

Zusatzkomponenten/Plug-Ins und Perl Skripte für ICINGA 1.0.2 installieren

Inhalt

Zusatzkomponenten/Plugins & Perl Skripte für ICINGA einbinden.	2
SNMP auf den ICINGA Server aktivieren	2
NRPE installieren für Windows Abfragen	4
PNP4NAGIOS	6
Bulk Mode mit NPCD konfigurieren	7
Das PNP4NAGIOS in das alte ICINGA Web Frontend einbinden	10
Microsoft SQL Server Express Check	11
Dell Hardware Check	12
Cisco Hardware Check.....	15

Zusatzkomponenten/Plugins & Perl Skripte für ICINGA einbinden.

Um ICINGA noch effektiver nutzen zu können benötigen wir für bestimmte Abfragen wie zum Beispiel für den Microsoft SQL Servern oder aber auch für die Dell Server Hardware und Cisco Switches bestimmte Plug-Ins bzw. Perl Skripte die man bei Nagios Exchange downloaden kann.

<http://exchange.nagios.org/directory/Plugins>

<http://www.monitoringexchange.org/>

Hier ein Beispiel:

Für die Netzwerkkarten von Unix und Windows Systemen, benutze ich das Perl Script **check_iftraffic3.pl** von Nagios Exchange --> http://exchange.nagios.org/directory/Plugins/Network-Connections,-Stats-and-Bandwidth/check_iftraffic3/details

Um solche Perl Skripte nutzen zu können, benötigt der Ubuntu Server noch ein paar Konfigurations-Schritte bzw. Software Pakete um die Abfragen fehlerfrei durchführen zu können.

SNMP auf den ICINGA Server aktivieren

Um z.B. das Perl Skript **check_iftraffic3.pl** für den Ubuntu Server nutzen zu können müssen wir SNMP auf den Server freischalten. Das funktioniert wie folgt:

sudo -s

mv /etc/snmp/snmpd.conf /etc/snmp/snmpd.backup

touch /etc/snmp/snmpd.conf

nano /etc/snmp/snmpd.conf

```
rocommunity    icinga
syslocation    "ICINGA IT Monitoring"
syscontact     email@adresse.de
```

nano /etc/default/snmpd

```
# snmpd control (yes means start daemon).
SNMPD_RUN=yes

# snmpd options (use syslog, close stdin/out/err).
#SNMPDOPTS='-Lsd -lf /dev/null -u snmp -g snmp -I -smux -p /var/run/snmpd.pid 127.0.0.1'
SNMPDOPTS='-Lsd -lf /dev/null -u snmp -I -smux -p /var/run/snmpd.pid -c /etc/snmp/snmpd.conf'
```

SNMPDOPTS='-Lsd -lf /dev/null -u snmp -I -smux -p /var/run/snmpd.pid -c /etc/snmp/snmpd.conf'

/etc/init.d/snmpd restart

und zum zweiten installieren wir noch ein paar Software Pakete um die Perl Skripte lauffähig zu machen, **bitte überprüft aber selbst, ob diese Software Pakete für euch von nutzen sind.**

Wenn Ihr diesem Howto von meiner Webseite gefolgt seid, dann benötigt Ihr diese untenstehenden Pakete -> [ICINGA 1.0.2 in einer Microsoft Hyper-V Maschine installieren](#)

Wenn nicht dann würde ich euch folgende Vorgehensweise vorschlagen, um neue Plug-Ins zu testen:

1. Plug-in/Perl Skript downloaden und in das Verzeichnis **`/usr/local/icinga/libexec`** kopieren
2. Folgende Rechte auf das Plugin setzen
 - **`chmod 4755 /usr/local/icinga/libexec/Plugin_Namen.pl`**
 - **`chown icinga:root /usr/local/icinga/libexec/Plugin_namen.pl`**
3. Sich als Icinga Benutzer auf der Console einloggen → **`su - icinga (Passwort)`**
4. **`#bash`**
5. Plugin testen → **`/usr/local/icinga/libexec/check_iftraffic3.pl -H Servername -C SNMP Community String -A IP-Adresse des Netzwerk Adapters -b 1000 -w 90 -c 95 -B -u m`**
6. abwarten was passiert → Fehlermeldungen im Notfall über Google herausfinden oder sich an das Forum von Nagios/Icinga wenden → <http://www.nagios-portal.org>

Das wichtigste Paket das benötigt wird ist CPAN

`perl -MCPAN -e "install Bundle::CPAN"`

`perl -MCPAN -e 'install Nagios::Plugin'`

alle anderen Pakete kamen nach und nach dazu, weil verschieden Perl Skripte verschiedene Anforderungen hatten.

`apt-get install perl-suid`

`apt-get install libnet-dns-perl`

`optional: apt-get install libdbd-sybase-perl`

`optional: apt-get install libdbd-odbc-perl`

`optional: apt-get install libconfig-general-perl`

`optional: apt-get install tdsodbc`

`apt-get install freetds-common`

`apt-get install freetds-dev`

Wenn ihr alles sauber installiert und konfiguriert habt, dann solltet ihr bei der ersten Abfrage folgenden Output bekommen.

**Average IN: 0Kbs (0.00%), Average OUT: 0Kbs (0.00%)
Total RX: 4300.68 Mbits, Total TX: 1818.22 Mbits|inUsage=0.00%;90;95 outUsage=0.00%;90;95 inBandwidth=0Kbs outBandwidth=0Kbs inAbsolut=563698855 outAbsolut=238317663**

NRPE installieren für Windows Abfragen

NRPE ist ein Addon, das erlaubt, Plug-Ins auf entfernten Linux-/Windows Rechnern auszuführen. Dies ist natürlich nützlich, wenn die lokale Ressourcen/Attribute wie Plattenbelegung, CPU-Last, Speicherbelegung usw. auf entfernten Hosts überwacht werden.

Hier wieder ein Beispiel:

Auf meiner gesamten Windows Farm setze ich den NSClient++ ein, in der Konfigurations- Datei [nsc.ini] kann man nun bestimmte Befehle hinterlegen die der ICINGA Server ab fragt.

In diesen Beispiel möchte ich gerne den Physikalischen Arbeitsspeicher abfragen und nicht den gesamten Arbeitsspeicher samt Auslagerungsdatei.

Auf den Client definiere ich folgendes in der [nsc.ini]:

```
[NRPE]
.....
; CheckMem/Check_nrpe Abfrage für physikalischen Arbeitsspeicher
check_mem=inject CheckMEM MaxWarn=80% MaxCrit=90% ShowAll=long type=physical
```

Auf den Server definiere ich folgendes in der [command.cfg]:

```
# 'check_nrpe' CheckMEM physikalisch
define command {
command_name    check_mem
command_line     $USER1$/check_nrpe -H $HOSTADDRESS$ -p 5666 -c CheckMEM -a MaxWarn=$ARG1$% MaxCrit=$ARG2$%
ShowAll=long type=physical
}
```

In der [Server.cfg] definiere ich die Abfrage wie folgt

Memory Auslastung

```
define service{
    use                windows-service,svr-pnp
    host_name          servername.domain.de
    service_description Memory Auslastung
    check_command       check_mem!95!99
}
```

Wie immer wechseln wir in das Ubuntu Source Verzeichnis

sudo -s (Passwort)

cd /usr/local/src

und starten den Download

wget <http://sourceforge.net/projects/nagios/files/nrpe-2.x>

Extrahieren das Paket mit folgendem Befehl

tar xvfz nrpe-2.12.tar.gz

wechseln in das Verzeichnis von NRPE

cd nrpe-2.12

Konfigurieren das Paket

./configure --prefix=/usr/local/icinga --with-nagios-user=icinga --with-nagios-group=icinga --with-nrpe-user=icinga --with-nrpe-group=icinga --enable-command-args

Und installieren im Anschluss das Paket

make all

make install-plugin

apt-get update

apt-get install nagios-nrpe-plugin

Ab jetzt ist auf den Server NRPE einsatzbereit, auf den Client muss man die folgenden Punkte in der nsc.ini aktivieren damit der Server per NRPE darauf zugreifen kann

NRPEListener.dll

```
[NRPE]
;# NRPE PORT NUMBER
;This is the port the NRPEListener.dll will listen to.
port=5666
;
;CheckMem/Check_nrpe abfrage für physikalischen Arbeitsspeicher
check_mem=inject CheckMEM MaxWarn=80% MaxCrit=90% ShowAll=long type=physical
;
;# ALLOWED HOST ADDRESSES
;This is a comma-delimited list of IP address of hosts that are allowed to talk to NRPE daemon.
;If you leave this blank the global version will be used instead.
allowed_hosts=IP Adresse vom ICINGA Server
```

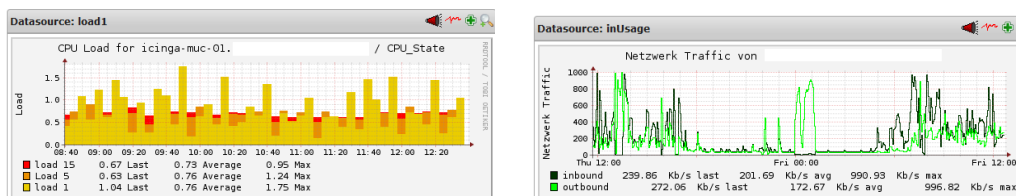
Hier noch ein paar Beispiel Befehle die man einsetzen kann

```
check_mem=inject CheckMEM MaxWarn=80% MaxCrit=90% ShowAll=long type=physical
check_cpu=inject checkCPU warn=80 crit=90 5 10 15
check_disk_c=inject CheckFileSize ShowAll MaxWarn=1024M MaxCrit=4096M File:WIN=c:\ATI\*. *
dont_check=inject dont_check This will "loop forever" so be careful with the inject command...
check_escape=inject CheckFileSize ShowAll MaxWarn=1024M MaxCrit=4096M "File: foo \" WIN=c:\\WINDOWS\\\".*"
nrpe_cpu=inject checkCPU warn=80 crit=90 5 10 15
```

PNP4NAGIOS

PNP4NAGIOS ist eine Zusatzsoftware für NAGIOS bzw. ICINGA, die alle zurückgelieferten Performancedaten, zum einen in eine RRD Datenbanken speichert und zum zweiten das Ganze Grafisch darstellt

Ich beschreibe euch wie ich PNP4NAGIOS installiert haben, die ausführlichere Dokumentation findet Ihr aber hier → <http://docs.pnp4nagios.org/de/pnp-0.6/start>



Um die Daten Grafisch darstellen zu können, benötigen wir die RRD Tools →

<http://oss.oetiker.ch/rrdtool/>

sudo apt-get install rrdtool librrds-perl

Nach der Installation von den RRD Tools beginne ich auch schon mit PNP4NAGIOS.

Ich wechsel wieder in das Source Verzeichnis von Ubuntu

cd /usr/local/src

wget <http://sourceforge.net/projects/pnp4nagios/files/PNP-0.6/pnp4nagios-0.6.5.tar.gz/download>

(Die Version 0.6.6 ist Bereits Online Verfügbar, habe es aber nicht mehr testen können)

Nach dem Download extrahiere ich das Paket

tar xvfz pnp4nagios-0.6.5.tar.gz

und wechsel in das Verzeichnis von PNP

cd pnp4nagios-0.6.5

Nun wird mit `./configure` das System analysiert

`./configure --with-nagios-user=icinga --with-nagios-group=icinga`

Link zum Original → <http://docs.pnp4nagios.org/de/pnp-0.6/install>

```
*** Configuration summary for pnp4nagios-0.6.5 07-09-2010 ***

General Options:
-----
Nagios user/group:      icinga icinga
Install directory:     /usr/local/pnp4nagios
HTML Dir:              /usr/local/pnp4nagios/share
Config Dir:            /usr/local/pnp4nagios/etc
Location of rrdtool binary: /usr/bin/rrdtool Version 1.3.8
RRDs Perl Modules:     FOUND (Version 1.3008)
RRD Files stored in:   /usr/local/pnp4nagios/var/perfdata
process_perfdata.pl Logfile: /usr/local/pnp4nagios/var/perfdata.log
Perfdata files (NPCD) stored in: /usr/local/pnp4nagios/var/spool

Web Interface Options: -----
HTML URL:              http://localhost/pnp4nagios
Apache Config File:    /etc/apache2/conf.d/pnp4nagios.conf

Review the options above for accuracy.  If they look okay,
type 'make all' to compile.
```

`make all`

`make install`

`make install-webconf`

`make install-config`

`make install-init`

`/etc/init.d/apache2 restart`

Bevor Ihr den Modus konfiguriert, solltet Ihr bitte nachlesen welcher Modus für euch der beste ist

<http://docs.pnp4nagios.org/de/pnp-0.6/modes>

Bulk Mode mit NPCD konfigurieren

`cd /usr/local/icinga/etc`

`sudo nano icinga.cfg`

`process_performance_data=` von 0 auf 1 ändern

```
# defined below).  Read the HTML docs for more
# performance data.
# Values: 1 = process performance data, 0 =
process_performance_data=1

# HOST AND SERVICE PERFORMANCE DATA PROCESSING
# These commands are run after every host and
# service has been checked.  These commands are executed on
```


STRG + W fürs suchen

STRG + O + Enter fürs Speichern

STRG + X für schließen

Zusätzlich werden einige neue Parameter in der *icinga.cfg* benötigt.

```
#####
# PNP4NAGIOS konfiguration
#####
#
# Service Performance-Daten
#
service_perfdata_file=/usr/local/pnp4nagios/var/service-perfdata
service_perfdata_file_template=DATATYPE::SERVICEPERFDATA\tTIMET::$TIMET$\tHOSTNAME::$HOSTNAME$\tSERVICEDESC::$SERVICEDESC$\tSERVICEPERFDATA::$SERVICEPERFDATA$\tSERVICECHECKCOMMAND::$SERVICECHECKCOMMAND$\tHOSTSTATE::$HOSTSTATE$\tHOSTSTATETYPE::$HOSTSTATETYPE$\tSERVICESTATE::$SERVICESTATE$\tSERVICESTATETYPE::$SERVICESTATETYPE$
service_perfdata_file_mode=a
service_perfdata_file_processing_interval=15
service_perfdata_file_processing_command=process-service-perfdata-file
#
# Host Performance-Daten ab Nagios 3.x
#
host_perfdata_file=/usr/local/pnp4nagios/var/host-perfdata
host_perfdata_file_template=DATATYPE::HOSTPERFDATA\tTIMET::$TIMET$\tHOSTNAME::$HOSTNAME$\tHOSTPERFDATA::$HOSTPERFDATA$\tHOSTCHECKCOMMAND::$HOSTCHECKCOMMAND$\tHOSTSTATE::$HOSTSTATE$\tHOSTSTATETYPE::$HOSTSTATETYPE$
host_perfdata_file_mode=a
host_perfdata_file_processing_interval=15
host_perfdata_file_processing_command=process-host-perfdata-file
#####
```

Die verwendeten Kommandos müssen Icinga wiederum bekannt gegeben werden.

nano /usr/local/icinga/etc/[company]/command.cfg

```
#####
# PNP4NAGIOS konfiguration
#####

define command{
    command_name process-service-perfdata-file
    command_line /bin/mv /usr/local/pnp4nagios/var/service-perfdata /usr/local/pnp4nagios/var/spool/service-perfdata.$TIMET$
}

define command{
    command_name process-host-perfdata-file
    command_line /bin/mv /usr/local/pnp4nagios/var/host-perfdata /usr/local/pnp4nagios/var/spool/host-perfdata.$TIMET$
}
#####
```

Folgende Dateien werden von mir umbenannt

mv /usr/local/pnp4nagios/etc/npcd.cfg-sample /usr/local/pnp4nagios/etc/npcd.cfg


```
mv /usr/local/pnp4nagios/etc/process_perfdata.cfg-sample
/usr/local/pnp4nagios/etc/process_perfdata.cfg
```

```
mv /usr/local/pnp4nagios/etc/rra.cfg-sample /usr/local/pnp4nagios/etc/rra.cfg
```

Jetzt müssen wir nur noch den NPCD Deamon starten.

```
/usr/local/pnp4nagios/bin/npcd -d -f /usr/local/pnp4nagios/etc/npcd.cfg
```

Richten den Deamon so ein das er beim nächsten Neustart automatisch mitgestartet wird.

update-rc.d npcd defaults

Ubuntu Rechner durchstarten

```
nano /etc/apache2/conf.d/pnp4nagios.conf
```

Im Apache Verzeichnis wurde von PNP die **pnp4nagios.conf** angelegt, leider ist hier ein kleiner Bug drin denn wir ändern müssen, Der Eintrag **AuthUserFile** bezieht sich auf eine NAGIOS Installation und nicht auf ICINGA, wenn man nun die PNP4NAGIOS Seite im Webbrowser aufrufen würde müsste man zweimal das Passwort eingeben um die Performance Daten einsehen zu können.

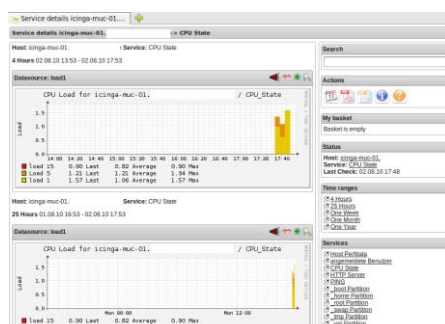
Hier ändere ich einfach den Pfad zur **htpasswd.users** Datei, statt **nagios** trage ich **icinga** ein.

```
#
# Use the same value as defined in nagios.conf
#
AuthName "Nagios Access"
AuthType Basic
AuthUserFile /usr/local/icinga/etc/htpasswd.users
Require valid-user
<IfModule mod_rewrite.c>
# Turn on URL rewriting
RewriteEngine On
Options FollowSymLinks
```

Wenn bis jetzt alles sauber funktioniert hat, kann PNP zum ersten Mal im Browser aufgerufen werden. Bei der Installation mit den Standardeinstellungen erfolgt der Aufruf über <http://Servername/pnp4nagios>

Beim ersten Aufruf sieht man die Seite **"PNP4Nagios Environment Tests"**, die verschiedene Tests von notwendigen Komponenten enthält. Offenkundig sollten alle Tests erfolgreich verlaufen, bevor es weitergehen kann. Bitte beachtet die Hinweise auf der Seite.

Sind alle Tests erfolgreich verlaufen, so kann die Datei **pnp4nagios/share/install.php** gelöscht oder umbenannt werden. Erst dann ist das Webinterface erreichbar.



Das PNP4NAGIOS in das alte ICINGA Web Frontend einbinden

Um nun die PNP4NAGIOS Graphen bereits in der ICINGA Weboberfläche nutzen zu können müssen vorab zwei Templates angelegt werden

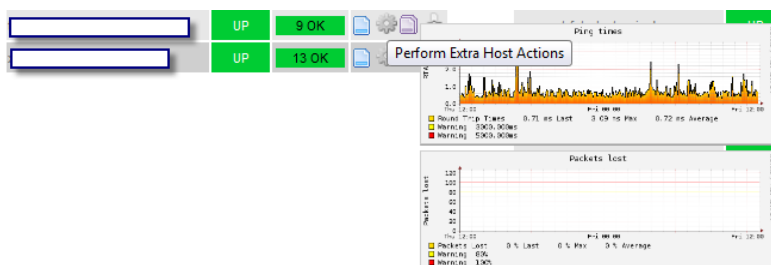
[templates.cfg]

```
#####
define host {
    name    host-pnp
    register 0
    action_url /pnp4nagios/index.php/graph?host=$HOSTNAME$&srv=_HOST_' class='tips'
    rel='/pnp4nagios/popup?host=$HOSTNAME$&srv=_HOST_'
}

define service {
    name    srv-pnp
    register 0
    action_url /pnp4nagios/index.php/graph?host=$HOSTNAME$&srv=$SERVICEDESC$' class='tips'
    rel='/pnp4nagios/popup?host=$HOSTNAME$&srv=$SERVICEDESC$'
}
#####
```

Außerdem gibt es die Möglichkeit, die Graphen bereits in der Statusübersicht beim Überfahren des **“Action Url Icons”** mit der Maus einzublenden.

Sieht wie folgt aus:



In den PNP-Quellen ist die Datei **contrib/ssi/status-header.ssi** bereits enthalten, die verwendeten URLs müssen aber unter Umständen angepasst werden. Wir gehen hier davon aus, dass PNP über **/pnp4nagios/index.php** erreichbar ist.

Die besagte Datei muss in das Verzeichnis **/usr/local/icinga/share/ssi/** kopiert werden und darf **NICHT** ausführbar sein. ICINGA würde die Datei sonst wirklich wie ein CGI behandeln und ausführen, was aber in diesem Fall zu Fehlern führen würde.

<http://docs.pnp4nagios.org/de/pnp-0.6/webfe>

Ab jetzt können alle Hosts und Services die Performance Daten zurückliefern mit folgenden Zusatz erweitert werden.

In der [Server.cfg] definiere ich für meine Windows Services und Host folgende Templates

Definition für Host und Service

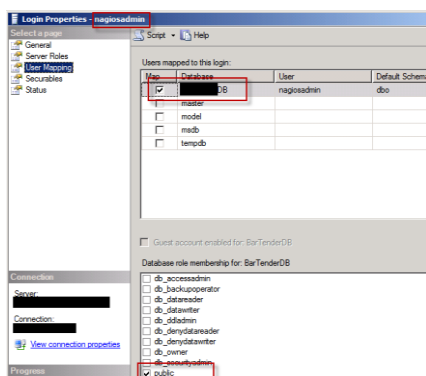
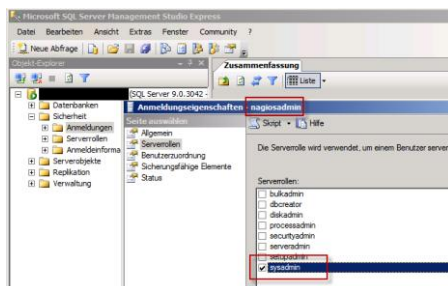
```
define host{
    use                windows-host, srv-pnp
    host_name          servername.domain.de
    .....
}
```

```
define service{
    use                windows-service, srv-pnp
    host_name          servername.domain.de
    .....
}
```

Microsoft SQL Server Express Check

Hier führe ich noch ein paar Beispiele auf wie ich manche Services in der Windows Umgebung Monitore.

Bei diesem Check muss natürlich vorab ein Benutzer für die Datenbank eingerichtet werden um den Check ausführen zu können. Bei der Serverrolle benötigt der Benutzer **sysadmin** Rechte und bei der Benutzerzuordnung müssen alle Datenbanken markiert werden plus **public** Rechte



Nun kann ich den ersten Test machen ob alles funktioniert

https://www.monitoringexchange.org/inventory/Check-Plugins/Database/SQLServer/check_mssql_monitor

Wir loggen uns in der Console wieder als icinga Benutzer ein

su – icinga (Passwort)

bash

```
/usr/local/icinga/libexec/check_mssql_monitor -H Servername\\Instanzname -U Benutzername -P Passwort -w 10 -c 20 --cpuWarning 200 --cpuCritical 400 --ioWarning 10 --ioCritical 20
```

Ergebnis

```
CHECK_MSSQL_MONITOR OK - SQL Server responded in 0.715336 seconds | time=0.715336s;10;20 cpu=0%;200;400 io=0%;10;20 idle=96%;;
```

[command.cfg]

SQL Express Server Check

```
define command{
    command_name    check_mssql_monitor
    command_line     $USER1$/check_mssql_monitor -H $HOSTADDRESS$\\Instanzname -U Username -P Passwort -w
$ARG1$ -c $ARG2$ --cpuWarning $ARG3$ --cpuCritical $ARG4$ --ioWarning $ARG5$ --ioCritical $ARG6$ -s
}
```

[Server.cfg]

SQL Express Monitor Check

```
define service{
    use                windows-service,srv-pnp
    host_name          servername.domain.de
    service_description SQL Express Monitor Check
    check_command       check_mssql_monitor!10!20!200!400!10!20
}
```

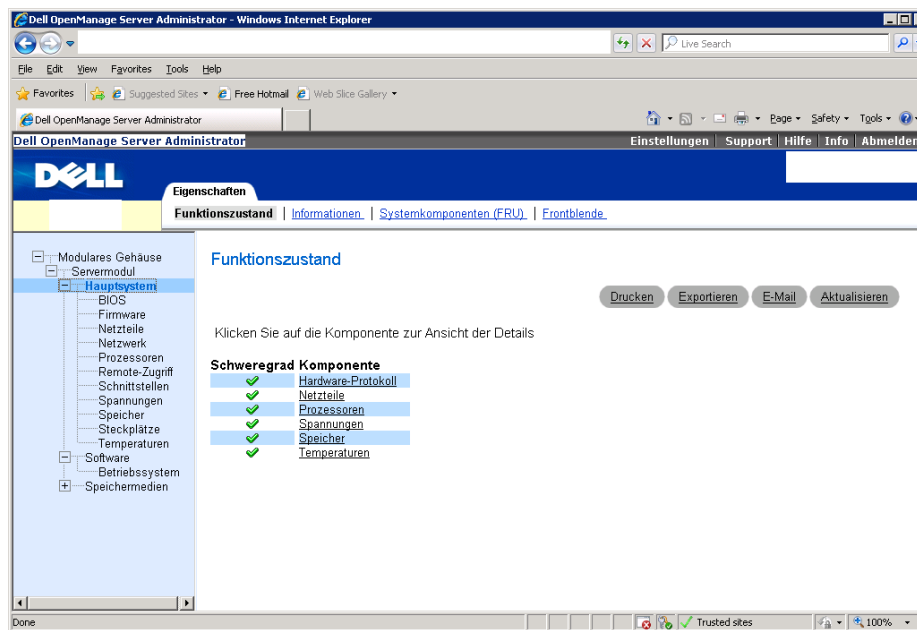
Dell Hardware Check

Um die Dell Hardware checken zu können benutzen ich das Plug-In **check_openmanage**

http://exchange.nagios.org/directory/Plugins/Hardware/Server-Hardware/Dell/check_openmanage/details

Voraussetzung für das Tool ist natürlich die Installation von dem → Dell OpenManage Server Administrator Managed Node auf den jeweiligen Servern.

http://support.dell.com/support/downloads/download.aspx?c=us&cs=555&l=en&s=biz&releaseid=R252582&SystemID=PWE_PNT_P3C_2850&servicetag=&os=WNET&osl=en&deviceid=2331&devlib=0&typecnt=0&vercnt=14&catid=&impid=&formatcnt=0&libid=0&typeid=-1&dateid=-1&formatid=-1&fileid=369525



Wir loggen uns in der Console wieder als icinga Benutzer ein

su – icinga (Passwort)

bash

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring -p -w 0=70,1=70,2=70,3=70 -c 0=80,1=80,2=80,3=80
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only storage
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only memory
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only fan
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only power
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only temp
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only cpu
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only voltage
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only bateries
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only amperage
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only intrusion
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only alertlog
```

```
/usr/local/icinga/libexec/check_openmanage -H Servername -C Communitystring --only esmhealth/esmlog
```

Ergebnis

```
OK - System: 'PowerEdge XXXX', SN: 'XXXXXXXX', hardware working fine, 1 logical drives, 2 physical
drives|temp_0_cpu#1_cpu1=38;120;125 temp_1_cpu#2_cpu2=39;120;125 temp_2_bmc_mem_temp_1=31;75;80
temp_3_bmc_mem_temp_2=31;75;80
```

```
[command.cfg]
```

```
# Dell CPU Check
define command{
    command_name    check_dell_cpu
    command_line    /usr/local/icinga/libexec/check_openmanage -H $HOSTADDRESS$ -C SNMP Password -p -w
0=70,1=70,2=70,3=70 -c 0=80,1=80,2=80,3=80
}
# Dell Memory Check
define command{
    command_name    check_dell_mem
    command_line    /usr/local/icinga/libexec/check_openmanage -H $HOSTADDRESS$ -C SNMP Password --only memory
}
# Dell Power Supplies Check
define command{
    command_name    check_dell_power
    command_line    /usr/local/icinga/libexec/check_openmanage -H $HOSTADDRESS$ -C SNMP Password --only power
}
# Dell Storage Check
define command{
    command_name    check_dell_storage
    command_line    /usr/local/icinga/libexec/check_openmanage -H $HOSTADDRESS$ -C SNMP Password --only storage
}
#####
```

```
[Server.cfg]
```

```
# Dell CPU Temp
define service{
    use                windows-service,srv-pnp
    host_name          Servername.Domain.de
    service_description _Dell CPU Temp
    check_command      check_dell_cpu
}
#####
# Dell Memory Check
define service{
    use                windows-service
    host_name          Servername.Domain.de
    service_description _Dell Memory Check
    check_command      check_dell_mem
}
#####
# Dell Power Supplies
define service{
    use                windows-service
    host_name          Servername.Domain.de
    service_description _Dell Power Supplies
    check_command      check_dell_power
}
#####
# Dell Storage Check
define service{
    use                windows-service
    host_name          Servername.Domain.de
    service_description _Dell Storage Check
    check_command      check_dell_storage
}
}
```

Cisco Hardware Check

Um die Cisco Hardware checken zu können benutzen ich folgendes Plug-In

check_interface_table_v2.pl → http://blog.netways.de/2009/06/08/check_interface_table_v2pl-alle-ports-auf-einen-schlag/

http://www.monitoringexchange.org/inventory/Check-Plugins/Network/check_interface_table_v2-pl

[command.cfg]

Cisco Port Check

define command {

command_name check_cisco_port

command_line \$USER1\$/check_interface_table_v2.pl -regex -H \$HOSTADDRESS\$ -C \$SNMP Passwort -w \$ARG1\$ -c \$ARG2\$ -include \$ARG3\$ -exclude \$ARG4\$ --StateDir '/usr/local/icinga/var/tmp/.ifState' --CacheDir '/usr/local/icinga/var/tmp/.ifCache' --HTMLDir '/usr/local/icinga/share/interfacetable/' --HTMLUrl 'http://servername/icinga/interfacetable/' -Human 1 -ifLoadGradient 1 -Error 1

}

[Server.cfg]

Check Interfaces

define service {

use cisco-service,srv-pnp

host_name switchname.domain.de



service_description Check Interfaces

check_command check_cisco_port!50!70!FastEthernet,GigabitEthernet,Tunnel!Null0,Loopback

register 1

}

Ergebnis:

Host:	(2 Items)					
<input type="checkbox"/> default	Check Interfaces		2010-08-30 11:42:17	1 / 5	WS-C37	152 interface(s), 72 /
<input type="checkbox"/> default	PING		2010-08-30 11:42:24	1 / 5	OK	rta 0.691ms, lost 0%

