Netze überwachen mit Nagios

LiveSecurity Editorial
für den Watchguard “Smarter Together” Contest 2003

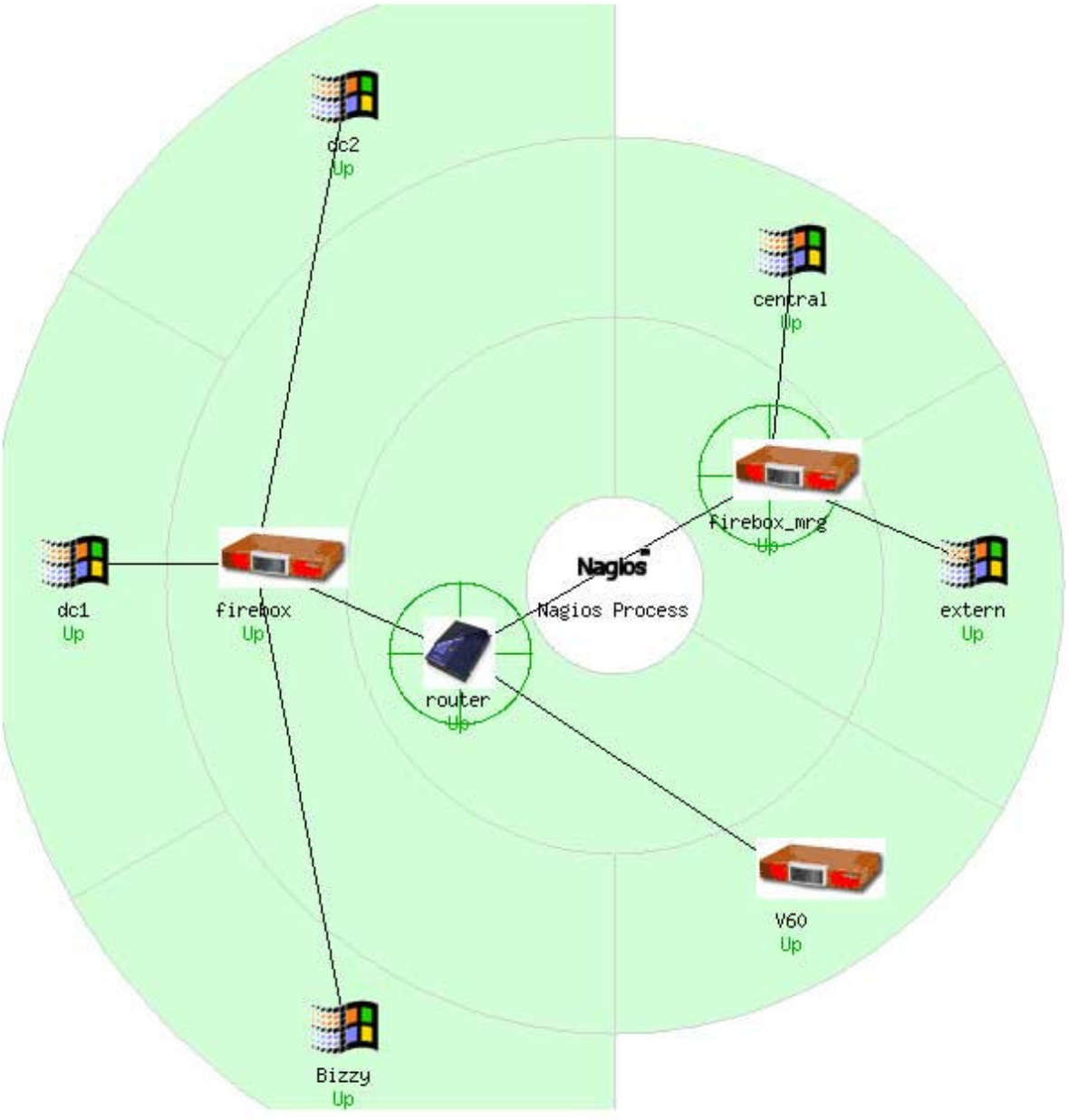
Daniel Heinze
daniel@opengeneration.org

26.05.2003

Der Webserver fällt aus …

und Sie als System-Administrator sind der Letzte, der es erfährt. In jedem Netzwerk, das eine gewisse Größe überschreitet und das kritische Dienste beinhaltet, ist eine zentrale Überwachungseinheit mehr als angebracht. Anstatt von der vollgelaufenen Festplatte immer erst am nächsten Morgen zu erfahren, oder von den Kunden auf den Ausfall des Fileservers hingewiesen zu werden, ist man gut beraten, eine zentrale Stelle zu besitzen, an der alle wichtigen Informationen zusammenlaufen und die den Administrator über kritische Vorkommnisse informiert.

Dabei können Überwachungsprogramme wie Nagios helfen. Nagios (ehemals NetSaint) ist ein Open Source System (veröffentlicht unter der GNU Public License, GPL), das verschiedene Mechanismen für die Überwachung unterschiedlichster Komponenten beinhaltet und sowohl die Möglichkeit von Notifications bietet, als auch über eine übersichtliche grafische Darstellungsweise verfügt.



Das Herz des Überwachungssystems bildet der Nagios Process. Er sammelt Informationen und schreibt sie in Logdateien nieder. Wenn er ein Problem erkennt,

sendet der zentrale Prozess eine Nachricht an die zuständigen Administratoren. Das Webinterface und dessen CGI-Skripte lesen die gesammelten Informationen aus den Files und stellen sie im Browser übersichtlich dar. Über eine Named Pipe, genannt External Command File, lassen sich zusätzliche Kommandos an den Nagios Prozess senden. Diese Schnittstelle nutzt auch das Webinterface: Es schreibt Befehle in die Pipe-Datei, Nagios liest sie und führt sie aus.

Installation von Nagios als zentrale Überwachungseinheit

Die eigentliche Installation des Nagios Demaons erweist sich als nicht besonders kompliziert. Um eine lediglich für einige Tests lauffähige Umgebung zu erhalten, ist die Standard-Server-Installation einer geläufigen Linux Distribution ausreichend, lediglich ein Webserver ist zwingende Voraussetzung. Die Nagios Sourcen (http://www.nagios.org/download) lassen sich, nachdem ein entsprechender User und die dazugehörige Gruppe angelegt wurde, mit

```
configure --with-cgiurl=/nagios/cgi-bin --with-hdmurl=/nagios/ \
          --with-nagios-user=nagios –with-nagios-grp=nagios \
          –-with-template-extinfo
make all
make install
```

kompilieren und installieren. Zu empfehlen ist auch die Installation der Beispiel-Konfigurationsdateien

```
make install-config
```

und der Startskripte

```
make install-init.
```

Der Nagios Prozess kann anschließend über Textdateien konfiguriert werden. Die Grundeinstellungen werden über die Datei >>nagios.cfg<< vorgenommen, die weitere Files mit einbindet (u. a. >>hostgroups.cfg<<, >>hosts.cfg<<, >>contactgroups.cfg<< und >>contacts.cfg>>). Die Dateien dienen hauptsächlich der Definition von Hosts, Diensten und Kontaktpersonen, sie sind gut kommentiert und leicht verständlich.

Plugin Architektur

Als nächstes müssen die verfügbaren Plugins heruntergeladen und installiert werden. Im Gegensatz zu anderen kommerziellen Implementierungen besitzt Nagios keine internen Mechanismen um Statusabfragen generieren zu können, sondern überlässt diese Arbeit externen Programmen (plugins). Diese Plugins sind compiled executables oder Scripte (Perl, shell, etc.), die Nagios benutzt, um den aktuellen Status von Hosts, Netzwerkkomponenten oder Services zu bestimmen. Die eigentliche Statusinformation steckt im Return-Code des Plugins (0 = OK, 1 = Warning, 2 = Critical, 3 = Unknown). Darüber hinaus liest Nagios die erste Zeile der Plugin-Ausgabe („stdout“), speichert die gewonnenen Informationen in den Logfiles und verschickt sie bei Bedarf als Benachrichtigung an die Admins.

Eine ausführliche Installationsanleitung und Tipps zu den verschiedenen Plugins findet sich unter http://www.nagios.org/docs.

Überwachung von Windows Servern mit dem NS Client Plugin

In praktisch jedem größeren Netzwerk hat es der Administrator mit dem einen oder anderen Windows Server zu tun. Um diesen mit Nagios überwachen zu können, wird ein betriebssystemspezifischer Agent, wie z. B. der NS Client (Netsaint Windows Client, http://nsclient.ready2run.nl/download.htm) benötigt. Der NS Client benutzt die Standard Windows API um die Performance-Daten von Windows abzufragen. Er dient demnach der allgemeinen Systemüberwachung und kann z. B. CPU Load, Memory Load, Disk Space, Service State oder System Uptime monitoren. Generell kann jeder Performance Counter eingebunden werden, der über das MS Windows Performance Tool abrufbar ist. Das bedeutet, dass beispielsweise auch Informationen über Exchange Server, SQl Server oder RAS verfügbar sind.

Die Kommunikation zwischen dem zu überwachenden Windows Server und der Nagios Zentrale erfolgt ebenfalls nach dem Plugin-Prinzip. Der NS Client muss auf dem Windows Host installiert sein, dann kann ihn das „check_nt“-Plugin auslesen.

V60		PING		OK	26-05-2003 20:10:45	14d 1h 9m 41s	1/3	PING OK - Packet loss = 0%, RTA = 1.48 ms
		SNMP-Trap	P	OK	23-05-2003 15:18:59	3d 4h 54m 15s	1/1	OK
central		CPU		OK	26-05-2003 20:06:24	4d 1h 19m 18s	1/3	CPU Load (15 min. 8%) (60 min. 11%) (1440 min. 11%)
		DHCP-Server Prozesse		OK	26-05-2003 20:13:12	3d 1h 41m 25s	1/3	tcpsvcs.exe: Running
		Disk C		OK	26-05-2003 20:03:41	4d 1h 19m 18s	1/3	C:\ - total: 7.65 Gb - used: 1.24 Gb (16%) - free 6.41 Gb (84%)
		Disk D		OK	26-05-2003 20:06:00	3d 7h 58m 15s	1/3	D:\ - total: 17.09 Gb - used: 11.67 Gb (68%) - free 5.42 Gb (32%)
		Memory		OK	26-05-2003 20:04:20	4d 1h 19m 18s	1/3	Memory usage: total:736.17 Mb - used: 100.09 Mb (14%) - free: 636.08 Mb (86%)
		PING		OK	26-05-2003 20:10:40	0d 10h 1m 45s	1/3	PING OK - Packet loss = 0%, RTA = 40.56 ms
		SNMP-Trap	P	CRITICAL	26-05-2003 18:14:32	0d 1h 59m 2s	1/1	Ausgabe "CENTRAL: Vorgang Sichern abgebrochen.Anzahl an Fehlern/Warnungen: 0/1"
		WINS-Server Prozesse		OK	26-05-2003 20:13:56	3d 9h 4m 45s	1/3	WINS.exe: Running
dc1		PING		OK	26-05-2003 20:11:16	0d 2h 17m 40s	1/3	PING OK - Packet loss = 0%, RTA = 1.26 ms
		SNMP-Trap	P	OK	22-05-2003 12:49:07	4d 7h 25m 5s	1/1	test
dc2		CPU		OK	26-05-2003 20:13:14	0d 22h 34m 55s	1/3	CPU Load (10 min. 3%) (60 min. 3%) (1440 min. 2%)
		DHCP-Server Prozesse		OK	26-05-2003 20:13:16	3d 1h 42m 55s	1/3	tcpsvcs.exe: Running
		DNS-Server Prozesse		OK	26-05-2003 20:13:14	1d 19h 21m 25s	1/3	DNS.exe: Running
		Disk C		OK	26-05-2003 20:13:12	3d 1h 42m 55s	1/3	C:\ - total: 4.24 Gb - used: 2.09 Gb (49%) - free 2.15 Gb (51%)
		Disk D		OK	26-05-2003 20:13:15	3d 1h 42m 55s	1/3	D:\ - total: 8.55 Gb - used: 6.45 Gb (75%) - free 2.10 Gb (25%)
		HTTP		OK	26-05-2003 20:11:12	4d 1h 13m 23s	1/3	HTTP ok: HTTP/1.1 200 OK - 0.064 second response time
		Memory		OK	26-05-2003 20:13:32	3d 1h 42m 55s	1/3	Memory usage: total:617.99 Mb - used: 474.23 Mb (77%) - free: 143.77 Mb (23%)
		PING		OK	26-05-2003 20:09:27	4d 1h 19m 18s	1/3	PING OK - Packet loss = 0%, RTA = 0.68 ms
		SMTP		OK	26-05-2003 20:13:12	3d 9h 2m 55s	1/3	SMTP OK - 0 second response time
		WINS-Server Prozesse		OK	26-05-2003 20:13:12	3d 1h 42m 55s	1/3	WINS.exe: Running
		snmp cpu		OK	26-05-2003 20:13:15	0d 22h 34m 55s	1/3	cpu OK - 4
extern		CPU		OK	26-05-2003 20:09:46	3d 7h 58m 15s	1/3	CPU Load (15 min. 1%) (60 min. 23%) (1440 min. 3%)
		DNS-Server Prozesse		OK	26-05-2003 20:09:43	0d 3h 36m 55s	1/3	DNS.exe: Running
		Disk C		OK	26-05-2003 20:07:02	3d 11h 18m 10s	1/3	C:\ - total: 7.81 Gb - used: 1.51 Gb (19%) - free 6.29 Gb (81%)
		Disk D		OK	26-05-2003 20:09:22	3d 11h 18m 10s	1/3	D:\ - total: 17.09 Gb - used: 1.84 Gb (11%) - free 15.25 Gb (89%)
		HTTP		OK	26-05-2003 20:09:11	0d 1h 59m 52s	1/3	HTTP ok: HTTP/1.1 200 OK - 0.597 second response time
		MS Exchange Internet Mail Prozesse		OK	26-05-2003 20:09:03	0d 9h 57m 55s	1/3	msexcimc.exe: Running
		MS Exchange Message Transfer Agent Prozesse		OK	26-05-2003 20:09:59	3d 11h 18m 21s	1/3	emsmta.exe: Running
		MS Exchange Verzeichnisdienst Prozesse		OK	26-05-2003 20:13:12	0d 9h 57m 35s	1/3	DSAMAIN.EXE: Running
		MS Exchange-Ereignisdienst Prozesse		OK	26-05-2003 20:09:38	3d 11h 30m 21s	1/3	events.exe: Running
		MS Exchange-Informationsspeicher Prozesse		OK	26-05-2003 20:13:12	0d 8h 36m 55s	1/3	store.exe: Running
		MS Exchange-Systemaufsicht Prozesse		OK	26-05-2003 20:09:17	3d 9h 4m 45s	1/3	mad.exe: Running
		Memory		OK	26-05-2003 20:01:14	3d 9h 11m 5s	1/3	Memory usage: total:959.78 Mb - used: 56.32 Mb (6%) - free: 903.46 Mb (94%)
		PING		OK	26-05-2003 20:13:12	0d 10h 1m 25s	1/3	PING OK - Packet loss = 0%, RTA = 47.67 ms
		SMTP		OK	26-05-2003 20:09:54	4d 1h 18m 59s	1/3	SMTP OK - 18 second response time

Anhand eines zu überwachenden Microsoft DNS Servers soll der Ablauf einer NS Client Installation und Konfiguration beschrieben werden.

Der Nagios Client wird auf dem Windows Server in ein beliebiges Verzeichnis kopiert und mit

```
pNSClient /install
```

als Dienst installiert. Anschließend kann der Dienst „NetSaint NT Agent“ manuell über die Windows Dienstekonfiguration gestartet werden.

Auf dem Linux Rechner muss in der Datei >>checkcommands.cfg<< definiert werden, mit welchen Parametern das check_nt-Plugin aufgerufen werden muss, um die

erwünschten Ergebnisse zu liefern. Um den Status des DNS Dienstes zu ermitteln, würde der Eintrag wie folgt aussehen:

```
define command{
  command_name    check_nt_service
  command_line    $USER1$/check_nt -H $HOSTADDRESS$ -p 1248 \
                  -v SERVICESTATE -d SHOWALL -l $ARG1$
}
```

Zusätzlich muss in der Datei >>services.cfg<< die Zuordnung von Host und auszuführendem Plugin aufgenommen werden.

```
define service{
      use                             generic-service
      host_name                       dc2
      service_description             DNS-Server Services
      is_volatile                     0
      check_period                    24x7
      max_check_attempts              3
      normal_check_interval           1
      retry_check_interval            1
      contact_groups                  nt-admins
      notification_interval           300
      notification_period             24x7
      notification_options            w,u,c,r
      check_command                   check_nt_service!DNS
}
```

Die Datei >>hostextinfo.cfg<< dient der Konfiguration der grafischen Darstellung der einzelnen Elemente in der Ausgabe über das Webinterface.

Nach einem Neustart des Nagios Demons sind nach kurzer Zeit schon erste Statusinformationen über den Webserver abrufbar.

Kommunikation über das SNMP Protokoll

Sehr viel detailliertere Informationen erhält man, wenn man sich des Simple Network Management Protokolls (SNMP) bedient. Prinzipiell werden für die Netzwerkverwaltung mit SNMP eine Management Station, ein SNMP Agent und eine Informationsbasis (Management Information Base, MIB) benötigt. Die Rolle der Management Station übernimmt Nagios, der mit Hilfe des SNMP-Plugins auf SNMP Agents zugreifen kann. Viele Netzwerkkomponenten wie Hosts, Router, Switches oder Firewalls sind mittlerweile mit SNMP ausgestattet, die darauf laufenden SNMP Agents können auf zwei unterschiedliche Arten mit dem Nagios Prozeß kommunizieren: zum einen, indem sie auf Anfragen nach Informationen antworten, zum anderen, indem sie die Management Station selbstständig mit wichtigen Informationen versorgen. Dazu wird das SNMP Protokoll benutzt, das folgende Schlüsselfunktionen enthält: GET (für den Abruf von Werten beim Agenten), SET (um Werte beim Agenten zu setzen) und TRAP (befähigt einen Agenten einer Management Station Ereignisse mitzuteilen). In der MIB werden vereinfacht ausgedrückt die Objekte angegeben, auf die beim jeweiligen Gerät Zugriff besteht (z. B. Angaben über durchgeführte Transaktionen pro Sekunde bei einem SQL Server).

Watchguard Firebox Vclass SNMP Trap Handling

Nagios stellt keinen kompletten Ersatz für die aufwendigen SNMP Management Anwendungen wie HP OpenView oder OpenNMS dar. Mit ein wenig Handarbeit ist es jedoch möglich, Nagios so zu konfigurieren, dass SNMP-Traps vom Linux UCD-SNMP

Demon (snmptrapd) empfangen werden und diese von Nagios ausgewertet werden können. Diese Möglichkeit des Eventhandlings bezeichnet man im Gegensatz zu den aktiven, von Nagios initiierten Abfragen als passive Dienstüberprüfung.

Dies ermöglicht beispielsweise sehr detaillierte Überwachungsmöglichkeiten für Windows 2000 Server (über zusätzliche MIBs von Drittanbietern), die weit über die bisher genannten Möglichkeiten hinausgehen. So können z. B. einzelne SQL-Server Replikationen oder Detailinformationen über den Exchange Storages übermittelt werden.

Auch die Firebox Vclass Serie von Watchguard ist SNMP fähig und demnach gut geeignet, um in ein anspruchsvolles Überwachungsszenario mit eingebunden werden. Dafür muss auf Firebox Seite lediglich über die Vcontroller Software unter *System Configuration -> SNMP* die IP-Adresse des Nagios Servers angegeben werden. Wenn mit SNMP-Traps gearbeitet werden soll, muss zusätzlich *Enable SNMP Traps* aktiviert werden.

Auf dem Nagios Server müssen die RPM-Pakete für den SNMP-Trap Demon installiert werden (ucd-snmp und ucd-snmp-utils). Anschließend ist das Erstellen einer Reihe von Skripten notwendig, die das Handling der eingegangenen Traps beschreiben.

> FIREBOX VCLASS – *trap* → SNMPTRAPD (SNMPTRAPD.CONF) → HANDLE-RAPID-TRAP → SUBMIT_CHECK_RESULT → NAGIOS.CMD → NAGIOS-PROCESS

In der Konfigurationsdatei des SNMP Trap Demons >>/etc/snmp/snmptrapd.conf<< wird definiert, wie mit hereinkommenden Traps umgegangen werden soll.

```
traphandle  RAPID-SYSTEM-CONFIG-MIB::rsAlarmTrap
/usr/local/nagios/libexec/eventhandlers/handle-rapid-trap
```

Dieser Eintrag bedeutet, dass alle einkommenden Traps, die durch `RAPID-SYSTEM-CONFIG-MIB::rsAlarmTrap` gekennzeichnet sind (MIB der VCLASS), an das Skript *handle-rapid-trap* übergeben werden sollen.

```
#!/bin/sh
# handle-rapid-trap

read host
read ip
read Trap
read Uptime
read rsAlarmId
read rsAlarmLabel
read rsAlarmTime
read rsAlarmLevel
read rsAlarmHostname
read rsAlarmMsg

case $host in 192.168.3.11)
      hostname="V60"
      ;;
esac

# CITICAL WARNING
state=2

/usr/local/nagios/libexec/eventhandlers/submit_check_result \
$hostname "SNMP-Trap" $state "$rsAlarmLabel  Error on $host: \
$rsAlarmMsg"

exit 0
```

Das Skript filtert alle relevanten Informationen aus dem eingehenden Trap heraus, belegt die entsprechenden Umgebungsvariablen (z. B. $host, $rsAlarmLabel, $rsAlarmMsg) und übergibt sie dem Skript *submit_check_result*.

```
#!/bin/sh
# submit_check_result

echocmd="/bin/echo"
CommandFile="/usr/local/nagios/var/rw/nagios.cmd"
datetime=`date +%s`
cmdline="[$datetime] PROCESS_SERVICE_CHECK_RESULT;$1;$2;$3;$4"
`$echocmd $cmdline >> $CommandFile`
```

Dieses Skript stellt einen Parser dar, der die Meldungen in eine von Nagios verständliche Syntax übersetzt und in der Datei *nagios.cmd* speichert. *nagios.cmd* ist die Quelle der oben beschriebenen Return-Werte aller Plugins.

Schließlich muss dem Nagios Demon in der >>nagios.cfg<< durch den Eintrag

```
check_external_commands=1
command_check_interval=1
```

mitgeteilt werden, dass er mit „Passive Checks“ (SNMP Traps) zu arbeiten hat.

Anschließend kann in der >>services.cfg<< angegeben werden, dass aufgefangene SNMP Traps behandelt werden sollen. So ermöglicht der Eintrag

```
define service{
      host_name                       V60
      service_description             SNMP-Trap
      is_volatile                     1
      check_period                    24x7
      normal_check_interval           5
      retry_check_interval            1
      active_checks_enabled           0
      passive_checks_enabled          1
      max_check_attempts              1
      contact_groups                  firewall-admins
      notification_interval           30
      notification_period             24x7
      notification_options            c,r
      check_command                   check_none
}
```

dass der Host „V60“ SNMP Traps empfangen kann.

Auf diese Art kann auf alle Ereignisse reagiert werden, die die MIB der Firebox Vclass zur Verfügung stellt. Auch Security relevante Ereignisse können übermittelt werden. So stellt die MIB beispielsweise auch einen Eintrag für die Entdeckung von Spoofing Attacks zur Verfügung. Durch die Konfiguration von Parametern dieser Art kann Nagios zusätzlich als kleines Intrusion Detection System betrieben werden.

Zu erwähnen ist, dass MIBs standardmäßig unter /usr/share/snmp/mibs erwartet werden, desweiteren ist der snmptrapd mit den Optionen –m All und –O v zu starten.

Die beschriebenen Konfigurationsmöglichkeiten reizen Nagios bei weitem nicht aus. Sie zeigen aber, wie mit einem kostenlosen Tool auf einfache Art und Weise grundlegende Netzwerk- und Hostüberwachung implementiert werden kann. Bei der Konfiguration für SNMP Traps ist etwas Handarbeit angesagt, anschließend steht aber ein zuverlässiges Realtime Notification System zur Verfügung, mit dem nicht nur die Verfügbarkeit von

Watchguard Vclass Fireboxes, sondern aller SNMP Trap fähigen Netzwerkkomponenten gewährleistet werden kann.