Aus Linux-Magazin 03/2006

Der Klassiker der Netzwerkadministration: Simple Network Management Protocol

Fernsicht

Michael Schwartzkopff

In bester Bademeister-Tradition brauchen Admins einen guten Überblick über ihr Netzwerk, um eventuelle Fehlerquellen frühzeitig zu entschärfen. Dank SNMP kein Problem: Das Protokoll liefert gezielte Einblicke in jede SNMP-fähige Komponente und gibt ihr bei Bedarf auch Anweisungen.

© photocase.com

Wer viele Rechner aus der Ferne administrieren will, setzt klassischerweise auf das Simple Network Management Protocol (SNMP, [1] bis [5]). Neben Computern verstehen auch Netzwerkgeräte wie Router und Switches dieses Protokoll, viele Drucker sind SNMP-fähig und etliche Applikation bringen SNMP-Agenten mit. Eine Managementkonsole genügt, um den Gerätezoo der meisten Firmen im Blick zu behalten, Ausfälle rasch zu bemerken oder sogar die komplette Administration zu erledigen.

Es scheint aber, dass mehr als 15 Jahre nach der Standardisierung von VersionÂ 1 [1] das vermeintlich uralte Protokoll in Vergessenheit gerät. Viele Entwickler erfinden lieber eigene Management-Protokolle und -Verfahren statt auf das bewährte SNMP zu setzen. Sie verzichten damit auf den großen Vorteil des Standards: seine enorme Verbreitung und Flexibilität. Veraltet ist SNMP noch lange nicht, die RFCs für Version 3 sind gerade drei Jahre jung [3].

Die Nagios-Community ([7], [8]), bekannt auch für ihren erfinderischen Übereifer, hat mit Nagios eines der beliebtesten und besten Überwachungs- und Management-Werkzeuge unter Linux entwickelt. Die Web-basierte Applikation verwendet ein eigenes Client-Server-Protokoll und bringt Agenten und Plugins [9] mit. Genau betrachtet sind viele dieser Plugins unnötig, da Nagios auch SNMP beherrscht.

SNMP statt Plugin

Auf der Nagios-Mailingliste tauchen immer wieder Fragen nach Plugins auf, deren Aufgaben viel einfacher mit SNMP zu erledigen wären. Der Trend setzt sich bei kommerziellen Anbietern fort. Beispielsweise haben HP und IBM so genannte Konnektoren ab 500 Euro im Angebot, die eine bessere Überwachung durch Openview oder Tivoli gewährleisten sollen. Findige Admins verzichten dankend, weil auch jeder SNMP-Agent die benötigten Informationen bereitstellt. Der ist in fast jedem System enthalten, ohne Mehrkosten, und lässt sich meist vielseitig erweitern und anpassen.

Das Management per SNMP [6] funktioniert nach dem Client-Server-Prinzip (Abbildung 1). Im Netz betreibt der Admin eine oder mehrere Network Management Stations (NMS), die ihre zugeordneten Network Management Elements (NME) überwachen. Auf einer NMS sammelt die Manager-Software Daten über den Zustand der NME. Die NMS kontaktiert dazu Agenten, die auf den NMEs laufen und auf UDP-Port 161 reagieren. SNMP dient als Kommunikationsprotokoll zwischen Manager und Agent. Die Daten sind in einem MIB-Baum strukturiert (Management Information Base, siehe auch Tabelle 1 mit einer Akronymliste). Die Structure of Management Information (SMI) gibt vor, wie MIBs aufgebaut sind.

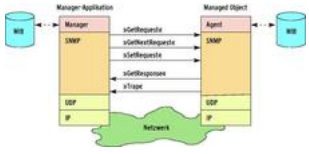

Abbildung 1: Auf jedem per SNMP verwalteten Gerät (Managed Object, rechts)

läuft ein SNMP-Agent (Server-Rolle). Er reagiert auf Fragen des Managers (Client-Rolle: »GetRequest«, »GetNextRequest«) und Änderungswünsche (»SetRequest«) oder sendet von sich aus Hinweise (»Trap«).

Tabelle 1: Abkürzungen

ASN.1	Abstract Syntax Notation Number One
BER	Basic Encoding Rules
CIX	Commercial Internet Exchange
IANA	Internet Assigned Numbers Authority
ISO	International Standards Organization
MIB	Management Information Base
MO	Managed Object
MRTG	Multi Router Traffic Grapher
NME	Network Management Element
NMS	Network Management Station
OID	Object ID
PDU	Protocol Data Unit
RBL	Realtime Blackhole List
RFC	Request For Comments
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
VACM	View based Access Control Model

Falls der Agent meldet, dass alles in Ordnung ist, zeigt beispielsweise das GUI des Administrators ein grünes Lämpchen, bei Problemen ein rotes. Um den Zustand abzufragen, sendet der Manager zunächst ein »GetRequest«-Paket (PDU, Protocol Data Unit), das der Agent mit einer »GetResponse« beantwortet. Den in der MIB an nächster Stelle liegenden Wert erfragt der Manager bequem per »GetNextRequest«. Auch darauf reagiert der Agent mit »GetResponse«.

Um das Verhalten eines NME zu steuern, sendet der Manager Befehle an einen Agenten, der daraufhin die Konfiguration der NME anpasst. Ein Befehl könnte etwa das Routing ändern. Dazu schickt der Manager dem Agenten einen SNMP-»SetRequest«, den der Agent - etwas unerwartet - mit einer »GetResponse« beantwortet. Die Antwort enthält die neuen Werte nach der Änderung.

Völlig aus der gewohnten Client-Server-Rollenverteilung fällt das »Trap«-Paket. Wenn ein Agent selbst ein Problem bemerkt und den Manager darüber informieren will, schickt er unaufgefordert ein »Trap« an den Manager (UDP-Port 162). Das ist per Default bei Authentifizierungsproblemen der Fall oder wenn ein Link (zum Beispiel ein Port am Switch) seinen Status ändert. Das Prinzip der SNMP-Nachrichten ist in Abbildung 1 zusammengefasst.

Alles in MIBs

SNMP-Agenten laufen auf sehr unterschiedlichen Geräten und die Manager-Applikationen gibt es auch von vielen Herstellern - genau hierin liegt der Vorteil der Standardisierung. Ein Kommunikationsprotokoll allein genügt aber nicht, um für globale Verständigung zu sorgen. Folgerichtig legt SNMP auch eine Struktur und eine Darstellungsform für die Management-Information fest. Welche Informationen es konkret gibt und wie Manager und Agent diese Daten adressieren, legt die Management Information Base fest.

Für die eindeutige Adressierung der einzelnen Infos innerhalb einer MIB sind OIDs zuständig (Object Identifier). Informationen über Objekte des Internets befinden sich im hierarchischen Baum unter »iso(1) org(3) dod(6) internet(1)« (Listing 1a). Zur Adressierung genügt auch die Folge der eingeklammerten Zahlen: ».1.3.6.1«.

Listing 1a: MIB-Baum

```
01 internet     OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }
02 mgmt         OBJECT IDENTIFIER ::= { internet 2 }
03 experimental OBJECT IDENTIFIER ::= { internet 3 }
```

```
04 private      OBJECT IDENTIFIER ::= { internet 4 }
05 enterprises  OBJECT IDENTIFIER ::= { private  1 }
```

MIBs beschreiben einen Satz von Objekten inklusive OID, Name, Syntax, Definition, Zugriffsrechten, Status und einer kurzen Erklärung. Die RFCs definieren die MIB-II [4] als Standard. Jeder Agent bietet eine MIB-II mit Daten über den TCP/IP-Stack. Um etwa anzufragen, wie lange ein System schon läuft, fragt der Manager nach der »SysUpTime« (Listing 1b). Deren OID lautet ».1.3.6.1.2.1.1.3.0«, ausgeschrieben: »iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) system(1) sysUpTime(3) sysUpTimeInstance(0)«. Die Beschreibung der OID (Zeile 8) verrät, dass dieses Objekt nicht die Uptime des Rechners wiedergibt, sondern die Zeitspanne seit dem letzten Start des SNMP-Agenten.

Listing 1b: MIB-II

```
01 mib-2  OBJECT IDENTIFIER ::= { mgmt  1 }
02 system OBJECT IDENTIFIER ::= { mib-2 1 }
03
04 sysUpTime OBJECT-TYPE
05   SYNTAX      TimeTicks
06   MAX-ACCESS  read-only
07   STATUS      current
08   DESCRIPTION
09     "The time (in hundredths of a second) since the
10     network management portion of the system was last
11     re-initialized."
12   ::= { system 3 }
```

Viele Hersteller packen zusätzliche Informationen in eigene MIBs. Cisco hat zum Beispiel die Herstellernummer 9. Die MIBs dieser Firma befinden sich unter ».1.3.6.1.4.1.9«, ausgeschrieben: »iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) cisco(9)«.

Für die MIB-Spezifikation verwendet SNMP die Beschreibungssprache SMI (Structure of Management Information, RFC 1155, [1]). SMI legt sowohl die allgemeine Struktur als auch die eindeutige Identifizierung von Netzwerk-Management-Informationen fest. SMI selbst ist in ASN.1 beschrieben (Abstract Syntax Notation Number 1), die es erlaubt, komplexe Datentypen und ihre zugeordneten Werte zu definieren.

ASN.1 BER (Basic Encoding Rules) legt Kodierungsregeln für Datentypen fest. Die räumen auch Zweifel aus, ob beispielsweise 10100101 als String oder als binär geschriebene Zahl 165 zu interpretieren ist; zusammen mit dem Wert kodiert BER auch den Datentyp. Die Versionsnummer 0 für SNMPv1 lautet in BER-Darstellung »02 01 00«. Die erste Stelle besagt, dass ein Feld vom Typ Integer folgt (02), das 1 Byte lang ist und den Wert 0 enthält. Vorteil: Selbst wenn ein Manager die Definition einer MIB eines seiner Agenten nicht kennt, kann das Programm dennoch die Werte der MIB korrekt darstellen. Auch für eine eventuelle Weiterverarbeitung ist es wichtig, den Datentyp zu kennen.

Die größte Schwäche von SNMPv1 ist die mangelhafte Authentifizierung. Lediglich ein so genannter Community-String teilt dem Agenten mit, ob ein Auftrag berechtigt ist. Jeder Agent unterscheidet einen Community-String für Get-Abfragen und einen für Set-Befehle. Jeder Manager, der diese Zeichenkette kennt, kann Daten aus dem Agenten auslesen oder diesem sogar Befehle geben.

« Zurück 1 2 3 4 5 6 Weiter »

Ähnliche Artikel

Net-snmp: Denial of Service möglich

mehr »