Markt+Technik

21 TAGE

Gerald Carter
Richard Sharpe

SAMBA

Schritt für Schritt zum professionellen Einsatz von Samba

Datei- und Druckdienste in heterogenen Netzen nutzen

Automatisierung, Sicherheit und Troubleshooting

Die aktuellen Samba-Versionen, verschiedene Tools und Skripts sowie das komplette

SAMS

Inhaltsverzeichnis

Stichwortverzeichnis

- A -

- B -

- C -

- - -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- W -

- Z -

Vorwort

Häufig beschweren sich die Leute bei mir, dass Samba schwer zu verwalten ist. Normalerweise stimme ich ihnen schließlich bei, aber sie verstehen oft nicht, warum Samba schwer zu verwalten sein kann. Die Administration von Samba ist kompliziert, weil Samba komplizierte Dinge tut. Es lässt einen Unix-Rechner wie einen Windows-Datei- und Druck-Server erscheinen. Es erledigt diese Aufgabe so gut, dass viele Windows-Benutzer nicht einmal wissen, dass auf dem Server, auf dem sie ihre Dateien speichern oder drucken, Unix läuft. Um dieses Kunststück zu vollziehen, muss Samba das Verbindungsstück sein, das diese zwei sehr verschiedenen Systeme zusammenhält. Dabei darf es weder die Vermutungen zerstreuen, die Windows-Clients über ihre Server anstellen, noch die Robustheit und Sicherheit zerstören, die einem Unix-System innewohnt.

Um Samba meistern zu können, ist es hilfreich, einen Experten zur Seite zu haben, der die Unterschiede zwischen Windows und Unix und die Methoden, mit denen Samba diesen Abgrund überbrückt, grundlegend versteht. Gerald Carter und Richard Sharpe sind solche Experten. Wir machten Gerald zu einem Mitglied des Samba-Teams, nachdem er den frühen experimentellen Code dokumentiert hatte, der geschrieben wurde, um die Windows-NT-Domain-Protokolle zu unterstützen. Gerald ist Autor des Dokuments »Samba NT-Domain Frequently Asked Questions (FAQ)« und obwohl der Code jetzt stabil ist, verwaltet er ihn immer noch. Jeder, der mit dem Samba-2.0-Code gearbeitet hat, hat Gerald erlebt, wie er geduldig Frage auf Frage in den Samba-Mailing-Listen beantwortet und anderen dabei hilft, das notwendige Wissen zu erlangen, damit ihre Systeme genau so funktionieren, wie sie es wollen.

Richard kam 1995 zum Samba-Team und schrieb das Originaldokument »What is SMB?«, das für viele Leute die erste klare Beschreibung dafür war, wie Microsoft-Netzwerke tatsächlich funktionieren. Er ist der Hauptverantwortliche für die smbtar-Backup-Komponente von Samba und verwaltet die SMBlib-Client-Library-Routinen. Richard ist in den Samba-Mailing-Listen gut bekannt und hilft Benutzern mit den tiefsten und dunkelsten Geheimnissen der Samba-Konfiguration und -Benutzung.

Ich freue mich sehr, dass ich gebeten wurde, das Vorwort zu Geralds und Richards Samba-Buch, *Samba in 21 Tagen*, zu schreiben. Sie sind in jeder Hinsicht wahre Samba-Experten. Zwar schreibe ich eine ganze Menge des Samba-Codes, aber Gerald und Richard zeigen den Leuten, wie sie diesen Code verwenden, und dafür möchte ich ihnen danken. Obwohl der Samba-Code selbst kompliziert ist, ermöglicht Ihnen das Buch, das sie geschrieben haben, Samba zu installieren, um die Windows- und Unix-Welten miteinander zu verbinden und es einfach aussehen zu lassen. Dafür können Sie Gerald und Richard nur danken.

Jeremy Allison

Samba-Team

Einführung

Willkommen in der Welt von Samba! In den folgenden 24 Kapiteln werden Sie ein Tool kennen lernen, das Sie bei der Integration Ihrer Windows-Clients und Unix-Server unterstützen kann. Sie können die 24 Kapitel so bearbeiten, wie Sie es wollen. Dieses Buch bietet Ihnen die meiner Meinung nach wichtigsten Informationen, die Sie zu Samba haben sollten.

Jedes Kapitel enthält Beispielkonfigurationen, die Sie ausprobieren können, Diagramme, die Konzepte illustrieren, und praktische Beispiele. Am Ende jedes Kapitels finden Sie einen Abschnitt, in dem die Dinge zusammengefasst werden, die Sie gelernt haben. Danach finden Sie Antworten auf einige weit verbreitete Fragen.

Wenn Sie alle Kapitel durchgearbeitet haben, stellen Sie das Buch an einen Platz, der sich nah bei Ihrem Schreibtisch oder Computer befindet, damit Sie jederzeit darauf zurückgreifen können. Die Kapitel enthalten Informationen und Beispiele, die ich in meinem Job als Administrator mehrerer Samba-Server täglich verwende. Ich hoffe, dass Ihnen diese Informationen nützen.

Was ist Samba?

Der Brockhaus definiert Samba als »Samba, die (portug.), lateinamerikanischer Tanz im Zweiviertel-Takt«. Das ist nicht die Art von Samba, über die ich hier reden werde. Tatsächlich werden Sie an keinem Punkt in diesem Buch dazu aufgefordert, zu tanzen (es sei denn, Sie fühlen sich danach, wenn irgend etwas funktioniert!).

Samba ist die Implementierung eines *Server-Message-Block-(SMB-)*Protokollservers, der auf fast jeder existierenden Variante von Unix laufen kann. Microsoft-Clients können dieses Protokoll benutzen, um auf Dateien und Drucker zuzugreifen, die sich auf Ihrem lokalen Unix-Rechner befinden, als wäre dieser ein nativer Windows-Server.

Samba ist ein Open-Source-Projekt wie Linux (ein Unix-artiges Betriebssystem für PCs). Der in C geschriebene Source-Code steht Ihnen immer zur Verfügung, so dass Sie ihn untersuchen, testen oder ändern können. Und Samba ist frei!

Dies führt dazu, dass Samba auf immer mehr Servern installiert wird, um Freigaben für Microsoft-Clients bieten zu können, ohne einen Windows-NT-Server oder einen anderen SMB-Server installieren zu müssen.

Unternehmen jeder Größe können von Samba profitieren. Sogar meine Mutter benutzt Samba! (Ohne Witz!) Ihr kleines Büronetzwerk, das aus drei Rechnern besteht, benutzt einen abgespeckten PC, auf dem Linux und Samba laufen und der Heimatverzeichnisse, freigegebene Festplatten und Drucker zur Verfügung stellt. Sie benutzt diese einfache, kosteneffektive Lösung, und ich liefere den Support (wir wohnen etwa drei Stunden voneinander entfernt).

Warum sollten Sie Samba kennen lernen?

Es gibt viele Gründe, warum Sie etwas über Samba lernen sollten, egal, ob Sie ein Vollzeit-Netzwerkadministrator sind oder einfach nur zwei PCs zu Hause stehen haben:

- Samba ermöglicht es Ihnen, Dateien und Drucker auf Ihren Unix-Servern für Ihre Microsoft-Clients freizugeben.
- Samba bietet eine Methode für die Authentifizierung von PC-Logins.
- Samba ist ein leistungsfähiger, stabiler, kosteneffizienter Ersatz für einige PC-Server.
- Es gibt einen wachsenden Arbeitsmarkt für Samba-Administratoren.

Wer sollte dieses Buch benutzen?

Dieses Buch wurde für diejenigen geschrieben, die generelle Kenntnisse über Unix-basierte Systeme haben. Sie müssen kein Experte sein, sollten aber Grundkenntnisse über Dinge wie `ps`, `grep` und `kill` mitbringen. Außerdem wäre es hilfreich, das `make`-Utility und den `gcc`-Compiler bereits zu kennen.

Was befindet sich auf der CD-ROM?

Vorausgesetzt, dass Sie für Übungszwecke ein Unix-System laufen haben, enthält die CD-ROM alles, was Sie für die Kompilierung, Installation und Konfiguration von Samba brauchen. Sie finden hier die aktuellste Version von Samba sowohl in Source-Code-Form als auch als vorkompilierte Binärversion. Außerdem habe ich Ihnen weiterführende Dokumentation in Form von FAQs, einigen nützliche Perl-Skripts und zusätzliche GUI-Konfigurationstools zu Ihrem Vergnügen (und hoffentlich auch zu Ihrem Nutzen) beigefügt. Anhang D bzw. die Datei `liesmich.txt` auf der CD-ROM gibt detailliertere Auskunft zum Inhalt.

Wie dieses Buch aufgebaut ist

Dieses Buch ist in 21 Lektionen, sprich: Tage eingeteilt, die Sie mit ein wenig Ehrgeiz und Kondition innerhalb von drei Wochen durcharbeiten können. In jeder Woche wird eine Vielzahl von Themen behandelt, jede Lektion baut auf der vorhergehenden auf und die Komplexität der Themen nimmt von Woche zu Woche zu.

Jede Lektion widmet sich einem Thema bei der Arbeit mit Samba. Eine abschließende Zusammenfassung, Erklärungen der neu eingeführten Begriffe sowie ein kleines Quiz, mit dem Sie Ihr frisch erworbenes Wissen auch gleich testen können, runden jedes Kapitel ab.

Die Anhänge A - C enthalten außerdem weiterführende Informationen zu Samba, Anhang D enthält eine nähere Aufstellung der Inhalte der beiliegenden CD-ROM.

Folgende Lektionen werden Sie in *Samba in 21 Tagen* durchnehmen:

Tag 1 »Einführung in Samba«: Hier erfahren Sie, was Samba ist, auf welchen Plattformen es läuft, was seine Vorteile gegenüber traditionellen Lösungen wie FTP sind und warum es frei erhältlich ist.

Tag 2 »Windows-Netzwerke«: Erläutert werden das Microsoft-Netzwerk-Modell, das Protokoll NetBIOS mit seinen Verfahren zur Namensregistrierung und -auflösung, natürlich SMB sowie typische Konstellationen in Microsoft-Netzwerken.

Tag 3 »Wie bekomme ich den aktuellsten Source-Code?« Und wie kompilieren und installieren Sie die jeweilige Version? In diesem Kapitel erfahren Sie alles Nötige.

Tag 4 »Installation und Testen der Konfiguration«: Hier richten Sie beispielhaft einen Samba-Server ein - Sie editieren die `smb.conf`, geben Gruppen-, Home-Verzeichnisse und Drucker frei, setzen Start-Skripte ein und testen die Konfiguration.

Tag 5 »Die Datei smb.conf: Samba mitteilen, was es tun soll«. Hier werfen Sie einen eingehenden Blick in Sambas Konfigurationsdatei, die zwar berüchtigt ist für ihre vielen Parameter, aber eigentlich nicht besonders schwer zu verstehen ist.

Tag 6 »Sicherheitsmodi und Passwörter«: An die grundlegende Konfiguration schließt sich die Absicherung Ihrer Samba-Konfiguration an. Hierzu unterstützt Samba Verbindungs- und Benutzer-Authentifizierung sowie Passwortverschlüsselung.

Tag 7 »Dateifreigaben«: Dieses Kapitel widmet sich der Art und Weise, wie Sie mit Samba Dateien für Ihre Clients freigeben bzw. wie Sie Dateien über diese Freigaben erstellen und nutzen können. Sie erfahren, wie Samba die Verfügbarkeit und die autorisierte Nutzung von Freigaben prüft.

Tag 8 »Drucker«: Der Freigabe der Dateien folgt die der Drucker. Dieses Kapitel zeigt Ihnen die zur kontrollierten Freigabe notwendigen Parameter sowie Möglichkeiten der automatischen Treiberinstallation unter Windows.

Tag 9 »GUI-Administrationstools«: Hier lernen Sie einige Tools kennen, mit deren Hilfe Sie Samba auch komfortabler als über die Kommandozeile steuern können - etwa über Ihren Browser.

Tag 10 »Automatisierung auf Server-Seite«: Dieses Feature erlaubt nicht nur die Einrichtung eines den Bedürfnissen Ihrer Clients angepassten Verhaltens von Samba, sondern auch die Emulation mehrerer Samba-Server.

Tag 11 »Troubleshooting«: Dieses Kapitel hilft Ihnen bei einem der zweifellos schwierigsten Themen - der Fehlerbehebung. Sie lernen die wichtigsten Tools kennen, um sich dieser sehr kreativen Aufgabe mit Erfolg widmen zu können.

Tag 12 »Fallstudie: Einen NT-Datei- und Drucker-Server ersetzen«. Zwar unterstützt Samba nicht alle NT-Funktionen, doch kann es einen NT-Datei- und Drucker-Server effektiv ersetzen. Dieses Kapitel zeigt Ihnen an einem Beispiel, wie dies funktioniert.

Tag 13 »Unix (smbclient, smbfs, smbwrapper und andere Utilities)«: Dieses Kapitel behandelt einige Utilities, welche die Funktionalität von Samba u.a. in Richtung Datensicherung oder -archivierung erweitern.

Tag 14 »Windows 9x und Windows NT«: Dieses Kapitel setzt von der Windows-Seite aus an - wie greifen Sie mit den mit Windows gelieferten Samba-Clients auf Ihren Server zu? Behandelt wird sowohl Windows 9x als auch Windows NT.

Tag 15 »Andere SMB-Clients«: Der Einsatz eines Samba-Servers in heterogenen Netzen bedeutet auch die Kommunikation mit DOS- und Mac-Clients - in diesem Kapitel lernen Sie einige von ihnen kennen.

Tag 16 »Passwortsynchronisation«: Der Abgleich der Benutzerpasswörter zwischen Samba-Server und -Clients kann problematisch sein. Dieses Kapitel zeigt Ihnen, wie Sie auch diese berüchtigte Hürde nehmen.

Tag 17 »SSL«: Interagieren Ihre Samba-Server und -Clients über eine ungeschützte Verbindung wie das Internet, müssen Sie z.B. SSL implementieren, um die Kommunikation abhörsicher zu verschlüsseln.

Tag 18 »NetBIOS-Namen ohne Broadcasts auflösen«: Um NetBIOS-Namen über Subnetzgrenzen hinweg aufzulösen, ohne Ihr Netzwerk mit Broadcast-Sendungen zu belasten, sollten Sie einen WINS-Server installieren. Dieses Kapitel zeigt, wie das geht.

Tag 19 »Browsing in lokalen Subnetzen«: Mit Hilfe von Samba können Sie im lokalen Subnetz browsen. Hier erfahren Sie, welche Parameter Sie dazu in der smb.conf setzen müssen.

Tag 20 »Browsing in Netzwerken mit Routern«: Samba macht auch das Browsen über Subnetzgrenzen hinweg sowie in Netzwerken mit Routern möglich. Dabei unterstützt Samba verteilte Arbeitsgruppen und NT-Domänen.

Tag 21 »Windows 9x-Domänenkontrolle«: Dieses Samba-Feature bietet eine Methode zur Authentifizierung von Freigabeverbindungen und Netzwerk-Logins. Hier erfahren Sie die dafür notwendigen Details.

Konventionen in diesem Buch

Sie finden in diesem Buch folgende Icons:

Hinweise geben Ihnen Kommentare und Randbemerkungen zum jeweiligen Thema sowie vollständige Erklärungen bestimmter Konzepte. Außerdem finden Sie hier Tastenkombinationen und Hinweise für eine effektivere Shell-Programmierung.

Hier warne ich Sie davor, sich selbst unglücklich zu machen, und zeige Ihnen, wie Sie die kleinen Fallen in der Programmierung vermeiden.

Absätzen mit neuen Begriffen ist das Icon »Neuer Begriff« vorangestellt. Der neue Begriff ist *kursiv* dargestellt.

Am Ende jedes Kapitels finden Sie die praktischen Abschnitte »Zusammenfassung« und »Frage&Antwort«.

Zusätzlich werden Sie in diesem Buch auf verschiedene typografische Konventionen treffen:

- Befehle, Variablen, Verzeichnisse und Dateien sind in einer `speziellen Schriftart` dargestellt.
- Befehle u.ä., die Sie eingeben sollen, sind **fett** markiert.
- Platzhalter in Syntax-Beschreibungen sind in einer *`speziellen kursiven Schriftart`* dargestellt. Hier sollen Sie den Platzhalter durch den tatsächlichen Dateinamen, Parameter oder ein anderes Element, für das er steht, ersetzen.

Woche 1: Erste Schritte

Tag 1: Einführung in Samba

Dieses Kapitel ist ein Überblick darüber, was Samba ist und was Samba kann. Ich bekomme häufig E-Mails von allen möglichen Leuten, die mir Fragen stellen wie z.B. »Kann Samba http-Verbindungen zu einem Windows-NT-Server über die NIS+-Passwort-Datenbank auf einem Solaris-2.6-Rechner authentifizieren?« und »Wurde Samba auf Windows NT portiert?«. Dieses Kapitel beantwortet einige der Fragen, die Sie möglicherweise zu Samba haben, und gibt Ihnen Informationen zu den Funktionen und der Verfügbarkeit von Samba.

Was ist Samba?

Samba ist ein *Open-Source-Software- (OSS-)Projekt, das 1991* von Andrew Tridgell an der Australian National University in Canberra, Australien, entwickelt wurde. Während dieser Zeit war Andrew Doktorand im Informatiklabor und benutzte PC-NFS, um auf Dateien auf Sun Workstations zuzugreifen. Als er eine Beta-Version von eXcursion von Digital bekam, begann er, den Client zu testen. Zu seiner Enttäuschung waren die Server, mit denen sich der eXcursion-Client verbinden konnte, jedoch nur unter VMS und Ultrix verfügbar. Da er wie die meisten Informatikstudenten neugierig war, begann Andrew darüber nachzudenken, das Dateifreigabeprotokoll auf Nicht-Digital-Workstations zu implementieren.

Zu dieser Zeit hatte er noch nie von NetBIOS oder SMB gehört. Tatsächlich war dies sein erster Ausflug in die Netzwerk-Socket-Programmierung. Kurze Zeit später hatte Andrew eine einigermaßen funktionierende Verbindung zur Sun Workstation über den eXcursion-Client. Die erste Implementierung seines Servers bestand aus einer Menge hartcodierter »magischer« Werte, die einfach die Reaktionen des Ultrix-Servers reproduzierten. Nach einer Unterredung mit einem Mitarbeiter von Digital lernte Andrew erstmals das NetBIOS-Protokoll kennen. Erst zwei Jahre nach seiner ersten Implementierung sah er die Spezifikationen für das SMB-Protokoll und erfuhr, was all die »magischen« Werte darstellten.

Andrew gab seine erste Implementierung im Januar 1992 frei. Während der nächsten zwei Jahre benutzte er meistens einen X-Terminal und hatte keinen Bedarf, sein Projekt weiterzuentwickeln. Während dieser Zeit lernte Andrew auch Linux kennen. Als das allgemeine Interesse an seinem SMB-Server wuchs, nahm er die Entwicklung von Samba wieder auf und der Rest ist, wie man so schön sagt, Geschichte.

Eine häufig gestellte Frage lautet: »Woher stammt eigentlich der Name Samba?« Die Antwort ist ehrlich gesagt ziemlich einfach. Ursprünglich hieß Andrews Software SMBserver. Wegen rechtlicher Probleme musste der Name geändert werden. Eines der Wörter, die Andrew fand, als er `/usr/dict/words` nach *Kombinationen der* Buchstaben *s*, *m* und *b* durchforstete, war Samba - und das war's.

Sie können von der Samba-Hauptdistributionssite in Australien den Source-Code für die Programmfamilie herunterladen und kompilieren oder vorkompilierte Binärdateien für bestimmte Plattformen erhalten. Sie sollten eine der Mirror-Sites weltweit wählen, die Ihrem Standort am nächsten liegt. Eine komplette Liste der Mirror-Sites finden Sie auf der Samba-Webpage unter http://samba.org.

Samba ist, einfach beschrieben, eine Programmfamilie, die es Ihnen ermöglicht, auf Dateien und Drucker auf einem Nicht-Windows-Server zuzugreifen und dabei die mitgelieferte Unterstützung des Windows-Clients für den Zugriff auf entfernte Ressourcen zu benutzen.

Genauer gesagt ist Samba eine freie Implementierung eines *SMB- (Server-Message-Block-Protocol-)Servers*, der hauptsächlich für Unix-basierte Systeme entwickelt wurde. Samba wurde jedoch auch auf andere Plattformen portiert. Viele PC-Clients benutzen das SMB-Protokoll, *das* kürzlich *in CIFS (Common Internet File System*) umbenannt wurde, um auf entfernte Dateisysteme und Drucker zuzugreifen. Diese werden im Windows-Fachjargon als Freigaben oder Dienste bezeichnet. Für viele Unternehmen ist dies ausreichend, damit sich die Idee von Samba verkauft, und möglicherweise ist das auch alles, wofür sie Samba jemals verwenden. Aber einige andere Merkmale von Samba sind sozusagen das Tüpfelchen auf dem i. Nachfolgend finden Sie eine Auflistung der Dinge, die Samba noch kann:

- Samba dient als NetBIOS-Nameserver (siehe Kapitel 18, »NetBIOS-Namen ohne Broadcasts auflösen«).
- Samba unterstützt NetBIOS-Browsing und Browse-Master-Auswahl (siehe Kapitel 19, »Browsing in lokalen Subnetzen«, und 20, »Browsing in Netzwerken mit Routern«).
- Samba enthält zwei SMB-Clients, über die Unix-Rechner auf freigegebene Dateien oder Drucker auf anderen SMB-Servern, PCs oder anderen Samba-Servern zugreifen können (siehe Kapitel 13, »Unix (SMB-Client, SMBFs, SMB-Wrapper und verschiedene Utilities«).

- Samba bietet Erweiterungen, über die Unix-Rechner Dateien auf entfernten SMB-Freigaben, wie z.B. Windows-Datei-Servern, sichern können (siehe Anhang B, »Tipps und Tricks«).
- Samba bietet ein Kommandozeilen-Utility für eingeschränkte entfernte Administrationsfunktionen für Windows-NT- und Samba-Server (siehe Anhang A, »Experimentelle PDC-Unterstützung«).
- Samba kann als Domänen-Controller für Windows-9x- und Windows-für-Workgroups-Clients agieren. Derzeit finden Entwicklungen statt, um auch Windows-NT-3.51/4.0-Clients Domänenkontrolle zu bieten, die schon teilweise implementiert wurden (siehe Kapitel 21, »Windows-9x-Domänenkontrolle«, und Anhang A).

Ein Samba-Server kann auf verschiedene Weise in ein bestehendes Netzwerk integriert werden. Hier einige übliche Beispiele:

- Ersetzt einen Windows-NT-Datei-/Drucker-Server aufgrund der Lizenzierungskosten (siehe Kapitel 12, »Fallstudie: Einen NT-Datei-und-Drucker-Server ersetzen«).
- Bietet ein Gateway für die Synchronisierung von Unix- und Windows-NT-Passwörtern (siehe Kapitel 16, »Passwortsychronisierung«).
- Agiert als »Home-Verzeichnis«-Server, damit Unix-Home-Verzeichnisse und Windows-Home-Verzeichnisse in einem gemeinsamen Bereich existieren können (siehe Kapitel 6, »Sicherheitsmodi und Passwörter«).
- Agiert als Drucker-Gateway zwischen vernetzten PC- und Unix-Druckern (siehe Kapitel 7, »Dateifreigaben«).
- Ermöglicht Unix-Rechnern, auf NT-Dateien zuzugreifen.

Dies sind nur einige Beispiele. Ihrer Phantasie und Ihren Fähigkeiten als Programmierer sind hier keine Grenzen gesetzt.

Traditionelle Lösungen

Möglicherweise sind Sie mit einigen der früher benutzten Lösungen vertraut, über die Windows-Clients auf entfernte Dateien oder Drucker zugreifen konnten. In einem Unternehmen, in dem ich beschäftigt war, konnte nur über FTP auf entfernte Dateien zugegriffen werden. Entferntes Drucken hieß, die Datei auf eine Diskette zu speichern und zum entfernten Rechner zu laufen (wir nannten diese Methode liebevoll »Turnschuhnetzwerk«). Obwohl diese Art ihre Vorteile hat (d.h. sie ist einfach), brauchen Benutzer heutzutage entfernten Zugriff, der die Zusammenarbeit fördert.

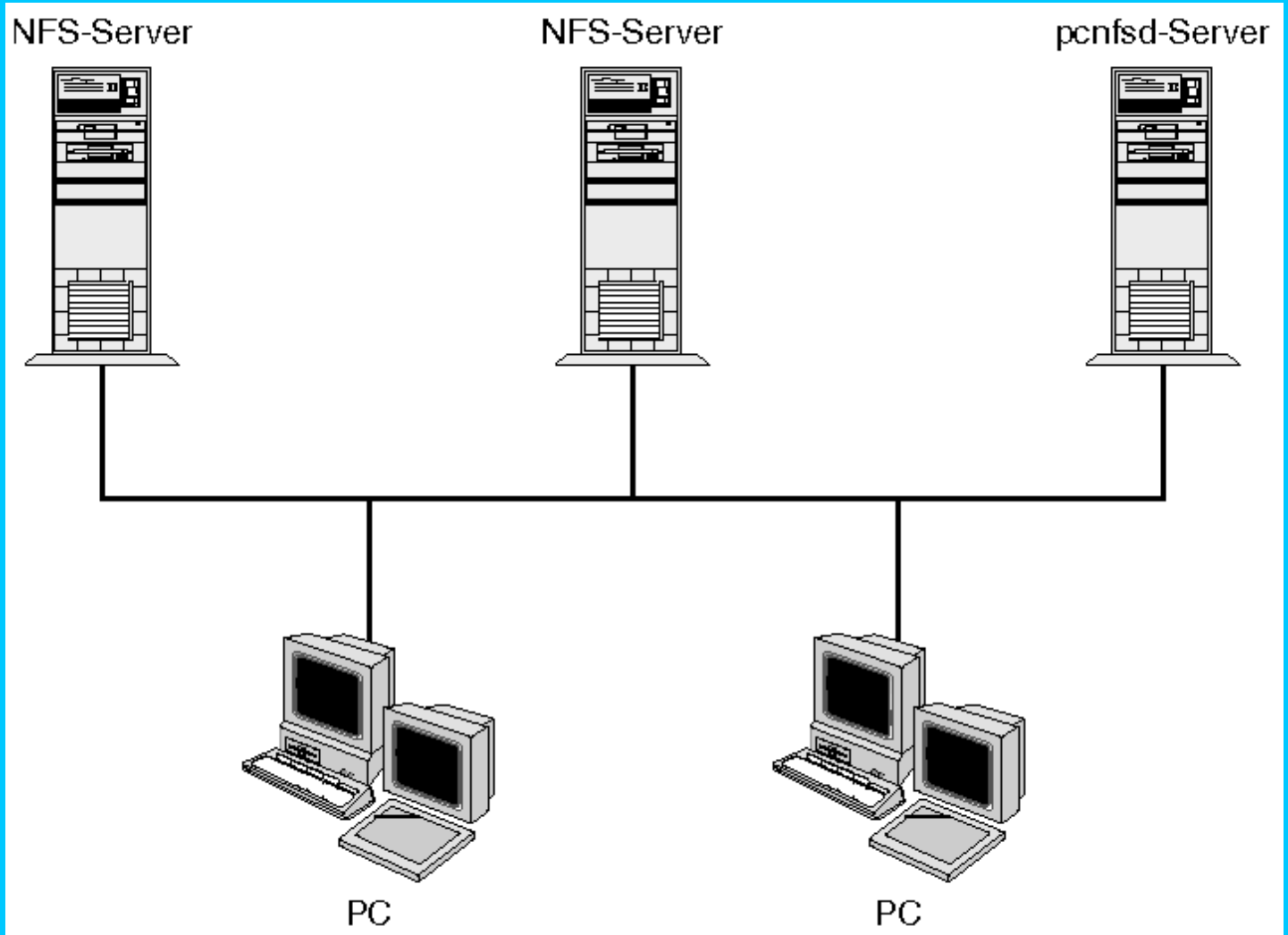
Während der späten achtziger Jahre begannen mehrere Unternehmen, einen Network-File-System-Client für PCs zu entwickeln, der als PC-NFS bekannt wurde. Viele Unternehmen begrüßten dies als Methode, PCs in existierende Unix-basierte Infrastrukturen zu integrieren.

Bestehende NFS-Server verlangten keine Software-Modifikationen, um PC-NFS-Clients zu bedienen. Die Authentifizierung der Clients wurde von einem Daemon durchgeführt, normalerweise `pcnfsd` oder `rpc.pcnfsd`, der auf einem Server lief. Die `Clients` übertrugen eine Authentifizierungsanfrage an den pcnfsd-Server. War diese erfolgreich, übertrug der Server die Unix-UID des Benutzers, die für alle weiteren NFS-Anfragen benutzt werden konnte.

Abbildung 1.1 zeigt, wie PCs mit existierenden NFS-Servern verbunden werden können. Es ist nur ein einziger Unix-Rechner notwendig, um die Authentifizierung der PC-Clients durchzuführen. In diesem Szenario werden die PCs sozusagen Bürger zweiter Klasse.

Die PC-NFS-Software führte im Wesentlichen zu zwei Problemen. Das erste bestand in den Einschränkungen des NFS-Protokoll selbst, was Dinge wie Datei-Locking betrifft. Datei-Locking war in Windows- und Unix-Dateisystemen (das ursprüngliche Ziel von NFS) auf verschiedene Weise implementiert. NFS benutzte für die Implementierung des Datei-Lockings einen separaten Prozess, der manchmal zu Problemen führte.

Abb. 1.1: Topologie mit existierenden NFS-Servern und einem einzigen pcnfsd-Server



Wenn z.B. der `lock-Daemon` abstürzte oder die Synchronisierung zwischen dem NFS- und dem `lock-Daemon` verloren ging, konnten die PC-Clients das notwendige Datei-Locking nicht mehr durchführen. Samba implementiert die SMB-Spezifikationen für das Datei-Locking und unterstützt auch opportunistisches Locking, gewöhnlich oplocks genannt. Silicon Graphics integrierte kürzlich oplock-Unterstützung in den Irix-Kernel, so dass die Dateiintegrität auch gesichert ist, wenn Samba und ein anderer Unix-Prozess gleichzeitig auf die Datei zugreifen. Das zweite Problem lag in der fehlenden nativen Unterstützung innerhalb der PC-Betriebssysteme begründet. Microsoft hat für den Aufbau seines Netzwerkmodells die Protokolle NetBIOS und SMB gewählt. Daher ist für den Client neben dem Betriebssystem selbst keine zusätzliche Software nötig, damit er sich mit einem Samba-Server verbinden kann. Für die Verbindung mit einem NFS-Server jedoch war zusätzliche Client-Software notwendig. Früher, als Festplatten noch kleiner waren, stellte diese zusätzliche Belastung oft ein Problem dar.

Die native SMB-Unterstützung bietet einige Vorteile. Einer ist, dass Benutzer sicher sein können, dass der für die Verbindung zu einem Samba-Server notwendige Client korrekt mit neuen Versionen des Betriebssystems funktionieren wird. Außerdem erhalten Benutzer die aktuellste Netzwerk-Client-Software, wenn sie ihr Betriebssystem aktualisieren. Bei der NFS-Software muss der NFS-Client separat aktualisiert werden, was manchmal zu Problemen führte.

Einer der größten Vorteile ist finanzieller Art. Die Lizenzgebühren für NFS-Clients summieren sich schnell, besonders in Unternehmen mit großen Netzwerken. Und, lassen Sie uns ehrlich sein, gibt es irgend jemanden, der es wirklich mag, die Client-Lizenzierung zu verwalten?

Wenn Sie Samba benutzen, vermeiden Sie all diese Probleme.

Heute sind in vielen Netzwerken bereits PC-Server in die Infrastruktur integriert (wenn sie nicht sogar andere Betriebssysteme komplett ersetzen). Hier besteht die traditionelle Lösung einfach darin, weitere Windows-NT-Server hinzuzufügen. Aber dies ist aus Kostengründen nicht für alle Unternehmen möglich. Mit einer Lösung - die Benutzung von Samba und einem der freien PC-Unix-Betriebssysteme wie Linux - können Administratoren den Vorteil günstiger Massen-Hardware nutzen und dabei trotzdem die Stabilität und Dienste bieten, die für ihre PC-Clients notwendig sind.

Auf welchen Plattformen läuft Samba?

Samba wird mit vollständigem Source-Code (in C geschrieben) verteilt und ist unter der GNU *General Public License* verfügbar. Wenn Sie nicht mit der GPL vertraut sind, finden Sie in diesem Kapitel später einen Abschnitt, der die Details beschreibt.

Da Samba mit Source-Code verteilt wird, kann es auf praktisch jeder Unix-Variante kompiliert werden, darunter

- Solaris2.x
- SunOS 4.x
- Ultrix
- Linux
- Irix
- HP-UX
- OSF1
- AIX
- NetBSD, FreeBSD und OpenBSD
- SCO
- DNIX

Lässt sich Samba unter Ihrer bestimmten Unix-Variante nicht kompilieren, senden Sie einen Bericht über Ihre Probleme an samba-bugs@samba.org, und dort wird jemand versuchen, eine Lösung für Ihr Problem zu finden. Eines der wundervollen Dinge in Bezug auf OSS-Projekte ist, dass Sie Ihre Probleme selbst beheben können. »Luke, use the source!« Sollten Sie tatsächlich Fehler im Source-Code finden oder beheben, senden Sie bitte ebenfalls eine Nachricht an bugs@samba.org.

Zusätzlich zu den wichtigsten Unix-Distributionen ist Samba auch auf die folgenden Betriebssysteme portiert worden:

- Amiga
- VMS
- OS/2
- MVS
- Stratus-VOS
- MPE/iX

Die GNU General Public License (GPL)

Viele OSS-Projekte werden unter der GNU GPL freigegeben, die von der Free Software Foundation entwickelt wurde. Der tragende Gedanke hinter der GPL ist, dass Software verteilt werden kann. Sie verlangt, dass

- alle Modifikationen an der Software unter der gleichen Lizenz wie die ursprüngliche Software freigegeben werden müssen (siehe Punkt 2 der GPL Version 2),
- der Source-Code, wenn er nicht mit den Binärdateien ausgeliefert wird, auf Anfrage verfügbar ist (siehe Punkt 3 der GPL Version 2).

Ohne die speziellen Abschnitte darzustellen, erklärt der folgende Auszug aus dem Vorwort der GPL die Absicht der Lizenz:

> Wenn wir von »freier« Software sprechen, meinen wir Freiheit, nicht den Preis. Unsere GPLs sollen sicherstellen, dass Sie die Freiheit haben, Kopien freier Software zu verteilen (und etwas für diesen Service zu berechnen, wenn Sie möchten), dass Sie den Source-Code erhalten bzw. ihn auf Wunsch verlangen können, dass Sie die Software ändern oder Teile davon für neue freie Software verwenden können und dass Sie wissen, dass Sie all diese Dinge tun können.
>
> Um Ihre Rechte zu schützen, müssen wir Einschränkungen machen, die es jedem verbieten, Ihnen diese Rechte zu verweigern oder Sie aufzufordern, auf diese Rechte zu verzichten. Diese Einschränkungen führen zu einer bestimmten Verantwortung Ihrerseits, wenn Sie Kopien der Software verteilen oder sie modifizieren.

Die Idee hinter dem Kopieren und der Verteilung der freien Software erklärt sich weitestgehend selbst. Die Abschnitte über die Modifizierung von Software, die unter der GPL lizenziert ist, sollten vielleicht etwas näher erklärt werden:

> (2b) Sie müssen dafür sorgen, dass jede von Ihnen verbreitete oder veröffentlichte Arbeit, die ganz oder teilweise von dem Programm oder Teilen davon abgeleitet ist oder solche enthält, Dritten gegenüber als Ganzes unter den Bedingungen dieser Lizenz ohne Lizenzgebühren zur Verfügung gestellt wird.

Abschnitt 2, Teil 2b der Version 2 der GPL bestimmt, dass jede Arbeit, die von anderer, unter GPL lizenzierter, Software abgeleitet ist, ebenfalls GPL sein muss. Dadurch garantiert die Lizenz, dass Software, die unter der GPL freigegeben wird, immer frei sein wird, auch wenn

Änderungen in existierenden Code eingefügt werden oder die Software in ein anderes Projekt eingebettet wird.

Die FSF bietet weitere Informationen über die Philosophie hinter freier Software und der GPL auf ihrer Website unter http://www.gnu.org/philosophy.

Samba wird unter der Version 2 der GPL verteilt. Den vollständigen Text dieser Lizenz finden Sie in der Samba-Distribution auf der beiliegenden CD-ROM. Weitere Informationen über die Free Software Foundation und das GNU-Projekt finden Sie unter http://www.gnu.org.

Zusammenfassung

Samba hat viele Merkmale. Hier die drei wichtigsten:

- Samba ist frei.
- Samba ermöglicht es Unix-Rechnern, Dateien und Drucker für PC-Clients zur Verfügung zu stellen.
- Samba unterstützt komplett NetBIOS-Browsing und -Namensauflösungen.

In den folgenden Kapiteln werden Sie lernen, wie Sie Samba installieren und die besonderen Merkmale konfigurieren.

Frage & Antwort

F. Kann Samba meinen Windows-NT-Server (nicht PDC) vollständig ersetzen?

. Samba kann Dateien und Drucker für Windows freigeben, ebenso wie ein Windows-NT-Server dies tun würde. Ich muss hier einige Besonderheiten erwähnen. Kapitel 12 beinhaltet eine Fallstudie für den Ersatz eines Windows-NT-Servers durch einen Linux-Rechner mit Samba.

F. Kann Samba meinen Windows-NT-PDC ersetzen?

. Nicht vollständig. Die Funktionen zur Domänenkontrolle von Samba für einen Windows-9x-Client sind stabil und vollständig, daher würde er den Unterschied wahrscheinlich niemals merken. Die Unterstützung für die Domänenkontrolle für Windows-NT-Clients befindet sich noch in der Entwicklung. Derzeit ist die Implementierung so weit gereift, dass ein Windows-NT-Client auf eine durch Samba kontrollierte Domäne zugreifen kann, aber die Domänenkontrolle umfasst wesentlich mehr als das. Die Funktionen von Samba zur Domänenkontrolle werden in den Kapiteln 21 und 22 dargestellt.

F. Ich habe das Betriebssystem meines Servers nicht in der Liste der Plattformen gefunden, auf denen Samba kompiliert werden kann. Wie kann ich herausfinden, ob sich Samba auf meinem Server kompilieren lässt?

. Wenn ich Ihre Unix-Version (oder andere Plattform) nicht aufgelistet habe, können Sie zwei Dinge tun. Versuchen Sie zunächst, Samba zu kompilieren, um zu sehen, ob sich irgend etwas nicht kompilieren oder konfigurieren lässt. Die zweite Möglichkeit besteht darin, einen Blick in den Samba-FAQ auf der Samba-Homepage zu werfen, wo Sie eine aktuelle Liste der unterstützten Plattformen und spezifische Informationen zu Ihrem Betriebssystem finden können.

F. Wie kann ich Andrew eine Pizza spendieren, um die Weiterentwicklung von Samba zu unterstützen?

. Es gibt verschiedene Möglichkeiten, wenn Sie Nahrung für den Geist (alias Pizza) zu Entwicklungszwecken spenden möchten. Hier die möglichen Methoden aus dem Samba-FAQ:

- Methode 1: Rufen Sie Ihre lokale Niederlassung einer internationalen Pizza-Kette an und fragen Sie, ob die Gutscheine auch in anderen Ländern eingelöst werden können. Pizza Hut bietet einen derartigen Service. Auf diese Art und Weise kam die gesamte Linux Users Group in Canberra auf Kosten einer Person in den USA in den Genuss eines Pizza-Abendessens.
- Methode 2: Rufen Sie eine Pizzeria in Canberra an, geben Sie der Person am anderen Ende Ihre Kreditkartennummer für einen Gutschein über einen bestimmten Betrag und sagen Sie ihr, dass Andrew den Gutschein abholen wird. (Vergessen Sie nicht, ihm dies vorher mitzuteilen.) Eine nette Person aus Deutschland hat das gemacht.
- Methode 3: Besorgen Sie sich einen Gutschein von Ihrem Lieblingsitaliener, der keine Niederlassungen im Ausland hat, und senden Sie ihn an Andrew. Er ist zwar völlig nutzlos, aber Andrew kann ihn neben den Gutschein an die Wand hängen, den er bereits aus Deutschland hat.
- Methode 4: Senden Sie ihm Ihre Lieblingspizza per Luftpost. Wahrscheinlich wird sie bei den Zollbehörden stecken bleiben oder von hungrigen Spürhunden auseinandergenommen werden, aber immerhin war es eine noble Geste.

Tag 2: Windows-Netzwerke

Bevor ich damit beginne, die Konfiguration des Innenlebens von Samba detailliert darzustellen, gebe ich Ihnen in diesem Kapitel einen Überblick über die Konzepte, die Grundlage dieser Details sind. Die folgenden Abschnitte erklären die grundlegenden Gedanken hinter dem Vernetzungsmodell für die Microsoft-Betriebssysteme, von Windows für Workgroups bis Windows NT. Dieses Kapitel ist sozusagen ein Blick aus 3.000 Meter Höhe auf die von Microsoft- (und anderen) Clients verwendeten Netzwerkprotokolle, aber Sie erhalten hier die nötigen Grundkenntnisse, um Ihre Samba-Server in späteren Kapiteln besser verstehen und konfigurieren zu können.

NetBIOS-Überblick

Wenn Sie bereits seit einiger Zeit mit Intel-basierten Rechnern arbeiten, sind Sie sicher mit dem Start-BIOS-Bildschirm vertraut, der beim Neustart des Rechners erscheint. *BIOS* steht für *Basic Input/Output System*. Mitte der achtziger Jahre wurden die Konzepte des Computer-BIOS auf die noch neuen Netzwerkkonzepte ausgeweitet. Das Resultat war das *Application Programming Interface (API)* eines Programmierers, das *NetBIOS* genannt wurde, was für *Network Basic Input/Output System* steht. Die allgemein akzeptierte Definition des *NetBIOS-API* zu jener Zeit war das *IBM PC Network Technical Reference Manual*, das im September 1994 von IBM veröffentlicht wurde.

Kurz danach, 1985, entwickelte IBM ein Netzwerkprotokoll, um das NetBIOS-API einzukapseln und zu erweitern. Das resultierende Protokoll wurde *NetBEUI* genannt, was für *NetBIOS Extended User Interface* steht. *NetBEUI* ist optimiert für kleine LANs, kann aber nicht geroutet werden.

Zusätzlich zu NetBEUI kann NetBIOS über IPX laufen. Dies würde die Koexistenz von Novell-Netzwerken und Microsoft-Netzwerken ermöglichen.

1987 standardisierte die *Internet Engineering Task Force (IETF)* in den *Requests for Comments (RFCs) 1001* und *1002* das Interface über TCP und UDP. NetBIOS über TCP/IP wird gewöhnlich NBT genannt.

Die RFCs 1001 und 1002 definieren drei Dienste, die von NetBIOS über TCP/IP zur Verfügung gestellt werden sollten:

- Name Service
- Session Service
- Datagram Service

Name Service

Ein *Name Service* bildet die Brücke zwischen der Art und Weise, wie Computer andere Rechner sehen und lokalisieren und wie Menschen sie

sehen und lokalisieren. Menschen können sich in der Regel Namen besser merken als Zahlen, aber Zahlen belegen weniger Speicherplatz im Arbeitsspeicher eines Computers. Also wird ein Mechanismus benötigt, der einen Namen in eine Zahl auflöst. Vom Prinzip her gleicht dies dem Heraussuchen einer Telefonnummer aus einem Telefonbuch.

Genau das tut der Name Service. Wenn Sie mit TCP/IP-Netzwerken vertraut sind, werden Sie wissen, dass jeder IP-Host einen Hostnamen hat, z.B. `bilbo,` und eine entsprechende IP-Adresse, z.B. 192.168.1.73. Wenn Sie an **einem anderen** Rechner `telnet bilbo` eingeben, muss der lokale Rechner, an dem Sie sich befinden, die entsprechende IP-Adresse zum Hostnamen `bilbo` finden, bevor er Pakete an den entfernten Rechner senden kann. Die Standardmethode für die Auflösung eines Internet-Hostnamens in eine IP-Adresse besteht entweder darin, eine lokale `/etc/hosts`-Datei oder einen Domain Name Server zu befragen. In beiden Fällen wird letztlich in einer Art »Telefonbuch« nachgeschaut, welche IP-Adresse zu dem Namen gehört.

Der NetBIOS-Name-Service, der in den RFCs 1001 und 1002 definiert ist, bietet dem Client die gleiche Art von Dienst wie DNS. Sie können den Hauptunterschied zwischen den beiden besser verstehen, wenn ich Ihnen erkläre, was ein NetBIOS-Name ist.

NetBIOS-Namen

NetBIOS-Namen existieren in einem flachen Namensbereich und bestehen aus 16 alphanumerischen Zeichen (z.B. a bis z, A bis Z und 0 bis 9) zusätzlich zu den folgenden:

- !
- @
- #
- $
- %
- ^
- &
- (
-)
- -
- '
- {
- }
- .
- ~

Nur 15 der Zeichen werden für die Benennung des Client-Rechners verwendet. Das sechzehnte Byte ist eine Zahl von 0x00 bis 0xFF, die den Ressourcentyp des Namens kennzeichnet. Die Namen können entweder exklusiv einem Benutzer oder Rechner gehören oder, bei Gruppennamen, gemeinsam benutzt werden. Der NetBIOS-Name MYMACHINE<00> z.B. ist ein eindeutiger Name, der den Rechner MYMACHINE kennzeichnet. Der Name MYDOMAIN<1e> ist ein Gruppenname, der von Browsing-Clients benutzt wird, um einen Master-Browser zu wählen. Wenn Sie etwas Hintergrundwissen über TCP/IP haben, ist es vielleicht hilfreich, sich vorzustellen, dass die NetBIOS-Ressourcentypen den TCP- und UDP-Portnummern entsprechen. In den Tabellen 2.1 und 2.2 sind alle aktuellen NetBIOS-Ressourcenbezeichnungen mit einer kurzen Erklärung aufgelistet.

Tabelle 2.1: Eindeutige NetBIOS-Ressourcentypen

Ressourcenbyte	Beschreibung
<00>	NetBIOS-Name, mit dem der Freigabename der Workstation bezeichnet wird
<03>	Messenger-Service-Name, der für das Übertragen und Empfangen von Nachrichten benutzt wird
<06>	RAS-Servername
<1b>	Domain-Masterbrowser-Name, der von einem Rechner benutzt wird, um den Primary Domain Controller einer Domain zu kontaktieren
<1f>	NetDDE-Dienst
<20>	Server-Dienstname, um Zugriffspunkte für freigegebene Dateien zu bieten
<21>	RAS-Client
<be>	Network Monitor Agent
<bf>	Network Monitor Utility

Tabelle 2.2: NetBIOS Gruppen-Ressourcentypen

Ressourcenbyte	Beschreibung
<1c>	Ein Domain-Gruppenname, der vom Domain Controller registriert wird und eine Liste von Computern enthält, die diesen Namen registriert haben
<1d>	Masterbrowser-Name, der vom Client benutzt wird, um auf den Masterbrowser zuzugreifen (möglicherweise ein lokaler Masterbrowser)
<1e>	Normaler Gruppenname, der bei der Wahl von Browse-Mastern benutzt wird
<20>	Internet-Gruppe, die benutzt wird, um eine Gruppe von Rechnern für administrative Zwecke zu identifizieren
MSBROWSE	Wird an den Domainnamen angehängt und per Broadcast übertragen, um anderen Masterbrowsern die Domain bekannt zu geben

Den praktischen Einsatz dieser Ressourcenbezeichnungen können Sie in der Ausgabe des Befehls `nbtstat.exe` sehen, die Sie in Listing 2.1 finden. Die Ausgabe wurde von einem Windows-95-OSR2-Rechner erstellt, und der untersuchte Rechner war eine Windows-NT-4.0-Workstation.

Listing 2.1: Ausgabe des Befehls nbtstat.exe

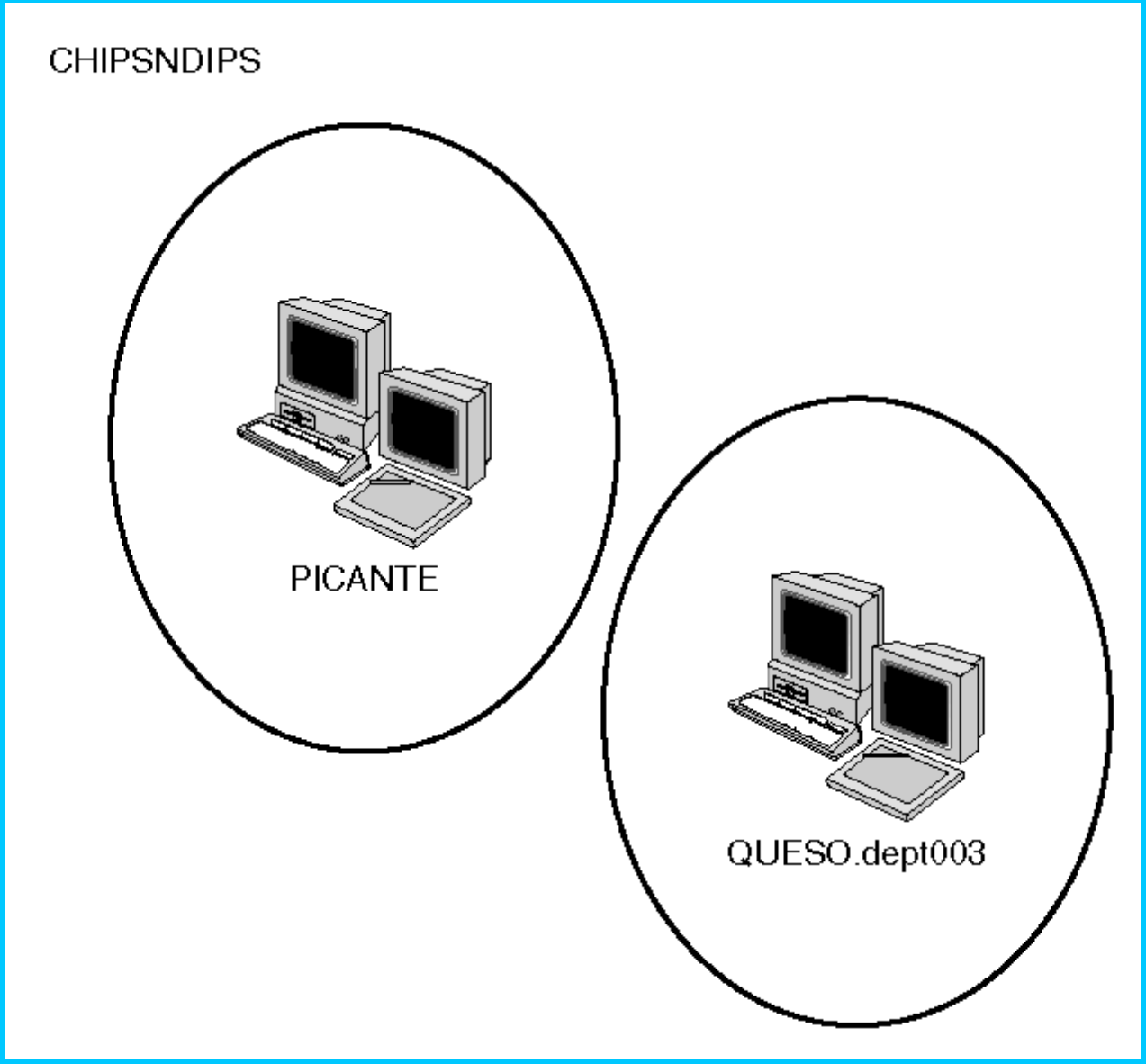
```
C:\users\jerry>nbtstat -a picante
    NetBIOS-Namentabelle des Remote-Computers
Name                Typ      Status
PICANTE        <00>   UNIQUE   REGISTRIERT
SALSA          <00>   GROUP    REGISTRIERT
PICANTE        <03>   UNIQUE   REGISTRIERT
JERRYC         <03>   UNIQUE   REGISTRIERT
PICANTE        <20>   UNIQUE   REGISTRIERT
SALSA          <1E>   GROUP    REGISTRIERT
SALSA          <1D>   UNIQUE   REGISTRIERT
..__MSBROWSE_.  <01>   GROUP    REGISTRIERT
MAC Address = 00-60-97-40-CD-18
```

Die Angaben in der Spalte `Type` und das Ressourcenbyte helfen Ihnen zu bestimmen, was der Name darstellt. So können Sie z.B. erkennen, dass der entfernte Rechner den Namen `PICANTE` hat, weil es ein eindeutiger Name mit der Bezeichnung `<00>` ist. Wenn Sie noch einmal einen Blick in Tabelle 2.1 werfen, sehen Sie diese Schlussfolgerung bestätigt.

Ich habe vorher erwähnt, dass NetBIOS-Namen in einem flachen Namensbereich existieren. Trotzdem ist es möglich, im gleichen logischen Subnetz einen Namensbereich von einem anderen zu unterscheiden, indem man benutzt, was als *NetBIOS-Scope* (NetBIOS-Bereich) bezeichnet wird. Der NetBIOS-Bereich ist ein Zeichenstring, der zusammen mit dem NetBIOS-Namen nicht länger als 256 Zeichen sein kann.

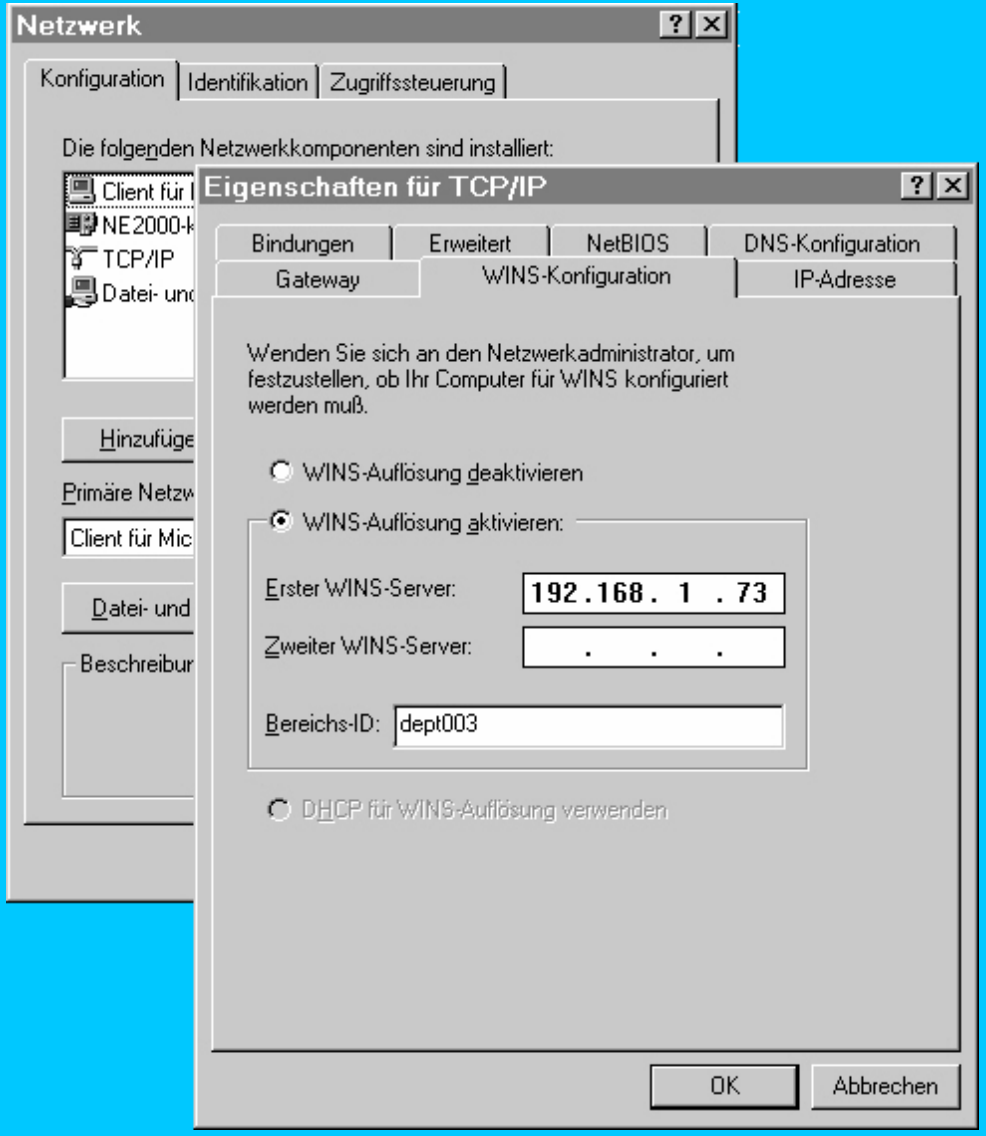
Der NetBIOS-Scope bietet keine hierarchische Organisation für den NetBIOS-Namensbereich. Er isoliert einfach nur die Namen in einem Bereich von Rechnern in einem anderen. Sie sollten diesen Bereich in der Regel einfach leer lassen, wenn Sie nicht einen besonderen Grund dafür haben, einen NetBIOS-Bereich einzurichten. Abbildung 2.1 zeigt die Einrichtung eines NetBIOS-Bereichs für die Segmentierung von Rechnern in einem Netzwerk. Die Rechner `PICANTE` und `QUESO` gehören beide der gleichen Arbeitsgruppe an, haben aber unterschiedliche Bereichs-IDs, die in der Abbildung an die Rechnernamen angehängt sind. `PICANTE` hat eine Bereichs-ID von »« und `QUESO` eine von »dept003«. Keiner von beiden kann mit dem anderen kommunizieren, solange beide eine unterschiedliche Bereichs-ID haben.

Abb. 2.1: Über die NetBIOS-Bereichs-ID NetBIOS-Clients segmentieren



Sie können unter Windows 95 die Bereichs-ID einrichten, indem Sie in der Systemsteuerung das Netzwerkkontrollfeld öffnen und sich unter den TCP/IP-Eigenschaften für Ihre Netzwerkadapterkarte die Registerkarte *WINS-Konfiguration* ansehen. Abbildung 2.2 zeigt das Feld *Bereichs-ID* mit einer Einstellung von »dept003«.

Abb. 2.2: Einstellung der Bereichs-ID im Netzwerkkontrollfeld von Windows 95



Registrierung und Auflösung

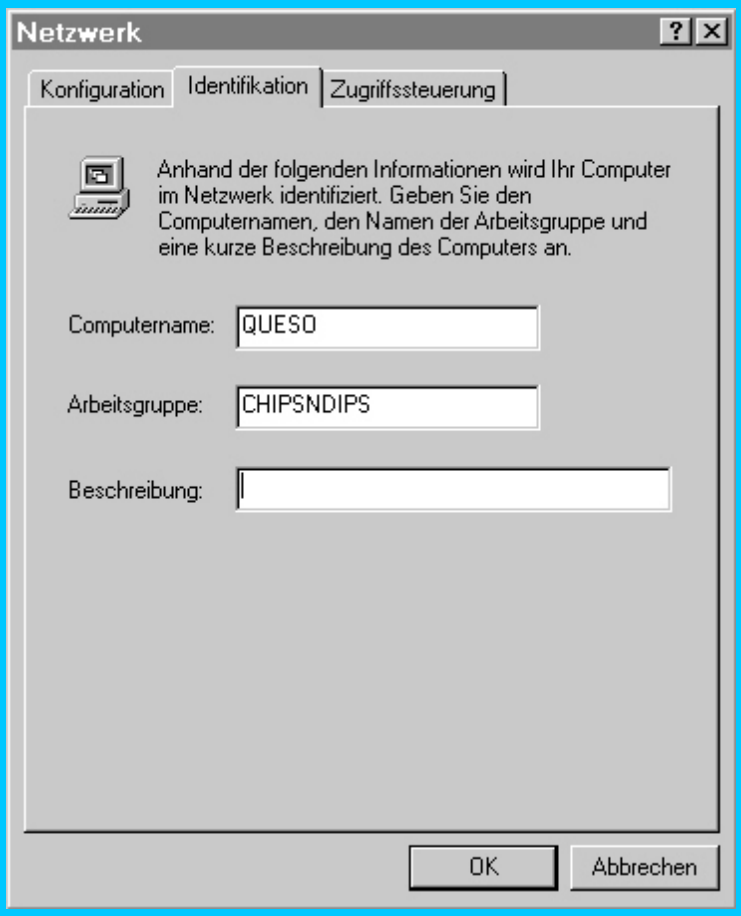
Der NetBIOS-Name-Service bietet eine Methode für die Registrierung der Namen von Client-Rechnern und deren Auflösung. Einen Namen *registrieren* heißt, dass der Client-Rechner ein erfolgreiches Namensangebot gemacht und den Namen für sich erhalten hat, so dass alle zukünftigen Anfragen an diesen Namen an den Client gehen werden. Während des Bootvorgangs versucht der Client, den Namen zu registrieren, indem er eine Anfrage an den Name Service sendet, welcher entweder kollaborativ oder singulär sein kann. Dieser Unterschied spielt jetzt keine Rolle.

Ist der gewünschte Name bereits an einen anderen Client vergeben, antwortet dieser, dass er den Namen bereits besitzt und ihn behalten möchte. Wenn Client Nr. 1 auf eine Anfrage von Client Nr. 2 antwortet, in der dieser den Namen von Client Nr. 1 verlangt, redet man auch davon, dass der erste Client seinen Namen verteidigt, d.h. dass er seinen Namen behalten möchte. NetBIOS-Clients werden ihre Namen immer behalten wollen, solange sie funktionieren. Die Verteidigung von registrierten Namen ist ein wichtiger Aspekt, da sie eine Methode bietet, Hosts zu entdecken, die inaktiv sind, aber den Besitz des Namens nicht offiziell aufgegeben haben.

Es ist außerdem hilfreich, sich klarzumachen, dass NetBIOS-Namen in eine bestimmte IP-Adresse aufgelöst werden, wenn ein NetBIOS-Interface über TCP/IP läuft. Stellen Sie sich vor, dass ein Client, der über DHCP erfolgreich eine IP-Adresse erhalten hat, einen Namen angibt, dann ohne Vorwarnung abstürzt und den Namen nicht offiziell freigeben kann. Wenn der Rechner neu gestartet wird, versucht er, den Namen wiederzubekommen, obwohl er potentiell eine andere IP-Adresse haben kann. Zwar können IP-Adressen dynamisch zugewiesen werden, aber NetBIOS-Namen sind hartcodiert. Wenn in einem Windows-Client der Client für Microsoft-Netzwerke installiert

ist, können Sie den Namen finden, indem Sie in den Netzwerkeigenschaften (über *Systemsteuerung* und *Netzwerk*) die Registerkarte *Identifikation* ansehen (siehe Abbildung 2.3).

Abb. 2.3: Die Einstellungen für den Computernamen und die Arbeitsgruppe des NetBIOS-Rechners in den Netzwerkeigenschaften unter Windows 95



Dies mag etwas verwirrend erscheinen, wenn Sie aus der Standard-TCP/IP-Welt kommen, in der ein Host-Name im DNS oder einer Host-Datei registriert ist und der Zusammenprall von Namen durch gute Administration vermieden wird. Ich gebe zu, dass es etwas seltsam ist, wenn ein Rechner Namen dynamisch registriert, indem er verlangt, dass sie auf dem lokalen Host statisch aufgezeichnet werden.

Nach der Registrierung von NetBIOS-Namen müssen jetzt andere Clients in der Lage sein, den Namen in eine Netzwerkadresse aufzulösen. In diesem Fall ist das eine IP-Adresse.

Es gibt zwei Methoden für die Registrierung und Auflösung von Namen: *Broadcast* und *Point-to-Point*. Broadcast-Registrierung oder -Auflösung heißt, dass die Anfrage an alle Hosts im gleichen logischen Subnetz mit der gleichen Bereichs-ID übertragen wird. Router können für die Übertragung von Broadcasts konfiguriert werden, aber dies ist in der Regel keine gute Idee, da dann zu viel Datenverkehr erzeugt wird.

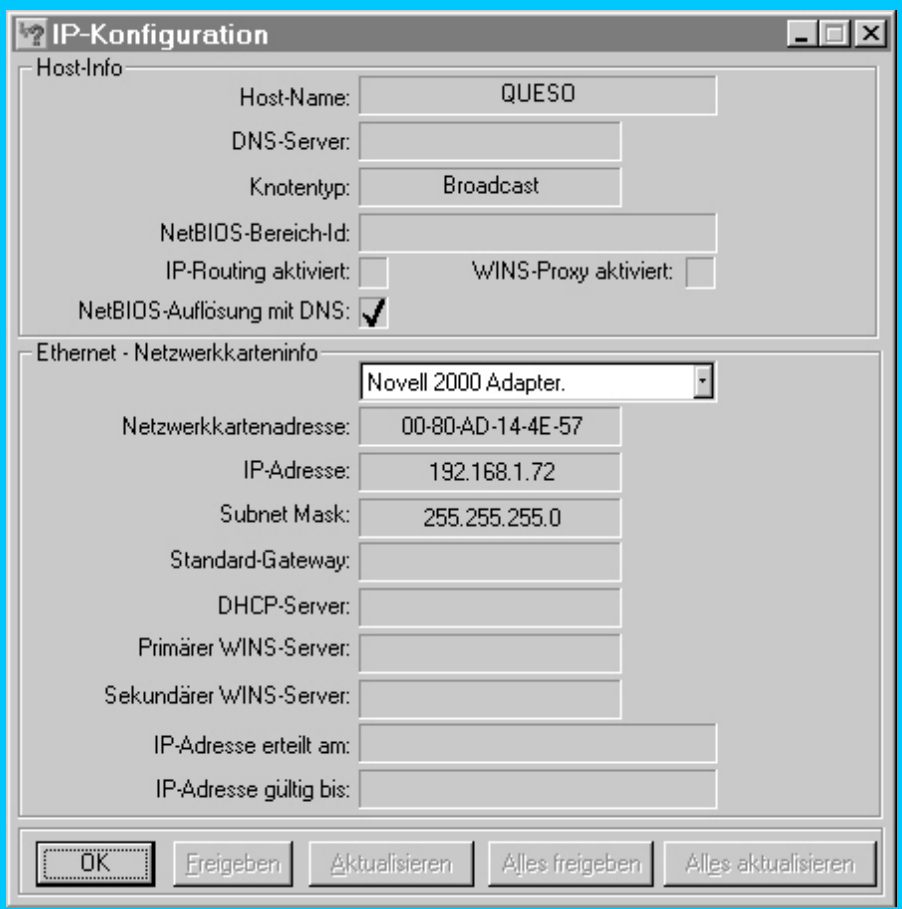
Point-to-Point-Registrierung und -Auflösung wird über einen *NetBIOS-Nameserver* (NBNS) durchgeführt. Der NBNS wird ebenfalls in den RFCs 1001 und 1002 definiert. Die NetBIOS-RFCs erlauben dem NBNS, in Hinsicht auf die Verwaltung und Authentifizierung von Namen unterschiedliche Grade von Verantwortung zu akzeptieren. Microsofts Implementierung des NBNS, der *Windows Internet Name Service (WINS)*, agiert als Agent und ermöglicht Clients sowohl die Registrierung von Namen als auch deren Auflösung in eine IP-Adresse. Samba kann auch als WINS-Server arbeiten (siehe Kapitel 18, »NetBIOS-Namen ohne Broadcasts auflösen«).

Die Broadcast-Point-to-Point-Registrierung und -Auflösung erstellt die folgende Taxonomie:

- *b-Knoten* - b-Knoten verwenden für die Registrierung und Auflösung von Namen nur Broadcasts.
- *p-Knoten* - p-Knoten übertragen Anfragen zur Registrierung und Auflösung per Unicast an den NetBIOS-Nameserver (NBNS).
- *m-Knoten* - m-Knoten führen die Namensregistrierung per Broadcast durch. Ist die Registrierung erfolgreich, wird der NBNS über Unicast-Pakete informiert. Zunächst wird versucht, Namen per Broadcast aufzulösen, und nur bei Fehlern wird der NBNS per Unicast kontaktiert. In der Praxis wird dieser Knotentyp nur selten benutzt.
- *h-Knoten* - h-Knoten wurden von Microsoft eingeführt, nachdem die RFCs 1001 und 1002 geschrieben waren; sie sind die logische Umkehrung der m-Knoten. h-Knoten, auch *Hybridknoten* genannt, benutzen für alle Namensregistrierungen und -auflösungen den NBNS und gehen nur dann zu Broadcasts über, wenn der NBNS nicht erreichbar ist bzw. die Anfragen nicht erfolgreich sind.

In der Praxis sind die meisten NetBIOS-Clients entweder b-Knoten, d.h., sie benutzen keinen WINS-Server, oder h-Knoten, was heißt, dass sie einen WINS-Server benutzen. Sie können sich den aktuellen Knotentyp ansehen, indem Sie das Windows-IP-Konfigurations-Utility (`winipcfg.exe`) starten und auf die Schaltfläche *Weitere Info* klicken. Abbildung 2.4. zeigt die weiteren Informationen, die dann angezeigt werden.

Abb. 2.4: Den NetBIOS-Knotentyp in der Windows IP-Konfiguration ansehen



Session-Service

Der NetBIOS-Session-Service bietet eine Methode zur Unterstützung zuverlässiger Verbindungen und zum Austausch von Nachrichten zwischen zwei Clients. Für jede einzelne Verbindung hat der Sender einen *Anrufernamen* (*calling name*) und der Empfänger einen *angerufenen Namen (called name)*. Ich erwähne diese Begriffe hier nur, weil sie manchmal in Fehlermeldungen der Windows-Clients sowie in einigen der in Samba enthaltenen Utilities auftauchen.

Vielleicht hilft es Ihnen, sich die Beziehung zwischen dem Nameserver und dem Session-Service als eine Telefonverbindung vorzustellen. Bevor Sie einen Anruf von Ihrem Freund Peter erhalten können, brauchen Sie eine Telefonnummer, die Ihnen von Ihrem Telekommunikationsanbieter zugewiesen wird (natürlich brauchen Sie zuerst auch noch ein Telefon). Dies entspricht der Registrierung Ihres NetBIOS-Namens. Wenn sowohl Peter als auch Sie registrierte Telefonnummern haben, kann Peter Sie anrufen.

Seine Seite der Verbindung ist der NetBIOS-Anrufername, und Ihre Seite der Verbindung ist der angerufene Name. Die Telefonkabel, die Sie verbinden, stellen den Session-Service dar, der die Daten (Ihre Konversation) überträgt. Jeder Anruf ist eine eigenständige Sitzung nur zwischen Ihnen und Peter (es sei denn, Ihr Telefon wird abgehört). Wenn Sie umziehen, ohne Peter davon zu unterrichten, und er versucht, Sie anzurufen, wird er eine fremde Person am anderen Ende haben, die den Anruf nicht entgegennimmt. Daher die NetBIOS-Meldung »Not listening on called name«. Sie erhalten diese Meldung, wenn Sie versuchen, eine Sitzung mit einem anderen Host einzugehen, aber den falschen Namen benutzen.

Dieser Dienst bietet NetBIOS-Anwendungen eine weitere Funktion: die Erkennung abgebrochener Sitzungen mit anderen NetBIOS-Clients. Ich erwähne dies mehr der Vollständigkeit halber, weniger als technische Einzelheit. Wenn Sie weitere Informationen über den NetBIOS-Session-Service wünschen, holen Sie sich eine Kopie der RFCs 1001 und 1002.

Datagram-Service

Der Datagram-Service ist sozusagen die Umkehrung des Session-Service. Er bietet einen verbindungslos-orientierten Dienst für die Übertragung von Paketen an einen bestimmten Host oder eine bestimmte Gruppe von Hosts innerhalb einer Arbeitsgruppe (Unicast/Multicast) oder die Übertragung an alle Hosts in einem logischen Subnetz oder einem NetBIOS-Bereich (Broadcast). Der Datagram-Service ermöglicht z.B. Dinge wie das Auffinden des aktuellen Browse-Masters für eine Arbeitsgruppe oder Domäne durch Übertragen der Anfrage per Multicast an den Arbeitsgruppennamen.

Vielleicht haben Sie den Begriff »Mailslots« *anstelle* des Begriffs »Datagramme« gehört. Mailslots ist der Microsoft-Name für NetBIOS-Datagramme.

Lassen Sie uns noch einmal zu dem Beispiel mit Ihrem Freund Peter zurückkehren. Nehmen wir an, Sie haben kein Telefon mehr. Wie könnte Peter Sie kontaktieren, vorausgesetzt, Sie wohnen nicht in seiner Nähe? Wahrscheinlich würde Peter Ihnen einen Brief schicken. Nehmen wir jetzt an, Sie haben Peter einen Brief geschickt, bevor Sie seinen erhalten haben. In diesem Beispiel ist Ihre Adresse der NetBIOS-Name und die Briefe sind Pakete. Peter sendet Ihnen jetzt einen weiteren Brief als Antwort auf Ihren Brief, aber Sie haben seinen ersten Brief nie erhalten. Diese Art von unzuverlässiger Auslieferung - ich kritisiere hier nicht die Post, es ist nur ein Beispiel - ist billiger als ein Anruf über große Entfernungen und weniger Bandbreiten-intensiv für das Telefonnetzwerk (d.h., es ist gar nicht bandbreiten-intensiv).

Um das Beispiel mit dem Postdienst fortzusetzen, nehmen wir an, Sie wollen Weihnachtskarten an all Ihre Freunde senden. Sie schicken 25 Karten los und alle kommen sicher an, aber Sie haben keinerlei Garantie, in welcher Reihenfolge Ihre Freunde die Karten erhalten. Haben Sie schon einmal eines dieser Gewinnspiele gesehen, in denen es heißt: »Der erste Anrufer gewinnt den Hauptpreis!« Das ist in etwa das Gleiche. Wenn das Unternehmen diese Aufforderung versendet, kann es nicht kontrollieren, wer sie zuerst bekommt, auch wenn die Briefe zu unterschiedlichen Zeiten versendet werden. Jetzt haben Sie eine Vorstellung des Datagram-Service.

CIFS-Überblick

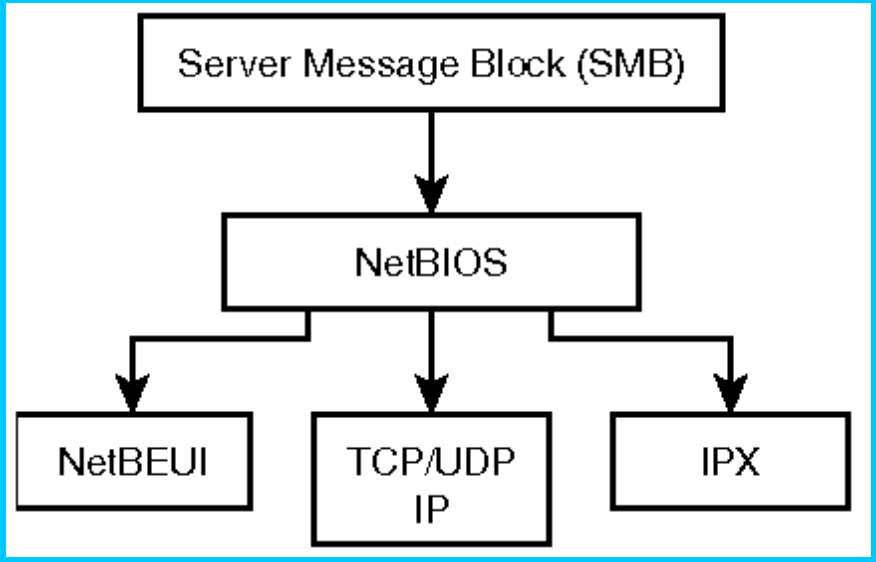
CIFS steht für *Common Internet File System* und wird oft austauschbar mit der Abkürzung *SMB* verwendet. Tatsächlich ist CIFS einfach die nächste Generation des SMB-Protokolls. Wie auch immer Sie es nennen, lassen Sie sich nicht verwirren.

SMB wurde 1987 erstmals durch ein gemeinsames Dokument von Microsoft und Intel namens »*Microsoft Network/OpenNET-File sharing Protocol*« definiert. Seitdem wurde das Protokoll vielen Änderungen unterzogen, und Paul Leach und Dilip Naik dokumentieren seine letzte Version im CIFS-1.0-Spezifizierungsentwurf. Da es sich um einen Entwurf handelt, ist die tatsächliche Dokumentation in der Praxis einfach »wie NT es macht«.

SMB über NBT

Das Server-Message-Block-Protokoll läuft über NetBIOS (siehe Abbildung 2.5). Wie in den vorherigen Abschnitten dargestellt, kann NetBIOS über NetBEUI, IPX/SPX und TCP/IP laufen. Samba implementiert nur SMB über TCP/IP.

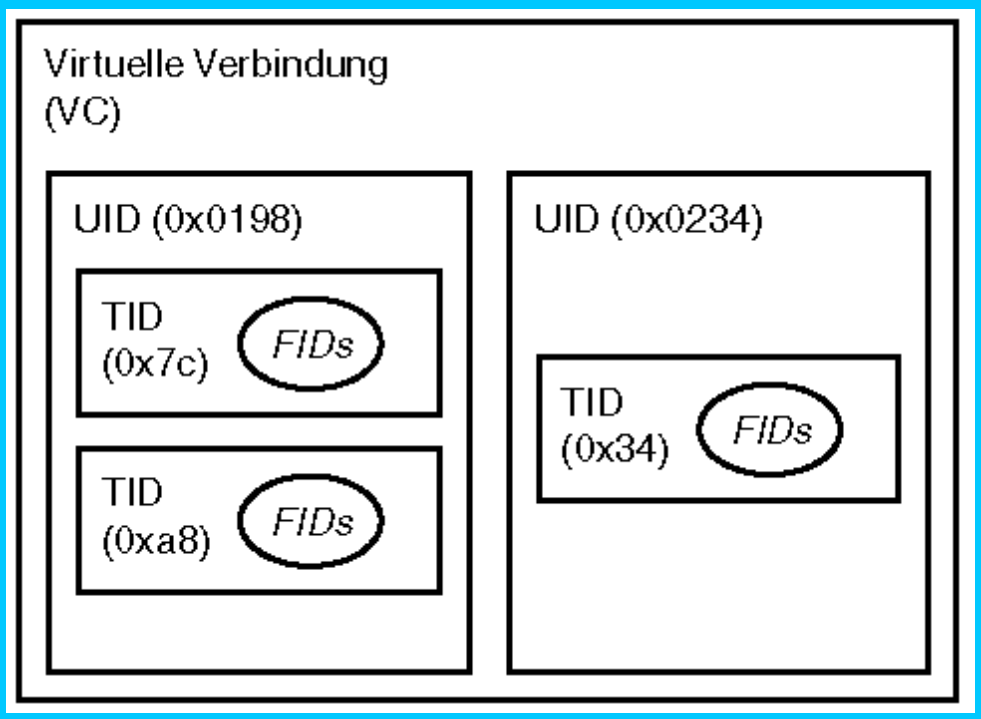
Abb. 2.5: SMB über NetBIOS und mögliche Transportschichtprotokolle



Verbindungsorientiert

SMB ist ein verbindungsorientiertes Protokoll, d.h. dass alle SMB-Pakete innerhalb einer virtuellen Verbindung (VC-Virtual Circuit) zwischen dem Client und dem Server in der Reihenfolge übertragen werden, in der sie versendet wurden. Wird die virtuelle Verbindung abgebrochen, werden alle Informationen, die sich in dieser Verbindung befinden, ungültig. Betrachten Sie die folgenden vier Dinge, die eine SMB-Verbindungsumgebung ausmachen: eine virtuelle Verbindung zwischen dem Client und dem Server, eine Sitzungs-UID, eine ID für den Ressourcen-Verzeichnisbaum (Tree ID - TID) und ein Dateiidentifikator (File Identifier - FID) (siehe Abbildung 2.6). Die UID wird in Schritt 2 des SMB-Protokollüberblicks näher beschrieben. Die TID stellt die freigegebene Ressource dar und die FID die individuellen Dateizugriffe.

Abb. 2.6: SMB-Verbindungsumgebung



Wird ein Teil der Verbindungsumgebung ungültig, wird alles, was sich in diesem Teil der Umgebung befindet, ebenfalls ungültig. Das SMB-Protokoll verwaltet keine Informationen über frühere Verbindungen, d.h., wird eine Verbindung beendet, muss alles wieder von Null neu aufgebaut werden. Eine Sitzungs-UID, die ein Client aus einer früheren Verbindung erhalten hat, gilt nicht für eine neue Verbindung. Alle TIDs und FIDs müssen ebenfalls erneut geöffnet werden. Alle Zugriffe auf Dateien, die sich in der Netzwerkressource befinden, werden nutzlos, wenn die Verbindung beendet wird.

Darum stürzen PC-Anwendungen, die auf einer Netzwerkfreigabe laufen, ab, wenn die Verbindung beendet wird, während die Anwendung läuft. Die Anwendung versucht, die Datei-Handles zu benutzen, die vorher ausgegeben wurden. Die meisten Clients versuchen still und leise, die beendete Sitzung wiederherzustellen, damit die Freigabeverbindung verfügbar ist, aber alle von den Anwendungen verwendeten Datei-Handles sind inzwischen ungültig.

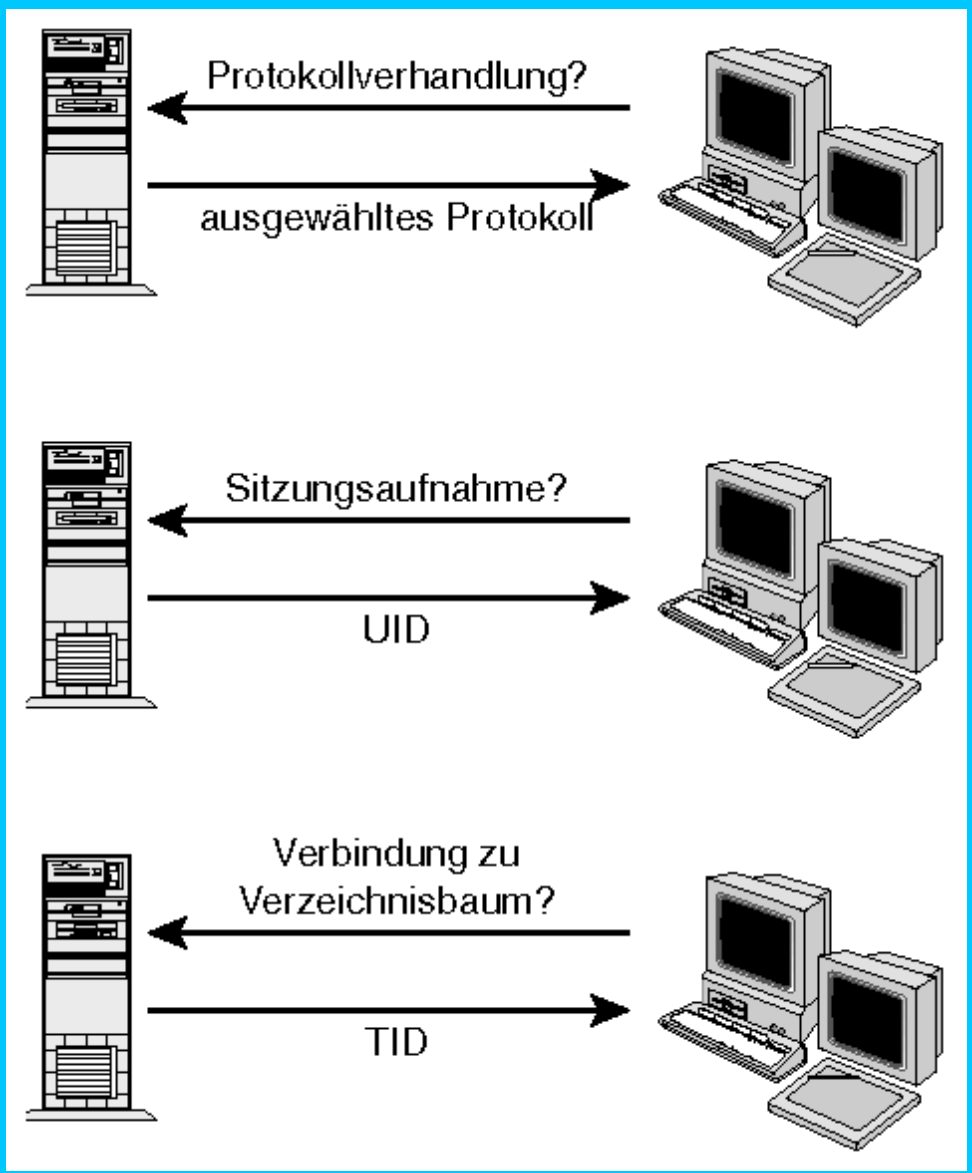
Abbildung 2.6 beschreibt die Umgebung für eine fiktive Verbindung. Nehmen wir jetzt an, dass die UID (0x0198) aus irgendeinem Grund beendet wird. Beide TIDs, x0x7c und 0xa8, werden ungültig, da sie Teil der UID-Umgebung sind. Nehmen wir nun an, die gesamte virtuelle Verbindung wird beendet. In diesem Fall werden alle UIDs, TIDs und FIDs ungültig, da sie alle innerhalb der virtuellen Verbindung bestehen.

Protokollüberblick

Wenn sich ein Benutzer mit einer Freigabe auf einem SMB-Server verbindet, z.B. über das Dialogfeld *Netzlaufwerk verbinden* im Windows-Explorer oder durch Browsing in der Netzwerkumgebung, werden drei Startphasen durchlaufen, bevor er auf die Inhalte der Freigabe zugreifen kann. Die beschriebenen Schritte beziehen sich auf Dateifreigaben, aber die gleichen Anfragen und Reaktionen gelten auch für die Verbindung zu entfernten Druckern.

Abbildung 2.7 zeigt die notwendigen Schritte für den Aufbau einer SMB-Verbindung zwischen einem Client und einem Server. Die chronologische Reihenfolge des Diagramms ist von oben nach unten und von links nach rechts.

Abb. 2.7: Eine SMB-Verbindung zu einer freigegebenen Ressource aufbauen



1. *Negprot* - Der erste Schritt für einen Verbindungsaufbau zu einer SMB-Freigabe, der oben in Abbildung 2.7 dargestellt ist, besteht in der Verhandlung über den zu verwendenden Protokolldialekt (Protocol Negotiation). Der Client sendet in einem Anfragepaket eine Textauflistung aller SMB-Dialekte, mit denen er vertraut ist. Jeder SMB-Dialekt (oder Protokoll-Level) unterstützt bestimmte Funktionen. Der aktuellste CIFS-Spezifizierungsentwurf listet die verschiedenen Befehle auf, die von jedem Level unterstützt werden. Der Server wählt das höchste Protokoll aus, das ihm bekannt ist, und gibt dem Client die entsprechende Protokollnummer aus der Liste zurück.
2. *sessetup&X* - Nachdem sich der Client und der Server auf einen gemeinsamen Protokolldialekt geeinigt haben, besteht der nächste Schritt darin, eine Sitzungsverbindung zwischen den beiden Rechnern einzurichten. Dieser Schritt wird im mittleren Teil der Abbildung 2.7 dargestellt. Der Client sendet eine Sitzungsanfrage, die einen Benutzernamen und einen Identitätsnachweis, z.B. ein Passwort, enthält. Der Server versucht, den anfragenden Benutzer zu authentifizieren. Ist dies erfolgreich, sendet der Server eine Sitzungs-UID an

den Client zurück. Diese UID ist für jede Sitzung einmalig und hat keinen Bezug zu der Server-internen Darstellung von Benutzern. Die Sitzungs-UID ist also nicht das Gleiche wie die Unix-UID oder die NT-Sicherheits-ID eines Benutzers. In Kapitel 6, »Sicherheitsmodi und Passwörter«, werden Sie lernen, dass diese Beschreibung nicht ganz akkurat ist. Das SMB-Protokoll kann in verschiedenen Sicherheitsmodi arbeiten, welche die Schritte 2 und 3 beeinflussen. Für unsere Zwecke jetzt ist die Beschreibung jedoch gut genug.

3. *tcon&X* - Der letzte notwendige Schritt, bevor ein Zugriff auf die Dateien in einer entfernten Freigabe gewährt wird, besteht in der erfolgreichen Verbindung zum Verzeichnisbaum in der freigegebenen Ressource und ist im unteren Teil von Abbildung 2.7 dargestellt. Der Client sendet eine Verbindungsanfrage für den Verzeichnisbaum mit der UID, die während der erfolgreichen Sitzungsaufnahme vom Server ausgegeben wurde. Hat der Benutzer ausreichende Zugriffsrechte auf die Freigabe, erhält der Client eine Verzeichnisbaumverbindungs-ID (TID). Die TID wird für alle Zugriffsanfragen auf Dateien verwendet, die sich in der Ressource befinden, die durch die TID bezeichnet wird.

Sind diese Schritte ausgeführt, kann der Benutzer Operationen in der Freigabe ausführen, z.B. ein Dokument drucken oder eine Datei, die sich in der Freigabe befindet, zum Bearbeiten öffnen.

Windows-Netzwerkmodelle

Peer-to-Peer-Netzwerke

Bevor es PCs gab, bestand das Netzwerkmodell aus einem zentralen Server und Terminals, auf die Benutzer zugreifen konnten. Diese Terminals hatten selbst keine Rechenleistung. Sie boten dem Benutzer nur eine interaktive Ansicht des Servers.

Mit der Invasion der PCs in den späten achtziger Jahren begannen Benutzer, ihre Dateien auf der lokalen Festplatte in ihren PCs zu speichern. Dies stellte jedoch ein Problem für die gemeinsame Nutzung von Dateien dar: etwas, das unproblematisch war, solange sich jeder von seinem Terminal in den gleichen Rechner (d.h. Mainframe) einloggte. Die Leute wollten ihre Dateien lokal speichern, damit sie auch während eines Server-Ausfalls (etwas, das sie nicht kontrollieren konnten) auf ihre Dateien zugreifen konnten, aber trotzdem anderen Benutzern den Zugriff auf die Dateien von ihren eigenen Rechnern ermöglichen. Dieses PC-zentrierte verteilte Modell wurde *Peer-to-Peer-Netzwerk* genannt, da alle Rechner gleichzeitig Client und Server sein konnten.

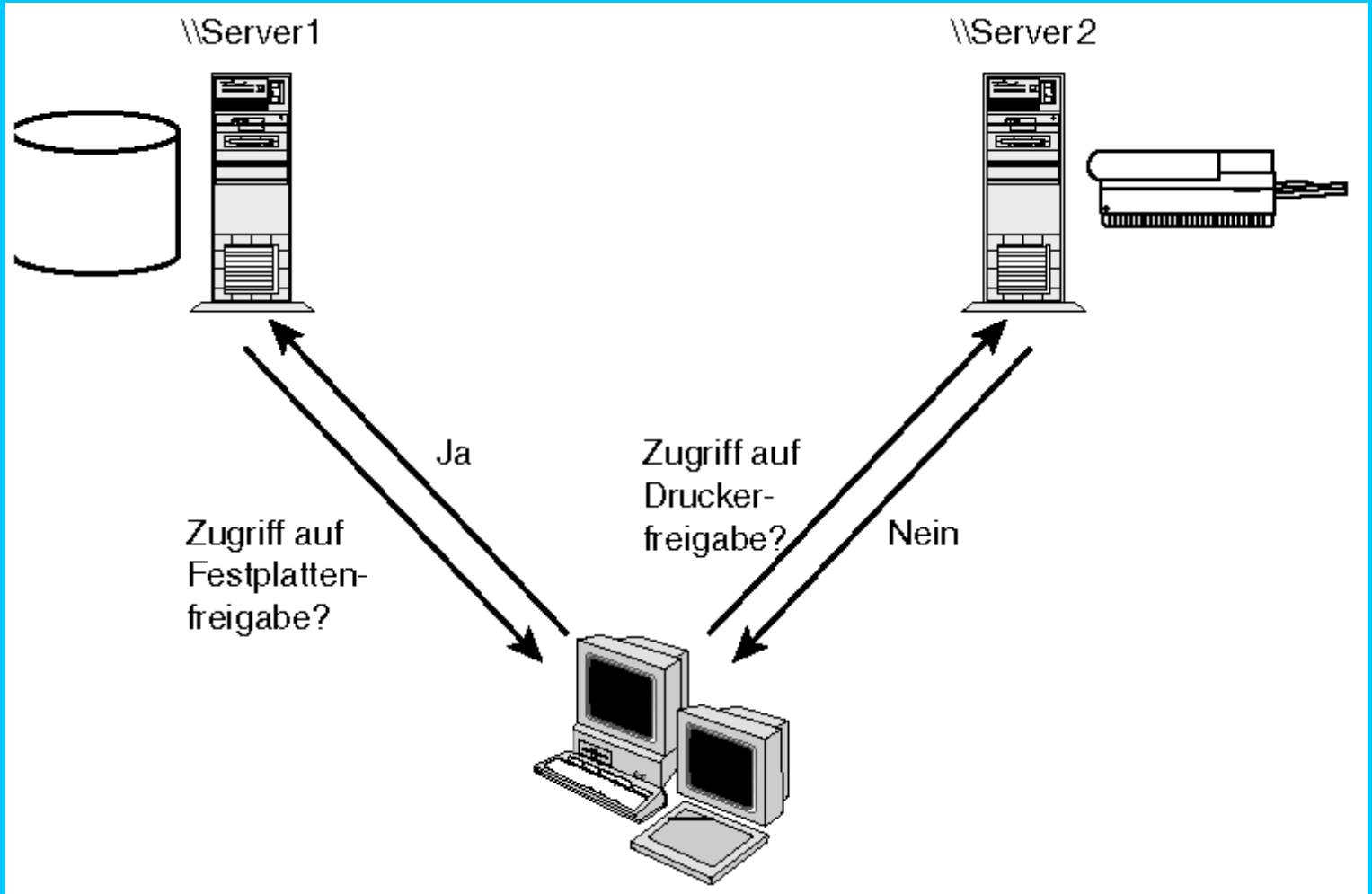
Arbeitsgruppen

Die Idee einer Arbeitsgruppe geht Hand in Hand mit dem Konzept des Peer-to-Peer-Netzwerks. Eine Arbeitsgruppe ist eine Gruppe von Leuten, die sich die Verantwortung teilen, ein gemeinsames Ziel zu erreichen. Jeder muss seinen Teil übernehmen. Eine Computer-Arbeitsgruppe ist nichts anderes. Wie Sie sehen werden, kann das Konzept einer Computer-Arbeitsgruppe in zwei Zusammenhängen benutzt werden.

Das erste Konzept einer Arbeitsgruppe ist eine administrative Gruppe von Rechnern, die Benutzer- und Gruppen-Account-Informationen nicht teilen. Erinnern Sie sich noch an Schritt 2 des SMB-Protokollüberblicks? Hier sendet der Client einen Benutzernamen und einen Beweis seiner Identität. Die Frage ist dann: »Wer authentifiziert diese Anfrage?« Jeder Rechner hat eine separate und lokale Kopie einer Account-Datenbank. Daher findet jegliche Authentifizierung lokal statt. Denken Sie daran, dass dieses Modell *Peer-to-Peer-Netzwerk* genannt wird, weil alle Rechner prinzipiell gleichwertig sind. Jeder PC hat die Möglichkeit, Datei- und Druckerfreigaben zur Verfügung zu stellen und Zugriffsanfragen zu authentifizieren. Diese Gleichwertigkeit heißt natürlich nicht, dass alle Rechner diese Aufgaben gleich gut durchführen. Bedenken Sie außerdem, dass der Konfigurationsaufwand bei dieser Variante gegebenenfalls sehr hoch sein kann.

Abbildung 2.8 verdeutlicht die Idee des Arbeitsgruppen-Authentifizierungsmodells. Der Client, der unten dargestellt ist, versucht, auf die Festplattenfreigabe auf SERVER1 zuzugreifen. SERVER1 ist alleine verantwortlich für die Authentifizierung der Sitzungsanfrage mit Hilfe seiner lokalen Account-Datenbank, was immer dies auch sein mag. Wenn der Client versucht, auf die Druckerfreigabe auf SERVER2 zuzugreifen, ist dieser Server dafür verantwortlich, die Verbindung zu authentifizieren. Das Ergebnis ist völlig unabhängig von dem Ergebnis der Verbindung zu SERVER1. Jeder Server hat eine lokale und unabhängige Account-Datenbank, die mit der des anderen Rechners nichts zu tun hat.

Abb. 2.8: Beispiel für ein Arbeitsgruppen-Netzwerkmodell



Das Konzept einer Computer-Arbeitsgruppe kommt auch beim Browsing zur Anwendung, das in den Kapiteln 19 und 20 näher dargestellt wird. Die Motivation für das Netzwerk-Browsing liegt in der Art und Weise, in der Ressourcen im Netzwerk erscheinen und aus dem Netzwerk verschwinden, während Hosts gestartet und beendet werden. Im Gegensatz zu einem zentralen Netzwerkmodell, wie z.B. die Lösung mit Mainframe und Terminals, ist es viel schwieriger, eine große Anzahl von Hosts zu überblicken, die je nach Willen des PC-Besitzers im Netzwerk an- und abgemeldet werden. Browsing ermöglicht es Benutzern, die aktuell verfügbaren Server und Ressourcen dynamisch anzusehen. In diesem Zusammenhang sind eine Domäne und eine Arbeitsgruppe äquivalent.

Domänen

Eine *Domäne* ist einer Arbeitsgruppe ähnlich, mit einer großen Ausnahme. In einer Domäne gibt es einen zentralen Authentifizierungsserver, der die Benutzer- und Gruppen-Accounts der Domäne verwaltet. Der Zugriff auf die Ressourcen in der Domäne wird durch den Domänen-Controller authentifiziert, egal auf welchem Rechner sich die Ressourcen befinden. Es handelt sich hierbei immer noch um ein Peer-to-Peer-Netzwerk, da alle Rechner die Möglichkeit haben, Datei- und Druckerfreigaben zur Verfügung zu stellen und die notwendige Authentifizierung durchzuführen. Der Unterschied besteht darin, dass die Authentifizierung mit Hilfe einer entfernten Account-Datenbank durchgeführt wird, die sich auf dem Domänen-Controller befindet.

Domänen vermeiden die Masse von Passwörtern, die notwendig ist, wenn jeder Rechner seine eigene lokale Account-Datenbank hat. Die Lösung versorgte Benutzer mit einem Account, der auf Wunsch Zugriff zu allen Ressourcen ermöglichen konnte.

Abb. 2.9: Beispiel für eine Domäne

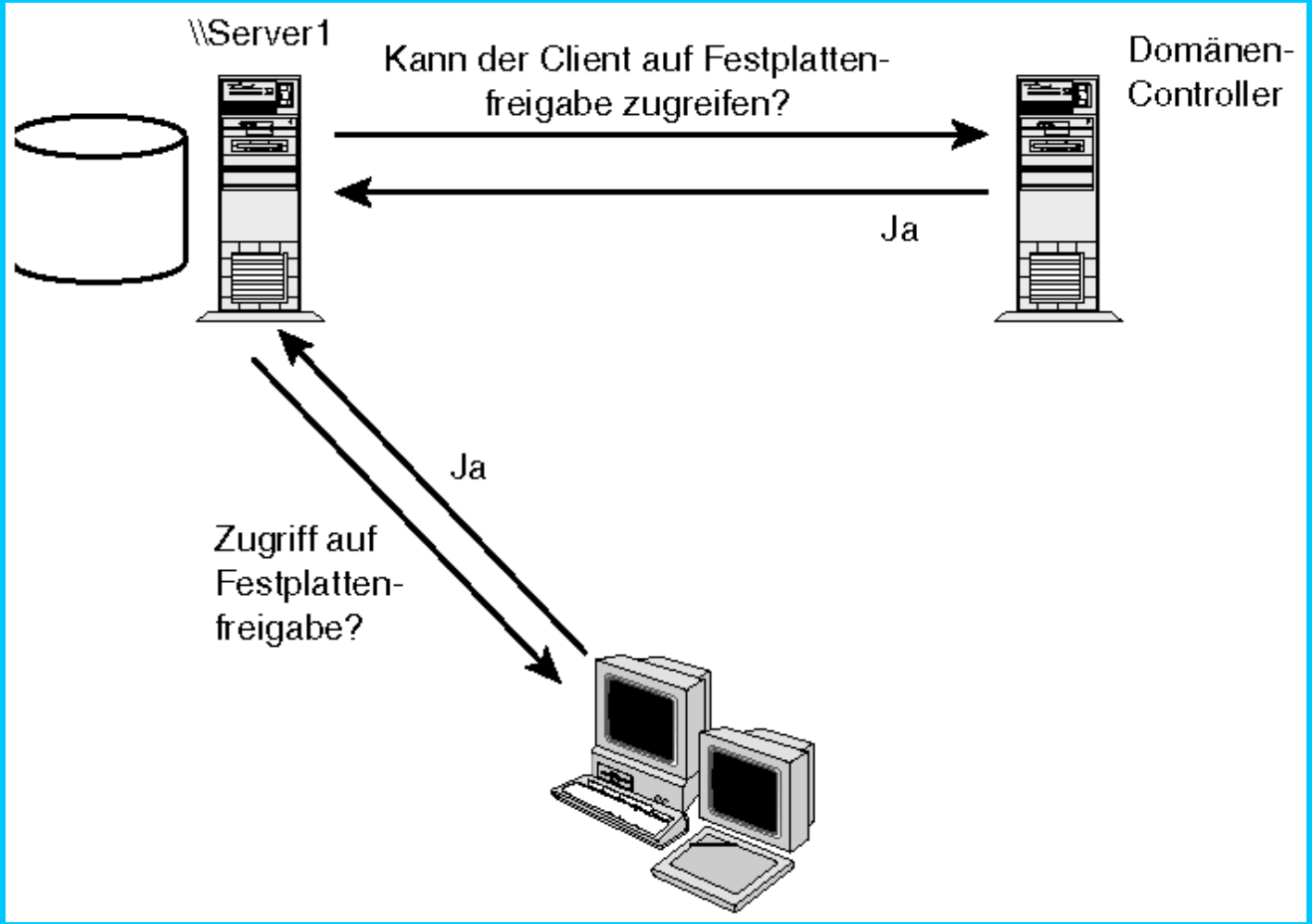


Abbildung 2.9 zeigt eine Beispielverbindung zu einem Server, der einer Domäne angehört. Zunächst sendet der Client die Verbindungsanfrage mit den Benutzerinformationen an SERVER1 und verlangt Zugriff auf eine Festplattenfreigabe. SERVER1 sendet dann eine Authentifizierungsanfrage an den Domänen-Controller (DC). Diese enthält die Benutzerinformationen, die ursprünglich vom Client übertragen wurden. Wenn der DC den Client erfolgreich authentifiziert, sendet er eine positive Antwort an SERVER1, der wiederum eine positive Verbindungsantwort an den Client zurückgibt. Das heißt, dass sich ein Client mit einem einzigen Benutzernamen und Passwort mit jedem Server in der Domäne verbinden kann, vorausgesetzt dass die Zugriffskontrollmechanismen wie z.B. Berechtigungslisten es erlauben. In Abbildung 2.8 brauchte der Client für jede Verbindung zu einem Server einen separaten Benutzernamen und ein Passwort.

Zusammenfassung

Die Protokolle NetBIOS und SMB umfassen noch viele Implementierungsdetails, die hier nicht dargestellt wurden. Die wichtigsten Punkte aus diesem Kapitel sind:

- NetBIOS bietet Clients drei Dienste: einen Name-Service, einen Session-Service und einen Datagram-Service.
- SMB ist ein verbindungsorientierter Dienst, der den NetBIOS-Session-Service benutzt.
- Eine SMB-Verbindung wird in drei Schritten aufgebaut: Verhandlung über den Protokolldialekt, Aufbau einer Sitzungsverbindung und Aufbau einer Verbindung zu einer bestimmten Freigabe.
- In Hinsicht auf Browsing sind ein Arbeitsgruppenname und ein Domänenname das Gleiche, dies gilt aber nicht in Bezug auf Authentifizierungsmodelle (siehe Abbildungen 2.8 und 2.9).

Frage & Antwort

F. Welche TCP- und UDP-Ports benutzt NetBIOS über TCP/IP?

. Der NBT-Name-Service benutzt Port 137/udp, der NBT-Session-Service Port 139/tcp und der NBT-Datagram-Service Port 138/udp.

F. Wo finde ich die NetBIOS-RFCs?

. Die ASCII-Versionen der RFCs 1001 und 1002 finden Sie auf der CD-ROM, die diesem Buch beiliegt. Natürlich können Sie sie auch

jederzeit von der IETF-Website (http://www.ietf.org) herunterladen.

F. Wo finde ich weitere Informationen über das SMB-Protokoll?

. Microsoft hat aus offensichtlichen Gründen das SMB- (bzw. CIFS-)Protokoll vorangetrieben. Das Unternehmen verwaltet eine Sammlung von Links und Informationen unter http://www.microsoft.com/workshop/networking/cifs.

Neue Begriffe

Broadcast-Namensregistrierung und -auflösung - Hosts übertragen Pakete an die Broadcast-Adresse des Netzwerks (z.B. `aaa.bbb.ccc.255` für ein Standard-Klasse-C-Netzwerk), um Namen zu registrieren und aufzulösen.

Angerufener Name (Called Name) - Der NetBIOS-Name des Servers, der in einer Verbindungsanfrage kontaktiert wird.

Anrufername (Calling Name) - Der NetBIOS-Name des Clients in einer NetBIOS-Verbindungsanfrage.

Domäne - Eine Sammlung von Rechnern, bei der die endgültige Authentifizierung von Verbindungsanfragen für jegliche Freigabe auf einem beliebigen Rechner von einem Domänen-Controller durchgeführt wird.

Domänen-Controller - Ein Rechner, der für die Authentifizierung von Dienstanfragen durch die Server in der Domäne verantwortlich ist.

NetBEUI - Erweiterte NetBIOS-Benutzerschnittstelle. NetBEUI ist ein von IBM entwickeltes Netzwerkprotokoll, das die Funktionalität des NetBIOS-API einbettet und erweitert.

NetBIOS - Network Basic Input/Output System. Ein API, das für die Programmierung von Netzwerkapplikationen entwickelt wurde.

NetBIOS-Scope - Ein String für die Segmentierung des NetBIOS-Namensbereichs. Hosts in einem NetBIOS-Bereich können Hosts in einem anderen nicht sehen.

Point-to-Point-Namensregistrierung und -auflösung - Hosts übertragen Pakete an einen einzelnen Host (Unicast), der als zentraler Nameserver für die Registrierung und Auflösung von Namen dient.

Arbeitsgruppe - Eine Sammlung von Rechnern, bei der jeder Rechner Client-Verbindungsanfragen für seine eigenen Freigaben authentifiziert.

Tag 3: Wie bekomme ich den aktuellsten Source-Code?

Manchmal finde ich in einer der Samba-Mailing-Listen oder einer Newsgroup eine Nachricht, in der es heißt: »Ich habe Samba Version 1.0 und kriege ____ nicht zum Laufen.« Die Antwort darauf lautet unweigerlich: »Aktualisieren Sie auf die neueste Version. Wenn es dann noch nicht funktioniert, stellen Sie Ihre Frage noch einmal.« Ich übertreibe vielleicht etwas, was die Version betrifft (um der Genauigkeit willen: während ich dies schreibe, ist die aktuellste Version 2.0), aber es ist wichtig, dass Sie sich darüber im Klaren sind, dass Code schnell weiterentwickelt und geändert wird, vor allem bei einem Open-Source-Software-Projekt wie Samba. Das Problem, auf das die Person gestoßen war, lag möglicherweise in einem Fehler begründet, der bereits korrigiert wurde.

Vielleicht installieren Sie Samba zum ersten Mal, oder der bisherige Samba-Administrator hat sich einen neuen Job gesucht, bei dem er fünfmal mehr verdient. Aus welchem Grund auch immer, irgendwann müssen Sie sich eine Kopie des aktuellsten Source-Codes besorgen und ihn selbst kompilieren. Vielleicht freuen Sie sich ja sogar auf diese Aufgabe.

Dieses Kapitel bietet Ihnen alle nötigen Informationen, um den aktuellsten Source-Code herunterzuladen und alle Optionen einzustellen, die für die Kompilierung wichtig sind. Außerdem werden Sie einen Blick auf die verfügbaren Binärdistributionen werfen (falls Ihnen nicht danach ist, die Dinge selber zu kompilieren).

Welche Samba-Version haben Sie derzeit?

Wenn bei Ihnen bereits Samba läuft und Sie wissen wollen, ob Sie die aktuellste Version haben, lesen Sie weiter! Wenn Sie Samba erstmals installieren, können Sie diesen Abschnitt für später aufsparen.

Es gibt zwei einfache Methoden, wenn Sie feststellen wollen, welche Samba-Version in Ihrem System installiert ist. Die erste Methode benutzt die Log-Dateien, die von den Daemons `smbd` und `nmbd` erstellt und hinterlassen werden, während die zweite Methode darin besteht, Informationen von den Prozessen selbst zu holen.

Schauen Sie sich zunächst die Log-Dateien an. Ein Samba-Server umfasst zwei Daemon-Prozesse: `smbd` und `nmbd`. Beide Prozesse erzeugen Logs, die standardmäßig in `/usr/local/samba/var` gespeichert werden und in der Regel `log.smb` bzw. `log.nmb` heißen. Sie können einen anderen Standort festlegen, indem Sie einen Wert in der Konfigurationsdatei definieren. Wenn Sie die Log-Dateien nicht in `/usr/local/samba/var` finden, besteht der nächste Schritt darin, die Konfigurationsdatei zu suchen, die normalerweise `smb.conf` heißt.

In der Regel befindet sich die Konfigurationsdatei in `/usr/local/samba/lib.` Wie für die meisten anderen Werte und Standorte in Samba ist es auch für `smb.conf` möglich, einen anderen Namen und Speicherplatz festzulegen, indem Sie über den Parameter `-s` ein Befehlszeilenargument an `smbd` und `nmbd` weiterleiten. Samba kann entweder über den Metadaemon `inetd` oder als eigenständiger Daemon gestartet werden. Um die von Ihrem

System benutzten Methoden zu bestimmen, werfen Sie über folgenden Befehl einen Blick in `/etc/inetd.conf`:

```
grep smbd /etc/indetd.conf
```

Wenn Ihr Rechner System-V-init-Skripte, wie z.B. Solaris 2.x oder RedHat Linux, benutzt, können Sie zum Start-Skript-Verzeichnis gehen, das in der Regel so ähnlich heißt wie `/etc/init.d` oder `/etc/rc.d`, und folgenden Befehl eingeben:

```
grep smbd *
```

Wenn Sie beim Starten von `smbd` bemerken, dass einem Parameter `-l` ein Dateiname folgt, ist dies Ihre Standard-Logdatei. Ansonsten geben Sie folgenden Befehl aus:

```
grep "log file" smb.conf
```

Die daraus resultierende Ausgabe sollte Ihnen einen absoluten Pfad zu einer Datei geben. Schauen Sie in diesem Verzeichnis nach den Samba-Logs.

Nachdem Sie die richtigen Logdateien gefunden haben, sollten Sie die Version des Samba-Daemons bestimmen können, der diese Logs erstellt hat, indem Sie die Datei durchsuchen:

```
root# grep "smbd version" log.smb | tail -l
smbd version 2.10-prealpha start
```

Wenn Sie keine Ausgabe erhalten, die der obigen ähnelt, heißt das entweder, dass die Logs durch einen `cron`-Job auf dem System überschrieben wurden oder dass sie von Samba selbst aufgrund ihrer Größe gelöscht wurden. Wenn Sie wollen, dass Samba diese Information erneut darstellt, können Sie die Logs entfernen und die `smbd`- und `nmbd`-Prozesse neu starten.

Es ist unmöglich, hier zu beschreiben, wie Samba auf jedem möglichen System neu gestartet wird. Dieses Beispiel aus einem Slackware-3.5-Linux-System sollte Ihnen eine generelle Vorstellung geben:

```
[root@bilbo /etc] killall smbd
[root@bilbo /etc] killall nmbd
[root@bilbo /etc] /etc/rc.d/rc.samba
```

Sollten Sie vergessen, die Logs vorher zu entfernen, hängt Samba die neuen Einträge einfach an die vorhandenen Logs an.

In Samba-Versionen vor 2.0 überschreibt `nmbd` standardmäßig die alte `log.nmb`-Datei, während `smbd` Log-Einträge anhängt. In Version 2.0 hängen sowohl `smbd` als `nmbd` Log-Einträge standardmäßig an.

Eine andere Möglichkeit, festzustellen, welche Samba-Version derzeit läuft, besteht darin, das `smbclient`-Utility zu benutzen, das in der Samba-Distribution enthalten ist. Obwohl diese Methode wesentlich einfacher ist, setzt sie voraus, dass `smbclient` installiert ist und Samba läuft.

Suchen Sie die `smbclient`-Binärdatei (in der Regel in `/usr/local/samba/bin`) und führen Sie folgenden Befehl aus:

```
smbclient -L localhost
```

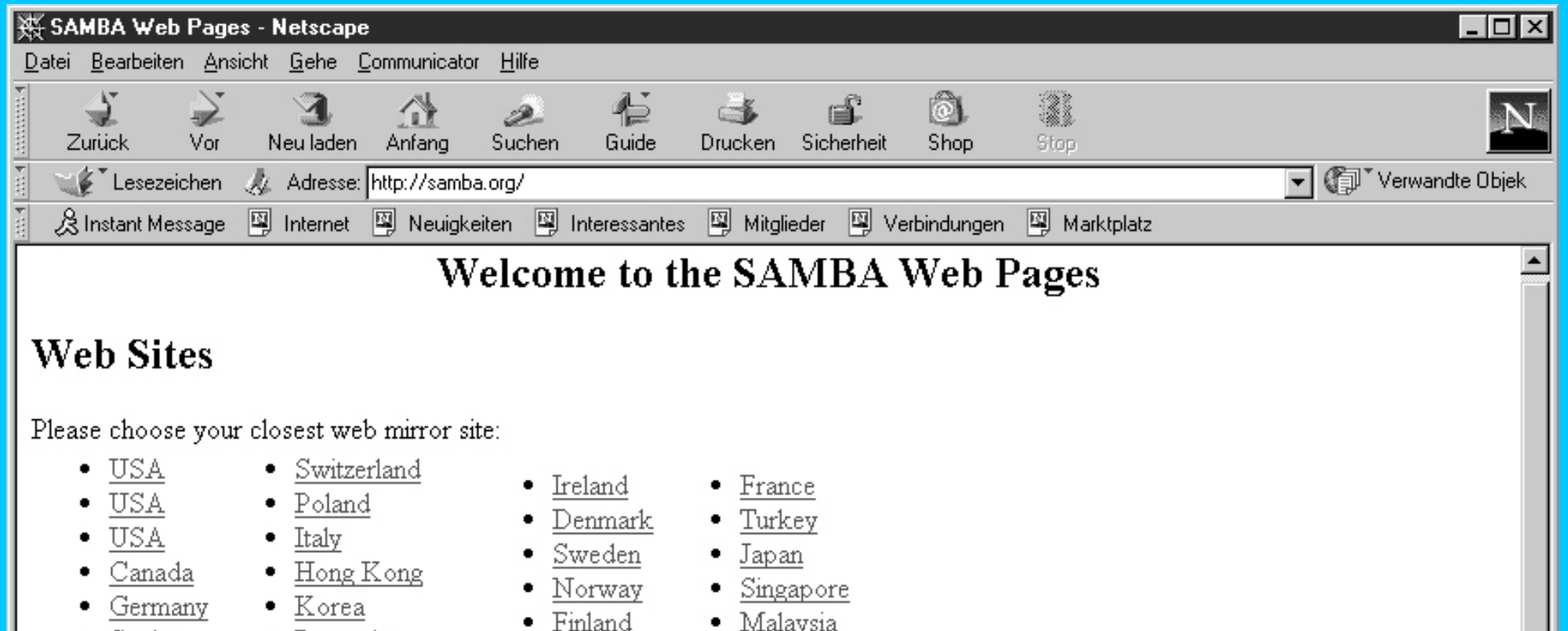
Als Ergebnis sollten Sie etwa Folgendes sehen (möglicherweise mit zusätzlichem Text; aaa, bbb, ccc und ddd sind Platzhalter für Zahlenwerte aus Ihrem definierten Netzwerk):

```
Added interface ip=aaa.bbb.ccc.ddd bcast=aaa.bbb.ccc.255 nmask=255.255.255.0
Domain=[CHIPSNDIPS]  OS=Unix  Server=[Samba 2.1.0-prealpha]
Sharename       Type      Comment
-------------   ------    ------------
...
```

Die Informationen zur Samba-Version auf dem Rechner, mit dem Sie verbunden sind (dem lokalen Rechner), sind in den eckigen Klammern (`[]`) dargestellt, die der Bezeichnung `Server=` folgen. Die Version im vorstehenden Beispiel ist `2.1.0-prealpha` (Testcode).

Wenn Sie nun wissen, dass Sie die neueste Version von Samba brauchen, wohin können Sie gehen? Als Startpunkt sollten Sie Ihren Browser an http://www.samba.org verweisen (siehe Abbildung 3.1). Diese Website bietet Links zu allen Samba-Mirror-Websites sowie den FTP-Mirrors. Hierbei handelt es sich um Server, die in kurzen und regelmäßigen Zeitabständen den Originalserver kopieren (»spiegeln«). Liegt ein solcher Server geografisch in Ihrer Nähe, ermöglicht er Ihnen so einen schnelleren Download der Dateien.

Abb. 3.1: Liste der Mirrors für http://www.samba.org



- Spain
- Portugal
- Russia

- Romania
- Hungary
- Czech Repuplic

- Iceland
- Iceland

- Australia
- Australia

Download sites

These contain the source and binary distributions but not the web pages.

- USA
- USA
- USA
- Canada
- United Kingdom
- Italy
- Italy
- Czech Republic
- Costa Rica

- Russia
- Poland
- Poland
- Greece
- Japan
- Japan
- Japan
- South Africa

- Austria
- Germany
- Germany
- Denmark
- Sweden
- Norway
- Finland
- Slovenia

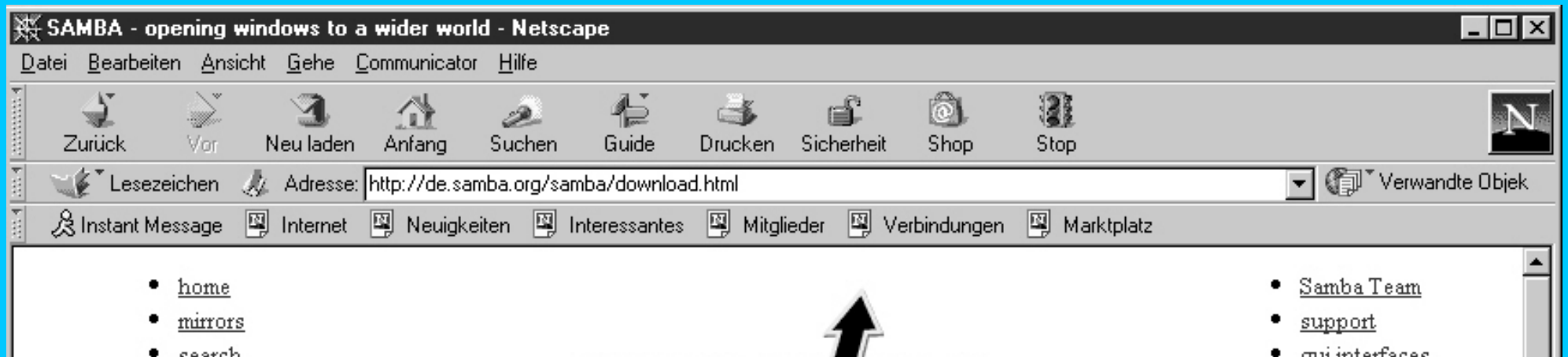
- Turkey
- Portugal
- Singapore
- Korea
- Hong Kong
- Australia
- France
- Brasil

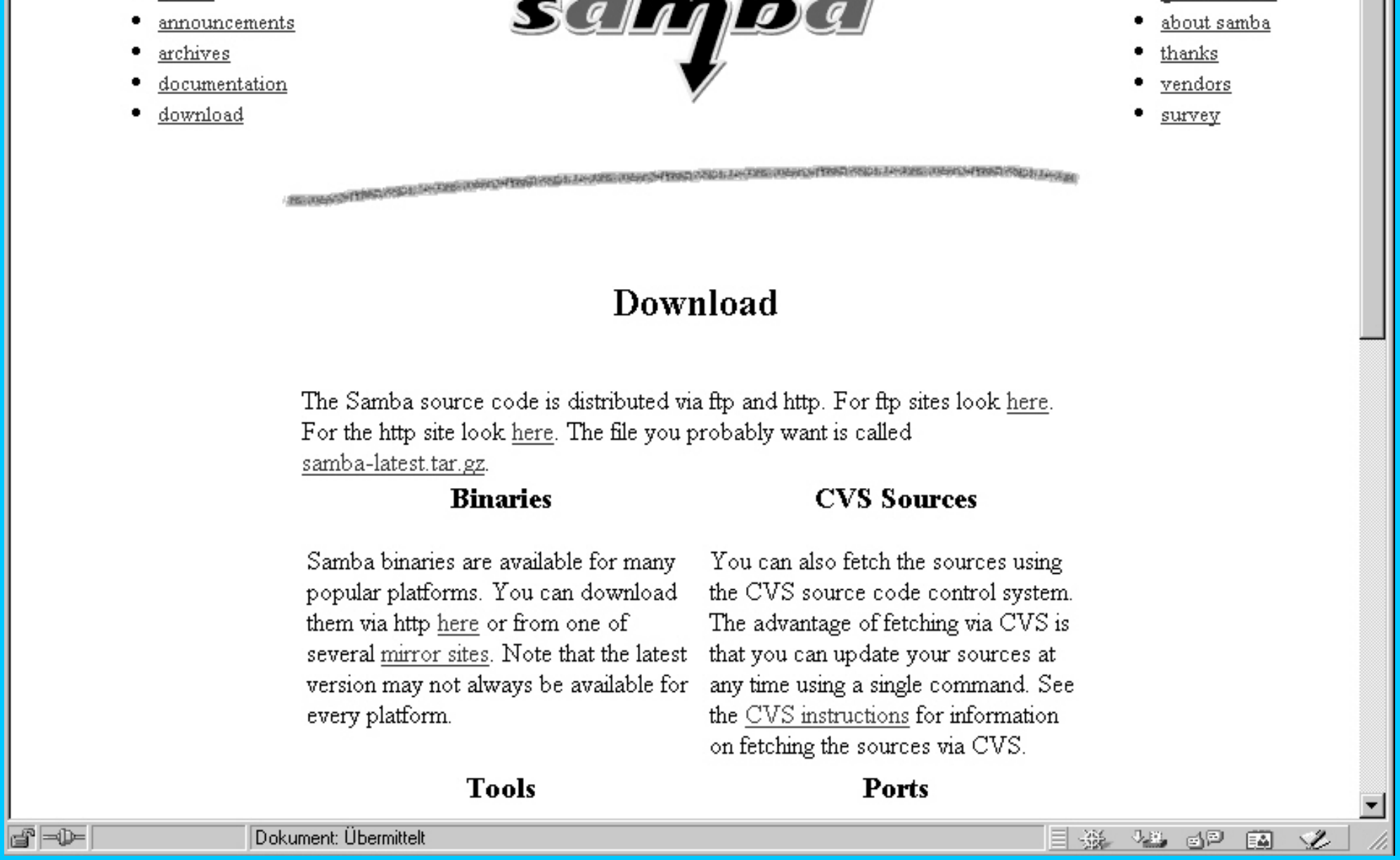
Please refer to these mirroring instructions for information on mirroring the Samba web pages.

Dokument: Übermittelt

Wenn Sie beim Herunterladen der Samba-Distribution HTTP benutzen, wählen Sie zunächst die Mirror-Site, die Ihrem Standpunkt am nächsten liegt. Wählen Sie dann auf der Homepage der Mirror-Site den Download-Link. Die sich dann öffnende Seite, die Sie in Abbildung 3.2 sehen können, sollte einen Link haben, über den Sie eine Datei namens `samba-latest.tar.gz` herunterladen können.

Abb. 3.2: Samba-Download-Seite





Wenn Sie es vorziehen zu browsen, um zu sehen, was verfügbar ist, können Sie eine Verzeichnisauflistung zum Herunterladen erhalten (siehe Abbildung 3.3). Auch hier sollten Sie eine Datei namens `samba-latest.tar.gz` vorfinden. Falls nicht, suchen Sie einfach nach `samba-#####.tar.gz` mit der höchsten Versionsnummer. Die Suche nach dem aktuellsten Distributionsarchiv über FTP entspricht in etwa dem Browsen der Verzeichnisauflistung über HTTP.

Nachdem Sie die Distribution heruntergeladen haben, wechseln Sie in ein temporäres Verzeichnis, in dem Sie nichts überschreiben können, wenn Sie die Source-Dateien entpacken. Ich benutze ein Verzeichnis namens `~/src` als Arbeitsverzeichnis für die Kompilierung von Source-Codes. Wenn ich nach dem Beenden etwas speichern möchte, verschiebe ich die Dateien an einen beständigeren Speicherplatz. Sie benötigen eine Arbeitsversion von GNU

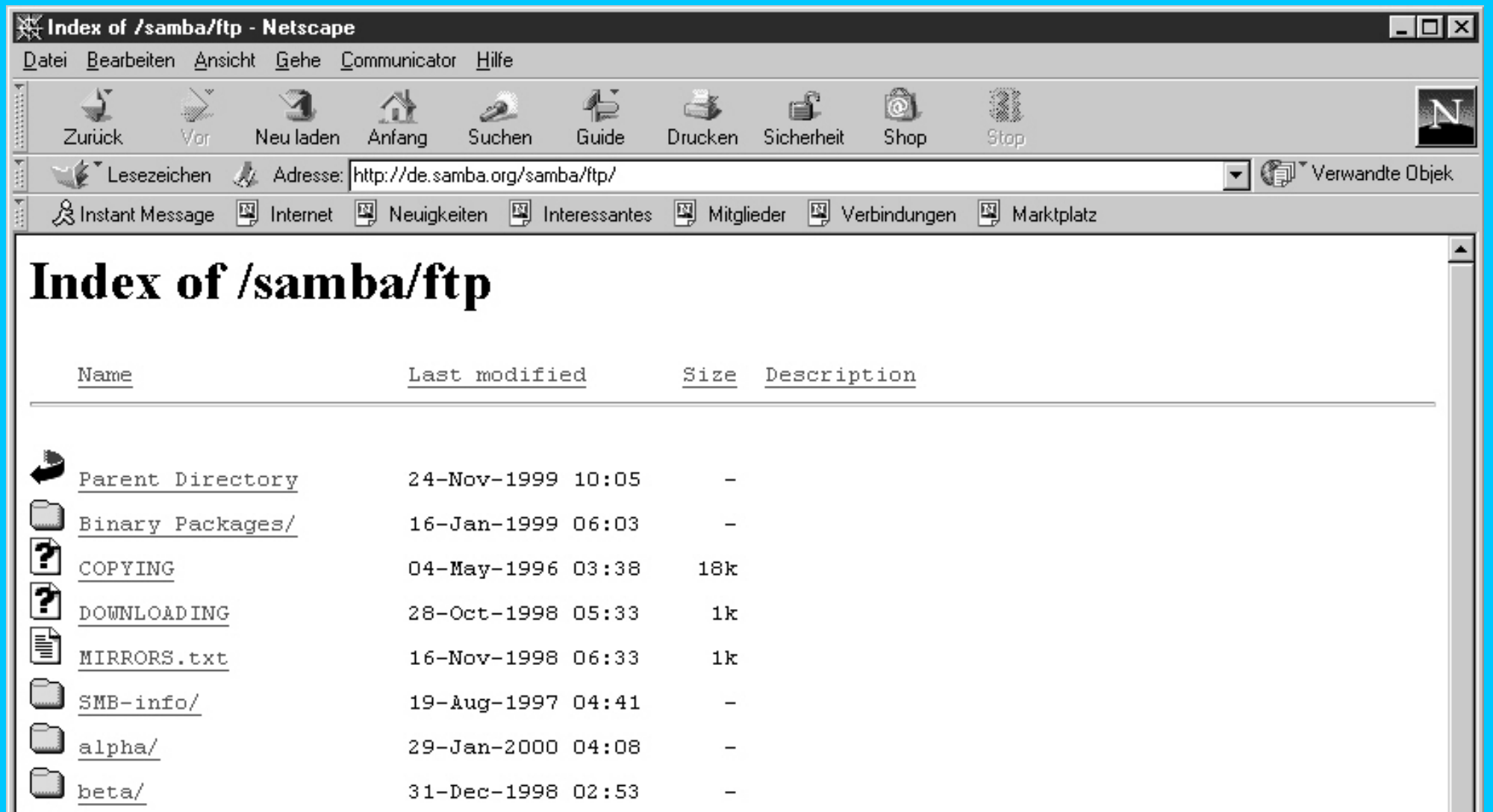
`gzip` und `tar`, um die Dateien auszupacken. Ist der Name der heruntergeladenen Datei z.B. `samba-latest.tar.gz`, packt der folgende Befehl die Dateien für Sie aus:

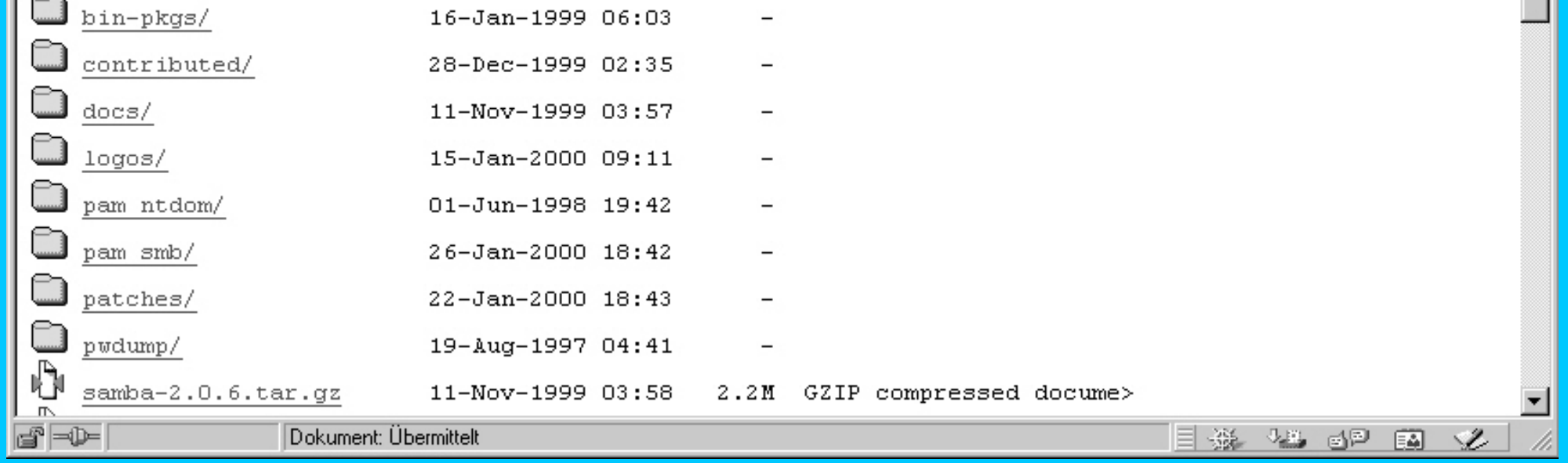
```
gzip -dc samba-latest.tar.gz | tar xvf -
```

Obwohl sich der Verzeichnisbaum von Zeit zu Zeit ändern kann, haben alle Versionen bisher drei Verzeichnisse gemeinsam:

- `docs/` - Dieses Verzeichnis enthält verschiedene Dokumentationen wie z.B. Manpages, HTML-Dateien und HOWTO-Dateien im ASCII-Format.
- `examples/` - Dieses Verzeichnis enthält mehrere Beispiele für viele verschiedene Betriebssysteme, die unterschiedliche Setup-Möglichkeiten beschreiben. Enthält hauptsächlich Beispiel-`smb.conf`-Dateien.
- `source/` - Dieses Verzeichnis enthält den Samba-Source-Code-Baum für die Distribution.

Abb. 3.3: Ordnerauflistung der Download-Archive auf der Samba-Website





Samba kompilieren

Nachdem Sie den Source-Code von Samba vorliegen haben, hängen die nächsten Schritte für die Kompilierung davon ab, welche Version von Samba Sie installieren möchten. In den Versionen vor 2.0 werden die Dinge im Vergleich zur Version 2.0 und höher ganz anders durchgeführt, da die Version 2.0 komplett überarbeitet wurde und GNU-`autoconf` verwendet.

Vor 2.0

Vielleicht ist es Ihnen aus irgendwelchen Gründen nicht möglich, auf die neueste Version zu aktualisieren. Vielleicht müssen Sie ein bereits existierendes System administrieren und eine der Standardeinstellungen ändern, die während der Kompilierung erstellt werden. Wie ich zuvor schon erwähnt habe, unterscheidet sich die Kompilierung der Versionen vor 2.0 von der aktuellen Methode. Der Vollständigkeit halber möchte ich noch erwähnen, dass die letzte Version vor 2.0 die Version 1.9.18p10 war.

Im Code-Baum der Versionen vor 2.0 befinden sich alle Source-Dateien in einem Verzeichnis, das auch ein Makefile enthält, das Sie vor der Kompilierung geringfügig ändern müssen:

```
# Das Basisverzeichnis für alle Samba-Dateien
BASEDIR = /usr/local/samba
```

Eine der Optionen, die Sie ändern sollten, ist das Basisverzeichnis für die Samba-Binärdateien. Die Inhalte dieses Verzeichnisses werden später in diesem Kapitel dargestellt. Zunächst einmal sollten Sie es als den Ort ansehen, an dem Samba Informationen sucht und ablegt. Die meisten dieser Speicherplätze können Sie über `smb.conf`-Optionen oder Befehlszeilenparameter selbst bestimmen. Ich ziehe es jedoch vor, Samba explizit mitzuteilen, wo sich die nötigen Dateien befinden, und keine Vermutungen darüber anzustellen, aber das ist meine persönliche Meinung:

```
# Die Verzeichnisse, in der die Dinge abgelegt werden.
# Wenn Sie mehrere Architekturen benutzen oder die
```

```
# Samba-Binaries über NFS freigeben, werden Sie diese
# Struktur wahrscheinlich ändern wollen.
# Hinweis: SBINDIR ist für Dateien, auf die Benutzer nicht
# zugreifen sollen
# gilt normalerweise nur für nmbd und smbd
# SBINDIR bedeutet ein sicheres Binärverzeichnis
BINDIR = $(BASEDIR)/bin
SBINDIR = $(BASEDIR)/bin
LIBDIR = $(BASEDIR)/lib
VARDIR = $(BASEDIR)/var
```

Wenn Sie die einzelnen Samba-Verzeichnisse über mehrere Verzeichnisse verteilen wollen, können Sie die Werte von `$ {BINDIR}`, `$ {LIBDIR}` und `{VARDIR}` ändern.

Sie können außerdem die Standorte für einzelne Dateien ändern, indem Sie eine der folgenden Variablen des Makefile ändern. Wenn Sie nicht sicher sind, was die einzelnen Dateien genau darstellen, können Sie ganz einfach die Standardeinstellungen beibehalten.

```
# geben Sie hier an, wo die verschiedenen Dateien zu finden
# sind
# dies kann durch Befehlszeilenparameter (siehe smbd(8))
# oder in smb.conf (siehe smb.conf (5)) überschrieben
# werden
SMBLOGFILE = $(VARDIR)/log.smb
NMBLOGFILE = $(VARDIR)/log.nmb
```

Bevor ich fortfahre, möchte ich auf die Konventionen hinweisen, über die bestimmte Abschnitte der Manpages für einen Eintrag durch Hinzufügen von `(Zahl)` definiert werden. In dem hier aufgelisteten Auszug aus dem Makefile heißt die Angabe `smbd (8)`, was bedeutet, dass sich die `smbd`-Manpage in Abschnitt 8 befindet. Standard-Installationsverzeichnisse vorausgesetzt befindet sich die `smbd`-Manpage in `/usr/local/samba/mab/man8`.

Die `smbd`- und `nmbd`-Daemons schreiben ihre Logging-Informationen in die Logdateien:

```
CONFIGFILE = $(LIBDIR)/smb.conf
LMHOSTSFILE = $(LIBDIR)/lmhosts
```

Die `smb.conf`-Datei habe ich bereits erwähnt. Die Datei `lmhosts`-Datei ist die NetBIOS-Entsprechung einer `/etc/hosts`-Datei:

```
DRIVERFILE = $(LIBDIR)/printers.def
```

Die Datei `printers.def` enthält Informationen über Druckertreiber, die von Windows-95- und Windows-98-Clients direkt heruntergeladen und installiert werden können, wenn sie sich mit dem Drucker verbinden:

```
SMB_PASWD = $(BINDIR)/smbpasswd
SMB_PASSWD_FILE = $(BASEDIR)/private/smbpasswd
```

Diese Dateien sind das Samba-`passwd`-Utility bzw. die `passwd`-Datei. Sie werden in Kapitel 6, »Sicherheitsmodi und Passwörter«, dargestellt.

```
# das Verzeichnis für Lock-Dateien
LOCKDIR = $(VARDIR)/locks
```

Im `lock`-Verzeichnis platziert Samba Dinge wie die aktuelle Browse-Liste (siehe Kapitel 19, »Lokales Subnetz-Browsing «, und 20, »Router-Netzwerke und Browsing«), Informationen zum Datei-Locking und die WINS-Datenbank (siehe Kapitel 18, »NetBIOS-Namen ohne Broadcasts auflösen«).

Einer der größten Vorteile der `autoconf`-Unterstützung liegt darin, dass Sie nicht viele Einzelheiten über Ihr Betriebssystem angeben müssen. In einem Makefile vor Version 2.0 müssen Sie die Parameter in dem Abschnitt des Makefile auskommentieren, der sich auf Ihr Betriebssystem bezieht. Der folgende Abschnitt gilt z.B. für Linux-Installationen, die Passwort-Shadowing benutzen, aber nicht `PAM`. Die Variablen `FLAGSM` und `LIBSM` wurden vorher durch ein vorangestelltes # ebenso auskommentiert wie alle betriebssystemspezifischen Abschnitte:

```
# Dies gilt für Linux mit Shadow-Passwörtern ohne PAM!
# beigetragen von Andrew.Tridgell@anu.edu.au
# fügen Sie -DLINUX_BIGCRYPT hinzu, wenn Sie Shadow-
# Passwörter haben, aber nicht die entsprechenden
# Libraries und Includes
FLAGSM = -DLINUX -DSHADOW_PWD -DFAST_SHARE_MODES
LIBSM = -lshadow
```

Ein weiterer Parameter, dessen Einstellung oft nützlich ist, ist die Variable `CC`. Standardmäßig ist `CC` für die Benutzung des `cc`-Compilers eingerichtet. Wollen Sie stattdessen den `gcc`-Compiler benutzen, müssen Sie die Zeile auskommentieren, in der es heißt: `CC = gcc`.

Neben der Einrichtung der Makefile-Variablen müssen Sie vielleicht noch einige unternehmensspezifische Werte im Header `local.h` einstellen, der sich im Verzeichnis `/source` befindet. Die Datei selbst und die Makros sind weitestgehend selbsterklärend. In der Regel ist es besser, die Datei so zu belassen, wie sie ist, es sei denn, Sie sind sich sicher, dass Sie etwas ändern müssen.

Ein Makro, das ich jedoch regelmäßig ändern musste, war `MAX_OPEN_FILES`. Der Standardwert in Versionen vor 2.0 ist hier `100`. Aufgrund einiger Umschreibungen des Codes wurde der neue Samba-Code in Hinsicht auf den Umgang mit Dateien wesentlich effizienter gestaltet. Die Standardeinstellung für die Version 2.0 ist `10.000`.

Ein Unternehmen, für das ich den Support lieferte, benutzte eine freigegebene Datenbank, und das Front End musste etwa 150 Dateien gleichzeitig öffnen. Ich löste mein Problem, indem ich den Wert für die maximale Anzahl offener Dateien auf `200` setzte. Offensichtlich gibt es mit der Version 2.0 keine derartigen Probleme! Ein anderer bemerkenswerter Unterschied ist, dass in der Version 2.0 diese Einstellung auch über einen Runtime-Parameter in der Datei `smb.conf` eingerichtet werden kann.

Nachdem Sie die notwendigen Änderungen an dem Makefile und der Headerdatei `local.h` (wenn überhaupt) durchgeführt haben, können Sie die Binaries erstellen, indem Sie **make** eingeben. Wenn alles gut geht, sollten sich `smbd`, `nmbd` und andere Utilities im gleichen Verzeichnis wiederfinden wie die Source-Dateien.

autoconf-Unterstützung von 2.0

Im Gegensatz zu den Versionen vor 2.0 ist das Kompilieren von Samba 2.0 sehr einfach. Es gibt ein `configure`-Skript, das sich im Verzeichnis `/source` befindet. Um die richtige Unterstützung für Ihr Betriebssystem zu aktivieren, wechseln Sie einfach in das Source-Verzeichnis und führen den folgenden Befehl aus:

```
./configure
```

Das Skript läuft und generiert das entsprechende Makefile für Sie.

Es ist zwar möglich, das erzeugte Makefile manuell zu editieren und Variablen einzustellen, aber es ist einfacher, einige der Optionen als Befehlszeilenparameter zu spezifizieren. Wenn Sie Informationen über die verfügbaren Optionen brauchen, geben Sie `./configure --help` ein.

In Listing 3.1 finden Sie exemplarische Auszüge der Ausgabe. Einige Zeilen habe ich gelöscht, damit das Listing nicht zu lang wird; diese Änderungen sind mit Klammern ({}) gekennzeichnet.

Listing 3.1: Beispielausgabe von ./configure --help

```
01: Usage: configure [options] [host]
02: Options: [Standards nach Beschreibung in Klammern]
03: Configuration:
04: {Zeilen ausgelassen}
05: Directory and file names:
06: --prefix=PREFIX         installiere architekturunabhängige Dateien in PREFIX
07: [/usr/local/samba]
08: --exec-prefix=EPREFIX   installiere architekturabhängige Dateien in EPREFIX
09: [wie prefix]
10: {Zeilen ausgelassen}
11: Host type: {Zeilen
12: ausgelassen}
13: Features and packages:
14: {Zeilen ausgelassen}
15: --enable and --with options recognized:
16: --enable maintainer-mode  einige make-Regeln für Verwalter aktivieren
17:   --with-smbwrapper       Mit SMB-Wrapper-Unterstützung
18:   --without-smbwrapper    ohne SMB Wrapper-Unterstützung (Standard)
19:   --with-afs              mit AFS-Unterstützung
20:   --without-afs           ohne AFS-Unterstützung (Standard)
21:   --with-dfs              mit DFS-Unterstützung
22:   --without-dfs           ohne DFS-Unterstützung (Standard)
23:   --with-krb4=base-dir    mit Kerberos-IV-Unterstützung
24:   --without-krb4          ohne Kerberos-IV-Unterstützung (Standard)
25:   --with-automount        mit AUTOMOUNT-Unterstützung
26:   --without-automount     ohne AUTOMOUNT-Unterstützung (Standard)
```

```
27:   --with-smbmount          mit SMBMOUNT-Unterstützung (nur Linux)
28:   --without-smbmount       ohne SMBMOUNT-Unterstützung (Standard)
29:   --with-ldap              mit LDAP-Unterstützung
30:   --without-ldap           ohne LDAP-Unterstützung (Standard)
31:   --with-nisplus           mit NISPLUS-Passwortdatenbank-Unterstützung
32:   --without-nisplus        ohne NISPLUS-Passwortdatenbank-Unterstützung (Standard)
33:   --with-nisplus-home      mit NISPLUS_HOME-Unterstützung
34:   --without- nisplus-home  ohne NISPLUS_HOME-Unterstützung (Standard)
35:   --with-ssl               mit SSL-Unterstützung
36:   --without-ssl            ohne SSL-Unterstützung (Standard)
37:   --with-mmap              mit experimenteller MMAP-Unterstützung
38:   --without-mmap           ohne MMAP- Unterstützung (Standard)
39:   --with-syslog            mit experimenteller SYSLOG-Unterstützung
40:   --without-syslog         ohne SYSLOG-Unterstützung (Standard)
41:   --wiht-netatalk          mit experimenteller Netatalk-Unterstützung
42:   --without netatalk       ohne experimentelle Netatalk-Unterstützung (Standard)
43:   --with-quotas            mit experimenteller Festplatten-Quota-Unterstützung
44:   --without-quotas         ohne experimentelle Festplatten-Quota-Unterstützung (Standard)
45:   --with-privatedir=DIR    Standort von smbpasswd (/usr/local/samba/private)
46:   --with-swatdir=DIR       Standort für SWAT-Dateien (/usr/local/samba/swat)
```

Die Liste der Optionen ist relativ lang, aber in der Realität tauchen viele von ihnen als Paar auf: eine für die Aktivierung der Option, die andere für die Deaktivierung. Schauen Sie sich z.B. die Optionen in Zeile 25 und 26 an. Die erste aktiviert `automount`-Unterstützung und die zweite deaktiviert sie. Sie sollten außerdem beachten, dass das Listing zeigt, welche Optionen standardmäßig aktiviert bzw. deaktiviert sind.

Wenn Sie den Standort des Top-Level-Installationsverzeichnisses (`/usr/local/samba`) in `/usr/samba` ändern wollen, geben Sie

```
--prefix=/usr/samba
```

als Parameter an, wenn Sie den Befehl **configure** ausführen. Um alle Standardoptionen für die Erstellung des Makefile zu akzeptieren, starten Sie einfach

```
./configure
```

Das `configure`-Skript erstellt ein Makefile im Verzeichnis `/source`. Jetzt müssen Sie nur noch **make** eingeben, um die Samba-Binaries zu übersetzen.

Was kopiert sich wohin, wenn ich make install eingebe?

Wenn die Binaries fertig sind, egal für welche Samba-Version, können Sie die Dateien in dem Verzeichnis installieren, das durch die Variable `prefix` (oder `BASEDIR`) im Makefile spezifiziert ist, indem Sie **make install** eingeben. Ich setze hier, und auch für den Rest des Buchs, voraus, dass Samba in das Standardverzeichnis `/usr/local/samba` installiert wird.

Der Samba-Verzeichnisbaum wird, wenn nötig, erzeugt, und die Binaries und anderen relevanten Dateien werden hineinkopiert. Nach Fertigstellung

sollten die folgenden Verzeichnisse im Samba-Installationsverzeichnis existieren, es sei denn, Sie haben die Standorte im Makefile geändert:

- `bin/` - Dieses Verzeichnis enthält die `smbd`- und `nmbd`-Binaries und andere in Samba enthaltene Utilities.
- `lib/` - Dieses Verzeichnis enthält die Dateien `smb.conf` und `lmhosts` sowie die Dateien für die Codepage-Unterstützung im Unterverzeichnis `codepages/`.
- `var/` - Dieses Verzeichnis ist leer, bis Samba zum erstenmal läuft. Dann erzeugen die Daemons `smbd` und `nmbd` die Lock-Dateien, die geteilten Speicherdateien, die Browse-Listen-Informationsdatei und möglicherweise die WINS-Datenbanken. Unter Samba 2.0 sind außerdem die Dateien `smbd.pid` und `nmbd.pid` zusammen mit der Prozess-ID der aktuell laufenden Daemons enthalten. Diese sind nützlich für einen einfachen Neustart. Das Verzeichnis ist auch der Standardstandort für die Samba-Logdateien.
- `man/` - Hier finden Sie in verschiedenen Unterverzeichnissen die Samba-Manpages. Wenn Sie die Seiten in Ihren Manpage-Suchpfad einbinden wollen, können Sie sie entweder an einen vorhandenen Manpage-Standort verschieben oder sie in Ihre `MANPATH`-Umgebungsvariable einfügen. Befinden sich die Samba-Manpages z.B. in `/usr/local/samba/man` unter der Bourne- oder bash-Shell, können Sie dieses Verzeichnis durch folgende Einstellung in den vorhandenen Suchpfad aufnehmen:

  ```
  MANPATH=$MANPATH:/usr/local/samba/man
  ```

- `swat/` - Dieses Verzeichnis enthält die Dateien für den GUI-`smb.conf`-Editor `SWAT`, der in Kapitel 9, »GUI-Administrationstools«, ausführlicher dargestellt wird.

Die Binärdistribution

Wenn Sie sich, aus welchen Gründen auch immer, dafür entscheiden, den Source-Code nicht selbst zu kompilieren oder dies auf Ihrem System nicht können (z.B. wegen eines fehlenden C-Compilers), bietet es sich jetzt an, über die Möglichkeit zu reden, nur die Binärdateien herunterzuladen. In Kapitel 1, »Einführung in Samba«, habe ich die Grundlagen der GPL dargestellt. Eine der Bedingungen der Lizenz ist, dass der Source-Code nicht mit den Binaries verteilt werden, aber auf Anfrage verfügbar sein muss.

Meinen ersten Job als Netzwerkadministrator hatte ich während des Studiums. Ich wusste noch nicht viel und lernte sozusagen »on the job«. Meine Aufgabe bestand darin, ein Computerlabor für Studenten aufzubauen. Eines der Dinge, die ich für das Labor kaufte, war eine Sparc Ultra mit Solaris 2.5.1. Stellen Sie sich mein Erstaunen vor, als ich begann, Software zu installieren, und feststellte, dass Sun keinen C-Compiler mit Solaris 2 lieferte!

Kurz und gut, Binaries können je nach Ihren Bedürfnissen sehr hilfreich sein. Wenn Sie nicht planen, die Standardeinstellungen während der Kompilierung zu ändern oder keine besonders außergewöhnlichen Unternehmensstrukturen haben, spart Ihnen das Herunterladen der Binaries wahrscheinlich einiges an Zeit. Denken Sie daran, dass der Source-Code immer verfügbar ist, falls Sie ihn später doch noch brauchen.

Es gibt eine Mailing-Liste, die speziell für Diskussionen über die Samba-Binärpakete eingerichtet wurde. Die Adresse ist samba-binaries@samba.org. Weitere Informationen über Samba-Mailing-Listen finden Sie unter http://samba.org/listproc.

Sie erhalten eine Samba-Binärdistribution in etwa auf die gleiche Art und Weise wie die Source-Code-Freigabe. Schauen Sie zunächst in die Samba-Homepage (http://samba.org) und wählen Sie die FTP- oder HTTP-Site, die Ihnen am nächsten liegt. Wenn Sie für das Herunterladen der Dateien HTTP benutzen, folgen Sie wieder dem Download-Link auf der Samba-Mirror-Homepage und suchen Sie nach Informationen über das Herunterladen von Binärpaketen. Wenn Sie FTP benutzen, suchen Sie nach einem Verzeichnis namens `bin-pkgs` oder `Binary_Packages` und wählen Sie dann Ihr Betriebssystem und bei der Arbeit mit einem PC gegebenenfalls die Linux-Distribution aus.

Binärpakete sind nicht für alle Plattformen verfügbar, auf denen Samba kompiliert werden kann, und auch nicht immer für die aktuellste Source-Code-Freigabe. Das liegt daran, dass die Pakete auf freiwilliger Basis kompiliert und auf den Server hochgeladen werden. Wenn möglich, werden die Binaries und verbundene Dateien mit den Tools für das native Betriebssystem archiviert. RedHat-Binaries z.B. werden unter Benutzung von RPM gespeichert und Solaris-Binaries im `pkgtool`-Format verteilt. Es würde den Rahmen dieses Buches sprengen, die Benutzung der Paket-Distributionstools verschiedener Betriebssysteme detailliert darzustellen. Wenn Sie weitere Informationen über das Binär-Distributionstool Ihres Betriebssystems brauchen (Solaris 2.x z.B. benutzt `pkgadd` und RedHat-Linuxsysteme `rpm`), lesen Sie bitte die Manpages für das Installationstool durch, welches Sie benötigen.

Sie haben nun die gewünschte Samba-Version heruntergeladen und die Binärdateien zu Ihrer Verfügung. In Kapitel 4, »Installation und Testen der Konfiguration«, werden Sie lernen, eine Beispielinstallation zum Laufen zu bringen.

Zusammenfassung

Sie haben zwei Optionen, wenn Sie Samba installieren wollen. Sie können entweder den Source-Code herunterladen und selbst kompilieren, oder Sie können sich eine Binärversion besorgen, sofern sie für Ihre Plattform verfügbar ist.

Um die Kompilierung von Samba 2.0 vorzubereiten, führen Sie das `configure`-Skript aus, das sich im Source-Verzeichnis der Distribution befindet, und spezifizieren alle notwendigen Optionen. Für Versionen vor Samba 2.0 müssen Sie das Makefile manuell editieren und die entsprechende Unterstützung für Ihr Betriebssystem einrichten. Sind alle Optionen korrekt eingerichtet, werden sowohl die Version 2.0 als auch alle früheren kompiliert, indem Sie den Befehl **make** ausführen. Nach Erzeugen der Binaries können Sie diese über den Befehl **make install** auf Ihrem lokalen System installieren.

Frage & Antwort

F. Ich habe Samba 2.0 heruntergeladen. Ich habe die Dateien entpackt und make eingegeben, aber das System gibt die Meldung »make: Fatal error: No arguments to build« aus.

. Sie müssen vor Ausführung von `make` das Skript `configure` laufen lassen.

Ergänzung zur deutschen Übersetzung: Samba und SuSE-Linux

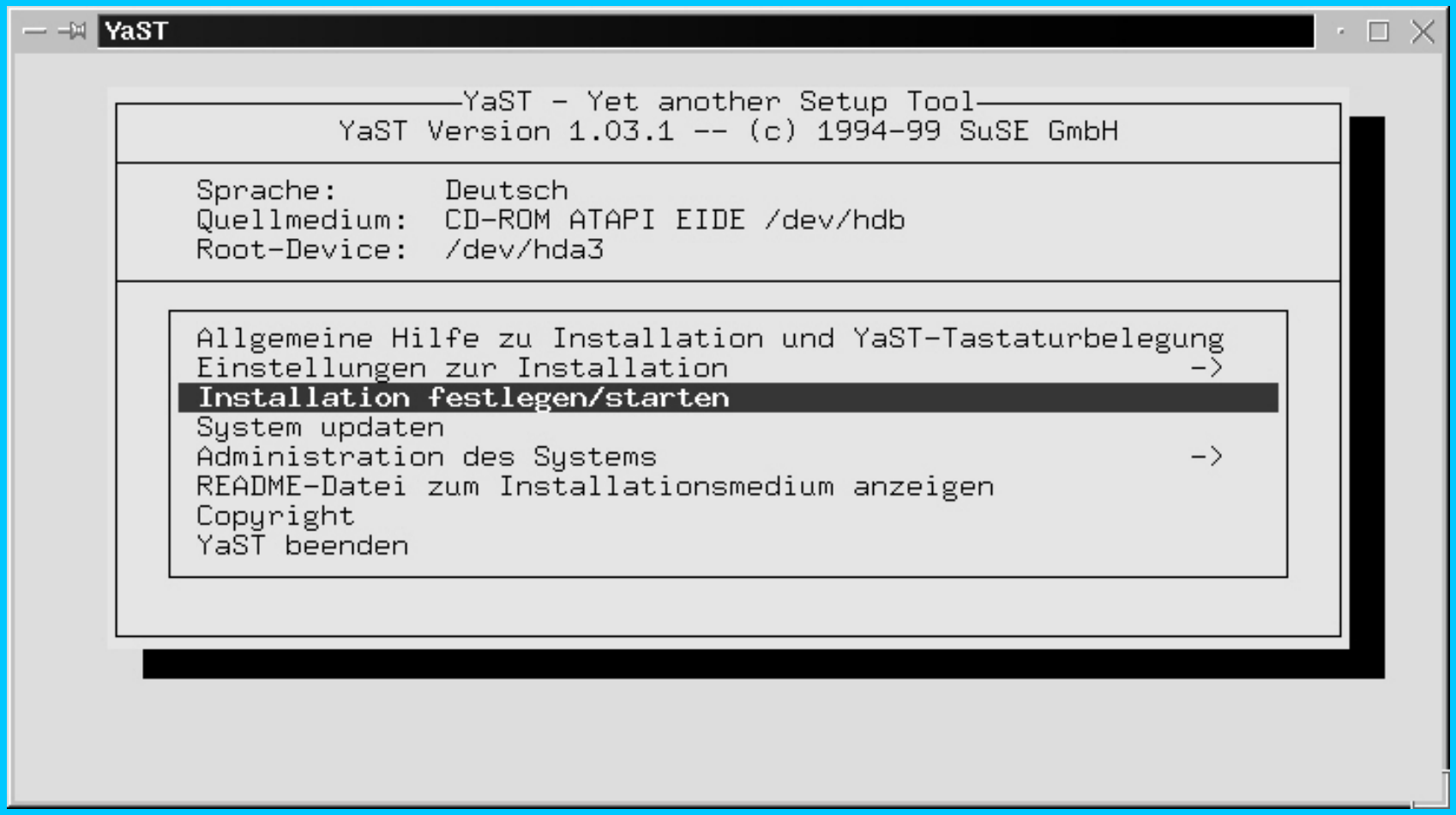
von Thomas Reuther

Wollen Sie vorerst nicht ganz so tief in die Kunst des Übersetzens der Sourcen einsteigen und erst einmal die bei SuSE-Linux mitgelieferte, schon übersetzte Version von Samba verwenden? Wenn Sie diese Frage für sich mit ja beantworten, dann sind Sie hier genau richtig! Ich werde Ihnen kurz darstellen, welche Schritte Sie durchführen müssen, um das Samba-Paket mit Hilfe des *YaST* zu installieren. Danach gebe ich Ihnen einen kurzen Überblick, wo sich die installierten Programme und Konfigurationsdateien befinden. Ich werde außerdem beschreiben, welche Schritte Sie vornehmen müssen, um swat zu aktivieren. Schließlich werde ich beispielhaft erläutern, wie Sie Benutzer anlegen und die `smb.conf` so anpassen können, dass Sie mit Ihrer Konfiguration schnell an das gewünschte Ziel gelangen.

Das Samba-Paket mit Hilfe des YaST installieren

Ich gehe davon aus, dass Sie Ihr SuSE-Linux-System installiert und bereits konfiguriert haben. Sie möchten nun das mitgelieferte Samba-Paket installieren. Melden Sie sich dazu als root an Ihrem System an und starten Sie den *YaST*. Legen Sie zunächst CD1 ein, wählen Sie dann den Menüpunkt *Installation festlegen/starten* aus und drücken Sie danach die Eingabetaste.

Abb. 3.4: Starten des YaST in einer Shell

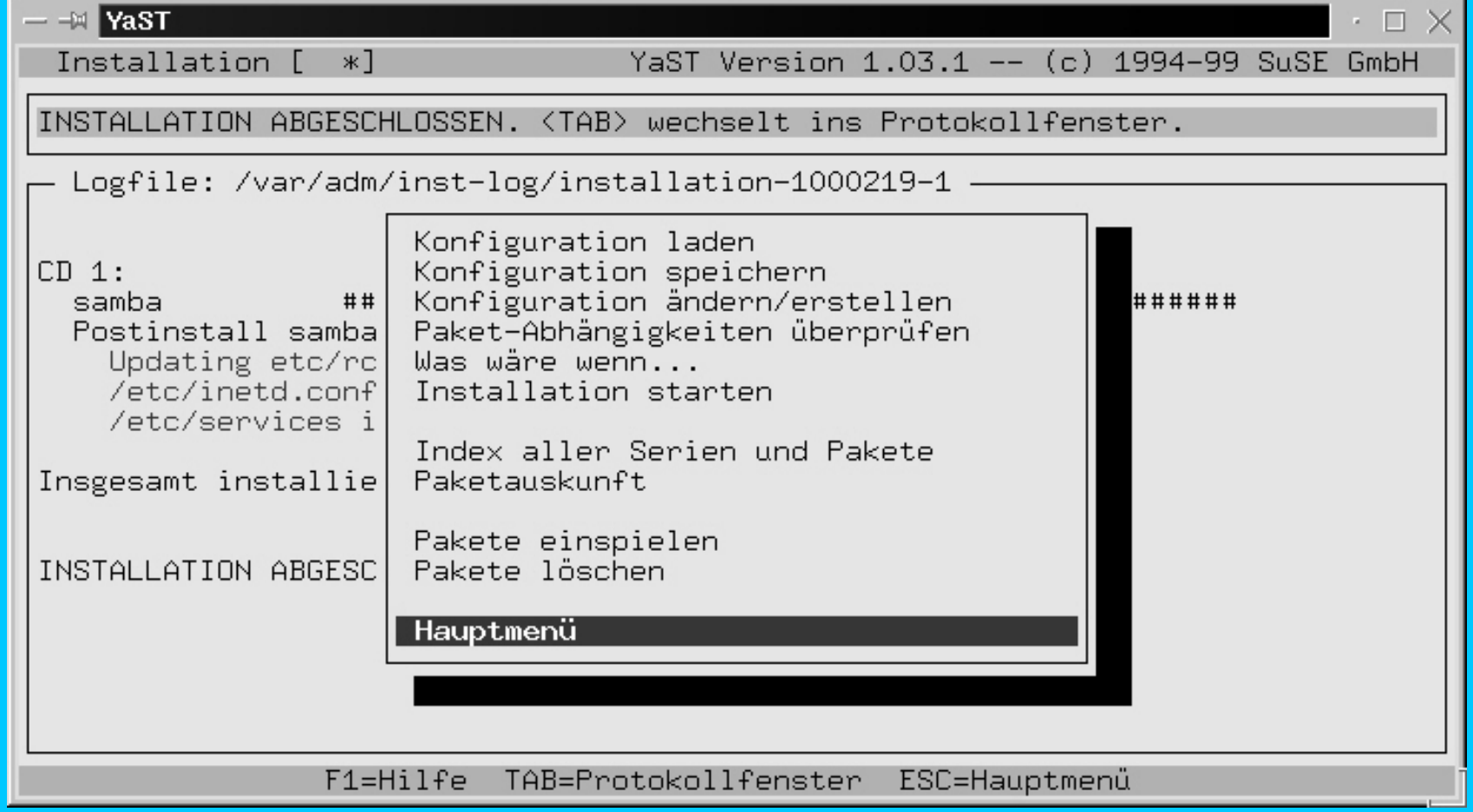


Wählen Sie den Menüpunkt *Konfiguration ändern/erstellen* aus und drücken Sie die Eingabetaste. Sie sollten nun in das Menü der Serienauswahl gelangt sein und eine ähnliche Darstellung sehen wie in Abbildung 3.5.

Abb. 3.5: Serienauswahl in SuSE-Linux

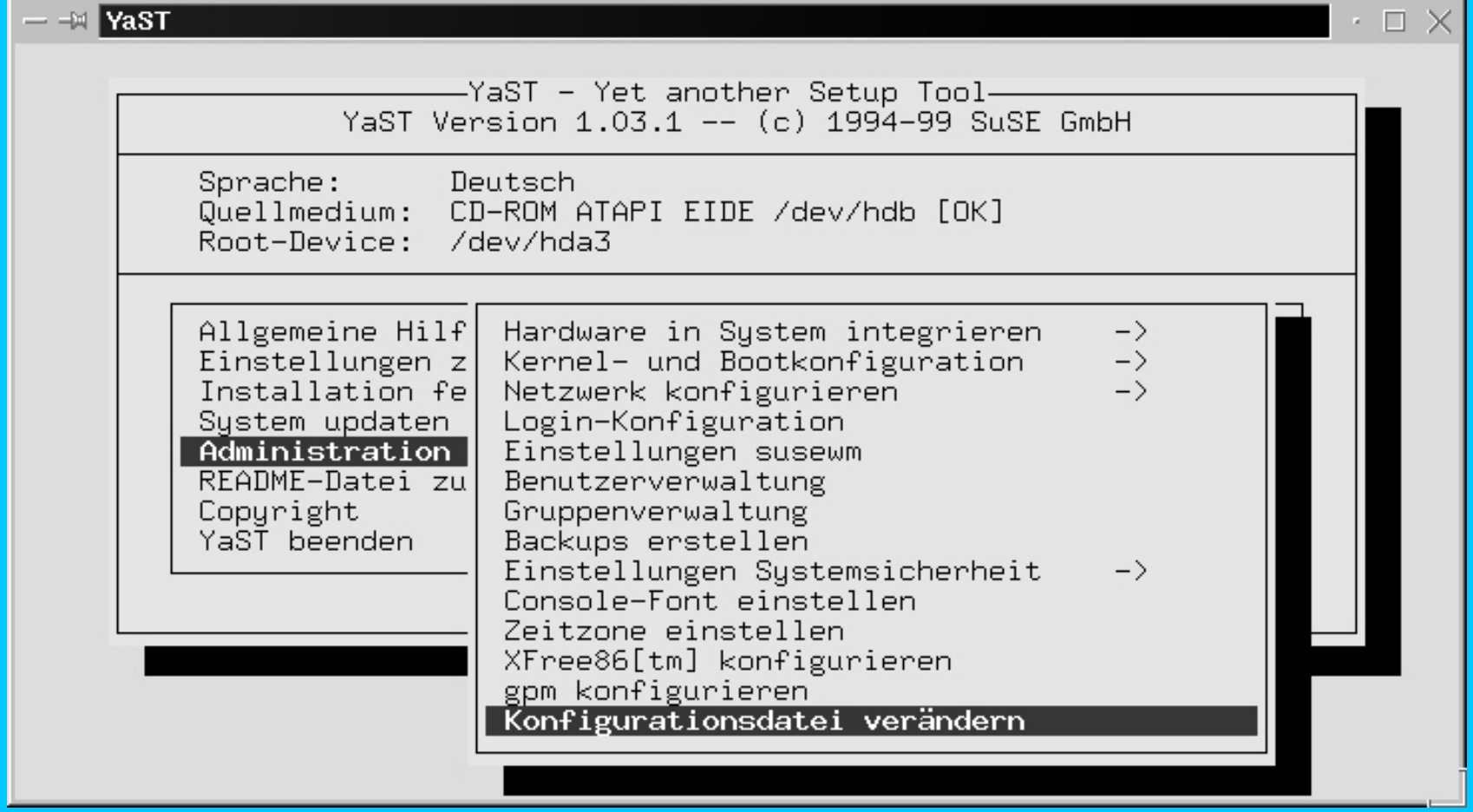
Das Samba-Paket ist in der Serie *n* enthalten. Wählen Sie diese Serie deshalb für die Installation aus. Suchen Sie nun mit den Pfeiltasten das Samba-Paket heraus und markieren Sie es mit Hilfe der Leertaste. Es sollte ein großes X vor dem Eintrag erscheinen. Drücken Sie nun so lange `[F10]`, bis der Menüpunkt *Installation starten* erscheint. Nachdem Sie diesen ausgewählt haben, wird Samba auf Ihrem SuSE-Linux-System installiert und Sie können den Fortschritt der Installation mit verfolgen. Wenn alles erfolgreich verlaufen ist, sollten Sie eine ähnliche Darstellung wie in Abbildung 3.6 sehen.

Abb. 3.6: Ende der Paketinstallation



Wählen Sie nun den Menüpunkt *Hauptmenü* aus. *YaST* startet daraufhin *SuSEconfig* und das installierte Samba-Paket-System wird in Ihrem System eingebunden. Nach Bestätigung der Meldungen von *SuSEconfig* finden Sie sich im Hauptmenü vom *YaST* wieder. Wählen Sie nun den Menüpunkt *Administration des Systems* und dann den Punkt *Konfigurationsdatei verändern* aus (Abb. 3.7).

Abb. 3.7: Konfigurationsdatei verändern auswählen



Als Nächstes können Sie menügeführt die `/etc/rc.config` editieren und die Startvariable `START_SMB` auf *yes* setzen (Abb. 3.8).

Abb. 3.8: Ende der Paketinstallationl

Damit werden zukünftig beim Start Ihres Linux-Systems auch automatisch smbd und nmbd gestartet. Beenden Sie nach dem Ändern die Maske mit `[F10]` und verlassen Sie dann den *YaST*. Sie haben nun bereits das Samba-Paket auf Ihrem SuSE-Linux-System installiert und dafür gesorgt, dass das Startskript für die beiden Samba-Daemons beim Systemstart ausgeführt wird. Als Nächstes werde ich Ihnen nun darstellen, wo Sie die Konfigurationsdateien und Samba-Tools finden und wie Sie vorgehen müssen, um swat zu starten.

Standort der Konfigurationsdateien

Nun wird es Zeit sich einmal anzuschauen, wo *YaST* die beiden Samba-Daemons, das Startskript und die Konfigurationsdatei `smb.conf` abgelegt hat. Es gibt also bereits eine Konfigurationsdatei von Samba, die Sie im Grunde nur noch etwas modifizieren müssen, um Ihre persönlichen Freigaben zu erstellen. Welche Möglichkeiten Sie dabei haben, das lesen Sie jedoch am besten in den Kapiteln 5, 6, 7 und 8 dieses Buches nach. Nun aber zu den Dateien im Einzelnen: Die zentrale Konfigurationsdatei `smb.conf` ist im Verzeichnis `/etc` zu finden. Dort existiert außerdem auch schon ein Dummy für eine `smbpasswd`-Datei für den Fall, dass Sie verschlüsselte Passwörter verwenden möchten oder müssen. Genaueres zur Passwortverschlüsselung finden Sie im Kapitel 6.2. Merken Sie sich hier einfach, dass Sie den `smbpasswd`-Befehl direkt in der Shell ausführen können und dass Ihre damit erzeugten Passwörter verschlüsselt in der `/etc/smbpasswd` zu finden sind. Somit sollte auch klar sein, wie die Zugriffsrechte auf diese Datei zu

setzen sind. Ein `ls -al /etc/smbpasswd` liefert schnell die Erkenntnis, dass auch hier schon vernünftige Einstellungen vorgenommen wurden, es sei denn, Sie haben Ihr System anders organisiert und müssen nachbessern. Bleibt nun noch die Frage, wo sich der smbd und nmbd sowie das zugehörige Startskript befinden. Die beiden Daemons liegen unter `/usr/sbin` und das Startskript findet sich unter `/sbin/init.d/` und trägt den Namen `smb`.

Nach einer kurzen Analyse findet man schnell heraus, dass es für die Runlevel 2 und 3 ausgeführt wird. Ich denke, dies ist auch in Ihrem Sinne, ansonsten können Sie an den geeigneten Stellen nachkonfigurieren.

Nun sollten Sie einen Überblick gewonnen haben, wie die Installation von Samba unter SuSE-Linux vorgenommen wird. Sie sollten nun auch in etwa wissen, wo sich die Konfigurationsdateien und Tools auf Ihrem System befinden. Somit ist es jetzt an der Zeit, *swat* unter SuSE-Linux zu aktivieren.

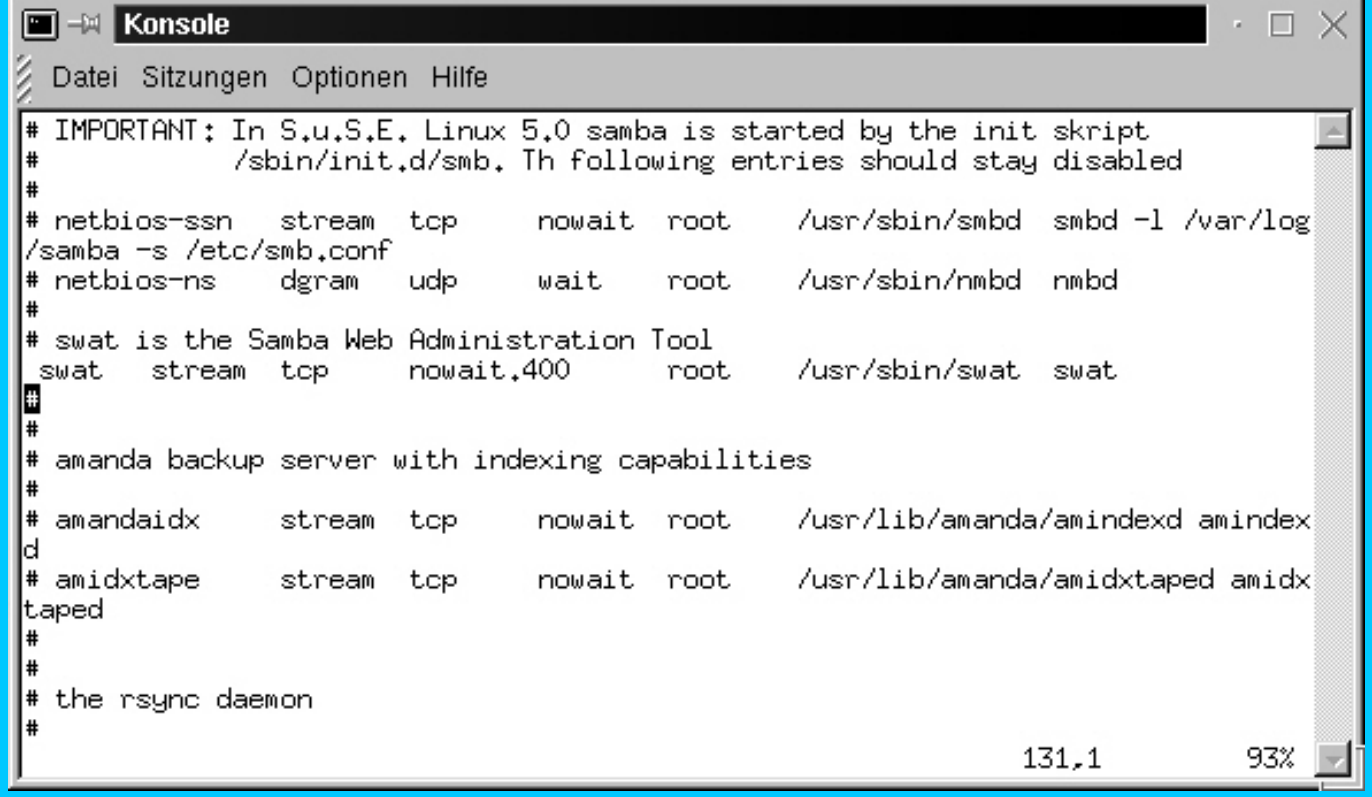
swat unter SuSE-Linux aktivieren

Eine ausführliche Beschreibung zum grafischen Konfigurationstool *swat* erhalten Sie im Kapitel 9 dieses Buches. Darüber hinaus finden sich dort neben *swat* noch einige weitere interessante Administrationstools. Wenn Sie die Sourcen der Tools benötigen, schauen Sie am besten im Internet oder auf der Buch-CD nach. Ich möchte auch hier noch einmal darauf hinweisen, dass bei diesen Tools zur Authentifizierung Klartextpasswörter verwendet werden. Sie sollten somit gegebenenfalls entsprechend vorsichtig damit sein, es sei denn, Ihnen gefällt die Vorstellung, dass irgendjemand (nicht Sie) Ihre Samba-Server aus dem Internet fernbedient.

Gehen wir zunächst einmal davon aus, dass Sie sich darüber keine Gedanken machen müssen und nun swat aktivieren wollen. Führen Sie dazu die folgenden Schritte durch:

Öffnen Sie die `/etc/inetd.conf` in einem Editor Ihrer Wahl und entfernen Sie die #-Zeichen in den Zeilen mit *swat* .... wie in Abbildung 3.9 zu sehen ist. Achten Sie außerdem darauf, dass der Eintrag nach `/usr/sbin/swat` verweist und dass dort auch *swat* zu finden ist.

Abb. 3.9: Editieren der /etc/inetd.conf



Nun ist es an der Zeit, Ihre Konfiguration ein wenig zu testen. Starten Sie am einfachsten Ihr Linux-System neu oder wechseln Sie die Runlevel entsprechend.

Wenn Sie nun Ihren Web-Browser starten, die IP-Adresse Ihres Samba-Servers angeben und sich mit Port 901 verbinden, sollten Sie *swat* so benutzen können, wie in Kapitel 9 ausführlich beschrieben wird.

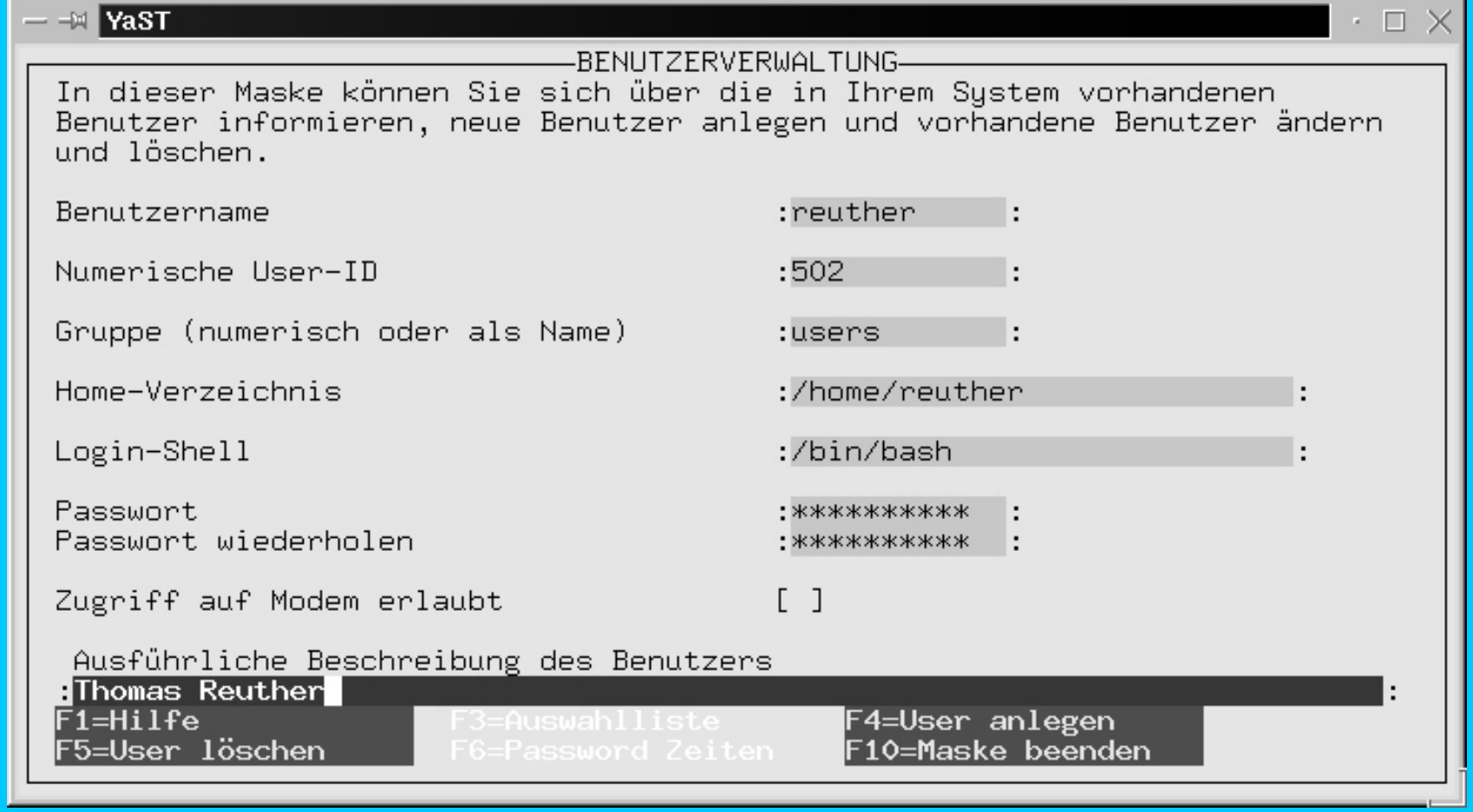
Als Nächstes möchte ich Ihnen kurz exemplarisch darstellen, wie Sie die Benutzerverwaltung mit Hilfe des *YaST* vornehmen können, um so Ihren Kollegen den Zugang zu Ihrem frisch installierten Samba-Server zu ermöglichen.

Benutzerverwaltung bei SuSE-Linux mit YaST

Nachdem Sie jetzt Samba konfiguriert haben, benötigen Sie nun auch noch Benutzer, die auf den Samba-Server zugreifen sollen. Zugegeben, eine Benutzerverwaltung mit verschiedenen Benutzern, Gruppen und deren Zugriffsrechten kann sehr schnell auch sehr kompliziert werden. Hier möchte ich deshalb nur darstellen, wie man prinzipiell einen Benutzer einrichtet, der sich dann von einem Windows-Client aus mit dem Samba-Server verbinden kann. Dies und vieles mehr können Sie einfach mit Hilfe des *YaST* auf Ihrem System einrichten, indem Sie die folgenden Schritte durchführen:

Starten Sie den *YaST*. Wählen Sie den Menüpunkt *Administration* des Systems und dann den Menüpunkt *Benutzerverwaltung*. Sie gelangen nun in eine Eingabemaske, in der Sie Ihre Benutzer auf Ihrem Linux-System einrichten können (Abb. 3.10).

Abb. 3.10: Benutzerverwaltung mit dem YaST



Nachdem Sie mit `[F4]` den Benutzer angelegt haben, können Sie weitere Benutzer hinzufügen, bevor Sie den *YaST* wieder verlassen. Wenn Sie Klartextpasswörter verwenden wollen, können Sie direkt mit dem letzten Schritt beginnen. Wollen oder müssen Sie jedoch Passwörter verschlüsseln, sollten Sie nun zunächst den `smbpasswd`-Befehl anwenden (vgl. Kapitel 6 dieses Buches).

Anpassen der /etc/smb.conf

Als letzten Schritt müssen Sie noch die `smb.conf` anpassen. Erinnern Sie sich noch? Ich sagte zu Anfang, dass sich bereits eine vorkonfigurierte `smb.conf` im Verzeichnis `/etc` auf Ihrem System befindet. Diese muss natürlich noch an die Gegebenheiten Ihres Netzwerks angepasst werden. Da die Möglichkeiten dieser Anpassung so vielfältig sind, dass ein sehr großer Teil dieses Buches sich damit beschäftigt, werde ich hier nur das mitgelieferte Beispiel darstellen und bei wichtigen Einstellungen auf die Buchkapitel verweisen. Dies soll Ihnen nur ein wenig beim Einstieg helfen und Ihnen Mut machen, mit Hilfe der anderen Kapitel in diesem Buch an der `/etc/smb.conf` herumzuexperimentieren. Die Flexibilität von Samba werden Sie erst schätzen lernen, wenn Sie versuchen, Ihre individuellen Lösungen für Ihr Netzwerk zu finden.

Hier nun die versprochene, mit Verweisen kommentierte `/etc/smb.conf`:

```
[global]        ; Beginn des globalen Bereichs
```

```
   workgroup = chipsndips          ; Arbeitsgruppenname
   guest account = nobody
   keep alive = 30
   os level = 2
   security = user                 ; Siehe Kap. 6

; Uncomment the following, if you want to use an existing
; NT-Server to authenticate users, but don't forget that
; you also have to create them locally!!!
;  security = server
;  password server = 192.168.1.10
encrypt passwords = yes            ; Siehe Kap. 6

   printing = bsd         ; Siehe Kap. 8
   printcap name = /etc/printcap
   load printers = yes

   socket options = TCP_NODELAY    ; Siehe Anhang B

   map to guest = Bad User


; Uncomment this, if you want to integrate your server
; into an existing net e.g. with NT-WS to prevent nettraffic
;  local master = no

; Please uncomment the following entry and replace the
; ip number and netmask with the correct numbers for
; your ethernet interface.
   interfaces = 192.168.1.100/255.255.255.0  ; Siehe Kap. 18,19,20

; If you want Samba to act as a wins server, please set
; 'wins support = yes'            ; Siehe Kap. 18
   wins support = no

; If you want Samba to use an existing wins server,
; please uncomment the following line and replace
; the dummy with the wins server's ip number.
;   wins server = 192.168.1.1      ; Siehe Kap. 18

; Do you wan't samba to act as a logon-server for
```

```
; your Windows 95/98 clients, so uncomment the
; following:

;   logon script =%U.bat           ; Siehe Anhang A
;   domain logons = yes
   domain master = yes
; [netlogon]
;   path = /netlogon

      ; Ende des globalen und
       Beginn des lokalen Bereichs
[homes]                            ; Siehe Kap. 7
   comment = Homeverzeichnis
   browseable = no
   read only = no
   create mode = 0750

; The following share gives all users access to the Server's CD drive,
; assuming it is mounted under /cd. To enable this share, please remove
; the semicolons before the lines
;
; [cdrom]                          ; Siehe Kap. 7
;   comment = Linux CD-ROM
;   path = /cd
;   read only = yes
;   locking = no

[printers]                         ; Siehe Kap. 8
   comment = All Printers
   browseable = no
   printable = yes
   public = no
   read only = yes
   create mode = 0700
   directory = /tmp
```

Ich hoffe, Ihnen mit meinen kurzen Ausführungen den Einstieg in die Installation und Konfiguration des Samba-Servers unter SuSE-Linux etwas erleichtert zu haben. Wenn Sie neugierig geworden sind und mehr wissen wollen, zögern Sie nicht, die Kapitel durchzuarbeiten, die Sie weiter interessieren! Ich wünsche Ihnen bei der Arbeit viel Spaß und Erfolg!

Tag 4: Installation und Testen der Konfiguration

In diesem Kapitel begleite ich Sie durch die Konfiguration und Installation eines Beispiel-Samba-Servers und beschreibe alle nötigen Vorbereitungen.

Stellen Sie sich vor, mein Chef gibt mir folgende Anweisungen:

Problem:

> »Ich brauche bis heute mittag drei Dinge von dir. Erstens eine Methode, über die ich Dokumente und Projektkalkulationen mit Joe Underling vom anderen Ende des Flurs gemeinsam nutzen kann. Zweitens stürzt mein PC dauernd ab. Mit diesem Band-Ding dauert es zu lange, bis ich alle meine wichtigen Dokumente wieder neu geladen habe. Ich brauche eine bessere Methode, um meine Dateien zu schützen. Und schließlich muss ich auf dem Drucker meiner Sekretärin nebenan drucken können.«

Szenario 1:

> »Hmmmm...«, denke ich. »Beide Rechner sind vernetzt. Ich könnte einen FTP-Server einrichten, um das Problem der gemeinsam genutzten Dokumente zu lösen. Da ich über die Benutzung von FTP nachdenke, könnte ich meiner Chefin zeigen, wie sie alle ihre Dateien auf eine Festplatte auf dem Server hochladen kann. Das wäre schneller als ein Bandlaufwerk. Um ihren PC mit dem Drucker zu verbinden, könnte ich wohl einen automatischen Umschalter besorgen und ein langes Parallelkabel an der Wand entlang bis zum Schreibtisch der Sekretärin verlegen. Hups. Zu lang für ein Parallelkabel. Ich könnte auch ein serielles Kabel benutzen. Nein. Das wäre zu langsam. Ich mache mich auf den Weg und denke, dass ich anfangen sollte, meine Bewerbungsunterlagen auf den neuesten Stand zu bringen.

Szenario 2:

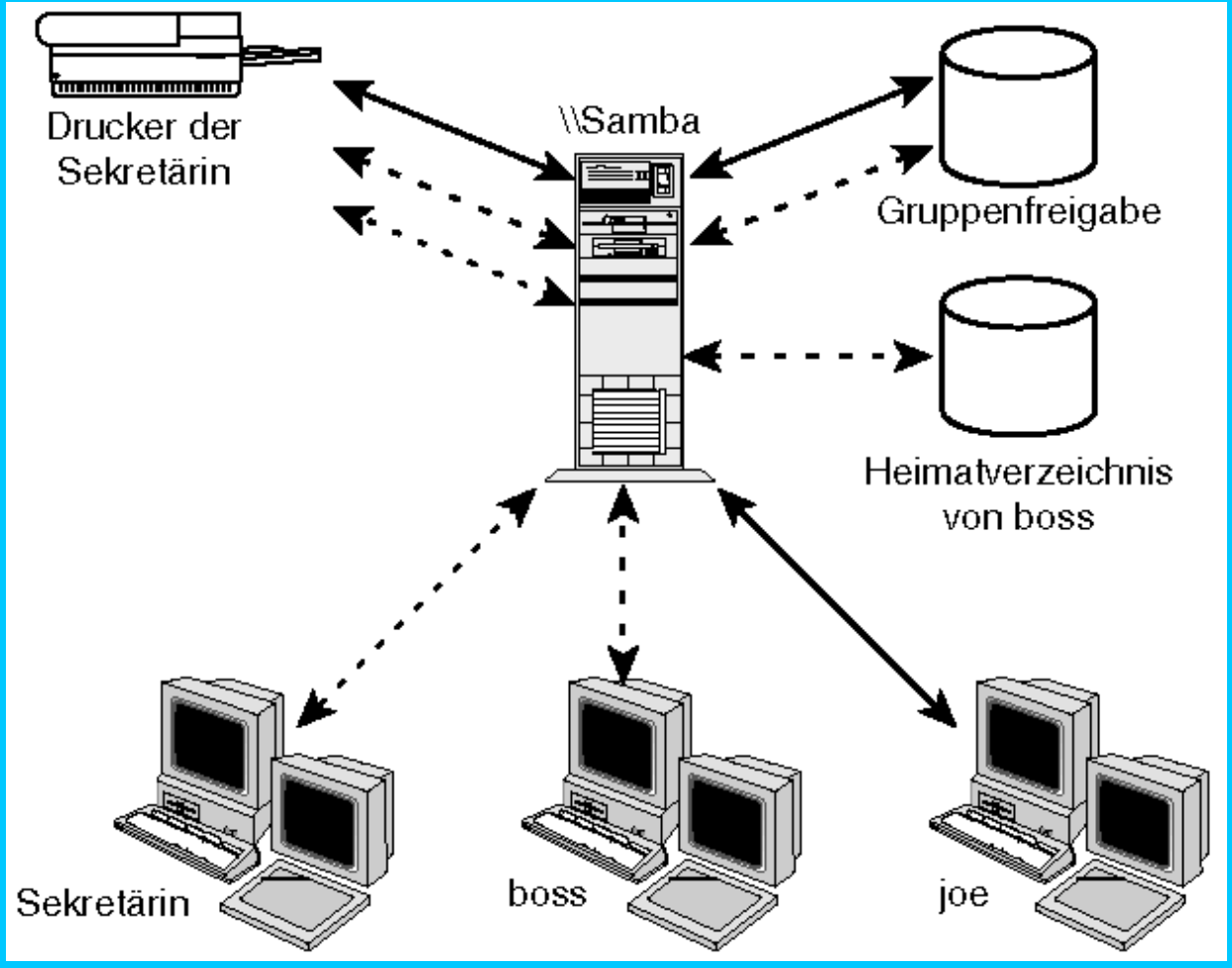
> Als erstes stelle ich fest, dass alle drei betroffenen PCs - der meiner Chefin, der der Sekretärin und Joes - mit dem lokalen Netzwerk verbunden sind. Danach gehe ich in mein Büro und suche nach einem externen Druckserver für den Drucker. Gefunden! Jetzt verbinde ich den Drucker mit dem Netzwerk. Danach logge ich mich in meinen Linux-Rechner ein, den ich gestern mit Hilfe meines Exemplars des Buchs »Jetzt lerne ich Linux« von Markt&Technik eingerichtet habe. Den aktuellen Samba-Source-Code habe ich bereits bei meiner allmorgendlichen Tasse Kaffee (schwarz, ohne Milch und Zucker) heruntergeladen und kompiliert.
>
> Ich richte eine einfache Konfigurationsdatei für Samba ein. Zuerst erstelle ich ein Home-Verzeichnis für meine Chefin auf dem Server, damit sie dort ihre wertvollen Dateien speichern kann. Dann richte ich eine Freigabe ein, auf die Joe Underling und meine Chefin zugreifen können, damit sie Dokumente gemeinsam nutzen können. Und schließlich erstelle ich eine Freigabe für den Drucker der Sekretärin und mache mich auf den Weg zu meiner Chefin, um ihr zu zeigen, wie sie all das benutzen kann.
>
> Nachdem ich die zwei Freigaben auf den Computer meiner Chefin gemountet und ihr gezeigt habe, wie sie auf die Dateien zugreifen kann, drucke ich eine Testseite, um sicherzugehen, dass der Drucker funktioniert. Meine Chefin ist so glücklich, dass sie nichts Neues lernen muss (außer, sich die Laufwerke H: und G: zu merken, für Home bzw. Gruppe), dass sie mir für den Rest der Woche frei gibt und eine Gehaltserhöhung noch dazu.

Vielleicht sind die Berichte über meine Notlage und meinen Triumph etwas übertrieben, aber man kann nie wissen. Es könnte so passieren!

In den restlichen Abschnitten dieses Kapitels führe ich Sie durch die Konfiguration von Samba, um die in Szenario 2 beschriebene Lösung zu implementieren. Linux diente hierbei nur als Beispiel. Die Lösung wird immer funktionieren, solange sich Samba unter dem Betriebssystem des Servers kompilieren lässt.

Abbildung 4.1 stellt dar, welcher Benutzer auf welche Freigaben zugreifen kann. Die gepunkteten Linien stehen für die Sekretärin, die gestrichelten für meine Chefin und die gefüllten für Joe Underling. Die Sekretärin sollte also nur auf den freigegebenen Drucker zugreifen können, Joe auf den Drucker und das Gruppen-Netzwerklaufwerk und meine Chefin auf beide vorher genannten Freigaben und zusätzlich auf ihr Home-Verzeichnis.

Abb. 4.1: Samba-Lösung für die Implementierung von (a) einer Gruppenfreigabe, (b) Home-Verzeichnissen und (c) einem Netzwerkdrucker



Welche Prozesse?

In Kapitel 3, »Wie bekomme ich den aktuellsten Source-Code«, habe ich erwähnt, dass ein Samba-Server aus zwei Haupt-Binaries besteht. Das erste ist `smbd`, das sich normalerweise in `/usr/local/samba/bin` befindet. `smbd` bearbeitet Anfragen für Datei- und Druckerfreigaben. Das zweite Binary, `nmbd`, ist normalerweise im gleichen Verzeichnis wie `smbd` zu finden. `nmbd` bearbeitet Anfragen für den NetBIOS-Name-Service und die Netzwerk-Browsing-Funktionen.

Den Abschnitt [global] in smb.conf konfigurieren

Bevor ich tatsächlich die Netzwerkfreigaben konfiguriere, die zur Lösung des Problems benötigt werden, muss ich einige Starteinstellungen konfigurieren, um Samba zum Laufen zu bringen. Sie sollten wissen, dass freigegebene Ressourcen in der SMB-Terminologie, egal ob es sich um Verzeichnisse oder Drucker handelt, als *Freigaben (Shares)* bezeichnet werden. Diese entsprechen einem NFS-exportierten Verzeichnis oder einem entfernten Drucker, der über `lpr` zur Verfügung gestellt wird. Manchmal wird eine SMB-Freigabe auch als *Dienst* bezeichnet. Ich benutze beide Begriffe austauschbar.

Zunächst muss ich folgende Fragen beantworten:

- Welchen Namen wird der Samba-Server haben?
- Welcher Arbeitsgruppe wird der Samba-Server angehören?

Nehmen wir für dieses Beispiel an, dass der Host-Name des Servers `eagle` ist und daher soll der NetBIOS-Name des Servers ebenfalls `EAGLE` lauten. Zwar sind die Namen nicht groß-/kleinsensitiv, aber ich werde in diesem Buch der Konvention folgen, dass ich DNS-Namen mit kleinen Buchstaben und NetBIOS-Namen mit Großbuchstaben darstelle. Die Arbeitsgruppe wird die gleiche sein, der auch der PC

angehört, also FOWLPLAY. In Kapitel 5, »Die Datei smb.conf: Samba mitteilen, was es tun soll«, finden Sie detaillierte Informationen über diese Entscheidungen und die entsprechenden smb.conf-Parameter.

Nachdem ich diese zwei Entscheidungen getroffen habe, kann ich die Starteinstellungen in der Datei smb.conf konfigurieren. Kommentaren in smb.conf wird das Zeichen ; oder # vorangestellt. Dabei ist es egal, welches der Zeichen verwendet wird. Das Erste kommt aus der Unix-Welt, während das Zweite aus der Windows-Welt stammt.

```
; globaler Parameterabschnitt von smb.conf
[global]
    ; den Netbios-Rechnernamen für den Server einrichten
    netbios name = EAGLE
    ; die Arbeitsgruppenzugehörigkeit einrichten
    workgroup = FOWLPLAY
    ; Samba für die Authentifizierung im User-Modus einrichten
    security = user
```

Die Kommentare erklären sich weitestgehend selbst. Die Zeile security = user bestimmt den Sicherheitsmodus, den Samba für die Authentifizierung von Benutzern verwendet. Weitere Informationen zur Benutzerauthentifizierung finden Sie in Kapitel 6, »Sicherheitsmodi und Passwörter«.

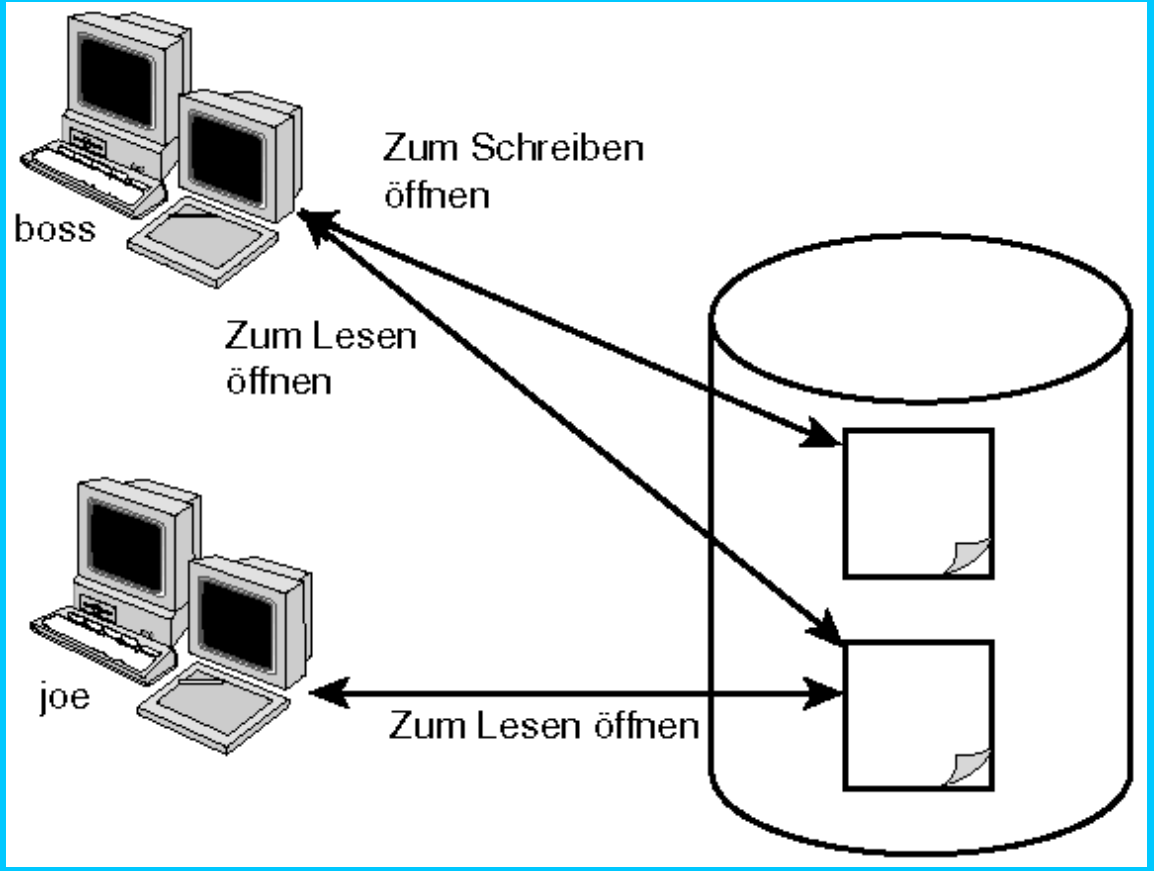
In Kapitel 3 habe ich bereits erwähnt, dass sich smb.conf standardmäßig in /usr/local/samba/lib/ befindet, und auch ich verwende diesen Standort. Über meinen bevorzugten Texteditor (vi) erstelle ich /usr/local/samba/lib/smb.conf und gebe den Text aus dem oben stehenden Code-Listing ein.

Das freigegebene Gruppenverzeichnis einrichten

Den ersten Teil des Problems, den ich angehe, ist die Konfiguration eines Netzwerklaufwerks, das mehreren Benutzern Lese- und Schreibzugriff bietet. Abbildung 4.2 stellt eine Beispielsituation dar, in der die Benutzer joe und boss auf Dateien auf der Festplattenfreigabe auf dem Server zugreifen sollen, die rechts dargestellt ist. Das Diagramm zeigt die Möglichkeit, dass beide Benutzer eine Datei gleichzeitig lesen können, aber ich erlaube jeweils nur einem Benutzer, eine Datei zum Bearbeiten zu öffnen.

Bevor ich die Verzeichnisse einrichte, auf die Joe und der Boss zugreifen können, muss ich zunächst sicherstellen, dass beide einen gültigen Account in /etc/passwd (oder im Falle von NIS oder NIS+ deren Netzwerkentsprechung) haben. Dann kann ich in /etc/group (oder der entsprechenden Netzwerk-Map) eine Gruppe einrichten und die zwei Accounts als Mitglieder hinzufügen. Für dieses Beispiel nenne ich die Gruppe boss1 und verwende die Benutzernamen joe und boss.

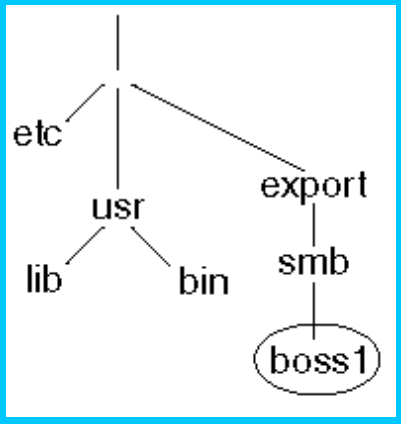
Abb. 4.2: Mehrere Benutzer greifen auf Dateien in einem freigegebenen Verzeichnis zu



Dieser Schritt wäre identisch, wenn ich eine Verzeichnisfreigabe für den Zugriff über Unix einrichten würde, da Samba es dem zugrunde liegenden Betriebssystem ermöglicht, den Zugriff auf die Dateien zu kontrollieren. Daher kann ich ein Sicherheitsmodell benutzen, mit dem ich vertraut bin, auch wenn die Windows-Clients keine Ahnung von Unix haben.

Ich benutze in der Regel das Verzeichnis `/export/` für alle Festplattenfreigaben, also halte ich mich auch hier daran. Ich richte das Verzeichnis `/export/smb/boss1/` als Verzeichnisfreigabe ein. Abbildung 4.3 stellt einen Teil der Verzeichnisstruktur des Servers dar. Das Verzeichnis, das über SMB freigegeben wird, ist eingekreist.

Abb. 4.3: Unix-Verzeichnisstruktur und der Teil, der über Samba freigegeben wird



Danach richte ich über die folgenden Befehle den Gruppeneigentümer und die Berechtigungen ein:

```
chgrp boss1 /export/boss1
chmod 770 /export/boss1
chmod g+s /export/boss1
```

Der letzte Schritt besteht darin, die Freigabe in der Konfigurationsdatei von Samba zu definieren, indem `smb.conf` folgende Einträge hinzugefügt werden:

```
; Freigabname
[boss1]
   ; aufzulistender Text, wenn die Freigabe von einem Client
   ; über Browsing angesehen werden kann
```

```
   comment = Freigegebenes Verzeichnis für die Gruppe boss1
   ; absoluter Pfad zum Festplattenverzeichnis
   path = /export/boss1
   ; soll die Freigabe beschreibbar sein?
   writable = yes
   ; Benutzer, die sich mit der Freigabe verbinden können sollen
   das @ bezeichnet eine Unix-Gruppe
   valid users = @boss1
   ; Datei-Locking aktivieren?
   locking = yes
   ; Standard-Berechtigungsmaske für Dateierstellung
   create mode = 0660
   ; Standard-Berechtigungsmaske für Verzeichniserstellung
   directory mode = 0770
```

Ich habe auch hier Kommentare eingefügt, um die allgemeine Bedeutung jedes Abschnitts zu beschreiben. Es ist im Moment nicht nötig zu verstehen, welche Funktion jeder Abschnitt hat. Festplattenfreigaben werden ausführlich in Kapitel 7, »Dateifreigaben«, dargestellt.

Das Home-Verzeichnis eines Benutzers einrichten

Das Einrichten von Home-Verzeichnissen in Samba ist der Einrichtung der Gruppenfreigabe, die im vorigen Abschnitt dargestellt wurde, sehr ähnlich. Der wichtigste Unterschied besteht darin, dass ich die Regel durchsetzen möchte, dass sich nur der jeweilige Eigentümer mit einem bestimmten Home-Verzeichnis verbinden kann. Anders gesagt, ich möchte nicht, dass Joe auf das Home-Verzeichnis meiner Chefin zugreifen kann. Das wäre nicht so gut.

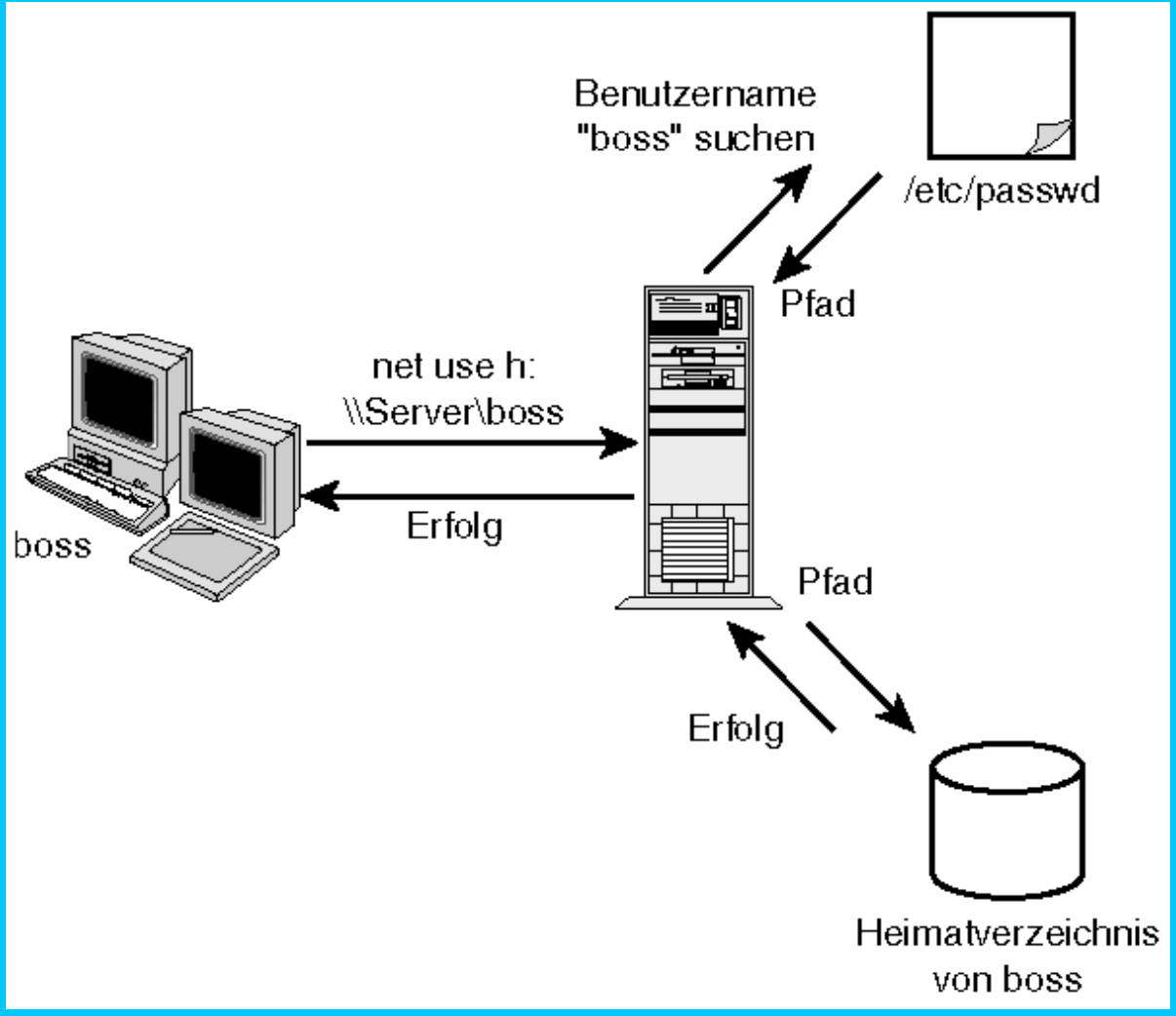
Als ich das freizugebende Gruppenverzeichnis eingerichtet habe, war es notwendig sicherzustellen, dass Joe und der Boss gültige Unix-Accounts auf dem Server haben. Normalerweise haben diese Unix-Accounts auch einen Bereich auf der Festplatte, der als das Home-Verzeichnis des Benutzers zugewiesen wird. Statt einen neuen Bereich zuzuweisen, konfiguriere ich Samba so, dass es das Home-Verzeichnis des Benutzers, das in der Unix-Account-Datenbank (d.h. `//passwd`) spezifiziert ist, gemeinsam benutzt. Hier sind die Parameter, die Sie in `smb.conf` einfügen müssen, um die Home-Verzeichnisse verfügbar zu machen:

```
; Freigabename
[homes]
   comment = Unix-Home-Verzeichnisbereich
   path = %H
   writable = yes
   valid users = %S
   create mode = 0600
   directory mode = 0700
   locking = no
```

Ich habe in diesen Abschnitt keine Kommentare eingefügt, da die meisten Einstellungen den Parametern für die Definition von `[boss1]` ähnlich sind. Es gibt einige Unterschiede, die Sie beachten sollten.

Der erste ist der Name der Freigabe oder des Dienstes, `[homes]`. Home-Verzeichnisse sind so etwas wie ein Spezialfall. Versucht ein Client, sich zu verbinden, wird der Freigabename `[homes]` durch den Benutzernamen ersetzt, den der verbindende Client überträgt. Das heißt, wenn meine Chefin versucht, sich mit ihrem Home-Verzeichnis zu verbinden, kann sie den Netzwerkpfad `\\EAGLE\homes` oder `\\EAGLE\boss` angeben. Samba interpretiert beide Angaben als die gleiche Freigabe. Abbildung 4.4 zeigt, wie Samba für jeden Benutzer während der Verbindungsaufnahme den `[homes]`-Dienst einrichtet. Existiert der Benutzer in `/etc/passwd`, benutzt Samba den Pfad für das Home-Verzeichnis, der in dem Benutzereintrag als Standort für den `[homes]`-Ordner angegeben ist.

Abb. 4.4: Samba erstellt und erweitert während der Laufzeit die Variable %H aus dem Home-Verzeichnis, das in /etc/passwd spezifiziert ist



Der zweite Unterschied ist der Eintrag `valid users = %S`. Die Variable `%S` wird während der Verbindungszeit in den Namen der Freigabe erweitert. Denken Sie daran, dass die `[homes]`-Freigabe während der Laufzeit in den vom Client übertragenen Benutzernamen umbenannt wird. Wenn Sie nur Verbindungen von einem Benutzer erlauben, der den gleichen Benutzernamen hat wie die Freigabe, verhindern Sie, dass Joe auf `\\EAGLE\boss` zugreifen kann. Diese Methode funktioniert auch dann, wenn die Berechtigungen für `/home/boss` auf allgemein beschreibbar (d.h. `rwxrwxrwx`) eingestellt sind.

Den Drucker einrichten

Nachdem ich die Festplattenfreigaben eingerichtet habe, ist es Zeit, mich dem Drucker zu widmen. Um das Beispiel fortzusetzen: ich habe den Drucker bereits korrekt auf dem Linux-Server eingerichtet. Wenn ich erfolgreich vom Linux-Rechner über den `lpr`-Befehl an den Netzwerkdrucker drucken kann, besteht der nächste Schritt darin, eine Druckerfreigabe einzurichten:

```
[global]
   printing = bsd

[printer1]
   comment = Drucker außerhalb des Büros
   printable = yes
   writable = no
```

Diese Definition veranlasst Samba, die Datei `/etc/printcap` nach einem Drucker namens `printer1` zu durchsuchen und den Standarddruckbefehl zu benutzen. Beachten Sie, dass ich auch den zusätzlichen Eintrag `printing = bsd` in den `[global]`-Abschnitt der `smb.conf` einfügen muss. Es gibt noch andere Arten von Unix-Drucksystemen. Samba kann auch mit diesen arbeiten, wie Sie in Kapitel 8, »Drucker«, sehen werden.

smb.conf überprüfen

Ich setze voraus, dass Ihre Binärdateien korrekt kompiliert und installiert wurden. Der nächste Schritt besteht darin, sicherzustellen, dass es keine Syntaxfehler in `smb.conf` gibt. Wenn Sie alle Samba-Utilities installiert haben (was automatisch der Fall ist, wenn Sie **make install** eingeben), sollten Sie ein Tool namens `testparm` im Verzeichnis `/usr/local//samba/bin` finden. Das Utility überprüft `smb.conf` und druckt alle Standardwerte aus, sofern diese nicht durch die Angabe anderer Parameter überschrieben wurden. Dies kann sehr hilfreich sein, um sicherzustellen, dass Samba das, was es Ihrer Meinung nach sehen sollte, auch sieht.

Ich habe im Abschnitt `[global]` der folgenden `smb.conf`-Datei das Wort *netbios* absichtlich falsch geschrieben.

```
; globaler Parameterabschnitt von smb.conf
[global]
   ; den Netbios-Rechnernamen für den Server einrichten
   netbis name = EAGLE
   ; die Arbeitsgruppenzugehörigkeit einrichten
   workgroup = FOWLPLAY
   ; Authentifizierung im User-Modus einrichten
   security = user
   printing = bsd
```

Dann habe ich `testparm` laufen lassen. Wenn Sie wollen, dass `testparm` eine `smb.conf`-Datei überprüft, die sich an einem anderen Ort als dem Kompilierungsstandard befindet, können Sie den Parameter `-s` *`Dateiname`* benutzen:

```
root# /usr/local/samba/bin/testparm -s smb.conf | head -6
Load smb config files from smb.conf
Unknown parameter encountered: "netbis name"
Ignoring unknown parameter "netbis name"
Processing section "[boss1]
Processing section "[homes]"
Processing section "[printer1]
```

Sie sehen, dass die Ausgabe einen unbekannten Parameter darstellt: `netbis name`.

Sie können `testparm` auch dazu benutzen, die Standardwerte von Parametern zu bestimmen. Die folgende Ausgabe zeigt z.B., dass der Standard-Gast-Account `nobody` ist.

```
root# /usr/local/samba/bin/testparm -s smb.conf| grep "guest account"
   guest account = nobody
```

smbd und nmbd starten

Nachdem Sie `smb.conf` fertiggestellt und die Samba-Binärdateien erzeugt haben (entweder durch Kompilieren des Source-Codes oder Herunterladen verfügbarer Binaries), besteht der nächste Schritt darin, die Samba-Daemons, `smbd` und `nmbd`, zu starten. Dafür gibt es zwei Möglichkeiten. Welche Sie wählen, hängt davon ab, wie viele Verbindungen zu Ihrem Samba-Server Sie erwarten, wie regelmäßig diese sind und wie viele Ressourcen Sie derzeit auf dem Server freigeben können.

Jede Client-Verbindung hat ihren eigenen `smbd`-Daemon. Der `smbd`-Prozess, der für einen Client verantwortlich ist, kann jedoch viele Verbindungen zu Freigaben verwalten. Jeder `smbd`-Prozess benutzt für seinen Arbeitsbetrieb etwa 1 Mbyte RAM, kann aber mehr Gesamtarbeitsspeicher zugewiesen bekommen. Der Arbeitsbetrieb eines Prozesses ist die Anzahl der Speicherseiten, die er im physikalischen Arbeitsspeicher behalten muss. Sie können ausrechnen, wie viel Gesamtarbeitsspeicher Samba allgemein belegen wird, indem Sie die Anzahl der gleichzeitigen Benutzer mit 1,5 multiplizieren. Einige Betriebssysteme benutzen Arbeitsspeicher-Mapping, damit Prozesse nicht modifizierbare Codeseiten gemeinsam nutzen können, um damit die Menge des tatsächlich benutzten Arbeitsspeichers zu reduzieren. Diese Formel gibt Ihnen zumindest einen Anhaltspunkt, um die Kapazität für Ihren bestimmten Server planen zu können.

Der Samba-Serverprozess kann entweder über den `inetd`-Metadaemon oder als Standalone-Daemon gestartet werden. Ich habe festgestellt, dass Freigaben bei Erstverbindungen schneller zur Verfügung stehen, wenn `smbd` und `nmbd` als Standalone-Daemon laufen. Wenn der Server aber für längere Zeit ohne SMB-Verbindungen inaktiv ist, können Sie `smbd` und `nmbd` über die `inetd.conf` laufen lassen, um die gesamte Arbeitsspeicherauslastung des Systems ein wenig zu reduzieren. Sie müssen entscheiden, welche Methode für Sie die beste ist, aber Sie können nur eine der beiden Methoden wählen.

Von inetd starten

Um Samba über den `inetd`-Daemon laufen zu lassen, müssen Sie zwei Dateien editieren. Falls Sie die SuSE-Linux-Distribution einsetzen, sollten Sie allerdings nicht zu dieser Variante greifen. Ich setze voraus, dass der Unix-Server weder NIS noch NIS+ benutzt, um Systemdateien über das Netzwerk zu verteilen.

Zuerst muss ich die folgenden Einträge in `/etc/services` einfügen. Stellen Sie sicher, dass es keine anderen Einträge für TCP Port 139 und UDP Port 137 gibt:

```
netbios-ssn      139/tcp
netbios-ns       137/udp
```

Danach fügen Sie mit Ihrem bevorzugten Texteditor die folgenden Einträge in der `/etc/inetd.conf` ein:

```
netbios-ssn stream tcp nowait root /usr/local/samba/bin/smbd smbd
netbios-ns dgram udp wait root /usr/local/samba/bin/nmbd nmbd
```

Nach Speichern der Änderungen muss ich dem `inetd`-Daemon mitteilen, die Konfigurationsdatei erneut zu lesen. Normalerweise erreiche ich dies, indem ich ein HUP-Signal an den Prozess sende. Hier sind die Befehle, die ich auf meiner Slackware-3.5.-Linux-Installation benutze:

```
root# ps -ax | grep inetd
102 ? S 0:00 /usr/sbin/inetd
root# kill -HUP 102
```

Als Daemon starten

Wird Ihr Samba-Server häufig Verbindungsanfragen erhalten, möchten Sie `smbd` und `nmbd` vielleicht als Daemon starten, um die Reaktionszeit zu verkürzen. Dies können Sie durch den Parameter `-D` erreichen, wenn Sie `smbd` und `nmbd` starten.

Unix-Derivate benutzen heute in der Regel eine von zwei Methoden, um Prozesse beim Booten zu starten. Eine wird als System-V-init-Methode bezeichnet und von Betriebssystemen wie Solaris, SuSE-Linux und RedHat Linux benutzt. Die zweite ist die ältere BSD-Methode, die auf Systemen wie SunOS und Slackware Linux zum Einsatz kommt.

System-V-init-Skripte

Zunächst stelle ich System-V-init-Skripte vor. Diese Skripte befinden sich normalerweise in `/etc/rc3.d` oder `/etc/init.d` mit Verknüpfungen von `/etc/rc#.d`, wobei # für einen bestimmten Runlevel steht. Sie sollten Samba in der Regel im Runlevel 3 starten. Die Skripte nehmen als einzelnes Befehlszeilenargument entweder `start` oder `stop` an, um den Dienst zu starten bzw. zu stoppen. Nachfolgend finden Sie ein Beispiel für ein Samba-Skript, das der System-V-Methode entspricht:

```
#!/bin/sh
#
# dieses Skript sollte wohl nur bei Runlevel 3 laufen
#
cd /etc
PATH=/usr/bin:/bin:/usr/sbin

case $1 in
'start')
   if [ -x /usr/local/samba/bin/smbd -a -f /etc/smb.conf ]; then
      echo 'Starting Samba...'
      /usr/local/samba/bin/nmbd -D
      /usr/local/samba/bin/smbd -D
   fi
   ;;
'stop')
   pid=`/bin/ps -x | egrep '(smbd|nmbd)' |sed -e 's/^ //' -e 's .*//'`
   if test "$pid"
   then
      kill $pid
      rm /usr/local/samba/var/locks/smbd.pid
      rm /usr/local/samba/var/locks/nmbd.pid
   fi
   ;;
*)
```

```
   echo "usage: /etc/init.d/samba {start|stop}"
   ;;
esac
```

Zum Starten der Samba-Daemons geben Sie **/etc/init.d/samba start** ein. Bedenken Sie, dass das Skript auf die Datei `smb.conf` zugreifen muss!

Wollen Sie die Prozesse stoppen, geben Sie den Befehl erneut ein, ersetzen jedoch **start** durch **stop**.

Start-Skripte in BSD-Manier

Die BSD-artigen Skripte bieten nicht die Flexibilität, sie starten und stoppen zu können, wie sie System-V-init-Skripte bieten, aber sie sind etwas einfacher zu handhaben, da es insgesamt weniger Skripte sind.

Wenn das System bootet, durchläuft es in der Regel einen Prozess, der dem folgenden ähnelt:

1. Startet `/etc/rc.S`, wenn der Single-User-Modus beginnt.
2. Startet `/etc/rc.M`, wenn der Multiuser-Modus beginnt.
3. `/etc/rc.M` wird normalerweise `/etc/rc.inet1` und `/etc/rc.inet2` starten, die dafür verantwortlich sind, die Netzwerk-Interfaces und -Dienste zu starten.
4. Schließlich wird `/etc/rc.local` gestartet, das alle lokalen System-Startprozesse beinhaltet.

Normalerweise wird der Samba-Serverprozess von `/etc/rc.local` gestartet. Hier ein Beispiel:

```
#!/bin/sh
#
# /etc/rc.local: Lokales System-Initialisierungsskript.
#
# Platziere alle lokalen Startbefehle hier:
if [-x /usr/local/samba/bin/smbd -a -f /etc/smb.conf ]; then
   echo 'Starting Samba...'
   /usr/local/samba/bin/nmbd -D
   /usr/local/samba/bin/smbd -D
fi
```

Es gibt Varianten mit geringen Abweichungen, die versuchen, System-V- und BSD-Startskripts zu kombinieren. Sie sollten dazu einen Blick in die Manpages und andere Dokumentationen zu Ihrem System werfen, um genau zu bestimmen, wo Sie die Startbefehle platzieren müssen.

Befehlszeilenargumente

In Kapitel 3 habe ich darauf hingewiesen, dass Sie die Standardkonfiguration von Samba durch Befehlszeilenargumente oder Parameter in der `smb.conf` ändern können. Die meisten `smb.conf`-Parameter werden in Kapitel 5 dargestellt. Dieser Abschnitt gibt einen kurzen Überblick über die meistbenutzten verfügbaren Befehlszeilenargumente. Die besten Nachschlagewerke für aktuelle Optionen sind die `smbd`- und `nmbd`-Manpages. Die Manpages für die Samba-Programmfamilie werden installiert, wenn Sie `make install` ausgeführt haben, um die Binärdateien an die richtige Stelle zu kopieren. Ist dies der Fall, können Sie über den Standardbefehl `man` auf die Manpages zugreifen.

Haben Sie keine Manpages installiert, finden Sie sie für Versionen vor 2.0 in der Samba-Distribution unter `docs/` und für die Version 2.0 oder höher unter `docs/manpages`. Um sich die Manpages anzusehen, suchen Sie die Dateien `smbd.8` und `nmbd.8` und führen den Befehl `nroff` aus:

```
nroff -man smbd.8 | more
```

Oder, alternativ (auf diese Weise können Sie besser scrollen):

```
nroff -man smbd.8 | less
```

Tabelle 4.1: smbd/nmbd-Befehlszeilenargumente

Option	Beschreibung
`-D`	Samba läuft als Daemon, d.h., der Prozess läuft im Hintergrund und reagiert auf Anfragen. Sie sollten `smbd` mit dieser Option als Datei-Server laufen lassen, der nicht nur auf unregelmäßige Anfragen antworten muss. Standardmäßig läuft `smbd` nicht als Daemon.
`-d` *`Debuglevel`*	Spezifiziert den Debug-Level, auf dem der Prozess laufen soll. Der Debug-Level ist eine Zahl zwischen 1 und 10.
`-l` *`Logdatei`*	Der Pfad zu der Datei, in die der Prozess Logeinträge schreiben soll.

`-s Konfigurationsdatei`	Der Pfad zur Konfigurationsdatei, die der Prozess benutzen soll.

Tabelle 4.1 listet einige häufig benutzte Optionen für `smbd` und `nmbd` auf. Eine komplette Auflistung finden Sie in den Manpages zum entsprechenden Prozess.

Die Installation testen

Bis hierher habe ich

- die Binärdateien für Samba kompiliert,
- die freizugebenden Verzeichnisse korrekt eingerichtet,
- Einstellungen für eine komplette `smb.conf`-Konfigurationsdatei eingegeben und
- entschieden, wie der Samba-Server starten soll.

Jetzt ist es an der Zeit zu überprüfen, ob ich bis hierhin Fehler gemacht habe. Ich werde auf einige übliche Fehler während der Einrichtung hinweisen und Ihnen Tips geben, wie Sie diese erkennen und korrigieren können.

Samba beinhaltet ein Utility namens `smbclient`, das ein FTP-ähnliches Interface für den Zugriff auf Samba-Server bietet. Tatsächlich wurde `smbclient` ursprünglich zum Testen von Samba verwendet. In diesem Fall ermöglicht es mir, den Samba-Server zu testen, ohne zusätzliche PCs benutzen zu müssen, die über ein Netzwerk mit dem Samba-Server verbunden sind. Obwohl `smbclient` nicht garantieren kann, dass es zu keinen Fehlern kommt, wenn die PC angeschlossen werden, sollte es grobe Fehler finden können.

Nachdem ich sichergestellt habe, dass die beiden Samba-Daemons laufen oder dass gültige Einträge vorhanden sind, um die Prozesse von `inetd.conf` zu starten, versuche ich eine Liste der Freigaben vom Server zu bekommen. Der Parameter `-L` *`NetBIOS-Name`* veranlasst `smbclient`, eine Liste der Freigaben von dem Server zu holen, der durch *`NetBIOS-Name`* spezifiziert ist. Ich benutze den Parameter `-N`, um die Aufforderung zur Eingabe eines Passworts zu vermeiden, da dies nicht notwendig ist, wenn ich nur die verfügbaren Freigaben sehen möchte:

```
root# /usr/local/samba/bin/smbclient -L EAGLE -N
added Interface ip=192.168.1.73 bcast=192.168.1.255 nmask=255.25.255.0
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0beta4]

Sharename    Type       Comment
---------    ------     --------
boss1        Disk       Freigegebenes Verzeichnis für Gruppe boss1
homes        Disk       Unix-Home-Verzeichnisbereich
printer1     Printer    Drucker außerhalb des Büros
IPC$         IPC        IPC-Service (Samba 2.0.0beta4)

Server               Comment
-------              -------

Workgroup            Master
---------            ------
FOWLPLAY
```

Alle drei von mir definierten Freigaben (`homes`, `boss1` und `printer1`) tauchen in der Liste der Freigaben auf. Die Freigabe `IPC$` wird automatisch erstellt und für das Browsing benutzt. Sie werden in späteren Kapiteln mehr darüber erfahren. Bis jetzt scheint alles in Ordnung zu sein.

Der nächste Schritt besteht darin, die individuellen Festplattenfreigaben zu überprüfen. Zunächst verbinde ich mich als `joe` mit der Freigabe `[boss1]`. Für die Verbindung mit einer Freigabe wird der Netzwerkpfad als *`//Servername/Freigabename`* dargestellt, wobei *`Servername`* und *`Freigabename`* durch die entsprechenden Werte ersetzt werden. Wenn ich mich in der bash-Shell unter einem anderen Benutzernamen verbinden will als dem, der in der Umgebungsvariable `$USER` gespeichert ist, muss ich den Parameter `-U` *`Benutzername`* verwenden:

```
root# /usr/local/samba/bin/smbclient //eagle/boss1 -U joe
added interface ip=192.168.1.73 bcast=192.168.1.255 nmask=255.255.255.0
Password:
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0beta4]
tree connect failed: ERRSRV - ERRinvnetname (Invalid network name in tree connect.)
```

Nach Eingabe des korrekten Passworts gibt `smbclient` die Fehlermeldung »Invalid network name in tree connect« zurück. In Kapitel 2,

»Windows-Netzwerke«, habe ich erwähnt, dass eine Verbindung zu einem Verzeichnisbaum eingegangen wird, nachdem der Client authentifiziert wurde und versucht, sich mit einer bestimmten Ressource zu verbinden. Nach einigen Überprüfungen finde ich heraus, dass ich statt `/export/boss1` den Verzeichnisnamen `/export/boss` angegeben habe und benenne es entsprechend um:

```
root# ls -l /export
total 1
drwxrws--- 2 root boss1 1024 Dec 29 09:50 boss/
root# mv /export/boss /export/boss1
root# ls -l /export
total 1
drwxrws--- 2 root boss1 1024 Dec 29 09:50 boss1/
```

Nachdem ich meinen Fehler korrigiert habe, versuche ich es erneut. Diesmal werde ich erfolgreich verbunden. Sie werden bemerken, dass die Hosts-Datei, die ich hochgeladen habe, mit den korrekten Berechtigungen sowie dem korrekten Gruppeneigentümer erstellt wurde. Ich kann eine ähnliche Methode benutzen, um die `[homes]`-Freigabe zu überprüfen:

```
root# smbclient //bilbo/boss1 -U joe
added interface ip=192.168.1.73 bcast=192.168.1.255 nmask=255.255.255.0
Password:
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0beta4]

smb: \> lcd /etc
the local directory is now /etc

smb: \> put hosts
putting file hosts as \hosts (101.073 kb/s) (average 101.074 kb/s)

smb: \> dir
   hosts                       621 Tue Dec 29 10:36:48 1998
   61967 blocks of size 4096. 15359 blocks available

smb: \> quit

root# ls -l /export/boss1
total 1
-rw-rw----   1 joe   boss1   621 Dec 29 10:36 hosts
```

Nachdem ich die Festplattenfreigaben getestet habe, muss ich nur noch den Netzwerkdrucker überprüfen. Der Prozess ist der gleiche, mit der Ausnahme, dass ich den Parameter `-P` benutze, um `smbclient` mitzuteilen, dass es sich mit der Freigabe als Drucker statt als Netzwerklaufwerk verbinden soll. Über den Befehl `put`, der eine Datei an einen Drucker lädt, wird `smbclient` informiert, die zu druckende Datei zu übertragen. Über den Befehl `lpq` kann ich mir die entsprechende Druckerwarteschlange ansehen, um sicherzustellen, dass Samba die Datei korrekt übertragen hat:

```
root# smbclient //bilbo/printer1 -P -U boss
Added interface ip=192.168.1.73 bcast=192.168.1.255 nmask=255.255.255.0
Password:
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0beta4]

smb: \> lcd /etc
the local directory is now /etc

smb: \> put hosts
putting file hosts as \hosts (101.073 kb/s) (average 101.074 kb/s)

smb: \> quit

root# lpq -P printer1
waiting for printer1 to become ready (offline ?)
Rank     Owner     Job     Files     Total Size
1st      boss      0       hosts     621 bytes
```

Andere in Samba integrierte Tools

Zusätzlich zu den bereits erwähnten Tools gibt es noch einige andere Utilities, die in der Samba-Distribution enthalten sind (siehe Tabelle 4.2). Diese finden Sie im Verzeichnis `/usr/local/samba/bin/`.

Tabelle 4.2: Andere Tools und Utilities in der Samba-Programmfamilie

Name des Tools	Beschreibung
`smbstatus`	Dieses Tool generiert Ausgaben über den aktuellen Status von Verbindungen und geschützten Dateien.
`nmblookup`	Dieses Utility fragt nach NetBIOS-Namensinformationen über TCP.
`testprns`	Dieses einfache Utility überprüft, ob ein Druckername als Freigabename benutzt werden kann, d.h., dass der Druckername in der `/etc/printcap` gefunden werden kann.
`make_smbcodepage`	Dieses Tool ermöglicht die Kompilierung und Dekompilierung von Codeseitendateien, die mit den Internationalisierungsfunktionen von Samba benutzt werden.
`smbtar`	Dieses Shell-Skript bietet eine Methode, Dateien über das Utility `smbclient` auf Unix-Bandlaufwerke auf entfernten SMB-Freigaben zu sichern und wiederherzustellen. Weitere Informationen über Backups finden Sie in Anhang B, »Tipps und Tricks«.
`smbrun`	Dieses kleine Zusatzprogramm wird von `smbd` benutzt, um Shell-Befehle auszuführen. Mit Zusatzprogramm meine ich, dass es als Interface agiert, damit `smbd` bestimmte Shell-Skripte oder andere Befehlszeilen-Tools starten kann.
`rpcclient`	Dieses Befehlszeilen-Tool, das zu `smbclient` analog ist, ermöglicht die Durchführung von DCE/RPC für Windows-NT- und Samba-Server. Weitere Informationen dazu finden Sie in Anhang A, »Experimentelle PDC-Unterstützung«.
`swat`	Dieses GUI-Administrationstool verwaltet die `smb.conf`-Datei. GUI-Administrationstools werden in Kapitel 9, »GUI-Administrationstools», ausführlicher dargestellt.
`smbpasswd`, `addtosmbpass`, `convert_smbpasswd`	Diese Tools bieten Methoden für die Manipulation der LanMan- und NT-Hashfunktionen, die in der Datei `private /smbpasswd` gespeichert sind, die von Samba benutzt wird, wenn die Verschlüsselung von Passwörtern aktiviert ist. Weitere Informationen über Verschlüsselung finden Sie in Kapitel 6.

`smbstatus` und `nmblookup` sind zwei der aufgelisteten Tools, die extrem nützlich sein können. Wir werden uns diese jetzt genauer ansehen und fangen mit `smbstatus` an.

Der `smbd`-Daemon für eine Verbindung läuft so lange als `root`, bis er für den verbundenen Benutzer eine Aufgabe ausführen muss. Dann wird die Prozess-UID zur Unix-UID des Benutzers und schaltet wieder zurück zu `root`. Dies kann zu Problemen führen, wenn es darum geht, welcher `smbd`-Prozess zu welchem Benutzer gehört. `smbstatus` gibt Informationen über aktuell eingeloggte Benutzer und geschützte Dateien aus. Die folgende Ausgabe zeigt, dass ich mit meinem Home-Verzeichnis verbunden bin und eine Datei zur Bearbeitung geöffnet habe:

```
root# smbstatus
Samba version 2.0.0beta4
Service      uid      gid      pid      machine
```

```
---------------------------------------------
jerryc     jerryc  users   472     queso (192.168.1.72) Tue Dec 29 11:31:05 1998

Locked files:
Pid     DenyMode    R/W      Oplock           Name
--------------------------------------------------
472     DENY_NONE   RDONLY   EXCLUSIVE+BATCHsrc/samba/source Makefile   Tue Dec 29 11:31:56 1998

Share mode memory usage (bytes):
   1048368(99%) free + 136(0%) used + 72(0%) overhead = 1048576(100%) total
```

Das zweite Tool, das ich oft benutze, ist `nmblookup`. Dieses Befehlszeilen-Utility bietet eine Methode, Informationen über NetBIOS-Namen abzurufen und die entsprechende IP-Adresse zu erhalten:

```
root# /usr/local/samba/bin/nmblookup eagle
Sending queries to 192.168.1.255
192.168.1.73 eagle<00>
```

`nmblookup` sucht standardmäßig nach den Namen des Typs `<00>`. Wenn Sie an den NetBIOS-Namen, über den Sie Informationen haben wollen, `#<xx>` anhängen, kann `nmblookup` auch einen anderen Ressourcentyp auflösen, wie z.B. den Messenger-Server-Typ `<03>`:

```
root# /usr/local/samba/bin/nmblookup eagle#03
Sending queries to 192.168.1.255
192.168.1.73 eagle<03>
```

`nmblookup` kann auch Gruppennamen auflösen:

```
root# /usr/local/samba/bin/nmblookup fowlplay
Sending queries to 192.168.1.255
192.168.1.73 fowlplay<00>
```

Und schließlich können Sie den Parameter `-S` verwenden, um eine Knotenstatusanfrage zum Namen durchzuführen und eine Ausgabe zu erhalten, die der des Befehls `nbtstat.exe -a` *`NetBIOS-Name`* ähnlich ist, den Sie von einem Windows-Rechner über die (MS-DOS-)Eingabeaufforderung starten:

```
root# /usr/local/samba/bin/nmblookup chipsndips -S
Sending queries to 192.168.1.255
192.168.1.73 chipsndips<00>
Looking up status of 192.168.1.72
received 8 names
     QUESO         <00> -         M <ACTIVE>
     CHIPSNDIPS    <00> - <GROUP> M <ACTIVE>
     QUESO         <03> -         M <ACTIVE>
     QUESO         <20> -         M <ACTIVE>
     CHIPSNDIPS    <1e> - <GROUP> M <ACTIVE>
     CARTEGW       <03> -         M <ACTIVE>
     CHIPSNDIPS    <1d> -         M <ACTIVE>
     .._MSBROWSE_. <01> - <GROUP> M <ACTIVE>
num_good_sends=0 num_good_receives=0
```

Zusammenfassung

Samba kann als effektiver Datei- und Druckerserver für PCs in einer Gruppenumgebung agieren. Die Mechanismen für die Zugriffskontrolle werden von dem zugrunde liegenden Betriebssystem gehandhabt, mit dem Sie vertraut sind, d.h. dem Standard-`lpr`-Drucksystem (oder anderen) und den grundlegenden Unix-Dateiberechtigungsbits. Das Erzeugen von Gruppenfreigaben oder entfernten Druckerfreigaben ist entsprechenden Aktionen unter Unix sehr ähnlich.

Zunächst müssen Sie sicherstellen, dass auf die Ressource von der Unix-Seite her zugegriffen werden kann, z.B. dass Benutzer über den Befehl `lpr -P`*`Druckername`* `Datei.txt` an den Drucker drucken und erfolgreich auf Dateien im freigzugebenden Verzeichnis zugreifen können.

Danach müssen die Freigaben in der Samba-Konfigurationsdatei definiert werden. Wie Sie in den nachfolgenden Kapiteln sehen werden, gibt es viele Parameter, die Freigaben definieren und kontrollieren.

Samba bietet außerdem viele Tools für die Administration des Servers und die Diagnose von Problemen. Zwei der Diagnose-Utilities, die Sie kennen gelernt haben, sind `smbstatus` und `nmblookup`.

Frage & Antwort

F. Kann ich Samba Dateisysteme freigeben, die über NFS gemountet wurden?

. Ja.

F. Wie viele gleichzeitige Verbindungen unterstützt ein Samba-Server?

. Theoretisch gibt es keine Einschränkungen. In der Praxis wird die Grenze durch die Server-Hardware bestimmt, insbesondere durch die Gesamtgröße des verfügbaren Arbeitsspeichers. Es kann auch von der Menge der Aktivität abhängen, die Sie von den `smbd`-Prozessen erwarten. Eine Situation, in der ein `smbd`-Prozess eine große Menge der CPU-Leistung belegt, ist, wenn auf einem Windows-Client ein Webbrowser läuft, der Dateien in den Cache eines Netzwerklaufwerks schreibt.

Neue Begriffe

Freigabe - Eine Freigabe, manchmal auch als *Dienst* bezeichnet, ist eine Ressource, wie z.B. ein Verzeichnis oder ein Drucker, der mit Hilfe des SMB-Protokolls für entfernte Rechner über ein Netzwerk zur Verfügung gestellt wurde.

Tag 5: Die Datei smb.conf: Samba mitteilen, was es tun soll

Die Datei `smb.conf` ist sozusagen das Herzstück von Samba. Sie wird sowohl von `smbd` und `nmbd` als auch von vielen der anderen Tools benutzt, die in der Samba-Programmfamilie enthalten sind. Und obwohl sie wahrscheinlich mehr Parameter hat als Godzilla Zähne, ist sie nicht sehr schwer zu verstehen. Dieses Kapitel bietet eine vertiefte Darstellung der Datei `smb.conf`. Sie werden sich den allgemeinen Aufbau der Datei, Variablen, die während der Laufzeit zur Anwendung kommen, und einige der globalen Parameter ansehen, die das allgemeine Verhalten von Samba kontrollieren.

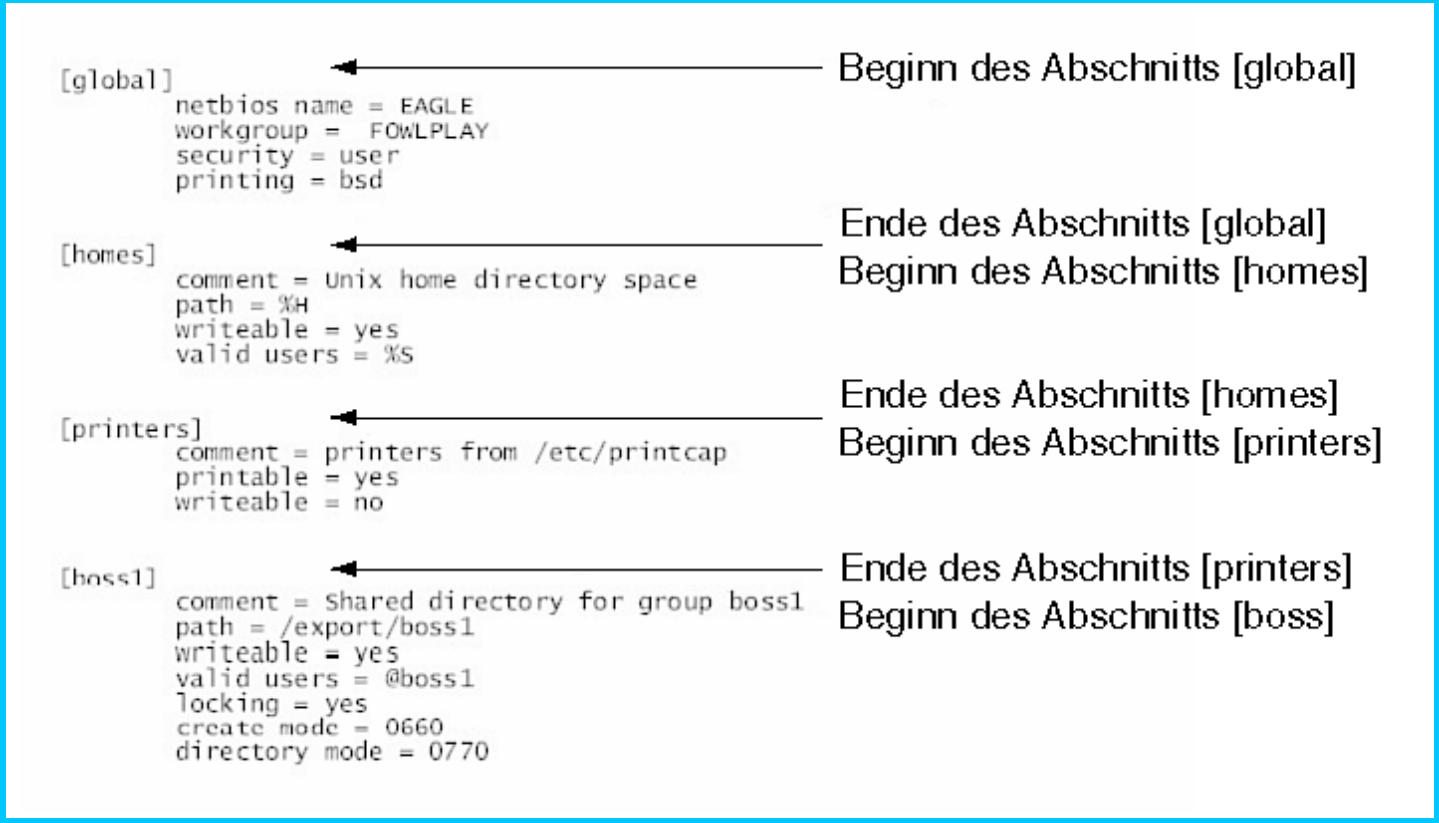
Aufbau

Eine Standard-`smb.conf`-Datei besteht aus verschiedenen Abschnitten, die jeweils mehrere Parameter enthalten. Diese Erklärung ist zwar richtig, aber nicht sehr hilfreich.

Die folgende Definition ergibt vielleicht mehr Sinn: Eine Samba-Konfigurationsdatei ist eine ASCII-Textdatei, die durch Abschnittsüberschriften logisch unterteilt ist, welche durch umschließende eckige Klammern (`[ ]`) gekennzeichnet sind. So wäre z.B. `[footbar]` eine gültige Abschnittsüberschrift. Die Namen der Abschnitte, Parameter und Werte sind nicht groß-/kleinsensitiv, es sei denn, sie gehören zum Betriebssystem, wie es für einen Verzeichnispfad der Fall ist. Jeder Abschnitt wird bis zur nächsten Abschnittsüberschrift fortgesetzt. Sambas `smb.conf` hat drei integrierte Abschnitte namens `[global]`, `[homes]` und `[printers]`.

Abbildung 5.1. stellt die integrierten Abschnitte und einen Beispielabschnitt dar. Weil Abschnittsüberschriften nicht groß-/kleinsensitiv sind, stehen die Bezeichnungen `[global]`, `[GLOBAL]` und `[Global]` alle für den gleichen Abschnitt. Die vier Einstellungen - `netbios name`, `workgroup`, `security` und `printing` - stehen für globale Parameter. Daher sind sie alle im Abschnitt `[global]` zu finden, der bei der nächsten Abschnittsüberschrift, `[homes]`, endet. Der letzte Abschnitt, `[boss1]`, steht für eine Festplattenfreigabe, die für diesen Server konfiguriert wurde.

Abb. 5.1: Der generelle Aufbau von smb.conf



[global]

Der Abschnitt `[global]` enthält Parameter, die für die allgemeine Funktionalität des Servers von Bedeutung sind. Die Parameter `netbios name` und `workgroup`, die in Kapitel 4, »Installation und Testen der Konfiguration«, kurz beschrieben wurden, sind Beispiele für globale Parameter. Diese und andere Parameter werden später in diesem Kapitel ausführlicher dargestellt.

[homes]

Der Abschnitt `[homes]` wurde bereits beim Beispiel-Samba-Server in Kapitel 4 kurz erwähnt. Diese spezielle Freigabe ermöglicht es Benutzern, auf ihre Home-Verzeichnisse zuzugreifen, ohne dass eine spezielle Freigabe für jeden Benutzer eingerichtet werden muss. Der Prozess läuft folgendermaßen ab:

1. Samba empfängt eine Verbindungsanfrage.
2. Die Datei `smb.conf` wird nach dem Namen der verlangten Freigabe durchsucht.
3. Wird der verlangte Name nicht gefunden und wurde die `[homes]`-Freigabe konfiguriert, durchsucht Samba die Datei `/etc/passwd` nach einem entsprechenden Benutzernamen.
4. Wird ein entsprechender Benutzername gefunden, erzeugt Samba eine Kopie der `[homes]`-Freigabe und ändert den Namen `homes` in den gefundenen Benutzernamen um. Ist kein Pfad angegeben, wird dieser auf das Home-Verzeichnis des Benutzers eingerichtet, wie er im `/etc/passwd`-Eintrag definiert ist.
5. Wird kein entsprechender Benutzername gefunden, gibt Samba die Fehlermeldung `Invalid resource in tree connection request` an den Client zurück.

[printers]

Der dritte integrierte Abschnitt, `[printers]`, ist `[homes]` ähnlich. Der Unterschied liegt in der Art der Ressource, die hier verfügbar gemacht wird. `[homes]` erzeugt Home-Verzeichnisse aus der `/etc/passwd`, während `[printers]` Drucker aus der `/etc/printcap` freigibt. Wenn Sie ein anderes Drucksystem als BSD benutzen, müssen Sie eine Hilfs-`printcap`-Datei für Samba erstellen, um Druckernamen zu authentifizieren. Weitere Informationen hierzu finden Sie in Kapitel 8, »Drucker«.

Die restlichen Abschnitte von smb.conf

Jeder Abschnitt außer `[global]` wird als freigegebene Ressource (kurz *Freigabe*) angesehen, daher müssen die Abschnitte generellen Benennungskonventionen für Freigaben folgen.

Um benutzerdefinierte Freigaben einzurichten, brauchen Sie nur eine Abschnittsüberschrift, wie z.B. `[foo]`, und die notwendigen Parameter eingeben, die in den Kapiteln 6, »Sicherheitsebenen und Passwörter«, und 7, »Dateifreigaben«, dargestellt werden. Der SMB-Client (z.B. ein Windows-PC) kann dann über den Netzwerkpfad `\\Servername\FOO` auf die Freigabe zugreifen. Ich habe vorher erwähnt, dass die Abschnittsüberschriften in `smb.conf` nicht groß-/kleinsensitiv sind. Daher beziehen sich `[foo]` und `[FOO]` auf die gleiche Freigabe. Darum kann der PC-Client `\\Servername\FOO` mounten, wenn die Freigabe als `[foo]` definiert ist.

Je nachdem wie sehr Sie es mögen, Ihre Arbeit zu dokumentieren - ich hoffe für die arme Person, die nach Ihnen kommt und Ihre Kreation übernehmen muss, dass Sie es sehr mögen -, können Sie freizügig Kommentare einfügen, indem Sie ein Semikolon (`;`) oder eine Raute (#) als erstes Zeichen einer Zeile einsetzen. Kommentare werden beim Zeilensprung beendet:

```
; Dies ist ein Kommentar
# und dies auch
```

Tabelle 5.1 ist eine Zusammenfassung der Formate für die `smb.conf`-Inhalte, die ich dargestellt habe.

Tabelle 5.1: Zusammenfassung der smb.conf-Formate

Eingabe	Format
Abschnitt	Zeile, die einen Zeichenstring enthält, der in eckige Klammern eingeschlossen ist, z.B. `[foo]`.
`[global]`	Spezieller Abschnitt, der Parameter enthält, die für die generellen Samba-Einstellungen und die Einstellungen der Standardfreigaben gelten.
`[homes]`	Dynamische Freigabe, die Namen aus der `/etc/passwd` holt.
`[printers]`	Dynamische Freigabe, die Druckernamen aus einer spezifizierten `printcap`-Datei holt.
Kommentar	Zeile, der das Zeichen `;` oder # vorangestellt ist.
Parameter	Konfigurationsparameter, gefolgt von = und einem Wert, z.B. `writable = yes`.

Variablen

Sie können in `smb.conf` verschiedene Variablen benutzen. Diese Makros, die durch das Zeichen % gekennzeichnet sind, werden während der Analyse der Konfigurationsdatei beim Ablauf ersetzt. Wenn z.B. Benutzer `jdoe` eine Anfrage für die Aufnahme einer Arbeitssitzung überträgt, analysiert Samba die `smb.conf` und ersetzt alle Entsprechungen der Variable `%U` durch `jdoe`. Tabelle 5.2 listet alle verfügbaren `smb.conf`-Variablen auf.

Tabelle 5.2: smb.conf-Variablen

Variable	Beschreibung
`%a`	Die Architektur des entfernten Rechners. Zuverlässigkeit wird nicht hundertprozentig garantiert, aber in der Regel ist es in der Praxis gut genug. Derzeit unterstützte Werte sind `Samba`, `WfWg`, `WinNT` und `Win95`. Windows 2000 ist tatsächlich Windows NT 5.0 und wird daher als `WinNT` erkannt
`%d`	Die Prozess-ID des aktuellen Server-Prozesses
`%g`	Die primäre Gruppe des Benutzernamens `%u`
`%G`	Die primäre Gruppe des Benutzernamens `%U`
`%h`	Der Name des Internet-Hosts, auf dem Samba läuft
`%H`	Das Home-Verzeichnis für den Benutzernamen `%u`
`%I`	Die IP-Adresse des Client-Rechners in Dezimalpunktschreibweise
`%L`	Der NetBIOS-Name des Servers
`%m`	Der NetBIOS-Name des Client-Rechners
`%M`	Der Internet-Host-Name des Client-Rechners

%N	Der Name Ihres NIS-Home-Directory-Servers, wie er in der `auto.home`-Map spezifiziert ist. Wenn Sie Samba ohne `AUTOMOUNT`-Unterstützung kompiliert haben, ist dies das Gleiche wie `%L`
%p	Der Pfad zum Home-Verzeichnis des Benutzers, wie er in `auto.home` definiert ist. Es wird vorausgesetzt, dass der NIS-Map-Eintrag durch einen Doppelpunkt getrennt und als `%N:%p` aufgeteilt ist
%P	Das Root-Verzeichnis des aktuellen Dienstes
%R	Das Protokoll, das während der Protokollabstimmungsphase bei Verbindungsaufnahme ausgewählt wurde
%S	Der Name der aktuellen Freigabe
%T	Aktuelles Datum und Zeit
%u	Benutzername der aktuellen Freigabe
%U	Der Benutzername, den der Client bei Aufnahme der Arbeitssitzung verlangt hat. Dies ist nicht unbedingt der gleiche wie der, der benutzt wurde
%v	Samba-Versionsnummer

Diese Variablen können auf vielfache Art und Weise benutzt werden. Eine Variable kann überall dort eingesetzt werden, wo ein Textstring zugelassen ist. Der folgende `[global]`-Parametereintrag z.B. würde Samba veranlassen, Verbindungsinformationen in eine Datei namens `/var/log/log.`*`NetBIOS-Name`* zu schreiben, wobei *`NetBIOS-Name`* durch den NetBIOS-Namen des Clients ersetzt wird.

```
log file = /var/log/log.%m
```

Hier ist noch ein Beispiel, das Samba mitteilt, abhängig vom Betriebssystem des sich verbindenden Clients ein anderes Domain-Logon-Skript zu benutzen:

```
logon script = %a.bat
```

Die Namen der verfügbaren Logon-Skripte wären `WfW.bat`, `Win95.bat` und `WinNT.bat`. Domain-Logons werden ausführlich in Kapitel 21, »Windows-9x-Domänenkontrolle«, und Anhang A, »Experimentelle PDC-Unterstützung«, dargestellt.

Um die Variablen ausführlich zu erklären, ist vielleicht die `[homes]`-Freigabe, die ich für das Beispiel in Kapitel 4 benutzt habe, besser geeignet:

```
; Freigabename
[homes]
   comment = Unix-Home-Verzeichnisbereich
   path = %H
   writable = yes
   valid users = %S
   create mode = 0600
   directory mode = 0700
   locking = no
```

Der Eintrag `valid users = %S` schränkt Verbindungen auf den Benutzer ein, dessen Benutzername dem der Freigabe entspricht. Denken Sie an meine frühere Erklärung der `[homes]`-Freigabe. Findet Samba eine Entsprechnung des Freigabenamens in der Datei `/etc/passwd`, wird eine Freigabe mit den Parametern aus der `[homes]`-Definition erzeugt, die in den entsprechenden Benutzernamen umbenannt wird. Daher ist der einzige Benutzer, dem die Verbindung erlaubt wird, der Eigentümer des Home-Verzeichnisses.

Noch ein letztes Beispiel, bevor es weitergeht. In meinem Beruf verwalte ich etwa 30 verschiedene Samba-Server, die unter verschiedenen Betriebssystemen laufen. In der Regel werden alle Server gleichzeitig auf die gleiche Version von Samba aktualisiert, aber es gibt immer einige Ausnahmen. Um schnell die installierte Version auf einem Server feststellen zu können, hat jede `smb.conf` im Abschnitt `[global]` einen Eintrag, der dem folgenden ähnelt:

```
server string = samba print server for administration [%v]
```

Mit dem Parameter `server string` legen Sie den Text fest, der neben dem Rechnernamen in Browse-Listen angezeigt wird, die über Tools wie die Netzwerkumgebung verfügbar sind. `%v` wird dynamisch auf die Version des aktuell laufenden `nmbd`-Prozesses aktualisiert. Um also festzustellen, welche Samba-Version auf einem Server läuft, benutze ich einfach den Befehl `net view \\Servername` von einem Windows-Rechner und untersuche den ausgegebenen String des Servers.

Parameter

Ein schnelles `grep` durch die `smb.conf`-2.0-Manpage legt über 130 einzelne globale Parameter und etwa 100 weitere offen, die mit Freigaben zu tun haben. Die `smb.conf`-2.0-Manpage ist etwa 8.500 Zeilen lang. Es erübrigt sich zu sagen, dass für die Konfiguration Ihres Servers ziemlich viele Optionen zur Verfügung stehen. In diesem Abschnitt lernen Sie einige der gebräuchlichsten Optionen kennen. Ich hebe die Darstellung einiger `[global]`-Optionen für spätere Kapitel auf, in denen der Kontext besser zur Funktion des Parameters passt. Eine komplette Auflistung der aktuellen `[global]`-Parameter finden Sie, wie immer, in der `smb.conf`-Manpage.

Die Werte für Parameter lassen sich, mit wenigen Ausnahmen, in drei Kategorien aufteilen:

- Bei der ersten wird der Wert als Zeichenstring eingegeben, wie z.B. `jerryc` oder `samba server`. Groß-/Kleinschreibung wird in Textstrings beibehalten.
- Die zweite ist ein boolescher Parameterwert, der `yes/no`, `true/false` oder `1/0` akzeptiert. Boolesche Werte sind nicht groß-/kleinsensibel, also sind `YES`, `Yes` und `yes` für Samba identisch.
- Die dritte Kategorie der Parameter akzeptiert einen numerischen Wert. Sie müssen jeden Parameter überprüfen, um zu bestimmen, ob es sich um eine ganze Zahl oder eine Basis handelt, wie z.B. einen Erstellungsmodus, der eine Oktalzahl ist.

Parameter haben die Form *`Name`* = *`Wert`*, z.B.:

```
netbios name = EAGLE
```

Nur das erste Gleichheitszeichen wird für die Analyse des Parameters und seines Werts benutzt. Der Wert beginnt beim ersten nicht leeren Zeichen nach dem Gleichheitszeichen und endet mit dem ersten Zeilenumbruch, dem kein \-Zeichen vorangestellt ist. Daher hat die folgende Einstellung die gleiche Bedeutung wie das obenstehende Beispiel:

```
netbios name        =         EAGLE
```

netbios name

Sie haben den Parameter `netbios name` oben schon kurz gesehen. Über diesen Parameter richten Sie NetBIOS-Rechnernamen des Samba-Servers ein. Wie die meisten anderen Parameter hat auch dieser einen Standardwert, nämlich den Hostnamen des Servers. Es ist möglich, diesen Parameter nicht einzurichten und den Standardwert zu benutzen, aber ich persönlich ziehe es vor, den Rechnernamen explizit zu definieren.

Standard: `netbios name = Internet-Hostname des Rechners`

Ich möchte die Themen Namensauflösung und Browsing an dieser Stelle nicht zu sehr vertiefen, aber ich habe die Erfahrung gemacht, dass es einfacher ist, wenn NetBIOS-Rechnername und Internet-Hostname gleich sind, es sei denn, Sie haben einen sehr guten Grund, verschiedene Namen zu verwenden. Wenn z.B. der Hostname des Servers `eagle` ist, würde ich den NetBIOS-Namen explizit wie folgt einrichten:

```
netbios name = EAGLE
```

Alle gültigen DNS-Namen, die nicht länger als 15 Zeichen sind, sind auch gültige NetBIOS-Namen. Das Gegenteil trifft nicht zu, da einige Zeichen, wie z.B. eine Tilde (~), für Rechnernamen benutzt werden können, aber nicht gültig sind, wenn es um DNS geht.

netbios aliases

In Kapitel 2, »Windows-Netzwerke«, habe ich im Abschnitt »NetBIOS-Überblick« erwähnt, dass es in einer NetBIOS-Verbindung einen »Anrufernamen« auf Seiten des Clients und einen vom Client verlangten »angerufenen« Namen gibt. Ein NetBIOS-Server antwortet nur auf Anfragen, die seinem angerufenen (*called*) Namen entsprechen. Der Parameter `netbios aliases` ermöglicht Samba, auf mehrere angerufene Namen zu antworten. Das heißt, Sie können den gleichen Server in einer Arbeitsgruppe unter mehreren Namen sehen, wenn Sie auf einem Windows-Client durch die Netzwerkumgebung browsen. Jeder Server-Name könnte verschiedene Freigaben zur Verfügung stellen, die sich alle auf dem gleichen Rechner befinden. Standardmäßig sind keine `netbios aliases` eingerichtet.

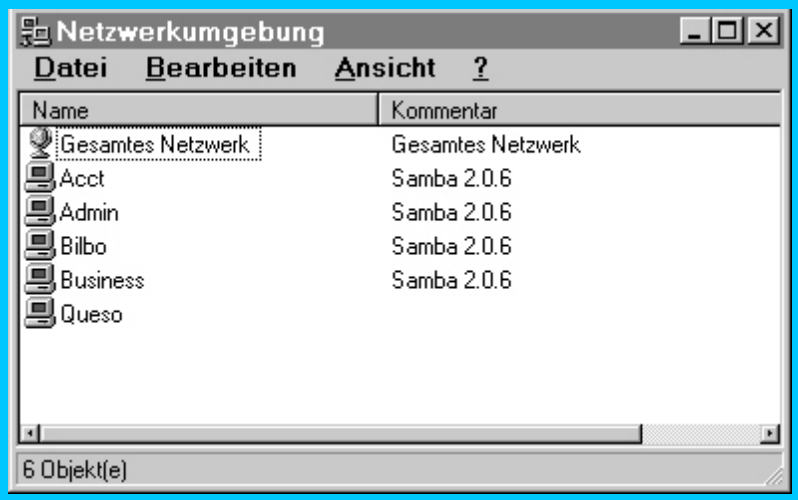
Standard: `netbios aliases =`

Abbildung 5.2 zeigt einen Server in der Windows-95-Netzwerkumgebung, wenn folgende Einstellung aktiviert ist:

```
netbios aliases = admin acct business
```

Der primäre NetBIOS-Name des Samba-Servers, `BILBO`, erscheint ebenfalls in der Auflistung. Sie sollten beachten, dass nur der primäre Name (d.h. `netbios name =` ...) benutzt wird, wenn auf Anfragen für Domänen-Logons reagiert werden soll oder wenn der Server als Browse-Server konfiguriert ist.

Abb. 5.2: Ein Beispiel für einen einzelnen Samba-Server mit mehreren NetBIOS-Aliasen



workgroup

Der Parameter `workgroup` entscheidet, zu welcher Arbeitsgruppe der Server gehört, wenn er auf Anfragen von Clients reagiert. Die Zugehörigkeit zur Arbeitsgruppe beeinflusst auch andere Einstellungen wie z.B. Domänen-Logons, Domänenzugehörigkeit und Browse-Dienste.

Der Standardwert für diesen Parameter wird während der Kompilierung durch das Makro `WORKGROUP` eingerichtet, das in dem Makefile definiert ist.

Standard: `workgroup = während Kompilierung definiert`

Ein Arbeitsgruppenname ist ein NetBIOS-Gruppenname und muss daher den Standardnamenskonventionen folgen (siehe Kapitel 2). Ein Beispiel:

```
workgroup = FOWLPLAY
```

server string

Der Parameter `server string` definiert den Textstring, der im Kommentarabschnitt des Windows-Druckmanagers angezeigt wird. Er wird auch mit dem NetBIOS-Namen des Rechners angezeigt, wenn Sie z.B. in der Netzwerkumgebung durch das Netzwerk browsen. Der Parameter akzeptiert, wie andere auch, `smb.conf`-Variablen. So können Sie hier z.B. über die Variable `%v` die aktuell laufende Version von Samba überprüfen.

Die Standardeinstellung ist:

Standard: `server string = Samba %v`

Ich gebe normalerweise eine etwas umfassendere Beschreibung ein, die den Standort eines Rechners bestimmt:

```
server string = Drucker-Server in Abteilung Einkauf [%v]
```

Folgendes Beispiel zeigt die Server in der aktuellen Arbeitsgruppe von einem Windows-NT-4.0-SP5-Rechner gesehen:

```
H:\>net view
Server-Name       Beschreibung
-------------------------------------------------------
\\BURRITO         Drucker-Server in Abteilung Einkauf [2.1.0-prealpha]
\\PIZZA           Samba [1.9.18p7]
Der Befehl wurde erfolgreich ausgeführt.
```

log file

Über diesen Parameter können Sie den während der Kompilierung festgelegten Standardstandort der `smbd`-Logdateien außer Kraft setzen.

Standard: `log file = bei Kompilierung eingerichtet`

Es gibt einige Besonderheiten für diesen Parameter. Sie sollten wissen, in welcher Reihenfolge die Dinge geordnet sind.

1. Wird während des Starts über den Parameter `-l` eine Datei spezifiziert, schreibt `smbd` die ersten Log-Einträge in die Datei, die in der Befehlszeile angegeben ist. Wird während des Starts kein Standort festgelegt, protokolliert `smbd` die Log-Informationen in der Datei, die während der Kompilierung angegeben wurde.
2. Wird bei einer Analyse der Konfigurationsdatei der `log-file`-Parameter gefunden, werden alle zukünftigen Log-Einträge in die Datei geschrieben, die durch den Wert des Parameters spezifiziert sind.

Weil Samba also zunächst nichts über den in der `smb.conf` definierten `log-file`-Standort weiß, schreibt es einige Startinformationen in die Logdatei, die es beim Starten kennt.

Sie können den während der Kompilierung festgelegten Standardstandort für die Logdatei von `nmbd` nur überschreiben, indem Sie beim Start den Parameter `-l` verwenden.

Dieses Beispiel erzeugt eine separate Logdatei für jeden Benutzer, der sich mit dem Server verbindet (oder versucht, sich zu verbinden):

```
log file = /var/log/log.%U
```

max log size

Der Parameter `max log size` nimmt als Wert eine ganze Zahl an, mit der die maximale Größe für die Logdatei in Kilobyte spezifiziert wird. Samba überprüft regelmäßig die Größe der Logdateien. Hat eine Logdatei die definierte maximale Größe überschritten, benennt Samba die Datei mit der Erweiterung `.old` um und erstellt eine neue. Existiert bereits eine Datei mit gleichem Namen (*`Logdatei`*`.old`), wird sie überschrieben. Der Standardwert ist auf 5 Mbyte gesetzt.

Standard: `max log size = 5000`

Sie können hier einen beliebigen Wert einfügen. Der folgende Eintrag richtet die maximale Größe der Logdatei auf 2 Mbyte ein:

```
max log size = 2000
```

syslog

Damit dieser Parameter in Kraft treten kann, müssen Sie während der Kompilierung die `syslog`-Unterstützung aktivieren:

```
./configure -- with-syslog
```

Der Parameter `syslog` nimmt als Wert eine ganze Zahl an und gleicht die Samba-Debug-Prioritäten mit den `syslog`-Log-Prioritäten ab. Die Entsprechungen finden Sie in Tabelle 5.3. Nur Samba-Debug-Meldungen mit einer Priorität, die kleiner als der definierte Wert ist, werden an den `syslogd`-Daemon übertragen. Standardmäßig werden daher nur Debug-Meldungen mit der Priorität 0 an `syslog` gesendet, obwohl der Wert auf `1` eingestellt ist.

Standard: `syslog = 1`

Tabelle 5.3 listet die verschiedenen Debug-Prioritäten und ihre `syslog`-Entsprechungen auf.

Tabelle 5.3: Samba-Debug-Level und entsprechende syslog-Prioritäten

Samba-Debug-Level	syslog-Level
0	LOG_ERR
1	LOG_WARNING
2	LOG_NOTICE
3	LOG_INFO
>3	LOG_DEBUG

Sollen alle Meldungen, die der Priorität `LOG_NOTICE` entsprechen, an den `syslogd`-Prozess übertragen werden, fügen Sie folgenden Eintrag in `smb.conf` ein:

```
syslog = 3
```

syslog only

Dieser boolesche Parameter bestimmt, ob Meldungen nur an den `syslog`-Daemon gesendet werden und nicht an die normalen Debug-Logdateien. Dieser Parameter wird zusammen mit dem Parameter `syslog` verwendet und setzt voraus, dass während der Kompilierung die `syslog`-Unterstützung aktiviert wurde. Standardmäßig werden Debug-Einträge zusätzlich zu den `syslog`-Dateien auch an die Standard-`smbd`- und -`nmbd`-Logdateien übertragen. Durch folgende Einstellung können Sie Samba veranlassen, Logging-Informationen nur an den `syslog`-Daemon weiterzugeben:

```
syslog only = yes
```

debug level

Über den Parameter `debug level`, der auch `log level` genannt wird, können Sie den maximalen Grad (Level) der Debug-Meldungen einstellen, die auf die Festplatte geschrieben werden. Der Parameter hat einen Standardwert von 2.

Standard: `debug level = 2`

Der Parameter `debug level` gilt sowohl für `smbd` als auch für `nmbd`. Sie werden Samba-Logs zu Debugging-Zwecken ausführlich in späteren Kapiteln verwenden. Hier ein Beispiel, in dem der Log-Level auf `5` eingestellt ist:

```
debug level = 5
```

Je höher der Debug-Level eingestellt ist, umso ausführlicher werden Meldungen in die Log-Dateien geschrieben.

Wenn Sie über die Option `-d` in der Befehlszeile einen Debug-Level definieren, setzt dieser Wert die Einstellung für den Parameter `debug level` außer Kraft.

lock directory

Mit diesem Parameter legen Sie einen Pfad für das Verzeichnis fest, in das Samba seine freigegebene Speicherdatei, Statusdatei, Browse-Liste, WINS-Datenbank (wenn WINS-Unterstützung aktiviert ist) und Lock-Dateien schreibt, die für die Implementierung des Parameters `max connections` verwendet werden. Der Parameter `max connections` wird in Kapitel 7 dargestellt, in dem Sie erfahren, wie Sie Samba für die Freigabe von Verzeichnissen konfigurieren. Der Zweck der Parameter besteht darin, die Anzahl der Benutzer einzuschränken, die sich gleichzeitig mit einer Freigabe verbinden können.

Während der Kompilierung wird normalerweise das Lock-Verzeichnis `/usr/local/samba/var/locks` als Standard festgelegt:

Standard: `lock directory = während der Kompilierung festgelegt`

In der Praxis sollten Sie die Standardeinstellung für das Lock-Verzeichnis z.B. dann ändern, wenn Sie mehreren Servern die Benutzung der gleichen Samba-Binärdateien ermöglichen möchten, indem Sie Tools auf einem NFS-gemounteten Dateisystem bereitstellen. Viele Unternehmen mounten solch ein Dateisystem in `/usr/local/`, um netzwerkspezifische Tools und Utilities freizugeben. Zwar können Sie Binärdateien zwischen Samba-Servern gemeinsam nutzen, aber es ist nicht möglich, ein Lock-Verzeichnis freizugeben. Daher sollten Sie für jeden Server ein lokales Verzeichnis festlegen, in das Samba die notwendigen Dateien platzieren kann.

```
lock directory = /var/spool/locks/samba
```

name resolve order

Der Parameter `name resolve order` entspricht der Datei `/etc/nsswitch.conf` auf Plattformen wie Linux, Solaris und IRIX. Mit diesem Parameter können Sie die Reihenfolge festlegen, in der versucht wird, Namen aufzulösen. Der Parameterwert ist eine durch Leerstellen getrennte Liste, für die vier Einträge zulässig sind. Tabelle 5.4 listet die möglichen Werte und Besonderheiten auf.

Tabelle 5.4: Zugelassene Einträge für den Parameter name resolve order

Wert	Beschreibung

`lmhosts`	Die Samba-Datei `lmhosts` wird auf eine Entsprechung des verlangten Namens durchsucht.
`hosts`	Dieser Wert weist Samba an, eine Standard-Hostname-/IP-Adresse-Auflösung durchzuführen und dafür die auf dem System zur Verfügung stehenden Mittel zu benutzen, z.B. Durchsuchen von `/etc/hosts`, DNS-Anfragen oder NIS/NIS+-Entsprechungen. Bedenken Sie, dass diese Methode nur benutzt wird, wenn der aufzulösende NetBIOS-Name die Server-Ressourcenkennung (`<20>`) hat.
`wins`	Ist über die Parameter `wins server` oder `wins support` (siehe Kapitel 18, »WINS«) ein WINS-Server definiert, kann der NetBIOS-Name über eine Anfrage an den WINS-Server aufgelöst werden.
`bcast`	Führt die normale NetBIOS-Namensauflösung per Broadcast durch, die voraussetzt, dass sich der in Frage kommende Host im gleichen Broadcast-Subnetz befindet (oder es vielleicht einen WINS-Proxyserver gibt).

Standardmäßig wird zuerst die lokale `lmhosts`-Datei durchsucht. Eine `lmhosts`-Datei ist die NetBIOS-Entsprechung zur Unix-Datei `/etc/hosts`. Danach versucht Samba, über Standardmethoden wie das Durchsuchen von `/etc/hosts` oder Anfragen an den DNS den Namen aufzulösen. Waren diese beiden Methoden erfolglos, kontaktiert der Server einen WINS-Server, falls in `smb.conf` einer spezifiziert wurde. Bleibt auch dies erfolglos, versucht Samba, den Namen über Broadcast-Anfragen aufzulösen.

Standard: `name resolve order = lmhosts hosts wins bcast`

Mit folgender Einstellung verwendet Samba für die Namensauflösung keine Broadcasts:

```
name resolve order = lmhosts wins hosts
```

deadtime

Über diesen Parameter können Sie die Anzahl von Minuten festlegen, die eine Verbindung (wie z.B. ein `smbd`-Prozess) inaktiv sein darf, bevor sie als terminiert angesehen und abgebrochen wird. Eine Verbindung gilt dann als inaktiv, wenn keine Aktivität erkannt wird und sie keine offenen Dateien enthält. Dies kann auf einem Server hilfreich sein, der viele Verbindungen handhabt, die nicht immer benutzt werden. Meine Benutzer haben die Angewohnheit, sich einzuloggen und niemals wieder auszuloggen, selbst wenn sie in den Urlaub gehen. Die meisten Clients haben eine Funktion, die die Verbindung automatisch wiederherstellt und diese Einstellung für den Benutzer transparent macht.

Die Standardeinstellung 0 bestimmt, dass die Verbindung niemals gekappt wird.

Standard: `dead time = 0`

Auf den Servern an meinem Arbeitsplatz habe ich diesen Wert auf 15 Minuten eingestellt:

```
dead time = 15
```

smbrun

Dieser Parameter definiert den absoluten Pfad zur Binärdatei `smbrun`, einem kleinen Programm, das vom `smbd`-Daemon zur Ausführung von Shell-Befehlen benutzt wird. Wenn Sie Samba über den Standardbefehl `make install` installiert haben, sollten Sie diesen Parameter nicht gebrauchen. Haben Sie die Samba-Binaries aber manuell an einem anderen Standort installiert als dem durch `$prefix` in der Makefile definierten, müssen Sie diesen Parameter einrichten. Wenn `smbd` die `smbrun`-Binärdatei nicht finden kann, protokolliert es entsprechende Debug-Meldungen in der Datei `log.smb`. Der Standardwert wird durch die Variable `$prefix` in dem Makefile bestimmt.

Standard: `smbrun = während Kompilierung eingerichtet`

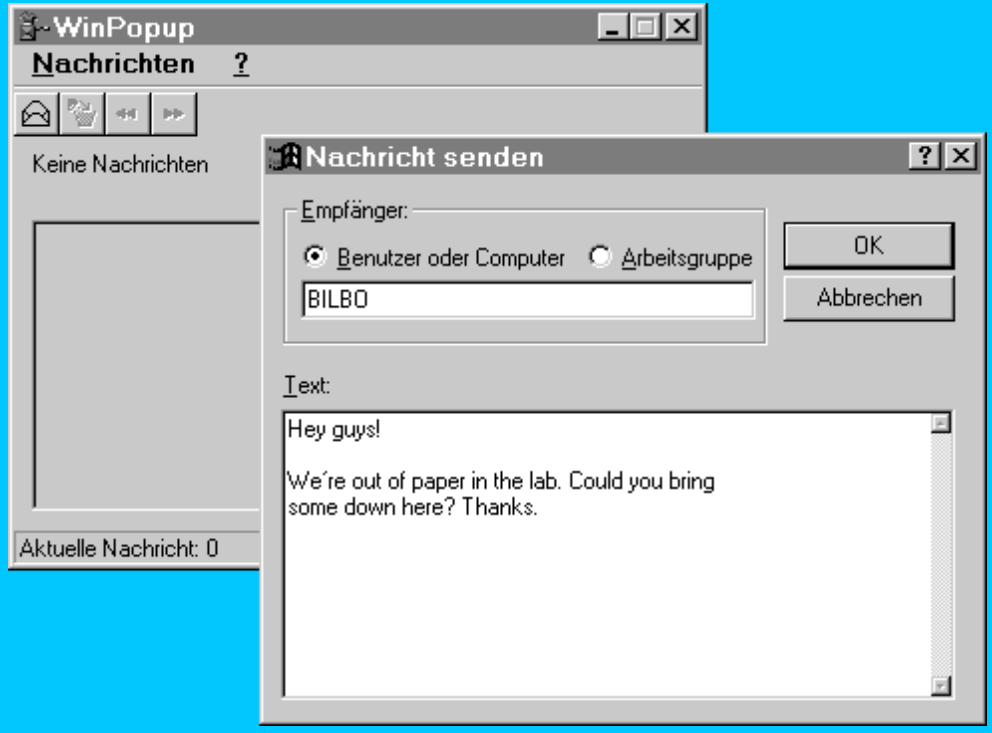
Haben Sie das Tool in ein anderes Verzeichnis installiert, wie z.B. `/usr/bin`, müssen Sie die Pfadangabe entsprechend ändern:

```
smbrun = /usr/bin/smbrun
```

message command

Der Parameter `message command` bestimmt die Aktion, die Samba durchführt, wenn es eine WinPopup-artige Nachricht empfängt. Sie wissen von der Darstellung der NetBIOS-Namen in Kapitel 2, dass Namen mit der Ressourcenbezeichnung `<03>` den Messenger Server darstellen. Die WinPopup-Meldungen werden an diesen Namen gesendet. Abbildung 5.3 zeigt das WinPopup-Windows-95-Utility, das eine Nachricht an den Samba-Server mit dem Namen `BILBO` sendet.

Abb. 5.3: Windows 95 OSR2 WinPopup.exe sendet und empfängt Nachrichten



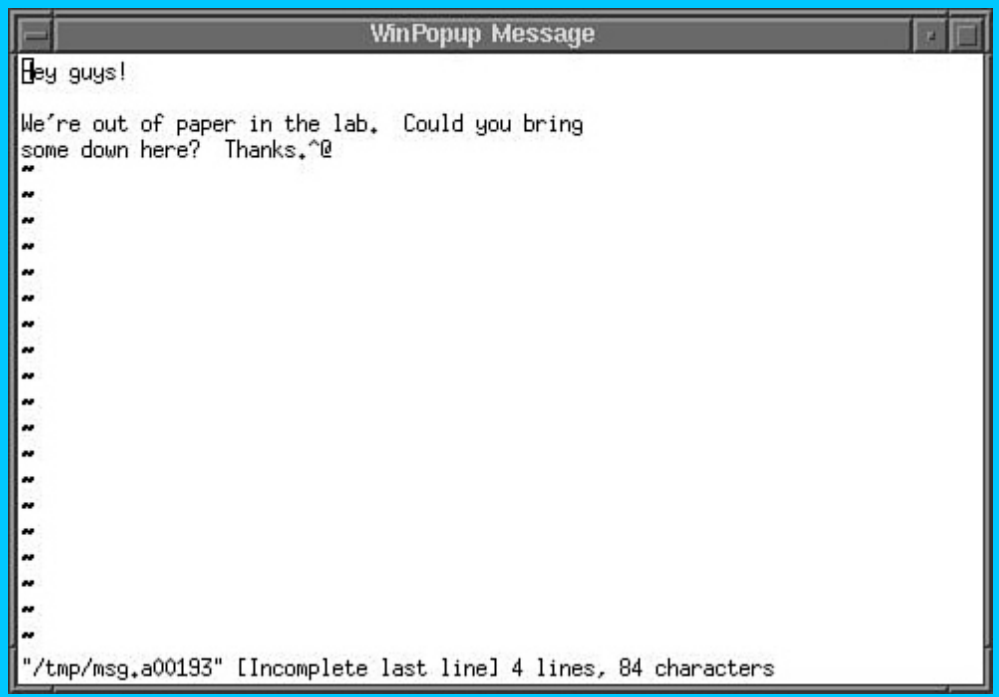
Standardmäßig verwirft Samba WinPopup-artige Nachrichten.

Standard: `message command = none`

Es gibt viele Möglichkeiten, die Nachricht zu senden. Ich habe das folgende Beispiel benutzt, um die Nachricht, die vom WinPopup-Client gesendet wurde (siehe Abbildung 5.3) auf meinem Linux-Rechner anzuzeigen (siehe Abbildung 5.4):

```
message command = /bin/bash -c '/usr/X11R6/bin/xterm -T "WinPopup Nachricht" \
-e /usr/bin/vim %s; rm %s' &
```

Abb. 5.4: Ausgeführtes message command bei Erhalt einer WinPopup-Nachricht



Sie könnten auch ein Befehlszeilen-Mail-Utilitiy wie z.B. `/bin/mailx` benutzen, um die Nachricht über SMTP zu übertragen.

Die WinPopup-Nachricht wird mit dem globalen `guest account` übertragen (in der Regel der Account `nobody`). Der Befehl kann neben den Standardmakros zusätzliche Variablen enthalten. Diese sind in Tabelle 5.5 aufgelistet.

Tabelle 5.5: Zusätzliche Variablen für den Parameter message command

Variable	Beschreibung
%s	Name der Datei, die den Textteil der Nachricht enthält.

%t	Name des Empfängers, an den die Nachricht gesendet wurde. Normalerweise ist dies der Name des Servers.
%f	Name des Clients, von dem die Nachricht stammt.

Es gibt einige Dinge, die Sie beim Einrichten des Parameters `message command` beachten sollten:

- Sie müssen absolute Pfade zu den Binaries benutzen, es sei denn, die von Ihnen verwendeten Befehle sind im Standardsuchpfad für die ausgeführte Shell enthalten.
- Sie müssen die erhaltene Nachricht explizit entfernen, sonst bleibt sie nach Ausführung des Befehls stehen.
- Das Programm sollte sofort wieder die Kontrolle an Samba zurückgeben, sonst kann der übertragende Client hängen, bis es zu einem Timeout kommt.

auto services

Dieser Parameter akzeptiert eine Liste aller Freigabenamen, die automatisch in der Browse-Liste für den Samba-Server sichtbar sein sollen. Dies ist wahrscheinlich in Bezug auf dynamisch erzeugte Dienste wie `[homes]` und `[printers]` am sinnvollsten. Die Standardeinstellung (keine Freigaben automatisch sichtbar) würde es nicht ermöglichen, die erweiterte Version dieser Freigaben zu sehen.

Standard: `auto services = none`

Die folgende Einstellung lässt die Home-Verzeichnisse für die Benutzer `jerryc` und `peteh` in einer Browse-Liste auftauchen. Dieser Parameter gibt nicht mehr Zugriffsrechte auf die Dateien in der Freigabe als ein Benutzer normalerweise hat.

Nehmen wir an, dass `jerryc` und `peteh` Benutzernamen in der lokalen `/etc/passwd` sind und dass die `[homes]`-Freigabe korrekt definiert wurde. Diese Freigaben sind normalerweise nicht verfügbar, bis der Benutzer die Verbindung zum Server aufgebaut hat. Die folgende Beispieleinstellung führt dazu, dass die Freigaben in der Browse-Liste gezeigt werden, egal welcher Benutzer sich mit dem Server verbindet.

```
auto services = jerryc peteh
```

Das heißt jedoch nicht, dass Benutzer sich mit diesen bestimmten Freigaben verbinden können. Sie können lediglich sehen, dass die Freigaben auf dem bestimmten Server verfügbar sind.

protocol

Während der Verhandlungsphase beim Aufbau einer SMB-Verbindung sendet der Client eine Liste der Protokolldialekte, die er unterstützt. Der Server wählt daraus den höchsten aus, der ihm bekannt ist. Wenn Sie sich dies noch einmal ansehen wollen, blättern Sie zurück zu Kapitel 2.

Mit dem Parameter `protocol` können Sie den höchsten SMB-Dialekt spezifizieren, den Samba beherrscht. Normalerweise sollten Sie diese Option nicht verwenden, damit Samba die Protokollauswahl automatisch handhaben kann. Die Standardeinstellung ermöglicht `smbd`, den höchsten möglichen SMB-Dialekt, NT1, auszuwählen.

Standard: `protocol = NT1`

In Tabelle 5.6 sind die zugelassenen Namen und eine kurze Beschreibung aufgelistet.

Tabelle 5.6: SMB-Dialekte

Name	Beschreibung
CORE	Die früheste Version von SMB, die keine Unterstützung für Benutzernamen bietet.
COREPLUS	Im Wesentlichen eine schnellere Version von CORE.
LANMAN1	Die erste moderne Version des Protokolls, die auch Unterstützung für lange Dateinamen bietet.
LANMAN2	Version 2 ist eine verbesserte Version des LANMAN1-Protokolls.
NT1	Dies ist die aktuellste Version des in Samba implementierten Protokolls, die auch vom Windows-NT-4.0-Service-Pack-3 verwendet wird. Die Version 2 dieses Protokolls wurde mit dem Service-Pack-4 für Windows NT freigegeben. Windows-NT-SP4-Clients arbeiten auch noch korrekt mit Samba mit der Version 1 des Protokolls.

time server

Ist dieser Parameter auf `true` gesetzt, kündigt `nmbd` sich als Zeitserver für Windows-Clients an und ermöglicht Ihnen daher, folgenden Befehl auf einem Windows-Client über die (MS-DOS-)Eingabeaufforderung auszuführen und die entsprechenden Ergebnisse zu erhalten:

```
C:\WINDOWS> net time
```

```
Aktuelle Zeit auf \\BILBO ist 1-27-1999 9:39 P.M.
Der Befehl wurde erfolgreich ausgeführt.
```

Auch wenn Sie diesen Parameter nicht einrichten, können Sie einen bestimmten Server immer nach der aktuellen Zeit befragen, indem Sie folgenden Befehl ausführen:

```
C:\WINDOWS> net time \\<Servername>
Standard:    time server = no
```

Standardmäßig antwortet Samba nicht auf `time-server`-Anfragen.

Standard: `time server = no`

Zusammenfassung

Zwar gibt es relativ viele `smb.conf`-Parameter, aber Sie müssen nur die einrichten, die Sie verwenden wollen, bzw. die, die Sie explizit definieren wollen, wenn Sie genau so übervorsichtig sind wie ich. Das folgende Beispiel ist wahrscheinlich die einfachste funktionierende `smb.conf`, die ich mir vorstellen kann. Sie implementiert einen einfachen Home-Verzeichnis-Server:

```
[global]
   workgroup = MYGROUP
[homes]
   writeable = yes
```

Es liegt an Ihnen und den Bedürfnissen Ihres Netzwerks, wie ausführlich Sie Ihre `smb.conf` gestalten.

Frage & Antwort

F. Wenn ich die Samba-Konfigurationsdatei ändere, muss ich dann die beiden Samba-Daemons beenden und neu starten?

. Für die meisten Änderungen an der Konfiguration müssen Sie nichts tun. Samba überprüft regelmäßig, ob die Konfigurationsdatei geändert wurde. Wenn ja, wird sie neu geladen. Aber es gibt hier einige Ausnahmen. Erstens, wenn Sie die Definition einer Freigabe ändern, können aktuell damit verbundene Benutzer die Änderungen erst sehen, wenn Sie die Verbindung beendet und die Freigabe neu gemountet haben. Zweitens, einige Änderungen erfordern, dass Samba neu gestartet wird, z.B. die Parameter `netbios name` oder `workgroup`.

F. Kann Samba mehr als einer Arbeitsgruppe gleichzeitig angehören?

. Nein. Samba kann nur Mitglied einer Arbeitsgruppe sein.

Tag 6: Sicherheitsmodi und Passwörter

Als ich nach meiner Woche Urlaub in Kapitel 4, »Installation und Testen der Konfiguration«, an meinen Arbeitsplatz zurückkam, nahm meine Chefin mich beiseite. »Diese Netzwerklaufwerksdinger, die du eingerichtet hast, sind großartig. Die Produktivität ist in ungeahnte Höhen geschnellt. Ich würde diese Methode gern unternehmensweit einsetzen, aber bevor ich sie dem Management empfehlen kann, brauche ich einige genaue Fakten zur Sicherheit des Ganzen. Kannst du mir erklären, wie Samba mein Passwort überprüft, wenn ich mich einlogge?«

Einen Moment stand ich da und dachte nach. Dann sagte ich: »Ich setze mich gerne hin und erkläre, wie das alles abläuft, aber erst brauche ich eine Stunde, um mir einen Kaffee zu holen und einige Dinge zu erledigen.«

»Hört sich gut an«, sagte meine Chefin. »Ich sehe dich dann in einer Stunde in meinem Büro.«

Ich ging den Flur hinunter zum Testlabor und versuchte mich daran zu erinnern, wo ich meine Lieblingstasse und meine Kopie dieses Buches gelassen hatte.

Konnte ich die Informationen rechtzeitig finden? Würde Samba unternehmensweit eingesetzt werden?

Bleiben Sie dabei und finden Sie es heraus!

Im vorangegangenen Kapitel habe ich den allgemeinen Aufbau der Samba-Konfigurationsdatei und einige der generellen und verschiedenen `[global]-Parameter` dargestellt. In diesem Kapitel geht es um die Parameter, die für die Authentifizierungsmethoden relevant sind, wenn sich Clients mit dem Samba-Server verbinden. Außerdem werde ich Ihnen etwas über Passwortsicherheit, Verschlüsselung und die Benutzung der IP-Adresse eines Clients für die Entscheidung, ob die Verbindung überhaupt authentifiziert wird, erzählen.

Sicherheitsmodi und der Parameter security

Das SMB-Protokoll bietet zwei grundsätzliche Modi für die Authentifizierung von Verbindungen. Der von Samba benutzte Modus wird durch die Einstellungen für den Parameter `security` im Abschnitt `[global]` der `smb.conf` definiert.

Der Eintrag für den Parameter `security` in der `smb.conf`-Manpage listet vier mögliche Eingaben auf, die verwendet werden können. Ich sprach aber von zwei grundlegenden Methoden, die das SMB-Protokoll für die Authentifizierung bietet. In der Realität sind von den vier Modi, die Samba unterstützt - `share`, `user`, `server` und `domain` - nur `share` und `user` fundamental unterschiedlich und stellen damit die zwei SMB-Sicherheitsmodi dar. Die anderen von Samba unterstützten Werte sind Variationen des User-Sicherheitsmodus:

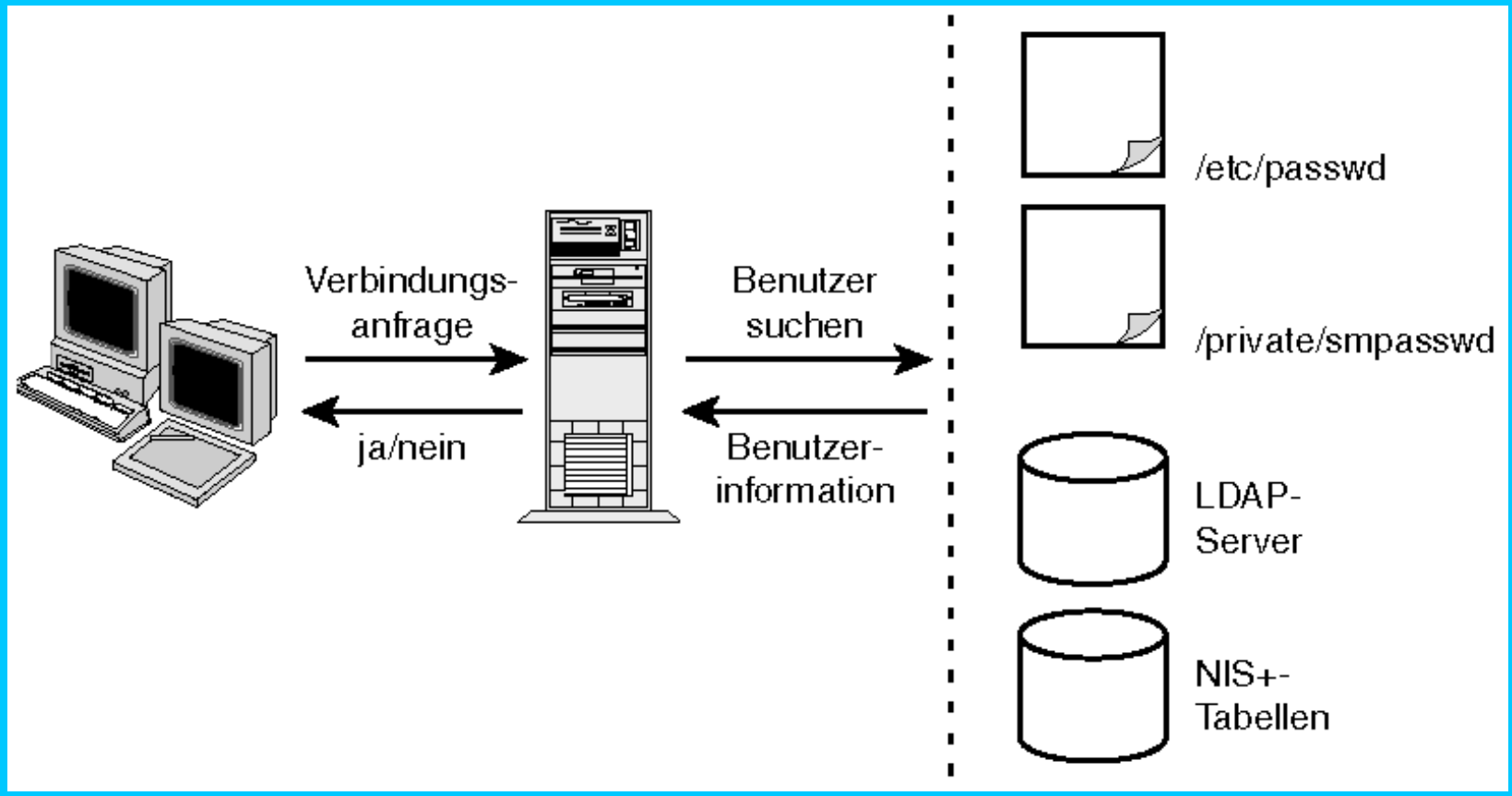
```
security = [share|user|server|domain]
```

Vor Version 2.0 war die Standardeinstellung für den Sicherheitsparameter `share`. Mit der Version 2.0 wurde diese Einstellung auf den User-Modus geändert.

Mit Samba 2.0 wurde ein Passwortdatenbank-API eingeführt, über das Entwickler durch Definition einer Sammlung von Funktionen verschiedene Authentifizierungsmethoden einbetten können. Das heißt, dass Sie mehrere Methoden zur Auswahl haben, wenn Sie entscheiden, wie Sie die Informationen zu Ihren Bentuzer-Accounts speichern wollen.

Abbildung 6.1 zeigt die möglichen Backends, die derzeit benutzt werden oder sich in der Entwicklungsphase befinden. Der Client verlangt eine Verbindung zum Server, und der Server kontaktiert die Account-Datenbank über ein definiertes Interface. Es ist nicht unbedingt notwendig zu wissen, welches Backend benutzt wird. Was Samba betrifft, bietet die Datenbank die benötigten Benutzerinformationen.

Abb. 6.1: Samba ermöglicht mehrere, voneinander unabhängige Benutzer-Account-Datenbanken

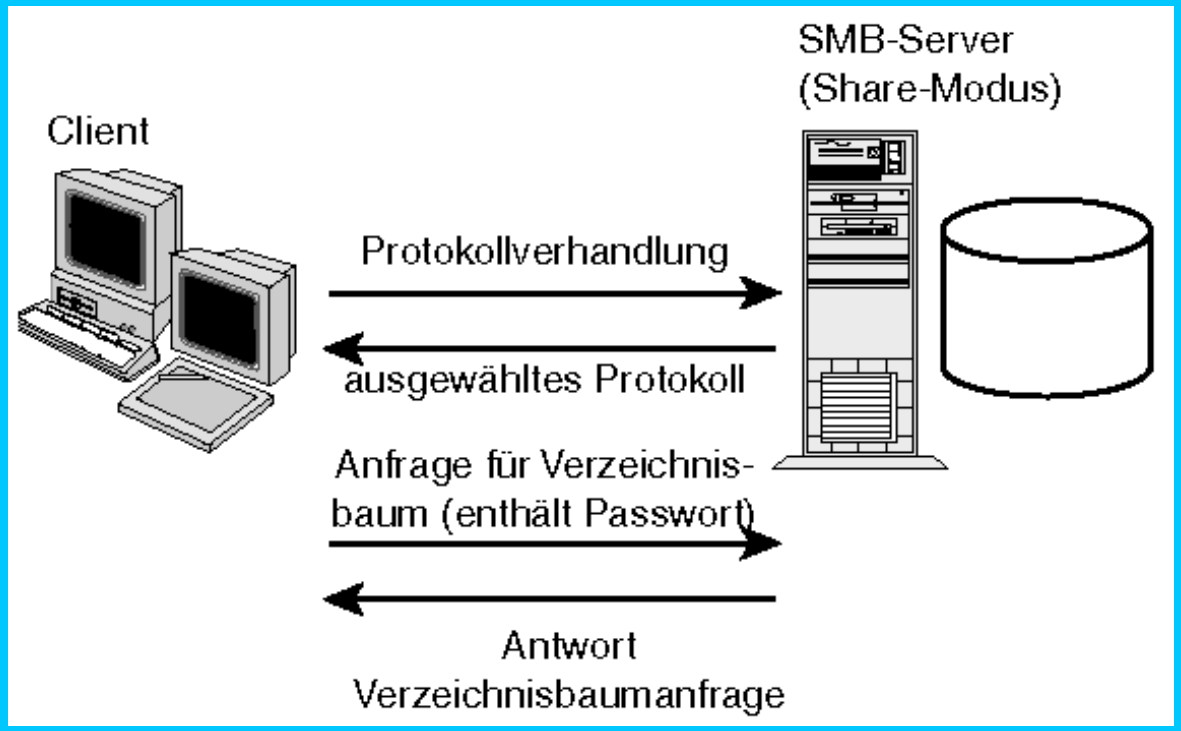


Derzeit wird experimentelle Unterstützung für den Zugriff auf NIS+-Tabellen und einen *Lightweight-Directory-Access-Protocol-(LDAP-)*Server entwickelt. Von den in Abbildung 6.1 dargestellten Möglichkeiten werden derzeit die Samba-`private/smbpasswd`-Datei und die Standard-Unix-`/etc/passwd`-Datei unterstützt. Beide Methoden werden später in diesem Kapitel ausführlich dargestellt. Jetzt sollten Sie nur wissen, dass beide Methoden für die Authentifizierung einen Benutzernamen und ein Passwort verlangen.

security = share

Im Share-Modus (Freigabeebene) sendet der Client während der Verbindungsanfrage ein Passwort. Ein zugehöriger Benutzername ist nicht erforderlich. Dies unterscheidet sich etwas von der Beschreibung, die ich in Kapitel 2, »Windows-Netzwerke«, gegeben habe, die eher auf den User-Modus zutrifft. Abbildung 6.2 verdeutlicht die zwei Schritte, die im Share-Modus für eine Verbindung zu einem SMB-Server benutzt werden.

Abb. 6.2: Während der Verbindungsanfrage wird im Share-Modus das Passwort übertragen



Möglicherweise kennen Sie bereits ein Beispiel für einen SMB-Server im Share-Modus. Der Share-Modus ist die Standardeinstellung für einen Datei- oder Drucker-Server unter Windows 95 (siehe Abbildung 6.3). Abbildung 6.4 zeigt das Dialogfeld in der Netzwerkfreigabe, über das Sie den Sicherheitsmodus ändern können.

Abb. 6.3: Zugriffssteuerung auf Freigabeebene (Share-Modus) unter Windows 95

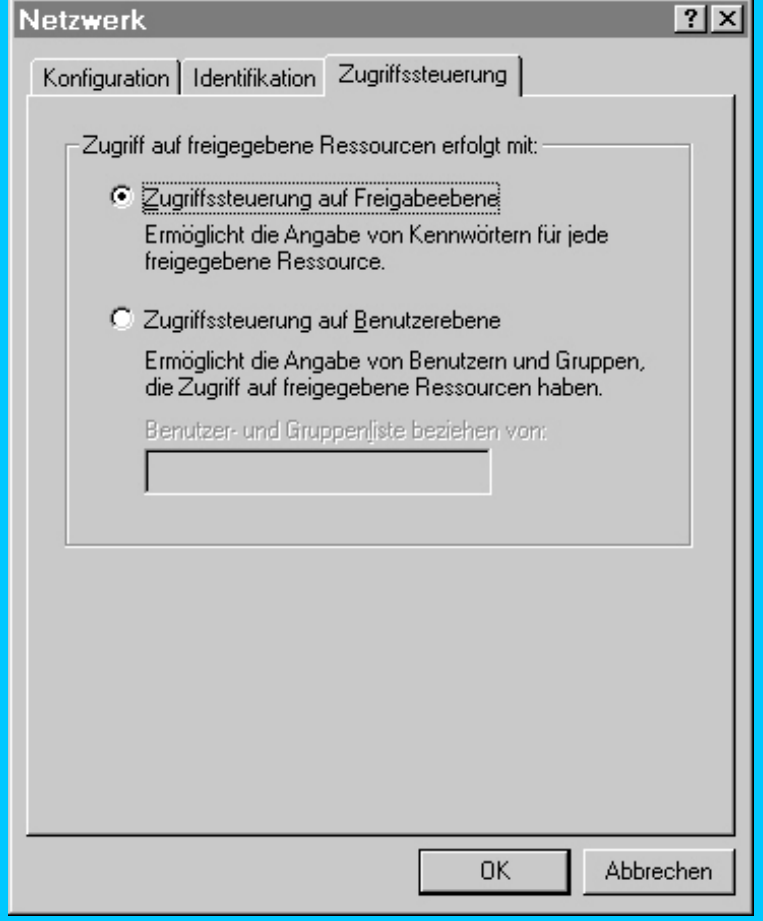
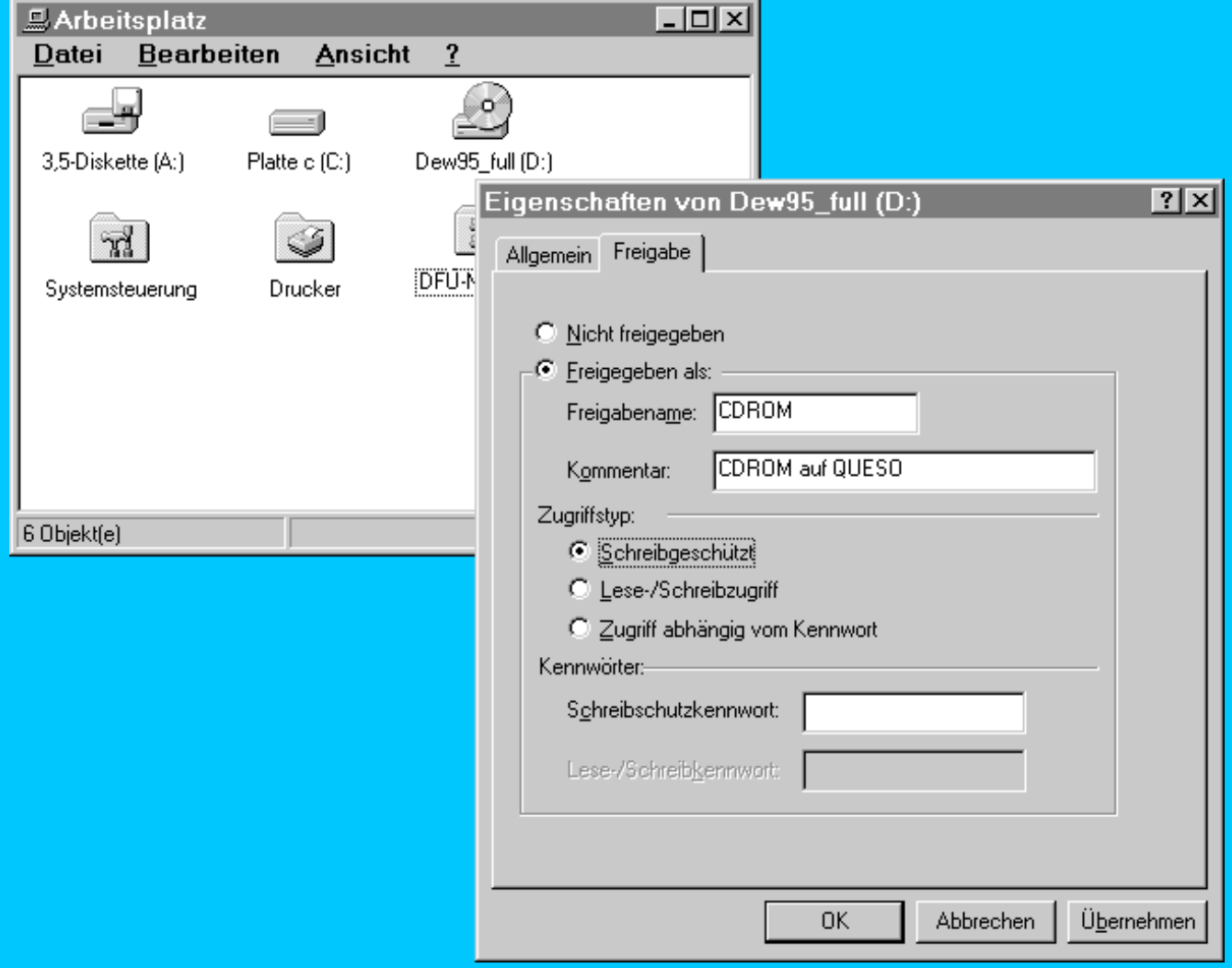


Abb. 6.4: Die Sicherheitsebene unter Windows 95 wählen



Vielleicht denken Sie mittlerweile: »Die Zugriffskontrolle auf Freigabeebene scheint dem Authentifizierungsmodell Benutzername/Passwort unter Unix zu widersprechen.« Sie haben Recht. Das Konzept des Share-Modus funktioniert in einer Multiuser-Umgebung wie Unix nicht gut, aber Samba versucht weitestgehend, das Unix-Sicherheitsmodell nicht zu beeinträchtigen.

Obwohl der Client erwartet, dass der SMB-Server im Share-Modus jeder Freigabe ein Passwort zuordnet, benutzt Samba das Standard-Unix-Benutzername/Passwort-Schema. Trotz der Tatsache, dass Clients der Verbindungsanfrage an einen Server im Share-Modus ein Passwort beifügen, übertragen viele Clients außerdem eine Sitzungsanfrage, die einen Benutzernamen enthält. Samba fügt diesen Namen einer Namensliste hinzu und versucht, ihn über das übertragene Passwort zu authentifizieren. Sie können andere Benutzernamen spezifizieren, die in die Liste aufgenommen werden sollen, indem Sie für die Freigabe den Parameter `user` definieren:

```
user = jerryc, smbguest, jdoe
```

Samba versucht die Verbindung zu authentifizieren, indem es jeden Benutzernamen mit dem Passwort vergleicht, bis eine Entsprechung gefunden wird oder auch nicht; dann wird die Verbindung zur Freigabe abgelehnt. Da Samba sowieso immer versucht, eine Kombination aus Benutzername und Passwort zu authentifizieren, wird die Zugriffskontrolle im Share-Modus nicht empfohlen. Es ist generell besser, eine Form der Authentifizierung auf Benutzerebene, die ich im nächsten Abschnitt darstelle, zu verwenden.

security = user

In Kapitel 2 (im Abschnitt »Protokollüberblick«) versucht ein Client, sich mit einem Server im User-Modus zu verbinden. Werden unverschlüsselte Passwörter benutzt, überträgt der Client während der Sitzungsaufnahme einen Benutzernamen und ein Passwort.

Da es in diesem Buch um die Benutzung von Samba und nicht um die Entwicklung eines SMB-Servers geht, werde ich Sie nicht sehr oft mit Paketausgaben langweilen. Aber ich denke, es ist recht nützlich, die Account-Informationen zu sehen, die während dieser Phase der Verbindung übertragen werden. Nachfolgend finden Sie eine Sitzungsanfrage von einem Windows-95-OSR2-Client, der sich mit einem Samba-Server verbindet. Die Pakete wurden über eine SMB-aktivierte Version von `tcpdump` abgefangen:

```
C:\\WINDOWS> net use h: \\bilbo\boss
Der Befehl wurde erfolgreich ausgeführt.
```

`tcpdump` ist ein Netzwerkpaket-Sniffer, der mit Source-Code verteilt wird. Die SMB-aktivierte Version können Sie sich unter http://samba.org herunterladen. Weitere Informationen über Paket-Sniffer finden Sie in Kapitel 11, »Troubleshooting«.

Nach Ausführung des Befehls generiert `tcpdump` folgende Ausgabe:

```
SMB PACKET: SMBsesssetupX (REQUEST)
SMB Command = 0x73
Error class = 0x0
Error code  = 0
Flags1      = 0x10
Flags2      = 0x0
Tree ID     = 0
Proc ID     = 28754
UID         = 1
MID         = 3586
Word Count  = 13
Com2=0x75
Res1=0x0
Off2=125
MaxBuffer=2920
MaxMpx=50
VcNumber=0
SessionKey}0xBE
CaseInsensitivePasswordLength=
[000] 54 45 53 54 50 41 53 53  00 00 00 00 00 00 42 4F  TESTPASS ......BO
[010] 53 53 00 00 00 00 00 00  42 4F 53 53 00 43 48 49  SS...... BOSS.CHI
[020] 50 53 4E 44 49 50 53 00  57 69 6E 64 6F 77 73 20  PSNDIPS. Windows
[030] 34 2E 30 00 57 69 6E 64  6F 77 73 20 34 2E 30 00  4.0. Windows 4.0
```

Sie können erkennen, dass das Passwort `testpass` und der Benutzername `boss` während der Sitzungsanfrage übertragen werden. Wenn Sie genau hinsehen, werden Sie auch feststellen, dass der Benutzername und das Passwort in Großbuchstaben umgewandelt werden. Dies kann ärgerlich sein und wird im Abschnitt »Passwortverschlüsselung« später in diesem Kapitel dargestellt.

Im User-Modus akzeptiert Samba die übermittelte Kombination aus Benutzername und Passwort und versucht, diese unter Benutzung seiner Account-Datenbank zu authentifizieren. Dieser Prozess ist unabhängig vom Backend des Benutzer-Accounts (z.B. verschlüsselte Passwörter, LDAP und `/etc/passwd`) immer gleich, obwohl der Beweis der Identität auch ein abgeleiteter Wert statt des eigentlichen Passworts selbst sein kann. Bitte beachten Sie die Hinweise zur SMB-Challenge/Response-Verschlüsselung im Abschnitt »Passwortverschlüsselung« später in diesem Kapitel. Ist die Authentifizierung der Sitzungsanfrage erfolgreich, braucht der Client während nachfolgender Verbindungsanfragen keine Benutzer-Account-Informationen zu übertragen.

In Abbildung 6.5 sind die drei Schritte für eine Verbindung zu einem SMB-Server im User-Modus dargestellt. Zunächst wird der Protokolldialekt ausgewählt, dann eine Sitzung zwischen dem Client und dem Server aufgebaut und schließlich die Verbindung zur Ressource konfiguriert.

security = server

Der Server-Modus von Samba ist im Prinzip eine Variante des User-Modus. Samba teilt dem Client mit, dass es sich im User-Modus befindet, und der Client führt eine normale Sitzungsanfrage durch. Samba nimmt dann die Informationen und sendet eine Sitzungsanfrage an den Rechner, der als Passwort-Server fungiert. Befindet sich der Passwort-Server im User-Modus und akzeptiert die Sitzungsanfrage, akzeptiert Samba die ursprüngliche Sitzungsanfrage des Clients.

Abb. 6.5: Verbindungsanfrage im User-Modus

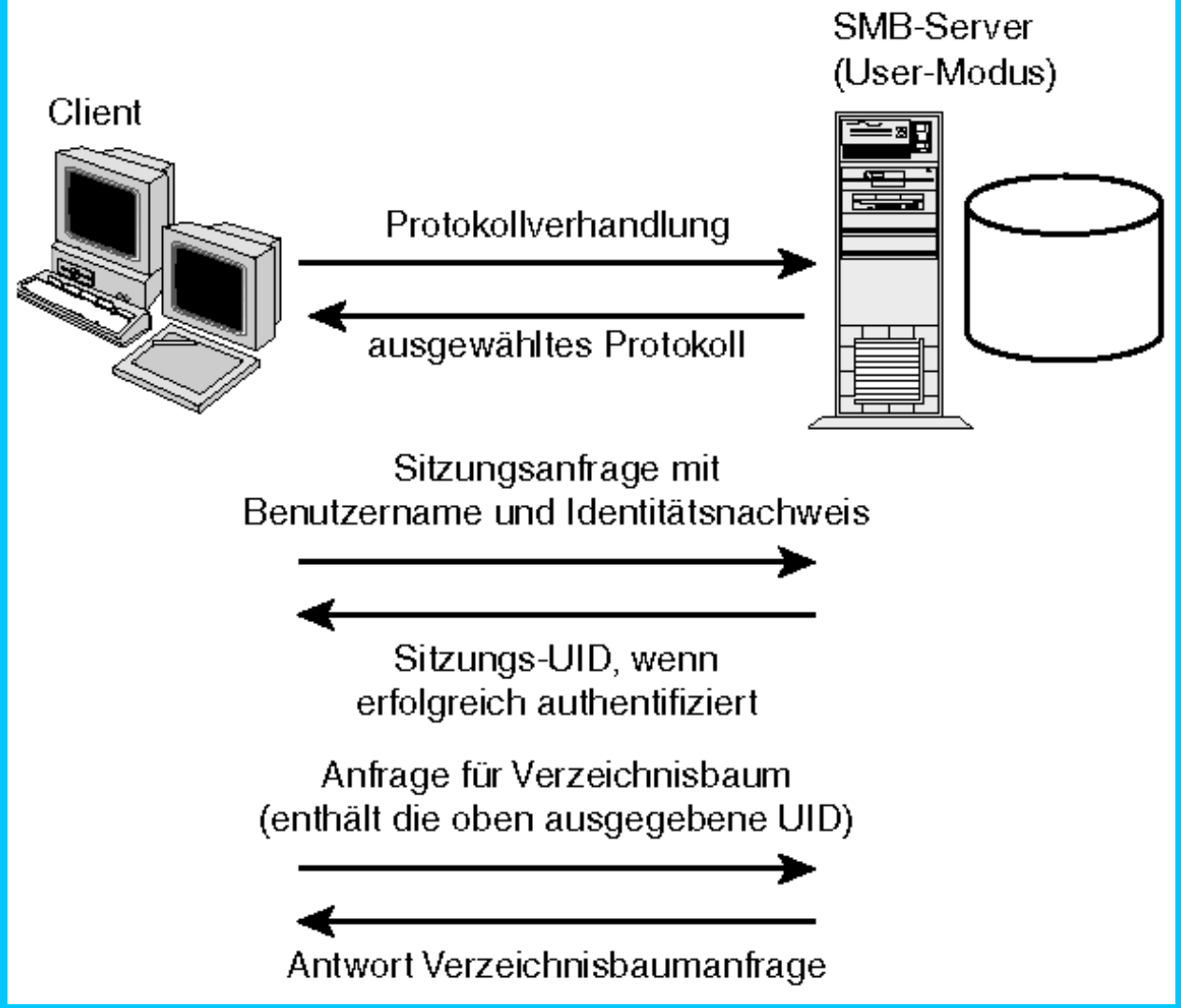
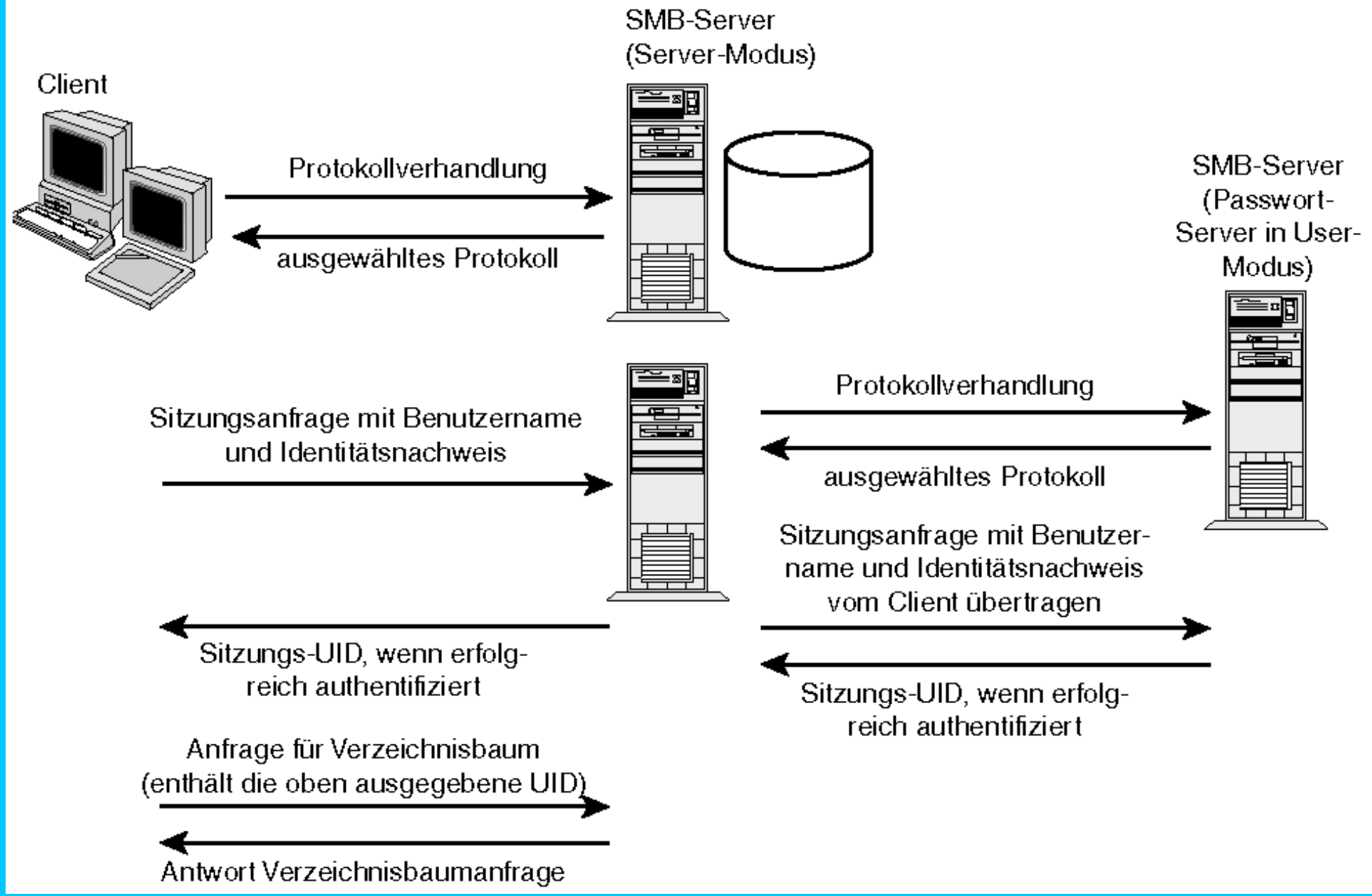


Abbildung 6.6 illustriert diesen Prozess. Die chronologische Reihenfolge des Diagramms ist von oben nach unten. Wenn Sie den Pfeilen folgen, sehen Sie, dass an dem Punkt, an dem der Client eine Sitzung mit dem Server verlangt, dieser eine Sitzungsanfrage an den Passwort-Server sendet. Erst wenn der Server eine Antwort vom Passwort-Server erhalten hat, wird die Anfrage des Clients akzeptiert oder abgelehnt.

Abb. 6.6: Ein Client, der sich mit einem Samba-Server im Server-Modus verbindet



Der Parameter `password server` hat folgende Syntax:

```
password server = NetBIOS-Name des SMB-Servers
```

Sie können mehrere NetBIOS-Namen auflisten, z.B.

```
password server = DOMAINPDC DOMAINBDC1 DOMAINBDC2
```

Mit dieser Einstellung kann Samba versuchen, jedem aufgelisteten Server nacheinander eine Sitzungsanfrage zu senden, bis ein Server antwortet. Das heißt, der nächste

Rechner in der Liste wird nur dann kontaktiert, wenn der vorstehende Rechner nicht erreichbar war. Es heißt nicht, dass Samba versucht, die anderen aufgelisteten Rechner zu kontaktieren, wenn die Verbindungsanfrage an den ersten Rechner scheitert.

Sie müssen den NetBIOS-Namen des Passwort-Servers benutzen (nicht die IP-Adresse), und Samba muss eine Möglichkeit haben, den Namen in eine IP-Adresse aufzulösen, um eine Verbindungsaufnahme zu versuchen.

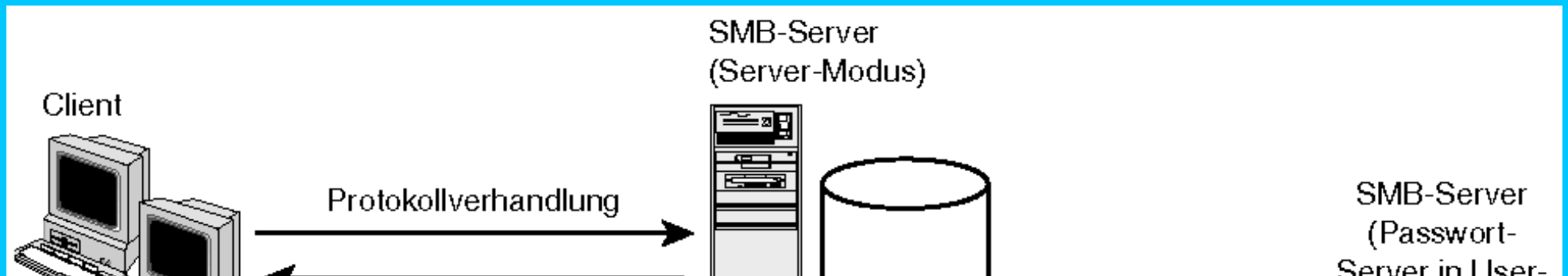
Sie können jeden SMB-Server im User-Modus als Passwort-Server verwenden, aber die Sicherheit Ihres Samba-Servers entspricht dann nur der des augewählten Passwort-Servers. Ich habe Sie gewarnt! Übliche Wahlen für einen Passwort-Server sind Ihr Windows NT Primary Domain Controller (PDC) oder ein anderer Samba-Rechner.

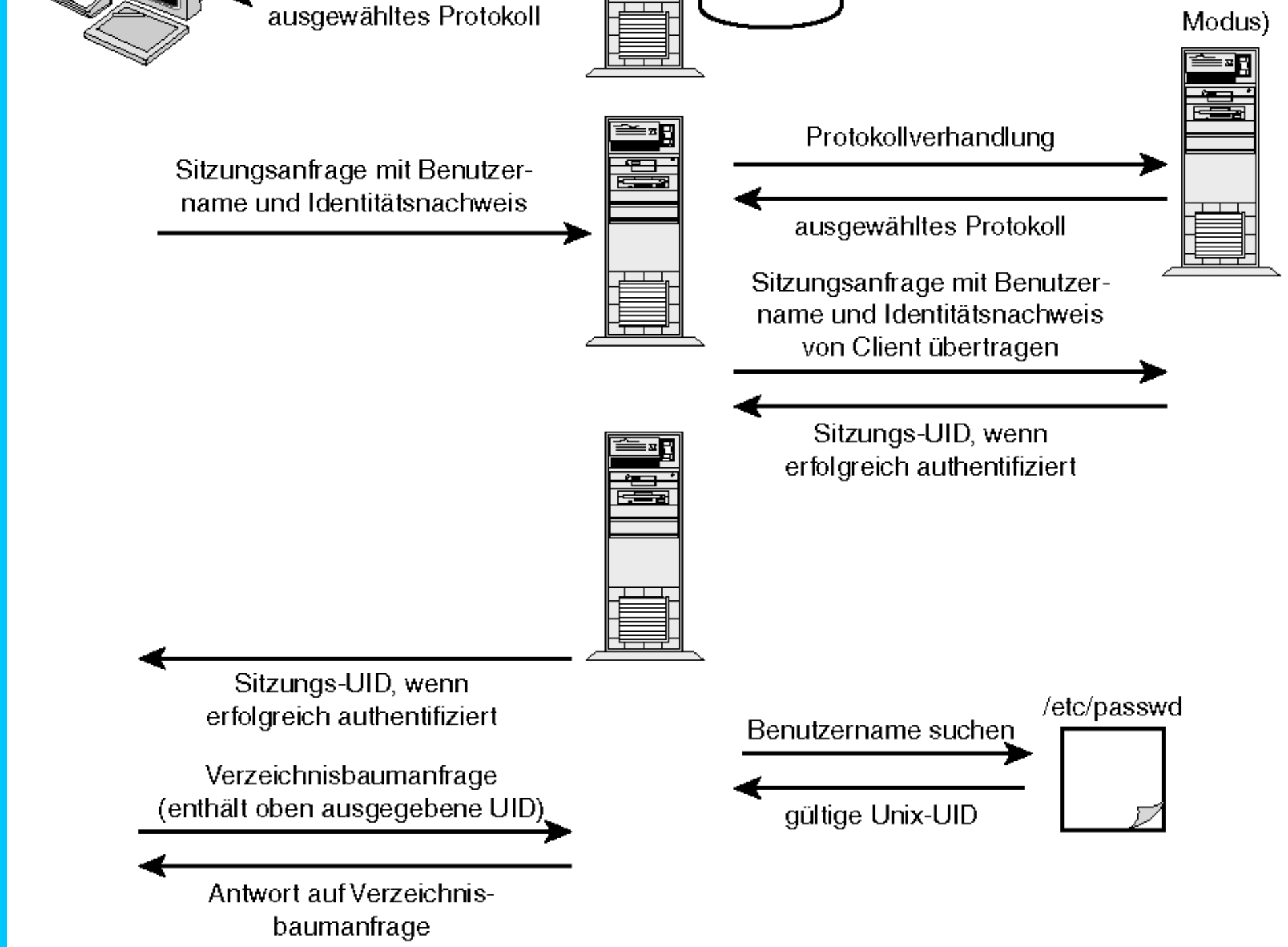
Der Server-Modus hat eine Besonderheit. Nachdem Samba die Sitzungsanfrage für den Client gewährt hat, muss es eine Methode haben, eine Unix-UID für den Benutzer zu bekommen, um den Zugriff auf Dateien kontrollieren zu können. Das heißt, dass zwar keine lokalen Accounts für die Authentifizierung der Verbindung verwendet werden, aber der Benutzer muss eine UID auf dem lokalen Server haben. Es gibt zwei mögliche Lösungen für dieses Problem:

- Sie können einen lokalen Account für alle Benutzer einrichten, die auf den Samba-Server zugreifen und einfach das Passwortfeld in der `/etc/passwd` (oder der Datei, in der die Passwörter gespeichert sind) deaktivieren. Normalerweise erreichen Sie dies, indem Sie das Sternchen (`*`) in das Feld `password` setzen.
- Sie können eine der Methoden benutzen, über die Benutzernamen innerhalb einer Freigabe zugeordnet werden, z.B. `force user`, die in Kapitel 7, »Dateifreigaben«, beschrieben werden. Oder verwenden Sie den Parameter `username map`, den ich später in diesem Kapitel darstelle.

Abbildung 6.7 sieht Abbildung 6.6 sehr ähnlich, da beide Abbildungen den gleichen Prozess darstellen. Abbildung 6.7 wurde aber um den Punkt erweitert, an dem Samba versucht, eine gültige Unix-UID für den in der Sitzungsanfrage angegebenen Benutzernamen zu erhalten. In einem gewissen Sinne transparent ist, dass die Suche nach dem Benutzernamen durch den Parameter `username map` gefiltert werden kann, bevor der Name tatsächlich in der `/etc/passwd` gesucht wird.

Abb. 6.7: Einem authentifizierten Benutzernamen wird eine Unix-UID zugeordnet




ausgewähltes Protokoll
Modus)
Protokollverhandlung
Sitzungsanfrage mit Benutzer-
name und Identitätsnachweis
ausgewähltes Protokoll
Sitzungsanfrage mit Benutzer-
name und Identitätsnachweis
von Client übertragen
Sitzungs-UID, wenn
erfolgreich authentifiziert
Sitzungs-UID, wenn
erfolgreich authentifiziert
/etc/passwd
Benutzername suchen
Verzeichnisbaumanfrage
(enthält oben ausgegebene UID)
gültige Unix-UID
Antwort auf Verzeichnis-
baumanfrage

security = domain

Der Domain-Modus von Samba entspricht im Wesentlichen dem Konzept des Server-Modus, mit der Ausnahme, dass der Samba-Server Mitglied einer Windows-NT-Domäne wird. Das heißt, der Samba-Server kann an Dingen wie vertrauten Beziehungen teilnehmen. Es gibt einige weitere Vorteile für die Benutzung von `security = domain` statt `security = server`. Diese Darstellung möchte ich auf Kapitel 12, »Fallstudie: Einen NT-Datei- und Drucker-Server ersetzen«, verschieben, in dem ich beschreibe, wie ein Windows-NT-Datei- und Drucker-Server durch einen Samba-Rechner im Domain-Modus ersetzt wird. Bis dahin betrachten Sie bitte die zwei Modi als äquivalent.

Benutzernamen und Passwörter

Jetzt wissen Sie, wie `smbd`-Verbindungen authentifiziert werden, und können sich auf die Details von Benutzernamen und Passwörtern konzentrieren.

Username Level

Wie Sie bereits in der Paketausgabe von einer Sitzungsanfrage gesehen haben, übertragen einige Clients den Benutzernamen komplett in Großbuchstaben. Standardmäßig versucht Samba, diesen Benutzernamen klein geschrieben zu suchen und dann nur mit dem ersten Buchstaben groß geschrieben, z.B. `boss` und `Boss`. Wenn Sie einen seltsamen Unix-Benutzernamen haben, wie z.B. `BobAcct`, der Bob in der Abteilung *Accounting* zugewiesen ist, kann Samba mit dieser Methode den Benutzernamen nicht finden.

Darum gibt es einen Parameter, über den die maximale Anzahl von Großbuchstaben in dem Benutzernamen bestimmt werden kann. Samba versucht dann, über eine Brute-Force-Methode den Benutzernamen zu finden, indem es alle Abwandlungen mit Großbuchstaben von 1 bis zum definierten Wert ausprobiert.

Stellen Sie den Parameter `username level` auf `4` ein und wenden Sie diese Einstellung auf Bobs Account-Namen an:

```
username level = 4
```

Sie können voraussetzen, dass als Benutzername während der Sitzungsanfrage `BOBACCT` übertragen wird. Samba versucht, folgende Namen in der Systempasswortdatei (oder anderen benutzten Backends) zu finden:

```
bobacct
Bobacct
bObacct
boBacct
bobAcct
bobaCct
bobacCt
bobaccT
Bobacct
BoBacct
BobAcct
```

Die Suche wird beendet, sobald der Benutzername gefunden ist. Je höher der Wert für `username level`, um so mehr Kombinationen von Groß-/Kleinbuchstaben werden ausprobiert und um so länger dauert es, bis die Suche erfolgreich oder auch nicht abgeschlossen ist. Sind alle Unix-Account-Namen im Standardformat klein geschrieben, wird dieser Parameter unnötig.

Username Map

Eines der Hauptprobleme bei der Integration von Unix- und PC-Betriebssystemen besteht in der Synchronisierung der Informationen zu den Benutzer-Accounts. Einige Unix-Varianten lassen nur die Benutzung von acht Zeichen oder weniger für den Benutzernamen zu, während einige Windows-Clients eine beliebige Zeichenkette mit Leerzeichen erlauben. Oft finden sich Administratoren mit der Aufgabe beschäftigt, zwei bereits etablierte Systeme mit existierenden Account-Namen zu integrieren. Mit dem Parameter `username map` können Sie eine Datei spezifizieren, die den während einer Verbindungsanfrage übertragenen Benutzernamen einem lokalen Benutzernamen zuordnet.

Diese Option ist standardmäßig nicht aktiviert.

Standard: `username map = none`

Um Zuordnungen zu verwenden, müssen Sie den Standort der Datei spezifizieren, in der die Zuordnungen enthalten sind:

`Beispiel: username map = /usr/local/samba/lib/users.map`

Jeder Eintrag in der Datei sieht so aus:

`Unix Benutzername = Client-Benutzername ...`

Wenn Sie z.B. den Benutzernamen `Administrator` oder `Admin` dem Account `sysadmin` zuordnen wollen, definieren Sie folgenden Eintrag:

```
sysadmin = Administrator Admin
```

Wenn ein Benutzer versucht, sich als `Administrator` mit einer Freigabe zu verbinden, muss er das Passwort für den Account `sysadmin` übermitteln.

Die Zuordnung wirkt sich auf alle Beispiele des Client-Benutzernamens aus, mit Ausnahme der Sitzungsaufnahme zu einem Passwort-Server, wenn `securitiy = server` gesetzt ist. Um bei obigem Beispiel zu bleiben: Wenn ein Benutzer versucht, sich mit dem Home-Verzeichnis von `Administrator` zu verbinden, würde er sich tatsächlich mit `\\server\sysadmin` verbinden.

Es ist möglich, Unix-Gruppen einem einzelnen Account zuzuordnen. Diese Zeile ordnet jeden Benutzer in der Gruppe `staff` dem Account `staffsmb` zu:

```
staffsmb = @staff
```

Es steht außerdem eine Wildcard zur Verfügung, über die Sie jeden vom Client übertragenen Namen zuordnen können. Dieser Eintrag ordnet alle Benutzer dem Account `guest` zu:

```
guest = *
```

In Hinsicht auf Abbildungen gibt es einen Punkt, den Sie beachten sollten: `smbd` analysiert die Datei Zeile für Zeile und führt eventuelle Abbildungen bis zum Ende der Datei durch. Dies kann zu mehrfachen Zuordnungen führen, von `Benutzername` zu `neuer_Benutzername1` zu `neuer_Benutzername2`. Wenn Sie die Analyse der Datei stoppen wollen, nachdem eine Zuordnung erfolgt ist, sollten Sie der Zeile ein Ausrufungszeichen (`!`) voranstellen. Wird keine Zuordnung in der Datei gefunden, benutzt Samba den ursprünglichen Benutzernamen.

Password Level

Werden für die Authentifizierung Klartextpasswörter benutzt, tauchen ähnliche Probleme auf wie die, die ich im Abschnitt über Benutzernamen in Bezug auf die Groß-/Kleinschreibung erwähnt habe. Dieser Ausschnitt aus der vorher gezeigten `tcpdump`-Ausgabe erinnert Sie daran, dass das Passwort, `testpass`, in Großbuchstaben übertragen wird:

```
[000] 54 45 53 54 50 41 53 53  00 00 00 00 00 00 42 4F  TESTPASS ......BO
[010] 53 53 00 00 00 00 00 00  42 4F 53 53 00 43 48 49  SS...... BOSS.CHI
[020] 50 53 4E 44 49 50 53 00  57 69 6E 64 6F 77 73 20  PSNDIPS. Windows
[030] 34 2E 30 00 57 69 6E 64  6F 77 73 20 34 2E 30 00  4.0. Windows 4.0
```

Der Parameter `password level` hat in etwa die gleiche Funktion wie der Parameter `username level`. Der Unterschied liegt darin, dass standardmäßig zwei Möglichkeiten für die Verwendung des Passworts ausprobiert werden: so, wie es vom Client übertragen wird, und komplett klein geschrieben.

Wie der Parameter `username level` nimmt auch `password level` als Wert eine ganze Zahl an, die die maximale Anzahl von Großbuchstaben definiert, die für das Passwort zugelassen sind. Samba versucht dann, den Benutzernamen zu authentifizieren, indem es Abwandlungen der Großbuchstaben im Passwort benutzt.

Je größer der Wert und je mehr Kombinationen Samba ausprobiert, um so länger dauert die Authentifizierungsphase. Sie müssen bestimmen, was für Ihren Server akzeptabel ist. Ein `password level = 8` bedeutet auf den meisten Systemen, dass beim Passwort nicht mehr zwischen Groß- und Kleinschreibung unterschieden wird. Ich habe festgestellt, dass die Einstellung 4 in der Regel akzeptabel ist und nicht allzu viele Umstände für existierende Passwörter macht. Es ist jedoch auch hilfreich, die Richtlinien für die Benutzung von Passwörtern dahingehend zu ändern, dass nicht mehr als vier Großbuchstaben verwendet werden dürfen.

Passwortverschlüsselung

Samba unterstützt sowohl die LanManager- als auch die Windows-NT-SMB-Passwort-Verschlüsselungsalgorithmen. Das heißt, Samba kann Benutzer auf die gleiche Art und Weise authentifizieren wie Microsoft-Server es können.

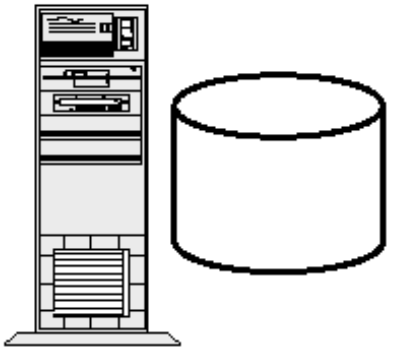
Wenn Sie mit der Verschlüsselung von Passwörtern unter Unix vertraut sind, erscheinen Ihnen einige Punkte vielleicht ähnlich. Z.B. sind die LanMan- und NT-Passwort-Hashwerte unwiderruflich, ebenso wie Unix-Passwörter, die in `/etc/passwd` (oder `/etc/shadow`) gespeichert sind. Unwiderruflich heißt, dass Sie nur dann feststellen können, ob ein Benutzer das korrekte Passwort eingegeben hat, wenn Sie das eingegebene Passwort verschlüsseln und diesen Wert mit der verschlüsselten Version vergleichen, die auf der Festplatte gespeichert ist. Es gibt keine Möglichkeit, einen LanMan/NT-Hash zu entschlüsseln außer über Brute-Force-Methoden, wie z.B. einen Wörterbuchangriff.

Sie sollten einen großen Unterschied zwischen den Unix- und LanMan/NT-Verschlüsselungsalgorithmen beachten. Der Algorithmus zur Erzeugung eines LanManager- oder NT-Passwort-Hashwerts produziert bei gleicher Eingabe immer das gleiche Resultat. Das heißt, wenn Sie das Passwort `testpass` zweihundertmal verschlüsseln, ist das verschlüsselte Passwort immer gleich. Dieser Prozess erzeugt, was als *Klartextentsprechung* bekannt ist.

Ich hoffe, dass das folgende Beispiel dies klarmacht. Sie können den Prozess in Abbildung 6.8 verfolgen.

Abb. 6.8: Beispiel für eine Challenge/Response-Authentifizierung zwischen einem Client und einem Server

(User-Modus)



1. Protokollverhandlung

2. ausgewähltes Protokoll(Verschlüsselungs-bit gesetzt und 8-Byte zufällige Chanllenge enthalten)

3. • Client verschlüsselt Passwort und hängt am Ende 40 Bit mit Nullen an
   • bricht den 168-Bit-Stream in drei 56-Bit-Schlüssel auf
   • verschlüsselt die 8-Byte-Challenge mit jedem der 56-Bit-Schlüssel, sodass eine 24-Byte-Antwort entsteht

3. Sitzungsanfrage mit Benutzer-name und 24-Byte-Antwort

4. Der Server benutzt die verschlüsselte Version des Benutzerpassworts, das auf der Festplatte gespeichert ist, um einen 24-Byte-Stream zu erzeugen, der mit der Antwort des Clients verglichen wird. Stimmen die Werte überein, wird der Client authentifiziert.

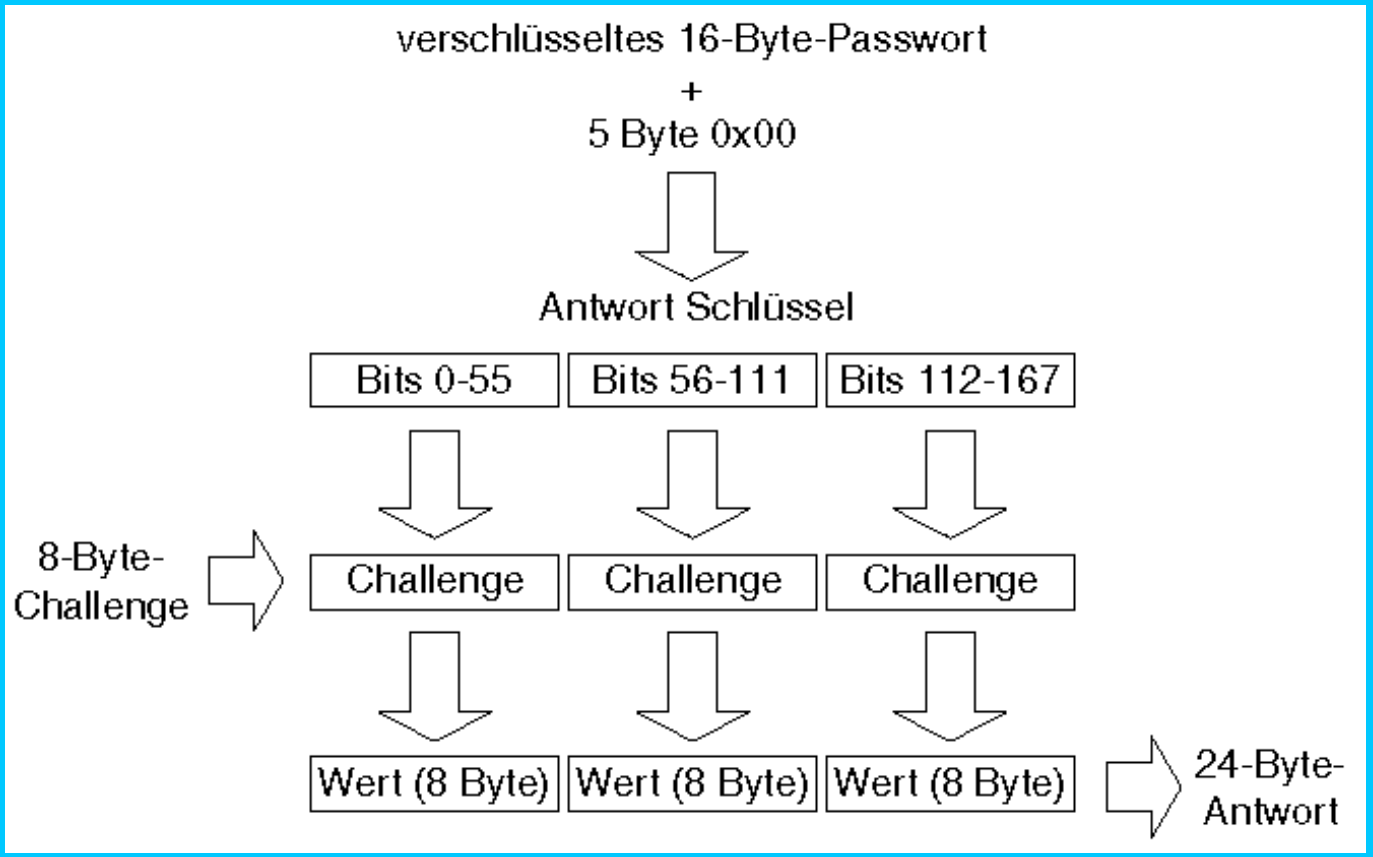
5. Sitzungs-UID, wenn erfolgreich authentifiziert

Anfrage für Verzeichnisbaum (enthält die oben ausgegebene UID)

Antwort Verzeichnisbaumanfrage

1. Der Client sendet eine Anfrage für eine Protokollverhandlung an den Server.

2. Unterstützt der Server verschlüsselte Passwörter, wird das entsprechende Bit im Response-Paket übertragen, und der Server fügt dem Paket eine 8-Bit-Challenge hinzu. Diese Aufforderung wird zufällig erzeugt und ist für jeden Client verschieden.
3. Abbildung 6.9 illustriert die Generierung der Client-Antwort. Der Client benutzt das verschlüsselte Passwort, das dem verhandelten Protokolldialekt entspricht (entweder LanMan oder NT) und hängt fünf Null-Bytes an (dies erzeugt einen 168-Bit-Stream), um drei verschiedene 56-Bit-DES-Schlüssel zu erzeugen, die dann jeweils für die Verschlüsselung der 8-Byte-Challenge benutzt werden. Die drei 8-Byte-Resultate werden zusammengefasst und bilden die 24 Byte lange Antwort, die an den Server übertragen wird. *Abb. 6.9: Die 24 Byte lange Antwort generieren*



4. Der Server führt dann unter Benutzung der verschlüsselten Version des Benutzerpasswords, die auf der Festplatte gespeichert ist, die gleichen Schritte durch. Das Resultat wird mit dem vom Client übertragenen Wert verglichen, um zu überprüfen, ob der Client das korrekte Passwort benutzt hat.
5. Stimmen der 24-Byte-Wert des Servers und die vom Client übermittelte Antwort überein, wird die Sitzungsanfrage (oder Freigabeverbindung im Falle des Share-Modus) akzeptiert. Stimmen sie nicht überein, hat der Client nicht das korrekte Passwort übertragen.

Machen Sie sich keine Gedanken, wenn Sie den Prozess nicht Wort für Wort wiederholen können. Ich habe ihn hier nur dargestellt, um einen Punkt zu beweisen. Das Passwort des Benutzers wird niemals über das Netzwerk übertragen. Das sorgt für erhöhte Sicherheit. Es werden nur die Daten übertragen, die aus dem Passwort generiert wurden.

Kommen wir nun zurück zu meinem vorangegangenen Kommentar über Klartextentsprechungen von Passwörtern. Der Server muss das verschlüsselte Passwort irgendwo speichern, damit er den 24-Byte-Wert generieren kann, um die Antwort des Clients zu authentifizieren. Denken Sie daran, dass das Passwort immer auf den gleichen Wert verschlüsselt wird. Wenn also jemand die verschlüsselte Version des Passworts kennt, kann diese Person an dem vorher beschriebenen Prozess teilnehmen, ohne das Passwort jemals kennen zu müssen!

Sind das zu viele Informationen auf einmal? Vielleicht können einige dieser Punkte helfen, sich für verschlüsselte oder Klartextpasswörter zu entscheiden:

- Mit Klartextpasswörtern kann Samba die gleiche Passwortdatenbank (d.h. die `/etc/passwd`) benutzen wie andere Unix-Dienste, z.B. login und FTP. Diese Dienste übermitteln Passwörter oft auch in Klartext über das Netzwerk. Samba überträgt also keine Benutzer-Account-Informationen, die nicht sowieso schon über das Netzwerk geschickt werden.
- Wenn Sie Klartextpasswörter benutzen, brauchen Sie nichts anderes als normale Unix-Systemdateien, die auf der Festplatte gespeichert werden.
- Windows NT ab SP3 mag keine Klartextpasswörter, und Sie können keinen Server browsen, der Verschlüsselung nicht unterstützt. NT fordert außerdem die Eingabe eines Passworts, wenn Sie sich mit nicht verschlüsselten Freigaben verbinden wollen, was bei häufigen Freigabeverbindungen extrem ärgerlich werden kann.
- Die Abstimmung zwischen `smbpasswd` und `unix passwd` kann schwierig sein. Weitere Informationen hierzu finden Sie in Kapitel 16, »Passwortsynchronisierung«.
- Verschlüsselte Passwörter können nicht von jemandem gelesen werden, der Zugriff auf die zwischen Client und Server übertragenen Pakete hat. Wenn Sie Klartextpasswörter benutzen, können die übertragenen Informationen über einfache Netzwerkanalyse-Tools wie `tcpdump` eingesehen werden.

Wenn Sie sich für verschlüsselte Passwörter entscheiden, die standardmäßig nicht aktiviert sind, können Sie die Funktion über folgende Einstellung in Ihrer `smb.conf` aktivieren:

```
encrypt passwords = yes
```

Haben Sie die Verschlüsselung von Passwörtern aktiviert, müssen Sie nun eine zweite Benuter-Account-Datei im Auge behalten. In dieser Datei, die normalerweise `smbpasswd` genannt wird und sich in einem Unterverzeichnis namens `private` unterhalb des Samba-Installationsverzeichnisses befindet, speichert Samba die LanMan- und NT-Hashwerte der Benutzerpasswörter. Das Format ist dem von `/etc/passwd` sehr ähnlich:

```
username:uis:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:account: flags:lastset:
```

Die Felder `username` und `uid` erklären sich von selbst. Die nächsten zwei Felder enthalten die zwei 16-Byte-Hashwerte des Benutzernamens, die von LanMan bzw. NT generiert wurden. Das Feld `account flags` bestimmt den Typ des Accounts wie z.B. Benutzer-Account oder Rechner-Account (Rechner-Accounts werden in Anhang A, »Experimentelle PDC-Unterstützung«, näher dargestellt). Das Feld `lastset` zeichnet den Zeitpunkt der letzten Passwortänderung auf.

Hier ein Beispieleintrag:

```
jerryc:1009:AAD3B435B51404EEAAD3B435B51404EE:31D6CFE0D16AE931B73C59D7E0C089C0:[U      ]:LCT-36918AD9:
```

Wollen Sie die Datei, die die verschlüsselten Passwörter enthält, an einem anderen Ort speichern oder sie umbenennen, können Sie die neuen Werte über den Parameter `smb passwd file` definieren. Der Wert sollte ein absoluter Pfad zur SMB-Passwortdatei wie der folgende sein:

```
smb passwd file = /etc/smbpasswd
```

Das Erzeugen der ursprünglichen SMB-`passwd`-Datei und das Einrichten der Passwörter kann eine extrem erschreckende Aufgabe sein, wenn Sie viele existierende Unix-Accounts haben.

Es gibt hier zwei übliche Lösungen. Beide verlangen, dass Sie zunächst einen ersten `smbpasswd`-Eintrag für jeden Benutzer einrichten. Dies kann ganz einfach gemacht werden, wenn Sie eines der Skripte benutzen, die in der Samba-Distribution enthalten sind:

```
cat /etc/password | mksmbpasswd.sh > /usr/local/samba/private/smbpasswd
```

Erhält der Unix-Rechner Account-Informationen von NIS oder NIS+, können Sie den oben stehenden Befehl `cat` je nach System entweder durch `ypcat` oder `niscat` ersetzen. Das Shellskript `mksmbpasswd.sh` befindet sich im Unterverzeichnis `source/script` der Samba-Distribution. Die resultierende `smbpasswd`-Datei enthält alle Benutzer aus `/etc/passwd` mit ihren LanMan- und NT-Hash-Passwortwerten, die auf 32 X eingestellt sind. Samba authentifiziert einen Benutzer, dessen Passworteintrag auf diesen Wert eingestellt ist, nicht.

Wollen Sie den Wert auf ein leeres Passwort setzen, müssen Sie

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

durch

```
NO PASSWORDXXXXXXXXXXXXXXXXXXXXX: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

ersetzen.

Geben Sie als `root` den folgenden Befehl aus:

```
/usr/local/samba/bin/smbpasswd -n Benutzername
```

Ersetzen Sie *`Benutzername`* durch den entsprechenden Benutzernamen. Samba speichert die verschlüsselten Passwörter in einer Datei namens `smbpasswd` und fügt ein Utility hinzu, das ebenfalls `smbpasswd` heißt, um die Einträge in der Datei zu manipulieren. Lassen Sie sich durch den Namen nicht verwirren.

Alternativ können Sie die `smbpasswd`-Datei manuell über einen Texteditor bearbeiten und die Eingabe selber ändern. Wenn Sie die `smbpasswd`-Datei tatsächlich manuell bearbeiten, sollten Sie jedoch sicherstellen, dass die LanMan- und NT-Passwortfelder 32 Zeichen enthalten, nicht mehr und nicht weniger. Haben die Felder nicht genau 32 Zeichen, kann Samba diesen Benutzer niemals authentifizieren.

Nachdem Sie den `smbpasswd`-Eintrag geändert haben, müssen Sie den folgenden Parameter `null passwords` im Abschnitt `[global]` der `smb.conf` auf `yes` setzen:

```
null passwords = yes
```

Nach Erzeugen der `smbpasswd`-Datei bleibt die Frage: »Wie fülle ich das `Passwortfeld für` jeden Eintrag?«

Lösung 1

Wenn Sie Samba derzeit mit Klartextpasswörtern benutzen, können Sie die verschlüsselten Passwortfelder nach und nach für jeden Benutzer füllen, indem Sie den booleschen Parameter `update encrypted` benutzen. Der Standardwert für diesen Parameter ist `no`. Um die Unterstützung zu aktivieren, müssen Sie im Abschnitt `[global]` der `smb.conf` folgenden Eintrag einfügen:

```
update encrypted = yes
```

Wenn Sie den Wert dieses Parameters auf yes setzen, müssen Sie sicherstellen, dass der Parameter `encrypt passwords` auf `no` gesetzt ist.

```
encrypt passwords = no
```

Ist der Parameter `update encrypted` auf `yes` gesetzt, schreibt Samba jedes Mal, wenn ein Benutzer erfolgreich eine Sitzungsaufnahme verlangt, die verschlüsselte Version des Klartextpassworts, das für diesen Benutzer übermittelt wurde. Die einzige Voraussetzung ist, dass der Benutzer einen gültigen Eintrag in der vorhandenen `smbpasswd`-Datei hat. Was immer der vorherige Wert des `passwd`-Felds war, es ist jetzt auf das aktuelle Passwort des Benutzers eingestellt. Natürlich hat diese Methode nur im User-Modus Sinn.

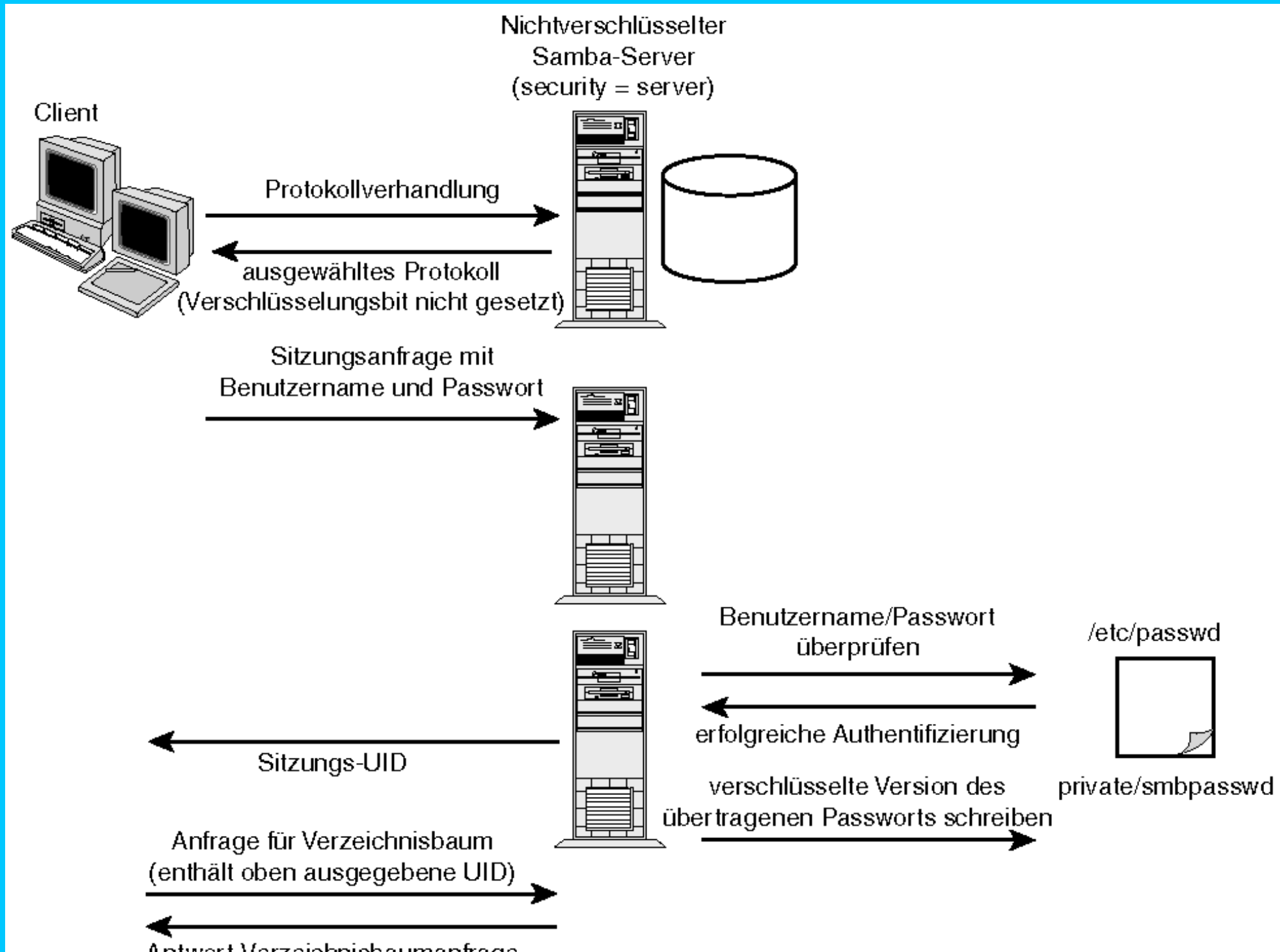
Mit dieser Lösung kann der Samba-Server einige Tage oder Wochen, wie lange auch immer er braucht, laufen, Passwörter abfangen und die `smbpasswd`-Datei ausfüllen. Enthält `smbpasswd` genügend Einträge, können Sie einfach die folgenden Parameter in `smb.conf` ändern und auf die Benutzung verschlüsselter Passwörter umschalten:

```
encrypt passwords = yes
update encrypted = no
```

Die meisten Ihrer Benutzer werden niemals erfahren, dass sich etwas geändert hat.

Abbildung 6.10 verdeutlicht, wie der Parameter `update encrypted` funktioniert. Zunächst wählen der Client und der Server den Protokolldialekt, dann sendet der Client in seiner Sitzungsanfrage den Benutzernamen und das Passwort in Klartext. Kann der Benutzer erfolgreich über `/etc/passwd` authentifiziert werden, verschlüsselt der `smbd` das Passwort und schreibt die Informationen in die `smbpasswd`-Datei. Sie sollten wissen, dass der Eintrag des Benutzers in der `smbpasswd` an diesem Punkt des Authentifizierungsprozesses niemals benutzt wird. Das verschlüsselte Passwort wird hier nur aufbewahrt.

Abb. 6.10: Über den Parameter update encrypted wird smbpasswd nach und nach gefüllt



Lösung 2

Ich habe vorher schon erwähnt, dass Samba ein Utility namens `smbpasswd` enthält, mit dem Sie Einträge in der `smbpasswd`-Datei manipulieren können. Dieses Tool finden Sie im Unterverzeichnis `/bin`. Es ist die Entsprechung des Unix-Programms `/bin/passwd`.

Wenn ein Benutzer einen neuen Unix-Account erhält, weisen die meisten Unternehmen ein zufälliges Passwort zu und zeigen dann dem Benutzer, wie er das Passwort ändern kann, damit es persönlicher oder leichter zu merken ist. Wenn Sie eine neue Samba-Infrastruktur aufbauen - das Wort könnte doch glatt aus einem Dilbert-Comic stammen, oder? -, könnten Sie Benutzern ganz einfach gleichzeitig mit ihrem neuen Unix-Account auch ein SMB-Passwort zuweisen. Zusammen mit den Standardanweisungen für die Änderung ihres Unix-Passworts über `/bin/passwd` könnten Sie Ihren Benutzern gleich die Anweisungen für die Änderung ihres SMB-Passworts über den Befehl `/usr/local/samba/bin/smbpasswd` mitgeben. Dies ist sicher die einfachste Lösung, da so die Verantwortung an den Benutzer übergeben wird, die Passwörter synchron zu halten, wenn dies gewünscht ist. Dies könnte aber je nach Kaliber Ihrer Benutzer zu vermehrten Anrufen bei den Supportleuten führen. Es ist sehr leicht durcheinander zu bringen, welches Passwort zu welchem Logon gehört, wenn die Accounts nicht mehr synchron sind. Ein Benutzer kann sehr unnachgiebig in seinem Glauben sein, dass er das korrekte Passwort für seinen Account eingegeben hat! Sie müssen selber entscheiden, welche Lösung für Sie die beste ist.

Nachfolgend finden Sie eine Beispielsitzung, in der ich mein SMB-Passwort über den Befehl `smbpasswd` ändere. Ich habe Kommentare in spitzen Klammern eingefügt, die Ihnen erklären, was eingegeben werden muss. (Sie haben doch nicht wirklich gedacht, dass ich Ihnen mein Passwort verrate, oder?)

```
[jerryc]@bilbo jerryc]408: /usr/local/samba/bin/smbpasswd
Old SMB password: <geben Sie das alte SMB-Passwort ein und drücken Sie Enter>
New SMB password: <geben Sie das neue SMB-Passwort ein und drücken Sie Enter>
Retype new SMB password: <geben Sie das neue SMB-Passwort noch einmal ein und drücken Sie Enter>
... Password changed for user jerryc
```

Windows-9x- und Windows-NT-Clients und verschlüsselte oder Klartextpasswörter

Ich möchte hier einen kurzen Hinweis zur Benutzung von Klartextpasswörtern und neueren Microsoft-Clients geben. Dieses und andere Themen, die für die Microsoft-32-Bit-Clients spezifisch sind, werden ausführlicher in Kapitel 14, »Windows 9x und Windows NT«, dargestellt.

Mit dem Service-Pack-3.0 für Windows NT 4.0 hat Microsoft den Standard geändert, so dass nur noch verschlüsselte Passwörter benutzt werden. Wenn Sie daher versuchen, sich mit einem nicht verschlüsselten Samba-Server zu verbinden, werden Sie folgende Fehlermeldung sehen:

```
Server ist nicht verfügbar. Mit diesem Konto kann mann sich nicht von dieser Station aus anmelden.
```

Sie werden das gleiche Verhalten bei Windows-95-Clients feststellen, die das SMB Network Redirector Update (`vrdrupd.exe`) haben. Windows 95 wird Sie aber einfach weiterhin auffordern, ein Passwort einzugeben. Dieser Patch aktualisiert das System:

```
\windows\system\Vredir.vxd
\windows\system\Vnetsup.xvd
```

Es ist möglich, mit diesen Clients einen nicht verschlüsselten Samba-Server zu benutzen. Dafür müssen Sie nur einen Wert in der Windows-Systemregistrierung einstellen. Die Details für diese Lösung finden Sie in Kapitel 14.

Zugriffskontrollen

Samba bietet außer der Standardauthentifizierung über Benutzername/Passwort einige zusätzliche Optionen für die Kontrolle von Verbindungsanfragen. Über diese Optionen können Sie auf Basis der IP-Adresse des Clients die Verbindungen kontrollieren, was sehr hilfreich sein kann, wenn Ihr Netzwerk mit einem größeren LAN (oder dem Internet) verbunden ist.

hosts allow

Sie können den Parameter `hosts allow` benutzen, um eine Liste von Hosts zu definieren, die sich mit einer bestimmten Freigabe verbinden können. Wird der Parameter im Abschnitt `[global]` der `smb.conf` benutzt, gilt er unabhängig von den einzelnen Freigabeeinstellungen für alle Freigaben.

Der Parameter nimmt als Wert eine Liste von IP-Adressen in Dezimalschreibweise an, wobei die Adressen vollständige Adressen oder Subnetz-Netzwerkadressen sein können. So würde z.B. `192.168.1.73` einem bestimmten Host die Verbindung gestatten, während `192.168.1.` Verbindungen von jedem Host im Class-C-Subnetz `192.168.1.` zulassen würde. Sie können Hostnamen statt IP-Adressen benutzen, wenn Samba die Namen auflösen kann. Dies heißt gewöhnlich, dass Sie als Wert den *Fully Qualified Domain Name (FQDN)* angeben. Es ist ebenfalls möglich, über das Schlüsselwort `EXCEPT` Hosts auszuschließen. Standardmäßig werden Verbindungen von jeder IP-Adresse akzeptiert.

Hier sind einige Beispiele:

```
hosts allow = 192.168.1.73 queso.my.net 191.168. EXCEPT 191.168.2.
```

Diese Einstellung ermöglicht Verbindungen von zwei bestimmten Hosts, `192.168.1.73` und `queso.my.net`, sowie Verbindungen von jedem Host im Class-B-Subnetz `191.168.` außer denen, die sich im Class-C-Subnetz `191.168.2.` befinden.

Hier ist ein Beispiel, für das eine Kombination aus IP-Adresse und Subnetzmaske verwendet wird:

```
hosts allow = 192.168.1.32/255.255.255.224
```

Dies ermöglicht Verbindungen von Hosts im Bereich `192.168.1.33` bis `192.168.1.63`. Die Broadcast-Adresse für das Subnetz ist `192.168.1.64`.

hosts deny

Der Parameter `hosts deny` ist die Ergänzung zum Parameter `hosts allow`. Er bietet die gleiche Funktion wie das Schlüsselwort `EXCEPT` innerhalb des Wertes von `hosts allow`, aber zu einem höheren Grad. Die Syntax ist die gleiche wie die von `hosts allow`. Standardmäßig werden keine Verbindungen abgelehnt:

```
hosts deny = 192.168.3. 192.168.1.72
```

hosts equiv und user hosts

Die nächsten zwei Parameter erwähne ich nur der Vollständigkeit halber und empfehle Ihnen, sie nicht zu benutzen, weil beide Methoden Möglichkeiten bieten, über die sich Benutzer mit Freigaben verbinden können und ohne Passwort authentifiziert werden. Dies kann ein ernsthaftes Sicherheitsloch in Ihrem Server darstellen. Seien Sie vorsichtig!

Über den Parameter `hosts equiv` können Sie den Standort einer Datei festlegen, die eine Liste von Hosts oder Benutzern, einen pro Zeile, enthält, die auf einen Dienst zugreifen können soll, ohne ein Passwort angeben zu müssen. Hier ein Beispiel:

```
hosts equiv = /etc/hosts.equiv
```

Der boolesche Parameter `user hosts` veranlasst Samba, die Unix-Benutzerdatei `~/.rhosts` zu verwenden, um spezielle Hosts zu bestimmen, die ohne Angabe eines Passworts auf Freigaben zugreifen können. Wie beim Parameter `hosts equiv` ist auch diese Option standardmäßig nicht aktiviert. Wollen Sie sie aktivieren, müssen Sie folgende Einträge in den Abschnitt `[global]` der `smb.conf` einfügen:

```
use rhosts = yes
```

Verschiedenes

Die letzen zwei Parameter, die ich darstellen werde, haben einen Bezug zu Sicherheit, passen aber nicht zu den anderen bereits behandelten Themen.

map to guest

Ohne zu sehr ins Detail zu gehen, können Sie über den Parameter `map to guest` festlegen, was Samba tun soll, wenn eine Verbindungsanfrage ungültige Benutzerinformationen enthält (z.B. ein ungültiges Passwort). Es gibt drei mögliche Antworten:

- `Never` - Samba lehnt Verbindungsanfragen mit einem ungültigen Passwort ab. Dies ist die Standardeinstellung.
- `Bad User` - Wenn der Client ein ungültiges Passwort überträgt, wird die Verbindung abgelehnt, es sei denn, der Benutzername ist nicht bekannt. In diesem Fall wird die Verbindung akzeptiert, und der Benutzer wird in den `guest account` aufgenommen, der in `smb.conf` spezifiziert ist.
- `Bad Password` - Diese Einstellung führt dazu, dass Kombinationen aus falschem Benutzernamen und Passwort als Gastverbindungen akzeptiert werden. Der verbindende Benutzer merkt hiervon jedoch nichts und beschwert sich möglicherweise, dass er nicht auf seine Dateien zugreifen kann, weil er als `guest account` verbunden ist.

Ich empfehle Ihnen, die Standardeinstellungen bestehen zu lassen, es sei denn, Sie haben einen guten Grund, sie zu ändern. Wenn Ihnen selbst kein guter Grund einfällt, ist dies wahrscheinlich Grund genug, die Einstellungen in Ruhe zu lassen.

root directory

Dies ist ein weiterer Parameter, der nicht häufig benutzt wird. Er weist Samba an, ein `chroot()` zum angegebenen Verzeichnis auszuführen, ganz ähnlich wie es bei anonymen FTP-Verbindungen gehandhabt wird. Dies ist nicht unbedingt notwendig, da Samba standardmäßig Zugriff auf Dateien außerhalb der Freigabe ablehnt. Allerdings wird hiermit eine zusätzliche Sicherheitsebene eingefügt, aber sie müssen sicherstellen, dass sich alle notwendigen Skripte, Systemdateien und Binaries unterhalb des Root-Verzeichnisses befinden. Um das Standard-Root-Verzeichnis `/` zu überschreiben, können Sie ganz einfach das Verzeichnis Ihrer Wahl spezifizieren:

```
root directory = /export/smb
```

Abschließende Kommentare

Ich dachte, es wäre besser, mit diesem Punkt zu schließen, statt ihn innerhalb eines Kapitels zu vergraben. Wenn Sie durch eine Firewall von anderen Netzwerken getrennt sind und nicht wollen, dass Clients, die sich hinter der Firewall befinden, auf Ihre internen SMB-Server zugreifen können, sollten Sie die eingehenden Ports 137, 138 und 139 blockieren. Dies ist insbesondere dann wichtig, wenn Sie Benutzer haben, die gern ihre gesamte Festplatte freigeben, weil »es so praktisch ist«. Gerade Windows 95/98-Clients bieten dies per Voreinstellung an.

In Kapitel 7 kommen Sie zu den Details der Freigabenkonfiguration, so dass Ihre Benutzer tatsächlich auf ihre Dateien zugreifen können. Oh, ich habe vergessen, die kleine Geschichte zu beenden, die ich am Anfang dieses Kapitels begonnen habe, oder?

Nachdem ich etwas weniger als eine Stunde gebraucht hatte, trank ich den letzten Schluck von meinem Kaffee (der mittlerweile lauwarm geworden war) und machte mich auf den Weg zu meiner Chefin. Nachdem ich ihr meine Entscheidung für verschlüsselte Passwörter erklärt und einen Strategieplan - noch ein Dilbert-Wort - für die Änderung existierender Benutzerpasswörter in verschlüsselte Passwörter auf relativ harmlose Art und Weise definiert hatte, gratulierte sie mir zu einer weiteren gut erledigten Aufgabe. Dann unterschrieb sie einen Kaufvertrag für einen neuen Laptop, damit ich während meines von der Firma bezahlten Urlaubs mit ihr in Kontakt bleiben konnte. (Es könnte passieren!)

Zusammenfassung

Das SMB-Protokoll unterstützt zwei Modi für die Authentifizierung von Verbindungen. Samba unterstützt sowohl den Share-Modus (Freigabeebene) als auch den User-Modus (Benutzerebene). Zusätzlich bietet Samba zwei weitere Varianten der Authentifizierung auf Benutzerebene: Server-Modus und Domain-Modus.

Sie können für alle `security`-Optionen in Samba entweder Klartext- oder verschlüsselte Passwörter benutzen. Klartextpasswörter werden über die Standard-Unix-Account-Datenbank `/etc/passwd` (oder ihre Netzwerkentsprechung) authentifiziert. Die Passwortverschlüsselung dagegen verlangt, dass Samba eine separate Datei verwaltet, in der die Hashwerte der verschlüsselten Passwörter gespeichert werden.

Frage & Antwort

F. Brauche ich externe Libraries, um die Passwortverschlüsselung in Samba zu aktivieren?

. Es ist richtig, dass der Administrator für ältere Versionen von Samba eine externe DES-Library besorgen musste, aber neuere Versionen von Samba benötigen diese nicht mehr. Der komplette benötigte Source-Code ist in der Samba-Distribution enthalten.

F. Kann ein Samba-Server so konfiguriert werden, dass er sowohl Klartext- als auch verschlüsselte Passwörter gleichzeitig unterstützen kann?

. Nein. Ein einzelner Samba-Server kann nicht gleichzeitig Klartext- und verschlüsselte Passwörter für die Authentifizierung von Benutzern verwenden. Es gibt jedoch eine Methode über den Parameter `netbios aliases`, mit der Sie dies umgehen können. Die nötigen Funktionen für die Implementierung einer derartigen Lösung werden in Kapitel 10, »Automatisierung auf Server-Seite«, dargestellt.

F. Können einige Freigaben für den Share-Modus und andere auf dem gleichen Server für den User-Modus konfiguriert werden?

. Nein. Der Samba-Parameter `security` ist eine `[global]`-Option.

Neue Begriffe

Klartextentsprechung eines Passworts - Diese wird generiert, wenn der benutzte Verschlüsselungsalgorithmus bei gleicher Eingabe immer den gleichen Byte-String erzeugt. Anders gesagt, ein Passwort wird immer zum gleichen Wert verschlüsselt. Kann ein Eindringling die verschlüsselte Version des Passworts abfangen, kann er erfolgreich am Challenge/Response-Authentifizierungsschema teilnehmen, das von SMB-Servern wie Samba und Windows NT benutzt wird.

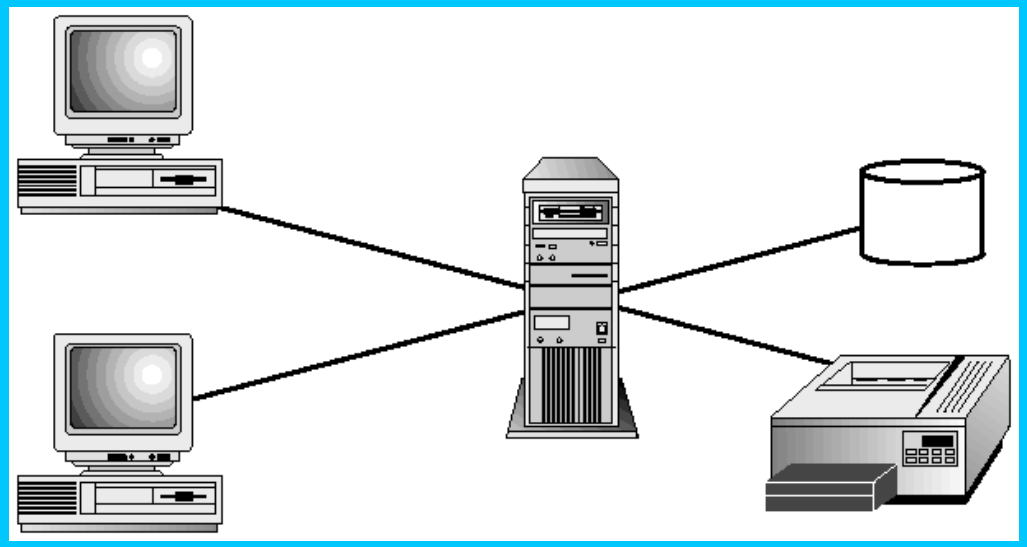
Tag 7: Dateifreigaben

von Richard Sharpe

In den Kapiteln 5, »Die Datei smb.conf: Samba mitteilen, was es tun soll«, und 6, »Sicherheitsmodi und Passwörter«, haben Sie sich das grundlegende Format für die Parameter in `smb.conf`, die Samba-Sicherheitsmodi und die Handhabung der Passwörter angesehen. Samba wurde entwickelt, um Dateien auf verschiedenen Rechnern gemeinsam nutzen zu können, und genau das ist es, was wir uns jetzt näher ansehen werden.

Die Freigabe von Dateien ermöglicht Ihnen, Dateien von verschiedenen Rechnern aus gemeinsam zu benutzen (siehe Abbildung 7.1). Normalerweise verfügt ein Datei-Server über mehr Festplattenspeicher als ein Client (Windows für Workgroups, Windows 95/98, Windows NT usw.). An den Server können auch die meisten Drucker angeschlossen sein, aber das Drucken wird im nächsten Kapitel ausführlich beschrieben.

Abb. 7.1: Ein Datei-Server, der Dateien und Drucker freigibt.



In diesem Kapitel führe ich Sie durch alle notwendigen Schritte, damit Sie Dateifreigaben auf einem Samba-Server einrichten können. Sie werden außerdem die meisten Freigabe- und globalen Parameter kennenlernen, die für die Freigabe von und den Zugriff auf Dateien relevant sind.

Wenn Sie meinen Ausführungen Schritt für Schritt folgen, sollten Sie sicherstellen, dass Ihre Clients keine verschlüsselten Passwörter benutzen, da diese zu unnötigen Komplikationen führen können. Weitere Informationen zur Deaktivierung verschlüsselter Passwörter auf Ihren Clients finden Sie in den Kapiteln 6 und 14, »Windows 9x und Windows NT«. Sie sollten sich über den Account `boss` in Ihren Client-Rechner einloggen. Ist Ihr Samba-Server der erste SMB-Server in Ihrem Netzwerk, erhalten Sie möglicherweise während des Einloggens eine Fehlermeldung, die besagt, dass Sie nicht über einen Logon-Server authentifiziert werden konnten. Ignorieren Sie diese Meldung fürs Erste.

Eine smb.conf-Datei aufbauen

Bevor Sie Dateien freigeben können, brauchen Sie eine funktionierende `smb.conf`-Datei, die Samba benutzen kann. In Kapitel 4, »Installation und Testen der Konfiguration», haben Sie sich eine `smb.conf`-Datei angesehen, hier werden Sie eine nun von Beginn an aufbauen.

Wie ich bereits erwähnt habe, hat die Datei `smb.conf` einen globalen und einen Freigabeabschnitt. Im Folgenden benutzen Sie den untenstehenden globalen Abschnitt und fügen Abschnitte für Dateifreigaben hinzu, während Sie verschiedene Methoden für die Kontrolle und Verwaltung von Dateifreigaben kennenlernen:

```
[global]
   workgroup = FOWLPLAY
   netbios name = EAGLE
```

```
   server string = My first server
   guest account = pcguest
   security = user
   password level = 8
```

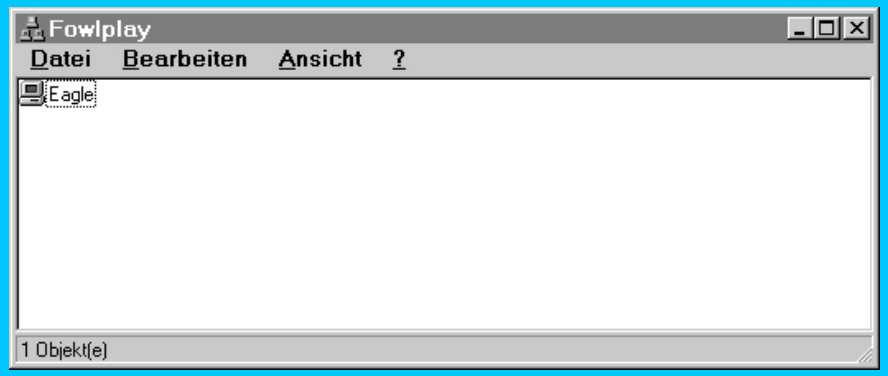
Als Erstes sollten Sie bei dieser `smb.conf`-Datei bemerken, dass sie keine Freigaben definiert, aber trotzdem funktioniert. Wie auch schon in Kapitel 5 ist in dieser `smb.conf`-Datei die Arbeitsgruppe, der Samba angehört, `FOWLPLAY` und der NetBIOS-Name des Servers, `EAGLE`.

Wenn Sie diese `smb.conf` auf Ihrem Server installieren und Samba neu starten (Sie müssen sich als `root` in Ihren Samba-Server einloggen, um das zu tun), sollten Sie den neuen Server in der Netzwerkumgebung in Windows 9x oder Windows NT 4.0 (für Windows für Workgroups 3.11 benutzen Sie den Dateimanager und wählen *Laufwerke*, *Netzlaufwerk verbinden*) sehen können.

Wenn Sie Samba bereits benutzen, ersetzen Sie Ihre vorhandene `smb.conf`-Datei nicht einfach durch obiges Beispiel. Machen Sie zuerst eine Backup-Kopie.

Nachdem Sie Samba neu gestartet haben (siehe Kapitel 4 für Details darüber, wie Sie Samba auf verschiedenen Plattformen starten), sollten Sie unter Windows 9x oder Windows NT die Netzwerkumgebung sehen, wie sie in Abbildung 7.2 dargestellt ist.

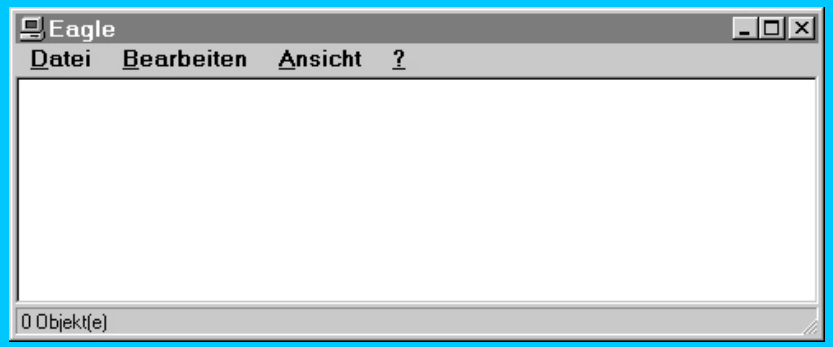
Abb. 7.2: Netzwerkumgebung für die Arbeitsgruppe FOWLPLAY



Hier sehen Sie, dass Ihr Server als `EAGLE` auftaucht (so haben Sie ihn in `smb.conf` genannt) und dass es sich um einen Samba-2.0.0Beta4-Server handelt. Wenn Sie den Parameter `netbios name` aus der vorstehenden `smb.conf` entfernen, wird Ihr Server mit einem Namen angezeigt, der aus den ersten Bestandteilen seines DNS-Namens besteht. Dies ist möglicherweise das, was Sie wollen, aber es wird nicht zu den Beispielen in diesem Kapitel passen.

Wenn Sie jetzt auf den oben stehenden Server doppelklicken, wird das Fenster in Abbildung 7.3 geöffnet.

Abb. 7.3: Auflistung der Freigaben auf dem Dateiserver EAGLE



Damit wird bestätigt, dass Ihr Samba-Server keine Freigaben hat oder zumindest keine, die browsebar sind (dies werde ich später darstellen).

Ihr nächster Schritt besteht darin, eine Freigabe einzurichten und zu sehen, was sich ändert.

Eine Freigabe einrichten

Um eine Freigabe einzurichten, müssen Sie Ihrer `smb.conf`-Datei im Definitionsbereich für Freigaben einen entsprechenden Abschnitt hinzufügen. Da Samba Verzeichnisse und die darin liegenden Dateien freigibt, sollten Sie zunächst ein entsprechendes Verzeichnis, das freigegeben werden soll, auf Ihrem Server suchen oder erstellen.

Hier erstelle ich ein Verzeichnis mit dem Namen `/home/first-share` und weise Samba an, es freizugeben:

```
mkdir /home/first-share
```

Sie können natürlich auch einen anderen Namen benutzen, wenn Sie zu den abenteuerlustigen Menschen gehören, aber dann müssen Sie im Folgenden alle Pfadnamen entsprechend ändern, und die Beispiele in diesem Buch sehen möglicherweise anders aus als das, was Sie sehen.

Dann fügen Sie unserer oben stehenden `smb.conf`-Datei Folgendes hinzu:

```
[first-share]
   comment = Meine erste Freigabe
   path = /home/first-share
   browsable = yes
```

Nachdem Sie nun Ihrer `smb.conf` eine Freigabe hinzugefügt haben, starten Sie Samba neu. Sie sollten (nach einer Weile) die neue Freigabe in der Netzwerkumgebung sehen können, wie in Abbildung 7.4 dargestellt.

Abb. 7.4: Die Netzwerkumgebung zeigt Ihre erste Freigabe!

Großartig, nun können Sie eine Freigabe auf Ihrem Samba-Server sehen. Was können Sie noch tun? Nun, bevor Sie sich die Freigabe ansehen, sollten Sie erst einmal einige Dateien einfügen:

```
cat > /home/first-share/file-1.txt
Now is the time for all good men
To come to the aid of their country
^D
cat > /home/first-share/file-2.txt
The time has come the walrus said
To talk of many things
^D
todos /home/first-share/file-2.txt
```

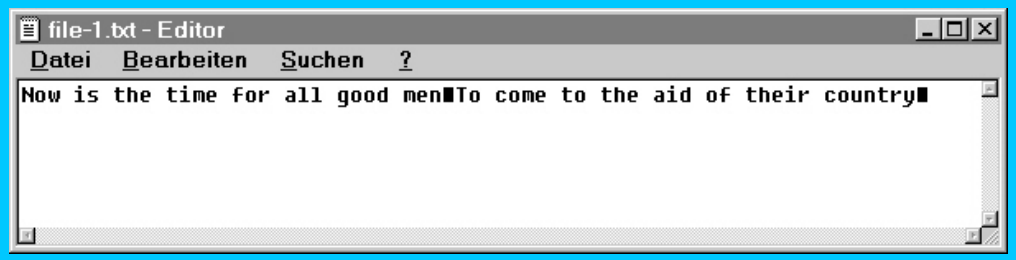
Beachten Sie, dass Sie möglicherweise das `todos-Utility` nicht auf Ihrem Server haben. In diesem Fall sollten Sie sich eine Kopie besorgen oder das Perl-Skript (das sich auf der diesem Buch beiliegenden CD-ROM befindet) eingeben, da Sie es brauchen werden. Das Skript ist bei einigen Distributionen auch unter dem Namen `u2dos` oder ähnlich verfügbar.

Nachdem Sie einige Dateien in die Freigabe auf Ihrem Server gestellt haben, schauen Sie sich an, was Ihr Client Ihnen zeigt. Doppelklicken Sie auf `first-share` im Fenster *Netzwerkumgebung*, und Sie sehen Ihre Dateien, wie in Abbildung 7.5 dargestellt.

Abb. 7.5: Die ersten zwei Dateien in Ihrer Dateifreigabe

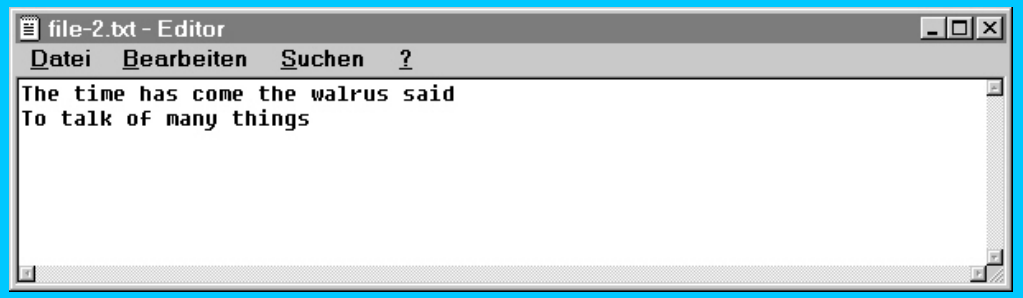
Sehen Sie sich nun die Dateien an. Wenn Sie auf `file-1-txt` doppelklicken, sollten Sie das in Abbildung 7.6 dargestellte Fenster sehen.

Abb. 7.6: file-1.txt enthält seltsame Zeichen



Schließen Sie dieses Fenster und doppelklicken Sie auf `file-2.txt`. Sie sollten genau das sehen, was in Abbildung 7.7 dargestellt ist.

Abb. 7.7: file-2.txt sieht normaler aus



Warum werden diese zwei Dateien so unterschiedlich im Editor dargestellt? Das Problem liegt in der unterschiedlichen Art und Weise begründet, in der Windows und Unix das Ende einer Zeile in Textdateien speichern. Unter Unix wird das Zeilenende durch das Zeilenvorschubzeichen (NL) gekennzeichnet (oktal 012, hex 0x0A), während unter Windows (und DOS) das Zeilenende durch ein Wagenrücklaufzeichen oder CR (oktal 015, hex 0x0C) gefolgt von einem Zeilenvorschubzeichen gekennzeichnet ist. Als Sie `file-2.txt` erstellten, haben Sie die Datei explizit in eine Textdatei im DOS-Format konvertiert (über `todos`), während `file-2.txt` als Textdatei im Unix-Format belassen wurde.

Obwohl der Zweck jeder Zeile, die Sie in der `smb.conf` eingefügt haben, offensichtlich sein mag, möchte ich dennoch jede einzeln darstellen und erklären.

[first-share]

Diese Zeile führt einen neuen Abschnitt für die Freigabe `first-share` ein. Jede Freigabe wird als neuer Abschnitt eingeführt, mit ihrem Namen in eckigen Klammern.

comment

Diese Zeile bietet einen beschreibenden Kommentar für die Freigabe. Sie dient der Dokumentation der Freigabe in `smb.conf` und wird unter dem Begriff Kommentar in der Auflistung in der Netzwerkumgebung angezeigt.

path

Die Zeile teilt Samba mit, welchen Teil des Dateisystems es für Clients freigeben soll. Sie sollten immer ein vorhandenes Verzeichnis angeben oder eine Datei, die als symbolischer Link zu einem Verzeichnis dient (siehe nachfolgend).

Existiert das Verzeichnis nicht, können Windows-Clients seltsame Fehlermeldungen ausgeben, wenn sie versuchen, auf die Freigabe zuzugreifen. Ein Beispiel:

```
Zugriff auf \\EAGLE\first-share nicht möglich: Der Netzwerkname kann entweder auf dem laufenden Netzwerk nicht gefunden werden oder ist fehlerhaft.
```

browsable

Streng genommen brauchen Sie diesen Eintrag nicht zu definieren, da die Standardeinstellung `yes` ist. Sie müssen die Einstellung nur dann ändern, wenn Sie nicht wollen, dass Clients bestimmte Freigaben sehen können.

Um die Auswirkung der Einstellung für `browsable` zu sehen, ändern Sie sie in Ihrer `smb.conf` auf

```
browsable = no
```

und starten Samba neu. Gehen Sie dann zurück zur `FOWLPLAY`-Netzwerkumgebung und doppelklicken Sie auf den Server `EAGLE`. Können Sie nun die von Ihnen erzeugten Freigaben sehen, wenn sich das Fenster öffnet?

Zugriffsrechte

Sie haben nun eine Dateifreigabe eingerichtet und müssen Sie nutzbar machen. Browsen Sie z.B. die Dateifreigabe (nachdem Sie die Einstellung `browsable` in Ihrer `smb.conf` wieder aktiviert, Samba beendet und neu gestartet haben). Versuchen Sie dann, eine Datei in die Freigabe zu kopieren oder eine neue Datei bzw. ein neues Verzeichnis in der Freigabe zu erstellen. Wie Sie feststellen werden, ist dies nicht möglich. Sie werden nun untersuchen, warum das so ist und wie Sie dies ändern können.

Die Dateifreigabe eignet sich hervorragend für die gemeinsame Benutzung von Dateien, die von Benutzern mit ausreichenden Berechtigungen für das freigegebene Verzeichnis gestellt wurden, aber Client-Workstations können keine Dateien in der Freigabe erzeugen. Auch wenn Sie sich als `root` einloggen (der Superuser unter Unix), können Sie an diesem Punkt keine Dateien in der Freigabe erzeugen. Wie Sie bald sehen werden, ist die Freigabe standardmäßig nur mit Leseberechtigungen versehen.

Damit Clients auf die Dateifreigabe zugreifen können, müssen Sie Samba einige Dinge über die Dateifreigabe mitteilen. Zunächst sollten Sie jedoch einen Blick darauf werfen, wie Samba bestimmt, ob ein Client Zugriff auf Dateifreigaben erhält.

In Kapitel 2, »Windows-Netzwerke«, haben Sie sich die Art und Weise angesehen, in der ein CIFS/SMB-Client auf eine Ressource oder eine Dateifreigabe zugreift. Um Ihr Gedächtnis aufzufrischen, sind hier noch einmal die erforderlichen Schritte, obwohl einige davon ausgelassen werden können, aufgelistet:

1. Der Client verhandelt über den Protokolldialekt, um festzulegen, welche Variante des CIFS/SMB-Protokolls er unterstützt.
2. Der Client führt dann möglicherweise ein Login im Netzwerk oder auf dem Server durch, abhängig vom benutzten Protokolldialekt und der Präsenz von Logon-Servern. An diesem Punkt übermittelt der Client einen Benutzernamen und ein Passwort, aber dieser Schritt kann übergangen werden, insbesondere bei älteren Clients.
3. Der Client verlangt eine Verbindung zu einem Verzeichnisbaum oder einer Freigabe.

Um die Verbindungsanfrage des Clients zu einer Freigabe weiterzuverarbeiten, bestimmt Samba zunächst, ob die verlangte Freigabe existiert. Der folgende einfache Ansatz überprüft, ob dies der Fall ist:

1. Die Datei `smb.conf` wird nach einem Abschnitt durchsucht, der mit dem verlangten Freigabenamen übereinstimmt. Wird einer gefunden, wird dieser benutzt.
2. Wird die Freigabe nicht gefunden, überprüft Samba, ob sie einen `[homes]`-Abschnitt in der Datei `smb.conf` hat. Falls ja, wird die Datei `passwd` durchsucht, um festzustellen, ob der Freigabename einem Benutzernamen entspricht. Ist das so, wird ein Klon der `[homes]`-Freigabe erzeugt (wie später in diesem Kapitel ausführlich dargestellt) und die neue Freigabe verwendet.
3. Wird die Freigabe immer noch nicht gefunden, überprüft Samba, ob die `smb.conf` einen `[printer]`-Abschnitt enthält. Falls ja, wird festgestellt, ob die verlangte Freigabe einem Drucker in der Datei `printcap` entspricht. Ist das der Fall, wird ein Klon der `[printers]`-Freigabe erzeugt und diese geklonte Freigabe benutzt. Der Abschnitt `[printers]` wird ausführlich in Kapitel 8, »Drucker«, dargestellt.
4. Wird die Freigabe auch hier nicht gefunden, überprüft Samba, ob eine Standardfreigabe existiert, ändert in diesem Fall den Namen der Standardfreigabe in den entsprechenden verlangten Freigabenamen und benutzt diese Freigabe.
5. Wird keine Freigabe gefunden, gibt Samba eine Fehlermeldung über einen ungültigen Netzwerknamen an den Client zurück.

Danach durchläuft Samba die folgende Prozedur, um festzustellen, ob ein Client Zugriff auf die gefundene Freigabe erhält und als welcher Benutzer dieser Client zugreifen kann. Die Schritte werden nacheinander

ausgeführt, und der Prozess wird mit dem ersten erfolgreichen Schritt beendet. Ist keiner der Schritte erfolgreich, wird der Zugriff zur Freigabe verweigert.

1. Wenn der Client einen Benutzernamen und ein Passwort überträgt, die authentifiziert werden können, wird der Zugriff zur Freigabe als authentifizierter Benutzer gewährt. Einige ältere Clients können ihre Benutzernamen über die Syntax `\\Server\Service%Benutzername` übertragen.
2. Hat der Client bereits einen gültigen Benutzernamen übermittelt und sendet nun ein korrektes Passwort (für die Freigabeanfrage), wird der Zugriff auf die Freigabe gewährt.
3. Der NetBIOS-Name des Clients und alle vorher verwendeten Benutzernamen werden über die Standardmechanismen des Betriebssystems (oder die Datei `smbpasswd`) mit dem übertragenen Passwort authentifiziert. Kann ein Benutzername erfolgreich authentifiziert werden, wird der Zugriff auf die Freigabe als authentifizierter Benutzer gewährt.
4. Wurden bereits vorher ein Benutzername und ein Passwort durch den Server (über ein `SessionSetupandX`) authentifiziert und hat der Client den Authentifizierungs-Token in der Freigabeanfrage übertragen, wird der Zugriff auf die Freigabe als Benutzername gewährt, der in dem Token definiert ist. Dieser Schritt wird übergangen, wenn für die Freigabe Revalidation spezifiziert ist (`revalidate = yes`).
5. Wenn für die Freigabe ein Feld `user =` definiert ist, der Client ein Passwort übertragen hat und die Kombination aus Benutzername, der für die Freigabe definiert wurde, und Passwort authentifiziert wurde, wird der Zugriff auf die Freigabe als definierter Benutzer gewährt.

Handelt es sich bei der Freigabe jedoch um eine `Guest-only`-Freigabe, wird der Zugang zur Freigabe als im Gast-Account definierter Benutzername gewährt, ohne dass einer der vorstehenden Schritte durchlaufen wird. Jedes übertragene Passwort wird ignoriert.

Wenn der als zugreifender Benutzer gewählte Benutzer sich in einer ungültigen Benutzerliste befindet (die später in diesem Kapitel beschrieben wird), wird die Verbindungsanfrage an diesem Punkt abgewiesen.

Über diese Prozedur kann Samba feststellen, unter welchem Account der Zugriff auf Dateien in der Freigabe gewährt wird. Der Zugriff auf Freigaben und die erlaubten Zugriffsberechtigungen auf Dateien werden jedoch noch von mehreren Parametern in der Datei `smb.conf` kontrolliert.

Parameter für den Zugriff auf Freigaben

Die folgenden Parameter sind auf die eine oder andere Weise relevant für den Zugriff auf Freigaben durch Clients. Die meisten Samba-Administratoren benutzen nicht viele dieser Parameter. Wie immer finden Sie die aktuelle Liste aller Parameter und die endgültige Darstellung ihrer Funktionen in den Manpages zu `smb.conf` für die aktuelle Version von Samba. Dort können Sie sich über den Befehl `man smb.conf` die Parameter genauer ansehen.

admin users

Dieser Freigabe-Parameter richtet die Benutzer ein, denen administrative Privilegien für die Freigabe gegeben werden. Wenn sie auf die Freigabe zugreifen, führen sie alle Dateioperationen als `root` aus.

Namen, die mit einem `@` beginnen, werden zunächst als NIS-Netzgruppe interpretiert und dann, wenn sie nicht in NIS gefunden werden, als Unix-Gruppen. Namen, die mit einem `+` beginnen, werden als Unix-Gruppen und Namen, die mit `&` beginnen, als NIS-Gruppen interpretiert.

Dieser Parameter kann sehr gefährlich sein, da jeder Benutzer in der `admin-users`-Liste alles tun kann, was er will, z.B. auch alle Dateien in der Freigabe löschen.

Standardmäßig gibt es keine administrativen Benutzer für eine Freigabe. Ein Beispiel:

```
admin users = root, fred
```

Diese Einstellung definiert `root` und `fred` als administrative Benutzer.

default service

Über diesen globalen Parameter wird der Name der Standardfreigabe festgelegt. Diese Standardfreigabe wird benutzt, wenn die von einem Client verlangte Freigabe nicht gefunden werden kann, und ihr Name wird in diesem Fall in den entsprechenden verlangten Namen umgeändert.

Normalerweise werden für die Standardfreigabe die Parameter `guest ok = yes` und `read only` eingerichtet.

Dieser Parameter hat keinen Standardwert. Ein Beispiel:

```
default service = lastchance
```

Mit dieser Einstellung wird `lastchance` zur Standardfreigabe.

guest account

Über diesen globalen Parameter wird der Name des Gast-Accounts definiert. Oft wird er auf `pcguest` eingestellt, und er muss in der auf dem Server benutzten Account-Datenbank eingetragen sein (z.B. der Datei `/etc/passwd`, NIS usw.). Normalerweise hat dieser Account kein gültiges Passwort, so dass sich niemand dort einloggen kann, weder von Unix noch von einem Client. Der Account kann nur benutzt werden, um den Zugriff auf Dateien zu kontrollieren.

Dieser Parameter kann im globalen Abschnitt und in einzelnen Freigabeabschnitten eingerichtet werden. Gast-Accounts, die in einem Freigabeabschnitt definiert sind, setzen globale Gast-Accounts außer Kraft.

Der Standardwert für diesen Parameter wird während der Kompilierung auf `nobody` gestellt. Ein Beispiel:

```
guest account = pcguest
```

Damit wird der Account mit dem Namen `pcguest` als Gast-Account definiert.

guest ok

Dieser Freigabe-Parameter bestimmt, ob der Zugriff zu einer Freigabe auch ohne Übertragung eines Benutzernamens und eines Passworts gewährt wird.

Wenn Clients einen Gastzugriff erhalten, greifen Sie auf die Dateien in der Freigabe als Gast-Account zu.

Ein Synonym für `guest ok` ist `public`.

Der Standardwert für diesen Parameter ist `no`. Ein Beispiel:

```
guest ok = yes
```

Damit kann über den Guest-Account auf die Freigabe zugegriffen werden.

guest only

Dieser Freigabe-Parameter bestimmt, dass nur Gastverbindungen zu einer bestimmten Freigabe erlaubt sind. Er muss in Verbindung mit `guest ok` oder `public` benutzt werden.

Der Standardwert für diesen Parameter ist `no`. Ein Beispiel:

```
guest only = yes
```

Damit wird festgelegt, dass auf diese Freigabe nur über den Guest-Account zugegriffen werden kann.

hosts allow

Dieser Parameter richtet die Liste der Hosts ein, die auf Freigaben zugreifen können. Wird er im Abschnitt `[global]` eingerichtet, bezieht er sich auf alle Freigaben, unabhängig von den Einstellungen für einzelne Freigaben. Wird der Parameter nicht im globalen Abschnitt verwendet, kann er für einzelne Freigaben definiert werden.

Hosts können über ihren Namen oder ihre IP-Adresse spezifiziert werden. Die vollständige Syntax für diesen Parameter ist die gleiche wie für die TCP-Wrappers-Datei `hosts_allow`. Weitere Details finden Sie in der Manpage (`man hosts_allow`).

Es gibt keinen Standardwert für diesen Parameter. Ein Beispiel:

```
hosts allow = 192.1.1. graham.goodies.com
```

Mit dieser Einstellung kann jeder Host im Subnetz `192.1.1.0/24` und das System mit dem Namen `graham.goodies.com` auf die Freigabe zugreifen.

hosts deny

Dieser Parameter ist die Umkehrung von `hosts allow`. Die für diesen Parameter aufgelisteten Hosts erhalten keinen Zugriff auf Freigaben, es sei denn, eine bestimmte Freigabe setzt die Liste mit einer eigenen Liste zugelassener Hosts außer Kraft. Gibt es Unstimmigkeiten zwischen den Parametern `hosts deny` und `hosts allow`, erhält `hosts allow` die Priorität.

Es gibt keinen Standardwert für diesen Parameter. Ein Beispiel:

```
hosts deny = 192.1.1. badhost.bad-company.com
```

Diese Einstellung verweigert jedem Host im Subnetz `192.1.1.0/24` und dem Host `badhost.bad-company.com` den Zugriff auf die Freigabe.

invalid users

Dieser Freigabe-Parameter spezifiziert eine Liste von Benutzern, die keinen Zugriff auf die Freigabe erhalten sollten. Der Parameter benutzt die gleiche Syntax wie der oben stehende Parameter `admin users`.

Es gibt keinen Standardwert für diesen Parameter. Ein Beispiel:

```
invalid users = root fred @bin
```

Mit dieser Einstellung wird den Benutzern `root` und `fred` sowie jedem Benutzer in der Gruppe `bin` der Zugriff auf die Freigabe verweigert.

max connections

Mit diesem Freigabe-Parameter wird die maximale Anzahl der Clients festgelegt, die sich mit der Freigabe verbinden können. Ist der hier spezifizierte Wert größer als 0, werden nach Erreichen der angegebenen Anzahl von Benutzern, die sich mit der Freigabe verbunden haben, keine Clients mehr zugelassen. Fällt die Anzahl der verbundenen Benutzer unter die angegebene Zahl, können sich wieder neue Benutzer bis zum definierten Maximum verbinden. Wird der Wert des Parameters auf `0` gesetzt, gibt es keine Einschränkung für die Anzahl der zugelassenen Verbindungen.

Dieser Parameter kann die Arbeitslast auf einem Samba-Server beschränken und bietet außerdem eine Methode für die Durchsetzung der Lizenzierungsbestimmungen, wenn Sie lizenzierte Software freigeben. Beachten

Sie, dass Sie die Anzahl der verbundenen Benutzer einschränken, nicht die Anzahl der aktiven Benutzer. Ein Benutzer, der sich zu Beginn des Tages mit der Freigabe verbindet und den ganzen Tag verbunden bleibt, ohne jemals etwas damit zu tun, belegt auch eine dieser maximalen Verbindungen.

Der Standardwert für diesen Parameter ist `0`, der, wie vorher schon erwähnt, uneingeschränkte Verbindungen zur Freigabe ermöglicht. Ein Beispiel:

```
max connections = 100
```

Mit dieser Einstellung werden nicht mehr als 100 Verbindungen zur Freigabe zugelassen.

read list

Dieser Freigabe-Parameter definiert eine Liste von Benutzern, denen Nur-Lese-Zugriff auf die Freigabe gewährt wird. Das heißt, diese Benutzer erhalten keinen Schreibzugriff auf die Freigabe, auch wenn die Freigabe beschrieben werden kann.

Es gibt keinen Standardwert für diesen Parameter. Ein Beispiel:

```
read list = fred @guests
```

Diese Einstellung definiert, dass `fred` und alle Benutzer in der Gruppe `guests` Nur-Lese-Zugriff auf die Freigabe erhalten. Das heißt, sie können die Freigabe nicht beschreiben.

read only

Dieser Freigabe-Parameter ist `das` Gegenteil des Parameters `writable`. Wird er aktiviert, können Clients keine Freigabe beschreiben.

Der Standardwert für diesen Parameter ist `yes`, d.h., die Freigabe ist im Nur-Lese-Modus. Wenn Sie für eine Freigabe hier keinen Wert definieren, kann sie nur gelesen werden. Ein Beispiel:

```
read only = no
```

Mit dieser Einstellung kann die Freigabe beschrieben werden. Den gleichen Effekt können Sie auch über den Parameter `writable = yes` erzielen.

valid users

Dieser Freigabe-Parameter listet die Benutzer auf, denen der Zugriff auf die Dateifreigabe gewährt wird. Er benutzt die gleiche Syntax wie der Parameter `admin users`.

Standardmäßig ist dieser Parameter leer, was bedeutet, dass jeder Benutzer auf die Freigabe zugreifen kann. Ein Beispiel:

```
valid users = fred @accounts
```

Mit dieser Einstellung sind der Benutzer `fred` und alle Benutzer in der Gruppe `accounts` die einzigen, die auf die Freigabe zugreifen können.

writable

Dieser Freigabe-Parameter (und sein Synonym `writeable`, für die, die Probleme mit der Rechtschreibung haben) zeigt an, ob Clients die Freigabe beschreiben können. Siehe auch `read only`.

Der Standardwert für diesen Parameter ist `no`, d.h. standardmäßig ist eine Freigabe also nur lesbar. Beispiele für die Benutzung des Parameters sind:

```
writable = yes
writeable = yes
read only = no
```

Alle diese Einstellungen führen dazu, dass jeder die Freigabe beschreiben kann, der die Berechtigungen hat, Dateien und Verzeichnisse in der Freigabe zu beschreiben.

write list

Dieser Freigabe-Parameter definiert eine Liste von Benutzern, die Schreib-/Lese-Zugriff auf eine Freigabe haben, unabhängig vom Wert des Parameters `read only`.

Ist ein Benutzer sowohl in der `read-list` als auch in der `write-list`, wird ihm Schreibzugriff gewährt.

Standardmäßig hat dieser Parameter keinen Wert, was bedeutet, dass für alle Benutzer der Wert des Parameters `read only` gilt. Ein Beispiel:

```
write list = root @admin
```

Diese Einstellung definiert, dass zumindest der Benutzer `root` und alle Benutzer in der Gruppe `admin` Schreibzugriff auf die Freigabe haben.

Zugriffsrechte für Ihre erste Freigabe first-share einrichten

Nachdem Sie sich die für den Zugriff auf Freigaben relevanten Parameter angesehen haben, können Sie damit beginnen, die Probleme zu korrigieren, auf die Sie vorher in puncto Zugriff gestoßen sind, als Sie nicht in Ihre Freigabe `first-share` schreiben konnten.

Die Probleme liegen darin begründet, dass für eine Freigabe standardmäßig folgender Parameter eingerichtet ist:

```
writable = no
```

Um sicherzustellen, dass weder Datei- noch Verzeichnisberechtigungen zu den Problemen führten, schauen Sie sich die Unix-Berechtigungen für das freigegebene Verzeichnis an und ändern Sie sie für jedermann zugänglich in RWX (0777 der Einfachheit halber) um. Wenn Sie Ihre `umask` nicht geändert haben, sieht `/home/first-share` auf Ihrem Samba-Server wie folgt aus:

```
ls -al /home/first-share
total 4
drwxr-xr-x    2  root  root    1024 Jan 5 14:23
drwxr-xr-x    17 root  root    1024 Jan 5 14:23
-rw-r--r--    1  root  root      69 Jan 5 14:22 file-1.txt
-rw-r--r--    1  root  root      59 Jan 5 14:23 file-2.txt
```

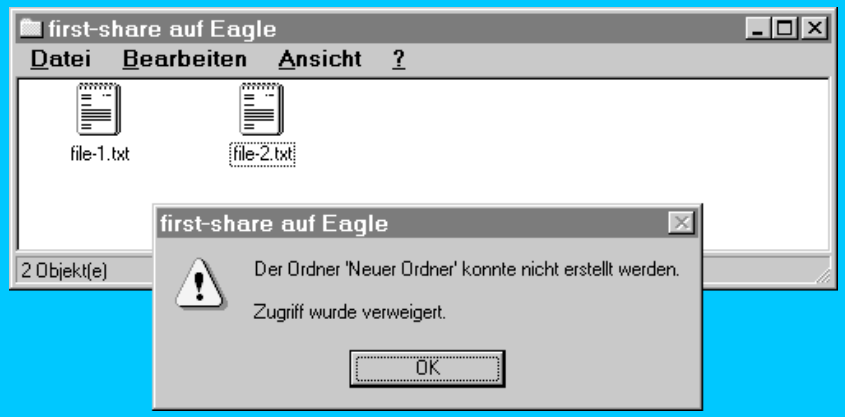
Ändern Sie die Berechtigungen für das Verzeichnis nun folgendermaßen auf 0777 um:

```
chmod 0777 /home/first-share
```

Damit sollte jeder Unix-Benutzer in das Verzeichnis schreiben können, da hiermit Benutzer-, Gruppen- und allgemeines Lese-, Schreib- und Ausführungsrecht eingerichtet wurde.

Wenn Sie aber nun versuchen, von Ihrem Client eine Datei in der Freigabe zu erstellen, erhalten Sie das, was Sie in Abbildung 7.8 sehen.

Abb. 7.8: Der Zugriff wird verweigert, obwohl das Verzeichnis schreib- und lesbar ist



Fügen Sie Ihrer Freigabe `first-share` Folgendes hinzu (und starten Sie Samba neu), damit Clients in die Freigabe schreiben können:

```
writable = yes
```

Versuchen Sie es! Sie können jetzt Verzeichnisse und Dateien in der Freigabe erstellen, wie Sie in Abbildung 7.9 sehen.

Abb. 7.9: Kann die Freigabe beschrieben werden, können Sie darin Ordner erstellen

Auf dem Samba-Server sieht Ihr freigegebenes Verzeichnis nun so aus:

```
ls -al /home/first-share
total 4
drwxrwxrwx    2  root  root    1024 Jan 5 14:23
drwxr-xr-x    17 root  root    1024 Jan 5 14:23
drwxr-xr-x    2  boss  boss    1024 Jan 6 01:09 Neuer Ordner
```

```
-rw-r--r--    1  root  root      69 Jan 5 14:22 file-1.txt
-rw-r--r--    1  root  root      59 Jan 5 14:23 file-2.txt
```

Der von Ihnen erstellte Ordner `Neuer Ordner` wurde als Verzeichnis erzeugt, und zwar mit Ihnen als Besitzer und Ihrer Gruppe als Gruppenbesitzer. Aber dies ist nur möglich, weil Sie das Verzeichnis `/home/first-share` auf den Modus `0777` eingestellt haben, was relativ gefährlich ist.

Sie könnten viele der anderen Parameter einrichten, die für den Zugriff auf `first-share` relevant sind, darunter

- `write list` - Die Liste der Benutzer, die Schreibzugriff auf die Freigabe haben.
- `valid users` - Die Liste der Benutzer, die auf die Freigabe zugreifen können.

Ihre Freigabe kann z.B. so aussehen:

```
[first-share]
   comment = Meine erste Freigabe
   path = /home/first-share
   browsable = yes
   writable = yes
   valid users = boss joe +users
   write list = root boss
```

`first-share` wurde dahingehend geändert, dass `boss`, `joe` und jeder Benutzer in der Unix-Gruppe `users` auf die Freigabe zugreifen, aber nur `boss` und `joe` in die Freigabe schreiben können.

Sie sollten sich jetzt ansehen, wie Dateiberechtigungen von Samba gehandhabt werden, damit Sie verstehen können, wie Unix-Dateiberechtigungen mit Anfragen für das Lesen oder Beschreiben von Dateien zusammenspielen.

Berechtigungen

Wie Sie im vorherigen Abschnitt gesehen haben, führt Samba zuerst grobe Überprüfungen durch, z.B. zum Lese-, Schreib- oder benutzerbasierten Zugriff auf Freigaben. Wenn eine verlangte Operation, wie z.B. das Öffnen einer Datei zum Lesen oder Bearbeiten, diese Zugriffsüberprüfungen durchlaufen hat, muss sie immer noch den normalen Beschränkungen des Betriebssystems für Dateien und Verzeichnisse entsprechen. Diese basieren auf dem Benutzer, den Samba als den zugreifenden Benutzer auf die Dateifreigabe definiert hat.

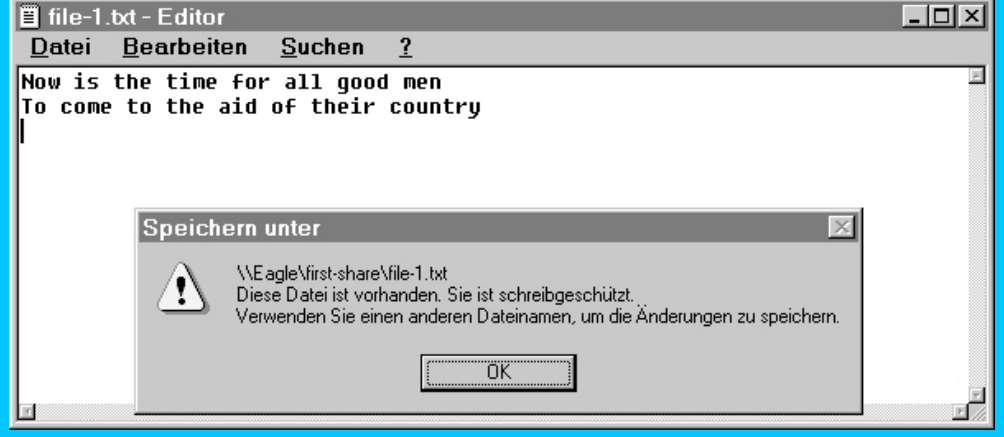
Das heißt, dass normale Unix-Berechtigungen für alle Dateioperationen gelten, nachdem Samba die Zugriffsberechtigung erteilt hat.

Erinnern Sie sich daran, dass Sie vorher die Dateien `file-1.txt` und `file-2.txt` im Editor öffnen konnten. Dies liegt daran, dass sie allgemein lesbar sind und daher jeder auf sie zugreifen kann. Hier ist noch einmal eine detaillierte Auflistung des Freigabeverzeichnisses:

```
ls -al /home/first-share
total 4
drwxr-xr-x    2  root  root    1024 Jan 5 14:23
drwxr-xr-x    17 root  root    1024 Jan 5 14:23
drwxr-xr-x    2  boss  boss    1024 Jan 6 01:09 Neuer Ordner
-rw-r--r--    1  root  root      69 Jan 5 14:22 file-1.txt
-rw-r--r--    1  root  root      59 Jan 5 14:23 file-2.txt
```

Wie Sie sehen können, haben die Dateien `file-1.txt` und `file-2.txt` den Status `0644`, das heißt, sie sind nicht gruppen- oder allgemein beschreibbar. Wenn Sie versuchen, diese Dateien von einem Client zu ändern, werden Sie keinen Erfolg haben. Öffnen Sie eine der Dateien im Editor und versuchen Sie, sie zu ändern. Wann erhalten Sie eine Fehlermeldung? Sie sollten dann eine erhalten, wenn Sie versuchen, Ihre Änderungen zu speichern. Irgendwann gibt der Editor die Fehlermeldung aus, die Sie in Abbildung 7.10 sehen.

Abb. 7.10: Der Editor kann file-1.txt nicht speichern



Diese Fehlermeldung liegt in der Art und Weise begründet, wie der Editor versucht, die Datei zu speichern. Er versucht, die Datei `file-1.txt` in der Dateifreigabe zu erzeugen, aber da diese Datei bereits existiert und Sie keine Berechtigung haben, die Datei zu bearbeiten, wird die Operation nicht ausgeführt.

Würden Sie versuchen, die Datei unter einem Namen zu speichern, der noch nicht in dem Verzeichnis (oder der Freigabe, die Begriffe sind aus Sicht desjenigen, der Samba verwaltet, austauschbar) existiert, wäre der Editor bereit und würde eine neue Datei erstellen.

Was Sie hier sehen, ist Folgendes: Nach allen Zugriffskontrollen, die Samba durchführt, überprüft es außerdem die normalen Berechtigungen für das Dateisystem, die der Benutzer (den Samba als mit der Dateifreigabe verbunden ansieht) für die Dateien in der Freigabe hat!

Wenn Sie keine Leseberechtigungen für die Verzeichnisse haben, werden Sie auch ihre Inhalte nicht betrachten können, obwohl Sie die Verzeichnisse selbst sehen können. Normalerweise erhalten Sie Meldungen mit dem Inhalt »Zugriff verweigert«, wenn Sie versuchen, durch solche Verzeichnisse zu browsen.

Nun, wenn Dateien und Verzeichnisse in einer Samba-Freigabe erzeugt werden, wer besitzt sie und welche Berechtigungen werden ihnen vererbt? Schauen Sie sich noch einmal das lange Listing für das Verzeichnis `first-share` an und werfen Sie einen Blick auf den Eintrag für den Ordner `Neuer Ordner`.

Hier noch einmal der Eintrag:

```
drwxr-xr-x    2  boss  boss    1024 Jan 6 01:09 Neuer Ordner
```

Sie sind der Besitzer des Verzeichnisses, das sich in Ihrer primären Gruppe befindet und das Dateirecht 0755 hat.

Lassen Sie uns nun einen kurzen Blick auf Unix-Dateirechte werfen, da Samba Nummern wie 0755 benutzen möchte, wie Sie vorher gesehen haben. Nachfolgend zeige ich Ihnen, was sie bedeuten und wie Sie sie konstruieren können.

Jede Datei in einem Unix-Dateisystem hat einen Besitzer, eine Gruppenzugehörigkeit und Dateirechte (manchmal auch Berechtigungen genannt), die aus vier Teilen bestehen:

- Die SETUID-, SETGID- und T-Bits.
- Die Benutzer- oder Besitzerberechtigungen, die eine beliebige Kombination aus Lese-, Schreib- und Ausführungsrecht sein können. Sie bestimmen, welche Zugriffsrechte der Besitzer auf die Datei hat.
- Die Gruppenberechtigungen, die eine beliebige Kombination aus Lese-, Schreib- und Ausführungsrecht sein können. Sie bestimmen, welche Zugriffsrechte jeder der Benutzer der Gruppe, in der sich die Datei befindet, auf die Datei hat.
- Die anderen oder allgemeinen Berechtigungen, die eine beliebige Kombination aus Lese-, Schreib- und Ausführungsrecht sein können. Sie bestimmen, welche Zugriffsrechte jeder auf die Datei hat, der nicht der Besitzer ist und sich nicht in der Gruppe des Besitzers befindet.

Ein Dateirecht wird normalerweise durch die drei Buchstaben `RWX` und den Bindestrich (-) dargestellt. `RWX` heißt Lesen, Schreiben und Ausführen, während `R-X` Lesen und Ausführen bedeutet. Um ein Dateirecht zu ändern, drücken Sie es als eine Serie von 12 Bits oktal aus. Daher wird das Recht einer Datei als vier oktale Ziffern dargestellt, wobei die erste Ziffer auf folgende Art codiert ist:

4=SETUID, 2=SETGID, 1=T oder STICKY Bit

Die verbleibenden drei Ziffern (die eigentlichen Berechtigungsbits) werden auf folgende Weise codiert:

4 = Lesen, 2 = Schreiben, 1 = Ausführen

Wenn Sie ein bestimmtes Recht für eine Datei konstruieren wollen, addieren Sie einfach die Codierungen für die von Ihnen gewünschten Berechtigungen. `RWX` summiert sich also oktal zu 7, `RW-` zu 6, `R-X` zu 5 usw.

Ein Dateirecht von 1755 heißt also:

- Für die Datei ist das T-Bit eingestellt, oder für das Verzeichnis wurde das Sticky Bit eingerichtet.

- Der Besitzer hat Lese-, Schreib- und Ausführungzugriff.
- Mitglieder der Besitzergruppe haben Lese- und Ausführungszugriff.
- Jeder andere hat Lese- und Ausführungszugriff.

Abschließend möchte ich noch erwähnen, dass eine Dateiberechtigung von 0755 von `ls -al` als `RWXR-XR-X` ausgegeben wird.

Samba handhabt die Erstellungsberechtigungen für Dateien und Verzeichnisse separat. Sie können eine ganze Reihe von Parametern einrichten, um sowohl die Eigentumsverhältnisse als auch die Berechtigungen für erstellte Dateien und Verzeichnisse zu kontrollieren.

Parameter für die Erstellung von Dateien und Verzeichnissen

Die folgenden Abschnitte listen viele der Freigabe-Parameter auf, die für die Berechtigungen und Eigentumsverhältnisse der von Samba erstellten Dateien und Verzeichnisse relevant sind. Wie immer finden Sie eine komplette Auflistung der Parameter und ihre aktuellen Funktionen in den Manpages zur `smb.conf` der aktuellen Samba-Version.

create mask, create mode

Diese Freigabe-Parameter sind synonym und kontrollieren die Berechtigungen, die bei der Erstellung von Dateien vergeben werden. Der gegebene Wert ist eine Bitmaske, die mit der aus dem verlangten DOS-Attrribut kalkulierten Unix-Maske verglichen wird.

Jedes nicht eingestellte Bit der Maske wird bei Erstellung der Datei aus den Berechtigungen für die Datei entfernt.

Standardmäßig hat die Erstellungsmaske einen Wert von `0744`, der festlegt, dass der Besitzer für neue Dateien `RWX`-Berechtigungen erhält, während Mitglieder der Besitzergruppe und alle anderen Benutzer nur die Berechtigungen `R--` erhalten.

Ein Beispiel:
```
create mask = 0755
```

Diese Einstellung definiert, dass der Besitzer `RWX`-Berechtigungen hat, Mitglieder der Besitzergruppe `R-X`-Berechtigungen und alle anderen Benutzer `R-X`-Berechtigungen erhalten.

directory mask, directory mode

Diese Freigabe-Parameter sind synonym und kontrollieren die Berechtigungen, die während der Erstellung von Verzeichnissen vergeben werden. Der gegebene Wert ist eine Bitmaske, die mit der Unix-Maske verglichen und aus dem verlangten DOS-Atrribut kalkuliert wird.

Jedes nicht eingestellte Bit in der Maske wird aus den Berechtigungen für die Datei entfernt, wenn es erstellt wird.

Standardmäßig hat die Verzeichnismaske einen Wert von `0755`. Ein Beispiel:
```
directory mask = 0744
```

Diese Einstellung definiert, dass der Besitzer `RWX`-Berechtigungen hat, Mitglieder der Besitzergruppe und alle anderen Benutzer nur die Berechtigungen `R--` erhalten.

Das Ausführungsbit hat eine spezielle Bedeutung für Verzeichnisse. Es ermöglicht einem Benutzer, in dieses Verzeichnis zu wechseln. Wenn also Benutzer keinen X-Zugriff auf ein Verzeichnis haben, können sie Dateien in dem Verzeichnis öffnen, wenn sie Lesezugriff auf die Dateien haben, aber sie können nicht in das Verzeichnis wechseln. Aufgrund der Art und Weise jedoch, wie Samba und Windows das Browsing von Ordnern handhaben, spielt das X-Bit für den Zugriff von Windows aus keine Rolle.

force create mode

Über diesen Freigabe-Parameter können Sie bestimmte Berechtigungsbits erzwingen, wenn Dateien in einer Freigabe erstellt werden. Dies können Sie tun, indem Sie ein bitweises `or` der hier spezifizierten Bits mit den Bits durchführen, die von `create mask` berechnet werden. Beachten Sie, dass damit der Paramter `force create mode` den Parameter `create mask` außer Kraft setzt.

Der Standardwert für diesen Paramter ist `0000`, was heißt, dass keine zusätzlichen Berechtigungsbits in den Parameter `create mask/mode` gezwungen werden. Ein Beispiel:
```
force create mode = 0755
```

Mit dieser Einstellung haben die erstellten Dateien eine Berechtigung von mindestens 0755 (oder `RWXR-XR-X`).

force directory mode

Über diesen Freigabe-Parameter können Sie bestimmte Berechtigungsbits erzwingen, wenn Verzeichnisse in einer Freigabe erstellt werden. Dies können Sie `tun,` indem Sie ein bitweises `or` der hier spezifizierten Bits mit den Bits durchführen, die von `create mask` berechnet werden. Beachten Sie, dass damit der Paramter `force directory mode` den Parameter `directory mask` außer Kraft setzt.

Der Standardwert für diesen Paramter ist `0000`, was heißt, dass keine zusätzlichen Berechtigungsbits in den Parameter `directory mask/mode` gezwungen werden. Ein Beispiel:

```
force directory mode = 0755
```

Mit dieser Einstellung haben die erstellten Verzeichnisse eine Berechtigung von mindestens 0755 (oder `RWXR-XR-X`).

force group

Dieser Freigabe-Parameter spezifiziert einen Unix-Gruppennamen, der als Standard-Primärgruppe für alle Benutzer verwendet wird, die auf die Freigabe zugreifen.

Standardmäßig hat dieser Parameter keinen Wert, was bedeutet, dass alle neuen Dateien und Verzeichnisse über die Anwendung der normalen Unix-Regeln einen Gruppenbesitzer erhalten (ist das SETGID-Bit auf das Vaterverzeichnis eingestellt, wird die Gruppe dieses Verzeichnisses benutzt, sonst die primäre Gruppe des Erstellers).

Ein Beispiel:

```
force group = users
```

Mit dieser Einstellung werden alle neuen Dateien in der Freigabe mit einem Gruppenbesitzer von `users` erstellt.

force user

Dieser Freigabe-Parameter spezifiziert einen Unix-Benutzernamen, der als Standardbenutzer für alle Benutzer verwendet wird, die auf die Freigabe zugreifen.

Standardmäßig hat dieser Parameter keinen Wert, was bedeutet, dass alle neuen Dateien in der Freigabe dem Unix-Benutzer gehören, der als mit der Freigabe verbunden angesehen wird (siehe den Abschnitt »Zugriffsrechte« vorher in diesem Kapitel). Ein Beispiel:

```
force user = boss
```

Diese Einstellung bedeutet, dass alle neue Dateien in der Freigabe Eigentum von `boss` sind.

Einige Beispiele

Sie haben nun viele der Parameter kennengelernt, die für die Erstellung von Dateien und den Zugriff auf Dateien in Samba relevant sind. Wie können Sie diese anwenden? Hier sind einige Beispiele.

Wenn Sie wollen, dass alle Dateien, die in einem bestimmten Verzeichnis erstellt werden, einer bestimmten Gruppe gehören, benutzen Sie den Parameter `force group`. Wollen Sie z.B., dass alle Dateien und Verzeichnisse, die in einer bestimmten Freigabe erzeugt werden, den Gruppen-Accounts gehören, verwenden Sie für die Freigabe den folgenden Parameter:

```
force group = accounts
```

Um zu verhindern, dass alle Dateien und Verzeichnisse, die in einer bestimmten Freigabe erstellt werden, allgemein offene Berechtigungen haben (um vielleicht Unix-Benutzer daran zu hindern, auf die Dateien in der Freigabe zuzugreifen), benutzen Sie die folgenden Parameter für die Freigabe:

```
create mask = 0750
directory mask = 0750
```

Sie müssen beide Parameter definieren, da Samba die Erstellung von Dateien und Verzeichnissen separat handhabt.

Modifizieren Sie Ihre Freigabe `first-share`, um einige dieser Änderungen einzufügen:

```
[first-share]
comment = Meine erste Freigabe
   path = /home/first-share
   browsable = yes
   writable = yes
   create mask = 0750
   create directory = 0750
   force group = users
```

Nachdem Sie Ihre `smb.conf` modifiziert haben, um `first-share` mit den vorher beschriebenen Parametern zu ändern, starten Sie Samba neu und erstellen von einem Client die Datei `new-file.txt` und das Verzeichnis `Neuer Ordner (2)`.

Wenn Sie sich das Freigabeverzeichnis komplett auflisten lassen, sollten Sie nun Folgendes sehen:

```
ls -al /home/first-share
```

```
total 4
drwxrwxrwx   4  root  root    1024 Jan 6 16:58
drwxr-xr-x  17  root  root    1024 Jan 5 14:23
drwxr-x---   2  boss  boss    1024 Jan 6 14:53 Neuer Ordner
drwxr-x---   2  boss  users   1024 Jan 6 16:58 Neuer Ordner (2)
-rw-r--r--   1  root  root      69 Jan 5 14:22 file-1.txt
-rw-r--r--   1  root  root      59 Jan 5 14:23 file-2.txt
-rwxr-----   1  boss  users      0 Jan 6 16:58 new-file.txt
```

Beachten Sie, dass die Datei `new-file.txt` und das Verzeichnis `Neuer Ordner (2)` beide den Gruppenbesitzer `users` und keine allgemeinen Berechtigungen haben.

Sie haben für das Verzeichnis `/home/first-share` die Berechtigungen 0777 eingerichtet, was sehr gefährlich ist. Eine bessere Methode, über die Sie Dateien in der Freigabe von Clients erstellen können, besteht darin, den Gruppenbesitzer des Verzeichnisses in eine Gruppe zu ändern, der Sie angehören.

Um dies zu tun, müssen Sie herausfinden, zu welchen Gruppen auf dem Unix-Rechner Sie gehören. Wenn Sie sich also als `boss` in Ihren Client eingeloggt haben, müssen Sie feststellen, welchen Gruppen `boss` angehört:

```
group boss
boss: boss wheel users
```

Ändern Sie dann den Gruppenbesitzer des Verzeichnisses `/home/first-share` in eine dieser Gruppen. `users` ist eine gute Wahl, besonders wenn Sie planen, dass andere Leute auf die Dateien in der Freigabe zugreifen und sie gemeinsam benutzen sollen. Sie müssen außerdem das Gruppen-Schreibbit für das Verzeichnis einrichten.

Um diese Änderungen durchzuführen, benutzen Sie die folgenden Befehle:

```
chgrp users /home/first-share
chmod 0775 /home/first-share
```

Wenn Sie sicherstellen wollen, dass diese Änderungen für alle Dateien und Verzeichnisse in der Freigabe gelten, fügen Sie den Befehlen `chgrp` und `chmod` den Parameter `-R` hinzu.

Befehle:

```
chgrp -R users /home/first-share
chmod -R 0775 /home/first-share
```

All dies verdeutlicht einen sehr nützlichen Aspekt von Samba: Sie können die Dateien in Ihren Dateifreigaben von Unix aus verwalten. Das heißt, Sie können auf alle Standard-Unix-Funktionen zugreifen, darunter Skripting (auch Perl) und `cron`-Dateien.

In einem Studentenlabor z.B., in dem von den Studenten verlangt wird, dass sie ihre Laborarbeiten zu einem bestimmten Datum und einer bestimmten Zeit übermitteln (indem sie es in die Freigabe `\\eagle\labwork` kopieren), bewirkt das folgende Shell-Skript, dass

- die Dateien Eigentum des Professors werden, damit Studenten ihre Beiträge nach dem Abgabetermin nicht mehr ändern können,
- der Professor eine Mail erhält, die ihm mitteilt, welche Studenten ihre Arbeit noch nicht übermittelt haben.

```
#!/bin/sh
# Finde heraus, wer seine Laborarbeit noch nicht übertragen
# hat, und ändere dann den Besitzer, damit Studenten nicht
# nach dem Abgabetermin abgeben können.
# Sende Mail an Professor über die, die ihre Arbeit nicht
# abgegeben haben. Offensichtlich gibt es ein kleines
# Risiko, dass jemand erneut übertragen kann. Dies könnte
# dadurch behoben werden, den Samba-Server zu beenden.
# Zunächst sollten alle Studenten daran gehindert werden,
# Dateien von PCs in das Verzeichnis einzufügen.
chmod 1700 /home/labwork
# Finde dann heraus, wem die übertragene Arbeit gehört
ls -ld /shares/labwork | tr -s " " " " | cut -f3 -d" " > /tmp/submitted.$$
# Ändere nun den Besitzer dieser Dateien
chown -R professor /home/labwork/*
# Finde nun heraus, wer noch nicht übertragen hat,
# basierend auf einer Studentenliste
diff -y /tmp/submitted.$$ /home/professor/students | grep \< | \ mail -s "Studenten, die Ihre Arbeit nicht rechtzeitig übertragen haben" professor
rm -f /tmp/submitted.$$
```

Dieses Shell-Skript ist nur ein Beispiel. In der Realität, in der Professoren oft mehrere Klassen haben usw., wären einige Änderungen nötig.

Spezielle Dateifreigaben

Jetzt, da Sie sich angesehen haben, wie Dateifreigaben eingerichtet werden, wie Sie den Zugriff auf diese Freigaben kontrollieren und wie Sie Zugriffsberechtigungen für Freigaben handhaben, ist es an der Zeit, einige spezielle Freigaben anzusehen, die Samba bietet.

Clients greifen gern auf ihre Home-Verzeichnisse zu, und wenn Samba das Home-Verzeichnis eines jeden Benutzers als separate Dateifreigabe darstellen müsste, hätten Administratoren ein schweres Leben. Stellen Sie sich vor, Sie müssten für jeden neuen Benutzer, den Sie dem System hinzufügen, einen Abschnitt in `smb.conf` einfügen und dann jedes Mal Samba neu starten.

Um das Leben all der überarbeiteten Administratoren etwas leichter zu gestalten, bietet Samba zwei spezielle Freigabeabschnitte: `[homes]` und `[printers]`. Ich werde den Abschnitt `[printers]` im nächsten Kapitel ausführlich darstellen. Die Details zum Abschnitt `[homes]` finden Sie nachfolgend.

Wenn ein Client eine Verbindung zu einer Dateifreigabe anfordert, werden die existierenden Dateifreigaben überprüft. Wird eine Entsprechung gefunden, wird diese Freigabe zur Verfügung gestellt. Wenn Samba aber keine Entsprechung findet, wird die angeforderte Freigabe als Benutzername behandelt und in der `passwd` gesucht. Existiert der Name und kann er authentifiziert werden, wird eine Freigabe durch Klonen der `[homes]`-Freigabe erstellt. Das heißt, dass die neue Freigabe die meisten der Parameter in der `[homes]`-Freigabe übernimmt.

Wenn die neue Freigabe erstellt wird, wird ihr Name in den Benutzernamen geändert und der Pfad der Freigabe auf das Home-Verzeichnis des Benutzers eingestellt, wenn kein solches im Abschnitt `[homes]` definiert ist.

Nachfolgend finden Sie ein Beispiel für eine `[homes]`-Freigabe in der `smb.conf`:

```
[homes]
   comment = Home-Verzeichnisse
   browsable = no
   writable = yes
```

Das ist alles, was Sie brauchen, damit Clients auf das angeforderte Home-Verzeichnis zugreifen können. Fügen Sie die oben stehenden Einträge in Ihre `smb.conf` ein und starten Sie Samba neu. Sie sollten dann unter Windows die DOS-Eingabeaufforderung aufrufen und folgenden Befehl ausführen können:

```
net use h: \\EAGLE\homes
```

Danach können Sie auf alle Dateien in Ihrem Home-Verzeichnis von Ihrem PC zugreifen.

Handhabung und Umsetzung von Dateinamen

Unix-Dateinamen und Windows-Dateinamen folgen verschiedenen Regeln.

Unix erlaubt fast jedes Zeichen in einem Dateinamen, außer dem Verzeichnistrennzeichen (/) und Escape, und unterscheidet in Namen zwischen großen und kleinen Buchstaben. Außerdem können Unix-Dateinamen sehr lang sein (bis zu 255 Zeichen). Auch Pfadnamen sind unter Unix oft bis zu 1.024 Zeichen lang.

DOS (6.22 und früher) dagegen hat die Einschränkung auf die 8.3-Namen, die nicht länger als acht Zeichen sein dürfen, mit einer Dateierweiterung, die aus nicht mehr als drei Zeichen bestehen darf. Außerdem benutzt DOS für Datei- und Verzeichnisnamen Großbuchstaben und beschränkt die Länge der Pfadnamen, die erheblich kürzer sind als die, die in Unix-Systemen erlaubt sind. Windows für Workgroups folgt den DOS-Beschränkungen.

Für Windows 95 (mit DOS 95) und Windows NT wurden viele dieser Einschränkungen aufgehoben, so dass Dateinamen länger als 11 Zeichen sein können und sowohl große als auch kleine Buchstaben in Dateinamen erlaubt sind. Aber auch Windows 95 und Windows NT haben Beschränkungen für die Länge von Datei- und Pfadnamen, die wesentlich kleiner sind als in Unix-Systemen. Windows 95 kürzt Dateinamen nach 127 Zeichen ab. Beträgt der komplette Pfadname mehr als 255 Zeichen (inklusive dem Server- und den Freigabenamen), weigert sich Windows 95, weitere Ordner oder Dateien zu erstellen. Windows NT unterliegt den gleichen Beschränkungen. Aber sowohl Windows 95 als auch Windows NT können mit längeren Datei- und Pfadnamen umgehen, wenn diese bereits in der Samba-Freigabe existieren (und vielleicht unter Unix erzeugt wurden).

Um für Kompatibilität mit älteren Clients (DOS, Windows für Workgroups, PATHWORKS usw.) und Anwendungen, die von 8.3-Dateinamen abhängen, zu sorgen, bietet Samba viele Freigabe-Parameter, die kontrollieren, wie Unix-Dateinamen und -Pfadnamen an Clients ausgegeben werden. Außerdem gibt es Parameter, die kontrollieren, wie die Groß-/Kleinschreibung beim Erzeugen neuer Dateien verwendet wird.

Samba bezeichnet diese Umsetzung der Dateinamen als *Name Mangling* und verwendet den folgenden allgemeinen Ansatz:

- Die ersten fünf alphanumerischen Zeichen vor dem ersten Punkt werden in Großbuchstaben umgewandelt und als erster Teil des umgesetzten Namens verwendet.
- An diesen ersten Teil des umgesetzten Namens wird eine Tilde (~) angehängt, gefolgt von einem aus zwei Zeichen bestehenden Hash-Wert des ursprünglichen `root`-Namens (d.h. ohne die ursprüngliche Erweiterung). Die Erweiterung wird aber in die Berechnung für den Hash-Wert eingefügt, wenn sie groß geschriebene Zeichen enthält oder länger als drei Zeichen ist. Statt der Tilde kann auch ein anderes Zeichen verwendet werden (über den Parameter `mangling char`), wenn Benutzer widersprechen oder Anwendungen Probleme mit Tilden haben.
- Die ersten drei Zeichen der ursprünglichen Erweiterung werden in Großbuchstaben umgewandelt und als Erweiterung für den umgesetzten Namen verwendet. Wenn der Dateiname keinen Punkt enthält, hat der umgesetzte Name keine Erweiterung.

- Dateien, die Unix-Namen haben, die mit einem Punkt beginnen, werden als versteckte DOS-Dateien behandelt. Ihre umgesetzten Namen ähneln den vorher dargestellten, allerdings mit der Ausnahme, dass der erste Punkt entfernt und eine Erweiterung von ___ (d.h. drei Underscores) hinzugefügt wird, unabhängig von der ursprünglichen Erweiterung.

Nachfolgend finden Sie ein Beispiel für die Umsetzung von Namen, die von einem Windows-95-DOS-Rechner durchgeführt wurde, der Ihre Freigabe `first-share` auflistet, nachdem einige neue Dateien und Ordner hinzugefügt wurden. Die umgesetzten Namen sehen Sie auf der linken Seite und die vollständigen Namen auf der rechten.

```
E:\>dir
Volume in drive E ist FIRST-SHARE
Directory of E:\
file-1   txt       69  01-05-99  2:22p  file-1.txt
file-2   txt       59  01-05-99  2:23p  file-2.txt
NEWFO~YX     <DIR>     01-06-99  2:53p  New Folder
File-1   txt       69  01-06-99  5:33p  File-1.txt
new-file txt        0  01-06-99  4:58p  new-file.txt
ANOTH~9Y     <DIR>     01-06-99  4:58p  another-new-folder
A-FIL~BH TXT       24  01-07-99 12:45a  a-file-with-a-long-name.txt
        5 file(s)          221 bytes
        2 dir(s)     33,488,896 bytes free
E:\>
```

Standardmäßig arbeitet Samba 2.0 wie ein Windows-NT-Server: D.h., es unterscheidet nicht zwischen Groß- und Kleinschreibung, behält aber die jeweilige Groß-/Kleinschreibung bei. Wenn Samba also Dateien öffnet, stellt es Dateinamen entsprechend in einer nicht groß-/kleinsensitiven Art und Weise dar, wenn es aber neue Dateien erstellt, behält Samba die Schreibweise bei, die der Client verwendet.

In den meisten Fällen sind die Standardeinstellungen von Samba genau richtig, aber Sie müssen möglicherweise einige dieser Einstellungen für spezielle Clients oder Anwendungen ändern.

Die folgenden Parameter sind für die Handhabung von Dateinamen relevant. Wie immer finden Sie eine vollständige Liste der Parameter und das letzte Wort zu ihren Funktionen in den Manpages zu `smb.conf` für die aktuelle Samba-Version.

mangled names

Dieser Freigabe-Parameter kontrolliert, ob Unix-Namen, die nicht mit DOS-Namen kompatibel sind, in DOS-kompatible Namen umgesetzt werden sollen. Standardmäßig setzt Samba Namen für Clients um, die Nicht-DOS-Namen nicht handhaben können.

Der Standardwert für diesen Parameter ist `yes`. Standardmäßig werden also Nicht-DOS-Namen in ihre DOS-kompatiblen Entsprechungen umgesetzt.

Wenn Sie den Wert für diesen Parameter auf `no` setzen, werden die Namen nicht umgesetzt. Dann sehen DOS-Clients und DOS-Befehlsprompts den Dateinamen einfach abgeschnitten, wie es den normalen DOS-Regeln entspricht.

mangle case

Dieser Freigabe-Parameter kontrolliert, ob Dateinamen umgesetzt werden, die nicht der Standardschreibweise (definiert über `default case`) entsprechen. Ist *dieser* Parameter aktiviert, werden Namen wie z.B. *Mail* in die Standardschreibweise umgesetzt.

Der Standardwert für diesen Parameter ist `no`. Damit wird definiert, dass Namen in gemischter Schreibweise nicht umgesetzt werden.

mangling char

Dieser Freigabe-Parameter definiert das Zeichen, das Samba als Mangling-Zeichen verwendet, wenn es Namen umsetzt. Standardmäßig ist dies die Tilde (~).

Ein Beispiel:

```
mangle char = ^
```

Mit dieser Einstellung benutzt Samba das (^) statt der Tilde.

case sensitive

Dieser Freigabe-Parameter bestimmt, ob Samba bei Dateinamen auf die Groß-/Kleinschreibung achten soll. Ist dieser Parameter auf `no` eingestellt, muss Samba eine nicht groß-/kleinsensitive Suche für alle Dateinamen durchführen, die von Clients übertragen werden.

Der Standardwert für diesen Parameter ist `no`.

default case

Dieser Freigabe-Parameter kontrolliert die Standardschreibweise für neue Dateien und sollte in Kombination mit dem Parameter `preserve case` verwendet werden.

Die Standardschreibweise ist Kleinschreibung.

preserve case

Dieser Freigabe-Parameter kontrolliert das Verhalten von Samba beim Erstellen neuer Dateien. Ist der Wert für diesen Parameter auf `yes` gesetzt, wird die Schreibweise benutzt, die der Client verwendet (auch gemischte Schreibweise), sonst wird die Schreibweise verwendet, die durch den Parameter `default case` definiert ist.

Standardmäßig wird die Schreibweise beibehalten.

short preserve case

Dieser Freigabe-Parameter kontrolliert das Verhalten von Samba beim Erstellen von Dateien mit DOS-kompatiblen Namen (d.h. 8.3-Namen in Großbuchstaben). Ist der Wert für diesen Parameter auf `yes` gesetzt, werden solche Dateien mit groß geschriebenen Namen erstellt, sonst wird die Schreibweise verwendet, die durch den Parameter `default case` definiert ist.

Dieser Parameter kann mit `preserve case = yes` verwendet werden, damit lange Dateinamen ihre Schreibweise beibehalten können, während kurze Namen klein geschrieben werden.

Der Standardwert für diesen Parameter ist `yes`.

Datei-Locking

Standardmäßig unterstützt Samba zwei verschiedene Arten von Datei-Locking, *share modes* und opportunistisches Locking, kurz *oplocks*.

share modes unterstützen die Standard-DOS/Windows-Zugriffsanfragen von `DENY_DOS`, `DENY_ALL`, `DENY_READ`, `DENY_WRITE`, `DENY_NONE` und `DENY_FCB`.

Unter den Unix-Versionen, die *Shared Memory* (gemeinsamen Speicher) unterstützen (die meisten Unix-Versionen), wird die Unterstützung für `share modes` unter Benutzung von Shared Memory implementiert, was sehr schnell ist. Wenn Ihre Unix-Version Shared Memory nicht unterstützt, wird die Unterstützung für `share modes` unter Benutzung von Lock-Dateien implementiert, was sehr langsam sein kann.

Wahrscheinlich müssen Sie `share modes` nicht deaktivieren. Sollte dies doch einmal der Fall sein, können Sie sie über den folgenden Befehl für jede einzelne Freigabe deaktivieren:

```
share modes = no
```

oplocks sind eine Leistungserweiterung, die zusammen mit dem Windows-NT-Server eingeführt wurden. Sie ermöglichen einem Client, viele Dateioperationen zwischenzuspeichern, solange der Client der einzige ist, der auf eine bestimmte Datei zugreift. Öffnet ein anderer Client die gleiche Datei, muss der Server dem Client mit dem oplock einen *oplock break* schicken, sodass dieser Client das lokale Zwischenspeichern beendet.

Wenn Clients oplocks erhalten können, sind Leistungssteigerungen von 30 Prozent und mehr möglich, da Clients aggressive Zwischenspeicherungen von Dateioperationen durchführen können (inklusive Öffnen und Schließen und möglicherweise erneutes Ausführen einiger Operationen im Zwischenspeicher).

oplocks sind standardmäßig in Samba aktiviert. In einigen Fällen ist es möglich, dass Client-Programme bei aktivierten oplocks nicht richtig funktionieren, also wollen Sie sie möglicherweise deaktivieren. Dies können Sie über den `oplocks`-Befehl für einzelne Freigaben erreichen:

```
oplocks = false
```

Sie können die oplocks für einzelne Dateien auch über den Parameter `veto oplock files` deaktivieren:

```
veto oplock files = /*.mbx/
```

Der Vollständigkeit halber möchte ich noch erwähnen, dass SGIs Irix 6.5.2f jetzt oplock-Unterstützung auf Kernel-Ebene bietet, und Linux und BSD dies ebenfalls bald bieten werden. Samba kann Kernel-oplocks erkennen und sie benutzen, wenn sie verfügbar sind. So können oplocks unterbrochen werden, wann immer ein lokaler Unix-Prozess oder eine NFS-Operation auf eine Datei zugreift, die der `smbd` geschützt hat. Dies bietet größere Datenkonsistenz zwischen SMB, NFS und lokalen Dateizugriffen.

Obwohl Sie normalerweise den Parameter `kernel` nicht brauchen, können Sie oplocks auch folgendermaßen deaktivieren:

```
kernel oplocks = off
```

Symbolische Links

Standardmäßig folgt Samba symbolischen Links im Unix-Dateisystem, die auf Dateien innerhalb des freigegebenen Verzeichnisses verweisen, nicht aber solchen zu Dateien/Verzeichnissen außerhalb dieses Verzeichnisses.

Zwei Freigabe-Parameter kontrollieren dieses Verhalten: `follow symlinks` und `wide links`.

Standardmäßig sind diese Parameter auf `yes` bzw. `no` gesetzt.

Wenn Sie `follow symlinks` auf `no` setzen werden keine symbolischen Links mehr verfolgt, was zu einer geringen Leistungsabnahme führt.

Die Einstellung `wide links = yes` führt dazu, dass auch Links außerhalb des freigegebenen Verzeichnisses verfolgt werden.

Handhabung von CD-ROMs

Ein Problem mit der Freigabe von CD-ROMs liegt darin, dass sie in das Dateisystem auf dem Samba-Server gemountet werden müssen. Wenn ein Benutzer die CD-ROM in einem CD-ROM-Laufwerk auswechselt und auf die neue CD zugreifen möchte, muss jemand oder etwas auf Ihrem Samba-Server eingreifen und die CD-ROM mounten. Wäre es nicht großartig, wenn die CD-ROM gemountet werden könnte, sobald der Client auf die CD-ROM-Freigabe zugreift?

Nun, Samba bietet eine solche Funktion mit den Befehlen `preexec/postexec` und `root preexec/root postexec`. Diese Parameter ermöglichen die Ausführung bestimmter Unix-Befehle, wenn ein Client sich mit einer Dateifreigabe verbindet bzw. wenn er die Verbindung zu einer Dateifreigabe beendet. Die Root-Version führt den Befehl einfach als `root` aus.

Eine CD-ROM-Freigabe könnte so aussehen:

```
[cdrom]
   comment = CD-ROM, bei Verbindung automatisch gemountet
   browsable = yes
   read only = yes
   path = /mnt/cdrom
   root preexec = /bin/mount /dev/hdd /mnt/cdrom
   root postexec = /bin/umount /mnt/cdrom
```

Natürlich ist das tatsächlich benutzte Gerät (`/dev/hdd`) abhängig von Ihrem System (dieses gilt für ein Linux-System mit der CD-ROM am zweiten IDE-Controller als Slave-Gerät).

Mit einer solchen Freigabe können Benutzer die CD-ROM im CD-ROM-Laufwerk auf dem Server auswechseln und sie dann erneut mappen (z.B. mit `net use /d v:` und dann `net use v: \\EAGLE\cdrom`).

Andere Parameter

Die folgenden Parameter sind alle auf irgendeine Art und Weise relevant für die Freigabe von Dateien, passen aber nicht richtig in einen der vorher beschriebenen Absätze.

maxopenfiles

Dieser globale Parameter kontrolliert die maximale Anzahl offener Dateien, die ein `smbd`-Dateifreigabeprozess für einen Client geöffnet haben kann. Seit Samba 2.0.0 ist der Standardwert für diesen Parameter `10.000` Dateien, obwohl `smbd` dies auf einen sinnvolleren Wert setzt, wenn das Betriebssystem so viele offene Dateien nicht unterstützt. Unter Linux wird `maxopenfiles` also standardmäßig auf etwa `246` eingestellt.

In früheren Samba-Versionen war `maxopenfiles` ein Parameter, dessen Wert während der Kompilierung festgesetzt wurde.

nis homedir und homedir map

Diese globalen Parameter weisen Samba an, den Standort von Home-Verzeichnissen über NIS zu holen. Sie werden in Situationen benutzt, in denen sich das Home-Verzeichnis eines Benutzers auf einem entfernten Rechner befindet und Samba über NFS auf dieses zugreifen würde.

Solange wie auf den tatsächlichen Home-Verzeichnis-Servern ebenfalls Samba läuft, kann ein Logon-Server die Heimatfreigabe so zurückgeben, dass sie sich auf einem anderen Server befindet. Dafür konsultiert er die NIS-Map, die über den Parameter `homedir map` definiert ist. Dies funktioniert nur, wenn es einen funktionierenden NIS-Server gibt und Samba als Logon-Server läuft.

Die Standardwerte für `nis homedir` und `homedir map` sind `false` bzw. `auto.home`.

ole locking compatibility

Dieser globale Parameter ermöglicht es Ihnen, die OLE-Kompatibilität für Bereichslocks zu deaktivieren, die Samba bietet. Einige Unix-Lockmanager können abstürzen oder andere Probleme haben, wenn die OLE-Funktion von Samba aktiviert ist, daher wollen sie sie vielleicht deaktivieren.

Der Standardwert für diesen Parameter ist `yes`.

strip dot

Dieser globale Parameter definiert, ob Samba abschließende Punkte (dots) von Unix-Dateinamen abschneidet. Einige CD-ROMs haben Dateinamen, die mit einem einzelnen Punkt enden.

Der Standardwert für diesen Parameter ist `no`.

Zusammenfassung

In diesem Kapitel haben Sie die Freigabe von Dateien und viele der Parameter betrachtet, die kontrollieren, wie Dateifreigaben für Clients verfügbar gemacht werden. Sie haben ebenfalls gelernt, wie Dateien in diesen Dateifreigaben erstellt werden und wie auf sie zugegriffen werden kann. Außerdem haben Sie einen Blick auf fortschrittlichere Funktionen im Zusammenhang mit Dateifreigaben geworfen.

Dabei haben Sie sich detailliert die einzelnen Schritte angesehen, die Samba durchläuft, um festzulegen, ob eine Freigabe existiert, ob ein bestimmter Client auf eine verlangte Freigabe zugreifen und ob dieser Client Dateien in der verlangten Freigabe lesen oder beschreiben kann. Mit diesen Informationen können Sie jetzt viele Probleme in der Konfiguration von Samba meistern.

Im nächsten Kapitel werden Sie sich ansehen, wie Sie Druckerfreigaben einrichten, wie Sie Samba installieren, damit es die automatische Installation von Druckertreibern unter Windows 9x unterstützt, und wie Sie von Unix-Systemen aus zu Windows-Clients drucken, die mit Druckern verbunden sind.

Frage & Antwort

F. Meine smb.conf-Datei enthält eine Dateifreigabe namens [docs]. In dieser Freigabe sollen die Autoren von Dokumenten ihre Dokumente speichern können. Der Gruppeneigentümer des Verzeichnisses ist docs, und alle Autoren sind Mitglieder dieser Gruppe. Die Berechtigungen für das Verzeichnis sind 0770, aber niemand kann in das Verzeichnis schreiben. Was habe ich falsch gemacht?

. Haben Sie einen der folgenden Parameter in der Freigabe definiert?

```
writable = yes
writeable = yes
read only = no
```

Denken Sie daran, dass eine Freigabe standardmäßig mit Nur-Lese-Rechten besetzt ist und Sie das Schreibrecht erst aktivieren müssen, bevor jemand in sie schreiben kann, unabhängig von den Verzeichnis- oder Dateiberechtigungen in der Freigabe.

F. Ich habe eine neue Freigabe namens kits definiert, aber niemand kann sich mit ihr verbinden. Einige Benutzer erhalten die Fehlermeldung: »Das angegebene Freigabeverzeichnis kann nicht gefunden werden.« Andere bekommen ein Dialogfeld, das besagt: »Kann nicht auf \\server\kits zugreifen ...« Was könnte das Problem sein?

. Überprüfen Sie die Pfadangabe in Ihrem Freigabeabschnitt für `[kits]`. Wenn der Pfad nicht existiert oder die Benutzer nicht darauf zugreifen dürfen, erhalten sie diese Art von Fehlermeldungen.

F. Wie klein kann ein [homes]-Abschnitt sein? Wenn die Freigabe nicht browsable ist, braucht sie doch eigentlich keinen Kommentar.

F. Sie müssen die Freigabe zum Beschreiben freigeben, damit Benutzer zumindest in ihre Home-Verzeichnisse schreiben können. Der kleinste [homes]-Abschnitt hat also zwei Zeilen, z.B.:

```
[homes]
writable = yes
```

F. Wie würden Sie verhindern, dass Dateien, die in der [homes]-Freigabe erstellt werden, allgemein lesbar sind?

. Hier müssen Sie die Parameter `create mode` und `directory mode` verwenden, um die korrekte Handhabung von Dateien und Verzeichnissen zu garantieren. Fügen Sie also einfach den folgenden Eintrag in Ihre `[homes]`-Freigabe ein:

```
create mode = 0750
directory mode = 0740
```

Dies verhindert alle allgemeinen Berechtigungen.

F. Wie würden Sie sicherstellen, dass nur die Rechner A, B und C auf die Dateifreigabe [docs] zugreifen können?

. Um sicherzustellen, dass nur die Rechner A, B und C auf eine Dateifreigabe zugreifen können, fügen Sie der Freigabe einfach eine `hosts-allow`-Aussage hinzu. Denken Sie daran, dass Sie Namen oder IP-Adressen, Netzgruppen usw. benutzen können. In Ihrem Fall fügen Sie also folgendes in die Freigabe `[docs]` ein:

```
hosts allow = A B C
```

Woche 2: Es geht weiter ...

Tag 8: Drucker

von Richard Sharpe

In Kapitel 7, »Dateifreigaben«, haben Sie sich angesehen, wie Dateifreigaben konfiguriert werden. In diesem Kapitel werden Sie die Details zum Drucken über Samba kennenlernen. Sambas Philosophie zum Thema Drucken lautet: Wenn Unix es drucken kann, kann Samba es auch.

Das Drucken mit Samba umfasst folgende Bereiche:

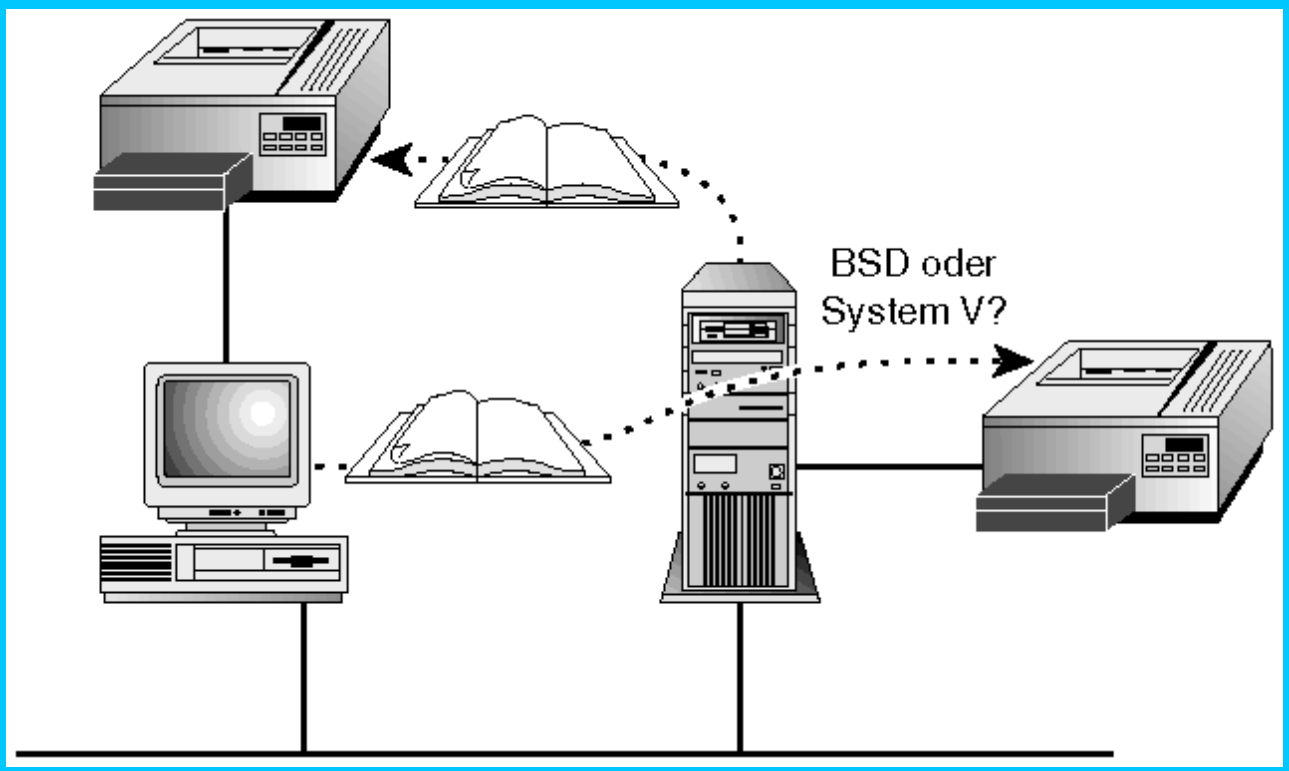
- Drucksysteme
- Konfiguration der Druckerfreigabe
- Automatische Installation des Druckertreibers
- Drucken von Unix an Windows-Systemen

Abbildung 8.1 zeigt, wie vielseitig Samba sein kann, da es das Drucken von Windows- (und DOS-)Clients an Drucker, die an Unix-Systeme angeschlossen sind, unterstützt und Unix-Systemen ermöglicht, an Drucker zu drucken, die mit Windows-Systemen verbunden sind.

Ich werde jeden dieser Bereiche darstellen und das Drucken auf Ihrem Samba-Server konfigurieren.

Wenn Sie bis jetzt Kapitel 7 noch nicht gelesen haben, sollten Sie noch einmal zurückgehen und es nachholen, da ich hier auf vielen der Konzepte aufbaue, die dort dargestellt sind.

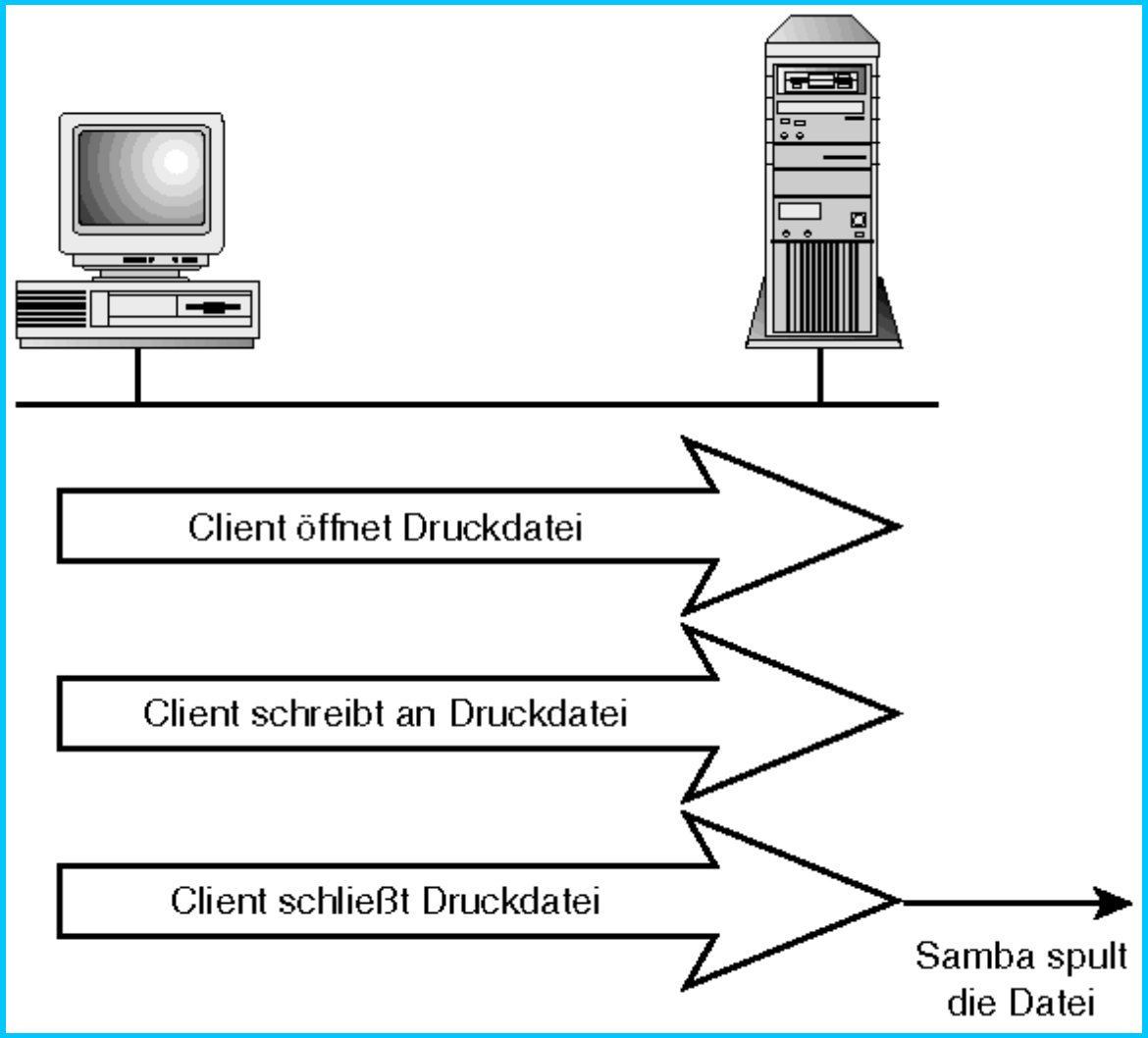
Abb. 8.1: Drucken mit Samba in einer vielseitigen Umgebung



Samba und Drucken

Aus dem Blickwinkel eines CIFS/SMB-Clients umfasst das Drucken das Öffnen einer Datei in einer speziellen Dateifreigabe, das Schreiben in diese Datei und das Schließen der Datei. Was nach Schließen der Datei mit ihr passiert, interessiert den Client nicht weiter, aber Benutzer wollen in der Regel, dass diese Datei gedruckt wird. Wenn Sie eine Druckerfreigabe unter Samba konfigurieren, kümmert Samba sich um das Drucken der Dateien. Abbildung 8.2 bietet einen detaillierten Blick auf das Drucken von einem CIFS/SMB-Client.

Abb. 8.2: Ein detaillierterer Blick auf das Drucken von einem Client



Wie Sie in Abbildung 8.2 sehen, durchläuft ein Client beim Drucken einer Datei folgende Schritte:

1. Der Client öffnet eine Datei in der Druckerfreigabe zum Beschreiben. Daher muss der Server Platz im Dateisystem zur Verfügung stellen, in dem die Datei gespeichert werden kann.
2. Der Client schreibt die Druckdatei. Er kann jegliche CIFS/SMB-Operationen benutzen. Ein bösartiger Client kann viele Daten schreiben, dann an den Anfang der Datei zurückgehen und einige Daten überschreiben.
3. Der Client schließt die Datei, und ab diesem Punkt sendet der Server den Druckauftrag an das Drucksystem (spoolt den Auftrag).

Einige Clients, wie z.B. Windows 95, öffnen Druckdateien in Druckerfreigaben mit leeren Dateinamen (d.h. ""). Lassen Sie sich hiervon nicht verwirren, wenn Sie eine `smbd`-Logdatei durchsehen.

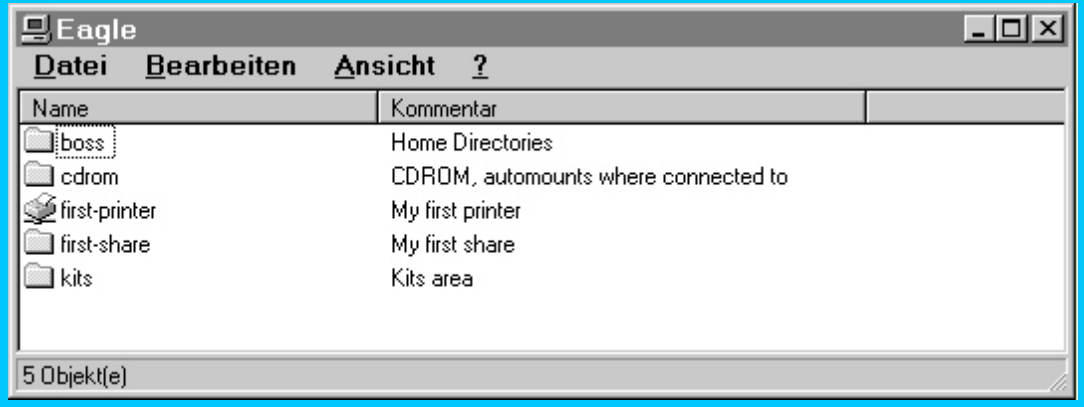
Abgesehen vom Drucken der Dateien wollen Clients oft auch den Status von Druckerwarteschlangen einsehen können. Samba unterstützt dies, indem es Informationen zur Druckerwarteschlange ausgibt, wenn ein Client diese Informationen verlangt.

Da also eine Druckerfreigabe im Wesentlichen eine Dateifreigabe mit einigen zusätzlichen Attributen ist, wissen Sie bereits das meiste, das Sie für die Erstellung einer Druckerfreigabe wissen müssen. Fast das Einzige, das Sie einer Dateifreigabe hinzufügen müssen, ist der Parameter `printable`. Hier ist ein erster Versuch für eine Druckerfreigabe:

```
[first-printer]
   comment = Mein erster Drucker
   path = /var/spool/samba
   printable = yes
```

Warum braucht eine Druckerfreigabe eine Pfadangabe? Nun, die Dateien, die der Client druckt, müssen an irgendeinen Platz im Dateisystem gespeichert werden, während sie geschrieben werden. Normalerweise wäre dieses Verzeichnis allgemein beschreibbar und hätte das Sticky Bit (t) gesetzt, damit niemand Dateien löschen kann, die ihm nicht gehören.

Abb. 8.3: Ihre erste Druckerfreigabe first-printer in der Netzwerkumgebung



Wenn Sie Samba nun neu starten, was sehen Sie? Abbildung 8.3 zeigt, was Sie in Windows 9x oder Windows NT in der Netzwerkumgebung sehen können, wenn Sie auf EAGLE doppelklicken.

Wie Sie sehen, wird Ihr Drucker angezeigt. Wenn Sie auf Ihrem Client einen Drucker konfigurieren und etwas an diese Freigabe übertragen, was passiert dann? Sie treffen auf eine Menge Probleme. Das erste ist, dass Samba annimmt, dass der mit dieser Freigabe verbundene Drucker `first-printer` heißt. Da der Drucker an diesem Punkt wahrscheinlich nicht existiert, endet die Datei, die Sie drucken wollen, in dem über den Pfadparameter definierten Verzeichnis und sitzt dort. Nachfolgend finden Sie ein detailliertes Listing des Verzeichnisses `/var/spool/samba`, nachdem eine Datei an `first-printer` übertragen wurde:

```
ls -al /var/log/samba
total 14
drwxrwxrwt   2 root    root    1024 Jan  8 12:09 .
drwxr-xr-x  14 root    root    1024 Dec 30 15:45 ..
-rwxr--r--   1 rsharpe sharpe 12240 Jan  8 11:59 rjspc1.a00652
```

Um dieses Problem zu beheben, können Sie den Parameter `printer` einfügen, um Samba mitzuteilen, welchen Drucker es benutzen soll. Um z.B. Druckaufträge an lp weiterzuleiten, benutzen Sie in der Druckerfreigabe folgenden Parameter:

```
printer = lp
```

Wenn Sie Ihrer Freigabe `first-printer` einen solchen Parameter hinzufügen und der Druckertyp der gleiche ist wie der auf Ihrem Client definierte (d.h. der Client hat den korrekten Treiber), sollten Sie nach einem Neustart von Samba den Drucker sehen können, wenn Sie an diese Warteschlange drucken.

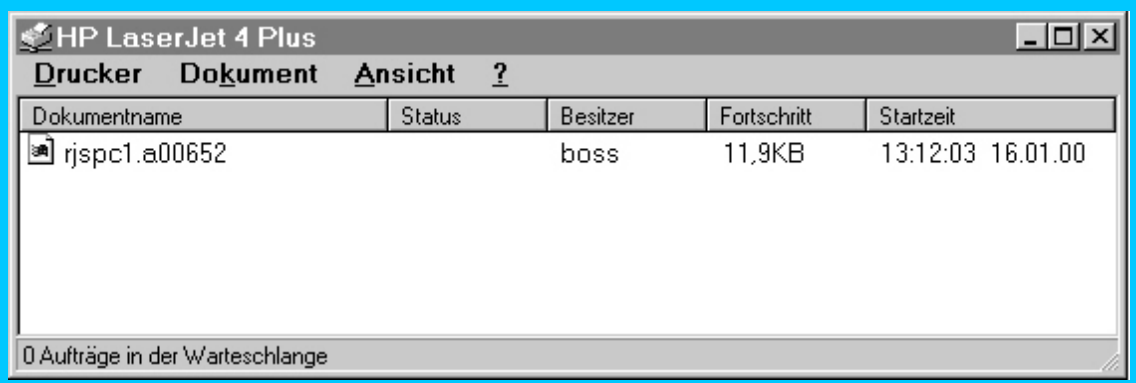
Leider wird der erste Auftrag, den Sie an den Drucker übertragen haben, nie in die Warteschlange gestellt, da Samba einen Druckauftrag nur weiterleitet, wenn er geschlossen wird. Der erste Auftrag sitzt einfach in dem durch den Pfadparameter definierten Verzeichnis, bis Sie ihn löschen.

Wenn Sie überprüfen wollen, ob das Drucken tatsächlich funktioniert, stoppen Sie die Warteschlange auf Ihrem Samba-Server, bei einem BSD-basierten Drucksystem z.B. über den Befehl:

```
lpc stop lp
```

Überprüfen Sie dann den Drucker von Ihrem Client aus. Unter Windows 9x oder Windows NT sollten Sie etwas Ähnliches sehen wie in Abbildung 8.4.

Abb. 8.4: Warteschlangeninformationen für Ihre Druckerfreigabe



Denken Sie daran, die `lp`-Warteschlange neu zu starten, damit zukünftige Druckaufträge gedruckt werden.

Das Drucken kann so einfach, aber auch eine sehr komplexe Aufgabe sein. Möglicherweise müssen Sie sich Gedanken machen über das von

Ihrem Samba-Server benutzte Drucksystem, über Druckertreiber, die PostScript mit der berüchtigten vorangestellten Zeichenkette `[Strg]-[D]` (^D) generieren, und eine Menge anderer Probleme. Viele dieser Probleme werden in den folgenden Abschnitten dargestellt.

Unterstützte Drucksysteme

Samba benutzt die Druckbefehle des Betriebssystems, um eine Datei zu drucken, die an eine Druckerfreigabe übertragen wurde. Es verwendet außerdem die anderen Befehle, die das Drucksystem bietet, um den Status von Druckerwarteschlangen und Druckaufträgen abzufragen, Druckerwarteschlangen anzuhalten, neu zu starten usw.

Es gibt jedoch viele verschiedene Drucksysteme unter Unix. Abgesehen von den ursprünglichen BSD und System V, die unterschiedliche Befehle zum Ausführen druckrelevanter Aufgaben verwenden, gibt es auch *PLP (Portable Line Printer)* und *LPRNG*, die beide auf dem BSD-Ansatz mit einigen Verbesserungen basieren. PLP und LPRNG wurden vom gleichen Autor, Patrick Powell, entwickelt.

Zusätzlich zu diesen Drucksystemen haben die beliebten Unix-Varianten AIX und HPUX ihre eigenen Drucksysteme. Außerdem unterstützt Samba das Drucken für QNX und SOFTQ.

Wenn Samba kompiliert wird, richtet es das Standarddrucksystem ein, indem es folgendermaßen nach Makros sucht (siehe `$SRCDIR/include/includes.h`):

1. Bei einem AIX-System wird das Drucksystem auf AIX eingestellt.
2. Bei einem HPUX-System wird das Drucksystem auf HPUX eingestellt.
3. Bei einem QNX-System wird das Drucksystem auf QNX eingestellt.
4. Bei einem System-V-System wird das Drucksystem auf SYSV eingestellt.
5. Sonst wird das Drucksystem auf BSD eingestellt.

Dies funktioniert für die meisten Systeme, muss aber geändert werden, wenn Ihr System nicht AIX, HPUX, QNX oder System V ist, aber das System-V-Drucksystem benutzt. Es muss auch geändert werden, wenn Sie PLP, LPRNG oder SOFTQ verwenden.

Wenn Sie das Standarddrucksystem tatsächlich ändern müssen, fügen Sie einfach folgenden Eintrag in den globalen Abschnitt ein:

```
printing = <Ihre Drucksystem-Auswahl>
```

Ist Ihr System wirklich außergewöhnlich und entspricht es keinem der Drucksysteme, müssen Sie möglicherweise individuelle Druckbefehle einrichten, damit das Drucken korrekt funktioniert.

Die [printers]-Freigabe

Wenn Sie jedesmal eine neue Druckerfreigabe einrichten müssten, wenn Sie Ihrem Samba-Server einen neuen Drucker hinzufügen wollen, könnte die Administration eines Samba-Servers kompliziert werden. Um Ihr Leben zu vereinfachen, bietet Samba ein Schema für Drucker, das der in Kapitel 7 beschriebenen `[homes]`-Freigabe ähnlich ist. Kurz: Wenn ein Client eine Verbindung zu einer Freigabe verlangt, folgt Samba diesem Ansatz:

1. Die Freigabe wird in der `smb.conf` gesucht und, falls vorhanden, zurückgegeben.
2. Wird die Freigabe nicht gefunden, gibt es aber einen `[homes]`-Abschnitt, sucht Samba in der `passwd`-Datei einen Benutzer mit dem gleichen Namen wie dem der verlangten Freigabe. Wird dieser gefunden, wird er als Freigabe zurückgegeben.
3. Ist auch dies nicht erfolgreich, existiert aber ein `[printers]`-Abschnitt, sucht Samba nach einem Drucker mit dem gleichen Namen wie die verlangte Freigabe und gibt dies als Freigabe zurück.
4. Findet Samba auch hier die Freigabe nicht, sucht es nach einer Standardfreigabe und gibt diese, falls vorhanden, zurück.
5. Ist auch dies nicht erfolgreich, gibt Samba eine Fehlermeldung aus, die besagt, dass der Netzwerkname nicht verfügbar ist.

Die `[printers]`-Freigabe sieht aus wie jede andere Druckerfreigabe. Tatsächlich können Sie diese zu einer `[printers]`-Freigabe machen:

```
[printers]
   comment = Alle Drucker in dieser Freigabe, aus printcap
   path = /var/spool/samba
   printable = yes
```

Wenn Samba die `[printers]`-Freigabe benutzt, weil die verlangte Freigabe nicht als Abschnitt existiert und nicht als `[homes]`-Freigabe aufgelöst werden kann, wird ein Klon der `[printers]`-Freigabe erzeugt, dem der Name der verlangten Freigabe und des verlangten Druckers gegeben wird. Das heißt, dass alle Drucker, die durch den `[printers]`-Abschnitt definiert werden, ihre Parameter aus dem `[printers]`-Abschnitt holen.

Woher bekommt Samba die Liste der Drucker? Aus der `printcap`-Datei. Basiert Ihr Drucksystem auf BSD oder benutzt es PLP oder LPRNG, finden Sie die `printcap`-Datei unter `/etc/printcap`. Benutzt Ihr System dagegen das System-V-Drucksystem, können Sie eine `printcap`-Datei erstellen oder den Parameter `printcap name` verwenden, der später in diesem Kapitel dargestellt wird.

Druckrelevante Parameter

Die folgenden Parameter beeinflussen auf die eine oder andere Weise die Funktionsweise der Druckerfreigaben. Die meisten Samba-Administratoren verwenden nicht viele dieser Parameter. Wie immer finden Sie die komplette Auflistung der Parameter und das letzte Wort zu ihren Funktionen in den Manpages zur `smb.conf` für die aktuellste Samba-Version. Sie können den Befehl `man smb.conf` verwenden, um einen Blick auf die Parameter zu werfen.

Viele der Parameter, die in den nächsten Abschnitten aufgelistet sind, nehmen Variablen wie `%p`, `%j` usw. an, die bei Ausführung der Befehle durch die druckrelevanten Informationen ersetzt werden. Tabelle 8.1 zeigt die Bedeutung vieler dieser Variablen.

Tabelle 8.1: Ersetzungen für Druckervariablen

Variable	Beschreibung
`%p`	Ersetzen durch Druckernamen
`%j`	Ersetzen durch Auftragsnummer
`%s`	Ersetzen durch vollen Pfadnamen für Spool-Datei.
`%f`	Ersetzen durch Namen der Spool-Datei (ohne Pfad)

load printers

Dieser globale Parameter definiert, ob Samba alle Drucker zum Browsen in die `printcap`-Datei lädt.

Der Standardwert für diesen Parameter ist `yes`, d.h. standardmäßig sind alle Drucker in Ihrer `printcap`-Datei für das Browsing verfügbar. Wenn Sie dies nicht wollen, fügen Sie einfach folgenden Eintrag in den globalen Abschnitt Ihrer `smb.conf` ein:

```
load printers = no
```

lppause command

Dieser Parameter definiert den Befehl, den Samba ausführt, um das Drucken eines bestimmten Druckauftrags anzuhalten. Dies sollte ein Befehl oder ein Skript sein, der bzw. das einen Warteschlangennamen und eine Jobnummer annimmt und den Auftrag anhält.

Dieser Parameter hat nur bei den Drucksystemen SysV und SOFTQ einen Standardwert.

Detaillierte Informationen finden Sie in den Manpages zur `smb.conf`.

lpq cache time

Dieser globale Parameter kontrolliert, wie lange `lpq`-Informationen zwischengespeichert werden. Er verhindert, dass `lpq command` zu oft aufgerufen wird. Der Wert wird in Sekunden ausgedrückt.

Der Standardwert für diesen Parameter ist 10 Sekunden.

lpq command

Dieser Parameter definiert den Befehl, den Samba ausführt, um Statusinformationen zur Druckerwarteschlange für Clients zu erhalten. Dies sollte ein Programm oder ein Skript sein, das einen Warteschlangennamen annimmt und Statusinformationen über den Drucker ausgibt.

Der Standardwert für diesen Parameter hängt vom Wert des Parameters `printing` ab.

lpresume command

Dieser Parameter definiert den Befehl, den Samba ausführt, um einen Druckauftrag für Clients neu zu starten oder fortzusetzen. Dies sollte ein Programm oder ein Skript sein, das einen Druckernamen und eine wieder aufzunehmende Auftragsnummer annimmt. Der Parameter bewirkt das Gegenteil von `lppause command`.

Dieser Parameter hat nur für die Drucksysteme SysV oder SOFTQ einen Standardwert.

Detaillierte Informationen finden Sie in den Manpages zur `smb.conf`.

lprm command

Dieser Parameter definiert den Befehl, den Samba ausführt, um einen Druckauftrag für Clients zu löschen. Dies sollte ein Programm oder ein

Skript sein, das einen Druckernamen und eine zu löschende Auftragsnummer annimmt.

Der Standardwert für diesen Parameter hängt vom Wert des Parameters `printing` ab.

Ein Beispiel:
```
lprm command = /usr/bin/lprm -P%p %j
```

Mit dieser Einstellung benutzt `lprm command /usr/bin/lprm` und erhält einen Warteschlangennamen und die Auftragsnummer.

min print space

Dieser Parameter definiert den mindestens auf der Festplatte zur Verfügung stehenden Speicherplatz, damit Clients den Druckauftrag durchführen können. Er wird in Kilobyte spezifiziert. Ein Wert von `0` (der Standardwert) heißt, dass Aufträge immer durchgeführt werden, unabhängig von freiem Platz auf der Festplatte.

postscript

Dieser Parameter legt fest, dass Samba Druckdateien als PostScript interpretieren soll. Samba fügt dann einen PostScript-Kommentar (`%!`) an den Anfang des Druckauftrags ein, womit Probleme mit PCs behoben werden, die darauf bestehen, die Zeichenkette `[Strg]+[D]` an den Anfang der Druckdaten zu setzen. Dies verwirrt PostScript-Drucker.

Der Standardwert für diesen Parameter ist `false` oder `no`.

print command

Dieser Parameter definiert den Befehl, den Samba ausführt, um einen Druckauftrag auszuführen. Dies sollte ein Programm oder ein Skript sein, das einen Druckernamen und einen Dateinamen annimmt und die Datei an den Drucker weiterleitet.

Der Befehl `print` muss mindestens eine der Variablen `%s` oder `%f` und kann die Variable `%p` enthalten.

Der Standardwert für diesen Parameter hängt vom Wert des Parameters `printing` ab.

Ein Beispiel:
```
print command = /usr/local/samba/bin/localprintscrip %p %s
```

Mit dieser Einstellung wird ein lokales Skript namens `localprintscript` aufgerufen, dem der Warteschlangenname und die Auftragsnummer übergeben wird.

printable

Dieser Parameter definiert, dass eine Freigabe eine Druckerfreigabe ist, über die ein Client Druckdaten an das in der Freigabe definierte Verzeichnis übertragen kann. Wenn eine Freigabe als `printable` bestimmt ist, ist `sie` standardmäßig auch `writable`. Jeglicher `read-only`-Parameter gilt nur für nicht druckenden Zugriff auf die Freigabe.

Der Standardwert für diesen Parameter ist `no`; standardmäßig sind Freigaben also nicht `printable`. Das heißt, Clients können sich auf diesem Weg nicht als Druckerfreigaben verbinden.

Um eine Freigabe für das Drucken einzurichten, fügen Sie einfach folgenden Eintrag in den Abschnitt für die Freigabe ein:
```
printable = yes
```

printcap name

Dieser Parameter (und sein Synonym `printcap)` wird benutzt, um Samba den Standort der `printcap-Datei` mitzuteilen, in der nach Druckern gesucht wird, wenn die `[printers]`-Freigabe verwendet wird.

In System-V-Systemen, die für eine Auflistung verfügbarer Drucker `lpstat` verwenden, können Sie den Parameter `printcap name` auf `lpstat` setzen, um automatisch eine Liste aller verfügbaren Drucker zu erhalten.

Der Standardwert für diesen Parameter ist `/etc/printcap`.

Ein Beispiel:
```
printcap name = /etc/myprintcap
```

Mit dieser Einstellung benutzt Samba die Datei `/etc/myprintcap`, wenn es nach Druckern sucht.

printer

Dieser Parameter teilt Samba den Namen des Druckers mit, an den Druckaufträge übertragen werden, wenn der Client die Druckdatei

geschlossen hat.

Dieser Parameter hat keinen Standardwert. Einige Beispiele:

```
printer = lp
printer = hplj4
```

printer driver

Dieser Parameter definiert, welchen Treibernamen Samba an Clients übergibt, die nach dem mit einem Drucker verbundenen Treiber fragen. Er wird mit der automatischen Installation von Druckertreibern verwendet, die später in diesem Kapitel dargestellt wird.

Dieser Parameter hat keinen Standardwert. Ein Beispiel:

```
printer driver = HP LaserJet 4 Plus
```

printer driver files

Dieser Parameter teilt Samba den Standort der Druckertreiber-Datei mit, die benutzt wird, wenn Treiber an Windows-9x-Clients übertragen werden.

Die Datei wird aus einer Windows-9x-`msprint.def`-Datei erstellt, wie es im Abschnitt »Automatische Treiberinstallation« beschreiben wird.

Der Standardwert für diesen Parameter ist `SAMBA_INSTALL_DIRECTORY/lib/printers.def`.

printer driver location

Dieser Parameter teilt Samba mit, welche Freigabe es übergeben soll, wenn Clients bei der automatischen Installation von Druckertreibern auf Windows-9x-Rechnern nach dem Standort der Treiberdateien fragen. Dies wird ausführlicher im Abschnitt »Automatische Treiberinstallation« dargestellt.

Dieser Parameter hat keinen Standardwert. Ein Beispiel:

```
printer driver location = \\%h\printers$
```

printing

Dieser Parameter teilt Samba das Drucksystem mit, das auf Ihrem Server verwendet wird. In der Regel wird das Drucksystem während der Kompilierung bestimmt, aber wenn Ihr System PLP, LPRNG oder SOFTQ benutzt oder sehr außergewöhnlich ist (z.B. auf SysV basierend, aber LPD benutzend), müssen Sie diesen Parameter manuell einstellen.

Derzeit werden acht Drucksysteme unterstützt:

- AIX
- BSD
- HPUX
- LPRNG
- PLP
- QNX
- SOFTQ
- SYSV

Der Standardwert für diesen Parameter wird, wie oben angegeben, während der Kompilierung festgelegt.

queuepause command

Dieser Parameter definiert den Befehl, den Samba ausführt, um eine Druckerwarteschlange anzuhalten. Dies sollte ein Programm oder ein Skript sein, das einen Druckernamen annimmt und die Warteschlange für diesen Drucker stoppt.

Der Standardwert für diesen Parameter hängt vom verwendeten Drucksystem ab.

queueresume command

Dieser Parameter teilt Samba mit, welchen Befehl es benutzen soll, um Druckerwarteschlangen wieder aufzunehmen. Dies sollte ein Programm oder ein Skript sein, das einen Druckernamen annimmt und die Warteschlange für diesen Drucker wieder aufnimmt.

Der Standardwert für diesen Parameter hängt vom verwendeten Drucksystem ab.

Automatische Treiberinstallation

Windows 95 und Windows 98 unterstützen die automatische Installation von Druckertreibern über *Point and Print*. Samba implementiert die erforderliche Funktionalität, um diese automatische Installation zu unterstützen.

Obwohl die benötigten Einstellungen für die Unterstützung der Point-and-Print-Installation ausführlich im Samba-Dokumentationsverzeichnis beschrieben ist (`PRINTER_DRIVER.txt`), werde ich sie hier detailliert darstellen.

Zunächst müssen Sie eine `[printers$]`-Freigabe einrichten, in der alle Treiberdateien abgelegt werden. Diese Dateifreigabe sieht wie folgt aus:

```
[printer$]
   path = /usr/local/samba/printer
   public = yes
   writable = no
   browsable = yes
```

Denken Sie daran, dieses Verzeichnis zu erstellen, damit die Dateifreigabe die Dateien dort ablegen kann.

Der nächste Schritt besteht darin, die Druckerdefinitionsdatei zu erstellen, damit Windows 9x weiß, wie es die Drucker installieren soll, die Sie für die automatische Installation verfügbar gemacht haben. Dafür müssen Sie sich die Windows INF-Dateien `msprint.inf` und `msprint2.inf` aus dem Verzeichnis `C:\WINDOWS\INF` von Ihrem Windows-Rechner holen. Manchmal befinden sich diese Dateien auch in einem anderen Verzeichnis. Wenn Sie nicht unterstützte oder aktualisierte Treiber verwenden, müssen Sie diese Treiber erst auf Ihrem Windows-9x-System installieren, dann die Datei `oemNN.inf` kopieren und `statt msprint.inf` diese Datei benutzen.

Die Datei heißt nicht genau `oemNN.inf`, sondern hat einen ähnlichen Namen. Sie können die von Ihnen benötigte Datei finden, indem Sie in jeder derartigen Datei nach dem entsprechenden Druckernamen suchen.

Wenn Sie die Dateien auf Ihren Samba-Server kopiert haben, müssen Sie das in Samba integrierte Programm `make_printerdef` benutzen, um Ihren Drucker in die Datei `printer.def` einzutragen. Suchen Sie den exakten Namen für den Drucker, den Sie definieren (d.h. den Namen, unter dem er Windows bekannt ist), indem Sie die entsprechende INF-Datei durchsuchen. Für Drucker, deren Namen mit den Buchstaben A bis K beginnen, suchen Sie in `msprint.inf`, für andere Namen in `msprint2.inf`. Für das folgende Beispiel verwenden Sie einen Drucker des Typs HP LaserJet 4 Plus und erstellen so einen neuen `printers.def`-Eintrag:

```
make_printerdef msprint.inf "HP LaserJet 4 Plus">> printers.def
```

Stellen Sie sicher, dass der neue Eintrag an das Ende der `printers.def`-Datei eingtragen wird. In diesen Beispielen platzieren Sie die Datei unter `/usr/local/samba/lib`.

Wenn `make_printerdef` ausgeführt wird, gibt es auf `stderr` die Dateien aus, die für die Installation benötigt werden. All diese Dateien müssen in die `[printer$]`-Freigabe kopiert werden, die Sie vorher definiert haben. Die Dateien befinden sich in der Regel alle im Verzeichnis `C:\WINDOWS\SYSTEM`. Für den HP LaserJet 4 Plus werden folgende Dateien benötigt:

```
FINSTALL.DLL, FINSTALL.HLP, HPPCL5MS.DRV, ICONLIB.DLL, PJLMON.DLL, UNIDRV.DLL, UNIDRV.HLP
```

Zum Abschluss müssen Sie Ihrer `smb.conf` noch einige zusätzliche Parameter hinzufügen. Einer geht in den globalen Abschnitt und spezifiziert den Standort der Druckerdefinitionsdatei:

```
[global]
   ...
   printer driver file = /usr/local/samba/lib/printers.def
   ...
```

Dies ist die Datei, in die Sie alle Druckerdefinitionseinträge platzieren, die mit dem Programm `make_printerdef` erstellt werden.

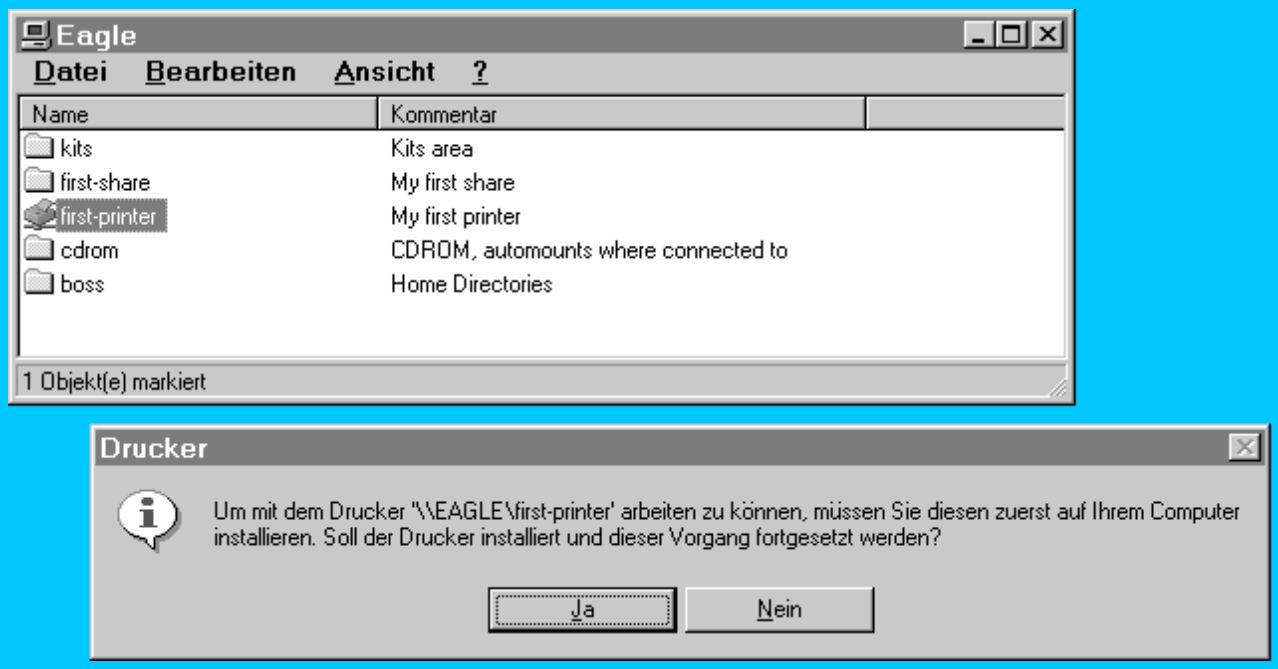
Die anderen Parameter, `printer driver` und `printer driver location`, müssen Sie für jede Druckerfreigabe definieren, für die Sie automatische Treiberinstallation ermöglichen wollen. Das folgende Beispiel zeigt Ihre Freigabe `first-printer` nach den Änderungen zur Unterstützung automatischer Installation:

```
[first-printer]
   comment = Mein erster Drucker
   path = /var/spool/samba
   printable = yes
   printer driver = HP LaserJet 4 Plus
   printer driver location = \\%h\PRINTER$
```

Machen Sie sich keine Gedanken über das `%h` in der letzten Zeile. Es ist eine der Variablen, die Samba in der `smb.conf`-Datei benutzen kann. Diese werden in Kapitel 10, »Automatisierung auf Server-Seite«, ausführlicher dargestellt.

Wenn Sie alle Änderungen an Ihrer `smb.conf` durchgeführt und Samba neu gestartet haben, können Sie die automatische Installation des Treibers für den definierten Drucker ausprobieren. Rufen Sie Ihren Server in der Netzwerkumgebung auf, wie in Abbildung 8.5 dargestellt. Doppelklicken Sie dann auf `first-printer`. Dies sollte die Installation des Druckers starten. Sie werden folgendes Fenster sehen.

Abb. 8.5: Ihr Drucker wird installiert



Klicken Sie auf *Ja*, um die Einrichtung Ihres Druckers fortzusetzen. Dies führt Sie zum Assistenten für die *Druckerinstallation*. Wählen Sie *Weiter* und Sie sehen das Fenster in Abbildung 8.6.

Abb. 8.6: Der Assistent für die Druckerinstallation weiß, dass er es mit mit einem HP LaserJet 4 Plus zu tun hat

Hier sehen Sie, dass der Assistent für die Druckerinstallation weiß, dass er es mit einem HP LaserJet 4 Plus zu tun hat. Das liegt daran, dass Samba ihm dies mitgeteilt hat. Wenn Sie die Druckerinstallation beenden und eine Testseite ausdrucken, sollten Sie sehen, dass Ihr Drucker die Testseite druckt.

Die automatische Treiberinstallation kann bei mittelgroßen oder großen Netzwerken enorm viel Zeit sparen, da es die Installation neuer Drucker für Windows-9x-Systeme in Windeseile erledigt.

Von Unix zu Windows drucken

Manchmal wollen Sie vielleicht von Ihren Unix-Servern auf einem Drucker drucken, der mit einem Windows-Client verbunden ist. Um dies einzurichten, müssen Sie einen entsprechenden Druckerfilter für das Drucksystem unter Unix konfigurieren.

In diesem Abschnitt wird dies für das Drucken auf Berkeley-artigen Systemen (LPD, PLP und LPRNG) und System-V-artigen Systemen dargestellt.

Samba enthält ein kleines Shell-Skript namens `smbprint` (oder `smbprint.sysv` für System-V-Systeme). Sie können dieses Programm benutzen, um Druckaufträge zu akzeptieren und sie an freigegebene Drucker auf entfernten Windows-Systemen weiterzuleiten. Dies wird auf unterschiedliche Art und Weise durchgeführt, abhängig davon, ob Ihr Unix-System ein BSD-artiges oder ein System-V-artiges Drucksystem verwendet.

In beiden Fällen muss der Drucker auf dem Windows-Rechner freigegeben sein. Dies können Sie tun, indem Sie den Befehl *Freigabe* aus dem Kontextmenü wählen, das sich öffnet, wenn Sie in Windows 9x oder Windows NT auf einen Drucker mit der rechten Maustaste klicken. Weitere Informationen finden Sie in Kapitel 14, »Windows 9x und Windows NT«.

Das Skript `smbprint` (oder `smbprint.sysv`) benutzt `smbclient`, um die Druckdaten von Unix an Windows (mit Zeilenende-Übersetzung, wenn notwendig) auf eine Druckerfreigabe zu kopieren. Das `smbclient`-Utility wird in Kapitel 13, »Unix (`smbclient`, `smbfs`, `smbwrapper` und andere Utilities)«, beschrieben.

Über ein BSD-artiges Drucksystem an Windows drucken

Das BSD-artige Drucksystem benutzt LPD und wird von einer Datei namens `/etc/printcap` kontrolliert. Alle Drucker werden in der Datei `printcap` definiert. Folgende Schritte führen dazu, dass ein bestimmter Drucker seine Druckaufträge an einen freigegebenen Drucker auf einem entfernten Windows-9x-System überträgt:

1. Definieren Sie den Drucker, d.h. bauen Sie entweder manuell oder über Ihr bevorzugtes Tool einen `printcap`-Eintrag auf.
2. Erstellen Sie eine Datei namens `.config` (stellen Sie sicher, dass der Name mit einem Punkt beginnt) im Spool-Verzeichnis (in der Regel `/var/spool/<Druckername>`) und fügen Sie der Datei folgende Zeilen hinzu:

   ```
   server = <Server-Name>
   service = <Name der Druckerfreigabe>
   password = <Passwort>
   ```

3. Ändern Sie in der `printcap`-Datei den Eingabefilter für Ihren Drucker auf (oder fügen Sie ihn hinzu)

   ```
   :if=<Verzeichnispfad zu smbprint>/smbprint:\
   ```

4. Starten Sie den Drucker neu.

Ist der Name des lokalen Druckers z.B. `hawk_print`, des entfernten Servers `HAWK`, der Druckerfreigabe `MY_PRINTER` und wird für den Zugriff auf die Freigabe kein Passwort verlangt, würde Ihre `.config`-Datei wie folgt aussehen:

```
server=HAWK
service=MY_PRINTER
password=""
```

Auf einem RedHat-Linux-System würden Sie die Datei im Verzeichnis `/var/spool/lpd/hawk_print` erstellen, und der Eintrag in der `printcap`-Datei würde wie folgt aussehen:

```
:if=/usr/bin/smbprint:\
```

Normalerweise müssen Sie diese Schritte nicht für Drucker ausführen, die sich auf Windows-NT-Systemen befinden, da Sie auf Windows NT LPD installieren können. Weitere Informationen über die Konfiguration entfernter LPD-Drucker finden Sie in der `printcap`-Manpage.

Über ein System-V-artiges Drucksystem an Windows drucken

Im Gegensatz zum BSD-artigen Drucksystem benutzt das System-V-artige Drucksystem `lp`. Samba verwendet für das Drucken an Windows-Rechner von System-V-Systemen das Skript `smbprint.sysv`, das sich im Verzeichnis `examples/printing` unterhalb des Samba-Source-Verzeichnisses befindet.

Dieses Skript ist eine modifizierte Version des BSD-Skripts. Um es zu benutzen, ändern Sie das Skript und spezifizieren den Windows-Server, die Freigabe und das Passwort. Diese befindet sich in einem Block:

```
server=admin
service=hplj2
password=""
```

Ändern Sie jeden dieser Einträge auf den entsprechenden Wert, installieren Sie dann das Skript als Interface-Skript für Ihre Warteschlange und beginnen Sie folgendermaßen mit dem Druck:

```
lpadmin -punixprintername -v/dev/null -i./smbprint.sysv
enable unixprintername
accept unixprintername
```

Ist der Name des lokalen Druckers z.B. `hawk_print`, des entfernten Servers `HAWK`, der Druckerfreigabe `MY_PRINTER` und wird für den Zugriff auf die Freigabe kein Passwort verlangt, würde Ihre `smbprint.sysv`-Datei wie folgt aussehen:

```
server=HAWK
service=MY_PRINTER
password=""
```

Mit folgenden Befehlen können Sie diesen Drucker einrichten und aktivieren:

```
lpadmin -phawk_print -v /dev/null -i/smbprint.sysv
enable hawk_print
accept hawk_print
```

Normalerweise ist dies für Windows-NT-Systeme nicht notwendig, da Windows NT LPD unterstützt und `lpadmin` einen System-V-basierten Drucker konfigurieren kann. Weitere Informationen finden Sie in der `lpadmin`-Manpage.

Zusammenfassung

In diesem Kapitel haben Sie etwas über das Drucken mit Samba erfahren. Sie haben sich angesehen, wie Sie Druckerfreigaben einrichten, und viele Parameter kennen gelernt, die Druckerfreigaben kontrollieren. Außerdem wurden die automatische Treiberinstallation und das Drucken von Unix an Windows-Rechner dargestellt.

Dabei haben Sie einen detaillierten Blick auf viele der für das Drucken relevanten Prozesse und der von Samba unterstützten Drucksysteme geworfen. Diese Informationen sollten es Ihnen ermöglichen, viele der Probleme bei der Konfiguration der Samba-Drucker zu beheben.

Im nächsten Kapitel werden Sie einige der GUI-Administrationstools kennen lernen, die für Samba verfügbar sind.

Frage & Antwort

F. Meine Benutzer drucken an einen PostScript-Drucker, aber es wird das PostScript statt der aktuellen Seite gedruckt. Wie kann ich das korrigieren?

. Viele Unix-Systeme haben einen PostScript-Konverter, der Textdateien in PostScript konvertiert (z.B. `enscript`). Der Filter ist intelligent genug, PostScript in Ruhe zu lassen, so dass es korrekt druckt. Einige Windows-Druckertreiber aber platzieren ein `[Strg]+[D]` (`^D`) an den Anfang der Datei. Dies verwirrt den Filter, der normalerweise nach `%!` sucht, das PostScript in dem Druckauftrag wird in PostScript konvertiert, und Sie erhalten sehr viele nicht brauchbare Seiten. Sie können dieses Problem vermeiden, wenn Sie den Parameter `postscript = yes` in die Druckerfreigabe einfügen.

F. Wir drucken von unseren Windows-Clients an Samba-Druckerfreigaben, die dann über externe oder interne Printserver (wie z.B. JetDirect- oder MarkNet-Karten) an HP LaserJet oder Lexmark-Drucker weitergeleitet werden. Einige Druckaufträge werden nicht korrekt ausgeführt, besonders solche, die Grafiken enthalten. Wie können wir dies beheben?

. Das hat wahrscheinlich damit zu tun, wie diese Art von Netzwerkdruckern Sequenzen am Ende einer Zeile behandeln. Beide erwähnte Typen bieten eine `raw`-Warteschlange (`RAW` bei HP, `printer` bei Lexmark), die die übertragenen Daten ganz einfach nimmt und ausdruckt. Beide bieten jedoch auch eine Warteschlange, die Standard-Unix-Textdateien in etwas konvertiert, mit dem die Drucker

umgehen können. Diese Warteschlangen (`TEXT` bei HP und `printer_cr` bei Lexmark) fügen immer dann einen Wagenrücklauf ein, wenn Sie eine Zeilenendemarke sehen. Dies sorgt dafür, dass Unix-Textdateien korrekt gedruckt werden. Wenn Sie dagegen eine Binärdatei von einem PC übertragen, ist die Wahrscheinlichkeit hoch, dass Sie die Binärdatei zerstören.

Sie sollten zwei Warteschlangen in Ihrer `printcap`-Datei zur Verfügung stellen. Eine, an die PCs drucken können (vielleicht über den Parameter `printer` in `smb.conf` definiert) und eine, an die Unix-Benutzer und -Programme drucken können. Samba sollte an die `raw`-Warteschlange drucken und Unix an die textbasierte.

F. Wir haben einen System-V-Rechner. Wie können wir Samba über alle Druckerwarteschlangen informieren, damit unsere Clients sie beim Browsing sehen können?

. Sie sollten folgenden Parameter in den globalen Abschnitt Ihrer `smb.conf` einfügen:

```
printcap name = lpstat
```

Tag 9: GUI-Administrationstools

von Richard Sharpe

In den letzten vier Kapiteln haben Sie einen eingehenden Blick auf die Konfiguration von Samba geworfen. Dies geschah jedoch in Form einer vertieften Darstellung vieler der Konfigurationsparameter, die in Ihrer `smb.conf` definiert werden können, und setzte voraus, dass Sie diese Datei über Ihren bevorzugten Editor bearbeiten können.

Diejenigen, die sich in der Samba-Konfiguration sehr gut auskennen, finden ohne Zweifel, dass die direkte Bearbeitung der `smb.conf` der schnellste Weg ist, um neue Freigaben hinzuzufügen oder Einstellungen zu ändern. Für viele andere Leute aber wäre ein einfaches GUI-Interface[1] für die `smb.conf`-Datei ein Segen. Eine solche Funktion ist sogar noch nützlicher, wenn Sie nur eine einfache Änderung durchführen wollen und nicht direkt in Ihren Samba-Server eingeloggt sind.

In diesem Kapitel werden Sie sich die folgenden GUI-Konfigurationstools für Samba mehr oder weniger detailliert ansehen:

- Das Samba Web Administration Tool, SWAT
- SMBedit, ein Windows-9x-basiertes Administrationstool
- Webmin, ein weiteres Web-basiertes Administrationstool
- `smbconftool`, ein Java-basiertes Administrationstool
- `smb-mode.el`, ein Emacs-Modus für die Bearbeitung der Datei `smb.conf`

Diese Tools können grob in drei Gruppen eingeteilt werden:

- Tools, die CGI-Skripte verwenden und daher von einem Browser auf jeder beliebigen Plattform benutzt werden können, darunter Windows-, Unix-, VMS-Rechner usw.
- Tools, die Windows-Anwendungen sind und daher nur auf Windows-Systemen funktionieren
- Tools, die Unix-Anwendungen sind oder auf dem Server laufen müssen, auf dem sich die Datei `smb.conf` befindet, und daher nur auf einem Unix-Rechner laufen können

In den folgenden Abschnitten werden Sie sich die Installation und Verwendung der meisten dieser Tools sowie ihre jeweiligen Vor- und Nachteile ansehen.

SWAT

Das *Samba Web Administration Tool (SWAT)* ist eine neue Funktion, die in Samba 2.0.0 integriert ist. Es ist ein Mini-Web-Server und eine CGI-Skripting-Anwendung, die dazu entwickelt wurde, über den `inetd` zu laufen und Zugriff auf die Datei `smb.conf` auf dem System zu bieten, auf dem SWAT läuft. Der `inetd` ist der Daemon, der das Starten der meisten Netzwerkdienste unter Unix handhabt und durch die Datei `/etc/inetd.conf` gesteuert wird. (Weitere Informationen zu `inetd` finden Sie über den Befehl `man inetd`.)

SWAT ermöglicht einer entsprechend autorisierten Person (mit dem `root`-Passwort) die Konfiguration aller Samba-Funktionen über Webseiten. SWAT bietet außerdem Hilfe-Links zu allen konfigurierbaren `smb.conf`-Optionen auf jede Seite, so dass Admininstratoren die Auswirkung jeder Änderung leicht verstehen können.

SWAT wird unter Samba 2.0.0 standardmäßig aufgebaut und installiert, aber je nach System und Installationsmethode müssen Sie möglicherweise einige zusätzliche Optionen konfigurieren, um SWAT benutzen zu können.

Wenn Sie Samba über RPM auf einem Linux-System installieren, übernimmt RPM alle notwendigen Konfigurationsaufgaben (inklusive der später erwähnten Änderungen). Wenn Sie aber manuelle Methoden verwenden, müssen Sie die folgenden Schritte ausführen:

1. Samba konfigurieren:

   `configure` bzw. `./configure`

2. Samba übersetzen:

   ```
   make
   ```

3. Samba installieren:

   ```
   make install
   ```

4. Der `/etc/services` eine Zeile wie die folgende hinzufügen:

   ```
   swat    901/tcp
   ```

Wenn Sie NIS benutzen, müssen Sie wahrscheinlich Ihre NIS-Service-Maps neu aufbauen.

5. `/etc/inetd.conf` eine Zeile wie die folgende hinzufügen:

   ```
   swat    stream    tcp    nowait.400    root /usr/local/samba/bin/swat swat
   ```

Wenn Sie Ihre Samba-Binärdateien an einem anderen Ort installiert haben, müssen Sie das Verzeichnis entsprechend ändern.

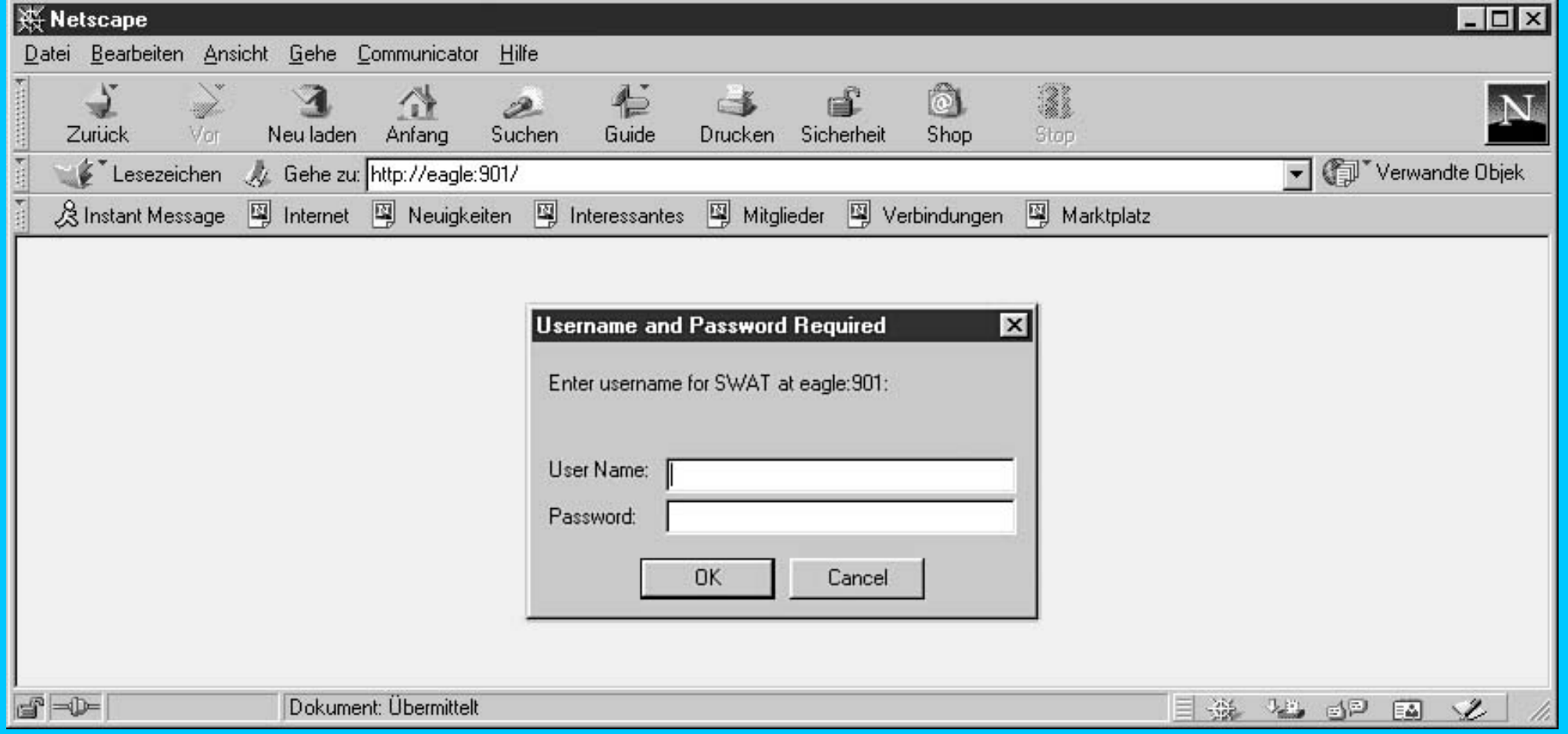
6. Haben Sie die Schritte 1 bis 5 absolviert, können Sie den `inetd` neu starten, indem Sie ein HUP-Signal übertragen. Dann liest der `inetd` seine `conf`-Datei erneut, und SWAT ist für den Einsatz bereit. Dafür können Sie verschiedene Methoden benutzen. Die einfachste ist `kill -HUP` *`PID`*, wobei *`PID`* für die Prozess-ID des `inetd`-Daemons steht.

Wenn Sie diese Schritte durchgeführt haben, sollten Sie über Ihren bevorzugten Browser auf SWAT zugreifen können. Um dies zu tun, besuchen Sie Ihren Samba-Server an Port 901, indem Sie in der Adresszeile Ihres Browsers `http://`*`Ihr-Server`*`:901/` eintragen und `Ihr Server` durch die IP-Adresse des Servers oder seinen DNS-Namen ersetzen.

Wenn Ihr Browser SWAT kontaktiert hat, wird Ihnen ein Autorisierungsdialogfeld präsentiert, das Ihren Benutzernamen und Ihr Passwort verlangt. Sie müssen hier einen ausreichend privilegierten Benutzer eingeben, wie z.B. `root`. Abbildung 9.1 zeigt, wie Sie einen Browser benutzen würden, um über SWAT auf `EAGLE` zuzugreifen.

Wenn Sie SWAT benutzen, senden Sie Ihren Benutzernamen und Ihr Passwort in Klartext über das Netzwerk. Es ist keine sehr gute Idee, jemandem zu ermöglichen, über SWAT Ihren Samba-Server entfernt über das Internet zu verwalten, da er dann Benutzernamen und Passwörter in Klartext über das Internet sendet.

Abb. 9.1: Von einem Browser auf SWAT zugreifen



Nachdem Sie sich eingeloggt haben, wird die Haupt-SWAT-Seite geöffnet (siehe Abbildung 9.2), auf der Sie die folgenden Bereiche auswählen können:

- *Home:* Führt Sie zurück zur SWAT-Homepage.
- *Globals:* Hier können Sie den globalen Abschnitt für diesen Samba-Server verwalten.
- *Shares:* Hier können Sie die Dateifreigaben für diesen Samba-Server verwalten.
- *Printers:* Hier können Sie die Druckerfreigaben für diesen Samba-Server verwalten.
- *Status:* Hier können Sie Statusinformationen zu diesem Samba-Server erhalten.
- *View:* Hier können Sie sich die aktuelle `smb.conf`-Datei ansehen.
- *Password:* Hier können Sie das Passwort für Ihren Samba-Server oder einen entfernten Rechner verwalten.

Abb. 9.2: Die SWAT-Homepage

Samba Web Administration Tool - Netscape

Datei Bearbeiten Ansicht Gehe Communicator Hilfe

Zurück Vor Neu laden Anfang Suchen Guide Drucken Sicherheit Shop Stop

Lesezeichen Gehe zu: http://eagle:901/ Verwandte Objek

Instant Message Internet Neuigkeiten Interessantes Mitglieder Verbindungen Marktplatz

Welcome to SWAT!

Please choose a configuration action using one of the above buttons

Documentation

- **Daemons**
  - smbd - the SMB daemon
  - nmbd - the NetBIOS nameserver
- **Administrative Utilities**
  - smbstatus - monitoring Samba
  - SWAT - web configuration tool
  - smbpasswd - managing SMB passwords
  - make_smbcodepage - codepage creation

Dokument: Übermittelt

Sie können jederzeit zur SWAT-Homepage zurückkehren, indem Sie auf die Schaltfläche *Home* klicken.

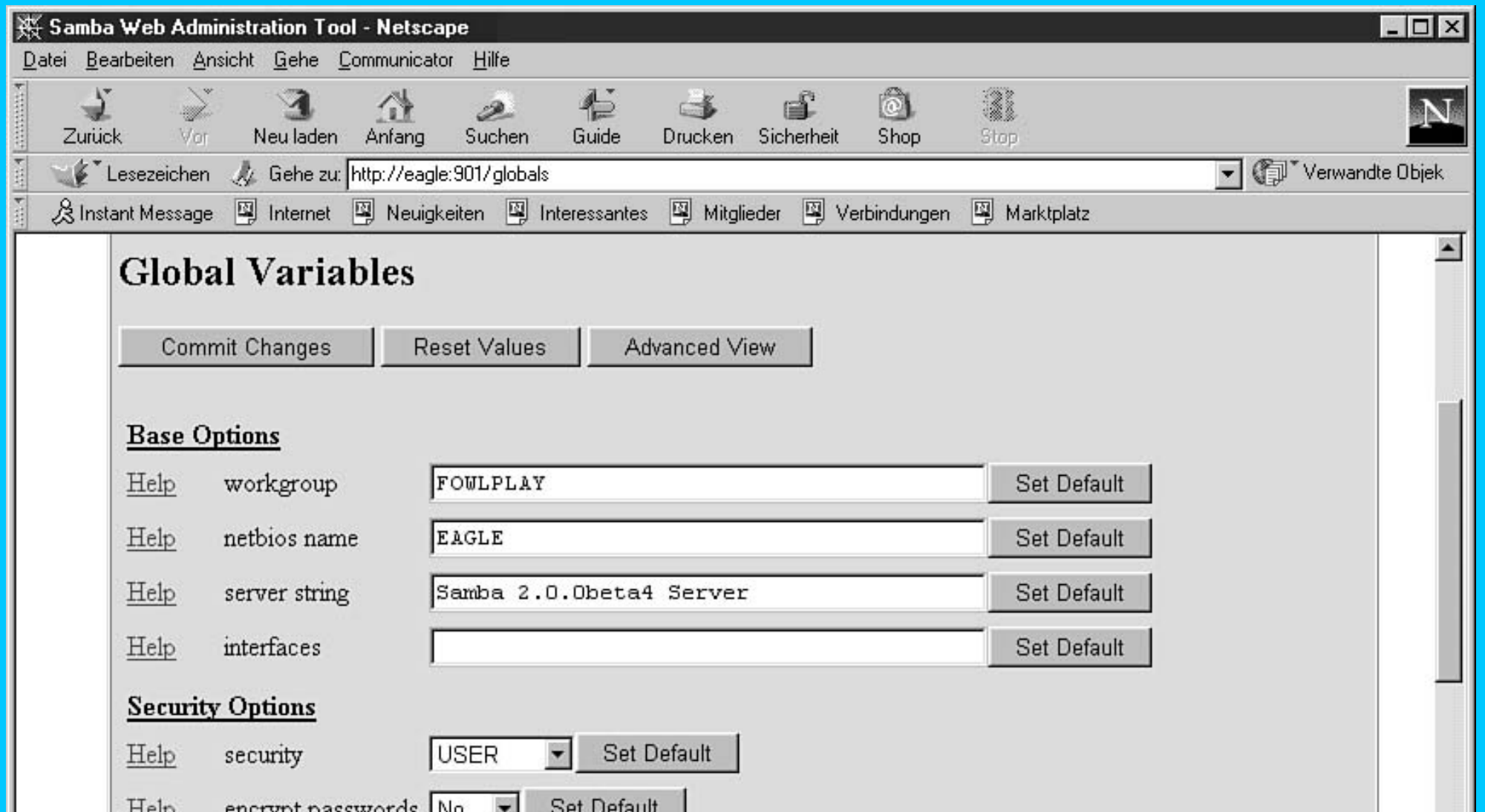
SWAT sollte auf jedem Betriebssystem funktionieren, auf dem Samba läuft, während einige der anderen in diesem Kapitel dargestellten Konfigurationstools eingeschränkter sind.

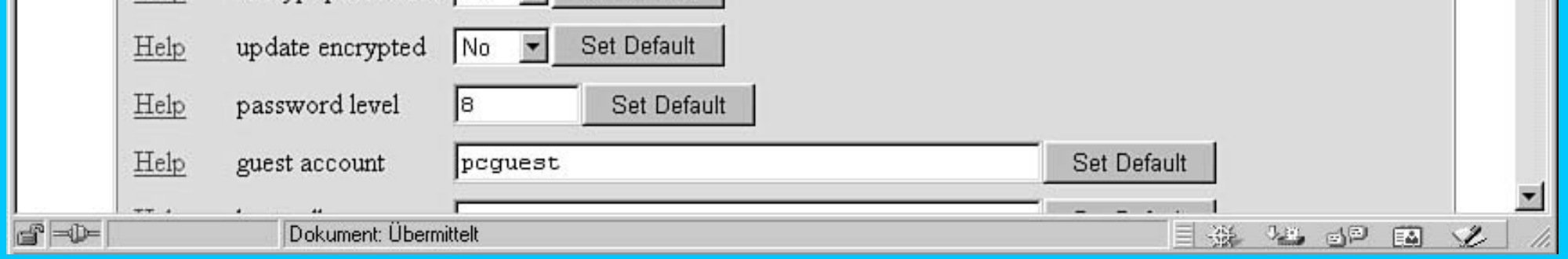
Die folgenden Abschnitte stellen jede der Konfigurationsseiten dar, auf die Sie zugreifen können.

Den Abschnitt [global] verwalten

Wenn Sie auf die Schaltfläche *Globals* klicken, öffnet SWAT eine Webseite, auf der Sie viele der relevantesten globalen Samba-Parameter modifizieren können. Diese Webseite sehen Sie in Abbildung 9.3. Die globalen Samba-Variablen sind nach verwandten Themen gruppiert.

Abb. 9.3: SWAT ermöglicht Ihnen die Modifizierung der Parameter im globalen Abschnitt





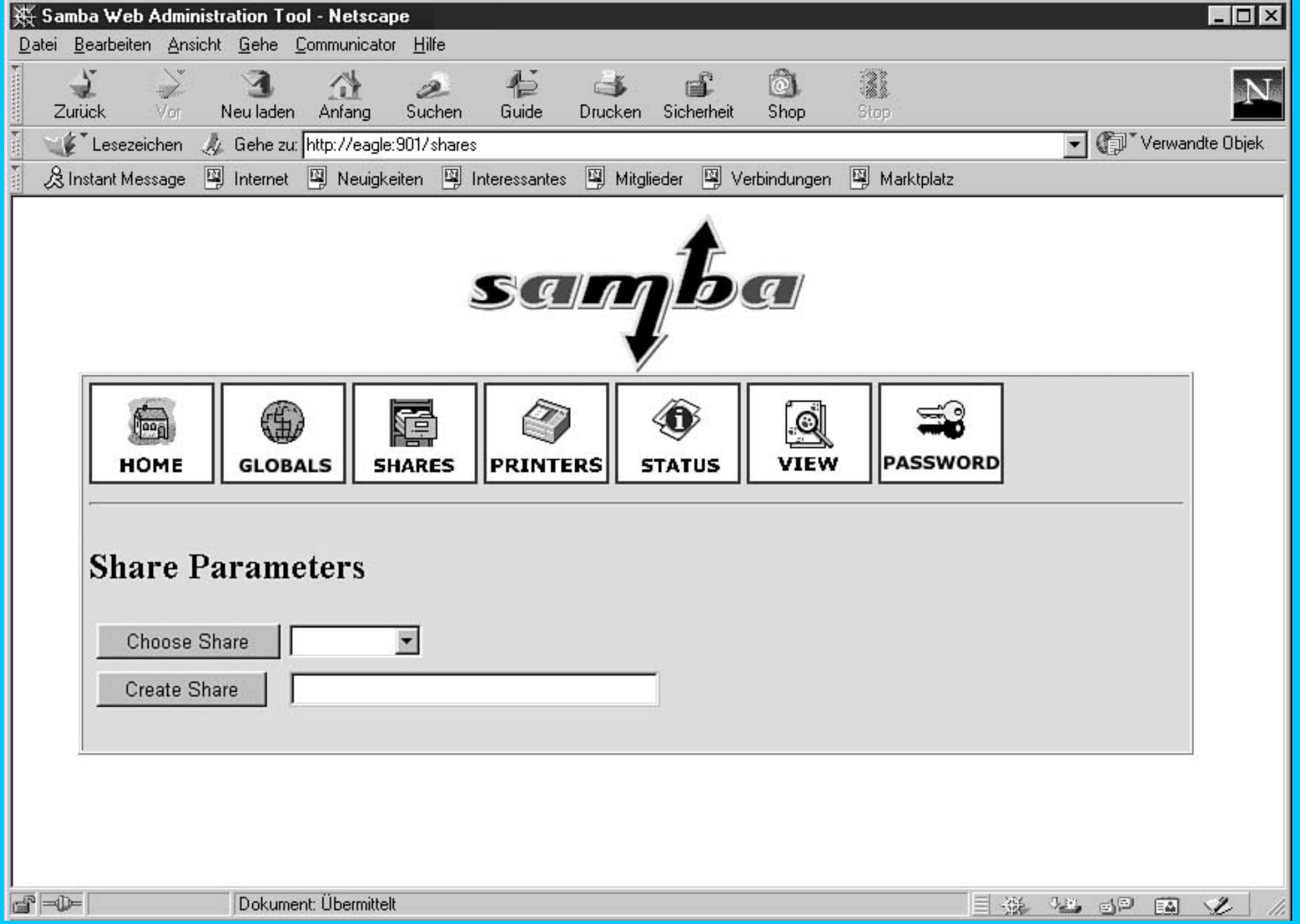
Wenn Sie auf die Schaltfläche *Advanced View* klicken, erhalten Sie die gleichen Gruppen verwandter Optionen, können aber jetzt alle bearbeiten.

Um eine Änderung durchzuführen, blättern Sie einfach zu dem Parameter, den Sie ändern wollen, geben den neuen Wert ein und klicken dann auf die Schaltfläche *Commit Changes*.

Dateifreigaben verwalten

Wenn Sie auf die Schaltfläche *Shares* klicken, öffnet SWAT eine Webseite, auf der Sie neue Freigaben einrichten und existierende Freigaben bearbeiten können. Diese Seite ist in Abbildung 9.4 dargestellt.

Abb. 9.4: Einrichten und Modifizieren von Freigaben mit SWAT

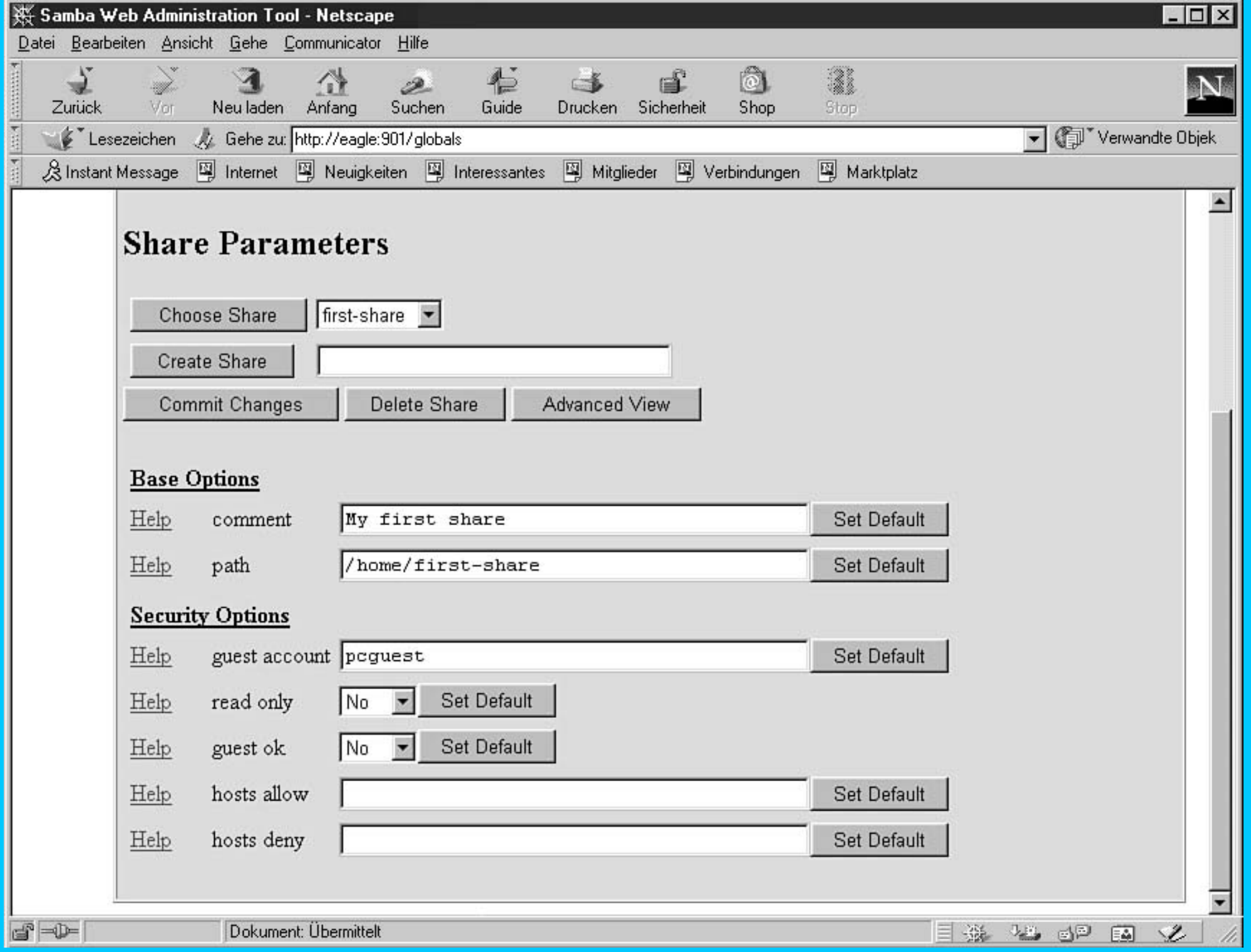


Um einen der Parameter in einer existierenden Freigabe zu ändern, wählen Sie die Freigabe aus der Dropdown-Liste neben der Schaltfläche *Choose Share* aus und klicken dann auf diese Schaltfläche. Sie sehen nun die Seite, die in Abbildung 9.5 dargestellt ist.

Wollen Sie eine neue Freigabe einrichten, geben Sie ihren Namen in das Feld neben der Schaltfläche *Create Share* ein und klicken danach auf *Create Share*. Sie sehen dann eine Seite, die der in Abbildung 9.5 ähnelt, mit dem Namen Ihrer neuen Freigabe als Auswahl im ersten Feld.

Sie sollten bemerken, dass die Abbildungen 9.4 und 9.5 die gleichen Felder oben auf der Seite haben, d.h. auf beiden Seiten finden Sie die Schaltflächen *Choose Share* und *Create Share*. Diese ermöglichen es Ihnen, eine Freigabe anzusehen oder eine neue Freigabe zu erstellen, ohne zur SWAT-Homepage zurückgehen zu müssen. Geben Sie einfach den Namen der Freigabe ein und klicken Sie auf *Choose Share*.

Abb. 9.5: Ändern oder Einrichten einer Freigabe über SWAT


Samba Web Administration Tool - Netscape
Datei Bearbeiten Ansicht Gehe Communicator Hilfe
Zurück Vor Neu laden Anfang Suchen Guide Drucken Sicherheit Shop Stop
Lesezeichen Gehe zu: http://eagle:901/globals Verwandte Objek
Instant Message Internet Neuigkeiten Interessantes Mitglieder Verbindungen Marktplatz
Share Parameters
Choose Share first-share
Create Share
Commit Changes Delete Share Advanced View
Base Options
Help comment My first share Set Default
Help path /home/first-share Set Default
Security Options
Help guest account pcguest Set Default
Help read only No Set Default
Help guest ok No Set Default
Help hosts allow Set Default
Help hosts deny Set Default
Dokument: Übermittelt

Auf dieser Seite können Sie

- eine andere Freigabe auswählen, indem Sie sie markieren und auf *Choose Share* klicken,
- eine neue Freigabe erstellen, indem Sie ihren Namen in das entsprechende Feld eingeben und auf *Create Share* klicken,
- alle bisher durchgeführten Änderungen übertragen, indem Sie auf *Commit Changes* klicken,
- die Freigabe löschen, indem Sie auf *Delete Share* klicken.

Wenn Sie Parameter ändern müssen, die nicht auf dieser Seite gezeigt werden, klicken Sie auf die Schaltfläche *Advanced View* und ändern die entsprechenden Parameter.

Die *Advanced-View*-Seite zeigt alle Parameter, die für die ausgewählte Freigabe relevant sind, gruppiert in folgende Abschnitte:

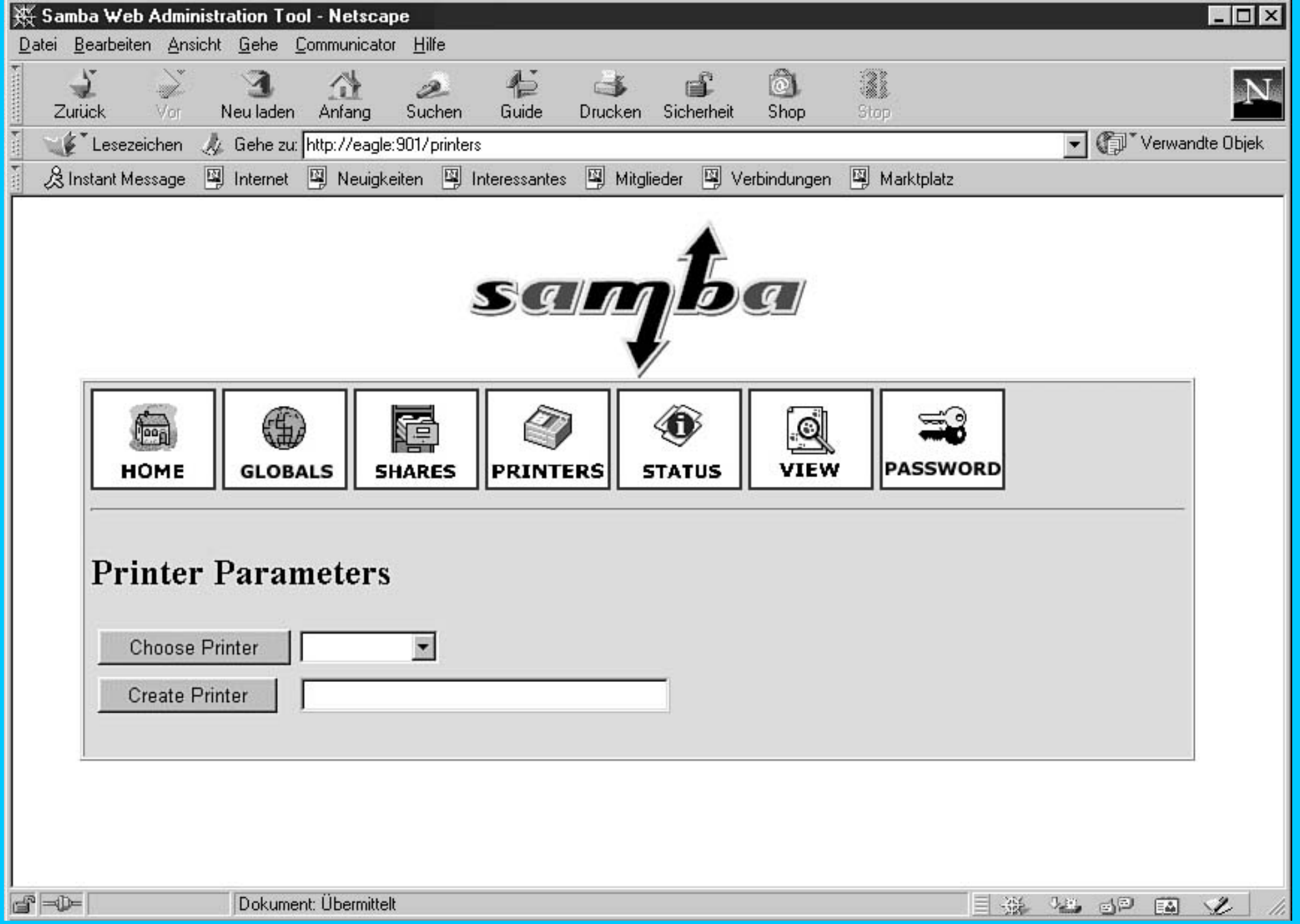
- Grundlegende Optionen (*Base Options*): Kommentar und Pfad
- Sicherheitsoptionen (*Security Options*): Benutzername, Gast-Account usw.
- Logging-Optionen (*Logging Options*): z.B. Status
- Tuning-Optionen (*Tuning Options*): z.B. maximale Verbindungen, Sync Always uws.
- Dateinamenbehandlung (*Filename Handling*): z.B. Parameter für die Schreibweise usw.
- Browse-Optionen (*Browse Options*): z.B. Oplocks und Strict Locking usw.
- Weitere Optionen (*Miscellaneous Options*)

Wenn Sie alle benötigten Änderungen durchgeführt haben, klicken Sie auf *Commit Changes*, und die Änderungen werden der Freigabe hinzugefügt. Samba nimmt die von Ihnen ausgewählten Änderungen sofort an.

Druckerfreigaben verwalten

Wenn Sie auf die Schaltfläche *Printers* klicken, öffnet SWAT eine Webseite, auf der Sie neue Drucker einrichten und existierende Drucker modifizieren können. Diese Seite ist in Abbildung 9.6 dargestellt.

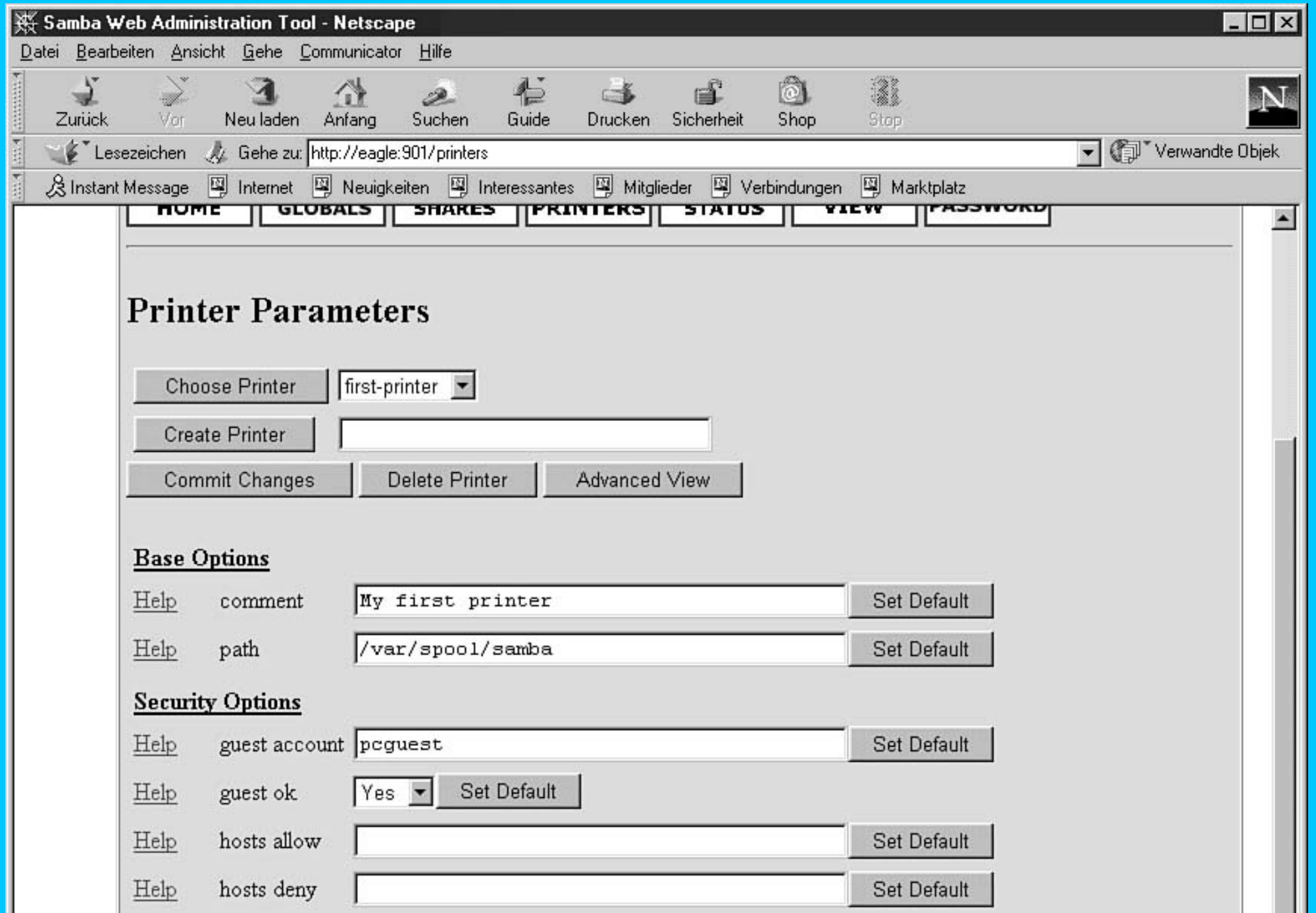
Abb. 9.6: Einrichten und Modifizieren von Druckern mit SWAT



Um einen existierenden Drucker zu modifizieren, wählen Sie ihn in der Dropdown-Liste neben der Schaltfläche *Choose Printer* aus und klicken dann auf *Choose Printer*. Sie sehen die Seite, die in Abbildung 9.7 dargestellt ist.

Um einen neuen Drucker einzurichten, geben Sie den Namen des Druckers in das Textfeld neben der Schaltfläche *Create Printer* ein und klicken dann auf *Create Printer*. Sie kommen zu einer ähnlichen Seite wie der in Abbildung 9.7, mit dem Namen Ihres neuen Druckers im ersten Feld.

Abb. 9.7: Einen Drucker über SWAT modifizieren oder einrichten



Auf dieser Seite können Sie

- einen anderen Drucker auswählen, indem Sie ihn markieren und auf *Choose Printer* klicken,
- einen neuen Drucker einrichten, indem Sie seinen Namen in das entsprechende Feld eingeben und auf *Create Printer* klicken,
- alle bisherigen Änderungen übertragen, indem Sie auf *Commit Changes* klicken,
- einen Drucker entfernen, indem Sie auf *Delete Printer* klicken.

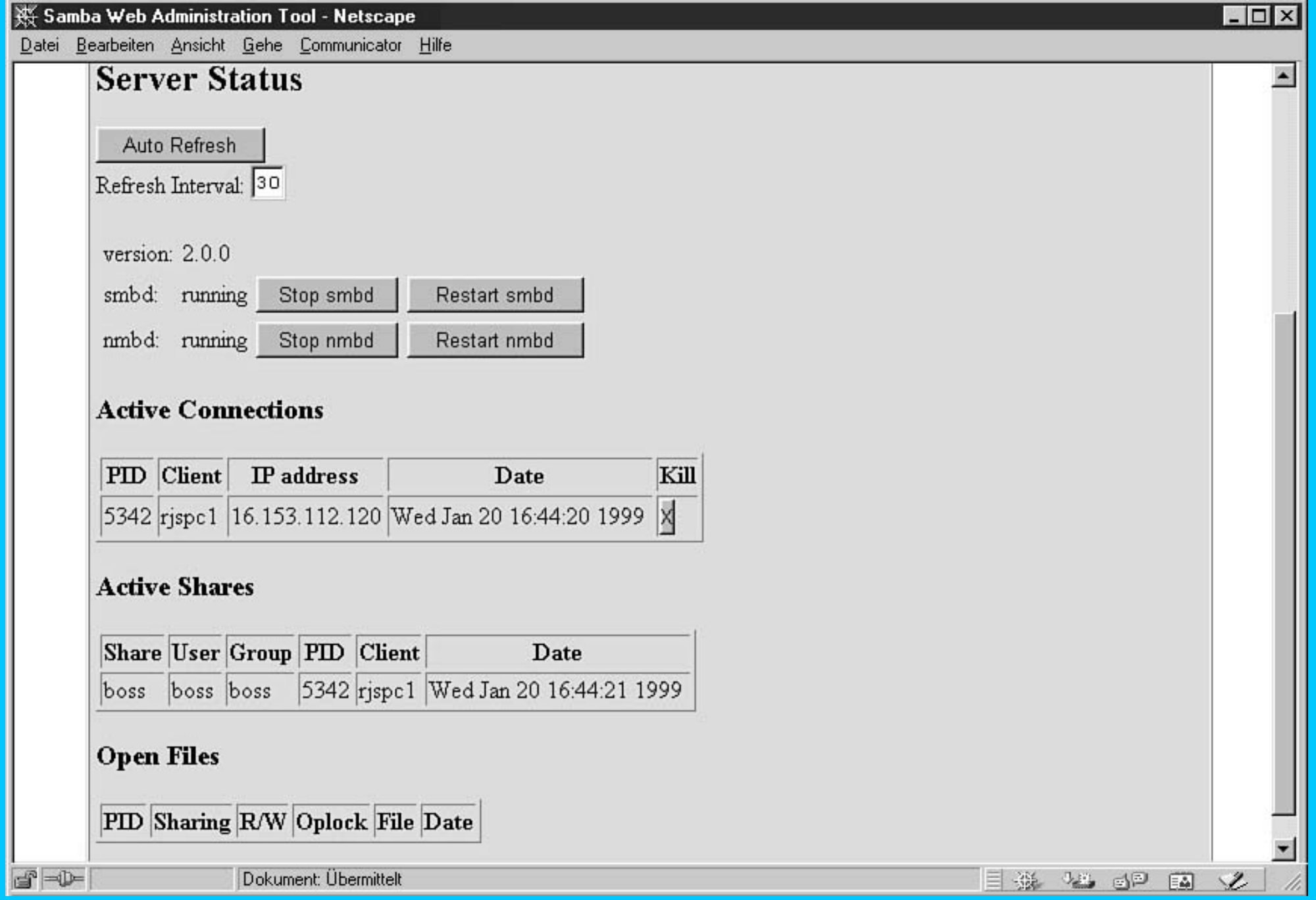
Wenn Sie Parameter ändern müssen, die nicht auf dieser Seite gezeigt werden, klicken Sie auf die Schaltfläche *Advanced View* und ändern die entsprechenden Parameter.

Wenn Sie alle benötigten Änderungen durchgeführt haben, klicken Sie auf *Commit Changes*, und die Änderungen werden der Freigabe hinzugefügt. Samba nimmt die von Ihnen ausgewählten Änderungen sofort an.

Statusinformationen abrufen

Wenn Sie auf die Schaltfläche *Status* klicken, öffnet SWAT eine Webseite, die Statusinformationen zu Samba bietet und es Ihnen ermöglicht, die Samba-Daemons zu starten oder anzuhalten. Außerdem können Sie hier die Verbindungen aktiver Benutzer trennen. Die von SWAT geöffnete Webseite sehen Sie in Abbildung 9.8.

Abb. 9.8: Die SWAT-Statusseite



Die Statusseite bietet auch eine Funktion, über die sie immer wieder aktualisiert wird. Spezifizieren Sie einfach ein Zeitintervall für die Aktualisierung und klicken Sie auf *Auto Refresh*.

Die gesamte smb.conf-Datei ansehen

Wenn Sie auf die Schaltfläche *View* klicken, öffnet SWAT eine Webseite, auf der Sie die gesamte `smb.conf`-Datei ansehen können. Diese Seite ist in Abbildung 9.9 dargestellt.

SWAT listet die Samba-Konfiguration so auf, wie sie in `smb.conf` definiert ist. Wenn Sie eine Auflistung wollen, in der die Werte aller von Samba verwalteten Parameter enthalten sind, klicken Sie einfach auf die Schaltfläche *Full View*.

Ihr Passwort ändern

Wenn Sie auf die Schaltfläche *Password* klicken, öffnet SWAT eine Webseite, auf der Sie Ihr Passwort für den Samba-Server, auf dem SWAT läuft, oder Ihr Passwort für einen anderen CIFS/SMB-Server in Ihrem Netzwerk ändern können. Sie können außerdem Benutzer hinzufügen und Benutzer deaktivieren bzw. aktivieren. Die entsprechende Seite ist in Abbildung 9.10 dargestellt.

Abb. 9.9: Die Datei smb.conf

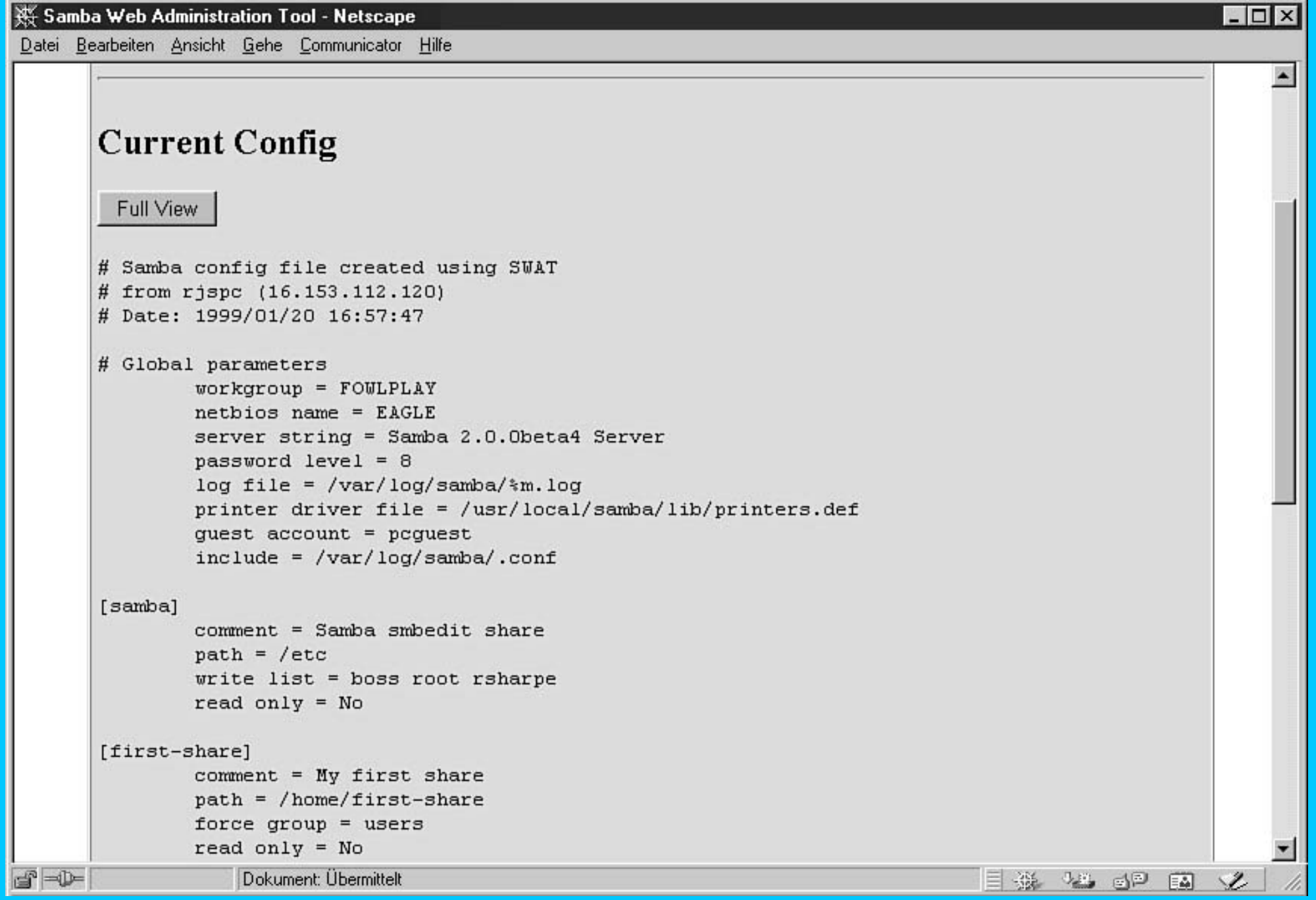


Abb. 9.10: Über SWAT Ihr Passwort ändern


Samba Web Administration Tool - Netscape
Datei Bearbeiten Ansicht Gehe Communicator Hilfe
HOME
GLOBALS
SHARES
PRINTERS
STATUS
VIEW
PASSWORD
Server Password Management
User Name : root
New Password :
Re-type New Password :
Change Password
Add New User
Disable User
Enable User
Client/Server Password Management
User Name : root
Old Password :
New Password :
Re-type New Password :
Remote Machine :
Change Password
Dokument: Übermittelt

SWAT bearbeitet nur Ihre `smbpasswd`-Datei, nicht Ihre normalen Unix-Passwortdateien.

Webmin

Webmin ist ein Web-basiertes System-Administrationspaket für Unix-Systeme. Es bietet Funktionen für die Verwaltung von Samba und für die Einrichtung von Accounts, die Konfiguration von DNS, Apache und sendmail sowie für die Durchführung vieler anderer Aufgaben der Systemadministration. Hier konzentrieren wir uns darauf, wie Webmin bei der Konfiguration von Samba zum Einsatz kommt.

Webmin besteht aus einem in Perl geschriebenen Mini-Web-Server und einer Sammlung von CGI-Skripten, die alle erforderlichen Funktionen implementieren, um die Systemadministration über das Web zu unterstützen. Sie müssen sich zunächst eine Version von Webmin besorgen und sie installieren.

Sie erhalten Webmin unter http://www.webmin.com/webmin/. Nachdem Sie Webmin als `gzip-tar`-Datei heruntergeladen haben, müssen Sie die Datei z.B. mit folgendem Befehl entpacken:

```
gzip -d webmin-VER_tar.gz
tar -xvf webmin-VER_tar
```

Auf einigen Systemen ist auch der Befehl `tar -xvzf webmin-`*VER*`_tar.gz` ausreichend. Für alle Befehle ersetzen Sie *VER* mit der aktuellen Version von Webmin. Zur Zeit der Bucherstellung war dies 0.77.

Nachdem Sie die Distribution entpackt haben, wechseln Sie einfach in das soeben erstellte Verzeichnis, in der Regel `webmin-`*VER*, wobei *VER* die Versionsnummer von Webmin darstellt. Lesen Sie dann die `README`-Datei für Anweisungen zur Installation. Als ich dieses Buch schrieb, wurde die Installation über folgenden Befehl ausgeführt:

```
./setup.sh
```

Beantworten Sie die Fragen, die Ihnen vom Installationsskript gestellt werden. Während der Installation werden Sie dazu aufgefordert, ein Passwort für den ersten Webmin-Benutzer, `admin`, einzugeben. Dieses Passwort brauchen Sie, wenn Sie sich mit der Webmin-Seite verbinden.

Nachdem Sie Webmin installiert haben, müssen Sie die Startskripte Ihres Systems ändern, um sicherzustellen, dass Webmin bei jedem Bootvorgang des Systems gestartet wird. Details hierzu würden den Rahmen dieses Buches sprengen. Bitte werfen Sie einen Blick in die Dokumentation zur Systemadministration für Ihr System.

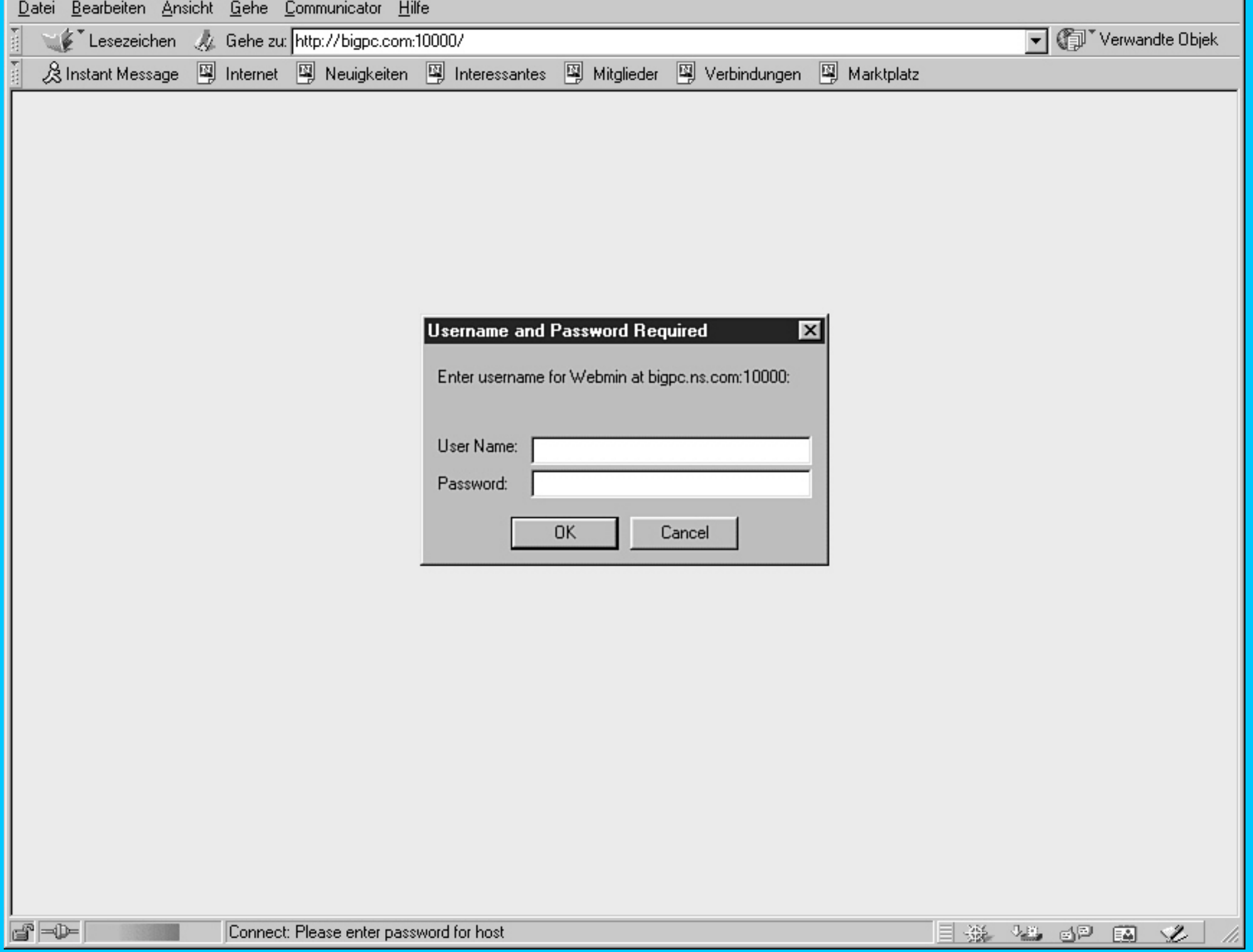
Webmin unterstützt derzeit die folgenden Betriebssysteme:

- Sun Solaris
- Caldera OpenLinux eServer
- Caldera OpenLinux
- Redhat Linux
- Slackware Linux
- Debian Linux
- SuSE Linux
- Corel Linux
- TurboLinux
- Mandrake Linux
- Delix DLD Linux
- MkLinux
- FreeBSD
- OpenBSD
- BSDI
- HP/UX
- SGI Irix
- DEC/Compaq OSF/1
- IBM AIX
- SCO UnixWare
- SCO OpenServer
- MacOS Server X

Vollständigere Informationen zu den von Webmin unterstützten Betriebssystemen finden Sie auf der Webmin-Webpage und im Installationsskript.

Nachdem Sie Webmin installiert und gestartet haben, können Sie mit Ihrem bevorzugten Browser darauf zugreifen, indem Sie zu Port 10.000 auf dem Server verbinden, auf dem Sie Webmin installiert haben. Wenn Sie die Portnummer, hinter der Webmin sitzt, vom Standardwert 10.000 in einen anderen Wert geändert haben, müssen Sie diese Zahl statt der 10.000 verwenden.

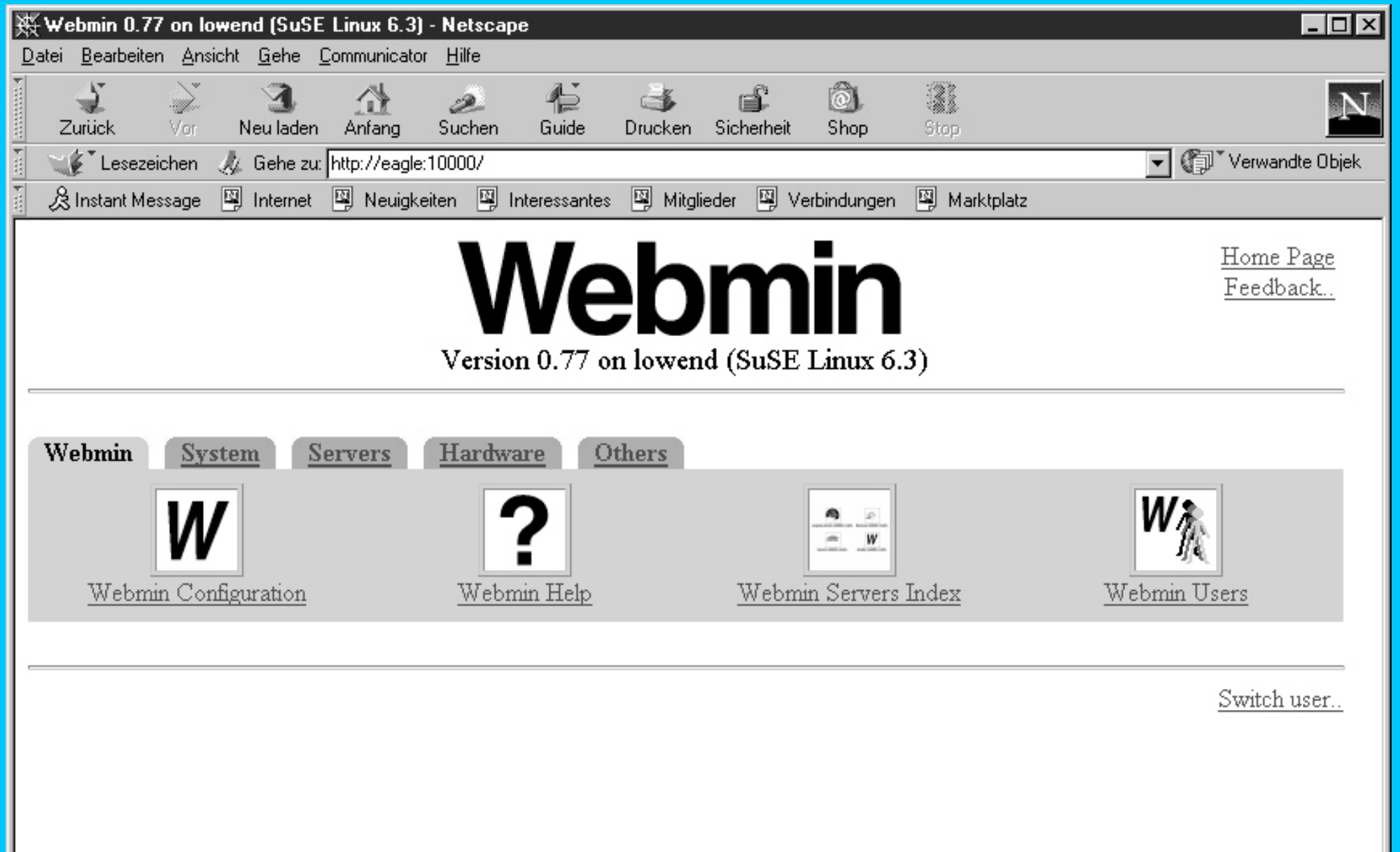
Abb. 9.11: Mit Webmin verbinden

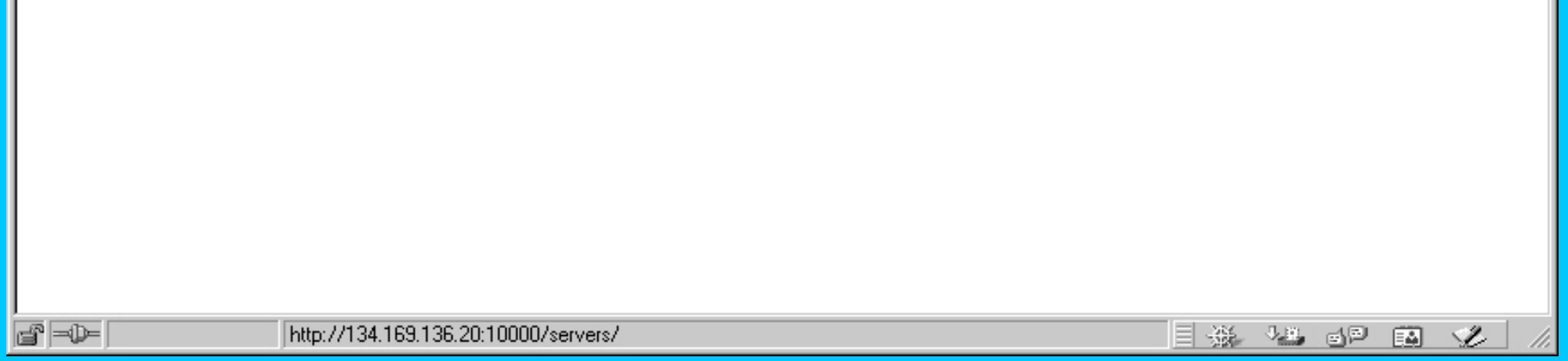

Datei Bearbeiten Ansicht Gehe Communicator Hilfe
Lesezeichen
Gehe zu:
http://bigpc.com:10000/
Verwandte Objek
Instant Message
Internet
Neuigkeiten
Interessantes
Mitglieder
Verbindungen
Marktplatz
Username and Password Required
Enter username for Webmin at bigpc.ns.com:10000:
User Name:
Password:
OK
Cancel
Connect: Please enter password for host

Nachdem Sie den korrekten Benutzernamen und das entsprechende Passwort eingegeben haben, kommen Sie zur Webmin-Homepage, die in Abbildung 9.12 dargestellt ist. Webmin kann viele Aspekte von Unix-Systemen verwalten, aber uns interessieren hier nur die Funktionen zur Verwaltung von Samba. Wählen Sie nunächst *Servers* und danach den Link *Samba Windows File Sharing* auf der Webmin-Homepage.

Auch hier sollten Sie wieder beachten, dass Webmin keine Unterstützung für *Secure Sockets Layer (SSL)* bietet und Sie also Benutzernamen und Passwörter in Klartext über das Netzwerk senden. Dies ist ein Sicherheitsproblem, und das mindeste, was Sie tun können, ist zu verhindern, dass jemand Ihre Samba-Server über das Internet verwalten kann.

Abb. 9.12: Die Webmin-Homepage



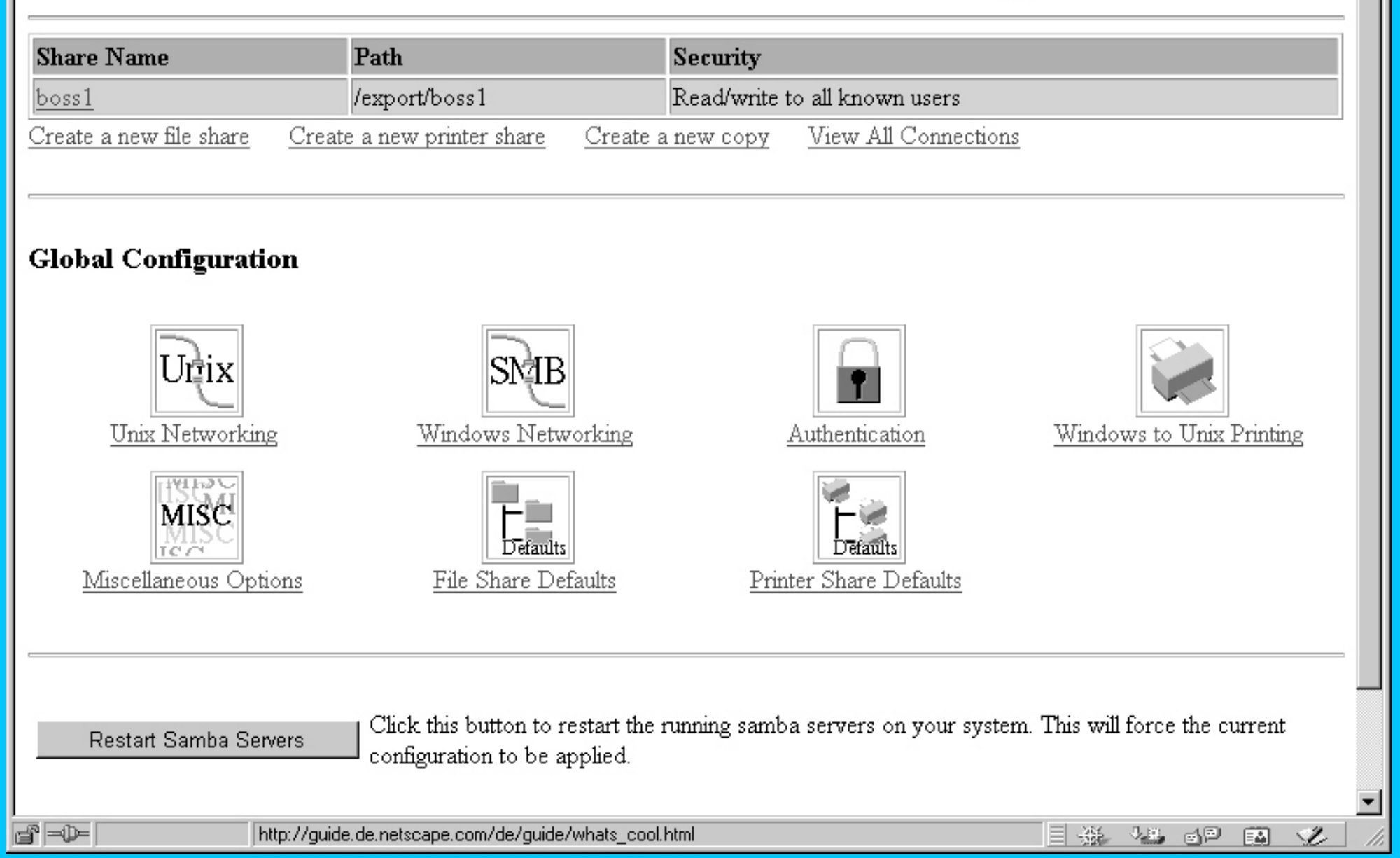


Nachdem Sie auf den Link *Samba Windows File Sharing* geklickt haben, öffnet Webmin die Samba-Exports-Seite, die in Abbildung 9.13 dargestellt ist. Über diese Seite können Sie Datei- und Druckerfreigaben sowie alle Aspekte der globalen Samba-Parameter verwalten. Sie können globale Konfigurationsabschnitte auswählen, indem Sie auf die Schaltflächen klicken, die Sie unten in Abbildung 9.13 sehen.

Sie können über diese Seiten die gleichen Funktionen ausführen, die Sie auch mit SWAT ausführen können.

Webmin kann keine Freigaben erstellen, die einen Bindestrich (-) enthalten, und funktioniert nicht mit `smb.conf`-Dateien, die `include`-Parameter enthalten.

Abb. 9.13: Die Webmin-Samba-Exports-Seite



smbconftool

`smbconftool` ist eine Java-Applikation, über die Sie die Datei `smb.conf` auf grafische Weise editieren können. Sie läuft auf dem Rechner, auf dem sich die `smb.conf` befindet. `smbconftool` kann zwar auf jedem Rechner laufen, der Java unterstützt, da das Programm Datei-IO benutzt, um auf die `smb.conf`-Datei zuzugreifen, aber es muss auf dem System laufen, in dem sich die `smb.conf` befindet. Weitere Informationen zu diesem Tool finden Sie unter http://www.eatonweb.com/samba/.

smb-mode.el

Dieses Tool ist ein Emacs-Modus, der von Fraser McCrossan geschrieben wurde und Administratoren bei der Bearbeitung der Datei `smb.conf` unterstützt. Wenn Sie `smb-mode.el` heruntergeladen und konfiguriert haben, erhalten Sie Funktionen wie:

- automatische Einzüge für Ihre `smb.conf`-Datei
- Editor-Operationen auf ganzen Abschnitten Ihrer `smb.conf`
- einen neuen Modus, den Outline-Mirror-Modus, der Ihnen einen Überblick über die gesamte `smb.conf` geben kann
- Parametersuche in der `smb.conf`-Manpage
- Parametervervollständigung
- einige weitere Funktionen

Um `smb-mode.el` mit dem Emacs zu benutzen, sollten Sie es von http://users.gtn.net/fraserm/smbmode.html herunterladen. Auf dieser Seite finden Sie auch Anweisungen zur Installation von `smb-mode.el`.

Diese Schritte müssen Sie ausführen:

1. Laden Sie `smb-mode.el` von http://users.gtn.net/fraserm/smbmode.html herunter.
2. Installieren Sie `smb-mode.el` in Ihrem Lisp-Verzeichnis.
3. Fügen Sie Ihrer `.Emacs`-Datei Lisp hinzu (oder erstellen Sie Ihre `.Emacs`-Datei, wenn sie noch nicht existiert).

Anweisungen für die Installation und Verwendung von `smb-mode.el` finden Sie auf der oben erwähnten Webseite.

Zusammenfassung

Die Konfiguration Ihrer `smb.conf`-Datei kann einer der schwierigsten Punkte sein, wenn es darum geht, dass Samba korrekt auf Ihrem System funktioniert. Die in diesem Kapitel dargestellten Tools machen die Verwaltung Ihrer `smb.conf` wesentlich einfacher.

Viele dieser Tools befinden sich jedoch noch in frühen Phasen ihrer Entwicklung und alle haben irgendein Problem (viele entfernen z.B. Ihre Kommentare, einige unterstützen keine `include`-Abschnitte usw.). Für erfahrene Administratoren liegt der Hauptnutzen eines Tools wie SWAT oder Webmin darin, dass sie `smb.conf` schnell über einen Webbrowser ändern können, wenn sie nicht in den Server eingeloggt sind.

Ich möchte noch einmal darauf hinweisen, dass die heute verfügbaren Web-basierten Tools keine SSL-Unterstützung bieten. Das heißt, dass Sie

Benutzernamen und Passwörter in Klartext über das Netzwerk senden. Dies stellt für viele sicherheitsbewußte Leute ein echtes Problem dar. Sie sollten zumindest vermeiden, dass jemand Ihre Samba-Server über das Internet verwalten kann!

Im nächsten Kapitel werden Sie sich mit dem Thema Automatisierung auf Server-Seite befassen und lernen, wie Sie Makros und andere fortschrittliche Techniken in Ihrer `smb.conf` verwenden können.

Frage & Antwort

F. Ich versuche, SWAT zu benutzen, erhalte aber immer eine Meldung wie: »Es gab keine Antwort. Der Server könnte inaktiv sein oder nicht antworten ...« Was ist das Problem?

. Dies hört sich an, als ob Samba nicht auf Verbindungen horcht oder Sie den falschen URL benutzen, wenn Sie sich mit SWAT verbinden wollen.

SWAT befindet sich normalerweise an Port 901; Sie sollten also den URL `http://`*`Ihr.Server.dom`*`:901/` benutzen, wobei Sie *`Ihr.Server.dom`* durch den DNS-Namen Ihres Samba-Servers ersetzen müssen.

Wenn Sie immer noch Probleme haben, stellen Sie sicher, dass SWAT in der Datei `/etc/services` aufgelistet ist und `/etc/inetd.conf` einen Eintrag für SWAT hat, wie es vorher dargestellt wurde.

---

1 GUI = Graphical User Interface, grafische Benutzeroberfläche

Tag 10: Automatisierung auf Server-Seite

Automatisierung kann als der Prozess für die Entwicklung einer Lösung definiert werden, die ohne menschliches Eingreifen ausgeführt werden kann. Automatisierung und Skalierbarkeit haben mehr miteinander zu tun, als man auf den ersten Blick glauben könnte. In diesem Kapitel werden Sie Methoden für die Automatisierung der Samba-Funktionen entwickeln, die Verbindungen von verschiedenen Clients handhaben. Auch wenn Sie nicht viele Benutzer, Computer oder Server verwalten müssen, können Sie davon profitieren, über Automatisierung nachzudenken.

Was ist Automatisierung auf Server-Seite?

Automatisierung auf Server-Seite beschreibt Aktionen, die der Server aufgrund der Konfiguration durch den Systemadministrator durchführt, wenn sich ein Client mit einer Freigabe, sei es Datei- oder Druckerfreigabe, verbindet. Vielleicht war diese Darstellung etwas langatmig, aber ich denke, die Definition wird verständlicher, wenn wir uns einige Beispiele ansehen.

Wenn Sie an zurückliegende Kapitel denken, haben Sie bereits zwei Beispiele für Automatisierung auf Server-Seite gesehen. Erinnern Sie sich noch an die `[homes]`-Freigabe? Wenn sich ein Benutzer mit seinem Home-Verzeichnis (z.B. `\\bilbo\joe`) verbinden will - vorausgesetzt, dass die `[homes]`-Freigabe definiert wurde -, versucht Samba, den Freigabenamen zu lokalisieren, erst in der `smb.conf`, dann in der lokalen `passwd`-Datei. Wird der Name in der `/etc/passwd` gefunden, erstellt Samba automatisch eine Kopie der `[homes]`-Freigabe und ändert ihren Namen in den Namen des Benutzers, der sich verbinden will. Das alles ist für den Benutzer transparent und erfolgt ohne jegliches menschliches Eingreifen. Wenn Sie für einen neuen Benutzer einen Account auf dem Unix-Rechner einrichten, müssen Sie keine Änderungen an der `smb.conf` vornehmen, damit sich der Benutzer mit seinem Home-Verzeichnis verbinden kann. Das ist, was ich mit Automatisierung auf Server-Seite meine. Die Lösung ist skalierbar und verwaltet sich selbst.

Die `[printers]`-Freigabe aus Kapitel 8, »Drucker«, ist ein weiteres Beispiel für die in Samba integrierten Automatisierungsfunktionen. Sie müssen nicht jeden auf dem System verfügbaren Drucker ausdrücklich definieren (obwohl Sie es könnten, wenn Sie wollten), damit der Drucker freigegeben werden kann. Samba holt sich seine Liste zugelassener Druckernamen aus der definierten `printcap`-Datei. Wenn Sie einen weiteren Drucker einrichten, wird Samba dies automatisch erkennen.

Bevor Sie damit beginnen, sich einige der eher benutzerdefinierten Methoden für die Automatisierung der Client-Unterstützung anzusehen, möchte ich darauf hinweisen, dass diese Beispiele nur Vorschläge und sicherlich nicht die einzige Art und Weise sind, in der die Parameter oder Mechanismen benutzt werden können. Sie sollten mit verschiedenen Kombinationen experimentieren. Ich hoffe, dass Ihnen die Beispiele gefallen und Sie bis zum Ende des Kapitels eigene Ideen haben, die Sie ausprobieren wollen. Samba sollte so funktionieren, wie Sie es wollen. Die Beispiele hier sollen Ihnen nur als Basis dienen, von der aus Sie arbeiten können.

Sie haben sich die in der `smb.conf` verfügbaren Variablen bereits in Kapitel 5, »Die Datei `smb.conf`: Samba mitteilen, was es tun soll« angesehen. Tabelle 5.2 listet diese Variablen auf, falls Sie noch einmal einen Blick darauf werfen wollen. Variablen bilden die Grundlage für die Individualisierung von Verbindungen. Einige der meistbenutzten Variablen, die ich im Rest dieses Kapitels verwenden werde, sind `%u`, `%U`, `%g`, `%G`, `%m`, `%L` und `%d`.

preexec- und postexec-Skripte

Vier Parameter ermöglichen Ihnen die Definition von Befehlen, die auf dem Server ausgeführt werden, wenn ein Client sich mit einer Freigabe verbindet oder die Verbindung beendet:

- `preexec`
- `postexec`
- `root preexec`
- `root postexec`

preexec und postexec

Schauen wir uns zunächst die Parameter `preexec` und `postexec` an. Beide nehmen eine Sammlung von Befehlen oder den Namen eines Skripts an, die auf dem Server unter Benutzung der UID `%u` aufgeführt werden. Vielleicht erinnern Sie sich noch aus Kapitel 5, dass `%u` der Benutzername der aktuellen Freigabe ist. Ich kann besser erklären, wann das Skript genau ausgeführt wird, wenn ich die Ausgabe des Befehls `net use` auf einem Windows-Rechner benutze:

```
C:\users\jerry>net use s: \\bilbo\src
Der Befehl wurde erfolgreich ausgeführt.
```

Der über `preexec` definierte Befehl wird zwischen dem Zeitpunkt ausgeführt, zu dem der Befehl `net use` läuft, und dem Zeitpunkt, zu dem Windows ausgibt, dass der Befehl erfolgreich ausgeführt wurde. Das Gleiche gilt für die `postexec`-Befehle. Der Windows-Client überträgt die Anfrage für die Verbindungslösung (`net use s: /d`). Samba führt dann den `postexec`-Befehl aus und teilt dem Windows-Rechner mit, dass die Verbindung zur Freigabe beendet wurde:

```
C:\users\jerry>net use s: /d
Der Befehl wurde erfolgreich ausgeführt.
```

Was können Sie mit diesen `preexec`- und `postexec`-Befehlen tun? Hier ist ein praktisches Beispiel. Mehr als einmal hat mich ein Benutzer angerufen, um mir zu sagen, dass sein Unix-Shell-Account anscheinend nicht richtig funktioniert. Nach ein paar Fragen wurde klar, dass der Benutzer von einem PC auf sein Home-Verzeichnis zugegriffen und einige Dateien gelöscht hatte, die ihm unwichtig erschienen. Ein Freund von mir bezeichnet diese Dateien spaßeshalber als »diese verflixten `dot`-Dateien«. Nun können Sie sich wahrscheinlich vorstellen, welche Dateien der Benutzer gelöscht hat. Hier ist ein Beispiel, das hilft, einige der Support-Anrufe für dieses spezielle Problem zu vermeiden:

```
[homes]
   preexec = /usr/local/bin/fix_dot_files.sh %H
```

Das Skript selbst ist sehr einfach.

```
#!/bin/sh

SKELDIR=/usr/local/etc/skel
EXPORT SKELDIR
home=$1

if [ ! -f $home/.login ]; then
   cp -p $SKELDIR/.login $home
fi

if [ ! -f $home/.logout ]; then
   cp -p $SKELDIR/.logout $home
fi

if [! -f $home/.profile ] then
   cp -p $SKELDIR/.profile $home
fi

if [ ! -f $home/.cahrc ]; then
   cp -p $SKELDIR/.cshrc $home
fi
```

Dies weist Samba an, das Shell-Skript `fix_dot_files` zu starten, wenn sich ein Benutzer mit seinem Home-Verzeichnis verbindet. Dieses spezielle Skript kopiert Standardversionen dieser »verflixten `dot`-Dateien« wie z.B. `.cshrc`, `.profile`, `.login` und `.logout`.

Das nächste Beispiel ist vielleicht ein bisschen dumm und könnte für den Benutzer etwas ärgerlich sein. Der `preexec`-Befehl sendet über eine WinPopup-Nachricht eine Kopie der `~/todo`-Datei an den verbunden Benutzer. Es wäre besser, diesen Befehl in ein Skript einzubetten, das zunächst die Existenz der Datei überprüft:

```
[homes]
   preexec = cat %H/todo | smbclient -M %m
```

Der `postexec`-Parameter ist dem `preexec`-Parameter ähnlich, abgesehen davon, dass der definierte Befehl ausgeführt wird, wenn der Benutzer eine Verbindung zu einer Freigabe beendet. Hier ist ein Beispiel, das alle Dateien aus dem Verzeichnis `~/tmp` des Benutzers entfernt:

```
[homes]
   postexec = /bin/rm -r %H/tmp/*
```

So kann ein Benutzer alle während einer Sitzung benötigten Dateien in seinem Home-Verzeichnis speichern. Damit wird die Sicherheit in einer Umgebung etwas erhöht, in der viele Leute einen Rechner benutzen, z.B. in einem Computerlabor für Studenten. Ich überlasse Ihnen die Entscheidung, ob es eine gute Idee ist, die Dateien automatisch auf Ihrem Server zu löschen.

Das Troubleshooting für diese Skripte kann extrem schwierig sein, da alle eventuellen Fehlermeldungen an Standard Error normalerweise verworfen werden. Mein Vorschlag ist, das Skript als normaler Benutzer manuell auszuführen, um seine korrekte Funktionsweise zu überprüfen. Ein verbreitetes Problem besteht darin, dass ein normaler Benutzer keinen Lese- oder Schreibzugriff auf die für das Skript notwendigen Dateien hat. Es ist auch hilfreich, alle Ausgaben, die von dem Skript generiert werden, in eine Datei zu protokollieren, die Sie sich später ansehen können.

root preexec und root postexec

Der einzige Unterschied zwischen den Parametern `root preexec` bzw. `root postexec` und den normalen `preexec`- bzw. `postexec`-Parametern besteht darin, dass erstere Befehle definieren, die als `root` auf dem Server laufen. Dies kann hilfreich sein für Dinge wie das Erstellen von Verzeichnissen, das Einrichten der Eigentumsverhältnisse von Dateien, die Protokollierung von Verbindungen an eine zentrale Datei wie `/var/adm/wtmp` oder das Mounten bzw. Unmounten von Dateigeräten wie z.B. CD-ROMs oder Disketten.

Sie sollten die über die Parameter `root preexec` und `root postexec` definierten Skripte als SUID-Binaries ansehen, die im Besitz von `root` sind. Das heißt, dass Sicherheitsaspekte wie z.B. die Änderung der Skripte durch Benutzer nicht auf die leichte Schulter genommen werden sollten. Vorausgesetzt dass der Benutzer die Datei `smb.conf` nicht modifizieren kann, können die Parameter, die an das Skript weitergegeben werden, nicht geändert werden. Daher sind die `root-preexec`- und `root-postexec`-Skripte nicht so gefährlich wie `root`-SUID-Binaries, die von einem Shell-Prompt laufen können.

Eine der Aufgaben an meinem derzeitigen Arbeitsplatz besteht darin, mehrere von Studenten benutzte Windows-9x/NT-Labors einzurichten, zu aktualisieren und zu verwalten. Wir haben entschieden, die Labore so unabhängig wie möglich aufzubauen, damit wir Dienstunterbrechungen auf ein Minimum reduzieren und SMB-Datenverkehr lokalisieren können. Alle Studenten erhalten Unix-Accounts, über die sich der Labor-Server Informationen aus NIS+-Tabellen holt. Alle Studenten haben einen Home-Verzeichnisbereich, der für ihren Account reserviert ist. Das Home-Verzeichnis, das die Studenten für das Labor benutzen, ist jedoch separat von ihrem Unix-Home-Verzeichnisbereich und befindet sich lokal auf dem Labor-Server.

Statt für jeden Benutzer zum Zeitpunkt der Account-Erstellung ein Home-Verzeichnis auf dem Labor-Server einzurichten, haben wir uns entschlossen, es dann einzurichten, wenn der Benutzer sich zum erstenmal in einen Rechner im Labor einloggt. So funktioniert es:

```
[homes]
   comment: PC Lab home directories
   root preexec: /usr/local/samba/bin/buildhome %U %G
```

```
   path = /export/home/%U
   valid user = %S
   create mode = 0600
   directory mode = 0700
```

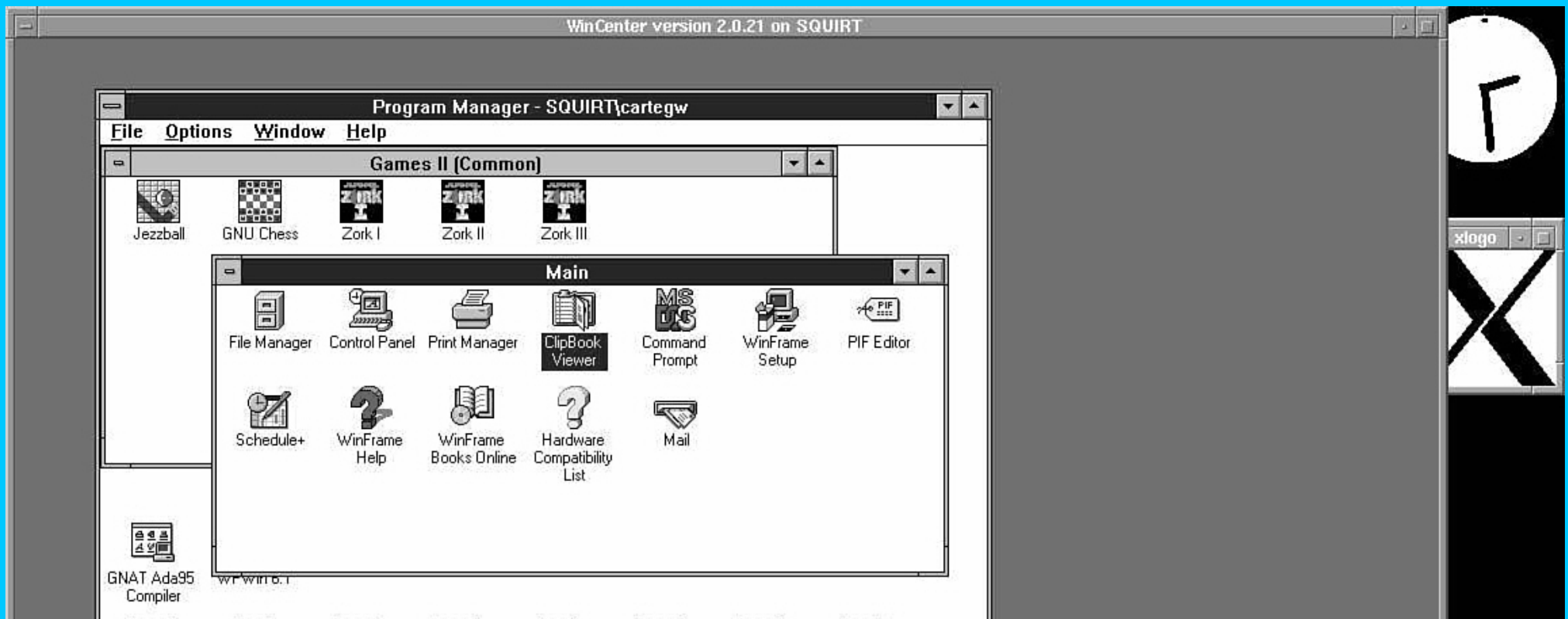
Sie sehen, dass der Pfad nicht das Home-Verzeichnis des Benutzers aus der `/etc/passwd` (d.h. `%H`) ist. Der Pfad ist auf einer lokalen Festplatte, `/export/home`, eingerichtet und jeder Benutzer hat dort ein Verzeichnis. Anfangs ist die Festplatte leer.

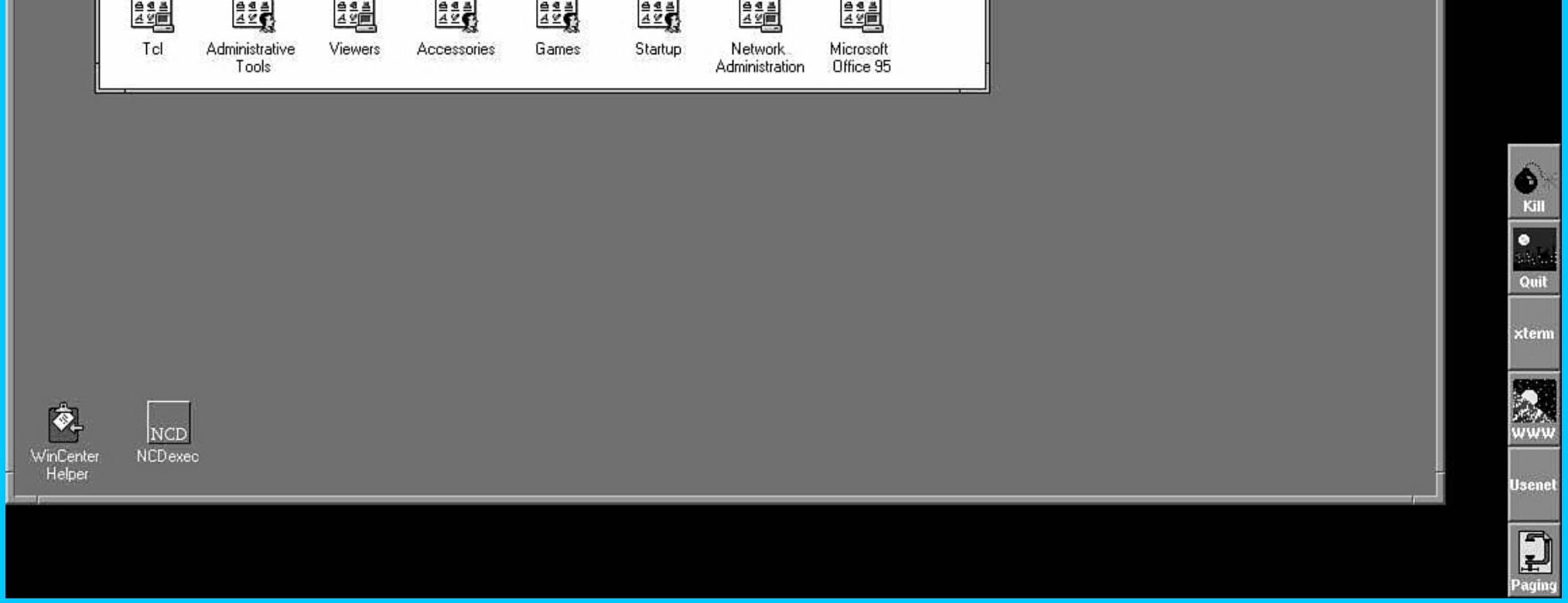
Hier kommt das `root-preexec`-Skript ins Spiel. Das Skript überprüft die Existenz des Verzeichnisnamens `/export/home/`*`Benutzername`* und erstellt es, falls es nicht vorhanden ist. Hier ist der Source-Code für ein einfaches `buildhome`-Skript:

```
#!/bin/sh
umask 077
user=$1
group=$2

# Existiert das Home-Verzeichnis des Benutzers ( export/home/$user )?
if [ ! -d /export/home/$user ]; then
   mkdir /export/home/$user
   chown $user /export/home/$user
   chgrp $group /export/home/$user
fi
```

Hier ist noch ein spaßiges Beispiel für etwas, das Sie mit `postexec`-Skripten machen können. An meinem Arbeitsplatz habe ich Zugriff auf einen älteren NCD-WinCenter-Server. Falls Sie die Cytrix-WinFrame-Multiuser-Version von Windows NT nicht kennen, sie ermöglicht es Benutzern, entfernt auf einen NT-Desktop zuzugreifen, so in etwa wie Benutzer über XDM auf Netzwerk-X-Terminals zugreifen können. Abbildung 10.1 zeigt einen WinCenter-2.0-Desktop direkt neben anderen X-Applikationen, die auf einem Solaris-2.6-Rechner laufen.

Abb. 10.1: NCD-WinCenter-Windows-NT-Desktop auf einer Sun Sparc 10 mit Solaris 2.6





Ich brauchte eine Methode, um von der WinCenter-Sitzung auf das SCSI-CD-ROM-Laufwerk auf meiner Sun Workstation zuzugreifen, also habe ich Samba für die lokale Freigabe der CD-ROM eingerichtet. Hier ist der `smb.conf`-Eintrag:

```
[cdrom]
   comment = 12X CD-Rom
   path = /cdrom/cdrom0
   root postexec = /usr/bin/eject cd
   read only = yes
   public = yes
```

Der Parameter `root preexec` ist nicht unbedingt notwendig, da ich in die Konsole eingeloggt bin, aber er würde grundsätzlich keinen Unterschied machen.

Jetzt kann ich eine CD-ROM einlegen (die durch den Solaris-Volume-Manager-Daemon, `vold`, gemountet wird) und in der WinCenter-Sitzung folgenden Befehl ausführen, damit ich auf die CD-ROM zugreifen kann:

```
net use f: \\mymachine\cdrom
```

Wenn ich die Verbindung über

```
net use f: /d
```

wieder aufhebe, wird die CD-ROM automatisch ausgeworfen.

%U und &u, %G und %g

Sie haben bereits einige Anwendungsbeispiele für Variablen gesehen. So haben Sie z.B. die Variable `%m` für den Wert des Parameters `log file` benutzt, um für Logs auf einer Pro-Rechner-Basis zu sorgen. Wenn Sie Folgendes verwenden, könnten Sie das Gleiche auf einer Pro-Benutzer-Basis erreichen:

```
log file = /usr/local/samba/var/log.%U
```

Bis jetzt habe ich den Unterschied zwischen %u und %U noch nicht richtig erklärt. Beide Variablen werden in einen Benutzernamen erweitert. %U steht dabei für den Benutzernamen, der während des Sitzungsaufbaus übertragen wird. Hier ist z.B. ein Teil der tcpdump-Ausgabe, die ich Ihnen in Kapitel 6, »Sicherheitsmodi und Passwörter«, gezeigt habe. Das Paket wurde von einem Windows-95-OSR2-Client übertragen und enthält das Passwort testpass und den Benutzernamen boss:

```
[000] 54 45 53 54 50 41 53 53  00 00 00 00 00 00 42 4F   TESTPASS ......BO
[010] 53 53 00 00 00 00 00 00  42 4F 53 53 00 43 48 49   SS...... BOSS.CHI
[020] 50 53 4E 44 49 50 53 00  57 69 6E 64 6F 77 73 20   PSNDIPS. Windows
[030] 34 2E 30 00 57 69 6E 64  6F 77 73 20 34 2E 30 00   4.0. Windows 4.0
```

Die Log-Datei für diese Verbindung wäre

```
/usr/local/samba/log.boss
```

Die Variable %u wird in den Benutzernamen der aktuellen Freigabe aufgelöst. Normalerweise sind %u und %U gleich, abgesehen von bestimmten Umständen, z.B. wenn der Parameter force user im User-Modus verwendet wird.

Hier ist ein Beispiel für eine Situation, in der %u und %U nicht gleich sind. Lassen Sie uns diese Freigabedefinition für [src] verwenden:

```
[src]
   root preexec = echo "%T : U is %U and u is %u" >> /var/log/log.src
   comment = /usr/local/src
   path = /usr/local/src
   create mode = 0644
   directory mode = 0755
   force user = jerryc
```

Wie Sie sehen, protokolliert der Wert für preexec einfach die Werte für %U und %u in die Datei. Dies ist der einfachste Weg zu bestimmen, welche Werte Samba benutzt.

smbclient ist ein gutes Test-Tool für diese Art von Experimenten. Ich spezifiziere einfach, dass ich mich mit der Freigabe [src] mit dem Benutzernamen boss verbinden will:

```
/usr/local/samba/bin/smb // bilbo/src -U boss
Added interface ip=192.168.1.73 bcast=192.168.1.255 nmask=255.255.255.0
Password:
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0beta5]
smb: \>
```

Wenn ich mir nun die Ausgabe ansehe, die in /var/log/log.src erstellt wurde, sehe ich, dass %U in den Benutzernamen erweitert wurde, den ich als den verbindenden definiert habe, nämlich boss, und dass der tatsächliche Benutzername, als der ich zugriff, jerryc war:

```
1999/01/09  10:08:02 : U is boss and u is jerryc
```

Die Variablen %G und %g sind direkt mit den Variablen %U und %u verwandt. Der Unterschied zwischen den zwei Gruppenparametern entspricht dem Unterschied zwischen den zwei Parametern für den Benutzernamen.

%L, %m und die include-Parameter

Den Parameter netbios aliases habe ich in Kapitel 5 schon erwähnt, aber ich habe nicht erklärt, warum jemand den gleichen Server in der Browse-Liste unter mehreren Namen aufführen wollte. Wie Sie sich erinnern, wird die Variable %L in den Namen erweitert, den der Client für die Verbindung mit dem Samba-Server benutzt hat.

Über den Parameter include können Sie an jedem Punkt der Konfigurationsdatei lexikalisch Text einfügen. Sie können für den der include-Direktive übergebenen Dateinamen Variablen verwenden. Wenn Sie die Variable %L benutzen, können Sie verschiedene Einstellungen einfügen, die darauf basieren, mit wem sich der Client verbinden wollte. Die Kombination aus Variablen und dem Parameter include bietet große Flexibilität für das Verhalten des Servers auf Basis des Anrufernamens (%m) oder des angerufenen Namens (%L).

include

Wenn Sie schon einmal ein Computerprogramm in C geschrieben haben, sind Sie mit der Präprozessor-Direktive #include *Dateiname* vertraut. Diese Direktive teilt dem Präprozessor mit, den gesamten Text der angegebenen Datei an dem Punkt lexikalisch in den Source-Code einzufügen. Sambas include-Parameter führt die gleiche Funktion aus.

Der Wert des Parameters ist ein Pfad zu einer Datei, deren Inhalte die aktuelle include-Zeile ersetzen werden. Kann Samba die angegebene Datei nicht öffnen, hat der include-Parameter keine

Auswirkung.

Lassen Sie uns diese Beispiel-`smb.conf`-Datei benutzen:

```
; smb.conf
[global]
   netbios name = EAGLE
   workgroup = FOWLPLAY
   security = user
   password level = 4
   include = /usr/local/samba/lib/shares.conf
```

Hier sind die Inhalte von `/usr/samba/lib/shares.conf`:

```
; shares.conf
[foo]
   comment = example disk share
   path = /export/smb/foo
[homes]
   writeable = yes
   valid user = %S
```

Die resultierende Datei nach Analyse wäre:

```
; smb.conf
[global]
   netbios name = EAGLE
   workgroup = FOWLPLAY
   security = user
   password level = 4
   ; shares.conf
[foo]
   comment = example disk share
   path = /export/smb/foo
[homes]
   writeable = yes
   valid user = %S
```

Welchen Unterschied macht dies und warum sollten Sie so etwas benutzen? Nehmen wir an, Sie haben drei Abteilungen: Accounting, Personal und Administration. Nehmen wir weiterhin an, dass jede Abteilung eine Gruppenfreigabe hat, auf die der jeweilige Unix-Server über das *Network File System (NFS)* zugreift, und eine zentrale `passwd`-Datei, die über einen Mechnismus wie `rdist` oder den *Network Information Service (NIS)* verteilt wird. Jeder Unix-Rechner agiert außerdem als Samba-Server für die PCs in der jeweiligen Abteilung. Von Zeit zu Zeit muss eine Person aus der einen Abteilung temporär auf eine Gruppenfreigabe auf einem Rechner in einer anderen Abteilung zugreifen. Die am leichtesten zu verwaltende Lösung bestünde darin, alle Gruppenfreigaben in einer Konfigurationsdatei zu definieren und diese dann während der Laufzeit in die Haupt-`smb.conf`-Datei einzufügen, wie Sie es in den Listings 10.1 und 10.2 sehen.

Listing 10.1: Eine Beispiel-smb.conf-Datei für jeden Abteilungsserver

```
; smb.conf-Datei für die Verwaltung von Gruppenfreigaben über den Parameter include
[global]
   netbios name = <Rechnernamen einfügen>
   workgroup = <Arbeitsgruppennamen der Abteilung einfügen>
   security = user
   password level = 4
[homes]
   comment = <Abteilungsnamen> Home-Verzeichnis
   writeable = yes
   path = /export/home/%U
; Gruppenfreigaben einfügen
   include = /opt/admin/sys/group_shares.conf
```

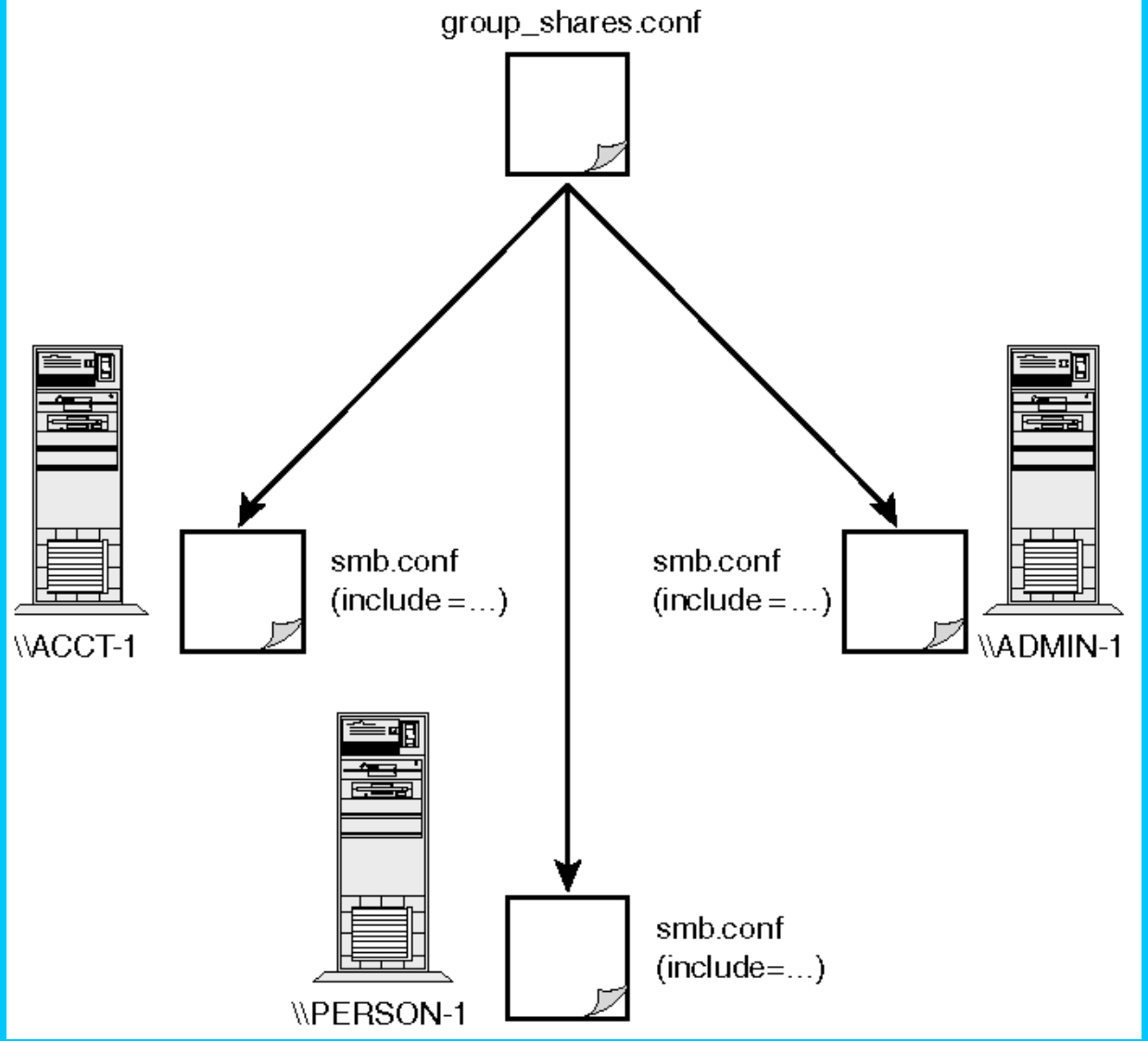
Listing 10.2: Inhalte von group_shares.conf

```
[acctgrp]
   comment = Accounting departmental share
   path = /export/acct/acctgrp
   create mode = 0660
   directory mode = 0770
   valid users = @acct
; Gruppenfreigabe Personal
[persgrp]
   comment = Personnel departmental share
   path = /export/personnel/persgrp
   create mode = 0660
   directory mode = 0770
   valid users = @personnel
; Gruppenfreigabe Administration
[admingrp]
   comment = Administration departmental share
   path = /export/admin/admingrp
   create mode = 0660
   directory mode = 0770
   valid users = @admin
```

Da für alle Server NFS zur Verfügung steht, ist es nur ein kleiner Schritt, ebenfalls NFS-Automount-Unterstützung einzuführen. Jeder Server benutzt diese `smb.conf`-Datei, die auf seine Rechnereinstellungen, wie z.B. den NetBIOS-Namen und den Namen der Arbeitsgruppe, abgestimmt ist. Die Datei `group_shares.conf` befindet sich auf einer Automounted-Freigabe (Solaris verwendet `/opt` als Mount-on-Demand-Punkt, aus Gewohnheit werde ich es also auch benutzen). Diese Umgebung ermöglicht es Ihnen, die Konfigurationsdateien für die Gruppenfreigabe zu ändern und die Änderungen für alle drei Server sichtbar zu machen.

Abbildung 10.2 bietet eine grafische Erklärung der oben aufgelisteten `smb.conf`-Datei. Alle drei Server, ACCT-1, PERSON-1 und ADMIN-1, haben eine lokale Konfigurationsdatei, die der in Listing 10.1 ähnelt. Die Direktive `include` am Ende der Datei teilt Samba mit, den Text aus der Datei `/opt/admin/sys/group_shares.conf` einzufügen, die oben im Diagramm dargestellt ist. Alle Server werden die gleiche Datei einfügen. Wenn also die Definition einer Gruppenfreigabe geändert werden muss, müssen die entsprechenden Änderungen nur in `group_shares.conf` vorgenommen werden, und sie werden dann von allen drei Servern gesehen.

Abb. 10.2: Gruppenfreigaben über den Parameter include verwalten



Variablen in include verwenden

Zwar war das oben stehende Beispiel für die Verwendung eingefügter Dateien völlig in Ordnung, aber es ist wohl üblicher, im eingefügten Dateinamen Variablen zu verwenden, um das Verhalten des Servers auf Basis der Client-Einstellungen zu ändern. Hier ist ein einfaches Beispiel, das Ihnen ermöglicht, verschlüsselte Passwörter für Windows-NT-Clients und Klartextpasswörter für Windows-95-Clients zu verwenden:

```
; smb.conf
[global]
   netbios name = EAGLE
   workgroup = FOWLPLAY
   include = /usr/local/samba/lib/%a.conf
; Freigabedefinition folgt
...
```

Denken Sie daran, dass die Variable %a in den Namen des Betriebssystems des Clients erweitert wird. Sie setzen voraus, dass sich nur Windows-9x- (win95) oder Windows-NT-Clients (winNT) verbinden werden. Hier sind die notwendigen Konfigurationsdateien, die diese zwei Clients unterstützen. Die Windows-95-Konfiguration sieht wie folgt aus:

```
; win95.conf
encrypt password = no
password level = 4
```

Der Windows-NT-Client benutzt folgende Datei:

```
; winNT.conf
encrypt passwords = yes
smb passwd file = /etc/smbpasswd
```

Ich hoffe, dass die Erklärung für die Verwendung des Parameters `netbios aliases` klarer wird, je öfter Sie den `include`-Parameter benutzen. Denken Sie daran, dass die Variable %L in den NetBIOS-Namen des Servers erweitert wird, den der Client während der Sitzungsanfrage verwendet hat. Wenn Sie im Dateinamen für den Parameter `include` die Variable `%L` benutzen, kann der gleiche Rechner als jeweils verschiedener Samba-Server erscheinen.

Gehen wir zurück zu unserem Beispiel der drei Abteilungen. Nehmen wir an, die Bandbreite Ihres Unternehmens wird erhöht, damit Sie einen zentralen Server für alle Abteilungen benutzen können. Wie können Sie den Parameter `include` mit NetBIOS-Aliasen verwenden, um die Änderungen für die Benutzer transparent zu gestalten und Ihre Aufgabe zu vereinfachen?

Zunächst konfigurieren Sie den Samba-Server unter Benutzung seines primären NetBIOS-Namens:

```
; smb.conf
   netbios name = server1
   workgroup = COMPANY-GRP
   security = user
   password level = 4
; Gruppenfreigabe Accounting
[acctgrp]
   comment = Accounting departmental share
   path = /export/acct/acctgrp
   create mode = 0660
   directory mode = 0770
   valid users = @acct
; Gruppenfreigabe Personal
[persgrp]
   comment = Personnel departmental share
   path = /export/personnel/persgrp
   create mode = 0660
   directory mode = 0770
   valid users = @personnel
; Gruppenfreigabe Administration
[admingrp]
   comment = Administration departmental share
   path = /export/admin/admingrp
   create mode = 0660
   directory mode = 0770
   valid users = @admin
```

Dann fügen Sie die Namen der existierenden Abteilungsserver als `netbios aliases` hinzu:

```
netbios aliases = acct-1 person-1 admin-1
```

Danach kopieren Sie die vorhandenen Konfigurationsdateien von den Abteilungsservern und nennen sie `acct-1.conf`, `person-1.conf` bzw. `admin-1.conf`. Sie sollten die Home-Verzeichnisse weiterhin separat halten, wenn Sie sie also auf die Festplatte des neuen Servers verschieben, teilen Sie sie in `/export/acct`, `/export/personel` und `/export/admin` auf. Jetzt müssen Sie Samba mitteilen, die Konfigurationsdatei zu laden, die dem während der Verbindung verwendeten Benutzernamen entspricht:

```
include = /usr/local/samba/lib/%L.conf
```

Die drei Konfigurationsdateien finden Sie in den Listings 10.3, 10.4 und 10.5.

Listing 10.3: Konfigurationseinstellungen für die Abteilung Accounting

```
; acct-1.conf
[homes]
   comment = Accounting home directories
   path = /export/acct/%U
   valid users = %S
[docs]
   comment = department documentation
   path = /export/acct/docs
   writeable = no
```

Listing 10.4: Konfigurationseinstellungen für die Abteilung Personal

```
; person-1.conf
[homes]
   comment = Personel home directories
   path = /export/personel/%U
   valid users = %S
[forms]
   comment = personel forms
   path = /export/personel/forms
```

Listing 10.5: Konfigurationseinstellungen für die Abteilung Administration

```
; admin-1.conf
[homes]
   comment = Administration home directories
   path = /export/admin/%U
   valid users = %S
```

Abbildung 10.3 zeigt die Resultate, wenn Sie das Netzwerk browsen. Physisch sind nur zwei Rechner verfügbar. `QUESO` ist ein Windows-95-Client und `SERVER1` ein Linux-Rechner. Die anderen drei Einträge - `ACCT-1`, `ADMIN-1` und `PERSON-1` - werden durch den Parameter `netbios aliases` in der `smb.conf`-Datei erstellt. Die Abbildungen 10.4, 10.5 und 10.6 zeigen die Freigaben, die jeder Server zur Verfügung stellt. Beachten Sie, dass jeder Server etwas unterschiedlich ist, aber alle haben die gemeinsamen Gruppenfreigaben.

Abb. 10.3: Browsing für jede der drei Konfigurationen des Samba-Servers. QUESO ist der Windows-95-Rechner, der für das Browsing des Netzwerk benutzt wird. SERVER1 ist der primäre NetBIOS-Name des Servers

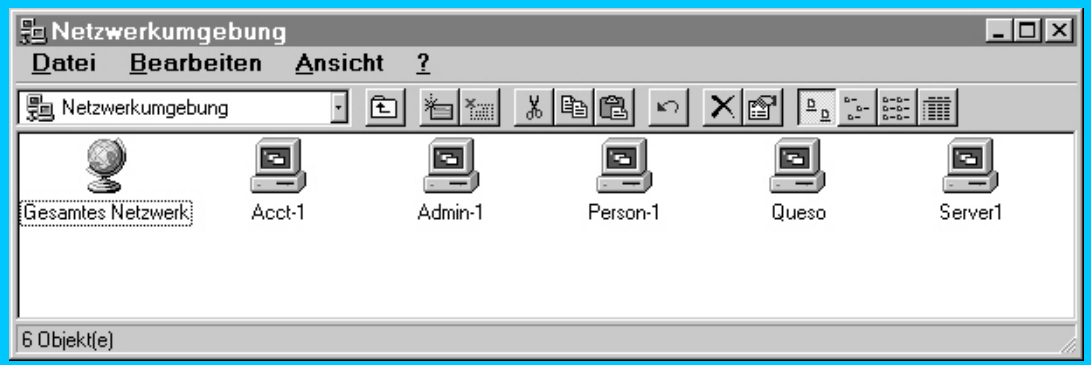


Abb. 10.4: Browsing der Freigaben auf ACCT-1

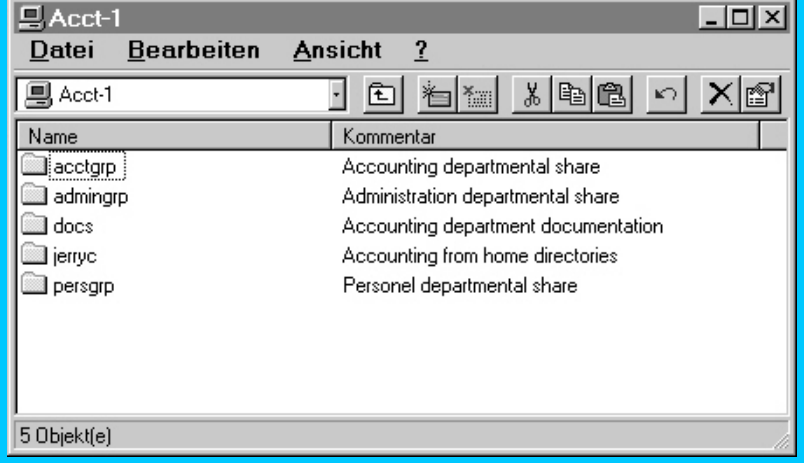


Abb. 10.5: Browsing der Freigaben auf ADMIN-1

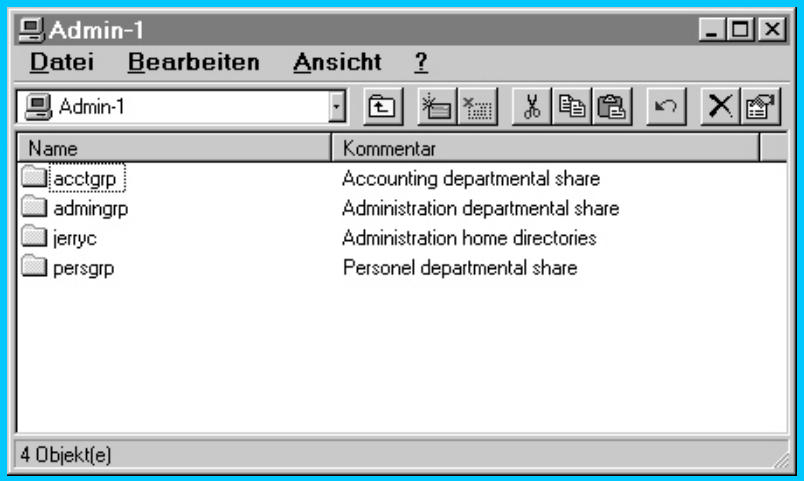
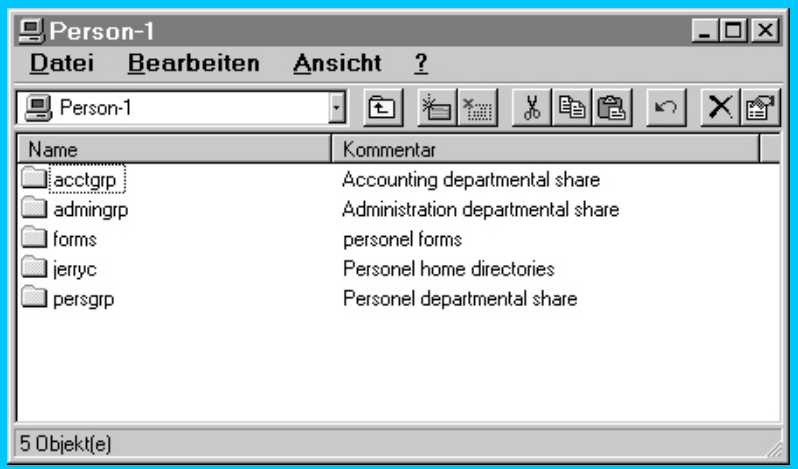


Abb. 10.6: Browsing der Freigaben auf PERSON-1



Vielleicht haben Sie einen Fehlerpunkt bereits entdeckt. Ich sagte, dass der Übergang für den Benutzer transparent sein sollte, aber wir sind von drei Arbeitsgruppen zu einer übergegangen. Ich gebe zu, dass ich dies ausgelassen habe. Es gibt derzeit keine Möglichkeit, Samba an mehr als einer Arbeitsgruppe gleichzeitig teilnehmen zu lassen. Daher setzte ich für unser Beispiel voraus, dass die Benutzer direkt über UNC-Netzwerkpfade in der Form von `\\Servername\Freigabename` auf die entsprechenden Server zugreifen.

%d

Ich habe diesen letzten Abschnitt in der Hoffnung eingefügt, dass er Ihre Kreativität anregen wird. Es kann schwierig sein festzustellen, welche `smbd`-Prozesse mit welchen Benutzern verbunden sind. Kapitel 4, »Installation und Testen der Konfiguration«, beschreibt, wie Sie über das Tool `smbstatus` diese Informationen erhalten können. In diesem Abschnitt zeige ich Ihnen, wie Sie `root-preexec`- und `-postexec`-Skripte verwenden, um die Möglichkeit zu implementieren, alle `smbd`-Prozesse zu beenden, die im Zusammenhang mit einem bestimmten Benutzer laufen. Die Methode umfasst drei grundlegende Teile:

- Die Befehle `root preexec` und `root postexec`, die in `smb.conf` spezifiziert sind.
- Das Verzeichnis, das die Ausgabedateien der `root-preexec`- und `root-postexec`-Skripte enthält.
- Das Shell-Skript, das die Prozesse tatsächlich beendet.

Ich werde jeden dieser Teile nacheinander darstellen. Zunächst untersuche ich die Befehle `root preexec` und `root postexec`. Ich möchte eine Prozess-ID mit einem Benutzernamen verbinden. Diese Information ist über die Variablen `%d` bzw. `%U` verfügbar. Der `root-preexec`-Befehl erzeugt eine Datei mit dem Namen der Prozess-ID für die Verbindung und enthält nur den durch den Client übertragenen Benutzernamen:

```
[apps]
   comment = global share drive
   root preexec = echo %U > /usr/local/samba/lib/proc/%d
   root postexec = /bin/rm /usr/local/samba/lib/proc/%d
   path = /export/smb/apps
```

Die Voraussetzung hierbei ist, dass sich jeder Client zusätzlich zu anderen Freigaben mit der Freigabe `[apps]` verbindet. Diese Voraussetzung können Sie einfach umgehen, indem Sie die Existenz von `/usr/local/samba/lib/proc/%d` überprüfen, bevor Sie versuchen, sie zu erzeugen. Für den gegenwärtigen Zeitpunkt setzen wir einfach voraus, dass sich alle Clients mit der Freigabe `[apps]` verbinden.

Nach Erzeugen der Dateien können Sie ein einfaches Skript erstellen, das nach dem Login-Namen des Benutzers in den Dateien sucht. Die gefundenen Dateien sind dann die Prozess-IDs der entsprechenden `smbd`s.

```
# !/bin/sh
# den Pfad einrichten
PATH=/usr/bin:/bin
export PATH
# Standort für die pid-Dateien einrichten
SMBD_DIR=/usr/local/samba/lib/proc
# der Benutzername sollte als der einzelne Befehlszeilenparameter übergeben werden
if [ "$1 eq "" ]; then
   echo 'Usage : killsmbd <username>'
else
   username=$1
fi
for pid in `grep $username $SMBD_DIR/*` ; do
   echo "Killing $pid"
   kill -9 $pid
done
```

Wenn Sie alle `smbd`-Prozesse für den Benutzer `jdoe` beenden wollen, können Sie einfach Folgendes eingeben:

```
killsmbd jdoe
```

Sie können ähnliche Methoden entwickeln, um `smbd`-Prozesse auf Basis des verbundenen Rechnernamens zu beenden, indem Sie Dateien erzeugen, die die Variable `%m` beinhalten.

Zusammenfassung

Samba bietet viele Tools für die Automatisierung von Aktivitäten auf der Server-Seite einer SMB-Verbindung. Wenn Sie die Parameter `preexec` und `postexec`, `smb.conf`-Variablen und die Direktive `include` verwenden, können Sie Ihren Server für ein sehr dynamisches Verhalten konfigurieren, das den Bedürfnissen Ihrer Clients entspricht.

Wird der Parameter `netbios aliases` in Verbindung mit dem Parameter `include` verwendet, haben Sie eine Methode zum Klonen mehrerer Samba-Server auf einem einzelnen Rechner.

Frage & Antwort

F. Kann eine Freigabe sowohl eine Einstellung für root preexec als auch für preexec haben?

. Ja, es ist möglich, für eine Freigabe sowohl eine `root-preexec`- als auch eine `preexec`-Einstellung einzurichten. Gleiches gilt für `root postexec` und `postexec`.

F. Können die Variablen %U, %u, %G und %g für den Parameter include benutzt werden?

. Ja. Damit könnten Sie eine Einstellung auf individueller oder auf Gruppenbasis einrichten. Ein Beispiel wäre das Einfügen einer Abteilungsfreigabe über `include = %G`, wobei `%G` die Definition für eine Freigabe enthalten würde.

Neue Begriffe

Automatisierung - Der Prozess, einen Job oder eine Aufgabe ohne menschliches Zutun zu erledigen.

Automatisierung auf Server-Seite - Die Dinge, die am Server-Ende einer Netzwerkverbindung eintreten und automatisiert sind. Beispiele wären die automatische Erstellung von Freigaben oder eine Neukonfiguration auf Basis des verbindenden Clients.

Tag 11: Troubleshooting

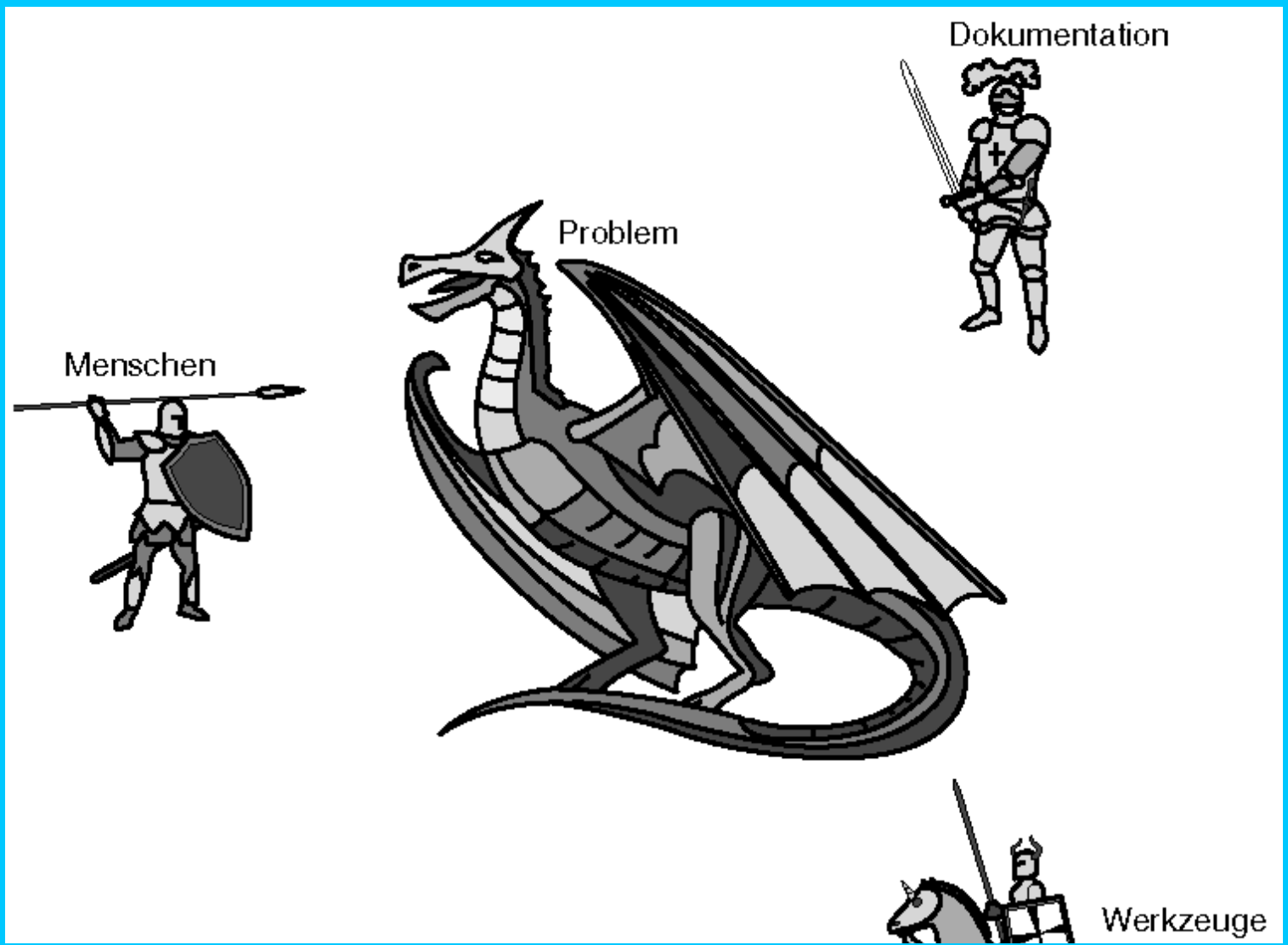
Egal, ob wir Samba zum ersten Mal konfigurieren, von einer früheren Version aktualisieren oder etwas einfach nicht funktioniert, manchmal brauchen wir alle ein wenig Hilfe. Vielleicht denken Sie, dass dies Kapitel 2 sein sollte (oder vielleicht ist es auch das zweite Kapitel, das Sie lesen). Ich habe mit der Darstellung der Fehlerbehandlung gewartet, bis Sie genügend Funktionen kennen gelernt haben, damit ich nicht erst Konzepte erklären muss, während ich versuche darzustellen, wie man Fehler in Verbindungen oder Konfigurationen korrigieren kann.

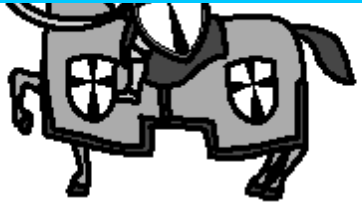
Das Hauptproblem der meisten Kapitel über Troubleshooting besteht darin, dass sie von Ihnen das verlangen, was ich als »eine kritische Masse« von Hintergrundinformationen bezeichne. Der Zeitpunkt, zu dem Sie Troubleshooting am dringendsten benötigen, liegt vor der Zeit, zu der Sie genügend Informationen haben, um die Dinge selbst zu lösen. Anscheinend ist dies eine Variante des Problems mit dem Huhn und dem Ei.

Aus diesem Grund glaube ich, dass die Behandlung jedes Problems irgendwo zwischen Kunst und Wissenschaft liegt. Sie brauchen ein gewisses Verständnis für die Grundlagen des Problems, aber meistens ist es ein »Heureka!«-Moment, der Sie dazu bringt, von Problem zu Lösung zu springen.

Troubleshooting beinhaltet oft das Betrachten einer Situation aus verschiedenen Blickwinkeln, wie in Abbildung 11.1 dargestellt. Die Informationen in diesem Kapitel sind in Schnipsel oder Blöcke aufgeteilt, damit Sie so viele Tools und Blickwinkel wie möglich kennen lernen können. Jede Methode stellt eine andere Facette einer umfassenderen Technik dar.

Abb. 11.1: Troubleshooting beinhaltet oft das Angehen des Problems aus verschiedenen Blickwinkeln, um seine Ursache einzuengen





Eines der wundervollen Dinge an einem Open-Source-Projekt wie Samba ist, dass potentiell Hunderte (oder Tausende) von Leuten einen Beitrag zur Dokumentation oder dem Source-Code leisten können. Zwar stellt dies ein kleines Problem dar, was die Verwaltung einer organisierten Verteilung betrifft, aber es bietet eine enorme Menge an Erfahrung, von der Sie profitieren können. Die Chancen stehen gut, dass jemand Ihr Problem zumindest kennt, wenn nicht gar bereits gelöst hat. Auch wenn bisher noch niemand auf Ihr spezielles Problem getroffen ist, sind die meisten bereit, Ihnen zu helfen.

Zufälligerweise arbeite ich gerade jetzt, da ich dieses Kapitel schreibe, mit einem anderen Netzwerkadministrator in einer Mailing-Liste zusammen, um eine Lösung für ein Konfigurationsproblem mit seinem Server zu finden. Dies scheint mir ein guter Startpunkt!

Dokumentation

Generell sind die Leute, die Dokumentationen schreiben, wie Sie. Diese Leute sind entweder Benutzer oder Systemadministratoren, die schnell Informationen und Antworten finden müssen. Aus diesem Grund besteht die meiste Dokumentation, abgesehen von Manpages, aus kurzen, ein bis zwei Seiten umfassenden, Erklärungen einer bestimmten Sache und einer Sammlung häufig gestellter Fragen.

Bevor Sie eine Ladung von Fragen und Postings in verschiedene Mailing-Listen oder Usenet-News-Gruppen schicken, in denen Sie um Hilfe bitten, denken Sie an Folgendes: Sie werden in der Regel bessere Antworten von anderen erhalten, wenn Sie einige Zeit für eigene Recherchen investiert haben. Erwarten Sie nicht von anderen, dass sie das Problem für Sie lösen. Dies hört sich sehr nach etwas an, was ein Psychologe sagen würde, oder?

DIAGNOSIS.txt

Wenn Sie eine Frage oder einen Hilferuf in ein beliebiges Kommunikationsmedium wie z.B. eine Mailing-Liste oder eine Newsgroup setzen, erhalten Sie als Antwort in der Regel eine andere Frage: »Haben Sie die Schritte in `DIAGNOSIS.txt` durchgeführt?« Wenn mich jemand in meinem Büro anruft und mir sagt, dass irgendeine Anwendung oder Netzwerkfunktion auf seinem PC nicht läuft, ist meine erste Reaktion immer: »Haben Sie Ihren PC neu gestartet? Wenn nicht, tun Sie das und rufen Sie mich zurück, wenn es immer noch nicht funktioniert.« Es gibt einige generelle Schritte, die Sie durchführen sollten, um dem Problem auf den Grund zu kommen.

Die Textdatei `DIAGNOSIS.txt` befindet sich im Verzeichnis `/docs/textdocs` für Samba 2.0 oder höher. Wenn Sie eine Samba-Version verwenden, die mit dem Betriebssystem verteilt wurde - Linux z.B. -, schauen Sie in das Verzeichnis `/usr/doc/samba`. Die Datei beschreibt einen Prozess für die Fehlerbehandlung Ihres Servers, der aus zehn Schritten besteht. Die Schritte bauen aufeinander auf und sollten nacheinander ausgeführt werden.

Der Sinn und Zweck hinter `DIAGNOSIS.txt` liegt in der Behebung von Verbindungsproblemen zwischen einem Client und dem Server. Dieser Diagnoseprozess verlangt einige Voraussetzungen:

- Sie haben Samba installiert und wollen eine Startkonfiguration testen.
- Sie haben Zugriff auf einen PC, auf dem irgendeine Version von Windows und TCP/IP läuft. Wenn Sie Windows 95/98 benutzen, muss außerdem der Microsoft Client für Windows installiert sein. Windows für Workgroups bezeichnet diesen Client als Unterstützung für Microsoft-Netzwerke, während Windows NT einen Arbeitsstationsdienst installiert, der die SMB-Client-Funktion handhabt.
- Der Samba-Server hat eine Freigabe namens `[tmp]` mit einem auf `/tmp` eingerichteten Pfad. Sie können diese Freigabe erstellen, indem Sie Ihrer vorhandenen `smb.conf` folgenden Abschnitt hinzufügen:

```
[tmp]
comment = Temporäre Nur-Lese-Freigabe
path = /tmp
writeable = no
```

Ich habe entschieden, keine komplette `smb.conf` als Beispiel einzufügen, da Sie sicherlich eher daran interessiert sind, die Konfiguration für Ihren bestimmten Server zu testen. Daher ermöglichen Ihnen die beschriebenen Schritte und Beispiele, Ihren Server in den Testprozess zu integrieren. Zu Ihrer Kenntnisnahme: ich benutze für diese Beispiele den Servernamen `BILBO` und einen Client-Rechner namens `QUESO`.

Schritt 1: smb.conf testen

Als Erstes sollten Sie über das Utility `testparm` sicherstellen, dass es in Ihrer `smb.conf` keine Syntaxfehler gibt, wie Sie es in Kapitel 4, »Installation und Testen der Konfiguration«, bereits gemacht haben. `testparm` generiert eine recht umfangreiche Ausgabe, da es sowohl die

Standardwerte für Parameter als auch die von Ihnen speziell eingerichteten Parameter anzeigt. Mit dem folgenden Befehl können Sie sich die Ausgabe nacheinander am Bildschirm ansehen:

```
testparm /etc/smb.conf |more
```

Sie sollten `/etc/smb.conf` dabei mit dem Standort der Konfigurationsdatei ersetzen, die Sie testen wollen. Sind Fehler vorhanden, werden sie am Anfang der Ausgabe dargestellt. Nachfolgend sehen Sie den `[global]`-Abschnitt einer Beispiel-`smb.conf`, die ich mit `testparm` überprüft habe:

```
[global]
   ; SMB-Einstellungen
   netbios name = BILBO
   workgroup = FOWLPLAY
   server string = Samba server [%v]

   ; Server-Einstellungen
   security = user
   hosts allow = 192.168.1.
   log file = /usr/local/samba/var/log.%m

   ;Passworteinstellungen
   password level = 4

   ; Standardfreigabe-Einstellungen
   lcking = no
   case sensitive = no
   public = guest
   writeable = no
```

Hier sind die ersten paar Zeilen, die `testparm` ausgibt:

```
[root@bilbo /root]539: /usr/local/samba/bin/testparm smb.conf- | more
Load smb config files from smb.conf-
Unknown parameter encountered: "lcking"
Ignoring unknown parameter "lcking"
ERROR: Badly formed boolean in configuration file: "guest".
Processing section "[netlogon]"
Processing section "[homes]"
Processing section "[src]"
Loaded services file OK.
Press enter to see a dump of your service definitions
# Global parameters
   workgroup = FOWLPLAY
   netbios name = BILBO
```

Der erste ausgegebene Fehler ist die falsche Schreibweise von `locking` (`lcking`) und der zweite der ungültige Wert, den ich dem Parameter `public` zuweisen wollte.

Schritt 2: IP-Konnektivität überprüfen

Nachdem Sie sichergestellt haben, dass die Syntax Ihrer `smb.conf` korrekt ist, werden Sie als nächsten Schritt überprüfen, ob Client und Server sich gegenseitig Pakete über IP senden können. Verwenden Sie zunächst den Befehl `ping`, um zu testen, ob der Server den Client »sehen« kann. Befindet sich `ping` nicht in Ihrem normalen `$PATH`, finden Sie den Befehl in der Regel in `/usr/sbin/`, `/bin` oder `/usr/bin`:

```
jerryc$ ping queso
PING queso (192.168.1.72): 56 data bytes
64 bytes from 192.168.1.72: icmp_seq=0 ttl=128 time=0.8 ms
64 bytes from 192.168.1.72: icmp_seq=1 ttl=128 time=0.8 ms
64 bytes from 192.168.1.72: icmp_seq=2 ttl=128 time=0.8 ms
64 bytes from 192.168.1.72: icmp_seq=3 ttl=128 time=0.8 ms

--- queso ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.8/0.8/0.8 ms
```

Den ping-Befehl können Sie mit der Tastenkombination [Strg]-[C] beenden.

Einige ping-Versionen sind standardmäßig wortreicher als andere. Der Befehl /usr/sbin/ping, der in Solaris 2.6 integriert ist, teilt Ihnen einfach nur mit, ob er überhaupt eine Antwort erhalten hat:

```
jerryc$ ping sunspot
sunspot.my.net is alive
```

Die Art der Ausgabe ist nicht so wichtig, solange Sie bestimmen können, dass der Server den Client erreichen kann.

Versuchen Sie nun, ping in die andere Richtung zu benutzen, vom Client zum Server. Das Tool ping.exe befindet sich normalerweise im Verzeichnis \windows\system und sollte daher standardmäßig in Ihrem Pfad enthalten sein:

```
C:\users\jerry>ping bilbo

Pinging bilbo [192.168.1.73] with 32 bytes of data:

Reply from 192.168.1.73: bytes=32 time=1ms TTL=64
Reply from 192.168.1.73: bytes=32 time=1ms TTL=64
Reply from 192.168.1.73: bytes=32 time=1ms TTL=64
Reply from 192.168.1.73: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.73:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 1ms, Maximum = 1 ms, Average = 1ms
```

Hat ein Rechner Probleme mit der Auflösung der Hostnamen in IP-Adressen, sollten Sie Ihre DNS-Einstellungen überprüfen. Unix-Derivate speichern die Liste der Name Server normalerweise in /etc/resolv.conf, während Sie bei Windows-Clients nur die IP-Adressen der DNS-Server über die TCP/IP-Protokolleinstellung einrichten können. Wenn nötig, sollten Sie sicherstellen, dass die Nameserver funktionieren und erreichbar sind. Dies kann z.B. dadurch erreicht werden, dass Sie einen Ping an die IP-Adresse des Nameservers schicken.

Schritt 3: smbd überprüfen

Für die ersten zwei Schritte war es nicht notwendig, die Samba-Daemons zu starten (oder für den Start bereitzuhalten, falls Sie die Prozesse von /etc/inetd.conf laufen lassen). Für diesen und die folgenden Schritte müssen Sie sowohl smbd als auch nmbd starten.

Wenn Sie sicher sind, dass die Samba-Daemons laufen oder bei Verbindungsaufnahme gestartet werden, holen Sie sich über smbclient eine Liste der Freigaben auf dem Server, wie Sie es bereits vorher einmal über den Befehl smbclient -L *Servername* -N getan haben:

```
jerryc$ smbclient -L bilbo -N
Added interface ip=192.168.1.73 bcast=192.168.1.255 nmask=255.255.255.0
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0]

       Sharename    Type    Comment
       ---------    ----    -------
       src          Disk    /usr/local/src
       tmp          Disk    Temporary Read-Only share
       IPC$         IPC     IPC Service (Samba server [2.0.0])

       Server               Comment
       ------               -------
       BILBO                Samba server [2.0.0]

       Workgroup            Master
       ---------            ------
       FOWLPLAY             BILBO
```

Läuft der smbd-Daemon nicht oder kann er aus irgendeinem Grund nicht auf TCP port 139 gesetzt werden, sehen Sie eine Meldung, die der folgenden ähnelt:

```
jerryc$ smbclient -L bilbo -N
Added interface ip=192.168.1.73 bcast=192.168.1.255 nmask=255.255.255.0
error connecting to 192.168.1.73:139 (Connection refused)
Connection to bilbo failed
```

Wenn Sie sich mit dem Server nicht verbinden können, weil Ihre Parameter `hosts allow` oder `hosts deny` falsch konfiguriert sind, teilt `smbclient` Ihnen mit, dass der Server aktiv ist, aber die Sitzung ablehnt:

```
jerryc$ smbclient -L bilbo -N
Added interface ip=192.168.1.73 bcast=192.168.1.255 nmask=255.255.255.0
session request to BILBO failed
session request to *SMBSERVER failed
```

Dies ist nicht der einzige mögliche Grund für das Scheitern einer Sitzungsaufnahme, aber der üblichste.

Wenn Sie Probleme haben, sollten Sie auch sicherstellen, dass die `subnet`-Maske und die Broadcast-Adresse sowohl auf dem Client als auch auf dem Server korrekt eingestellt sind. Samba versucht, diese automatisch zu bestimmen, aber es ist möglich, dass es dies nicht schafft. Sie können die Adresse und die Netzmaske, die Samba benutzen sollte, explizit über den Parameter `interfaces` einrichten. Ich werde diesen Parameter ausführlich in Kapitel 20, »Browsing in Netzwerken mit Routern«, darstellen. Das Format für den Wert des Parameters ist eine Kombination aus IP-Adresse und Netzmaske, z.B.:

```
interfaces = 192.168.1.73/255.255.255.224
```

Wenn Sie es vorziehen, können Sie die Netzmaske als dezimale Zahl darstellen, die die Anzahl der zu verwendenden Bits bezeichnet. Denken Sie daran, dass ein logisches `AND` mit den Bits einer Netzmaske und der IP-Adresse des Rechners durchgeführt wird, um die Netzwerkadresse zu bestimmen. Das folgende Beispiel entspricht der vorherigen Einstellung:

```
interfaces = 192.168.1.73/27
```

Schritt 4: nmbd überprüfen

Jetzt ist der Zeitpunkt gekommen, an dem Sie überprüfen, ob der `nmbd` korrekt installiert wurde. Benutzen Sie das Utility `nmblookup`, um zu versuchen, den NetBIOS-Namen des Servers aufzulösen. Der folgende Befehl sollte die IP-Adresse des Samba-Servers zurückgeben:

```
nmblookup -B Servername __SAMBA__
```

Die Angabe `-B Servername` teilt Samba mit, die IP-Adresse des Servernamens als Broadcast-Adresse und `__SAMBA__` als aufzulösenden NetBIOS-Namen zu verwenden. Dies ist ein spezieller Name, auf den nur Samba-Server reagieren werden. Sie sollten das Argument *`Servername`* durch den NetBIOS-Namen Ihres Samba-Servers ersetzen. Ein Beispiel:

```
jerryc$ nmblookup -B BILBO __SAMBA__
Sending queries to 192.168.1.73
192.168.1.73 __SAMBA__<00>
```

Wenn `nmblookup` die IP-Adresse Ihres Servers nicht zurückgibt, liegt es wahrscheinlich daran, dass der `nmbd` nicht korrekt installiert wurde. Wenn Sie den `smbd` und `nmbd` von `inetd.conf` starten, stellen Sie sicher, dass alle Befehlszeilenparameter, die Sie an den `nmbd` weiterleiten, tatsächlich während des Starts verwendet werden. Einige `inetd`-Implementierungen schränken die Anzahl der Parameter ein, die in der Befehlszeile an eine Anwendung weitergeleitet werden können. Wenn der `nmbd` nicht alle Befehlszeilenparameter erkennen kann, denken Sie darüber nach, ein Skript zu schreiben, das den `nmbd` startet, und dieses Skript vom `inetd` starten zu lassen.

Schritt 5: Die Client-Software auf dem PC überprüfen

Nachdem Sie sichergestellt haben, dass `smbd` und `nmbd` installiert sind und laufen, überprüfen Sie den Status der installierten Client-Software auf dem PC. Dieser Schritt ist Schritt 4 sehr ähnlich. Verwenden Sie wieder `nmblookup`, um das NetBIOS-Interface auf dem Client zu überprüfen:

```
nmblookup -B Client-Name '*'
```

Dieser Befehl benutzt die IP-Adresse des Client-Rechners, um die Anfrage für einen beliebigen Namen per Broadcast zu übertragen. Die Ausgabe sollte die IP-Adresse des Client-PCs anzeigen. Ist das nicht der Fall, überprüfen Sie noch einmal, ob die TCP/IP-Einstellungen für den Client korrekt sind, und stellen Sie sicher, dass, im Falle von Windows 9x und Windows für Workgroups der Client für Microsoft-Netzwerke installiert ist:

```
jerryc$ nmblookup -B queso '*'
Sending queries to 192.168.1.72
192.168.1.72 *<00>
```

Wenn Sie einen Windows-NT-Rechner benutzen, stellen Sie sicher, dass die Dienste Server und Arbeitsstation laufen und dass die NetBIOS-Schnittstelle mit TCP/IP verbunden ist. Weitere Details über die Konfiguration von Windows-Clients finden Sie in Kapitel 14, »Windows 9x und Windows NT«.

Schritt 6: Die Broadcast-Adresse überprüfen

Überprüfen Sie als Nächstes, ob die konfigurierte Broadcast-Adresse korrekt eingerichtet ist. Denken Sie daran, dass viele der Anfragen für

die NetBIOS-Namenregistrierung und -auflösung standardmäßig per Broadcast übertragen werden; daher sollten Sie sicherstellen, dass die Adresse korrekt ist.

Ein Befehlszeilenargument, das Sie bisher noch nicht mit `nmblookup` verwendet haben, ist der Parameter `-d debug level`. Dieser Parameter führt hier die gleiche Funktion aus wie im Zusammenhang mit `smbd` und `nmbd`. Der einzige Unterschied liegt darin, dass die `debug`-Ausgabe in den Standard-Output statt in die Logdateien geschrieben wird. Das eingefügte Wildcard-Zeichen (*) bedeutet, dass `nmblookup` Broadcast-Anfragen an alle Namen im lokalen Broadcast-Subnetz übertragen soll:

```
jerryc$ nmblookup -d 2 '*'
Added interface ip=192.168.1.73 bcast=192.168.1.255 nmask=255.255.255.0
Sending queries to 192.168.1.255
Got a positive name query response from 192.168.1.72 (192.168.1.72)
Got a positive name query response from 192.168.1.73 (192.168.1.73)
192.168.1.72 *<00>
192.168.1.73 *<00>
```

Ist die Broadcast-Adresse korrekt konfiguriert, sollten Sie mehrere Meldungen mit dem Inhalt »Got a positive name query response from ...« sehen, auch wenn Sie nur zwei Rechner im Netzwerk haben, so wie es hier bei mir der Fall ist. Die tatsächliche Anzahl der Antworten ist unwichtig, solange Sie eine von einem anderen Rechner als dem Server erhalten.

Wenn Sie keine dem Beispiel ähnliche Ausgabe sehen, müssen Sie möglicherweise mit dem Parameter `interfaces` in der `smb.conf` experimentieren, um das Interface und die Netzmaske manuell einzurichten, die für den `smbd` und `nmbd` relevant sind. Haben Sie mehr als ein Netzwerk-Interface, ist Samba standardmäßig nur mit dem ersten verbunden.

Schritt 7: Lokale Verbindung mit einer Freigabe

Jetzt können Sie die Sicherheitsoptionen überprüfen, die Sie in Ihrer `smb.conf` konfiguriert haben. Damit meine ich, dass Sie die Passworteinstellungen überprüfen, um sicherzustellen, dass sich ein Benutzer mit einer Freigabe verbinden kann.

Wenn Sie es noch nicht getan haben, sollten Sie jetzt sicherstellen, dass Ihre `smb.conf` die Freigabe `[tmp]` enthält, das Verzeichnis `/tmp` existiert und es allgemein lesbar ist. Dann versuchen Sie, wieder über `smbclient`, sich über einen gültigen Account mit der Freigabe `[tmp]` zu verbinden:

```
jerryc$ smbclient '\\bilbo\tmp' -U jerryc
Added interface ip=192.168.1.73 bcast=192.168.1.255 nmask=255.255.255.0
Password: geben Sie hier Ihr Passwort ein
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0]
smb: \> dir
   .X11-unix         DH      0    Tue Jan 12 20:11:12 1999
   log.nmb.nmb              305    Wed Oct 14 01:03:44 1998
   .ICE-unix         DH      0    Tue Jan 12 20:21:54 1999
   .X0-lock          HR     11    Tue Jan 12 20:11:11 1999

      61967 blocks of size 4096. 14548 blocks available
```

Ich möchte hier eine Anmerkung zur Verwendung von nach vorn bzw. nach hinten geneigten Schrägstrichen (Slash und Backslash) (/ und \) machen. SMB-Freigaben werden mit Ihrem *Universal-Naming-Convention- (UNC-)*Namen, d.h. `\\Servername\Freigabename`, bezeichnet. Unix-Befehle analysieren einen Backslash (\) normalerweise als Escape-Zeichen. Daher müssen Sie, wenn Sie für den UNC-Namen Backslashes verwenden wollen, den Pfad entweder in einzelne oder doppelte Anführungszeichen einschließen. Alternativ ermöglicht Ihnen `smbclient`, stattdessen Slashes zu verwenden, z.B. `//bilbo/tmp`. Es bleibt Ihrer persönlichen Präferenz überlassen, welche Methode Sie wählen.

Gibt `smbclient` eine Fehlermeldung wie »Invalid tree in network connect« zurück, stellen Sie sicher, dass der in `smb.conf` definierte Pfad für die Freigabe, mit der Sie sich verbinden wollen, gültig ist. Überprüfen Sie außerdem, ob der Benutzer die entsprechenden Berechtigungen (Lesen oder Schreiben, was auch immer angemessen ist) für den Zugriff auf das Verzeichnis hat.

Bezieht sich die zurückgegebene Fehlermeldung auf einen ungültigen Login-Namen oder ein ungültiges Passwort, stellen Sie sicher, dass Sie

das Passwort korrekt eingegeben haben. Sie können außerdem über das Befehlszeilenargument `-U Benutzername` den Benutzernamen explizit einrichten, den `smbclient` während der Sitzungsanfrage benutzen soll. Weitere Probleme können sein: die Aktivierung verschlüsselter Passwörter, aber kein gültiger `private/smbpasswd`-Eintrag für den Benutzer, falsche Einstellungen für den Parameter `valid users` für die Freigabe oder eine Einstellung für den `password level`, die für die Anzahl der Großbuchstaben im Passwort zu niedrig ist.

Schritt 8: Den Server von einem DOS-Prompt browsen

Jetzt prüfen Sie, ob der PC-Client eine Liste der Freigaben vom Server erhalten kann. Versuchen Sie den folgenden Befehl von einem DOS-Prompt unter Windows:

```
net view \\Servername
```

Ersetzen Sie *Servername* durch den NetBIOS-Namen Ihres Samba-Servers. Hier ist die generierte Ausgabe nach Ausführung dieses Schrittes auf meinem Test-Server:

```
C:\users\jerry>net view \\bilbo
Freigegebene Ressourcen auf \\BILBO

Name        Typ       Lokal     Beschreibung
-------------------------------------------------------------
jerryc      Platte              Linux home directories
scr         Platte              /usr/local/src
tmp         Platte              Temporaty Read-Only share
Der Befehl wurde erfolgreich ausgeführt.
```

Wenn Sie versuchen, von einem Windows-NT-Client zu browsen, aber auf dem Server verschlüsselte Passwörter nicht aktiviert haben, werden Sie wahrscheinlich eine Meldung sehen, die besagt »Zugriff wurde verweigert«. Das liegt daran, dass Windows NT Unterstützung für verschlüsselte Passwörter verlangt, um den Server zu browsen, aber auch das kann problematisch sein. Dies wird Sie jedoch nicht daran hindern, sich mit einer bestimmten Freigabe auf dem Server zu verbinden, wie Sie im nächsten Schritt sehen werden.

Wenn Sie folgende Fehlermeldung erhalten

```
Computer ist nicht verfügbar. Mit diesem Konto kann man sich nicht von dieser Station aus anmelden.
```

sollten Sie sicherstellen, dass Sie den Zugriff auf den Server nicht durch eine `hosts-deny`/`hosts-allow`-Einstellung in Ihrer `smb.conf` oder über ein Programm wie TCP Wrappers eingeschränkt haben.

Diese Fehlermeldung

```
Der Computer ist nicht verfügbar oder der Netzwerkpfad konnte nicht gefunden werden.
```

bedeutet, dass der PC den NetBIOS-Namen nicht auflösen konnte. Dies können Sie beheben, indem Sie entweder die `nmbd`-Installation auf dem Server korrigieren oder einen anderen Mechanismus für die Namenauflösung konfigurieren, z.B. `lmhosts`-Dateien oder einen WINS-Server. In Kapitel 18, »NetBIOS-Namen ohne Broadcasts auflösen«, wird die Namenauflösung ohne Broadcasts detailliert dargestellt.

Schritt 9: Vom PC mit einer Freigabe verbinden

Wenn Sie die Schritte 7 und 8 erfolgreich abgeschlossen haben, sollte dieser Schritt korrekt funktionieren, mit einer möglichen Ausnahme. Bevor wir fortfahren, versuchen Sie, sich über den Befehl `net use` mit der `[tmp]`-Freigabe zu verbinden:

```
C:\users\jerry>net use t: \\bilbo\tmp
Der Befehl wurde erfolgreich ausgeführt.
```

Der Befehl verlangt möglicherweise ein Passwort, wenn Sie in Windows mit einem anderen Passwort eingeloggt sind als Ihrem Account auf dem Samba-Server. Ist das der Fall, geben Sie das korrekte Passwort ein und drücken Sie `[Return]`.

Ein Problem mit Windows 9x ist, dass Sie keinen anderen Benutzer spezifizieren können, der für die Sitzungsanfrage benutzt wird, im Gegensatz zum Windows-NT-Befehl `net.exe`, der den Parameter `/user:Benutzername` enthält. Wenn Sie in Windows unter einem anderen Benutzernamen eingeloggt sind als dem, den Sie für die Verbindung benutzen wollen, müssen Sie sich aus Windows ausloggen und noch einmal mit dem entsprechenden Benutzernamen neu anmelden. Die andere Möglichkeit besteht darin, die Samba-Option `username map` zu konfigurieren, damit der Name einem gültigen Account zugewiesen wird.

Haben Sie den korrekten Benutzernamen und das entsprechende Passwort benutzt und Samba für Klartextpasswörter konfiguriert, können sich aber immer noch nicht mit einer Freigabe verbinden, müssen Sie wahrscheinlich die Einstellung für `password level` in der `smb.conf` ändern. Denken Sie daran, dass Windows 9x in Klartext übertragene Passwörter zunächst in Großbuchstaben konvertiert.

Schritt 10: Browsing von der Netzwerkumgebung

Der letzte Test überprüft, ob das Netzwerk-Browsing funktioniert. Ich muss Sie warnen, dass Browsing ein »komplizierter Tanz« sein kann, wie eine der Samba-Dokumentationen es nennt. Wenn Sie den Samba-Server nicht sehen können, nachdem Sie in der Netzwerkumgebung zur korrekten Arbeitsgruppe navigiert sind, warten Sie bis, bis ich Ihnen eine umfassende Erklärung des Netzwerk-Browsings und der relevanten Fehlerbehandlung in den Kapiteln 19, »Browsing in lokalen Subnetzen«, und 20, »Browsing in Netzwerken mit Routern«, gebe.

Andere Dokumentation

Das Samba-Verzeichnis `/docs` ist voller Informationen. Ich sage nicht, dass sie hier Antwort auf jede erdenkliche Frage finden, aber das Verzeichnis bietet doch viel. Wenn Sie jemals feststellen, dass etwas fehlt, und Sie Informationen über Ihre Lösung aufschreiben wollen, können Sie Ihre Dokumente an die Samba-Verwalter weiterleiten und, hoffe ich, jemand anderem helfen, der auf ähnliche Schwierigkeiten getroffen ist.

Aus der Menge der Dateien im Dokumentationsverzeichnis möchte ich hier nur noch einige weitere erwähnen. Eine recht nützliche ist `UNIX_INSTALL.txt`. Diese Datei führt Sie durch die notwendigen Schritte für das Herunterladen, Kompilieren und Installieren des aktuellsten Source-Codes, die Sie bereits in den Kapiteln 3, »Wie bekomme ich den aktuellen Source-Code«, und 4, »Installation und Testen der Konfiguration« durchgeführt haben. Die meisten Informationen in dieser Datei kennen Sie bereits, aber es ist immer hilfreich, die gleichen Dinge mehrmals zu hören.

`ENCRYPTION.txt` ist eine andere Datei, die hilfreich ist, wenn Sie verschlüsselte LanManager- und Windows-NT-Passwörter verwenden wollen. Diese Datei beschreibt generell, wie die Passwort-Hashwerte generiert werden, und führt die notwendigen Schritte für die Aktivierung dieser Funktion auf. Wenn Sie Kapitel 6, »Sicherheitsmodi und Passwörter«, gelesen haben, sollten Sie mit den beschriebenen Begriffen und Algorithmen sowie den Schritten zur Aktivierung der Verschlüsselung vertraut sein.

Ich werde einige andere Dateien im Verzeichnis `/docs` erwähnen, wenn ich WINS-Unterstützung, Browsing und Domänenkontrolle später in diesem Buch darstelle. Für jetzt können Sie einfach einen Blick auf die Dokumentation werfen, damit Sie wissen, dass sie verfügbar ist, wenn Sie ein Problem haben.

Das Problem von beiden Enden angehen

Ich habe bereits über die Dokumentation gesprochen, die erklärt, was Sie tun sollten, um ein Problem anzugehen, und jetzt werde ich einige Tools darstellen, die Ihnen dabei helfen, Informationen zu sammeln und Ihr Problem hoffentlich zu lösen.

Ein altes Sprichwort sagt: »Zu einem Streit gehören immer zwei.« Dieses Sprichwort trifft sicherlich auf Netzwerkverbindungen zu. Ich habe es noch nie geschafft, mit dem Loopback-Interface (`127.0.0.1`) besonders viel Spaß zu haben. Sie gelangen darüber nicht wirklich irgendwo hin.

Jedes Netzwerkmodell umfasst mindestens zwei Rechner. Um Fehler in einer SMB-Verbindung zu beheben, müssen Sie sich sowohl den Client als auch den Server ansehen. Damit kommen wir wieder zu den verschiedenen Blickwinkeln, die für die Untersuchung des Problems verfügbar sind. Ich stelle mir dies gern als einen Krieger vor, der seinen Feind einkreist und nach schwachen Stellen in der Abwehr seines Gegners sucht. Das scheint vielleicht etwas zu dramatisch, aber ich denke, Sie verstehen die Idee.

Logdateien

Eines der besten Tools für das Debuggen von Samba ist Samba selbst. Die Menge der Informationen, die von `smbd` und `nmbd` aufgezeichnet werden kann, ist enorm. Wenn Sie die Debug-Loglevel einrichten (siehe Parameter `debug level` in `smb.conf` oder den Befehlszeilenparameter `-d`), entscheiden Sie, wie viele Informationen Sie ansehen wollen. Standardmäßig gibt Samba Debug-Informationen des Levels 2 und niedriger aus. Damit werden normalerweise Verbindungen und alle Fehlermeldungen des Systems protokolliert, z.B. wenn die `smbpasswd`-Datei nicht geöffnet werden kann. Hier ist ein Beispiel für Einträge durch den `smbd`-Daemon, wenn er mit einem Debug-Level von 2 läuft:

```
[1999/01/12 23:52:28, 1] smbd/service.c:make_connection(484)
bilbo (192.168.1.73) connect to service tmp as user jerryc (uis=1009, gid=100) (pid 436)
```

Debug-Level reichen von 0, für kritische Fehler, bis 10, für Entwicklungszwecke. Eine Zusammenfassung der verschiedenen Level sehen Sie in Tabelle 11.1. Wenn Sie den Debug-Level eines bestimmten laufenden Prozesses ändern wollen, können Sie den Level erhöhen, indem Sie dem Prozess ein USR1-Signal senden:

`kill -USR1 pid`

Alternativ können Sie den Debug-Level herabsetzen, indem Sie dem Prozess das USR2-Signal senden:

`kill -USR2 pid` *Tabelle 11.1: Beschreibung der Debug-Level*

Debug-Level	Beschreibung

0	Systemkritische Fehlermeldungen, z.B. wenn es nicht möglich ist, die System-Passwortdatei zu öffnen
1-2	Generelle alltägliche Protokollierung von Verbindungen und Benutzerauthentifizierung
3-5	Debugging-Einrichtung, Konfiguration und Source-Code
>6	Entwicklung

Hier ist ein Beispiel, in dem die Debug-Logs benutzt werden, um ein Problem in `smb.conf` zu beheben:

```
[src]
comment = /usr/local/src
path = /usr/local/sr
create mode = 0644
directory mode = 0755
```

Der Benutzer versuchte, sich von einer Windows-NT-4.0-Workstation über den Befehl `net use` mit der Freigabe zu verbinden, und erhielt die Fehlermeldung: »Der Netzwerkpfad wurde nicht gefunden«. Der folgende Eintrag wurde im `smbd`-Debug-Log gefunden:

```
[1999/01/13 16:15:16, 0] smbd/service.c:make_connection(437)
Can't change directory to /usr/local/sr (No such file or directory)
```

Wie Sie sehen, existierte der Pfad für die Freigabe nicht. Nach der Korrektur konnte sich der Benutzer erfolgreich mit `\\bilbo\src.` verbinden. Der oben stehende Log-Eintrag wurde als Level-0-Debug-Meldung aufgezeichnet.

Wenn Samba mit einem sehr hohen Debug-Level läuft, werden sehr viele Informationen aufgezeichnet, die schnell die Festplatte füllen, auf der die Debug-Logs gespeichert werden. Ich kann die Verwendung von Debug-Levels höher als 2 für die normale alltägliche Verwendung nicht empfehlen. Wenn Sie höhere Level verwenden wollen, sollten Sie den Parameter `max log size` benutzen, um die maximale Größe der Logdateien in Kilobyte zu kontrollieren.

Damit Sie einen Eindruck davon bekommen, wie viele Informationen `smbd` protokollieren kann, finden Sie in Listing 11.1 die Ausgabe von der gleichen Verbindung zu `\\bilbo\src` bei einem Debug-Level 10. Ziemlich lang, hm? Tatsächlich enthält das Listing nur die Schritte während der Anfrage für die Protokollverhandlung.

Listing 11.1: Level-10-Debug-Ausgabe aufgezeichnet von smbd für eine Protokollverhandlungsanfrage während eines Verbindungsversuchs zu einer Freigabe von einem Windows-95-Client

```
[1999/01/13 16:34:10, 6] param/loadparm.c:lp_file_list_changed(1767)
  lp_file_list_changed()
  file /etc/smb.conf -> /etc/smb.conf last mod_time: Wed Jan 13 16:14:54 1999

[1999/01/13 16:34:10, 5] smbd/connection.c:claim_connection(127)
  trying claim /usr/local/samba/var/locks STATUS. 100000
[1999/01/13 16:34:10, 8] lib/util.c:fcntl_lock(2632)
  fcntl_lock 8 7 0 1 1
[1999/01/13 16:34:10, 8] lib/util.c:fcntl_lock(2693)
  Lock call successful
[1999/01/13 16:34:10, 8] lib/util.c:fcntl_lock(2632)
  fcntl_lock 8 7 0 1 2
[1999/01/13 16:34:10, 8] lib/util.c:fcntl_lock(2693)
  Lock call successful
[1999/01/13 16:34:10, 5] smbd/reply.c:reply_special(147)
  init msg_type=0x81 msg_flags=0x0
[1999/01/13 16:34:10, 6] lib/util_sock.c:write_socket(185)
  write_socket(6,4)
[1999/01/13 16:34:10, 6] lib/util_sock.c:write_socket(188)
  write_socket(6,4) wrote 4
[1999/01/13 16:34:10, 10]
```

```
lib/util_sock.c:read_smb_length_return_keepalive(445)
  got smb length of 170
[1999/01/13 16:34:10, 6] smbd/process.c:process_smb(564)
  got message type 0x0 of len 0xaa
[1999/01/13 16:34:10, 3] smbd/process.c:process_smb(565)
  Transaction 1 of length 174
[1999/01/13 16:34:10, 5] lib/util.c:show_msg(459)
   size=170
   smb_com=0x72
   smb_rcls=0
   smb_reh=0
   smb_err=0
   smb_flg=24
   smb_flg2=3
[1999/01/13 16:34:10, 5] lib/util.c:show_msg(465)
   smb_tid=0
   smb_pid=51966
   smb_uid=0
   smb_mid=0
   smb_wct=0
[1999/01/13 16:34:10, 10] lib/util.c:show_msg(475)
  smb_bcc=135
[1999/01/13 16:34:10, 10] lib/util.c:dump_data(2832)
  [000} 02 50 43 20 4E 45 54 57  4F 52 4B 20 50 52 4F 47  .PC NETW ORK
PROG
  [1999/01/13 16:34:10, 10] lib/util.c:dump_data(2840)
  [010} 52 41 4D 20 31 2E 30 00  02 58 45 4E 49 58 20 43  RAM 1.0. .XENIX
C
[1999/01/13 16:34:10, 10] lib/util.c:dump_data(2840)
  [020} 4F 52 45 00 02 4D 49 43  52 4F 53 4F 46 54 20 4E  ORE..MIC ROSOFT
N
[1999/01/13 16:34:10, 10] lib/util.c:dump_data(2840)
  [030} 45 54 57 4F 52 4B 53 20  31 2E 30 33 00 02 4C 41  ETWORKS 1.03..LA
[1999/01/13 16:34:10, 10] lib/util.c:dump_data(2840)
  [040} 4E 4D 41 4E 31 2E 30 00  02 57 69 6E 64 6F 77 73  NMAN1.0..Windows
[1999/01/13 16:34:10, 10] lib/util.c:dump_data(2840)
  [050} 20 66 6F 72 20 57 6F 72  6B 67 72 6F 75 70 73 20  for Wor kgroups
[1999/01/13 16:34:10, 10] lib/util.c:dump_data(2840)
[060} 33 2E 31 61 00 02 4C 4D  31 2E 32 58 30 30 32 00 3  .1a..LM 1.2X002.
[1999/01/13 16:34:10, 10] lib/util.c:dump_data(2840)
  [070} 02 4C 41 4E 4D 41 4E 32  2E 31 00 02 4E 54 20 4C  .LANMAN2 .1..NT
L
[1999/01/13 16:34:10, 10] lib/util.c:dump_data(2840)
  [080} 4D 20 30 2E 31 32 00                              M 0.12.
```

Menschen

Menschen können die beste Informationsquelle sein, um Probleme zu beheben. Denken Sie aber daran, dass alle Leute, die Sie möglicherweise in den verschiedenen Mailing-Listen oder Newsgroups kontaktieren, Ihnen helfen, weil Sie es wollen, und nicht, weil sie dafür bezahlt werden. Dazu gehören auch die, die Samba entwickeln und in der Regel als das Samba-Team bezeichnet werden. Das heißt, dass es letztendlich an Ihnen liegt, Ihre Probleme zu beheben.

Wenn Sie in einer Mailing-Liste oder Newsgroup Fragen postieren oder beantworten, sollten Sie sich an die übliche Internet-Etiquette (oder Netiquette) halten. Falls Sie dies nicht tun, werden Sie feststellen, dass die Leute weniger hilfreich sind. Wenn Sie jedoch in Ihren Postings rücksichtsvoll sind, wird Ihnen normalerweise jemand antworten.

Weitere Informationen zu den Samba-Mailing-Listen und wie Sie an Ihnen teilnehmen können, finden Sie unter http://samba.org/listproc. Unter anderem sind folgende Mailing-Listen verfügbar:

- samba@samba.org - Dies ist die Haupt-Samba-Mailing-Liste für allgemeine Informationen über die aktuellste Samba-Distribution (z.B. Samba 2.0).
- samba-technical@samba.org - In dieser Mailing-Liste wird über die Entwicklung von Samba diskutiert. Wenn Sie daran teilhaben

wollen, werden Sie Mitglied der Liste, öffnen Sie `vi` und beginnen Sie damit, sich durch den Source-Code zu arbeiten.

- samba-ntdom@samba.org - Diese Liste legt ihren Schwerpunkt auf das Testen und Debuggen von Sambas Unterstützung für den Primary Domain Controller für Windows-NT-Domänen.
- samba-bugs@samba.org - Diese Adresse ist keine Mailing-Liste, sondern eine Adresse, an die Sie über aktuelle Fehler in den Samba-Anwendungen berichten können.

Die Usenet-Newsgroup `comp.protocols.smb` ist eine weitere gute Informationsquelle für die Konfiguration und das Testen von Samba.

Netzwerk-Sniffer

Wie ich Ihnen vorher schon erzählt habe, erhielt ich einen meiner ersten Jobs als Netzwerkadministrator aufgrund einer Assistentenstelle an der Universität. Ich arbeitete Teilzeit und musste ein kleines Netzwerk verwalten, das aus PCs und einem einzelnen Sparc-IPX-Mail- und Web-Server mit SunOS 4.1.3 bestand. Nach einiger Zeit konnte ich das Betriebssystem auf dem Server auf Solaris 2.5 aktualisieren und einen zweiten Datei-Server installieren. Als ich versuchte, ein Problem zu lokalisieren, fragte mich ein anderer Systemadministrator, der später ein sehr guter Freund wurde: »Hast du dir die `snoop`-Ausgabe schon angesehen?« (`snoop` ist eine Software für die Paketüberprüfung, die in Solaris-2.x-Rechner integriert ist). Ich sagte: »Wie?«, und so kam ich erstmals mit Paket-Sniffern in Berührung. Obwohl das Ganze wesentlich weniger schädlich als das Schnüffeln von Kleber ist, kann es wesentlich abhängiger machen!

Ein *Paket-* (oder *Netzwerk-*)*Sniffer* ist ein Utility, über das Sie Pakete abfangen können, die der Host-Rechner auf dem Netzwerk sieht. Wenn Sie eine gemeinsam benutzte Medienumgebung verwenden (wie z.B. Ethernet), sieht das Netzwerk-Interface des Rechners alle Pakete auf dem lokalen Subnetz. Wenn Sie irgendeine geschaltete Umgebung verwenden, können Sie nur die Pakete sehen, die an oder von Ihrem Rechner oder an die Broadcast-Adresse des Netzwerks übertragen werden.

Welchen Vorteil bieten Paket-Sniffer? Wenn Sie Samba einfach nur benutzen wollen und nicht neugierig sind, was hinter den Kulissen passiert, können Sie diesen letzten Abschnitt einfach überschlagen und direkt mit Kapitel 12, »Fallstudie: Einen NT-Datei- und Drucker-Server ersetzen«, weitermachen. Wenn Sie aber auch nur eine gewisse Neugier verspüren, lesen Sie weiter.

Der Vorteil in der Benutzung eines Paket-Sniffers liegt darin, dass Sie genau das zu sehen bekommen, was Samba sieht. Es ist nicht nur einmal vorgekommen, dass jemand mit einem Paket-Sniffer einen Fehler in Samba, Windows NT, Windows 9x und [fügen Sie hier Ihr bevorzugtes Netzwerkbetriebssystem ein] gefunden hat. Sie wissen, was ich meine.

Ich werde in diesem Abschnitt zwei Paket-Sniffer darstellen. Einer ist frei verfügbar, der andere ist ein kommerzielles Produkt.

tcpdump-smb

`tcpdump` ist ein frei verfügbares Netzwerkpaket-Analysetool, das ursprünglich von Van Jacobson geschrieben wurde. Teile des Tools wurden später von Steven McCanne umgeschrieben, und der Code wird heute von der Network Research Group am Lawrence Berkeley National Laboratory verwaltet. `tcpdump` verlangt die `libcap`-Bibliothek, ein benutzerbasiertes, rechnerunabhängiges Interface für das Abfangen von Paketen.

Andrew Tridgell, der Originalautor von Samba, hat eine Sammlung von Patches geschrieben, damit `tcpdump` SMB-Pakete analysieren kann. Die aktuellste Version von `tcpdump`, den Source-Code für `libcap` und die Patch-Sammlung für die SMB-Unterstützung können Sie von `ftp://ftp.samba.org/pub/samba/tcpdump-smb` herunterladen.

`tcpdump` bietet gegenüber dem anderen Paketausgabe-Utility, das Sie sich später ansehen werden, den Vorteil, dass es komplett befehlszeilenbasiert ist. Daher läuft es sehr gut in einer Remote-Telnet-Sitzung. Kompiliert ist das Binary unabhängig und kann leicht auf den entfernten Rechner kopiert und dort ausgeführt werden. Sie müssen keine speziellen Gerätetreiber oder eine andere Software installieren.

Zusätzlich dazu ist `tcpdump` frei verfügbar. Das heißt, Sie können Pakete auf einem Irix-Rechner abfangen und die Ausgabedatei an andere Systemadministratoren senden, die andere Plattformen verwenden. Diese wiederum können sich die Datei einfach ansehen, wenn sie `tcpdump` für ihr System kompiliert haben.

Vielleicht ist es einfacher, diese Darstellung fortzusetzen, wenn wir uns gemeinsam eine Beispielausgabe ansehen. Für die erste Paketverfolgung habe ich den Befehl `net view \\bilbo` von meinem Windows-95-Client, `queso`, laufen lassen. Auf dem Slackware-Linux-Rechner `bilbo` läuft Samba 2.0. Ich habe die Pakete auf `bilbo` über den Befehl

```
tcpdump port 139 and host queso
```

abgefangen, der die Ausgabe in Listing 11.2 generiert hat.

```
queso.1083 > bilbo.netbios-ssn: P 73:231(158) ack 5 win 8756
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=154
SMB PACKET: SMBnegprot (REQUEST) bilbo.netbios-ssn > queso.1083: P 5:87(82) ack 231 win 32736
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=78
SMB PACKET: SMBnegprot (REPLY) queso.1083 > bilbo.netbios-ssn: P 231:392(161) ack 87 win 8674
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=157
SMB PACKET: SMBsesssetupX (REQUEST)
SMB PACKET: SMBtconX (REQUEST) (CHAINED)
bilbo.netbios-ssn > queso.1083: P 87:180(93) ack 392 win 32736
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=89
SMB PACKET: SMBsesssetupX (REPLY)
SMB PACKET: SMBtconX (REPLY) (CHAINED)
```

Ich habe die Teile der Ausgabe herausgenommen, die Informationen für die Raw-Pakete und die SMB-Flags anzeigen. Eine Leerzeile trennt jedes Paket. Wenn Sie sich die jeweiligen Zeilen mit der Bezeichnung `SMB PACKET` ansehen, werden Sie die drei Schritte für eine SMB-Verbindung zu einem Server im User-Modus erkennen:

Listing 11.2: tcpdump-Ausgabe des Befehls net view \\Servername, der auf einem Windows-95-Client ausgegeben wurde

```
SMB PACKET: SMBnegprot (REQUEST)
SMB PACKET: SMBnegprot (REPLY)
SMB PACKET: SMBsesssetupX (REQUEST)
SMB PACKET: SMBtconX (REQUEST) (CHAINED) SMB PACKET: SMBsesssetupX (REPLY)
SMB PACKET: SMBtconX (REPLY) (CHAINED)
```

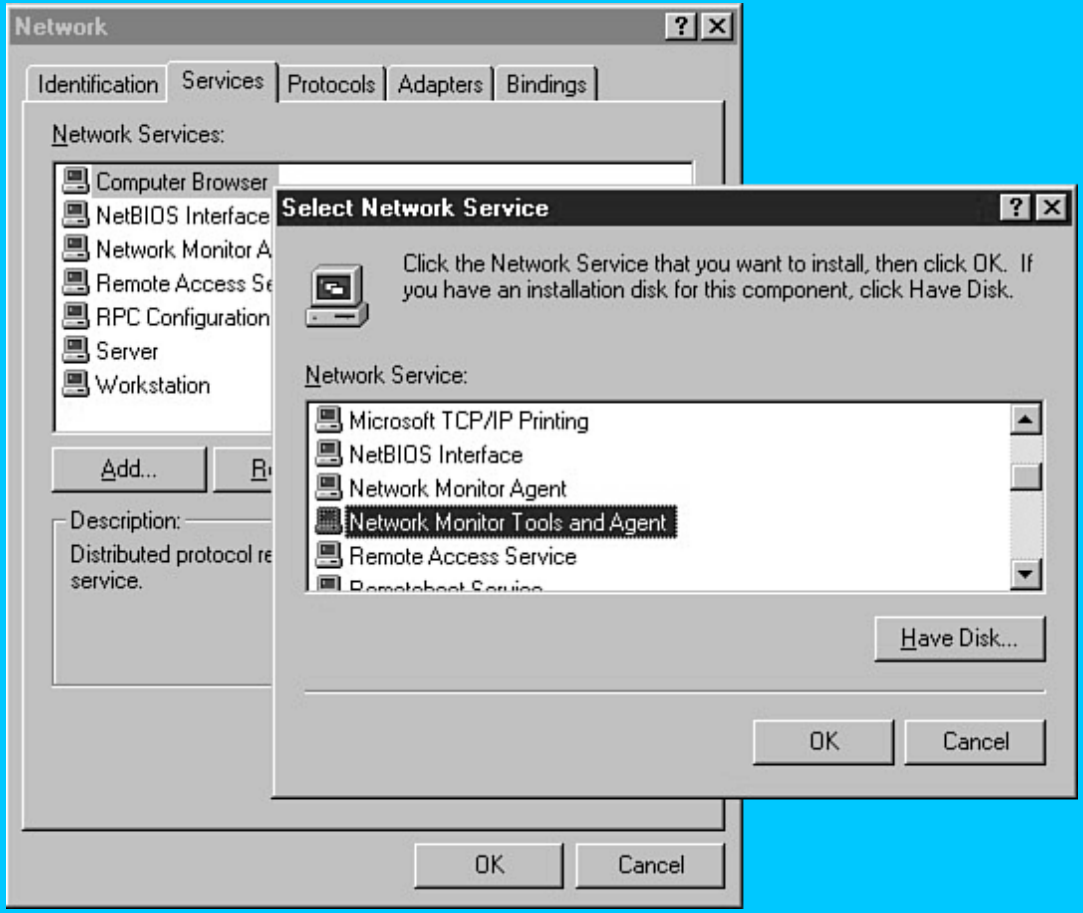
Das erste Paket stellt den Schritt dar, bei dem Client und Server sich für den SMB-Protokolldialekt entscheiden.

Ich habe bisher noch nicht erwähnt, dass bestimmte SMB-Befehle in einem Paket zusammengefasst werden können. Im Anfragepaket sendet der Client die Sitzungsanfrage und die Verbindungsanfrage für den Verzeichnisbaum. Der Server sendet die Antworten für beide Anfragen in einem Antwortpaket zurück.

Microsofts Network Monitor

Microsoft hat ein Paketausgabe-Tool in die Windows-NT-Server- und die *System-Management-Software-(SMS-)*CD-ROM integriert, das *Network Monitor* genannt wird (alias `netmon`). `netmon` besteht aus zwei Teilen: einem Agent und dem Tool selbst. Beide müssen installiert werden, damit die Software korrekt funktioniert. Abbildung 11.2 zeigt das Installationsfenster unter Windows NT 4.0 Server.

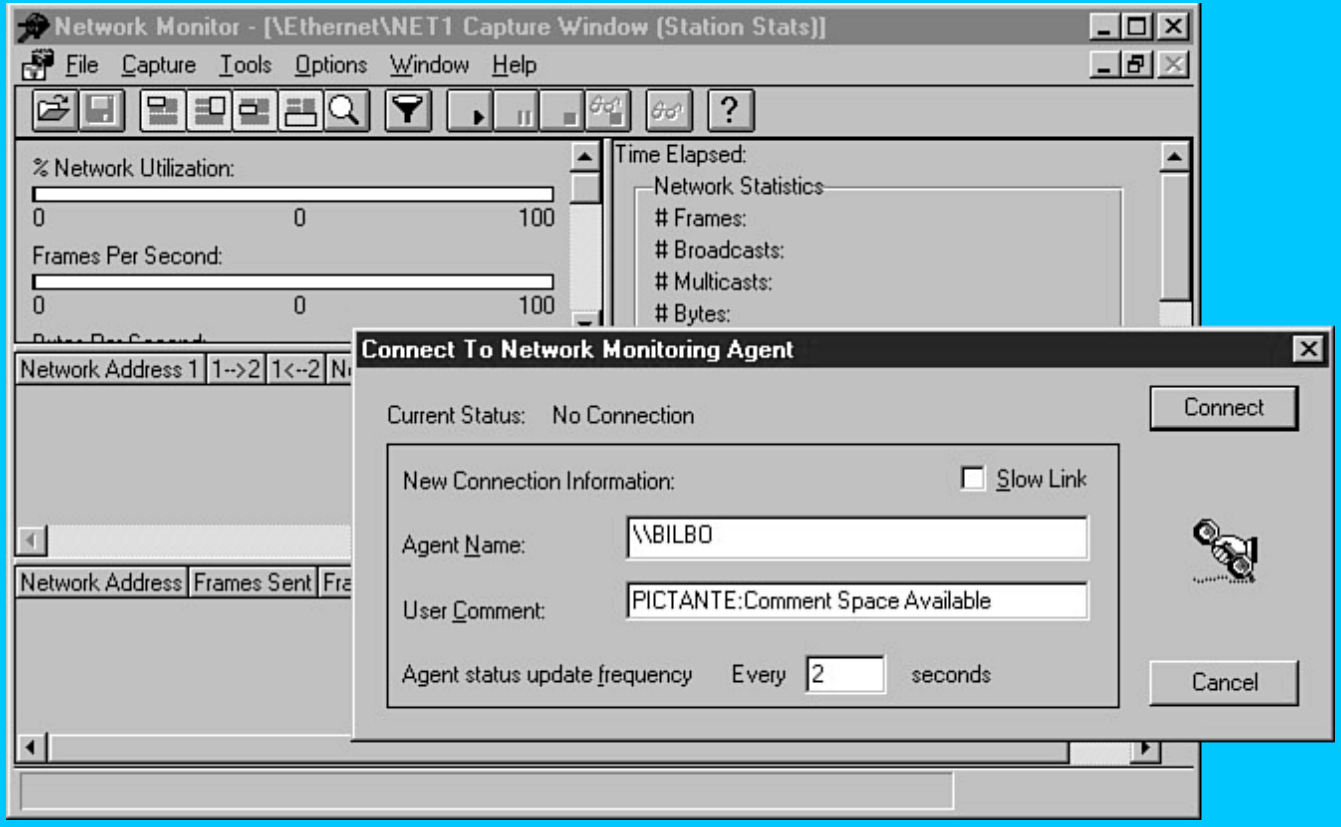
Abb. 11.2: Den Network-Monitor-Agent und die Tools auf einer Windows-NT-4.0-Workstation installieren



Es gibt zwei verschiedene Versionen von `netmon`, die beide nicht frei verfügbar sind wie `tcpdump`. Mit der Version, die sich auf der Windows-NT-4.0-Server-CD-ROM befindet, können Sie nur Pakete ansehen, die vom oder an den lokalen Rechner gesendet werden. Die Version auf der SMS-CD-ROM ermöglicht, dass das Netzwerk-Interface in *Promiscuous Mode* gesetzt wird, in dem alle Pakete auf dem geteilten Medium angesehen werden können. Beide Versionen können auch lokal auf Windows-NT-Workstations und Windows-9x-Clients laufen, wenn der Netzwerkmonitoragent installiert wurde.

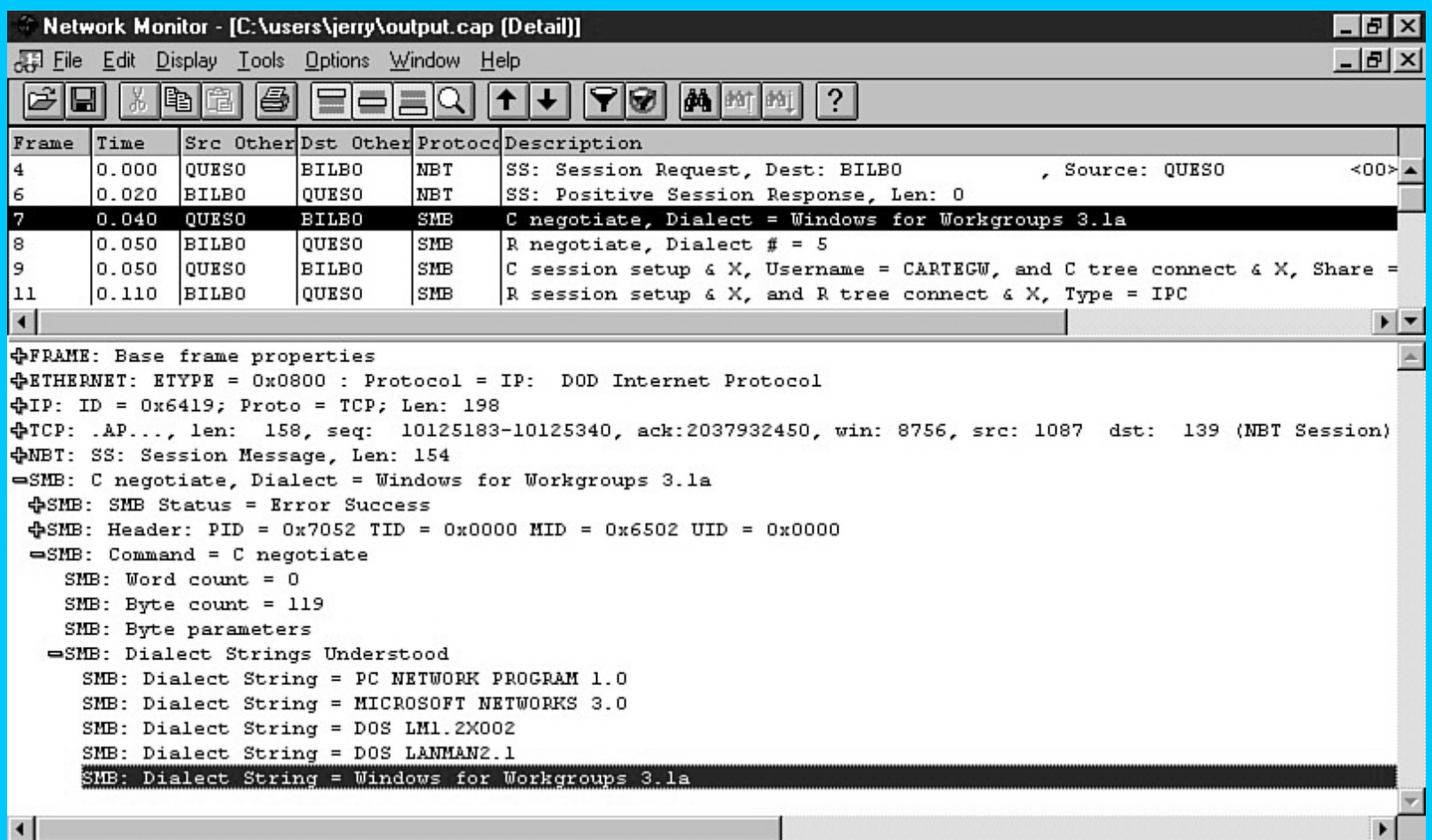
Obwohl `netmon` nicht auf einem Rechner in einem Telnet-Fenster laufen kann, ist es möglich, Pakete entfernt abzufangen. Dafür muss aber der Netzwerkmonitoragent auf dem entfernten Rechner installiert sein. Haben Sie dies getan, können Sie das entfernte Netzwerk, mit dem Sie sich verbinden wollen, wie in Abbildung 11.3 gezeigt, spezifizieren.

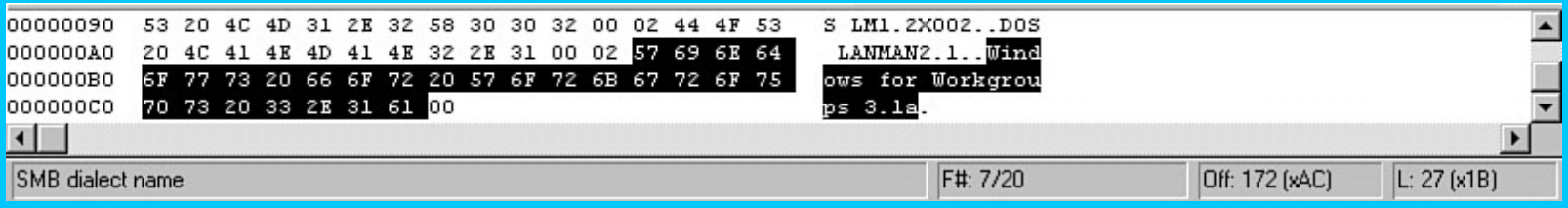
Abb. 11.3: Verbindung zu einem Remote-Network-Monitor-Agent



Warum sollten Sie `netmon` benutzen, wenn `tcpdump` frei verfügbar ist? Das ist eine gute Frage. Network Monitor ist sehr gut für die Analyse vieler Arten von Paketen, darunter DNS-Anfragen, NFS-Pakete, IPX und sogar Microsofts eigene Implementierung von DCE/RPC. Abbildung 11.4 zeigt die gleiche Paketanalyse, die Sie vorher mit `tcpdump` gesehen haben (siehe Listing 11.2), wenn sie in `netmon` generiert wird.

Abb. 11.4: Netzwerk-Paketanalyse in Network Monitor





tcpdump und netmon gemeinsam anwenden

In Network Monitor werden Ausgaben auf eine sehr nette Art und Weise dargestellt, aber was ist, wenn Sie die benötigten Pakete nur auf einem entfernten Unix-Rechner abfangen können? Hier ist eine einfache Methode, über die Sie das Beste aus beiden Welten kombinieren können.

Benutzen Sie zunächst die SMB-aktivierte Version von `tcpdump`, um die Pakete auf dem entfernten Netzwerk abzufangen, und speichern Sie die Ausgabe in einer Datei:

```
tcpdump -s 1000 -t -w output.dump host <Hostname>
```

Verwenden Sie nun das Tool `capconvert`, um die `tcpdump`-Ausgabedatei in das CAP-Format von `netmon` zu konvertieren. Die Source-Datei für `capconvert` können Sie unter `ftp://ftp.samba.org/pub/tcpdump-smb/capconvert.c` herunterladen.

Wenn Sie einen Solaris-2.x-Rechner benutzen und lieber Suns eigenen Paket-Sniffer, `snoop`, verwenden, können Sie sich einen `snoop-CAP`-Konvertierer unter `ftp://ftp.samba.org/pub/tcpdump-smb/snoop2cap.c` herunterladen.

Jetzt müssen Sie nur noch die konvertierte Datei auf einen Windows-Rechner kopieren, um sie in `netmon` ansehen zu können.

Zusammenfassung

Die Behandlung eines Problems liegt irgendwo zwischen Kunst und Wissenschaft. Dieses Kapitel hat Ihnen die Informationen über verfügbare Tools gegeben, mit denen Sie den wissenschaftlichen Teil Ihrer Aufgabe erledigen können. Dinge wie gute Dokumentation, Log-Einträge und Paket-Sniffer können Ihnen alle notwendigen Informationen geben, um das Problem zu bestimmen. Aber die Analyse dieser Informationen und deren Benutzung, um eine Lösung für das Problem zu finden, braucht Zeit, Erfahrung und Kreativität.

Verschiedene Mailing-Listen und Usenet-Newsgroups können Sie in Kontakt mit Leuten bringen, die Ihnen mit der Kunst des Troubleshootings behilflich sein können. Denken Sie daran, dass Sie Ihre Hausaufgaben machen und sich das Problem aus allen Blickwinkeln ansehen, statt nur das kaputte Teil zu betrachten.

Frage & Antwort

F. Gibt es Archive für die verschiedenen Samba-Mailing-Listen, in denen ich nachsehen kann, ob jemand meine Frage schon einmal gestellt hat?

. Ja. Es gibt ein durchsuchbares Archiv für alle Mailing-Listen, die von `samba.org` verwaltet werden. Weitere Informationen finden Sie unter http://samba.org/listproc.

F. Wo finde ich Informationen über kommerziellen Support für Samba?

. Die Haupt-Samba-Website beinhaltet eine Seite, auf der Sie eine Liste von Unternehmen finden, die kommerziellen Support für Samba bieten. Eine »offizielle« Samba-Support-Struktur wird derzeit organisiert und in naher Zukunft verfügbar sein.

Neue Begriffe

Paket-Sniffer - Eine übliche Bezeichnung für eine Klasse von Netzwerk-Tools, entweder Software oder Hardware, die die rohen Daten anzeigen können, die über ein Netzwerk übertragen werden. Diese Utilities werden auch als Netzwerkverfolger oder Paketverfolger bezeichnet. Einige bieten auch Funktionen für die Analyse der Pakete und zeigen die Informationen in einem für das menschliche Auge besser lesbaren Format an.

ZURÜCK Inhalts-verzeichnis Stichwort-verzeichnis VOR

Tag 12: Fallstudie: Einen NT-Datei- und Drucker-Server ersetzen

Ich habe Besprechungen wie diese hassen gelernt. Ich gehe die Folienpräsentation noch einmal in meinem Kopf durch. Wenn ich nur eine Netzwerkverbindung unter meinem Tisch hätte, dann könnte ich wenigstens etwas Nützliches tun, wie z.B. meine E-Mail abrufen oder so etwas.

Ich merke, dass meine Chefin sich bereit macht, mich bald vorzustellen. »...und hier ist jetzt unser firmeninterner Experte in der Netzwerkadministration mit der Kostenanalyse für den Ersatz des Servers.« Meine Chefin liebt es, in ihren Sätzen das Wort *Experte* unterzubringen. Ich trinke noch einen Schluck Kaffee, mache mich auf den Weg in die Mitte des Raumes und stelle mich neben den Projektor. Ich drücke die Leertaste, um meinen Notebook aus dem Schlafmodus zu holen, und beginne zu reden. »Wir wollen uns heute einige Zahlen ansehen, die die Kosten der Dienste vergleichen, die wir unseren Benutzern im Netzwerk bieten«, beginne ich. Ich kann hören, wie die Festplatte meines Notebooks zu arbeiten beginnt, und die erste Folie erscheint wie gerufen ...

... das Fazit ist folgendes: Wenn wir eine Kombination aus Linux und Samba auf gewöhnlicher PC-Hardware benutzen, können wir den existierenden Datei-Server durch einen neueren Rechner ersetzen, der zweimal so schnell ist und etwa halb so viel kostet. Zweitens wird es keine mit dem Server verbundenen Client-Lizenzierungsgebühren pro Arbeitsplatz oder pro Verbindung geben. Und schließlich wird die Änderung für den Benutzer transparent sein.« Ich stoße einen stillen Erleichterungsseufzer aus, als ich mich wieder hinsetze, nur um festzustellen, dass mein Kaffee mittlerweile kalt geworden ist.

»Wenn diese Lösung so gut ist, wie sie sich anhört, warum haben wir das nicht gleich so gemacht?«, fragt einer der Abteilungsleiter.

Ich zucke meine Schultern ein wenig und erinnere mich an die Person, die den letzten Stoß Windows-NT-Server für das Unternehmen installiert hat. »Die Zeiten ändern sich», erkläre ich. »Egal was hinter der Planung stand, die uns an diesen Punkt gebracht hat, ist die Lösung, die ich präsentiert habe, heute für uns die beste, und ich glaube, dass sie uns auch in Zukunft gut bekommen wird.«

»Gut gemacht«, sagt meine Chefin, als wir nach der Besprechung zurück zum Büro gehen. »Ich lasse Mike die Bestellungen für die neue Hardware bis heute abend erledigen.«

»Sie ist immer so überoptimistisch, was diese Bestellungen angeht«, denke ich bei mir und muss lächeln. »Hört sich gut an«, antworte ich, als ich um die Ecke zum Labor biege und mich auf meinen Weg mache. Ich fange an, in meinem Kopf die Dinge durchzugehen, die ich erledigen muss, um den Windows-NT-Datei-Server durch einen Linux-Rechner zu ersetzen. »Wo habe ich bloß die Kaffeetasse hingetan?«, murmle ich.

Bis hierher habe ich die Möglichkeiten von Samba und die Einrichtung der Datei `smb.conf` dargestellt. Jetzt ist an der Zeit, das bisher Gelernte praktisch umzusetzen. In diesem Kapitel führe ich Sie Schritt für Schritt durch den Prozess, einen Windows-NT-4.0-Server durch einen Linux-Rechner mit Samba zu ersetzen. Der Windows-NT-Rechner bietet Festplatten- und Druckerfreigaben. Der Samba-Server übernimmt einfach die Aufgabe, die Freigaben zur Verfügung zu stellen. Wenn alles gut geht, wird der Endanwender niemals bemerken, dass der NT-Server ersetzt wurde.

Das vorhandene Netzwerk

Zunächst muss ich festlegen, welchen Anforderungen mein Samba-Server entsprechen muss. Ich stelle eine Liste auf:

- Alle Benutzer in der Domäne sollten auf die neuen Freigaben auf dem Samba-Server zugreifen können, ohne dass ein synchronisierter Unix-Account auf dem Rechner verlangt wird. Das bedeutet, dass der existierende NT-Domänen-Account Zugriff auf die freigegebenen Ressourcen des neuen Servers bieten sollte.
- Der Samba-Server sollte in der gleichen Arbeitsgruppe sein und den gleichen NetBIOS-Rechnernamen benutzen wie der existierende Server, um die Verwirrung für den Benutzer so gering wie möglich zu halten.
- Die Zugriffskontrollmechanismen für Dateien sollten die gleichen sein, damit ein Benutzer, der Zugriff auf eine Datei auf dem existierenden Server hat, auf die gleiche Datei auf dem Samba-Server zugreifen kann. Außerdem sollte ein Benutzer, der keinen Zugriff auf eine Datei auf dem existierenden Server hat, auch unter der neuen Konfiguration nicht auf diese Datei zugreifen können.

Die ersten zwei Punkte lassen sich ohne Komplikationen realisieren, aber mit der letzten Anforderung werde ich einige Arbeit haben.

Hier sind die Netzwerkressourcen, die der aktuelle Windows-NT-Server bietet. Ich habe diese Liste etwas vereinfacht, damit ich einige Zeit damit verbringen kann, jeden Dienst einzeln anzusehen.

- `[users]` - Diese Freigabe enthält die Home-Verzeichnisse für die Benutzer in der Domäne.
- `[docs]` - Eine gemeinsame Festplattenfreigabe für die Gruppenzusammenarbeit. Alle Benutzer können innerhalb der Freigabe Verzeichnisse erstellen, wird aber eine Datei erzeugt, wird der Zugriff über die Standard-NTFS-ACLs kontrolliert.
- `[canon]` - Ein Netzwerkdrucker, der für alle Benutzer in der Domäne verfügbar ist.

Abb. 12.1: Ein Überblick über das Netzwerk, das den zu ersetzenden Windows-NT-Server enthält

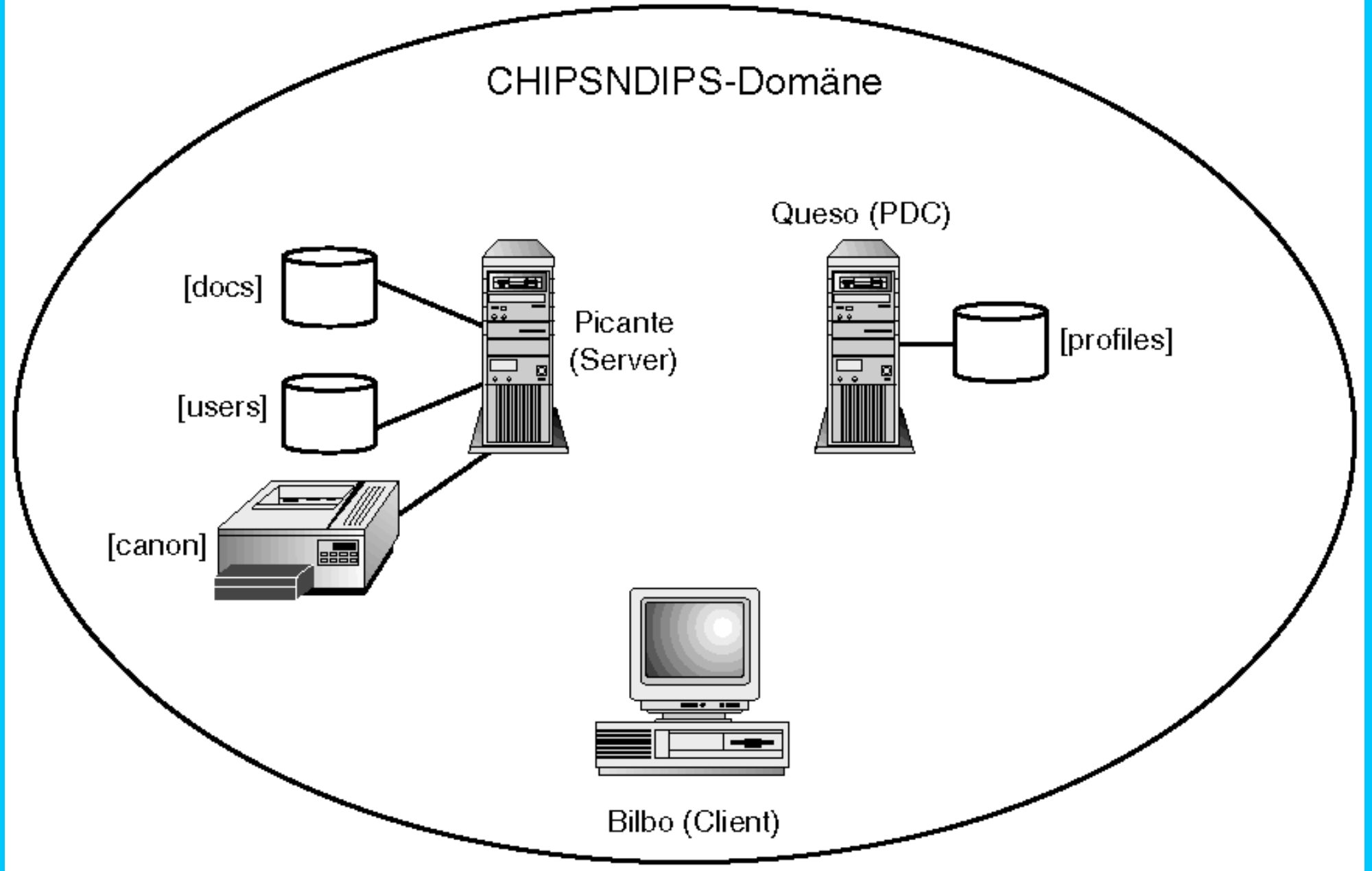


Abbildung 12.1 stellt den derzeitigen Aufbau dar. Ich werde in einem einzelnen Domänenmodell mit einem *Primary Domain Controller (PDC)* arbeiten, der alle Benutzerauthentifizierungen handhabt. Die Anzahl der Client-Rechner ist für meine Zwecke hier unwichtig. Um den neuen Server zu testen, brauche ich den PDC für die Durchführung der Authentifizierung, den neuen Server und einen Windows-NT-Client-Rechner.

Der Linux-Server

Als Hintergrundinformation und um sicherzustellen, dass wir alle auf dem gleichen Stand sind, möchte ich klarstellen, dass ich einen Server benutzen werde, der mit der Slackware-Linux-3.5-Distribution aufgebaut wurde, die auf dem Linux-Kernel 2.0.34 basiert. Hier ist die Ausgabe von `uname -a`:

```
Linux picante 2.0.34 #2 Thu Jun 4 22:36:07 PDT 1998 i586 unknown
```

Der Rechner selbst ist ein Dell Pentium 233 mit einer 8-Gbyte-IDE-Festplatte und 128 Mbyte RAM.

Die Samba-Installation ist Version 2.0, die ich als Source-Code heruntergeladen und über die folgenden drei Befehle kompiliert und installiert habe:

```
./configure
make
make install
```

Der Ersetzungsprozess

Ich werde fünf Schritte durchlaufen, um den Windows-NT-Server durch den neueren Linux-Rechner zu ersetzen:

1. Existierende NT-Domänen-Accounts und alle notwendigen Abgleichungen innerhalb des Linux-Rechners bearbeiten
2. Die Dateien und Drucker-Spool-Dateien vom Windows-NT-Server auf den Linux-Rechner verschieben
3. Die entsprechenden Parameter in der Samba-Konfigurationsdatei (`smb.conf`) konfigurieren
4. Den Samba-Server der NT-Domäne hinzufügen
5. Den neuen Server testen

Schritt 1: Benutzer und Gruppen

Als ich zum ersten Mal über die Dienste sprach, die mein Samba-Server bieten muss, habe ich entschieden, dass sich alle existierenden Domänenbenutzer in der Lage sein sollten, sich mit dem neuen Server zu verbinden, ohne explizit von einem zweiten Account zu wissen. Es ist keine akzeptable Option, einen separaten Account zu erstellen und zu verwalten, auch nicht einen, dessen Passwort zum Domänen-Account synchron ist. Die Tatsache, dass Benutzer keine gültige Shell auf dem Linux-Rechner haben werden, macht diese Entscheidung möglich. Wenn Samba den Windows-NT-PDC für die Benutzerauthentifizierung benutzen kann, gibt es noch eine Sache weniger, über die ich mir den Kopf zerbrechen muss.

NTs relative IDs und Unix-UIDs und -GIDs

Bevor ich zu den speziellen Punkten komme, die für Benutzer-Accounts relevant sind, möchte ich einige Hintergrundinformationen für diejenigen bieten, die damit vertraut sind, wie Windows NT einen Account intern darstellt. Ich denke, das dies helfen wird, einige der Entscheidungen zu rechtfertigen, die ich später treffen muss.

Windows NT benutzt wie Unix eine numerische Zahl, um Accounts intern darzustellen. Unix macht dabei jedoch einen Unterschied zwischen einem Domänen-Account und einem lokalen Account. Wenn Sie mit Suns *Network Information Service (NIS)* vertraut sind, ist es hilfreich, sich den PDC als den NIS-Master vorzustellen, und Domänen-Accounts sind jene, die in der NIS-`passwd`-Map aufgelistet sind. Wenn ein Account mit dem Benutzernamen `jdoe` sowohl in der lokalen Datei `/etc/passwd` als auch in der NIS-`passwd`-Map vorhanden ist, wird nur einer der Accounts gesehen, je nach

Suchreihenfolge, die in `/etc/nsswitch.conf` definiert ist: Dateien oder NIS. Deshalb habe ich vorher erwähnt, dass Unix die Unterscheidung zwischen lokalen und globalen Accounts nicht ermöglicht.

Ein anderer Unterschied zwischen den beiden Betriebssystemen besteht darin, dass Windows-NT-Gruppen- und -Benutzer-Accounts im gleichen Zahlenbereich existieren. Es ist sehr gut möglich, wenn nicht sogar üblich, einen Unix-Rechner mit einer Gruppe zu sehen, die eine ID von 0 (`wheel`) hat, und einem Benutzer-Account, der ebenfalls eine ID von 0 (`root`) hat. Es kommt in diesem Fall zu keinen Problemen, da Unix-GIDs und -UIDs völlig separat voneinander existieren. Windows-NT-Gruppen- und -Benutzer-Accounts existieren nebeneinander. Daher ist es unmöglich, eine Gruppe und einen Benutzer-Account mit einer ID von 1002 zu haben. Windows-NT-Gruppen und -Benutzer werden in einer monoton sich erhöhenden Reihenfolge erzeugt, die mit 1000 beginnt. Diese ID-Nummer wird *relative ID* oder *RID* genannt. Einer RID wird entweder die ID des lokalen Rechners oder die ID der Domain angehängt, um den Account vollständig zu qualifizieren und so die Unterscheidung zwischen lokalen und Domänen-Accounts sicherzustellen. Diese Rechner-IDs werden *Security Identifiers* oder *SIDs* genannt.

Vielleicht fragen Sie sich, ob Gruppen und Benutzer im gleichen Zahlenbereich existieren. Wie können Sie dann bestimmen, ob es sich bei der ID 1002 um eine Gruppe oder einen Benutzer handelt? Windows NT weist ein Flag für den Account-Typen zu, das mit jedem Objekt gespeichert wird. Die RID allein ist für die Identifizierung des Account-Typs nicht ausreichend.

Jetzt, da ich verstehe, wie Windows NT und Unix Benutzer und Gruppen darstellen, wie komme ich dann vom einen zum anderen und wieder zurück? Die einfachste Methode, wenn der Samba-Server Mitglied einer NT-kontrollierten Domäne ist, besteht darin, einfach dem Benutzernamen für den NT-Account einen Benutzernamen auf dem Unix-Rechner zuzuweisen.

Moment mal! Hatte ich nicht gesagt, dass ich nicht wollte, dass sich Benutzer mit einem zweiten Account auf dem Samba-Server abgeben müssten? Hier ist die Lösung. Ich erstelle Accounts für die Benutzer auf meinem Linux-Rechner, überlasse aber dem PDC jegliche Authentifizierung. So kann ich alle Passwortfelder in `/etc/passwd` deaktivieren, und die Benutzer haben nur einen Account zu verwalten.

Warum braucht man also überhaupt einen Unix-Account? Eine Sache, die derzeit nicht in Samba implementiert ist, ist die Unterstützung für Windows-NT-Zugriffskontrolllisten auf Freigaben, die Samba zur Verfügung stellt. Daher benutze ich die Standard-Unix-Dateiberechtigungen, was bedeutet, dass jeder Benutzer für seine Arbeit eine Unix-UID und -GID benötigt. Ergibt das einen Sinn?

Die Linux-Benutzer-Accounts und -Gruppen einrichten

Da Sie nun verstehen, warum ich Accounts für die Domänenbenutzer auf dem Linux-Rechner einrichten muss, kann ich aus zwei Optionen wählen. Erstens kann Samba sie, wenn notwendig, automatisch einrichten, wenn ein Benutzer sich verbindet. Die zweite Lösung besteht darin, die Benutzer und Gruppen manuell einzurichten. Mit manuell meine ich nicht, dass Automatisierung über Skripte nicht möglich ist, sondern dass die Accounts ohne Intervention von Samba erstellt werden.

Ich wähle aus folgendem Grund die zweite Option: Der existierende NT-Server hat bereits Dateien, die von Domänenbenutzern erstellt wurden und deren Eigentum sie sind. Dies ist wahrscheinlich in den meisten Fällen so. Ich kann Linux-Benutzer-Accounts aus dem Nichts einrichten, aber es kann schwierig

sein, existierende Gruppenmitgliedschaften korrekt zu übertragen. Ich denke, dies wird noch klarer werden, je weiter ich komme.

Zunächst brauche ich eine Auflistung der Benutzer-Accounts vom PDC. Die Windows-NT-Version des Befehls `net.exe` hat eine weitere Option, die ich bisher noch nicht erwähnt habe. Über die Option `user` kann ich Informationen über lokale und Domänen-Accounts erhalten. Wenn ich den folgenden Befehl ausgebe, erhalte ich eine Auflistung aller Domänenbenutzer. Mein Windows-NT-4.0-PDC ist der Rechner `SALSA`.

```
E:\users>net user /domain

Benutzerkonten für \\SALSA

-------------------------------------------------------------
Administrator            daphnie                  dot
freddie                  Guest                    jerryc
scooby                   shaggy                   velma
wacko                    yacko
Der Befehl wurde erfolgreich ausgeführt.
```

Wenn ich Informationen über einen speziellen Benutzer-Account brauche, kann ich der Option `user` ein weiteres Argument hinzufügen, um anzugeben, welcher Benutzer gesucht werden soll. Um z.B. weitere Informationen über einen Benutzer namens `scooby` zu erhalten, gebe ich den Befehl `net user scooby /domain` aus. Die Ausgabe dieses Befehls sehen Sie in Listing 12.1.

Listing 12.1: Account-Informationen über Domänenbenutzer ausgegeben vom Befehl net user

```
E:\users>net user scooby /domain
Benutzername            scooby
Vollständiger Name
Beschreibung
Benutzerbeschreibung
Ländereinstellung               000 (System-Standardvorgabe)
Konto aktiv                     Ja
Konto abgelaufen                Nie
letztes Setzen des Kennworts    1/21/99 7:48 AM
Kennwort läuft ab               1/22/99 7:48 AM
Kennwort änderbar               1/21/99 7:48 AM
Kennwort erforderlich           Ja
Benutzer kann Kennwort ändern   Ja

Erlaubte Arbeitsstationen         Alle
Anmeldescript
Benutzerprofil                    \\salsa\users\scooby\profile
Basisverzeichnis                  \\picante\users\scooby
Letzte Anmeldung                  Nie

Erlaubte Anmeldezeiten            Alle
```

```
Lokale Gruppenmitgliedschaften
Globale Gruppenmitgliedschaften   *Domain Users    *Accounting
Der Befehl wurde erfolgreich ausgeführt.
```

Nun brauche ich einige Informationen über die Domänengruppen. Es gibt einen analogen Parameter zur Option `user` für den Befehl `net.exe`, der die Liste der lokalen oder Domänengruppen und damit verbundene Informationen anzeigt:

```
E:\users>net group /domain

Gruppenkonten für SALSA

-------------------------------------------------------------
*Accounting          *Dept Heads          *Domain Admins
*Domain Guests       *Domain Users        *Web Developers
Der Befehl wurde erfolgreich ausgeführt.
```

Die Option `group` akzeptiert einen Gruppennamen, wenn ich Informationen über die aktuellen Mitglieder brauche:

```
E:\users>net group "Dept Heads" /domain
Gruppenname          Dept Heads
Beschreibung

Mitglieder

-------------------------------------------------------------
freddie              velma
Der Befehl wurde erfolgreich ausgeführt.
```

Da ich jetzt Zugriff auf alle notwendigen Informationen habe, möchte ich die Accounts einrichten. Auf der CD-ROM zu diesem Buch finden Sie ein einfaches Perl-Skript namens `nt2passwd`, das die Ausgabe des Befehls `net user /domain` akzeptiert und gültige `/etc/passwd`-Einträge produziert. Das Skript richtet außerdem für jeden Benutzer ein Home-Verzeichnis ein. Falls ich mich entscheide, dies nicht zu tun, wird das entsprechende Feld im `passwd`-Eintrag auf `/dev/null` gesetzt. Hier ist die Ausgabe, nachdem ich die Liste der Domänenbenutzer durch das Skript laufen ließ. Zuerst habe ich die Ausgabe von `net user /domain` abgefangen und habe sie mit einer Datei verkettet:

```
E:\users>net user /domain > users.txt
```

Dann habe ich die Datei `users.txt` auf meinen Linux-Rechner übertragen und das `nt2passwd`-Skript laufen lassen:

```
# ./nt2passwd users.txt
Enter the uid to start with: 1000
Enter the gid to use: 100

Do you want to create a home directory for the users? (y/n) y
Please enter the base directory for the users home:  /export/home
Do you want me to make the home directories for you? (y/n) y
```

```
Please enter a username for [administrator] of 8 characters or less: ntadm
```

Sie haben wahrscheinlich bereits bemerkt, dass Linux keine Benutzernamen mit mehr als acht Zeichen unterstützt. Wenn das `nt2passwd`-Skript einen nicht zugelassenen Namen entdeckt, fordert es Sie auf, einen gültigen Benutzernamen einzugeben. Existiert der neue Benutzername bereits, fragt das Skript, ob der Benutzer einen anderen Namen probieren möchte. Der Benutzer kann den Account komplett überspringen, indem er mit einem `n` antwortet. Eine Datei mit Einträgen in der Form von

`UnixBenutzername=NTBenutzername`

wird erstellt und erhält den Namen `Benutzername.map`, um die Entsprechungen aufzuzeichnen. Ich kann diese Datei mit dem `smb.conf`-Parameter `username map` benutzen, damit Windows-NT-Benutzernamen korrekt mit einem gültigen Unix-Account verbunden werden. Folgende Map-Datei wird bei meinem Beispiel erzeugt:

```
ntadmin=administrator
```

`nt2passwd` erstellt eine Datei namens `passwd.new`, die alle neu eingerichteten Accounts enthält. All dies ist notwendig, um die Datei an meine existierende Datei `/etc/passwd` anzuhängen. `nt2passwd` überprüft, ob es UID-Konflikte gibt, wenn die Accounts eingerichtet werden, also ist dies kein Problem:

```
# cat passwd.new >> /etc/passwd
```

Danach muss ich Einträge für die NT-Domänengruppen in der `/etc/group` einfügen. Auch hierfür benutze ich wieder die Ausgabe des Befehls `net.exe`, um die Einträge zu erstellen. Zunächst leite ich die Ausgabe von `net group /domain` in eine Datei weiter:

```
net group /domain > groups.txt
```

Danach gebe ich die Ausgabe des Befehls `net group` an das Perl-Skript namens `nt2group` weiter. Dieses Skript nimmt einen zusätzlichen Parameter an, nämlich den Namen einer Datei, die den entsprechenden Linux-Gruppennamen für die Windows-NT-Domänen-Gruppennamen enthält. Pro Zeile wird eine Zuordnung eingegeben, die beiden Namen werden dabei durch einen Doppelpunkt (`:`) getrennt. Hier ist die Beispielzuordnungsdatei, die ich benutze. Die NT-Gruppennamen sind links und die Linux-Gruppennamen rechts dargestellt:

```
accounting:acct
dept heads:dptheads
domain users:users
web developers:webdev
```

Nach Starten des `nt2group`-Skripts

```
# nt2group groups.txt group.map
Enter the gid to start with: 200
```

werden die folgenden Einträge in einer Datei namens `group.new` erstellt. Es gibt keinen Eintrag für die Gruppe `users`, weil diese Gruppe auf meinem Linux-Rechner bereits existiert.

```
acct:*:200:
dptheads:*:201:
webdev:*:203:
```

Ich hänge diese Einträge an die auf dem System vorhandene Datei `/etc/group` an, um die Gruppen zu erstellen:

```
cat group.new >> /etc/group
```

Jetzt sind die Benutzer und die Gruppen eingerichtet. Der letzte Schritt besteht darin, die Gruppen mit den entsprechenden Benutzernamen zu füllen. Dafür verwende ich wieder das nützliche Utility `net.exe`. Denken Sie daran, dass Sie die Mitglieder einer Gruppe bestimmen können, indem Sie den Befehl `net group` *`Gruppenname`* `/domain` ausführen. Auf der CD-ROM finden Sie ein weiteres Perl-Skript namens `add2group`, das diese Ausgabe benutzt und die aktualisierte Version der Datei `/etc/group` an Standard Output weiterleitet.

Schauen Sie sich dazu ein Beispiel an. Hier ist die Ausgabe von `net group Accounting /domain` für die Domäne `CHIPSNDIPS`:

```
Gruppenname     Accounting
Beschreibung

Mitglieder

-------------------------------------------------------------
daphnie            scooby           velma
Der Befehl wurde erfolgreich ausgeführt.
```

Hier ist die Beispielausführung für die Linux-Gruppe namens `acct` über die gleiche Gruppenzuordnung, die ich vorher mit dem Tool `nt2group` durchgeführt habe. Der erste Parameter ist die Ausgabe des Befehls `net group Accounting`, das zweite Argument ist die Gruppen-Zuordnungsdatei, die ich mit dem `nt2group`-Skript verwendet habe, und der letzte Befehlszeilenparameter ist die Datei, die die Zuordnungen der Benutzernamen enthält und vom `nt2passwd`-Skript erstellt wurde.

Das Skript zeigt tatsächlich auch die nicht modifizierten Einträge an, aber ich habe die Ausgabe hier aus Platzgründen gekürzt:

```
# add2group accounting.txt group.map username.map
[...Ausgabe gelöscht...]
acct:*:200:daphnie,scooby,velma
[...Ausgabe gelöscht]
```

Ich wiederhole den gleichen Schritt für jede Domänengruppe. Hier sind die Ergebnisse für die vier Gruppeneinträge:

```
users:100:games
acct:*:200:daphnie,scooby,velma
dptheads:*:201:freddie,velma
webdev:*:203:freddie,jerryc,shaggy
```

Beachten Sie, dass für die Gruppe `users` keine Namen aufgelistet sind. Das liegt daran, dass ich den Benutzer dem Eintrag in `/etc/group` nicht hinzufügen muss, wenn dies die primäre Gruppen-ID des Benutzers ist, wie sie in `/etc/passwd` definiert ist. Das Skript lässt alle Benutzernamen aus, die nicht in `/etc/passwd` gefunden werden. Wenn ich z.B. entscheide, keinen Account für `Administrator` einzurichten, wäre er einfach übergangen worden, wenn er Mitglied der Gruppe `Accounting` gewesen wäre.

Nun habe ich endlich die Benutzer-Accounts eingerichtet, die Gruppen erstellt und den notwendigen sekundären Gruppen Benutzer hinzugefügt. Wow! Das war viel Arbeit! Glücklicherweise ist dies der härteste Teil des Prozesses für den Ersatz des Windows-NT-Servers.

Ich habe vorher erwähnt, dass Samba, wenn notwendig, Benutzer-Accounts automatisch für Sie einrichten kann, wenn sich ein bestimmter Benutzer verbindet. Dies können Sie erreichen, indem Sie ein Skript für den `smbd` definieren, das gestartet wird, wenn ein Benutzer hinzugefügt oder entfernt werden soll. Diese zwei globalen Parameter, `add user script` und `delete user script`, sind neu in Version 2.0. Sie sind Teil einer Entwicklung, das zur Verfügung zu stellen, was »Gerätemodus« genannt wird. Diese Funktion ermöglicht die Anwendung eines Samba-Gerätes in einem Windows-NT-Netzwerk, das hauptsächlich aus Adressparametern besteht. Alle Benutzer und Gruppen können aus dem Nichts nach Bedarf erstellt werden. Halten Sie sich auf dem Laufenden, da Samba ständig weiterentwickelt wird, um diese Art von Funktion zu unterstützen.

Wenn Sie vorhandene Dateien übertragen, die verbundene Zugriffskontrolllisten haben, ist es besser, die Dinge außerhalb von Samba zu konfigurieren, so wie ich es getan habe.

Schritt 2: Dateien und Druck-Spool-Dateien verschieben

Das Verschieben von Dateien von einem Windows-NT-Server auf meinen Linux-Rechner könnte ein absoluter Alptraum sein! Es gibt jedoch einige gute System-Management-Techniken, die dabei helfen können, die Dinge etwas einfacher zu machen. Ich kann den notwendigen Aufwand vermindern, indem ich den Verzeichnisbaum organisiere und Dateien mit gemeinsamen Eigentümern und gemeinsamen ACLs gruppiere.

[users]

Ich fange damit an, einen genaueren Blick auf die `[users]`-Freigabe zu werfen, die der NT-Datei-Server bietet. Der Aufbau der Freigabe entspricht dem Exportieren des Verzeichnisses `/home/` zu anderen Unix-Rechnern über NFS und dem Platzieren der Home-Verzeichnisse der Benutzer eine Ebene tiefer. Anders gesagt, `\\PICANTE\users` enthält ein Verzeichnis für jeden Domänenbenutzer. Das Home-Verzeichnis eines Benutzers ist tatsächlich `\\PICANTE\users\`*`Benutzername`*.

Die Verzeichnisberechtigungen für Ordner, die in der `[users]`-Freigabe enthalten sind, sind sehr direkt. Jeder Benutzer hat vollständige Kontrolle über sein entsprechendes Verzeichnis. Nur ein Administrator (z.B. `PICANTE\Administrator`) kann Verzeichnisse auf oberster Ebene in der Freigabe erstellen. Dies bedeutet die Unix-Berechtigungen.

```
drwx------   Benutzername   Gruppenname   Benutzername/
```

Um Dateien in Home-Verzeichnisse zu übertragen, brauche ich nur die Dateien des Benutzers an den in `/etc/passwd` definierten Standort zu kopieren. Dann kann ich über den Befehl `chown` die Berechtigungen einrichten:

```
chown -R Benutzername Home-Verzeichnis
chgrp -R Gruppenname Home-Verzeichnis
chmod -R 700 Home-Verzeichnis
```

Ich ersetze *`Benutzername`* durch den entsprechenden Namen und *`Home-Verzeichnis`* durch den absoluten Pfad zum Home-Verzeichnis des Benutzers. *`Gruppenname`* ist die primäre Gruppen-ID des Benutzers aus `/etc/passwd`.

Nun definiere ich die `[users]`-Freigabe in `smb.conf`, um den gleichen Dienst wie vorher zu bieten:

```
[users]
comment = Home-Verzeichnisse für Domänenbenutzer
path = /export/home
```

```
create mode = 0600
directory mode = 0700
```

[docs]

Die Freigabe `[docs]` ist aufgrund der Unterschiede in den Datei- und Verzeichnis-ACLs etwas schwieriger zu übertragen. Ich muss mich mit zwei Möglichkeiten befassen. Die eine besteht darin, dass ein einzelner Benutzer oder eine einzelne Gruppe in der Zugriffskontrollliste benutzt wird. Dann kann das Verzeichnis in ein entsprechendes Unix-Berechtigungsbit-Modell umgewandelt werden, aber bei der anderen Möglichkeit ist eine Entsprechung nicht so leicht zu realisieren. Es kommt zu einem Problem, wenn mehrere Benutzer oder Gruppen in der Zugriffskontrollliste eingefügt sind.

Schauen wir uns zunächst die Möglichkeit des Zugriffs von einem einzelnen Benutzer oder einer einzelnen Gruppe an und wie Sie damit umgehen können.

Abb. 12.2: Ein Diagramm der Verzeichnis-Zugriffskontrolllisten in der Freigabe [docs]

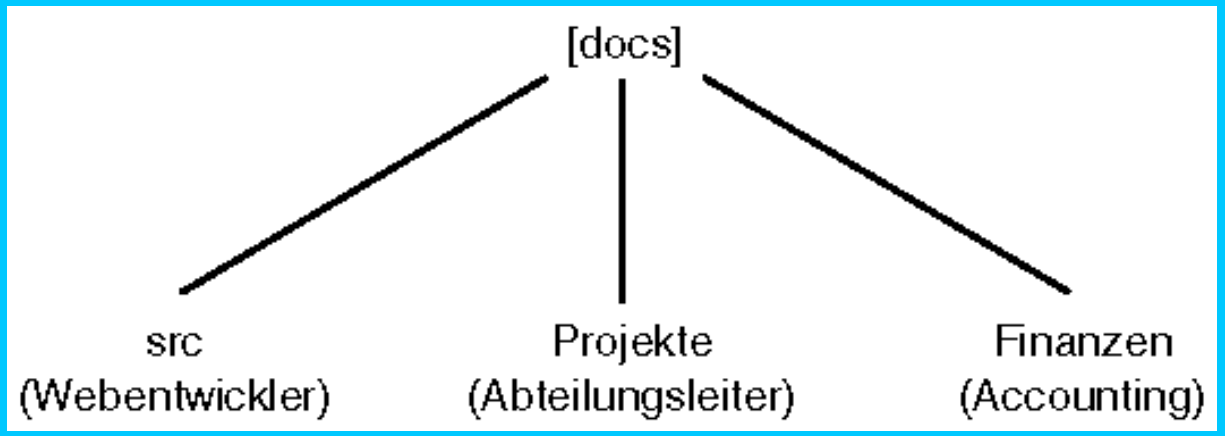


Abbildung 12.2 zeigt die verschiedenen ACLs auf Verzeichnissen innerhalb des `[docs]`-Verzeichnisbaums. Obwohl die verschiedenen Verzeichnisse unterschiedliche Besitzer haben, nehme ich an, dass der Zugriff weitestgehend begrenzt ist. So besitzt z.B. die Gruppe `acct (Accounting)` das Verzeichnis `\\PICANTE\docs\finances` und alles, was darunter liegt, und die Gruppe `webdev` (`Web Development`) kontrolliert den Verzeichnisbaum `\\PICANTE\docs\src`. Diese Art von Zugriffskontrolle kann über die Unix-Eigentümer- und -Gruppenberechtigungsbits dargestellt werden. Ich werde außerdem das Gruppen-ID-Bit einrichten, damit sich die Gruppeneigentumsverhältnisse im entsprechenden Verzeichnis nach unten fortsetzen:

```
drwxrws---   root   acct     finances/
-rw-rw----   root   acct     finances/department.xls
drwxrws---   root   webdev   src/
-rw-rw----   root   webdev   src/calendar.html
```

Unter Unix können Verzeichnisse nicht in Besitz mehrerer Benutzer oder Gruppen gleichzeitig sein. Unter Windows NT übrigens auch nicht, aber das Unix-Berechtigungsmodell basiert ausschließlich auf den Eigentumsverhältnissen. Es gibt keine Methode, Mitglieder auch nur zweier Gruppen auf Dateien zugreifen zu lassen. Es ist entweder Zugriff für eine Gruppe oder allgemeiner Zugriff. Windows NT trennt Besitztum von Zugriffsrecht durch die Anwendung von Zugriffskontrolllisten, die mehreren Benutzern und Gruppen Einträge ermöglichen, wobei jeder eine eindeutige Einstellung für die Berechtigung hat. Die einzige Methode, dies ohne richtige NT-ACL-Unterstützung zu umgehen, besteht in der Verwendung einer Kombination der `smb.conf`-Parameter `valid users`, `write list`, `read list` und `force user`. Im Wesentlichen werden die ersten drei Parameter die Datei-

oder Verzeichnis-ACL. Der Parameter `force user` kann nützlich sein, aber alle »Ersteller-Besitzer«-Informationen gehen verloren.

Mein Beispiel setzt einen einzelnen Benutzer oder eine einzelne Gruppe in der ACL voraus. Ich glaube nicht, dass diese Voraussetzung die Nützlichkeit von Samba in dieser bestimmten Umgebung verschlechtert, sondern es lässt einfach den Bedarf für eine besser organisierte Verzeichnisstruktur aufkommen. Obwohl NT-ACLs derzeit in Version 2.0 noch nicht unterstützt werden, sind Entwicklungen im Gange, um die notwendigen Mechanismen zu implementieren.

Bevor Sie die Dateien verschieben, sollten Sie sich Notizen über die aktuellen Verzeichniseigentumsverhältnisse und die relevanten ACL-Informationen machen. Dann kann ich die entsprechenden Berechtigungen manuell auf dem Linux-Server einrichten. Wenn z.B. `\\PICANTE\docs\Projects` in Besitz der Gruppe `Dept Heads` wäre und zumindest »Change«-Berechtigungen für das Verzeichnis eingerichtet wären (d.h. `rwxd`, falls Sie nicht mit Windows-NT-ACLs vertraut sind), würde ich die entsprechenden Berechtigungen auf dem Linux-Rechner einrichten, indem ich eine Reihe von `chown`-, `chgrp`- und `chmod`-Befehlen ausführe:

```
# chown -R root /export/docs
# chgrp -R dptheads /export/docs
# chmod -R 770 /export/docs
# chmod -R g+s /export/docs
# ls -ld /export/docs
drwxrws---   root   dptheads   /export/docs/Projects/
```

Um dies zu aktivieren, konfiguriere ich die `[docs]`-Freigabe in `smb.conf` dahingehend, dass für Dateien immer die Gruppen-Lese-Schreib-Bits eingerichtet sind und für Verzeichnisse Lesen-Schreiben-Ausführen:

```
[docs]
   comment = domain group share
   path = /export/docs
   create mode = 0660
   directory mode = 0770
```

[canon]

Ich werde das Drucken von einem Windows-Client zu einem Samba-Server nicht im Detail darstellen. Aber es gibt zwei Dinge, die Sie beachten sollten.

Ein Windows-NT-Rechner benutzt spezielle Mechanismen, um an einen anderen Windows-NT-Rechner zu drucken. Diese unterscheiden sich von dem Aufruf, den er macht, um an einen Windows-9x-Server oder einen Samba-Server zu drucken. Das bedeutet, dass es derzeit noch nicht möglich ist, einen Samba-Server gegen einen Windows-NT-Drucker-Server auszutauschen, ohne einiges an Vorarbeiten zu leisten. Diese beinhalten normalerweise eine geringfügige Neukonfiguration der Netzwerkdruckerverbindung des Clients. Das Problem mit der Implementierung echten NT-Druckens liegt darin, dass, wenn Sie einen Teil der Funktionalität unterstützen, Windows NT erwartet jedoch, dass Sie alles unterstützen. Unterstützung für dies sollte bald in Samba verfügbar sein. Teile des Codes wurden bereits geschrieben und der Rest befindet sich in der Entwicklung.

Der zweite Punkt, der tatsächlich mit dem ersten Problem verwandt ist, ist, dass Samba das Herunterladen von Druckertreiberdateien nicht unterstützt, wenn ein Client sich erstmals mit einem Drucker verbindet. Werfen Sie einen Blick in Kapitel 8, »Drucker«, wenn Sie diesen Punkt noch einmal durchgehen möchten. Auch hier können Sie die Funktionalität bald erwarten.

Wenn Sie eine Drucker-Spool-Datei von einem NT-Server an einen Samba-Server übertragen, sollte Ihnen klar sein, dass Samba administrative Domänengruppen, wie z.B. die »Druckoperatoren«, nicht im wahrsten Sinne des Wortes unterstützt. Wenn nötig, können Sie eine Art von Unterstützung

implementieren, indem Sie den Freigabeparameter `valid users` benutzen. Abgesehen von diesen Dingen, auf die Sie achten sollten, läuft die Konfiguration Ihres Druckers genau so, wie sie in Kapitel 8 dargestellt wurde. Hier ist die Druckerfreigabe, die ich benutzen werde:

```
[canon]
   print command = lpr -P%p %s; rm %s
   comment = domain printer
   printable = yes
   writeable = no
   public = no
```

Natürlich muss ich den Drucker auch korrekt in der lokalen `/etc/printcap`-Datei konfigurieren.

Schritt 3: smb.conf konfigurieren

Ich habe es fast geschafft. Als Nächstes muss ich den Abschnitt `[global]` der Datei `smb.conf` konfigurieren. Zunächst nehme ich mir die Parameter vor, mit denen ich bereits sehr vertraut bin: `netbios name` und `workgroup`:

```
[global]
netbios name = PICANTE
workgroup = CHIPSNDIPS
```

Damit erfülle ich die zweite Anforderung, die ich am Anfang dieses Kapitels dargestellt habe: Der Rechner sollte als der gleiche wie der NT-Server auftauchen, wenn er über einen Netzwerk-Browsing-Mechanismus wie die Netzwerkumgebung angesehen wird.

Danach muss ich den Sicherheitsmodus konfigurieren. Aus Platz- und Zeitgründen werde ich nicht detailliert darstellen, wie Domänensicherheit im Netzwerk implementiert wird. Statt dessen sollten Sie mir einfach glauben, dass Samba mit diesem Sicherheitsmodus als vollständiges Domänenmitglied agieren und an Vertrauensbeziehungen teilnehmen kann:

```
security = domain
```

Damit Samba im Domain-Modus arbeiten kann, muss ich einen Server definieren, der Authentifizierungsanfragen validieren kann, ganz ähnlich wie im Server-Modus. Der Parameter `password server` funktioniert hier ganz genau so wie mit der Einstellung `security = server`. Der Passwort-Server sollte der PDC für meine Domäne sein. Habe ich mehrere BDCs (Backup Domain Controllers), könnte ich diese ebenfalls in die Liste einfügen:

```
password server = QUESO
```

Abschließend muss ich angeben, dass ich verschlüsselte Passwörter verwenden werde, obwohl die Verwaltung einer zusätzlichen `smbpasswd`-Datei nicht notwendig ist. Dieser Parameter schaltet das Flag in der Antwort zur Protokollverhandlung ein, das anzeigt, ob ich Passwortverschlüsselung unterstütze:

```
encrypt passwords = yes
```

Es ist besser, dem Abschnitt `[global]` Ihrer `smb.conf` auch die folgenden Parameter hinzuzufügen, es sei denn, Sie wissen ganz genau, was Sie tun:

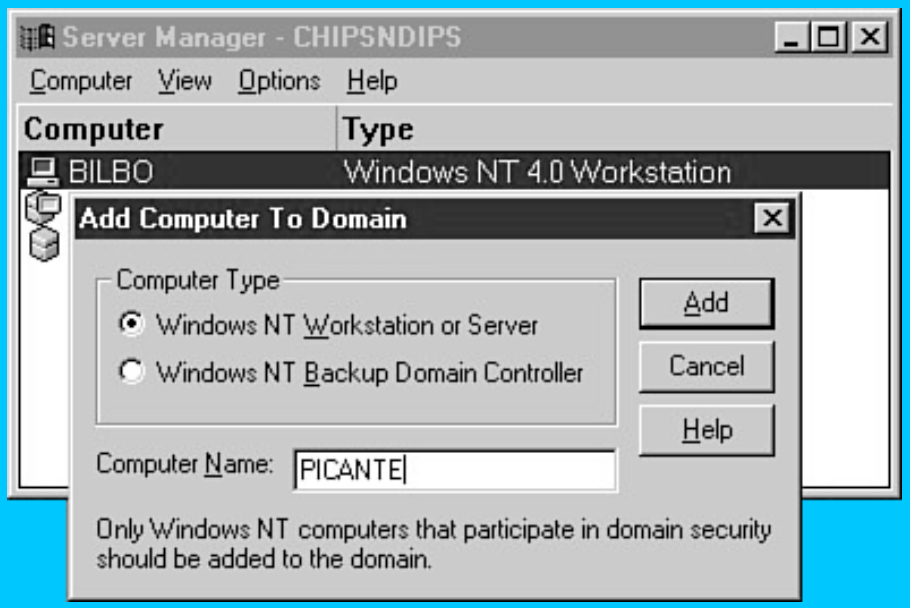
```
os level = 0
domain master = no
local master = no
preferred master = no
```

Der Grund dafür liegt darin, dass ein Windows-NT-PDC der Domain-Master-Browser sein muss. Dies wird in Kapitel 19, »Browsing des lokalen Subnetzes«, klarer werden.

Schritt 4: Den Samba-Server an der NT-Domäne teilnehmen lassen

Das Einfügen eines Samba-Servers in eine NT-Domäne ist ein Prozess, der in zwei Schritten ausgeführt wird. Derzeit unterstützt Samba nicht die Option, einen Rechner-Account in der Domäne beim Eintritt einzurichten; daher muss ich über den Server-Manager für Domänen einen Account auf dem PDC einrichten. Abbildung 12.3 zeigt den Prozess für das Einfügen eines neuen Servers in die Domäne. Falls Sie sich fragen, warum ich nicht einfach den existierenden Rechner-Account für `PICANTE` benutzen kann, so liegt das daran, dass der Samba-Rechner keine Ahnung hat, wie das Passwort für den Account lautet. Zwar ist es möglich, diese Information herauszufinden, aber es ist die einfachste Lösung, den Server der Domäne neu hinzuzufügen.

Abb. 12.3: Über den Server-Manager für Domänen einen Rechner-Account in der Domäne CHIPSNDIPS einrichten



Ist der Account eingerichtet, kann ich über das Tool `smbpasswd` der Domäne beitreten. Es ist sehr wichtig, dass weder `smbd` noch `nmbd` laufen, während ich versuche, der Domäne beizutreten. Ich sollte außerdem sicherstellen, dass das Verzeichnis `/usr/local/samba/private` existiert, da `smbpasswd` hier das aktuelle Rechnerpasswort für den Samba-Server speichern wird. Um der Domäne tatsächlich beizutreten, geben Sie den folgenden Befehl aus:

```
# /usr/local/samba/bin/smbpasswd -j CHIPSNDIPS 1999/01/21 22:43:38 : change_trust_account_password:
```

```
Changed password for domain CHIPSNDIPS.
Joined domain CHIPSNDIPS.
```

Natürlich sollten Sie `CHIPSNDIPS` mit dem Namen Ihrer Domäne ersetzen. Jetzt ist es an der Zeit, `smbd` und `nmbd` zu starten und die Dinge auszuprobieren.

Schritt 5: Die Konfiguration testen

Die beste Methode sicherzustellen, dass alles korrekt funktioniert, besteht darin, sich in einen existierenden NT-Client einzuloggen und zu sehen, ob alles gleich aussieht. Nachdem ich mich in eine Windows-NT-Workstation eingeloggt habe, überprüfe ich zunächst, dass ich auf mein Home-Verzeichnis zugreifen kann. Abbildung 12.4 zeigt, dass die Netzwerkfreigabe `\\PICANTE\users` auf Laufwerk H: gemountet ist. Abbildung 12.5 zeigt, dass sogar mein Befehlsprompt im korrekten Verzeichnis startet!

Abb. 12.4: Mein Arbeitsplatz zeigt die aktuellen Netzwerk-Laufwerksverbindungen

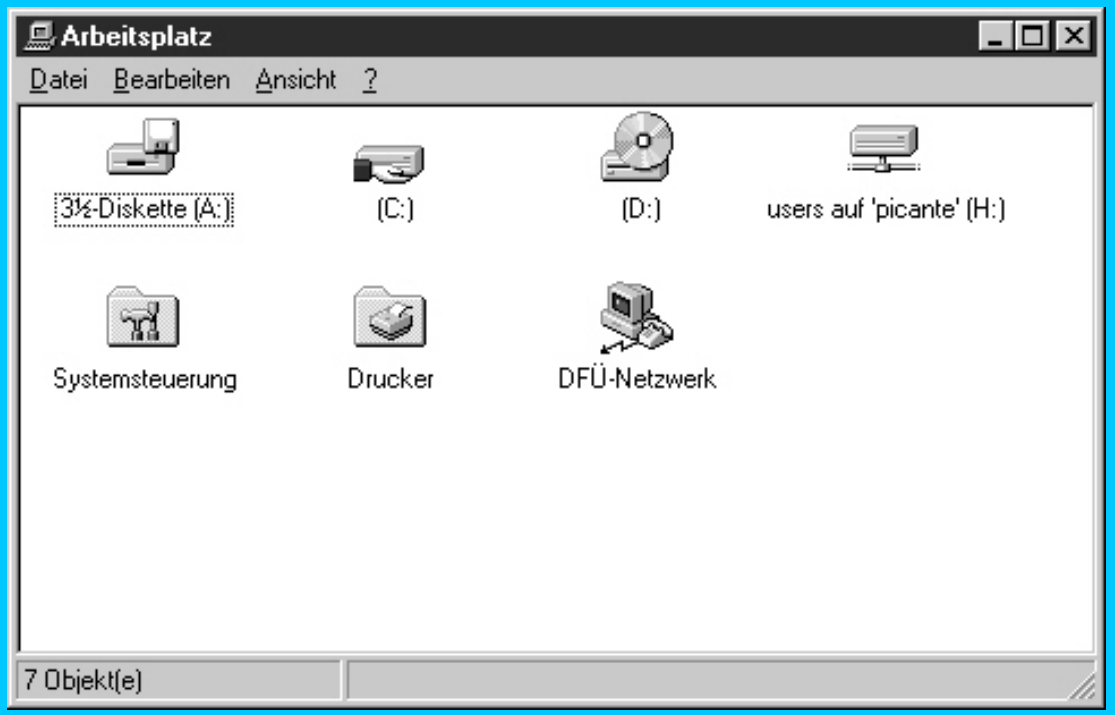
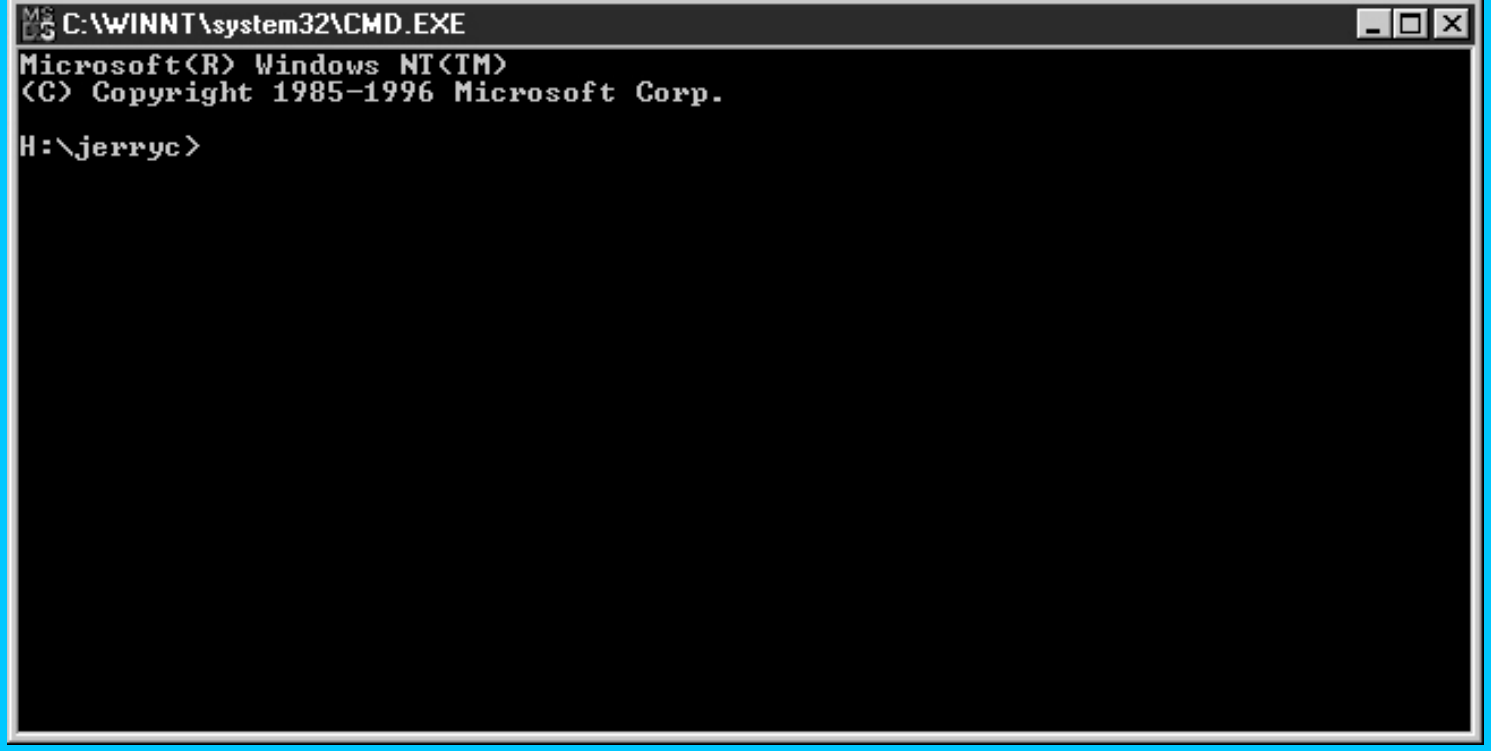


Abb. 12.5: Windows-NT-Befehlsprompt startet standardmäßig im Home-Verzeichnis des Benutzers



Danach überprüfe ich das Netzwerk-Browsing. Der Server `PICANTE` erscheint in der Browse-Liste in Abbildung 12.6 wie vorgesehen. Nachdem ich sichergestellt habe, dass der Server die korrekte Freigabeliste (siehe Abbildung 12.7) anzeigt, warte ich einfach, ob jemand die Auswechslung der Server bemerkt.

Abb. 12.6: Die Netzwerkumgebung zeigt den aktuellen Rechner für CHIPSNDIPS

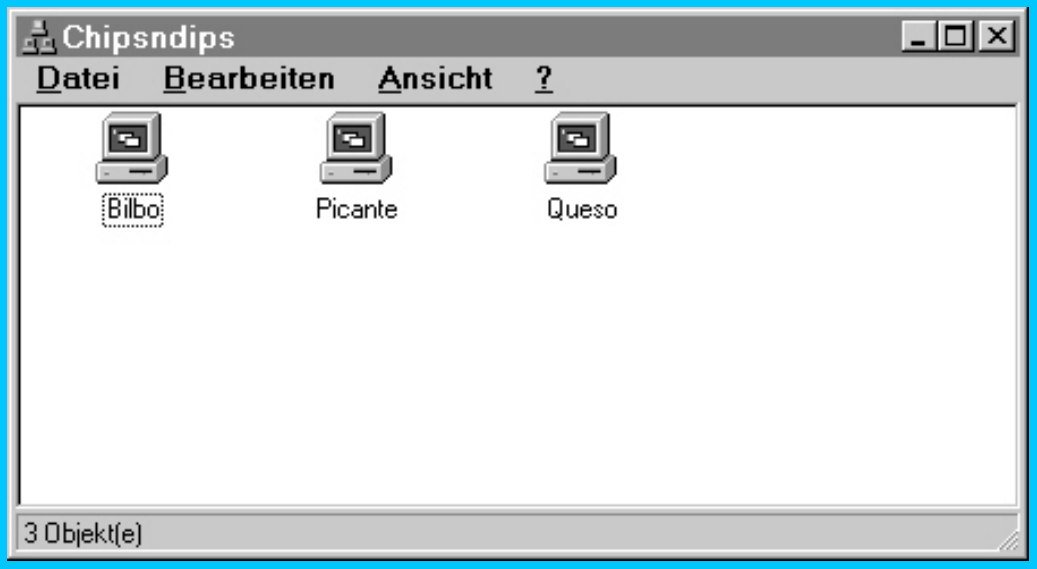
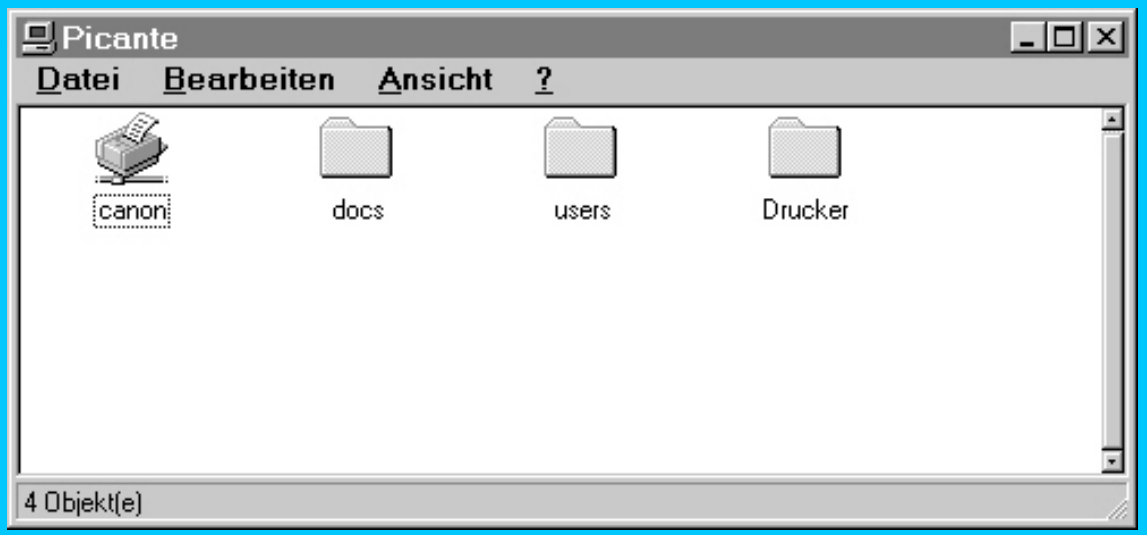


Abb. 12.7: Die Liste der Freigaben für PICANTE



Zusammenfassung

Obwohl Samba nicht alle NT-Funktionen unterstützt, kann es einen Windows-NT-Datei- und Drucker-Server in einem existierenden Netzwerk effektiv ersetzen. Samba entspricht den drei Anforderungen, die ich gestellt habe:

- Alle Domänenbenutzer sollten auf die neuen Freigaben auf dem Samba-Server zugreifen können, ohne dass ein synchronisierter Unix-Account auf dem Rechner verlangt wird.

  Der vorhandene Windows-NT-Domain-Controller führt sämtliche Login-Authentifizierungen durch.
- Der Samba-Server sollte in der gleichen Arbeitsgruppe erscheinen und den gleichen NetBIOS-Rechnernamen wie der existierende Server verwenden, um die Verwirrung für die Benutzer so gering wie möglich zu halten.

  Die `smb.conf`-Parameter `netbios name` und `workgroup` ermöglichen es uns, die Identität des alten Servers anzunehmen.
- Die Zugriffskontrollmechanismen für Dateien sollten die gleichen sein, damit ein Benutzer, der Zugriff auf eine Datei auf dem existierenden Server hat, auch auf die gleiche Datei auf dem Samba-Server zugreifen kann. Außerdem sollte ein Benutzer, der keinen Zugriff auf eine Datei auf dem existierenden Server hat, auch unter der neuen Konfiguration nicht auf diese Datei zugreifen können.

  Durch zugelassene Einträge in der `/etc/passwd` und `/etc/group` kann ich die Standard-Unix-Berechtigungsbits für die Zugriffskontrolle benutzen.

Frage & Antwort

F. Warum ist security = domain besser als security = server?

. Es gibt zwei Gründe, warum `security = domain` besser ist. Erstens kann Samba mit dieser Methode an den Vertrauensverhältnissen der

Domäne teilnehmen. Dies ist im Server-Modus nicht möglich. Zweitens muss im Server-Modus jeder `smbd`-Prozess eine offene Verbindung mit dem Authentifizierungsserver beibehalten. Dies kann einen Windows-PDC schnell überfordern. Im Domain-Modus ist diese Verbindung so lange notwendig, bis die Authentifizierung durchgeführt ist, und so werden wertvolle Ressourcen gespart.

Neue Begriffe

SID - Der Securtiy Identifier, der aus einem Zahlenstring besteht und benutzt wird, um zwischen NT-Rechnern und -Accounts zu unterscheiden.

RID - Eine 32-Bit-Nummer, die Windows NT verwendet, um eine Benutzer-ID mit einem relativen Identifier zu bezeichnen. Die RID wird an die Rechner- oder Domänen-SID angehängt, um den Account vollständig zu qualifizieren.

BDC - Ein Backup Domain Controller ist ein Rechner, der Account-Informationen zur Domäne vom Primary Domain Controller repliziert, um die Last der Authentifizierungsverbindungen zu verteilen.

ACL - Eine Zugriffskontrollliste (Access Control List) ist ein Attribut, das mit Dateien, Verzeichnissen und Druckern verbunden ist und die Möglichkeit zur Manipulation eines bestimmten Objekts einschränkt oder gewährt.

Tag 13: Unix (smbclient, smbfs, smbwrapper und andere Utilities)

von Richard Sharpe

Viele Leute denken, dass SMB-Clients nur für DOS- und Windows-basierte Systeme verfügbar sind. Für einige Leute ist es eine große Überraschung, dass es auch Unix-basierte SMB-Clients geben könnte. Welcher Nutzen sollte überhaupt darin bestehen, von einem Unix-Rechner auf SMB-Server zugreifen zu können?

Tatsächlich gibt es mehrere Möglichkeiten für SMB-Clients von Unix-Rechnern, und Samba bietet einige davon, während andere Entwickler weitere zur Verfügung stellen. Diese Clients sind u.a.:

- `smbclient`, ein Standardbestandteil von Samba, das ein Befehlszeilen-Utility für den Zugriff auf SMB-Server bietet. Es kann verwendet werden, um Dateien zwischen Unix- und Windows-Rechnern zu kopieren oder Dateien von einem SMB-Server zu sichern, wird aber auch von einer Reihe anderer Utilities benutzt, wie z.B. `smbprint` und `smbtar`.
- `smbfs`, ein virtuelles Dateisystem für Linux, über das Windows-Dateisysteme auf Unix-Systemen gemountet werden können.
- `smbwrapper` und `smbsh`, die neuesten Utilities, über die Benutzer Dateisysteme auf Windows-Systemen innerhalb einer Unix-Shell durchsuchen können.
- Sharity, ein weiteres virtuelles Dateisystem, das auch SSL-Unterstützung bietet.
- Verschiedene Utilities, die `smbclient` für die Durchführung ihrer Funktionen verwenden, wie z.B. `smbprint` und `smbtar`.

Zusätzlich verwenden auch andere Open-Source-Pakete Samba-Clients wie `smbclient`, um Teile ihrer Funktionalität zur Verfügung zu stellen. Ein bemerkenswertes Beispiel ist das Netzwerk-Backup-Utility Amanda, das `smbclient` für das Backup von Windows-Clients verwendet.

Dieses Kapitel stellt diese Clients dar, untersucht, wofür und wie sie benutzt werden, und vergleicht ihren Nutzen

smbclient

`smbclient` ist ein CIFS/SMB-Client-Programm für Unix. Es ist ein Befehlszeilen-Utility, das dem bekannten FTP-Utility ähnlich ist. Das folgende Beispiel zeigt eine `smbclient`-Sitzung, bei der sich der Benutzer mit einem Windows-95-System verbunden und alle Dateien in der Freigabe aufgelistet hat:

```
[root@eagle samba-book]# smbclient //eagle/first-share
Added interface ip=16.153.112.110 bcast=16.153.112.255 nmask=255.255.255.0
Password:
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0beta4]
smb: \> ls
   file-1.txt           69 Mon Jan 11 15:35:14 1999
   file-2.txt           59 Mon Jan 11 15:35:14 1999
   Neuer Ordner    D     0 Sat Jan 16 12:21:56 1999
       55729 blocks of size 16384. 3516 blocks available
smb: \> ^D
```

Hier haben Sie sich mit `\\eagle\first-share` verbunden und die Dateien darin aufgelistet. Sie hätten sich jedoch ebenso leicht mit einem anderen Windows-System verbinden können. Beachten Sie, dass ich nicht das Standard-DOS/Windows-Namensschema von `\\eagle\first-share` verwendet habe. Die korrekte Verwendung des DOS-Windows-artigen Namens hätte verlangt, alle Backslashes zu verdoppeln, da Unix sie als ein Escape-Zeichen behandelt. Da `smbclient` sowohl Slashes als auch Backslashes versteht, ist es daher einfacher, erstere zu verwenden.

Für die Puristen unter Ihnen hier aber auch noch die zweite korrekte Form des `smbclient`-Befehls:

```
smbclient \\\\eagle\\first-share
```

smbclient teilt uns dann mit, dass es ein Interface hinzufügt, und fragt uns nach einem Passwort. Sie werden immer zur Eingabe eines Passworts aufgefordert, aber es ist nicht immer unbedingt eins notwendig. Da Sie sich mit Eagle verbunden haben, einem Unix-Rechner mit Samba im User-Modus, war ein Passwort notwendig. smbclient sendet immer den Benutzernamen des eingeloggten Benutzers (es sei denn, Sie haben in der Befehlszeile -U *Benutzername* eingegeben), und Sie sollten das Passwort für den eingeloggten Benutzer auf dem entfernten Rechner (Eagle) eingeben. Ist für den Benutzer auf dem entfernten Rechner kein Passwort notwendig oder verwendet der entfernte Rechner den Share-Modus und hat die Freigabe kein Passwort, drücken Sie einfach [Return].

Wenn smbclient sich mit dem entfernten Rechner verbindet und auf die entsprechende Freigabe zugreifen kann, wird Ihnen ein Prompt präsentiert, an dem Sie einen Befehl eingeben können. Sie haben **ls** eingegeben und erhielten eine detaillierte Auflistung der Dateien in der Freigabe, die Sie vorher gesehen haben. Jede Datei und jedes Verzeichnis wird in einer Zeile dargestellt, die den Namen, eventuell das D-Flag (wenn es sich um ein Verzeichnis handelt), die Größe und das Erstellungsdatum zeigt.

Natürlich bietet smbclient noch viele weitere Funktion, darunter das Kopieren von Dateien in beide Richtungen. Wenn Sie weitere Informationen zu smbclient-Befehlen wünschen, geben Sie in der Befehlszeile **help** ein. Alle smbclient-Befehle werden später in diesem Kapitel dargestellt, aber zunächst sollten Sie einen Blick auf die Befehlszeilenoptionen von smbclient werfen.

smbclient-Befehlszeilenoptionen

Generell hat ein smbclient-Befehl folgende Form:

```
smbclient Freigabename [Passwort] [Optionen]
```

Freigabename ist hier in Form von //Server/Freigabe (oder in Form des alternativen DOS/Windows-artigen Freigabenamens, wie vorher) dargestellt, *Passwort* ist das Passwort für die Freigabe oder den Benutzer, der sich in den Server einloggt, und *Optionen* sind alle, die in diesem Abschnitt aufgelistet sind.

[-s *smb.conf*]

smbclient benutzt die Samba-Konfigurationsdatei smb.conf, um einige Aspekte seiner Funktionen zu kontrollieren. Über diesen optionalen Parameter können Sie eine andere *smb.conf*-Datei spezifizieren.

Diese Option war in Samba-Versionen vor 2.0.0 nicht dokumentiert.

[-B *IP-Adresse*]

Um die IP-Adresse des spezifizierten Servers aufzulösen, überträgt smbclient NetBIOS-Name-Service-Anfragen per Broadcast. Über diese Option können Sie die zu verwendende Broadcast-Adresse spezifizieren.

Diese Option war in den Manpages in Samba-Versionen vor 2.0.0 nicht dokumentiert.

[-O *Socket-Optionen*]

Über diese Option können Sie Socket-Optionen auf TCP-Ebene für die zum Server gerichtete Verbindung einrichten. Weitere Informationen über die für diese Option zugelassenen Werte finden Sie beim Parameter socket options in der smb.conf(5)-Manpage.

[-R *name resolve order*]

smbclient kann eine Reihe von Systemeinrichtungen verwenden, um Server-Namen in IP-Adressen aufzulösen. Über diese Option können Sie definieren, welche Systemeinrichtungen verwendet werden und in welcher Reihenfolge. Die Auflösungsoptionen sind lmhosts, host, wins und bcast.

Die Standardreihenfolge für die Auflösung ist `lmhosts`, `host`, `wins` und `bcast`.

Jede Option wird in der angegebenen Reihenfolge probiert, bis man eine IP-Adresse erhält oder bis keine Auflösungsmethode mehr zur Verfügung steht, was dazu führt, dass keine Verbindung zum Server aufgebaut werden kann.

Die einzelnen Optionen haben folgende Bedeutung:

`lmhosts`	Eine IP-Adresse wird in der Samba-Datei `lmhosts` gesucht, die normalerweise im gleichen Verzeichnis gespeichert ist wie `smb.conf`.
`host`	Eine IP-Adresse wird in der Datei `/etc/hosts`, NIS oder DNS gesucht, abhängig von den Systemeinstellungen für Ihr Betriebssystem.
`wins`	Eine IP-Adresse wird durch Übertragen einer Anfrage an den *Windows-Internet-Name-Service-(WINS-)*Server gesucht, der in der `smb.conf` aufgelistet ist. Ist kein WINS-Server definiert, wird diese Methode ignoriert.
`bcast`	Eine IP-Adresse wird durch Übertragen einer NetBIOS-Namensanfrage per Broadcast an alle bekannten Interfaces gesucht, die über den Parameter `interfaces` in `smb.conf` definiert sind. Gibt es keinen `interfaces`-Parameter in `smb.conf`, geht die Broadcast-Übertragung an alle bekannten Interfaces.

[-M *NetBIOS-Name*]

Über diese Option können Sie Nachrichten an andere Rechner senden. `smbclient` benutzt das WinPopup-Protokoll und versucht, den NetBIOS-Namen über die definierte Auflösungsreihenfolge oder die Standardauflösungsreihenfolge in eine IP-Adresse aufzulösen.

Sie können Nachrichten nur an NetBIOS-Namen senden.

Die zu übertragende Nachricht wird in STDIN eingegeben und mit einem `[Strg]`+`[D]` (`^D`) abgeschlossen.

Ein Beispiel:

```
echo "Backup starting" | smbclient -M Controller
```

[-i *Scope*]

Über diese Option können Sie den NetBIOS-Bereich einrichten, den `smbclient` verwendet, wenn es NetBIOS-Namen generiert. Diese Option wird selten benutzt. Weitere Informationen über NetBIOS-Namen finden Sie in den RFCs 1001 und 1002.

Diese Option war in den Samba-Versionen vor 2.0.0 nicht dokumentiert.

[-N]

Über diese Option können Sie die Aufforderung zur Passworteingabe unterdrücken, wenn `smbclient` sich mit einem Server verbindet. Sie ist am nützlichsten, wenn Sie auf eine Freigabe zugreifen, die kein Passwort hat.

[-n *NetBIOS-Name*]

Über diese Option können Sie einen NetBIOS-Namen spezifizieren, den der Client als den NetBIOS-Namen des lokalen Systems benutzt. Standardmäßig benutzt `smbclient` die groß geschriebene Version des Hostnamens des lokalen Rechners.

[-d *Debug-Level*]

Über diese Option können Sie den Level der Debug-Meldungen spezifizieren, der ausgegeben wird. Der Standardwert für diesen Parameter ist

0. Je höher der Level, desto mehr Informationen gibt `smbclient` über seine Aktivitäten aus. Auf höheren Levels sind die Ausgaben sehr umfangreich, darunter Details über Protokollmeldungen, die meistens kryptisch und nur für Entwickler interessant sind.

Auf Level 0 werden nur kritische Fehler und ernsthafte Warnungen protokolliert.

[-P]

In Versionen vor 2.0.0 konnten Sie über diese Option spezifizieren, dass Sie sich mit einer Druckerfreigabe verbinden wollten.

Diese Option wird in Samba 2.0.0 und höher nicht mehr benötigt (und stillschweigend ignoriert), da der Server jetzt den Gerätetyp bestimmen kann.

[-p *Port*]

Über diese Option können Sie den TCP-Port ändern, an dem `smbclient` verbunden ist, wenn es sich mit einem Server verbindet. Die Standard-Portnummer ist der bekannte TCP-Port für CIFS/SMB, auch *NetBIOS Session Service* genannt, oder Port 139. Sie sollten diese Option niemals verwenden müssen.

Diese Option war in Samba 2.0.0 fehlerhaft, wurde aber in 2.0.1 und höher korrigiert.

[-h]

Über diese Option können Sie die Auslastungsmeldung für `smbclient` ausgeben. Die Auslastungsmeldung wird auch ausgegeben, wenn in einer Option ein Fehler gefunden wird.

Diese Option war in den Manpages vor Samba 2.0.0 nicht dokumentiert, ist aber relativ offensichtlich.

[-I *Ziel-IP*]

Über diese Option können Sie die IP-Adresse des Servers spezifizieren, zu dem die Verbindung aufgenommen werden soll.

Normalerweise benutzt `smbclient` die NetBIOS-Prozedur zur Namensauflösung, die vorher in diesem Kapitel dargestellt wurde, um einen NetBIOS-Namen in eine IP-Adresse aufzulösen. Wenn Sie diese Option verwenden, können Sie `smbclient` dazu zwingen, sich mit dem angegebenen Server zu verbinden. Sie sollten jedoch beachten, dass die meisten Microsoft-Clients verlangen, dass der NetBIOS-Name in der Sitzungsanfrage dem NetBIOS-Namen des Servers entspricht. Dies bedeutet in der Regel, dass Sie sowohl die Option `-I` als auch die Option `-N` spezifizieren müssen.

[-E]

Die Option definiert, dass `smbclient` Meldungen und Debugging-Ausgaben an STDERR statt an STDOUT schreiben sollte.

Standardmäßig sendet `smbclient` Meldungen und Debugging-Informationen an STDOUT.

[-U *Benutzername*]

Über diese Option können Sie den Benutzernamen festlegen, der verwendet wird, um sich auf dem Server einzuloggen, nachdem eine Verbindung aufgebaut wurde.

Ist diese Option nicht spezifiziert, benutzt `smbclient` standardmäßig eine groß geschriebene Version der Umgebungsvariablen `USER` oder `LOGNAME`, und zwar in dieser Reihenfolge. Existiert keine der Umgebungsvariablen, wird der Benutzername `GUEST` verwendet.

Enthält die Umgebungsvariable `USER` ein Prozentzeichen (%), wird alles, was diesem Zeichen folgt, als Passwort behandelt. Damit können Sie *`Benutzername%Passwort`* in der Umgebungsvariable `USER` spezifizieren und das Übertragen von Passwörtern in der Befehlszeile (in der der Befehl `ps` sie sehen und anderen zeigen kann) verhindern.

Wenn der Server, mit dem Sie sich verbinden, eine ältere Version des SMB-Protokolls benutzt und keine Benutzernamen unterstützt, wird diese Option ignoriert.

Einige Server verlangen, dass der Benutzername groß geschrieben wird, und einige, dass es ein gültiger NetBIOS-Name ist (weniger als 16 Zeichen, groß geschrieben usw.).

[-L *NetBIOS-Name*]

Über diese Option können Sie eine Liste aller vom Server zur Verfügung gestellten Freigaben erhalten. Möglicherweise müssen Sie auch die Option `-I` spezifizieren, wenn Sie versuchen, einen Host in einem anderen Netzwerk zu erreichen, oder Ihre NetBIOS-Namen nicht mit Ihren Host-Namen übereinstimmen.

[-t *Terminal-Code*]

Mit dieser Option können Sie definieren, wie `smbclient` Dateinamen interpretiert, die vom Server kommen. Dies ist notwendig, da Unix üblicherweise andere Multibyte-Zeichensammlungen benutzt als Windows. Wenn Sie diese Option einrichten, konvertiert `smbclient` von SMB-Dateinamen zu Windows-Dateinamen und umgekehrt.

Werte für den Terminal-Code sind u.a. `sjis`, `euc`, `jis7`, `jis8`, `junet`, `hex` und `cap`. All dies sind Multibyte-Zeichenkodierungssysteme mit folgender Bedeutung:

`sjis`	Shift-JIS- (Japanese-Industrial-Standard-)Kodierungssystem
`euc`	Extended-Unix-Coding-System
`jis7`, `jis8`	7- und 8-Bit-JIS-Kodierungssystem
`junet`	Japanese-Unix-Network-Kodierungssystem
`hex`, `cap`	Andere Multibyte-Kodierungssysteme

Diese Option war in Samba-Versionen vor 2.0.0 nicht dokumentiert.

[-m *max protocol*]

Diese Option wird in Samba 2.0.0 und höher ignoriert. `smbclient` versucht jetzt immer, sich auf der höchsten vom Server unterstützten Protokollebene zu verbinden.

In Versionen vor Samba 2.0.0 (z.B. 1.9.18p10) war diese Option nicht dokumentiert. Sie konnten darüber die maximale Protokollebene einrichten, über `smbclient` verhandeln und so einige Funktionen vermeiden (wie z.B. das Bestehen darauf, sich in den Server einzuloggen).

[-W *Arbeitsgruppe*]

Über diese Option können Sie die Arbeitsgruppe spezifizieren, die `smbclient` bei der Suche nach NetBIOS-Namen und damit beim Aufbau von Verbindungen zu Servern verwendet. Sie brauchen diese Option möglicherweise für die Verbindung zu einigen Servern.

[-T *tar-Optionen*]

Mit dieser Option können Sie tar-kompatible Backups von Dateien auf CIFS/SMB-Servern erstellen. Die Option nimmt eine Reihe von sekundären Flags an (die später in diesem Kapitel dargestellt werden), die ihr Verhalten kontrollieren.

[-D *Verzeichnis*]

Über diese Option können Sie in ein Startverzeichnis wechseln, bevor Sie mit der Ausführung der gewünschten Operation beginnen. Wird meistens mit der Option `-T` (tar) benutzt.

[-c *Befehlsstring*]

Über diese Option können Sie eine Reihe von Befehlen spezifizieren, die von `smbclient` ausgeführt werden, statt von STDIN angefordert zu werden. Wird diese Option benutzt, ergibt sich daraus `-N`.

Diese Option wird hauptsächlich in Skripte und für die Einrichtung einiger Parameter mit der Funktion `-T` (tar) benutzt.

Die tar-Form des `smbclient`-Befehls (`smbclient` *`Freigabename`* `-T...`) hat die Optionen, die Sie in Tabelle 13.1 finden.

Tabelle 13.1: -T-Optionen

Option	**Beschreibung**
`c` *`tar-Datei`*	Definiert, dass eine tar-Datei erstellt wird, indem Dateien aus der Freigabe kopiert werden, die in der `smbclient`-Befehlszeile spezifiziert ist, modifiziert durch die `-C`-Option, wenn es eine gibt. Sie muss vom Namen der zu erstellenden tar-Datei gefolgt werden, der entweder eine Datei, ein Bandgerät (z.B. `/dev/st0`) oder `-` für die Standardausgabe sein kann (was Ihnen ermöglicht, die tar-Datei in einen anderen Befehl einzubetten).
`x` *`tar-Datei`*	Definiert, dass eine tar-Datei wiederhergestellt wird, indem Dateien an den Server und die Freigabe kopiert werden, die in der `smbclient`-Befehlszeile spezifiziert sind, modifiziert durch die `-C`-Option, wenn es eine gibt. Sie muss vom Namen der wiederherzustellenden tar-Datei gefolgt werden, der entweder eine Datei, ein Bandgerät (z.B. `/dev/st0`) oder `-` für die Standardausgabe sein kann. Von `c` und `x` kann nur eins gleichzeitig spezifiziert werden.

I *include-Ausdruck*	Fügt Dateien und Verzeichnisse ein, wie im Ausdruck definiert. Dateinamen-Globbing (manchmal auch als Wildcards bezeichnet) wird implementiert, wenn auch die Option r hinzugefügt wird. Werden zusätzliche Namen am Ende von tar-Optionen eingefügt, nimmt smbclient an, dass es sich um Namen für Dateien und Verzeichnisse handelt, die in das tar-Archiv eingefügt werden sollen.
X *exclude-Ausdruck*	Schließt Dateien und Verzeichnisse aus, wie im Ausdruck definiert. Dateinamen-Globbing (manchmal auch als Wildcards bezeichnet) wird implementiert, wenn auch die Option r hinzugefügt wird. (Siehe Hinweis nach dieser Tabelle.)
b *Blockgröße*	Ermöglicht die Spezifizierung der Blockgröße für die tar-Datei. Der eingegebene Wert muss größer als null sein, dann wird die tar-Datei in Blöcken zu Blockgröße x 512 Byte gelesen oder geschrieben.
g	Spezifiziert inkrementellen Modus, in dem nur Dateien gesichert werden, deren Archivbit gesetzt ist. Kann nur in Verbindung mit der Unteroption c benutzt werden.
q	Spezifiziert Ruhemodus, in dem während der Verarbeitung von Dateien keine Diagnosemeldungen ausgegeben werden.
r	Spezifiziert, dass das in smbclient integrierte Dateinamen-Globbing verwendet werden sollte, um den vorher beschriebenen include- oder exclude-Ausdruck zu interpretieren.
N Datei	Über diese Option können Sie festlegen, dass nur Dateien, die neuer als *Datei* sind, gesichert werden sollen. Kann nur mit der Unteroption c verwendet werden.
a	Führt dazu, dass smbclient das Archivbit zurücksetzt, wenn die Datei gesichert wurde. Kann nur mit den Unteroptionen g und c benutzt werden.

Dateinamen-Globbing funktioniert auf eine von zwei Arten. Wurde Samba mit HAVE_REGEX_H aufgebaut, kann der Ausdruck ein regulärer Ausdruck sein, und der reguläre Ausdrucksvergleich wird benutzt, um die in das tar-Archiv einzufügenden Dateien zu suchen. Beachten Sie, dass der reguläre Ausdrucksvergleich sehr rechenintensiv sein kann. Sie werden wirklich bemerken, wie viel langsamer smbclient ist, wenn Sie REGEX-Unterstützung integriert haben und die Option -r benutzen. Wurde Samba ohne HAVE_REGEX_H aufgebaut, wird eingeschränktes Dateinamen-Globbing unterstützt, wobei * und ? als Wildcards im Ausdruck behandelt werden. Dies ist für die meisten Leute ausreichend und außerdem viel schneller.

Diese Optionen werden direkt nach der -T-Option spezifiziert. Ein Beispiel:

```
smbclient //Server/Freigabename -Tc server.tar
```

Hier wird eine Backup-tar-Datei namens server.tar erstellt, die durch Kopieren aller Dateien von *//Server/Freigabename* aufgebaut wird.

Die tar-Komponente von smbclient handhabt alle langen Windows-Dateinamen, der volle Pfadname kann aber nicht länger als 1.024 Zeichen sein. Da Windows-95-Pfadnamen auf höchstens 256 Zeichen beschränkt sind und diese Einschränkung auch für Windows NT gilt, kann der tar-Befehl von smbclient Windows-Dateien jeder Art vollständig handhaben.

Wie immer sollten Sie einen Blick in die Manpages zu smbclient werfen, um das letzte Wort zu seinen Optionen und Funktionen zu finden.

smbclient-Ausgabe

`smbclient` handhabt informative Ausgaben (für den Benutzer) wie folgt:

Alle informativen Ausgaben werden in die gleiche Datei geschrieben, an die auch Debug-Ausgaben übertragen werden. Normalerweise ist das STDOUT, aber wenn `smbclient` seine Ausgaben an STDOUT sendet (weil vielleicht der Benutzer `-Tc -` spezifiziert hat, d.h. tar an STDOUT), werden informative und Debug-Ausgaben an STDERR übertragen.

Sie können die Ausgabe auch an STDERR übertragen, indem Sie für `smbclient` die Option `-E` spezifizieren.

smbclient-Ausdrücke (Wildcards oder Masken)

Für viele Befehle akzeptiert `smbclient` einen *Ausdruck* (oder eine *Maske (mask)* in der `smbclient`-Manpage). Dieser Ausdruck wird benutzt, um entsprechende Dateinamen vom Server für die Weiterbearbeitung durch den Befehl zu holen.

Diese Ausdrücke können Standard-Windows-Wildcards enthalten, wie z.B. `*` und `?`. Die Befehle, die diese Ausdrücke akzeptieren, verarbeiten nur die Dateien weiter, die den Ausdrücken entsprechen.

`dir *a*` z.B. gibt alle Dateien zurück, die irgendwo in ihrem Namen ein *a* haben.

smbclient-Befehle

Nachdem `smbclient` sich mit dem gewünschten Server und der Freigabe verbunden und eingeloggt hat, präsentiert es dem Benutzer die folgende Eingabeaufforderung:
`smb:\>`

\ bezeichnet hier das aktuelle Verzeichnis. Es ändert sich, wenn Sie sich in dem entfernten System bewegen.

An diesem Punkt können Sie jeden beliebigen `smbclient`-Befehl ausgeben. In den folgenden Abschnitten werden die einzelnen Befehle dargestellt.

Wenn Sie Dateinamen angeben müssen, die Leerstellen enthalten, schließen Sie diese Dateinamen in doppelte Anführungsstriche ein (z.B. "eine Datei mit einem langen Namen").

Nachfolgend werden optionale Parameter in eckigen Klammern dargestellt, z.B. `[optional]`. Erforderliche Parameter werden in spitzen Klammern gezeigt, z.B. `<verlangt>`.

? [*Befehl*] | help [*Befehl*]

Diese Befehle bieten Hilfe zu allen Befehle. Sie nehmen einen optionalen Parameter an, in Form des Befehlsnamens, zu dem Sie Hilfe brauchen. Wird kein Parameter angegeben, erhalten Sie eine Liste verfügbarer Befehle.

Ein Beispiel:
```
smb: \> ? mkdir
HELP mkdir:
<directory> make a directory
smb: \>
```

Diese Syntax bittet um Hilfe zum Befehl `mkdir`. Sie sehen außerdem die Resultate dieses Befehls.

! [*Shell-Befehl*]

Über diesen Befehl können Sie einen Shell-Befehl ausführen oder von innerhalb `smbclient` eine Shell aufrufen. Er kann benutzt werden, um sich die Inhalte einer vom entfernten System abgerufenen Datei anzusehen, ohne `smbclient` beenden zu müssen.

Im folgenden Beispiel wird der Befehl benutzt, um eine Datei anzusehen, die vom entfernten Server abgerufen wurde, ohne `smbclient` zu verlassen:
```
smb: \> ! cat text.txt
A file with some characters in it.
smb: \>
```

Hier haben Sie `cat` verwendet, um die Datei `test.txt` zu listen.

archive [*Level*]

Über diesen Befehl können Sie das Verhalten von `mget`-Befehlen in Hinsicht auf die Handhabung des Archivbits auf DOS/Windows-Dateien kontrollieren. Ist kein Level definiert, gibt der Befehl einfach den aktuellen Wert für die Archiveinstellung aus.

Wollen Sie einen Level spezifizieren, muss es ein Wert zwischen 0 und 3 sein, der folgende Bedeutung hat:

- 0 - Ruft alle Dateien ab, ohne auf das Archivbit zu achten.
- 1 - Ruft nur die Dateien ab, bei denen das Archivbit gesetzt wurde. Das Archivbit selbst wird nicht geändert.
- 2 - Ruft nur die Dateien ab, bei denen das Archivbit gesetzt ist, und setzt bei diesen Dateien das Archivbit zurück.
- 3 - Ruft alle Dateien ab, unabhängig von der Einstellungen für das Archivbit, und setzt das Archivbit auf allen Dateien zurück.

Dieser Befehl ist nicht in den Manpages, aber in der Hilfefunktion von `smbclient` dokumentiert.

Nachfolgend finden Sie zwei Beispiele für die Verwendung des Befehls `archive`:

```
smb: \> archive
Archive level is 0
smb: \> archive 3
smb: \> archive
Archive level is 3
smb: \>
```

Hier verwenden Sie `archive`, um den aktuellen `archive`-Level zu überprüfen, setzen ihn auf 3 und stellen dann sicher, dass er auf 3 gesetzt wurde.

blocksize <*blocksize*>

Über diesen Befehl können Sie die Blockgröße spezifizieren, die mit dem `tar`-Befehl von `smbclient` benutzt werden soll. Der Wert des Parameters muss eine ganze Zahl größer als null sein.

Wird der Befehl benutzt, veranlasst er den `tar`-Befehl Blöcke von *`Blockgröße`* `x TBLOCK` (aktuell ist `TBLOCK` auf 512 gesetzt) Byte zu lesen oder zu schreiben.

Ein Beispiel:

```
smb: \> blocksize 40
blocksize is now 40
smb: \>
```

Damit wird die Blockgröße auf 40 gesetzt, d.h. 40x512-Byte-Blöcke oder 20.480-Byte Blöcke.

cancel <*Job-ID*>

Dieser Befehl löscht den Job mit der Nummer *`Job-ID`* in der Druckerwarteschlange.

Dieser Befehl ist nicht in den Manpages, aber in der Hilfefunktion zu `smbclient` dokumentiert.

Ein Beispiel:

```
smb: \> cancel 7
Job 7 cancelled
smb: \>
```

Dies zeigt Ihnen, wie Sie einen Druckauftrag löschen. Um dies zu tun, müssen Sie mit einer Druckerfreigabe verbunden sein, die einen solchen Job in der Warteschlange hat.

cd [*Verzeichnis*]

Über diesen Befehl können Sie in das aktuelle Arbeitsverzeichnis auf dem Server wechseln und es ausgeben.

Ist ein Verzeichnis spezifiziert, wird das aktuelle Arbeitsverzeichnis auf dem Server in das angegebene geändert.

Ist kein Parameter gegeben, wird das aktuelle Arbeitsverzeichnis auf dem Gerät ausgegeben, an das die Debug-Ausgabe geht.

Ein Beispiel:

```
smb: \> cd "Neuer Ordner"
smb: \> \Neuer Ordner\
```

Damit wechseln Sie in das Verzeichnis `Neuer Ordner`.

del <*Ausdruck*>

Über diesen Befehl können Sie Dateien aus dem Arbeitsverzeichnis auf dem Server löschen. Alle Dateien, die *Ausdruck* entsprechen, werden gelöscht.

Ein Beispiel:

```
smb: \> ls
   file-1.txt             69 Mon Jan 11 15:35:14 1999
   file-2.txt             59 Mon Jan 11 15:35:14 1999
   Neuer Ordner    D       0 Sat Feb  6 10:51:47 1999
   609913.doc      A  431104 Mon Jan 18 22:25:20 1999

    55729 blocks of size 16384. 6871 blocks available
smb: \> del file-1.txt
smb: \> ls
   file-2.txt             59 Mon Jan 11 15:35:14 1999
   Neuer Ordner    D       0 Sat Feb  6 10:51:47 1999
   609913.doc      A  431104 Mon Jan 18 22:25:20 1999

    55729 blocks of size 16384. 6871 blocks available
smb: \>
```

Dies zeigt uns, wie die Datei `file-1.txt` gelöscht wird. Zunächst listen Sie die verfügbaren Dateien auf, löschen dann `file-1.txt` und überprüfen abschließend, ob die Datei gelöscht wurde.

dir [*Ausdruck*] | ls [*Ausdruck*]

Über diese Befehle erhalten Sie eine Verzeichnisauflistung der Dateien in einem beliebigen Verzeichnis auf dem Server.

Wird ein *Ausdruck* angegeben, werden nur die Dateien und Verzeichnisse aufgelistet, die dem Ausdruck entsprechen.

Ist kein Ausdruck spezifiziert, werden alle Dateien und Verzeichnisse im aktuellen Verzeichnis aufgelistet.

Siehe auch `ls` später in diesem Kapitel.

Ein Beispiel:

```
smb: \> dir
   file-2.txt             59 Mon Jan 11 15:35:14 1999
   Neuer Ordner    D       0 Sat Feb  6 10:51:47 1999
   609913.doc      A  431104 Mon Jan 18 22:25:20 1999

    55729 blocks of size 16384. 6871 blocks available
smb: \>
```

du

Über diesen Befehl können Sie abrufen, wie viel Speicherplatz von den Dateien im aktuellen Verzeichnis belegt wird. Ist rekursiver Modus aktiviert, durchläuft er rekursiv alle Verzeichnisse, um den belegten Speicherplatz zu kalkulieren, sonst wird nur der Speicherplatz für die Dateien im aktuellen Verzeichnis ausgegeben.

Dieser Befehl ist nicht in den Manpages, aber in der Hilfefunktion von `smbclient` dokumentiert.

Ein Beispiel:

```
smb: \> recurse
directory recursion is now off
smb: \> du
    55729 blocks of size 16384. 5912 blocks available
Total number of bytes: 431163
smb: \> recurse
directory recursing is now on
smb: \> du
    55729 blocks of size 16384. 5912 blocks available
Total number of bytes: 15979461
smb: \>
```

Hier haben Sie sichergestellt, dass der rekursive Modus deaktiviert wird, bevor Sie den Befehl `du` benutzen. In Zeile 5 sehen Sie, dass nur 431.163 Byte ausgegeben werden, aber nachdem Sie in Zeile 6 den rekursiven Modus aktiviert haben, sehen Sie, dass in Zeile 10 15.979.461

Byte ausgegeben werden.

exit | quit | q

Diese Befehle beenden `smbclient`. Vorher trennt `smbclient` noch die Verbindung zum Server. Es ist das Gleiche wie `[Strg]+[D]` (^D).

get <*Remote-Datei*> [*lokale Datei*]

Über diesen Befehl können Sie eine entfernte Datei vom Server auf den lokalen Rechner kopieren. Der Name der entfernten Datei muss angegeben werden, und wenn der Benutzer einen Namen für eine lokale Datei spezifiziert, wird die entfernte Datei kopiert und erhält den Namen der lokalen Datei.

Gibt es keinen lokalen Dateinamen, wird die entfernte Datei kopiert und erhält den gleichen Namen.

Ist die Option `lowercase` aktiviert, werden alle lokalen Dateinamen als klein geschriebene Versionen der gleichen entfernten Dateien erstellt.

Alle Dateien werden im Binärmodus kopiert, es sei denn, Sie haben die Option `translation` aktiviert, dann werden CRLF-Paare auf dem lokalen Rechner in LF-Entsprechungen umgewandelt.

Ein Beispiel:

```
smb: \> get file-2.txt
getting file file-2.txt of size 59 as file-2-txt (0.414512 kb/s) (average 0.414512 kb/s)
smb: \>
```

Hier haben Sie den Befehl `get` benutzt, um die Datei `file-2.txt` zu kopieren. Ist die Datei kopiert, gibt `smbclient` einige Statistiken über die Übertragung aus.

lcd [*Verzeichnis*]

Über diesen Befehl können Sie Ihr lokales Verzeichnis wechseln. Wird ein gültiges Verzeichnis angegeben, wechselt `smbclient` in dieses Verzeichnis auf dem lokalen Rechner. Wird kein Verzeichnis spezifiziert, gibt `smbclient` das aktuelle Arbeitsverzeichnis in der Ausgabedatei aus.

Ein Beispiel:

```
smb: \> lcd ..
the local directory is now /
smb: \> lcd /usr/local/sbin
the local directory is now /usr/local/sbin
smb: \>
```

Hier benutzen Sie den Befehl `lcd` zweimal, und `smbclient` gibt jedes Mal aus, wo Sie sich befinden.

lowercase

Über diesen Befehl können Sie die Schreibweise lokaler Dateien kontrollieren, wenn Dateien über die Befehle `get` und `mget` von CIFS/SMB-Servern kopiert werden. Er schaltet den aktuellen Wert von `lowercase` um, der standardmäßig auf `OFF` eingestellt ist.

Wird `lowercase` auf `ON` gesetzt, werden lokale Dateinamen (wenn sie nicht vom Benutzer spezifiziert sind) als klein geschriebene Versionen der gleichen entfernten Dateien erstellt. Dies ist sehr nützlich, wenn Sie MS-DOS-Dateien von einem Server kopieren, da Unix-Dateinamen normalerweise klein geschrieben werden.

mask <*Ausdruck*>

Über diesen Befehl können Sie einen Ausdruck festlegen, der benutzt wird, um Dateien in rekursiven (wird später in diesem Kapitel dargestellt) `mput`- und `mget`-Operationen zu filtern.

Bei rekursiven `mput`- und `mget`-Operationen müssen zwei Selektionen durchgeführt werden. Zunächst müssen alle relevanten Verzeichnisse ausgewählt werden. Sodann müssen alle relevanten Dateien selektiert werden. Diese zwei Selektionsoperationen sind normalerweise verschieden. Über den Befehl `mask` können Sie einen Ausdruck spezifizieren, der für die Auswahl der Dateien benutzt wird, die in rekursiven Operationen weiterverarbeitet werden.

Ein separater Ausdruck wird mit den Befehlen `mget` und `mput` spezifiziert, die die Verzeichnisse auswählen, die im rekursiven Modus weiterverarbeitet werden.

Die Maske ist standardmäßig leer, was bedeutet, dass alle Dateien selektiert werden. Wenn Sie die Maske ändern, bleibt sie so, bis Sie sie

erneut ändern.

Ein Beispiel für diesen Befehl sehen Sie bei der Darstellung der Befehle `mput` und `mget` später in diesem Kapitel.

md <*Verzeichnis*> | mkdir <*Verzeichnis*>

Über diese Befehle können Sie neue Verzeichnisse im aktuellen Verzeichnis auf dem Server erstellen, wenn Sie die entsprechenden Privilegien haben. Das neue Verzeichnis erhält den angegebenen Namen.

Ein Beispiel:

```
smb: \> ls
   file-2.txt                59 Mon Jan 11 15:35:14 1999
   Neuer Ordner      D        0 Sat Feb  6 10:51:47 1999
   609913.doc        A   431104 Mon Jan 18 22:25:20 1999

    55729 blocks of size 16384. 5909 blocks available
smb: \> md new-dir
smb: \> ls
   new-dir           D        0 Thu Feb 11 01:44:32 1999
   file-2.txt               59 Mon Jan 11 15:35:14 1999
   Neuer Ordner      D        0 Sat Feb  6 10:51:47 1999
   609913.doc        A   431104 Mon Jan 18 22:25:20 1999

    55729 blocks of size 16384. 5909 blocks available
smb: \>
```

Hier überprüfen Sie das aktuelle Verzeichnis auf dem entfernten Server, erstellen das Verzeichnis `new-dir` und listen dann noch einmal das Verzeichnis auf, um sicherzustellen, dass das neue Verzeichnis erstellt wurde.

mget <*Ausdruck*>

Über diesen Befehl können Sie alle Dateien, die *`Ausdruck`* entsprechen, vom Server auf den lokalen Rechner kopieren.

Ist aber rekursiver Modus aktiviert, selektiert *`Ausdruck`* die Verzeichnisse, die weiterverarbeitet werden, während der Befehl `mask` die zu kopierenden Dateien spezifiziert. Wenn Sie bei aktiviertem rekursiven Modus über `mget` Dateien kopieren und der `mget`-Befehl die Verzeichnisse selektiert, erstellt er die Verzeichnisstruktur auf dem lokalen Rechner. Das heißt, er behält die Verzeichnisstruktur der entsprechenden Dateien und Verzeichnisse für die kopierten Dateien bei.

Ein Beispiel:

```
smb: \> ls
   new-dir           D     0 Thu Feb 11 11:57:48 1999
   file-4.txt             59 Mon Jan 11 15:35:14 1999
   2nd-dir           D     0 Thu Feb 11 11:57:57 1999

    55729 blocks of size 16384. 6887 blocks available
smb: \> prompt
prompting is now off
smb: \> mget *.txt
getting file file-4.txt of size 59 as file-4.txt (1.2259 kb/s) (average 1.2259 kb/s)
smb: \> recurse
directory recursion is now on
smb: \> mask *.txt
smb: \> mget *dir
getting file file-1.txt of size 59 as file-1.txt (0.789275 kb/s) (average 0.960286 kb/s)
getting file file-2.txt of size 59 as file-2.txt (1.25254 kb/s) (average 1.04127 kb/s)
getting file file-3.txt of size 59 as file-3.txt (1.15234 kb/s) (average 1.06698 kb/s)
smb: \>
```

Sie haben hier beim ersten `mget`-Befehl nur eine Datei kopieren können. Aber nachdem Sie für den zweiten `mget`-Befehl rekursiven Modus aktiviert und eine Maske eingerichtet hatten, konnten Sie mehrere Dateien kopieren. Diese Dateien befanden sich in den Unterverzeichnissen `new-dir` und `2nd-dir`.

more <*Datei*>

Über diesen Befehl können Sie eine entfernte Datei abrufen und auf Ihrem Standard-Pager ansehen.

Dieser Befehl ist nicht in den Manpages, aber in der Hilfefunktion von `smbclient` dokumentiert.

mput <*Ausdruck*>

Über diesen Befehl können Sie alle Dateien, die *`Ausdruck`* entsprechen, vom lokalen Rechner auf den Server kopieren.

Ist jedoch rekursiver Modus aktiviert, selektiert *`Ausdruck`* die Verzeichnisse, die weiterverarbeitet werden, während der Befehl `mask` die zu kopierenden Dateien spezifiziert.

Ein Beispiel:

```
smb: \> prompt
prompting is now off
smb: \> mput *.txt
putting file browse.txt as \browse.txt (20.2602 kb/s) (average 20.2602 kb/s)
putting file evi.txt as \evi.txt (37.6367 kb/s) (average 30.4219 kb/s)
putting file file-2.txt as \file-2.txt (2.05775 kb/s) (average 26.431 kb/s)
putting file file-4.txt as \file-4.txt (5.76166 kb/s) (average 25.442 kb/s)
putting file nmblookup.txt as \nmblookup.txt (5.55096 kb/s) (average 23.7844 kb/s)
smb: \>
```

newer <*Datei*>

Über diesen Befehl legen Sie fest, dass `mget`-Befehle nur Dateien kopieren, die neuer sind als die angegebene lokale *`Datei`*.

Dieser Befehl ist nicht in den Manpages, aber in der Hilfefunktion von `smbclient` dokumentiert.

Ein Beispiel:

```
smb: \> newer log.log
Getting files newer than Sun Jan 17 12:33:26 1999
smb: \>
```

print <*Datei*>

Über diesen Befehl können Sie eine *`Datei`* auf dem lokalen Rechner über eine Druckerfreigabe auf dem Server drucken.

Da eine Datei auch gedruckt werden kann, indem Sie sie an eine Druckerfreigabe (über `put`) kopieren, ist dieser Befehl überflüssig und sollte nicht mehr benutzt werden. Er wird aber noch für Skripte beibehalten, die vielleicht bereits angewendet werden.

Ein Beispiel:

```
smb: \> print log.log
putting file log.log as log.log (19.3522 kb/s) (average 19.3522 kb/s)
smb: \>
```

Hier können Sie sehen, dass die Datei tatsächlich über den Befehl `put` übertragen wurde.

printmode <*Modus*>

Dieser Befehl ist ab Samba 2.0.0 veraltet und hat keine Auswirkung auf andere Befehle. Er wird aus Gründen der Kompatibilität mit älteren Skripten beibehalten, die möglicherweise im Einsatz sind.

In Versionen vor Samba 2.0.0 (z.B. 1.9.18p10) wurde das Drucken implementiert, indem auf dem Server eine Spool-Datei geöffnet und die zu druckende Datei in die Spool-Datei kopiert wurde. Nach Öffnen der Spool-Datei wurde der Wert von `printmode` übertragen. Die meisten modernen Server aber ignorieren den Druckmodus sowieso.

prompt

Über diesen Befehl können Sie einstellen, ob Sie dazu aufgefordert werden, die Übertragung für jede Datei während der Ausführung der `mget`- und `mput`-Befehle zu bestätigen oder ob diese Befehle ihre Übertragungen stillschweigend durchführen.

Der Startwert von `prompt` ist `ON`, und dieser Befehl schaltet den aktuellen Wert um. Mit der Einstellung `ON` werden Sie aufgefordert, jede Übertragung zu bestätigen. Mit der Einstellung `OFF` finden alle Übertragungen ohne Eingabe von Ihnen statt.

put <*lokale Datei*> [Remote-Datei]

Über diesen Befehl können Sie Dateien vom lokalen Rechner an den Server kopieren. Die lokale Datei muss spezifiziert werden. Ist ein Name für die Remote-Datei angegeben, wird die Remote-Datei erstellt und die Inhalte der lokalen Datei werden in diese kopiert.

Ist kein Name für die Remote-Datei angegeben, wird eine entfernte Datei mit dem gleichen Namen wie lokale Datei erstellt und die Inhalte der lokalen Datei werden in diese kopiert.

Ein Beispiel:

```
smb: \> put log.log
putting file log.log as \log.log (329.792 kb/s) (average 329.793 kb/s)
smb: \>
```

pwd

Dieser Befehl gibt das aktuelle entfernte Verzeichnis aus.

Er ist nicht in den Manpages, aber in der Hilfefunktion von `smbclient` dokumentiert.

queue

Über diesen Befehl können Sie die Jobs in einer entfernten Warteschlange auflisten, müssen dafür aber mit einer Druckerfreigabe verbunden sein.

rd <*Verzeichnis*> | rmdir <*Verzeichnis*>

Über diese Befehle können Sie ein Verzeichnis vom Server löschen. Sie löschen das angegebene Verzeichnis aus dem aktuellen Verzeichnis, wenn Sie Zugriffsrechte darauf haben und das Verzeichnis leer ist.

Wenn Sie versuchen, ein Verzeichnis zu löschen, das nicht leer ist, erhalten Sie die Fehlermeldung `ERRDOS -ERRnoaccess (Access denied)`.

recurse

Über diesen Befehl können Sie `mput`- und `mget`-Befehle dazu veranlassen, bei der Übertragung von Dateien auch alle Unterverzeichnisse zu verarbeiten. Standardmäßig ist `recurse` auf `OFF` eingestellt. Er veranlasst auch den Befehl `du`, sich auf Unterverzeichnisse auszubreiten.

Wird `recurse` auf `ON` geschaltet, verarbeiten die Befehle `mput` und `mget` alle Unterverzeichnisse, die dem Ausdruck entsprechen, der mit den Befehlen `mget` oder `mput` spezifiziert ist. Die Dateien, die tatsächlich von diesen Befehlen verarbeitet werden, sind über den Befehl `mask` spezifiziert. Wird kein `mask`-Befehl ausgegeben, werden alle Dateien verarbeitet.

Ein Beispiel:

```
smb: \> mask *.txt
smb: \> mget *dir
```

Damit werden alle Unterverzeichnisse eingeschlossen, die `*.dir` und alle Dateien kopiert, die `*.txt` entsprechen.

Wird `recurse` auf `OFF` geschaltet, werden lediglich Dateien aus dem aktuellen Arbeitsverzeichnis weiterverarbeitet, und zwar nur die, die dem mit den Befehlen `mget` oder `mput` spezifizierten Ausdruck entsprechen.

Siehe auch `mget` und `mput` vorher in diesem Kapitel.

rm <*Ausdruck*>

Über diesen Befehl können Sie alle Dateien im aktuellen Arbeitsverzeichnis löschen, die *`Ausdruck`* entsprechen.

setmode <*Datei*> [[+|-]] [r|s|h|a]>

Über diesen Befehl können Sie Dateiattribute auf dem Server einstellen. Er ist dem DOS-Befehl `attrib` ähnlich. Sie können hierüber DOS-Attribute wie *read-only* (schreibgeschützt) (`r`), *system* (`s`), *hidden* (versteckt) (`h`) oder *archive* (`a`) setzen. Sie können mehrere dieser Attribute auf einmal spezifizieren. Außerdem können Sie sowohl Attribute, die hinzugefügt werden sollen, als auch solche, die entfernt werden sollen, gleichzeitig angeben (z.B. `r-s`).

Ein Beispiel:

```
smb: \> setmode log.log +hs

perm set 6 0
smb: \>
```

tar <c|x>[*IxbgNarq*-Parameter]

Über diesen Befehl können Sie Dateien vom Server in einer lokalen tar-Datei, auf einem Band oder in STDOUT sichern. Die Unteroptionen

und Parameter sind die gleichen, die unter `smbclient -tar`-Optionen beschrieben wurden. Sie müssen entweder `c` oder `x` einfügen, was bedeutet, dass ein neues Archiv erstellt bzw. ein existierendes Archiv extrahiert wird.

Dieser Befehl kann benutzt werden, um entfernte Windows-Systeme zu sichern. Beispiele für die Verwendung der Befehlszeilenversion dieses Befehls finden Sie in Anhang B, »Tipps und Tricks«.

tarmode <[no]<full|inc|reset|noreset|hidden|quiet|verbose>>+

Über diesen Befehl können Sie eine ganze Reihe von Aspekten des Verhaltens des `tar`-Befehls kontrollieren. Die Parameter für `tarmode` habe folgende Bedeutung:

`full`	Vollständiges Backup
`inc`	Backup der Dateien, für die das Archivbit gesetzt ist
`reset`	Zurücksetzen des Archivbits auf allen gesicherten Dateien
`system`	Backup der Dateien mit gesetztem Systembit
`hidden`	Backup der Dateien mit gesetztem Hidden-Bit
`verbose`	Ausgabe von Backup-Informationen während der Ausführung
`quiet`	keine Ausgabe von Informationen während der Ausführung

Wenn Sie ein `no` vor den Parameter platzieren, negiert es seine Bedeutung, und Sie können mehr als einen Parameter auf einmal spezifizieren, getrennt durch Leerzeichen.

Die Standardwerte für `tarmode` sind `full` und `verbose`.

Ein Beispiel:

```
smb: \> tarmode nosystem hidden
tarmode is now full, nosystem, hidden, noreset, verbose
smb: \>
```

Hier haben Sie spezifiziert, dass Sie keine Systemdateien wollen, versteckte Dateien aber schon. `smbclient` teilt uns mit, dass `tarmode` auch `noreset` und `verbose` enthält.

translate

Dieser Befehl ist in keiner Samba-Version dokumentiert. Er kann benutzt werden, um die Handhabung der verschiedenen Zeilenenden zwischen Unix und DOS/Windows zu kontrollieren.

Ist `translate` beim Kopieren von Dateien von einem CIFS/SMB-Server zum lokalen Rechner auf `ON` geschaltet, wird CRLF in LF übersetzt. Ganz ähnlich wird LF in CRLF übersetzt, wenn Sie Dateien von einem lokalen Rechner auf einen CIFS/SMB-Server kopieren.

Ist `translate` auf `OFF` gesetzt, werden keine übertragenen Dateien geändert.

Standardmäßig ist `translate` in `smbclient` auf `OFF` gesetzt.

smbclient-Beispiele

Nachdem Sie sich nun die Befehle und Befehlszeilenoptionen für `smbclient` angesehen haben, finden Sie nachfolgend einige Beispiele für die Benutzung von `smbclient`.

Utilities wie `smbtar` und `smbprint` sind weitere Quellen für Beispiele, da sie `smbclient` für die Ausführung ihrer Funktionen benutzen.

Freigaben auf einem Server auflisten

Wenn Sie überprüfen, warum Sie auf eine Freigabe auf einem Server nicht zugreifen können, besteht einer der ersten Debugging-Schritte darin, eine Auflistung der Freigaben auf diesem Server zu holen. Zwar können Sie unter Windows 9x oder Windows NT die Netzwerkumgebung starten, aber es ist einfacher, `smbclient` zu verwenden. Hier ein Beispiel:

```
[root@eagle]# smbclient -L eagle -N
Added interface ip=16.153.112.110 bcast=16.153.112.255 nmask=255.255.255.0
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0beta4]

   Sharename      Type    Comment
   ---------      ----    -------
   samba          Disk    Samba smbedit share
```

```
   first-share   Disk    My first share
   kits          Disk    Kits Area
   cdrom         Disk    CDROM, automounts where connected to
   first-printer Printer My first printer
   printer$      Disk
   IPC$          IPC     IPC Service (Samba 2.0.0beta4 Server)

   Server           Comment
   ------           -------
   EAGLE            Samba 2.0.0beta4 Server

    Workgroup       Master
    ---------       ------
    FOWLPLAY        EAGLE
    NCINET          RJSPC1
[root@eagle]# exit
```

Hier sehen Sie, dass `EAGLE` ein Samba-2.0.0beta4-Server unter Unix in der Arbeitsgruppe `FOWLPLAY` ist. Sie sehen außerdem alle Freigaben auf `EAGLE` und alle bekannten Arbeitsgruppen.

Eine Datei drucken

Um eine Datei zu drucken, kopieren Sie die Datei einfach an eine Druckerfreigabe. Dies können Sie wie folgt in einer Befehlszeile tun:

```
[root@eagle]# cat log.smb | smbclient //eagle/first-printer XXXXXXX -N -c "put - fred"
```

Hier ist ein erfolgreiches Ergebnis:

```
Added interface ip=16.153.112.110 bcast=16.153.112.255 nmask=255.255.255.0
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0beta4]
putting file - as \fred (13.8289 kb/s) (average 13.8289 kb/s)
[root@eagle]#
```

Hier holen Sie die Datei über `cat` aus STDOUT und kopieren die Datei in `smbprint` von `-` zu einem entfernten Dateinamen, den Sie angeben. Sie sollten vielleicht etwas wie `$$` (wird in den meisten Shells in die PID übersetzt) im Zieldateinamen angeben, um Namensüberschneidungen zu verhindern.

Das Passwort wurde als `XXXXXXX` eingegeben. Sie sollten das korrekte Passwort für den Remote-Account benutzen.

Wenn Sie noch kein erfahrener Unix-Benutzer sind, fragen Sie sich vielleicht, welche Art von Datei `?` ist. Dies ist eine Standard-Unix-Konvention, über die Programme ihre Eingaben aus STDIN erhalten und ihre Ausgaben an STDOUT übertragen können. Mit dieser Methode können Programme wie `smbclient` in einer Unix-Pipeline benutzt werden und sinnvolle Dinge tun.

Lassen Sie uns einen genaueren Blick auf den Befehl werfen, den wir vorher in diesem Abschnitt benutzt haben. Er besteht aus `cat log.smb`, gefolgt vom Pipe-Symbol (`|`), gefolgt vom `smbclient`-Befehl. Standardmäßig sendet `cat` seine Ausgabe über STDOUT, und das Pipe-Symbol verbindet zwei Programme, so dass STDOUT des ersten mit STDIN des zweiten Programms verbunden wird. Damit `smblient` sich die Datei holt, müssen Sie ihm mitteilen, so lange aus STDIN zu kopieren, bis es ein Dateiende (EOF) sieht. Das ist, was Sie vorher getan haben. Sie haben `smbclient` mitgeteilt, aus `-` zu kopieren, was das Programm in diesem Fall als Hinweis auf STDIN behandelt.

Ganz ähnlich kann `smbclient` Dateien kopieren und an STDOUT übertragen. Ein Beispiel:

```
smbclient //eagle/first-share XXXXXX -N -c 'get file-1.txt -' | lpr -Pmyprinter
```

Hier wird eine Datei namens `file-1.txt` kopiert und über STDOUT an den nächsten Befehl in der Pipeline übertragen, der in diesem Fall der Unix-Druckbefehl, `lpr`, ist.

Dateien in einem Verzeichnis auflisten

Obwohl Sie `smbclient` interaktiv verwenden können, um eine Verzeichnisauflistung zu erhalten, zeigt Ihnen das folgende Beispiel, wie Sie dies über die Befehlszeile tun:

```
[root@eagle]# smbclient //rjspc1/c -N -D csw -c dir
```

Dieser Befehl hat folgendes Ergebnis:

```
Added interface ip=16.153.112.110 bcast=16.153.112.255 nmask=255.255.255.0
Got a positive name query response from 16.153.112.120 (16.153.112.120)
```

```
.               D        0    Thu Dec 18 20:20:52 1997
..              D        0    Thu Dec 18 20:20:52 1997
METER.DLL       A     5216    Mon Nov  6 10:35:58 1995
CSWPROMO.EXE    A  1314816    Tue Jul 30 12:17:44 1996
CSHDOC.WIR      A    22528    Tue Jul 30 12:18:16 1996
HOSTDSK.WIR     A     5120    Wed Jan 15 09:28:54 1997

    45035 blocks of size 32768. 735 blocks available
[root@eagle]# exit
```

In diesem Beispiel benutzen Sie `-N`, damit `smbclient` kein Passwort verlangt, `-D csw` für die Angabe eines Startverzeichnisses und `-c dir`, um festzulegen, dass ein Verzeichnisbefehl ausgeführt werden soll.

Aufträge in einer Druckerwarteschlange auflisten

Sie können alle Aufträge in einer Druckerwarteschlange auf einem CIFS/SMB-Server auflisten, indem Sie Folgendes am Befehlsprompt eingeben:

```
[root@eagle]# smbclient //eagle/first-printer XXXXXXX -c queue
```

Sie erhalten folgendes Ergebnis:

```
Added interface ip=16.153.112.110 bcast=16.153.112.255 nmask=255.255.255.0
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0beta4]
1285    12228    rjspc1.a00652
1286    12228    rjspc1.a00972
[root@eagle]#
```

Hier haben Sie wieder `-c` benutzt, um den auszuführenden Befehl zu definieren. Sie können sehen, dass sich zwei Aufträge in der Warteschlange befinden.

Eine Datei vom entfernten Rechner kopieren

Zwar können Sie Dateien interaktiv kopieren, aber hier sehen Sie, wie eine Datei von einem CIFS/SMB-Server über die Befehlszeile kopiert werden kann:

```
[root@eagle]# smbclient //rjspc1/c -N -D csw -c "get meter.dll"
```

Sie erhalten folgendes Ergebnis:

```
Added interface ip=16.153.112.110 bcast=16.153.112.255 nmask=255.255.255.0
Got a positive name query response from 16.153.112.120 (16.153.112.120)
getting file meter.dll of size 5216 as meter.dll (69.7773 kb/s) (average 69.7774 kb/s)
[root@eagle]#
```

Hier haben Sie wieder die Option `-D` benutzt, um ein Startverzeichnis einzurichten, und die Option `-c`, um den auszuführenden Befehl zu definieren.

smbfs

Stellen Sie sich vor, einfach ein CIFS/SMB-Dateisystem auf einen Unix-Rechner zu mounten. Unter Linux können Sie dies über `smbfs` erreichen, einem Dateisystem, das das CIFS/SMB-Protokoll versteht.

Sie brauchen zwei Dinge, um `smbfs` benutzen zu können:

- Einen Linux-Kernel mit `smbfs`-Unterstützung
- Utilities, über die Sie `smbfs`-Dateisysteme mounten und unmounten können

Linux-Kernel enthalten schon seit einiger Zeit `smbfs`, und Sie können `smbfs` in den Kernel einkompilieren oder es als ladbares Modul benutzen. Da es den Rahmen dieses Buches sprengen würde, Ihnen zu erklären, wie Sie neue Kernel kompilieren oder ein ladbares Modul konfigurieren, sollten Sie auf ein Linux-Buch zurückgreifen, in dem diese Dinge dargestellt werden. Eine ganze Reihe aktueller Linux-Distributionen bieten direkt integrierte `smbfs`-Unterstützung.

Wenn Sie einen Kernel mit `smbfs`-Unterstützung haben, brauchen Sie die Utilities, über die Sie `smbfs`-Dateisysteme mounten und unmounten können, da die Standard-Linux-Befehle `mount` und `umount` die `smbfs`-Dateisysteme nicht unterstützen. Sie haben hier zwei Optionen.

Erstens enthält Samba ab Version 1.9.18p10 und 2.0.0 die Utilitites `smbmount`, `smbumount` und `smbmnt`. Mit diesen Utilities können Sie

Dateisysteme von CIFS/SMB-Servern mounten und unmounten.

Zweitens enthält das Paket `smbfs-2.0.1-4` ähnliche Utilities, darunter `smbmount` und `smbumount` (unterschiedliche Code-Basis zu den vorher erwähnten). Auch diese Utilities ermöglichen Ihnen das Mounten und Unmounten von Dateisystemen von CIFS/SMB-Servern.

Der wichtigste Faktor, den Sie bei der Wahl zwischen diesen beiden Paketen beachten sollten, ist, dass die Samba-2.0.0-Version der `smbmount`-Utilities nur auf Linux-Kernel höher als 2.1.70 und den neuen Linux-2.2-Kernel kompiliert werden kann.

Wenn Sie ein `smbfs`-Dateisystem unter Linux gemountet haben, sieht es weitestgehend wie jedes andere Dateisystem aus. Es muss jedoch einige Kompromisse eingehen in Bereichen, in denen ein CIFS/SMB-Dateisystem nicht die Funktionalität bietet, die Unix zur Verfügung stellt. Nachfolgend sehen Sie, wie ein solches Dateisystem aussieht, wenn es unter Linux gemountet wurde:

```
Filesystem 1024-blocks Used  Available Capacity Mounted on
/dev/hdb3   876101     430377  400465     52%   /
/dev/sda2   991124       5626  934298      1%   /home
/dev/hdc    572804     572804       0    100%   /mnt/cdrom
//rjspc1/c 1441120    1409344   31776     98%   /mnt/smb
```

Hier können Sie sehen, dass das Dateisystem von `//rjspc1/c` auf `/mnt/smb` gemountet ist. Wie sieht eine lange Auflistung eines Verzeichnisses innerhalb dieses Dateisystems aus? Hier ist ein Beispiel von `/mnt/smb/Samba`:

```
[root@eagle]# ls -al /mnt/smb/Samba
total 773
drwxr-xr-x    1  root  root       512 Jan 17 19:48 .
drwxr-xr-x    1  root  root       512 Jan   1 1970 ..
drwxr-xr-x    1  root  root       512 Jan 18 01:04 Fol1-2
drwxr-xr-x    1  root  root       512 Jan 18 01:05 Fol2-1
-rwxr-xr-x    1  root  root     15391 Nov 27 10:50 log.ulysses
-rwxr-xr-x    1  root  root    770141 Jan 17 19:48 samba-1.9.18p10-source.tar.gz
```

Sie sehen, dass die Auflistungen wie reguläre Unix-Dateien aussehen, aber das Aussehen kann irreführend sein. Das Dateisystem wurde von einem Windows-95-Rechner gemountet, der nicht dem Konzept der Benutzer oder Gruppen folgt, was das Eigentum der Dateien betrifft. `smbfs` erfindet die Benutzer- und Gruppeninformationen basierend auf Informationen (oder Standardwerten), die `smbmount` gegeben werden. Außerdem sind auch die Berechtigungen nicht unbedingt vollständig, da `smbfs` diese Informationen von anderen Informationen (oder Standardwerten) ableitet, die `smbmount` gegeben wurden.

Ohne Sorgfalt könnte es gut sein, dass eine Datei scheinbar vom aufgelisteten Besitzer gelöscht werden kann, obwohl der Server dies tatsächlich nicht zulassen würde.

Dennoch kann `smbfs` für einen Systemadministrator eine sehr bequeme Methode für den Zugriff auf Dateisysteme von CIFS/SMB-Servern für Backup- oder Admininstrationszwecke sein.

Die folgenden Abschnitte stellen die Funktionsweisen der zwei vorher erwähnten Utilities dar. Generell müssen beide ähnliche Probleme lösen, gehen diese aber aufgrund ihrer Eigenschaften unterschiedlich an.

Beispiele für Samba-smbmount

Damit Sie diese Utilities benutzen können, müssen Sie Samba mit `smbmount`-Unterstützung übersetzen. Dafür starten Sie `configure`, übergeben die Option `--with-smbmount` und beenden den Aufbau wie in Kapitel 3, »Wie bekomme ich den aktuellsten Source-Code?«, beschrieben. Die Utilities können nur auf einem Kernel 2.1.71 oder höher kompiliert werden.

Das Samba-Utility `smbmount` wurde entwickelt, indem nicht benötigter Code aus `smbclient` entfernt und die Fähigkeit hinzugefügt wurde, ein entferntes CIFS/SMB-Dateisystem zu mounten. Dies erreichen Sie durch Ausführen des `smbmnt`-Utility, das alle notwendigen Aufgaben durchführt, nachdem alle von `smbmnt` benötigten Befehlszeilenparameter aufgebaut und die gesammelten Parameter überprüft wurden.

Diese Version des `smbmount`-Utility ist daher in seiner Benutzung von Befehlszeilenparametern `smbclient` sehr ähnlich. Die von `smbmount` akzeptierten Befehlszeilenparameter finden Sie im Abschnitt »`smbclient`«.

Ein Beispiel:

```
smbmount //rjspc1/c -c 'mount /mnt/smb -u 123 -g 456'
```

Hier wird `//rjspc1/c` auf `/mnt/smb` gemountet und erhält eine lokale UID von 123 sowie eine lokale GID von 456. Außerdem können Sie hier auch die Benutzung von `-c` sehen.

Weitere Informationen über diese Utilities finden Sie in den Manpages zu `smbmount`, `smbmnt` und `smbumount`.

Beispiele für smbfs-smbmount

Wenn Sie diese Sammlung von Utilities benutzen wollen, müssen Sie das `smbfs`-Paket installiert haben. Sie werden das Paket in Ihrer Distribution als `smbfs-2.0.1-4` finden. Es geht über den Rahmen dieses Buches hinaus, Ihnen zu erklären, wie solche Pakete installiert werden. Bitte werfen Sie einen Blick in die Dokumentation zu Ihrem Linux-System für Anweisungen zur Installation von Paketen.

Nach der Installation können Sie `smbmount` benutzen, um entfernte CIFS/SMB-Dateisysteme auf Ihren Linux-Rechner zu mounten, und `smbumount` für das Unmounten.

Ein Beispiel für die Verwendung dieser Version von `smbmount`:

```
smbmount //rjspc1/c /mnt/smb -u 123 -g 456
```

Hier wird `//rjspc1/c` auf `/mnt/smb` gemountet und erhält eine lokale UID von 123 sowie eine lokale GID von 456.

Weitere Informationen über diese Utilities finden Sie in den Manpages zu `smbmount` und `smbumount`.

smbwrapper

Dies ist ein Experiment für den Zugriff auf CIFS/SMB-Dateisysteme für existierende ausführbare Dateien. Es verwendet die Fähigkeit einer ganzen Reihe von Systemen, Shared Libraries vorzuladen, die die Standard-Systembibliotheken außer Kraft setzen. `smbwrapper` bietet eigene Versionen von vielen der Standard-Systembibliotheks-Routinen. Diese Routinen überprüfen, ob die Operation für eine Datei in einem CIFS/SMB-Dateisystem ausgeführt wird, und ruft in diesem Fall eine entsprechende Routine auf, die das SMB-Protokoll versteht und die Operation durchführt. Wird die Operation jedoch nicht für eine Datei in einem CIFS/SMB-Dateisystem durchgeführt, wird die Standard-Systemversion der Operation aufgerufen.

Die `smbwrapper`-Funktion ist neu in Samba 2.0.0 und wird nicht standardmäßig aufgebaut. Um sie zu kompilieren, müssen Sie Samba mit der Option `--with-smbwrapper` konfigurieren und dann Samba neu kompilieren und installieren. Details hierzu finden Sie in Kapitel 3.

Diese Funktion wurde auf einer ganzen Reihe von Systemen getestet, darunter:

- Linux 2.0 mit glibc2 (Red Hat 5.1)
- Linux 2.1 mit glibc
- Solaris 2.5.1 mit gcc
- Solaris 2.6 mit gcc
- SunOS 4.1.3 mit gcc
- IRIX 6.4 mit cc
- Digital Unix 4.0 mit gcc

Wurde `smbwrapper` kompiliert und installiert, können Sie es über die Eingabe von **smbsh** starten. Beim Starten fordert `smbsh` Sie zur Eingabe eines Benutzernamens und Passworts auf, die dem Benutzernamen und Passwort entsprechen, die Sie für den Zugriff auf die Arbeitsgruppe oder Domäne, in der Sie sind, benutzen wollen.

Nachfolgend finden Sie ein Beispiel für die Benutzung von `smbwrapper`:

```
[root@eagle]# smbsh
Username: boss
Password:
```

Hier haben Sie `smbsh` gestartet und den Benutzernamen `boss` übermittelt, mit dem entsprechenden Passwort für diesen Account. Der Benutzername und das Passwort werden an jeden Server übermittelt, der mit `security=user` operiert:

```
[root@eagle]# ls /smb/rjspc1
C
```

Hier verlangen Sie eine Dateiauflistung des Verzeichnisses `/smb/rjspc1`. Der `/smb`-Teil teilt `smbwrapper` mit, dass Sie sich auf ein CIFS/SMB-Dateisystem beziehen, und das `/rjspc1`-Pseudoverzeichnis teilt `smbwrapper` mit, dass Sie die Freigaben auf dem Server `rjspc1` auflisten wollen. Wie Sie sehen, stellt `rjspc1` nur eine Freigabe zur Verfügung:

```
[root@eagle]# ls /smb/rjspc1/c/samba
Fol1-2        log.ulysses
Fol2-1        samba-1.9.18p10-source.tar.gz
```

Hier haben Sie eine Auflistung der Dateien in einem Verzeichnis namens `samba` in der einen verfügbaren Freigabe von `rjspc1` verlangt. Sie sehen eine Auflistung der Dateien in diesem Verzeichnis. Dies demonstriert, dass Standard-Unix-Programme mit CIFS/SMB-Dateisystemen arbeiten können, auf die über `smbsh` zugegriffen wird:

```
[root@eagle]# more /smb/rjspc1/c/samba/log.ulysses
doing parameter lock directory = /opt/samba/var/locks
doing parameter share modes = yes
doing parameter default case = lower
doing parameter case sensitive = no
doing parameter preserve case = yes
doing parameter short preserve case = yes
Processing section "[homes]"
doing parameter comment = Home Directories
doing parameter browseable = yes
```

Hier sehen Sie, dass ein anderes Standard-Unix-Programm mit CIFS/SMB-Dateisystemen arbeiten kann, auf die über `smbsh` zugegriffen wurde.

Tatsächlich wurde `smbsh` mit einer großen Anzahl von Unix-Programmen und -Utilities getestet und funktioniert recht gut mit allen.

Es gibt jedoch einige Operationen, die nicht mit `smbsh` funktionieren:

- Ausführen einer Datei von einer Freigabe
- Benutzung eines Programms, das `mmap` verwendet
- Weiterleiten an eine Datei in einer Freigabe

Zwar bietet `smbsh` viele Funktionen von `smbclient` in einer Unix-freundlicheren Art und Weise und kann in vielen Situationen als Ersatz für `smbclient` benutzt werden, aber `smbsh` funktioniert noch nicht unter einigen Betriebssystemen und unterstützt einige Funktionen noch nicht, die `smbclient` unterstützt. Daher werden Sie wahrscheinlich noch für einige Zeit sowohl `smbclient` als auch `smbsh` in zukünftigen Distributionen sehen.

smbprint

`smbprint` ist ein einfaches in Samba integriertes Shell-Skript, über das Unix-Systeme an Drucker auf anderen CIFS/SMB-Servern drucken können. Obwohl `smbprint` bereits detailliert in Kapitel 8, »Drucker«, dargestellt wurde, lohnt es sich, hier weitere Details darzustellen.

`smbprint` benutzt für die Ausführung seiner Aufgaben `smbclient` und seine Befehlszeilenfunktionen sowie seine Fähigkeit, STDIN zu handhaben, um die zu druckende Datei in einer einfachen Pipeline weiterzuverarbeiten. Ein Beispiel:

```
(
   echo translate
   echo print -
   cat
)| smbclient "\\\\$Server\\$Freigabe" $Passwort -u $Server -N -P
```

Diese Syntax ist nicht genau die gleiche, die `smbprint` benutzt, verdeutlicht aber die gleichen Prinzipien:

1. In einer Subshell teilen zwei `echo`-Aussagen `smbclient` mit, alle Eingaben zu übersetzen und von STDIN (`-`) zu drucken. Sie führen diese Befehle in einer Subshell (umschlossen von Klammern) aus, damit Sie über STDIN einen Befehlsstream an `smbprint` übertragen können.
2. Der Befehl `cat` kopiert von STDIN an STDOUT, aber STDOUT ist mit dem nächsten Schritt der Pipeline verbunden und STDIN mit der von `lpd` zu druckenden Datei. Dieser Befehl wird ebenfalls in der Subshell ausgeführt, in der die zwei vorherigen ausgeführt werden.
3. `smbclient` wird gestartet, verbindet sich mit dem Server, der in `$Server` spezifiziert ist und der Freigabe, die in `$Freigabe` spezifiziert ist (wobei `$Server` und `$Freigabe` von `smbprint` eingerichtet sind), benutzt die anderen erhaltenen Parameter und verarbeitet STDIN.
4. Da aber STDIN mit der vorher erwähnten Subshell verbunden ist, erhält es den Befehl `translate`, dann den Befehl `print -` und schließlich die zu druckende Datei, alles in STDIN.

smbtar

Um das Leben der Systemadministratoren leichter zu machen, enthält Samba ein Shell-Skript, über das Sie tar-Backups entfernter CIFS/SMB-Server durchführen können. Dieses Skript ist `smbtar`, das `smbclient` für die Durchführung seiner Funktionen benutzt.

`smbtar` hat folgendes generelles Format:

`smbtar *Optionen Dateien*`

`smbtar` erhält alle notwendigen Informationen über die Befehlszeile und nimmt folgende Befehlszeilenparamter an:

`-s *Server*`	Dieser notwendige Parameter spezifiziert den Server, auf dem sich die Freigabe befindet, die Sie sichern wollen.
`[-p *Passwort*]`	Dieser optionale Parameter spezifiziert das Passwort für die entfernte Freigabe oder das Benutzerpasswort für den Server. Es gibt kein Standardpasswort.
`[-x *Freigabe*]`	Dieser optionale Parameter spezifiziert die Freigabe, auf die zugegriffen wird. Wird hier nichts angegeben, verbindet `smbtar` sich standardmäßig mit einer Freigabe namens `backup`.
`[-X]`	Dieser optionale Parameter definiert, dass alle in der Befehlszeile angegebenen Dateinamen aus der erstellten oder wiederhergestellten tar-Datei ausgeschlossen werden sollen.
`[-d *Verzeichnis*]`	Dieser optionale Parameter spezifiziert das Startverzeichnis, in das gewechselt werden soll, bevor Dateien gesichert oder wiederhergestellt werden.
`[-u *Benutzer*]`	Dieser optionale Parameter spezifiziert den Benutzernamen, unter dem die Verbindung zum Server aufgebaut werden soll. Gibt es hier keine Angabe, wird standardmäßig der Login-Name des Benutzers verwendet, der den `smbtar`-Befehl ausführt.
`[-t *Band*]`	Dieser optionale Parameter spezifiziert das Bandgerät oder die Datei, auf dem oder in der die Daten gesichert bzw. von dem oder aus der die Daten wiederhergestellt werden. Wird hier keine Angabe gemacht, benutzt `smbtar` die Umgebungsvariable `TAPE` oder, wenn auch die nicht gesetzt ist, `tar.out`.
`[-b *Blockgröße*]`	Dieser optionale Parameter spezifiziert den Band-Blockfaktor. Der Wert muss eine ganze Zahl größer als 0 sein. Der Standardwert ist 20.
`[-N *Dateinamen*]`	Dieser optionale Parameter spezifiziert, dass `smbtar` nur Dateien sichern sollte, die neuer als `*Dateinamen*` sind.
`[-i]`	Über diesen optionalen Parameter wird ein inkrementelles Backup durchgeführt. Das heißt, nur die Dateien mit gesetztem DOS-Archivbit werden gesichert.
`[-a]`	Über diesen Parameter wird das Archivbit auf allen gesicherten Dateien zurückgesetzt. Standardmäßig wird das Archivbit nicht geändert.
`[-r]`	Über diesen optionalen Parameter wird eine Wiederherstellung statt eines Backups durchgeführt.
`[-l *Loglevel*]`	Dieser optionale Parameter spezifiziert den Debug-Level und wird über das Flag `-d` an `smbclient` weitergeleitet. Der Standard-Loglevel ist 0.
`[-v]`	Über diesen optionalen Parameter arbeitet `smbtar` im Verbose-Modus.
`[*Dateinamen*]`	Diese optionale Liste von Dateien wird eingeschlossen oder ausgeschlossen, abhängig davon, ob die Option `-X` eingefügt wurde.

Auf einigen Systemen, auf denen die Funktion `getopts` nicht korrekt in der Standard-Systemshell implementiert ist (z.B. Digital Unix), sollten Sie die erste Zeile des `smbtar`-Skripts von `#!/bin/sh` in `#!/usr/bin/ksh` ändern. Ohne diese Änderung funktioniert `smbtar` nicht richtig und gibt Fehlermeldungen über `OPTIND` aus.

Andere Clients

Wahrscheinlich werden Sie auch auf andere CIFS/SMB-Clients treffen als die, die vorher in diesem Kapitel erwähnt wurden. Einer ist z.B. Sharity, ein Produkt für viele Unix-Derivate, das `smbfs` ähnelt, aber die Fähigkeit hat, SSL für die Verbindung zu Servern zu benutzen. Dies bietet erhebliche Verbesserungen in Hinsicht auf Sicherheit. Weitere Informationen über Sharity finden Sie auf der folgenden Webseite: http://www.obdev.at/Products/Sharity.html.

Zusammenfassung

In diesem Kapitel haben Sie sich viele der Unix-Clients detailliert angesehen, die mit Samba verfügbar oder für das CIFS/SMB-Protokoll relevant sind. Diese Clients bieten alle einen relativ Unix-freundlichen Zugriff auf CIFS/SMB-Dateisysteme.

In Kapitel 14, »Windows 9x und Windows NT«, werden Sie sich die verschiedenen SMB-Clients für Windows und in Kapitel 15, »Andere SMB-Clients«, andere Clients ansehen, z.B. für Mac OS, OS/2 Warp und MS-DOS.

Frage & Antwort

F. Wie würden Sie in smbsh die translate-Funktion von smbclient zur Verfügung stellen?

. Leiten Sie die Datei durch `fromdos` oder `todos`, je nachdem, in welche Richtung die Datei geht.

F. Ich habe in meinem Unternehmen gerade Amanda installiert, benutze Samba 1.9.18p10, und meine Backups sind verfälscht. Warum hat Amanda so viele Probleme mit Samba?

. In Samba 1.9.18p7 oder so wurde ein Fehler gefunden, bei dem sowohl die tar-Ausgabe als auch informative Meldungen über den Fortgang des Backups an STDOUT übertragen wurden; daher wurde das Archiv beschädigt.

Es gibt verschiedene Korrekturen für dieses Problem. Die Leute von Amanda haben einen Patch auf ihrer Website zur Verfügung gestellt, der das Problem löst. Sie können auch dem `smbclient`-Befehl, den Amanda benutzt, die Option `-E` hinzufügen. Und schließlich wurde ein Patch in den 2.0- und den 2.1-Source-Code integriert, der dieses Verhalten stoppt und informative Meldungen an STDERR lenkt, wenn STDOUT für das tar-Archiv benutzt wird. Er ist in Samba 2.0.1 und höher verfügbar.

F. Wenn ich Dateien auf einem Windows-95-Rechner wiederherstelle, fällt smbclient manchmal aus, nachdem es mir mitgeteilt hat, dass es die Erstellungszeit einer Datei nicht aktualisieren konnte. Was kann ich tun?

. Windows 95 und Windows NT verhalten sich unterschiedlich, wenn das Erstellungsdatum einer Datei geändert werden soll. Windows NT führt die verlangte Aufgabe ohne Probleme durch, während Windows 95 die verlangte Aufgabe durchführt und eine Fehlermeldung zurückgibt. Die Version von `smbclient` in Samba 2.0.0 beinhaltet eine Korrektur für dieses Problem (es ignoriert die Antwort vom Server, wenn es die Erstellungszeit ändert); Sie sollten also auf die Version 2.0.0 aktualisieren.

Tag 14: Windows 9x und Windows NT

Der Zugriff auf SMB-Server von neueren Windows-Betriebssystemen ist relativ leicht zu konfigurieren. Bis hierher habe ich noch keines der Konfigurationsdetails erwähnt, aber höchstwahrscheinlich haben Sie die grundlegenden notwendigen Komponenten bereits installiert.

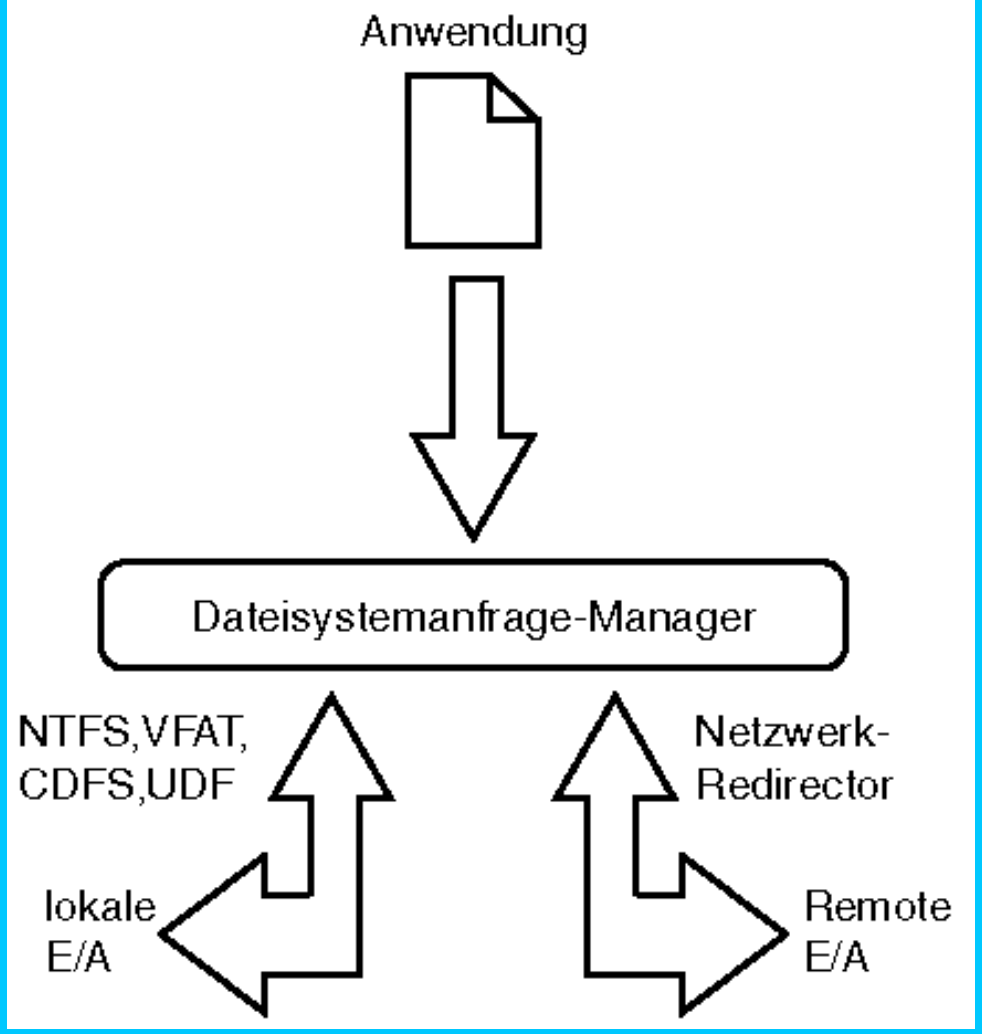
In diesem Kapitel werde ich darstellen, wie Sie den in Windows 9x und Windows NT integrierten SMB-Client installieren, konfigurieren und benutzen. Ich behandle Windows-95- und Windows-98-Clients zusammen, da diese beiden fast identisch sind. Es gibt nur wenige Veränderungen in der Netzwerkkonfiguration zwischen den beiden Versionen. Windows NT ist ein vollkommen anderes Betriebssystem und wird daher separat dargestellt.

Der Windows-Netzwerk-Redirector

Bevor wir uns auf die Details für die Konfiguration des Windows-Netzwerk-Clients konzentrieren, möchte ich mir ein bisschen Zeit nehmen und Ihnen ein wenig über die Windows-Netzwerkarchitektur erzählen. Sie sollten beachten, dass der Windows-9x-Netzwerk-Redirector und der, der Teil von Windows NT ist, sich erheblich unterscheiden, aber die grundlegende Theorie hinter beiden ist ähnlich.

Abbildung 14.1 zeigt eine vereinfachte Darstellung, wie Windows den Zugriff auf entfernte Netzwerkfreigaben handhabt. Ich setze für meine Beschreibung voraus, dass es sich nur um Festplattenfreigaben handelt. Die Erklärung für entfernte Drucker ist ähnlich.

Abb. 14.1: Eine generelle Implementierung des Netzwerk-Redirectors



Die Implementierung umfasst drei grundlegende Teile:

- Die Anwendung, die Zugriff auf die Festplatte verlangt.
- Irgendeine Art von Dateisystemanfrage-Manager. Windows 95 und 98 implementieren dies über einen installierbaren FileSystem Manager. Der Arbeitsstationsdienst handhabt Anfragen auf Benutzerebene in Windows NT.
- Den Netzwerk-Redirector.

Der Prozess funktioniert in etwa so:

1. Die Anwendung verlangt Zugriff auf die Festplatte.
2. Die Anfrage wird an den Dateisystemanfrage-Manager übertragen.
3. Dieser leitet die Anfrage dann an den entsprechenden Dateisystem-Gerätetreiber weiter. Ist der verlangte Zugriff lokal, wird der lokale Dateisystemtreiber (NTFS, VFAT, CDFS, UDF usw.) die Anfrage bearbeiten. Gilt die Anfrage einer Festplatte auf einer Netzwerkfreigabe, wird die Information als SMB-Paket übersetzt und in das Netzwerk gesendet.

4. Der verlangte Festplattenzugriff wird über den Dateisystemanfrage-Manager an die Anwendung zurückgegeben.

Ich gebe zu, dass dies möglicherweise ein zu vereinfachter Blick auf den Netzwerk-Redirector ist, aber immerhin werden Sie so eine allgemeine Vorstellung von dem bekommen, was ich meine, wenn ich vom Netzwerk-Redirector spreche.

Windows 9x

Windows 98 ist die aktuellste Ausgabe in der Linie von Windows-Betriebssystemen, die für den persönlichen Gebrauch entwickelt wurden. Wie das FAT16-Dateisystem hat es die ursprüngliche Intention und Kapazität übertroffen. Aber eine der großartigsten Verbesserungen zwischen Windows 95 und den vorherigen 16-Bit-Versionen liegen im Netzwerkbereich. Der Netzwerk-Code ist robuster und stabiler. Ich sage nicht, dass er perfekt ist, aber er ist, sagen wir, Windows für Workgroups um Lichtjahre voraus. (An meinem Arbeitsplatz liefere ich immer noch Support für etwa 125 WfWG-3.11-Clients.)

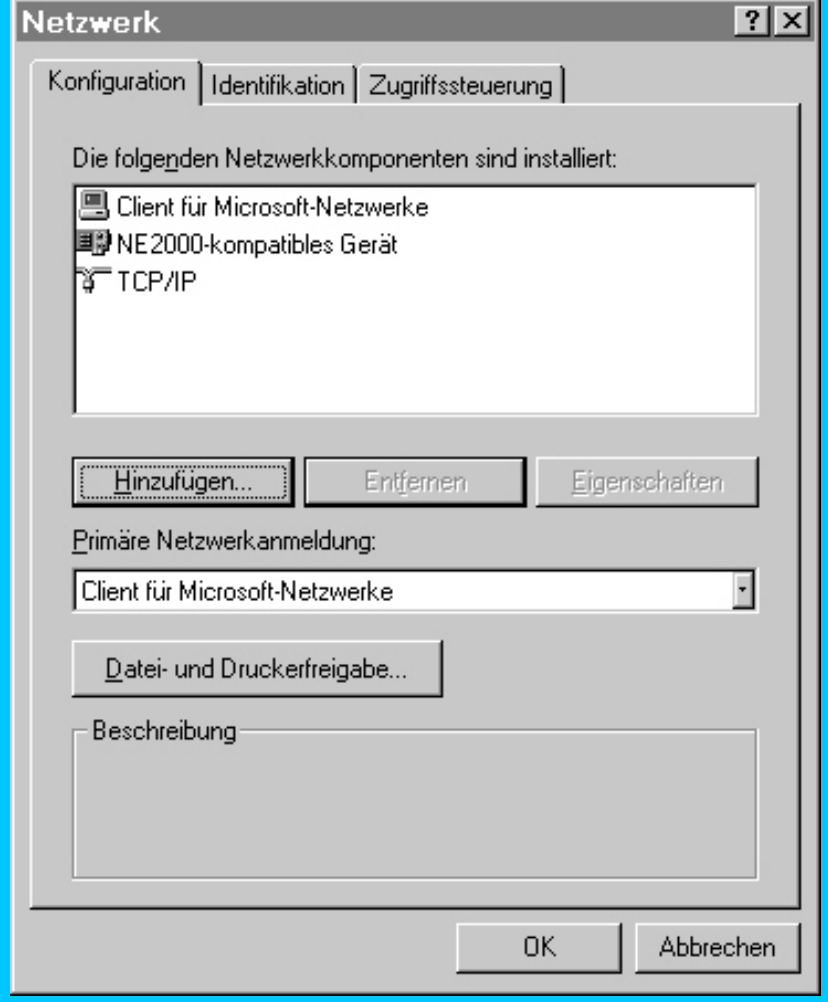
Den Client konfigurieren

Ich werde mich auf drei Komponenten für die Konfiguration des Windows-95-Clients konzentrieren:

- Den Treiber für die Netzwerkkarte
- Das TCP/IP-Protokoll
- Den SMB-Client (alias Client für Microsoft-Netzwerke)

Diese sehen Sie im Netzwerk-Kontrollfenster in Abbildung 14.2.

Abb. 14.2: Die notwendigen Netzwerkkomponenten für die Verbindung zu Freigaben auf einem Samba-Server



Die Netzwerkkarte

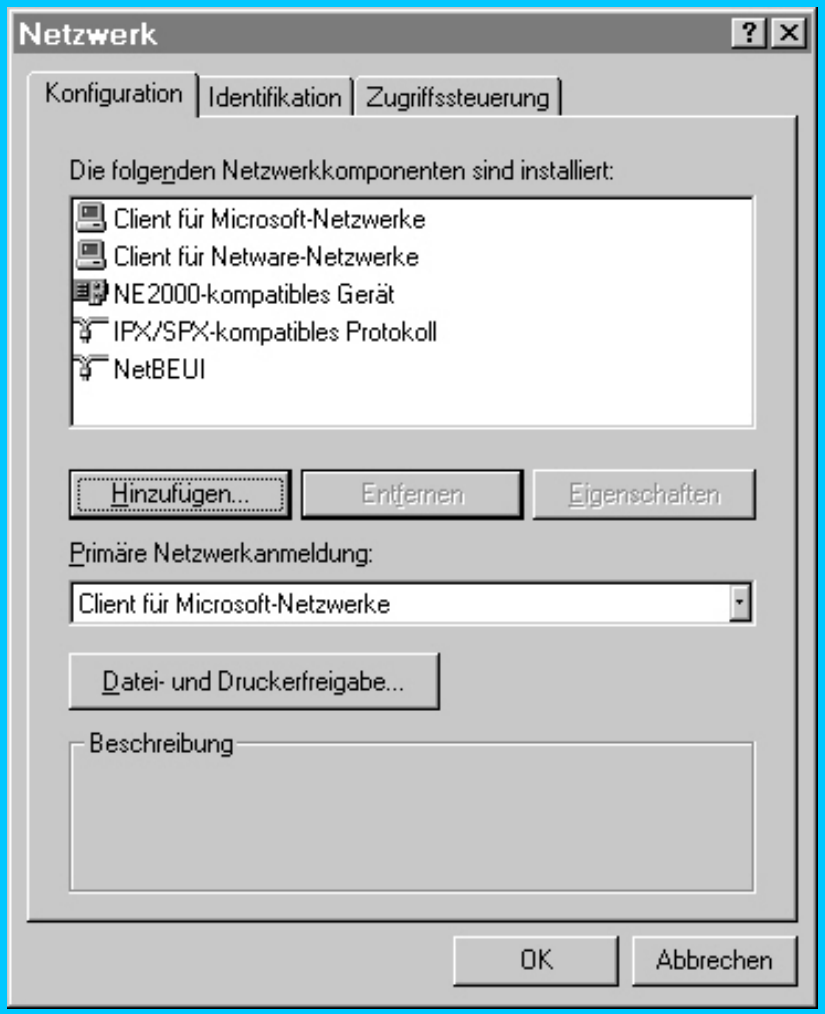
Der Treiber für die Netzwerkkarte ist gewöhnlich das Erste, was Sie installieren werden. Eine detaillierte Beschreibung der Installation jeder erdenklichen Netzwerkkarte geht über den Rahmen dieses Buches hinaus, aber Windows 9x erkennt und konfiguriert die meisten modernen Netzwerkkarten automatisch. Wenn Sie ältere 16-Bit-ISA-Karten benutzen, müssen Sie wahrscheinlich die Karte manuell konfigurieren und die Treiber-Software selber installieren. Netzwerkkarten werden normalerweise mit einer Diskette vertrieben, die Anweisungen für die Installation und die aktuellsten Treiber enthält. Es ist jedoch immer eine gute Idee, einen Blick auf die Website des Herstellers zu werfen, um zu sehen, ob Sie die aktuellste Version der Treiberdiskette haben.

Bei der Erstinstallation einer Netzwerkkarte fügt Windows 9x zusätzlich zum Gerätetreiber für die Netzwerkkarte vier Netzwerkkomponenten hinzu. Dies passiert unabhängig davon, ob Windows die Netzwerkkartentreiber automatisch lokalisiert und installiert. Die vier Komponenten sind:

- Client für Microsoft-Netzwerke
- Client für NetWare-Netzwerke
- IPX/SPX-kompatibles Netzwerkprotokoll
- NetBEUI--Netzwerkprotokoll

Abbildung 14.3 zeigt das Netzwerk-Kontrollfenster nach der Installation einer neuen NE-2000-kompatiblen Karte. Windows fand statt des genauen Herstellers und Modells eine Entsprechung des Chipsatzes auf der Karte, darum wird die Netzwerkkarte als NE-2000-kompatible Karte bezeichnet.

Abb. 14.3: Die von Windows 9x installierten Standard-Netzwerkkomponenten



Sie sollten die folgenden zwei Netzwerkprotokolle entfernen, indem Sie auf den entsprechenden Eintrag in der Liste und dann auf die Schaltfläche *Entfernen* klicken:

- IPX/SPX-kompatibles Netzwerkprotokoll
- NetBEUI

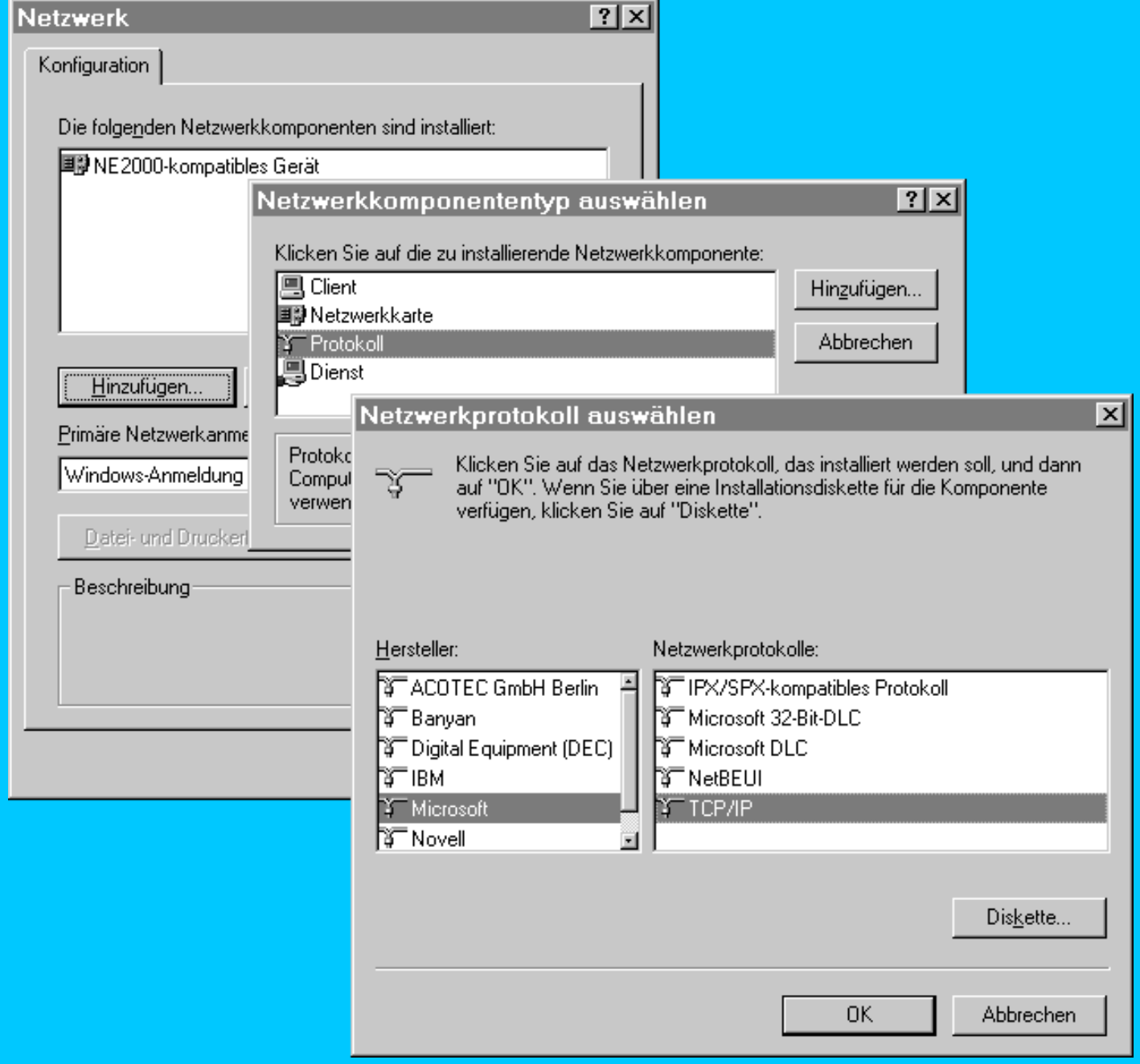
Dies wird auch die zwei Clients entfernen. Sie werden einen von ihnen später brauchen, aber darüber sollten Sie sich jetzt keine Gedanken machen. Die resultierende Liste sollte nur den Treiber für Ihre spezielle Netzwerkkarte enthalten, wie Sie es in Abbildung 14.4 sehen.

Abb. 14.4: Das Windows-95-Netzwerkkontrollfenster nach Entfernen der Standardkomponenten

Das TCP/IP-Protokoll

Nachdem Sie die Netzwerkkarte korrekt installiert haben, besteht der nächste Schritt darin, das TCP/IP-Netzwerkprotokoll hinzuzufügen. Dafür klicken Sie einfach auf die Schaltfläche *Hinzufügen* im Netzwerk-Kontrollfenster und wählen *Protokoll* aus der Liste, die nun erscheint. Klicken Sie auf die Schaltfläche *Hinzufügen* in diesem Fenster. Das sich jetzt öffnende Fenster enthält zwei Listen. Die Liste auf der linken Seite enthält eine Zusammenfassung der verfügbaren Protokolle nach Unternehmensnamen. Wenn Sie ein Unternehmen auswählen, zeigt die Liste auf der rechten Seite die Protokolle, die von diesem Hersteller verfügbar sind. Sie sollten links auf *Microsoft* klicken und rechts auf *TCP/IP*. Diese drei Fenster sehen Sie in Abbildung 14.5.

Abb. 14.5: Installation des Microsoft-TCP/IP-Netzwerkprotokolls



Wenn Sie im Dialogfeld *Netzwerkprotokoll auswählen* auf *OK* klicken, wird das TCP/IP-Protokoll in die Liste der installierten Netzwerkkomponenten eingefügt (siehe Abbildung 14.6). Sie können verschiedene Optionen für TCP/IP auf dem PC konfigurieren, indem Sie TCP/IP in der Liste markieren und auf die Schaltfläche *Eigenschaften* klicken.

Abb. 14.6: Liste installierter Netzwerkkomponenten nach Hinzufügen des TCP/IP-Protokolls

In der ersten Einstellungsseite im anschließend erscheinenden Fenster können Sie die lokale IP-Adresse und die Subnetzmaske eintragen. Es gibt zwei Möglichkeiten: Die erste besteht darin, eine IP-Adresse von einem *DHCP- (Dynamic-Host-Configuration-Protocol-)*Server zu erhalten und die zweite darin, die zwei Einstellungen manuell vorzunehmen. Für dieses Beispiel werden Sie die IP-Adresse und die Subnetzmaske manuell konfigurieren (siehe Abbildung 14.7).

Abb. 14.7: Einstellungen für die IP-Adresse im Dialogfeld Eigenschaften für TCP/IP

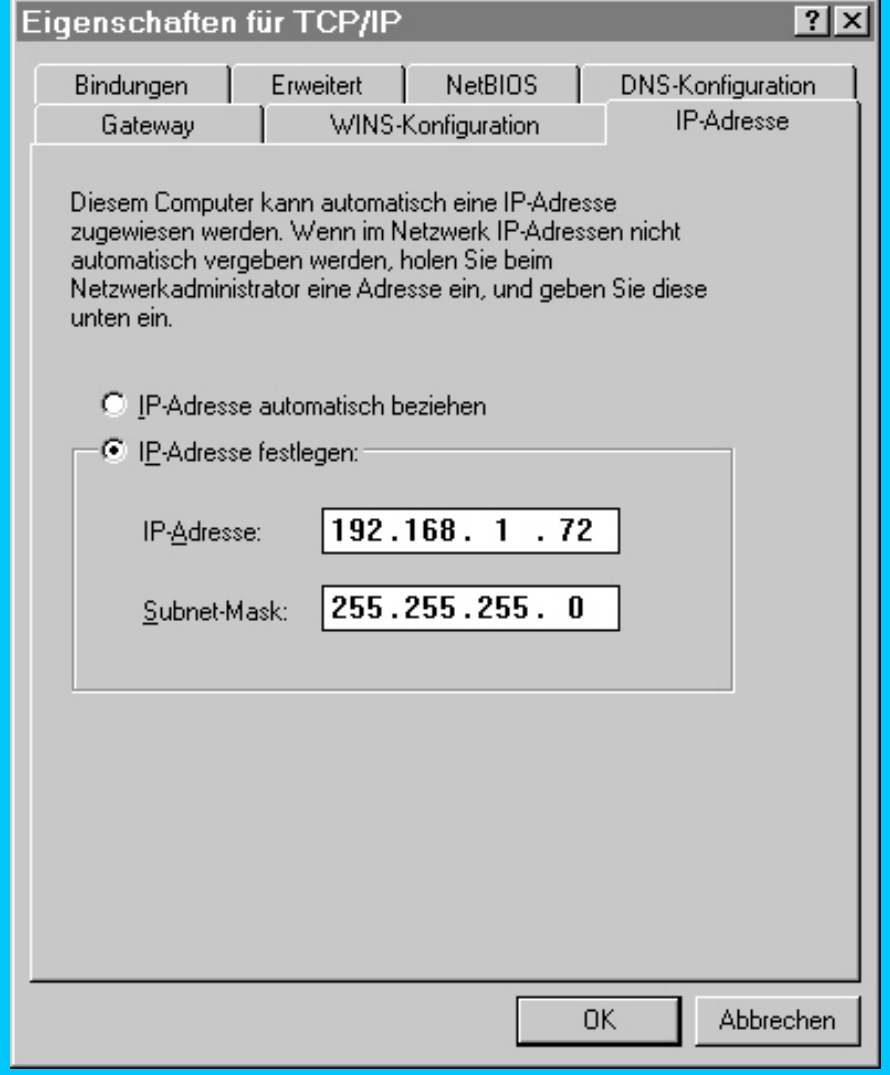
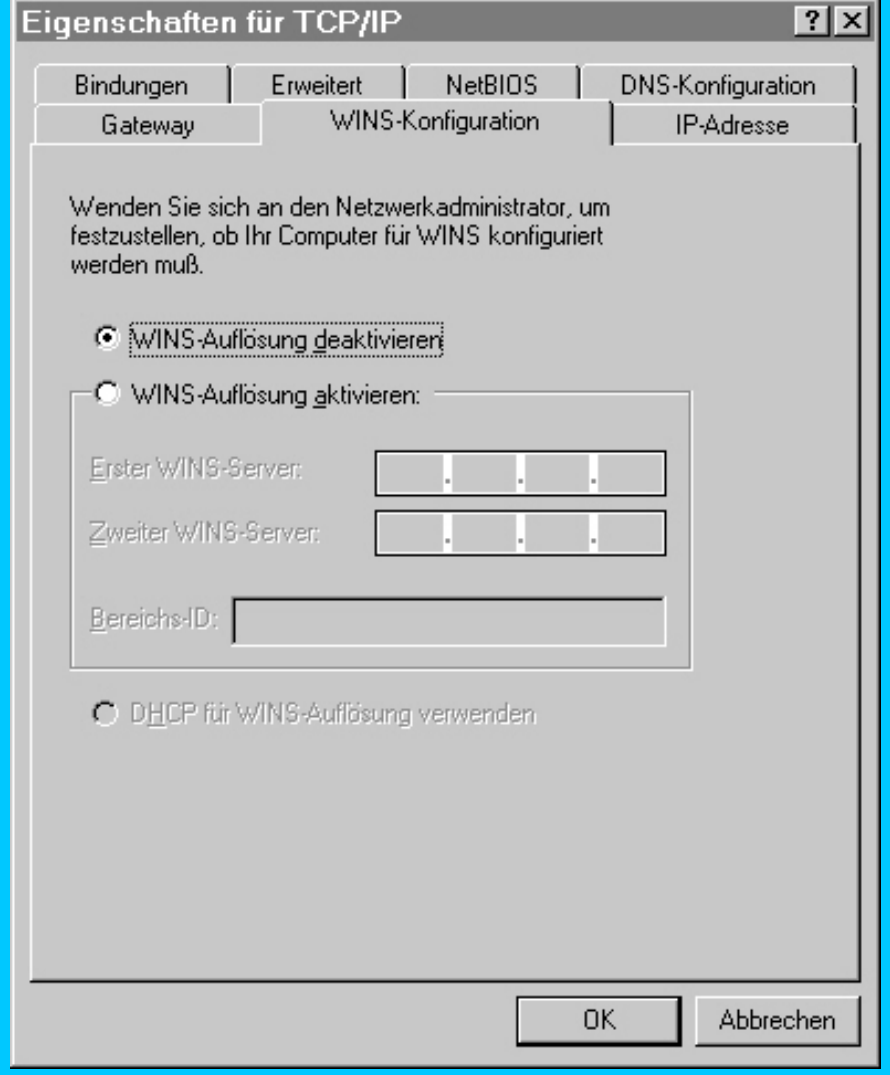


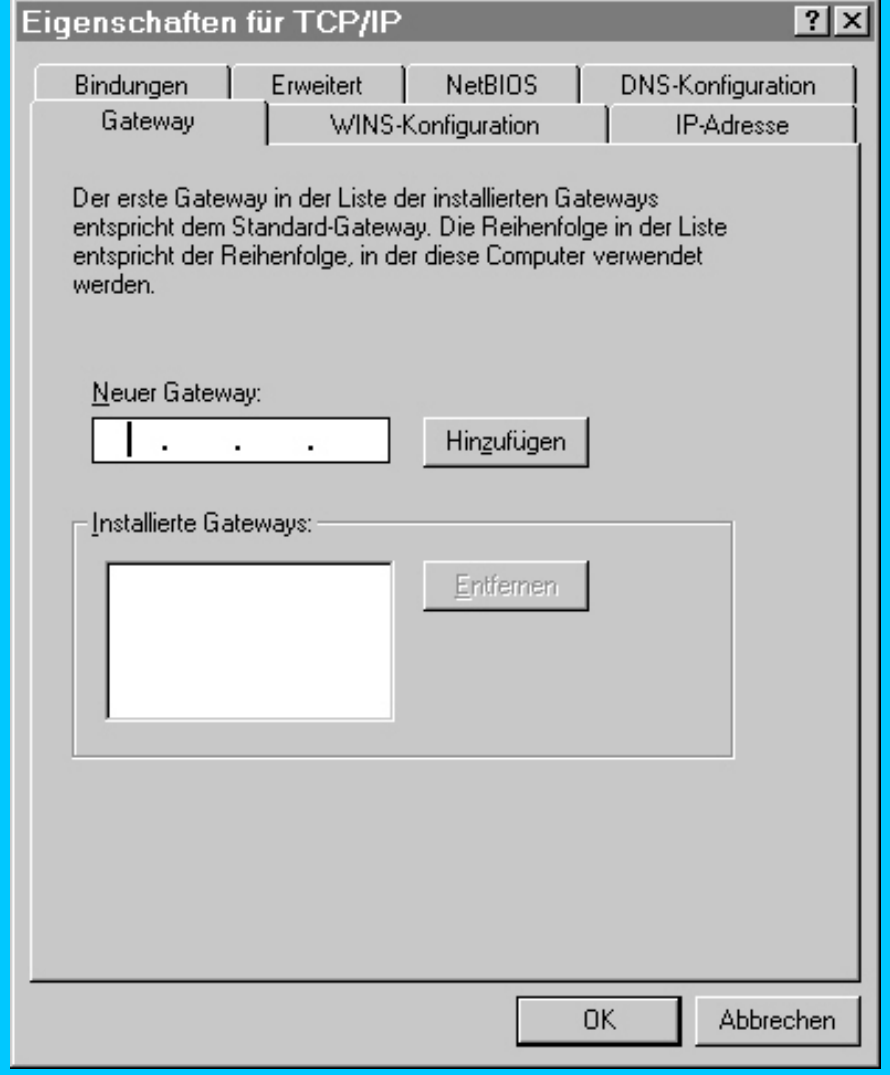
Abbildung 14.8 zeigt die nächste Registerkarte für die TCP/IP-Einstellungen. Sie haben die Registerkarte *WINS-Konfiguration* bereits in Kapitel 2, »Windows-Netzwerke«, gesehen. Über diese Registerkarte können Sie WINS-Auflösung aktivieren und die IP-Adresse Ihres WINS-Servers spezifizieren. Ist WINS-Auflösung aktiviert, können Sie auch eine Bereichs-ID definieren. Außerdem können Sie die IP-Adressen für die WINS-Server auch vom DHCP-Server erhalten. An diesem Punkt sollten Sie WINS einfach deaktivieren. Weitere Informationen über WINS finden Sie in Kapitel 18, »NetBIOS-Namen ohne Broadcasts auflösen«.

Abb. 14.8: Die Registerkarte WINS-Konfiguration im Dialogfeld Eigenschaften für TCP/IP



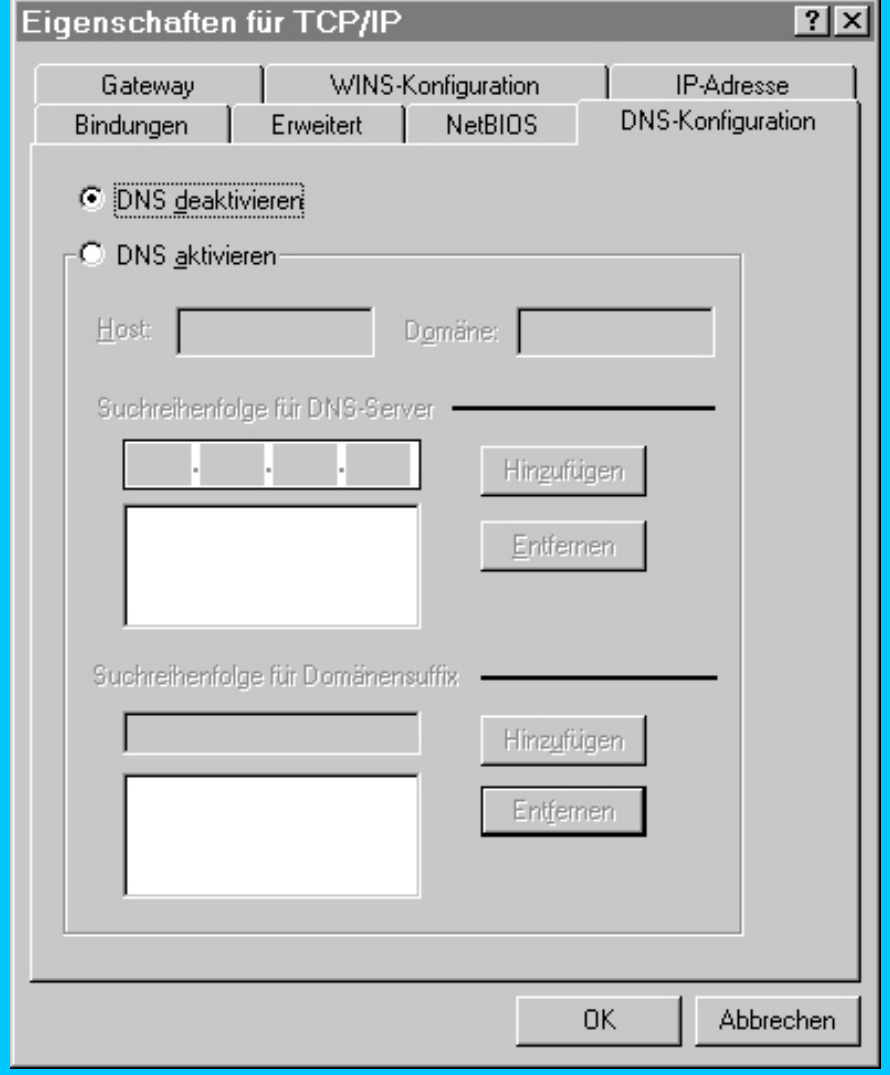
Wenn Ihr Subnetz ein Standard-Gateway für den Datenverkehr hat, der für ein anderes IP-Netzwerk bestimmt ist, können Sie die IP-Adresse des Routers in der Registerkarte *Gateway* festlegen (siehe Abbildung 14.9). Für das kleine Netzwerk, das ich zu Hause installiert habe, verwende ich keinen Standard-Gateway, deshalb ist dieses Feld leer.

Abb. 14.9: Den Standard-Gateway im Dialogfeld Eigenschaften für TCP/IP einrichten



Die letzten TCP/IP-Einstellungen, die ich hier darstellen werde, sind die DNS-Einstellungen für den PC (siehe Abbildung 14.10). Sie haben die Wahl, DNS entweder zu aktivieren oder zu deaktivieren. Wenn Sie DNS aktivieren, können Sie die IP-Adressen Ihrer DNS-Server, die Suchreihenfolge für Domänensuffixe sowie den Hostnamen und die Domäne des PC definieren. Für mein kleines Netzwerk zu Hause sind auch diese Einstellungen nicht notwendig. Ich möchte noch darauf hinweisen, dass Sie die IP-Adressen der DNS-Server auch über DHCP einrichten können, wenn Sie es wollen.

Abb. 14.10: Die Registerkarte DNS-Konfiguration im Dialogfeld Eigenschaften für TCP/IP

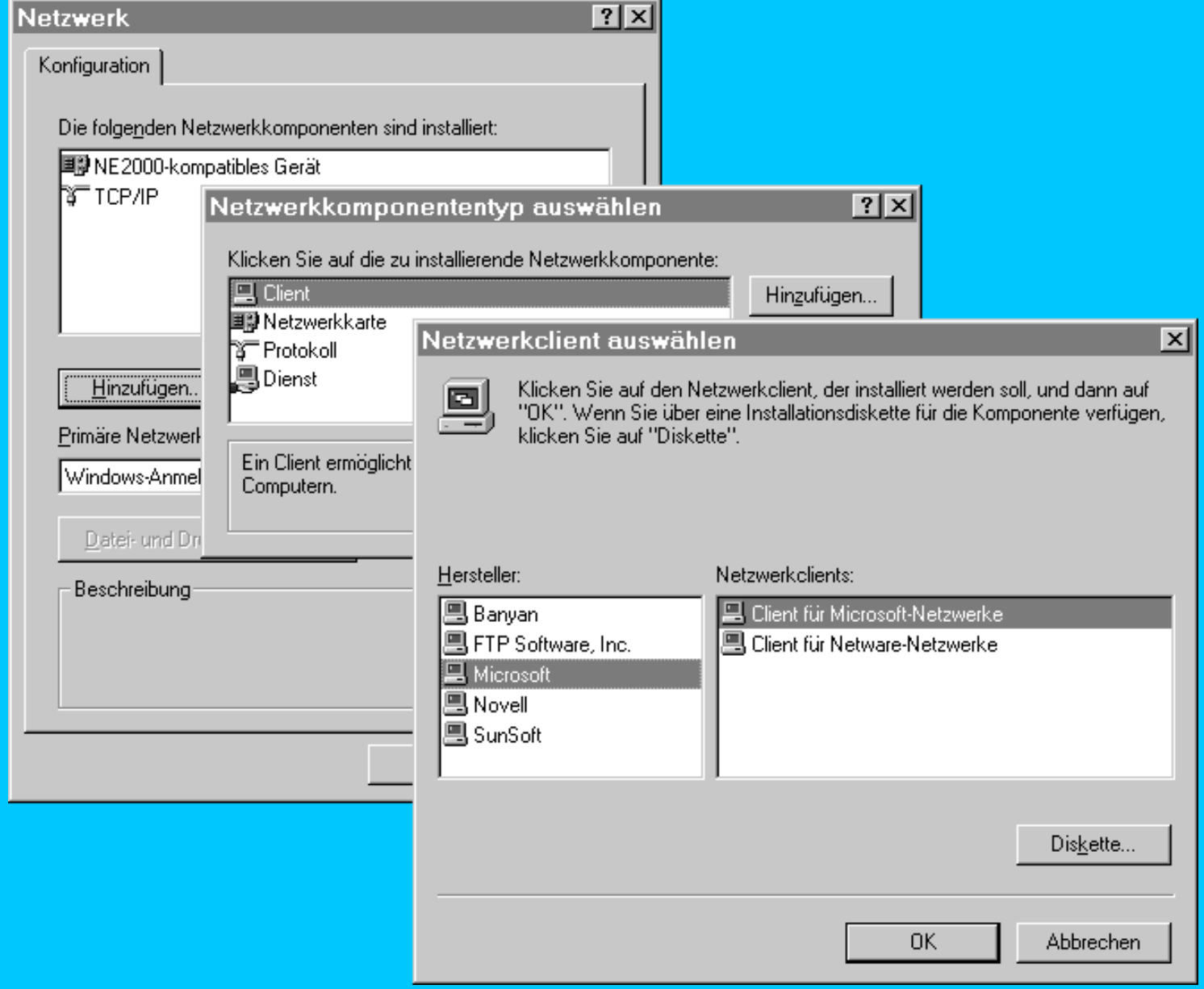


Der Client für Microsoft-Netzwerke

Die letzte Netzwerkkomponente, die Sie installieren müssen, ist der SMB-Client. Wie ich schon in vorherigen Kapiteln erwähnt habe, hat Microsoft das SMB-Protokoll als Ressourcenfreigabe-Mechanismus für sein Netzwerkmodell gewählt. Daher wird der SMB-Client Client für Microsoft-Netzwerke genannt.

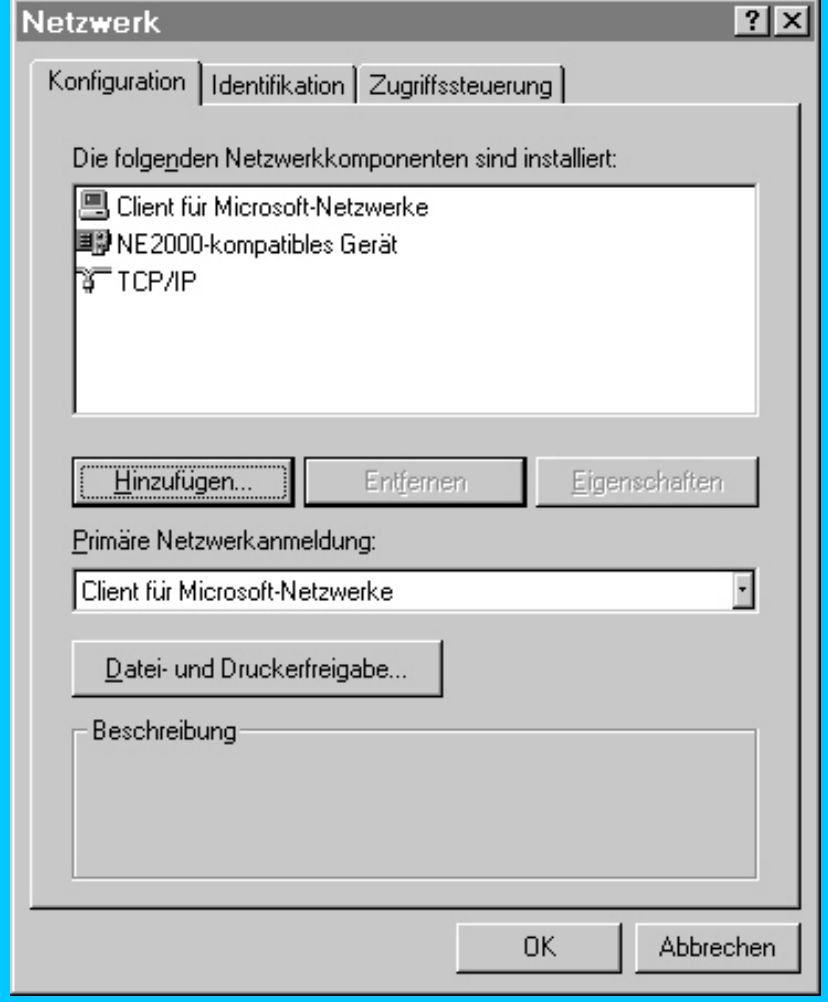
Um den Microsoft-Netzwerk-Client zu installieren, klicken Sie auf die Schaltfläche *Hinzufügen* im Netzwerkkontrollfenster, wie Sie es beim Hinzufügen des TCP/IP-Protokolls getan haben. Wählen Sie dann aus der Liste der Netzwerkkomponententypen *Client* und klicken Sie auch in diesem Fenster auf *Hinzufügen*. Sie sehen ein Diaglogfeld, das dem sehr ähnlich ist, über das Sie das zu installierende Netzwerkprotokoll gewählt haben. Diesmal enthält die Liste der Hersteller auf der linken Seite die Hersteller, die Clients statt Protokolle entwickelt haben. Wählen Sie zuerst *Microsoft* in der linken Liste und dann *Client für Microsoft-Netzwerke* in der rechten Liste. Klicken Sie abschließend im Dialogfeld *Netzwerkclient auswählen* auf *OK*, um den ausgewählten Client hinzuzufügen. Abbildung 14.11 zeigt die drei Fenster, die notwendig sind, um den korrekten Client zu installieren.

Abb. 14.11: Den Client für Microsoft-Netzwerke zur Installation auswählen



Das Netzwerkkontrollfenster sollte nun die drei Netzwerkkomponenten anzeigen, die Sie brauchen (siehe Abbildung 14.12).

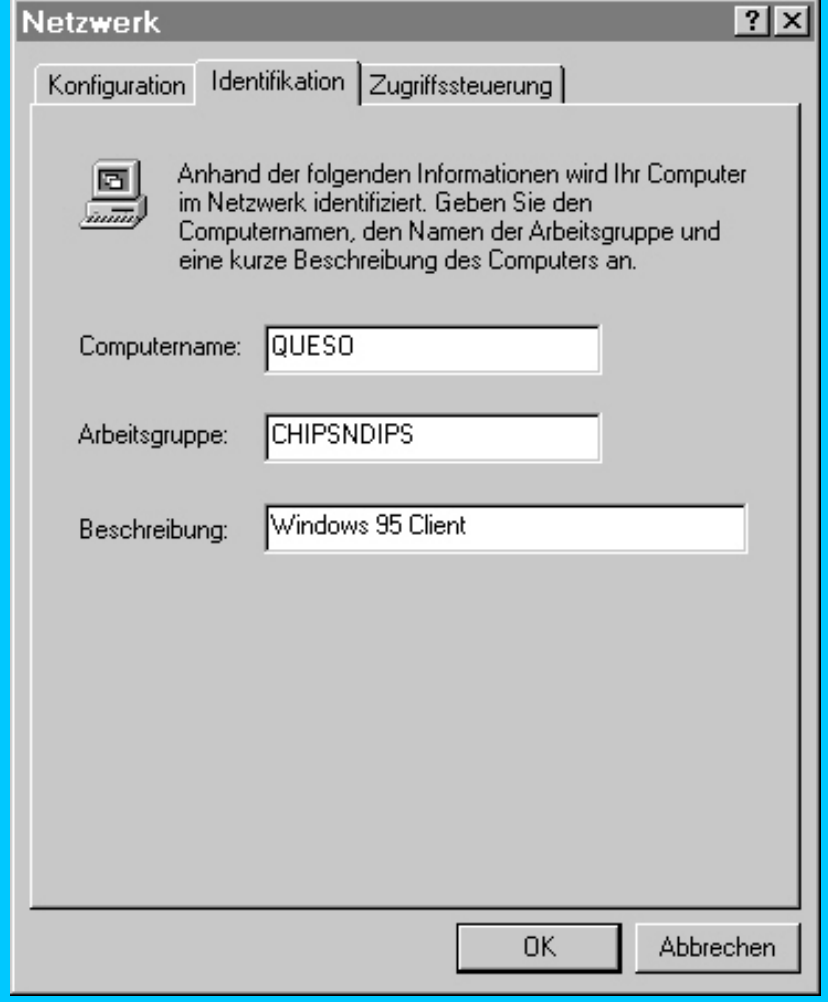
Abb. 14.12: Das Netzwerkkontrollfenster nach Installation des Netzwerkadapters, des TCP/IP-Protokolls und des Microsoft Netzwerk-Clients



Den Namen des Rechners und der Arbeitsgruppe einrichten

Der letzte Schritt, um einen Windows-9x-PC für den Zugriff auf Freigaben auf einem SMB-Server zu konfigurieren, besteht darin, den NetBIOS-Rechnernamen und den Arbeitsgruppennamen einzurichten. Sie können zu Kapitel 2 zurückblättern, um sich noch einmal in Erinnerung zu rufen, aus was ein zugelassener NetBIOS-Name besteht. Um diese zwei Strings zu definieren, muss der Client für Microsoft-Netzwerke installiert sein. Die Registerkarte *Identifikation* (siehe Abbildung 14.13) ist nur dann im Netzwerkkontrollfenster verfügbar, wenn der korrekte Client installiert ist. Im Feld *Beschreibung* auf der Registerkarte *Identifikation* können Sie den Text eingeben, der neben dem Rechnernamen erscheint, wenn Sie Datei- und Druckerfreigaben auf dem PC aktiviert haben. Diese Einstellung entspricht dem Samba-Parameter `server string` in der `smb.conf`. Nachdem Sie die gewünschten Werte eingegeben haben, klicken Sie auf die Schaltfläche *OK*. Windows beginnt, die notwendigen Dateien von den installierten `*.cab`-Dateien oder von der Windows-CD-ROM zu kopieren. Ist das Betriebssystem fertig, fordert es Sie auf, Ihren Rechner neu zu starten. Wenn Sie dies getan haben, können Sie zum nächsten Schritt übergehen und sich in das Netzwerk einloggen.

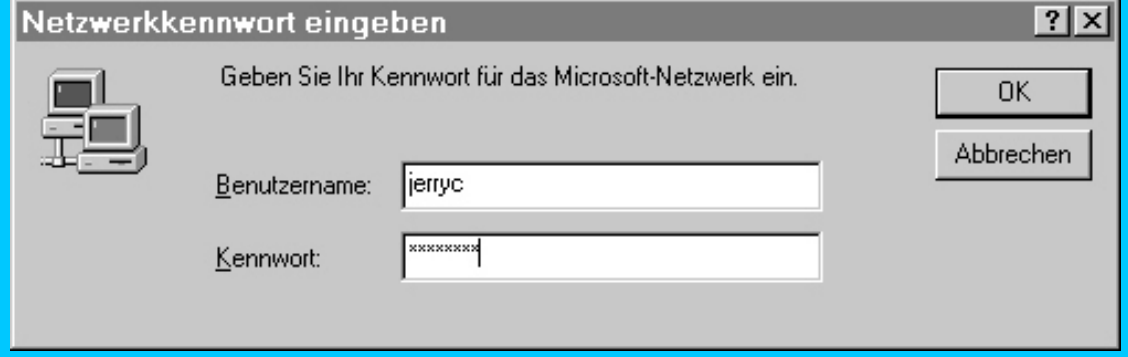
Abb. 14.13: Den NetBIOS-Rechnernamen, den Namen der Arbeitsgruppe und eine Beschreibung für einen Windows-95-OSR2-Client eingeben



In das Netzwerk einloggen

Wenn der PC neu gestartet ist, werden Sie dazu aufgefordert, sich in das Netzwerk einzuloggen (siehe Abbildung 14.14). Es gibt einige Punkte, die Sie bei der Anmeldung in einem Netzwerk beachten sollten:

Abb. 14.14: In ein Microsoft-Netzwerk einloggen



- Der Benutzername und das Passwort, die Sie bei der Anmeldung eingeben, sind die Standardwerte, die Windows für die Verbindung zu entfernten Servern benutzt. Unter Windows 9x können Sie keine unterschiedlichen Benutzernamen für die Verbindung zu verschiedenen Servern benutzen. Wenn Sie sich aber über den bei der Anmeldung angegebenen Benutzernamen und das Passwort nicht mit einem Server verbinden können, verwendet Windows den gleichen Benutzernamen, fordert Sie aber auf, ein neues Passwort einzugeben.
- Der Benutzername und das Passwort, die Sie eingeben, werden nicht beim Anmelden authentifiziert, sondern erst, wenn Sie sich mit einem Server verbinden.
- Sie müssen sich einloggen, um auf Nicht-Gast-Freigaben auf einem SMB-Server zugreifen zu können.

Verschlüsselte und Klartextpasswörter

Die ursprünglichen Distributionen von Windows 95 benutzten Klartextpasswörter als Standard für SMB-Verbindungen. Wenn ein SMB-Server auf eine Protokollverhandlungsanfrage antwortet, enthält das Antwortpaket ein Bit, das angibt, ob der Server die Challenge/Response-Verschlüsselung unterstützt, die in Kapitel 6, »Sicherheitsmodi und Passwörter«, beschrieben ist. Mit der Freigabe des Netzwerk-Redirector-Updates für Windows 95 (`vrdrupd.exe`) hat Microsoft den Standard geändert, so dass Windows-95-Clients kein Klartextpasswort an einen Server übertragen würde, der Verschlüsselung nicht unterstützt. Das bedeutet, wenn Sie versuchen, sich über einen Windows-95-Client, der die folgenden Dateien (oder spätere Versionen):

```
windows\system\vredir.vxd     4.00.1114 6/2/97 11:14a 156,773
windows\system\vnetsup.vxd    4.00.1112 6/2/97 11:12a 17,595
```

installiert hat, mit einem nicht verschlüsselten SMB-Server zu verbinden, werden Sie kontinuierlich dazu aufgefordert, ein Passwort einzugeben, obwohl Sie ein gültiges Passwort für den Account eingegeben haben. Es gibt zwei mögliche Lösungen:

- Richten Sie den Samba-Server mit Hilfe der in Kapitel 6 beschriebenen Schritte für die Benutzung verschlüsselter Passwörter ein.
- Aktivieren Sie die Benutzung von Klartextpasswörtern für den Windows-95-Client.

Wenn Sie die zweite Lösung wählen, müssen Sie folgenden Registrierungsschlüssel hinzufügen und den Client neu booten:

```
[HKLM\System\CurrentControlSet\Services\VxD\VNETSUP]
"EnablePlainTextPassword"=dword:00000001
```

Das Bearbeiten der Registry kann Ihr System unbrauchbar machen. Gehen Sie mit extremer Vorsicht vor, wenn Sie den Registrierungseditor (`regedit.exe`) benutzen.

Bis hierher habe ich in dieser Hinsicht Windows 98 noch nicht erwähnt. Windows 98 verhält sich genauso wie Windows 95 mit installiertem Redirector-Update. Sie können die gleichen zwei Lösungen anwenden, und die gleiche Registrierung aktiviert Klartextpasswörter auf dem Windows-98-Client, wenn Sie dies wünschen.

Mit Freigaben verbinden

Windows unterstützt zwei Interfaces für die Verbindung mit entfernten SMB-Festplattenfreigaben. Das eine ist ein Befehlszeilen-Interface über den Befehl `net.exe`, das andere ein grafisches Interface, das im Windows Explorer enthalten ist. Sie können auch den Dateimanager benutzen, um Laufwerkverbindungen zu etablieren, aber dies ist weniger beliebt.

Zunächst benutzen Sie das Befehlszeilen-Tool `net.exe`. Wenn Sie alle Optionen sehen wollen, die für den `net`-Befehl verfügbar sind, geben Sie Folgendes ein:

```
C:\WINDOWS> net /?
```

Wollen Sie die Optionen für einen bestimmten Befehl sehen - das Argument `use` z.B. - geben Sie Folgendes ein:

```
C:\WINDOWS> net Option /?
```

Die grundlegende Syntax für den Befehl `net use` lautet:

```
net use X: \\Servername\Freigabename
```

wobei `X:` der Laufwerksbuchstabe für die Freigabe und `\\Servername\Freigabename` der UNC-Netzwerkpfad zur Freigabe ist. Ist die Verbindung erfolgreich, sollten Sie eine Ausgabe wie die folgende sehen:

```
C:\WINDOWS>net use h: \\eagle\jerryc
Der Befehl wurde erfolgreich ausgeführt.
```

An diesem Punkt wurde die Freigabe `\\eagle\jerryc` dem Laufwerksbuchstaben `H:` zugeordnet, und auf alle Dateien, die darin enthalten sind, kann auf die gleiche Art und Weise zugegriffen werden wie auf Dateien und Verzeichnisse, die sich auf der lokalen Festplatte, `C:`, befinden.

Es ist auch möglich, sich mit einer SMB-Freigabe zu verbinden, ohne ihr einen Laufwerksbuchstaben zuzuordnen, indem Sie Folgendes eingeben:

```
net use \\Servername\Freigabename
```

Die Verbindung ist im Wesentlichen die gleiche wie für den entsprechenden Laufwerksbuchstaben. Der Unterschied liegt darin, dass Sie den UNC-Namen statt eines Laufwerksbuchstabens, wie z.B. `H:`, benutzen müssen, um auf Dateien zuzugreifen. Davon abgesehen gibt es keine wesentlichen Unterschiede.

Wie ich bereits im Abschnitt über das Einloggen erwähnt habe, versucht Windows, den Benutzernamen und das Passwort, die während der Anmeldung übertragen wurden, für die Verbindung zur entfernten Freigabe zu benutzen. Wenn die Sitzungsaufnahme aus Gründen der Authentifizierung versagt, ermöglicht Windows Ihnen, ein anderes Passwort einzugeben, benutzt aber den gleichen Benutzernamen. Wenn dies passiert, sehen Sie ein Prompt wie das Folgende:

```
C:\users\jerry>net use h: \\eagle\jerryc
```

```
Das Kennwort ist für \\EAGLE\JERRYC ungültig. Weitere Informationen erhalten Sie vom Netzwerkadministrator.
Geben Sie das Passwort ein für \\EAGLE\JERRYC:********
Der Befehl wurde erfolgreich ausgeführt.
```

Wenn Sie Statistiken über eine aktuelle Verbindung sehen möchten, verwenden Sie den Befehl

```
net use X:
```

wobei `X:` der Laufwerksbuchstabe für die Verbindung ist, die Sie ansehen wollen. Hier sind die Informationen für die Verbindung, die ich vorher zu `\\eagle\jerryc` aufgebaut habe:

```
C:\users\jerry>net use h:
Lokaler Name      H:
Remote-Name       \\EAGLE\JERRYC
Ressourcentyp     Platte
Der Befehl wurde erfolgreich ausgeführt.
```

Wenn Sie eine Verbindung zu einer Freigabe beenden wollen, geben Sie folgenden Befehl ein:

```
net use X: /d
```

Damit wird die Verbindung gelöscht. Sie sollten `X:` durch den entsprechenden Laufwerksbuchstaben ersetzen.

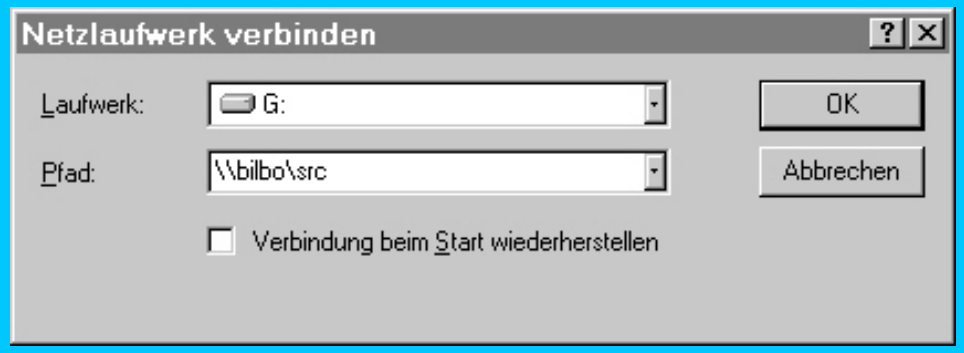
Der Befehl `net use` kann außerdem eine Liste aller aktuellen Freigabeverbindungen anzeigen. Wenn Sie `net use` ohne andere Befehlszeilenargumente ausführen, zeigt Ihnen die Ausgabe alle aktuellen Freigabeverbindungen, auch die, die keinem Laufwerksbuchstaben zugeordnet sind:

```
C:\users\jerry>net use

Status       Lokal            Remote
-------------------------------------------------------------
OK           H:               \\EAGLE\JERRYC
OK                            \\EAGLE\SRC
Der Befehl wurde erfolgreich ausgeführt.
```

Das andere verfügbare Interface für die Einrichtung von Verbindungen zu Netzwerkressourcen sind die Dialogfelder, die der Windows Explorer bietet. Das GUI-Interface bietet meiner Meinung nach nicht so viel Flexibilität wie die Befehlszeilen-Tools, aber Sie können darüber Dauerverbindungen zu Freigaben einrichten. Damit meine ich, dass Windows jedes Mal, wenn Sie sich in das Netzwerk einloggen, versucht, die Verbindung zu den Freigaben wieder herzustellen.

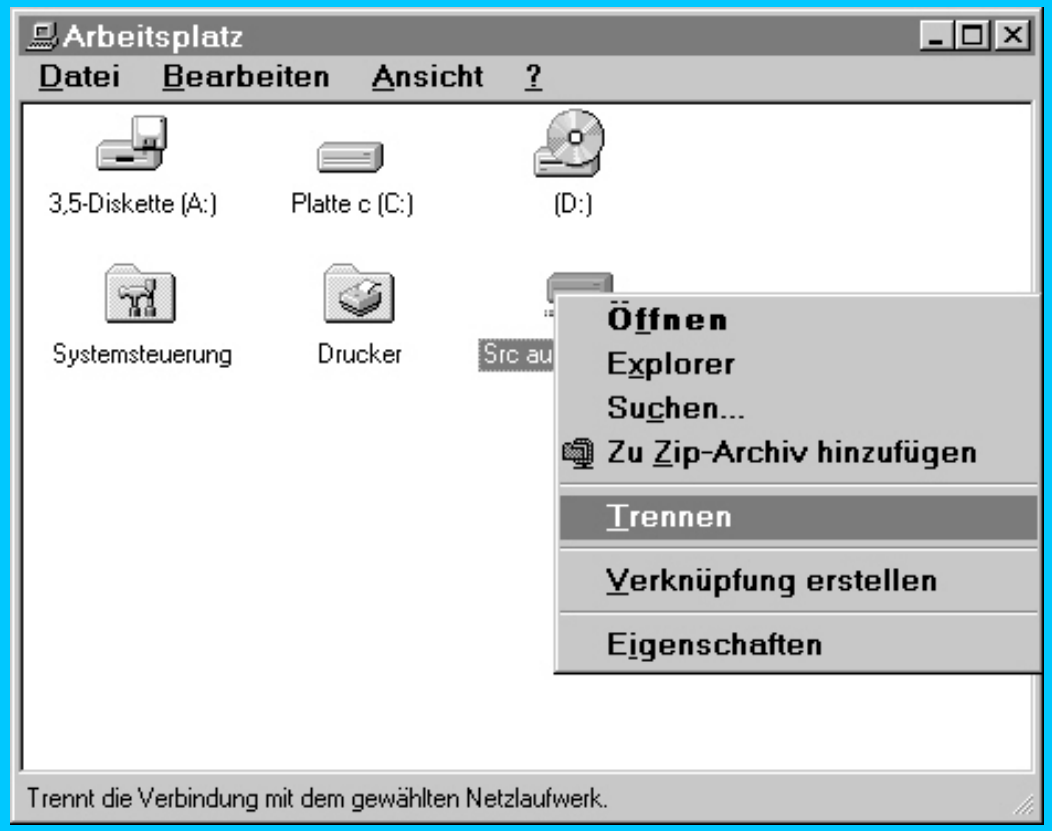
Abbildung 14.15 zeigt das Dialogfeld *Netzlaufwerk verbinden*, das Sie öffnen können, indem Sie entweder auf das Icon *Arbeitsplatz* oder *Netzwerkumgebung* auf Ihrem Desktop rechts klicken und die Option *Netzlaufwerk verbinden* wählen. Das Fenster bietet ein Popup-Menü, aus dem Sie den verfügbaren Laufwerksbuchstaben wählen können, mit dem Sie sich verbinden wollen. Im Feld *Pfad* können Sie den Pfad für die Verbindung manuell definieren, und über die Option *Verbindung beim Start wiederherstellen* können Sie definieren, ob Windows sich an diese Verbindung erinnern sollte, wenn Sie sich beim nächsten Mal einloggen. Beachten Sie, dass Windows Sie über dieses Interface dazu zwingt, einen Laufwerksbuchstaben einzugeben. Wenn Sie sich über das Explorer-Interface mit einer Freigabe verbinden wollen, müssen Sie ihr ein Laufwerk zuweisen. Außerdem gibt es kein Feld, in das Sie ein zu benutzendes Startpasswort eingeben können. Windows verwendet wieder zuerst den Benutzernamen und das Passwort, die Sie beim Anmelden eingegeben haben, und fordert Sie auf, ein neues Passwort einzugeben, falls die Authentifizierung scheitert.

Abb. 14.15: Das Dialogfeld Netzlaufwerk verbinden



Wenn Sie die Verbindung zum Netzwerklaufwerk trennen wollen, klicken Sie auf den entsprechenden Laufwerksbuchstaben im Fenster *Arbeitsplatz*, um die Menüoption *Trennen* aufzurufen (siehe Abbildung 14.16). Die Wahl dieser Option informiert Windows, dass Sie das Netzwerklaufwerk trennen wollen.

Abb. 14.16: Ein Netzwerklaufwerk trennen



Sie können zwischen beiden Interfaces beliebig hin- und herschalten. Laufwerke, mit denen Sie sich über das GUI-Interface verbunden haben, können Sie über den Befehl `net use` trennen und umgekehrt.

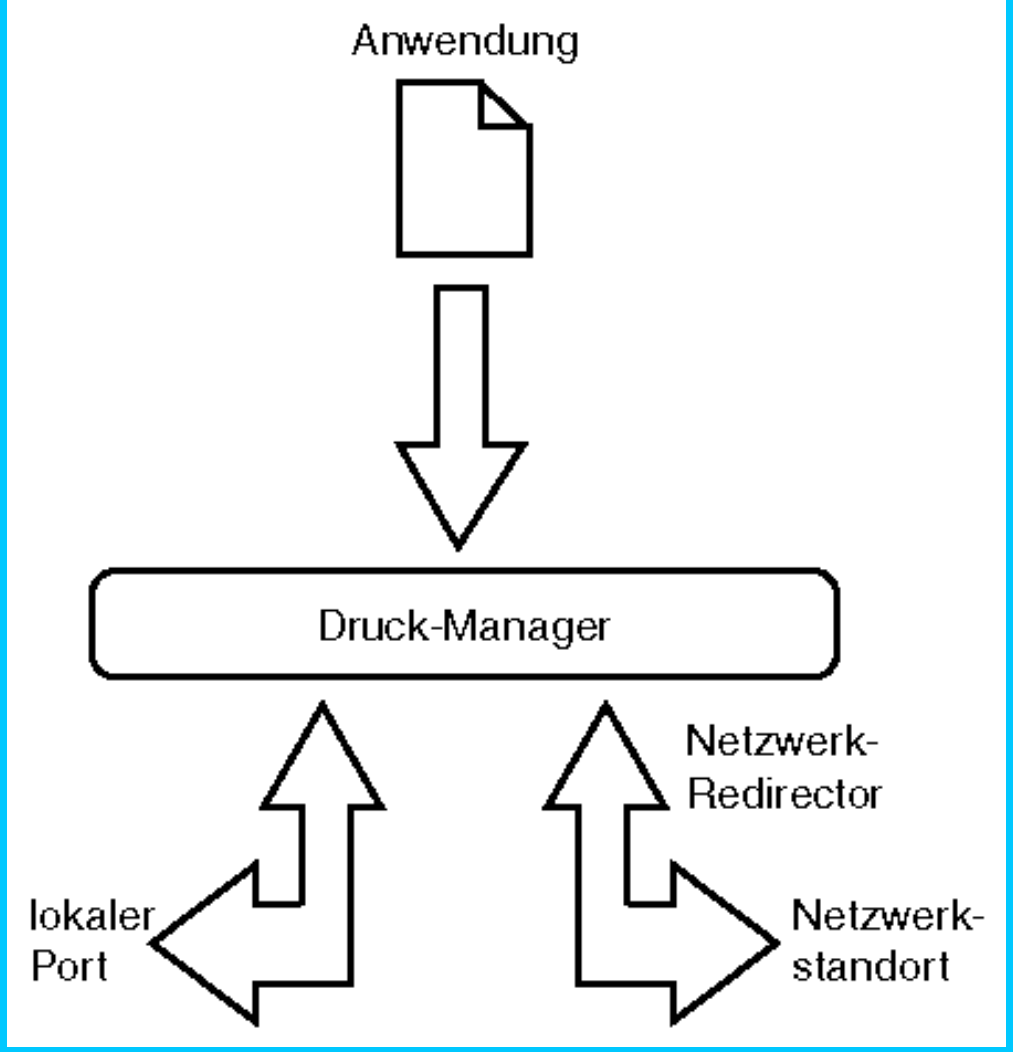
Mit Druckern verbinden

Wie Sie sich mit einem Netzwerkdrucker verbinden, hängt von den Anwendungen ab, die Aufträge an ihn senden müssen. Wenn Sie ältere DOS-Programme haben, die Zugriff auf den Drucker verlangen, besteht die beste Lösung darin, dem Netzwerkdrucker einen verfügbaren LPT-Port zuzuordnen, den entsprechenden Druckertreiber zu installieren und ihn so einzurichten, dass er an den zugeordneten Port druckt. Brauchen nur Windows-Anwendungen Zugriff auf den Drucker, ist es möglich, lediglich den Netzwerkdrucker mit dem UNC-Netzwerkpfad zu verbinden.

Der Unterschied in der Einrichtung liegt in der Tatsache begründet, dass DOS-Programme eher direkt an den LPT-Port drucken (einige sind sogar hartcodiert, nur an LPT1 zu drucken) und die Windows-Druckertreiber übergehen. Wenn Sie einem Netzwerkstandort den LPT-Port zuweisen, werden alle an den Port gesendeten Daten an den Netzwerkdrucker weitergeleitet. Lassen Sie uns einen Blick auf die Details beider Einrichtungen werfen, beginnend mit der, die das Drucken von DOS-Anwendungen unterstützt.

Abbildung 14.17 zeigt den grundlegenden Gedanken hinter dem Drucken mit DOS an einen Netzwerkstandort. Beachten Sie, dass die Abbildung der Erklärung des Windows-Netzwerk-Redirectors in Abbildung 14.1 ähnelt. Dies liegt darin begründet, dass die Logik grundlegend die gleiche ist. Statt direkt an den Port zu drucken, geht die Ausgabe der DOS-Anwendung über den Windows-Redirector, der die Ausgabe entweder an einen lokalen LPT-Port oder den Netzwerkstandort überträgt.

Abb. 14.17: Drucken an einen zugewiesenen LPT-Port



Es gibt zwei Methoden, um einem UNC-Pfad für einen Netzwerkdrucker einen lokalen LPT-Port zuzuweisen. Diese sind fast identisch mit den Methoden für die Zuweisung von Netzwerklaufwerken.

Der LPT-Port, den Sie verwenden, muss nicht wirklich existieren, damit Sie ihn als Anschlusspunkt für den Netzwerkdrucker benutzen können. Sie können ihn unter Windows LPT1 bis LPT9 benutzen, unabhängig davon, ob die Ports auf dem lokalen System präsent sind oder nicht.

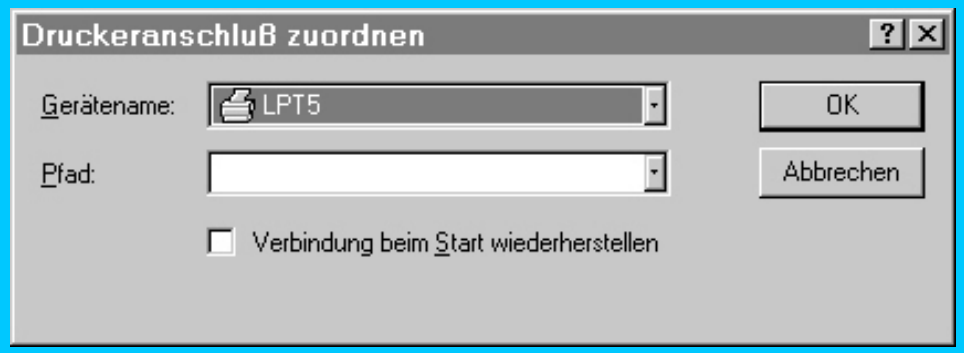
Sie können über den Befehl `net.exe` LPT-Ports ebenso wie Laufwerksverbindungen zuweisen. Die Syntax ist sehr ähnlich:

```
net use LPTn: \\Servername\Druckername
```

Sie sollten *n* durch einen verfügbaren Port zwischen 1 und 9 ersetzen. *Servername* und *Druckername* stehen für den UNC-Pfad zum entfernten Drucker.

Wenn Sie es vorziehen, das grafische Interface zu benutzen, können Sie das Dialogfeld *Druckeranschluß zuordnen* aufrufen, indem Sie auf das Drucker-Icon in einem offenen Arbeitsplatzfenster rechts klicken und die Option *Druckeranschluß zuweisen* aus dem Kontextmenü wählen.

Abb. 14.18: Das Windows-95-Dialogfeld Druckeranschluß zuordnen



Egal welche Methode Sie für die Zuweisung des Ports benutzen, Sie können bei Erfolg die Verbindung sehen, wenn Sie den Befehl `net use` ohne Argumente ausführen, wie Sie es vorher getan haben:

```
C:\users\jerry>net use

Status       Lokaler Name      Remote-Name
-------------------------------------------------------------
OK           M:                \\EAGLE\FILES
OK           LPT5              \\EAGLE\CANONBJC
Der Befehl wurde erfolgreich ausgeführt.
```

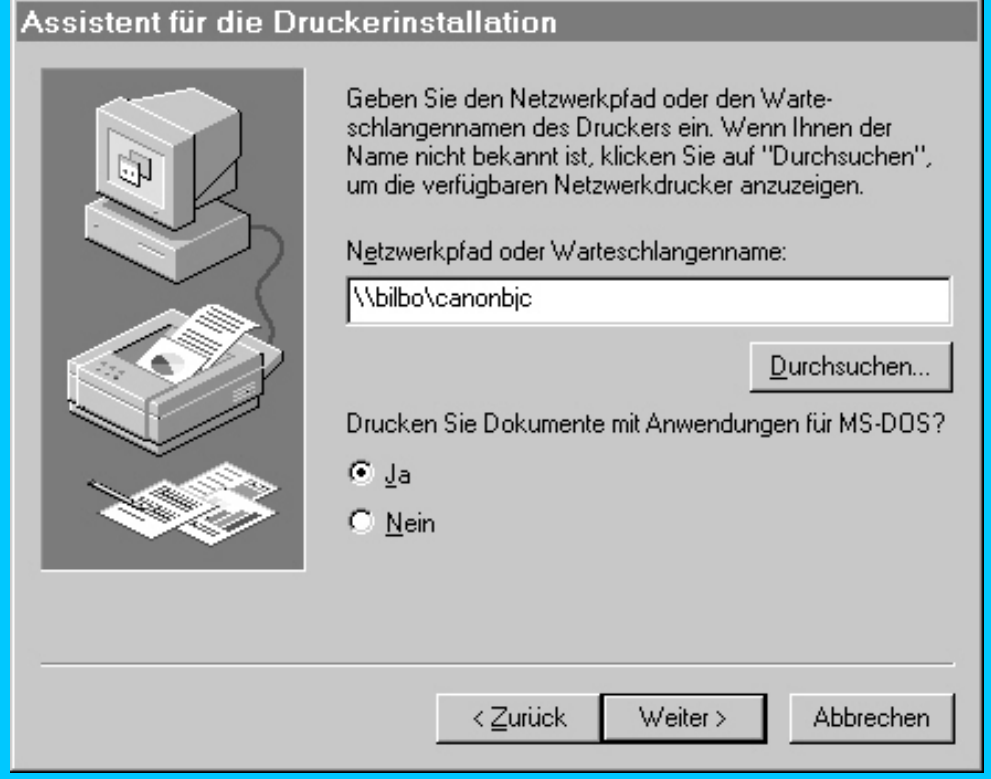
Sie können jetzt den lokalen Druckertreiber einrichten, damit er an den zugewiesenen LPT-Port druckt. Die Einstellung *Anschluß für die Druckausgabe* finden Sie normalerweise auf der Registerkarte *Details* bei den Druckereigenschaften, wie Sie in Abbildung 14.19 sehen können.

Abb. 14.19: Den lokalen Drucker einrichten, damit er an den zugewiesenen LPT-Port druckt

Es gibt eine weitere Möglichkeit, dem Netzwerkdrucker einen lokalen Port zuzuweisen. Wenn Sie den Drucker zum ersten Mal installieren, lässt Windows 95 Sie wählen, ob Sie DOS-Anwendungen ermöglichen wollen, auf diesem Drucker zu drucken. Wenn Sie *Ja* wählen, wird das Dialogfeld *Druckeranschluß zuordnen* geöffnet, und Windows richtet den Drucker, wenn verbunden, ein, so dass er seine Ausgabe an den zugeordneten Druckerport für Sie überträgt. Abbildung 14.20 zeigt das Dialogfeld, in dem Sie gefragt werden, ob Sie DOS-Unterstützung wollen oder nicht.

Wenn alle Programme, die drucken sollen, native Windows-Anwendungen sind, darunter auch Windows-3.1-Software, ist es leichter, wenn Sie sich einfach während der Installation mit dem UNC-Netzwerkpfad des Druckers verbinden. Wenn Sie während der Installation festlegen, dass der Drucker ein Netzwerkdrucker ist, präsentiert Windows Ihnen ein editierbares Feld, in dem Sie entweder nach dem Drucker suchen oder den Netzwerkpfad manuell definieren können (siehe Abbildung 14.20).

Abb. 14.20: Der Windows-95-Assistent für die Druckerinstallation ermöglicht Ihnen die Definition des UNC-Pfadnamens für den Netzwerkdrucker und fragt Sie, ob auch DOS-Programme auf diesen Drucker zugreifen können



Windows NT

Obwohl es das gleiche Benutzer-Interface bietet wie Windows 95 und 98, ist Windows NT ein von Grund auf komplett anderes Betriebssystem. Der Vorteil, das gleiche GUI zu haben, liegt darin, dass viele der Schritte für die Benutzung des Windows-NT-SMB-Clients ähnlich denen für die Benutzung des Windows-9x-Clients sind. Lassen Sie uns mit der Installation der notwendigen Netzwerkkomponenten beginnen.

Den Client konfigurieren

Ein Bereich, in dem sich die internen Unterschiede zwischen Windows NT und Windows 9x nach außen zeigen, ist der Netzwerk-Code. Beide Betriebssysteme benutzen eine Art von Netzwerk-Redirector, wie ich ihn im ersten Teil dieses Kapitels beschrieben habe. Unter Windows NT ist dies der Arbeitsstationsdienst. Die anderen zwei benötigten Komponenten sind die Netzwerkkarte und das TCP/IP-Protokoll, ebenso wie in Windows 9x.

Die Netzwerkkarte installieren

Wieder beginnen Sie mit der Installation der Treiber-Software für die Netzwerkkarte. Im Gegensatz zu Windows 9x verlangt Windows NT, dass Sie den Netzwerkkartentreiber manuell hinzufügen und konfigurieren. Das bedeutet, dass Sie einige genaue Details Ihrer speziellen Karte kennen müssen. Auch hier würde es wieder den Rahmen dieses Buches sprengen, Sie durch alle möglichen Schritte für die Installation jeder Netzwerkkarte zu führen; also werde ich nur einige Punkte erwähnen, die für alle gelten.

Bevor Sie mit der Installation beginnen, stellen Sie sicher, dass Sie die aktuellsten Treiber des Herstellers haben. Ich habe noch nie einen Fall gesehen, in dem

die gewöhnlichen Windows-NT-Treiber besser waren als die, die vom Hersteller der Karte entwickelt wurden. Bei einigen neueren Karten werden Sie vielleicht feststellen, dass nur die Treiber des Herstellers zur Verfügung stehen.

Wenn Sie die Karte in Ihren PC eingebaut haben und das Netzwerkdialogfeld öffnen, werden Sie feststellen, dass die Netzwerkkarten, Protokolle und Dienste alle auf verschiedenen Seiten aufgelistet sind. Denken Sie daran, dass ich sagte, dass dies einer der Bereiche ist, in denen Sie die Unterschiede sehen können.

Gehen Sie zuerst zur Registerkarte *Netzwerkkarte*. Klicken Sie auf *Hinzufügen* und suchen Sie den entsprechenden Treiber für Ihre Karte (siehe Abbildung 14.21). Wenn Sie einen anderen Treiber installieren müssen, werden Sie nach dem Standort der Dateien gefragt, wenn Sie auf die Schaltfläche *Diskette* klicken.

Nachdem Windows NT die notwendigen Treiberdateien von der Windows-NT-Installations-CD-ROM oder der Treiberdiskette, wenn Sie benutzerdefinierte Treiber-Software installieren, kopiert hat (siehe Abbildung 14.22), können Sie zum nächsten Schritt übergehen und das TCP/IP-Protokoll installieren.

Abb. 14.21: Eine neue Netzwerkkarte installieren

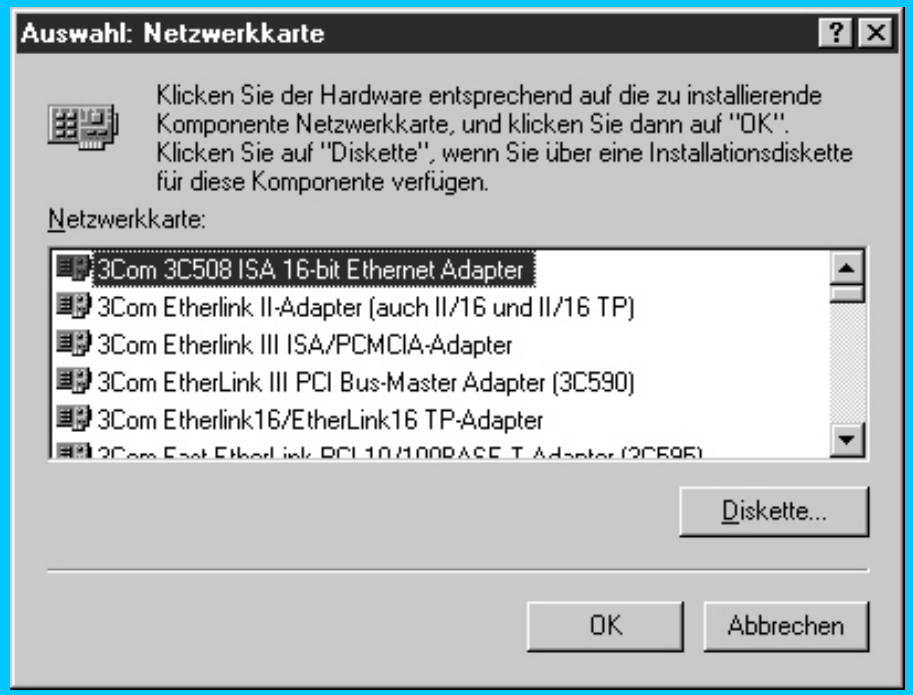
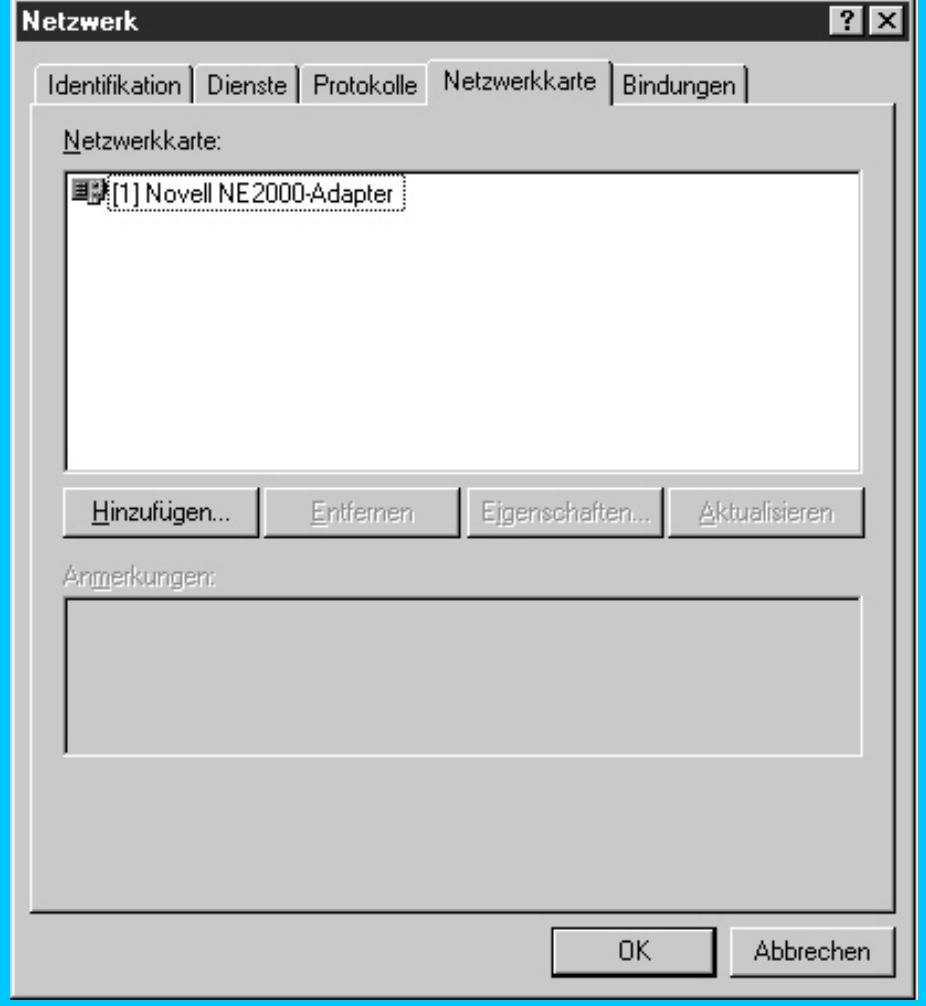


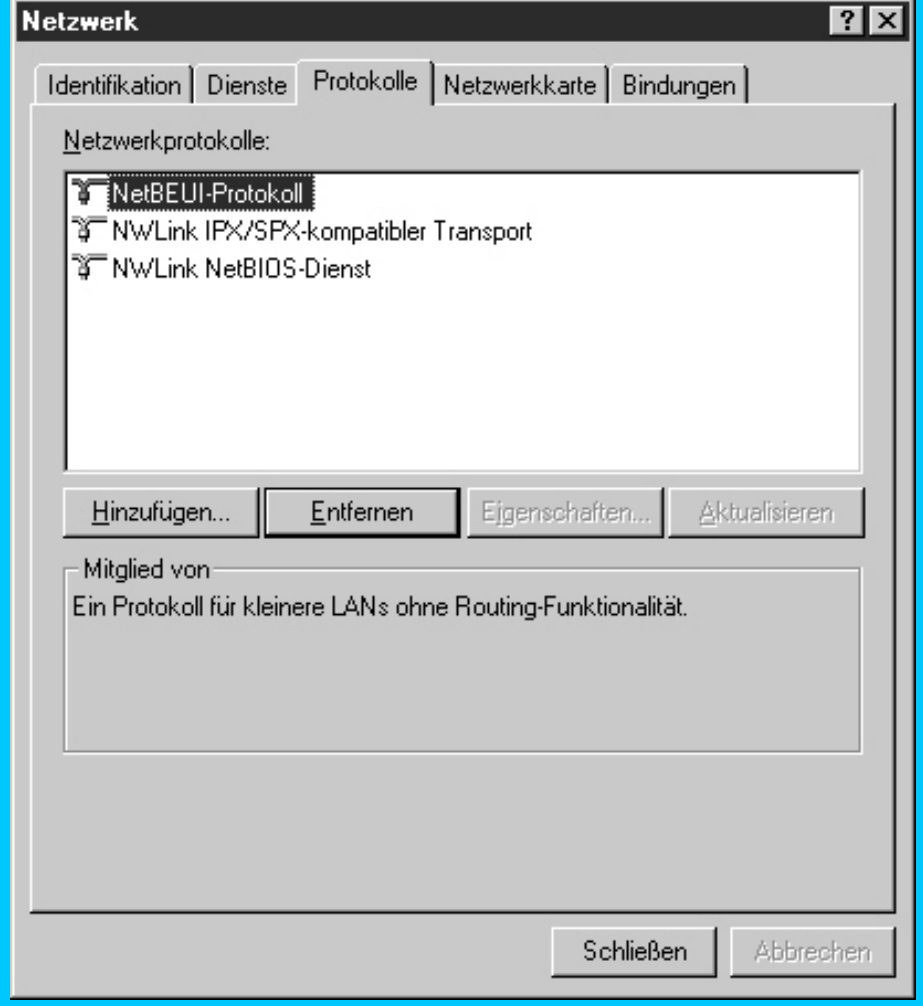
Abb. 14.22: Der neu installierte Netzwerkkartentreiber für eine NE2000- Ethernet-Karte



Das TCP/IP-Protokoll installieren

Windows NT fügt, wie Windows 9x, automatisch zwei Protokolle für Sie hinzu, wenn die erste Netzwerkkarte installiert ist. Abbildung 14.23 zeigt die zwei NWLink-Einträge sowie den Eintrag für das NetBEUI-Protokoll. Der NWLink-NetBIOS-Eintrag hängt vom NWLink-IPX/SPX-Protokoll ab und wird mit ihm installiert, daher beziehe ich mich auf beide als einen Eintrag.

Abb. 14.23: Standard-Netzwerkprotokolle, die von Windows NT installiert werden



Zunächst fügen Sie das TCP/IP-Protokoll hinzu, indem Sie auf die Schaltfläche *Hinzufügen* klicken und den Eintrag für das Protokoll aus der Liste wählen, die Sie in Abbildung 14.24 sehen. Dann entfernen Sie die NWLink-Einträge und das NetBEUI-Protoll aus der installierten Liste, damit nur das TCP/IP-Protokoll bestehen bleibt.

Wenn Sie das IPX/SPX-Protokoll entfernen, entfernt Windows NT automatisch auch den NWLink-NetBIOS-Eintrag (siehe Abbildung 14.25).

Abb. 14.24: Das TCP/IP-Protokoll für die Installation auswählen

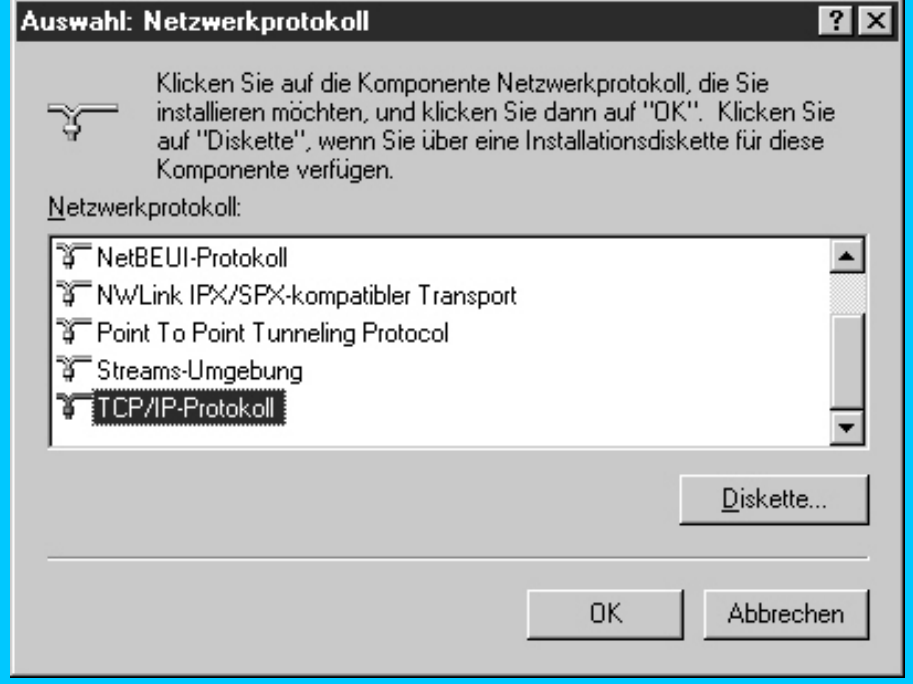
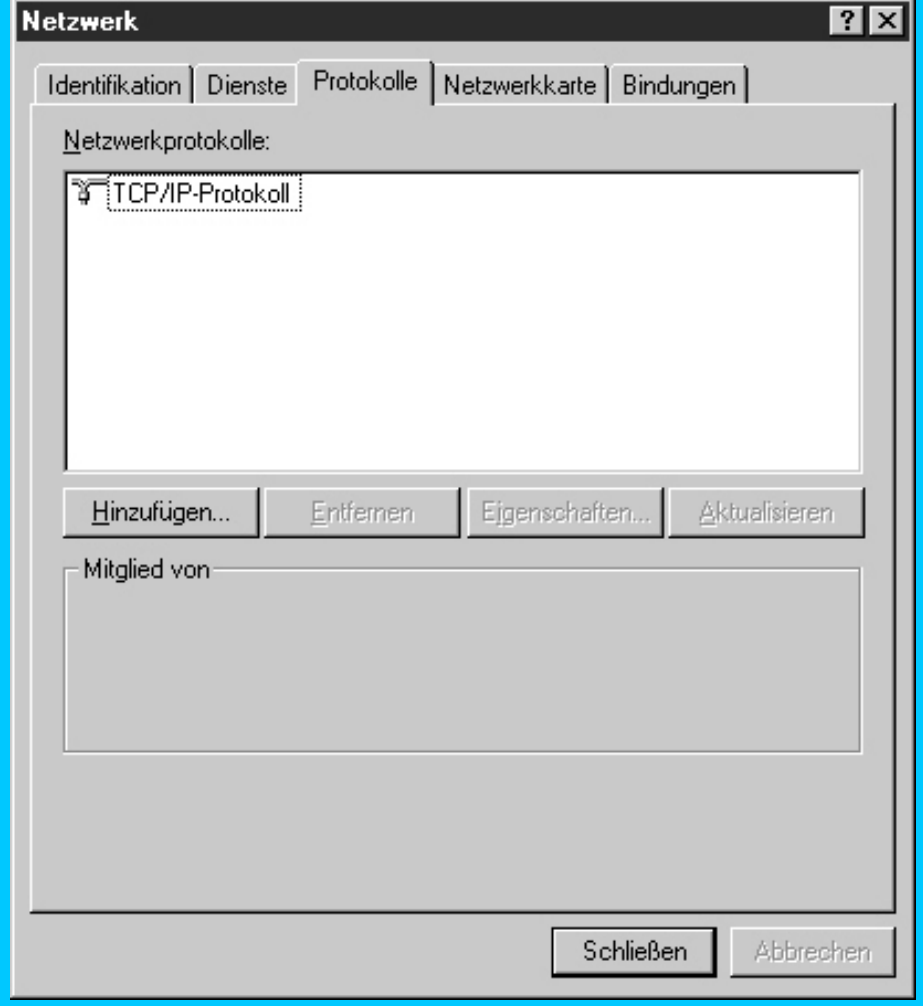
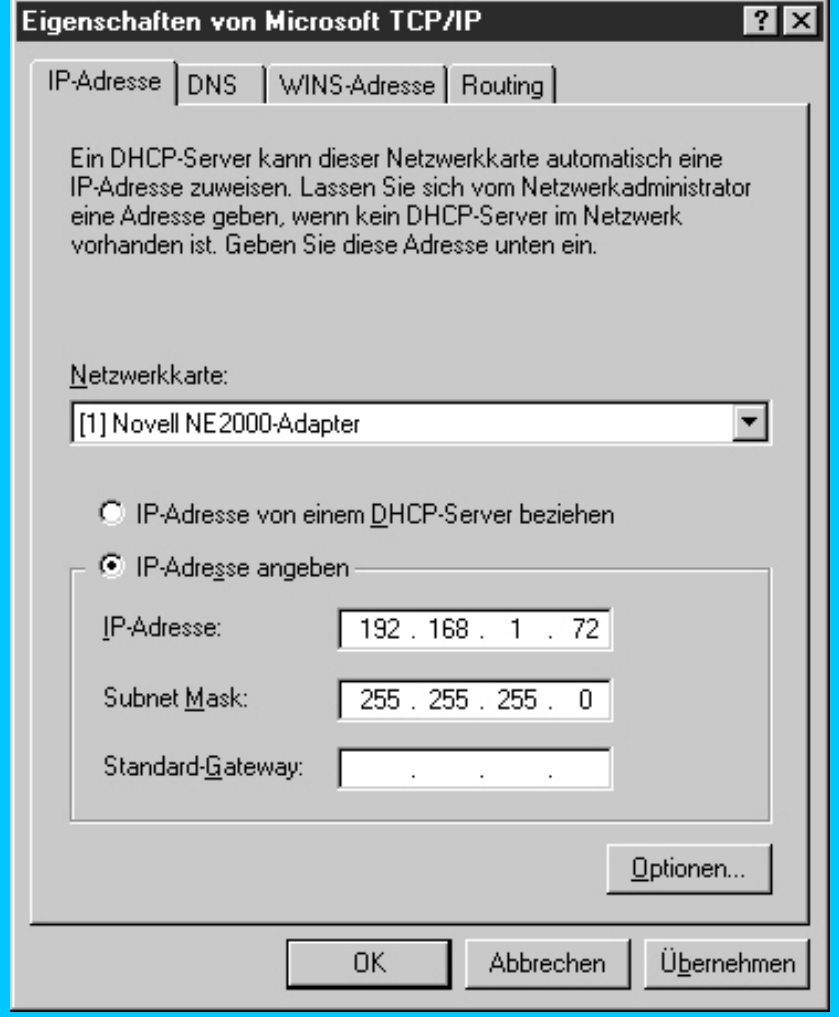


Abb. 14.25: Die installierten Netzwerkprotokolle nach Entfernen der Einträge für das NWLink- und das NetBEUI-Protokoll



Machen Sie sich jetzt keine Gedanken über die Konfiguration der TCP/IP-Einstellungen. Abbildung 14.26 zeigt, was passiert, wenn Sie das Netzwerkdialogfeld schließen. Windows NT überprüft die Netzwerkkarte und die Protokollbindungen und fordert Sie auf, die notwendigen TCP/IP-Einstellungen festzulegen. Die einzigen Informationen, die ich im Moment angeben möchte, sind die IP-Adresse des Rechners und die Subnetzmaske. Wenn Ihr System die Einrichtung eines Standard-Gateways oder die Angabe der IP-Adressen Ihrer DNS-Server verlangt, sollten Sie dies jetzt erledigen.

Abb. 14.26: Das Dialogfeld Eigenschaften für TCP/IP



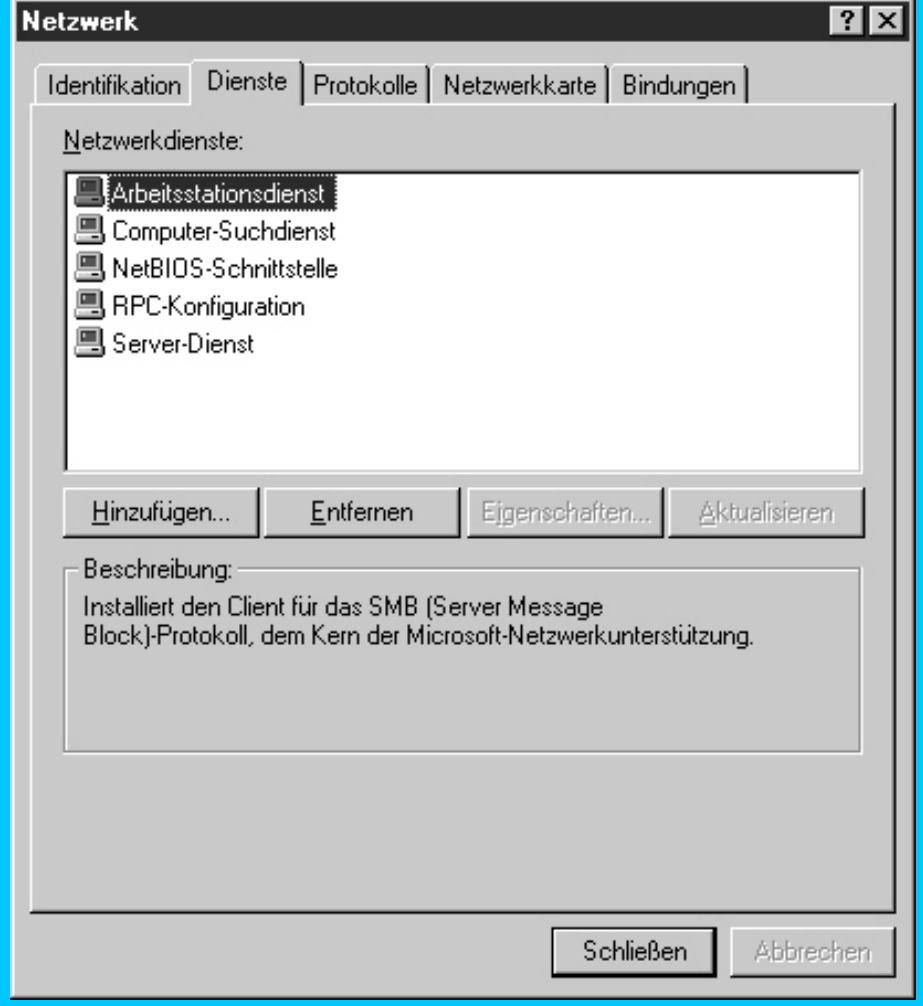
Der Arbeitsstationsdienst

Ich habe am Anfang dieses Abschnitts erwähnt, dass der Arbeitsstationsdienst den Windows-NT-SMB-Redirector implementiert. Der ergänzende Dienst, der einem Windows-NT-Rechner ermöglicht, Dateien freizugeben, ist der Server-Dienst.

Der Arbeitsstationsdienst verlangt keine Konfiguration und wird standardmäßig installiert, wenn Sie Netzwerke unter Windows NT einrichten. Ich denke, Microsoft hat angenommen, dass auch Windows-NT-Rechner in irgendeiner Art von Microsoft-Netzwerkumgebung laufen würden. Denken Sie daran, dass Microsoft das SMB-Protokoll für die Implementierung von Datei- und Druckerfreigaben in seinem Netzwerkmodell gewählt hat. Das war wahrscheinlich eine relativ sichere Annahme.

Abbildung 14.27 zeigt den Arbeitsstations- und andere Dienste, die standardmäßig installiert werden. Sie akzeptieren einfach die Einträge, die in der Registerkarte *Dienste* eingerichtet wurden.

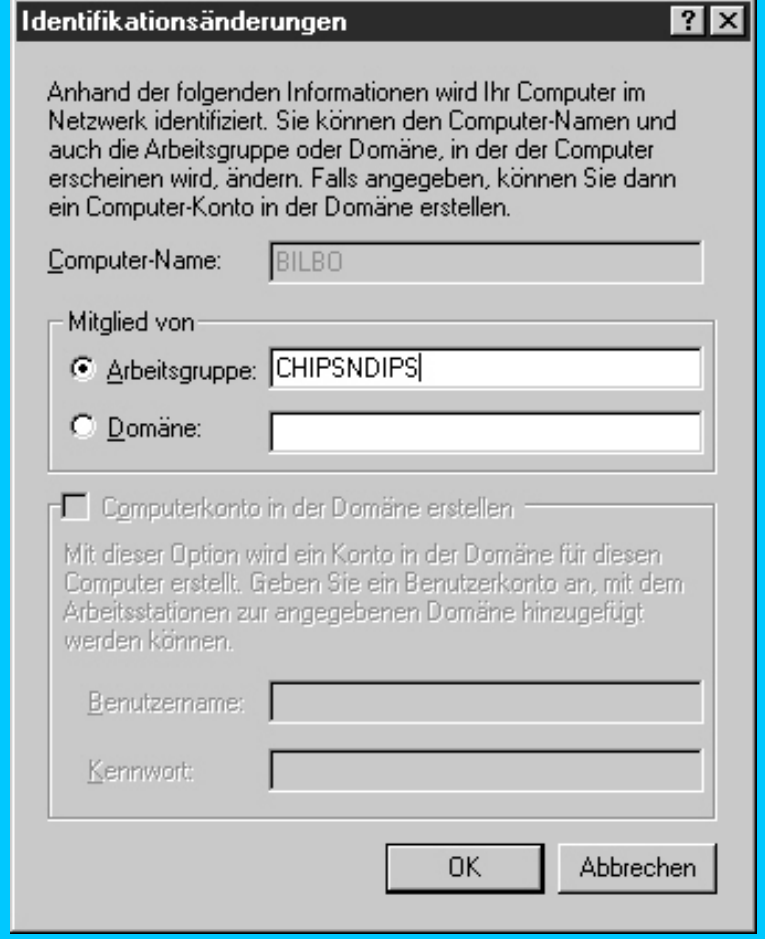
Abb. 14.27: Die Standard-Netzwerkdienste einschließlich des Arbeitsstationsdienstes, die von Windows NT installiert werden



Den NetBIOS-Rechnernamen und den Arbeitsgruppennamen einrichten

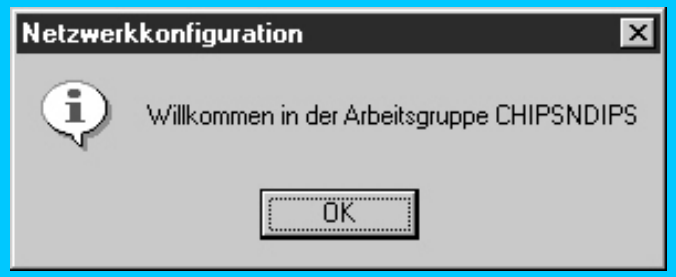
Das Letzte, was Sie noch tun müssen, bevor Sie die von Ihnen durchgeführten Änderungen durch Schließen des Netzwerkdialogfelds akzeptieren, ist die Einrichtung des NetBIOS-Rechnernamens und des Arbeitsgruppennamens. Der Rechnername wird während der Installation des Betriebssystems definiert, der Name der Arbeitsgruppe nicht. Die erste Registerkarte im Netzwerkdialogfeld ist die Registerkarte *Identifikation*. Wenn Sie auf die Schaltfläche *Ändern* klicken, können Sie die Werte ändern, wie Sie in Abbildung 14.28 sehen. Für meine Zwecke hier mache ich den Client zu einem Mitglied einer Arbeitsgruppe. Die Arbeitsgruppe, die ich in meinem privaten Netzwerk benutze, nenne ich `CHIPSNDIPS`, da die meisten der Rechner nach irgendeinem mexikanischen Dip benannt sind, wie z.B. `QUESO`, `SALSA` und `PICANTE`.

Abb. 14.28: Den Namen der Arbeitsgruppe im Netzwerkdialogfeld einrichten



Wenn Sie auf die Schaltfläche *OK* klicken, um die Änderung der Arbeitsgruppe anzunehmen, werden Sie ein Fenster sehen, in dem es heißt: »Willkommen in der Arbeitsgruppe CHIPSNDIPS« (siehe Abbildung 14.29). Natürlich heißt Ihre Meldung Sie in der Arbeitsgruppe willkommen, die Sie auf Ihrem Rechner eingegeben haben. Jetzt können Sie das Netzwerkdialogfeld schließen und Ihren Rechner neu starten.

Abb. 14.29: Willkommen in der Arbeitsgruppe CHIPSNDIPS



Zeit für einen Neustart

Nachdem Sie alle notwendigen TCP/IP-Einstellungen definiert, den Namen der Arbeitsgruppe eingerichtet und das Netzwerkdialogfeld geschlossen haben,

fordert Windows NT Sie dazu auf, den Rechner neu zu starten. Akzeptieren Sie den Neustart, holen Sie sich eine Tasse Kaffee und harren Sie der Dinge, die da kommen.

Nach dem Neustart sollten Sie sich mit dem Account, der für Sie eingerichtet wurde, einloggen. Eines der besten Zitate, die ich jemals über Windows 95 gehört habe, war, dass »es Ihnen all die Sicherheit gibt, die Sie verdienen!« Im Grunde genommen gibt es keine echte Sicherheit. Gewiss können manche Dinge Ihnen das Gefühl von Sicherheit vermitteln, aber nichts würde einen Gymnasiasten mit einigen Minuten Zeit aus dem System heraushalten.

Windows NT wurde mit dem Gedanken an Sicherheit entwickelt. Es gibt viele Mechanismen, die Ihnen dabei helfen, den Zugriff auf lokale Ressourcen auf einem Rechner zu kontrollieren. Einer dieser Mechanismen ist die Voraussetzung, dass sich alle Benutzer über irgendeinen Account anmelden müssen, entweder über einen lokalen Account oder einen Domänen-Account, bevor sie auf einen brauchbaren Desktop zugreifen können.

Nachdem Sie sich nun eingeloggt haben und alles installiert ist, lassen Sie uns ansehen, wie Sie sich mit entfernten Freigaben verbinden.

Service Pack 3 für Windows NT 4.0

Bevor Sie versuchen, sich mit dem Samba-Server zu verbinden, sollte ich noch erwähnen, dass Microsoft die gleiche Entscheidung für Windows NT und verschlüsselte Passwörter getroffen hat wie für Windows 95. Sie können noch einmal zum Abschnitt »Verschlüsselte und Klartextpasswörter« zurückblättern, in dem es um das Netzwerk-Redirector-Update für Windows 95 geht, das den Client davon abhält, die Klartextform des Benutzerpassworts über das Netzwerk zu übertragen, wenn der SMB-Server verschlüsselte Passwörter nicht unterstützt.

Bei Service Pack 3 für Windows taucht das gleiche Problem mit nicht verschlüsselten SMB-Servern auf. Wenn Sie versuchen, sich von einem Windows-NT-Client (Server oder Workstation), auf dem das Service Pack 3 oder höher installiert ist, mit einem nicht verschlüsselten SMB-Server zu verbinden, gibt der Netzwerk-Redirector folgende Meldung aus:

```
Computer ist nicht verfügbar. Mit diesem Konto kann man sich nicht von dieser Station aus anmelden.
```

Die gleichen zwei möglichen Lösungen für das Problem mit dem Windows-95-Redirector-Update existieren auch für das Service Pack 3:

- Richten Sie den Samba-Server mit Hilfe der in Kapitel 6 beschriebenen Schritte für die Benutzung verschlüsselter Passwörter ein.
- Aktivieren Sie die Benutzung von Klartextpasswörtern für den Windows-NT-Client.

Wenn Sie die zweite Lösung wählen, müssen Sie folgenden Registrierungsschlüssel hinzufügen und den NT-Client neu booten:

```
[HKLM\System\CurrentControlSet\Services\Rdr\Parameters]
"EnablePlainTextPassword"=dword:00000001
```

Das Bearbeiten der Registry kann Ihr System unbrauchbar machen. Gehen Sie mit extremer Vorsicht vor, wenn Sie den Registrierungseditor (`regedit.exe` oder `regedt32.exe`) benutzen.

Mit Freigaben verbinden

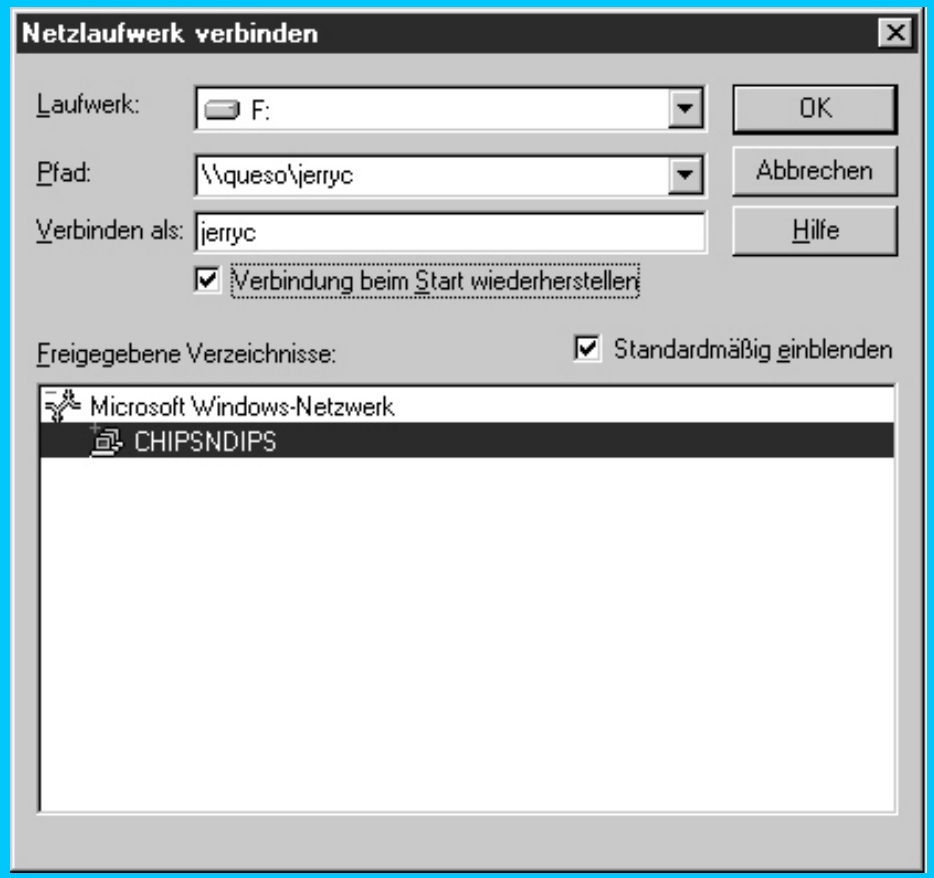
Ich werde in diesem Abschnitt nicht ganz so wortreich sein, wie ich in dem entsprechenden Abschnitt über die Verbindung zu Freigaben von einem Windows-95-Client war, da es in diesem Bereich eine Menge Überschneidungen gibt. Stattdessen werde ich nur den wichtigsten Unterschied darstellen, den Sie sehen werden.

Wie Windows 9x unterstützt auch Windows NT ein Befehlszeilen- und ein grafisches Interface für die Verbindung zu entfernten SMB-Dateifreigaben. Es gibt einen wichtigen Unterschied in den Methoden im Vergleich zu Windows 9x. Unter Windows NT können Sie sich unter einem anderen Benutzernamen mit dem Server verbinden als dem, der für die Anmeldung beim lokalen Rechner benutzt wird. Wenn Sie den Befehl `net use` ausführen, um ein Laufwerk zu mounten, können Sie einen zusätzlichen Parameter für den Benutzernamen spezifizieren, der für die Verbindung verwendet wird. Wenn ich mich in die PC-Konsole mit dem Benutzernamen `jcarter` einlogge, kann ich Windows NT mitteilen, für den Sitzungsaufbau den Benutzernamen `jerryc` zu verwenden, indem ich Folgendes eingebe:

```
net use h: \\bilbo\jerryc /user:jerryc
```

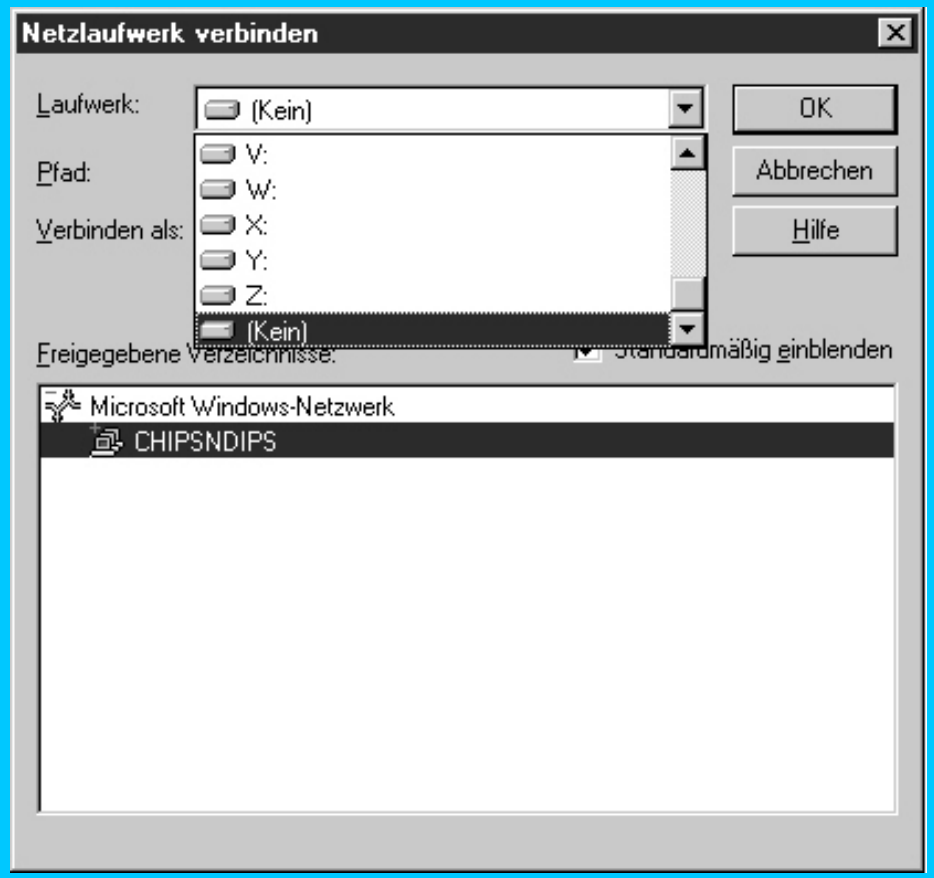
Sie werden außerdem in Abbildung 14.30 bemerken, dass das Dialogfeld *Netzlaufwerk verbinden* im Vergleich zu Windows 95 ein zusätzliches Feld hat. Das Feld *Verbinden als* ist die grafische Entsprechung des Parameters `/user:` für den Befehl `net use`.

Abb. 14.30: Die Windows-NT-Version des Dialogfelds Netzlaufwerk verbinden



Das Windows-NT-GUI für die Verbindung unterstützt außerdem die Verbindung zu einem UNC-Pfad, indem Sie einen Laufwerksbuchstaben spezifizieren, dem die Verbindung zugeordnet wird. Um dies zu tun, wählen Sie einfach *(kein)* aus dem Popup-Menü für das Laufwerk (siehe Abbildung 14.31).

Abb. 14.31: Verbindung zu einer Netzwerkdateifreigabe ohne Verwendung eines Laufwerksbuchstabens



Zwar können Sie sowohl unter Windows 9x als auch unter Windows NT Dauerverbindungen erstellen, die das Betriebssystem beim Einloggen in den PC versucht wiederherzustellen, aber nur unter Windows NT können Sie dies über den Befehl `net use` tun. Sie können das Standardverhalten, ob Windows NT sich an Verbindungen erinnern soll oder nicht, global einrichten, indem Sie Folgendes eingeben:

```
net use /persistent:Option
```

`Option` ist entweder `yes` oder `no`. Sie können das Standardverhalten außer Kraft setzen, indem Sie dem tatsächlichen Befehl den Parameter `/persistent:`*`Option`* hinzufügen, um die Ressource zuzuordnen:

```
net use h: \\eagle\jerryc /persistent:yes
```

Der NT-SMB-Redirector hat einen kleinen Nachteil: Sie können sich nicht über mehrere Benutzernamen mit dem gleichen Servernamen verbinden. Nehmen wir z.B. an, ich habe einen SMB-Server namens `SERVER1`, der entweder Windows NT, Windows 9x oder Samba ist und die folgenden Freigaben bietet: `[docs]` und `[forms]`. Zunächst verbinde ich mich mit `\\server1\docs` über den gültigen Account mit dem Namen `sysadmin`:

```
net use p: \\server1\docs /user:admin
```

```
Der Befehl wurde erfolgreich ausgeführt.
```

Dann versuche ich, mich über meinen normalen Account `jerryc` mit `\\server1\forms` zu verbinden:

```
net use p: \\server1\forms /user:jerryc
Systemfehler 1219 ist aufgetreten.

Mit diesem Konto kann man sich nicht von dieser Station aus anmelden.
```

Sie können dieses Problem umgehen, indem Sie NT denken lassen, dass Sie sich mit einem anderen Server verbinden. Die einfachste Methode, dies zu tun, besteht darin, die IP-Adresse des Servers statt seines Namens zu verwenden. Um mich z.B. mit der Freigabe `[forms]` zu verbinden, gebe ich folgenden Befehl aus:

```
net use p: \\192.168.1.75\forms /user:jerry
Der Befehl wurde erfolgreich ausgeführt.
```

Diesmal kann Windows NT sich erfolgreich verbinden. Sie können auch die Samba-Option `netbios aliases` verwenden, um Clients zu ermöglichen, sich unter mehreren Namen mit dem gleichen Server zu verbinden statt die IP-Adresse zu benutzen.

Mit Druckern verbinden

Die Verbindung zu einem Netzwerkdrucker funktioniert unter Windows NT, aus praktischen Gründen, nicht anders als unter Windows 9x. Da wahrscheinlich kaum jemand DOS-Anwendungen laufen lässt, die unter Windows NT drucken müssen, werde ich nur die direkte Verbindung zum Netzwerkdrucker-Server darstellen. Wenn Sie von DOS-Anwendungen drucken müssen, können Sie über die gleiche Prozedur wie für Windows 9x einen LPT-Port zuordnen. Abbildung 14.32 zeigt den ersten Schritt für die Installation eines Netzwerkdruckers auf einem NT-4.0-Client. Nachdem Sie festgelegt haben, dass die Verbindung für einen Netzwerkdrucker-Server und nicht für den lokalen Computer gilt, fragt das Betriebssystem Sie nach dem Netzwerkpfad, ebenso wie Windows 9x dies tut. Sie können den Netzwerkpfad entweder manuell einrichten oder ihn im unteren Teil des Fensters *Mit Drucker verbinden* suchen.

Abb. 14.32: Mit einem Drucker auf einem Netzwerk-Server verbinden

Wenn Sie sich mit einem Samba-Server oder einem anderen Server verbinden, der die Druckertreiber nicht zum Herunterladen verfügbar hat, werden Sie aufgefordert, den Hersteller und das Modell des Druckers anzugeben, damit Windows NT die Treiberdateien vom Installationsmedium kopieren kann.

Zusammenfassung

Die Konfiguration eines Windows-9x- oder Windows-NT-Clients für die Verbindung zu einem entfernten SMB-Server, wie z.B. Samba, in einem TCP/IP-Netzwerk verlangt drei Komponenten:

- Eine funktionierende Netzwerkkarte
- Eine korrekt konfigurierte Version des TCP/IP-Protokolls
- Irgendeine Art von SMB-Netzwerk-Redirector. Windows 9x nennt diesen den Client für Microsoft-Netzwerke, während Windows NT diese Funktion im Arbeitsstationsdienst implementiert.

Obwohl beide Betriebssysteme sehr ähnlich erscheinen, gibt es einen bedeutenden Unterschied, der für den Endbenutzer sichtbar ist. Unter Windows 9x können Sie sich nicht unter einem anderen Benutzernamen mit einem Server verbinden als dem, der für das ursprüngliche Login benutzt wurde, während Windows NT diese Funktionalität unterstützt.

Frage & Antwort

F. Kann ich sowohl das IPX/SPX-Netzwerkprotokoll als auch TCP/IP auf meinem Computer installieren?

. Ja, aber einige Postings in den Samba-Mailing-Listen weisen darauf hin, dass dies zu Problemen in Hinsicht auf Netzwerk-Browsing führen kann. Einige Leute haben darüber berichtet, dass sie keine Server in dem Netzwerk finden können, wenn beide Protokolle installiert sind.

F. Warum beschwert sich Windows NT und gibt die Fehlermeldung 2138 aus, wenn ich versuche, Netzwerklaufwerke zuzuordnen?

. Der Arbeitsstationsdienst muss gestartet werden, damit Sie auf das Netzwerk zugreifen können. Sie können den Dienst manuell starten, indem Sie in einem Befehlsprompt-Fenster `net start workstation` eingeben.

Woche 3: Andere SMB-Clients

Tag 15: Andere SMB-Clients

Wie Sie bereits in Kapitel 13, »Unix (smbclient, smbfs, smbwrapper und andere Utilities)«, gesehen haben, stellen Windows-Clients nicht das einzige Betriebssystem dar, das sich mit SMB-Servern verbinden kann. In diesem Kapitel werden Sie zwei weitere SMB-Clients kennenlernen: einen für DOS und einen für den Macintosh.

Vielleicht denken Sie, dass seit der ersten Freigabe von Windows 95 und später Windows 98 DOS der Vergangenheit angehört, dass es nur noch in kleinen Geschichten von älteren Professoren erwähnt wird, die von ihrer Abschlussarbeit auf einem Intel 8086-Rechner erzählen, auf dem das Disk Operating System Version 1.0 lief. Welcher intelligente Mensch sollte schließlich ein Betriebssystem benutzen wollen, das nur eine Sache auf einmal tun kann und eine Speicherbegrenzung von 640 Kbyte hat?

Um die Wahrheit zu sagen, ich benutze DOS immer noch relativ häufig. DOS passt bequem auf eine Diskette, die dazu benutzt werden kann, ein System zu booten. Sie können dann damit fortfahren, Dinge zu verwenden, die Windows 9x eher nicht verwendet (wie z.B. Festplatteneditoren).

Ich habe auch festgestellt, dass diese Boot-Disketten sehr hilfreich sind, wenn sie zusammen mit Disk-Imaging-Software wie z.B. Ghost von Symantec benutzt werden. Wenn ich einen SMB-Client auf einer DOS-Diskette benutze, kann ich in das Netzwerk booten, alle notwendigen Laufwerke mounten und den Rechner von einer Image-Datei laden, die sich auf einem Samba-Server befindet. Der gesamte Prozess dauert etwa 15 bis 20 Minuten, bis ich einen funktionierenden Windows-95-Rechner hochgefahren und im Netzwerk habe. Natürlich sind noch einige andere kleine Details beteiligt, die dies möglich machen, wie z.B. die Benutzung von DHCP und identischer Hardware auf allen Laborrechnern. Eine andere nützliche Situation haben Sie, wenn Sie Windows auf einem Rechner installieren müssen, der kein CD-ROM-Laufwerk hat. Es scheint ewig zu dauern, Windows 95 von Disketten zu laden, aber es geht vergleichsweise extrem schnell, wenn Sie eine freigegebene CD-ROM mounten und die Software über das Netzwerk laden. In diesem Kapitel stelle ich dar, wie Sie über Microsofts DOS-Netzwerk-Client Netzwerk-Boot-Disketten erstellen.

Es gibt auch andere SMB-Clients, die Ihnen dabei helfen können, Nicht-Windows-Rechner in Ihre Samba-Infrastruktur zu integrieren. Vielleicht benutzt die Grafikabteilung in Ihrem Unternehmen Macintoshs, aber die großen Dateien, die die Leute dort produzieren, müssen an andere Abteilungen übertragen werden, die PCs benutzen. Sie werden sich einen SMB-Client namens DAVE für Macintosh-Systeme ansehen, über den diese auf Freigaben auf einem Samba-Server zugreifen und dort Dateien speichern können. Dies stellt einen zentralen Distributionspunkt für alle Ihre Benutzer dar.

Microsoft-Netzwerk-Client Version 3.0 für MS-DOS

Dieser Abschnitt erklärt, wie Sie eine bootfähige DOS-Diskette erstellen, über die Sie SMB-Freigaben mounten können. Sie werden über drei grundlegende Schritte unterrichtet:

- Die Client-Software besorgen
- Installation der Netzwerk-Client-Software auf der lokalen Festplatte
- Erstellen der Netzwerk-Boot-Diskette mit den Dateien, die in Schritt 2 installiert wurden

Die Software besorgen

Microsoft verteilt seinen Netzwerk-Client für DOS frei. Wenn Sie eine Kopie der Windows-NT-4.0-Server-CD-ROM haben, können Sie sich die Zeit für das Herunterladen sparen, indem Sie auf das aus zwei Disketten bestehende Set wie folgt zugreifen:

```
X:\clients\msclient\disks
```

X: ist hierbei der Laufwerksbuchstabe für Ihr CD-ROM-Laufwerk.

Haben Sie keine Kopie der CD-ROM, können Sie die Installationsdisketten für den DOS-Client unter `ftp://ftp.microsoft.com/bussys/Clients/MSCLIENT/` herunterladen.

Das Verzeichnis hat zwei Dateien:

```
DSK3-1.EXE
DSK3-2.EXE
```

Laden Sie beide Dateien in ein temporäres Verzeichnis herunter (z.B. `c:\temp`) und extrahieren Sie jede Diskette über folgenden Befehl:

```
C:\> mkdir c:\temp\disk1
C:\> cd c:\temp\disk1
C:\> c:\temp\dsk3-1.exe
C:\> mkdir c:\temp\disk2
C:\> cd c:\temp\disk2
C:\> c:\temp\dsk3-2.exe
```

Egal welche Methode Sie verwenden, um die Netzwerk-Client-Dateien zu erhalten (CD-ROM oder FTP), sollten Sie an diesem Punkt bereit sein, die Dateien auf zwei 1,44-Mbyte-Disketten zu kopieren und den Client zu installieren.

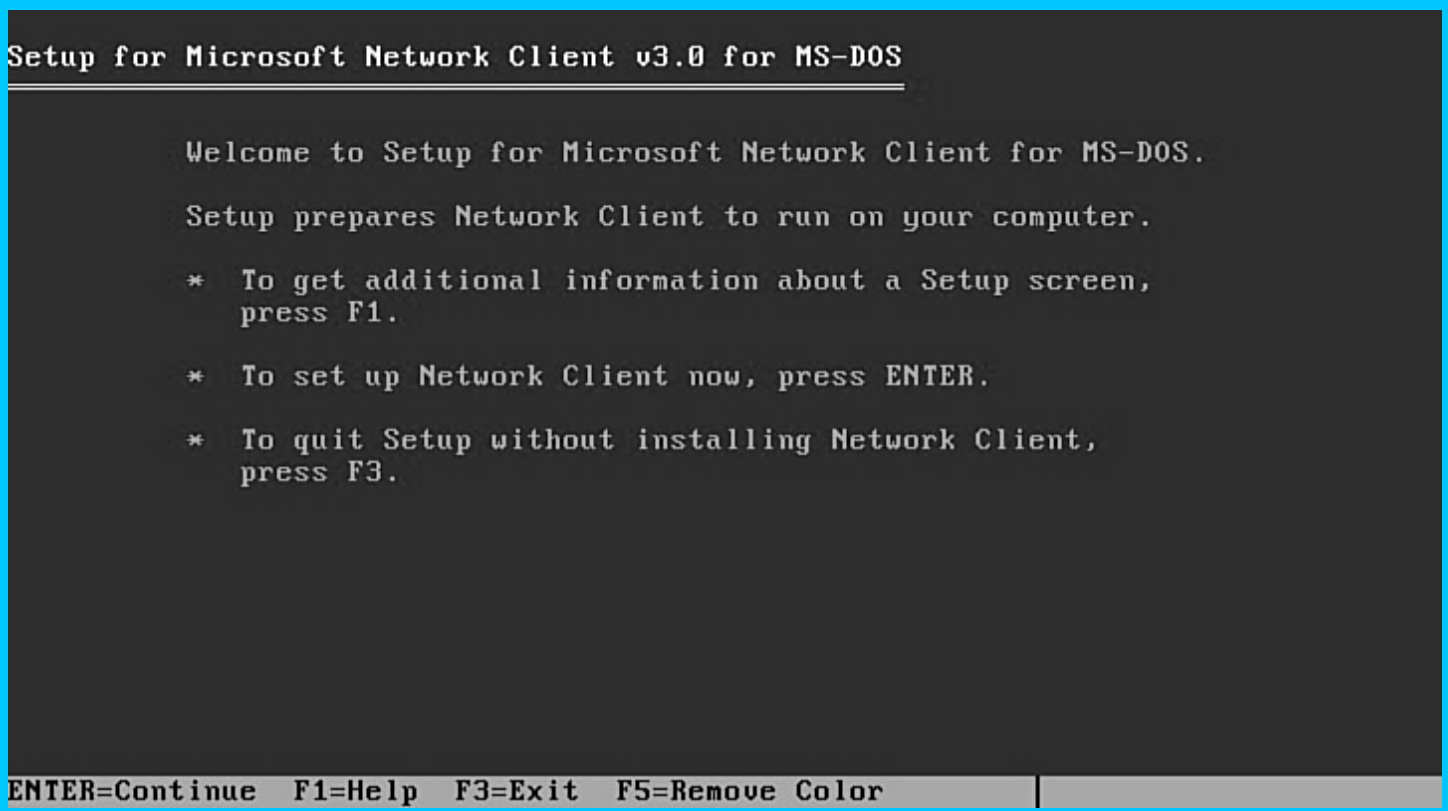
Den Client installieren

Um eine Netzwerk-Boot-Diskette zu erstellen, durchlaufen Sie zunächst die Schritte für die Installation des DOS-Netzwerk-Clients auf eine Festplatte. Kopieren Sie dann nur die benötigten Dateien auf die Boot-Diskette. Die Praxis hat mir gezeigt, dass es einfacher ist, die Dinge unter normalen Bedingungen zum Laufen zu bringen, bevor man versucht, die Boot-Diskette zu erstellen. Ich verwende MS-DOS 6.22 als Betriebssystem, dies nur der Vollständigkeit halber.

Nachdem Sie die Dateien auf zwei Disketten kopiert haben, können Sie die Software installieren. Für dieses Beispiel setze ich voraus, dass der Buchstabe Ihres Diskettenlaufwerks `A:` ist. Legen Sie die erste Diskette in den Client-Rechner ein und geben Sie **C:\>a:setup.exe** ein.

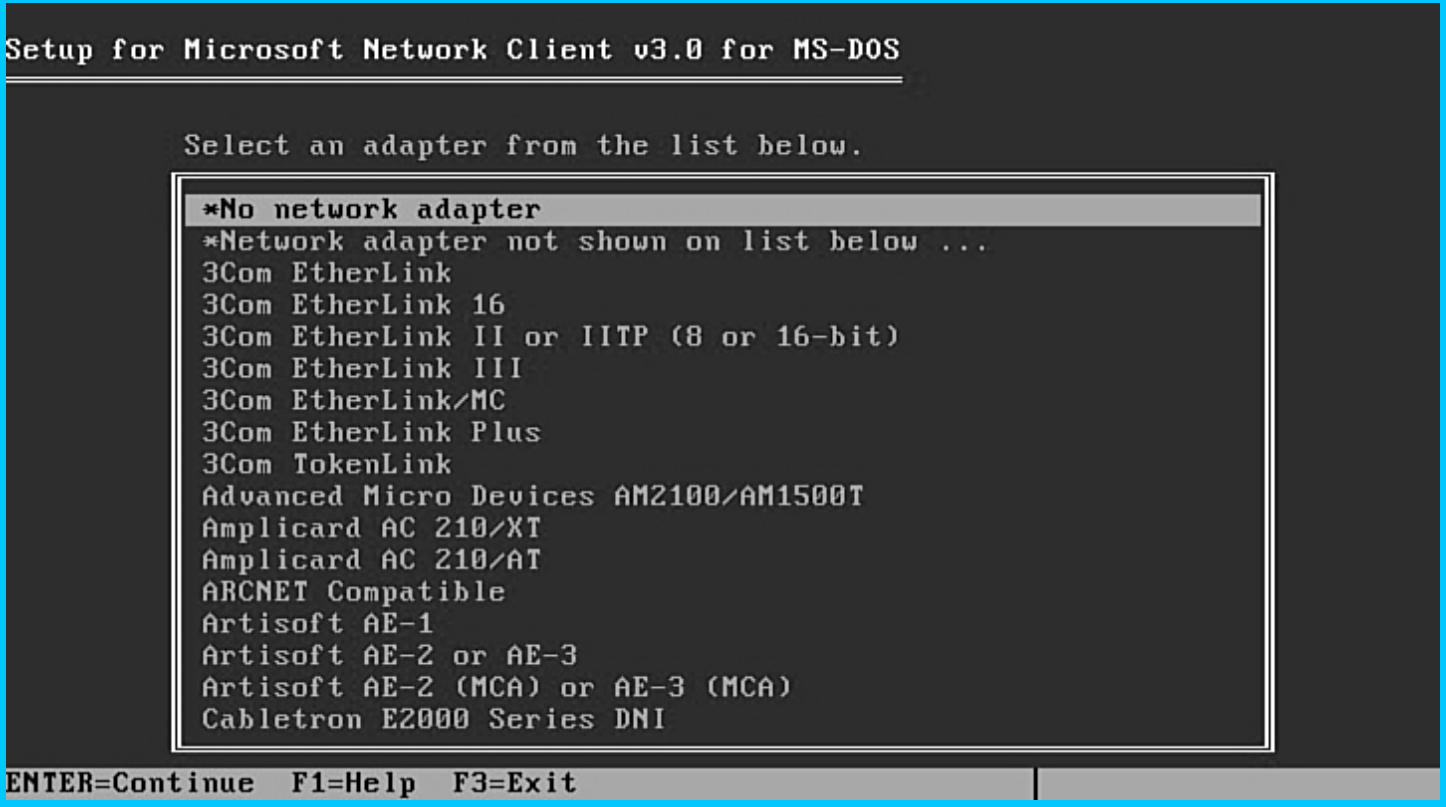
Als Ergebnis sollten Sie den Bildschirm sehen, der in Abbildung 15.1 dargestellt ist.

Abb. 15.1: Installationsbildschirm für Microsofts Netzwerk-Client 3.0



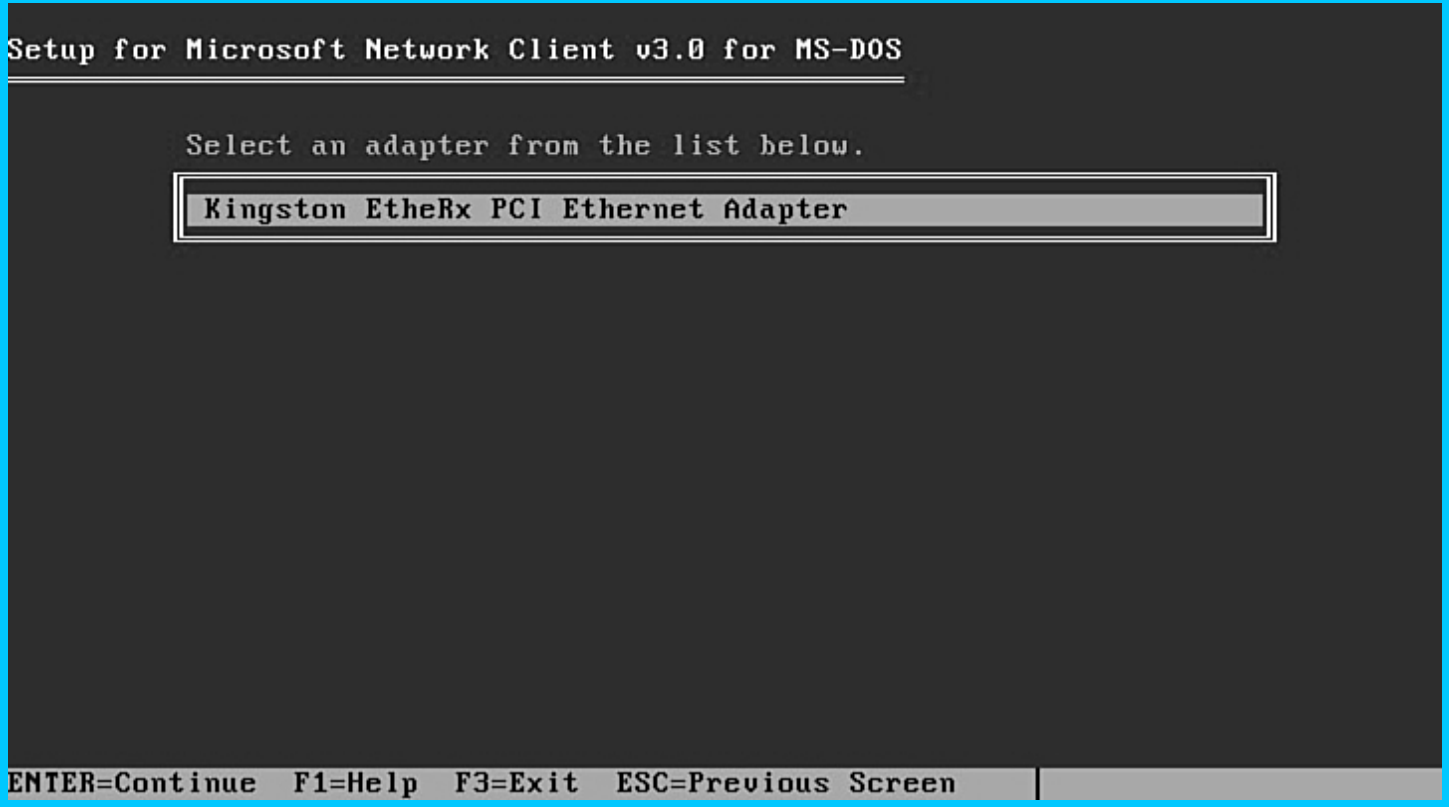
Wenn Sie mit dem nächsten Bildschirm fortfahren, akzeptieren Sie den Standard-Installationsstandort `C:\NET`. Nachdem das Setup-Programm für einige Sekunden das System durchsucht hat, kehrt es mit einer Liste möglicher Treiber für Netzwerkkarten zurück, die Sie installieren können (siehe Abbildung 15.2).

Abb. 15.2: Einen Treiber für den Netzwerkadapter zur Installation auswählen



Dies kann der komplizierteste Teil der Installation sein. Wenn Sie Ihre bestimmte Netzwerkkarte nicht in der Liste sehen - und die Chancen sind hoch, dass sie sie nicht sehen -, suchen Sie den MS LanMan für DOS-Treiber für Ihre Karte. Ich verwende normalerweise die Standard-*NDIS2*-Treiber, die mit der Netzwerkkarte ausgeliefert werden. Das Setup-Programm für den Netzwerk-Cient versucht, eine Datei namens `OEMSETUP.INF` in dem Verzeichnis zu finden, das als Standort für die Kartentreiber designiert ist. Findet das Setup-Programm diese Datei nicht, beschwert es sich, dass es keine zulässigen Treiber gefunden hat. Dies kann etwas mühselig sein. Vielleicht brauchen Sie einige Versuche, aber es sollte möglich sein, Ihre Karte aus einer Liste von Treibern zu wählen, die gefunden wurden. Meine Netzwerkkarte ist in der Liste in Abbildung 15.3 dargestellt.

Abb. 15.3: Einen benutzerdefinierten Treiber für eine Netzwerkkarte zur Installation auswählen



Der nächste Schritt besteht darin, einen Benutzernamen zu spezifizieren, der standardmäßig verwendet wird. Dies dient tatsächlich mehr der Bequemlichkeit, da Sie immer einen anderen angeben können, wenn Sie sich tatsächlich in das Netzwerk einloggen.

Der nächste Bildschirm (siehe Abbildung 15.4) ermöglicht Ihnen die Änderung von Setup-Optionen vor der Installation. Über die Nameneinstellung können Sie den NetBIOS-Rechernamen und den Arbeitsgruppennamen für das Browsing sowie den Domänennamen ändern, der für die Authentifizierung benutzt wird, wenn die Option `Logon to Domain` aktiviert ist. Abbildung 15.5 zeigt die Standardeinstellungen für diese drei Werte. Ändern Sie diese in `QUESO`, `CHIPSNDIPS` bzw. `CHIPSNDIPS`.

Abb. 15.4: Der Bildschirm für die generelle Konfiguration des Netzwerk-Clients

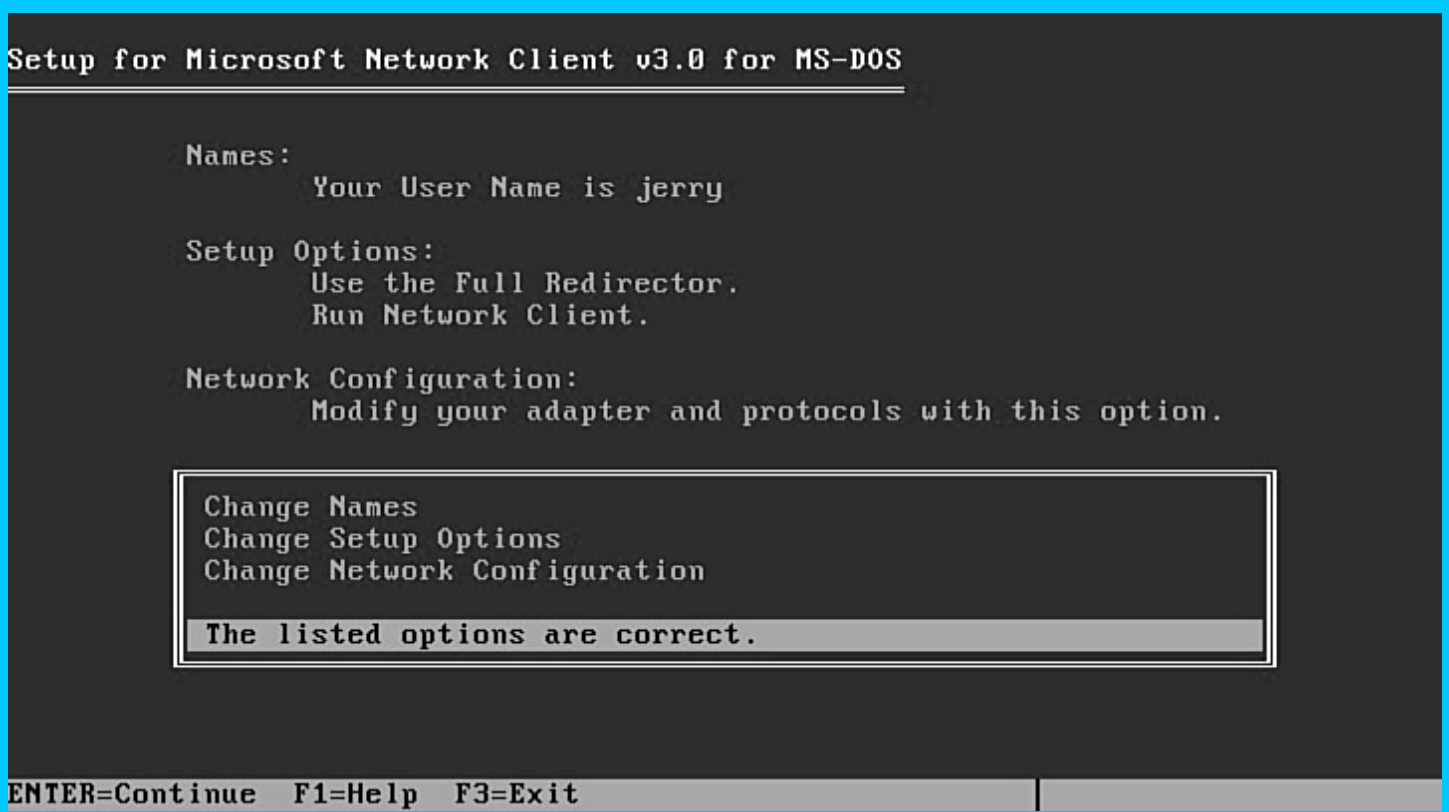
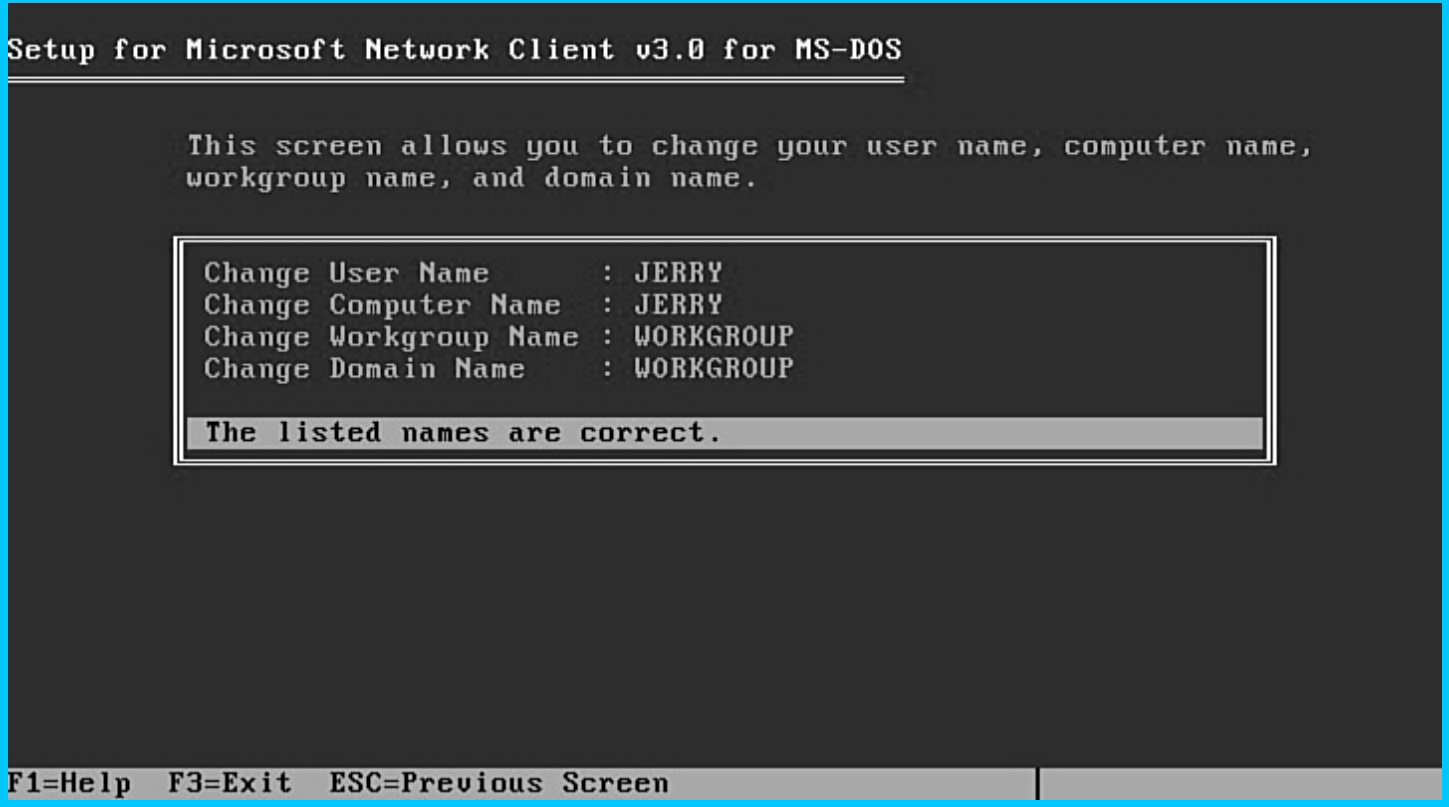


Abb. 15.5: Der Bildschirm für die Einstellung des NetBIOS-Namens



Die Setup-Optionen sollte ich vielleicht etwas näher erklären. Die Einstellung in Abbildung 15.6 zeigt, dass Ihre Konfiguration den Full Redirector verwendet. Sie haben zwei Optionen. Der volle Netzwerk-Redirector ermöglicht Domänen-Logins - und ja, Sie können auch Login-Skripte erhalten - und benutzt etwa 100 Kbyte Speicher. Der Basis-Redirector benutzt wesentlich weniger Speicher, ermöglicht aber nur das Mounten von Druckerports und Netzwerklaufwerken. Der Basis-Redirector verlangt außerdem, dass Sie die Passwortverschlüsselung auf Ihrem Server aktivieren. Der volle Redirector unterstützt Klartext- und verschlüsselte Passwörter. Wählen Sie für den gegenwärtigen Zeitpunkt den vollen Redirector. Wenn Sie später feststellen, dass Sie die Menge des benutzten Speichers reduzieren müssen, können Sie das Setup-Programm erneut starten, um Ihre Einstellungen zu ändern.

Abb. 15.6: Der Bildschirm für die detaillierte Konfiguration des Netzwerk-Clients

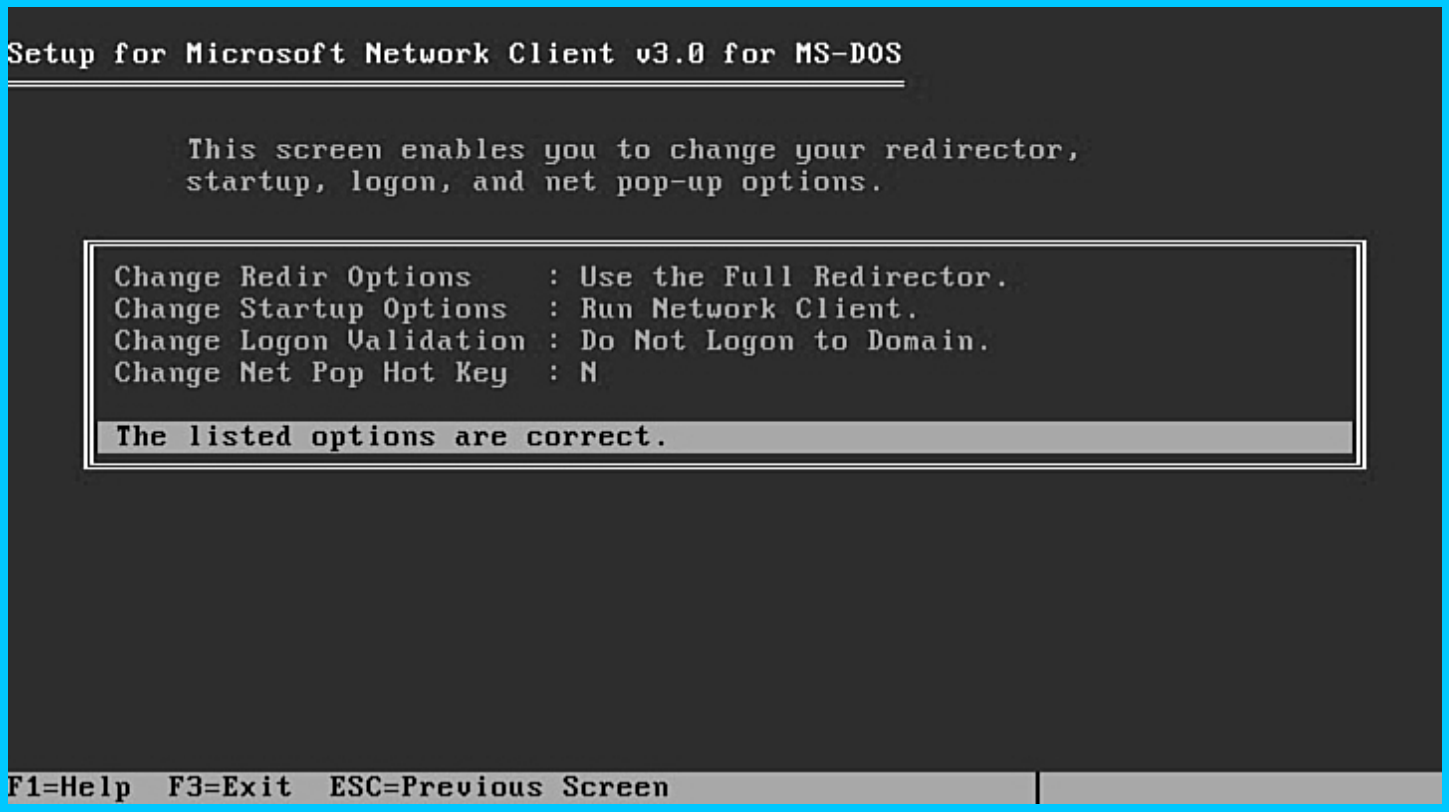
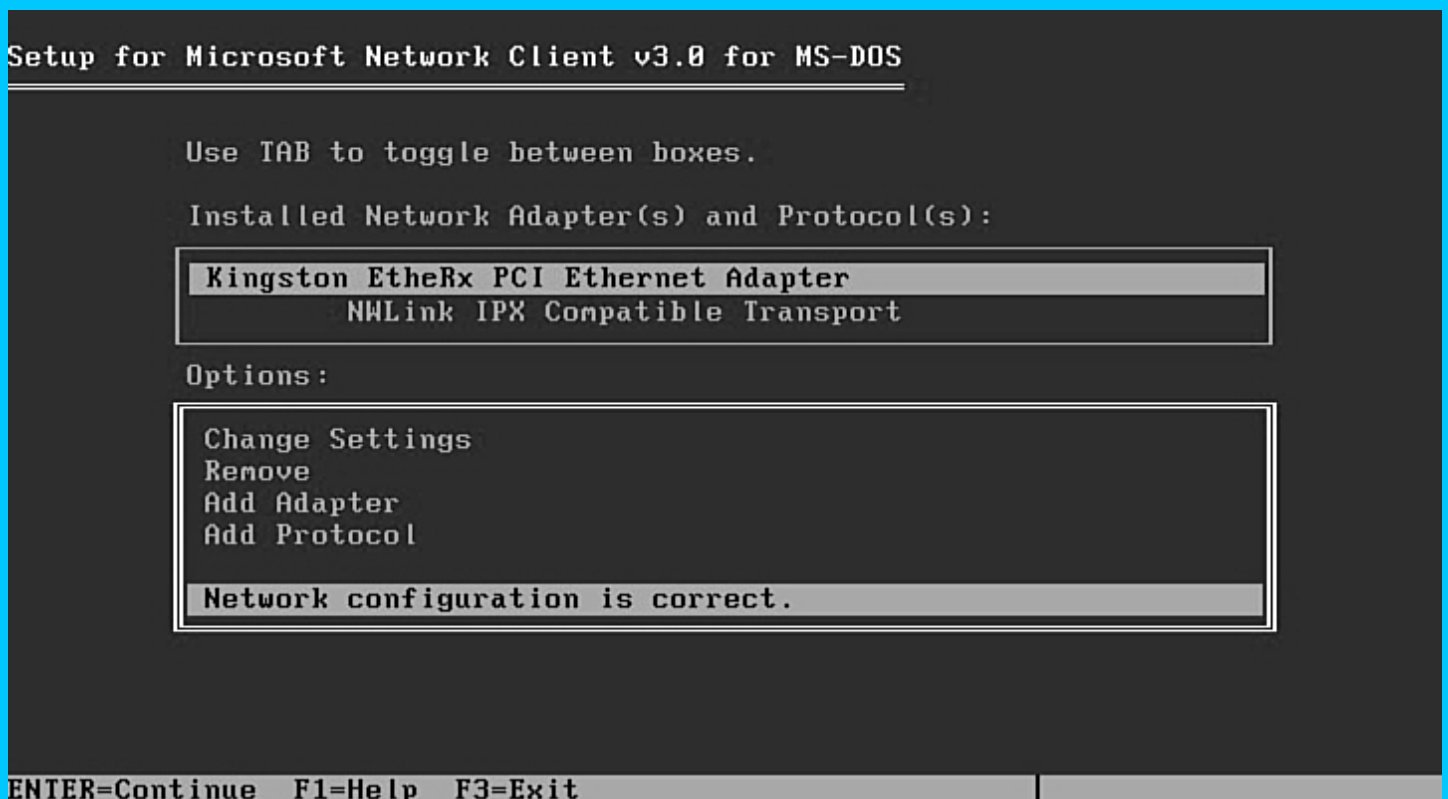


Abbildung 15.7 zeigt die installierten Netzwerkkomponenten. Hier wählen Sie, welche Protokolle installiert werden sollen. Wie üblich,

installieren Sie nur das TCP/IP-Protokolll und, wie immer, ist das Standardverhalten, etwas anderes zu installieren. Sie haben Ihre Netzwerkkarte bereits ausgewählt, also sollten Sie diese Einstellung jetzt nicht ändern müssen.

Abb. 15.7: Installierte Netzwerkkomponenten



Fügen Sie zunächst TCP/IP hinzu. Abbildung 15.8 zeigt die Liste verfügbarer Protokolle. Nachdem Sie das TCP/IP-Protokoll hinzugefügt haben, gehen Sie über die `[Tab]`-Taste zum NWLink-Eintrag, den Sie in Abbildung 15.9 sehen, und löschen Sie ihn, indem Sie die Option `Remove` benutzen, die auf dem Bildschirm gelistet ist. Bevor Sie zum vorherigen Bildschirm, der in Abbildung 15.4 dargestellt ist, zurückkehren, konfigurieren Sie manuell die Einstellungen für Ihre IP-Adresse und Subnetzmaske. Abbildung 15.10 zeigt die Standardwerte für die Netzwerkadressen. Die einzigen, die Sie hier betreffen, sind die Subnetzmaske, die IP-Adresse und `Disable DHCP` (setzen Sie den Wert auf `1`). Wenn Sie über einen Router auf Hosts zugreifen müssen, können Sie außerdem ein Standard-Gateway spezifizieren.

Abb. 15.8: Liste verfügbarer Netzwerkprotokolle für die Installation

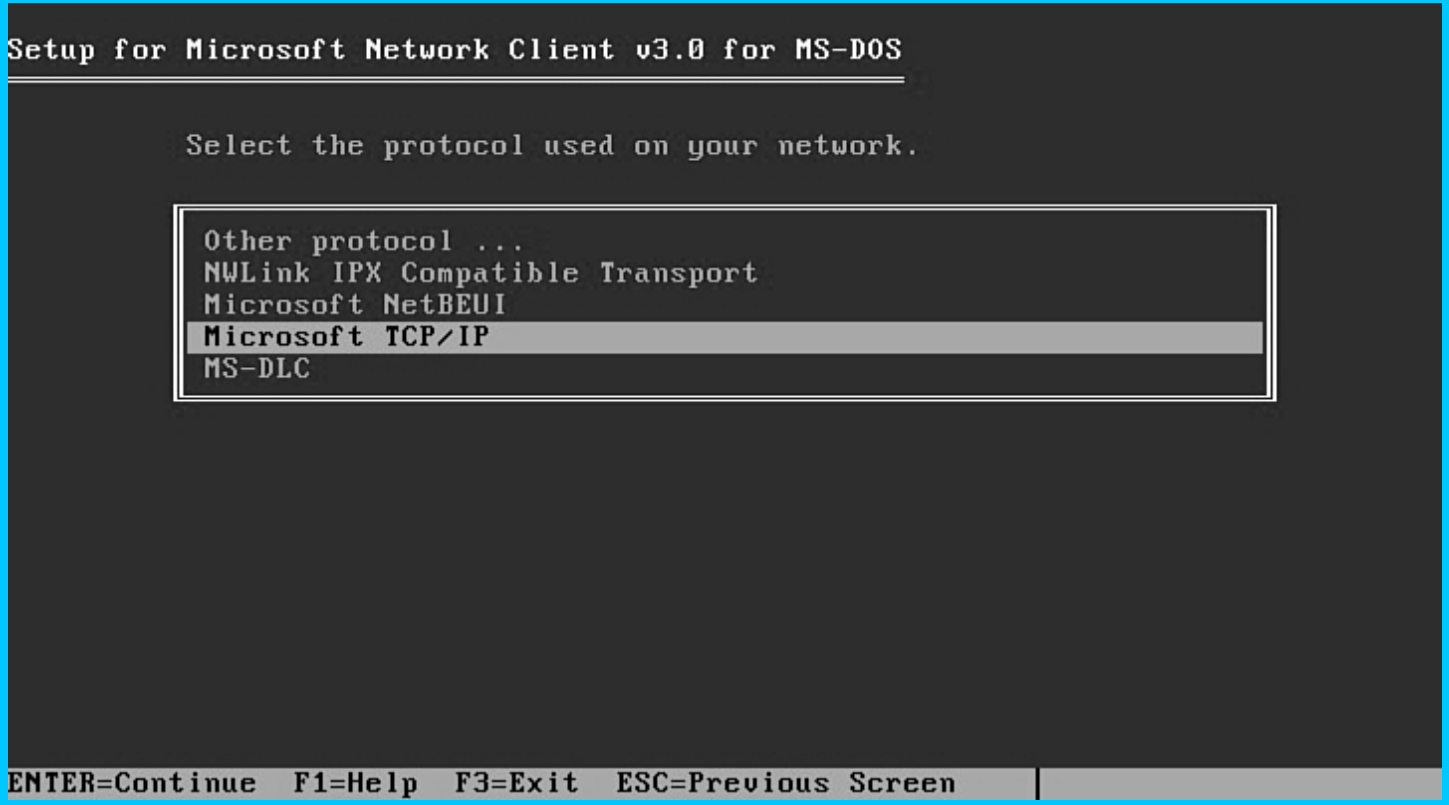


Abb. 15.9: Installierte Netzwerkkomponenten nach Hinzufügen des TCP/IP-Protokolls

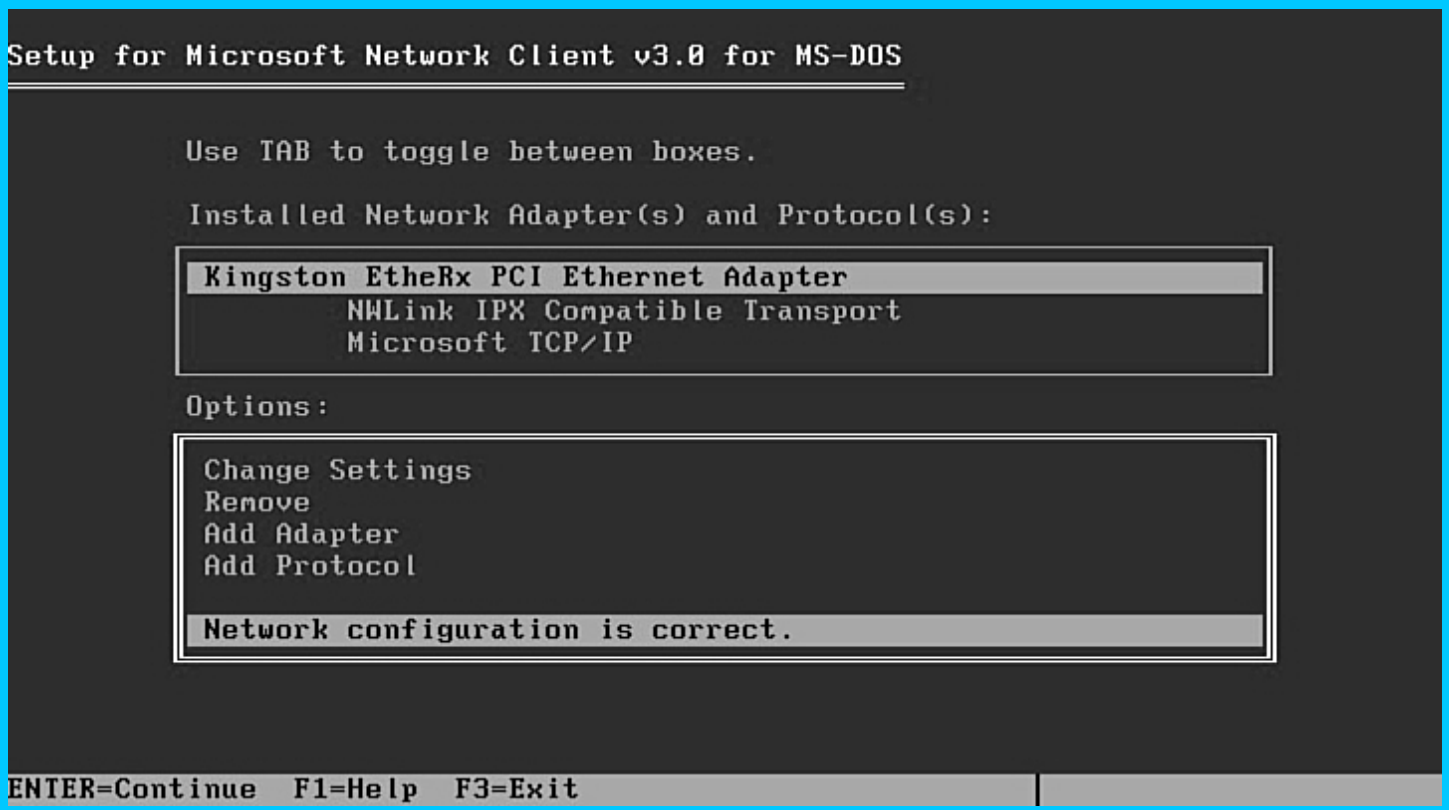
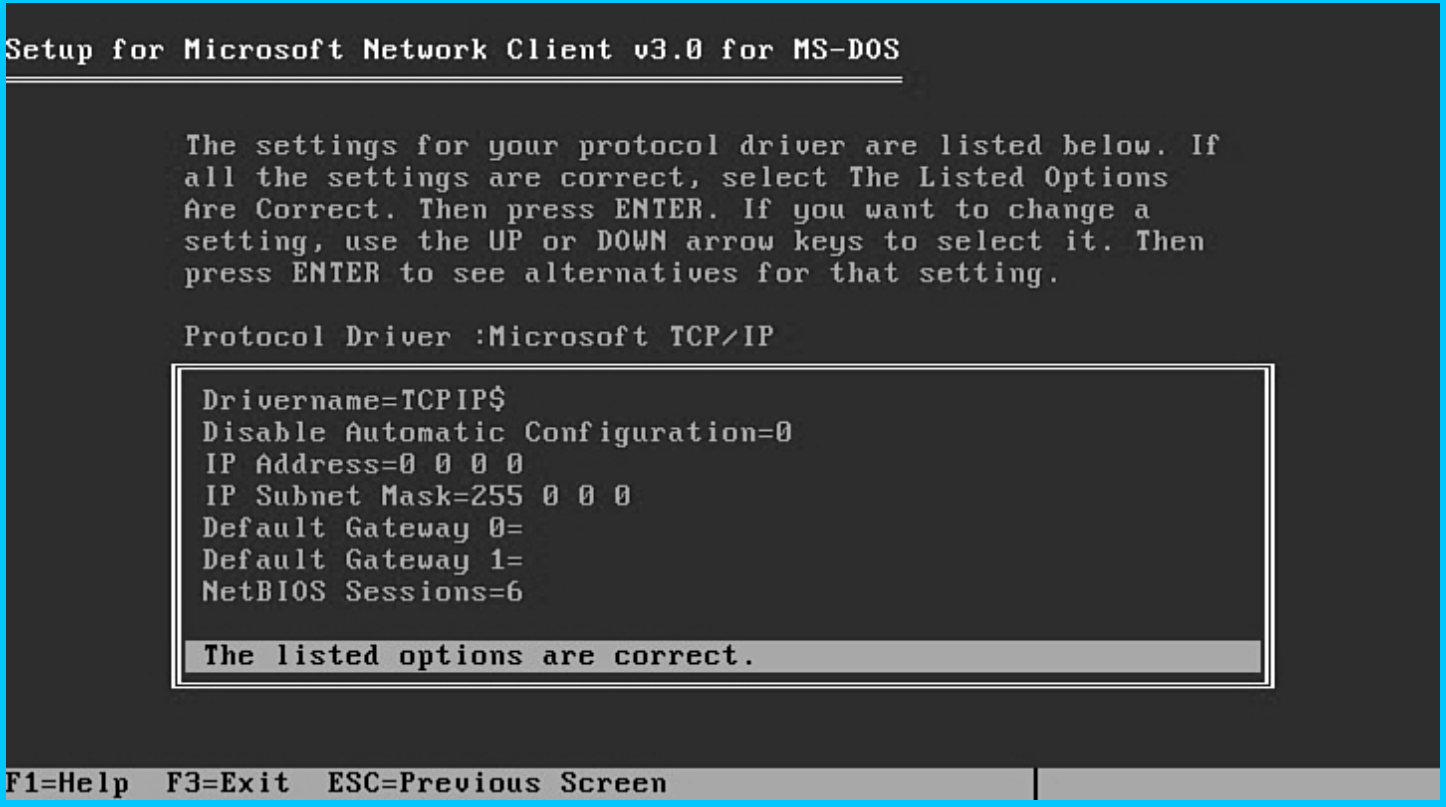


Abb. 15.10: Liste der Standardwerte für Netzwerkadressen



Jetzt ist die Netzwerkkonfiguration korrekt, und Sie kehren zum Bildschirm in Abbildung 15.4 zurück. Nach Kopieren der notwendigen Dateien informiert Sie das Setup-Programm, dass alles erfolgreich abgeschlossen wurde und Sie den Rechner neu starten sollten, damit die neue Konfiguration in Kraft tritt.

Wenn der Rechner hochfährt, werden Sie aufgefordert, sich in das Netzwerk einzuloggen. Sie werden eine ähnliche Aufforderung wie meine sehen:

```
Type your user name, or press ENTER if it is Jerry:
```

Nachdem Sie `[Return]` gedrückt haben, werden Sie aufgefordert, Ihr Passwort einzugeben:

```
Type your password:*******
```

Danach sucht der DOS-Netzwerk-Client nach einer Passwort-Cache-Datei, um das angegebene Passwort darüber zu authentifizieren. Da Sie sich zum ersten Mal einloggen, findet er natürlich keine derartige Datei. Daher fragt er Sie, ob Sie eine erstellen wollen. Aufgrund meiner natürlichen Vorsicht gegenüber dem Zwischenspeichern von Passwörtern auf der Festplatte eines unsicheren Computers lehne ich ab:

```
There is no password-list for JERRY.
Do you want to create one? (Y/N) [N]:
```

Nachdem Sie informiert wurden, dass der Befehl erfolgreich beendet wurde, können Sie die Basisoptionen für den Befehl `net.exe` benutzen, um auf Laufwerke zuzugreifen und das Netzwerk zu browsen. Sie können z.B. über den Befehl `net use` ein freigegebenes Zip-Laufwerk von einem Windows-NT-4.0 oder auch einem Samba-Server mounten:

```
C:\> net use z: \\picante\zipdisk
Der Befehl wurde erfolgreich ausgeführt.
```

Der DOS-Netzwerk-Client hat jedoch das gleiche Problem wie der Windows-9x-Client. Alle Verbindungsversuche zu Servern benutzen den Login-Namen, der eingegeben wurde, als der Client auf dem Rechner gestartet wurde. Es gibt keine Methode, dies zu umgehen.

Die Netzwerk-Boot-Diskette erstellen

Sie haben jetzt einen funktionieren Netzwerk-Client auf der lokalen Festplatte und können damit beginnen, die Netzwerk-Boot-Diskette zu erstellen. Folgender Befehl erstellt eine einfache Boot-Diskette:

```
C:\> format a: /s
```

Nachdem die Systemdateien auf die Diskette übertragen wurden, erstellen Sie zwei Verzeichnisse, eins mit dem Namen `DOS` und das andere mit dem Namen `NET`. Das erste Verzeichnis wird alle notwendigen DOS-Utilities enthalten, wie z.B. Speichermanager, und das zweite alle notwendigen Dateien für den DOS-Netzwerk-Client.

Wenn Sie sich das \NET-Verzeichnis auf der lokalen Festplatte ansehen, werden Sie etwa 1,6 Mbyte Dateien finden. Das Auffinden der richtigen Dateien, die auf die Diskette kopiert werden müssen, kann aus mehreren Versuchen bestehen. Deshalb finden Sie die Dateien, die Sie auf der Diskette benutzen müssen, in Listing 15.1. Danach folgt Listing 15.2 mit der `config.sys`-Datei der Festplatte und Listing 15.3 mit der `autoexec.bat`-Datei.

Listing 15.1: Dateien auf der Netzwerk-Boot-Diskette

```
   Datenträger in Laufwerk A: hat keine Bezeichnung
   Datenträgernummer 2629-09D8

Verzeichnis von A:\

dos        <DIR>              02-01-99    2:20a
net        <DIR>              02-01-99    2:20a
autoexec   bat           245   02-01-99    2:40a
command    com        54,645   05-31-94    6:22a
config     sys           132   02-01-99    2:39a
    5 Datei(en)        55,022 Bytes

Verzeichnis von A:\DOS

.          <DIR>              02-01-99     2:20a
..         <DIR>              02-01-99     2:20a
format     com        22,974   05-31-94     6:22a
sys        com         9,432   05-31-94     6:22a
emm386     exe       120,926   05-31-94     6:22a
fdisk      exe        29,336   05-31-94     6:22a
vi         exe        46,130   01-24-96     2:23p
himem      sys        29,136   02-13-94     6:21a
    8 Datei(en)       257,934 Bytes

Verzeichnis von A:\NET
.          <DIR>              02-01-99     2:20a
..         <DIR>              02-01-99     2:20a
hosts                   715   08-31-94     7:37p
lmhosts                 817   08-31-94     7:36p
networks                395   08-31-94     6:52p
protocol                795   08-31-94     6:52p
services              5,973   05-08-95     2:34p
wfwsys     cfg          840   02-01-99    12:54a
netbind    com        8,513   08-31-94    12:00a
umb        com        3,325   08-31-94    12:00a
connect    dat           40   02-01-99     2:44a
ktc40      dos        49,057   07-21-95     7:31p
protman    dos        21,940   08-31-94    12:00a
nemm       dos         2,619   08-31-94    12:00a
tcpdrv     dos         4,174   08-31-94    12:00a
protman    exe        13,782   08-31-94    12:00a
emsbfr     exe         4,294   08-31-94    12:00a
nmtsr      exe        22,826   08-31-94    12:00a
ping       exe        66,460   08-31-94    12:00a
tcptsr     exe        71,040   08-31-94    12:00a
tinyrfc    exe        37,024   21-01-94     7:39p
net        exe       450,326   02-07-95    12:40p
system     ini           497   02-01-99     2:42a
protocol   ini           356   02-01-99     1:51a
tcputils   ini           233   08-31-94    12:00a
net        msg        76,234   03-03-95     7:11p
neth       msg       123,066   03-03-95     7:12p
shares     pwl           622   02-01-99    12:54a
ifshlp     sys         4,644   08-31-94    12:00a
    29 Datei(en)       970,607 Bytes
```

```
    42 Datei(en)      1,283,563 Bytes
                     18,432 Bytes frei
```

Listing 15.2: config.sys-Datei von der Netzwerk-Boot-Diskette

```
device=a:\dos\himen.sys
device=a:\dos\emm386.exe noems
dos=high,umb
files=99
buffers=45
lastdrive=z
device=a:\NET\ifshlp.sys
```

Listing 15.3: autoexec.bat-Datei von der Netzwerk-Boot-Diskette

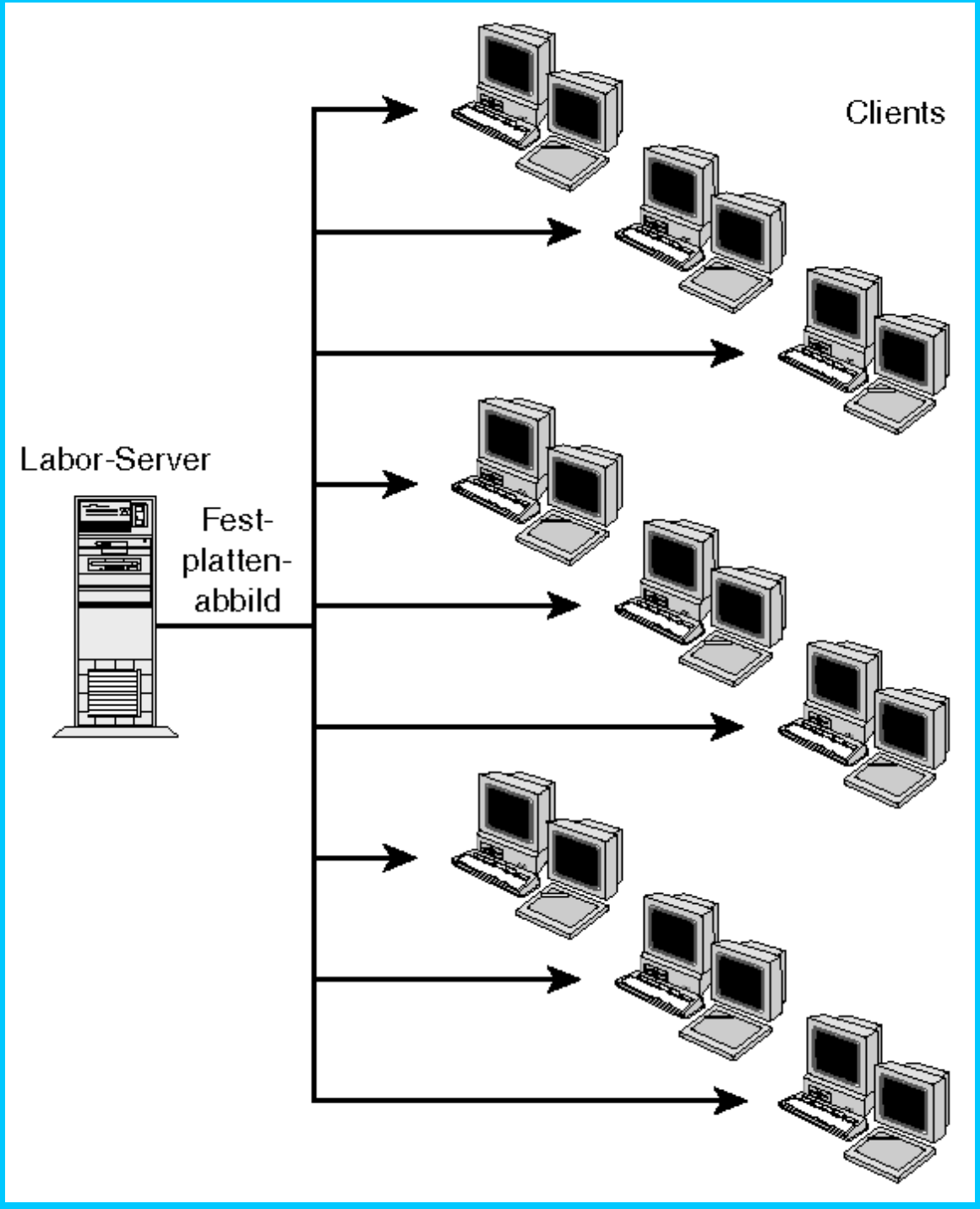
```
@echo off
set prompt=$p$g
set dircmd=/l/oe/p
set copycmd=/v
set PATH=a:\NET;a:\dos

a:\NET\net initialize
a:\NET\netbind.com
a:\NET\umb.com
a:\NET\tcptsr.com
a:\NET\tinyrfc.exe
a:\NET\nmtsr.exe
a:\NET\emsbfr.exe
a:\NET\net start
```

Ich überlasse Ihnen die Entscheidung, was Sie mit dieser Diskette tun. Ich werde Ihnen von der Situation erzählen, die mich zum ersten Mal dazu brachte, eine dieser Disketten zu erstellen.

Wie ich schon vorher erwähnt habe, ist eine meiner Aufgaben an meinem Arbeitsplatz die Verwaltung mehrerer öffentlicher Computerlabors. Jedes Labor enthält Rechner mit homogener Hardware. Daher kann ich die Festplatten kopieren und sie einfach ohne große Probleme auf einen anderen Rechner übernehmen. Es gibt viele Software-Pakete für das Klonen von Festplatten. Sie können bis zu einem gewissen Grad sogar den Befehl `xcopy` benutzen. Das spezielle Paket, das ich benutze, ermöglicht es, dass eine gesamte Festplatte in eine einzelne Datei ausgegeben wird. Diese Dateien können recht groß sein und sich aufgrund von Updates und Korrekturen im Labor häufig ändern.

Ursprünglich benutzte ich ein beschreibbares CD-ROM-Laufwerk, um die Image-Datei auf eine CD-ROM zu kopieren, aber dies stellte sich als problematisch heraus. Seit ich über die notwendige Software gestolpert bin, kann ich einfach über das Netzwerk eine Image-Datei des Servers an jeden Rechner im Labor gleichzeitig erstellen (siehe Abbildung 15.11)! Wenn Sie bedenken, dass ein standardmäßig konfigurierter Rechner das Abbild in etwa 20 Minuten lädt und ein Labor fast 50 Rechner hat, können Sie sich vorstellen, wie viel Zeit ich gespart habe.

Abb. 15.11: Alle Rechner im Labor über eine Netzwerkverbindung gleichzeitig vom Server laden



DAVE 2.1 für das Macintosh-Betriebssystem

DAVE ist ein SMB-Client, der von Thursby Software Systems (http://www.thursby.com) entwickelt wurde und vertrieben wird. Mit DAVE können sich Ihre Macintosh-Rechner mit SMB-Servern verbinden. DAVE wird derzeit zum Einzelhandelspreis von 149 Dollar verkauft und funktioniert mit Windows NT, Windows 95, WfWg 3.11 (mit TCP/IP) und Samba.

Lassen Sie mich zunächst sagen, dass ich nicht jeden Tag einen Mac benutze, so dass mir das Herumspielen mit DAVE viel Spaß gemacht hat. Ich war sehr beeindruckt von der allgemeinen Einfachheit der Installation und den Konnektivitätsoptionen. Das sollte als farbenprächtige Broschüre reichen. Sie finden Produktbeschreibungen für den Thursby-Client auf der Website des Unternehmens. Jetzt sollten wir uns einige Konfigurationseinzelheiten ansehen.

Die minimalen Hardware- und Software-Anforderungen für DAVE sind:

- Ein Macintosh-Computer mit einem 68020-Prozessor oder höher
- Mac OS Version 7.5 oder höher
- 8 Mbyte RAM

- Apples MacTCP oder Open Transport TCP/IP 1.1 oder höher
- Notwendige Hardware für den Einsatz von TCP/IP (z.B. eine funktionierende Netzwerkkarte)

Bevor Sie DAVE installieren, sollten Sie sicherstellen, dass Ihre TCP/IP-Konfiguration korrekt funktioniert.

DAVE installieren

Für dieses Beispiel habe ich eine Testversion von DAVE v2.1 von http://www.thursby.com/ heruntergeladen. Auf dem Client-System läuft Mac OS 7.6 mit 16 Mbyte RAM. Sie können auf Thursbys Website ein Formular ausfüllen, um einen temporären Registrierungsschlüssel zu erhalten. Sie können außerdem das Benutzerhandbuch (etwa 172 Seiten) im Adobe-Acrobat-Reader-Format herunterladen. Ich fand dies ebenfalls recht hilfreich.

Nachdem Sie die Client-Software heruntergeladen und extrahiert sowie den Registrierungsschlüssel bereit haben, können Sie DAVE installieren, indem Sie das Installer-Icon ausführen (siehe Abbildung 15.12), das sich im Ordner `DAVE 2.1` befindet.

Nachdem Sie das Installationsprogramm gestartet und sich durch die Lizenzbestimmungen gearbeitet haben, kommen Sie zum DAVE-Installer-Fenster, das Sie in Abbildung 15.13 sehen. Für dieses Beispiel benutzen Sie Easy Install. Dieser Bildschirm bietet Ihnen auch die Möglichkeit, bestimmte Teile des Clients zu installieren oder die Software zu entfernen. Zusätzlich zum Client enthält DAVE SMB-Datei-und-Drucker-Freigabeunterstützung, so dass die Beziehung zweiseitig ist.

Abb. 15.12: Das Installer-Icon

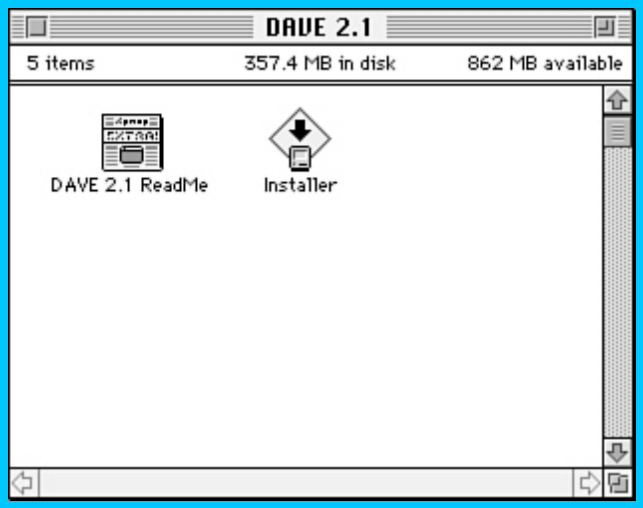
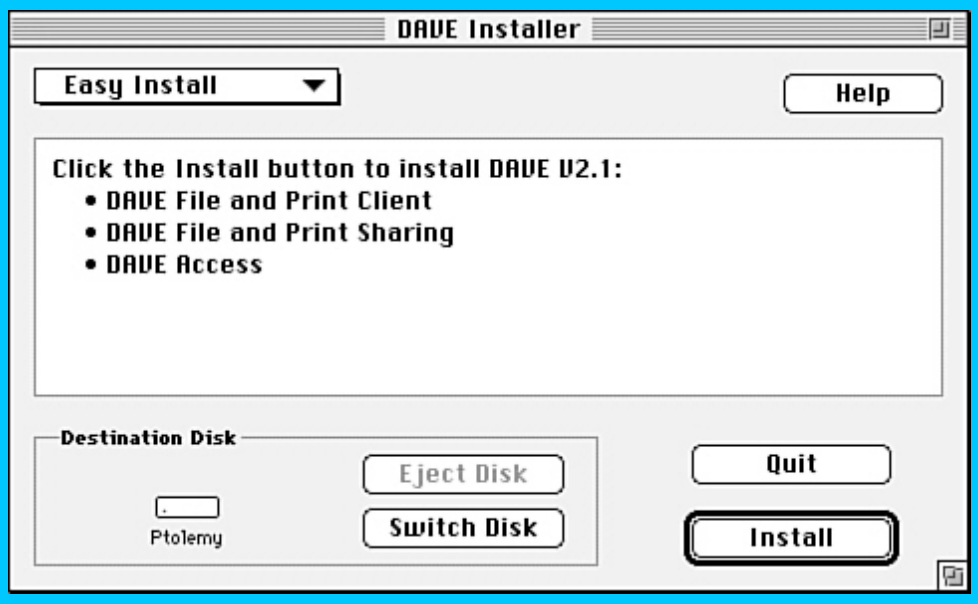


Abb. 15.13: Das DAVE-Installer-Fenster für die Auswahl der Optionen

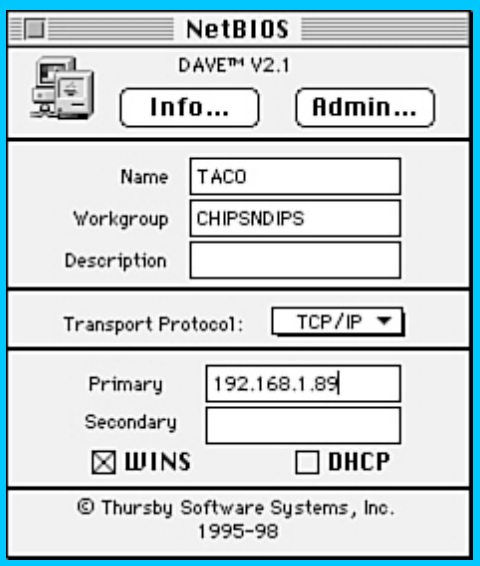


NetBIOS konfigurieren

Nachdem DAVE installiert ist, startet das Client-System neu. Nachdem es wieder hochgefahren ist, können Sie das NetBIOS-Interface konfigurieren. Sie können auf diese Einstellungen zugreifen, indem Sie im Menü *Control Panels,* das Sie im *Apple*-Menü finden, die Option

NetBIOS wählen. Haben Sie bis jetzt noch keine Registrierungsinformationen für die Software eingegeben, werden Sie nun aufgefordert, Ihren Namen, Ihr Unternehmen und Ihre Lizenznummer anzugeben. Wenn Sie dies erledigt haben, sollten Sie den Bildschirm in Abbildung 15.14 sehen.

Abb. 15.14: Das DAVE-Fenster für die NetBIOS-Einstellungen



An diesem Punkt sollte Ihnen das Ganze vertraut erscheinen. Das Feld *Name* ist für den Computernamen, nicht für einen Benutzernamen. *Workgroup* ist die Arbeitsgruppe, die für Browsing-Zwecke verwendet wird. Es muss nicht die gleiche sein wie die Domäne, in die Sie sich später einloggen werden. Das optionale Feld *Description* entspricht dem Samba-Parameter `server string`. Definieren Sie außerdem manuell die IP-Adresse Ihres WINS-Servers. Diese Information könnte Sie auch von einem DHCP-Server erhalten.

Eine Sache, die DAVE bietet, ist die erweiterte Kontrolle über die NetBIOS-Einstellungen. Die Schaltfläche *Info* im NetBIOS-Kontrollfenster hat eine ähnliche Funktion wie der Windows-Befehl `nbtstat.exe` (siehe Abbildung 15.15). Über diese Schaltfläche können Sie Dinge sehen wie z.B. welche Namen lokal zwischengespeichert werden, welche Namen der Client registriert hat und welche NBT-Sitzungen aktiv sind. Über die Schaltfläche *Admin* können Sie z.B. die Scope-ID, den Knotentyp und verschiedene Zeit- und Wiederholungseinstellungen für die Namensregistrierung und -auflösung einrichten (siehe Abbildung 15.16).

Abb. 15.15: Registrierte Namen ansehen

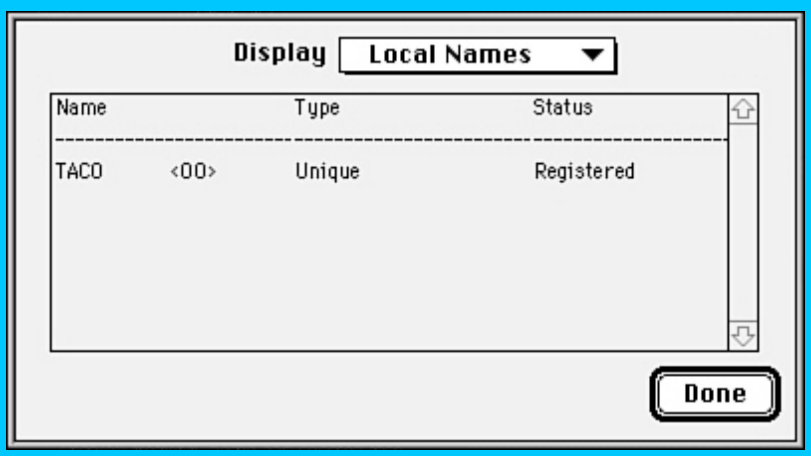
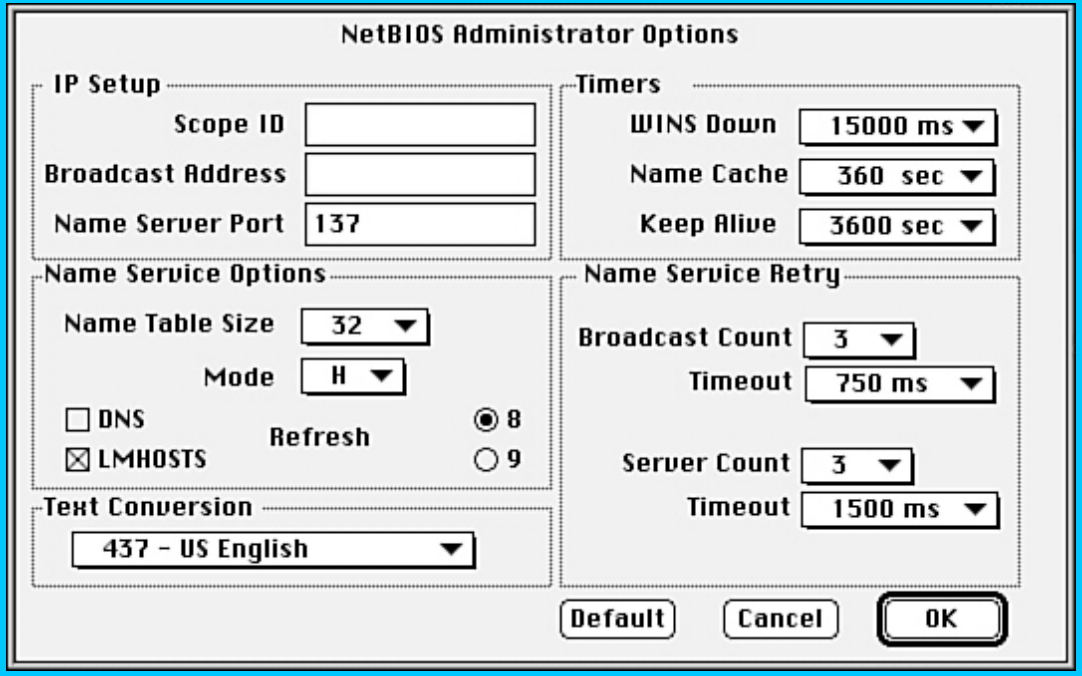


Abb. 15.16: NetBIOS-Parameter einrichten

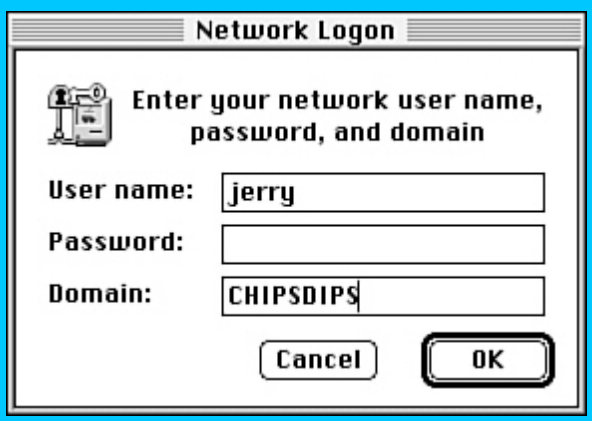


Wenn Sie fertig sind, schließen Sie das Fenster. Das Betriebssystem fordert Sie auf, Ihre Änderungen zu speichern. Wenn Sie aus der Windows-Welt kommen, sind Sie vielleicht überrascht, dass Sie den Computer nicht neu starten müssen, nachdem Sie diese Einstellungen definiert haben. Jedes Mal, wenn Sie Informationen in der Registerkarte *Identifikation* im Netzwerkdialogfeld in Windows 95 ändern, müssen Sie den Rechner neu starten, bevor die Änderungen in Kraft treten.

Einloggen und auf Server zugreifen

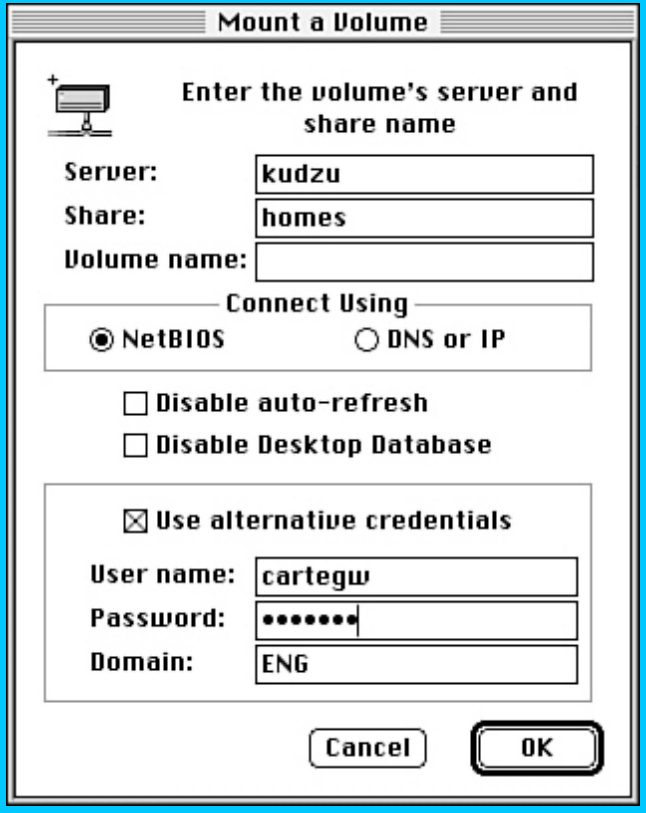
Wenn Sie auf Freigaben zugreifen, verhält sich DAVE mehr wie Windows NT als wie Windows 95, da DAVE es Ihnen ermöglicht, einen anderen Benutzernamen und ein anderes Passwort für die Verbindung zu einer Freigabe zu benutzen. Sie müssen sich jedoch zuerst über das DAVE-Access-Programm einloggen, das vom *Apple*-Menü gestartet werden kann. Abbildung 15.17 zeigt das Dialogfeld *Network Logon*, auf das Sie zugreifen können, indem Sie *Log On* aus dem *Access*-Menü wählen. Ich habe entschieden, mich in eine Domäne einzuloggen, die von einem Samba-Server kontrolliert wird. Natürlich würde es keinen Sinn ergeben, wenn das Login-Skript ausgeführt werden würde, aber DAVE unterstützt die Sprache AppleScript.

Abb. 15.17: Über DAVE in eine von Samba kontrollierte Domäne einloggen



Nachdem Sie sich erfolgreich eingeloggt haben, lässt DAVE Sie Freigaben von SMB-Servern mounten. Abbildung 15.18 zeigt das Fenster, das geöffnet wird, wenn Sie die Option *Mount a Volume* aus dem *Access*-Menü wählen. Das Beispiel zeigt die Einstellungen, die ich benutze, um mich mit dem Verzeichnis `home` an meinem Arbeitsplatz zu verbinden.

Abb. 15.18: Über DAVE eine SMB-Festplattenfreigabe mounten



Wenn Sie es vorziehen, über ein Interface, das der Windows-Netzwerkumgebung ähnlich ist, nach Freigaben zu suchen, ermöglicht Ihnen das *Chooser*-Fenster (siehe Abbildung 15.19) Browsing über AppleShare und SMB, wenn Sie sie installiert haben. Haben Sie den gewünschten Server und die Freigabe gefunden, unterstützt DAVE eine Funktion, die die Verbindung zur Freigabe beim Login wiederherstellt (siehe Abbildung 15.20), ähnlich wie eine Dauerverbindung unter Windows NT.

Abb. 15.19: Im Mac-Fenster Chooser können Sie das gesamte Netzwerk durchsuchen

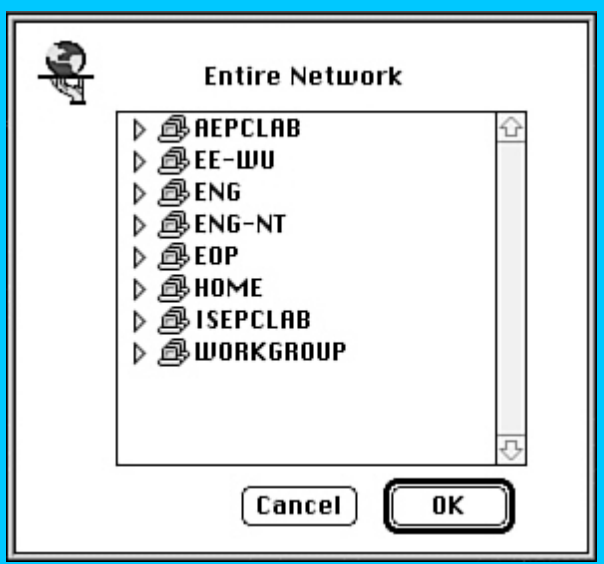
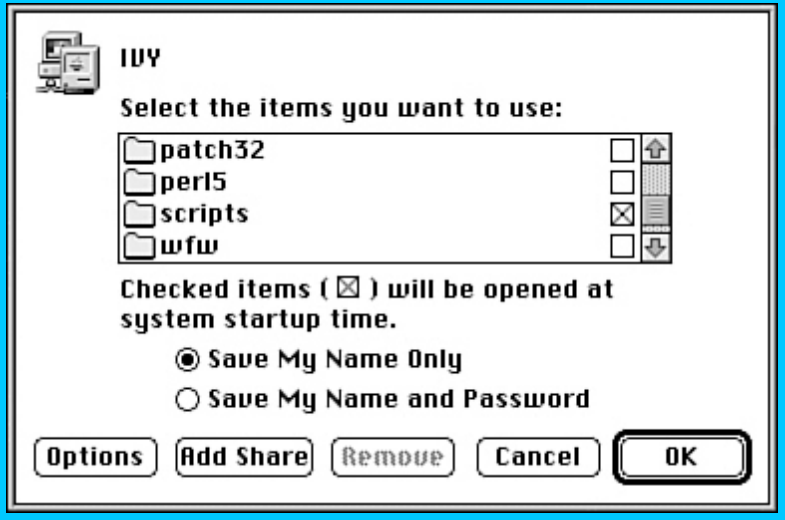


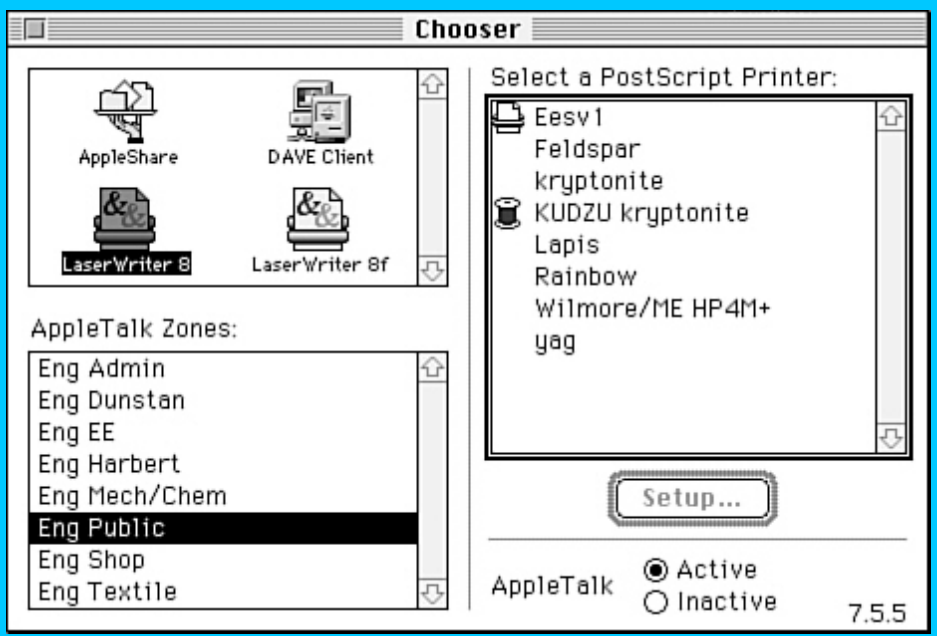
Abb. 15.20: Eine Freigabe für die Wiederverbindung beim Login zuordnen



Drucker werden ganz ähnlich ausgewählt wie Festplattenfreigaben. Abbildung 15.21 zeigt das Fenster, das bei Verbindung zu einem entfernten Drucker geöffnet wird. Es ist notwendig, den Drucker im *Chooser*-Fenster unter dem Eintrag *LaserWriter* zu konfigurieren, damit die Netzwerkverbindung gespulte Druckaufträge akzeptiert. Abbildung 15.22 zeigt die Liste der Drucker. Der Eintrag mit dem Namen `KUDZU Kryptonite` ist der Drucker, auf den ich über DAVE zugreife. Sie sollten beachten, dass die Dokumentation für DAVE angibt, dass im Moment nur PostScript-Drucker unterstützt werden.

Abb. 15.21: Über DAVE mit einem entfernten Drucker verbinden

Abb. 15.22: Den zugeordneten Drucker für die Konfiguration auswählen



Zusammenfassung

Dieses Kapitel hat Ihnen zwei sehr verschiedene SMB-Clients vorgestellt. Microsofts DOS-Netzwerk-Client wird von verschiedenen Orten frei verteilt. Thursby Software Solutions Produkt DAVE ist eine kommerzielle Lösung für die Integration von Macitosh-Clients in ein SMB-basiertes Netzwerk.

Frage & Antwort

F. Wenn ich versuche, über den MS-DOS-Netzwerk-Client mit dem Basis-Redirector ein Laufwerk von einem Samba-Server zu mounten, erhalte ich ständig eine Fehlermeldung, die besagt, dass das Passwort ungültig ist. Woran liegt das?

. Der Basis-Redirector des DOS-Clients überträgt nur den LanManager 24-Byte-Hashwert. Ist auf Ihrem Samba-Server die Passwortverschlüsselung nicht aktiviert, müssen Sie entweder den vollen Redirector von DOS benutzen oder die Verschlüsselung von Passwörtern auf Ihrem Server aktivieren.

F. Ich scheine Probleme mit der Benutzung von DAVE 2.1 und Samba 2.0 zu haben. Keine der Dateien auf dem Samba-Server wird angezeigt.

. Der Grund hierfür liegt darin, dass Samba sein Dateisystem als NTFS angibt. Die Einstellung `fstype = Samba` in Ihrer `smb.conf` sollte das Problem lösen.

Neue Begriffe

NDIS2 - Version 2 der Network Desktop Interface Specification. Diese erstellt ein definiertes Interface zwischen der Hardware und dem höher gelegenen Netzwerkprotokoll, das Hersteller verwenden können, um konforme Gerätetreiber für Netzwerkkomponenten zu schreiben. Es gibt auch eine Version 3 dieser Spezifikation, die NDIS3 genannt wird.

Tag 16: Passwortsynchronisation

In den vorhergehenden Kapiteln haben Sie sich angesehen, wie eine ganze Reihe verschiedener Clients auf Datei- und Druckerfreigaben zugreifen können, die von einem Samba-Server zur Verfügung gestellt werden. Sie haben sich außerdem angesehen, wie Samba einen lokalen Benutzer-Acount mit allen Dateizugriffen verbindet. In den meisten Fällen übermitteln Benutzer einen Benutzernamen und ein Passwort, um auf eine Ressource auf einem Samba-Server zugreifen zu können. Wenn Sie eine komplexe Umgebung verwalten, in der Sie NT-Server, Unix-Server (einige mit Samba) und vielleicht auch einige Windows-9x-Systeme haben, werden Sie wahrscheinlich ein Problem mit der Passwortsynchronisation haben.

In diesem Kapitel geht es um

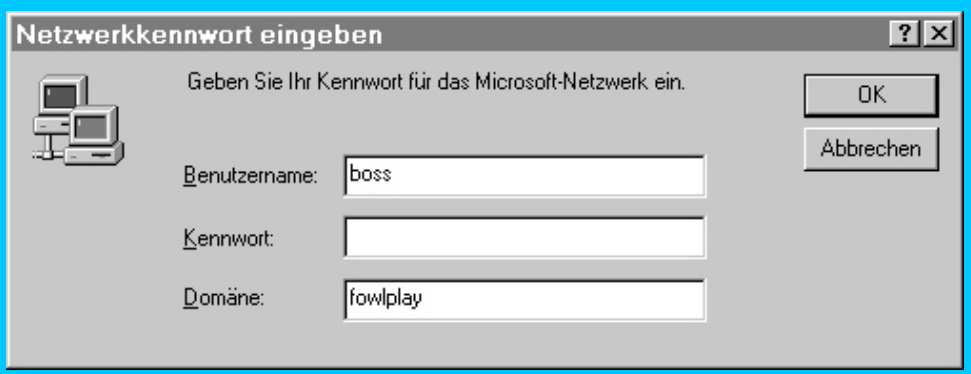
- das Problem mit der Passwortsynchronisation,
- einen Überblick über einige Ansätze zur Passwortsynchronisation,
- die Samba-basierte Passwortsynchronisation,
- die PAM-basierte Passwortsynchronisation,
- einige Windows-NT-Ansätze in Bezug auf dieses Problem,
- LDAP als eine potentielle Lösung,
- die verbleibenden Probleme.

Einige der Lösungen für dieses Problem basieren auf der Unterstützung für verschlüsselte Passwörter und der neuen PDC-Unterstützung, die in Samba 2.0.0 und höher integriert ist. Es ist vielleicht gut, erst einmal die Kapitel 21, »Windows-9x-Domänenkontrolle«, und 22, »Experimentelle PDC-Unterstützung«, zu lesen, bevor Sie dieses Kapitel angehen.

Was ist das Problem?

Was ist das Problem? Nehmen wir an, Sie wollen sich in ein Windows-9x-System einloggen und schauen auf das Dialogfeld, das Sie in Abbildung 16.1 sehen.

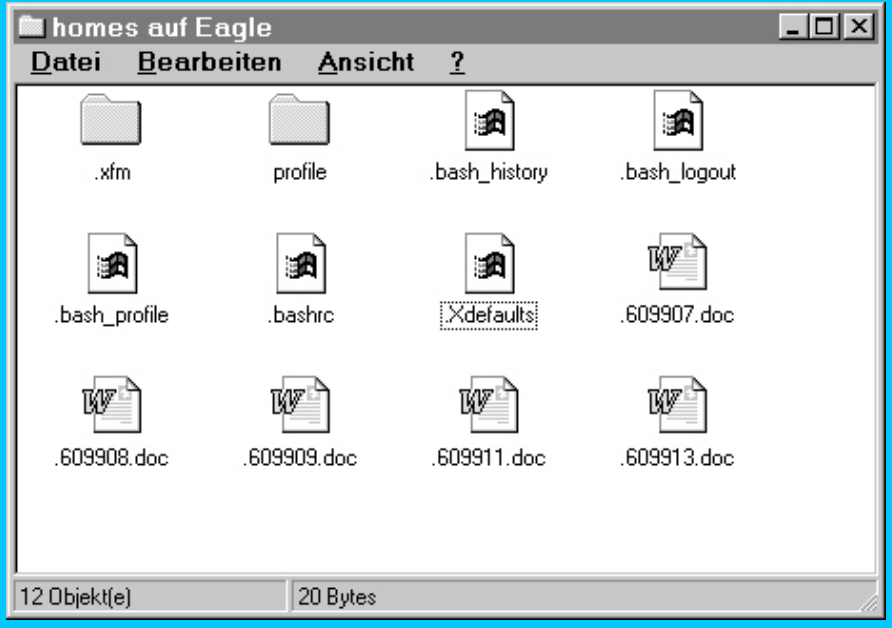
Abb. 16.1: Das Login-Dialogfeld unter Windows 9x



Geben Sie Ihren Benutzernamen (hier `boss`) und Ihr Passwort an, loggen Sie sich in das Netzwerk ein, und Sie erhalten Zugang zu Ihren Dateien. Wo wird Ihr Passwort verwaltet? Wenn Sie Ihr Passwort dann ändern (auf das Sie über die Systemsteuerung zugreifen können), wo wird dieses Passwort geändert?

Wenn Sie sich eingeloggt haben, erhalten Sie möglicherweise Zugriff auf ein Home-Verzeichnis, in dem Sie Dateien und Verzeichnisse speichern können. Abbildung 16.2 zeigt ein Beispiel des Home-Verzeichnisses, das auf einer Samba-Freigabe für den Benutzer `boss` zur Verfügung steht.

Abb. 16.2: Das Home-Verzeichnis nach Einloggen in das Netzwerk



Nun werden Sie sich auch in Unix-Rechner in Ihrem Netzwerk einloggen und Zugriff auf den gleichen Account-Namen haben müssen, den Sie für das Einloggen in Ihre Windows-9x-Systeme verwenden. Nachfolgend sehen Sie eine Beispiel-Login-Sitzung, während der Sie sich in ein Linux-System mit dem Account-Namen `boss` einloggen und Zugang zu den gleichen Dateien erhalten, auf die Sie von Windows in Abbildung 16.2 zugegriffen haben:

```
Red Hat Linux release 5.2 (Apollo)
Kernel 2.0.36 on an i686
login: boss
Password:
[boss@eagle]$ ls -al
total 2261
drwx------  4 boss  boss    1024 Jan 19 18:34 .
drwxr-xr-x 21 root  root    1024 Jan 19 16:41
-rw-r--r--  1 boss  boss    3768 Jan 10 12:59 .Xdefaults
-rw-r--r--  1 boss  boss      24 Jan 10 12:59 .bash_logout
-rw-r--r--  1 boss  boss     220 Jan 10 12:59 .bash_profile
-rw-r--r--  1 boss  boss     124 Jan 10 12:59 .bashrc
drwxr-xr-x  2 boss  boss    1024 Jan 10 12:59 .xfm
-rwxr--r--  1 boss  boss  551424 Jan 13 00:46 609907.doc
-rwxr--r--  1 boss  boss  309248 Jan 12 15:29 609908.doc
-rwxr--r--  1 boss  boss   48640 Jan 13 00:10 609911.doc
-rwxr--r--  1 boss  boss  431104 Jan 18 22:25 609913.doc
drwxr-xr-x  5 boss  boss    1024 Jan 22 14:28 profile
```

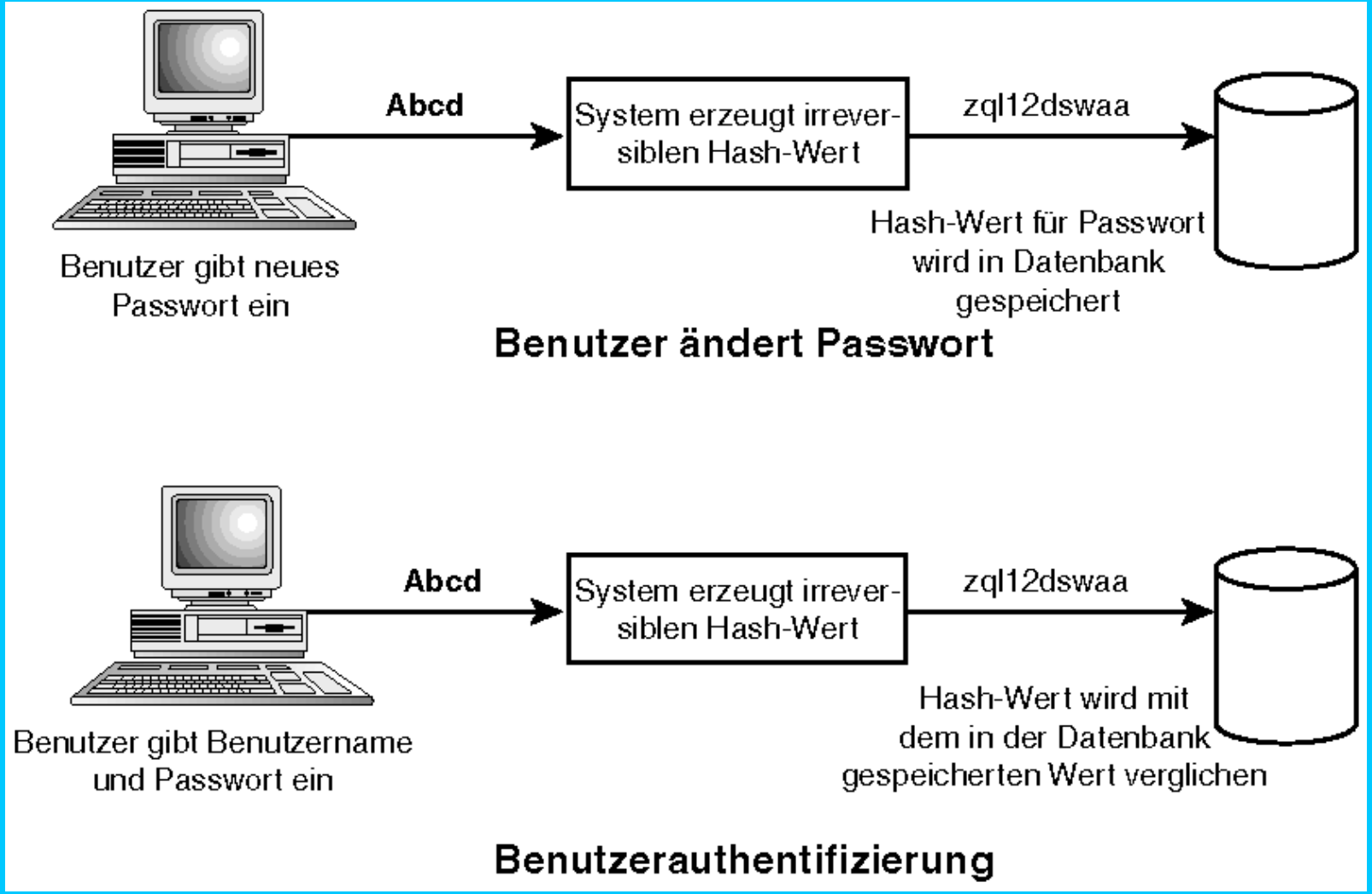
Welches Passwort wurde benutzt, um Sie in jedem dieser Befehle zu authentifizieren? Wenn Sie Ihr Passwort in einer Umgebung ändern, wird es dann für beide Umgebungen geändert? Ist es möglich, dass nur ein Passwort verwendet wird?

Damit Sie das Problem verstehen und den Hintergrund einiger der Lösungen beurteilen können, müssen Sie mehr darüber wissen, wo Windows NT und Unix Passwörter für ihre Benutzer speichern und auf welche Art und Weise diese Passwörter gespeichert werden.

Zunächst einmal speichert weder NT noch Unix in den entsprechenden Passwortdateien Klartextpasswörter. Beide führen eine irreversible Kalkulation auf Grundlage des Klartextpassworts durch, das ein Benutzer überträgt, wenn er ein Passwort ändert. Der resultierende Wert wird in den Passwortdatenbanken gespeichert, die jedes System verwaltet. Wenn ein Benutzer sich einloggen möchte, wird die gleiche irreversible Kalkulation noch einmal durchgeführt und das Resultat mit dem Wert verglichen, der in der Passwortdatenbank gespeichert ist. Stimmen die Werte überein, hat der Benutzer das korrekte Passwort übertragen. Beide Prozesse werden in Abbildung 16.3 dargestellt.

Zwar führen beide Betriebssysteme eine irreversible Berechnung auf Grundlage des Passworts durch und speichern das Resultat, aber jedes benutzt einen anderen Algorithmus. Windows NT berechnet einen MD4-Hashwert auf Grundlage des Benutzerpassworts, während Unix das Benutzerpasswort als Schlüssel für einen modifizierten DES-Algorihmus verwendet und diesen speichert. Da jede dieser Operationen in einem angemessenen Zeitrahmen irreversibel ist, ist es nicht möglich, einen Passworttyp in den anderen zu konvertieren. Das bedeutet, erhält ein System einen Windows-Hashwert, kann es keinen korrekten Unix-Hashwert für den gleichen Benutzer generieren und umgekehrt.

Abb. 16.3: Ändern von Passwörtern und Authentifizierung



Windows NT speichert Passwörter in der *SAM- (Security-Account-Manager-)*Datenbank, die Teil der Registry ist. In einer Windows-NT-Domäne werden Passwörter auf dem Primary Domain Controller gespeichert und an Backup Domain Controller repliziert. Obwohl jeder NT-Rechner auch seine eigene lokale SAM-Datenbank haben kann, ermöglicht diese nur lokale Logins. Beim Einloggen in die Domäne wird die SAM-Datenbank auf den Domänen-Controllern konsultiert. Da Backup Domain Controller eine Kopie der SAM-Datenbank erhalten, gibt es im Wesentlichen nur eine Quelle für Authentifizierungsinformationen in einer Windows-NT-Domäne.

Unix speichert Passwörter in der Passwortdatei (`/etc/passwd`) oder der Shadow-Passwortdatei für erhöhte Sicherheit (da die Passwortdatei allgemein lesbar und es möglich ist, einen erfolgreichen Wörterbuchangriff durchzuführen, wenn jemand Zugriff auf die verschlüsselten Passwörter für einen Rechner hat). In einer vernetzten Umgebung werden verschiedene Schemata benutzt, um eine einzelne Quelle für Authentifizierungsinformationen für Unix zu bieten. Das beliebteste ist ein Schema, das von Sun Microsystems entwickelt wurde und *Network Information Service* oder NIS genannt wird. Vor einiger Zeit hat Sun eine aktualisierte Version von NIS namens NIS+ vorgestellt, die eine Reihe von Problemen mit NIS behandelt, insbesondere Sicherheitsprobleme. NIS+ ist allerdings nicht unter so vielen Unix-Versionen implementiert wie NIS.

Wenn Sie sowohl Windows NT als auch Unix in Ihrem Netzwerk haben und dieses Netzwerk umfassend ist, ist die Wahrscheinlichkeit groß, dass Sie zwei Sammlungen von Passwörtern haben, eine für Windows NT und eine für Unix. Außerdem kann Samba Windows-NT-artige Passwörter verwenden und verwalten, wenn es für die Benutzung verschlüsselter Passwörter konfiguriert ist. Wenn Sie Samba unter Unix benutzen, um Datei- und Druckerfreigaben für Windows-9x- und Windows-NT-Systeme zu bieten und verschlüsselte Passwörter verwenden, haben Sie ebenfalls zwei Sets von Passwörtern, eins für die Windows-Clients und eins für Unix.

Das bedeutet, der Account `boss` hat ein Passwort für Windows und ein Passwort für Unix.

Dies führt zu einem Problem: Wie können Sie diese Passwörter synchron halten, da Benutzer es nicht mögen, sich an zwei Passwörter erinnern oder ein einzelnes Passwort an zwei verschiedenen Orten ändern zu müssen? Wenn Sie den einfachen Ansatz nehmen und Samba in einer Unix-Umgebung (oder einer gemischten Windows-NT- und Unix-Umgebung) einrichten, haben Sie ein Problem mit der Synchronisation der Passwörter, da Ihre Benutzer Passwörter an zwei Orten haben werden und diese manuell synchron halten müssen. In einigen Fällen macht sich das Problem vielleicht nicht bemerkbar, aber wenn Ihre Windows-Benutzer einen Unix-basierten POP-Server oder RADIUS-Server usw. verwenden, werden sie das Problem sicher bemerken.

In den folgenden Abschnitten sehen Sie sich Ansätze für den Umgang mit dem Problem der Passwortsynchronisation an.

Ansätze für die Passwortsynchronisation

Die Passwortsynchronisation stellt Sie vor zwei Fragen, die Sie beantworten müssen:

- Versuchen Sie, nur ein Passwortformat zu haben, oder verwalten Sie verschiedene Passwortformate, d.h. eins für jede Umgebung (z.B. Windows NT, Unix usw.)?
- Versuchen Sie, nur eine Master-Kopie der Passwortdatenbank (vielleicht mit Backup-Kopien, die von der Master-Kopie aktualisiert werden) zu halten oder kann jede Umgebung ihre eigene(n) Passwortdatenbank(en) haben, in der Sie sicherstellen, dass Passwörter aktualisiert werden, wenn ein Benutzer sein Passwort ändert?

Wenn Sie nur ein Passwortformat verwenden, können Sie die Passwort-Hashwerte auf Client-Systemen generieren, wenn Benutzer ihre Passwörter ändern. Verwenden Sie aber mehrere Passwortformate, müssen Sie entweder beim Client mehrere Passwort-Hashwerte generieren, wenn ein Passwort geändert wird (was bedeutet, dass Clients geändert werden müssen, wenn der Umgebung ein neues Passwortformat hinzugefügt wird), oder Sie müssen das Klartextpasswort an einen Server übertragen, damit dieser die entsprechenden Hashwerte generiert.

In jedem Fall sollten Sie besser die Daten verschlüsseln, die Sie an den Authentifizierungsserver übertragen, um zu verhindern, dass jemand, der die Übertragung der Passwortänderung verfolgen kann, entweder Passwörter oder Passwortentsprechungen abfängt. (Eine Passwortentsprechung kann von Angreifern benutzt werden, die ihre eigenen Client-Programme schreiben.)

Können Sie eine einzelne Authentifizierungsdatenbank halten (auch wenn Sie in dieser Datenbank verschiedene Passwort-Hashwerte haben), wird Ihr Leben viel einfacher sein, wenn Sie die Datenbank aktualisieren - z.B. wenn die Benutzer ihre Passwörter ändern. Die Verwaltung mehrerer Authentifizierungsdatenbanken dagegen bedeutet, dass Sie besondere Sorgfalt an den Tag legen müssen, um sicherzustellen, dass Aktualisierungen an jede Datenbank weitergegeben werden, und es wird immer Gelegenheiten geben, dass die Datenbanken nicht länger synchron sind. Die Probleme werden gemischt, wenn sich in diesen Datenbanken Passwort-Hashwerte in verschiedenen Formaten befinden, da dann irgendwo auf dem System Klartextpasswörter gespeichert werden müssen, bis alle Datenbanken aktualisiert sind.

Die verfügbaren Lösungen für das Problem der Passwortsynchronisation umfassen folgende Ansätze:

- Richten Sie Windows NT so ein, dass es für die Authentifizierung NIS auf Unix-Rechnern benutzt. Dieser Ansatz ermöglicht Ihnen, alle Passwörter an einem Ort unter NIS zu halten, aber er bedeutet auch, dass Ihre NT-Systeme Standalone-Geräte sind, die nicht an einer Domäne teilnehmen können. Für diese Lösung brauchen Sie eine Software namens NISGINA, die Sie unter http://www.dcs.qmw.ac.uk/~williams/ erhalten können. Diese Lösung hat nichts mit Samba zu tun und wird nicht weiter dargestellt.
- Richten Sie Samba als Logon-Server ein (und möglicherweise als einen Primary Domain Controller, wenn Sie NT-Geräte haben) und verwenden Sie verschlüsselte Passwörter in Samba. Sie können Samba dann dahingehend konfigurieren, dass es Passwortänderungen in Ihr spezielles Unix-Passwortsystem überträgt. Leider ermöglicht diese Lösung alleine nicht, dass Passwortänderungen auf Unix-Seite in das Samba-Passwortsystem gelangen.
- Erweitern Sie die gerade beschriebene Lösung um *Pluggable Authentication Modules* (PAM: verfügbar für Solaris und Linux und Teil der OSF DTE, die heute von den meisten Unix-Herstellern ausgeliefert, aber leider auf den meisten Systemen nicht dargestellt werden), damit Unix-Authentifizierung für die Samba-Passwortdatenbank durchgeführt werden kann. In einigen Fällen ermöglichen die PAM außerdem, dass Passwortänderungen auf Unix-Systemen ihren Weg in die Samba-Passwortdatei finden. Dies wird später in diesem Kapitel ausführlicher dargestellt.
- Verwenden Sie PAM, um alle Unix-Authentifizierungen mit einem SMB-Server irgendeiner Art durchzuführen, entweder mit Samba oder einem echten Windows NT Primary Domain Controller. Diese Lösung ermöglicht derzeit nicht, dass Passwortänderungen auf Unix-Seite in die Passwortdatenbank übernommen werden.
- Benutzen Sie etwas wie *LDAP* (*Lightweight Directory Access Protocol*) für die zentrale Verwaltung der Authentifizierungsinformationen und lassen Sie alle Systeme Anfragen an den LDAP-Server übertragen. Dies müsste auf Unix-Seite um PAM für LDAP-Zugriff erweitert werden und würde von einer LDAP-Unterstützung in Samba abhängen. Im Moment ist die LDAP-Unterstützung in Samba noch in der Entwicklung und sollte nicht für Produktionsumgebungen benutzt werden, aber es ist klar, dass LDAP einer der besten Ansätze für die Zukunft ist.

Sie können auch SWAT für die Änderung von Benutzerpasswörtern verwenden. SWAT leidet jedoch unter einer Reihe von Problemen, unter anderem darunter dass es keine Unterstützung für SSL bietet, so dass alle vom Benutzer übertragenen Passwörter für jeden Eindringling sichtbar sind.

Samba-basierte Passwortsynchronisation

Samba kann für die Authentifizierung von Windows-95- und Windows-NT-Clients eingerichtet werden. Dies erfordert (besonders für Windows-NT-Systeme), dass Samba verschlüsselte Passwörter und die `smbpasswd`-Datei benutzt.

Sie müssen dem globalen Abschnitt Ihrer `smb.conf` die folgenden Parameter hinzufügen:

```
encrypt passwords = yes
smb passwd file = /usr/local/samba/private/smbpasswd
```

Natürlich kann sich, je nach Ihrer Unix-Version, die `smbpasswd`-Datei auch an einem anderen Ort befinden. In Linux-Distributionen, wie z.B. RedHat und TurboLinux, befindet sie sich in `/etc`.

Sie müssen außerdem die `smbpasswd`-Datei erstellen und sie mit allen Accounts bestücken, die in die `smbpasswd`-Datei vorgeladen werden sollen. In der `smbpasswd` speichert Samba alle NT-Passwort-Hashwerte für alle Windows-Benutzer. Die Datei hat ein ähnliches Format wie die Unix-Passwortdatei.

Um die `smbpasswd` anfangs zu füllen, müssen Sie einen Befehl ausführen wie den folgenden:

```
cat /etc/passwd | mksmbpasswd.sh > /usr/local/samba/private/smbpasswd
```

oder, wenn Sie NIS für die Unix-Authentifizierung verwenden:

```
ypcat passwd | mksmbpasswd.sh > /usr/local/samba/private/smbpasswd
```

Das `mksmbpasswd.sh`-Skript finden Sie in Samba 2.0 im Samba-Source-Verzeichnis im Unterverzeichnis `Script`.

Aufgrund der Sensitivität der Informationen, die in der `smbpasswd` gespeichert werden (wie z.B. die verschlüsselten Passwörter in der `passwd`-Datei, die Sie in einer Shadow-Passwortdatei speichern können), sollte das Verzeichnis, in dem die `smbpasswd`-Datei gespeichert ist, im Besitz von `root` sein. Außerdem sollte niemand außer `root` auf das Verzeichnis zugreifen können. Und schließlich sollte die `smbpasswd`-Datei selbst für niemanden außer `root` zugänglich sein. Die folgenden Befehle setzen die korrekten Berechtigungen für `smbpasswd`, nachdem Sie die Datei erstellt haben:

```
chown -R root.root /usr/local/samba/private
chmod 500 /usr/local/samba/private
chmod 600 /usr/local/samba/private/smbpasswd
```

Wenn Sie die `smbpasswd` erstellt und Samba neu gestartet haben, können Ihre Windows-Benutzer ihre Passwörter ändern, und die `smbpasswd` wird aktualisiert. Die Unix-Passwörter für ihre Accounts auf dem Samba-Server werden jedoch nicht geändert, es sei denn, Sie richten weitere globale Parameter in Ihrer `smb.conf` ein.

Sie *können* Samba so einrichten, dass es das Unix-Passwort eines Benutzers ändert, wenn dieser sein Windows-Passwort ändert. Dafür müssen Sie den Parameter `unix password sync` benutzen und eventuell auch die Parameter `passwd chat`, `passwd chat debug` und `passwd program` setzen.

Diese Funktionalität ist verfügbar, weil Samba jetzt den API-Aufruf für die Passwortänderung unterstützt und Zugriff auf die Klartextversion des neuen Passworts des Benutzers hat. Sie sollten jedoch wissen, dass die neuen Informationen verschlüsselt über das Netzwerk übertragen werden.

Da die entsprechenden Informationen verfügbar sind, kann Samba für den Benutzer die Ausführung eines Befehls organisieren, der die Passwortänderung auch in die Unix-Umgebung überträgt. Dieser Befehl kann einfach `/bin/passwd` sein oder auch `/bin/yppasswd` oder ein entsprechender lokaler Passwortbefehl.

In den folgenden Abschnitten sind die `smb.conf`-Parameter dargestellt, die für die Passwortsynchronisation relevant sind.

unix passwd sync

Dieser globale Parameter spezifiziert, ob Samba versucht, das Unix-Passwort eines Benutzers zu synchronisieren, wenn das Windows-Passwort dieser Person in `smbpasswd` geändert wird. Ist der Parameter auf `True` eingestellt, wird das über den Parameter `passwd program` definierte Programm als `root` aufgerufen, um das Passwort des Benutzers unter Unix zu ändern. Das Passwortprogramm muss als `root` aufgerufen werden, da Samba keinen Zugriff auf die Klartextversion des alten Benutzerpassworts hat. Der Standardwert für diesen Parameter ist:

```
unix password sync = False
```

Damit Samba Passwortänderungen synchron hält, ändern Sie diesen Wert auf `True`.

passwd chat

Dieser globale Parameter spezifiziert die Passwort-Chat-Sequenz, die Samba verwendet, um das Passwort des Benutzers unter Unix zu ändern, wenn diese Person ihr Windows-Passwort ändert. Dieser String nimmt die Form einer Sequenz von Ein- und Ausgabepaaren an, die der `smbd` benutzt, um festzulegen, was er an das `passwd`-Programm senden und was er zurückbekommen sollte. Empfängt der `smbd` nicht die erwartete Antwort, wird das Passwort des Benutzers nicht geändert.

Die Chat-Sequenz ist normalerweise spezifisch für Ihr Unternehmen und hängt ab von der Ausgabeform für Meldungen des Befehls `passwd` oder `yppasswd` auf Ihrem System bzw. des Befehls, den Sie für die Änderungen von Passwörtern auf Ihrem System verwenden.

Die Chat-Sequenz kann die Makros `%o` und `%n` enthalten, die durch das alte bzw. neue Passwort ersetzt werden. Sie kann außerdem die üblichen

Makros `/n`, `/r`, `/t` und `/s` enthalten, die für Zeilenvorschub, Wagenrücklauf, Tabulator bzw. Leerzeichen stehen.

Enthält der String in einer Antwort ein Sternchen (`*`), entspricht dies einer beliebigen Zeichenfolge. Zusätzlich können doppelte Anführungszeichen benutzt werden, um Strings mit eingebetteten Leerzeichen zu spezifizieren. Ein Punkt (`.`) in einem beliebigen Teil der Sequenz bedeutet, dass kein String gesendet wird, wenn der Punkt in einer Ausgabesequenz vorkommt, oder keine Antwort erwartet wird, wenn der Punkt in einer Antwortsequenz auftaucht. Der Standardwert für diesen Parameter ist:

```
passwd chat = \ *old*password* %o\n *new*password* %n\n *changed*
```

passwd chat debug

Dieser globale Parameter bestimmt, ob das `passwd`-Chat-Skript im Debug-Modus laufen soll. Wenn eingeschaltet, werden die vom `passwd`-Programm erhaltenen und an das Programm gesendeten Strings mit einem Debug-Level von 100 protokolliert. Um den String sehen zu können, müssen Sie Ihren Debug-Level auf 100 einstellen.

Diese Option ist gefährlich, da sie dazu führt, dass Klartextpasswörter in der `passwd`-Datei gesehen werden können.

Der Standardwert für diesen Parameter ist:

```
passwd chat debug = False
```

passwd program

Dieser globale Parameter teilt dem `smbd` mit, welches Programm gestartet wird, um das Unix-Passwort eines Benutzers zu ändern. Jedes vorkommende `%u` wird durch den Benutzernamen in der Befehlszeile ersetzt. Der Standardwert für diesen Parameter ist:

```
passwd program = /bin/passwd
```

Sie sollten diesen Parameter nicht ändern müssen.

PAM-basierte Passwortsynchronisation

Zwar können über die Parameter `unix password sync` und `passwd chat` die Passwortänderungen, die auf Windows-Systemen durchgeführt wurden, in Ihrem Unix-Passwortsystem reflektiert werden, aber wie sieht es mit Änderungen von Unix-Passwörtern aus?

Einer der Ansätze für die Verwaltung der Authentifizierung und Synchronisation in einer Unix-Umgebung ist die Anwendung der *PAM* oder *Pluggable Authentication Modules*. PAM sind eine Technik, die von Sun Microsystems entwickelt und in Solaris und Linux implementiert wurde. Es gibt außerdem Unterstützung im OSF CDE (Common Desktop Environment), das von Unix-Herstellern für ihre Workstations weitestgehend unterstützt wird. Es ist jedoch nicht klar, ob PAM in diesen Umgebungen für Systemadministratoren zugänglich sind.

PAM stellen sicher, dass alle Authentifizierungsentscheidungen über Shared Libraries mit definierten Eintrittspunkten implementiert werden. Die für jeden Authentifizierungstyp zu verwendenden Bibliotheken sind in einer Sammlung von Konfigurationsdateien für die PAM spezifiziert. Der Systemadministrator kann dann die entsprechende PAM-Konfigurationsdatei ändern, um die Art und Weise zu ändern, in der die Authentifizierung durchgeführt wird. PAM spezifizieren die Eingangspunkte für die Authentifizierung, die Änderung von Passwörtern und viele andere sicherheitsrelevante Funktionen. Weitere Informationen über PAM finden Sie in Ihrer Sytemadministrationsdokumentation oder auf folgenden Webseiten:

http://www.sun.org/software/solaris/pam
http://parc.power.net/morgan/Linux-PAM/

Mindestens drei PAM-Shared-Libraries ermöglichen Unix-Systemen die Authentifizierung über SMB-Server (entweder Samba oder Windows NT). Eine ermöglicht den Benutzern unter eingeschränkten Bedingungen, sowohl ihre Unix- als auch ihre SMB-Passwörter zu ändern. Diese Bibliotheken sind:

- `pam_smb` von David Airlie. Dieses Modul ermöglicht Unix-Systemen die Authentifizierung über SMB-Server, unterstützt aber nicht die Aktualisierung von Passwörtern. Daher kann es für die Authentifizierung über einen Samba-Server oder einen Windows-NT-Server benutzt

werden.

- `pam_ntdom` von Luke Leighton. Dieses Modul ermöglicht Unix-Systemen die Authentifizierung über Domänen-Controller. Es kann die Authentifizierung über einen Samba-Server, der als Primary Domain Controller läuft, und auch über einen Windows-NT-PDC durchführen. Dieses Modul unterstützt jedoch nicht die Änderung der Benutzerpasswörter.
- `pam_smbpass` von Stephen Langasek. Dieses Modul ermöglicht Unix-Systemen die Authentifizierung über SMB-Server und, unter eingeschränkten Bedingungen, die Aktualisierung der Passwörter von Unix-Benutzern in Sambas `smbpasswd`-Datei.

pam_smb: Konfiguration und Installation

Sie erhalten `pam_smb` über folgende Website: http://www.csn.ul.ie/~airlied/pam_smb.

Wenn Sie das Paket auf Ihr System kopiert haben, müssen Sie es in einem lokalen Verzeichnis entpacken und dann übersetzen und für die Benutzung konfigurieren. Benutzen Sie folgenden Befehl, um das Paket zu entpacken, wenn die tar-Version auf Ihrem System gzip-Archive unterstützt:

```
tar zxvf pam_smb-1_1_tar.gz
```

Unterstützt Ihre tar-Version keine gzip-Archive, verwenden Sie folgenden Befehl:

```
gzip -d pam_smb-1_1_tar.gz
tar xvf pam_smb-1_1_tar
```

Wenn Sie das Archiv entpackt haben, sollten Sie in das Verzeichnis wechseln, das den `pam_smb`-Source-Code enthält, und das Paket kompilieren. Die notwendigen Schritte sind:

```
cd <Source-Verzeichnis>
./configure
make
```

Damit sollten Sie eine Datei namens `pam_smb_auth.sp` erhalten. Diese Datei müssen Sie in das Verzeichnis für die PAM kopieren, das unter Linux `/lib/security` und unter Solaris `/usr/lib/security` ist.

Ihr nächster Schritt besteht darin, PAM für die Benutzung des neuen Moduls, das Sie erstellt haben, zu konfigurieren. Dafür müssen Sie die verschiedenen PAM-Konfigurationsdateien editieren und die Datei `pam_smb.conf` in `/etc` erstellen.

Sie brauchen folgende Änderungen in den PAM-Konfigurationsdateien. Für Linux müssen Sie `/etc/pam.d/login` bearbeiten und folgende Zeile hinzufügen:

```
auth    required    /lib/security/pam_smb_auth.so
```

Diese Zeile sollte vor der Zeile für `pam_pwdb.so`, aber hinter die Zeile für `pam_securetty.so` eingefügt werden.

Dies teilt PAM mit, dass zuerst Sicherheitsanforderungen überprüft und dann über `pam_smb_auth.so` authentifiziert werden sollte.

Für Solaris müssen die Zeile `other` in `/etc/pam.conf` folgendermaßen ändern:

```
other    auth required    /usr/lib/security/pam_smb_auth.so.1
```

Sowohl für Linux als auch Solaris können Sie in der Befehlszeile Parameter einfügen, darunter:

- `debug`, für Debug-Informationen über syslog
- `use_first_pass`, eine Standard-PAM-Befehlszeilenoption, die diesem Modul mitteilt, keine Aufforderung für ein Passwort auszugeben, sondern ein vorher eingegebenes Passwort zu verwenden
- `nolocal`, für die Authentifizierung eines Benutzernamen-/Passwortpaares, das nicht in der lokalen Passwortdatei ist

`pam_smb_auth` funktioniert auf folgende Art und Weise, wenn sich ein Benutzer in ein Unix-System einloggt:

1. Der Benutzer-Account muss in der Unix-Passwortdatei sein, damit sich der Benutzer einloggen kann. Ist für das Modul aber in der Befehlszeile `nolocal` spezifiziert, muss der Benutzer keinen Account in der lokalen Passwortdatei haben, dies kann aber zu Sicherheitsproblemen führen.
2. Haben Benutzer einen gültigen Eintrag in der Passwortdatei, werden sie anhand dieser Informationen authentifiziert. Das heißt, sie werden nicht über einen SMB-Server authentifiziert.
3. Hat der Benutzer kein Passwort in der Passwortdatei (d.h. ein `*` oder `!!` im Passwortfeld), benutzt `pam_smb_auth` die Inhalte der Datei `/etc/pam_smb.conf`, um den Benutzer über den spezifizierten SMB-Server zu authentifizieren.

Die `pam_smb`-Konfigurationsdatei (`/etc/pam_smb.conf`) ist eine einfache Textdatei, die die folgenden drei Zeilen enthält:

```
Domänenname
Primary Server
Backup Server
```

Für die Umgebung, die Sie in diesem Buch eingerichtet haben, kann diese Datei z.B. wie folgt aussehen:

```
FOWLPLAY
EAGLE
EAGLE2
```

Da pam_smb auf SMBlib basiert, das keine NetBIOS-Namensauflösungen durchführt, müssen alle Namen in der pam_smb-Konfigurationsdatei in Ihrer Hosts-Datei oder dem DNS verfügbar sein.

Dieses Modul ermöglicht es den Benutzern nicht, ihre Passwörter auf Remote-SMB-Servern zu aktualisieren, also können Benutzer ihr SMB-Passwort nicht von Unix-Rechnern aus ändern. Dies müssen sie von einem Windows-System aus tun.

pam_ntdom: Installation und Konfiguration

Sie erhalten pam_ntdom aus dem Verzeichnis /samba/ftp/pam_ntdom/ auf Ihrer bevorzugten Samba-Mirror-Website. Um zu diesem Verzeichnis zu kommen, wechseln Sie zu Ihrer Mirror-Site und wählen den Download-Link, der Sie zur HTTP-Site führt. Sie müssen im Wesentlichen die gleichen Schritte ausführen wie für pam_smb, um pam_ntdom zum Laufen zu bringen, da die Aufbauumgebung für pam_ntdom auf der von pam_smb basiert. Das heißt:

1. Entpacken Sie die Distribution.
2. Konfigurieren Sie das Paket (./configure).
3. Übersetzen Sie das Paket (make).

Wenn Sie pam_ntdom übersetzt haben, sollten Sie eine Datei namens pam_ntdom_auth.so im Source-Verzeichnis vorfinden. Diese Datei müssen Sie in das entsprechende Verzeichnis kopieren, wie vorher beschrieben (/lib/security für Linux-Systeme und /usr/lib/security für Solaris-Systeme).

Der nächste Schritt besteht darin, PAM in einer ähnlichen Art und Weise zu konfigurieren, wie ich es vorher für pam_smb beschrieben habe.

Für ein Linux-System fügen Sie die folgende Zeile in /etc/pam.d/login ein:

```
auth    required    /lib/security/pam_ntdom_auth.so
```

Diese Zeile sollte vor der auth-Zeile für security und hinter der für pwdb, falls in Ihrer PAM-Konfiguration enthalten, eingefügt werden.

Für ein Solaris-System ändern Sie die Zeile other in /etc/pam.conf wie folgt:

```
other   auth required   /usr/lib/security/pam_ntdom_auth.so.1
```

Dieses Modul unterstützt die gleichen Optionen, die pam_smb unterstützt, mit der gleichen Bedeutung. (Tatsächlich ist der Code für pam_ntdom im Wesentlichen der gleiche wie für pam_smb, nur mit geänderter Authentifizierungsroutine.)

Abschließend müssen Sie die pam_ntdom-Konfigurationsdatei einrichten, die /etc/pam_smb.conf heißt. Die Einträge in dieser Datei sind die gleichen wie für pam_smb und haben die gleiche Bedeutung. Eine Beispielkonfigurationsdatei wäre:

```
[root@remote1] ls /etc/pam_smb.conf
FOWLPLAY
EAGLE
EAGLE2
```

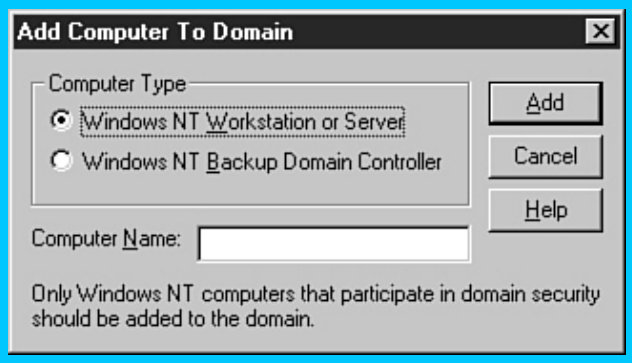
Diese Datei spezifiziert, dass die Authentifizierung über die Domäne FOWLPLAY durchgeführt wird, der primäre Authentifizierungsserver ist EAGLE und der sekundäre EAGLE2.

pam_ntdom unterstützt nur die Authentifizierung über Domänen-Controller, so dass die vorher spezifizierten Server Windows Domain Controller (PDCs oder BDCs) oder ein Samba-Server, der als Primary Domain Controller konfiguriert ist, sein müssen.

Damit ein Unix-System pam_ntdom für die Authentifizierung in einer Domäne laufen lassen kann, muss das Unix-System der Domäne hinzugefügt werden. Dies wird über zwei verschiedene Methoden erreicht, abhängig davon, ob die Authentifizierung über Windows-NT-Domänen-Controller oder Samba als Domänen-Controller stattfindet.

Damit ein Unix-System über einen Windows-NT-Domänen-Controller authentifizieren kann, müssen Sie ein solches Unix-System manuell über den Server-Manager für Domänen in die Domäne einfügen. Abbildung 16.4 zeigt ein Beispiel für das Hinzufügen eines Computers in eine Domäne. Sie müssen das Unix-System als eine Windows-NT-Workstation oder einen NT-Server hinzufügen.

Abb. 16.4: Einer Windows-NT-Domäne ein Unix-System hinzufügen



Für die Authentifizierung über einen Samba-Server, der als PDC läuft, müssen Sie der Samba-Domäne jedes Unix-System, das authentifizieren will, hinzufügen. Dies wird in der Datei `NTDOMAIN.txt` im Samba-Dokumentationsverzeichnis (`docs/textdocs`) beschrieben.

`pam_ntdom` ermöglicht es Unix-Benutzern derzeit nicht, ihre Passwörter von Unix aus zu ändern. Sie müssen die Änderungen von einem Windows-System durchführen. `pam_ntdom` wird aber wahrscheinlich bald Funktionen zur Passwortänderung bieten können.

pam_smbpass: Installation und Konfiguration

Sie erhalten `pam_smbpass` unter `ftp://ftp.netexpress.net/pub/pam`. Es ist als Source-Distribution (`pam_smbpass-0_5.tgz`) oder als ein RedHat-rpm verfügbar. Sie können sich eine Distribution aussuchen, aber hier nehme ich an, dass Sie sich für die Source-Distribution entschieden haben. Dieses Modul ist Linux-spezifisch und für die Systeme möglich, die PAM unterstützen.

Folgende Schritte müssen Sie für dieses Paket ausführen:

1. Entpacken Sie die Distribution. Dafür müssen Sie möglicherweise gzip verwenden, wenn der `tar`-Befehl Ihres Systems komprimierte tar-Dateien nicht handhaben kann.
2. Übersetzen Sie die Distribution. Dieses Paket enthält ein Makefile; starten Sie daher einfach `make` im Source-Verzeichnis.

Wenn Sie die Distribution übersetzt haben, müssen Sie das resultierende Modul nach `/lib/security` kopieren:

```
cp pam_smbpass.so /lib/security
```

Der nächste Schritt besteht darin, die korrekte PAM-Konfigurationsdatei in ähnlicher Art und Weise wie für `pam_smb` und `pam_ntdom` aufgelistet zu modifizieren. Fügen Sie dafür `/etc/pam.d/login` folgende Zeile hinzu:

```
auth    required    /lib/security/pam_ntdom_auth.so
```

Die Zeile sollte hinter der `auth`-Zeile für `security` eingefügt werden und die Zeile für `pwdb` ersetzen, wenn sie in Ihrer PAM-Konfigurationsdatei vorhanden ist.

Dieses Modul akzeptiert die folgenden PAM-Befehlszeilenoptionen:

- `debug`, protokolliert Debugging-Informationen
- `audit`, protokolliert noch mehr Debugging-Informationen
- `use_first_pass`, benutzt, wenn möglich, das Passwort, das bereits von vorherigen Modulen angefordert wurde
- `try_first_pass`, eine Variation des vorherigen Befehls (weitere Informationen finden Sie in der PAM-Dokumentation)
- `use_authtok`, ähnlich wie `try_first_pass`, funktioniert aber nicht, wenn das neue `PAM_AUTHTOK` nicht vorher gesetzt wurde
- `not_set_pass`, teilt dem Modul mit, `PAM_`-Teile nicht mit Passwörtern einzurichten, die von diesem Modul benutzt werden
- `nodelay`, verhindert fehlerhafte Authentifizierung aufgrund einer Verzögerung von etwa einer Sekunde

Der Vorteil des `pam_smbpass`-Moduls liegt darin, dass es sowohl die `passwd`-Datei als auch die `smbpasswd`-Datei auf einem Samba-Server aktualisieren kann. Dies gibt Benutzern eine Methode, ihre Passwörter von Unix aus zu modifizieren und sowohl ihr Unix- als auch ihr Windows-Passwort ändern zu lassen. In dieser Hinsicht ist das Modul allen anderen PAM-Modulen, die Sie sich bisher angesehen haben, überlegen.

`pam_smbpass` kann aber nur die `smbpasswd`-Datei auf dem Rechner ändern, auf dem es läuft. Das heißt, es kann nicht in einer Umgebung benutzt werden, in der Sie mehrere Unix-Systeme haben und Benutzer sich in jedes Unix-System einloggen können, um ihr Passwort zu ändern.

LDAP-basierte Ansätze

Wenn Ihre Umgebung groß ist und aus vielen Unix-Systemen mit einer großen Anzahl an Windows-Systemen besteht, nützt Ihnen wahrscheinlich keiner der bisher dargestellten Ansätze. Das liegt hauptsächlich daran, dass es derzeit keinen weithin implementierten einzelnen Standard für

Passwordstandorte oder -datenbanken gibt.

Über die letzten paar Jahre hat sich ein Standard namens *Lightweight Directory Access Protocol* oder *LDAP* entwickelt, der verspricht, die Probleme zu reduzieren, auf die Sie derzeit in Hinsicht auf die Passwortsynchronisierung treffen, indem Passwörter an einem Ort gehalten werden. Die LDAP-Unterstützung für Samba befindet sich derzeit noch in der Entwicklungsphase. Für alle Unix-Versionen wird eine Interaktion mit LDAP erwartet, ebenso für zukünftige Windows-Versionen.

Wenn alle Systeme in Ihrer Umgebung LDAP für die Aufbewahrung und Änderung von Passwörtern benutzen, wird es keine Problem in Hinsicht auf die Passwortsynchronisierung mehr geben.

In Anhang C, »Sambas Zukunft«, finden Sie weitere Informationen über die zukünftige Unterstützung für LDAP in Samba.

Probleme

Das verbleibende Problem mit dem größten Teil der Ansätze für Passwortsynchronisierung, die Sie sich bis hierher angesehen haben, ist, dass sie von Windows zu Unix funktionieren, nicht aber von Unix zu Windows. Zukünftige Versionen einiger Module, wie z.B. `pam_smb` und `pam_ntdom`, werden den gleichen Code enthalten, den Samba derzeit für die entfernte Änderung von Passwörtern in einer Windows-Umgebung enthält, und daher auch dieses letzte Problem lösen.

Letztendlich liegt die Lösung des Problems jedoch in LDAP.

Zusammenfassung

Sie haben sich die Möglichkeiten angesehen, die für Systemadministratoren zur Verfügung stehen, um sicherzustellen, dass Benutzerpasswörter zwischen ihren Windows- und Unix-Umgebungen abgeglichen werden. Leider gibt es derzeit noch einige Lücken in dem, was getan werden kann, aber wenn Ihr System aus einem einzelnen Unix-System besteht, das als Server für eine Anzahl von Windows-Systemen dient, wird das Leben für Sie relativ leicht sein.

Im nächsten Kapitel werden Sie sich die Unterstützung ansehen, die Samba für SSL bietet, damit Sie Zugang zu Windows-Datei- und Druckerressourcen über unsichere Netzwerke anbieten können.

Frage & Antwort

F. Unser Unternehmen hat einen großen Samba-Server und eine große Anzahl an Windows-9x-Systemen, die darauf zugreifen. Einige der Benutzer müssen jedoch auch auf ihre Unix-Accounts zugreifen. Welches ist die beste Methode, um sicherzustellen, dass alle Passwortänderungen in beiden Umgebungen sichtbar sind?

. Wenn Sie außer Ihrem Samba-Server keine anderen Unix-Systeme haben, besteht die beste Methode für Sie darin, verschlüsselte Passwörter in Samba und `pam_smbpass` für die Unix-Authentifizierung zu verwenden. So gelangen Passwortänderungen Ihrer Windows-Benutzer in die `smbpasswd`-Datei von Samba, und alle Änderungen durch Ihre Unix-Benutzer werden auch in der `smbpasswd`-Datei durchgeführt.

Haben Sie jedoch mehr als ein Unix-System, können Benutzer von anderen Unix-Rechnern als Ihrem Samba-Server Ihre Windows-Passwörter nicht von diesen Rechnern aus ändern, sondern nur vom Samba-Server aus. Auf diesen anderen Unix-Systemen kann die Authentifizierung immer noch über `pam_smb` anhand der `smbpasswd`-Datei durchgeführt werden, aber Passwortänderungen gelangen nicht in die `smbpasswd`-Datei.

F. Können wir etwas wie pam_smb verwenden, um Authentifizierung über einen Samba-Server durchzuführen, der als Primary Domain Controller läuft. Wenn ja, welche Einschränkungen gibt es?

. Ja, `pam_smb` kann für die Authentifizierung über einen Samba-Server, der als PDC läuft, verwendet werden. Die Einschränkungen liegen darin, dass solche Benutzer ihre Passwörter nicht von Unix aus aktualisieren können. Sie müssen dies von Windows aus tun.

F. Gibt es andere Methoden für die Änderung von Benutzerpasswörtern als die Verwendung der Standard-Windows- oder Unix-Funktionen?

. Ja, SWAT kann Passwörter über das Standard-Windows-NT-API für die Passwortänderung ändern. Wenn Sie aber mit SWAT kommunizieren, werden Ihr altes und Ihr neues Passwort in Klartext über das Netzwerk übertragen.

Neue Begriffe

PAM - Pluggable Authentication Modules. PAM wurden von Sun Microsystems entwickelt, von der Linux-Gemeinde aufgegriffen und werden weitreichend in Linux-Distributionen eingesetzt.

Tag 17: SSL

von Richard Sharpe

In den letzten Kapiteln haben Sie sich angesehen, wie Sie Samba auf Ihrem Unix-Server zum Laufen bringen und wie Samba mit Clients kommuniziert. Clients können sich sogar über das Internet mit Ihrem Samba-Server verbinden. Die große Frage ist: Wie sieht es mit der Sicherheit aus?

Wenn sich Clients über das Internet mit Ihrem Samba-Server verbinden, kann jeder, der in der Lage ist, den Datenverkehr abzuhören, die Dateien rekonstruieren, an denen entfernte Benutzer eventuell arbeiten. Mit den entsprechenden Programmen können diese Leute sogar Daten in Dateien einfügen, die von entfernten Benutzern bearbeitet werden.

Die beste Methode, um diese Art von Problemen zu verhindern, besteht darin, die zwischen Clients und Ihrem Samba-Server übertragenen Dateien zu verschlüsseln. Unterstützt Samba Verschlüsselung? Ja. Tatsächlich bietet Samba Unterstützung für *Secure Sockets Layer* oder *SSL*, die gleiche Verschlüsselungsspezifizierung, die auch von Webbrowsern verwendet wird, damit sie sicher mit Webservern über das Internet kommunizieren können.

Dieses Kapitel zeigt Ihnen, wie Sie Samba mit SSL-Unterstützung übersetzen und die SSL-Unterstützung in Samba konfigurieren. Sie werden sich außerdem ansehen, welche SMB-Clients SSL unterstützen und wie Sie sicheren Zugriff von Windows-Clients über das Internet einrichten können.

Dabei werden Sie auch einige Konzepte kennen lernen, z.B. was Zertifikate sind, woher Sie diese bekommen und wie Sie sie installieren, damit Clients und Samba-Server sie verwenden können.

SSL mit Samba benutzen

SSL ist ein Standard und ein Protokoll, mit dem zwei Parteien sicher über ein unsicheres Netzwerk kommunizieren können. SSL verwendet die Public-Key-Verschlüsselung und digitale Zertifikate für die Authentifizierung einer oder beider Parteien. Es benutzt symmetrische Verschlüsselung mit zufälligen Sitzungsschlüsseln, die während der Authentifizierungsphase gewählt werden, damit beide Parteien für die Dauer einer Sitzung Informationen sicher untereinander austauschen können.

Die SSL-Unterstützung in Samba basiert auf SSLeay, die frei verfügbare SSL-Source-Code-Bibliothek von Eric Young (daher kommt das *eay* im Namen).

Um Samba mit SSL-Unterstützung aufzubauen, müssen Sie folgende generelle Schritte ausführen:

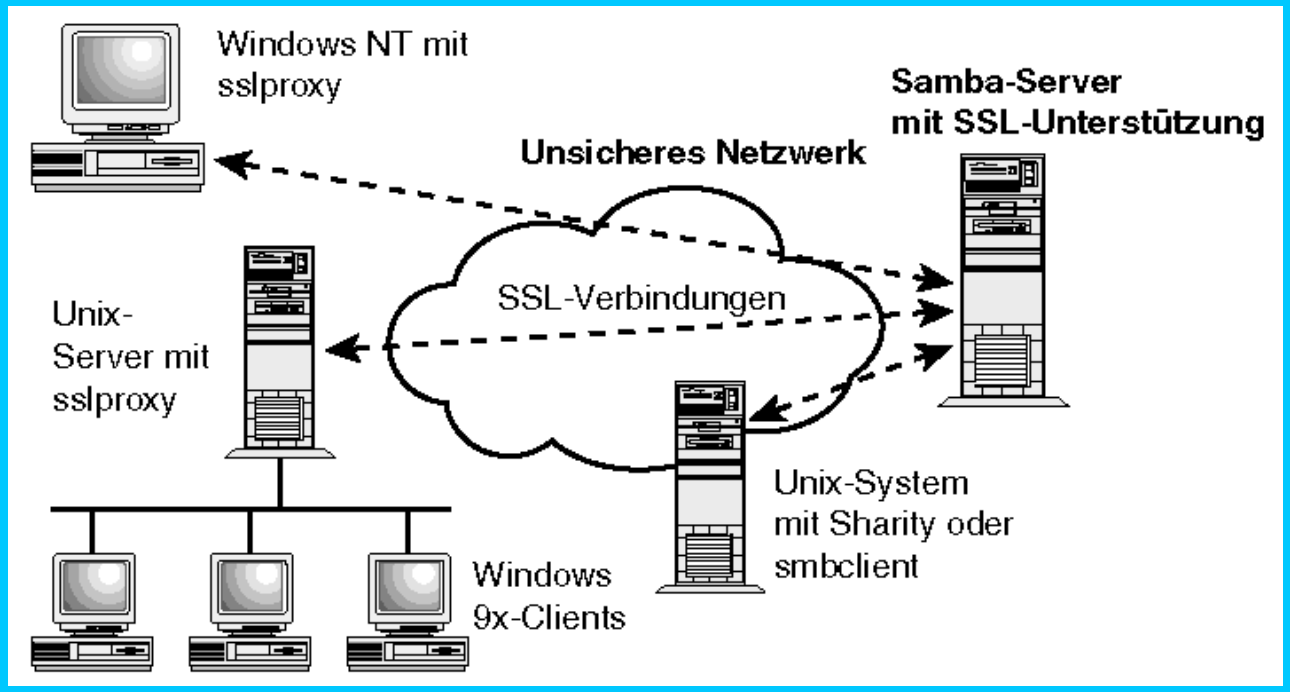
1. Besorgen Sie sich SSLeay, übersetzen Sie es und installieren Sie es. Die aktuellste Version von SSLeay erhalten Sie unter http://www.cryptsoft.com. Auf dieser Site finden Sie auch die SSLeay-FAQ. Diesen Schritt müssen Sie vor der Konfiguration von Samba mit SSL-Unterstützung durchführen, da Samba sonst nicht übersetzt werden kann.
2. Übersetzen Sie Samba mit aktivierter SSL-Unterstützung und installieren Sie es.
3. Besorgen Sie sich oder erstellen Sie X509-Zertifikate mindestens für Ihre Samba-Server und, je nach Bedarf, auch für Client-Rechner, die über ein unsicheres Netzwerk auf Ihren Samba-Server zugreifen werden.
4. Konfigurieren Sie Samba für die Benutzung von SSL, teilen Sie Samba mit, wo sich das Server-Zertifikat und der Private Key befinden, und setzen Sie andere notwendige Parameter.
5. Richten Sie eine Umgebung ein, die SMB-Clients mit SSL-Unterstützung den Zugriff auf Samba ermöglicht.

Um SSL mit Samba benutzen zu können, müssen Sie sowohl Samba als auch Ihren Clients irgendwie Unterstützung für SSL bieten. Sie brauchen nur einige Clients zu finden, die SSL aktiviert haben, und schon kann es losgehen. Leider bieten nur wenige Clients SSL, darunter:

- Sharity, ein Produkt von Object Development, über das Linux und andere Betriebssysteme CIFS/SMB-Dateisysteme mounten können
- `smbclient`, wenn mit SSL-Unterstützung kompiliert
- `sslproxy`, eine Software, die unter der GPL verfügbar ist und es Nicht-SSL-Clients ermöglicht, Zugriff auf SSL-Server zu erhalten

Abbildung 17.1 zeigt, wie ein Samba-Server mit SSL eingesetzt werden kann, damit er sicheren entfernten Zugriff auf Dateien und Drucker über das Internet zur Verfügung stellt. Die Wolke stellt ein unsicheres Netzwerk dar, wie z.B. das Internet, in dem jemand eventuell Ihren übertragenen Datenverkehr abhören kann.

Abb. 17.1: Einen Samba-Server mit SSL-Unterstützung einsetzen



Ohne SSL-Unterstützung gehen CIFS/SMB-Clients einfache TCP-Verbindungen zum Server ein und erhalten über diese Zugriff auf Dateien. Die übertragenen Befehle und Daten sind jedoch für jeden klar sichtbar. Hier benutzen Sie `sslproxy` für Systeme wie Windows NT oder Windows 9x und Sharity oder `smbclient` für Unix-Systeme. Dann werden Befehle und Daten von entfernten Benutzern, die über ein unsicheres Netzwerk übertragen werden, verschlüsselt und daher davor geschützt, von jemandem gesehen zu werden, der die Daten abhören kann.

Standardmäßig wird die SSL-Unterstützung nicht in Samba einkompiliert, und könnte es auch nicht, da SSLeay nicht mit Samba geliefert wird. Das liegt an den Exportbestimmungen der USA. Wenn Samba standardmäßig mit aktivierter SSL-Unterstützung und SSLeay verteilt werden würde, wäre es sehr schwierig, es über in den USA basierten Websites zum Herunterladen zur Verfügung zu stellen. Wenn Sie sich allerdings SSLeay besorgen, es übersetzen und dann Samba mit aktivierter SSL-Unterstützung kompilieren, bekommen Sie vollständige symmetrische 128-Bit-Verschlüsselung, nicht die abgespeckte 40-Bit-Unterstützung, die die aktuellen Exportbestimmungen der USA erlauben.

SSL besorgen und übersetzen

Den Source-Code für SSL erhalten Sie unter `ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL`. Zur Zeit der Bucherfassung war die aktuellste Version SSLeay-0.9.0b. Diese Site liegt in Brisbane, Australien, also sollte es mit dem Herunterladen der Software keine Probleme geben, egal wo Sie sich befinden. Auf alle Fälle werden Kopien der Software auch von anderen Sites rund um die Welt verwaltet. Wenn Sie bereits den Anweisungen in Kapitel 3, »Wie bekomme ich den aktuellsten Source-Code?«, gefolgt sind, sollten Sie SSLeay von der oben angegebenen Website herunterladen können.

Nachdem Sie SSLeay heruntergeladen haben, entpacken Sie es in ein Verzeichnis und übersetzen es. Folgende Schritte müssen dafür in der Regel durchgeführt werden:

```
gzip -d SSLeay-0.9.0b_tar.gz
tar -zvf SSLeay-0.9.0b_tar
cd SSLeay-0.9.0b
```

```
./Configure Betriebssystem und Compiler
make
make install
```

Um zu bestimmen, welche Kombination aus *`Betriebssystem und Compiler`* Sie angeben sollten, geben Sie einfach `./Configure` ein, dann erhalten Sie eine Auflistung aller unterstützten Kombinationen.

Erhalten Sie eine Fehlermeldung wie

```
[ba]sh: ./Configure: No such file or directory
```

wenn Sie versuchen, `./Configure` auszuführen, ist Ihr Perl-Interpreter wahrscheinlich nicht an dem Standort, den `Configure` erwartet. `Configure` ist ein Perl-Skript und erwartet, dass Perl sich in `/usr/local/bin/perl` befindet. Ist Ihr Perl-Interpreter an einem anderen Platz, editieren Sie einfach `Configure` und ändern Sie die erste Zeile (`#!/usr/local/bin/perl`), um den tatsächlichen Standort von Perl in Ihrem System anzugeben.

Außerdem haben viele Systeme einen `tar`-Befehl, der `gzip`-Dateien direkt unterstützt. In diesem Fall können Sie SSLeay mit folgendem Befehl entpacken:

```
tar _zxvf SSLeay-0.9.0b_tar.gz
```

Wenn Sie SSL entpackt und installiert haben, können Sie zum nächsten Schritt übergehen und Samba mit aktivierter SSL-Unterstützung übersetzen. Beachten Sie jedoch, dass die folgenden Anweisungen vorausssetzen, dass Sie SSLeay am Standardstandort installiert haben, d.h. `/usr/local/ssl`. Müssen Sie SSLeay an einem anderen Platz installieren, sollten Sie den Hinweis dazu im nächsten Abschnitt beachten.

Samba mit SSL übersetzen

Wenn Sie SSLeay aufgebaut und installiert haben, können Sie fortfahren, Samba mit SSL-Unterstützung zu übersetzen, aber Sie sollten beachten, dass die SSL-Unterstützung in Samba 2.0.0 fehlerhaft ist. Sie werden sich die aktuellste Samba-Version (2.0.6 oder höher) besorgen müssen. Bitte beachten Sie die Hinweise in Kapitel 3, wie Sie sich die aktuellste Version herunterladen können.

Nachdem Sie sich die Samba-Version 2.0.3 oder höher besorgt haben, sollten Sie folgende Schritte ausführen, um Samba mit SSL-Unterstützung aufzubauen:

```
./configure _with-ssl
make
make install
```

Haben Sie SSL an einem anderen Platz als `/usr/local/ssl` installiert, müssen Sie die erste Zeile wie folgt ändern:

```
./configure _with-ssl --with-sslinc=Standort von SSL
```

Jetzt können Sie eine `smb.conf`-Datei aufbauen, die es Samba ermöglicht, Verbindungen von SSL-Clients zu akzeptieren, aber lassen Sie uns zunächst das SSL-Protokoll näher ansehen, z.B. was Zertifikate sind und wie SSL sie benutzt.

Wenn Samba mit SSL-Unterstützung übersetzt worden und der SSL-Modus aktiviert ist, müssen die Parameter `ssl server cert` und `ssl server key` gültige Werte haben. Außerdem verlangt der `smbd` den Passwortsatz, der für die Verschlüsselung des Server-Schlüssels benutzt wird, wenn er verschlüsselt wird.

Zertifikate und all das

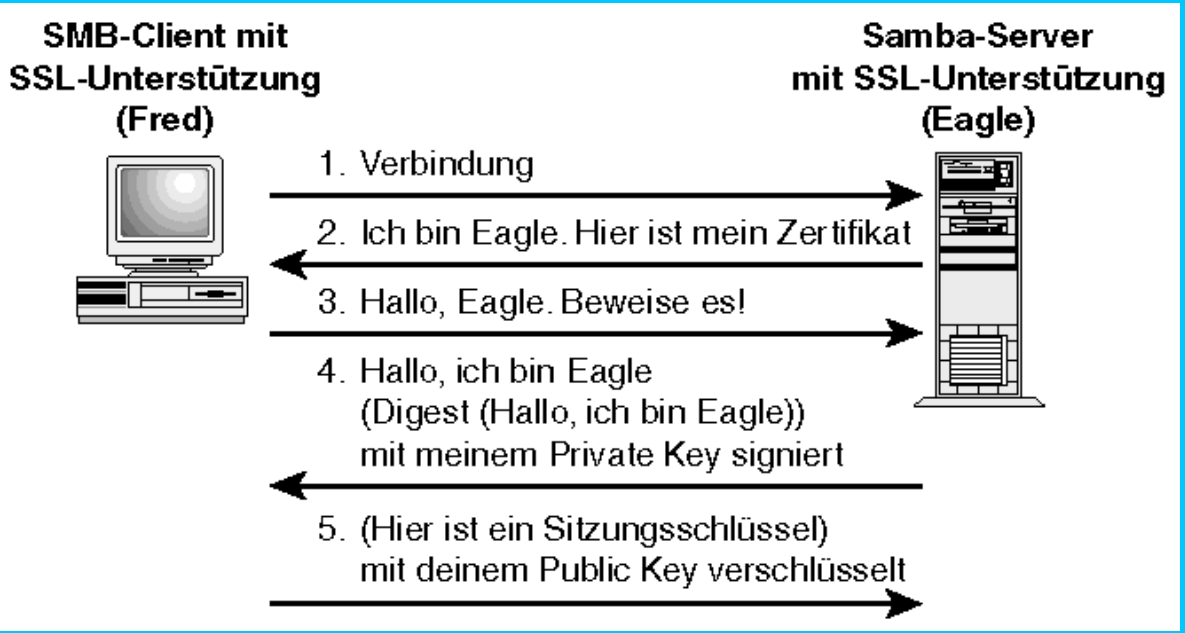
Das SSL-Protokoll verlässt sich auf die Public-Key-Verschlüsselung, damit zwei kommunizierende Parteien einen verschlüsselten Kommunikationskanal zwischen sich aufbauen können. Während der SSL-Verhandlung (Handshake), wird ein zufälliger Schlüssel generiert, der mit der symmetrischen Verschlüsselungsmethode benutzt wird, die wiederum für den tatsächlichen Austausch von Informationen in der SSL-Verbindung verwendet wird. Das bedeutet, wenn der SSL-Handshake abgeschlossen ist, werden keine Public-Key-Algorithmen mehr verwendet. Stattdessen werden Algorithmen wie Triple-DES, IDEA usw. mit dem zufällig gewählten Schlüssel benutzt, der während des SSL-Handshakes übertragen wurde. Die symmetrische Verschlüsselung wird benutzt, weil sie

schneller ist als die Public-Key-Technologie (hundertmal schneller - siehe z.B. Bruce Schneier, *Applied Cryptography*).

SSL geht jedoch noch weiter, da es einer oder beiden Parteien ermöglicht zu überprüfen, ob sie tatsächlich mit der Person kommunizieren, die sich als ihr Kommunikationspartner ausgibt.

Abbildung 17.2 zeigt einen Überblick über das SSL-Protokoll; sie ist ein Auszug aus http://home.netscape.com/products/security/ssl/howitworks.html.

Abb. 17.2: SSL und Zertifikate



In Schritt 1 verbindet sich der Client über TCP mit dem Server. Nach einem ersten Handshake, während dem die Parteien bestimmen, dass SSL benutzt wird und welche Version, sendet der Server in Schritt 2 sein Zertifikat zurück. Das Zertifikat enthält den Public Key von Eagle und ist von einer Zertifizierungsstelle signiert.

In Schritt 2 verlangt der Client, dass Eagle einen Beweis seiner Identität überträgt. Eagle generiert eine Nachricht (Hallo, ich bin Eagle), berechnet einen Digest (MD5) dieser Nachricht und signiert den Digest (d.h. verschlüsselt den Digest mit seinem eigenen Private Key). Wenn der Client die Nachricht erhält, berechnet er ebenfalls den Digest der Klartextnachricht und entschlüsselt den von Eagle übertragenen und signierten Digest mit dem Public Key von Eagle (aus dem Zertifikat). Entspricht der entschlüsselte Digest dem berechneten, kann er nur von Eagle stammen.

In Schritt 5 schließlich generiert der Client einen zufälligen Sitzungsschlüssel, verschlüsselt diesen mit dem Public Key von Eagle und überträgt ihn an Eagle. Nur Eagle sollte den Sitzungsschlüssel entschlüsseln können. Von diesem Punkt an können Eagle und Client nun symmetrische Verschlüsselung benutzen, um Informationen auszutauschen, und sich sicher sein, dass niemand sonst ihre Nachrichten entschlüsseln kann.

Möchte Eagle die Identität des Clients überprüfen, kann er den Client in ähnlicher Art und Weise nach einem Zertifikat fragen. So können beide Parteien ihre Identität bestätigen und sich sicher sein, dass sie tatsächlich mit dem kommunizieren, den sie für ihren Kommunikationspartner halten.

Dieses Protokoll verlässt sich auf die Tatsache, dass festgestellt werden kann, ob der Public Key, den Eagle an den Client Fred übertragen hat, tatsächlich Eagle gehört. Wie wird dies erreicht? Über das Zertifikat, das Eagle während Schritt 2 übertragen hat. So läuft die Prozedur ab:

1. Eagle oder der Systemadministrator auf Eagle generiert ein Schlüsselpaar (Public und Private Keys) für Eagle. Da der Private Key so wichtig ist und niemals anderen preisgegeben werden sollte, wird er gewöhnlich mit einem Passwortsatz verschlüsselt.
2. Der Systemadministrator generiert dann einen *Certificate Signing Request (CSR)*, der identifizierende Informationen über Eagle in Form eines Zertifikats an Eagles Public Key bindet, und sendet diesen an eine Zertifizierungsstelle (Certification Authority - CA) für die Signierung. Diese identifizierenden Informationen bestehen u.a. aus dem eindeutigen Namen des Servers (d.h. seinem DNS-Namen), seinem Standort, der E-Mail-Adresse des Anfragenden usw.
3. Die von Ihnen gewählte Zertifizierungsstelle führt einige Authentifizierungsschritte durch, um irgendwie sicherzustellen, dass der von Ihnen übergebene Public Key tatsächlich für den Rechner gilt, der ihn benutzen wird.
4. Die CA signiert oder verschlüsselt Ihr Zertifikat mit dem Private Key. Dann sendet sie das signierte Zertifikat an Sie zurück. Das Zertifikat ist jetzt von einer Zertifizierungsstelle signiert und bestätigt, dass der darin enthaltene Public Key zu dem Rechner gehört, der im Zertifikat als eindeutiger Name angegeben ist.

5. Sie laden das Zertifikat auf Ihren Server und teilen Ihrem Server mit, wo er sowohl das Zertifikat als auch den dafür relevanten Private Key findet.

Mit dieser Prozedur bestätigt eine dritte Partei (oder Parteien), dass der in dem Zertifikat angegebene Public Key dem Rechner gehört, der das Zertifikat anbietet. Normalerweise wird eine der Komponenten des eindeutigen Namens in dem Zertifikat der DNS-Name des Rechners sein, der das Zertifikat zur Verfügung stellt. Daher werden die Parteien, die ein Zertifikat erhalten, sicherstellen, dass der Name im Zertifikat den DNS-Namen (oder einem von ihnen) des Systems entspricht, von dem das Zertifikat kommt.

Das SSL-Modul in Samba stellt derzeit nur sicher, dass die andere Seite ein gültiges Zertifikat präsentiert (wenn verlangt). Es überprüft keine Entsprechungen zwischen eindeutigem Namen und DNS-Namen. Diese Funktion wird in einer zukünftigen Samba-Freigabe implementiert sein.

Zertifikate erhalten

Bevor Sie Samba mit SSL für Verbindungen von Clients konfigurieren können, müssen Sie Zertifikate für alle Rechner haben, die Zertifikate brauchen. Wenn Samba im SSL-Modus arbeitet, brauchen Sie mindestens ein Server-Zertifikat. Im Gegensatz zu einem Webserver werden Sie aber wahrscheinlich auch Client-Zertifikate haben wollen.

Warum? Nun, wenn ein Benutzer auf einen Webserver zugreift und sensible Informationen eingeben soll, muss der Benutzer wirklich sicher sein, dass der Webserver auch tatsächlich der Rechner ist, der er vorgibt zu sein. Im Fall eines Samba-Servers jedoch greifen normalerweise die Clients auf Informationen auf dem Server zu, die sensibel sein könnten. Sie sollten daher sicher sein können, dass nur autorisierte Clients auf diese Informationen zugreifen. Aus diesem Grund sollten Sie Zertifikate für Clients haben, die über ein unsicheres Netzwerk auf Samba zugreifen.

Zwar können Sie alle benötigten Zertifikate z.B. von Verisign oder Thawte erhalten, aber dies ist eine teure Option, wenn die von Ihnen benötigten Zertifikate sehr eingeschränkte Anwendungsmöglichkeiten haben. Sie werden ja nur an Ihre eigenen Server übermittelt. In diesem Fall werden Sie wahrscheinlich als Ihre eigene Zertifizierungsstelle agieren und eigene Zertifikate signieren wollen.

SSLeay bietet alle notwendigen Funktionen, um als Zertifizierungsstelle zu agieren. So können Sie Ihre eigenen Zertifikate signieren, was sehr sinnvoll ist, da Sie höchstwahrscheinlich die Clients kennen, die diese Zertifikate benutzen.

Die meisten der Prozeduren für die Einrichtung einer CA und die Signierung von Zertifikaten werden in der SSLeay-Dokumentation und im SSLeay-Dokument im Samba-Verzeichnis `docs` detailliert dargestellt, aber auch hier werde ich sie in den folgenden Abschnitten wiederholen.

Richten Sie sich als Zertifizierungsstelle (CA) ein

Dafür brauchen Sie eine Datenbank für alle von der CA signierten Zertifikate und das Schlüsselpaar, das von der CA verwendet wird. Die CA braucht mindestens einen Private Key, um Zertifikate signieren zu können.

Sie sollten für das meiste, was nun folgt, Ihren Pfad eingerichtet haben, damit Sie Programme von `/usr/local/ssl/bin` oder dem Standort, an dem Sie SSL installiert haben, einfach ausführen können. Richten Sie also die Umgebungsvariable `PATH` auf den entsprechenden Standort ein:

```
PATH=$PATH:/usr/local/ssl/bin
```

Dies kann ein anderer Befehl sein, wenn Sie eine andere Shell benutzen (wie z.B. `csh` oder `tcsh`, wo Sie möglicherweise `setenv` benutzen). Konsultieren Sie die entsprechenden Startdateien für Ihre Shell.

Danach initialisieren Sie SSLeays Generator für zufällige Zahlen. Die Initialisierungsinformationen werden in einer Datei namens `.rnd` im Home-Verzeichnis des Accounts gespeichert, von dem Sie arbeiten. Verwenden Sie für die Initialisierung folgende Befehle:

```
cat > /tmp/random.txt
```

Drücken Sie dann ein oder zwei Minuten lang beliebige Tasten auf Ihrer Tastatur. Wenn Sie genügend Zeichen eingegeben haben, drücken Sie `[Strg]+[D]`, um die Datei zu beenden. Initialisieren Sie dann den SSLeay-Generator für zufällige Zahlen mit:

```
ssleay genrsa -rand /temp/random.txt
rm -f /tmp/random.txt
```

Löschen Sie die Datei `random.txt`, da es eventuell möglich ist, Ihren Private Key darüber zu erhalten. Wenn der Befehl `genrsa` ausgeführt wird, werden Sie eine Ausgabe über die Generierung eines RSA-Private-Key sehen. Sie können diese Informationen ignorieren.

Danach müssen Sie die Datenbank einrichten, die von der Zertifizierungsstelle benutzt wird. Diese Datenbank würde sich normalerweise in einem Verzeichnis unter `/usr/local/ssl` befinden (wenn Sie SSLeay dort installiert haben).

Wählen Sie einen Namen für das Verzeichnis, in dem sich die Datenbank für die CA befindet, z.B. `myCA`. Dieser Standort muss in die `ssleay.cnf`-Datei in `/usr/local/ssl/lib` und in das `CA.sh`-Skript eingegeben werden. Bearbeiten Sie zuerst `/usr/local/ssl/lib/ssleay.cnf` und ändern Sie den `dir`-Eintrag im Abschnitt `CA_default` von

```
[ CA_default ] dir    =./demoCA        # Where everything is kept
...
```

auf

```
[ CA default ] dir    =/usr/local/ssl/myCA    # Where everything is kept
...
```

Ändern Sie danach das `CA.sh`-Skript, um auch dort den Standort der CA-Datenbank anzugeben. Sie werden drei Zeilen finden, in denen die Variablen `CATOP`, `CAKEY` und `CACERT` definiert sind. Ändern Sie `CATOP`, damit der Standort der Datenbank angezeigt wird. Z.B:

```
CATOP=/usr/local/ssl/myCA
```

Erstellen Sie dann das Verzeichnis, in dem sich die Datenbank befindet, und setzen Sie dafür die Berechtigungen 0700 (damit niemand auf die Private Keys zugreifen kann):

```
mkdir /usr/local/ssl/myCA
chmod 700 /usr/local/ssl/myCA
CA.sh -newca
chmod -R 700 /usr/local/ssl/myCA
```

Nach Ausführen des Befehls `CA.sh` wird Ihnen eine Reihe von Fragen gestellt, die sich auf den eindeutigen Namen der CA beziehen:

```
[root@myca]#"> [root@myca]# CA.sh -newca
mkdir: cannot make directory `/usr/local/ssl/myCA': File exists
CA certificate filename (or enter to create)

Making CA certificate ...
Using configuration from /usr/local/ssl/lib/ssleay.cnf
Generating a 1024 bit RSA private key
...+++++
.....................................+++++
writing new private key to '/usr/local/ssl/myCA/private/./cakey.pem'
Enter pass phrase: (1)
Verifying password - Enter PEM pass phrase: (2)
-----
You are about to be asked to enter information that will be
incorporated into your certifcate request.
What are you about to enter is what is called a Distinguished Name oder a DN. There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AU (3)
State or Province Name (full name) [Some-State]:South Australia (4)
Locality Name (e.g., city) []:Adelaide (5)
Organization Name (e.g., company) [Internet Widgits Pty Ltd]:NS Widgits Pty Ltd (6)
Organizational Unit Name (e.g., section) []: (7)
Common Name (e.g., YOUR name) []:myserver.mydom.com.au (8)
Email Address []:rsharpe@ns.aus.com (9)
[root@myca]#
```

Beachten Sie Folgendes in Bezug auf die fettgedruckten Zahlen in Klammern im obigen Listing:

1. Geben Sie hier Ihren Passwortsatz für den Private Key der CA ein.
2. Bestätigen Sie hier Ihren Passwortsatz. Vergessen Sie diesen Passwortsatz nicht, sonst müssen Sie den Private Key für die CA erneut generieren.
3. Die verbleibenden Punkte beziehen sich auf den eindeutigen Namen für den Rechner, den Sie definieren. Geben Sie hier den aus zwei Buchstaben bestehenden ISO-Code für Ihr Land ein.
4. Geben Sie hier, wenn vorhanden, Ihren Staat oder die geographische Unterteilung ein, z.B. Südaustralien wie in obigem Beispiel oder Kalifornien, Quebec usw.
5. Geben Sie hier den Namen Ihres Standorts ein, z.B. Adelaide, Fremont, Montreal usw.
6. Geben Sie hier den Namen Ihres Unternehmens ein, z.B. NS Widgits Pty Ltd. Pty Ltd ist übrigens eine rechtliche Bezeichnung für ein Unternehmen in Australien.
7. Geben Sie hier den Namen Ihrer Abteilung ein, z.B. Produktion. In obigem Beispiel wurde dieses Feld leer gelassen.
8. Geben Sie hier Ihren Eigennamen ein, z.B. `myserver.mydom.com.au`. Dies wird oft der Name sein, der mit dem DNS-Namen des Rechners, der das Zertifikat ausgibt, verglichen wird.
9. Geben Sie hier Ihre E-Mail-Adresse ein.

Zertifikate für jeden Rechner generieren

Danach müssen Sie für jeden Rechner ein Zertifikat generieren, der eins braucht. Folgende Schritte müssen ausgeführt werden:

1. Generieren Sie ein Schlüsselpaar für den Client:

   ```
   ssleay genrsa -des3 1024 > client1.pem
   ```

   Dies verlangt, dass ein Schlüsselpaar mit 1024-Bit-Schlüsseln generiert wird. Der Private Key wird über Triple-DES mit einem von Ihnen spezifizierten Passwortsatz verschlüsselt.
2. Generieren Sie einen *Certificate Signing Request (CSR)*

   ```
   ssleay req -new -key client1.pem -out client1-csr.pem
   ```

   Damit wird `client1.pem` in einen CSR konvertiert.

Wenn Sie den CSR generieren, werden Sie aufgefordert, alle Informationen für den eindeutigen Namen des Rechners einzugeben, für den das Zertifikat bestimmt ist. Die hier angegebenen Informationen sollten sich auf den Client beziehen, für den Sie ein Zertifikat generieren. Insbesondere der Eigenname sollte dem DNS-Namen des Clients entsprechen.

Die Zertifikate signieren

Wenn Sie den CSR generiert haben, müssen Sie ihn signieren. Verwenden Sie dafür den folgenden Befehl:

```
ssleay ca -policy your policy 365 \ -infiles client1-csr-pem > client1-cert.pem
```

Jetzt haben Sie ein Zertifikat. Sie müssen den Private Key (`client1.pem`) und das Zertifikat an den Client weitergeben, der beide an einem entsprechenden Platz auf dem Rechner speichert. Normalerweise gehen sie in das SSLeay-Verzeichnis `/usr/local/ssl/certs`. Haben Sie allerdings SSLeay an einem anderen Ort installiert, platzieren Sie sie in das `certs`-Verzeichnis an diesem Standort. Wenn Sie ein anderes Paket als SSLeay verwenden, sollten Sie in der Dokumentation für Ihr Paket Informationen dazu finden, wo Sie Zertifikate ablegen.

Sicherstellen, dass das CA-Zertifikat auf jedem Rechner vorhanden ist, der SSL benutzt

Um das übertragene Zertifikat zu überprüfen, braucht SSL das Zertifikat der CA. Dies wird normalerweise an einem bekannten Ort gespeichert, obwohl Sie den meisten Anwendungen auch mitteilen können, wo sich diese Bereiche befinden.

Eine der Dateien, die generiert wurden, als wir vorher die CA erstellt haben, war eine Datei namens `cacert.pem`. Dies ist das Zertifikat der CA. Die folgenden Schritte platzieren das Zertifikat der CA in das entsprechende Verzeichnis und erstellen einen gehashten Namen:

```
cp /usr/local/ssl/myCA/cacert.pem /usr/local/ssl/certs/myCA.pem
cd /usr/local/ssl/certs
ln -s myCA.pem `ssleay x509 -noout -hash < myCA.pem`.0
```

Wenn Sie einen gehashten Namen erstellen, können Sie Zertifikate für mehr als eine CA im Verzeichnis `certs` aufbewahren, und SSL kann das entsprechend ausprobieren, um ein angebotenes Zertifikat zu überprüfen.

Generell würden Sie diese Schritte auf jedem Client-Rechner ausführen, der Zertifikate überprüft, außer dass `cacert.pem` von einem anderen System kommt.

Samba für die Benutzung von SSL konfigurieren

Wenn Sie Samba mit SSL-Unterstützung übersetzen, werden Sie bemerken, dass eine ganze Reihe neuer `smb.conf`-Parameter in den Manpages für `smb.conf` auftaucht. Damit Samba die SSL-Unterstützung, die Sie in Samba einkompiliert haben, verwenden kann, müssen Sie einige dieser Parameter entsprechend einrichten.

Damit Samba SSL benutzt, müssen Sie mindestens die Parameter `ssl`, `ssl server cert` und `ssl server key` setzen.

Diese Parameter gehen alle in den globalen Abschnitt Ihrer `smb.conf`. Hier ist ein Beispiel für die Einrichtung der Parameter:

```
ssl = yes
ssl server cert = /root/keys/real-cert.pem
ssl server key = /root/keys/key.pem
```

Diese Einträge aktivieren SSL, spezifizieren, wo sich die Zertifikate der CA befinden, und teilen Samba mit, wo sein Server-Zertifikat und sein Private Key sind.

In den folgenden Abschnitten werden alle SSL-relevanten Parameter dargestellt. Diese Parameter sind nur verfügbar, wenn Samba mit SSL kompiliert wurde.

ssl

Dieser globale Parameter aktiviert die SSL-Unterstützung, die in Samba einkompiliert wurde. Wenn Sie SSL mit diesem Parameter aktivieren, müssen Sie die Parameter `ssl server cert` und `ssl server key` ebenfalls setzen. Standardmäßig ist SSL deaktiviert, auch wenn Sie es in Samba einkompiliert haben.

Hier ist ein Beispiel, das Ihnen zeigt, wie Sie SSL aktivieren:

```
ssl = yes
```

ssl CA certDir

Dieser globale Parameter spezifiziert den Standort des Verzeichnisses, das die Zertifikate für alle Zertifizierungsstellen enthält, denen Ihr Unternehmen oder Ihr System vertraut.

Der Dateiname für jedes CA-Zertifikat in diesem Verzeichnis ist tatsächlich der Hash-Wert, der aus dem eindeutigen Namen im Zertifikat ermittelt wird. Details über die Generierung dieser Hash-Werte finden Sie früher in diesem Kapitel im Abschnitt über die Sicherstellung, dass sich das CA-Zertifikat auf jedem Rechner befindet, der SSL benutzt. Eine andere Methode, die verlangten Informationen einzurichten, besteht in der Verwendung des Parameters `ssl CA certFile`. Normalerweise verwenden Sie entweder den einen oder den anderen Parameter, nicht beide.

Sie brauchen diesen Parameter nicht zu verwenden, wenn Client-Zertifikate nicht überprüft werden.

Standardmäßig hat dieser Parameter keinen Wert.

Das folgende Beispiel zeigt, wie Sie das Verzeichnis, das die CA-Zertifikate enthält, auf `/usr/local/ssl/certs` einrichten:

```
ssl CA certDir = /usr/local/ssl/certs
```

Erhält ein Client ein Zertifikat, sucht Samba in diesem Verzeichnis nach dem Zertifikat der CA.

ssl CA certFile

Dieser globale Parameter ist eine alternative Methode für die Spezifizierung des Standorts vertrauenswürdiger CAs. Über diesen Parameter spezifizieren Sie eine Datei, die alle Zertifikate vertrauenswürdiger CAs enthält, die verkettet sind.

Verwenden Sie diesen Parameter, wenn Sie nur ein Zertifikat haben und keine Hash-Werte für CA-Zertifikate generieren wollen.

Eine andere Methode, die verlangten Informationen einzurichten, besteht in der Verwendung des Parameters `ssl CA certDir`. Normalerweise verwenden Sie entweder den einen oder den anderen Parameter, nicht beide.

Sie brauchen diesen Parameter nicht zu verwenden, wenn Client-Zertifikate nicht überprüft werden.

Standardmäßig hat dieser Parameter keinen Wert.

Das folgende Beispiel zeigt, wie Sie die Zertifikatsdatei auf `/usr/local/ssl/certs/myCAcert.pem` einrichten:

```
ssl CA certFile = /usr/local/ssl/certs/myCAcert.pem
```

Erhält ein Client ein Zertifikat, sucht Samba in dieser Datei nach dem Zertifikat der CA.

ssl ciphers

Über diesen globalen Parameter können Sie Samba mitteilen, welche Verschlüsselungsmethoden es benutzen kann. Sie sollten diesen Parameter nur verwenden, wenn Sie die Standardschlüssel außer Kraft setzen müssen, die SSLeay während der Verhandlungsphase bietet, und genau wissen, was Sie tun.

Folgende Werte können für diesen Parameter benutzt werden:

- `DEFAULT`
- `DES-CFB-M1`
- `NULL-MD5`
- `RC4-MD5`
- `EXP-RC4-MD5`
- `RC2-CBC-MD5`
- `EXP-RC2-CBC-MD5`
- `IDEA-CBC-MD5`
- `DES-CBC-MD5`
- `DES-CBC-SHA`
- `DES-CBC3-MD5`
- `DES-CBC3-SHA`
- `RC4-64-MD5`
- `NULL`

Weitere Details hierzu finden Sie im SSLeay-Source-Code oder den SSL-Spezifikationen (http://www.netscape.com/info/SSL.html).

Ein Beispiel für die Verwendung dieses Parameters:

```
ssl ciphers = DEFAULT
```

ssl client cert

Dieser globale Parameter teilt `smbclient` mit, welches Zertifikat es benutzen soll, wenn der Server ein Client-Zertifikat verlangt. Dies ist derzeit die einzige Methode, für `smbclient` ein Zertifikat zu spezifizieren. Wenn Sie also mehrere Zertifikate für `smbclient` haben, werden Sie die Datei `smb.conf` editieren müssen, um sie zu verwenden.

Der Standardwert für diesen Parameter ist:

```
ssl client cert = /usr/local/ssl/certs/smbclient.pem
```

ssl client key

Dieser globale Parameter teilt `smbclient` mit, wo sich der Private Key für `smbclient` befindet. Dieser ist nur notwendig, wenn der Server ein Client-Zertifikat verlangt und eins für `smbclient` spezifiziert wurde.

Der Standardwert für diesen Parameter ist:

```
ssl client key = /usr/local/ssl/certs/smbclient.pem
```

ssl compatibility

Über diesen globalen Parameter können Sie bestimmen, ob SSLeay Bug-Kompatibilität mit anderen SSL-Implementierungen bieten sollte. Da es derzeit keine Clients mit anderen SSL-Implementierungen gibt, werden Sie diesen Parameter nicht benötigen. Der Standardwert für diesen Parameter ist:

```
ssl compatibility = no
```

ssl hosts

Dieser globale Parameter spezifiziert die Hosts, für die SSL-Verbindungen verlangt werden, wenn der SSL-Modus aktiviert ist. Sie können Hosts nach IP-Adresse, Adressbereich, Netzgruppe oder Namen auflisten.

Dieser Parameter ist mit dem Parameter `ssl hosts resign` verwandt, und zwar insofern als SSL-Verbindungen für alle Client-Verbindungen verlangt werden, wenn keiner der beiden Parameter

gesetzt und der SSL-Modus aktiviert ist. Es gibt keinen Standardwert für diesen Parameter.

Ein Beispiel für die Verwendung dieses Parameters ist:

```
ssl hosts = host1 host2
```

Dies bestimmt, dass nur für die mit dem Parameter aufgelisteten Hosts (z.B. `host1` und `host2`) SSL-Verbindungen verlangt werden. Alle anderen Hosts können Nicht-SSL-Verbindungen eingehen.

ssl hosts resign

Dieser globale Parameter spezifiziert die Hosts, für die keine SSL-Verbindungen verlangt werden, wenn der SSL-Modus aktiviert ist. Sie können Hosts nach IP-Adresse, Adressbereich, Netzgruppe oder Namen auflisten.

Dieser Parameter ist mit dem Parameter `ssl hosts` verwandt, und zwar insofern als SSL-Verbindungen für alle Client-Verbindungen verlangt werden, wenn keiner der beiden Parameter gesetzt und der SSL-Modus aktiviert ist. Es gibt keinen Standardwert für diesen Parameter.

Ein Beispiel für die Verwendung dieses Parameters ist:

```
ssl hosts resign = host1 host2
```

Dies bestimmt, dass für die mit dem Parameter aufgelisteten Hosts (z.B. `host1` und `host2`) keine SSL-Verbindungen verlangt werden. Alle anderen Hosts müssen SSL benutzen.

ssl require clientcert

Dieser globale Parameter bestimmt, ob der Server vom Client verlangt, Zertifikate zu präsentieren. Ist er auf `yes` gesetzt, müssen Clients gültige Zertifikate übertragen, und der Server benutzt die Informationen, die im Parametern `ssl CA certDir` oder `ssl CA certFile` gegeben sind, um die CA zu überprüfen, die die vom Client übertragenen Zertifikate ausgegeben hat.

Kann das Zertifikat eines Clients nicht authentifiziert werden, wird die Verbindung beendet.

Ist dieser Parameter auf `no` (die Standardeinstellung) gesetzt, brauchen Clients keine Zertifikate.

ssl require servercert

Dieser globale Parameter bestimmt, ob `smbclient` ein Zertifikat vom Server verlangt. Ist er auf `no` gesetzt (die Standardeinstellung), verlangt `smbclient` bei Verbindungsaufnahme kein Zertifikat vom Server. Ist er auf `yes` gesetzt, verlangt `smbclient` ein Zertifikat vom Server und benutzt die Informationen, die im Parametern `ssl CA certDir` oder `ssl CA certFile` gegeben sind, um die CA zu überprüfen, die das übertragene Zertifikat ausgegeben hat.

ssl server cert

Dieser globale Parameter spezifiziert den Standort des Zertifikats für den Samba-Server (`smbd`). Dieser Parameter *muss* gesetzt werden, wenn der SSL-Modus aktiviert ist (standardmäßig ist er nicht gesetzt).

Die Datei, die das Zertifikat des Servers enthält, kann auch den Private Key für den Server enthalten.

Ein Beispiel für die Verwendung des Parameters ist:

```
ssl server cert = /root/keys/samba-cert.pem
```

Dies teilt Samba mit, dass das Server-Zertifikat in der Datei `/root/keys/samba-cert.pem` gefunden werden kann. Wenn ein Client eine SSL-Verbindung zu Samba eingeht, händigt es die Inhalte dieser Datei als Zertifikat aus.

ssl server key

Dieser globale Parameter spezifiziert den Standort des Private Key für den Samba-Server (`smbd`). Dieser Parameter *muss* gesetzt werden, wenn der SSL-Modus aktiviert ist (standardmäßig ist er nicht gesetzt). Außerdem muss der Public Key im Zertifikat zum Private Key passen.

Wenn `smbd` im SSL-Modus startet, holt es sich den Private Key, der über diesen Parameter spezifiziert ist. Ist der Private Key durch einen Passwortsatz verschlüsselt, fährt `smbd` nicht fort, bis der Passwortsatz eingegeben ist. Sicherheit ist eine gute Begründung für die Verschlüsselung des Private Key für den Server, aber sie führt zu einigen Schwierigkeiten, wenn Sie den Server neu starten.

Ein Beispiel für die Verwendung des Parameters ist:

```
ssl server key = /root/keys/samba-cert.pem
```

Dies teilt Samba mit, dass der Private Key in der Datei `/root/keys/samba-cert.pem` ist. Normalerweise werden Sie aufgefordert, einen Passwortsatz einzugeben, um diese Datei zu

entschlüsseln.

ssl version

Dieser globale Parameter spezifiziert die Version von SSL, die Samba benutzt, wenn der SSL-Modus aktiviert ist. Folgende Werte können Sie für diesen Parameter einstellen:

ssl2 SSL v2 wird benutzt
ssl3 SSL V3 wird benutzt
sslor3 Ermöglicht Verhandlung über die Verwendung von SSL V2 oder SSL V3
tls1 TLS V1 wird benutzt

Der Standardwert für diesen Parameter ist:

```
ssl version = ssl2or3
```

sslproxy

`sslproxy` ist ein Proxy, der zwischen einem Client ohne SSL-Unterstützung und einem SSL-Server sitzt. Er bietet SSL-Zugriff für den Client ohne SSL-Unterstützung und kann in zwei verschiedenen Modi operieren:

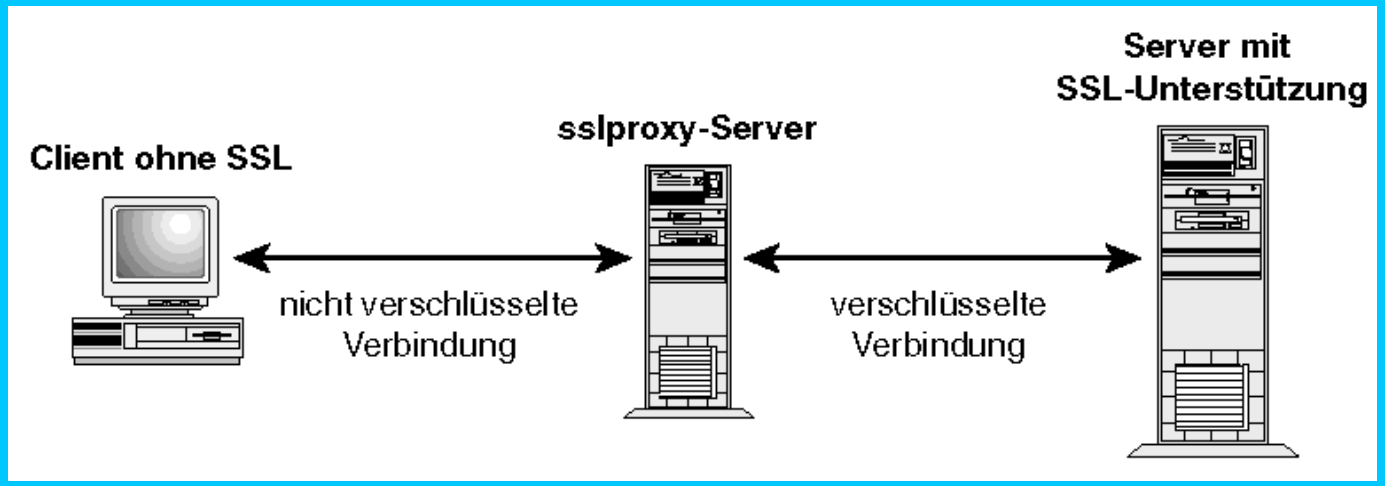
- Transparenter Modus, in dem er SSL-Dienste und Verschlüsselung/Entschlüsselung transparent für den Client ohne SSL-Unterstützung bietet.
- NetBIOS-Modus, in dem er erkennt, dass der Client mit einem NetBIOS-Server kommuniziert und SSL in der Art und Weise initiiert, die Samba erwartet.

Abbildung 17.3 zeigt, wie `sslproxy` benutzt werden könnte. Beachten Sie, dass `sslproxy` entweder auf den Client- oder den Server-Rechner platziert werden kann (mit der entsprechenden Einrichtung), aber ich zeige es hier auf einem separaten System. Wenn Sie `sslproxy` auf dem Client-System platzieren, verbinden sich Client-Programme normalerweise mit `127.0.0.1`.

Wenn Sie `sslproxy` benutzen, verbinden sich Nicht-SSL-Clients mit dem System, auf dem `sslproxy` läuft, statt mit dem SSL-Server.

`sslproxy` wurde von Christian Starkjohan geschrieben und wird in Source-Form verteilt. Sie erhalten `sslproxy` unter http://www.obdev.at/Products. Sie können es auf Unix-Rechnern und unter Windows NT übersetzen. Dazu dekomprimieren Sie einfach die Source-Datei, entpacken das Archiv und führen den Befehl `make` aus, während Sie sich im Source-Verzeichnis befinden.

Abb. 17.3: sslproxy benutzen



Beispiele

Sie haben sich nun detailliert angesehen, wie Sie Samba mit SSL-Unterstützung konfigurieren, wie Sie SSL unter Samba konfigurieren, wie Sie Zertifikate erstellen usw. Aber was ändert sich eigentlich, wenn SSL-Unterstützung verwendet wird? Dieser Abschnitt wirft einen kurzen Blick auf die Änderungen.

Nachfolgend sehen Sie ein Beispiel für das Starten von Samba, wenn der SSL-Modus aktiviert und korrekt konfiguriert wurde:

```
[root@bigpc]# /etc/rc.d/init.d/smb start
Starting SMB services: smbd Enter PEM pass phrase: (1)
nmbd

[root@bigpc]# ps ax | grep mbd (2)
446  ?  S  0:00 smbd -D
455  ?  S  0:00 nmbd -D
[root@bigpc]#
```

Folgendes sollten Sie in Bezug auf die fettgedruckten Zahlen in Klammern beachten:

1. Der `smbd` fragt nach dem Passwortsatz für den Private Key, der über den Parameter `ssl server key` spezifiziert ist. Sie müssen diesen eingeben, bevor `smbd` weitermacht. In der Regel führt dies dazu, dass Ihr Systemstart hängt, und Sie sollten vielleicht andere Methoden benutzen, um Samba zu starten, wenn Sie SSL aktiviert haben und nicht wollen, dass Ihr System während des Starts hängt. (Was ist, wenn niemand sieht, dass Ihr System neu startet?)
2. Sie überprüfen, ob `smbd` gestartet wurde, und es wurde gestartet.

An diesem Punkt wurde Samba mit SSL-Unterstützung gestartet. Wie sieht es aus, wenn Sie versuchen, sich von `smbclient` auf dem gleichen Rechner zu verbinden?

```
[root@bigpc]# smbclient //bigpc/fred (1)
Added interface ip=16.153.112.65 bcast=16.153.112.255 nmask=255.255.255.0
SSL: Certificate OK: /C=AU/ST=South Australia/L=Adelaide/O=NS Widgets/CN=Richard Sharpe/Email=rsharpe@ns.com.au (2)
SSL: Certificate OK: /C=AU/ST=South Australia/L=Adelaide/O=NS Widgets/CN=Richard Sharpe/Email=rsharpe@ns.com.au
SSL: negotiated cipher: DES-CBC3-SHA (3)
Password: (4)
Domain=[FOWLPLAY] OS=[Unix] Server=[Samba 2.0.0]
smb: \> ls
   smb.conf        501 Sun Jan 24 21:21:08 1999

    61945 blocks of size 16384. 58392 blocks available
smb: \>
```

Folgendes sollten Sie in Bezug auf die fettgedruckten Zahlen in Klammern beachten:

1. Sie sollten `smbclient` in der gleichen Art und Weise wie sonst starten.
2. Wenn `smbclient` sich mit dem Server verbindet, präsentiert es ein Zertifikat, das an den Benutzer übertragen wird.
3. Die SSL-Routinen teilen uns auch mit, welcher Schlüssel gewählt wurde.
4. Von diesem Punkt an ist alles wie gehabt.

Um schließlich `sslproxy` zu benutzen, damit Clients ohne SSL-Unterstützung über SSL auf Samba zugreifen können, müssen Sie folgendes Setup durchführen:

1. Starten Sie `sslproxy` auf dem System, auf dem es laufen soll, mit mindestens den folgenden Befehlszeilenoptionen:

   `sslproxy -l 139 -R` *`Server`* -r 139 -n -c *`Zertifikatsdatei`*

   Sie müssen eventuell den vollständigen Pfad zu `sslproxy` angeben.
2. Benutzen Sie Ihren Client, um sich mit dem `sslproxy`-System zu verbinden statt mit dem SSL-Server. Dafür müssen Sie möglicherweise den Namen des `sslproxy`-Systems in Ihre `lmhosts`-Datei einfügen, damit Sie den NetBIOS-Namen des `sslproxy`-Systems in seine IP-Adresse auflösen können.

Zusammenfassung

Sie haben sich in diesem Kapitel detailliert angesehen, wie Sie SSL-Unterstützung in Samba implementieren können. Diese Unterstützung ermöglicht Ihnen, sicheren Zugriff auf Samba-Server über unsichere Netzwerke zu bieten. Wenn Sie aber solchen Zugriff über unsichere Netzwerke nicht zur Verfügung stellen müssen, sollten Sie SSL-Unterstützung besser nicht aktivieren.

Teil V, »Samba für Fortgeschrittene«, behandelt Themen wie WINS-Unterstützung, Browsing in lokalen Subnetzen und Browsing in Router-Netzwerken.

Frage und Antwort

F. Wenn Microsoft keine SSL-CIFS/SMB-Clients bietet, wie kann ich dann die SSL-Unterstützung in Samba benutzen?

. Sie könnten `smbclient` für einfachen Befehlszeilenzugriff auf Samba über SSL verwenden. Für Windows-NT-Clients können Sie `sslproxy` kompilieren und dies zwischen Ihrem Windows-NT-Server und Samba einsetzen. Für Windows-9x-Clients müssen Sie `sslproxy` auf einem separaten Server laufen lassen. Beispiele sind in Abbildung 17.3 dargestellt.

F. Braucht man normalerweise Client-Zertifkate, wenn man SSL mit Samba benutzt?

. Ja. Tatsächlich ist das Server-Zertifikat, das Samba bietet, viel weniger wichtig als Client-Zertifikate. Das liegt daran, dass Sie normalerweise sicher sein wollen, dass nur autorisierte Benutzer Samba über ein unsicheres Netzwerk benutzen können. Wenn Sie Zertifikate an Clients ausgeben und ihre Benutzung verlangen, erhalten Sie ein gewisses Maß an Sicherheit, dass nur autorisierte Benutzer auf Ihren Server zugreifen (sie müssen ein Zertifikat haben). Beachten Sie, dass mit der derzeitigen Implementierung von SSL in Samba lediglich ein gültiges Zertifikat verlangt wird. Es gibt keine Überprüfungen, ob der Client wirklich der ist, der auf dem Zertifikat angegeben ist.

F. Müssen wir Zertifikate von einer Zertifizierungsstelle wie Verisign (http://www.verisign.com) oder Thawte (http://www.thawte.com) besorgen?

. Es ist nicht klar, ob eine der Zertifizierungsstellen Zertifikate für Samba signiert. Wie auch immer, Sie brauchen nicht wirklich ein Zertifikat von einer der kommerziellen Zertifizierungsstellen, da Sie keine Verbindungen zur Öffentlichkeit erlauben werden.

Da Sie Zertifikate an die Clients ausgeben, die auf Ihren Samba-Server zugreifen können, können Sie leicht Ihre eigenen Zertifikate signieren und Ihr CA-Zertifikat auf allen Systemen installieren, die es brauchen.

Tag 18: NetBIOS-Namen ohne Broadcasts auflösen

Es ist endlich passiert! Meine Chefin hat sich einverstanden erklärt, eine weitere Person für die Administration des Netzwerks einzustellen.

»Vielleicht kann ich jetzt einiges verbessern, statt den ganzen Tag lang nur Probleme zu behandeln«, denke ich mir, als das Flugzeug landet. Meine Chefin fand es eine gute Idee, mich und meinen neuen Kollegen zur aktuellsten Konferenz über die Integration von Unix und Windows zu schicken. Normalerweise freue ich mich auf diese Konferenzen. Sie sind ziemlich interessant, und ich treffe immer irgendwelche alten Freunde, die auch an der Konferenz teilnehmen.

Am nächsten Morgen gehen wir zwei gerade die paar Treppen hoch, die zum Konferenzraum führen, in dem die erste Sitzung stattfindet, als ich eine Stimme von hinten höre, die meinen Namen ruft. Ich drehe mich um und sehe das vertraute Gesicht von Ian.

»Schön dich zu treffen«, sage ich begeistert, als wir uns kräftig die Hände schütteln. »Wie geht es Dir?«

Ian antwortet: »Gut. Beschäftigt wie immer. Immer noch an der CMU. Und du?«

»Genau das Gleiche, abgesehen von der CMU«, ich lächle. »Ian, lass mich dir Gary Danz vorstellen. Er ist seit ein paar Wochen bei uns.«

Gary streckt seine Hand aus und sagt ein paar höfliche Begrüßungsworte.

»Schön, dich kennenzulernen, Gary«, antwortet Ian. »Hey, wir sollten losgehen. Ich möchte einen Platz in der Nähe der Kaffeekanne und der Bagels haben.« Wir Drei gehen zur Tür des Konferenzraums und lachen über irgendeinen schlechten Witz, den wir sicher vorher schon einmal irgendwo gehört haben.

Ich habe mir oft gedacht, dass es toll wäre, ein fotografisches Gedächtnis zu haben und in der Lage zu sein, ohne Zögern Namen und Gesichter zuzuordnen. Oder vielleicht eine Brille zu haben, die eine visuelle Suche durch irgendeine Online-Datenbank durchführt und die Gesichter der Leute sucht, die ich schon einmal getroffen habe, aber an deren Namen ich mich nicht erinnern kann. Ein solcher *Namensdienst* wäre sicher bei Leuten mit schlechtem Namensgedächtnis äußerst beliebt. Wenn Sie darüber nachdenken, habe ich in der gerade erzählten Geschichte als eine Art Namensdienst agiert, als ich Ian Gary vorstellte.

Als generelle Regel gilt, dass Menschen andere Menschen mit einem Namen ansprechen, aber es ist unmöglich, sich an die Namen aller Menschen in der Welt zu erinnern oder diese auch nur kennen zu lernen. Rechner in einem Netzwerk treffen auf das gleiche Problem. Zwar ist es möglich, die Namen aller Rechner im Internet zu kennen, aber das Durchsuchen einer einzigen globalen Hosts-Datei in Echtzeit wäre weder praktikabel noch effizient. Außerdem ändern Administratoren von Zeit zu Zeit die Namen der Rechner; daher ist es auch ein Problem, die Daten aktuell zu halten.

Dieses Kapitel legt seinen Schwerpunkt auf Nameserver und die Dienste, die diese zur Verfügung stellen. Ich werde mich vor allem darauf konzentrieren, wie Samba den *Windows Internet Name Service (WINS)* und die Implementierung eines NetBIOS-Nameservers benutzt. Abschließend werde ich darstellen, wie DNS die Auflösung von NetBIOS-Namen beeinflusst und welche Parameter für die Konfiguration dieses Verhaltens in Samba verfügbar sind.

WINS

In Kapitel 2, »Windows-Netzwerke«, habe ich auch das Konzept eines *NetBIOS-Nameservers (NBNS)* erwähnt, als ich NetBIOS-Namen und die Auflösung der Namen in Netzwerkadressen durch die Clients darstellte. Diese Idee wird im RFC 1001 präsentiert, in dem die Theorie bezüglich NetBIOS über TCP definiert ist.

Der RFC stellt zwei Typen von Nameservern dar. Der erste agiert als Bulletin Board für Clients, auf dem Namen hinterlegt und entfernt werden, ohne dass eine Authentifizierung der Namen selbst stattfindet. Am anderen Ende des Spektrums finden Sie einen Server, der möglicherweise volle Authentifizierung für jede Anfrage zur Registrierung eines Namens bietet. Diese Authentifizierung stellt sicher, dass es zu keinen Überschneidungen der Namen kommt, wenn sie für den anfragenden Client ausgeführt wird. Ich werde dies in einem kleinen Moment ausführlicher darstellen. WINS ist eine Implementierung des letzten Typs von NBNS.

Windows NT 4.0 Server wird mit einem WINS-Server geliefert, der installiert werden kann. Samba hat ebenfalls die Fähigkeit, als WINS-Server zu agieren. Allerdings implementiert Samba derzeit das Protokoll nicht für die Replikation einer WINS-Datenbank mit einem anderen WINS-Server, sei es Windows NT oder Samba.

Dies wirkt sich so aus, dass Sie, wenn Sie Samba als Ihren WINS-Server wählen, keine Möglichkeiten haben, einen Ausfall zu überbrücken, sollte der Samba-Server abstürzen. In der Praxis verwende ich Samba als den einzigen WINS-Server in einem Unternehmen, das über 400 Clients hat, verteilt auf mindestens 15 bis 20 Subnetze. Jeder Client verlangt, dass der WINS-Server verfügbar ist, um korrekt funktionieren zu können. Der Samba-Rechner ist unglaublich stabil. Obwohl es also technisch korrekt ist, dies als einen einzelnen Ausfallpunkt zu bezeichnen, möchte ich das nicht mit Instabilität gleichsetzen.

Sie werden in Kapitel 19, »Browsing in lokalen Subnetzen», sehen, dass ein WINS-Server nicht das Gleiche ist wie ein Browse-Master, obwohl diese zwei oft verwechselt werden.

Wofür brauchen Sie WINS?

Sie werden sich aus Kapitel 2 erinnern, dass ein NetBIOS-Client Namen über zwei Methoden registriert und auflöst: Broadcast (an jeden Rechner im Subnetz gesendet) oder Unicast (an einen bestimmten Rechner gesendet). Ein Problem, das bereits erwähnt wurde, ist die Menge des Broadcast-Datenverkehrs, die eine große Anzahl von NetBIOS-Clients generieren kann. Ein anderes Problem besteht darin, dass Broadcasts standardmäßig auf ein logisches Subnetz limitiert sind. Das bedeutet, dass Hosts in verschiedenen Subnetzen nicht miteinander kommunizieren können.

WINS wurde entwickelt, um diese beiden Probleme zu lösen. Erstens wird Broadcast-Datenverkehr wesentlich reduziert, indem Clients so konfiguriert werden, dass sie einen Namen über einen einzelnen Rechner registrieren und auflösen (siehe Abbildung 18.1). Zweitens können NetBIOS-Cients in verschiedenen Subnetzen Namen im gleichen Namensbereich registrieren und auflösen, da Anfragen für einen Namen nun direkt an einen Rechner übertragen werden. Abbildung 18.2 zeigt, wie das funktioniert. Zunächst registriert der Client `NACHO` seinen Namen beim WINS-Server. Dann bittet `QUESO` den WINS-Server, den Namen `NACHO` für ihn aufzulösen. Der Server antwortet mit der IP-Adresse, die vom ersten Rechner zuvor registriert wurde. Für die Konsolidierung mehrerer Subnetze, sei es für das Browsing oder für den Zugriff auf Remote-Server, wird WINS zur Voraussetzung.

Abb. 18.1: Namensregistrierung über (a) IP-Broadcasts und (b) Point-to-Point mit einem WINS-Server

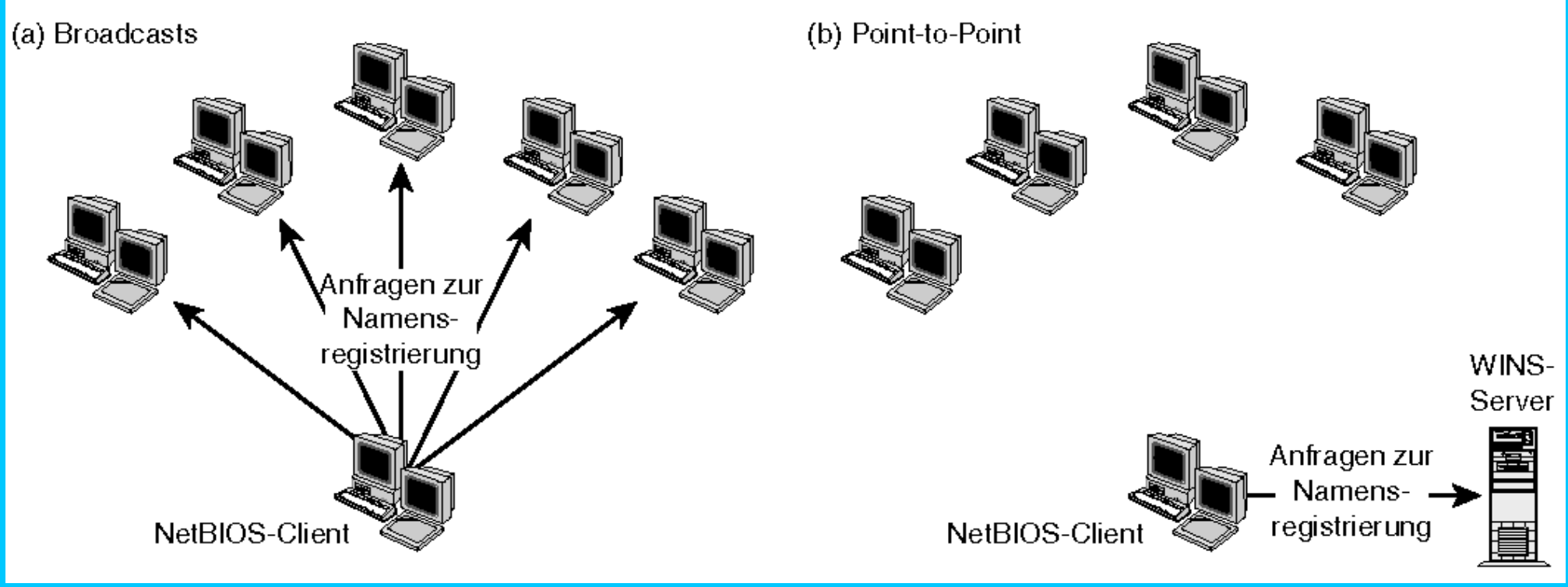
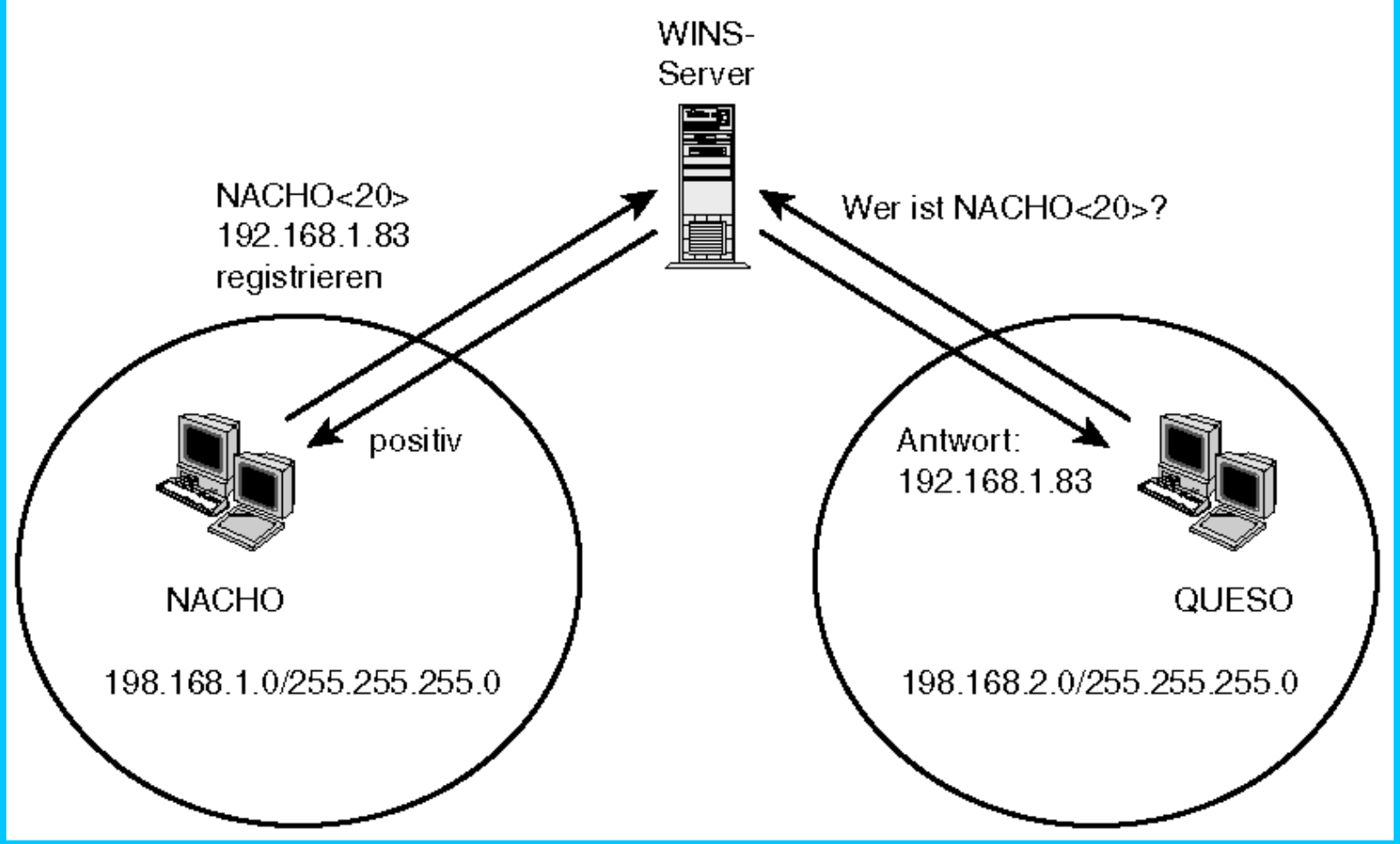


Abb. 18.2: Die Benutzung eines WINS-Servers ermöglicht Clients in verschiedenen Subnetzen die Existenz im gleichen Namensbereich



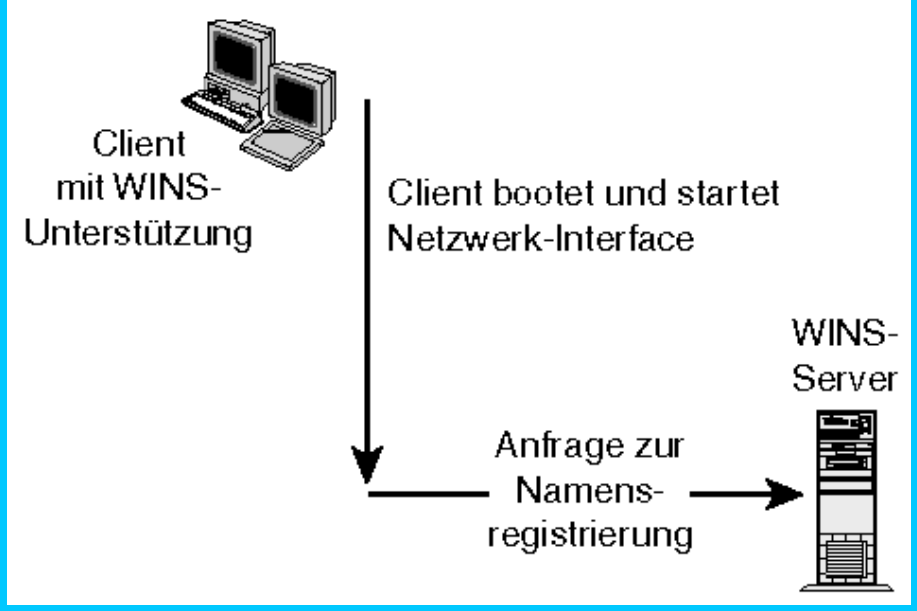
Um diese zwei Probleme besser zu verstehen, lassen Sie uns anschauen, wie NetBIOS-Clients mit einem WINS-Server interagieren. Sehen wir uns zuerst die Namensregistrierung an.

Wenn ein Client mit WINS-Unterstützung in das Netzwerk bootet, sendet er eine Anfrage zur Namensregistrierung direkt an den WINS-Server, wie in Abbildung 18.3 dargestellt. Die Anfrage für die Namensregistrierung ist im Wesentlichen die gleiche wie eine Broadcast-Namensanfrage, mit der Ausnahme, dass sie Point-to-Point vom Client an den WINS-Server überträgt.

Der WINS-Server kann eine von drei Antworten wählen:

- Keine Antwort - Der Server ist nicht aktiv oder so konfiguriert, dass er Pakete vom Host ignoriert.
- Positiv - Der Server findet keine übereinstimmenden Einträge in seiner Datenbank, die eine Namensüberschneidung mit einem anderen Client anzeigen. Daher kann der Client den Namen erfolgreich registrieren, und der Eintrag wird in der WINS-Datenbank gespeichert.
- Negativ - Wenn der WINS-Server einen Namen in seiner Datenbank findet, der dem Namen in der Registrierungsanfrage entspricht, gibt er eine Namensanfrage an die IP im gefundenen Eintrag aus. Er versucht dies mehrmals. Gibt es keine Antwort vom Client, dem der Name gehört, kann der neue Client den Namen erfolgreich registrieren. Antwortet der Besitzer des Namens aber, wird dem anfragenden Client eine negative Antwort gesendet, und er kann den Namen nicht registrieren.

Abb. 18.3: Anfragepaket für eine Namensregistrierung, das von einem Client an einen WINS-Server übertragen wird



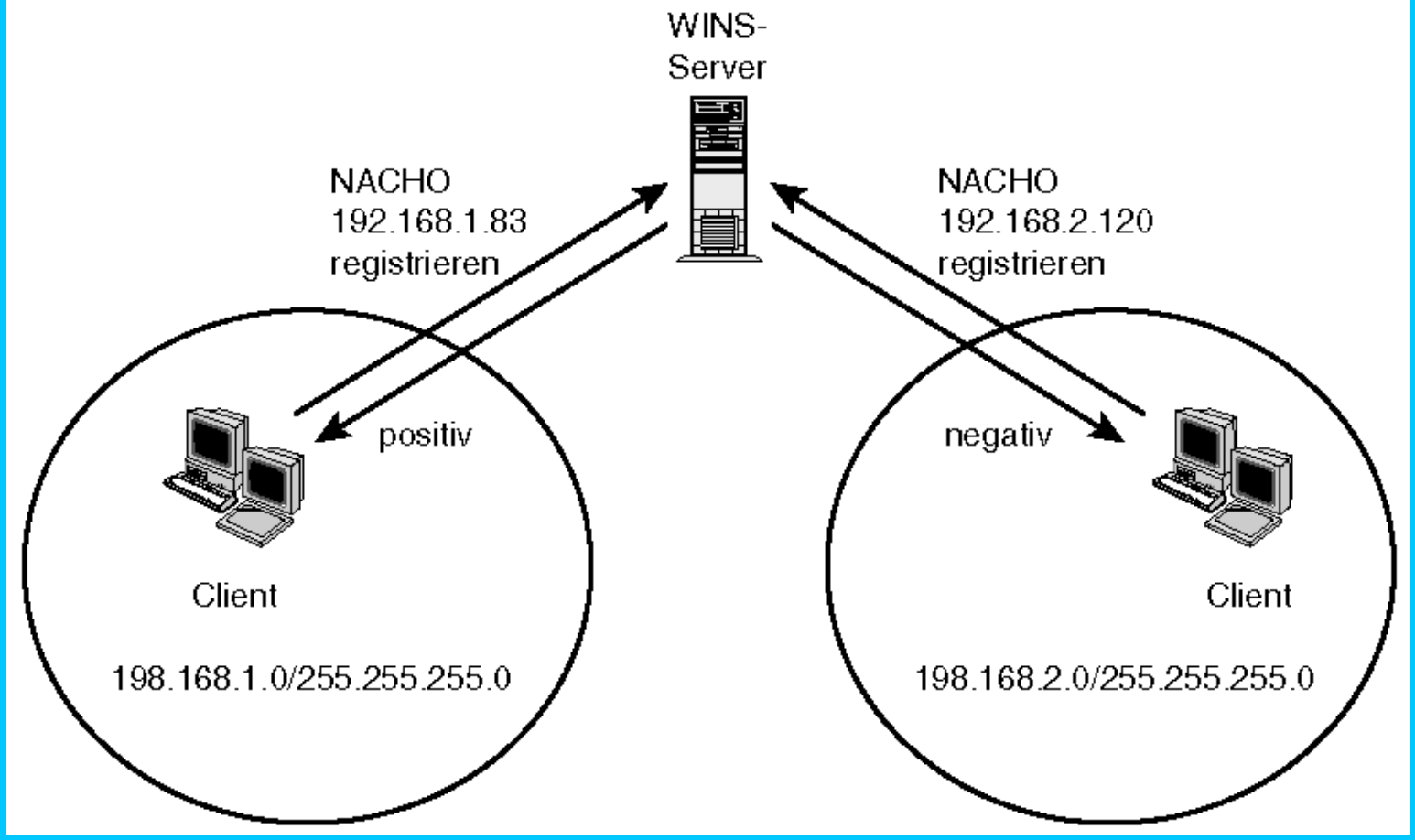
Die Auflösung von Namen findet in etwa auf die gleiche Art und Weise statt. Ein Client überträgt den aufzulösenden Namen in einem Paket an den WINS-Server. Der Server sucht dann in seiner Datenbank nach dem Namen und antwortet, wenn er ihn findet, mit der entsprechenden IP-Adresse. Andernfalls sendet der Server eine negative Antwort.

wins server

Da Sie jetzt grundlegende Kenntnisse darüber haben, was WINS ist und warum es benötigt wird, werde ich darstellen, wie Sie Samba für die Benutzung eines WINS-Servers konfigurieren. Vielleicht fragen Sie sich, warum Samba einen anderen WINS-Server benutzen wollte, wenn es selbst als einer agieren kann.

Wenn Sie Ihr NetBIOS-Netzwerk implementieren, sollten Sie nur einen WINS-Server installieren. Sie können mehrere Server haben, wenn diese in der Lage sind, Datenbanken untereinander abzugleichen, wie z.B. der Windows-NT-4.0-WINS-Server. Der Grund dafür ist, dass der Zweck für die Verwendung eines WINS-Servers darin liegt, die Namensbereiche von allen IP-Subnetzen in einem zusammenzufassen. In Abbildung 18.4 versuchen zwei verschiedene Hosts, den eindeutigen Namen `NACHO` gemeinsam zu erhalten. Gibt es keinen WINS-Server, ist dies möglich, da sich die zwei Rechner in verschiedenen Subnetzen befinden. Wird jedoch ein WINS-Server eingesetzt, wird der Namensbereich der zwei Subnetze kombiniert (zumindest für Clients mit WINS-Unterstützung). Der Client auf der rechten Seite kann den Namen nicht beim Server registrieren, da der Client auf der linken Seite den Namen bereits besitzt.

Abb. 18.4: Durch Installation eines WINS-Servers den NetBIOS-Namensbereich vereinigen



Wenn Sie mehrere nicht synchronisierte WINS-Server verwenden, bleibt der Namensbereich fragmentiert. Wenn Sie also einen existierenden WINS-Server in Ihrem Netzwerk installiert haben und mit den anderen dort registrierten NetBIOS-Clients zusammengehören wollen, müssen Sie sich ebenfalls beim WINS-Server registrieren.

Dafür müssen Sie den Samba-Parameter `wins server` benutzen. Dieser Parameter akzeptiert als Wert die IP-Adresse Ihres existierenden WINS-Servers. Es ist möglich, stattdessen den DNS-Namen des WINS-Servers anzugeben, aber die Spezifizierung der IP-Adresse ist die bevorzugte Methode. Standardmäßig wird kein WINS-Server definiert:

```
wins server = none
```

Vorausgesetzt, dass Sie einen Rechner (möglicherweise einen anderen Samba-Server) an der IP-Adresse `198.162.1.80` als einen WINS-Server konfiguriert haben, können Sie spezifizieren, dass der aktuelle Rechner diesen Server für die Namensregistrierung und -auflösung verwenden soll, indem Sie folgenden Wert in Ihre `smb.conf`-Datei einfügen:

```
wins server = 192.168.1.80
```

wins support

Der Parameter `wins support` akzeptiert einen booleschen Wert, der Sambas WINS-Server-Funktion entweder aktiviert oder deaktiviert. Standardmäßig ist Samba nicht aktiviert, als WINS-Server zu agieren:

```
wins support = no
```

Um Samba als einen WINS-Server zu aktivieren, setzen Sie diesen Parameter einfach auf

```
wins support = yes
```

und konfigurieren die benötigten Einstellungen auf dem Client. Für Windows-Clients wurde dies in Kapitel 14, »Windows 9x und Windows NT«, dargestellt.

Die Datenbank, die von `nmbd` benutzt wird, wenn Samba als WINS-Server agiert, hat die Form einer flachen Textdatei mit dem Namen `wins.dat`, die sich in der Standardinstallation im `lock`-Verzeichnis `/usr/local/samba/var/locks/` befindet. Sie brauchen sich über all die Informationen, die in jedem WINS-Eintrag aufgeführt sind, nicht den Kopf zu zerbrechen. Einige Felder erklären sich allerdings ganz von selbst. Hier ist ein Beispiel-Listing:

```
"CHIPSNDIPS#00" 919228124 255.255.255.255 c4R
"CHIPSNDIPS#1b" 919228124 192.168.1.72 44R
"CHIPSNDIPS#1c" 919228124 192.168.1.72 c4R
"CHIPSNDIPS#1e" 919228124 255.255.255.255 c4R
"QUESO#00" 919228124 192.168.1.72 46R
"QUESO#03" 919228124 192.168.1.72 46R
"QUESO#20" 919228124 192.168.1.72 46R
```

Sie sollten die hexadezimale Zahl erkennen, die dem Rautenzeichen (#) im Namen folgt. Dies ist die gleiche Syntax, die `nmblookup` verwendet, um das NetBIOS-Ressourcenbyte auszugeben.

Ist Samba als WINS-Server aktiviert, fragt es bei sich selbst an, wenn es einen Namen erhält, der aufgelöst werden muss. Somit agiert Samba gleichzeitig als WINS-Server und WINS-Client.

Sie tendieren vielleicht zu folgenden Einstellungen:

```
wins support = yes
```

und

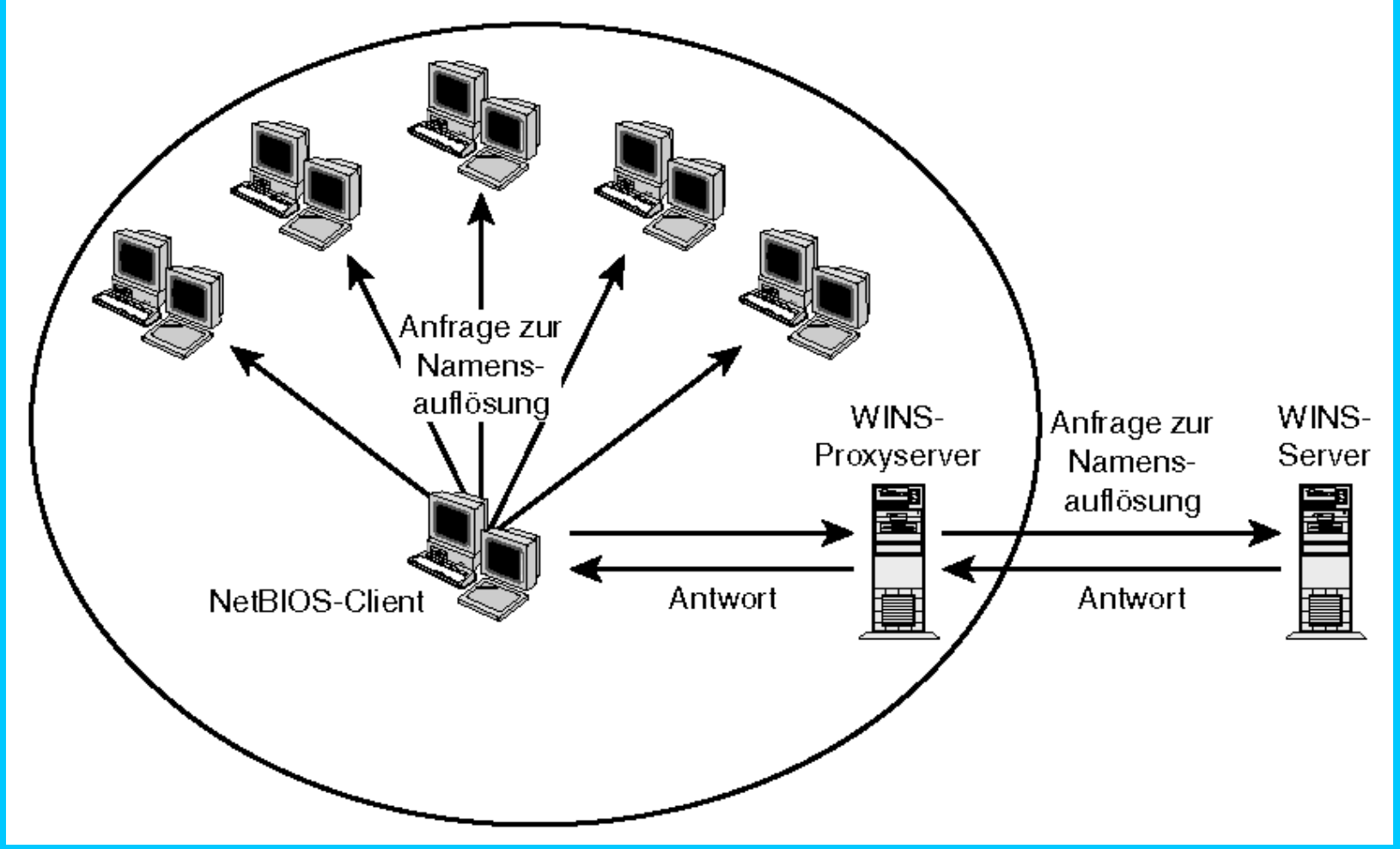
```
wins server = IhreIP-Adresse
```

Dies wäre falsch. Die Parameter `wins server` und `wins support` schließen sich aus. Sie sollten nicht beide gleichzeitig gesetzt werden.

wins proxy

Wenn die TCP-/IP-Einstellungen auf Ihren PC-Clients manuell eingerichtet werden, wird es wahrscheinlich schwer sein, WINS zu implementieren, ohne jeden Rechner aufzusuchen und die notwendigen Einstellungen zu aktualisieren. Eine mögliche Option bestünde darin, einen Samba-Server auf jedem Subnetz einzurichten, der als Proxy-Agent für Broadcast-Anfragen zur Namensauflösung agiert. Abbildung 18.5 illustriert diese Idee.

Abb. 18.5: Einen WINS-Proxy-Agent einsetzen, der bei der Namensauflösung behilflich ist



Der boolesche Parameter `wins proxy` kontrolliert, ob der `nmbd` für andere Clients auf Broadcast-Anfragen antwortet. Standardmäßig ist dies nicht der Fall:

```
wins proxy = no
```

Wenn Sie diese Einstellung aktivieren, leitet `nmbd` Namensanfragen sowohl für die Registrierung als auch die Auflösung an den in `smb.conf` spezifizierten WINS-Server (möglicherweise sich selbst im Fall von `wins support = yes`) weiter und gibt die Antwort an den anfragenden Client wieder zurück.

dns proxy

Wenn Samba als WINS-Server (`wins support = yes`) konfiguriert ist, ist es möglich, dem `nmbd` mitzuteilen, alle Namensanfragen im DNS suchen zu lassen, wenn sich die entsprechenden Namen nicht in der lokalen WINS-Datenbank finden lassen. Der Parameter `dns proxy` kontrolliert dieses Verhalten. Samba befragt standardmäßig den DNS, wenn ein Name nicht in WINS gefunden wird:

```
dns proxy = yes
```

Ist `dns proxy` aktiviert, startet der `nmbd` einen anderen `nmbd`-Prozess, um die DNS-Anfragen durchzuführen, weil die Standard-`gethostbyname()`-Funktion in Unix ein blockierender Aufruf ist. Der zusätzliche Prozess ermöglicht dies, ohne den normalen NetBIOS-Name-Service zu benachteiligen.

Wenn Sie keine Standverbindung zu Ihren DNS-Servern haben, entscheiden Sie sich möglicherweise dafür, `dns proxy` zu deaktivieren. Dies verhindert, dass Samba jedes Mal eine

Aufwärtsverbindung zu Ihren Nameservern initialisiert, wenn eine Anfrage für einen nicht existenten Namen übertragen wird:

```
dns proxy = no
```

Dies wäre der Fall, wenn Sie über Anwählverbindungen mit Ihren DNS-Servern verbunden wären.

lmhosts

Eine Methode für die Auflösung von Namen, über die ich bisher noch nicht wirklich gesprochen habe, ist die Suche nach einem Namen in einer lokalen `lmhosts`-Datei. Die `hosts`-Datei des LanManagers entspricht funktionell der Unix-Datei `/etc/hosts`, mit der Ausnahme, dass sie IP-Adressen NetBIOS-Namen zuordnet statt Hostnamen. Sambas `lmhost`-Datei und eine, die von Windows-Clients verwendet wird, unterscheiden sich etwas im Format, also werde ich beide darstellen.

Informationen zur Syntax von Sambas `lmhosts`-Datei finden Sie in der entsprechenden Manpage (`man lmhosts`). Jeder Eintrag sieht wie folgt aus:

`IP-Adresse    Rechnername`

Der Eintrag für meinen derzeitigen Samba-Server ist z.B.:

```
192.168.1.73    queso
```

Denken Sie daran, dass bei NetBIOS-Namen nicht zwischen Groß- und Kleinschreibung unterschieden wird, also hätte ich genauso gut den Namen `QUESO` verwenden können. Der Eintrag *`Rechnername`* kann jeder gültige Name sein, der vom Tool `nmblookup` akzeptiert wird. So kann ich z.B. einen bestimmten Namenstyp spezifizieren, indem ich einen NetBIOS-Ressourcentyp an den Namen anhänge:

```
192.168.1.73    queso#20
192.168.1.73    chipndips#1b
```

Der erste Eintrag bezieht sich auf die Server-Ressource `queso`. Denken Sie daran, dass `<20>` der Ressourcentyp ist, mit dem Freigabepunkte bezeichnet werden. Der zweite Eintrag wird benutzt, um den Domain Master Browser für die Gruppe `chipsndips` zu finden.

Normalerweise befindet sich Sambas `lmhosts`-Datei im gleichen Verzeichnis wie seine Konfigurationsdatei, also `/usr/local/samba/lib`. Es ist möglich, einen anderen Standort zu spezifizieren, wenn Sie den `nmbd` mit dem Flag `-H` *`Dateiname`* starten. Das folgende Beispiel startet `nmbd` als Daemon mit `/etc/lmhosts` als Standard:

```
/usr/local/samba/bin/nmbd -H /etc/lmhosts -D
```

Ich möchte noch eine letzte Bemerkung über Sambas Benutzung einer `lmhosts`-Datei machen. Die Inhalte der Datei betreffen nur die Namensauflösung von Samba-Hosts. Samba verwendet keine Einträge, um Anfragen aufzulösen, die es von anderen Hosts empfängt.

Das Windows-Format für die `lmhosts`-Datei ist etwas umfangreicher, aber im Wesentlichen das Gleiche. In seiner einfachsten Form sieht ein Eintrag hier aus wie einer in Sambas `lmhosts`-Datei:

`IP-Adresse    Rechnername`

Microsofts Format ermöglicht jedoch das Einfügen anderer Dateien über eine `#include`-Direktive. Dies gibt Ihnen die Möglichkeit, eine zentrale `lmhosts`-Datei für alle Clients zu verwenden:

```
#include \\publicserv\global\lmhosts
```

Natürlich müssen Sie eine Entsprechung für `publicserv` definieren, bevor Sie `#include` ausführen, damit die Dinge funktionieren. Weitere Informationen über Microsofts `lmhosts`-Datei finden Sie in der Datei `lmhost.sam`, die mit dem TCP/IP-Protokoll installiert wird.

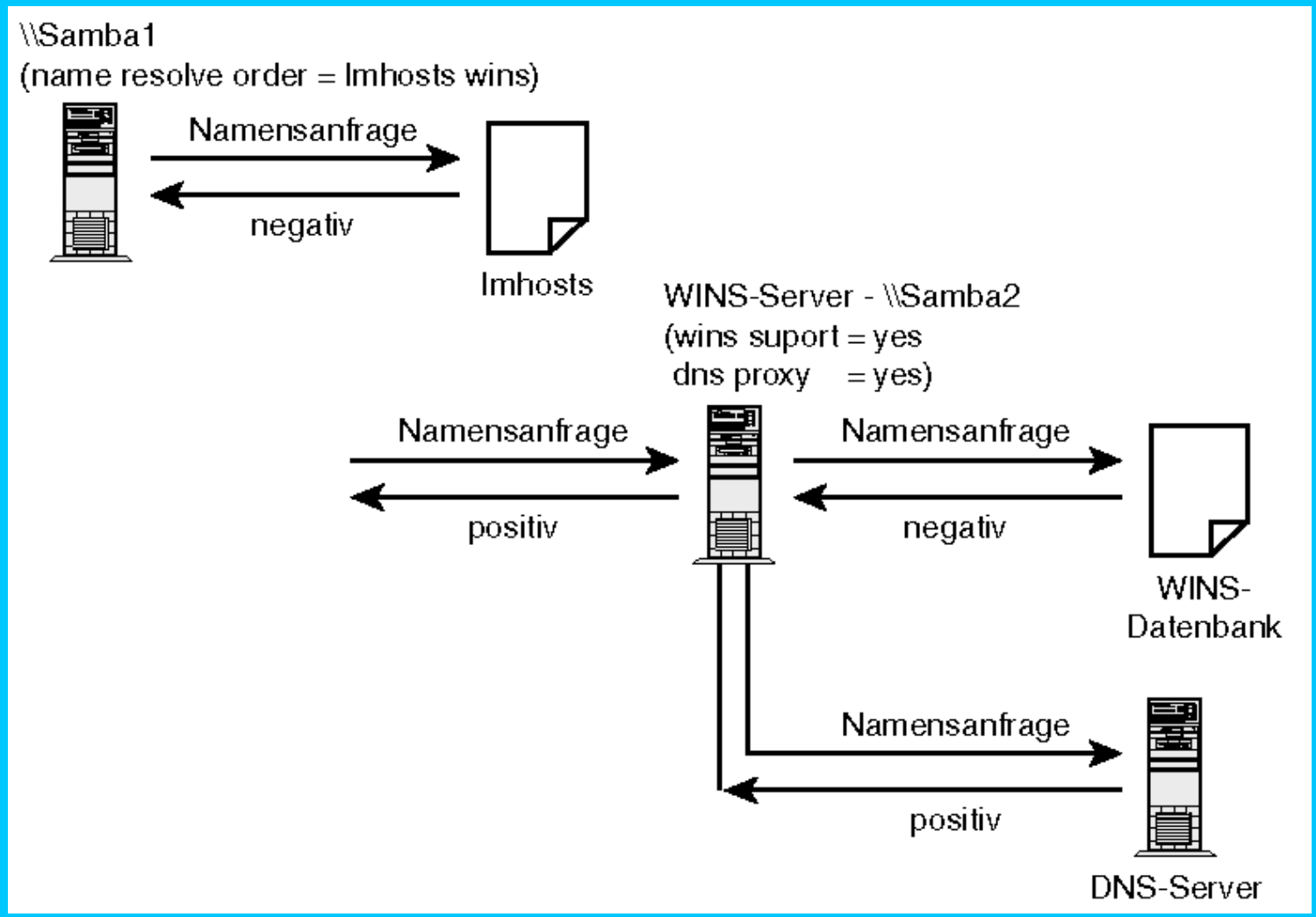
name resolve order

Obwohl der Parameter `name resolve order` bereits in Kapitel 5, »Die Datei `smb.conf`: Samba mitteilen, was es tun soll«, erwähnt wurde, möchte ich ihn hier noch einmal darstellen, nachdem alle möglichen Methoden untersucht wurden. Insbesondere möchte ich darauf hinweisen, dass dieser Parameter nur bestimmt, wie Samba seine eigenen Namensanfragen auflöst, und nicht die, die es von anderen erhält. Wenn also die folgende Reihenfolge definiert ist, würde Samba selbst keine Namensanfragen per Broadcast übertragen und nicht versuchen, Namen über den DNS aufzulösen:

```
name resolve order = lmhosts wins
```

Es würde nur in der lokalen `lmhosts`-Datei suchen und dann den konfigurierten WINS-Server befragen. Ist der kontaktierte WINS-Server ein anderer Samba-Server (oder möglicherweise er selbst) und hat dieser Server die Einstellung `dns proxy` aktiviert, ist es möglich, dass der Name letztendlich über DNS aufgelöst wird. Abbildung 18.6 zeigt, wie dies passieren kann. `SAMBA1` versucht einen Namen aufzulösen. Nachdem er den Host nicht in seiner eigenen `lmhosts`-Datei gefunden hat, befragt der Server WINS. Der WINS-Server, `SAMBA2`, versucht, den Namen in seiner internen WINS-Datenbank zu finden. Da er den Namen nicht finden kann, kontaktiert `SAMBA2` den DNS-Server der Domäne. Diesmal wird der Name gefunden und die entsprechende IP-Adresse zurück an `SAMBA1` geleitet.

Abb. 18.6: Ein Beispiel für den Parameter name resolve order



Obwohl der Name letztendlich über DNS aufgelöst wurde, war es nicht `SAMBA1`, der den Domain Name Server befragt hat. Dieses Beispiel gilt auch dann noch, wenn `SAMBA1` und `SAMBA2` der gleiche Rechner sind. Dies ist die feine Unterscheidung, die ich meinte, als ich an einem früheren Punkt sagte, dass Samba gleichzeitig als WINS-Client und WINS-Server agieren kann.

WINS und Windows 2000

Vielleicht haben Sie einiges von dem Rummel um die nächste Freigabe von Windows NT (jetzt als Windows 2000 bezeichnet) mitbekommen. Einer der Hauptunterschiede zwischen dem neuen Betriebssystem und den derzeitigen Versionen von Windows NT besteht darin, dass »WINS in NT 5 nicht mehr vorhanden ist«. Ich kann nicht mehr zählen, wie oft ich das gehört habe.

Windows 2000 wird die Fähigkeit haben, das CIFS/SMB-Protokoll zu verwenden, ohne es über NetBIOS zu schichten. Diese NetBIOS-lose Implementierung wird es ermöglichen, für alle Namensauflösungen DNS zu verwenden. Sie können den Vorteil dieser Funktion jedoch nur in homogenen Windows-2000-Umgebungen nutzen. Existieren in dem Netzwerk auch ältere Windows-NT- oder Windows-9x-Clients, wird WINS immer noch notwendig sein. Daher können Sie trotz all der Diskussionen darüber, dass WINS mit der nächsten Ausgabe des Aushängeschild-Betriebssystems von Microsoft verschwindet, erwarten, dass NetBIOS und WINS für mindestens noch einige Jahre erhalten bleiben.

Zusammenfassung

Der Windows Internet Name Service (WINS) ermöglicht die Verwaltung von NetBIOS-Namen über mehrere IP-Subnetze. Wenn alle Client-Namen bei einem einzigen WINS-Server registriert werden, ist es möglich, Dinge wie Browsing über Subnetzgrenzen (siehe Kapitel 20, »Router-Netzwerke und Browsing») und Domänen-Logons über einen Router (siehe Kapitel 21, »Windows-9x-Domänenkontrolle«) zu implementieren. Wenn Sie NetBIOS-Clients in einer Umgebung mit mehreren Subnetzen benutzen oder die Zahl der Broadcast-Pakete, die der Namensauflösung oder -registrierung gelten, vermindern wollen, sollten Sie einen WINS-Server in Ihrem Netzwerk installieren.

Frage & Antwort

F. Können NetBIOS-Clients Namen auch dann auflösen, wenn der WINS-Server im Falle eines Absturzes nicht erreichbar ist?

. Wenn der Client als ein H-Knoten konfiguriert ist, wie es für die meisten Microsoft-Clients der Fall ist, die sich für die Benutzung von WINS registrieren, geht der Client zu Broadcasts über, um Namen zu registrieren und aufzulösen, wenn er den WINS-Server nicht kontaktieren kann. Nur wenn der aufzulösende Name einem Rechner im logischen Subnetz gehört (d.h. über Broadcast-Mechanismen erreicht werden kann), wird der Client eine positive Antwort erhalten. Eine komplette Darstellung der verschiedenen NetBIOS-Knotentypen finden Sie in Kapitel 2.

F. Kann ich, technisch gesehen, mehrere WINS-Server gleichzeitig laufen lassen?

. Ja, das ist möglich. Der NetBIOS-Namensbereich bleibt jedoch segmentiert. Nur Clients, die beim gleichen WINS-Server registriert sind, können sich über Router sehen.

F. Welchen Bezug hat WINS zu Dynamic DNS?

. WINS und DNS, dynamisch oder nicht, sind zwei komplett verschiedene Dinge. Dynamic DNS (DDNS) ist im Wesentlichen eine Implementierung von DNS, die die automatische Aktualisierung von Zonen ermöglicht, wann immer Änderungen in der Domäne vorgenommen werden. Microsofts DNS-Server befragt seinen WINS-Server, wenn eine DNS-Suche nicht erfolgreich ist, aber die zwei sind trotzdem verschiedene Dienste. DNS handhabt IP-Namen und WINS NetBIOS-Namen.

Neue Begriffe

NetBIOS Name Server (NBNS) - Ein in RFC 1001 definierter Server, der die Point-to-Point-Registrierung und -Auflösung von NetBIOS-Namen ermöglicht.

Windows Internet Name Service (WINS) - Microsofts Implementierung eines RFC-1001/1002-konformen NBNS.

lmhosts - Eine ASCII-Datei, die Zuordnungen von IP-Adressen und ihren NetBIOS-Namen enthält. Dies ist funktionell die NetBIOS-Entsprechung zur Unix-Datei `/etc/hosts`.

Tag 19: Browsing in lokalen Subnetzen

In den vorhergehenden Kapiteln haben Sie sich angesehen, wie Sie Samba für die Freigabe von Dateien und Druckern konfigurieren, wie Sie es verwalten und wie Sie über SSL auf Samba zugreifen können, falls dies notwendig ist. Viele Windows-Benutzer verwenden jedoch das Netzwerk-Browsing (den Suchdienst), um die Ressourcen zu lokalisieren, die Sie für ihre Benutzung konfiguriert haben. In diesem Kapitel sehen Sie sich an, wie Samba das Browsing in lokalen Subnetzen unterstützt. Kapitel 20, »Router-Netzwerke und Browsing«, stellt dar, wie Samba Browsing über ein WAN unterstützt.

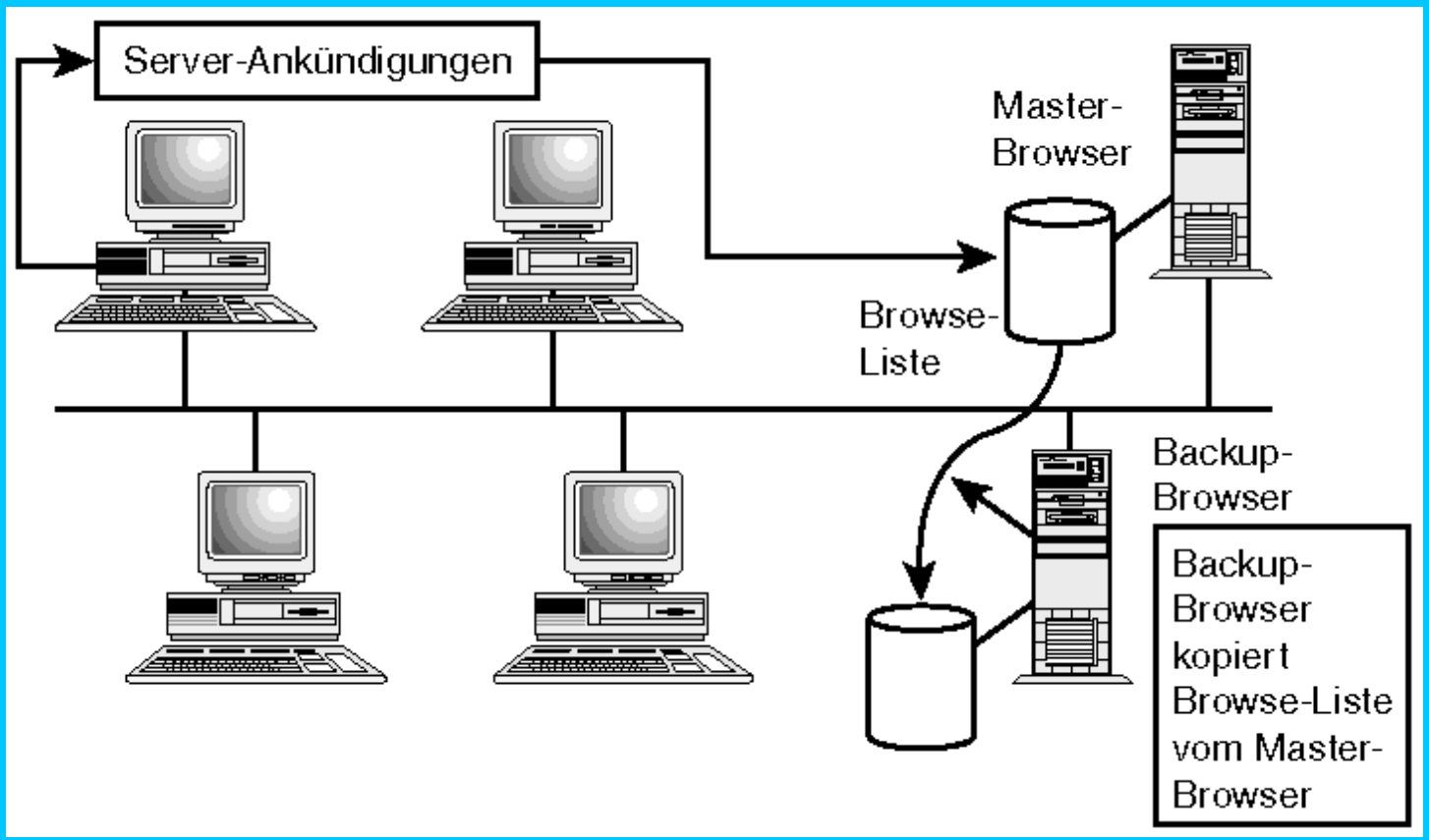
Dieses Kapitel

- gibt eine kurze Einführung zum Thema Browsing,
- stellt alle Samba-Parameter dar, die für das Browsing in lokalen Subnetzen relevant sind,
- bietet einige Beispiele für Sambas Browsing-Unterstützung,
- schlägt Methoden zum Troubleshooting von Browsing-Problemen vor.

Browsing - eine Einführung

Windows-Benutzer durchsuchen das Netzwerk, um Server und Freigaben zu finden, die sie benutzen müssen. Unter Windows 9x und Windows NT wird dies über die Netzwerkumgebung erreicht. Browsing wird jedoch nur von Systemen im Netzwerk unterstützt, die als Browse-Server operieren. Diese Browse-Server können Systeme mit Windows-NT-Server, Windows-NT-Workstation, Windows 9x, Windows für Workgroups oder Samba sein (und möglicherweise auch LMU- und ASU-basierte Server). Abbildung 19.1 zeigt ein Netzwerk mit Browse-Servern und -Clients.

Abb. 19.1: Browsing in einem Windows-Netzwerk



Wenn Windows-Benutzer das Netzwerk durchsuchen, führt Windows folgende Schritte durch:

- Sendet eine `QueryBrowserServers`-Anfrage an das Netzwerk, um die Liste der Browse-Server für die Arbeitsgruppe oder

Domäne, deren Mitglied es ist, zu erhalten

- Wählt einen zufälligen Server aus der Liste und sendet einen `NetServerEnum2`-Aufruf (oder `NetServerEnum`) an diesen Server und fragt ihn nach der Browse-Liste

Die Browse-Liste ist eine Liste aller Server in der Arbeitsgruppe oder Domäne des Master-Browse-Servers und aller Domänen in dem Netzwerk. Die Browse-Liste wird vom Master-Browser über eine gewisse Zeitspanne aufgebaut, während er auf die *Server-Ankündigungen* und *Domänen-Ankündigungen* horcht, die von allen Servern und Domain-Master-Browsern (werden später in diesem Kapitel dargestellt) im Netzwerk gemacht werden. Abbildung 19.2 zeigt die Ergebnisse, wenn ein Netzwerk durchsucht wird, das die Domäne/Arbeitsgruppe `FOWLPLAY` und eine andere Domäne/Arbeitsgruppe namens `Nsdom` enthält. Wenn Benutzer auf eine der Domänen oder Arbeitsgruppen klicken, werden ihnen alle Server in der Domäne oder Arbeitsgruppe angezeigt. Ein Beispiel hierfür sehen Sie in Abbildung 19.3.

Abb. 19.2: Das Browsing des Netzwerks zeigt zwei Domänen/Arbeitsgruppen

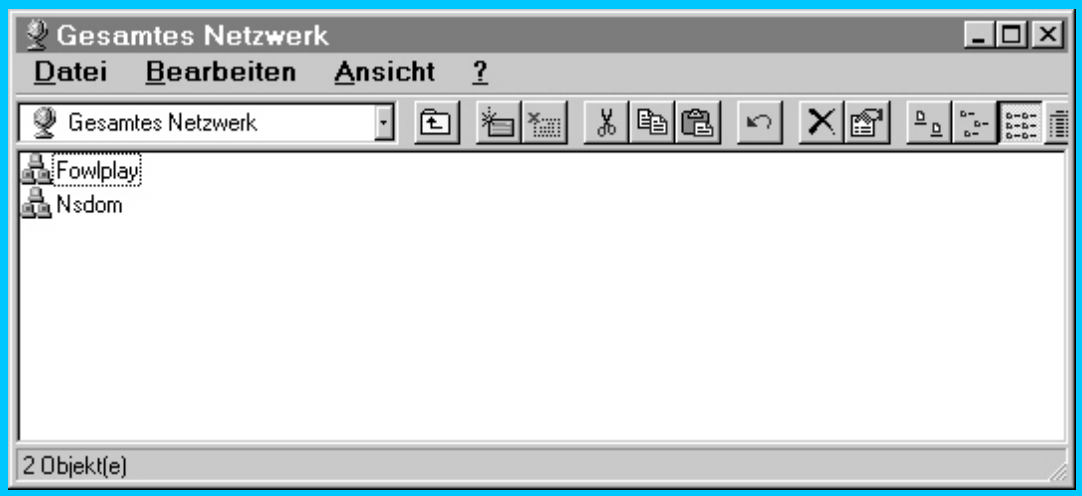
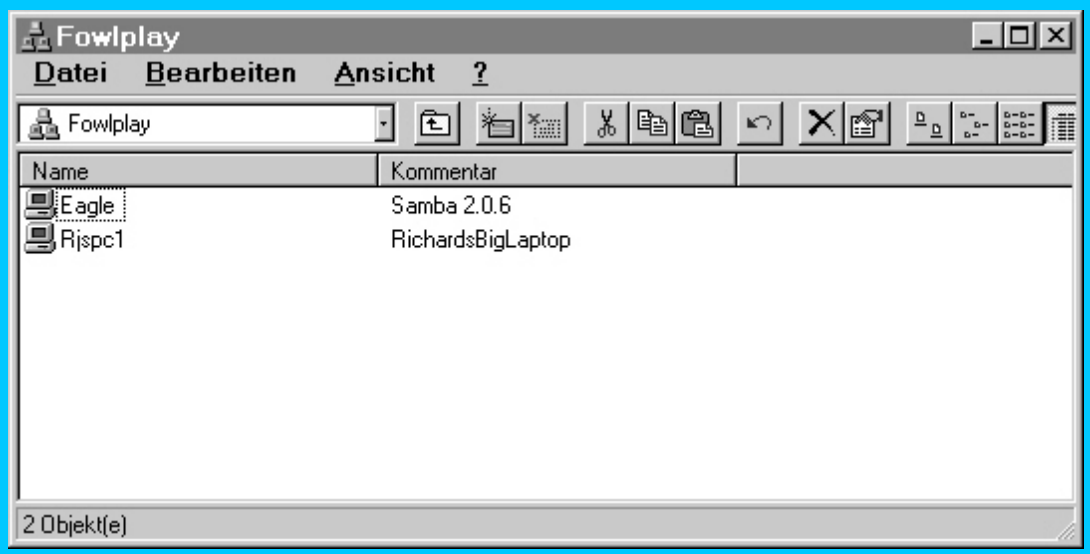


Abb. 19.3: Das Browsing einer Domäne oder Arbeitsgruppe zeigt die Server



Um Redundanz zu bieten und die Browsing-Last über mehrere Systeme zu verteilen, hat ein Windows-Netzwerk normalerweise mehrere Backup-Browser zusätzlich zum Master-Browser. Samba kann entweder als Backup-Browser oder als Master-Browser operieren, abhängig von den Werten einiger Parameter in `smb.conf`.

Wenn sie diese Browsing-Architektur benutzen, können Windows-Clients zwei wichtige Vorteile genießen:

- Sie brauchen keine CPU- und Speicherressourcen für die Verarbeitung von Server-Ankündigungen und die Verwaltung von Browse-Listen bereitstellen.
- Sie brauchen keine Browse-Listen langsam aufbauen, während sie jede Ankündigung sehen. Stattdessen können sie einfach einen Browser kontaktieren, der höchstwahrscheinlich schon wesentlich länger aktiv war als sie selbt.

Zusätzlich zur Verwaltung der Browse-Liste für eine Arbeitsgruppe oder Domäne verwaltet der Master-Browse-Server (oder Master-Browser) eine Liste aller Browse-Server (oder Backup-Browser) im lokalen Subnetz. Diese Liste wird den Clients zur Verfügung gestellt, wenn sie eine `QueryBrowserServers`-Anfrage ausgeben (als Reaktion darauf, dass der Benutzer das Netzwerk durchsuchen will).

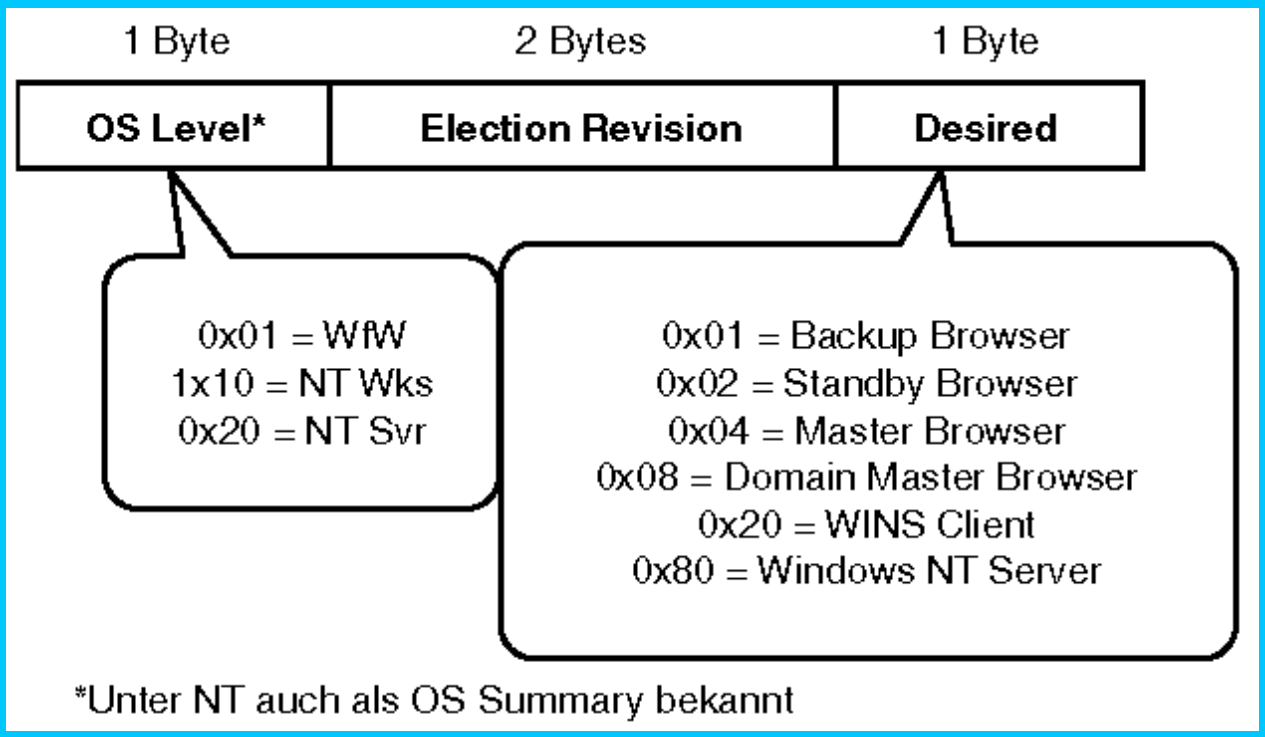
Ein Backup-Browser kontaktiert alle 15 Minuten den Master-Browser, um die aktuellste Kopie der Browse-Liste zu erhalten. Diese Liste von

Servern und Domänen wird zwischengespeichert und benutzt, um auf `NetServerEnum[2]`-Anfragen von Browse-Clients zu antworten.

Wenn ein Backup-Browser aber den Master-Browser nicht kontaktieren kann, erzwingt er eine *Wahl*. Eine Wahl ist der Prozess, der von allen Browsern im Netzwerk verwendet wird, um zu bestimmen, welcher der Master-Browser und welche die Backup-Browser sein sollen.

Wenn ein System in einem Windows-Netzwerk eine Browser-Wahl erzwingen möchte, sendet es eine Wahlanfrage aus. In jeder Wahlanfrage ist ein Vier-Byte-Feld enthalten, das *Election Criteria* genannt wird. Die Struktur dieses Feldes ist in Abbildung 19.4 dargestellt, zusammen mit den Feldern, die Sie kontrollieren können.

Abb. 19.4: Das Feld Wahlkriterium



Wenn der aktuelle Master-Browser eine Wahlanfrage erhält, untersucht er die Werte im Feld *OS-Level* (oder *OS Summary*) und die Felder *Election Revision* und *Desired*, um zu bestimmen, was zu tun ist. Sie können Sambas Verhalten in Bezug auf Browser-Wahlen kontrollieren, indem Sie den Parameter `os level` zusammen mit anderen Parametern entsprechend einstellen (siehe Abschnitt »Samba-Browsing-Parameter« später in diesem Kapitel).

Ist sein eigener OS-Level höher als der anderer Browser, tritt ein Browser in den Status der Wahlführung ein. In diesem Status sendet er bis zu vier weitere Wahlanfragen in Intervallen von 200 bzw. 400 Millisekunden, je nachdem ob er ein Master-Browser oder ein Backup-Browser ist. Gewinnt ein Browser viermal hintereinander jede Wahl, wird er der Master-Browser für diese Arbeitsgruppe oder Domäne. Entspricht sein OS-Level dem eines anderen Browsers im Netzwerk, werden andere Felder wie z.B. *Election Revision* oder *Desired* überprüft, um festzulegen, wer die Wahl gewinnt.

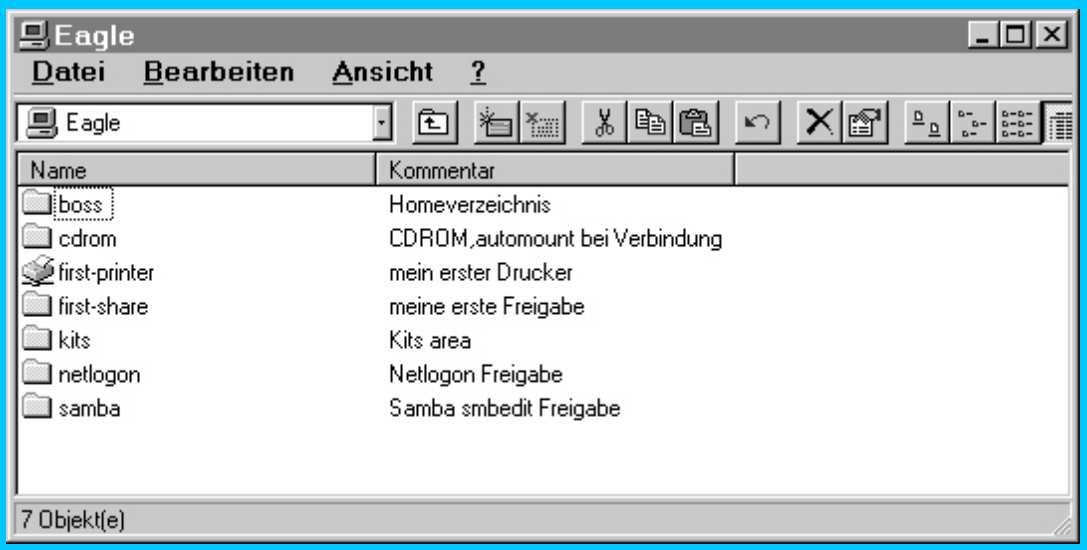
Erhält ein Master-Browser eine Wahlanfrage, die anzeigt, dass ein anderes System die Wahl gewinnen wird, stuft er sich selbst als Backup-Browser herunter.

Wenn ein System zum Master-Browser wird, sendet es per Broadcast eine `RequestAnnouncement`-Anfrage an Rechner im Subnetz. Die

Systeme antworten auf die `RequestAnnouncement`-Anfrage nach einem zufälligen Zeitraum innerhalb der nächsten 30 Sekunden mit einem `ServerAnnouncement`. Dies hilft, die Last auf dem Master-Browser zu reduzieren.

Klickt ein Windows-Benutzer auf einen Server, der in einer Browsing-Ansicht (z.B. in der Netzwerkumgebung) sichtbar ist, sendet Windows eine `NetShareEnum`-Anfrage an den ausgewählten Server und zeigt die Ergebnisse an. Ein Beispiel hierfür sehen Sie in Abbildung 19.5. Obwohl dieser letzte Schritt keine Interaktion mit einem Master-Browser oder einem Backup-Browser beinhaltet, kontrolliert eine Reihe von Samba-Parametern die dargestellten Informationen. Technisch gesehen kann auch der Samba-Server, von dem Sie eine Liste der Freigaben haben wollen, entweder der Master-Browser oder ein Backup-Browser sein. Diese Funktion ist jedoch unabhängig von der Funktion, Freigaben aufzulisten und Datei- und Druckerfreigaben zur Verfügung zu stellen.

Abb. 19.5: Browsing der Freigaben auf einem Server



Samba-Browsing-Parameter

Das Verhalten von Samba in Bezug auf das Browsing wird von einer Reihe von `smb.conf`-Parametern kontrolliert. Diese Parameter beeinflussen u.a. folgende Dinge:

- Ob Samba Browsing-Dienste bietet.
- Ob Samba versucht, ein Master-Browser zu werden oder ein Backup-Browser wird.
- Welche Freigaben angezeigt werden, wenn ein Samba-Server von einem Windows-Client-System durchsucht wird.

In den folgenden Abschnitten werden die einzelnen Parameter beschrieben.

announce as

Dieser globale Parameter spezifiziert, als welchen Server-Typ Samba sich selbst ankündigt. Die gültigen Optionen sind:

`NT`	Ankündigung als NT-Server in einem Host-Announcement. Dies ist die Standardeinstellung.
`Win95`	Ankündigung als Windows-95-System in einem Host-Announcement.
`WfW`	Ankündigung als Windows-für-Workgroups-System in einem Host-Announcement.

Generell sollten Sie den Standardwert für diesen Parameter nicht ändern müssen:

```
announce as = NT
```

announce version

Dieser globale Parameter spezifiziert die Versionsnummern, die Samba (`nmbd`) benutzt, wenn es eine Host-Ankündigung überträgt. Standardmäßig verwendet Samba derzeit die Versionsnummer 4.2 (Major-Version 4 und Minor-Version 2), die größer ist als der von Windows NT benutzte Wert 4.0.

Generell sollten Sie den Standardwert nicht ändern müssen:

```
announce version = 4.2
```

auto services

Dieser globale Parameter stellt die Liste der Freigaben zur Verfügung, die automatisch in die Browse-Liste aufgenommen werden sollen.

Dieser Parameter ist am nützlichsten für das Hinzufügen spezieller Freigaben von Home-Verzeichnissen und Druckern, die normalerweise nicht sichtbar wären. Ein Beispiel:

```
auto services = boss    lp postscript
```

Dies teilt Samba (`nmbd`) mit, dass die Dienste `lp` und `postscript` für das Browsing verfügbar gemacht werden sollen. Bitte beachten Sie jedoch, dass dies nur beeinflusst, was die Leute sehen können, wenn Sie auf das Icon für Ihren Server in der Netzwerkumgebung klicken.

Wenn Sie einfach nur in der Lage sein wollen, alle Drucker in Ihrer `printcap`-Datei zu browsen, verwenden Sie den Parameter `load printers`.

browsable

Dieser Freigabe-Parameter spezifiziert, ob individuelle Freigaben in Browse-Listen erscheinen. Wenn Sie diesen Parameter für eine Freigabe auf `no` setzen

```
browsable = no
```

können Sie diese Freigabe vor Clients verbergen, denn der Standardwert ist `yes`:

```
browsable = yes
```

Für diesen Parameter können Sie auch sein Synonym `browseable` verwenden.

browse list

Dieser globale Parameter bestimmt, ob Samba überhaupt Browse-Listen an Clients weitergibt. Normalerweise sollten Sie den Standardwert für den Parameter nicht ändern müssen:

```
browse list = yes
```

comment

Dieser Freigabe-Parameter spezifiziert, welcher Text neben einer Freigabe erscheint, wenn ein Client den Server durchsucht. Hier ist ein Beispiel:

```
comment = Meine erste Freigabe
```

interfaces

Dieser globale Parameter spezifiziert, dass Samba mehrere Netzwerk-Interfaces für Browsing und andere Aktivitäten einrichten und verwenden sollte. Standardmäßig operiert Samba nur auf dem primären Interface eines Systems.

Dieser Parameter nimmt als Wert eine Liste von IP/Netzmaske-Paaren an, entweder im CIDR-Format (a.b.c.d/Prefix-Bits) oder im klassischeren NETMASK-Format (a.b.c.d/e.f.g.h). Ein Beispiel:

```
interfaces = 192.168.1.0/24 192.168.2.0/255.255.255.0
```

lm announce

Dieser globale Parameter spezifiziert, ob Samba (`nmbd`) LanManager-artige Host-Announcements wie die, die von OS/2 und anderen Clients benötigt werden, ausgeben soll, damit der Samba-Server in deren Browse-Listen erscheinen kann.

Der Parameter kann folgende Werte annehmen:

`true`	Ausgabe von LanMan-artigen Host-Announcements in Zeitabständen, die durch den Parameter `lm interval` definiert sind.
`false`	Keine Ausgabe von LanMan-artigen Host-Announcements.
`auto`	Keine Ausgabe von LanMan-artigen Host-Announcements, bis ein solches im Netzwerk gesehen wird. Danach werden sie in den über den Parameter `lm interval` definierten Zeitabständen ausgegeben.

Der Standardwert ist:

```
lm announce = auto
```

lm interval

Zusammen mit dem Parameter `lm announce` verwendet, spezifiziert dieser globale Parameter den Zeitabstand in Sekunden, in dem LanMan-artige Host-Announcements ausgegeben werden, wenn der Parameter `lm announce` auf `true` oder `auto` gesetzt ist.

Ist dieser Parameter auf 0 gesetzt, werden keine LanMan-Host-Announcements ausgegeben, unabhängig von der Einstellung für den Parameter `lm announce`. Der Standardwert für diesen Parameter ist `60`, was bedeutet, dass alle 60 Sekunden LanMan-Host-Announcements ausgegeben werden:

```
lm interval = 60
```

load printers

Dieser globale Parameter bestimmt, ob Samba alle Drucker in der `printcap`-Datei für das Browsing lädt. Dies funktioniert jedoch nur, wenn Sie einen `[printers]`-Abschnitt definiert haben. Ist das nicht der Fall, hat der Parameter keine Funktion. Standardmäßig ist der Parameter auf `yes` gesetzt:

```
load printers = yes
```

local master

Dieser globale Parameter spezifiziert, dass Samba versuchen sollte, beim Starten lokaler Master-Browser im Subnetz zu werden. Ist er auf `no` gesetzt, nimmt Samba nicht an Wahlen teil, kann sie aber erzwingen, wenn es entdeckt, dass kein lokaler Master existiert.

Ist dieser Parameter auf `yes` gesetzt, den Standardwert, nimmt Samba an den Wahlen für den lokalen Master-Browser teil:

```
local master = yes
```

Ob Samba der lokale Master-Browser wird, hängt von den Werten weiterer Parameter in `smb.conf` und einer Reihe anderer Parameter ab. Das bedeutet, Samba muss eine Browser-Wahl gewinnen, um lokaler Master-Browser zu werden.

netbios aliases

Dieser globale Parameter spezifiziert die zusätzlichen NetBIOS-Namen, die Samba für sich selbst als Aliase angibt. Damit kann ein einziges System in Browse-Listen unter mehreren Namen auftauchen.

Agiert das System als Browse- oder Logon-Server, wird keiner der Aliase für den Browse- oder Logon-Server verwendet.

Standardmäßig definiert Samba keine `netbios aliases`. Ein Beispiel:

```
netbios aliases = bald money pinkfloyd
```

Dies bestimmt, dass Samba die NetBIOS-Namen `bald`, `money` und `pinkfloyd` als Aliase für den Samba-Server registriert.

netbios name

Dieser globale Parameter richtet den NetBIOS-Namen ein, unter dem Samba im Netzwerk bekannt ist. Standardmäßig ist dies die erste Komponente des DNS-Namens des Servers. Wenn dieser Rechner ein Browse- oder Logon-Server ist, werden diese Dienste unter dem spezifizierten NetBIOS-Namen bekannt gegeben (und nur unter diesem Namen, nicht unter einem Alias).

Um den NetBIOS-Namen Ihres Servers auf `eagle` einzurichten, verwenden Sie Folgendes:

```
netbios name = eagle
```

os level

Dieser globale Parameter spezifiziert den Wert des Feldes *OS Level* oder *OS Summary*, den Samba in Wahlanfragen verwendet. Der benutzte Wert bestimmt, ob Samba ein Master-Browser im lokalen Netzwerk wird oder nicht.

Windows NT Server verwendet den Wert 32, Windows NT Workstation den Wert 16 und Windows 95 und Windows für Workgroups den Wert 1. Bei einer Wahl gewinnt das System mit dem höchsten OS-Level. (Gibt es ein Unentschieden, werden andere Faktoren benutzt, um zu bestimmen, welches System gewinnt.)

Wenn Sie diesen Wert auf 33 setzen, ist es sicher, dass ein Samba-Server immer gewinnt, während die Einstellung 0 dazu führt, dass Samba immer verliert. Die Einstellung 17 stellt sicher, dass Samba Wahlen gegen Windows NT Workstation gewinnt, aber gegen Windows NT Server verliert.

Der Standardwert für diesen Parameter ist 0, was bedeutet, dass Samba Wahlen gegen jeden anderen Browser verliert. Der folgende Wert setzt den OS-Level Ihres Servers auf 33 und stellt sicher, dass er Wahlen immer gewinnt:

```
os level = 33
```

preferred master

Dieser globale Parameter spezifiziert, ob Samba der bevorzugte Master-Browser für seine Arbeitsgruppe wird oder nicht. Ist dieser Parameter beim Start auf `yes` gesetzt (der Standardwert ist `no`), erzwingt Samba eine Wahl. Seine Chancen, Master-Browser zu werden, hängen dann von den Werten einer Reihe anderer Parameter ab (z.B. `os level`).

Dieser Parameter sollte mit Vorsicht benutzt werden. Sind nämlich mehrere Systeme bevorzugter Master-Browser, werden sie regelmäßig Browser-Kriege durchführen.

Ein Synonym für diesen Parameter ist `preferred master`.

server string

Dieser globale Parameter bestimmt, welcher beschreibende String neben dem Namen eines Samba-Servers in Browse-Listen erscheint. Dies kann jeder beliebige Text sein, der folgende Makros enthalten kann:

%v	Dieses Makro wird durch die Samba-Versionsnummer ersetzt, z.B. 2.0.0.
%h	Dieses Makro wird mit dem Hostnamen des Samba-Servers ersetzt.
%L	Dieses Makro wird durch den NetBIOS-Namen des Samba-Servers ersetzt.

Der Standardwert für diesen Parameter ist:

```
server string = Samba %v
```

Um weitere Informationen in Ihren Server-String einzufügen, wie z.B. den DNS-Hostnamen, versuchen Sie folgende Einstellung:

```
server string = %v on host %h
```

workgroup

Dieser globale Parameter spezifiziert die Arbeitsgruppe, in der ein Samba-Server erscheinen wird. Samba (`nmbd`) setzt den über diesen Parameter spezifizierten Namen als Arbeitsgruppe in Host-Ankündigungen ein. Er wird auch als Domänenname für den Samba-Server verwendet, wenn der Domain-Modus aktiviert ist oder Samba als Primary Domain Controller operiert. Der Standardwert ist:

```
workgroup = WORKGROUP
```

Um den Namen Ihrer Arbeitsgruppe auf `FOWLPLAY` zu ändern, fügen Sie Folgendes in den globalen Abschnitt Ihrer `smb.conf` ein und starten Samba neu:

```
workgroup = FOWLPLAY
```

Browsing-Beispiele

Bis hierher haben Sie erfahren, was Browsing ist und was all die `smb.conf`-Parameter bedeuten, die für das Browsing im lokalen Subnetz relevant sind. In diesem Abschnitt präsentiere ich eine Beispiel-`smb.conf`-Datei und zeige Ihnen, wie diese das Browsing im Netzwerk beeinflusst.

Zunächst sind hier die relevanten Einträge aus dem globalen Abschnitt der `smb.conf`-Datei für Ihren Test-Server `eagle`:

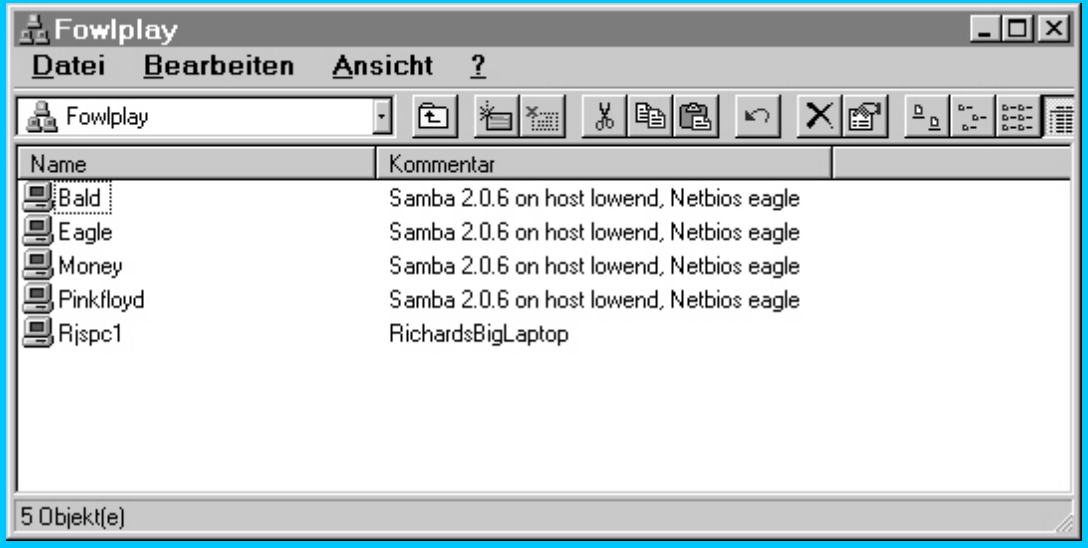
```
[global]
workgroup = FOWLPLAY
server string = Samba %v on host %h, NetBIOS %L
guest account = pcguest
netbios name = EAGLE
netbios aliases = BALD MONEY PINKFLOYD
os level = 33
preferred master = yes
```

Folgende Hinweise gelten für diese Parameter:

1. Die ausgewählte Arbeitsgruppe/Domäne ist FOWLPLAY.
2. Der Server zeigt den String `Samba 2.0.0 on host` *`Hostname`*`, NetBIOS eagle`, da auf dem Rechner die Samba-Version 2.0.0 läuft und ich den NetBIOS-Namen später auf `eagle` gesetzt habe.
3. Der Server kündigt sich selbst als `eagle` an und stellt unter diesem Namen auch die entsprechenden Dienste zur Verfügung, falls er als Browser oder Logon-Server operiert.
4. Der Server kündigt sich selbst unter allen Namen an, die über diesen Parameter eingerichtet sind.
5. Der Server gewinnt Wahlen gegen alle Windows-Systeme, da sein Level höher ist als die, die von anderen Windows-Betriebssystemen benutzt werden.
6. Der Server ist ein bevorzugter Master-Browser und erzwingt beim Start eine Wahl.

Abbildung 19.6 zeigt das Ergebnis, wenn dieser Server durchsucht wird.

Abb. 19.6: Browsing der Arbeitsgruppe FOWLPLAY



In Abbildung 19.6 sehen Sie die Auswirkungen der Parameter `netbios name` und `netbios aliases`. Die Browse-Liste für die Arbeitsgruppe `FOWLPLAY` zeigt einen Server namens `Eagle` sowie weitere Server mit den Namen `Bald`, `Money` und `Pinkfloyd`. Mit jedem dieser Server wird der Kommentar `Samba 2.0.0 on host linux1, NetBIOS eagle` angezeigt, was eine direkte Auswirkung der Einstellungen für den Parameter `server string` ist. Aus diesem Kommentar können Sie schließen, dass der Hostname für jeden dieser Rechner `linux1` ist.

Sie können hier nicht erkennen, welches System der Master-Browser ist. Ein paar schnelle Überprüfungen mit `nmblookup` auf einem Samba-Server können dies allerdings für uns herausfinden:

```
[root@eagle]# nmblookup -S eagle
Sending queries to 16.153.112.255
16.153.112.110 eagle<00>
Looking up status of 16.153.112.110
received 18 names
    EAGLE           <00> -           M <ACTIVE>
    EAGLE           <03> -           M <ACTIVE>
    EAGLE           <20> -           M <ACTIVE>
    .._MSBROWSE_.   <01> - <GROUP>   M <ACTIVE>
    BALD            <00> -           M <ACTIVE>
    BALD            <03> -           M <ACTIVE>
    BALD            <20> -           M <ACTIVE>
    FOWLPLAY        <00> - <GROUP>   M <ACTIVE>
    FOWLPLAY        <1b> -           M <ACTIVE>
    FOWLPLAY        <1c> - <GROUP>   M <ACTIVE>
    FOWLPLAY        <1d> -           M <ACTIVE>
    FOWLPLAY        <1e> - <GROUP>   M <ACTIVE>
    MONEY           <00> -           M <ACTIVE>
    MONEY           <03> -           M <ACTIVE>
    MONEY           <20> -           M <ACTIVE>
    PINKFLOYD       <00> -           M <ACTIVE>
    PINKFLOYD       <03> -           M <ACTIVE>
    PINKFLOYD       <20> -           M <ACTIVE>
num_good_sends=0 num_good_receives=0 [root@eagle]# nmblookup -S bigpc
Sending queries to 16.153.112.255
16.153.112.99 bigpc<00>
Looking up status of 16.153.112.99
received 6 names
    BIGPC           <00> -           M <ACTIVE>
    FOWLPLAY        <00> - <GROUP>   M <ACTIVE>
    BIGPC           <03> -           M <ACTIVE>
    BIGPC           <20> -           M <ACTIVE>
    FOWLPLAY        <!E> - <GROUP>   M <ACTIVE>
    ADMINISTRATOR   <00> -           M <ACTIVE>
num_good_sends=0 num_good_receives=0
```

In der Ausgabe von `nmblookup` für die Namen `eagle` und `bigpc` sehen Sie, dass `eagle` den Namen `FOWLPLAY` als Namen des Typs `<1d>` registriert hat. Das bedeutet, dass `eagle` der lokale Master-Browser für die Arbeitsgruppe `FOWLPLAY` ist. Es ist auch möglich, den Befehl `nmblookup -M` zu verwenden, um Master-Browser zu suchen:

```
[root@linux1 /root]# nmblookup -M fowlplay
Sending queries to 16.153.112.255
16.153.112.110 fowlplay<1d>
```

Hier sehen Sie, dass der Master-Browser `16.153.112.110` ist. Wenn Sie `nmblookup` noch einmal in der folgenden Art und Weise laufen lassen, können Sie den Master-Browser finden:

```
nmblookup -S -A 16.153.112.110
```

Eine andere Methode, die Identität des Master-Browsers herauszufinden, besteht darin, `nmblookup` das Flag `-T` hinzuzufügen:

```
nmblookup -T -M fowlplay
```

Dies teilt `nmblookup` mit, IP-Adressen zurück in DNS-Namen zu übersetzen und diese zusammen mit den vorher gezeigten IP-Adressen auszugeben.

Probleme beim Browsing

Die meisten Browsing-Probleme können durch sorgfältige Einrichtung der relevanten Parameter in Ihrer `smb.conf` gelöst werden. Wenn Master-Browser jedoch ausfallen und Wahlen abgehalten werden, sehen Sie möglicherweise Meldungen wie: »Die Serverliste ist zur Zeit nicht verfügbar.« Möglicherweise sehen Sie auch, dass große Teile des Netzwerks verschwinden, wenn es nicht genügend Backup-Browser im Netzwerk gibt.

Sie können `nmblookup` auf einem Samba-Server benutzen, um zu untersuchen, welche Knoten in der Arbeitsgruppe/Domäne vorhanden sind und welcher Knoten der Master-Browser ist, wenn überhaupt.

Folgendes kann u.a. zu Problemen in Bezug auf das Browsing führen:

- Auf Ihrem System ist kein Gast-Account konfiguriert, oder Sie haben keinen Eintrag `guest account` in Ihrer `smb.conf`. Beides kann dazu führen, dass Clients Server nicht durchsuchen können, um die Liste der verfügbaren Freigaben anzusehen.
- Windows-NT-Systeme, die ein Passwort verlangen, um Samba-Server zu browsen.
- Große Teile des Netzwerks, die zufällig verschwinden und wieder auftauchen. Dies wird durch zu viele Wahlen im Netzwerk hervorgerufen. Zwei Browser kämpfen miteinander, um Master-Browser zu werden.

Samba hat einige wichtige Dateien, die nützlich sind, wenn Sie Browsing-Probleme zu lösen versuchen. Sie haben sie schon in früheren Kapiteln kennen gelernt, aber hier sind sie noch einmal:

`browse.dat`	Diese Datei enthält die Browse-Liste. Sie besteht aus einer Zeile pro Server in der Browse-Liste.
`wins.dat`	Diese Datei enthält alle Einträge, die `nmbd` in seiner WINS-Datenbank verwaltet.

Zusammenfassung

In diesem Kapitel wurden das Browsing in lokalen Subnetzen und alle hierfür relevanten `smb.conf`-Parameter detailliert dargestellt. Sie sollten jetzt in der Lage sein, die meisten Probleme rund um das Browsing in einem Windows-Netzwerk zu lösen, in dem Samba SMB-Freigaben zur Verfügung stellt.

Im nächsten Kapitel werden Sie sich das Browsing in Router-Netzwerken ansehen. Außerdem werden Sie sehen, wie Sie Samba konfigurieren können, damit Browsing-Informationen über Subnetzgrenzen synchron gehalten werden.

Frage & Antwort

F. Wir können unseren Samba-Server nicht durchsuchen. Das heißt, keine der Freigaben wird angezeigt, wenn wir auf das Server-Icon klicken. Wie können wir dies korrigieren?

. Das liegt möglicherweise daran, dass Sie die Freigaben nicht für das Browsing aktiviert haben. Standardmäßig sind Freigaben in Samba nicht durchsuchbar. Um Browsing zu aktivieren, fügen Sie den Freigaben, die beim Browsing sichtbar sein sollen, folgende Zeile hinzu:
`browsable = yes`

F. Wir planen, unseren Samba-Server auf einen anderen Rechner zu verschieben, möchten aber, dass unsere Benutzer den neuen Rechner in ihren Browse-Listen als den aktuellen Server sehen, der HOBBIT heißt.

. Sie müssen den neuen Server ein NetBIOS-Alias angeben lassen oder ihm den NetBIOS-Namen des alten Servers geben. Tatsächlich ist es der Mühe wert, einem Samba-Server einen NetBIOS-Namen zu geben, der sich auf den Dienst bezieht, den er ausübt, statt standardmäßig die erste Komponente der DNS-Adresse des Servers zu benutzen. Sie können einem Samba-Server wie folgt einen NetBIOS-Namen geben:

```
netbios name = hobbit
```

NetBIOS-Aliase werden folgendermaßen eingerichtet:

```
netbios aliases = hobbit
```

F. Wie kann ich sicherstellen, dass mein Samba-Server der Master-Browser in unserer Arbeitsgruppe wird? Manchmal stellen wir fest, dass ein Windows-95-PC Master-Browser wird, was nicht so gut ist.

. Um sicherzustellen, dass Samba der Master-Browser wird, müssen Sie dafür sorgen, dass Samba die Browser-Wahlen gewinnt. Dafür müssen Sie Ihrem Samba-Server einen OS-Level geben, der höher ist als der eines anderen potentiellen Browse-Servers im Netzwerk - z.B. 33. Fügen Sie dafür Folgendes in den globalen Abschnitt Ihrer `smb.conf` ein:

```
os level = 33
```

F. Wir haben in der printcap-Datei auf unserem Samba-Server viele Drucker definiert. Diese werden alle in der Browse-Liste angezeigt, wenn Benutzer den Server durchsuchen, aber sie nützen den Benutzern nicht viel, da sie meistens keine Treiber für die Drucker haben. Wir haben die Drucker, die für die PC-Benutzer interessant sind, in unserer smb.conf-Datei definiert. Wie können wir Samba davon abhalten, all diese unnötigen Drucker anzuzeigen?

. Standardmäßig lädt Samba alle Drucker in Ihre `printcap`-Datei. Um dieses Verhalten umzustellen, fügen Sie einfach Folgendes in den globalen Abschnitt Ihrer `smb.conf` ein:

```
load printers = no
```

Tag 20: Browsing in Netzwerken mit Routern

von Richard Sharpe

Samba unterstützt Browsing in lokalen Subnetzen und über geroutete Netzwerke. Im vorigen Kapitel wurde das Browsing im lokalen Subnetz detailliert dargestellt. In diesem Kapitel erfahren Sie, wie Samba Browsing in gerouteten Netzwerken oder über Subnetzgrenzen hinaus unterstützt.

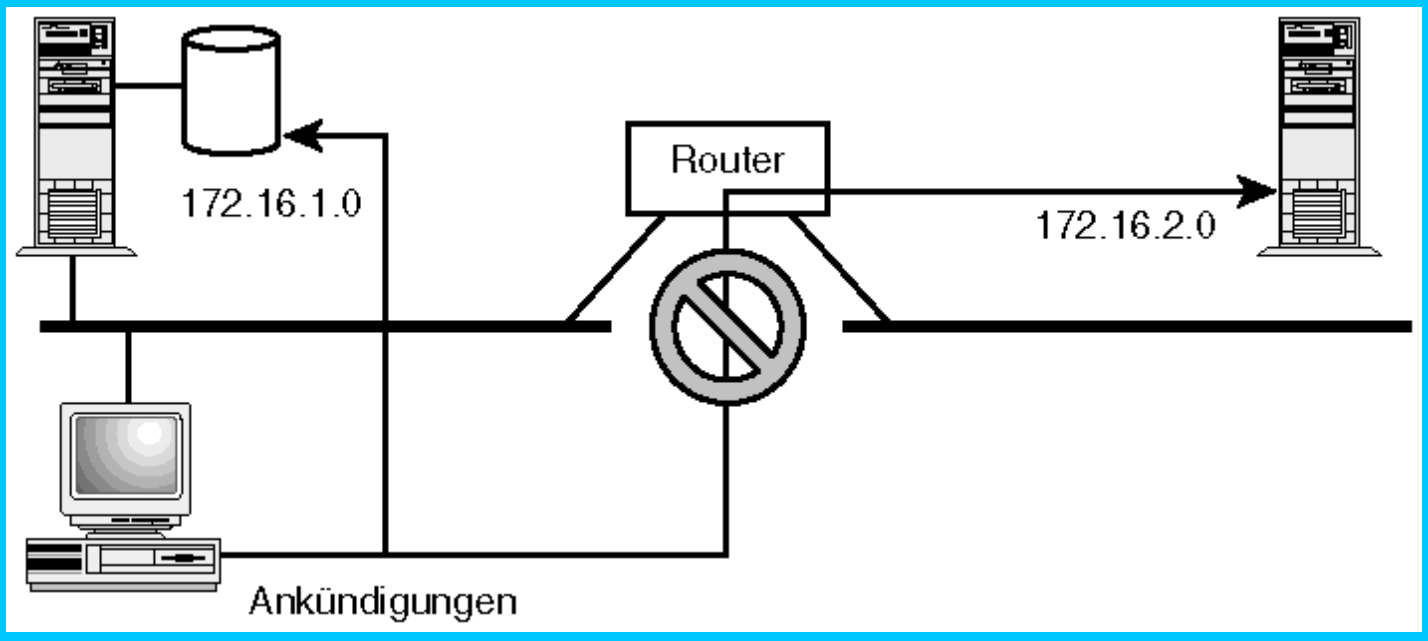
Zunächst werden Sie sich die Unterschiede zwischen Browsing im lokalen Subnetz und Browsing in Netzwerken mit Routern ansehen. Dann stelle ich dar, wie Samba konfiguriert werden kann, um das Browsing in Router-Netzwerken zu unterstützen. Danach werfen Sie einen Blick auf eine Beispielkonfiguration, bevor Sie abschließend einige Troubleshooting-Techniken in Bezug auf das Browsing in Netzwerken mit Routern kennen lernen.

Browsing über Subnetzgrenzen

Erinnern Sie sich aus dem letzten Kapitel daran, dass CIFS/SMB-Server Host-Ankündigungen senden, um die Master-Browser auf ihre Existenz aufmerksam zu machen. Zusätzlich dazu senden Domain-Master-Browser Arbeitsgruppen/Domain-Announcements, um anderen Master-Browsern die Existenz ihrer Arbeitsgruppe/Domäne im Netzwerk mitzuteilen.

All diese Ankündigungen werden jedoch als UDP-Broadcasts an die Broadcast-Adresse für das Subnetz übertragen, in dem sich der Absender befindet. Ist Ihr Subnetz z.B. `172.16.1.0/24`, werden diese Ankündigungen an die Adresse `172.16.1.255` gesendet. Zwar sollten alle Knoten in Ihrem Subnetz diese Broadcasts sehen, aber Knoten in anderen Subnetzen sehen diese Broadcasts in der Regel nicht. Abbildung 20.1 stellt diese Situation ausführlicher dar.

Abb. 20.1: Ankündigungen werden per Broadcast im lokalen Subnetz übertragen



Wir wollen sicherstellen, dass in einem Netzwerk, das aus mehreren verbundenen Subnetzen besteht, alle Subnetze genügend Informationen erhalten, um die im Netzwerk verfügbaren Server durchsuchen zu können. Samba bietet mehrere Methoden, um dies zu erreichen:

- Remote-Announcement von einem Samba-Server zu einem Subnetz
- Remote-Browser-Synchronisation von einem Subnetz zu einem anderen
- Normale Windows-Netzwerk-Browser-Synchronisation

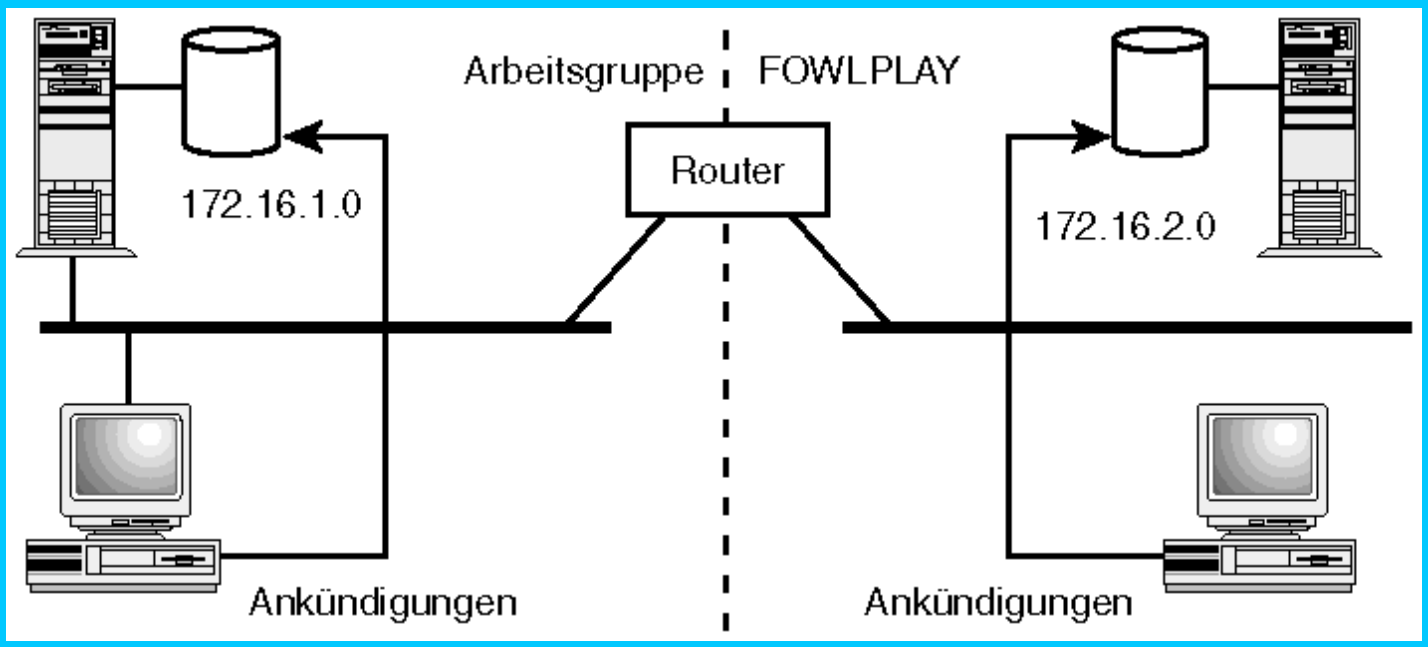
Sie müssen jedoch bei allem, was nun folgt, zwischen Arbeitsgruppen und Domänen unterscheiden. In Standard-Windows-Netzwerken können sich Arbeitsgruppen nicht auf Subnetze ausweiten. Das bedeutet, wenn die Arbeitsgruppe `FOWLPLAY` in den Subnetzen

`172.16.1.0/24` und `172.16.2.0/24` existiert, handelt es sich tatsächlich um zwei separate Arbeitsgruppen. Dies wird in Abbildung 20.2 dargestellt.

Das Windows-Netzwerkmodell behandelt sie als zwei separate Arbeitsgruppen, weil Host-Ankündigungen in einem beliebigen Subnetz die anderen Subnetze nicht erreichen; daher können Browse-Server die Existenz irgendeines Servers in anderen Subnetzen nicht sehen.

In einer Windows-NT-Domäne dagegen wird ein Rechner, der Primary Domain Controller, als Domain-Master-Browser designiert. Zusätzlich dazu enthält jedes Subnetz einen lokalen Master-Browser sowie mehrere Backup-Browser. Ein lokaler Master-Browser stellt die Browse-Liste für dieses Subnetz zusammen und unterrichtet den Domain-Master-Browser mit einem `MasterBrowserAnnouncement` von seiner Existenz. Ein lokaler Master-Browser findet seinen Domain-Master, indem er den Namen *`Arbeitsgruppe`*`<1b>` über WINS übersetzt, da der Domain-Master-Browser diesen Namen bei WINS registriert. In der Domäne `FOWLPLAY` würde der Domain-Master-Browser also den Namen `FOWLPLAY<1b>` registrieren.

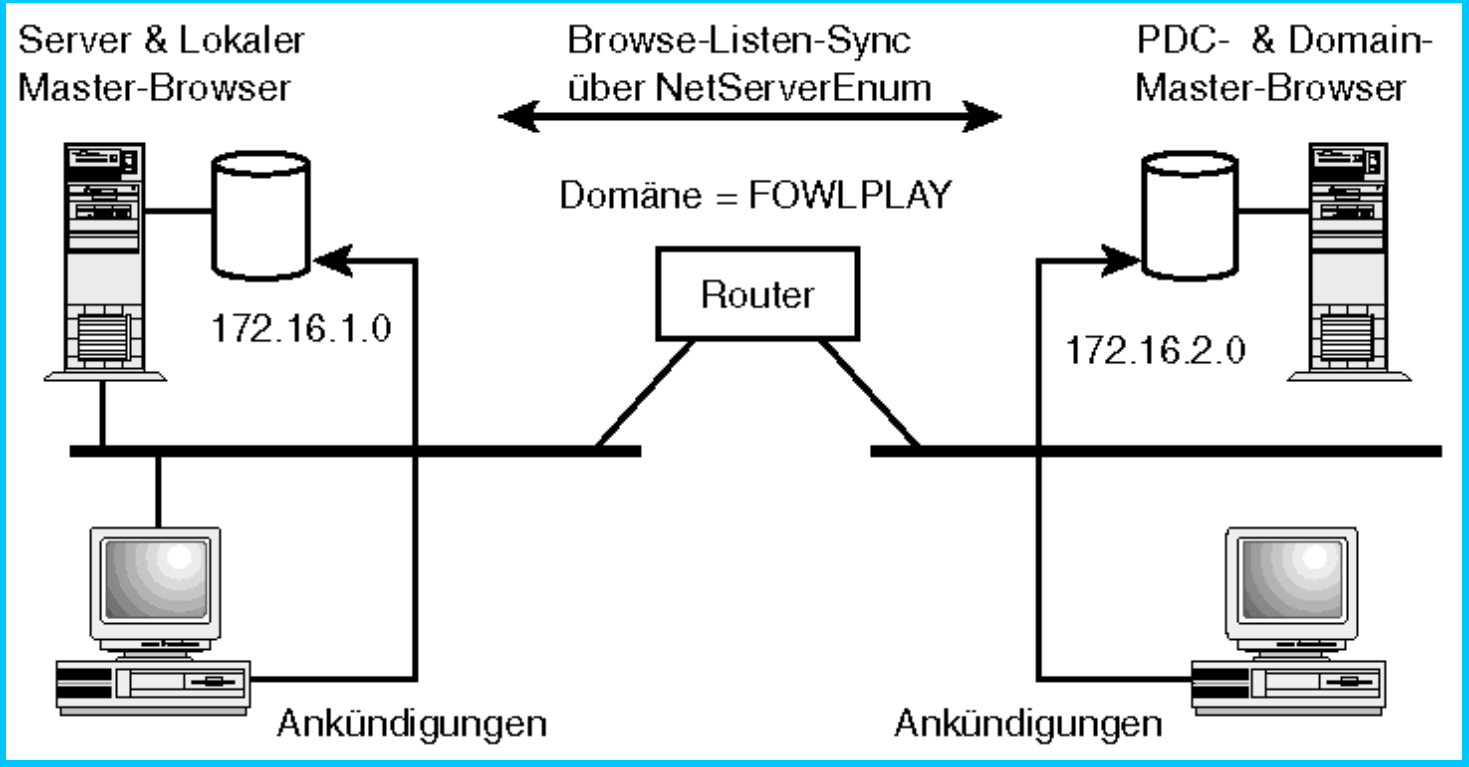
Abb. 20.2: Eine Arbeitsgruppe, die auf Subnetze verteilt ist



Der Domain-Master-Browser sendet alle 15 Minuten eine `NetServerEnum`-Anfrage an die Master-Browser in jedem Subnetz und nimmt die Server-Liste von jedem lokalen Master-Browser in seine eigene Server-Liste auf. So kann der Domain-Master-Browser eine komplette Liste aller Server in der Domäne aufbauen.

Zusätzlich sendet der lokale Master-Browser in jedem Subnetz ebenfalls alle 15 Minuten eine `NetServerEnum`-Anfrage an den Domain-Master-Browser, um eine Liste aller Server in der Domäne zu erhalten. Mit dieser Methode hat jedes Subnetz eine Browse-Liste für sich verfügbar, in der die ganze Domain enthalten ist. Abbildung 20.3 stellt diesen Prozess dar.

Abb. 20.3: Browsing in einer NT-Domäne



In einer Arbeitsgruppe richten Sie Samba dann so ein, dass Host-Ankündigungen per Broadcast in die entfernten Subnetze, mit denen es verbunden ist, übertragen werden. Haben Sie in jedem Ihrer Subnetze Samba-Server, können Sie diese entsprechend einrichten, damit sie ihre Browse-Listen untereinander abgleichen. In einer NT-Domäne konfigurieren Sie einen Rechner als Domain-Master-Browser, und ein Server in jedem Subnetz wird der lokale Master-Browser.

Samba trennt die Funktionen des Primary Domain Controller und des Domain-Master-Browsers. Das heißt, Samba bietet verschiedene Parameter, die kontrollieren, ob Samba als Primary Domain Controller oder Domain Master Browser operiert. Unter Windows NT muss der gleiche Rechner beide Funktionen zur Verfügung stellen. Es ist möglich, eine Windows-NT-Domäne durcheinander zu bringen, wenn Sie einen Samba-Server als Domain-Master-Browser einrichten. Möglicherweise stellen Sie dann fest, dass Clients versuchen, sich in den Samba-Server einzuloggen.

In jedem Fall brauchen Sie aber in der Regel einen WINS-Server, da die Browse-Listen, die von den Servern zusammengestellt werden, nur die Namen der Server enthalten, die in den Host-Ankündigungen gesehen werden. Das heißt, sie enthalten nicht die IP-Adresse der Server. Damit NetBIOS-Namen in IP-Adressen aufgelöst werden, wird irgendeine Art von WINS-Unterstützung benötigt.

Samba-Konfiguration für das Browsing über Subnetzgrenzen

Nachfolgend werden alle `smb.conf`-Parameter dargestellt, die für das Browsing über Subnetzgrenzen in Samba relevant sind. Einige dieser Parameter wurden auch im letzten Kapitel schon erwähnt, da sie sowohl das Browsing im lokalen Subnetz als auch das Browsing über mehrere Subnetze betreffen.

domain master

Dieser globale Parameter spezifiziert, ob Samba ein Domain-Master-Browser ist. Ist dieser Parameter gesetzt, weist er Samba (`nmbd`) an, mit der Zusammenstellung Domänen-weiter Browse-Listen (oder Arbeitsgruppen-weiter Browse-Listen, wenn Sie Arbeitsgruppen-übergreifende Subnetze haben) zu beginnen. Samba (`nmbd`) beansprucht außerdem einen speziellen Domänen-spezifischen NetBIOS-Namen (*`Arbeitsgruppe`*`<1b>`, wobei *`Arbeitsgruppe`* durch den Namen Ihrer Arbeitsgruppe ersetzt wird). Samba wartet dann darauf, von lokalen Master-Browsern mit `MasterBrowserAnnouncements` kontaktiert zu werden, und beginnt, Browse-Listen von jedem lokalen Master-Browser zu sammeln, der den Samba-Server kontaktiert.

Lokale Master-Browser in anderen Subnetzen kontaktieren Samba für seine eigene Browse-Liste und übertragen ihre eigenen Browse-Listen,

wenn Samba danach fragt. So werden Domänen-weite Browse-Listen verwaltet.

Es ist gefährlich, diesen Parameter zu setzen, wenn Sie bereits ein Windows-NT-System haben, das als Primary Domain Controller läuft. Große Teile des Netzwerks können hierbei durcheinander kommen und nicht mehr in der Lage sein, Login-Server zu finden.

Der Standardwert für diesen Parameter ist:
`domain master = no`

local master

Dieser globale Parameter spezifiziert, dass Samba versuchen sollte, lokaler Master-Browser in einem Subnetz zu werden. Ist er gesetzt, nimmt Samba (`nmbd`) an Wahlen für den lokalen Master-Browser teil. Samba sendet außerdem `MasterBrowserAnnouncements` an den Domain-Master-Browser. Es findet den Master-Browser, indem es den Namen *`Arbeitsgruppe`*`<1b>` über WINS auflöst, der vom Domain-Master-Browser registriert wird.

Es ist nicht garantiert, dass Samba lokaler Master-Browser wird, sondern es ist nur sicher, dass es an den Wahlen zum lokalen Master-Browser teilnimmt. Um sicherzustellen, dass Samba lokaler Master-Browser wird, müssen Sie den Wert für den Parameter `os level` in Samba höher setzen als für jeden anderen potentiellen Browse-Server im Netzwerk.

Der Standardwert für diesen Parameter ist:
`local master = yes`

Wird dieser Wert auf `no` gesetzt, nimmt Samba nicht an Browser-Wahlen teil und kann daher nicht lokaler Master-Browser werden.

netbios aliases

Dieser globale Parameter spezifiziert die zusätzlichen NetBIOS-Namen, die Samba für sich selbst als Aliase angibt. Damit kann ein einziges System in Browse-Listen unter mehreren Namen auftauchen.

Agiert das System als Browse- oder Login-Server, wird keiner der Aliase für den Browse- oder Logon-Server verwendet.

Diese Aliase erscheinen in Browse-Listen über mehrere Subnetze, wenn die Konfiguration entsprechend eingerichtet wurde.

Standardmäßig gibt Samba keine `netbios aliases aus`. Ein Beispiel:
`netbios aliases = bald money pinkfloyd`

netbios name

Dieser globale Parameter richtet den NetBIOS-Namen ein, unter dem Samba im Netzwerk bekannt ist. Standardmäßig ist dies die erste Komponente des DNS-Namens des Servers. Ist dieser Rechner ein Browse-Server oder ein Logon-Server, werden diese Dienste unter dem spezifizierten NetBIOS-Namen bekanntgegeben (und nur unter diesem Namen, nicht unter einem Alias).

Dieser NetBIOS-Name erscheint in Browse-Listen über mehrere Subnetze, wenn die Konfiguration entsprechend eingerichtet wurde.

os level

Dieser globale Parameter spezifiziert den Wert des Feldes *OS Level* oder *OS Summary*, den Samba in Wahlanfragen verwendet. Der benutzte Wert bestimmt, ob Samba ein Master-Browser im lokalen Netzwerk wird.

Windows NT Server verwendet den Wert 32, Windows NT Workstation den Wert 16, und Windows 95 und Windows für Workgroups den Wert 1. Bei einer Wahl gewinnt das System mit dem höchsten OS-Level. (Gibt es ein Unentschieden, werden andere Faktoren benutzt, um zu bestimmen, welches System gewinnt.)

Wenn Sie diesen Wert auf 33 setzen, ist es sicher, dass ein Samba-Server immer gewinnt, während die Einstellung 0 dazu führt, dass Samba immer verliert. Die Einstellung 17 stellt sicher, dass Samba Wahlen gegen Windows NT Workstation gewinnt, aber gegen Windows NT Server verliert.

preferred master

Dieser globale Parameter spezifiziert, ob Samba der bevorzugte Master-Browser für seine Arbeitsgruppe ist. Ist dieser Parameter beim Start auf `yes` gesetzt (der Standardwert ist `no`), erzwingt Samba eine Wahl. Seine Chancen, Master-Browser zu werden, hängen dann von den

Werten einer Reihe anderer Parameter ab (z.B. `os level`).

Dieser Parameter sollte mit Vorsicht benutzt werden. Sind nämlich mehrere Systeme bevorzugter Master-Browser, werden sie regelmäßig Browser-Kriege führen, was zu Verstopfung im Netzwerk führt; dies wiederum resultiert in schlechter Performance.

Browser-Kriege, in denen mehrere Server in einem Subnetz darum kämpfen, Browse-Master zu werden, führen zu zwei Problemen:

- Unnötiger Broadcast-Datenverkehr in Ihrem Subnetz, der in schlechter Performance resultieren kann.
- Reduzierte Browsing-Funktionen, da der Master-Browser ständig die Browse-Liste für die Domäne neu aufbauen muss.

Ein Synonym für diesen Parameter ist `preferred master`.

remote announce

Dieser globale Parameter spezifiziert, dass Samba (`nmbd`) regelmäßig Host-Ankündigungen an die spezifizierte IP-Adresse unter dem spezifizierten Arbeitsgruppennamen senden sollte.

Damit können Sie Ihren Samba-Server in den Browse-Listen entfernter (Subnetz-) Arbeitsgruppen erscheinen lassen, wenn dies nach den korrekten Windows-Netzwerkregeln nicht möglich wäre.

Ein Beispiel:
```
remote announce = 172.30.0.255/FOWLPLAY
```

Mit dieser Einstellung sendet Samba Host-Ankündigungen in der Arbeitsgruppe `FOWLPLAY` an die Adresse `172.30.0.255`.

Standardmäßig ermöglichen viele Router die Weiterleitung von gerichteten Broadcasts (z.B. ein Datagramm an die Broadcast-Adresse eines Subnetzes wie `172.30.0.255,` wobei die Subnetzmaske `255.255.255.0` ist) nicht. Sie müssen gerichtete Broadcasts in Ihren Routern aktivieren, um Remote-Ankündigungen an die Broadcast-Adresse in einem anderen Subnetz senden zu können.

Können Sie über Ihren Router keine gerichteten Broadcasts weiterleiten, müssen Sie Remote-Ankündigungen an den lokalen Master-Browser in den entfernten Subnetzen übertragen.

remote browse sync

Dieser globale Parameter bestimmt, dass Samba (`nmbd`) regelmäßig die Synchronisation der Browse-Listen mit den angegebenen Master-Browsern in entfernten Subnetzen verlangen sollte. Die Mechanismen für die Synchronisation funktionieren nur mit anderen Samba-Servern, aber sie stellen die Synchronisation von Browse-Listen zwischen Samba-Servern über Subnetzgrenzen sicher.

Ein Beispiel:
```
remote browse sync = 172.30.0.255
```

Mit dieser Einstellung verlangt Samba vom Master-Browser an der angegebenen Adresse, Browse-Listen abzugleichen.

server string

Dieser globale Parameter bestimmt, welcher beschreibende Text neben dem Namen eines Samba-Servers in Browse-Listen erscheint. Dies kann jeder beliebige Text sein, der folgende Makros enthalten kann:

%v	Dieses Makro wird durch die Samba-Versionsnummer ersetzt, z.B. 2.0.0.
%h	Dieses Makro wird mit dem Hostnamen des Samba-Servers ersetzt.
%L	Dieses Makro wird durch den NetBIOS-Namen des Samba-Servers ersetzt.

Der Standardwert für diesen Parameter ist:
```
server string = Samba %v
```

Um weitere Informationen in Ihren Server-String einzufügen, wie z.B. den DNS-Hostnamen, versuchen Sie folgende Einstellung:
```
server string = %v on host %h
```

wins proxy

Dieser globale Parameter spezifiziert, ob Samba (`nmbd`) als WINS-Proxy agiert: ob es auf Namensanfragen per Broadcast von anderen Hosts antwortet, die WINS nicht verstehen oder nicht für WINS konfiguriert sind. Der Standardwert für diesen Parameter ist:
```
wins proxy = no
```

Sie müssen diesen Parameter möglicherweise für ältere Clients setzen und wahrscheinlich eine Form von WINS-Dienst für entfernte Subnetze spezifizieren, wenn Sie `remote announce` oder `remote browse sync` verwenden oder in einer NT-Domäne sind und Browse-Listen-Synchronisation verwenden.

wins server

Dieser globale Parameter spezifiziert die IP-Adresse oder den DNS-Namen des WINS-Servers, bei dem Samba (`nmbd`) sich registrieren sollte. Dieser Parameter sollte nur verwendet werden, wenn Samba die WINS-Funktion nicht selbst zur Verfügung stellt.

Wenn Sie Browsing über Subnetzgrenzen verwenden, müssen Sie Samba in den entfernten Subnetzen eventuell so konfigurieren, dass es auf einen WINS-Server in zentraleren Subnetzen weist. Hier ist ein Beispiel:

```
wins server = 172.30.0.1
```

wins support

Dieser globale Parameter bestimmt, ob Samba (`nmbd`) als WINS-Server agieren sollte. Ist er gesetzt, agiert Samba (`nmbd`) als WINS-Server. Es sollte niemals mehr als einen Samba-WINS-Server in Ihrem Netzwerk geben, und Samba unterstützt derzeit noch nicht die Microsoft-WINS-Replikationsprotokolle; daher kann Samba nicht als WINS-Server eingerichtet werden, wenn ein Windows-System diese Funktion bereits durchführt. Der Standardwert für diesen Parameter ist:

```
wins support = no
```

Sie brauchen irgendwo in Ihrem Netzwerk einen WINS-Server, wenn Sie das Browsing über Subnetzgrenzen verwenden.

workgroup

Dieser globale Parameter spezifiziert die Arbeitsgruppe, in der ein Samba-Server erscheinen wird. Samba (`nmbd`) setzt den über diesen Parameter spezifizierten Namen als Arbeitsgruppe in Host-Announcements ein. Er wird auch als Domänenname für den Samba-Server verwendet, wenn der Domain-Modus aktiviert ist oder Samba als Primary Domain Controller operiert. Der Standardwert ist:

```
workgroup = Arbeitsgruppe
```

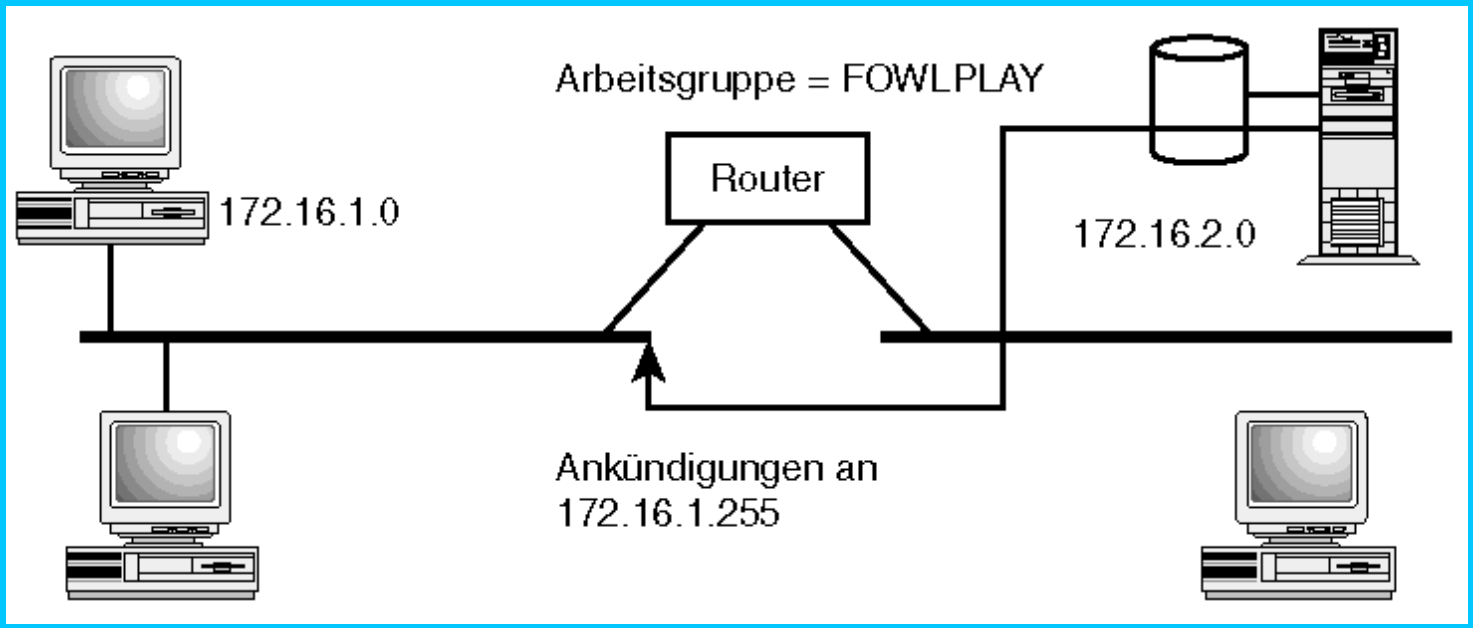
Beispielkonfigurationen

In diesem Abschnitt werden Sie sich einige Beispielkonfigurationen ansehen. Zunächst stelle ich die Verwendung von `remote announce` dar, damit eine Arbeitsgruppe, die über Subnetze verteilt ist, ihre Samba-Server durchsuchen kann. Danach untersuchen Sie die Einrichtung eines Domain-Master-Browsers, um eine Domäne einzurichten, die über mehrere Subnetze verteilt ist.

Arbeitsgruppen über Subnetzgrenzen

Wenn Sie eine Arbeitsgruppe haben, die über mehrere Subnetze verteilt ist, können Sie den Parameter `remote announce` verwenden, damit Clients in anderen Subnetzen mit den Teilen des Netzwerks in Kontakt bleiben können, in denen sich die Server befinden. Abbildung 20.4 stellt ein Beispiel hierfür dar.

Abb. 20.4: Eine Arbeitsgruppe über Subnetzgrenzen zusammenhalten



Sie müssten folgende Einstellung verwenden, um sicherzustellen, dass das Subnetz `172.16.1.0` von dem Server in Subnetz `172.16.2.0` weiß:

```
remote announce = 172.16.1.255/FOWLPLAY
```

Dies garantiert, dass alle Knoten innerhalb von `172.16.1.0` von der Existenz des Samba-Servers in `172.16.2.0` hören. Es hilft jedoch den Knoten in `172.16.2.0` nicht, etwas über die Server herauszufinden, die eventuell in `172.16.1.0` sind.

Damit etwas Derartiges funktionieren kann, müssen Sie sicherstellen, dass Ihr Router gerichtete Broadcasts weiterleitet. Funktionieren gerichtete Broadcasts zwischen den zwei Subnetzen nicht, können Sie `172.16.1.255` durch die IP-Adresse des lokalen Master-Browsers im Subnetz ersetzen.

Wenn Sie in jedem Ihrer Subnetze Samba-Server haben, können Sie den Parameter `remote browse sync` verwenden, um die Browse-Listen zwischen allen Subnetzen synchron zu halten. Um dies zu tun, führen Sie folgende Schritte durch:

1. Bestimmen Sie einen Ihrer Samba-Server als Domain-Master-Browser. Dafür fügen Sie in den globalen Abschnitt der `smb.conf` auf dem Rechner, der Domain-Master-Browser werden soll, den Parameter `domain master = yes` ein.
2. Spezifizieren Sie über den Parameter `remote browse sync` die entfernten Netzwerke, mit denen Samba Browse-Listen synchronisieren soll.
3. Stellen Sie sicher, dass die Samba-Server in jedem entfernten Netzwerk die lokalen Master-Browser verwenden.

Wenn Ihr Netzwerk die Subnetze `172.16.1.0`, `172.16.2.0` und `172.30.3.0` umfasst und Ihr Domain-Master-Browser in `172.30.1.0` auf dem Samba-Server ist, würden Sie folgende Parameter in den globalen Abschnitt der `smb.conf`-Datei einfügen:

```
domain master = yes
remote browse sync = 172.16.2.255 172.16.3.255
```

Leiten Ihre Router keine gerichteten Broadcasts weiter, müssen Sie die Broadcast-Adressen durch die IP-Adresse des Samba-Servers ersetzen, der lokaler Master-Browser in jedem der aufgelisteten Subnetze ist.

Auf dem Samba-Server, der lokaler Master-Browser in jedem der entfernten Subnetze sein soll, sollten Sie die folgenden Parameter in den globalen Abschnitt der `smb.conf`-Datei einfügen:

```
local master = yes
preferred master = yes
os level = 33
```

Sie können den Wert für den Parameter `os level` auf jeden Wert höher als `32` setzen.

Domänen über Subnetzgrenzen

Wenn Sie eine Domäne haben, die über mehrere Subnetze verteilt ist, müssen Sie wahrscheinlich gar nichts zusätzlich tun, damit alles korrekt funktioniert. Das liegt daran, dass Windows-NT-Domänen diese Situation bereits meistern.

Wenn Sie Samba jedoch als Primary Domain Controller einrichten, fügen Sie folgenden Eintrag in den globalen Abschnitt Ihrer `smb.conf`-Datei ein:

```
domain master = yes
```

Zusätzlich sollten Sie sicherstellen, dass Sie Folgendes in der `smb.conf`-Datei jedes Samba-Servers in Subnetzen haben:

```
local master = yes
```

Diese Einstellungen stellen sicher, dass es im Netzwerk einen Domain-Master-Browser gibt und dass irgendein Knoten versucht, lokaler Master-Browser in jedem Subnetz zu werden.

Wenn Sie bereits einen Windows-NT-Server als Primary Domain Controller haben, sollten Sie kein Samba-System zum

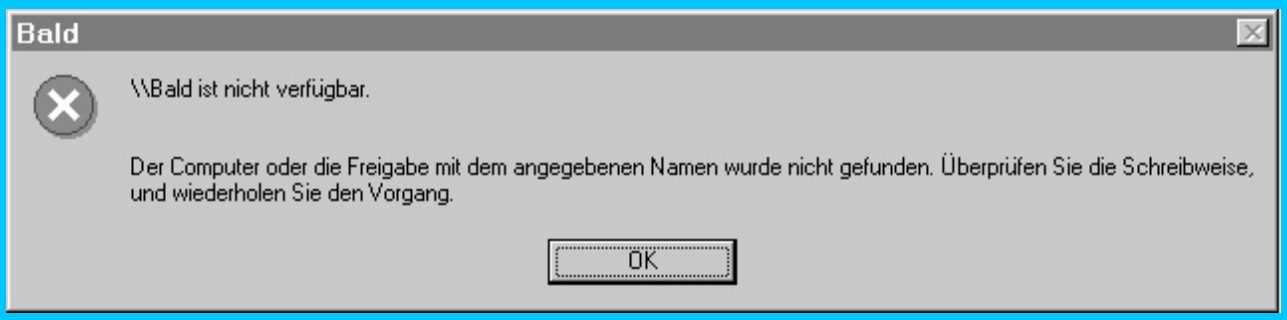
Domain-Master-Browser machen. Dies kann viele Clients verwirren, die dann versuchen, sich in Ihren Samba-Server statt in den Primary Domain Controller einzuloggen.

Troubleshooting

Wenn Sie Probleme in Bezug auf das Browsing beheben wollen, denken Sie daran, dass Browse-Listen nur Namen enthalten - die Namen der Server, die Host-Ankündigungen in der Arbeitgruppe oder Domäne gesendet haben, und die Namen der Domänen, die vom Master-Browser angekündigt wurden. Das bedeutet, dass es sehr wichtig ist, dass WINS in Ihrem Netzwerk korrekt funktioniert.

Wenn Sie alle Server in der Netzwerkumgebung sehen, aber keine Liste der Freigaben auf einem bestimmten Server erhalten können, sollten Sie sicherstellen, dass die WINS-Server funktionieren und Ihre Clients auf die WINS-Server zugreifen können. Abbildung 20.5 zeigt ein Beispiel für eine Fehlermeldung, die Sie sehen könnten.

Abb. 20.5: Fehlermeldung beim Zugriff auf einen Server in der Netzwerkumgebung



Sie können noch auf folgende Probleme treffen:

- Sie können keine Knoten in der Netzwerkumgebung sehen oder Sie sehen die Knoten nicht mehr, die vorher da waren. Dieses Problem liegt an Browsern, die ihre Rolle ändern und oft Wahlen durchführen. Sie sollten sicherstellen, dass Samba Wahlen nicht zu oft erzwingt. Sie können dies überprüfen, wenn Sie sich die Datei `log.nmb` im Samba-Log-Verzeichnis ansehen.
- Sie werden aufgefordert, ein Passwort anzugeben, wenn Sie einen Server durchsuchen wollen. Dieses Problem liegt normalerweise darin begründet, dass Sie keinen Account auf dem Rechner haben, den Sie durchsuchen wollen, oder dass Ihr Passwort für diesen Rechner ein anderes ist als das, mit dem Sie sich in das Netzwerk eingeloggt haben.

Zusammenfassung

In diesem Kapitel haben Sie sich angesehen, wie Browsing in einem Netzwerk mit Routern funktioniert. Sie haben gesehen, dass Samba sowohl Arbeitsgruppen als auch NT-Domänen, die über mehrere Subnetze verteilt sind, unterstützen kann.

In Kapitel 21, »Windows-9x-Domänenkontrolle«, werden Sie untersuchen, wie DOS- und Windows-Clients (Windows für Workgroups und Windows 9x) sich in eine von Samba verwaltete Domäne einloggen können. In einem weiteren Kapitel werden Sie sich die Primary-Domain-Controller-Funktionen ansehen, die in Samba integriert sind.

Frage & Antwort

F. Wir haben eine Reihe von Samba-Servern in einem Netzwerk, das bereits eine Windows-NT-Domäne mit eigenem Primary Domain Controller hat. Einige unserer Client-Rechner wollen sich anscheinend in den Samba-Server einloggen und können sich deshalb nicht in der Domäne anmelden. Was ist das Problem und wie können wir es beheben?

. Ihr Problem liegt wahrscheinlich darin, dass Samba als Domain-Master-Browser konfiguriert ist. Wird er hochgefahren, bevor der PDC gestartet wird, hat der Samba-Server den Domänennamen `DOMÄNE<1B>` registriert (wobei `DOMÄNE` durch den Namen Ihrer Domäne ersetzt wird). Entfernen Sie alle Domain-Master-Parameter aus der Datei `smb.conf` auf Ihrem Samba-Server und starten Sie Samba neu. Sie müssen eventuell auch Ihren Primary Domain Controller neu starten.

F. Unsere Arbeitsgruppe ist über mehrere Subnetze verteilt, und wir haben Samba-Server in jedem Subnetz. Wir haben Probleme mit dem Browsing. Ich habe gelesen, dass nur eine Windows-NT-Domäne das Browsing über Subnetzgrenzen unterstützt. Muss ich unseren Haupt-Samba-Server durch einen Windows-NT-PDC ersetzen, um Browsing über das gesamte Netzwerk bieten zu können?

. Zwar unterstützt das Microsoft-Windows-Netzwerkmodell das Browsing über Subnetzgrenzen in einer Arbeitsgruppe nicht, Samba aber schon. Sie müssen Ihren Haupt-Samba-Server nicht ersetzen. Sie können ihn einfach als Domain-Master-Browser konfigurieren, den Parameter `remote browse sync` aktivieren und sicherstellen, dass Ihre Samba-Server in den entfernten Subnetzen lokale

Master Browser sind.

F. Wir können Server in entfernten Subnetzen in unserem Netzwerk durchsuchen, aber wenn jemand versucht, die auf einem Server zur Verfügung gestellten Freigaben zu durchsuchen, erhält er eine Fehlermeldung wie »Network name not found« und es funktioniert nicht. Wie können wir dies korrigieren?

. Dies liegt normalerweise daran, dass Sie WINS nicht korrekt in den Clients im entfernten Netzwerk eingerichtet haben. Browse-Listen stellen nur die NetBIOS-Namen von Servern und Domänen zur Verfügung. Die Auflösung eines NetBIOS-Namens in eine IP-Adresse in einem Router-Netzwerk verlangt die Benutzung von WINS (oder einer `lmhosts`-Datei, aber WINS ist besser). Wenn Sie Clients haben, die WINS nicht verstehen, müssen Sie einen Samba-Server im lokalen Subnetz als WINS-Proxy einrichten.

Tag 21: Windows-9x-Domänenkontrolle

»Nein, es tut mir leid ... ich verstehe. Ja, ich weiß, dass Ihre Bookmarks in Netscape wichtig sind. Es ist nur so, dass sie auf der lokalen Festplatte Ihres PC gespeichert waren, und als er abstürzte ... nun, ich kann leider nichts für Sie tun.« Ich höre ein lautes Klicken, als der andere Teilnehmer an dieser Unterhaltung wütend auflegt.

»Es muss einen besseren Weg geben«, seufze ich. »Das Beste wäre, sein Netscape-Profileverzeichnis auf einem Netzwerklaufwerk zu speichern. Dann würde es mit in die allnächtlichen Backups aufgenommen werden. Ich muss aber die Cache-Dateien gar nicht speichern. Ich könnte das Caching ausschalten ... oder ich könnte das Cache-Verzeichnis auf der lokalen Festplatte einrichten. Nachdem die Lesezeichendateien auf einem Netzwerklaufwerk sind, könnte ich sogar je nach Bedarf Einstellungen ändern und Probleme behandeln, ohne in irgendein Büro gehen oder die Dinge über das Telefon erklären zu müssen!«

»Das gefällt mir«, sage ich mit einem halben Grinsen. Es ist ein Grinsen der Zufriedenheit. »Hmmm, aber wie kann ich garantieren, dass das Home-Verzeichnis des Benutzers gemountet wird und sich auf dem richtigen Laufwerksbuchstaben befindet?«, frage ich mich. «Wenn ich nur sicherstellen könnte, dass jeder beim Einloggen die gleichen Freigaben mountet.«

Kennen Sie das? Wenn Sie schon einmal PCs in einem beliebigen Netzwerk verwaltet haben, haben Sie wahrscheinlich all die Geschichten und Beschwerden gehört. In diesem Kapitel möchte ich Ihnen zeigen, wie Sie Samba als *Domain Controller (DC)* für Windows-9x-Clients einrichten können, um einige dieser Probleme zu lösen. Das gleiche Setup funktioniert auch für Windows für Worksgroups und den MS-DOS-Client, aber ich werde mich hier nicht auf diese konzentrieren.

Ich hoffe, dass Sie nach der Lektüre dieses Kapitels einige Mechanismen kennen gelernt haben, die Ihr Leben etwas einfacher machen. Ich weiß, dass einige dieser Dinge mir wirklich geholfen haben.

Domänen und Arbeitsgruppen

Erinnern Sie sich noch daran, dass ich in Kapitel 2, »Windows-Netzwerke«, über Domänen und Arbeitsgruppen gesprochen habe? Ich hoffe, dass die zwei Konzepte mittlerweile etwas mehr Sinn ergeben, falls Sie noch nicht mit ihnen vertraut waren. Lassen Sie uns die zwei Abbildungen noch einmal ansehen (Abbildung 21.1 und 21.2).

Abb. 21.1: Beispiel für eine Arbeitsgruppe

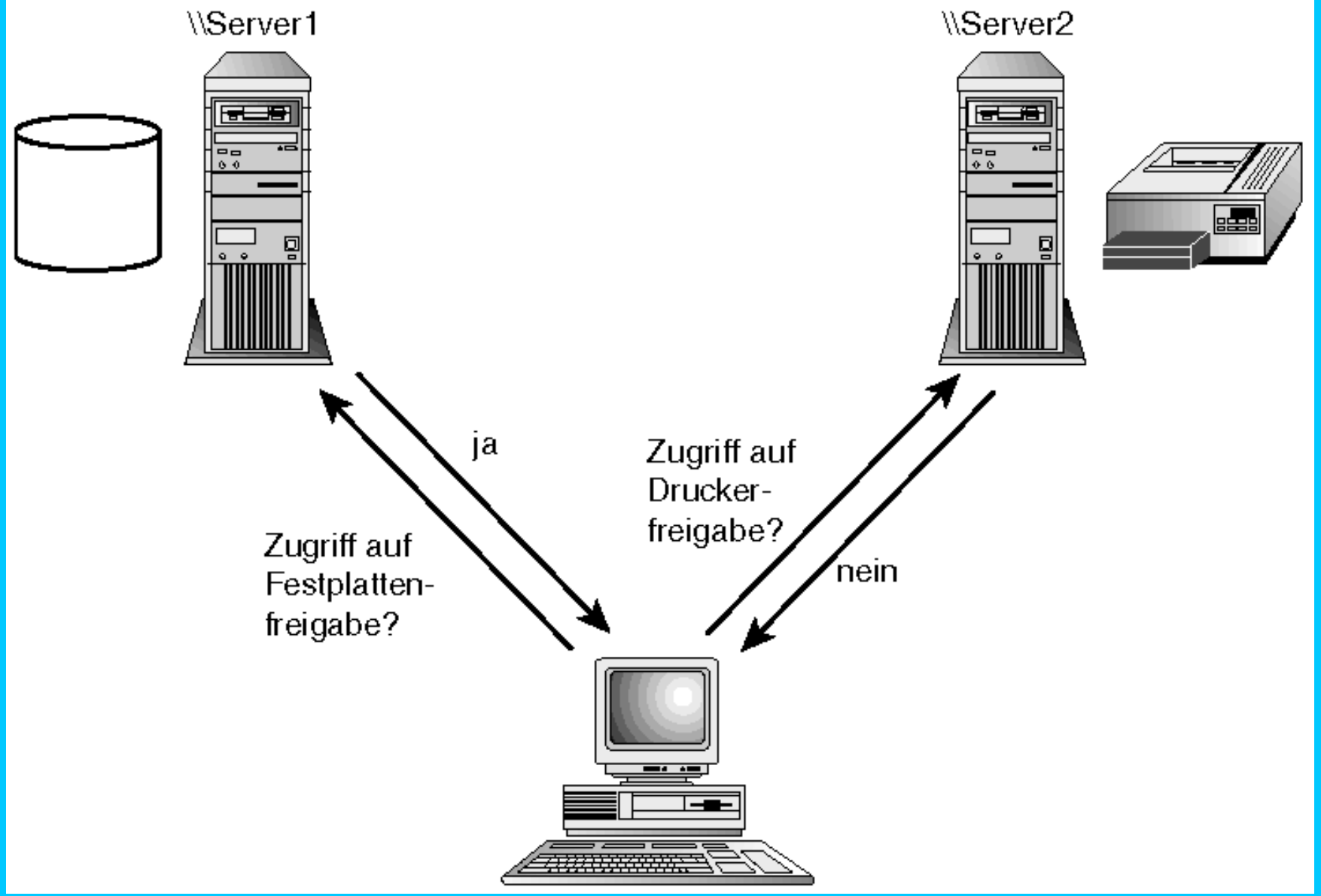
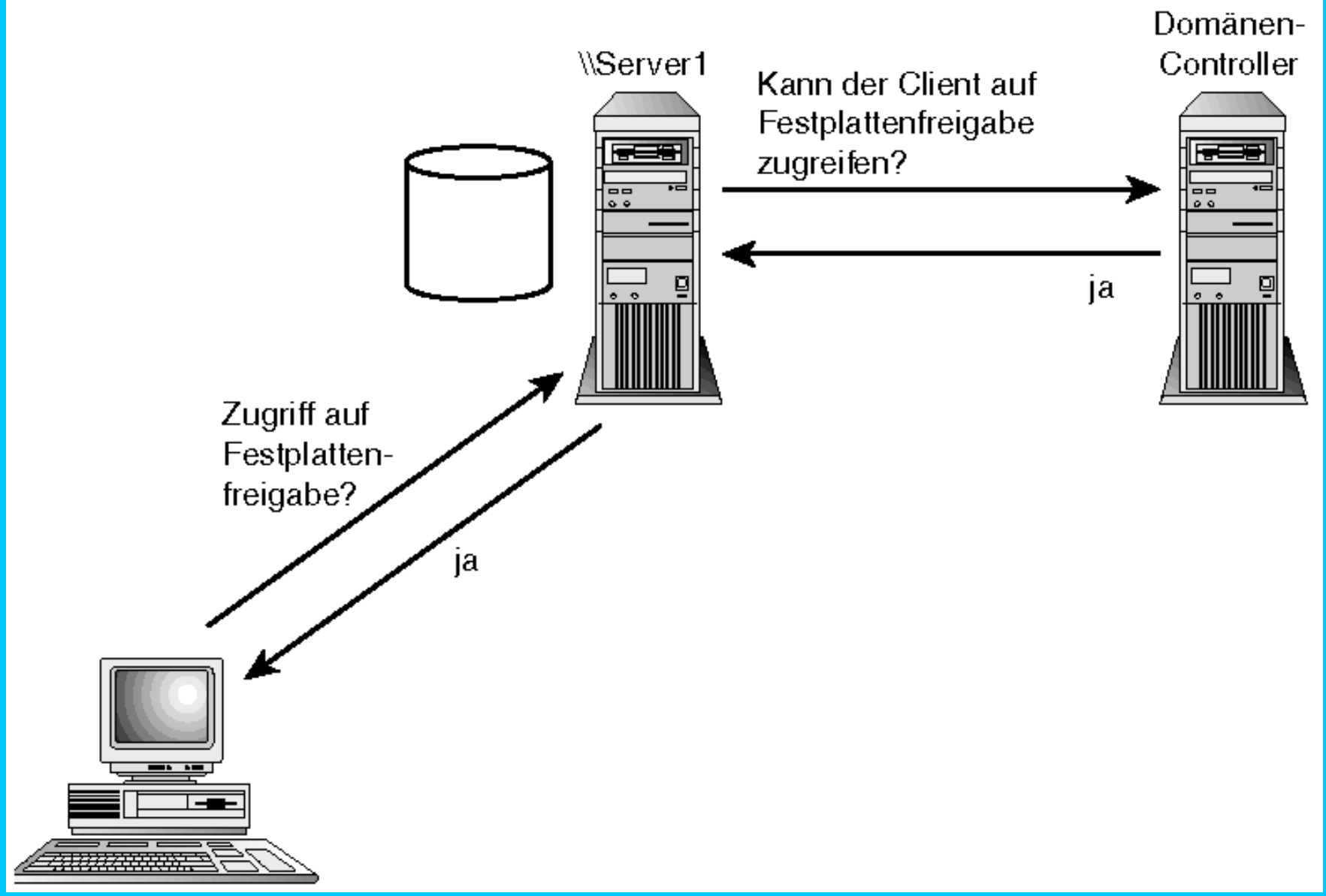


Abb. 21.2: Beispiel für eine Domäne



Während Sie sich die Abbildungen 21.1 und 21.2 ansehen, denken Sie an das Konzept des Einloggens in das Netzwerk. An diesem Punkt ist *das Netzwerk* eine vage Idee, so wie wenn jemand sagt: »Nun, du weißt, was die Leute sagen«. Wer sind *die Leute*? Niemand weiß es genau.

Der Unterschied zwischen der Anmeldung in einem Netzwerk in einer Arbeitsgruppenumgebung und einem anderen in einer Domänenumgebung besteht darin, dass in einer Domäne das Login wirklich authentifiziert wird! In einer Arbeitsgruppe werden die Benutzerinformationen vom lokalen Rechner einfach zwischengespeichert, um sie im Falle einer Verbindung an andere Server weiterzuleiten. Wenn Sie nun noch einmal zu Abbildung 21.2 zurückgehen, werden Sie feststellen, dass der gleiche DC, der das Login authentifiziert hat, jede Verbindung zu einer Freigabe authentifiziert, die von einem anderen Mitglied der Domäne zur Verfügung gestellt wird. Diese Realisierung bringt uns zu meinen neuen Illustrationen von Arbeitsgruppen und Domänen in den Abbildungen 21.3 und 21.4.

Wie hilft uns das weiter? Zunächst einmal, wenn der Benutzer beim Login in eine Arbeitsgruppe ein falsches Passwort oder einen falschen Benutzernamen eingibt, akzeptiert das Windows-9x-System dies, ohne eine Reaktion zu zeigen. Ohne Verbindung zu einem Server ist es für das lokale System natürlich nicht möglich, eine Reaktion zurückzugeben.

Lassen Sie mich hier einen kurzen Rückzieher machen. Windows 9x kann den Benutzernamen und das Passwort anhand einer lokalen Passwort-Cache-Datei authentifizieren. Diese Dateien können im Verzeichnis `\windows` gefunden werden und haben die Erweiterung `*.pwl`. Meiner Meinung nach sind diese Dateien teuflisch. Vielleicht nicht so teuflisch, dass sie das Ende der Welt bedeuten, aber immerhin. Sie verwenden schwache Verschlüsselungsalgorithmen und sind leicht genug zu knacken. Daher stellen sie ein Loch in meinem heiligen Netzwerk-Sicherheitsmodell dar und müssen entsprechend behandelt werden.

Abb. 21.3: Einloggen in das Netzwerk in einer Arbeitsgruppenumgebung

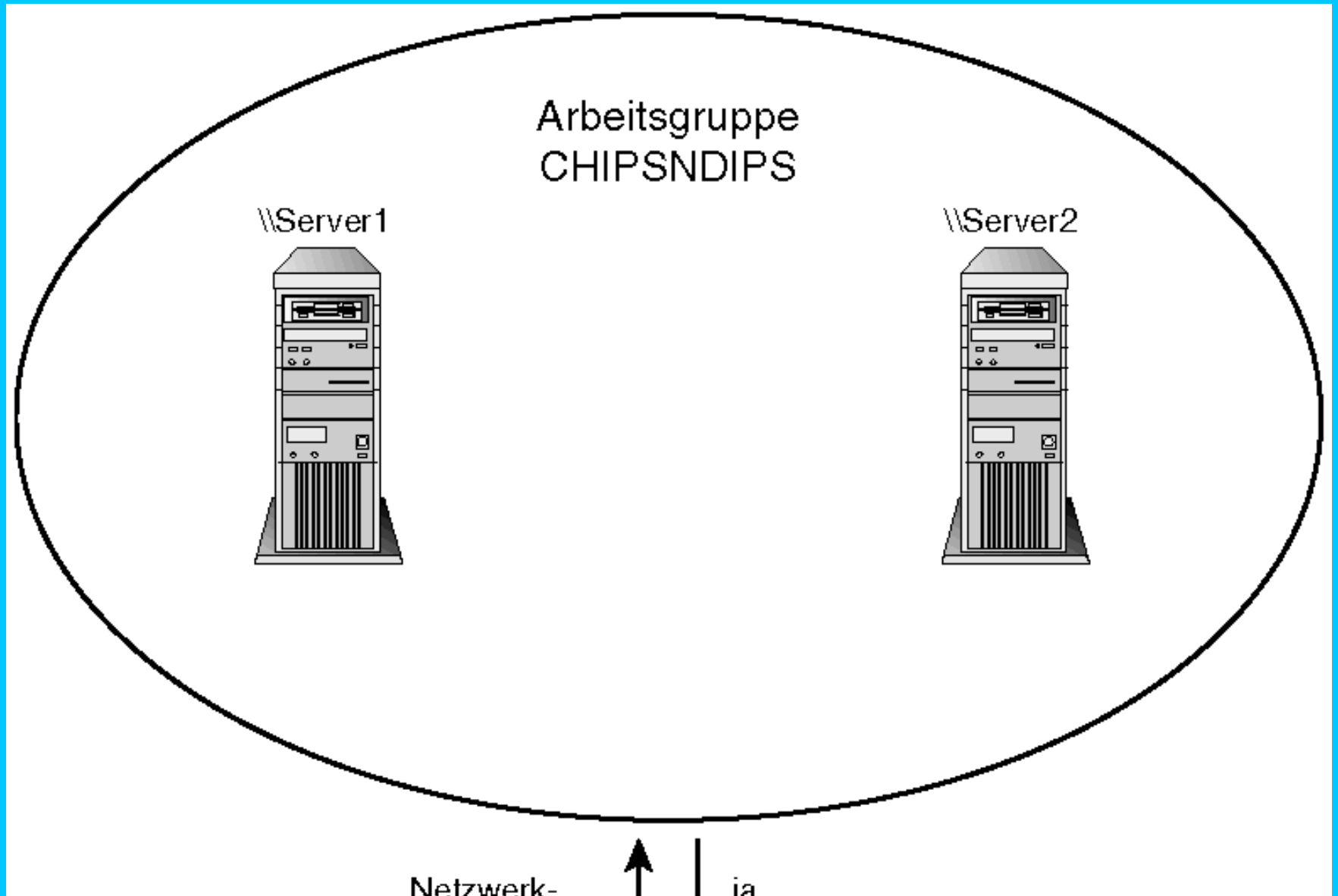
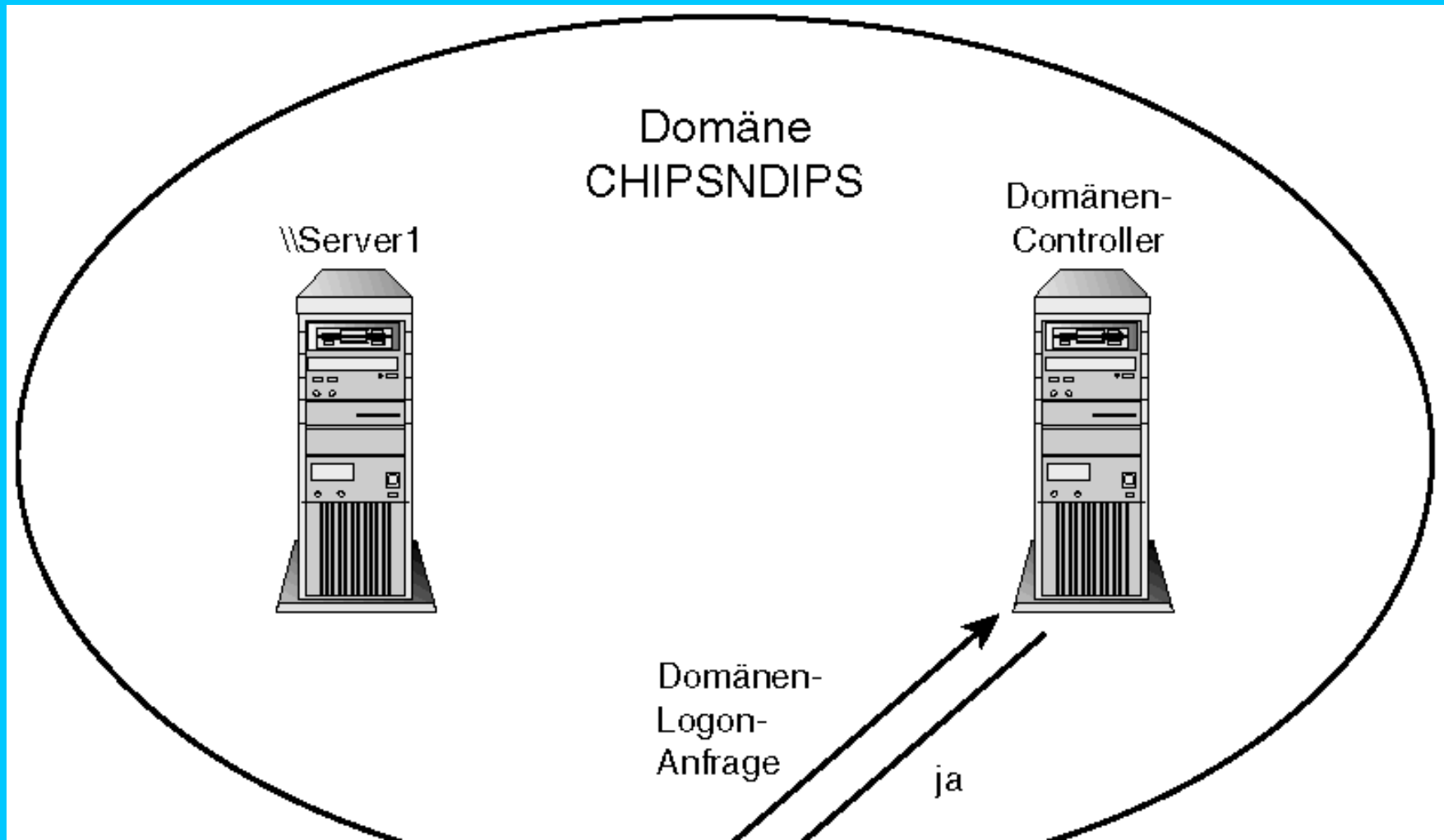
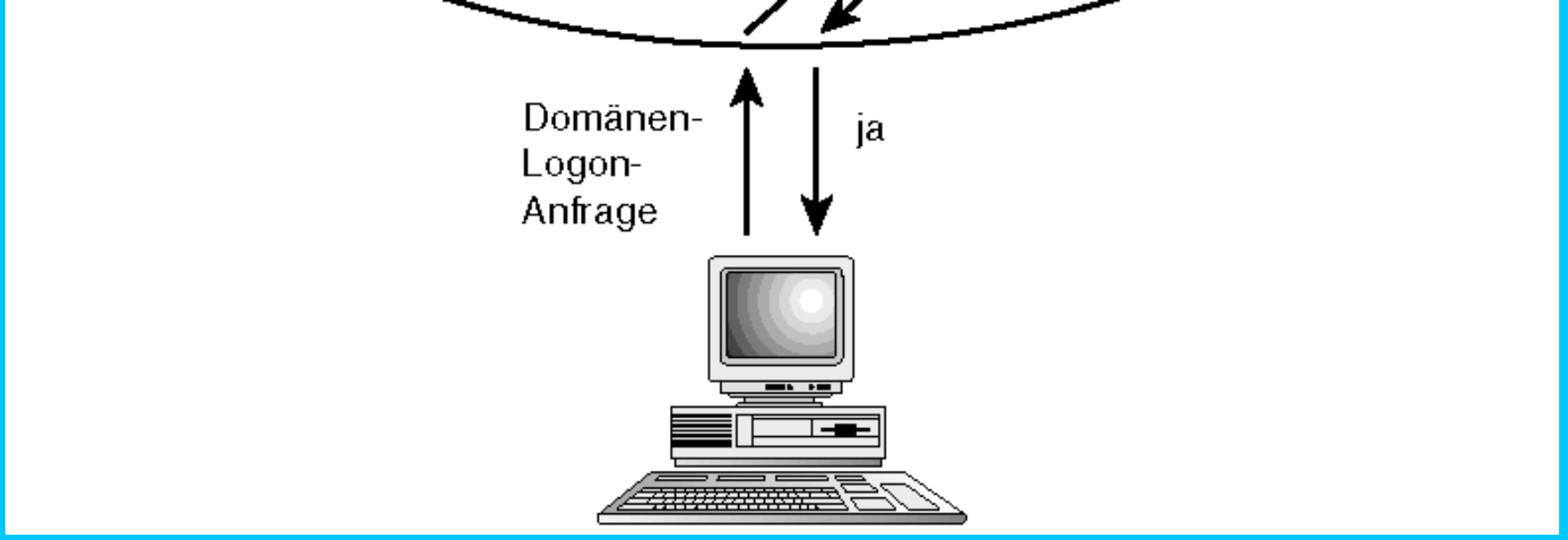


Abb. 21.4: Einloggen in das Netzwerk in einer Domänenumgebung





Okay. Vielleicht habe ich heute zu viel Kaffee getrunken, aber hier ist der Knüller. Passwort-Cache-Dateien sind leicht zu knacken. Sie sind außerdem ärgerlich. Wenn Sie Ihr Passwort auf dem DC ändern, müssen Sie Ihr Passwort auch auf dem lokalen Windows-Rechner ändern. Der Hauptzweck für eine Passwort-Cache-Datei besteht darin, alle Passwörter zu sammeln, die ein Benutzer für den Zugriff auf Server in einer Arbeitsgruppe braucht. Wenn Sie sich in das System mit `pass1` einloggen und sich dann mit `pass2` mit `\\server-a` verbinden, versucht Windows zuerst, sich mit `pass1` und dann mit `pass2` mit dem Server zu verbinden, das in Ihrer entsprechenden `*.pwl`-Datei gespeichert wurde. Windows öffnet Ihre Passwort-Cache-Datei mit Ihrem Login-Passwort als Schlüssel, damit das System alle in der Datei gefundenen Passwörter verwenden kann. In einer Domänenumgebung braucht ein Benutzer nur ein Passwort, also ist der ursprüngliche Gedanke hinter Passwort-Cache-Dateien nicht mehr gültig. Könnte jemand eine dieser `*.pwl`-Dateien, die ein Domänenpasswort enthält, stehlen und knacken, hätte der Eindringling außerdem Zugriff auf jeden Server, der das Recht verteilt, sich über das Netzwerk in diesen Account einzuloggen.

Aus diesen Gründen setze ich voraus, dass Sie das Passwort-Caching deaktiviert haben. Dies können Sie über den Policy-Editor tun, über den ich am Ende dieses Kapitels kurz reden werde, oder Sie können einfach alle `*.pwl`-Dateien in Ihrem `\windows`-Verzeichnis löschen, den folgenden Eintrag in die lokale System-Registry einfügen und dann neu starten:

```
[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Network]
"DisablePwdCaching"=dword:00000001
```

Das Bearbeiten der Registry kann Ihr System unbrauchbar machen. Gehen Sie mit äußerster Sorgfalt vor, wenn Sie den Registrierungseditor (`regedit.exe`) benutzen.

Ist das Passwort-Caching deaktiviert, kann ich unbeirrt zu meiner ursprünglichen Aussage stehen: Beim Einloggen in eine Arbeitsgruppe findet keine Authentifizierung statt, bis die erste Verbindung zu einem SMB-Server eingegangen wird.

Jetzt kehre ich wieder zu meinem Ausgangspunkt zurück. Wenn sich ein Benutzer in eine Arbeitsgruppe einloggt, kann das lokale System keine Reaktion zur Gültigkeit des Benutzername/Passwort-Paars zurückgeben. Wenn sich ein Benutzer jedoch in eine Domäne einloggt, werden der Benutzername und der Beweis für die Identität an den Domänen-Controller gesendet, der eine Antwort zurückgibt, in der das Login entweder akzeptiert oder abgelehnt wird.

Wie hilft Ihnen das alles? In einem gewissen Sinne hilft es mehr dem Benutzer. Nehmen wir an, ein Benutzer hat den Login-Namen `beckett`. Nehmen wir weiterhin an, dass er sich in die Arbeitsgruppe einloggt, aber seinen Namen falsch als `beclett` eingibt. Jedesmal, wenn der Benutzer versucht, sich während dieser Arbeitssitzung mit einer Netzwerkressource zu verbinden, übertragen die lokalen Sitzungen den Namen `becklett` als Benutzernamen für die Sitzungsaufnahme. Dies würde wahrscheinlich in Anrufen an Ihre Supportabteilung enden, in denen sich der Benutzer beschwert, dass »das Netzwerk mein Passwort nicht mag, aber ich weiss, dass ich es richtig eingebe«. Würde die gleiche Situation in einer Domänenumgebung stattfinden, würde der DC sich über ein falsches Benutzername/Passwort-Paar beschweren, damit das lokale System dies dem Benutzer mitteilen kann.

Das ist genug der theoretischen und philosophischen Erklärungen der Domänenkontrolle. Lassen Sie uns nun ansehen, wie Sie sie implementieren können und mit welcher Art von Tricks und kleinen Spielzeugen ich arbeite.

Den Samba-Domain-Controller einrichten

Die erste Voraussetzung für die Konfiguration von Samba als Domänen-Controller ist, dass der Server im User-Modus operiert. `security = server` funktioniert ebenfalls, da Samba dann bekannt gibt, dass es den User-Modus verwendet.

```
security = user
```

So ist es. Samba kann nicht im Share-Modus arbeiten und gleichzeitig als Domain-Controller agieren.

Danach sollten Sie sicherstellen, dass Samba der Master-Browser für die Domäne ist. Denken Sie an die NetBIOS-Ressourcentypen, über die ich in Kapitel 2 gesprochen habe, die mit Namen verbunden sind. Ein Domänen-Controller, ob es sich nun um einen Windows-NT-PDC oder einen Samba-Rechner handelt, muss den NetBIOS-Namen *`Domänenname`*`<1b>` erfolgreich registrieren können. Der Ressourcentyp `<1b>` bezeichnet den Master-Browser für eine Domäne, und darüber lokalisieren Windows-Clients den Login-Server der Domäne. Dies sollte aus den Kapiteln 19, »Browsing in lokalen Subnetzen«, und 20, »Browsing in Netzwerken mit Routern«, klar sein. Die einfachste Methode, dies jetzt zu konfigurieren, besteht im Einfügen folgender Parameter in Ihre `smb.conf`-Datei:

```
os level = 64
domain master = yes
local master = yes
preferred master = yes
```

Die dritte Voraussetzung ist, dass Sie Samba mitteilen, dass der Server als Domänen-Controller agieren sollte, indem Sie den Parameter `domain logons` setzen:

```
domain logons = yes
```

Und schließlich müssen Sie eine Freigabe namens `[netlogon]` in der `smb.conf` erstellen. Alle Windows-9x-Clients, die versuchen, sich in eine Domäne einzuloggen, verbinden sich mit dieser Freigabe. Die Freigabe `[netlogon]` muss nicht unbedingt Daten enthalten, sie muss nur existieren, und Benutzer

müssen sich erfolgreich mit ihr verbinden können. Hier ein einfaches Beispiel, das Sie benutzen werden:

```
[netlogon]
path = /export/smb/netlogon
writeable = no
public = no
```

Ich möchte hier eine Sache erwähnen, die optional ist, die ich aber sehr zu schätzen gelernt habe. Mit der Domänenkontrolle kommt die Fähigkeit, eine Batch-Datei zu spezifizieren, die gestartet wird, wenn ein Benutzer sich erfolgreich in die Domäne einloggt. Diese Batch-Datei sollte irgendwo in die `[netlogon]`-Freigabe platziert werden; sie wird über den Parameter `logon script` spezifiziert. Der Dateiname selbst kann jedes der Standard-`smb.conf`-Makros enthalten, aber Parameter, die an die Batch-Datei geleitet werden, können dies nicht. Die folgende Einstellung z.B. funktioniert genauso, wie Sie es sich vorstellen:

```
logon script = %U.bat
```

Das Login-Skript für eine Verbindung wird auf *`Benutzername`*`.bat` gesetzt, wobei *`Benutzername`* aus den Informationen zur Sitzungsanfrage geholt wird.

Würde sich der Benutzer `beckett` erfolgreich einloggen, so würde der Windows-Client folgende Datei zu starten versuchen:

```
\\server\netlogon\beckett.bat
```

Wenn Sie jedoch den aktuellen Benutzernamen als Parameter an die Batch-Datei weitergeben wollten, würde Folgendes nicht funktionieren (die Batch-Datei würde `%U` tatsächlich als einzelnes Befehlszeilenargument erhalten):

```
logon script = logon.bat %U
```

Für dieses Beispiel werden Sie eine einzige Batch-Datei für alle Benutzer verwenden. Das Skript führt nur einen Befehl aus, der das Home-Verzeichnis des Benutzers mountet. Fügen Sie der `smb.conf` folgenden Eintrag hinzu:

```
logon script = logon.bat
```

Erstellen Sie nun eine Textdatei in `\\server\netlogon` mit dem Namen `logon.bat`. Stellen Sie außerdem sicher, dass die Batch-Datei DOS-formatierten Text mit einem CR- und einem LF-Zeichen am Ende jeder Zeile verwendet. Sie können die Datei entweder mit dem in Windows integrierten Editor erstellen oder mit `vi` und jeder Zeile eine `[Strg]`+`[V]`- und `[Strg]`+`[M]`-Sequenz anhängen. Hier ist die `logon.bat`-Datei, die Sie für diese Beispiele verwenden werden:

```
echo Mapping home directory...
net use h: \\picante\homes
```

Aus Gründen der Vollständigkeit zeigt Listing 21.1 die gesamte `smb.conf`-Datei, die Sie für Ihren Server eingerichtet haben. Sie werden bemerken, dass Sie Klartextpasswörter benutzen, aber Sambas Funktionen zur Domänenkontrolle funktionieren auch mit verschlüsselten Passwörtern. Für jede Methode sollten Sie sicherstellen, dass Sie die Benutzer-Accounts korrekt eingerichtet haben. Wenn Sie Klartextpasswörter verwenden, überprüfen Sie, ob Sie diese Funktion auf Windows-98-Clients und den benötigten Windows-95-Clients aktiviert haben (siehe Kapitel 12, »Fallstudie: Einen NT-Datei- und Drucker-Server ersetzen«).

Listing 21.1: Samba-Konfigurationsdatei für einen einfachen Windows-9x-Domain-Controller

```
;
;    Thu Jan 23 11:00:27 CST 1999
```

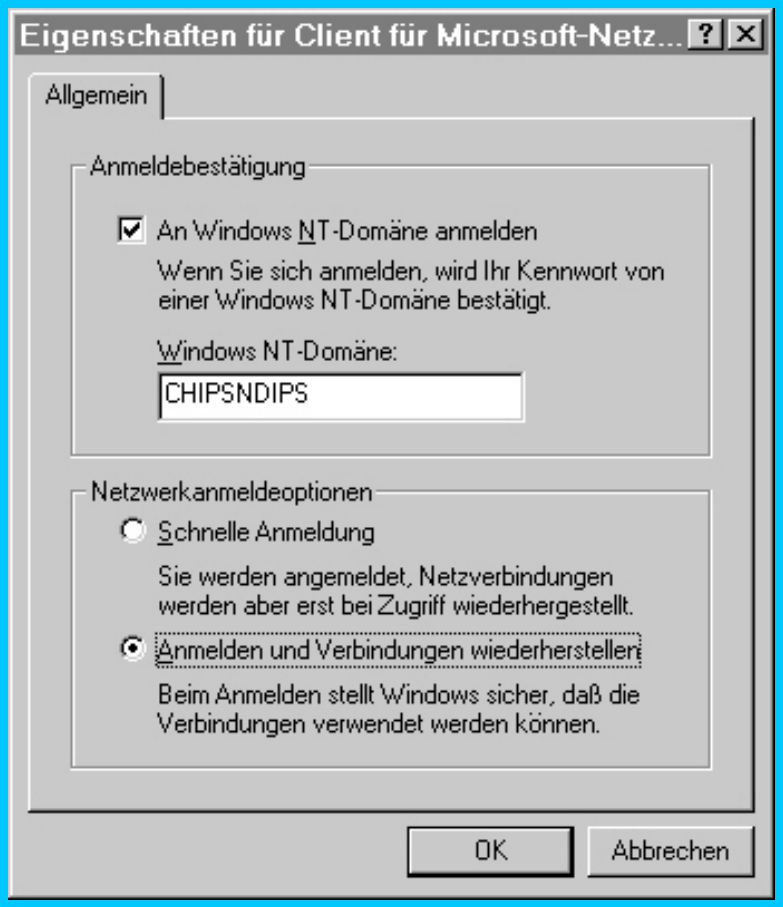
```
;    Jerry Carter
;
;    Sams Teach Yourself Samba in 24 Hours
;
;    smb.conf for Windows 9x domain controller[global]
    netbios name = picante
    workgroup = CHIPSNDIPS
    security = user
    password level = 4
    domain logons = yes
    logon script = logon.bat

    os level = 64
    domain master = yes
    local master = yes
    preferred master = yes

    browseable = yes
    writeable = yes
    locking = no
    case sensitive = no
    default case = lower
    preserve case = yes
    short preserve case = no

[netlogon]
    comment = NETLOGON service
    path = /export/smb/netlogon
    locking = no
    public = no
    writeable = no

[homes]
    comment = Home directories for CHIPSNDIPS domain users
    path = %H
    create mode = 0600
    directory mode = 0700
    browseable = no
    valid users = %S
```

Einen Windows-9x-Client einrichten

Nachdem Sie nun den Samba-Server konfiguriert und gestartet haben, besteht der nächste Schritt darin, den Windows-Client zu konfigurieren, damit er sich in die Domäne einloggt. Vorausgesetzt, dass Sie den Windows-Rechner bereits für den Zugriff auf den SMB-Server konfiguriert haben, sind nur wenige Schritte notwendig. Abhängig von Ihrer Netzwerktopologie, kann sogar nur ein Schritt nötig sein.

Abbildung 21.5 zeigt das Fenster *Eigenschaften für den Client für Microsoft-Netzwerke*. Um darauf zuzugreifen, öffnen Sie das Netzwerkkontrollfenster, markieren den Eintrag *Client für Microsoft-Netzwerke* und klicken auf die Schaltfläche *Eigenschaften*. Wählen Sie die Option *An Windows NT-Domäne anmelden* im Abschnitt *Anmeldebestätigung*, und geben Sie den Namen der Domäne ein, die Sie benutzen wollen. Klicken Sie dann so lange auf *OK*, bis das Netzwerkfenster wieder geschlossen wird. An diesem Punkt will Windows eventuell einige Dateien von der Windows-95-Installations-CD kopieren (warum das System Dateien kopieren muss, wenn der Client bereits installiert ist, ist für mich nicht nachvollziehbar, aber ...). Danach müssen Sie das System neu starten, damit die Änderungen in Kraft treten können.

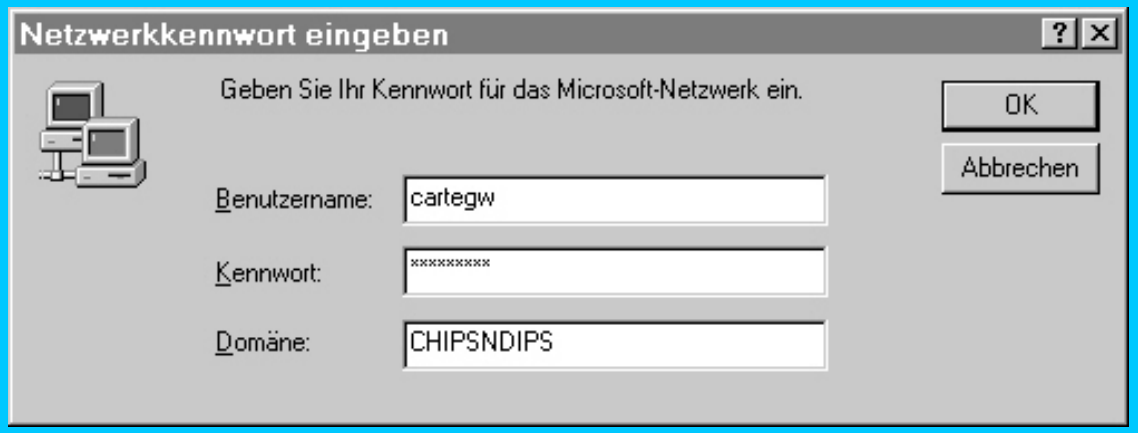
Abb. 21.5: Einen Windows-95-OSR2-Client konfigurieren, damit er sich in die Domäne CHIPSNDIPS einloggt



Testen und Troubleshooting

Nachdem der PC neu gestartet ist, sollten Sie das vertraute Login-Dialogfeld sehen, mit der Ausnahme, dass es diesmal ein zusätzliches Feld mit dem Domänennamen enthält (siehe Abbildung 21.6), der in Abbildung 21.5 spezifiziert wurde.

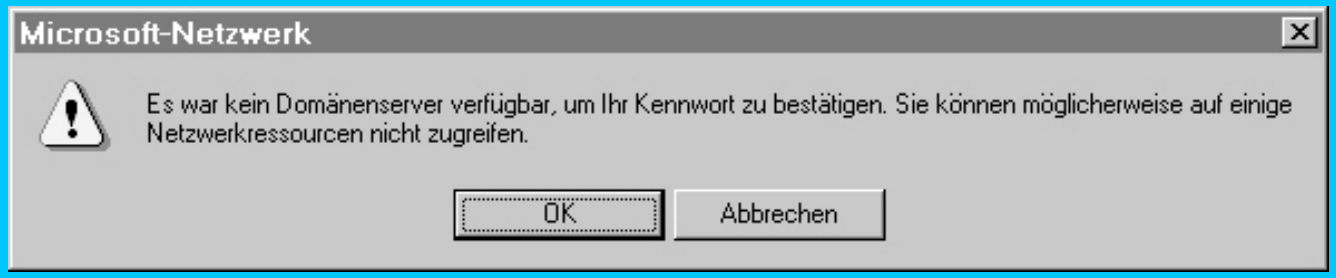
Abb. 21.6: Das Netzwerk-Login-Dialogfeld unter Windows, das die Domäne (CHIPSNDIPS) enthält, über die authentifiziert wird



Es war kein Domänenserver verfügbar, um Ihr Kennwort zu bestätigen ...

Sie werden möglicherweise zwei übliche Fehlermeldungen sehen, wenn Sie versuchen, sich in die Domäne einzuloggen. Abbildung 21.7 zeigt die erste. Die Meldung ist Windows Art und Weise, Ihnen mitzuteilen: »Ich habe herumgefragt, um zu sehen, ob irgend jemand versuchen würde, diesen Benutzernamen und dieses Passwort für mich zu authentifizieren, aber niemand hat geantwortet.«

Abb. 21.7: Windows-Fehlermeldung, wenn der Client den Besitzer des NetBIOS-Namen Domänenname<1b> nicht finden kann. In diesem Beispiel war der Domänenname CHIPSNDIPS



Dies passiert, wenn Windows den Besitzer des NetBIOS-Namens `CHIPSNDIPS<1b>` nicht finden kann oder der Name in eine IP-Adresse aufgelöst werden konnte, Windows aber keine Antwort vom Server erhalten hat.

Warum sollte der Client den Domänen-Server nicht finden können? Erinnern Sie sich, als ich sagte, dass je nach Ihrer Netzwerktopologie einige Konfigurationsschritte ausgelassen werden können? Wenn Ihr Samba-Domain-Controller und Ihr Windows-Client in verschiedenen Subnetzen sind, kann der Client den Namen `CHIPSNDIPS<1b>` nicht über die normalen Broadcast-Methoden auflösen, weil IP-Broadcast-Pakete normalerweise nicht von Routern weitergeleitet werden. Damit der Domänen-Controller gefunden wird, muss der Windows-Client einen WINS-Server kontaktieren, bei dem der

Domänen-Controller seinen Namen registriert hat. Ich empfehle Ihnen, Ihren Domänen-Controller zunächst mit einem Client im gleichen logischen Subnetz zu testen, aber wenn Sie über Router testen müssen, sollten Sie noch einmal zu Kapitel 18, »NetBIOS-Namen ohne Broadcasts auflösen«, zurückblättern, um zu sehen, wie Sie WINS verwalten.

Eine mögliche Erklärung für die aktuellen Probleme ist, dass der Samba-Domain-Controller den Namen `CHIPSNDIPS<1b>` nicht erfolgreich registrieren konnte. Sie können dieses Problem in der `nmbd`-Debug-Logdatei (d.h. `/usr/local/samba/var/log.nmb`) überprüfen. Konnte Samba den Namen erfolgreich registrieren, werden Sie Einträge wie diese sehen:

```
[1999/01/23 10:06:46, 0] nmbd/nmbd_become_dmb.c:become_domain_master_stage2(118)
   *****

   Samba server PICANTE is now a domain master browser for workgroup CHIPSNDIPS on subnet 192.168.1.74

   *****
[1999/01/23 10:07:01, 0] nmbd/nmbd_become_lmb.c:become_local_master_stage2(406)
   *****

   Samba name server PICANTE is now a local master browser for workgroup CHIPSNDIPS on subnet 192.168.1.74

*****
```

Ich habe dies in den Kapiteln 19 und 20 ausführlicher dargestellt, aber der fundamentale Unterschied zwischen einem Domain-Master-Browser (DMB) und einem lokalen Master-Browser (LMB) besteht darin, dass Sie nur einen DMB pro Domäne haben können. Sie sollten dagegen einen LMB für jedes Subnetz haben. Natürlich sollte der DMB auch der LMB für sein logisches Subnetz sein.

Wenn Sie den DMB für eine bestimmte Domäne finden wollen, benutzen Sie das Tool `nmblookup`, um den Namen *`Domänenname`*`<1b>` aufzulösen:

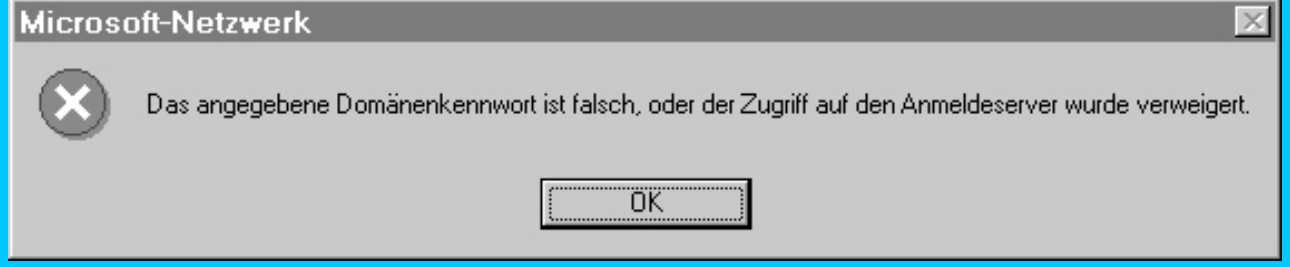
```
# nmblookup CHIPSNDIPS#1b
Sending queries to 192.168.1.255
192.168.1.74 CHIPSNDIPS<1b>
```

Die Ausgabe sollte die IP-Adresse des Samba-Domain-Controllers anzeigen. Ist das nicht der Fall, sollten Sie Ihre `smb.conf`-Datei noch einmal überprüfen und sicherstellen, dass die Parameter `os level`, `domain master`, `local master` und `preferred master` so konfiguriert sind, wie ich es vorher dargestellt habe. Überprüfen Sie außerdem, ob der `nmbd`-Daemon tatsächlich läuft.

Das angegebene Domänenkennwort ist falsch ...

Das zweite weit verbreitete Problem, auf das Sie treffen können, wenn Sie versuchen, sich in eine Domäne einzuloggen, ist, dass der Server gefunden wurde, aber »Nein« sagte (siehe Abbildung 21.8). Es gibt im Wesentlichen zwei Möglichkeiten hierfür, abgesehen davon, dass Sie einfach den Benutzernamen und das Passwort falsch eingegeben haben.

Abb. 21.8: Der Domänen-Controller lehnt die Authentifizierung Ihres Passworts ab



Die erste ist, dass der Server die Sitzungsanfrage abgelehnt hat. Ich habe bereits in Kapitel 4, »Installation und Testen der Konfiguration«, über einige Dinge gesprochen, die dies hervorrufen können. Um Ihr Gedächtnis aufzufrischen: Dies kann passieren, wenn die IP-Adresse des Clients nicht im Wert für den Parameter `hosts allow` in Ihrer `smb.conf` aufgeführt ist oder in der Auflistung für `hosts deny` vorkommt.

Eine andere Möglichkeit ist, dass der Server für die Benutzung von Klartextpasswörtern eingerichtet ist, aber der Client nicht auf Klartext herabstufen will. Sie sollten den zwei Lösungen folgen, die bereits in früheren Kapitel aufgelistet waren:

- Richten Sie den Samba-Server für die Benutzung von verschlüsselten Passwörtern ein, wie in Kapitel 6, »Sicherheitsmodi und Passwörter«, beschrieben.
- Ermöglichen Sie dem Windows-9x-Client die Benutzung von Klartextpasswörtern, wie in Kapitel 14, »Windows 9x und Windows NT«, beschrieben.

Erfolgreich in die Domäne einloggen

Wenn Sie sich erfolgreich in die Domäne einloggen können, sollten Sie sehen, wie das Login-Skript in einem DOS-Fenster ausgeführt wird, wie in Abbildung 21.9 dargestellt. Das Schöne daran ist, dass, mit einigen Ausnahmen, der Windows-Client den Unterschied zwischen einer von Samba kontrollierten Domäne und einer von Windows NT kontrollierten nicht feststellen kann. Daher werden auch Ihre Benutzer höchstwahrscheinlich den Unterschied nicht merken!

Abb. 21.9: Nach einem erfolgreichen Login wird das Windows-NT-Login-Skript ausgeführt

Ein Vorteil der Benutzung eines Login-Skripts gegenüber der Windows-Einstellung für Dauerverbindungen, die Laufwerksverbindungen initiieren, besteht darin, dass Sie das Login-Skript ändern können, ohne überhaupt am Client-Rechner sitzen zu müssen. Sie können Netzwerkverbindungen neu zuweisen und sogar Patches für das Betriebssystem und für Anwendungen im Login-Skript laufen lassen. Toll! Das ist, was mir gefällt.

Sonstiges

Was Sie bis hierhin gesehen haben, ist ziemlich gut. Sie können sicherstellen, dass Benutzer beim Einloggen die korrekten Informationen angeben, Sie können sicher sein, dass jeder Benutzer zumindest einige notwendige Verbindungen zu Netzwerklaufwerken hat, und Sie können andere Dinge während des Logins ausführen. Was können Sie noch tun?

Ich habe es vorher schon gesagt: Das Schöne an dieser Art von Setup ist, dass der Windows-Client meistens gar nicht weiß, dass es sich nicht um einen Windows-NT-Server handelt, der als Domänen-Controller agiert. Die Themen, die ich jetzt darstellen werde, sind nicht Samba-spezifisch, sondern Teil des Windows-9x-Netzwerkmodells. Egal, ob Sie das Modell mögen oder nicht, Sie müssen die Tools benutzen, die Ihnen gegeben werden, um Ihre Aufgabe so gut wie möglich zu erledigen.

Profile

Ein Benutzerprofil entspricht in etwa der Sammlung von `dot(.)`-Dateien, die Unix verwendet, um Logins, Logouts und das Verhalten von Anwendungen zu kontrollieren. Liegt Ihr Hintergrund eher in der Windows-Terminologie, können Sie sich das Profil als eine Sammlung von benutzerspezifischen `*.ini`-Dateien und Programmgruppen vorstellen. Das Fazit ist, dass Profile es einem Benutzer ermöglichen, seine Umgebung anzupassen, ohne sie permanent mit einem einzelnen Rechner zu verbinden.

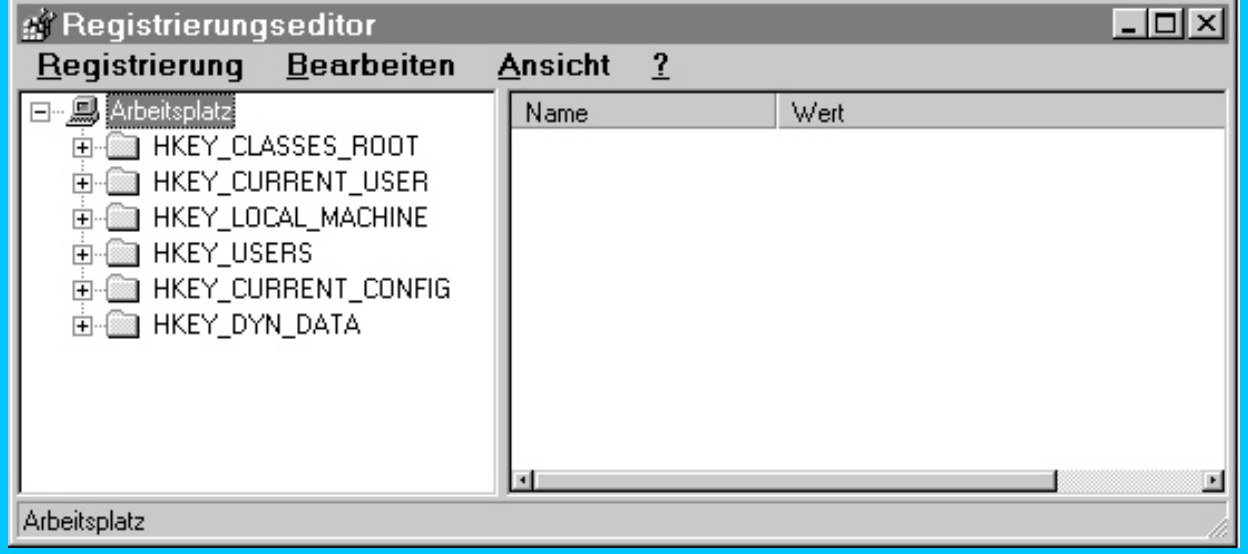
Die Windows-Registry 101

Betrachten Sie diesen Abschnitt als Hintergrundinformation, damit wir alle auf dem gleichen Stand sind. Wenn Sie bereits mit der System-Registry vertraut sind, haben Sie einen Moment Geduld.

Die Windows Registry ist eine Datenbank, die aus zwei Binärdateien besteht, `system.dat` und `user.dat`. Die Datei `system.dat` befindet sich immer im `\windows`-Verzeichnis (oder dem Verzeichnis, in dem Sie Windows installiert haben). Normalerweise ist auch die Datei `user.dat` dort zu finden. Im Fall von wandernden Profilen jedoch wird sie von einem für den Benutzer spezifischen Platz heruntergeladen.

Die zwei Dateien sind in einer baumartigen Struktur mit sechs Hauptwurzeln zusammengefasst. Abbildung 21.10 zeigt die System-Registry, wie sie im Windows-95-Registrierungseditor dargestellt wird. Ich werde mich hier nur mit zwei dieser Wurzeln beschäftigen: HKEY_LOCAL_MACHINE (HKLM) und HKEY_CURRENT_USER (HKCU).

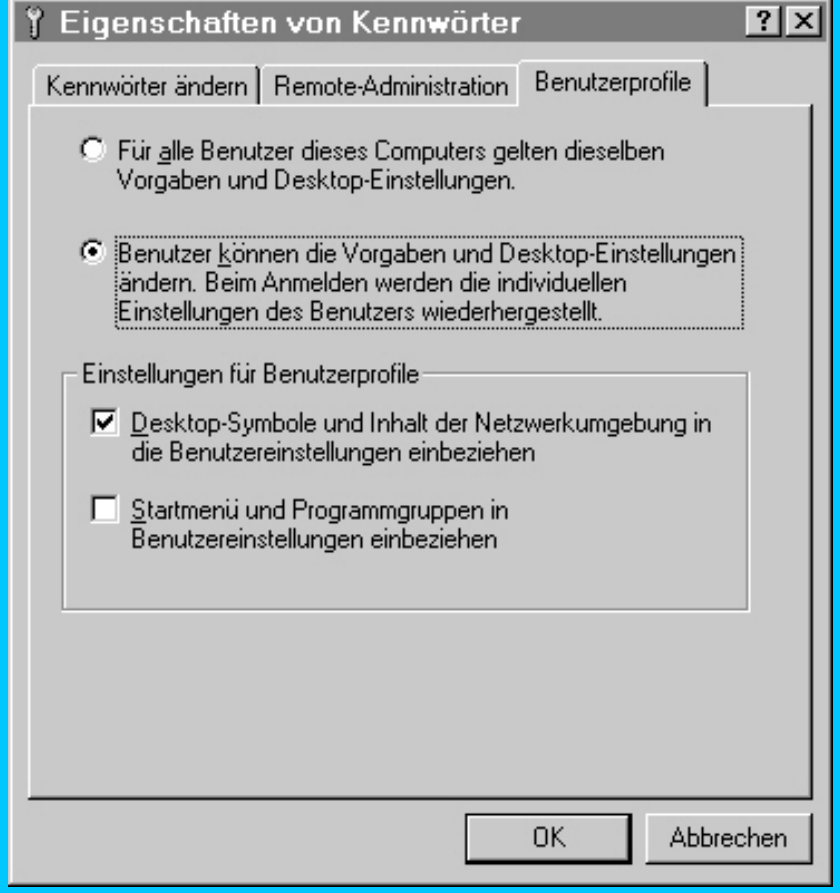
Abb. 21.10: Die System-Registry, wie sie vom Windows-95-Registrierungseditor (regedit.exe) dargestellt wird



Die *HKLM-Struktur* (jede der sechs Wurzeln wird als Struktur bezeichnet) sollte nur Informationen enthalten, die für den Rechner lokal sind. Das ist nicht immer der Fall, da Anwendungen nicht immer die Registry benutzen, die sie sollten.

Die *HKCU-Struktur* enthält Informationen, die für den aktuell eingeloggten Benutzer relevant sind. Beispiele für die Einstellungen, die in der HKCU-Struktur gespeichert sind, sind u.a. Hintergrund- und Bildschirmschoner-Einstellungen, die Liste zuletzt verwendeter Dateien für Anwendungen und andere benutzerspezifische Dateistandorte. Der lokale Rechner kann so eingerichtet werden, dass er die gleichen Benutzerinformationen für jeden verwendet, der sich einloggt, oder dass er Informationen für jeden Benutzer verfolgt. Ersteres ist das Standardverhalten von Windows. In diesem Fall ist die HKCU-Struktur das Gleiche wie die HKEY_USERS- (HKU-)Struktur. Sie können individuelle Profile über die Registerkarte *Benutzerprofile* im Dialogfeld *Passworteigenschaften* aktivieren.

Abb. 21.11: Benutzerprofile unter Windows 95 aktivieren



Was kann noch in einem Benutzerprofil enthalten sein?

Die Möglichkeit, individuelle Einstellungen in der Registry zu verwalten, kann an sich schon sehr hilfreich sein, aber Windows ermöglicht es Ihnen außerdem, das Start-Menü und die Desktop-Icons mit einem Benutzerprofil zu verbinden.

Sie können individuelle Benutzerprofile auf zwei Arten erstellen. Mit der ersten Methode kann nur ein einzelner Rechner das Benutzerprofil verwenden, das lokal gespeichert ist. Über die zweite Methode kann das Profil dem Benutzer im Netzwerk folgen, so dass jeder Rechner, in den sich der Benutzer einloggt, Zugriff auf das Profil hat. Letzteres wird als umherziehendes oder wanderndes Benutzerprofil bezeichnet.

Wenn der Windows-Client mit aktivierten Benutzerprofilen für das Einloggen in eine Domäne konfiguriert wird, versucht das lokale System automatisch, das Profil im Home-Verzeichnis des Benutzers zu speichern. Das Profil wird dann beim Login vom Netzwerk auf die lokale Festplatte zwischengespeichert und beim Logout wieder zurückgegeben. Abbildung 21.12 zeigt ein Beispiel hierfür.

Wenn Sie Samba als Domänen-Controller benutzen, wird der Platz, an dem Windows das wandernde Profil im Netzwerk speichert, über den Parameter `logon path` in der `smb.conf`-Datei festgelegt. Der Standardwert für diesen Parameter ist ein Unterverzeichnis im Home-Verzeichnis des Benutzers mit dem Namen `profile`. Viele Leute in den Samba-Mailing-Listen haben berichtet, dass es besser ist, Profile in einer separaten Freigabe zu speichern. Definieren Sie z.B. zunächst eine Freigabe namens `[profile]`:

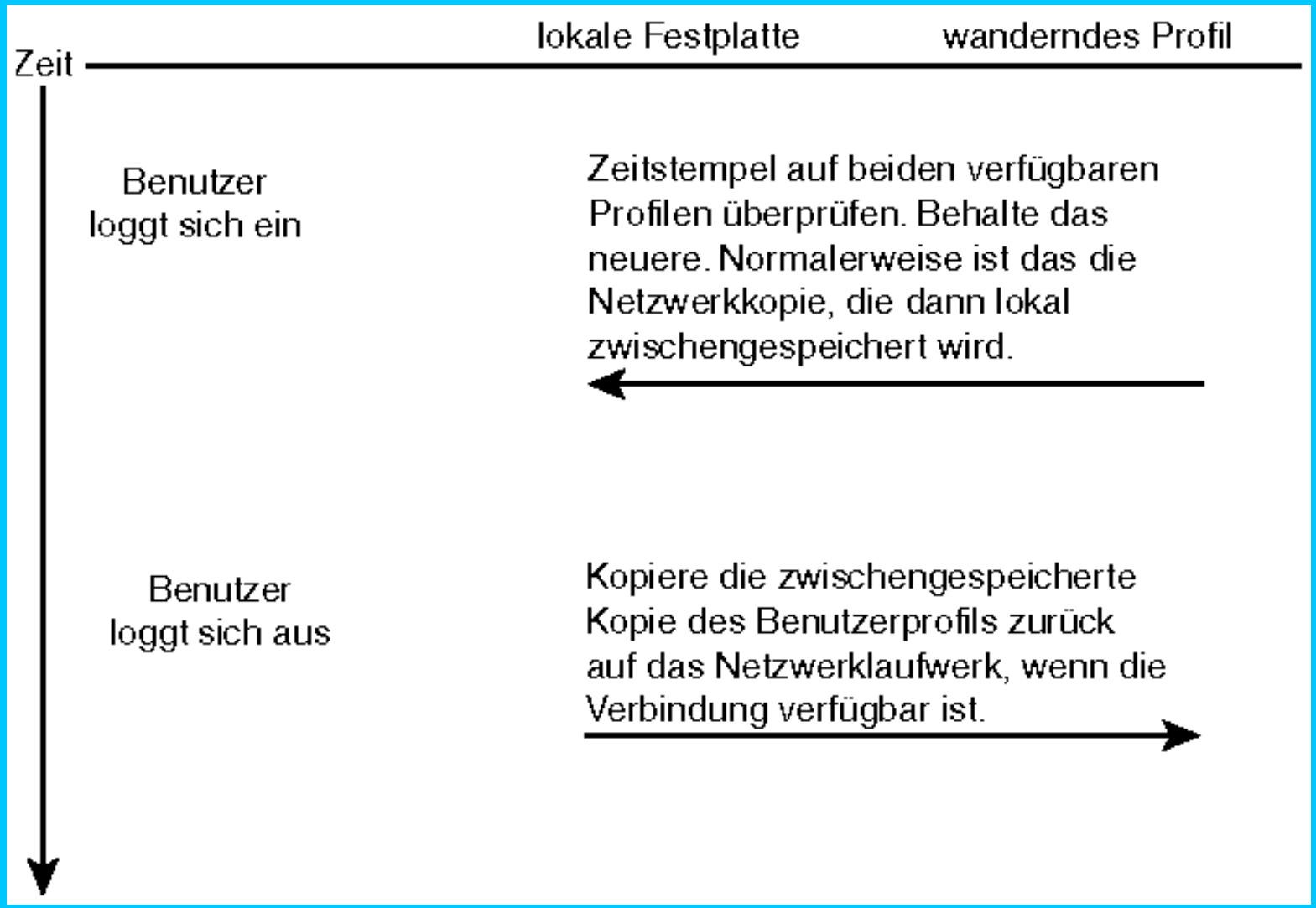
```
[profile]
```

```
comment = Windows user profiles
path = /export/profile
create mode = 0600
directory mode = 0770
browseable = yes
writeable = yes
```

Setzen Sie dann den Parameter `logon path` auf:

```
logon path = \\Servername\profile\%U
```

Abb. 21.12: Speichern von Profilinformationen im Home-Verzeichnis des Benutzers



Weitere Informationen über wandernde Profile finden Sie im Windows 95 Resource Kit, das im Windows-Hilfeformat auf der Windows-95-Installations-CD im Verzeichnis `\admin\reskit` verfügbar ist.

Was Profile für Sie tun können

Ich muss eine Rechtfertigung für meine Erklärung von Benutzerprofilen liefern. Wie können sie meine Aufgabe wirklich vereinfachen, statt sie noch schwieriger zu machen? Hier ist ein Beispiel, das ich verwende, um die Windows-95-Studentenlabors zu managen, die ich an meinem Arbeitsplatz verwalte.

Nehmen wir an, Sie haben einen bestimmten Shortcut, den Sie auf den Desktop jedes Benutzers (oder auch in das Start-Menü) platzieren wollen. Was können Sie tun?

- Sie können Anweisungen für die Erstellung des Shortcuts ausgeben. (Das ist schlecht.)
- Sie können zu jedem Rechner gehen und die Verknüpfung manuell hinzufügen. (Auch schlecht.)
- Sie können das Login-Skript benutzen, um den Shortcut zu kopieren. (Das ist okay.)
- Sie können im Home-Verzeichnis des Benutzers ein `preexec`-Skript einrichten, das die notwendigen Dateien in das wandernde Profil des Benutzers kopiert. (Das ist noch besser.)

Obwohl die dritte Methode funktionieren würde, ziehe ich die letzte vor, da ich lieber Skripte in Perl oder einer Shell-Sprache schreibe als Batch-Dateien zu verwenden. Zusätzlich können Sie `smb.conf`-Variablen an `preexec`- oder `postexec`-Skripte übergeben und die Variablen korrekt interpretieren lassen.

Lassen Sie mich die Bühne vorbereiten. Sie haben den Logon-Pfad wie folgt definiert:

```
logon path = \\%N\profile\%U
```

Sie haben folgende Einstellungen für die `[profile]`-Freigabe konfiguriert:

```
[profile]
   comment = Windows user profiles
   path = /export/profile
   preexec = /usr/local/bin/buildprofile %U
   create mode = 0600
   directory mode = 0700
   browseable = yes
```

Hier ist das `buildprofile`-Skript:

```
#!/bin/sh

user=$1
umask 077
if [ !-f /export/profile/$user/desktop/somelink.lnk ]; then
   cp -p /usr/local/samba/lib/somelink.lnk/export/profile/$user/desktop/
fi
```

In diesem Skript ist `somelink.lnk` der Name der Shortcut-Datei, die Sie an den Desktop jedes Benutzers verteilen wollen. Wenn Windows sich mit der `[profile]`-Freigabe verbindet, um auf das Profil des Benutzers zuzugreifen, führt der `smbd` das `preexec`-Skript aus, das den notwendigen Shortcut kopiert. Mit dieser Art von Konfiguration können Sie alle Arten von komplizierten Tricks durchführen. Seien Sie kreativ!

Policies

System-Policies (Richtlinien) sind eng mit der Windows-95-Registry verbunden. Die Beziehung kann wie folgt erklärt werden: Policies definieren, was erlaubt ist und was abgelehnt wird. Die Windows-Registry enthält die aktuellen Policy-Einstellungen in Form von Registrierungsschlüsseln.

Zu Beginn dieses Kapitels habe ich eine Registry-Einstellung benutzt, um das Passwort-Caching zu deaktivieren. Das ist ein Beispiel für eine Policy. Es liegt in der Verantwortung des Betriebssystems, die Policy-Einstellungen, die in der Registry aufgezeichnet werden, durchzusetzen.

Genau wie Windows einen Registrierungseditor, `regedit.exe`, bietet, stellt es auch einen Policy-Editor mit dem Namen `poledit.exe` zur Verfügung, den Sie in Abbildung 21.13 sehen. Die Darstellung jedes Details der Policy-Dateien und -Vorlagen würde über den Rahmen dieses Buches hinausgehen. Stattdessen werde ich mich darauf konzentrieren, wie Windows-Clients konfiguriert werden, um die Policy-Dateien von einem Server herunterzuladen und was Sie mit System-Policies tun können.

Abb. 21.13: Der Windows-95-System-Policy-Editor

Der System-Policy-Editor kann von der Windows-Installations-CD heruntergeladen werden. Anweisungen für die Installation des Tools finden Sie in einer Datei mit dem Namen `poledit.txt` in `\admin\apptools\poledit`.

Der Policy-Editor ermöglicht es Ihnen, eine Datei zu erstellen, die während des Logins mit der lokalen System-Registry verbunden werden kann. Sie können Windows mitteilen, die Policy-Datei automatisch herunterzuladen:

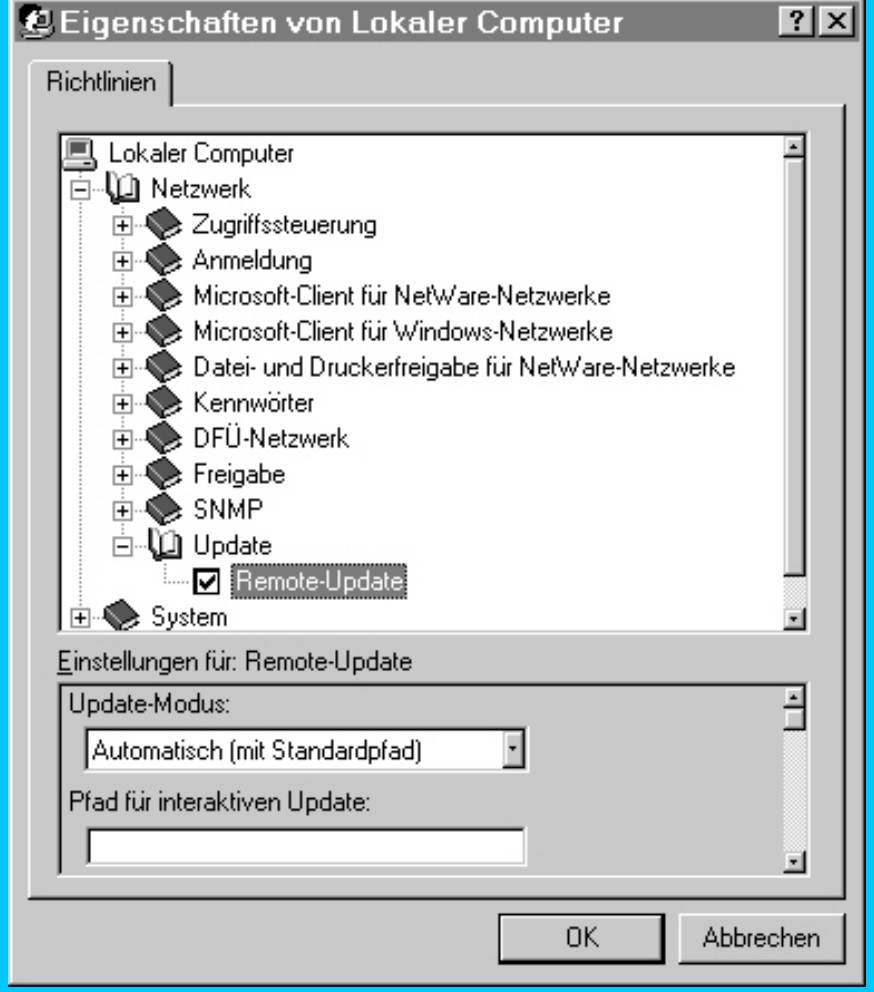
`\\`*`server`*`\netlogon\config.pol`

`server` ist hier der NetBIOS-Name des Domänen-Controllers, der das Login durch Setzen des folgenden Registrierungsschlüssels authentifiziert hat:

```
[HKLM\System\CurrentControlSet\control\Update]
"UpdateMode"=dword:00000001
```

Denken Sie an die übliche Warnung in Hinsicht auf die Benutzung des Registrierungseditors. Alternativ können Sie den Policy-Editor auf dem lokalen Rechner installieren und die Registry öffnen. Dann können Sie, wie in Abbildung 21.14 gezeigt, den Update-Modus einrichten.

Abb. 21.14: Den Policy-Update-Modus über den System-Policy-Editor einrichten

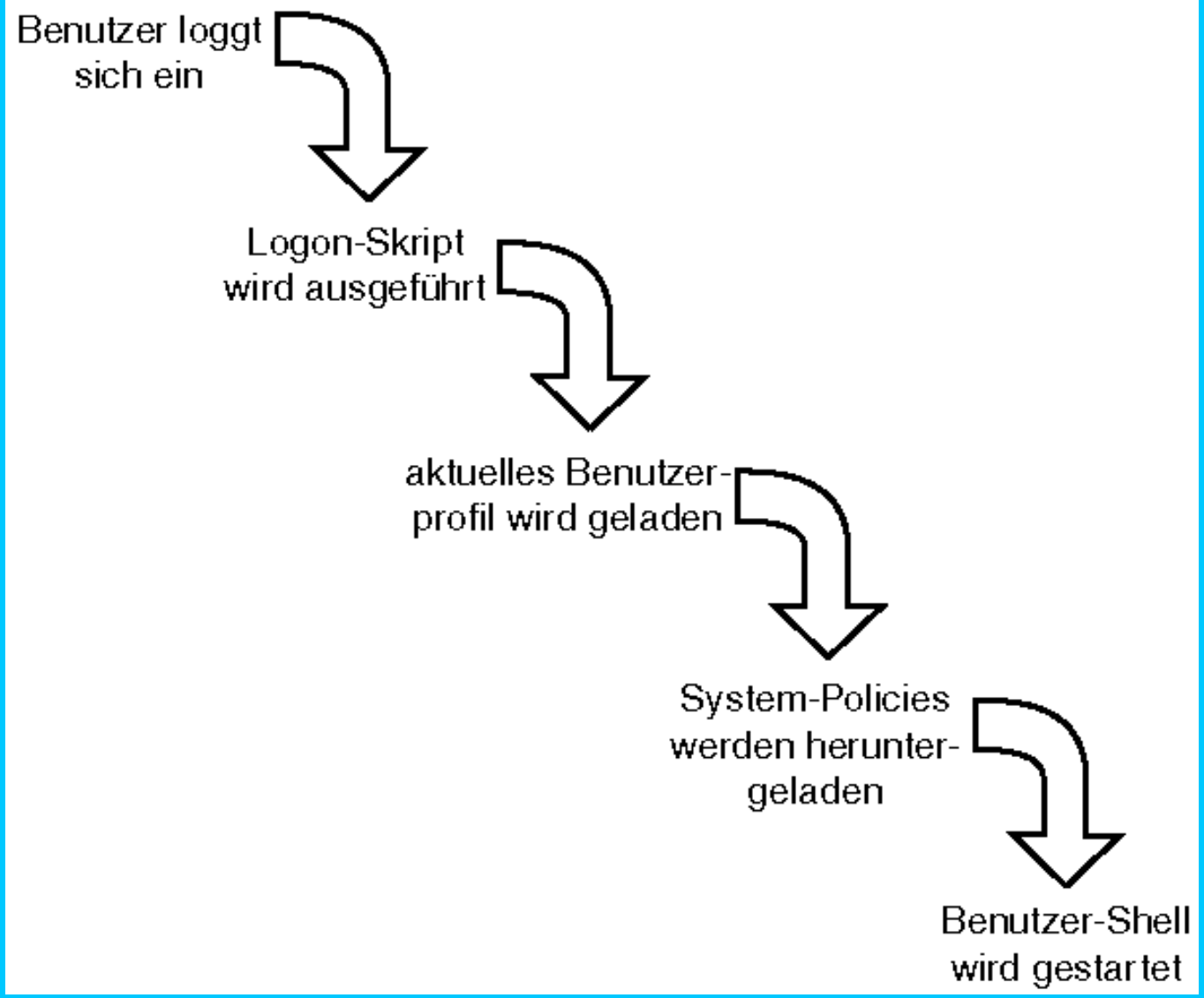


Da Sie sich nun den Policy-Editor angesehen haben, stellt Abbildung 21.15 den Prozess des Einloggens in das Netzwerk dar und zeigt, wann die Policy-Datei, wenn überhaupt, heruntergeladen wird.

Um eine Policy-Datei zu erstellen (z.B. `config.pol`), starten Sie den Policy-Editor und wählen *Neu* aus dem Menü *Datei*. Sie können u.a. folgende Einstellungen vornehmen:

- Zugriff auf Teile der Systemsteuerung verhindern
- Teile aus dem Startmenü und dem Desktop entfernen, wie z.B. die Netzwerkumgebung
- Verlangen, dass der Client das Benutzer-Login authentifiziert, bevor Zugriff auf den Desktop gewährt wird
- Datei- und Druckerfreigaben deaktivieren

Abb. 21.15: Chronologischer Überblick über das Einloggen im Netzwerk



Experimentieren Sie mit den verfügbaren Einstellungen. Weitere Informationen über System-Policies finden Sie in den Windows-95- und Windows-98-Resource-Kits. Verfügbare Dokumentation finden Sie auf der Windows-Installations-CD.

Wenn Sie Ihre Policy-Datei erstellt haben, speichern Sie die Einstellungen in einer Datei mit dem Namen `config.pol` in der `[netlogon]`-Freigabe auf dem DC. Stellen Sie außerdem sicher, dass Sie das Locking für diese Freigabe deaktiviert haben. Sie sollten eventuell der Definition der Freigabe in `smb.conf` Folgendes hinzufügen:

```
locking = no
```

Noch ein letztes Wort zu Policies. Zwar bieten sie einiges an Kontrolle, aber es gibt keinen Mechanismus in Windows 95, der verhindert, dass ein lokaler Benutzer die Policy-Einstellungen in der Registry ändert. Das als kleine Warnung.

Zusammenfassung

Sambas Fähigkeiten zur Domänenkontrolle für Windows-95- und Windows-98-Clients bieten eine Methode für die Authentifizierung von Netzwerk-Logins und nicht nur von Freigabeverbindungen. Die vier Voraussetzungen für einen Samba-DC sind:

- Der Server muss im User-Modus operieren.
- Der DC muss als Domain-Master-Browser für die Domäne konfiguriert sein.
- Domänen-Logins müssen in der `smb.conf`-Datei aktiviert sein (d.h. `domain logons = yes`).
- Eine Freigabe mit dem Namen `[netlogon]` wurde korrekt in der `smb.conf` konfiguriert und ist für die Benutzer zugänglich.

Neben den Standard-Domänen-Logins und Login-Skripten unterstützt Samba auch die Windows-9x-Funktionen zum Herunterladen von System-Policies und das Speichern von Benutzerprofilen auf Netzwerklaufwerken.

Frage & Antwort

F. Ich habe Samba erfolgreich als Domänen-Controller eingerichtet und benutze Login-Skripte. Wenn ein Benutzer sich einloggt, läuft zwar die Batch-Datei korrekt ab, aber das Windows-NT-Logon-Skript-Fenster hängt, bis der Benutzer schließlich irgendwann auf die Schaltfläche Abbrechen klickt. Was mache ich falsch?

. Sie machen nichts falsch. Dieses kleine Ärgernis wird dadurch hervorgerufen, dass die Batch-Datei ihr Ausführungsende signalisiert, indem Sie eine Datei mit dem Namen `LMSCRIPT.$$$` im aktuellen Arbeitsverzeichnis erstellt. Das Windows-NT-Logon-Skript-Fenster würde dann die Dinge bereinigen. Das Problem tritt zutage, wenn die Batch-Datei die Datei `LMSCRIPT.$$$` nicht erstellen kann. Daher weiß das NT-Logon-Skript-Fenster nicht, dass die Batch-Datei ihre Aufgabe beendet hat. Wahrscheinlich sind Sie zu einem Laufwerk und einem Verzeichnis gewechselt, in das der Benutzer nicht schreiben darf. Um dies zu korrigieren, fügen Sie am Ende des Skripts eine Zeile ein, die zurück zur lokalen Festplatte des Benutzers geht (z.B. `c:`).

Anhänge

Anhang A: Experimentelle PDC-Unterstützung

Lassen Sie mich diesen ersten Anhang damit beginnen, Ihnen zu sagen, dass dies ein kleiner Blick in die Zukunft ist. Samba kann minimal als *Primary Domain Controller (PDC)* agieren, indem es Domänen-Logins für Windows-NT-Clients durchführt. Diese Funktion ist jedoch derzeit noch nicht komplett vorhanden und wird offiziell noch nicht unterstützt. Auch wenn Sie sich an Entwicklungscode (auch Prealpha-Test-Code genannt) nur zögerlich herantrauen, seien Sie vorsichtig!

Wenn die Unterstützung nicht offiziell ist, warum erwähne ich sie dann hier überhaupt? Weil sie so toll ist!

Dies ist so wichtig, dass ich es noch einmal sagen möchte. Es ist derzeit keine Samba-Version im Umlauf (dazu gehört auch die Version 2.0), die PDC-Funktionen für Windows-NT-Domänen enthält, die *offiziell* unterstützt werden. Inoffizielle Unterstützung findet sich in Form einer Mailing-Liste (samba-ntdom@samba.org), die sich mit dem Debugging und Testen des Codes beschäftigt.

Die Fähigkeit, einen PDC auf Ihrem Unix-Server laufen zu lassen, eröffnet enorme Möglichkeiten, die sonst nicht existieren würden. Die Verwaltung von Benutzer-Accounts z.B. wird einfacher, wenn die Informationen auf einem Rechner leicht zugänglich sind (oder zumindest die Methoden, dies zu tun, bekannt sind). Ein anderer Vorteil liegt darin, dass Sie die gleiche Hardware für andere Unix-Dienste einsetzen können. Vorausgesetzt Sie haben einen ausreichend großen Server, gibt es keinen Grund, dass Ihr Samba-PDC nicht auch als NIS-Master operieren könnte. Ohne dies müssten Sie einen völlig neuen Rechner mit einer potentiell komplett anderen Architektur kaufen, installieren und verwalten (denken Sie an den Fall von Sparc- gegen Intel-Plattformen). Außerdem, wer möchte schon noch ein weiteres Server-Betriebssystem administrieren müssen? Hier kommt ganz einfach das Problem zum Tragen, genügend Leute zu haben, um all die Dienste korrekt zu verwalten, die durch die Support-Organisation für Ihr Netzwerk geboten werden.

Im restlichen Teil dieses Anhangs werden Sie sich ansehen, welche Windows-NT-PDC-Funktionen in Samba implementiert wurden und wie Sie diese konfigurieren. Wie in Kapitel 21, »Windows-9x-Domänenkontrolle«, werden Sie auch hier einen kurzen Blick auf System-Policies und Benutzerprofile werfen. Abschließend werden Sie sich ansehen, wie diese Unterstützung durch die bevorstehende Freigabe von Windows NT 5.0 betroffen ist, das jetzt offiziell als Windows 2000 bezeichnet wird.

Was bereits implementiert wurde und was nicht

Windows-NT- und Windows-9x-Domänenkontrolle verwenden unterschiedliche Mechanismen. Wie Sie in Kapitel 21 gesehen haben, ist die Windows-9x-Domänenkontrolle stabil und wird vollständig unterstützt. Die Mechanismen für die Implementierung der Windows-NT-Domänenkontrolle wurden nur zum Teil fertiggestellt und die Unterstützung befindet sich noch in der Entwicklungsphase.

Was ist also an einer Windows-NT-Domäne anders? Nun, zunächst einmal ist das NT-Domänen-Kontrollprotokoll derzeit nicht dokumentiert. Es wurde also in Samba implementiert, indem Bits und Pakete angestarrt wurden, die zwischen Domänenmitgliedern und Domänen-Controllern über das Netzwerk übertragen wurden. Keine schlechte Leistung! Ein anderes Problem ist die Menge der Funktionalität, die unterstützt werden muss, nur um sich einfach in die Domäne einzuloggen. Vielleicht erinnern Sie sich noch daran, dass ich in Kapitel 12, »Fallstudie: Einen NT-Datei- und Drucker-Server ersetzen«, sagte, dass Windows NT, wenn Sie einen Teil der Funktionalität unterstützen, erwartet, dass Sie alles unterstützen. Dies wird ganz besonders deutlich, wenn Sie versuchen, Domänenkontrolle und die damit verbundenen Tools wie z.B. den Benutzer-Manager für Domänen oder den Server-Manager für Domänen zu implementieren.

Wenn dies so ein großes, monströses Projekt ist, fragen Sie sich vielleicht: »Wie viel wurde denn bisher erreicht?« und »Wie viel muss noch getan werden?« Da dieser Teil von Samba sich in der aktiven (derzeit sehr aktiven) Entwicklung befindet, ist der beste Ort für Informationen über den aktuellen Status der Unterstützung der Online-NT-Domain-FAQ. Das Inhaltsverzeichnis hierfür kann unter dem Link *Documentation* auf der Samba-Homepage gefunden werden.

Da der Samba-Code ein derart bewegliches Ziel ist und es wirklich keine Meilensteine wie eine verteilte Freigabe gibt, kann es schwierig sein, genau zu sagen, welche Version des Entwicklungscodes ich benutze, außer wenn ich das Datum des letzten Herunterladens angebe. Sie

werden hören, dass die Leute den aktuellen Entwicklungscode (auch als *HEAD-Branch* bezeichnet) mit einer Prealpha-Versionsnummer bezeichnen. Der letzte HEAD-Branch-Code z.B. bezeichnet sich selbst als 2.1.0-prealpha, wenn die `smb.conf`-Variable `%v` angezeigt wird.

Hier ist eine Liste der aktuell oder partiell implementierten Funktionen:

- Die Fähigkeit, als PDC für Windows-NT-3.51- und -4.0-Clients zu agieren.
- Die Fähigkeit, über den Benutzer-Manager für Domänen unter Windows NT 4.0 Informationen über Domänen-Accounts anzusehen, die in der Datei `smbpasswd` gespeichert sind.
- Die Erlaubnis, Samba-Freigaben über den Server-Manager für Domänen unter Windows NT 4.0 anzusehen.
- Das Setzen von Windows-9x-Clients in den User-Modus. Die Clients können jedoch keine Liste von Domänen-Accounts durchsuchen, um Berechtigungen für Datei- und Druckerfreigaben zu spezifizieren.
- Windows-NT-Domänenmitglieder, die ihr Rechner-Account-Passwort regelmäßig ändern können. Ich werde Vertrauens-Accounts von Rechnern später darstellen.
- Die Möglichkeit für Benutzer, ihr Passwort, das in `smbpasswd` aufgezeichnet ist, zu ändern, während sie in einen Windows-NT-Client eingeloggt sind, indem sie die Tastensequenz `[Strg]+[Alt]+[Entf]` benutzen und dann *Passwort ändern* wählen.
- Die Möglichkeit, Windows-NT-Gruppen und -Benutzernamen Ihren Unix-Entsprechungen zuzuordnen. Sie können z.B. spezifizieren, dass alle Mitglieder der Unix-Gruppe `wheel` Mitglied der Windows-NT-Gruppe `Domain Admins` sein sollten.
- Unterstützung für die Einbindung von Domänenbenutzern und -gruppen in die NTFS ACLs.

Dies ist eine relativ lange Liste! Nun, hier sind ein paar Dinge, die bis jetzt noch nicht implementiert sind und daher in der Liste fehlen:

- Die Möglichkeit für einen Samba-PDC, an Vertrauensstellungen mit einem anderen PDC, Samba oder Windows NT teilzunehmen.
- Die Möglichkeit, die Account-Datenbank des Systems mit einem *Backup Domain Controller (BDC)* zu replizieren. Dies bezieht sich auf das Protokoll, das ein Windows-NT-PDC benutzt, um aktualisierte Benutzerinformationen an einen Windows-NT-BDC zu übertragen.
- Unterstützung für wahres Windows-NT-artiges Drucken. Derzeit bringt Samba Windows-NT-Clients dazu, zu einer niedrigeren Version des SMB-Protokolls (z.B. LanMan statt NTLM) herabzustufen. Bitte blättern Sie zurück zu Kapitel 2, »Windows-Netzwerke«, wenn Sie sich die SMB-Protokollebenen noch einmal ansehen wollen.
- Die Möglichkeit, Windows-NT-ACLs auf Samba-Freigaben zu implementieren statt der Standard-Unix-Berechtigungsbits.

Obwohl die Liste der Dinge, die bereits implementiert wurden, länger ist als die der Dinge, die noch nicht implementiert wurden, wird sich die NT-Domänenunterstützung ständig weiterentwickeln. Die offizielle PDC-Funktionalität wird voraussichtlich in Version 2.1 beinhaltet sein.

Wie bekomme ich den Code?

Sie haben also beschlossen, die Dinge auszuprobieren. Okay, der nächste Schritt besteht darin, den aktuellsten HEAD-Branch-Source-Code herunterzuladen. Die beste Methode, den Source-Code für den HEAD-Samba-Branch zu bekommen, besteht in der Benutzung von *CVS*, was für *Concurrent Versions System* steht. Über CVS können die Samba-Entwickler den Source-Code in etwa so »ausleihen«, wie Sie ein Buch aus einer Bibliothek ausleihen würden. Der Unterschied ist, dass ein Entwickler eine Kopie des Codes mitnimmt. Dies ist anders als Revision Control Software, wie z.B. SCCS, bei der nur eine Person auf einmal den Code ändern kann.

Wenn ein Entwickler die Kopie des Source-Baums verändert, müssen diese Änderungen an den Haupt-Branch zurückgegeben werden. CVS fasst die übertragenen Änderungen zusammen, die andere hinzugefügt haben, seit der Code zuletzt ausgeliehen wurde. Dies ist dem sehr ähnlich, was ein `diff`-Skript oder das Unix-Utility `patch` tut.

Samba-Entwicklungscode ist über Anonymous CVS für diejenigen verfügbar, die keinen Schreibzugriff auf den Source-Baum haben, d.h. jeder, der nicht Mitglied des Samba-Teams ist. Um anonymen CVS-Zugriff nutzen zu können, müssen Sie zunächst eine Kopie des CVS-Clients herunterladen und installieren.

Weitere Informationen über CVS finden Sie unter http://www.cyclic.com/. Der Source-Code für den CVS-Client ist verfügbar unter `ftp://download.cyclic.com/pub/`.

Der CVS-Client-Source-Code unterstützt die GNU-`autoconf`-Tests genau wie Samba, also sind Kompilierung und Installation unkompliziert. Ist der Client installiert, können Sie den HEAD-Branch-Source-Code herunterladen, indem Sie sich zunächst in den Samba-CVS-Server einloggen:

```
cvs -d :pserver:cvs@cvs.samba.org:/cvsroot login
```

Sie sollten dies in dem Verzeichnis tun, in das der Source-Baum heruntergeladen werden soll. Wenn Sie nach einem Passwort gefragt werden, geben Sie **cvs** ein.

Der nächste Befehl erstellt ein Unterverzeichnis mit dem Namen `./samba` und holt eine Kopie des HEAD-Branch-Source-Codes:

```
cvs -d :pserver: cvs@cvs.samba.org:/cvsroot co samba
```

Wenn der CVS-Client-Source-Code das Herunterladen des Source-Baums beendet hat, können Sie Samba mit den gleichen Methoden kompilieren, die Sie auch für die Version 2.0.6 verwenden (siehe Kapitel 3, »Wie bekomme ich den aktuellsten Source-Code«). Es tut mir leid, aber aus ganz offensichtlichen Gründen gibt es hierfür keine Binärdistribution. Da es sich um Entwicklungscode handelt, läuft der Kompilierungsprozess eventuell nicht so sauber und fehlerfrei ab wie bei einer verteilten, stabilen Version von Samba. Damit meine ich, dass Sie mehr Fehlermeldungen sehen werden oder auf Probleme bei der Kompilierung auf Ihrer Plattform treffen könnten. Sie sollten Berichte über alle Probleme per E-Mail an samba-bugs@samba.org senden. Außerdem sollten Sie Mitglied der Mailing-Liste samba-ntdom@samba.org werden, um über Updates, Korrekturen und Änderungen auf dem Laufenden zu sein.

Nachdem Sie `configure` und `make` ausgeführt haben, starten Sie `make install`, um das Installationsverzeichnis zu erstellen und die Binaries zu kopieren. Wenn Sie Samba bereits vorher auf Ihrem Server installiert haben, sollten Sie ein Backup des Verzeichnisses `/usr/local/samba` machen, bevor Sie die neue Version installieren. Von diesem Punkt an sollten Sie besonders vorsichtig sein, was alle Dateien im Verzeichnis `/usr/local/samba/private` betrifft.

Konfiguration eines Samba-PDC für eine Windows-NT-Domäne

Wenn Sie etwas Hintergrundwissen in Bezug auf die Verwaltung eines normalen Windows-NT-Domain-Controllers haben, werden Sie feststellen, dass dieser Schritt sehr ähnlich ist. Zunächst sollten Sie eine funktionierende `smb.conf`-Datei konfigurieren. Sie müssen den Server im User-Modus konfigurieren und die Passwortverschlüsselung aktivieren. Folgende Einstellungen sind ein Muss:

```
security = user
encrypt passwords = yes
```

Nehmen wir an, Sie haben nur die folgende Freigabe konfiguriert. Dann haben Sie etwas zum Testen, wenn Sie fortfahren:

```
[homes]
comment = user home directories
path = %H
valid users = %S
create mode = 0600
directory mode = 0700
locking = no
```

Zusätzlich zu diesen Einstellungen wähle ich als NetBIOS-Namen für den Server den Namen `BURRITO`, und die Arbeitsgruppe wird wieder `CHIPSNDIPS` sein:

```
[global]
netbios name = BURRITO
workgroup = CHIPSNDIPS
```

Starten Sie nun die Samba-Prozesse. Wenn Samba zum ersten Mal als Domänen-Controller läuft, sucht es eine Datei mit dem Namen *`DOMÄNENNAME`*`.SID` im gleichen Verzeichnis wie die Datei `smbpasswd`. *`DOMÄNENNAME`* sollte das sein, was als Name Ihrer Arbeitsgruppe in `smb.conf` definiert wurde. Für dieses Beispiel wäre die Datei also `/usr/local/samba/private/CHIPSNDIPS.SID`. Existiert die Datei nicht, generiert Samba eine zufällige Domänen-SID und speichert sie in diese Datei. Sonst liest es die Inhalte der Datei und verwendet den Wert, der darin gespeichert ist. Denken Sie daran, dass Windows NT Rechner über eine SID identifiziert und in diesem Fall eine Domäne.

Stellen Sie sicher, dass Sie die vom `smbd` erstellte SID-Datei nicht ändern. Wenn Sie es tun, können sich alle Rechner, die Mitglieder der Domäne sind, nicht mehr einloggen und müssen der Domäne neu hinzugefügt werden.

An diesem Punkt ist es vielleicht eine gute Idee, einen Benutzer-Account als Test-Account für das Einloggen in die Domäne und die Verbindung zu Freigaben zu erstellen. Verwenden Sie hierfür den Benutzernamen `speedy`. Nachdem Sie den Unix-Account mit den Ihnen zur Verfügung stehenden Methoden (Slackware Linux hat einen Befehl mit dem Namen `adduser`, der gut genug funktioniert) in `/etc/passwd` eingerichtet haben, müssen Sie einen Account für den Benutzer in Sambas `smbpasswd`-Datei einrichten, indem Sie folgendes ablaufen lassen:

```
root# /usr/local/samba/bin/smbpasswd -a speedy
New SMB password: Passwort eingeben
Retype new SMB password: Passwort eingeben
Added user speedy.
Password changed for user speedy
```

Wenn Sie den aktuellen Entwicklungscode benutzen, muss Samba auf dem Server laufen, und das Tool `smbpasswd` muss den primären NetBIOS-Namen, der in der lokalen `smb.conf` spezifiziert ist, auflösen können. Läuft Samba nicht, sehen Sie eventuell eine Fehlermeldung wie die folgende:

```
root# /usr/local/samba/bin/smbpasswd -a speedy
error connecting to 192.168.1.74:139 (Connection refused)
cli_establish_connection: failed to connect to BURRITO<00> (192.168.1.74)
error connecting to 192.168.1.74:139 (Connection refused)
cli_establish_connection: failed to connect to BURRITO<00> (192.168.1.74)
cli_connect_serverlist: Domain password server not available.
get_member_domain_sid: unable to initialise client connection.
Can't setup password database vectors.
```

Wenn `smbpasswd` den NetBIOS-Namen des Servers nicht auflösen kann, wird folgende Fehlermeldung angezeigt:

```
root# /usr/local/samba/bin/smbpasswd -a speedy
cli_connect_serverlist: Can't resolve address for PICANTE
cli_connect_serverlist: Domain password server not available.
get_member_domain_sid: unable to initialise client connection.
Can't setup password database vectors.
```

Wenn Sie aber den zwei Anforderungen entsprechen, sollten Sie mit einer Meldung begrüßt werden, die besagt, dass der Benutzer erfolgreich hinzugefügt wurde. Verwenden Sie nun `smbclient` und den von Ihnen eingerichteten Test-Account, um sicherzustellen, dass bis hierhin alles in Ordnung ist. Versuchen Sie, sich mit dem Home-Verzeichnis des Benutzers zu verbinden:

```
root# /usr/local/samba/bin/smbclient //burrito/speedy -U speedy
Added interface ip=192.168.1.74 bcast=192.168.1.255 nmask=255.255.255.0
Password:
Domain=[CHIPSNDIPS] OS=[Unix] Server=[Samba 2.1.0-prealpha]
smb: \>
```

Dieser Schritt gibt Ihnen etwas Sicherheit, dass die Dinge so laufen, wie sie sollten. Konfigurieren Sie danach Samba für die Durchführung von Domänen-Logins, wie Sie es für einen Windows-9x-Domänen-Controller getan haben. Denken Sie daran, dass Sie zusätzlich zur Aktivierung des Parameters `domain logons` auch die Freigabe `[netlogon]` konfigurieren. Nachfolgend finden Sie die relevanten Einträge, die Sie Ihrer `smb.conf` hinzufügen müssen:

```
[global]
   ; frühere Einträge kommen hierhin
   domain logons = yes

[netlogon]
   comment = NETLOGON service
   path = /export/smb/netlogon
   locking = no
   public = no
   writeable = no
```

Obwohl es nicht notwendig ist, sollten Sie auch ein Login-Skript spezifizieren, das für die Benutzer gestartet wird. Diese Option wird genauso wie in Kapitel 21 gesetzt:

```
logon script = logon.bat
```

Die Batch-Datei zeigt als Beispiel eine Meldung des Tages an. Wie Sie bald sehen werden, gibt es einen alternativen Weg, um das Home-Verzeichnis eines Benutzer zu mounten, daher werde ich das diesmal nicht im Login-Skript tun. Hier ist meine Beispiel-Batch-Datei:

```
@echo off
echo ******************************************************
echo *    Willkommen in einer Samba-gesteuerten NT-Domäne   *
echo *    Samba in 21 Tagen                                 *
echo *    Markt&Technik                                     *
echo ******************************************************
```

```
pause
```

Nachdem Sie nun den Domain Controller konfiguriert haben, müssen Sie der Domäne einen NT-Client hinzufügen.

Die Clients hinzufügen

Die Unterstützung für die Domänenkontrolle für Windows-NT-Clients wurde für Windows NT 3.51 (Service Pack 5) und Windows NT 4.0 (Service Pack 5) getestet. Es ist auch möglich, einen anderen Samba-Server als Domänenmitglied hinzuzufügen. Denken Sie an die Möglichkeiten!

Das Hinzufügen jedes Client-Typs zu einer Windows-NT-Domäne umfasst die gleichen Schritte:

1. Einen Rechner-Account (auch Rechnervertrauens-Account genannt) auf dem Domänen-Controller für den Client einrichten.
2. Den Client dazu bringen, der Domäne beizutreten.
3. Den Client neu starten (außer natürlich, wenn es sich um einen Samba-Server mit `security = domain` handelt).

Ich werde diese Schritte einzeln darstellen. Aber zunächst werde ich die Frage beantworten »Was ist ein Workstation-Vertrauens-Account?«.

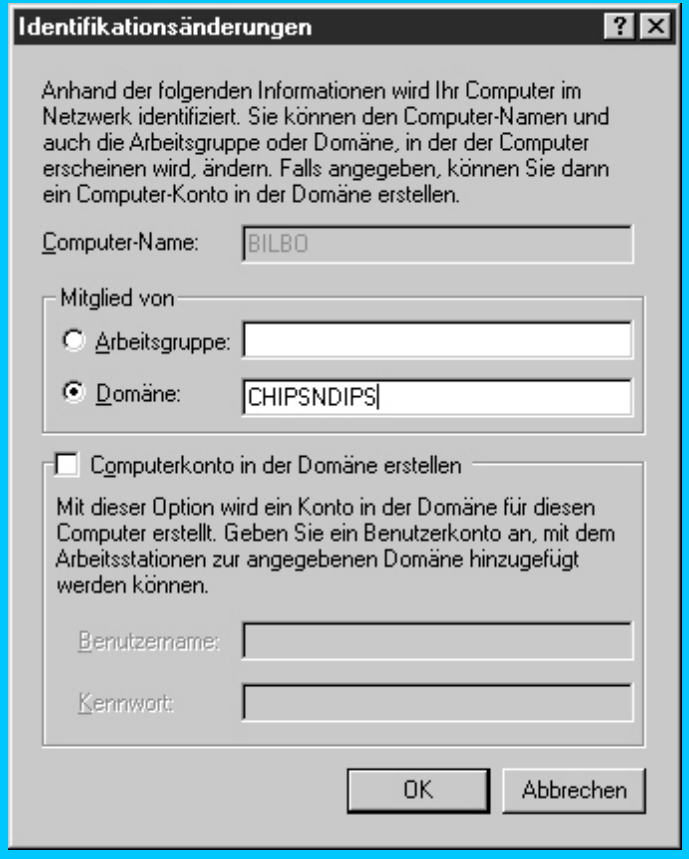
Rechner-Vertrauens-Accounts

Ein *Vertrauens-Account* ist tatsächlich ein Benutzer-Account, der vom Domänenmitgliedsrechner benutzt wird. Generell hat Unix keinen Account für Workstations. Eine Ausnahme hierzu wäre irgendeine Art von sicheren Verteilungssystemen wie z.B. Sun Microsystems NIS+.

Die Idee dahinter ist, dass sich der Rechner in die Domäne einloggen muss, bevor der DC Benutzer-Logins von dem Rechner erlaubt. Aus diesem Grund werden Windows-9x-Clients nicht als wahre Mitglieder der Domäne angesehen, auch wenn sie sich in eine Domäne einloggen. Damit meine ich, dass der Windows-9x-Rechner keinen Vertrauens-Account hat und der Domänen-Controller ihm nicht wirklich vertrauen kann, dass er richtige Informationen präsentiert. Es gibt keine Methode sicherzustellen, dass jemand wirklich derjenige ist, der er vorgibt zu sein, wenn die Informationen von einem Windows-9x-Client übertragen werden.

Samba unterstützt derzeit die Windows-NT-Methode nicht, die beim Versuch, der Domäne beizutreten, einen Vertrauens-Account auf dem PDC einrichtet (siehe Abbildung A.1). Daher müssen Sie den Account auf dem Samba-PDC manuell einrichten.

Abb. A.1: Das Dialogfeld unter Windows NT 4.0 für die Spezifizierung eines administrativen Accounts, der benutzt wird, um beim Eintrittsversuch in die Domäne den Vertrauens-Account einzurichten



Wie ist der Benutzername für einen Vertrauens-Account? Das ist eine gute Frage. Es ist der NetBIOS-Name des Rechners, dem am Ende ein $-Zeichen angehängt wird. Ist der Name Ihres Windows-NT-Clients z.B. `BILBO`, würde für den Vertrauens-Account der Name `BILBO$` benutzt werden. Das Startpasswort für den Account ist auf den NetBIOS-Namen des Rechners eingerichtet, diesmal ohne Zusatz und klein geschrieben. Für mein vorher erwähntes Beispiel wäre das Startpasswort für den Account `BILBO$` also `bilbo`. Richtig sicher, oder?

Da jeder Vertrauens-Account einfach ein Benutzer-Account ist, der von einem Rechner verwendet wird, muss jedem auch eine RID zugeordnet werden (siehe Kapitel 12, wenn Sie noch einmal etwas über RIDs nachlesen wollen). Alle Benutzer-Accounts, darunter auch Vertrauens-Accounts und Gruppen existieren im gleichen Zahlenbereich. Daher brauchen Sie einen Mechanismus um sicherzustellen, dass jeder Vertrauens-Account eine eindeutige RID hat. Wie können Sie dies für Benutzer tun? Im Falle von Benutzern und Gruppen generieren Sie die RID über eine mathematische Funktion aus der entsprechenden UID oder GID.

Warum kann man nicht die gleiche Funktion für die Generierung einer RID für einen Vertrauens-Account benutzen? Vielleicht sehen Sie das Problem bereits. Unter normalen Bedingungen besitzt eine Unix-Workstation keine UID. Dies können Sie umgehen, indem Sie einen Benutzer-Account in der `/etc/passwd` mit dem Namen des Vertrauens-Accounts einrichten (in diesem Fall `BILBO$`). Der `/etc/passwd`-Eintrag ist einfach ein Platzhalter für eine UID und wird nicht für die Authentifizierung verwendet. Hier ist der Eintrag, den ich in die lokale Passwortdatei auf meinem Linux-Server eingefügt habe:

```
bilbo$:*:10000:1000:WinNT trust account:/dev/null:/bin/false
```

Sie sehen, dass ich das Passwortfeld deaktiviert habe, das Home-Verzeichnis eingerichtet habe, auf `/dev/null` zu verweisen und eine ungültige Shell spezifiziert habe. Denken Sie daran, dass Sie nur die richtige UID für den Rechner brauchen.

Jetzt können Sie `/usr/local/samba/bin/smbpasswd` verwenden, um den Vertrauens-Account für den Client einzurichten. Die Option `-a` zeigt, dass Sie einen Account hinzufügen, und die Option `-m` teilt dem Tool `smbpasswd` mit, dass der Account ein Vertrauens-Account für einen Rechner ist:

```
root# /usr/local/samba/bin/smbpasswd -a -m bilbo
Added user bilbo$
Password changed for user bilbo$
```

Dies richtet das Passwort in der Datei `private/smbpasswd` auf den Standardwert für Rechner-Accounts ein, dem Rechnernamen in Kleinbuchstaben.

Einige Unix-Versionen unterstützen keine Benutzernamen, die länger als acht Zeichen sind. Da ein NetBIOS-Name bis zu 15 Zeichen lang sein kann, werden Sie sich eventuell mit diesen Namensunterschieden beschäftigen müssen. Sie sollten die Details für Ihr bestimmtes System überprüfen. Obwohl Slackwares Utility `adduser` keine Benutzernamen mit mehr als acht Zeichen erlaubt, können Sie einen solchen Namen manuell in der `/etc/passwd` einfügen.

Wenn ein Client der Domäne beitritt, ändert er sein Passwort für den Vertrauens-Account auf einen zufälligen Wert und sendet dieses neue Passwort in verschlüsselter Form an den PDC. Das neue Passwort wird jedoch nicht benutzt, bis sich der Rechner nach einem Neustart in die Domäne einloggt.

Der Domäne beitreten (Windows-NT-4.0-Client)

Ist der Client-Rechner ein anderer Samba-Server, folgen Sie den Anweisungen für den Beitritt zu einer Domäne, die Sie in Kapitel 12 finden. Dieser Abschnitt stellt dar, wie Sie einen Windows-NT-Rechner dazu bekommen, einer Samba-Domäne beizutreten. Das Beispiel benutzt die Version 4.0 des Betriebssystems, aber der Prozess ist für Windows NT 3.51 sehr ähnlich. Tatsächlich entsprechen die Schritte fast genau denen, die für den Beitritt zu einer Domäne, die von einem Windows-NT-PDC kontrolliert ist, nötig sind.

Zunächst müssen Sie auf dem Windows-NT-Client das Netzwerkkontrollfeld öffnen (doppelklicken Sie dazu auf das Netzwerk-Symbol in der Systemsteuerung) und auf der Registerkarte *Identifikation* die Schaltfläche *Ändern* wählen. Das Fenster in Abbildung A.1 wird geöffnet.

Nachdem Sie den gewünschten Domänennamen eingegeben und auf *OK* geklickt haben, sollten Sie mit der Willkommensmeldung begrüßt werden, die Sie in Abbildung A.2 sehen. Klicken Sie einfach auf *OK* in diesem Fenster und schließen Sie dann die Netzwerkkontrolle. Windows NT fordert Sie zu einem Neustart auf. Nachdem der Rechner neu gestartet ist, können Sie sich über den Test-Account, `speedy`, den Sie vorher eingerichtet haben, in die Domäne einloggen.

Abb. A.2: Willkommen in der Domäne CHIPSNDIPS

Wenn Sie eine Fehlermeldung erhalten, die anzeigt, dass der Domänen-Controller nicht gefunden werden konnte, stellen Sie sicher, dass Samba läuft, dass der Client-Rechner den Namen `CHIPSNDIPS<1b>` auflösen und sich tatsächlich mit dem Server verbinden kann. Überprüfen Sie z.B., ob der Rechner nicht in einer `hosts-deny`-Einstellung auf dem Server aufgelistet ist.

Besagt die gezeigte Fehlermeldung, dass der Client sich nicht mit dem Domänen-Controller verbinden konnte und Sie den Rechner-Account des Clients vom Administrator überprüfen lassen sollten, stellen Sie sicher, dass Sie den Rechner-Account korrekt hinzugefügt haben. Schließlich sind Sie der Administrator.

Den NT-Client neu starten

Nachdem Sie den NT-Client neu gestartet haben, sollten Sie sich über den gültigen Domänen-Account, `speedy`, einloggen können. Stellen Sie sicher, dass Sie im Popup-Menü *Domäne* die Domäne `CHIPSNDIPS` auswählen statt des lokalen Rechners. Nach dem Einloggen drücken Sie wieder `[Strg]+[Alt]+[Entf]`, um das Windows-NT-Dialogfeld *Sicherheit* zu öffnen. Sie sollten in dem Fenster sehen können, dass Sie als `CHISNDIPS\speedy` eingeloggt sind.

Wenn Sie bei Ihrem Versuch, sich einzuloggen, eine Fehlermeldung erhalten, die besagt: »Die Domäne CHIPSNDIPS ist nicht verfügbar«, stellen Sie sicher, dass Samba läuft und dass der Client-Rechner den Namen `CHIPSNDIPS<1b>` über die in Kapitel 11, »Troubleshooting«, dargestellten Methoden auflösen kann. Je nach System müssen Sie vielleicht einen WINS-Server verwenden. Einen Überblick über WINS und warum es benötigt wird, finden Sie in Kapitel 18, »NetBIOS-Namen ohne Broadcasts auflösen«.

Zusätzliche Parameter

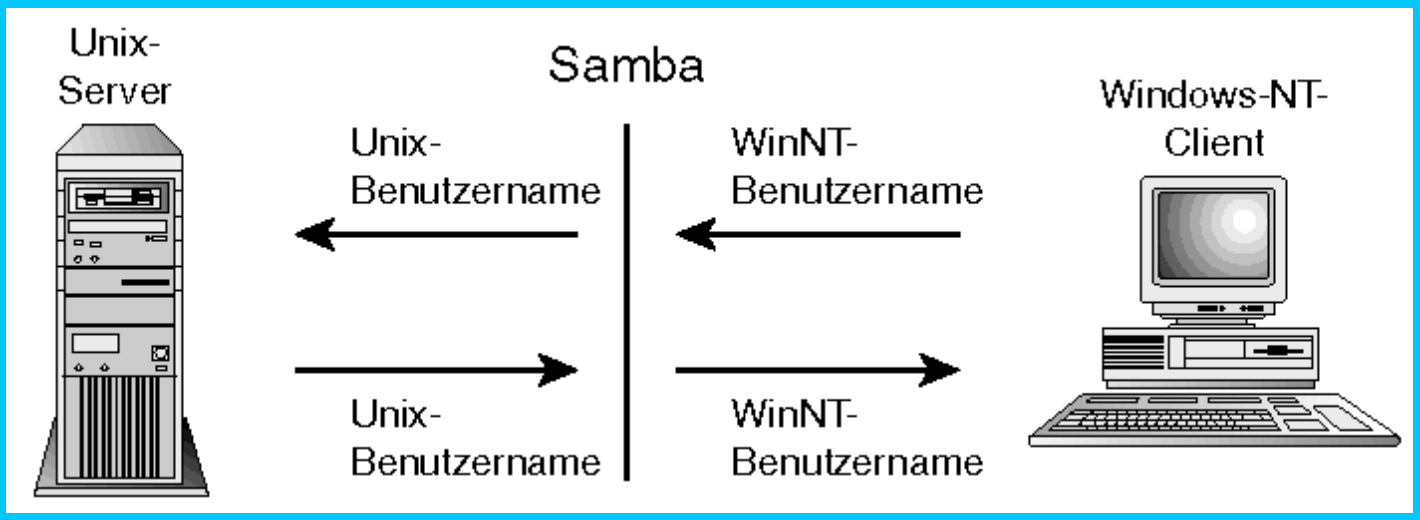
Natürlich umfasst eine Windows-NT-Domäne mehr als einfach nur das Einloggen. In Windows NT sind viele Sicherheitsmechanismen integriert, von denen Sie profitieren wollen. Ein NTFS-Dateisystem unterstützt z.B. Zugriffskontrolllisten für die Einrichtung von Berechtigungen auf Dateien, Verzeichnissen und Druckern. Sie brauchen einen Mechanismus, mit dem Sie Domänenbenutzer und Domänengruppen in die ACLs auf der lokalen Festplatte einfügen können. Eine andere Funktion, die Sie sich ansehen werden, ist die

Unterstützung für wandernde Benutzerprofile und System-Policies, wie Sie sie auch unter Windows 9x verwenden können.

Gruppen und Benutzer

Abbildung A.3 beschreibt, wie Samba einem Windows-NT-Namen einen Unix-Benutzernamen zuordnet und umgekehrt. Der generelle Gedanke dahinter ist, dass man einen Benutzername zuerst während der Sitzungsanfrage erhält. Der Benutzername wird dann durch eine Zuordnungsfunktion gefiltert und der daraus resultierende Benutzername anhand der Passwortdatenbank des Servers authentifiziert. Der Windows-NT-Client wird den Unix-Benutzernamen nicht erfahren, und der Unix-Rechner wird niemals den Windows-NT-Benutzernamen kennen. Samba isoliert die zwei Betriebssysteme voneinander.

Abb. A.3: Zuordnung von Windows-NT-Benutzern und -Gruppen zu Unix-Benutzern und -Gruppen



domain group map

Drei Parameter unterstützen diese Funktion. Der erste ist `domain group map`. Sein Wert definiert den Standort einer Datei, die Zuordnungen zwischen Unix- und NT-Gruppen enthält. Die Gruppen werden als Domänen- oder globale Gruppen behandelt.

```
domain group map = /usr/local/samba/lib/domain_group.map
```

Die Syntax der Map-Datei ist sehr einfach. Jeder Eintrag hat folgendes Format:

```
UnixGruppenname = NTGruppenname
```

`UnixGruppenname` ist der Name der Gruppe, wie er in der `/etc/group` (oder der Netzwerkentsprechung) definiert ist. *`NTGruppenname`* ist der Name, der auf den Windows-NT-Clients angezeigt werden soll. Der folgende Eintrag z.B. definiert eine NT-Gruppe mit dem Namen `Accounting` und ordnet ihr die Unix-Gruppe mit dem Namen `acct` zu:

```
acct = Accounting
```

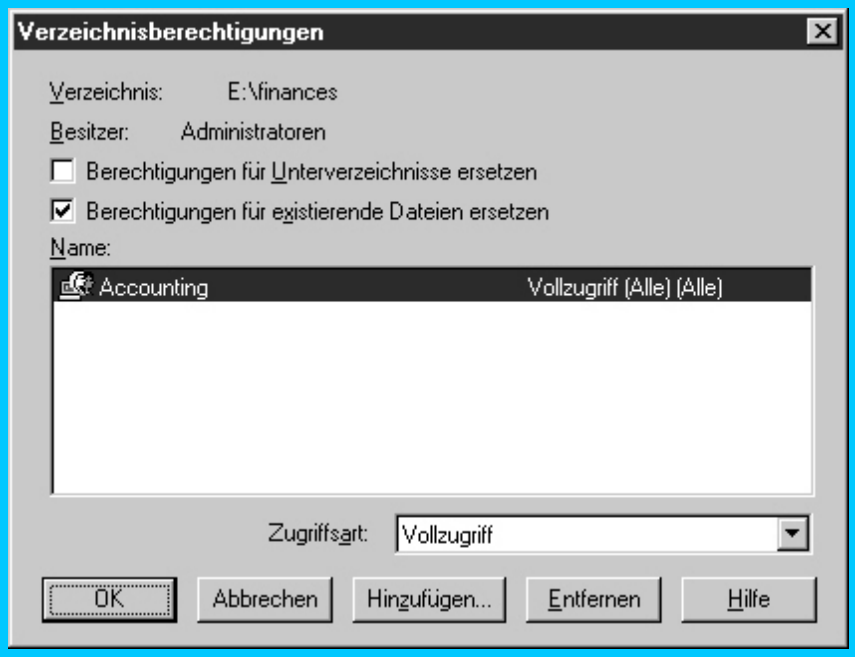
Hier ist die Ausgabe des Befehls `net group /domain`:

```
D:\>net group /domain
Die Anforderung wird auf dem primären Domänen-Controller für Domäne CHIPSNDIPS verarbeitet.
Gruppenkonten für \\BURRITO

-------------------------------------------------------------
*Accounting          *adm              *bin
*daemon              *disk             *Domain Admins
*Domain Guests       *Domain Users     *dptheads
*floppy              *kmen             *lp
*mail                *man              *mem
*news                *nogroup          *root
*sys                 *tty              *users
*uucp                *webdev           *wheel
*wrks
Der Befehl wurde erfolgreich ausgeführt.
```

Die Gruppe `Accounting` ist mit den anderen Domänengruppen aufgelistet. Beachten Sie, dass alle Unix-Gruppen, die nicht in den Map-Dateien der Domänengruppen enthalten sind, ebenfalls aufgelistet sind. Die Unix-Gruppe `acct` ist nicht aufgelistet, weil sie in der Map-Datei enthalten ist.

Zunächst erstelle ich ein Verzeichnis mit dem Namen `acct` auf einer NTFS-Partition, setze als Eigentümer die Domänengruppe `Accounting` und gewähre nur dieser Gruppe Zugriff. Das Windows-Explorer-Fenster zeigt die Berechtigungen und den Eigentümer, die in Abbildung A.4 dargestellt sind.

Abb. A.4: Anzeige der Berechtigungen für das Verzeichnis acct



Danach füge ich der Gruppe `acct` in der `/etc/group` den Benutzer `speedy` hinzu. Der daraus resultierende Eintrag sieht wie folgt aus:

```
acct:*:200:daphnie,scooby,velma,speedy
```

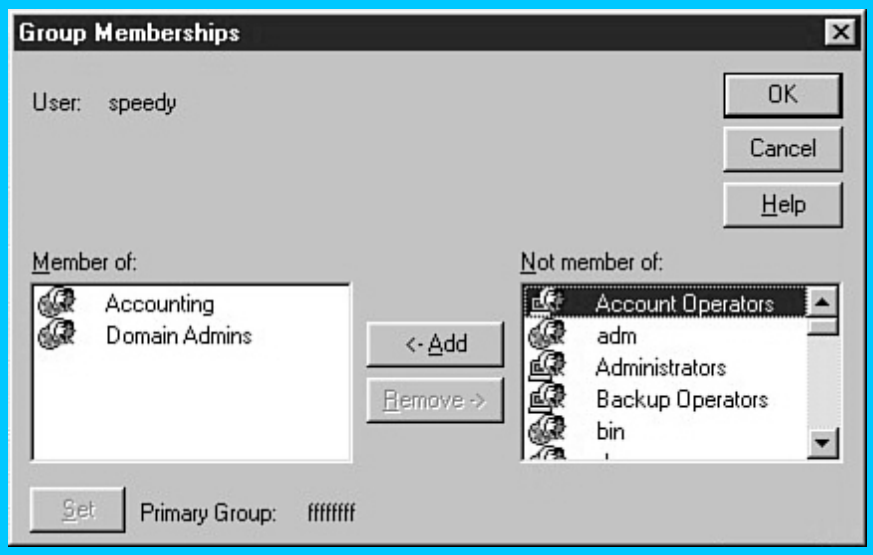
Wenn ich mich in den NT-Client einlogge und versuche, in `E:\finances` eine Datei zu erstellen, habe ich aufgrund der Mitgliedschaft in der Gruppe `acct` volle Kontrolle über das Verzeichnis.

Einige globale Gruppen werden immer dargestellt: die allgemein bekannten Domänengruppen wie z.B. `Domain Admins`, `Domain Users` und `Domain Guests`. Um einen Account als einen Domänenadministrator zu konfigurieren, müssen Sie zunächst eine entsprechende Unix-Gruppe wählen, die für die Zuordnung verwendet wird. Ich benutze eine Gruppe namens `ntadmin` und füge dem Eintrag in der `/etc/group` den Account `speedy` hinzu:

```
ntadmin:*:16:speedy
```

Wenn ich mir jetzt über den Benutzer-Manager für Domänen die Domänenbenutzer auf dem Samba-PDC anschaue, kann ich sehen, dass der Benutzer `speedy` Mitglied sowohl der Gruppe `Accounting` als auch der Gruppe `Domain Admins` ist (siehe Abbildung A.5).

Abb. A.5: Anzeige der Gruppenmitgliedschaft für den Benutzer speedy im Benutzer-Manager für Domänen



domain user map

Der Parameter `domain user map` funktioniert im Wesentlichen genau wie der Parameter `domain group map`: Er akzeptiert als Wert den Pfad zu einer Datei. Die Datei enthält Zuordnungen von Unix-Benutzernamen zu Windows-NT-Benutzernamen.

```
domain user map = /usr/local/samba/lib/domain_user.map
```

Das Format einer Zuordnung für Domänennamen ist:

`UnixBenutzername = [\\Domänenname\\]NTBenutzername`

`UnixBenutzername` ist ein existierender Account auf dem Unix-Server, *`NTBenutzername`* ist der Name, der zugeordnet wird. *`Domänenname`* ist hier optional. Wird keiner angegeben, nimmt Samba an, dass die Domäne dem Wert für den Parameter `workgroup` entspricht. Der Sinn dahinter ist, Samba zu ermöglichen, NT-Benutzer von anderen vertrauenswürdigen Domänen zuzuordnen. Da Vertrauensstellungen bisher noch nicht in Samba implementiert sind, ist dies für den Zweck dieses Beispiels unnötig.

Der Parameter `domain user map` ist nicht das Gleiche wie der Parameter `username map`. Verwechseln Sie die beiden nicht! Letzterer bezieht sich ganz und gar nicht auf Windows-NT-Domänen.

Nehmen wir an, Sie wollten einen Domänen-Account mit dem Namen `Administrator` einrichten, der dem Unix-Account `root` zugeordnet wird. Folgende Schritte müssten Sie dafür durchführen:

1. Richten Sie den Wert für den Parameter `domain user map` in `smb.conf` wie folgt ein:

   ```
   domain user map = /usr/local/samba/lib/domain_user.map
   ```

2. Fügen Sie diesen Eintrag in `/usr/local/samba/lib/domain_user.map` ein:

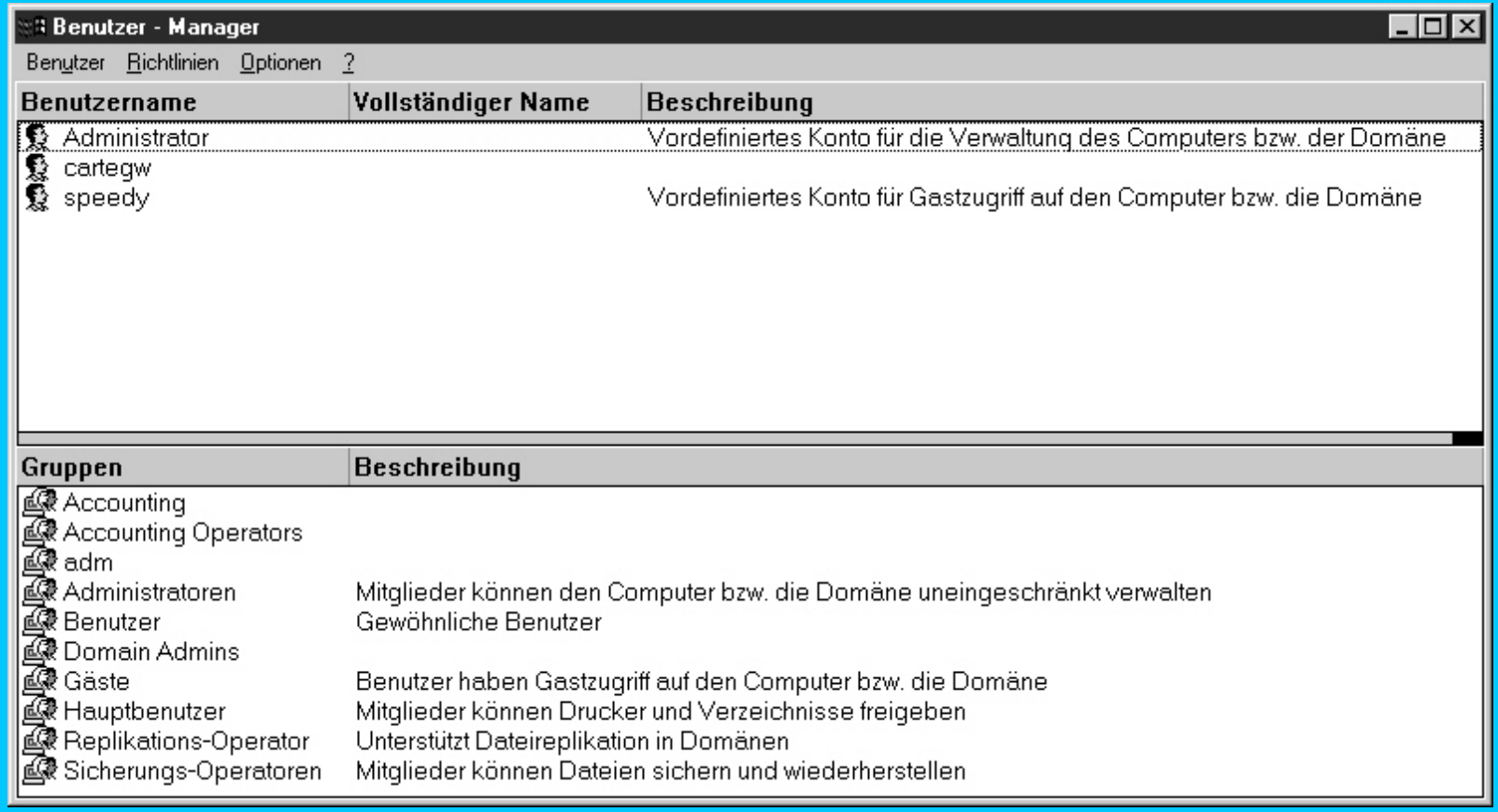
   ```
   root = Administrator
   ```

3. Fügen Sie, wenn notwendig, einen Eintrag für `root` in die `smbpasswd`-Datei ein, indem Sie Folgendes ausführen:

   ```
   root# /usr/local/samba/bin/smbpasswd -a root
   New SMB password: Passwort eingeben
   Retype new SMB password: Passwort eingeben
   Added user root.
   Password changed for user root
   ```

Jetzt können Sie sich von einem Windows-NT-Client mit dem Accountnamen `Administrator` und dem Passwort, das Sie in der `smbpasswd`-Datei für `root` eingerichtet haben, in die Domäne einloggen. Der Account `Administrator` ist auch im Benutzer-Manager für Domänen zu sehen (siehe Abbildung A.6).

Abb. A.6: Anzeige der verfügbaren Domänen-Benutzer-Accounts, darunter Administrator, für die Domäne CHIPSNDIPS im Benutzer-Manager für Domänen



local group map

Der dritte Zuordnungsparameter ist `local group map`, über den Sie lokale Gruppen auf dem Samba-PDC definieren können. In unserem Beispiel spielt er keine Rolle, also werde ich ihn nachfolgend kurz beschreiben.

Zunächst müssen Sie den Standort der Datei angeben, die die Zuordnungen von lokalen NT-Gruppennamen zu Unix-Gruppennamen enthält:

```
local group map = /usr/local/samba/lib/local_group.map
```

Das Format der lokalen Map-Datei ist:

`UnixGruppenname` `= [BUILTIN\]`*`NTGruppenname`*

Wieder ist *`UnixGruppenname`* die in der `/etc/group` definierte Gruppe, der *`NTGruppenname`* zugeordnet wird. Der String *`BUILTIN`*`\` sollte allen allgemein bekannten lokalen Gruppennamen vorangestellt werden, z.B. `Administrators` und `Users`.

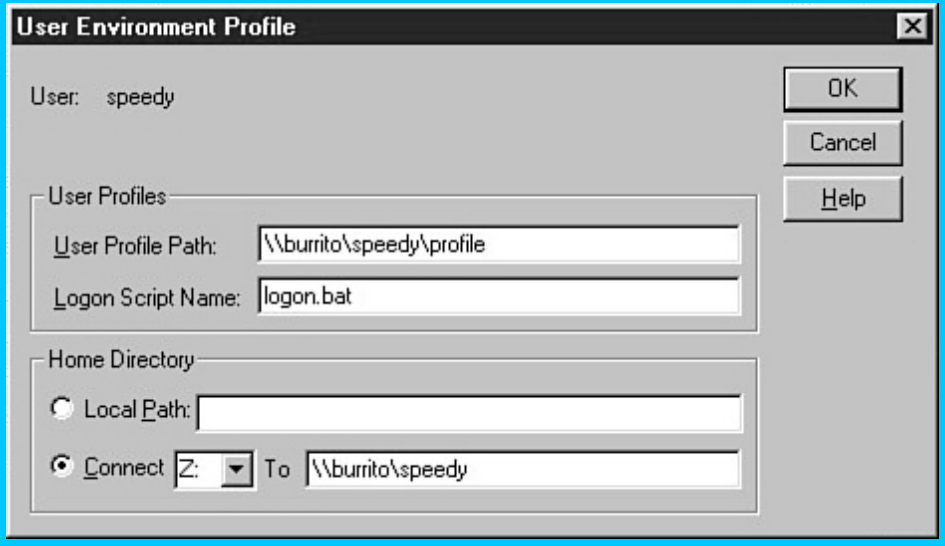
Um z.B. der lokalen Gruppe `Administrators` auf dem Samba-PDC die Unix-Gruppe `wheel` zuzuordnen, müssen Sie einen Eintrag wie den folgenden in die lokale `group-map`-Datei einfügen:

```
wheel=BUILTIN\Administrators
```

logon home, logon drive und logon path

Teil der Informationen eines Benutzer-Accounts in einer Windows-NT-Domäne sind der Standort und der Mount-Point für das Home-Verzeichnis des Benutzers sowie der Standort des wandernden Benutzerprofils (siehe Abbildung A.7). Das Login-Skript des Benutzers ist ebenfalls unter diesen Informationen zu finden. Sie können einige der Felder, die in dem Fenster gezeigt werden, mit `smb.conf`-Parametern verbinden.

Abb. A.7: Benutzer-Account-Informationen, wie sie im Benutzer-Manager für Domänen angezeigt werden



Schauen Sie sich zunächst die Einstellungen an, die sich auf das Home-Verzeichnis des Benutzers beziehen. Der Bereich *Basis-Verzeichnis* im Fenster spezifiziert einen Laufwerksbuchstaben, mit dem der Netzwerk-Homepfad verbunden werden sollte. Dies sind die Parameter `logon drive` bzw. `logon home`:

```
logon drive = Z:
logon home = \\%N\%U
```

Diese Einstellungen, zusammen mit den in Abbildung A.7 gezeigten, sind die Standardwerte für diese Parameter.

Der Parameter `logon home` akzeptiert als Wert einen Pfad, der als Home-Verzeichnis des Benutzers gemountet wird. Mit Samba ist dies einfach über die `[homes]`-Freigabe zu erledigen. Wenn Sie wollen, können Sie auch Folgendes spezifizieren:

```
logon home = \\Server\Users\Benutzername
```

Dies setzt voraus, dass Sie eine Freigabe mit dem Namen `[users]` definiert haben und dass jeder Benutzer ein Verzeichnis hat, dessen Name *`Benutzername`* entspricht. *`Server`* ist der Name des eventuell entfernten Servers, der die *`[users]`*-Freigabe zur Verfügung stellt.

Der Parameter `logon drive` akzeptiert als Wert einen Laufwerksbuchstaben, an dem der Windows-NT-Client das Home-Verzeichnis des Benutzers mountet. Der Standard ist die Benutzung des Laufwerks `Z:`. Sie können hier einen beliebigen Buchstaben angeben. Sie sollten aber einen Laufwerksbuchstaben angeben, von dem Sie relativ sicher sind, dass er auf dem Client verfügbar ist. Obwohl Sie z.B. auch `C:` angeben könnten, wäre dies wahrscheinlich keine gute Idee.

Der Parameter `logon path` schließlich funktioniert ebenso wie für Windows-9x-Clients: Er definiert den Standort im Netzwerk, an dem das wandernde Benutzerprofil gespeichert ist. Standardmäßig werden die Profilinformationen im Home-Verzeichnis des Benutzers gespeichert. Aus den in Kapitel 21 bereits dargestellten Gründen sollten Sie eine separate Freigabe einrichten, in der alle Benutzerprofile gespeichert werden. Definieren Sie z.B. eine Freigabe namens `[profile]`, wie Sie es im letzten Kapitel getan haben, und richten Sie `logon path` wie folgt ein:

```
logon path = \\Server\profiles\Benutzername
```

Die `[profiles]`-Freigabe muss nicht unbedingt auf dem Samba-PDC sein.

Profile und Policies

Benutzerprofile und System-Policies sind konzeptionell die gleichen wie in Windows 9x, aber die Wichtigkeit der beiden ist aufgrund der in Windows NT verfügbaren Sicherheitsmechanismen wesentlich größer.

Stellen Sie sich dieses Beispiel vor: Nehmen wir an, dass Sie System-Policies verwenden, um die Option *Beenden* aus dem Startmenü zu entfernen. Unter Windows 9x kann ein Benutzer einfach die Einstellung rückgängig machen, um die Option *Beenden* wieder einzufügen, `explorer.exe` beenden, die Shell neu starten - et voilà: die Option *Beenden* ist wieder da.

Wenn Sie aber die gleiche Policy unter Windows NT einsetzen, können Sie den Benutzer davon abhalten, sie zu ändern, indem Sie ACLs für verschiedene Registrierungsschlüssel benutzen. Windows NT bietet nicht nur die Policy-Optionen, sondern auch die Methoden, die Policy-Einstellungen durchzusetzen, etwas, das in Windows 9x fehlt.

Benutzerprofile werden unter Windows NT mehr zu einer Notwendigkeit und sind weniger der Luxus, der sie unter Windows 9x sind. Ich behandle Windows-NT-Clients eher wie Unix-Workstations als wie PCs, und zwar in dem Sinn, dass der Rechner in der Regel abgeschlossen

ist, um Änderungen zu verhindern, und dass der Benutzer einen spezifizierten Standort hat (normalerweise das Home-Verzeichnis), an dem er Dateien speichern kann. Dies gilt ganz besonders für öffentliche Laborumgebungen.

Deshalb wird es notwendig, Benutzern eine Umgebung zu bieten, die eingerichtet werden kann und ihnen von Rechner zu Rechner folgt. Benutzerprofile in einer Samba-Controller-Domäne sind standardmäßig wandernde Profile. Dies sollte aus dem Standardwert für den Parameter `logon path` hervorgehen.

Als generelle Regel kann man sagen, dass die Verwaltung von Benutzerprofilen in einer von Samba kontrollierten Domäne genauso ist wie die in einer von Windows NT kontrollierten. Eine Möglichkeit ist z.B., ein `Default-User`-Profil in der `[netlogon]`-Freigabe auf dem PDC einzurichten, das für alle neuen Benutzer verwendet wird. Diese Methode funktioniert gleich, egal ob Ihr PDC ein Samba-Server oder ein Windows-NT-Rechner ist.

Microsoft bietet auf seiner Website einige gute Dokumente über die Administration von Benutzerprofilen. Weitere Informationen finden Sie unter http://www.microsoft.com/ntserver/nts/techdetails/.

Es gibt ein Problem, das auftaucht, wenn Sie einen Samba-Server als PDC benutzen. Wenn Sie keine Kopie der Windows-NT-4.0-Server-CD-ROM haben, woher bekommen Sie dann die Server-Tools wie z.B. den Server-Manager, den Benutzer-Manager für Domänen und den System-Policy-Editor?

Glücklicherweise können Sie diese von der Microsoft-Web- oder FTP-Site herunterladen. Der Policy-Editor wird mit den Service Packs 3 und 4 für Windows NT 4.0 verteilt, obwohl er nicht standardmäßig auf Workstations installiert wird. Die Windows-NT-4.0-Versionen des Server-Managers und des Benutzer-Managers für Domänen sind zum Herunterladen unter http://ftp.microsoft.com/Softlib/MSFILES/ verfügbar.

Der Dateiname ist `SRVTOLL.EXE`. Eine Version dieser Tools, die unter Windows 9x installiert werden kann, ist unter dem gleichen URL erhältlich und hat den Namen `NEXUS.EXE`.

rpcclient

In den Anfangstagen von Samba brauchte Andrew Tridgell Tools, mit denen er den `smbd` testen konnte, also wurde `smbclient` geboren. In den frühen Tagen der Implementierung der Samba-PDC-Unterstützung brauchte Luke Leighton ein Tool, mit dem er die DCE/RPC-Funktionalität im `smbd` testen konnte, und das war `rpcclient`.

`rpcclient` ist immer noch mehr ein Test-Tool als etwas, mit dem Sie eine Aufgabe zu Ende bringen können. Ich erwähne es hier einfach deshalb, weil es die wundervolle Funktion bietet, Informationen über Windows-NT-Rechner zu erhalten und einzurichten. Sie werden sich einige Beispiele ansehen, um die Möglichkeiten zu erkennen. Ich möchte Sie warnen, dass die Ausgabe kryptisch sein kann. Welchen Wert `rpcclient` für Sie hat, ist Ihre Entscheidung.

Loggen Sie sich zunächst über einen lokalen Account auf dem Rechner in eine Windows-NT-4.0-Workstation ein:

```
root# /usr/local/samba/bin/rpcclient -S bilbo -U jerry -W BILBO
Added interface ip=192.168.1.72 bcast=192.168.1.255 nmask=255.255.255.0
Enter password:
smb: \>
```

Sie können eine Liste verfügbarer Befehle erhalten, indem Sie `help` am Prompt `smb: \>` eingeben:

```
smb: \> help
help

svcenum      regenum      regdeletekey regcreatkey  regquerykey
regdeleteval regcreateval reggetsec    regtesttsec  ntlogin
wksinfo      srvinfo      srvsessions  sevshares    srvconnections
srvfiles     lsaquery     lookupsids   lookupnames  enumusers
addgroupmem  addaliasmem  creategroup  createalias  delgroup
delalias     ntpass       samuser      samtest      enumaliases
enumgroups   samgroups    quit         q            exit
```

```
bye           help          ?             !
```

Ich werde nicht alle Befehle erklären, aber Sie können sehen, dass es sich um Funktionen handelt, die Gruppen hinzufügen oder löschen sowie nach Freigaben und Registrierungsschlüsseln oder Werten fragen.

Danach besorge ich einfach einige Informationen über den Server:

```
smb: \> srvinfo
srvinfo

Server Info Level 101:
        BILBO         Wk    Sv    Din    NT    PtB
        platform_id   :               500
        os version    :               4.0
```

Daraus kann ich den Namen des Servers (`BILBO`), den ich bereits kannte, und auch die Version des Betriebssystems auf dem Server, in diesem Fall 4.0, bestimmen.

Sie können auch eher interne Informationen herausfinden, wenn Sie wollen. Windows NT enthält einen Dienst namens *Local Security Authority (LSA)*, der dafür verantwortlich ist, Dinge wie Objektzugriffe und Benutzer-Logins zu überprüfen. Sie können Informationen über die SID des Rechners erhalten, indem Sie den Windows-NT-LSA befragen:

```
smb: \> lsaquery
lasaquery

LSA Query Info Policy
Domain Member     - Domain: CHIPSNDIPS SID: S-1-5-21-123486-344389-124325
Domain Controller - Domain: BILBO SID: S-1-5-21-1842630440-1322791361-134157935
```

Wie ich bereits gesagt habe ist `rpcclient` nicht wirklich ein produktives Tool, sondern es ist ein einfaches Test-Tool für die Samba-PDC-Unterstützung. Es gibt Ihnen einen kleinen Einblick in einige Methoden, die Sie benutzen können, um Informationen über entfernte Windows-NT-Rechner herauszufinden, wie z.B. Einstellungen in der Registry und Benutzer-Accounts, während Sie selbst an einer Unix-Konsole sitzen.

Windows 2000

Wie Sie bereits gehört haben, wird sich mit Windows 2000 (formal als Windows NT 5.0 bekannt) einiges ändern. In einer homogenen Windows-2000-Umgebung werden viele Änderungen zu sehen sein. Viele Leute werden aber Windows NT 4.0 und Windows 2000 gemeinsam laufen lassen. Daher werden Dinge wie WINS realistisch gesehen noch für einige Zeit vorhanden bleiben.

Derzeit (Windows NT 5.0 Beta 2) können sich Windows-2000-Clients nicht in eine von Samba kontrollierte Domäne einloggen. Dies liegt an Änderungen in der Art und Weise, wie Windows 2000 versucht, mit einem PDC zu kommunizieren. Aber seien Sie versichert, dass die Entwicklung für die Implementierung vollständiger PDC-Funktionalität in Samba fortgesetzt wird, und irgendwann wird die Unterstützung für die aktuellste Windows-NT-Version implementiert sein.

Zusammenfassung

Der aktuellste Samba-Entwicklungscode enthält experimentelle Unterstützung für Windows-NT-Domänen-Logins. Hier besteht ein Unterschied zu Sambas Fähigkeit, Windows-9x-Domänen-Logins zu authentifizieren. Derzeit werden die Funktionen für einen Primary Domain Controller offiziell nicht unterstützt. Aber die Dinge werden sich ändern und verbessern, bis dieser Dienst komplett verteilt werden kann.

Einige der weiter verbreiteten Funktionen sind bereits implementiert, darunter die Durchführung von Domänen-Logins, Zuordnungsfunktionen von Windows-NT-Benutzern und -Gruppen zu Unix-Benutzern und -Gruppen sowie wandernde Benutzerprofile und System-Policies.

Frage & Antwort

F. Kann Samba 2.0 als Windows-NT-Domain-Controller agieren?

. Es gibt eine eingeschränkte Funktionalität für PDC-Unterstützung, die in Samba 2.0 implementiert ist, aber sie ist defekt, wenn Sie mit dem aktuellsten HEAD-Branch-Code verglichen wird. Wenn Sie mit Samba-PDC-Funktionen experimentieren wollen, empfehle ich Ihnen, sich stattdessen den aktuellsten HEAD-Branch-Source-Code zu besorgen.

F. Wo kann ich Hilfe bekommen und an wen berichte ich Fehler, wenn ich die PDC-Unterstützung teste?

. Der Mailing-Liste samba-ntdom@samba.org gehören Leute an, die sich damit beschäftigen. Für Anweisungen, wie Sie der Mailing-Liste beitreten können, gehen Sie zu http://samba.org/listproc. Dies wird Ihre beste Informationsquelle sein.

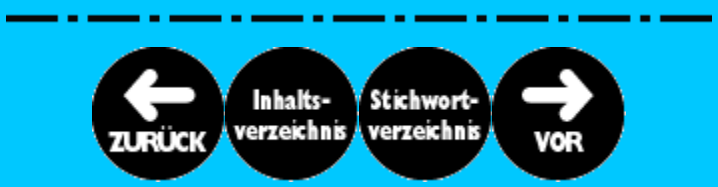
F. Ich scheine nicht gerade viel Dokumentation zu den Samba-PDC-Funktionen finden zu können. Wo sollte ich danach suchen?

. Eine Version des Samba-NT-Domain-FAQs ist auf der CD-ROM enthalten. Die aktuellste Version finden Sie online auf der Samba-Website. Den Link finden Sie im Abschnitt `Documentation`.

Neue Begriffe

HEAD-Branch - Dies ist der aktuellste Samba-Source-Code. Daran arbeiten derzeit die Entwickler, und er ist nicht das Gleiche wie die letzte verteilte Version, z.B. Samba 2.0.

Local Security Authority (LSA) - Der Dienst, der unter Windows NT dafür verantwortlich ist, Zugriff auf lokale Ressourcen, wie eine Konsole oder Netzwerk-Logins, zu kontrollieren und zu authentifizieren.



Anhang B: Tipps und Tricks

von Jerry Carter und Richard Sharpe

Eines der spaßigen Dinge im Leben ist, etwas in einer Art und Weise zum Funktionieren zu bringen, wie Sie es nie zuvor getan haben. Ein guter Freund von mir kam in den Besitz einer Festplatte, die fast zerstört war. Er musste einige Daten, die darauf waren, wiederherstellen, aber der Motor, der die Festplatte drehte, war sehr schwach. Da die Festplatte bereits fehlerhaft war, dachte er, er hätte nichts zu verlieren, entfernte das Gehäuse, drehte die Platten so lange mit einem Schraubenzieher, bis sie schnell genug waren und der Motor wieder übernehmen konnte. Das war wirklich toll! Das ist die Art von Geschichten, die Sie immer wieder an einem Konferenztisch erzählen können.

Dieser Anhang ist eine Zusammenfassung verschiedener Tricks und Tipps, die mein Leben verbessert haben (zumindest was die Verwaltung von Samba-Servern betrifft). Ich bezwecke damit, Ihnen einige Ideen zu geben, die Ihnen helfen können und Ihre eigene Kreativität anspornen. Zwar mag es sich widersprüchlich anhören, aber der Hauptgrund, aus dem ich im Computerbereich landete, waren die kreativen Aussichten. Nun, das und der Film *Tron*, den ich immer noch auf Video habe.

Performance-Tuning

Nur selten muss ich die Performance-Parameter meiner Samba-Server an meinem Arbeitsplatz bearbeiten. Vielleicht ist das ein Vorteil der Hardware, auf der unsere Samba-Server laufen oder der zugrunde liegenden Netzwerkarchitektur. Manchmal aber taucht doch ein Problem auf.

Bevor ich fortfahre, lassen Sie mich sagen, dass jeder Abschnitt über Performance Tuning in einem Buch nur ein einfacher Ratschlag ist und auch so behandelt werden sollte. Es gibt einige gute generelle Vorschläge, aber hauptsächlich sollten Sie experimentieren. Jedes Netzwerk ist anders, und es gibt einfach zu viele Variablen in dem Problem, um alle Schwierigkeiten in jeder Situation lösen zu können.

Wenn Sie Übertragungsraten über Samba oder einen anderen Mechanismus vergleichen wollen, ist eine gute Beurteilung der rohen Übertragungsgeschwindigkeit möglich, wenn sie die Leistung mit der des FTP-Protokolls vergleichen. Alle TCP/IP-Protokolle, die von Microsoft für seine Betriebssysteme verteilt werden, enthalten einen FTP-Client. Messen Sie einfach die Übertragungsrate einer größeren Datei, etwa 8 Mbyte oder so, über FTP und den Standard-DOS-Befehl `copy`. Sie sollten feststellen, dass die Lese- und Schreibraten für eine bestimmte Methode ähnlich sind.

In dem kleinen Netzwerk, das ich zu Hause habe, konnte ich eine Datei über FTP mit etwa 990 Kbps herunterladen. Über den Windows-NT-Befehl `copy` konnte ich die Datei vom gleichen Server mit Samba mit etwa 890 Kbps herunterladen. Die Übertragungsrate war für das Lesen und Schreiben ähnlich. Der Windows-Client war ein Dell Pentium 400 MHz mit Windows NT 4.0 Server. Er hatte 128 Mbyte RAM und eine EIDE-Festplatte. Der Linux-Server war ein Pentium 133 mit 64 Mbyte RAM und ebenfalls einer EIDE-Festplatte. Die Linux-Distribution war Slackware 3.5 mit der Version 2.0.34 des Linux-Kernels.

Netzwerkbandbreite

Wenn Sie Performance betrachten, gibt es mehrere Blickwinkel, aus denen Sie schauen sollten. Der offensichtlichste, auch wenn er oft übersehen wird, ist die Netzwerkbandbreite selbst. Wenn Sie das Netzwerkkabel übersättigen, gibt es nicht viel, was Sie tun können, außer entweder den Netzwerkdatenverkehr zu vermindern oder die Bandbreite zu erhöhen.

Eine einfache Methode, einen Unix-Rechner zu überprüfen, besteht darin, die Befehle `ifconfig` oder `netstat -i` zu verwenden. Sie sollten die Statistiken über einen gewissen Zeitraum beobachten, um eine Vorstellung von der generellen Anwendung zu bekommen. Beide geben Ihnen einen Überblick über die gesamten übertragenen und empfangenen Fehler zusätzlich zu der Anzahl der Kollisionen. Nachfolgend finden Sie eine Ausgabe des Linux-Befehls `ifconfig` von meinem Rechner zu Hause.

Um die Kollisionen (durch die `coll:`-Statistiken angezeigt) zu produzieren, habe ich einen Host von meinem Linux-Rechner mit `echo`-Anfragen überflutet und eine große Datei (etwa 15 Mbyte) vom Host an meinen Samba-Server übertragen, damit der Datenverkehr in beide Richtungen fließt:

```
eth0 Link encap:Ethernet Hwaddr 00:00:F4:D8:6C:0D
     inet addr:192.168.1.72 Bcast:192.168.1.255 Mask:255.255.255.0
     UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
     RX packets:161777 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:159703 errors:0 dropped:0 overruns:0 carrier:0 coll:21734
Interrupt:10 Base address:0xff40 DMA chan:4
```

Wenn die Statistiken zu Ihrem Netzwerkdatenverkehr nicht normal erscheinen, ist eine mögliche Erklärung eine fehlerhafte Netzwerkkarte im Netzwerk. Ich habe Fälle gesehen, in denen eine Netzwerkkarte in einem PC kaputt gegangen ist und anfing, zufällig Pakete in das Netzwerk zu übertragen.

Server-Tuning

Wenn Sie sicher sind, dass das Problem nicht in der Bandbreite des Netzwerks liegt, ist ein anderer Blickwinkel notwendig, aus dem die Dinge angesehen werden können, die Seite des Servers. Generell fällt das Tuning auf Server-Seite in zwei Bereiche:

- Einstellungen für das Datei-Locking
- TCP- und Socket-Einstellungen

Einstellungen für das Datei-Locking

Seit der Freigabe 1.9.18 unterstützt Samba korrekt die SMB-Mechanismen für *opportunistisches Locking*, kurz *oplocks*. Diese ermöglichen Clients, Dateioperationen auf aggressive Art und Weise zwischenzuspeichern. Je nach Client-Einstellungen kann dies zu einer großen Performance-Steigerung führen; es ist standardmäßig aktiviert. Es wird empfohlen, dass Sie für Samba-Versionen höher als 1.9.18 die oplock-Unterstützung nicht deaktivieren. Aktuelle Bewertungen ergaben eine Leistungssteigerung von bis zu 30 Prozent, wenn oplocks aktiviert sind.

Samba bietet auch die Option, `strict locking` zu aktivieren. In diesem Fall überprüft Samba bei jedem Zugriff auf eine Datei den Datei-Lock, statt dies nur dann zu tun, wenn ein Client eine Datei-Lock-Überprüfung verlangt. Wenn Sie `strict locking` aktivieren, werden Sie auf manchen Systemen eine Performance-Verschlechterung feststellen. Standardmäßig ist `strict locking` nicht aktiviert.

Nicht unbedingt relevant für das Locking, aber für Festplatten-E/A im allgemeinen ist der boolesche Parameter `strict sync`. Dieser Parameter, der standardmäßig nicht aktiviert ist, kontrolliert, ob der `smbd` Anfragen von PC-Clients für ein `sync` der Daten auf die Festplatte entspricht. Oft will die Applikation den Speicher einfach entleeren statt für die Daten ein `sync` durchzuführen. Dies ist ein subtiler Unterschied, aber unter Unix blockiert ein `sync()`-Aufruf, bis alle ausstehenden Daten in den Kernel-Puffern auf die Festplatte geschrieben wurden. Dies stellt eine Belastung für die CPU dar und verlangsamt Ihren Dateizugriff. Das Problem wurde erstmals mit der Freigabe von Windows 98 erkannt. Der Windows Explorer schrieb während der Kopie einer Datei die Daten in kleinen Brocken und verlangte nach jedem ein `sync`.

Mit der Standardeinstellung ist es in Hinsicht auf Datenkorruption nur dann gefährlich, wenn das Betriebssystem, auf dem Samba ausgeführt wird, abstürzt. Daher ist die Gefahr sehr klein. Sie werden sicher feststellen, dass die Vorteile das kleine Risiko überwiegen.

Socket-Einstellungen

Die wichtigste Methode für das Tuning der Sockets innerhalb von Samba wird über den Parameter `socket options` ausgeführt. Die möglichen Werte für den Parameter sind:

- `SO_KEEPALIVE`
- `SO_REUSEADDR`
- `SO_BROADCAST`
- `TCP_NODELAY`
- `IPTOS_LOWDELAY`
- `IPTOS_THROUGHPUT`
- `SO_SNDBUF=`*`ganze Zahl`*
- `SO_RCVBUFF=`*`ganze Zahl`*
- `SO_SNDLOWAT=`*`ganze Zahl`*
- `SO_RCVLOWAT=`*`ganze Zahl`*

Die letzten vier Optionen akzeptieren als Wert eine ganze Zahl. Sie könnten z.B. `SO_RCVBUF=16384` spezifizieren. Wenn Sie eine der ersten sechs Optionen aktivieren und das Default Receive Window definieren wollen, könnten Sie Folgendes hinzufügen:

```
socket options = TCP_NODELAY SO_RCVBUF=13384
```

Die Vorteile jeder Option hängen von Ihrer Netzwerktopologie und den Server-Besonderheiten ab. Um Samba zu tunen, brauchen Sie etwas Hintergrundwissen über TCP/IP, z.B. was ist das *Receive Window* und welche Auswirkungen hat es, wenn Sie es ändern. Aber Sie können natürlich auch die Einstellungen einfach blind ändern und sehen, was passiert. Da solche Themen über den Rahmen dieses Buches hinausgehen, empfehle ich Ihnen, die Manpages zur Funktion `setsocketopt` für Ihr System zu lesen und auch einen Blick in die

smb.conf-Manpage zu werfen. Standardmäßig wird die Option TCP_NODELAY aktiviert und sonst nichts:

```
socket options = TCP_NODELAY
```

Client-Tuning

Es ist schwierig, dieses Thema vertieft darzustellen, weil so viele verschiedene Clients und PCs alle möglichen Arten von Netzwerk-Hardware installiert haben können. Microsoft hat eine Liste der konfigurierbaren TCP/IP-Einstellungen für Windows 95 im Knowledge-Base-Artikel mit dem Titel *Windows TCP/IP Registry Entries* (Q158474) veröffentlicht. Ich konnte keinen ähnlichen Artikel für Windows NT finden, aber Dr. Karanjit S. Siyan hat eine gute Referenz mit dem Titel *Windows NT TCP/IP* geschrieben, die von New Riders herausgegeben wird (ISBN 1-56205-887-8).

Um auf die MS Knowledge Base im Web zuzugreifen, gehen Sie zu http://support.microsoft.com/support.

Eine normalerweise verzwickte Einstellung ist die Größe des angezeigten TCP Default Receive Window. Der Standard für Windows 9x ist 8 Kbyte. Einige Leute haben jedoch über bessere Perfomance berichtet, wenn der Wert auf 16.384 Byte erhöht wird. Sie werden mit Ihren Clients experimentieren müssen, um sicher zu sein.

Verschiedene andere Einstellungen

Die Performance Ihres Servers kann auch durch andere Umstände beeinträchtigt werden, die nicht wirklich im Netzwerk begründet liegen. Einer ist der aktuelle Log-Level für smbd und nmbd. Samba kann sehr wortreich sein in Hinsicht auf Debug-Meldungen, die in die Logs geschrieben werden. Das ist gut, wenn Sie versuchen, ein Problem zu lösen, aber es kann Festplattenplatz belegen und die Samba-Daemons verlangsamen, da sie ständig Ausgaben in Logdateien ausschütten. Ziehen Sie in Betracht, die Daemons mit dem Standard-Log-Level 2 laufen zu lassen. Dies gibt Ihnen genügend Informationen für grundlegendes Troubleshooting, ohne Samba zu überlasten.

Ein anderer Grund für langsame Reaktionen während des Logins kann darin bestehen, dass die Einstellungen für Parameter wie username level und password level auf einen sehr hohen Wert wie z.B. 8 gesetzt sind. Denken Sie daran, dass dies dazu führt, dass Samba alle Kombinationen der Groß-/Kleinschreibung bis zum definierten Maximum ausprobiert. Je höher der Level, um so mehr Strings sind für die Authentifizierung zu vergleichen.

Mehrere Samba-Server auf einem Rechner

Eine Sache, die PC-Benutzer und die meisten Netzwerkadministratoren gemeinsam haben, ist, dass wir alle immer denken, wir bräuchten mehr Hardware. Die Spielzeuge sind größer, aber der Hintergedanke bleibt der gleiche. Aber natürlich gilt für uns, dass wir die Hardware auch wirklich brauchen!

Es ist möglich, mehrere Samba-Server auf einem einzigen Rechner mit mehreren Netzwerk-Interfaces zu konfigurieren. Wenn Sie aber das gleiche Ziel über die Einstellungen für die Parameter netbios aliases und include erreichen können, wie es in Kapitel 10, »Automatisierung auf Server-Seite«, beschrieben wurde, kann ich Ihnen nur empfehlen, dies zu tun. Sie sollten nur dann mehr als einen Samba-Server pro Rechner installieren, wenn Sie mehrere Versionen der smbd- und nmbd-Binaries laufen lassen müssen, wenn Sie z.B. Version 2 für Dateifreigaben laufen lassen, aber auch den aktuellsten HEAD-Branch-Code installieren wollen, um als PDC zu agieren.

Zunächst müssen auf dem Rechner beide Netzwerk-Interfaces korrekt konfiguriert sein. Ich habe für den zweiten Samba-Server ein virtuelles Netzwerk-Interface benutzt. Unter Solaris wird dieser fiktionale Netzwerkadapter z.B. als le0:1 bezeichnet.

Danach müssen Sie für jeden Server eine separate smb.conf-Datei einrichten. Natürlich können Sie für übliche Einstellungen die Direktive include verwenden.

In jeder smb.conf müssen Sie zwei Parameter setzen. Der erste ist das Interface, an das Samba sich anbinden sollte. Nehmen wir z.B. an, Sie haben zwei Netzwerkkarten mit den IP-Adressen 192.168.1.90 bzw. 192.168.1.91. Ich werde die Subnetzmaske 255.255.255.0 verwenden. Daher enthält jede smb.conf folgende Einstellung:

```
interfaces = 192.168.1.90/24
```

oder

```
interfaces = 192.168.1.91/24
```

Sie müssen außerdem die Adresse spezifizieren, an der ein bestimmter Samba-Server auf Verbindungen horchen sollte. Standardmäßig werden Verbindungen an jede Adresse akzeptiert. Die Adresse, die Sie über den Parameter `socket address` spezifizieren, muss der für den Parameter `interfaces` entsprechen. Hier ist ein Beispiel für das erste Netzwerk-Interface:

```
socket address = 192.168.1.90
```

Die Verwendung des Parameters `socket address` bricht höchstwahrscheinlich die Standard-Browsing-Regeln. Um dies zu umgehen, müssen Sie einen WINS-Server benutzen und den Samba-Rechner dort registrieren lassen.

Wenn Sie die zwei Server jetzt starten, sollte jeder nur mit einem Interface verbunden sein. Wenn Sie beim Start von `smbd` und `nmbd` auf Probleme treffen, überprüfen Sie die Debug-Logs auf Meldungen wie »not able to bind to address«. Sie können auch versuchen, einen Server zu starten und dann einige Minuten warten, bevor Sie den zweiten Server starten.

Diese Art von Umgebung kann sehr komplex und Fehler können nur schwer zu finden sein. In einem Fall hatte ich einen Samba-PDC auf einem virtuellen Netzwerk-Interface und Samba 1.9.18p7 auf dem realen Interface. Beide Server sollten sich bei einem anderen Samba-Rechner registrieren, der als ein WINS-Server agierte, aber der PDC konnte den Namen *`DOMÄNE`*`<1b>` nicht registrieren. Das Problem lag darin, dass Pakete am primären Interface verloren gingen und der PDC auf dem virtuellen Interface nie die Antwort des WINS-Servers zu sehen bekam. Ich konnte den Fehler beheben, indem ich den PDC startete, fünf Sekunden wartete und dann den 1.9.18p7-Server startete. Wenn Sie entscheiden, ein derartiges System laufen zu lassen, machen Sie sich darauf gefasst, ein wenig herumsuchen zu müssen, bis die Dinge funktionieren.

Die Festplatte eines Remote-PC sichern

Eines der größten Probleme mit PCs liegt im Backup ihrer Festplatten. Benutzer kümmern sich nicht darum, aber wenn sie etwas verlieren, rennen sie zu den Systemadministratoren und verlangen ihre Dateien zurück. In einigen Unternehmen ist es zur Richtlinie geworden, dass die Inhalte der Festplatte auf jedem PC nicht gespeichert werden und dass Dateien, die Benutzer sichern wollen, auf dem Samba-Server gespeichert werden müssen (vielleicht in ihrer `homes`-Freigabe).

In manchen Unternehmen ist dieser Ansatz aber nicht durchzuführen und deshalb werden andere Lösungen gebraucht. Hier ist eine Möglichkeit, das Problem zu lösen. Der Ansatz benutzt `smbclient`, um die Dateien zu sichern, Shell-Skripte und eventuell `cron`-Skripte.

`smbclient` wurde bereits in Kapitel 13, »Unix (smbclient, smbfs, smbwrapper und andene Utilities)«, dargestellt. Das Tool hat einen tar-Modus, mit dem Sie Dateien von einer entfernten CIFS/SMB-Freigabe kopieren und sie als eine TAR- (GNUtar-)Datei speichern können. Hier verwenden Sie diese Funktion, um PCs zu sichern. Sie müssen `smbclient` nicht von Ihrem primären Samba-Server laufen lassen. Tatsächlich müssen Sie es überhaupt nicht von einem Samba-Server laufen lassen. Installieren Sie einfach die Samba-Binaries auf dem System, von dem Sie Backups durchführen wollen.

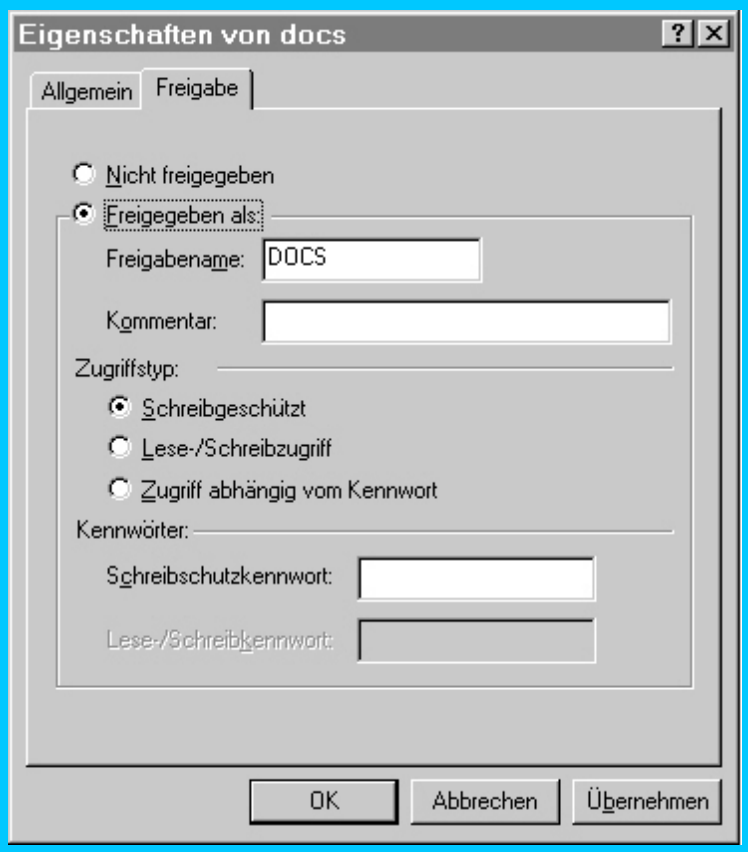
Folgende Schritte müssen Sie für das Backup von PCs mit `smbclient` durchführen:

1. Entscheiden Sie, welche PCs Sie sichern wollen. Dies ist der einfachste Schritt.
2. Richten Sie für jeden PC, der gesichert werden soll, eine Freigabe ein, die Zugriff auf die Dateien und Verzeichnisse für das Backup bietet.
3. Richten Sie die `cron`-Skripte und Shell-Skripte ein, die Sie für das Backup der Dateien auf jedem PC verwenden wollen.

Sie werden sich nachfolgend auf die Schritte 2 und 3 konzentrieren. `smbclient` kann für das Backup von Dateien von Windows für Workgroups, Windows 9x und Windows NT benutzt werden. Um eine Freigabe auf einem PC einzurichten, starten Sie Windows Explorer (oder verwenden Ihre bevorzugte Methode) und wählen das Verzeichnis aus, das alle Dateien enthält, die Sie sichern wollen. Ein Beispiel hierfür ist in Abbildung B.1 dargestellt. Sie sollten das Verzeichnis nur für den Lesezugriff freigeben. Der Vorgang ist für Windows für Workgroups etwas anders. Sie sollten der Freigabe den Namen `backup` oder einen anderen für alle PCs gleichen Namen geben.

Wenn Sie PCs derart einrichten, kann jeder die Dateien auf diesen PCs entfernt lesen. Sind Sie mit dem Internet verbunden, sollten Sie sicherstellen, dass CIFS/SMB-Zugriff vom Internet auf Ihre internen Systeme durch eine Firewall verhindert wird.

Abb. B.1: Freigabe eines Verzeichnisses unter Windows 9x und Windows NT



Wenn Sie Dateifreigaben auf den PCs, die Sie sichern wollen, eingerichtet haben, sollten Sie die `cron`- oder Shell-Skripte einrichten, die die Dateien in diesen Freigaben sichern.

Zwar sind die Skripte wahrscheinlich für jedes Unternehmen spezifisch, aber die Form des `smbclient`-Befehls ist der folgenden ähnlich:

```
smbclient //$client/backup -Tc Gerät oder Datei
```

`$client` ist hier eine Shell-Variable, die den aktuell zu bearbeitenden Client enthält, und *`Gerät oder Datei`* ist der Name des Geräts oder der Datei, auf das bzw. in die das Backup gespeichert wird. Dies kann Folgendes sein:

- `/dev/st0` für ein SCSI-Bandlaufwerk
- `/home/backup/`*`$client`*`.\`Datum +%d%m%Y\`.tar`, um das Backup in eine Datei zu speichern, die durch den Namen des Clients und ein Datum identifiziert wird
- `- |` *`ein anderer Befehl`*, um das Backup an eine andere Funktion zur Verarbeitung weiterzuleiten

Wenn Sie Dateien ausschließen müssen, können Sie den oben stehenden Befehl wie folgt ändern:

```
smbclient //$client/backup -TcX Gerät oder Datei Exclude-Dateien
```

wobei *`Exclude-Dateien`* den in Kapitel 13 dargestellten Regeln folgt. Die Wildcard-Zeichen `*` und `%` funktionieren hier.

Sie sollten sich auch Amanda als Backup-Tool ansehen, da es `smbclient` für das Backup von PCs verwendet, mehrere Backups parallel ermöglicht und einfache Funktionen zur Handhabung von Bandlaufwerken bietet. Die Amanda-Homepage finden Sie unter http://www.cs.umd.edu/projects/amanda.

Faxen

Samba kann benutzt werden, um es Benutzern zu ermöglichen, Faxe direkt von Ihren PCs aus zu versenden. Die grundlegende Strategie ist:

1. Installieren Sie HylaFax (www.vix.com/hylafax) oder mgetty+sendfax (ftp://sunsite.unc.edu/pub/linux/system/Serial/mgetty+sendfax) auf Ihrem Samba-System.

2. Richten Sie unter Samba eine Druckerfreigabe ein, vielleicht mit dem Namen FAX, die Druckaufträge an die LPD-artige Warteschlange FAX weiterleitet. Aufträge an diese Warteschlange werden als PostScript weitergeleitet.
3. Stellen Sie für die Warteschlange FAX einen Druckfilter zur Verfügung, der Folgendes tut:

   Durchsucht PostScript-Dateien nach einem String wie `Fax-Nummer:d [d*][-d*]`

   Leitet die PostScript-Datei entweder an HylaFax oder mgetty+sendfax weiter
4. Auf Clients wird die Warteschlange als beliebiger PostScript-Drucker konfiguriert.
5. Ein Benutzer bereitet das Fax einfach als Textverarbeitungsdokument vor und druckt es an die entfernte Warteschlange.

Eine detaillierte Darstellung darüber, wie Sie HylaFax oder mgetty+sendfax erhalten, installieren und konfigurieren, um ausgehende Faxe zu handhaben, würde den Rahmen dieses Buches sprengen.

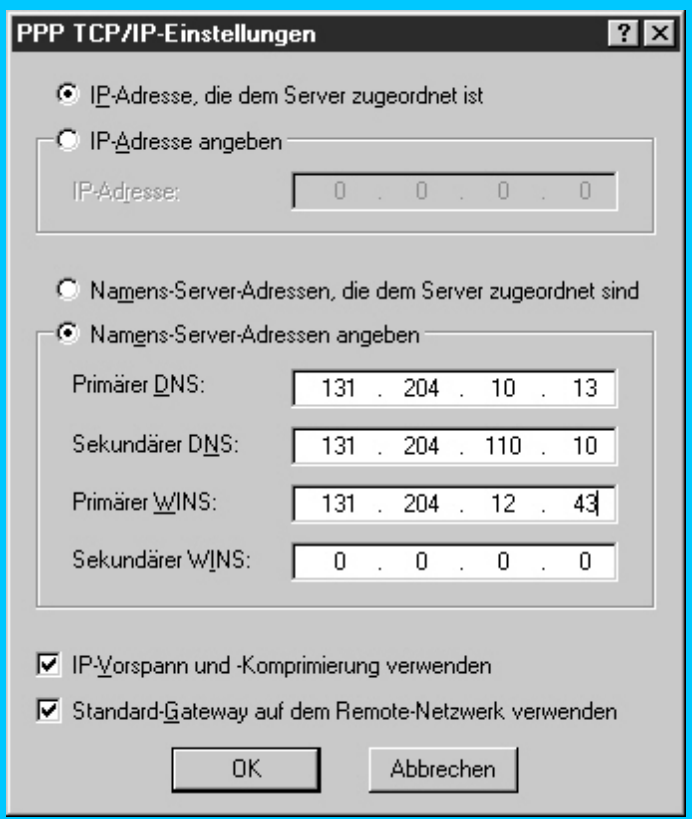
Samba über eine PPP-Verbindung

Der Zugriff auf einen Samba-Server über eine PPP-Verbindung ist dem Zugriff auf einen Server in einem entfernten Netzwerk sehr ähnlich. Die meisten Probleme, auf die Leute treffen, hängen mit der Auflösung von Namen zusammen. Wenn Sie Namen erfolgreich auflösen können, verschwinden viele Probleme von allein.

Für meine Beispiele in diesem Abschnitt werde ich ein Notebook benutzen, auf dem Windows NT 4.0 Workstation läuft. Die präsentierten Ideen sind für die Konfiguration eines Windows-95-PPP-Clients im Wesentlichen gleich. Da Sie sich ansehen werden, wie Sie auf Samba über eine PPP-Verbindung zugreifen können, werde ich auch nicht erklären, wie Sie einen PPP-Server einrichten. Ich werde mich stattdessen von den Details der PPP-Server-Implementierung und der Kontrolleinstellungen auf den Clients distanzieren.

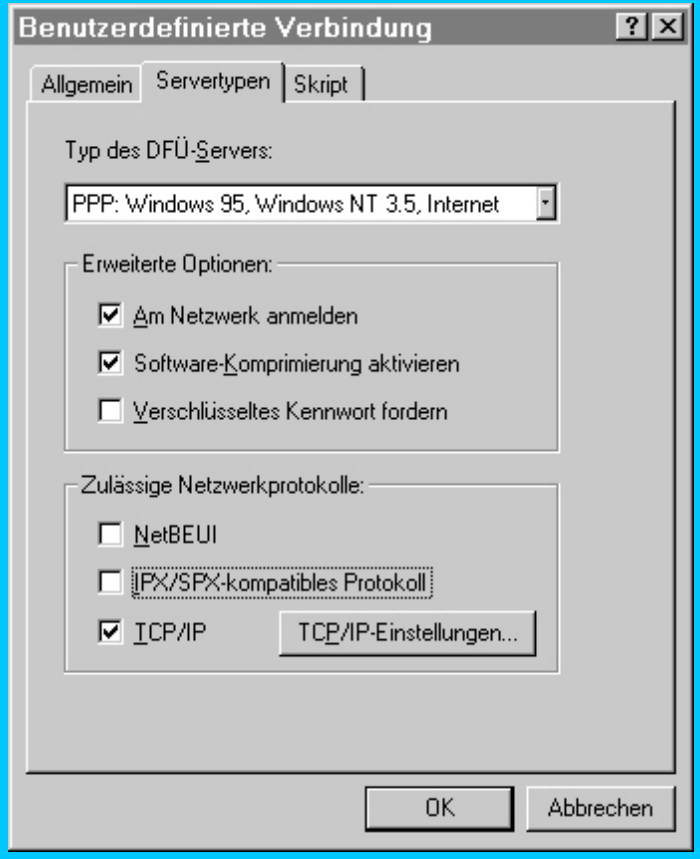
Die erste Sache, die Sie definieren müssen, ist die IP-Adresse Ihres WINS-Servers. Abbildung B.2 zeigt das Fenster für die TCP/IP-Eigenschaften unter Windows NT 4.0. Wie Sie sehen können, habe ich die IP-Adressen der DNS-Server und die einzige WINS-Adresse manuell eingegeben. Je nach Server können diese Adressen bei Aufnahme einer neuen Verbindung dynamisch vergeben werden.

Abb. B.2: Einstellungen für DNS- und WINS-Server für eine PPP-Verbindung unter Windows NT 4.0



Wenn Sie Windows 9x benutzen, sollten Sie außerdem sicherstellen, dass Sie die Option *Am Netzwerk anmelden* (siehe Abbildung B.3) bei den Verbindungseigenschaften markieren und den Client für Microsoft Netzwerke installiert haben, als würden Sie eine Verbindung zu einem LAN konfigurieren.

Abb. B.3: Verbindungseigenschaften unter Windows 9x. Auswahl der Option Am Netzwerk anmelden



Ich habe festgestellt, dass ein anderes wichtiges Detail ist, dass der anrufende Client ein Mitglied der gleichen NetBIOS-Arbeitsgruppe sein sollte wie der WINS-Server. In meinem Fall ist der WINS-Server auch der Domain-Master-Browser der Arbeitsgruppe. Daher kann ich alle Arbeitsgruppen sehen, die auch der WINS/DMB-Server sieht. Es ist nicht notwendig, dass der WINS-Server gleichzeitig auch der PPP-Server ist.

Das Browsing ist, wie immer, der komplizierteste Teil in der Konfiguration einer Anwählverbindung. Vorausgesetzt Sie haben die Option *Am Netzwerk anmelden* unter Windows 9x aktiviert (dies ist in Windows NT integriert), sollten Sie sich mit bestimmten Servern verbinden können, auch wenn das Browsing nicht funktioniert. Dafür müssen nur Ihre WINS-Server-Einträge korrekt sein.

Alle Standardoptionen für den Befehl `net.exe` funktionieren normal. Ich kann z.B. über folgenden Befehl auf mein Home-Verzeichnis zugreifen:

```
net use h: \\burrito\jerry
```

Es ist ebenfalls möglich, über eine PPP-Verbindung Domänen-Logins durchzuführen, indem Sie entweder einen WINS-Server oder einen statischen `lmhosts`-Eintrag benutzen, um den NetBIOS-Namen *`DOMÄNE`*`<1b>` aufzulösen. Aus Spaß habe ich mit diesem einfachen Setup einer von Samba kontrollierten Windows-NT-Domäne über eine Verbindung durch unseren Modem-Pool an meinem Arbeitsplatz einen Windows-NT-4.0-Server hinzugefügt. Ich konnte mich außerdem über die Anwählverbindung in die Domäne einloggen und mein wanderndes Profil herunterladen. Ich würde Ihnen wegen der Übertragungsgeschwindigkeiten nicht empfehlen, dies über eine Anwählverbindung zu tun, aber das Logon-Skript und die Laufwerksverbindungen funktionieren sehr gut.

Einfache Tricks für Domänen-Logon-Skripte

Im Gegensatz zu Windows NT bietet Windows 9x keine einfache Methode, um den Benutzernamen der aktuell eingeloggten Person zu bestimmen. Das ärgert mich. Deshalb zeige ich Ihnen nun, wie ich das umgehen kann.

Zunächst habe ich ein allgemeines Logon-Skript für alle Windows-9x-Clients, das durch folgende Einstellung gesetzt ist:

```
logon script = logon.bat
```

Die Batch-Datei selbst (die in Listing B.1 gezeigt wird) ist relativ einfach. Der Großteil des Skripts ist speziell für mein Unternehmen entwickelt. Aber die Entwicklung ist vielseitig.

Listing B.1: Logon-Skript für Windows-95-Clients

```
1.  @echo off
2.
3.  rem #
```

```
4.  rem # 971007 Jerry Carter  -  COE Network Services
5.  rem #
6.  rem # Login script for Windows 95 clients
7.  rem #
8.
9.  echo Executing Windows 95 login script....
10.
11. rem # Clean up old pcnfsPro 2.0 files if necessary...
12. if exist %winbootdir%\pcnfswin.ini call \\ivy\scripts\delpro.bat %winbootdir%
13.
14. rem ---------- Set the PC's system clock
15. net time \\kudzu /set /yes
16.
17. rem ---------- Mount the normal network drives
18. echo Mapping I: to \\kudzu\apps
19. net use i: \\kudzo\apps
20. echo Mapping H: to \\kudzu\homes
21. net use h: \\kudzu\homes
22.
23 rem ---------- Now the user specific stuff
24. if not exist h:\user.bat GOTO no_user
25. call h:\user.bat
26. :no_user
27.
28. :group
29. rem ---------- Mount the default group drive
30. %COMSPEC% /c \\kudzu\netlogon\group.bat %GROUP%
31.
32. rem ---------- Mount any office shares
33. if "%MACHINE_GROUP%" == "" goto no_machine_group
34. %COMSPEC% /c \\kudzu\netlogon\off-dirs.bat %MACHINE_GROUP%
35. GOTO end
36.
37. :no_machine_group
38. regedit /s /w \\kudzu\netlogon\update.reg
39. GOTO end
40.
41. :end
42. m ******* end of logon.bat
```

Zunächst sollten Sie bemerken, dass das Home-Verzeichnis des Benutzers immer als Laufwerk `H:` gemountet wird (Zeile 21). Dann sollten Sie sehen, dass das Logon-Skript `h:\user.bat` aufruft, wenn es existiert (Zeilen 24 und 25). Die Batch-Datei ist benutzerspezifisch und ich werde in ihr die Umgebungsvariable `USERNAME` setzen.

Wenn eine Verbindung zu einem Home-Verzeichnis eingegangen wird, habe ich ein `preexec`-Skript, das die `~\user.bat`-Datei erzeugt, wenn sie noch nicht existiert. Hier sind die Einstellungen für die `[homes]`-Freigabe:

```
[homes]
   preexec = /usr/samba/lib/netlogon/user.pl %U
   comment = Unix Home Directories
   browseable = no
   path = %H
   writeable = yes
   wide links = no
   create mode = 0600
   directory mode = 0700
   invalid users = @ugrad
```

Listing B.2 enthält den Source-Code für das Perl-Skript, das als Wert für den Parameter `preexec` definiert ist.

Listing B.2: Perl-Skript zum Einrichten von ~/USER.BAT

```
#!/usr/local/bin/perl5
```

```
$user = $ARGV[0]

#get the user's home directory
( $name, $passwd, $uid, $gid, $quota, $comment, $gcos, $dir, $shell ) = getpwnam($user);
( $group, $passwd, $gid ) = getgrgid ( $gid );

if ( ! -f "$dir/user.bat" ) {

   open ( USERINFO, "> $dir/user.bat" ) || die "Couldn't open user.bat!!\n";
   print USERINFO "\\\ivy\\bin\\winset USERNAME=$user\015\n";
   print USERINFO "set USERNAME=$user\015\n";
   print USERINFO "set GROUP=$group\015\n";
   close ( USERINFO );
   chmod 0600, "$dir/user.bat"

}

exit 0
```

Der Befehl `winset` ist ein Tool, das auf der Windows-95-CD-ROM enthalten ist und über das Sie einen Wert im globalen Umgebungsbereich von einem DOS-Befehlsprompt setzen können. Nach dem Einloggen ist diese Information in der Variablen `%USERNAME%` verfügbar.

Die `user.bat` kann auch für die Person spezifische Befehle enthalten, z.B. eine Zuordnung zu einem LPT-Port oder das Kopieren einiger Dateien. Ich habe folgende Befehle eingefügt, um das Verzeichnis `\\Server\netlogon` zu mounten:

```
\\ivy\bin\winset USERNAME=jerry
set USERNAME=jerry
set GROUP=uucp

echo Mapping X: to //burrito/netlogon
net use x: \\burrito\netlogon
```

Die Einträge `\\ivy\bin\winset USERNAME` und `set USERNAME` sind beide notwendig. Der `winset`-Befehl lässt die Variable nicht in der aktuellen Befehls-Shell in Kraft treten. Daher wird der Standardbefehl `set USERNAME` verwendet, damit ich mich von innerhalb des Login-Skripts auf seinen Wert beziehen kann.

Zusammenfassung

Samba bietet viele Möglichkeiten, die Dinge zu tun, die Ihre Aufgabe vereinfachen. Im Fall von Performance Tuning über die `smb.conf` können Sie so viele Details bearbeiten, wie Sie wollen. Oft funktionieren die Standardeinstellungen von Samba gut genug, aber Sie sollten ein wenig experimentieren.

Samba ist ein Tool, das in jeder Art und Weise benutzt werden sollte, die Sie sich vorstellen können. Ich hoffe, dass die in diesem Anhang dargestellten Beispiele Ihre Kreativität geweckt haben und Sie Ihre eigenen Tipps und Tricks entwickeln werden.

Frage & Antwort

F. Wo finde ich weitere Informationen über andere Tipps und Tricks?

. Die besten Informationsquellen sind die Usenet Newsgroup `comp.protocols.smb` und die Haupt-Samba-Mailing-Liste unter samba@samba.org. Sie können das Samba-Archiv für Mailing-Listen durchsuchen, indem Sie den Links auf der Samba-Homepage folgen.

F. Ich habe gerade einen eigenen Tipp oder Trick entwickelt. Wie kann ich anderen Leuten davon berichten?

. Die besten Orte sind die Newsgroup `comp.protocols.smb` und die verschiedenen Samba-Mailing-Listen.

Anhang C: Sambas Zukunft

Jetzt, da Sie am Ende angekommen ist, ist die ganz normale Frage: »Was passiert nun?« Ich habe vorher schon gesagt, dass Samba ein bewegliches Ziel ist, was seine Entwicklung betrifft. Dieser Anhang stellt einige der Dinge dar, die auf der Liste der Dinge zu finden sind, die noch erledigt werden müssen und hoffentlich alle in naher Zukunft implementiert werden. Ich werde nicht nur einen Blick auf die Zukunftspläne für Samba werfen, sondern auch darstellen, welche Auswirkungen diese Pläne für Sie haben könnten.

Eine Sache, an die Sie denken sollten, ist, dass OSS-Projekte wie Samba sehr auf Grundlage eines Angebot-und-Nachfrage-Schemas funktionieren. Funktionen werden in der Regel implementiert, wenn jemand einen genügend hohen Bedarf und die verfügbaren Ressourcen hat, die er in das Projekt eingeben kann. Daher sind die hier präsentierten Funktionen die aktuelle Zielsetzung für die Entwickler. Mit allgemeinen Änderungen und einer Änderung in der Nachfrage können Ressourcen auf neue Funktionen verteilt werden, an die bisher noch niemand denkt.

Unterstützung für den Primary Domain Controller

Einer der größten Anstöße (ich sage einer, nicht der einzige) im Moment ist die Weiterentwicklung in Hinsicht auf die Implementierung der notwendigen Funktionen, damit Samba als Primary Domain Controller für Windows-NT-Domänen agieren kann. Derzeit ist vorsichtig geplant, die Entwicklungen bis zur Freigabe von Samba 2.1 abzuschließen. Aber aufgrund der Komplexität der Details und Tests, um die man sich kümmern muss, mag dies eine sehr optimistische Planung sein.

Ein wichtiges Projekt ist die Implementierung der Fähigkeit, an Vertrauensstellungen mit anderen Domänen-Controllern, entweder Samba oder Windows NT, teilzunehmen. Eines der gesetzten Ziele des PDC-Projekts ist, Samba als identisch mit einem NT-Server erscheinen zu lassen, wenn es über das Netzwerkkabel gesehen wird. Die Fähigkeit, an Vertrauensstellungen teilzunehmen und diese einzugehen, ist eines der letzten großen Teile, die das Puzzle komplett machen.

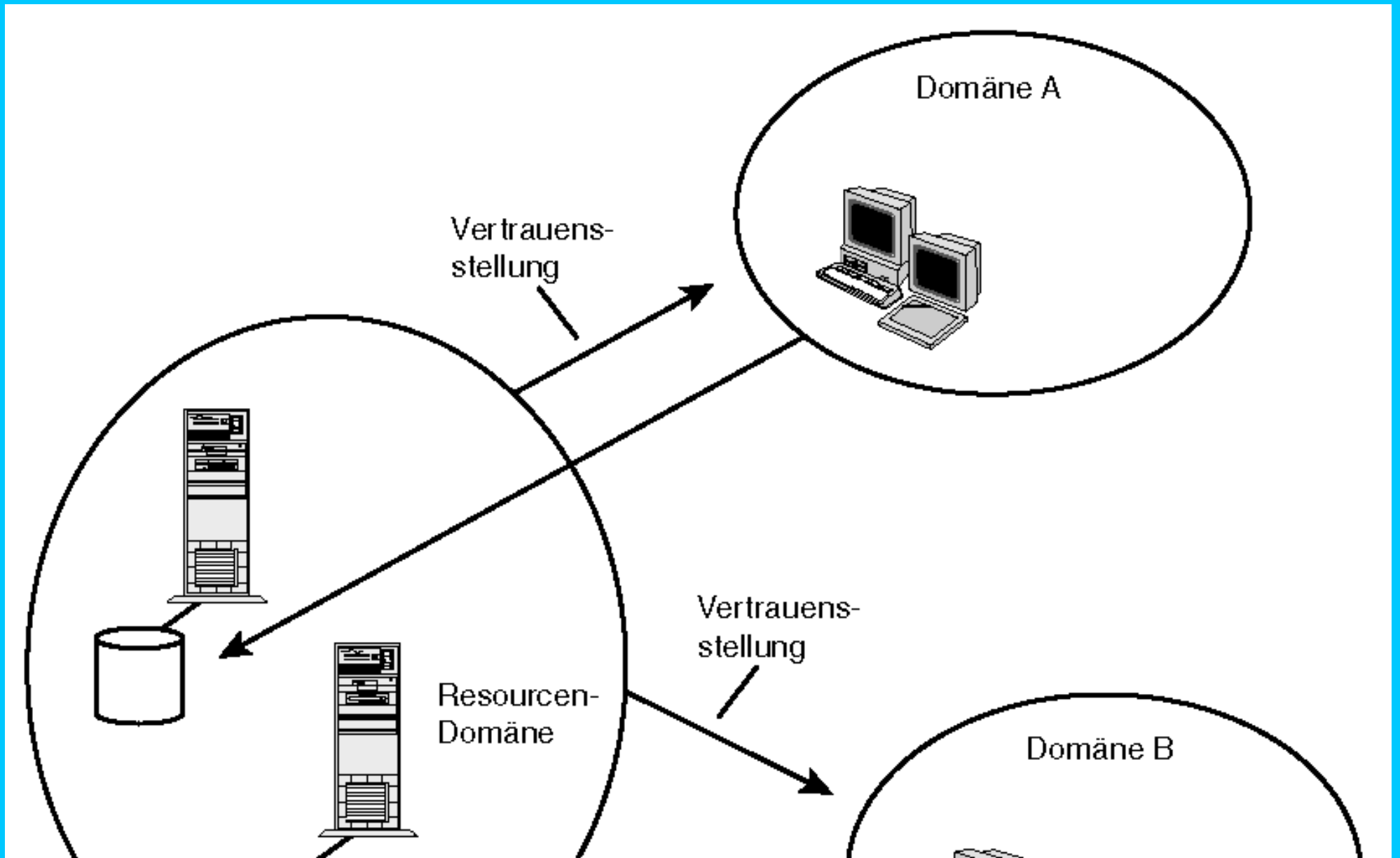
Domänen-Vertrauensstellungen werden nur dann wichtig, wenn Sie mit einem Mehr-Domänen-Modell arbeiten. Wenn Sie sich nicht mit Vertrauensstellungen auskennen und nicht wissen, warum sie wichtig sind, stellen Sie sich Folgendes vor: Ihr Ziel ist, eine Sammlung von Ressourcen für alle Benutzer zur Verfügung zu stellen, ohne die Authentifizierung für diese Benutzer durchzuführen. Das Unternehmensnetzwerk wurde in drei Abschnitte aufgeteilt, die auf den Funktionen der damit verbundenen Abteilungen basieren (siehe Abbildung C.1). Wenn Netzwerkressourcen zwischen zwei oder mehr Domänen freigegeben werden müssen, besteht die einfachste Methode darin, sie in eine Single-Resource-Domäne zu platzieren, die den zugreifenden Domänen zutraut, die Benutzerauthentifizierung für sie durchzuführen. Vertrauensstellungen sind eine Methode. Wenn die `ResourceDomain` der `DomainA` vertraut, können die Benutzer in `DomainA` auf Ressourcen in der `ResouceDomain` zugreifen. Benutzer in der `ResourceDomain` können aber nicht auf Ressourcen in `DomainA` zugreifen, ohne dort einen Account zu haben.

Durch die Implementierung von Domänen-Vertrauensstellungen könnte Samba dann in der Mehr-Domänen-Topologie funktionieren. Bisher kann

Samba eine Windows-NT-Domäne kontrollieren, tut dies aber auf eine isolierte Art und Weise.

Obwohl das Protokoll, das für die Replikation von Windows-NT-SAM-Datenbanken zwischen einem Primary Domain Controller und Backup Domain Controllern verwendet wird, nicht dekodiert wurde, gibt es nur wenige Bemühungen, dies in Samba zu tun. Das liegt daran, dass eine Replikation von Account-Datenbanken zwischen Rechnern sinnlos wird, wenn beide Rechner auf eine von einem LDAP-Server freigegebene Datenbank zugreifen können. Ich werde das als nächsten Punkt darstellen. Wenn Sie die Account-Datenbank wegen der Verteilung der Arbeitslast oder Überbrückung von Ausfällen replizieren wollen, müssen Sie nicht Ihr eigenes Protokoll implementieren, wenn LDAP eine Replikation sowieso schon ermöglicht.

Abb. C.1: Mehrere Domänen und eine Single-Resource-Domäne



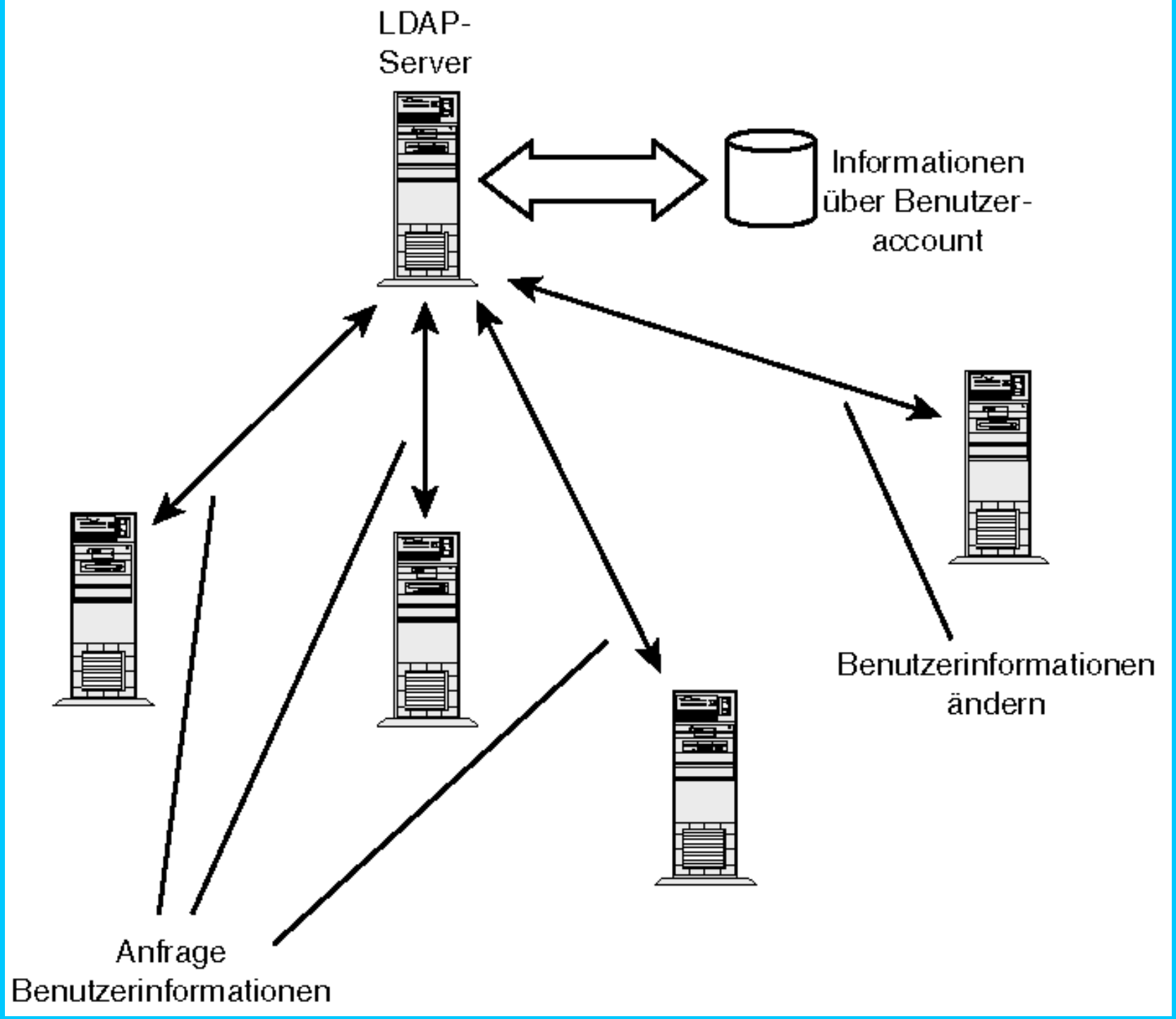
Account-Datenbanken

Account-Datenbanken ist ein passendes Thema, um der Unterstützung für NT-Domänenkontrolle zu folgen. In Kapitel 2, »Windows-Netzwerke«, habe ich erwähnt, dass mit der Version 2 ein Passwortdatenbank-API eingeführt wurde, und ich habe auch bereits die Benutzung eines LDAP-Servers dargestellt. Primär wurde dieses API entwickelt, um die notwendigen Informationen zu verwalten, die Sambas PDC-Funktion für Windows-NT-Domänen unterstützen. Es zog die Informationen zum Benutzer-Account aus den Routinen heraus, die notwendig sind, um auf die Daten zuzugreifen und sie zu modifizieren. Als Ergebnis existieren nun mehrere alternative Benutzer-Account-Datenbanken im Gegensatz zu der einzelnen flachen ASCII-Datei (`/usr/local/samba/private/smbpasswd`). Ist die Passwortverschlüsselung nicht aktiviert, funktioniert Samba normal und verwendet Klartextpasswörter, die mit der `/etc/passwd` verglichen werden. Wollen sie eins dieser experimentellen Backends aktivieren, müssen Sie die Passwortverschlüsselung aktivieren und Samba-Kompilierungsoptionen konfigurieren, um das notwendige Flag zu integrieren.

Schauen Sie sich zuerst die LDAP-Unterstützung von Samba an. Abbildung C.2 stellt eine mögliche Zusammenstellung mit einem einzelnen Rechner als LDAP-Server dar, der alle Benutzer-Account-Informationen verwaltet. Alle drei Samba-Server erhalten Informationen für die Benutzerauthentifizierung von einem einzelnen Punkt. Zwar fügt dies einen einzelnen Ausfallpunkt in Ihre Topologie ein, aber es bietet auch einen einzelnen Punkt für Änderungen, die von allen gesehen werden können. Sie können natürlich auch replizierte LDAP-Server benutzen, um die Arbeitslast auszugleichen und für eine Ausfallüberbrückung zu sorgen.

Derzeit befindet sich die Unterstützung für ein LDAP-Datenbank-Backend in der experimentellen Testphase. Das bedeutet, dass mindestens eine Person einen LDAP-Server bereits erfolgreich konfiguriert und benutzt hat, um Benutzer-Accounts zu speichern. Zwar wurde für die LDAP-Unterstützung kein Freigabedatum gesetzt, aber die Chancen stehen gut, dass Sie eine Produktionsfreigabe zur gleichen Zeit erwarten können, zu der die Primary-Domain-Controller-Funktion offiziell bekanntgegeben wird.

Abb. C.2: Ein einzelner LDAP-Server bietet mehreren Samba-Servern Informationen über Benutzer-Accounts



Es wurden auch Entwicklungen begonnen, NIS+-Tabellen zu verwenden, um Windows-NT-artige Benutzer-Account-Informationen zur Verfügung zu

stellen. Diese `smbpasswd`-Tabelle steht separat von der Standard-`passwd.org_dir` und enthält Informationen wie eine Windows-NT-Liste von Rechnern, auf dem der Benutzer die Erlaubnis hat, sich einzuloggen, und den Standort des wandernden Benutzerprofils. Zwar wurde bereits Code geschrieben, aber das NIS+-Datenbank-Backend ist noch nicht so weit fortgeschritten wie die LDAP-Unterstützung.

Ein anderes Beispiel dafür, welche Möglichkeiten das neue Passwort-API eröffnen kann, ist die Benutzung einer `gdbm`-Tabelle, die `smbpasswd`-Informationen enthält. Dies bietet erhöhte Suchgeschwindigkeit ohne die Komplexität eines Netzwerkdatenbank-Servers.

NTFS-Zugriffskontrolllisten

Wenn Sie sich die Eigenschaften einer Samba-Festplattenfreigabe ansehen, werden Sie bemerken, dass das Dateisystem als Typ NTFS angegeben wird. Der Grund dafür hängt mit der Unterstützung für lange Dateinamen zusammen, wenn Freigaben für Windows-95-Clients zur Verfügung gestellt werden. Die tatsächliche Implementierung von NTFS-Zugriffskontrolllisten für Verzeichnisse und Dateien, die in Samba-Freigaben gespeichert sind, ist aber auch eines der Dinge, die noch getan werden müssen. Obwohl die Entwicklung hierfür bisher noch keine richtige Substanz hat, ist es möglich, dass Sie diese Funktion noch während der 2.0-Phase zu sehen bekommen.

Was ist der tragende Gedanke dahinter, NTFS-ACLs auf einem Unix-Dateisystem implementieren zu wollen, das über SMB freigegeben wird? Vielleicht erinnern Sie sich aus Kapitel 12, »Fallstudie: Einen NT-Datei und Drucker-Server ersetzen«, noch an eines der Probleme, die ich hatte, wenn zwei oder mehr Gruppen Zugriff auf einen bestimmten Ordner hatten. Die Schwierigkeit lag darin begründet, dass Unix-Dateiberechtigungsbits an den Eigentümer gebunden sind, sei dies ein Benutzer oder eine Gruppe. Es gibt keine Methode, das Eigentumsrecht für ein einzelnes Verzeichnis an zwei Gruppen zu vergeben. Dies ist unter Windows NT jedoch üblich.

Das Ziel liegt darin, Benutzern zu ermöglichen, dass sie ACLs auf einen Ordner oder eine Datei setzen können, indem sie die unter Windows NT verfügbaren Standard-Methoden, z.B. Windows Explorer, verwenden, und Samba diese Berechtigungen erzwingen zu lassen. Der Hintergedanke, abgesehen von dem vorher erwähnten, ist, dass Samba in eine Windows-NT-Umgebung platziert wird und sich so verhalten kann, wie es der Benutzer erwarten würde (z.B. NTFS-ACLs und Domänen-Sicherheitsmodell).

Wahres Windows-NT-Drucken

Wie ich in Kapitel 12 beschrieben habe, unterstützt Samba derzeit nicht die volle Skala der Windows-NT-Druckfunktion, aber diese Funktion ist fast fertiggestellt. Etwa 75 Prozent des Codes, der für die Unterstützung der Windows-NT-Druckfunktionen notwendig ist, wurde bereits geschrieben, aber bis zum Zeitpunkt der Bucherfassung ist nichts davon in den Samba-Source-Code-Baum eingefügt worden.

Unter den Funktionen, die im Endprodukt unterstützt werden, sind:

- Drucken über die Microsoft-DCE/RPC-Funktionen, die ein Windows-NT-Rechner benutzt, wenn er an einen anderen Windows-NT-Rechner druckt.
- Die Fähigkeit, Druckertreiberdateien vom Samba-Server herunterzuladen, wie es derzeit für Windows-95-Clients unterstützt wird.

Die Windows-NT-Druckfunktion wird keine Unterstützung für das Windows-*Enhanced-Metafile- (EMF-)*Format beinhalten, da ein Windows-NT-Client, der einen Druckauftrag im EMF-Format an einen Remote-Server überträgt, verlangt, dass der Server den Auftrag unter Benutzung der Windows-Druckertreiber verarbeitet. Da es unmöglich wäre (oder zumindest nicht sehr spaßig), dies auf einem Unix-Rechner zu tun, unterstützt Samba nur Druckaufträge, die im RAW-Format übertragen werden, so dass sie einfach ohne Modifikationen an das Unix-Drucksystem weitergeleitet

werden können.

Höchstwahrscheinlich wird diese Funktion ebenfalls in der 2.1-Freigabe von Samba enthalten sein.

WINS-Replikation

Microsoft-Windows-NT-Server werden mit einem WINS-Server ausgeliefert, der von der CD-ROM installiert werden kann. Dies habe ich kurz in Kapitel 18, »NetBIOS-Namen ohne Broadcasts auflösen«, erwähnt. Sie wissen auch, dass Samba als WINS-Server agieren kann, aber derzeit keine Art der automatischen Replikation seiner WINS-Datenbank mit einem anderen Server, Windows NT oder Samba, bietet.

Es gibt zwei Ansätze, um irgendein WINS-Replikationsprotokoll innerhalb von Samba zu implementieren. Ein Ansatz besteht darin, noch ein anderes nicht dokumentiertes (Überraschung!) Microsoft-Protokoll zu dekodieren. Der Vorteil besteht darin, dass Samba dann mit einem anderen WINS-Server, der auf einer Windows-NT-Plattform läuft, interagieren könnte. Der Nachteil liegt in der Arbeit, die notwendig ist, um die während des Replikationsprozesses über das Kabel gesendeten Pakete zu dekodieren.

Der zweite Ansatz besteht darin, ein komplett neues und möglicherweise besseres Protokoll zu entwickeln, das von Samba benutzt wird. Zwar ist die Protokollentwicklung nicht banal, aber dieser Ansatz hat den Vorteil, direkt aus Sambas internem Code entwickelt zu werden. Die Entwicklung für diese Implementierung wäre wahrscheinlich weniger komplex als die Dekodierung des Microsoft-Protokolls.

Es gibt jedoch auch einen dritten Ansatz, der besagt: »Lasst uns beides tun!« Dieser Vorschlag besteht darin, das Microsoft-WINS-Replikationsprotokoll aus Gründen der Interoperabilität zu implementieren, aber ein feiner abgestimmtes Protokoll für die Replikation zwischen zwei Samba-Servern zu benutzen. Es ist nicht bekannt, wann welcher Weg, wenn überhaupt, gewählt wird.

Derzeit ist Samba als WINS-Server isoliert, darum ist diese Replikationsfunktion wichtig. Es gibt keine Methoden für die Überbrückung eines Ausfalls, Redundanz oder Ausgleich der Arbeitslast. Die dynamische Struktur der NetBIOS-Namen macht es außerdem wichtig, dass Samba die Replikation automatisch handhabt statt durch eine Methode, die versucht, die WINS-Datenbank über `rdist` zwischen Servern zu verteilen. Dies würde tatsächlich nicht funktionieren, da die WINS-Replikation dahingehend entwickelt werden sollte, mehreren primären Servern zu ermöglichen, sich gegenseitig zu aktualisieren, im Gegensatz zu dem Modell, das für primäre und sekundäre DNS-Server verwendet wird.

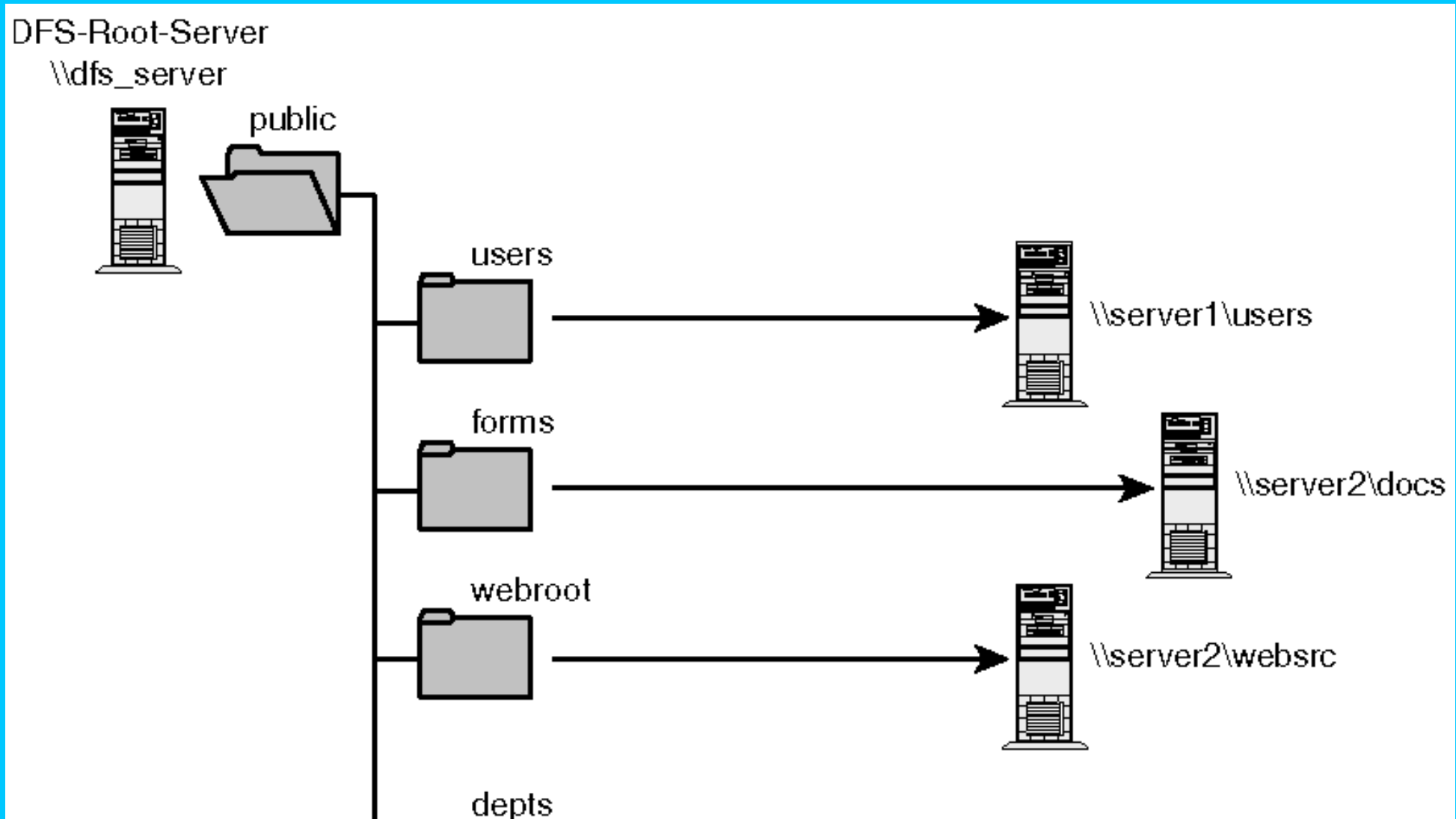
Distributed File System

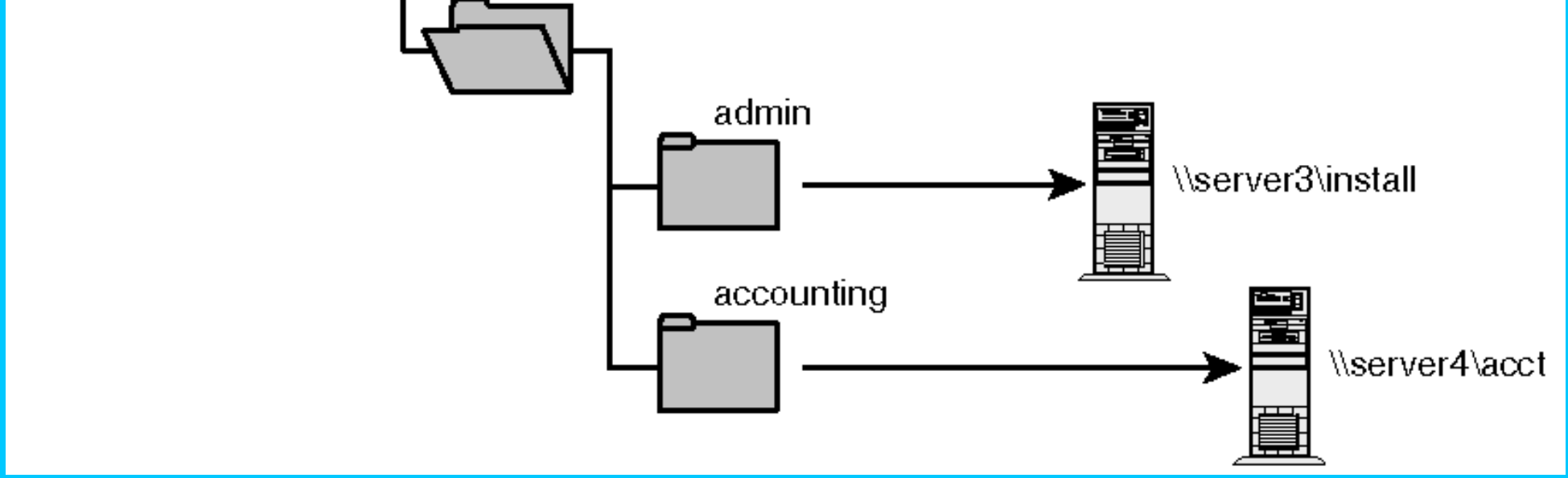
Jemand hat mir gegenüber einmal einen sehr witzigen Kommentar in Hinsicht auf entfernte Festplattenfreigaben in einer Windows-Umgebung gemacht. Er sagte: »Was passiert, wenn man keine Buchstaben mehr hat?« Er benutzte dies natürlich als Beispiel für das Argument, dass NFS einen einheitlicheren Blick des Netzwerks bietet, als es die Standard-UNC-Netzwerkpfade tun. Natürlich habe ich ihm zugestimmt.

Im Vergleich zu NFS können UNC-Netzwerkpfade sehr nervig und schwer zu ändern sein, ohne irgendeine Applikation zu beschädigen. Wie fassen Sie z.B. mehrere SMB-Server in einem Rechner mit nur einem NetBIOS-Namen zusammen, ohne Ihren Benutzern das Leben schwer zu machen? Es muss einen Mechanismus geben, um Netzwerkfreigaben zu ordnen, so in etwa wie Verzeichnisse durch ein Dateisystem auf einer Festplatte geordnet sind. Dies ist der Ansatz, von dem NFS ausgeht.

Das *Distributed File System (DFS)* ermöglicht es einem Administrator, eine »Freigabe der Freigaben« zu erstellen. So können Benutzer einige Verbindungen zu DFS-Root-Servern beibehalten und auf alle benötigten Dateien zugreifen. Abbildung C.3 zeigt, wie DFS den Namensbereich von Servern zusammenfasst und es Benutzern ermöglicht, einen einzelnen Netzwerk-Verzeichnisbaum zu sehen.

Abb. C.3: Einen DFS-Server als Zugriffspunkt für mehrere SMB-Server verwenden





Der Benutzer kann z.B. `\\dfs_server\public` auf einem Laufwerksbuchstaben, z.B. `N:`, mounten. Wenn der Benutzer auf eine Datei in `N:\users` zugreift, bezieht sich die tatsächliche Anfrage auf den UNC-Netzwerkpfad `\\server1\users`. Wenn Sie als Administrator entscheiden, dass die Freigabe `[users]` auf einem anderen Server als `Server1` platziert werden soll, müssen die Benutzer dies wissen. Alles, was Sie tun müssen, ist, die Referenzkonfiguration auf dem DFS-Server für das `users`-Verzeichnis zu ändern, und alle Clients verweisen automatisch auf den neuen Server. Ebenso werden alle Versuche, auf `N:\dept\admin` zuzugreifen, an `\\server3\install` verwiesen.

Eine *DFS-Root-Freigabe* agiert als Mount-Punkt für andere Freigaben. Das Root-Verzeichnis kann *Verbindungspunkte* enthalten, die andere Netzwerk-Festplattenfreigaben oder normale Verzeichnisse darstellen. Die Unterverzeichnisse können ebenfalls Unterverzeichnisse und Verbindungspunkte enthalten. Es ist außerdem möglich, dass ein Verbindungspunkt auf einen anderen DFS-Root-Server verweist. So könnte eine Abteilung ihren eigenen Namensbereich über DFS verwalten und diesen dann für andere innerhalb des Unternehmens durch Konfiguration einer *Referenz* vom Haupt-DFS-Root-Server zur Verfügung stellen.

Zusammen mit einem *Directory Service* wie LDAP verwendet, können so mehrere Server konfiguriert werden, um einen einzelnen Verbindungspunkt zu bedienen. Wenn ein Administrator zwei Server einrichtet, um eine öffentliche Festplattenfreigabe zu replizieren, würde DFS die Client-Anfrage an einen der zwei SMB-Server weiterleiten, die im Verzeichnisdienst aufgelistet sind. DFS bietet keine Methode für die Unterstützung der Festplattenreplikation, sondern nur die Referenz an einen von mehreren Servern.

Dieser kurze Überblick sollte ausreichen, um Ihr Interesse an Sambas zukünftiger Unterstützung für DFS zu wecken. An diesem Punkt wurde etwas

Code geschrieben, aber es müssen noch weitere Entwicklungen durchgeführt werden, um die Unterstützung portabel und verwaltbar zu machen. Obwohl es kein festgelegtes Datum für die Freigabe der DFS-Unterstützung gibt, hat sie eine relativ hohe Priorität.

Windows 2000

Windows 2000 wird viele neue Funktionen bieten und viele der vorhandenen ändern. Statt nun in die Details von Active Directory, Kerberos und NetBIOS-losen Datenfreigaben zu gehen, möchte ich einfach nur sagen, dass Samba so lange mit zukünftigen Versionen von Microsofts Betriebssystemen operieren wird, wie es einen ausreichenden Bedarf dafür gibt. Es ist interessant zu bemerken, dass Microsoft Regressionstests seiner Windows-Clients anhand von Samba durchführt. Dies gilt nur für Samba als Datei-Server und natürlich nicht als PDC.

Zusammenfassung

Die Entwicklung von Samba wird fortgesetzt werden, so lange ein Bedarf dafür da ist. Anscheinend gibt es einen Bedarf, sonst würde ich dieses Buch nicht schreiben und Sie würden es nicht lesen! Die Funktionen, die in diesem Anhang aufgelistet sind (PDC-Funktion, steckbare Benutzeracccount-Datenbanken, Unterstützung für NTFS-ACLs, WINS-Replikation, DFS-Unterstützung und Interoperabilität mit Windows 2000) sind nur einige der Möglichkeiten. Die OSS-Gemeinde kann starke Verbindungen miteinander formen, und ein bestimmtes Projekt kann zu einem Familienmitglied werden. Wenn Sie Interesse haben, werden Sie Mitglied einer Mailing-Liste, nehmen Sie teil am Geschehen und geben Sie Ihre Wunschliste bekannt.

Frage & Antwort

F. Wo kann ich etwas über die aktuellsten Pläne zur Entwicklung von Samba erfahren?

. Der beste Startpunkt ist die Samba-Website unter http://samba.org. Wenn Sie aber wirklich herausfinden wollen, was die Leute denken, sollten Sie sich die verschiedenen Mailing-Listen ansehen, die in Kapitel 11, »Troubleshooting«, vorgestellt wurden.

F. Ich habe eine Idee für eine wirklich tolle Funktion, die in Samba eingefügt werden sollte. Was kann ich damit tun?

. Das beste wäre, Ihre Idee erst zu implementieren und dann die Patches an samba-bugs@samba.org zu senden. Wenn es Ihnen aus irgendwelchen Gründen nicht möglich ist, dies zu tun, senden Sie eine Nachricht an samba-bugs@samba.org mit »WISH« als erstem Wort in der Betreffzeile.

Neue Begriffe

Distributed File System (DFS) - Eine Art von Netzwerkdateisystem, das es SMB-Freigaben ermöglicht, mit dem DFS-Baum gemountet und dann für die Clients über einen einzelnen vereinigten Namensbereich verfügbar gemacht zu werden.

DFS-Root-Freigabe - Das Root-Verzeichnis des DFS-Baums. Dies ist der Teil, der von Benutzern auf ihren lokalen Rechner gemountet wird, genau wie eine SMB-Freigabe. Der tatsächliche Standpunkt einer Freigabe im Netzwerk, die innerhalb des DFS-Baums gefunden wird, ist für den Benutzer transparent.

Verbindungspunkte - Ein Verzeichnis mit einem DFS-Baum, das als Mount-Punkt für eine entfernte Festplattenfreigabe agiert.

DFS-Referenzierung - Der Prozess, einen Verbindungspunkt in einen UNC-Netzwerkpfad aufzulösen.

Directory Service - Eine Netzwerkdatenbank, die verwendet wird, um Objekte wie Drucker, Rechner und Benutzer zu suchen. Der DS hilft Applikationen, Namen in Netzwerkstandorte aufzulösen.

Anhang D: Die CD-ROM

Nachfolgend ein Überblick über den Inhalt der wichtigsten Verzeichnisse auf der CD-ROM:

- /RFC/

  Dieses Verzeichnis enthält die RFCs 1001 (Protocol Standard for a NetBIOS service on a TCP/UDP Transport: Concepts and methods) und 1002 (Protocol Standard for a NetBIOS service on a TCP/UDP Transport: Detailed Specifications).

- /Samba Gui Config Tool 1-0/

  Dieses Verzeichnis enthält ein grafisches Konfigurations-Tool zum Editieren der smb.conf.

- /Samba/samba-2.0.6/

  Hier finden Sie die aktuelle Samba-Version 2.0.6 als tar-Archiv im Quellcode sowie in der gleichen Form die Pakete zu SSL und Webmin.

- /Samba/samba-2.0.3/

  Hier finden Sie die Samba-Version 2.0.3 sowohl im Quellcode wie auch jeweils als Binary für die wichtigsten Distributionen.

- /Scripts/

  Hier finden Sie die Skripts aus der Lektion zu Tag 12.

- /Sharity/

  Enthält den gleichnamigen CIFS-Unix-Client, der es Ihnen ermöglicht, Windows 95-, Windows NT- und sogar OS/2-Freigaben unter Unix zu importieren.

- /SMBedit/

  Dieses Verzeichnis enthält einen smb.conf-Editor für Windows 95 und Windows NT.

- /tcpdump/

  Hier finden das gleichnamige Tool zur Netzwerküberwachung. Lesen Sie die Readme, bevor Sie versuchen, tcpdump aus den Quellen zu kompilieren.

- /tofrodos-1.3/

  Dieses Verzeichnis enthält ein Tool zur Konvertierung von Textdateien von DOS nach Unix (»fromdos«) bzw. von Unix nach DOS (»todos«).

- /Samba_in_21_Tagen/

Dieses Verzeichnis enthält das vorliegende Buch im HTML-Format.