



ICINGA 1.7.0 in einer Virtuellen Umgebung mit
UBUNTU 12.04 installieren

ICINGA 1.7.0 in einer Virtuellen Umgebung mit UBUNTU 12.04 installieren

Inhalt

Ubuntu/ICINGA Einleitung	2
Grundinstallation UBUNTU 12.04 Desktop	3
ICINGA Server in das Microsoft Active Directory einbinden	8
Ubuntu vorbereiten für die ICINGA Installation	16
Benutzer und Gruppen für ICINGA anlegen	18
ICINGA und die IDOUtils kompilieren und installieren.....	19
MySql Datenbank anlegen	21
ICINGA Webkomponenten installieren	22
Nagios Plug-Ins 1.4.15 installieren	23
ICINGA Konfiguration überprüfen	23
Icinga 1.7.0 Grundkonfiguration	25
Ordnerstruktur anlegen für das Firmenweite Monitoring	25
hostgroup.cfg	26
localhost.cfg	27
templates.cfg.....	27
icinga.cfg.....	29
timeperiods.cfg.....	30
icinga01.htdom.local.cfg	31
ads01.htdom.local	34
Windows Firewall und SNMP Dienst konfigurieren	38
commands.cfg	39
Zusatzkomponenten/Plugins & Perl Skripte für ICINGA einbinden.	39
SNMP auf den ICINGA Server aktivieren	40
NRPE installieren für Windows Abfragen.....	41
commands.cfg die zweite	43

Ubuntu/ICINGA Einleitung

Diese Anleitung ist dazu gedacht, dir eine einfache Anweisung zu liefern, wie die Software ICINGA 1.7.0 innerhalb von 60 Minuten aus dem Quellcode installiert und deinen lokalen Rechner überwacht.

Diese Anleitung enthält Anweisungen für eine einzige Linux-Distributionen: Ubuntu 12.04 LTS Desktop. Ähnliche Distributionen werden wahrscheinlich auch funktionieren, darunter Fedora, OpenSuse, RedHat, CentOS, Debian und SLES. Leider habe ich persönlich nicht die Zeit alle Distributionen zu testen bzw. zu Dokumentieren.

Was dabei herauskommt:

Wenn du dieser Anweisungen folgst, wirst du am Schluss ein lauffähiges ICINGA 1.7.0 System haben:

- Icinga und die Nagios Plug-Ins werden unterhalb von `/usr/local/icinga` installiert sein
- Icinga wird so konfiguriert sein, dass es einige Dinge auf deinem lokalen System bzw. entfernten Windows System überwacht (CPU-Auslastung, Plattenbelegung, usw.)
- Das Icinga-Web-Interface ist erreichbar unter <http://<servername>/icinga/>
- Eine Datenbank, die von Icinga mit Hilfe von IDOUtils gefüllt wird
- Der ICINGA Server wird in einer Microsoft Active Directory integriert

Voraussetzungen:

Bei einigen Teilen der Installation benötigst du root-Zugang. IDOUtils benutzt die libdbi und die libdbi-Treiber für verschiedene Datenbanken. Die Development-Libraries werden ebenfalls benötigt.

Auf los geht's los.

Meine Testumgebung wurde realisiert mit einem Microsoft Hyper-V 2.0 Server. In dieser Virtuellen Umgebung habe ich zwei Active Directory Server (2008 R2/Server8), SharePoint 2010 Server, Exchange 2010 Server und natürlich einen ICINGA Server installiert.

Also viel Spaß beim Lesen und nachmachen.

Referenzen für dieses Howto waren wie immer:

NAGIOS Portal → <http://www.nagios-portal.org> und die Icinga Webseite → <http://www.icinga.org/>

Weitere Howto's zu ICINGA/NAGIOS findest du auf meiner Webseite <http://www.thurnhofer.net>

Grundinstallation UBUNTU 12.04 Desktop

Von folgender Webseite bekommt man das ISO Image für UBUNTU 12.04 Desktop x86/x64

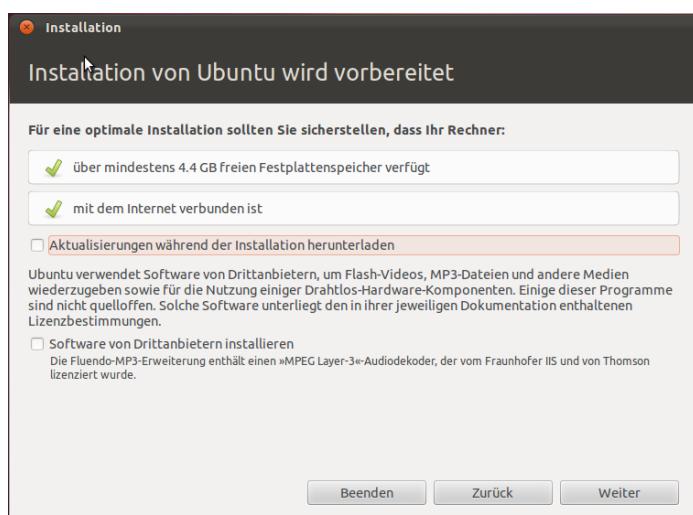
X86 → <http://releases.ubuntu.com/12.04/ubuntu-12.04-desktop-i386.iso>

X64 → <http://releases.ubuntu.com/12.04/ubuntu-12.04-desktop-amd64.iso>

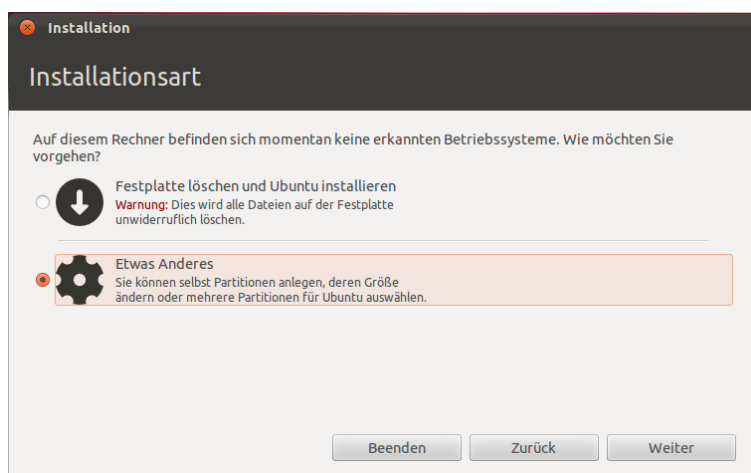
In Oracle VM VirtualBox lege ich eine neue Virtuelle Maschine an und binde das ISO Image ein, danach starte ich die Virtuelle Maschine und beginne die Installation von UBUNTU 12.04 Desktop.



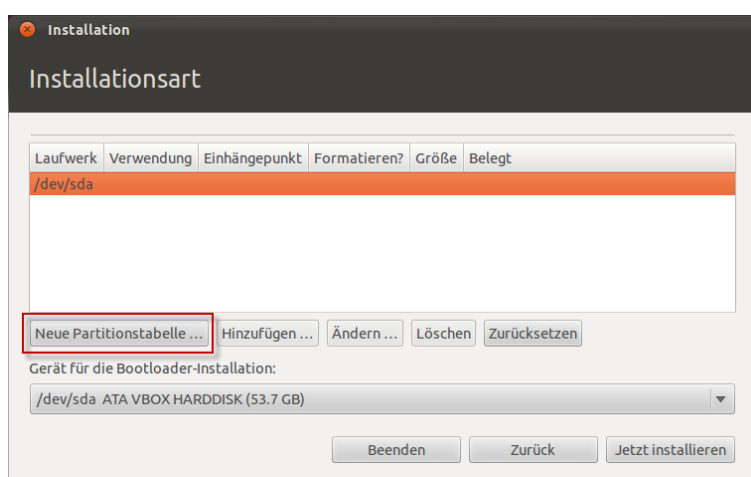
Wähle Deutsch aus und drück die auf Ubuntu installieren



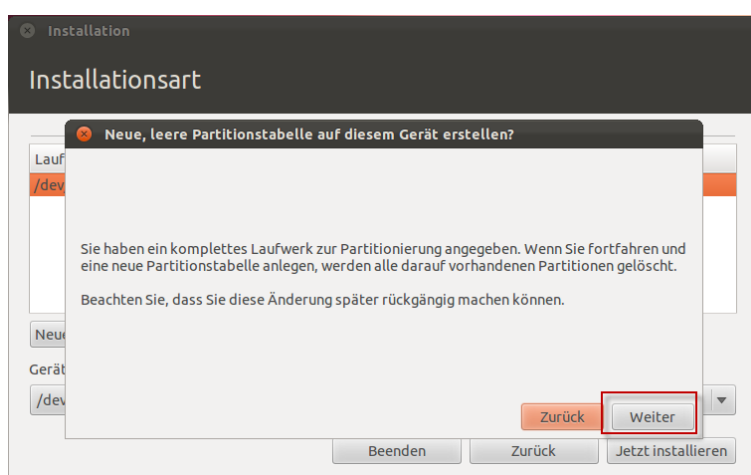
Im nächsten Fenster klicke ich auf Weiter



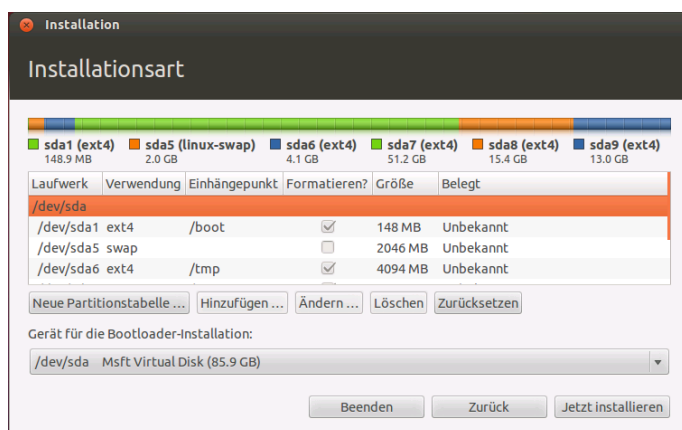
Bei der Partitionierung der Festplatte wähle ich „**Etwas Anderes**“ und klicke auf Weiter



Im nächsten Fenster wähle ich **Neue Partitionstabelle...**



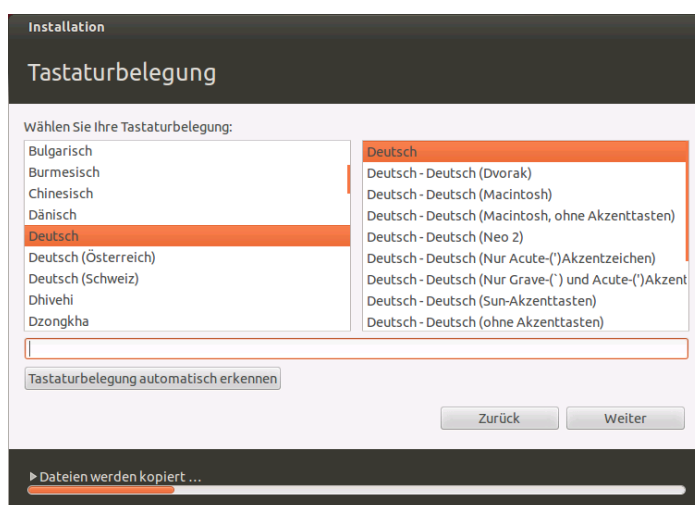
Bestätige das folgende Fenster mit Weiter und richte die Partitionen ein.



Da ich eine Virtuelle Maschine habe reichen mir 80GB als Gesamtgröße, die Gesamtgröße teile ich dementsprechend auf. **150MB /boot, 2GB /swap, 4GB /tmp, 50GB /root, 15GB /var, Rest /home**



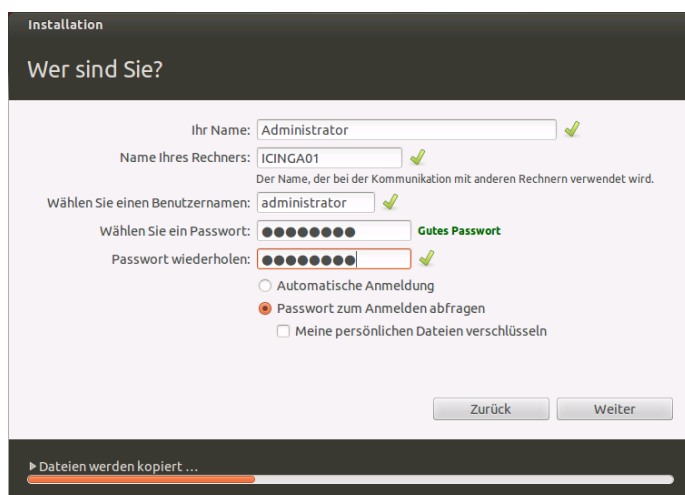
Wähle meinen Standort aus und klicke auf Weiter



Wähle meine Tastaturbelegung aus und klicke auf Weiter

Hinweis: Hyper-V Maschinen die über das Tool **vmconnect.exe** geöffnet werden, haben Probleme mit der Alt-Gr Taste oder Strg+Alt, das Pipe Zeichen, Backslash-Zeichen, @-Zeichen, €-Zeichen zu tippen, diese funktionieren schlicht weg nicht, egal welches Tastaturlayout man einstellt. Hier habe ich mir einen Wolf gesucht um eine Lösung herbeizuführen, leider ohne wirklichen Erfolg.

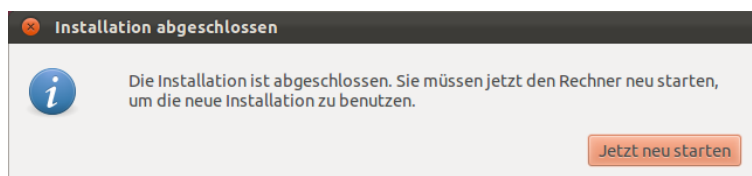
Also habe ich mir einen Ultra VNC Viewer lokal installiert, und später wenn der Server/Client installiert wurde den Remotezugriff erlaubt. Und siehe da, hier funktionieren alle Tasten so wie es sein soll. 😊



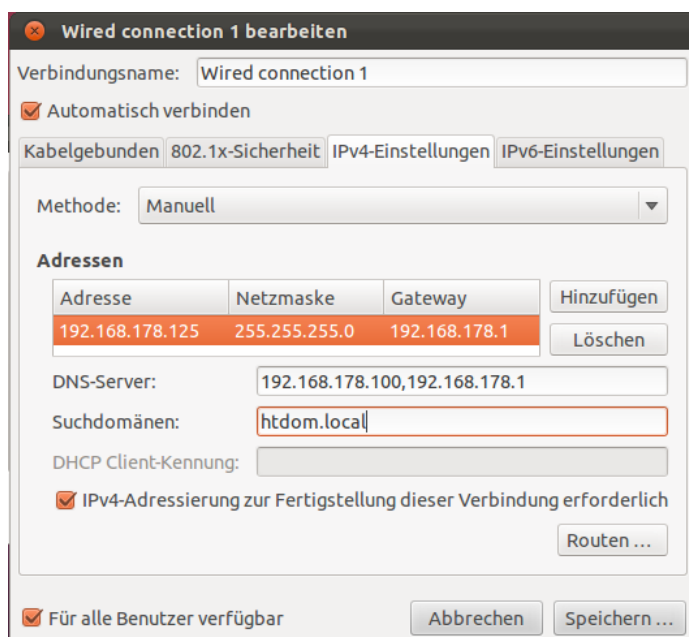
Vergebe einen Computernamen und Benutzernamen mit Passwort und klicke auf Weiter.



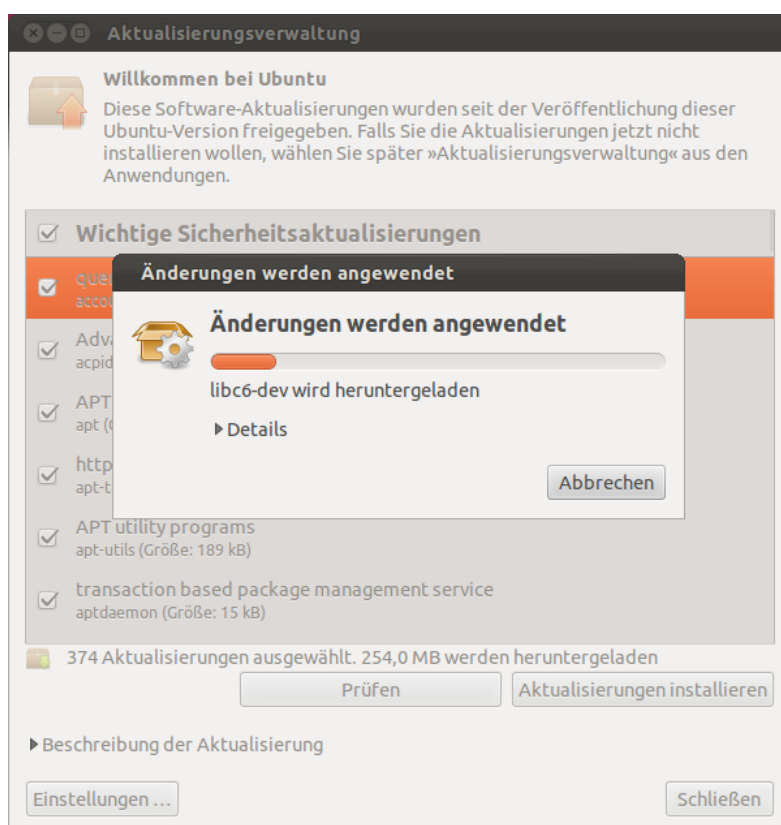
Die Installation von Ubuntu 12.04 beginnt.



Nach der Installation machen wir den obligatorischen Neustart.



Vergeben nach dem ersten Login eine Statische IP-Adresse.



Und installiere alle Updates und Sparchpakete von Ubuntu 12.04 LTS Desktop.

ICINGA Server in das Microsoft Active Directory einbinden

Um sicher zu gehen das der Domaincontroller richtig aufgelöst wird, frage ich den DNS Server mit der Forward und Reverse Lookup Adresse ab.

host ads01

host 192.168.178.100

ping htdom.local

nslookup !

```

administrator@ICINGA02: ~
administrator@ICINGA02:~$ host ads01
ads01.htdom.local has address 192.168.178.100
administrator@ICINGA02:~$ host 192.168.178.100
100.178.168.192.in-addr.arpa domain name pointer ads01.htdom.local.
administrator@ICINGA02:~$ ping htdom.local
ping: unknown host htdom.local
administrator@ICINGA02:~$ nslookup !
nslookup htdom.local
Server:          192.168.178.100
Address:         192.168.178.100#53

Name:   htdom.local
Address: 192.168.178.100

administrator@ICINGA02:~$ sudo nano /etc/hosts
[sudo] password for administrator:
administrator@ICINGA02:~$ ping htdom.local
PING ads01.htdom.local (192.168.178.100) 56(84) bytes of data:
64 bytes from ads01.htdom.local (192.168.178.100): icmp_req=1 ttl=128 time=0.638 ms
64 bytes from ads01.htdom.local (192.168.178.100): icmp_req=2 ttl=128 time=0.268 ms
^C
--- ads01.htdom.local ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.268/0.453/0.638/0.185 ms
administrator@ICINGA02:~$

```

Wenn das sauber funktioniert dann starte ich die Installation von Likewise, wenn **nicht** dann muss man noch kurz die **/etc/hosts** Datei anpassen, damit der ping auf die Domäne funktioniert.

```

administrator@ICINGA02: ~
GNU nano 2.2.6                                Datei: /etc/hosts
127.0.0.1    localhost
192.168.178.125 icinga02.htdom.local icinga02
192.168.178.100 ads01.htdom.local ads01 htdom.local

```

```

root@ICINGA02: ~
GNU nano 2.2.6                               Datei: /etc/nsswitch.conf

# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:         compat lsass
group:          compat lsass
shadow:         compat

hosts:          files dns mdns4
#hosts:         files mdns4_minimal [NOTFOUND=return] dns mdns4
networks:       files

```

Da ich eine *.local Domain für die Testumgebung nutze, muss ich noch die **/etc/nsswitch.conf** Datei anpassen. Die ursprüngliche Zeile **hosts:** kommentiere ich aus und schreibe eine neue Zeile.

nano /etc/nsswitch.conf

hosts: files dns mdns4

Wenn das alles erledigt ist beginne ich die Installation.

sudo apt-get install likewise-open likewise-open-gui

Nach der Installation, nehme ich den ICINGA Server in der Domäne auf.

sudo domainjoin-cli join --ou ComputersUnix htdom.local Administrator

```

root@ICINGA02: ~
administrator@ICINGA02:~$ sudo -s
[sudo] password for administrator:
root@ICINGA02:~# domainjoin-cli join --ou ComputersUnix htdom.local Administrator
Joining to AD Domain: htdom.local
With Computer DNS Name: icinga02.htdom.local

Administrator@HTDOM.LOCAL's password:

Error: Undocumented exception [code 0x0009efc]

An undocumented exception has occurred. Please contact Likewise technical support and use the error code to identify this exception.
root@ICINGA02:~#

```

20 mal Probiert und es ist nix passiert. ☹ Die Integration bringt mir jetzt einen Undokumentierten Errorcode. Trotzdem funktionierte die Anlage vom Computerkonto im AD. Mal sehen welche Auswirkungen der Fehler noch hat.

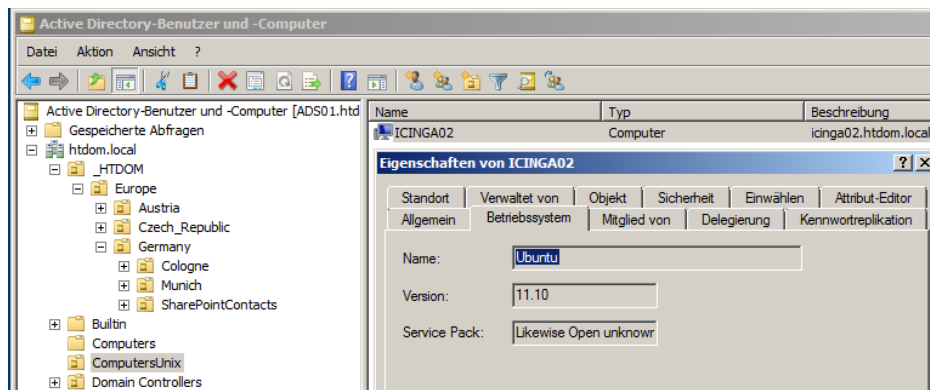
```

administrator@icinga01: ~
root@icinga01:/# gedit /etc/resolv.conf
root@icinga01:/# domainjoin-cli join htdom.local Administrator
Joining to AD Domain: htdom.local
With Computer DNS Name: icinga01.htdom.local

Administrator@HTDOM.LOCAL's password:
SUCCESS

```

Ich habe die Integration mit mehreren Ubuntu Versionen (10.04, 10.10, 11.04, 12.04) getestet und immer eine Success Meldung bekommen, bei der jetzigen 11.10 tritt dieser Error Code auf.



Nach dem ersten reboot nach der Integration, teste ich den Domain Join In mit dem Befehl

sudo lw-get-status

```

root@ICINGA02: ~
administrator@ICINGA02:~$ sudo -s
[sudo] password for administrator:
root@ICINGA02:~# lw-get-status
LSA Server Status:

Compiled daemon version: 5.0.0.0
Packaged product version: 0.0.0.0
Uptime:          0 days 0 hours 10 minutes 54 seconds

[Authentication provider: lsa-activedirectory-provider]

Status:      Online
Mode:        Un-provisioned
Domain:      HTDOM.LOCAL
Forest:      htdom.local
Site:        Site-Germany-Munich
Online check interval: 300 seconds
[Trusted Domains: 1]

[Domain: HTDOM]

DNS Domain:      htdom.local
Netbios name:    HTDOM
Forest name:     htdom.local
Trustee DNS name:
Client site name: Site-Germany-Munich
Domain SID:      S-1-5-21-1614496689-3584411676-4085633081
Domain GUID:     7cb766ab-1e78-d441-a074-9a6c0dc64b36
Trust Flags:     [0x001d]
  
```

Die Benutzerinformationen der Domäne kann ich noch mit folgenden Befehl abrufen

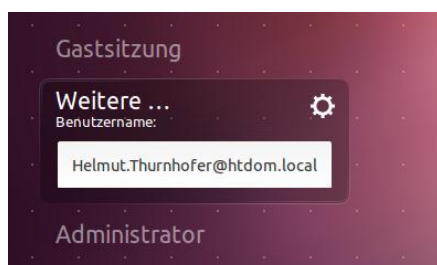
sudo lw-enum-users

```

root@ICINGA02: ~
root@ICINGA02:~# lw-enum-users
User info (Level-0):
=====
Name:      administrator
Uid:       41943540
Gid:       41943553
Gecos:     Administrator
Shell:     /bin/bash
Home dir:  /home/likewise-open/HTDOM/administrator

User info (Level-0):
=====
Name:      gast
Uid:       41943541
Gid:       41943554
Gecos:     <null>
Shell:     /bin/bash
Home dir:  /home/likewise-open/HTDOM/gast
  
```

Im nächsten Schritt loge ich mich auf den ICINGA Server mit einem Domänen Benutzer ein.



```
helmut.thurnhofer@ICINGA02: ~
helmut.thurnhofer@ICINGA02:~$ ls -la
insgesamt 144
drwxr-xr-x 19 helmut.thurnhofer domänen-benutzer 4096 2012-04-12 11:46 .
drwxr-xr-x  4 root                root        4096 2012-04-12 11:44 ..
drwxr-xr-x  2 helmut.thurnhofer domänen-benutzer 4096 2012-04-12 11:44 Arbeitsf
läche
-rw-----  1 helmut.thurnhofer domänen-benutzer 127 2012-04-12 11:46 .bash_hi
story
-rw-r--r--  1 helmut.thurnhofer domänen-benutzer 220 2012-04-12 11:44 .bash_lo
gout
```

Auch das scheint sauber zu funktionieren.

„Optional habe ich auch folgendes Script bei meinen vorigen Tests benutzt und es hat ebenfalls funktioniert.“

Auf folgender Webseite → <http://www.beyondtrust.com/Technical-Support/Downloads/PowerBroker-Identity-Services-Open-Edition> bekommt man die Software/Script „**pbis-open-7.0.0.867.linux.x86.deb.sh**“ Nach einer kurzen Registrierung bekommt man einen E-Mail zugesandt und kann im Anschluss das Script für die Ubuntu Umgebung downloaden.

Hier die Dokumentation die ich verwendet habe → [http://www.beyondtrust.com/Technical-Support/Downloads/files/PBISO/Manuals/PBISO Installation and Administration Guide V6.5/Default.htm](http://www.beyondtrust.com/Technical-Support/Downloads/files/PBISO/Manuals/PBISO%20Installation%20and%20Administration%20Guide%20V6.5/Default.htm)



PBIS Open 7.0.0.867 - Stable Release
Likewise Open integrates Linux, UNIX, and Mac computers in Microsoft Active Directory.

Download the PowerBroker Identity Services, Open Edition installer for your type of computer and operating system from the list and then make it executable with `chmod a+x`. Or, right-click the installer, click **Properties**, click the **Permissions** tab, select **Execute for Owner**, and then click **Close**. Then double-click the installer to run it.

After installing PowerBroker Identity Services, Open Edition, run the following command as root. Your AD account must have privileges to join computers to the domain:
Linux: `/opt/pbis/bin/domainjoin-cli join domainName ADjoinAccount`

Log on with your AD credentials by using the form `DOMAIN\username`. Example: `likewisedemo.com\hoenstiv`. On the command line, you must include an escape character: `DOMAIN\\username`.

PBIS Open 7.0 Linux Download Packages

Version by Architecture	Download Package	Application Integration	Typical Platforms
Linux 2.4/2.6 kernel 32-bit RPM	DOWNLOAD Stk	COMING SOON	
Linux 2.4/2.6 kernel 64-bit RPM	DOWNLOAD Stk	COMING SOON	
Linux 2.4/2.6 kernel 32-bit DEB	DOWNLOAD Stk	COMING SOON	
Linux 2.4/2.6 kernel 64-bit DEB	DOWNLOAD Stk	COMING SOON	

Nach dem Download, setzt man die Rechte für das Script und führt es im Anschluss mit Root-Rechten aus.

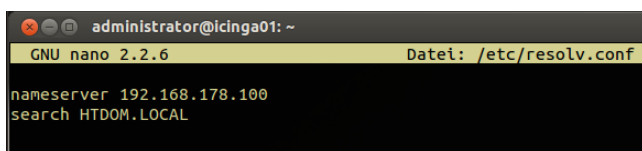
```
chmod a+x pbis-open-7.0.0.867.linux.x86.deb.sh  
./pbis-open-7.0.0.867.linux.x86.deb.sh
```

Wenn das Script durchgelaufen ist, passt man noch die **resolv.conf** Datei im etc Verzeichnis an um den Richtigen DNS Server der Domäne anzusprechen. **Wenn das vergessen wird funktioniert der Join In nicht! → Bis jetzt nur unter der Version 12.04 Beta nötig gewesen**

```
sudo nano /etc/resolv.conf
```

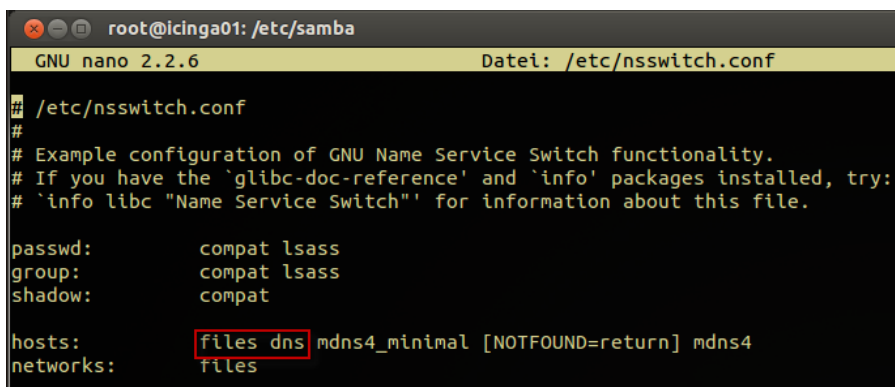
„Mir ist aufgefallen, dass nach jeden Neustart die **resolv.conf** vom System überschrieben wird. Um dieses zu vermeiden, kann man die Datei nach einer Änderung mit folgenden Befehl schützen“

Sperren → **chattr +i /etc/resolv.conf**
Entsperren → **chattr -i /etc/resolv.conf**



```
administrator@icinga01: ~  
GNU nano 2.2.6      Datei: /etc/resolv.conf  
nameserver 192.168.178.100  
search HTDOM.LOCAL
```

Auf was auch noch zu achten ist wäre die **/etc/nsswitch.conf** Datei, dort muss ein Eintrag verändert werden.



```
root@icinga01: /etc/samba  
GNU nano 2.2.6      Datei: /etc/nsswitch.conf  
## /etc/nsswitch.conf  
#  
# Example configuration of GNU Name Service Switch functionality.  
# If you have the 'glibc-doc-reference' and 'info' packages installed, try:  
# 'info libc "Name Service Switch"' for information about this file.  
  
passwd:          compat lsass  
group:           compat lsass  
shadow:          compat  
  
hosts:           files dns mdns4_minimal [NOTFOUND=return] mdns4  
networks:        files
```

Unter **hosts:** muss der **dns** Eintrag nach **files** kommen, die ist nötig wenn man eine *.local Domänenendung hat. **(Musste ich auch nachlesen weil der Domain Join In am Anfang nicht funktioniert hat.)**

Um sicher zu gehen das der Domaincontroller richtig aufgelöst wird, frage ich den DNS Server mit der Forward und Reverse Lookup Adresse ab.

```
host ads01  
host 192.168.178.100
```



```

administrator@icinga01: ~
administrator@icinga01:~$ host ads01
ads01.HTDOM.LOCAL has address 192.168.178.100
administrator@icinga01:~$ host 192.168.178.100
100.178.168.192.in-addr.arpa domain name pointer ads01.htdom.local.
administrator@icinga01:~$

```

Wenn das sauber funktioniert dann starte ich den Join in in die Domäne mit folgenden Befehl.

domainjoin-cli join htdom.local Administrator

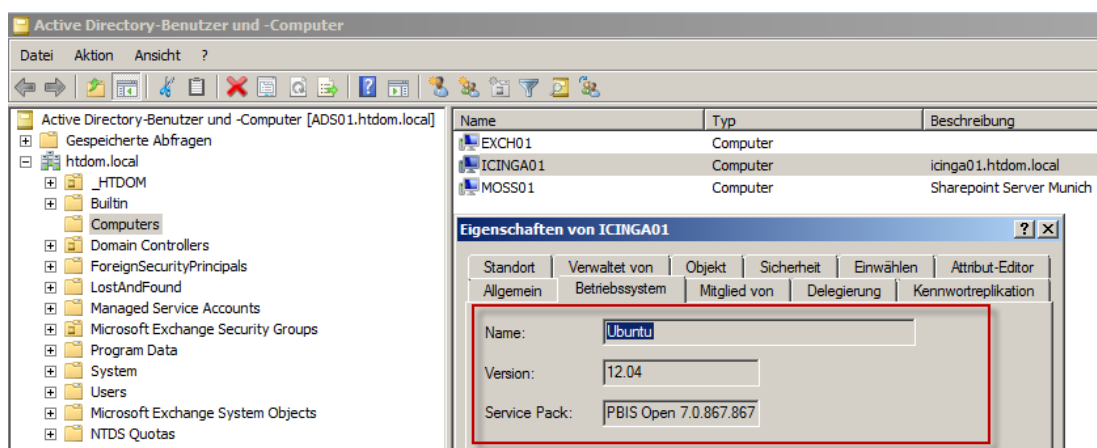
Passwort nicht vergessen 😊

```

root@icinga01:~# gedit /etc/resolv.conf
root@icinga01:~# domainjoin-cli join htdom.local Administrator
Joining to AD Domain: htdom.local
With Computer DNS Name: icinga01.htdom.local

Administrator@HTDOM.LOCAL's password:
SUCCESS

```



Danach sollte der Join In funktioniert haben, wenn dem nicht so ist, dann passt irgendwas mit der Namensauflösung nicht.

Man kann noch bei der Fehlersuche das Loglevel anschalten:

sudo domainjoin-cli --loglevel verbose join htdom.local Administrator
gedit /var/log/auth.log

Mit folgenden Befehlen kann man überprüfen ob alles funktioniert hat

sudo /opt/pbis/bin/get-status

```

root@icinga01: ~
root@icinga01:~# /opt/pbis/bin/get-status
LSA Server Status:
Compiled daemon version: 7.0.0.867
Packaged product version: 7.0.867.63997
Uptime: 0 days 0 hours 15 minutes 10 seconds
[Authentication provider: lsa-activedirectory-provider]
[Domain: HTDOM]
Status: Online
Mode: Un-provisioned
Domain: HTDOM.LOCAL
Domain SID: S-1-5-21-1614496689-3584411676-4085633081
Forest: htdom.local
Site: Site-Germany-Munich
Online check interval: 300 seconds
[Trusted Domains: 1]

[Domain: HTDOM]
DNS Domain: htdom.local
Netbios name: HTDOM
Forest name: htdom.local
Trustee DNS name: Site-Germany-Munich
Client site name: Site-Germany-Munich
Domain SID: S-1-5-21-1614496689-3584411676-4085633081
Domain GUID: 7cb766ab-1e78-d441-a074-9a6c9dc64b36
Trust Flags: [0x001d]
[0x0001 - In forest]
[0x0004 - Tree root]
[0x0008 - Primary]
[0x0010 - Native]
Trust type: Up Level
Trust Attributes: [0x0000]
Trust Direction: Primary Domain
Trust Mode: In my Forest Trust (MFT)
Domain Flags: [0x0001]
[0x0001 - Primary]

[Domain Controller (DC) Information]
DC Name: ADS01.htdom.local
DC Address: 192.168.178.100
DC Site: Site-Germany-Munich
DC Flags: [0x000031rd]
DC Is PDC: yes
DC Is time server: yes
DC has writeable DS: yes
DC Is Global Catalog: yes

```

sudo domainjoin-cli query

```

root@icinga01: ~
root@icinga01:~# domainjoin-cli query
Name = icinga01
Domain = HTDOM.LOCAL
Distinguished Name = CN=ICINGA01,CN=Computers,DC=htdom,DC=local
root@icinga01:~#
user-by-name htdom\Administrator

```

sudo /opt/pbis/bin/get-dc-name htdom.local

```

root@icinga01: ~
root@icinga01:~# /opt/pbis/bin/get-dc-name htdom.local
Printing LWNED_DC_INFO fields:
=====
dwDomainControllerAddressType = 23
dwFlags = 12797
dwVersion = 5 htdom\Administrator
wLToken = 65535
wNTToken = 65535
pszDomainControllerName = ADS01.htdom.local
pszDomainControllerAddress = 192.168.178.100
pucDomainGUID(hex) = 7C B7 66 AB 1E 78 D4 41 A0 74 9A 6C 0D C6 4B 36
pszNetBIOSDomainName = HTDOM
pszFullyQualifiedDomainName = htdom.local
pszDnsForestName = htdom.local
pszDCSiteName = Site-Germany-Munich
pszClientSiteName = Site-Germany-Munich
pszNetBIOSHostName = ADS01
pszUserName = <EMPTY>
root@icinga01:~#

```

sudo /opt/pbis/bin/find-user-by-name htdom\Administrator ***id htdom\Administrator***

```

root@icinga01: ~
root@icinga01:~# /opt/pbis/bin/find-user-by-name htdom\Administrator
User info (Level-0):
=====
Name: ht-dom.local HTDOM\administrator
SID: S-1-5-21-1614496689-3584411676-4085633081-500
Uid: 41943540
Gid: 41943553
Gecos: Administrator
Shell: /bin/sh
Home dir: /home/local/HTDOM/administrator
Logon restriction: NO
root@icinga01:~# id htdom\Administrator
uid=41943540(HTDOM\administrator) gid=41943553(HTDOM\domänen-benutzer) Gruppen=41943553(HTDOM\domänen-benutzer),41943612(HTDOM\abgelehnte^rodc-kennwortreplikationsgruppe),41943552(HTDOM\domänen-admins),41943560(HTDOM\richtlinien-ersteller-besitzer),41943558(HTDOM\schema-admins),41943559(HTDOM\organisations-admins),41944167(HTDOM\organization^management)
root@icinga01:~#

```

Mit folgenden Befehl kann man sich die laufenden Services ansehen

sudo /opt/pbis/bin/lwsm list

Und mit folgenden Befehl kann man die Services durchstarten

sudo /opt/pbis/bin/lwsm restart lwio

```

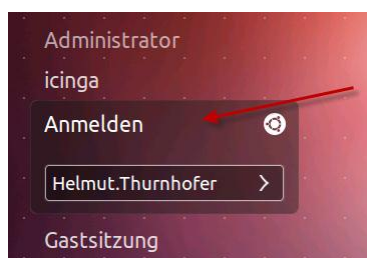
root@icinga01: ~
root@icinga01:~# /opt/pbis/bin/lwsm list
lwreg          running (container: 853)
dcerpc         running (container: 3000)
eventloghtdom running (container: 3037)
lsass          running (container: 3588)
lwio -by-name  running (container: 3551)
netlogon       running (container: 2152)
rdr            running (to: 3551)
reapsysl       running (container: 3566)
usermonitor    running (container: 3618)
root@icinga01:~# /opt/pbis/bin/lwsm restart lwio
Stopping service reverse dependency: usermonitor
Stopping service reverse dependency: lsass
Stopping service reverse dependency: rdr
Stopping service reverse dependency: reapsysl
Stopping service: lwio
Starting service: lwio
Starting service reverse dependency: reapsysl
Starting service reverse dependency: rdr
Starting service reverse dependency: lsass
Starting service reverse dependency: usermonitor
root@icinga01:~#

```

Hinweis: Unter Ubuntu 12.04 fehlt bei der Anmeldung das Feld „Weitere oder other User“ für die Auswahl der LDAP Anmeldung. Hier habe ich folgenden Trick im Internet gefunden um den Fehler zu beheben. → <http://www.linuxreaders.com/2012/05/10/ubuntu-12-04-enable-other-user-login-root/#.T7zZfsXRSkw>

sudo echo "greeter-show-manual-login=true" >> /etc/lightdm/lightdm.conf

Nach der Befehlseingabe den Server herunterfahren und ausschalten, wenn man jetzt den Server wieder startet, sollte ein Neuer Eintrag beim Login erscheinen.



Ubuntu vorbereiten für die ICINGA Installation

Benötigte Software installieren (Apache Server, GCC-Compiler, C/C++ Development Libraries, GD-Deploment Libraries, MySql Server, Perl)

sudo -s (Passwort)

apt-get install apache2 build-essential libgd2-xpm-dev

apt-get install mysql-server mysql-client libdbi-dev libdbd-mysql

apt-get install libconfig-inifiles-perl

MySql Paket verlangt bei der Installation ein Passwort.



(PHP 5, SNMP, Open SSL, PHPMyAdmin)

apt-get install php5 php5-mysql

apt-get install snmp snmpd libnet-snmp-perl

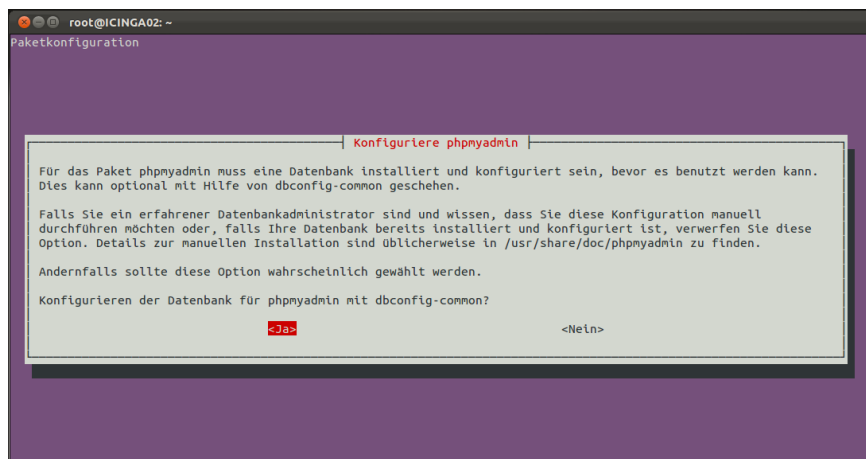
apt-get install openssl libcurl4-openssl-dev

Optional für die Grafische MySql Datenbankverwaltung: ***apt-get install phpmyadmin***

PHPMYAdmin Paket benötigt ebenfalls noch ein paar Konfigurationsangaben



Hier wähle ich den Apache Server aus und klick im Anschluss auf OK



Lass mir vom System die Datenbank einrichten



Und vergebe meine Passwörter.

Benutzer und Gruppen für ICINGA anlegen

sudo -s

/usr/sbin/useradd -m icinga
passwd icinga (2x Passwort eingeben)

Wenn noch nicht vorhanden dann die icinga Gruppe anlegen

/usr/sbin/groupadd icinga

Damit du über das Webinterface Befehle an Icinga senden kannst, legst du noch eine neue Gruppe ***icinga-cmd*** an und fügst den Webbenutzer und den Icinga Benutzer dieser Gruppe hinzu.

/usr/sbin/groupadd icinga-cmd
/usr/sbin/usermod -a -G icinga-cmd icinga
/usr/sbin/usermod -a -G icinga-cmd www-data

```

root@ICINGA01: ~
administrator@ICINGA01:~$ sudo -s
[sudo] password for administrator:
root@ICINGA01:~# /usr/sbin/useradd -m icinga
root@ICINGA01:~# passwd icinga
Geben Sie ein neues UNIX-Passwort ein:
Geben Sie das neue UNIX-Passwort erneut ein:
passwd: Passwort erfolgreich geändert
root@ICINGA01:~# /usr/sbin/groupadd icinga
groupadd: Gruppe »icinga« existiert bereits.
root@ICINGA01:~# /usr/sbin/groupadd icinga-cmd
root@ICINGA01:~# /usr/sbin/usermod -a -G icinga-cmd icinga
root@ICINGA01:~# /usr/sbin/usermod -a -G icinga-cmd www-data
root@ICINGA01:~#
  
```

ICINGA und die IDOUtils kompilieren und installieren

Wechsel in das lokale Source-Verzeichnis von Ubuntu

```
cd /usr/local/src
```

Download von den ICINGA 1.7.0 Quelldateien

ICINGA Quelldateien Entpacken

In das ICINGA Verzeichnis wechseln

```
wget http://sourceforge.net/projects/icinga/files/icinga/1.7.0/icinga-1.7.0.tar.gz
```

```
tar xvzf icinga-1.7.0.tar.gz
```

```
cd icinga-1.7.0
```

Führe das Icinga-configure-Script aus. Durch die Nutzung des --help-Flags erhältst du Hilfe zu den Optionen.

```
./configure --with-command-group=icinga-cmd --enable-idoutils
```

oder mit SSL

```
./configure --with-command-group=icinga-cmd --enable-idoutils --enable-ssl
```

```
root@ICINGA01: /usr/local/src/icinga-1.7.0
*** Configuration summary for icinga-core 1.7.0 05-15-2012 ***:
-----
General Options:
-----
Icinga executable:      icinga
Icinga user/group:      icinga,icinga
Command user/group:     icinga,icinga-cmd
Apache user/group:      www-data,www-data
Default Embedded Perl:  no
Event Broker:           yes
ido2db lockfile:        ${prefix}/var/ido2db.lock
ido sockfile:           ${prefix}/var/ido.sock
idomod tempfile:        ${prefix}/var/idomod.tmp
Build IDOUtils:         libdbi, instance_name=default
Install ${prefix}:      /usr/local/icinga
Lock file:              ${prefix}/var/icinga.lock
Temp file:              /tmp/icinga.tmp
Chk file:               ${prefix}/var/icinga.chk
HTTP auth file:         ${prefix}/etc/htpasswd.users
Lib directory:          ${exec_prefix}/lib
Bin directory:          ${exec_prefix}/bin
Plugin directory:       ${exec_prefix}/libexec
Eventhandler directory: ${exec_prefix}/libexec/eventhandlers
Log directory:          ${prefix}/var
Check result directory:  ${prefix}/var/spool/checkresults
Temp directory:         /tmp
State directory:        ${prefix}/var
Ext Cmd file directory:  ${prefix}/var/rw
Init directory:         /etc/init.d
Apache conf.d directory: /etc/apache2/conf.d
Mail program:           /bin/mail
Host OS:                linux-gnu
Environment Prefix:     ICINGA_

Web Interface Options:
-----
HTML URL:      http://localhost/icinga/
CGI URL:       http://localhost/icinga/cgi-bin/
Traceroute (used by WAP):

Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.

!!! Please take care about the upgrade documentation !!!
```

Jetzt wird der ICINGA-Source-Code kompiliert mit

```
make all
```

Installieren bitte die Binaries, das Init-Script, Beispiel-Konfigurationsdateien und setze die Berechtigungen für das External-Command-Verzeichnis.

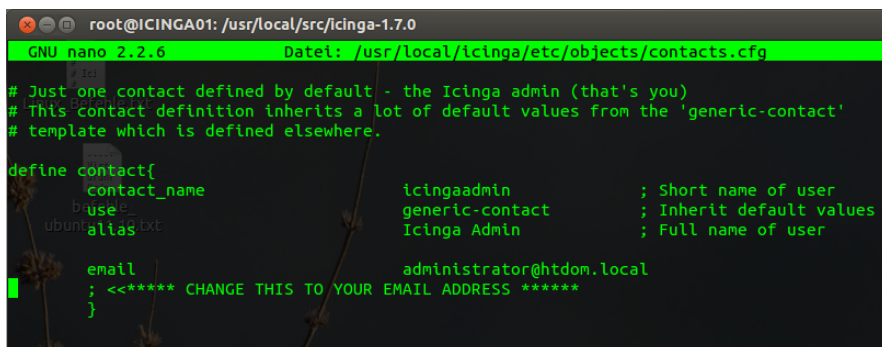
```
make install
make install-init
make install-config
make install-commandmode
make install-idoutils
```

oder kürzer

```
make fullinstall
make install-config
```

Ändere bitte die `/usr/local/icinga/etc/objects/contacts.cfg`-Konfigurationsdatei mit deinem bevorzugten Editor und passen die E-Mail-Adresse in der icingaadmin-Kontaktdefinition an, so dass sie die Adresse enthält, die im Falle von Alarmen benachrichtigt werden soll.

```
nano /usr/local/icinga/etc/objects/contacts.cfg
gedit /usr/local/icinga/etc/objects/contacts.cfg
```



```
root@ICINGA01: /usr/local/src/icinga-1.7.0
GNU nano 2.2.6 Datei: /usr/local/icinga/etc/objects/contacts.cfg
# Just one contact defined by default - the Icinga admin (that's you)
# This contact definition inherits a lot of default values from the 'generic-contact'
# template which is defined elsewhere.
define contact{
  contact_name icingaadmin ; Short name of user
  use generic-contact ; Inherit default values from parent template
  alias Icinga Admin ; Full name of user
  email administrator@htdom.local
  ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}
```

Wenn du die IDOUTils mit ssl kompiliert hast, aktivieren Sie ssl in der idomod.cfg mit

```
use_ssl=1
output_type=tcpsocket
output=127.0.0.1
```

Jetzt ändern wir noch die Beispiel Dateien im Verzeichnis `/usr/local/icinga/etc`

Mit dem Befehl `mv` benennen wir die Dateien um und mit `cp` kopieren wir sie nur, und behalten die Originale.

```
cd /usr/local/icinga/etc
cp idomod.cfg-sample idomod.cfg
cp ido2db.cfg-sample ido2db.cfg
```

Aktivieren des idomod-Eventbroker-Moduls

nano /usr/local/icinga/etc/icinga.cfg (# vor broker_modul entfernen)

STRG + W für suchen

STRG + O (Enter) Speichern

STRG + X fürs beenden von nano

```
# Example:
# broker_module=<modulepath> [moduleargs]

broker_module=/usr/local/icinga/lib/idomod.so config_file=/usr/local/icinga/etc/idomod.cfg
```

MySQL Datenbank anlegen

Mit folgendem Befehl baut man eine Verbindung zum MySQL Datenbank Server auf, im Anschluss legen wir die Datenbank „icinga“ an und Berechtigungen dementsprechend

mysql -u root -p (Passwort)

CREATE DATABASE icinga;

***GRANT USAGE ON *.* TO 'icinga'@'localhost' IDENTIFIED BY 'icinga' WITH
MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0;***

Berechtigungen für den Benutzer icinga setzen

***GRANT SELECT , INSERT , UPDATE , DELETE , DROP , CREATE VIEW ON icinga.* TO
'icinga'@'localhost';***

FLUSH PRIVILEGES ;

quit

So sieht es aus wenn man die Befehle in der Konsole eingibt

```
root@ICINGA01:/usr/local/icinga/etc# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 50
Server version: 5.5.22-0ubuntu1 (Ubuntu)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE icinga;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT USAGE ON *.* TO 'icinga'@'localhost' IDENTIFIED BY 'icinga' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SELECT , INSERT , UPDATE , DELETE , DROP , CREATE VIEW ON icinga.* TO 'icinga'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES ;
```

Datenbankschema der Datenbank „icinga“ hinzufügen (*mysql.sql*)

cd /usr/local/src/icinga-1.7.0/module/idoutils/db/mysql

mysql -u root -p icinga < mysql.sql (Passwort)

```
root@ICINGA01: /usr/local/src/icinga-1.7.0/module/idoutils/db/mysql
root@ICINGA01:/usr/local/icinga/etc# cd /usr/local/src/icinga-1.7.0/module/idoutils/db/mysql
root@ICINGA01:/usr/local/src/icinga-1.7.0/module/idoutils/db/mysql# mysql -u root -p icinga < mysql.sql
Enter password:
```

nano /usr/local/icinga/etc/ido2db.cfg

überprüfen ob alle Einträge passen

db_servertype=mysql

db_port=3306

db_user=icinga

db_pass=icinga

ICINGA Webkomponenten installieren

cd /usr/local/src/icinga-1.7.0

make install-cgis

make install-html

Installieren Sie die Icinga-Web-Konfigurationsdatei im Apache conf.d-Verzeichnis.

make install-webconf

Legen Sie ein icingaadmin-Konto an, um sich am klassischen Web-Interface anmelden zu können.
Merken Sie sich das Passwort, das Sie diesem Konto geben - Sie brauchen es später.

htpasswd -c /usr/local/icinga/etc/htpasswd.users icingaadmin (2x Passwort)

/etc/init.d/apache2 restart

Zusätzlichen neuen Benutzer hinzufügen

htpasswd /usr/local/icinga/etc/htpasswd.users <USERNAME> (2x neues Passwort)

Der Benutzer muss aber lokal am ICINGA Server angelegt sein

Nagios Plug-Ins 1.4.15 installieren

Wechsel in das lokale Source-Verzeichnis von Ubuntu

```
cd /usr/local/src
```

Download von den Nagios Plug-Ins Quelldateien

Nagios Plug-Ins Quelldateien Entpacken

In das Nagios Plug-Ins Verzeichnis wechseln

```
wget http://surfnet.dl.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.15.tar.gz  
tar xvf nagios-plugins-1.4.15.tar.gz  
cd nagios-plugins-1.4.15
```

Führe das Konfigurations- Script aus

```
./configure --prefix=/usr/local/icinga --with-cgiurl=/icinga/cgi-bin --with-htmurl=/icinga --with-nagios-user=icinga --with-nagios-group=icinga
```

Kompilieren das Ganze und installieren im Anschluss die Plug-Ins

```
make  
make install
```

IDOUutils und Icinga starten - IDOUutils muss vor Icinga gestartet werden

```
/etc/init.d/ido2db start  
update-rc.d ido2db defaults
```

ICINGA Konfiguration überprüfen

```
/usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
```

Anstatt die Pfade für das Binary und die Konfigurationsdatei anzugeben kannst du auch folgenden Befehl eingeben:

```
/etc/init.d/icinga checkconfig
```

Die Ausführung ergibt einen Return-Code ungleich Null, wenn die Konfiguration Fehler enthält. Das kann sinnvoll sein, wenn man Icinga automatisch starten will. Gibt man folgenden Befehl ein

```
/etc/init.d/icinga show-errors
```


dann wird eine Datei mit den gefundenen Fehlern erstellt. Falls es Fehler gibt, wird der Inhalt der Datei angezeigt ("show-errors" ab Icinga 1.0.2).

ICINGA Dienst starten

/etc/init.d/icinga start

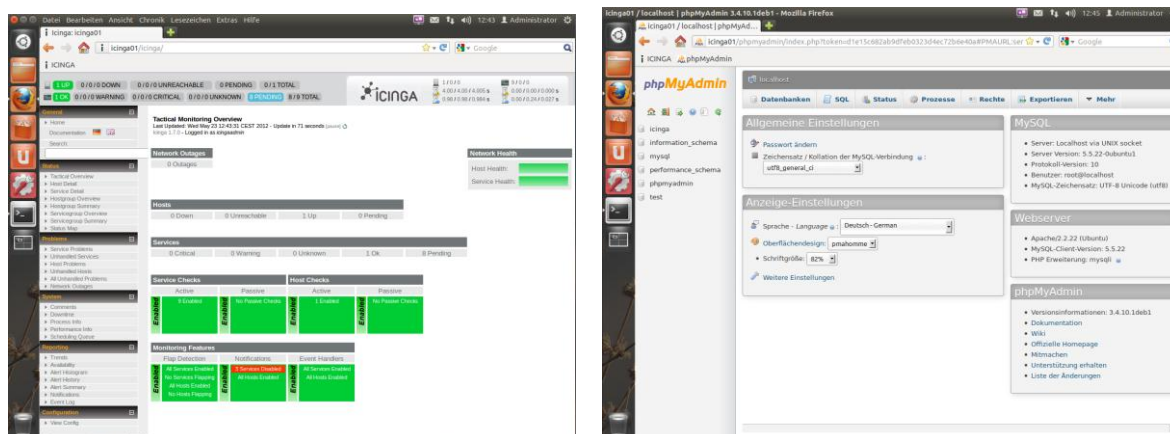
Fügen ICINGA zu der Liste der System-Services hinzu und sorgen somit für einen automatischen Start, wenn das System hochfährt.

update-rc.d icinga defaults

Stelle sicher, dass die Firewall-Einstellungen deiner Maschine einen Zugriff auf das Web-Interface ermöglichen, wenn du von anderen Rechnern darauf zugreifen willst.

iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT

Fertig - Herzlichen Glückwunsch! ICINGA 1.7.0 + PHPMyAdmin 3.4.10 wurde erfolgreich installiert.



Icinga 1.7.0 Grundkonfiguration

Es gibt viele Möglichkeiten ICINGA richtig zu konfigurieren, hier möchte ich euch meine Lösung vorschlagen, wie ihr schnell und einfach eine kleine Umgebung in eurem Netzwerk überwachen könnt. Umso komplexer die Umgebung umso mehr Überlegung muss investiert werden damit ICINGA sauber konfiguriert wird.

Sehr gute Anregung → http://www.nagios-wiki.de/nagios/tips/nagios_optimal_konfigurieren

Oder das Buch von Wolfgang Barth → http://www.amazon.de/Nagios-System-Netzwerkmonitoring-Wolfgang-Barth/dp/3937514910/ref=sr_1_2?ie=UTF8&s=books&qid=1273768468&sr=8-2

Ordnerstruktur anlegen für das Firmenweite Monitoring

Aber nun zu meiner Lösung die ich für mich jetzt am besten gefunden habe und auch heute noch einsetze. Ich lege im Verzeichnis */usr/local/icinga/etc* einen neuen Ordner mit meinen Firmennamen (*htdom*) an.

```
sudo -s (Password)
su - icinga
bash
cd /usr/local/icinga/etc
mkdir htdom
cd htdom
```

In diesen Ordner *htdom* lege ich weitere Unterordner mit meinen Standorten (standort1, 2 und 3) an.

```
mkdir munich vienna zurich
```

Kopiere mir folgende Dateien (*commands.cfg*, *contacts.cfg*, *templates.cfg*, *timeperiods.cfg*) vom Original Ordner */usr/local/icinga/etc/objects* in das Verzeichnis */usr/local/icinga/etc/htdom*

```
cp /usr/local/icinga/etc/objects/commands.cfg /usr/local/icinga/etc/htdom
cp /usr/local/icinga/etc/objects/ contacts.cfg /usr/local/icinga/etc/htdom
cp /usr/local/icinga/etc/objects/ templates.cfg /usr/local/icinga/etc/htdom
cp /usr/local/icinga/etc/objects/ timeperiods.cfg /usr/local/icinga/etc/htdom
```

hostgroup.cfg

Lege in meinen **htdom** Ordner eine neue Datei, mit dem Namen **hostgroup.cfg** an und fülle diese Datei mit folgendem Inhalt.

```
cd /usr/local/icinga/etc/htdom
touch hostgroup.cfg
nano hostgroup.cfg
```

Der erste Eintrag beinhaltet meine erste Hostgruppe mit dem Firmennamen (htdom), der Alias ist frei wählbar (z.B. Firma HTDOM Inc.), bei den Members möchte ich alle Server der Firma zusammenfassen, das wird mit dem Stern* markiert, bei den Eintrag **hostgroup_members** verschachteln ich alle anderen Hostgruppen der einzelnen Standorte (location1, 2 und 3), Der Rest der Datei denke ich ist Selbsterklärend.

```
#####
# HOST GROUP Firma HTDOM Inc.
#####
define hostgroup {
    hostgroup_name    htdom
    alias              Firma HTDOM Inc.
    members            *
    hostgroup_members  munich,vienna,zurich
}
#####
# HOST GROUP Standort Muenchen
#####
define hostgroup {
    hostgroup_name    munich
    alias              Server Muenchen
    members            icinga01.htdom.local,server2. htdom.local,server3.htdom.local,usw. (Bitte ersetzen)
}
#####
# HOST GROUP Standort Wien
#####
define hostgroup {
    hostgroup_name    vienna
    alias              Server Wien
    members            server4.htdom.local,server5.htdom.local (Bitte ersetzen)
}
#####
# HOST GROUP Standort Zuerich
#####
define hostgroup {
    hostgroup_name    zurich
    alias              Server Zuerich
    members            server6.htdom.local,server7.htdom.local (Bitte ersetzen)
}
#####
```

localhost.cfg

Im nächste Schritt kopiere ich die Original Datei **localhost.cfg** in meinen Ordner standort1, 2 und 3.
Nach dem kopieren benenne ich die Datei so um, wie mein Server heißen. (Bitte vergesst die Endung der Datei .cfg nicht, alle Dateien die nicht so enden ignoriert ICINGA)

```
cp /usr/local/icinga/etc/objects/localhost.cfg /usr/local/icinga/etc/htdom/munich
cp /usr/local/icinga/etc/objects/localhost.cfg /usr/local/icinga/etc/htdom/vienna
cp /usr/local/icinga/etc/objects/localhost.cfg /usr/local/icinga/etc/htdom/zurich
```

```
mv /usr/local/icinga/etc/htdom/munich/localhost.cfg /usr/local/icinga/etc/htdom/munich
/icinga01.htdom.local.cfg
```

```
mv /usr/local/icinga/etc/htdom/vienna/localhost.cfg
/usr/local/icinga/etc/htdom/vienna/ads02.htdom.local.cfg
```

```
mv /usr/local/icinga/etc/htdom/zurich/localhost.cfg
/usr/local/icinga/etc/htdom/zurich/moss01.htdom.local.cfg
```

templates.cfg

Wenn das erledigt ist dann Konfiguriere ich meine **templates.cfg** Datei
Nähere Informationen findet Ihr hier: <http://docs.icinga.org/latest/de/objectdefinitions.html>

```
#####
# CONTACT TEMPLATES
#####
define contact{
    name                generic-contact        ; The name of this contact template
    service_notification_period 24x7            ; service notifications can be sent anytime
    host_notification_period 24x7              ; host notifications can be sent anytime
    service_notification_options w,u,c,r,f,s   ; send notifications for all service states, flapping events, and
    host_notification_options d,u,r,f,s        ; send notifications for all host states, flapping events, and scheduled
    service_notification_commands notify-service-by-email ; send service notifications via email
    host_notification_commands notify-host-by-email ; send host notifications via email
    register             0                    ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL CONTACT
}
#####
# LINUX HOST TEMPLATES
#####
define host{
    name                linux-host            ; The name of this host template
    notifications_enabled 1                    ; Host notifications are enabled
    event_handler_enabled 1                    ; Host event handler is enabled
    flap_detection_enabled 1                   ; Flap detection is enabled
    failure_prediction_enabled 1               ; Failure prediction is enabled
    process_perf_data     1                    ; Process performance data
    retain_status_information 1                 ; Retain status information across program restarts
```

```

retain_nonstatus_information 1 ; Retain non-status information across program restarts
check_period 24x7 ; By default, Linux hosts are checked round the clock
check_interval 5 ; Actively check the host every 5 minutes
retry_interval 1 ; Schedule host check retries at 1 minute intervals
max_check_attempts 5 ; Check each Linux host 10 times (max)
check_command check-host-alive ; Default command to check Linux hosts
notification_period 24x7 ; Linux admins hate to be woken up, so we only notify during the day
notification_interval 30 ; Resend notifications every 2 hours
notification_options d,u,r ; Only send notifications for specific host states
contact_groups admins ; Notifications get sent to the admins by default
register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!
}

```

```
#####
```

WINDOWS HOST TEMPLATES

```
#####
```

```

define host{
name windows-host ; The name of this host template
notifications_enabled 1 ; Host notifications are enabled
event_handler_enabled 1 ; Host event handler is enabled
flap_detection_enabled 1 ; Flap detection is enabled
failure_prediction_enabled 1 ; Failure prediction is enabled
process_perf_data 1 ; Process performance data
retain_status_information 1 ; Retain status information across program restarts
retain_nonstatus_information 1 ; Retain non-status information across program restarts
check_period 24x7 ; By default, Linux hosts are checked round the clock
check_interval 5 ; Actively check the host every 5 minutes
retry_interval 1 ; Schedule host check retries at 1 minute intervals
max_check_attempts 5 ; Check each Linux host 10 times (max)
check_command check-host-alive ; Default command to check Linux hosts
notification_period 24x7 ; Linux admins hate to be woken up, so we only notify during the day
notification_interval 30 ; Resend notifications every 2 hours
notification_options d,u,r ; Only send notifications for specific host states
contact_groups admins ; Notifications get sent to the admins by default
register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!
}

```

```
#####
```

WINDOWS SERVICE TEMPLATES

```
#####
```

```

define service{
name windows-service ; The 'name' of this service template
active_checks_enabled 1 ; Active service checks are enabled
passive_checks_enabled 1 ; Passive service checks are enabled/accepted
parallelize_check 1 ; Active service checks should be parallelized (disabling this can lead to
obsess_over_service 1 ; We should obsess over this service (if necessary)
check_freshness 0 ; Default is to NOT check service 'freshness'
notifications_enabled 1 ; Service notifications are enabled
event_handler_enabled 1 ; Service event handler is enabled
flap_detection_enabled 1 ; Flap detection is enabled
failure_prediction_enabled 1 ; Failure prediction is enabled
process_perf_data 1 ; Process performance data
retain_status_information 1 ; Retain status information across program restarts
retain_nonstatus_information 1 ; Retain non-status information across program restarts
is_volatile 0 ; The service is not volatile
check_period 24x7 ; The service can be checked at any time of the day
check_interval 5 ; Actively check the host every 5 minutes
retry_interval 1 ; Schedule host check retries at 1 minute intervals
max_check_attempts 5 ; Check each Linux host 10 times (max)
contact_groups admins ; Notifications get sent out to everyone in the 'admins' group
notification_options c,r ; Send notifications about warning, unknown, critical, and recovery events
}

```

```
notification_interval    20           ; Re-notify about service problems every hour
notification_period      24x7         ; Notifications can be sent out at any time
register                 0           ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!
}
```

```
#####
# LINUX SERVICE TEMPLATES
#####
define service{
name                linux-service    ; The name of this service template
active_checks_enabled 1              ; Active service checks are enabled
passive_checks_enabled 1             ; Passive service checks are enabled/accepted
parallelize_check     1              ; Active service checks should be parallelized (disabling this can lead to
obsess_over_service   1              ; We should obsess over this service (if necessary)
check_freshness       0              ; Default is to NOT check service 'freshness'
notifications_enabled 1              ; Service notifications are enabled
event_handler_enabled 1              ; Service event handler is enabled
flap_detection_enabled 1             ; Flap detection is enabled
failure_prediction_enabled 1         ; Failure prediction is enabled
process_perf_data     1              ; Process performance data
retain_status_information 1          ; Retain status information across program restarts
retain_nonstatus_information 1       ; Retain non-status information across program restarts
is_volatile           0              ; The service is not volatile
check_period          24x7          ; The service can be checked at any time of the day
check_interval        5             ; Actively check the host every 5 minutes
retry_interval        1             ; Schedule host check retries at 1 minute intervals
max_check_attempts    5             ; Check each Linux host 10 times (max)
contact_groups        admins        ; Notifications get sent out to everyone in the 'admins' group
notification_options   c,r          ; Send notifications about warning, unknown, critical, and recovery events
notification_interval  20           ; Re-notify about service problems every hour
notification_period    24x7         ; Notifications can be sent out at any time
register              0             ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!
}
```

```
#####
```

icinga.cfg

Als nächstes gebe ich noch in der Hauptkonfigurations-Datei **icinga.cfg** an, in welchen Ordner ICINGA meine neuen Konfigurationsdateien findet.

<http://docs.icinga.org/latest/de/configmain.html>

Ich öffne als root die **icinga.cfg** und füge folgenden Eintrag hinzu.

```
sudo nano /usr/local/icinga/etc/icinga.cfg
sudo gedit /usr/local/icinga/etc/icinga.cfg
cfg_dir=/usr/local/icinga/etc/htdom
```

alle anderen CFG Dateien die in den Ordner **/usr/local/icinga/etc/objects** verweisen. Kommentiere ich mit einem # aus.

```
# You can specify individual object config files as shown below:
#cfg_file=/usr/local/icinga/etc/objects/commands.cfg
#cfg_file=/usr/local/icinga/etc/objects/contacts.cfg
#cfg_file=/usr/local/icinga/etc/objects/timeperiods.cfg
#cfg_file=/usr/local/icinga/etc/objects/templates.cfg

# Definitions for monitoring the local (Linux) host
#cfg_file=/usr/local/icinga/etc/objects/localhost.cfg

# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/icinga/etc/objects/windows.cfg

# Definitions for monitoring a router/switch
#cfg_file=/usr/local/icinga/etc/objects/switch.cfg

# Definitions for monitoring a network printer
#cfg_file=/usr/local/icinga/etc/objects/printer.cfg

# Definitions for ido2db process checks
#cfg_file=/usr/local/icinga/etc/objects/ido2db_check_proc.cfg

# Definitions for broker modules like idoutils.cfg
cfg_dir=/usr/local/icinga/etc/modules

# Definitions for monitoring the HTDOM Structure
cfg_dir=/usr/local/icinga/etc/htdom
```

Und da ich gerne das Europäische Datumsformat lesen möchte, ändere ich noch den Eintrag **date_format** von **us** (Standard) in **euro** um.

```
# DATE FORMAT OPTION
# This option determines how short dates are displayed. Valid options
# include:
#   us           (MM-DD-YYYY HH:MM:SS)
#   euro         (DD-MM-YYYY HH:MM:SS)
#   iso8601      (YYYY-MM-DD HH:MM:SS)
#   strict-iso8601 (YYYY-MM-DDTHH:MM:SS)
#
date_format=euro
```

timeperiods.cfg

Bei der Datei `timeperiods.cfg` sollte man unbedingt die Feiertage von seiner Region anpassen, diese Einstellung sind wichtig damit ICINGA weiß wann es den Administratoren genehm ist, dass man Sie benachrichtigt. ☺

```
#####
# Some U.S. holidays
# Note: The timeranges for each holiday are meant to *exclude* the holidays from being
# treated as a valid time for notifications, etc. You probably don't want your pager
# going off on New Year's. Although you're employer might... :-)

define timeperiod {
    name                us-holidays
    timeperiod_name     us-holidays
    alias               U.S. Holidays
    january 1           00:00-00:00 ; New Years
    monday -1 may       00:00-00:00 ; Memorial Day (last Monday in May)
    july 4              00:00-00:00 ; Independence Day
    monday 1 september  00:00-00:00 ; Labor Day (first Monday in September)
    thursday -1 november 00:00-00:00 ; Thanksgiving (last Thursday in November)
    december 25         00:00-00:00 ; Christmas
}
#####
```

This defines a modified "24x7" timeperiod that covers every day of the
year, except for U.S. holidays (defined in the timeperiod above).

```
define timeperiod {
timeperiod_name    24x7_sans_holidays
    alias            24x7 Sans Holidays
    use              us-holidays           ; Get holiday exceptions from other timeperiod
    sunday           00:00-24:00
    monday           00:00-24:00
    tuesday          00:00-24:00
    wednesday        00:00-24:00
    thursday         00:00-24:00
    friday            00:00-24:00
    saturday         00:00-24:00
}
```

#####

icinga01.htdom.local.cfg

#####

HOST DEFINITION

#####

```
define host {
    use                linux-host           ; Host Template (templates.cfg)
    host_name          icinga01.htdom.local ; FQDN - Servername
    alias              ICINGA IT Monitoring Server ; Frei definierbarer Anzeigename
    address            192.168.0.235        ; Statische IP Adresse vom Server
    hostgroups         munich               ; Hostgruppenname (hostgroup.cfg)
}
```

#####

SERVICE DEFINITION

#####

```
define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description PING
    check_command      check_ping!100.0,20%!500.0,60%
}
```

#####

```
define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description angemeldete Benutzer
    check_command      check_local_users!10!20
}
```

#####


```

define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description _boot Partition
    check_command      check_local_disk!20%!10%!/boot
}

#####

define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description _root Partition
    check_command      check_local_disk!20%!10%!/
}

#####

define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description _var Partition
    check_command      check_local_disk!20%!10%!/var
}

#####

define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description _home Partition
    check_command      check_local_disk!20%!10%!/home
}

#####

define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description _tmp Partition
    check_command      check_local_disk!20%!10%!/tmp
}

#####

define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description _swap Partition
    check_command      check_local_swap!20!10
}

#####

define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description Alle Prozesse
    check_command      check_local_procs!250!400!RSZDT
}

```

```
#####
```

```
define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description CPU State
    check_command       check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}
```

```
#####
```

```
define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description HTTP Server
    check_command       check_http
    notifications_enabled 0
}
```

```
#####
```

```
#define service {
    #use                linux-service
    #host_name          icinga01.htdom.local
    #service_description Netzwerk Traffic
    #check_command       check_unixtraffic!192.168.0.235!1000!90!95 -B -u m
#}
```

```
#####
```



Bitte nicht wundern warum der letzte Service **ausgehasht** wurde, aber der Service **check_unixtraffic** ist noch nicht in der **commands.cfg** definiert und würde jetzt so auf einen Fehler laufen. Dazu aber später mehr.

Sieht nach erfolgreicher Abfrage von ICINGA wie folgt aus

Service Status Details For All Hosts						
Host --	Service --	Status --	Last Check --	Duration --	Attempt --	Status Information
icinga01.htdom.local	Alle Prozesse	OK	24-05-2012 11:20:50	0d 0h 42m 39s	1/5	PROCS OK: 76 processes with STATE = RSZDT
	CPU State	OK	24-05-2012 11:21:17	0d 0h 42m 12s	1/5	OK - load average: 0.00, 0.05, 0.11
	HTTP Server	OK	24-05-2012 11:21:44	0d 0h 41m 45s	1/5	HTTP OK: HTTP/1.1 200 OK - 453 bytes in 0,001 second response time
	PING	OK	24-05-2012 11:22:12	0d 0h 41m 17s	1/5	PING OK - Packet loss = 0%, RTA = 0.04 ms
	_boot Partition	OK	24-05-2012 11:22:39	0d 0h 40m 50s	1/5	DISK OK - free space: /boot 79 MB (59% inode=99%):
	_home Partition	OK	24-05-2012 11:23:06	0d 0h 40m 23s	1/5	DISK OK - free space: /home 11421 MB (97% inode=99%):
	_root Partition	OK	24-05-2012 11:18:33	0d 0h 39m 56s	1/5	DISK OK - free space: / 42811 MB (92% inode=94%):
	_swap Partition	OK	24-05-2012 11:19:01	0d 0h 39m 28s	1/5	SWAP OK - 100% free (1951 MB out of 1951 MB)
	_tmp Partition	OK	24-05-2012 11:19:28	0d 0h 39m 1s	1/5	DISK OK - free space: /tmp 3576 MB (96% inode=99%):
	_var Partition	OK	24-05-2012 11:19:55	0d 0h 38m 34s	1/5	DISK OK - free space: /var 12763 MB (92% inode=99%):
	angemeldete Benutzer	OK	24-05-2012 11:20:23	0d 0h 38m 6s	1/5	USERS OK - 1 users currently logged in

11 Matching Service Entries Displayed

Firma HTDOM Inc. (htdom)

Host	Status	Services	Actions
icinga01.htdom.local	UP	11 OK	 

Server Muenchen (munich)

Host	Status	Services	Actions
icinga01.htdom.local	UP	11 OK	 

ads01.htdom.local

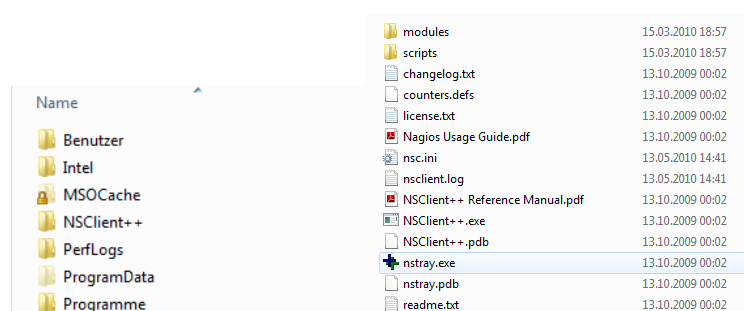
Ähnlich läuft es auch bei den Windows Server/Client ab, nur benötige ich für den Windows Server/Client noch zusätzlich einen Agent der im Hintergrund läuft. Diesen Agent bekommt man im Internet → <http://nsclient.org/nscp/> → und es handelt sich um den NSClient++ Agent.

Download → x86 oder x64 Clients →

<http://sourceforge.net/projects/nscplus/files/nscplus/NSClient%2B%2B%200.3.8/NSClient%2B%2B-0.3.8-Win32.zip>

<http://sourceforge.net/projects/nscplus/files/nscplus/NSClient%2B%2B%200.3.8/NSClient%2B%2B-0.3.8-x64.zip>

Diesen Client extrahiere ich und benenne den Ordner um in NSClient++ und kopiere diesen Ordner auf jeden Server/Client der überwacht werden soll lokal in das Verzeichnis C:\



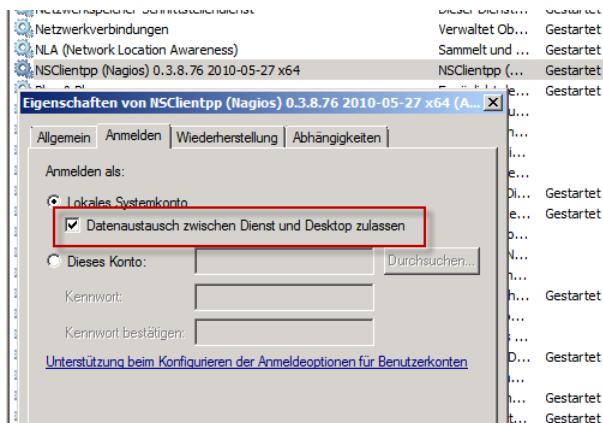
Danach rufe ich eine CMD Console auf und Wechsel in das C:\NSClient++ Verzeichnis

```
cd \
cd NSClient++
NSClient++.exe /install
```

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Helmut Thurnhofer>cd \
C:\>cd "NSClient++"
C:\NSClient++>"NSClient++.exe" /install
```

Nach erfolgreicher Installation des Dienstes, öffnen wir die **services.msc** über Start → Ausführen und setzen das Häkchen „**Datenaustausch zwischen Dienst und Desktop zulassen**“ im Reiter Anmelden, danach können wir den Dienst getrost starten. Überprüft im Reiter Allgemein das der Dienst im Starttyp Automatisch drinstehen hat.



Im nächsten Schritt muss noch lokal am Client die **nsc.ini** bearbeitet werden, ich führe hier nur die Dinge auf, bei denen ich am Anfang vom Eintrag das Semikolon entfernt habe.

Unter [modules] wurden folgende DLL's freigeschaltet

FileLogger.dll
 CheckSystem.dll
 CheckDisk.dll
 NSClientListener.dll
 NRPEListener.dll
 SysTray.dll
 CheckEventLog.dll
 CheckHelpers.dll

Unter [Settings] wurden folgende Punkte konfiguriert

password=#icinga# (Wenn Ihr hier ein Passwort einträgt, dann muss das in der commands.cfg ebenfalls mit angegeben werden → **command_line \$USER1\$/check_nt -H \$HOSTADDRESS\$ -p 12489 -s#icinga# -v \$ARG1\$ \$ARG2\$**)

```
# 'check_nt' command definition
define command{
    command_name    check_nt
    command_line    $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s#icinga# -v $ARG1$ $ARG2$
}
```

allowed_hosts=192.168.0.235

use_file=1

shared_session=1 (Wegen Windows Vista/7 bzw. Server 2008/2008 R2)

Unter [NSClient]

allowed_hosts=192.168.0.235

port=12489

Unter [Check System]

CPUBufferSize=1h

Unter [NSCA Agent]

hostname=ads01.htdom.local

nsca_host=192.168.0.235

jetzt starten wir nochmal den NSClient Dienst durch mit ***net stop/start NSClientpp*** (Hier aufpassen das das Shortcut Icon unten rechts bei der Uhr ebenfalls beendet wurde)



```
Administrator: Eingabeaufforderung
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Administrator>net stop nsclientpp && net start nsclientpp
NSClientpp (Nagios) 0.3.8.76 2010-05-27 x64 wird beendet..
NSClientpp (Nagios) 0.3.8.76 2010-05-27 x64 wurde erfolgreich beendet.

NSClientpp (Nagios) 0.3.8.76 2010-05-27 x64 wird gestartet.
NSClientpp (Nagios) 0.3.8.76 2010-05-27 x64 wurde erfolgreich gestartet.
```

```
#####
# HOST DEFINITION
#####

define host {
    use                windows-host                ; Host Template (templates.cfg)
    host_name          ads01.htdom.local            ; FQDN - Servername
    alias              1. Domänencontroller HTDOM   ; Frei definierbarer Anzeigenname
    address            192.168.0.231                ; Statische IP Adresse vom Server
    hostgroups         munich                       ; Hostgruppenname (hostgroup.cfg)
    #parents           fritz.box                    ; Unmittelbarer Nachbar (Physikalischer Switch/Router)
}

#####
# SERVICE DEFINITION
#####

define service {
    use                windows-service
    host_name          ads01.htdom.local
    service_description Service Check
    check_command      check_nt!SERVICESTATE!-d SHOWALL -l 'NSClientpp'
}

#####

define service {
    use                windows-service
    host_name          ads01.htdom.local
    service_description Server Uptime
    check_command      check_nt!UPTIME
}

#####
```

```
define service {
    use                windows-service
    host_name          ads01.htdom.local
    service_description CPU Auslastung
    check_command      check_nt!CPULOAD!-l 10,90,95,30,90,95
}

#####

#define service {
    #use                windows-service
    #host_name          ads01.htdom.local
    #service_description Memory Auslastung
    #check_command      check_nt!MEMUSE!-w 80 -c 90
#}

#####

define service {
    use                windows-service
    host_name          ads01.htdom.local
    service_description _C:\ Festplatte
    check_command      check_nt!USEDISKSPACE!-l c -w 80% -c 90%
}

#####

define service {
    use                windows-service
    host_name          ads01.htdom.local
    service_description _D:\ Festplatte
    check_command      check_nt!USEDISKSPACE!-l d -w 80% -c 90%
}





#####
```

Bitte nicht wundern warum der Service **ausgehasht** wurde, aber der Service **check_nt!MEMUSE** ist noch nicht in der **commands.cfg** definiert und würde jetzt auf einen Fehler laufen. Dazu aber später mehr.





Sieht nach erfolgreicher Abfrage von ICINGA wie folgt aus

Host +-	Service +-	Status +-	Last Check +-	Duration +-	Attempt +-	Status Information	
ads01.htdom.local	CPU Auslastung	OK	24-05-2012 14:07:11	0d 0h 23m 31s	1/5	CPU Load 0% (10 min average) 5% (30 min average)	<input type="checkbox"/>
	Server Uptime	OK	24-05-2012 14:08:26	0d 0h 22m 16s	1/5	System Uptime - 15 day(s) 10 hour(s) 52 minute(s)	<input type="checkbox"/>
	Service Check	OK	24-05-2012 14:09:03	0d 0h 21m 39s	1/5	NSClientpp: Started	<input type="checkbox"/>
	_C:\ Festplatte	OK	24-05-2012 14:07:53	0d 0h 22m 49s	1/5	c: - total: 59,90 Gb - used: 16,91 Gb (28%) - free 42,99 Gb (72%)	<input type="checkbox"/>
	_D:\ Festplatte	OK	24-05-2012 14:08:31	0d 0h 22m 11s	1/5	d: - total: 100,00 Gb - used: 11,09 Gb (11%) - free 88,91 Gb (89%)	<input type="checkbox"/>

Firma HTDOM Inc. (htdom)

Host	Status	Services	Actions
ads01.htdom.local	UP	5 OK	 
icinga01.htdom.local	UP	11 OK	 

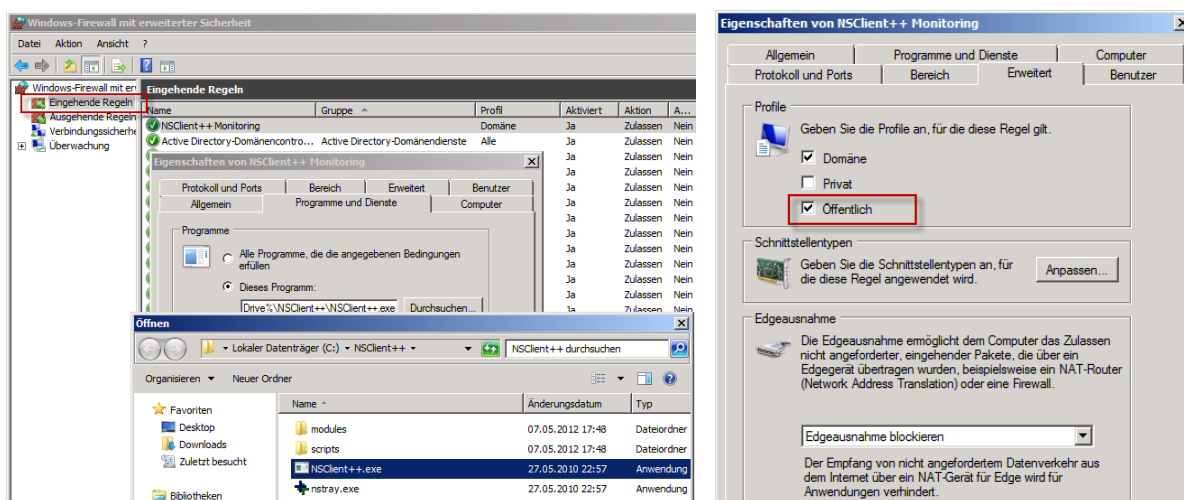
Server Muenchen (munich)

Host	Status	Services	Actions
ads01.htdom.local	UP	5 OK	 
icinga01.htdom.local	UP	11 OK	 

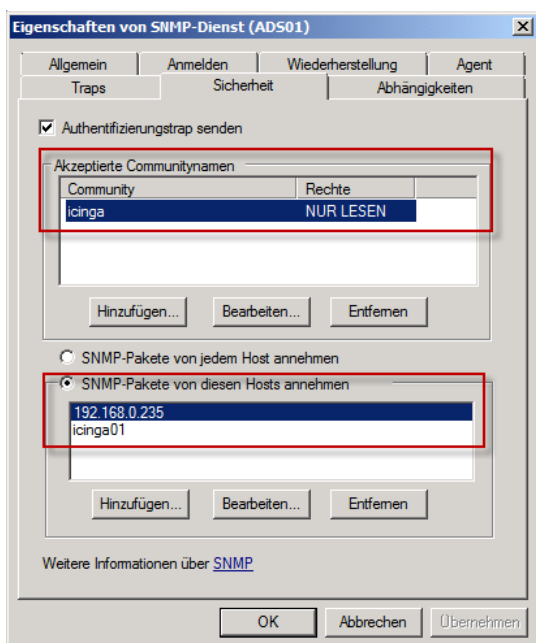
Windows Firewall und SNMP Dienst konfigurieren

Sollte die Abfrage auf Fehler laufen und das wird sie, muss noch die Windows Firewall konfiguriert werden.

Hier benötigen wir eine **Eingehende Regel**, die der NSClient++.exe erlaubt, abgefragt zu werden. Wir definieren die Regel nur auf dem **Domänen Profil**, sollte ein Windows 2008 R2 Core Server abgefragt werden muss zusätzlich zum Domänen Profil auch das **Öffentliche Profil** dazu konfiguriert werden.



Der SNMP Dienst wird ebenfalls benötigt, diesen bitte installieren und konfigurieren.



commands.cfg

Die letzte Änderung für die Grundkonfiguration ist das ändern der Service abfragen in der **commands.cfg**

Bei dem Service **check_host_alive** ändere ich das Command von check_ping in check_icmp

```
# 'check-host-alive' command definition
define command{
    command_name    check-host-alive
    command_line    $USER1$/check_icmp -H $HOSTADDRESS$ -w 3000.0,80% -c 5000.0,100% -p 5
}
```

Und dasselbe mache ich auch noch bei dem Command **check_ping**

```
# 'check_ping' command definition
define command{
    command_name    check_ping
    command_line    $USER1$/check_icmp -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$ -p 5
}
```

Grund dafür ist das mir der **check_ping** keine Performance Daten ausgibt **check_icmp** aber schon, die Daten benötige ich zu einen späteren Zeitpunkt.

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
```

gedit /etc/default/snmpd

```
# snmpd control (yes means start daemon).
SNMPDPRUN=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 -c /etc/snmp/snmpd.conf'

# snmptrapd control (yes means start daemon). As of net-snmp version
# 5.0, master agentx support must be enabled in snmpd before snmptrapd
# can be run. See snmpd.conf(5) for how to do this.
TRAPDPRUN=no
```

SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g 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 oder nicht.**

Folgende Vorgehensweise würde ich euch vorschlagen, um neue Plug-Ins zu testen:

1. Plug-in/Perl Skript downloaden und in das Verzeichnis **/usr/local/icinga/libexec** kopieren
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
2. Sich als Icinga Benutzer auf der Console einloggen → **su - icinga (Passwort)**
3. **#bash**
4. 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**
5. 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'
```

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

```
Average IN: 0Kbs (0.00%), Average OUT: 0Kbs (0.00%)<br>Total RX: 16.63 Mbits, Total TX: 2.12  
Mbits|inUsage=0.00%;90;95 outUsage=0.00%;90;95 inBandwidth=0Kbs outBandwidth=0Kbs  
inAbsolut=2179437 outAbsolut=277565
```

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  
    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/nrpe-2.13/nrpe-2.13.tar.gz
```

Extrahieren das Paket mit folgendem Befehl

```
tar xvfz nrpe-2.13.tar.gz
```

wechseln in das Verzeichnis von NRPE

```
cd nrpe-2.13
```

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
```

Hinweis: Sollte folgender Fehler kommen

```
checking for socklen_t... yes  
checking for type of socket size... size t  
checking for SSL headers... SSL headers found in /usr  
checking for SSL libraries... configure: error: Cannot find ssl libraries
```

Bitte noch folgendes machen, damit die Konfiguration sauber durchläuft.

```
dpkg -L libssl-dev  
ln -s /usr/lib/x86_64-linux-gnu/libssl.so /usr/lib/libssl.so
```

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

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
```

commands.cfg die zweite

Um jetzt die letzten Services der beiden Server abfragen zu können muss ich noch ein bisschen in der **commands.cfg** konfigurieren.

Um jetzt den Arbeitsspeicher mit NRPE vom 1. Domänencontroller **ADS01** abfragen zu können

```
#####

define service {
    use                windows-service
    host_name          ads01.htdom.local
    service_description Memory Auslastung
    check_command       check_nt!MEMUSE!-w 80 -c 90
}

#####
```

öffne ich die commands.cfg und definiere folgende Zeile

```
# '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
}
```

Und Lokal auf dem Server in der nsc.ini folgende Zeile

```
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=192.168.0.235
```

Überprüfe die Icinga Konfiguration und starte den Icinga Service durch.

/etc/init.d/icinga checkconfig

/etc/init.d/icinga restart

Host	Service	Status	Last Check	Duration	Attempt	Status Information	
ads01.htdom.local	ADDS Dienst	OK	24-05-2012 17:00:12	0d 3h 4m 35s	1/5	NTDS: Started	<input type="checkbox"/>
	ADDS Zertifikat Dienst	OK	24-05-2012 17:01:30	0d 3h 4m 9s	1/5	CertSvc: Started	<input type="checkbox"/>
	Active Directory Check	OK	24-05-2012 17:01:55	0d 3h 10m 43s	1/5	TCP OK - 0,000 second response time on port 389	<input type="checkbox"/>
	CPU Auslastung	OK	24-05-2012 17:01:30	0d 3h 4m 17s	1/5	CPU Load 0% (10 min average) 0% (30 min average)	<input type="checkbox"/>
	DNS Server	OK	24-05-2012 17:01:56	0d 3h 1m 51s	1/5	DNS: Started	<input type="checkbox"/>
	Memory Auslastung	OK	24-05-2012 17:02:20	0d 0h 0m 39s	1/5	Memory usage: total:2047,57 Mb - used: 713,85 Mb (35%) - free: 1333,72 Mb (65%)	<input type="checkbox"/>
	Microsoft SMB Check In	OK	24-05-2012 17:02:22	0d 6h 19m 44s	1/5	TCP OK - 0,000 second response time on port 445	<input type="checkbox"/>

Bei der Netzwerkkarte verfare ich genauso, in der ads01.htdom.local.cfg definiere ich folgende Zeilen.

```
#####

define service {
    use                windows-service
    host_name          ads01.htdom.local
    service_description Netzwerk Traffic
    check_command       check_wintraffic!192.168.0.231!1000!90!95 -B -u m
}

#####
```

Und in der **commands.cfg** schreibe ich folgendes rein

```
# Netzwerk Traffic Windows
define command {
    command_name       check_wintraffic
    command_line        $USER1$/check_iftraffic3.pl -H $HOSTADDRESS$ -C icinga -A $ARG1$ -b $ARG2$ -w $ARG3$ -c $ARG4$
}
```

Überprüfe wieder die Icinga Konfiguration und starte den Icinga Service durch.

/etc/init.d/icinga checkconfig

/etc/init.d/icinga restart

Memory Auslastung	OK	24-05-2012 17:12:20	0d 0h 12m 49s	1/5	Memory usage: total:2047,57 Mb - used: 711,40 Mb (35%) - free: 1336,17 Mb (65%)	<input type="checkbox"/>
Microsoft SMB Check In	OK	24-05-2012 17:12:22	0d 6h 31m 54s	1/5	TCP OK - 0,000 second response time on port 445	<input type="checkbox"/>
Netlogon Dienst	OK	24-05-2012 17:12:48	0d 3h 13m 9s	1/5	Netlogon: Started	<input type="checkbox"/>
Netzwerk Traffic	OK	24-05-2012 17:14:55	0d 0h 0m 14s	1/5	Average IN: 0Kbs (0.00%), Average OUT: 0Kbs (0.00%) Total RX: 75.06 Mbits, Total TX: 7.92 Mbits	<input type="checkbox"/>
Server Uptime	OK	24-05-2012 17:13:14	0d 3h 12m 53s	1/5	System Uptime - 0 day(s) 1 hour(s) 22 minute(s)	<input type="checkbox"/>

Mit dem Icinga Server verfare ich genauso nur das ich hier ein weiteres Perl Script eingesetzt habe weil das **check_iftraffic3** nicht mehr funktioniert hat, Hier verwende ich das Perl Script **check_iftraffic4.pl**

So sieht der Befehl aus wenn ich in am Icinga Server als icinga User absetze

su – icinga

bash

/usr/local/icinga/libexec/check_iftraffic4.pl -H icinga01 -C icinga -A 192.168.0.235 -r -l 1000 -u g -w 90 -c 95

Average IN: 0bBs (0.00%), Average OUT: 0bBs (0.00%) Total RX: 17.32875 Mbytes, Total TX: 579.925 Mbytes|inUsage=0.00%;90;95 outUsage=0.00%;90;95 inBandwidth=0bBs outBandwidth=0bBs inAbsolut=17328604 outAbsolut=579924403

In der **commands.cfg** definiere ich jetzt folgendes

```
# Netzwerk Traffic Linux
define command {
    command_name    check_linuxtraffic
    command_line    $USER1$/check_iftraffic4.pl -H $HOSTADDRESS$ -C icinga -A $ARG1$ -r -l $ARG2$ -u g -w $ARG3$ -c $ARG4$
}
```

In der **icinga01.htdom.local** definiere ich folgende Zeilen

```
#####

define service {
    use                linux-service
    host_name          icinga01.htdom.local
    service_description Netzwerk Traffic
    check_command       check_linuxtraffic!192.168.0.235!1000!90!95
}

#####
```

icinga01.htdom.local	Alle Prozesse	OK	24-05-2012 17:25:25	0d 10h 47m 37s	1/5	PROCS OK: 71 processes with STATE = RSZDT	<input type="checkbox"/>
	CPU State	OK	24-05-2012 17:27:45	0d 10h 47m 10s	1/5	OK - load average: 0.42, 0.25, 0.29	<input type="checkbox"/>
	HTTP Server	OK	24-05-2012 17:26:17	0d 10h 46m 43s	1/5	HTTP OK: HTTP/1.1 200 OK - 453 bytes in 0,001 second response time	<input type="checkbox"/>
	Netzwerk Traffic	OK	24-05-2012 17:28:13	0d 0h 0m 16s	1/5	Average IN: 0bBs (0.00%), Average OUT: 0bBs (0.00%) Total RX: 21.63625 Mbytes, Total TX: 718.28125 Mbytes	<input type="checkbox"/>
	PING	OK	24-05-2012 17:26:43	0d 10h 46m 15s	1/5	OK - 192.168.0.235: rta 0,035ms, lost 0%	<input type="checkbox"/>

Ich für meinen Teil wäre jetzt fertig, wenn alles richtig gemacht wurde, habt Ihr ein lauffähiges ICINGA 1.7.0 Server.

Wünsche euch viel Spaß damit. ☺

Gruß Helmut Thurnhofer