

Hochschule für Technik und Wirtschaft Furtwangen
Computer-Networking - Fakultät Informatik

Dokumentation Netzwerkmanagement

Christian Lammers
<lammers@informatik.hs-
furtwangen.de>

Markus Korzendorfer
<markus@dselect.de>

T_EXed with L^AT_EX

Inhaltsverzeichnis

Inhaltsverzeichnis	4
1 Nagios	5
1.1 Installation	5
1.2 Konfiguration	5
2 Einführung MRTG	9
3 Installation und Konfiguration von MRTG unter Debian	9
4 Ergebnisse der Netzprotokollierung	11
5 Alternativen	11

1 Nagios

Nagios ist ein Open-Source Programm mit dem einzelne Hosts und Dienste sowie Netzwerke überwacht werden können.

1.1 Installation

Um die aktuelle Version von Nagios, derzeit 3.0.1-1, zu benutzen, musste die Datei */etc/apt/sources.list* angepasst werden. Der Zusatz von

```
deb http://people.teamix.net/~svelt/debian/etch/nagios3/current/ ./
deb http://people.teamix.net/~svelt/debian/etch/php/current/ ./
```

ermöglichte die Installation der aktuellsten Pakete. Mit

```
apt-get install nagios3 nagios3-common nagios-plugins
```

konnte die Installation gestartet werden. Für die Darstellung der Nagios Webseiten wird der Webserver Apache verwendet. Dieser wird mit

```
apt-get install apache2 libapache2-mod-php5
```

installiert. Gleichzeitig wird noch das php5 Modul für den Webserver mitinstalliert.

1.2 Konfiguration

Zunächst wird die *.htpasswd.users* Datei für den Zugriff auf Nagios konfiguriert.

```
htpasswd -c /etc/nagios3/htpasswd.users admin
htpasswd /etc/nagios3/htpasswd.users nagios
```

Die beiden Befehle legen einen *admin* und *nagios* Benutzer in der *.htpasswd.users* Datei an. In */etc/nagios/cgi.cfg* wird in den Direktiven

```
authorized_for_system_information=admin
authorized_for_configuration_information=admin
authorized_for_system_commands=admin
authorized_for_all_services=admin,nagios
authorized_for_all_hosts=admin,nagios
authorized_for_all_service_commands=admin
authorized_for_all_host_commands=admin
```

festgelegt, dass der Benutzer *admin* Zugriff auf jegliche Bereiche von Nagios erhält. Der normale *nagios* Benutzer allerdings kann sich allerdings nur die Überwachung der Rechner und Dienste anzeigen lassen und keine Konfiguration vornehmen kann.

Die Konfigurationsdateien für die einzelnen Hosts sowie den Diensten werden in */etc/nagios3/conf.d* abgelegt. Um Nagios anzuweisen in diesem Pfad nach Konfigurationsdateien zu suchen, wird in der */etc/nagios3/nagios.cfg* folgender Eintrag konfiguriert.

```
cfg_dir=/etc/nagios3/conf.d
```

Im folgenden wird die Konfiguration von Nagios gezeigt, um die Aufgabenstellung zu erfüllen.

- Prüfen Sie, ob die Dienste der virtuellen Nodes 141.28.65.141(ping), 141.28.65.142(ping, HTTP, MySQL), 141.28.65.xxx(den freien Speicher Ihres Servers)

Die Hosts wie oben genannt, werden in der Datei */etc/nagios3/conf.d/hosts.cfg* angelegt. Deren Syntax sieht wie folgt aus.

```
define host {
    use          generic-host
    host_name    141.28.65.141
    alias        sysadmin-server1
    address      141.28.65.141
    parents      localhost
    hostgroups   ping
}
define host {
    use          generic-host
    host_name    141.28.65.142
    alias        sysadmin-server2
    address      141.28.65.141
    parents      localhost
    hostgroups   ping, http, mysql
}
```

Zu sehen, ist jeweils die *address* die den jeweiligen Host mit seiner IP-Adresse beschreibt. Die *host_name* Direktive dient lediglich dazu, den Host zu identifizieren. Hier koennte ein beliebiger Name, der zu dem Host passt stehen. Über *hostgroups* wird deklariert, welche Dienste auf dem Host überwacht werden sollen. Die Hosts 141.28.15.26, 141.28.2.19, 141.28.64.100, 141.28.64.197, 141.28.64.210, 141.28.78.79, 141.28.33.99 sollen auf Erreichbarkeit, sowie Bereitstellung der Dienste geprüft werden, dabei soll kein dauerhafter Scan erfolgen. Dies wird anhand eines Hosts dargestellt, um den Umfang nicht zu sprengen.

```
define host {
    use          generic host
    host_name    kenny.informatik.hs-furtwangen.de
    alias       kenny
    active_checks_enabled 0;
    address     141.28.64.197
    hostgroups  ping, ssh
}
```

Über die Direktive *active_checks_enabled 0;* wird Nagios angewiesen, den Host nur einmal zu scannen.

Die Überwachung des CVS-Servers geschieht wie folgt:

```
define host{
    use          generic-host
    host_name    son.foo.fh-furtwangen.de
    alias       son.foo
    address     141.28.64.51
    hostgroups  cvs
}
```

Die *hostgroups* Direktive *cvs* muss extra definiert werden, da es kein geeignetes Plugin dafür gibt. Dafür wird die Datei *services.cfg* um einen Service wie folgt erweitert:

```
define service{
    hostgroup_name    cvs
    service_description CVS
    check_command     check_tcp!2401
    use               generic_service
    notification_interval 0
}
```

Für die Überwachung des LDAP Servers *idp.alpha.HS-FURTWANGEN.de* läuft nach dem gleichen Muster ab, wie die des CVS Servers. Es wird zunächst wieder der Host in der *hosts.cfg* angelegt:

```
define host{
    use          generic-host
    host_name    idp.alpha.HS-FURTWANGEN.de
    alias       idp.alpha
    address     141.28.2.1
    hostgroups  ldap
}
```

Die *hostgroups* Direktive *ldap* muss auch extra angelegt werden.

```
define service{
    hostgroup_name      ldap
    service_description LDAP
    check_command       check_tcp!389
    use                 generic_service
    notification_interval 0;
}
```

Die Dienste ping, pop und smtp benötigen keine extra Definition. Diese sind standardmäßig in den Nagios Plugins enthalten. Hier wird lediglich eine zusätzlich Host definiert, der den entsprechenden *hostgroups* hinzugefügt wird.

```
define host{
    use                generic-host
    host_name          averell.foo.fh-furtwangen.de
    alias              averell.foo
    address             141.28.64.55
    hostgroups         ping,pop,smtp
}
```

Der SSH Dienst für die Hosts 141.28.64.179 (kenny schon oben aufgeführt) und 141.28.64.253 wird wie folgt konfiguriert:

```
\define host{
    use                generic-host
    host_name          hack-fw.foo.fh-furtwangen.de
    alias              hack-fw.foo
    address             141.28.64.253
    hostgroups         ssh
}
```

Der SSH Dienst auf dem Host 141.28.78.70 ist vermutlich nur von luckyluke erreichbar, weil der Dienst konfiguriert wurde, nur von diesem IP-Adressbereich Verbindungen zu erlauben. Die 2-D Status sieht damit wie folgt aus.

2 Einführung MRTG

MRTG ist ein universelles Programm, welche per SNMP Werte von Systemen ausliest und grafisch aufbereitet. MRTG kann zu umfangreichen Auswertungen genutzt werden, solange die Werte per SNMP oder Kommandozeile zu ermitteln sind.

MRTG selbst ist in Perl geschrieben und damit auf nahezu jeder Plattform einsetzbar. Zur Aufnahme der Daten nutzt MRTG eine Konfigurationsdatei, aus der MRTG die Datenquellen ermittelt.

Die gesammelten Daten werden in einer besonderen Datenbank gespeichert, welche durch ihr Design nie grösser wird. Ältere Werte werden dabei nach einer bestimmten Zeit zusammengefasst und gespeichert. Damit ergibt sich eine Datendarstellung, dass die letzten Messwerte sehr genau sind und je weiter die Messung in der Vergangenheit liegt, um so weniger Messwerte vorliegen. Dies entspricht aber den meisten Anforderungen, da immer nur die letzten Minuten oder Tage interessant sind und langfristig eher Trends erkennbar sein müssen, als einzelne Werte.

MRTG wird $\tilde{A}\frac{1}{4}$ ber einen Taskplaner möglichst regelmässig aufgerufen. Alternativ kann MRTG auch als Prozess gestartet werden, welcher permanent aktiv ist und selbst in Intervallen die Abfragen durchführt.

Ein zweiter Bestandteil wieder Lösung ist die Aufbereitung der Daten zu Bildern und Webseiten. Da MRTG kein eigener Webserver ist, generiert MRTG bei jedem Durchlauf eine Menge von HTML und PNG-Dateien, welche über einen Webserver bereitgestellt werden können. Werden viele Daten ermittelt, dann ist die zyklische Generierung von Bildern sehr Rechenintensiv, so dass eine andere Variante zum Zuge kommt. MRTG kann auch als CGI- Skript eingebunden und mit RRDTOOL kombiniert werden. Dann werden die Bilder vom Webserver im Moment des Zugriffs erzeugt.

3 Installation und Konfiguration von MRTG unter Debian

Die Installation von MRTG gestaltet sich ein wenig aufwenig. Benötigt werden

- gcc (GNU C Compiler)
- perl (Version 5.005+)
- gd (Grafikbibliothek zur Erzeugung der Grafiken)
- libpng (PNG-Bibliothek, um die PNG-Grafiken zu erzeugen)
- zlib (Bibliothek, die benötigt wird, um komprimierte Grafiken zu erzeugen)

Die benötigte Software findet man unter

- gcc (<http://gcc.gnu.org/>)

3 INSTALLATION UND KONFIGURATION VON MRTG UNTER DEBIAN

- perl (<http://www.perl.com/>)
- gd (<http://www.boutell.com/gd/>)
- libpng (<http://www.libpng.org/pub/png/>)
- zlib (<http://www.info-zip.org/pub/infozip/zlib/>)
- mrtg (<http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/pub>)

Zuerst müssen die von MRTG benötigten Bibliotheken übersetzt werden.

Man beginnt mit `zlib`:

```
tar -xvzf zlib.tar.gz
mv zlib-?.?.?/ zlib
cd zlib
./configure
make
cd ..
```

Anschliessend compiliert man `libpng`:

```
tar -xvzf libpng-x.x.x.tar.gz
mv libpng-x.x.x libpng
cd libpng
make -f scripts/makefile.std CC=gcc ZLIBLIB=../zlib ZLIBINC=../zlib
rm *.so.* *.so
cd ..
```

Nun kann man `gd` übersetzen und installieren:

```
tar -xvzf gd-1.8.3.tar.gz
mv gd-1.8.3 gd
cd gd
make INCLUDEDIRS="-I. -I../zlib -I../libpng"
LIBDIRS="-L../zlib -L. -L../libpng"
LIBS="-lgd -lpng -lz -lm"
cd ..
```

Jetzt hat man die Vorbereitung für die Installation von MRTG abgeschlossen. Die aktuelle Version von MRTG sollte nun vorliegen und kann mit `configure` und `make` MRTG installieren.

Der nächste Schritt ist nun MRTG zu konfigurieren. Dies wird mit einer `mrtg.cfg` Datei initiiert, die definiert was man protokollieren möchte. Die Config-Datei muss man nicht selber schreiben, da MRTG einen `cfgmaker` mitliefert. Dies ist ein Skript, welches alle Interfaces einer Komponente erkennt und ausgibt. Dieses kann man im Unterverzeichnis `bin` finden.

```
cfgmaker --global 'WorkDir: /home/httpd/mrtg'\
--global 'Options[_]: bits,growright'\
--output /home/mrtg/cfg/mrtg.cfg\
xxx.xxx.xxx.xxx
```

Hier wird eine Config-Datei von einer bestimmten IP-Adresse, die man in die letzte Zeile schreibt, in `/home/mrtg/cfg/mrtg.cfg` erstellt. Man sollte nur die Interfaces angeben, die man auch benötigt. Die Auswertung in Form von PNG-Grafiken werden unter `/home/httpd/mrtg` abgelegt, die im WebServer sichtbar gemacht werden sollten.

Nun kann man MRTG starten:

```
<mrtg-bin>/mrtg /home/mrtg/cfg/mrtg.cfg\
--logging /home/mrtg/cfg/mrtg.log
```

Dies veranlasst eine einmalige Messung aller Komponenten und Interfaces, die in der Config-Datei geschrieben wurden. Dies ist aber nicht empfehlenswert, da die Stärken von MRTG in diesem Fall nur wenig bis gar nicht ausgenutzt würden. Besser ist es das MRTG-Skript alle 5 Minuten zu starten und nach einer gewissen Zeit die Daten auszuwerten. Dies geschieht am einfachsten, wenn man in `/etc/crontab` folgenden Eintrag hinzufügt:

```
*/5 * * * * mrtg-user <mrtg-bin>/mrtg /home/mrtg/cfg/mrtg.cfg\
--logging /home/mrtg/cfg/mrtg.log
```

Jetzt nur noch mit `/etc/crontab crontab` den Job aktivieren und es wird alle 5 Minuten der Verkehr am jeweiligen Interface gemessen und im entsprechenden `httpd` Verzeichnis die Grafik aktualisiert.

4 Ergebnisse der Netzprotokollierung

Kann ich erst machen, wenn ich wieder in FuWa bin.

5 Alternativen

Neben MRTG gibt es noch zahlreiche andere Netzwerkmonitoring-Tools. Neben recht teuren Lösungen wie `HPOpenView` von Hewlett Packard oder `Tivoli` von IBM, freuen sich auch zunehmend kostenlose immer mehr zu grösserer Beliebtheit. Zu nennen wäre hier an der Stelle `Cacti`, dass auf dem `RRDtool` basiert.

`Cacti` ist ein Frontend für `RRDTool` welches alle nötigen Daten über die zu erstellen- den Graphen in MySQL speichert. `Cacti` sorgt ausserdem für das Einsammeln der Daten die zur Erstellung der Graphen benötigt werden. Dies kann entweder per SNMP Abfragen geschehen, oder über vom User bereitgestellte Skripte, womit quasi alle mit `RRDTool`-Graphen

darstellbaren Grössen abgefragt und angezeigt werden können. Aber auch die Darstellungen von Umweltbedingungen wie Temperatur, Luftfeuchtigkeit und vieles mehr sind prinzipiell möglich.

Sowohl für Datenquellen als auch für Graphen lassen sich eigene Templates definieren, sodass man auch hier vollkommene Freiheit im Aussehen/Verhalten haben kann.

`Cacti` bietet die Möglichkeit der Nutzerverwaltung, sodass man einzelne Nutzer beispielsweise auf bestimmte Graphen einschränken kann während anderen wieder die komplette Verwaltung Ihrer Graphen und Einstellungen möglich ist.