FreeNAC Installation Guide

There are three key documents available on http://FreeNAC.net/en/community, the Users, Technical and Installation Guides. FreeNAC administrators will need to read all three. Each is divided into several subpages, if you wish to see it all on one page, click the "Printer-friendly version" link below.

This document describes how to get FreeNAC installed.

Note: An alternative, complete step-by-step installation guidefor FreeNAC v2.1 and Suse 10 has also been contributed in Word format, see http://opennac.svn.sourceforge.net/viewvc/*checkout*/opennac/branches/2.... . It is dated Jan'07, limited to the older v2.1 and is Suse specific though.

Contents

START HERE

Welcome to the FreeNAC installation Guide. There are several methods to install FreeNAC, this section explains what they are and how to get started.

A list of hardware requirements is also provided, to help with server dimensioning/ planning.

Installation Methods

Introduction

FreeNAC is available as a Virtual Machine, a 'tarball'', from Subversion (SVN) and a Ubuntu/Debian package. There are also 'stable' and development versions.

This page explains each of these options below. If you are in a hurry and don't want to read the whole page, **we recommend**:

1. If you have a fast Internet connection, or are not used to administering Linux, the Virtual Machine is probably the easiest way to get up and running.
2. If you are a Ubuntu system administrator, try the package.
3. If you are used to linux, install using the latest stable branch sources from the Subversion respository.

Next step:

- to install the VM, follow the instructions Quick install using a Virtual Machine .
- For all other methods, follow the Installation Guide, starting with Step1: Linux platform Installation .

Which Operating System?

FreeNAC runs on Linux, this installation guides explains how to get it up and running on Linux. It could also run on UNIX systems, for example earlier versions ran on Solaris, but this Installation Guide does not cover these.

FreeNAC does not run under windows. In theory it is possible since PHP and Apache also run on Windows, but for example "vmpsd" would have to be ported.

Virtual machine

FreeNAC is quite complex, and needs a machine with GNU/Linux and many OpenSource source tools installed.

To get you started quickly with FreeNAC, we have built a Virtual Appliance (also called a Virtual Machine, VM) with Linux, the modules needed, and FreeNAC installed in a 'demo mode'.

The VM is ideal if you just want to test FreeNAC, or feel daunted by the length installation documentation for FreeNAC.

On the other hand, the VM is only updated every few months, and its slow to download.

Tarball

A tarball is a slang name for a compressed archive (see also wikipedia) containing source code, that is typically compiled and can be installed on many platforms.

Analysis: Tarballs require more knowledge, are not subject to version control (its difficult to see what you have changed, or to update to new bug fix releases) and are not integrated into the Linux package management. **No internet access is required on the target server** either. But for the same reasons its an ideal format for advanced system admisistrators.

Tarballs are also easy to generate, so the FreeNAC team often announce new releases in this format first.

Subversion (SVN) repository

The FreeNAC team use SVN (see also wikipedia) to manage the source code. Work on the source code usually takes place in parallel in two copies:

- The development release (also called unstable or trunk), where new features are added and tested.
- The stable releases (also called branches, e.g. V2.1, v2.2). When new features are addes, have been well tested, a 'branch' is created corresponding to a new version.

Analysis: Subversion can be difficult to learn at first, it is not integrated into the Linux packagement and required internet access (direct or via a proxy). The advantages are significant though: it allows the source code to be pulled directly from the FreeNAC respository and **allows easy, regular updates to the latest version of a branch** (via the *svn update* command).

It is recommended that you use the latest 'stable' release, unless you are involved in testing or development.

Package

Much production software is usually provided as a package, that is easy to install or deinstall and pre-destined for an exact operating system. Tthe operating system 'knows' what packages have been installed, on what other packages it depends, and how to remove those packages.

Packages are a very good way to distribute software since installation time is quicker (easier for the user), and dependancies between packages is taken care of. The difficulties lie in the generation of packages (the many different vriants of Linux/UNIX methods, the dependancies between versions), and the necessity to make appropriate installation/deinstallation scripts.

For resource reasons, the FreeNAC team has not generated packages until recently (Oct'07). The initial focus has been on creating Ubuntu (debian compatible) packages, since this is the preferred platform used by many of the FreeNAC core team.

Packages take time to generate and test, and so will often be behind Subversion and tarball software. However **its a good choice for Ubtunu system administrators running production servers**.

We welcome community contributions for other distrubutions (RedHhat or Suse rpm, gentoo sources etc. :-)

Hardware Requirements

To run FreeNAC, you'll need

- a Linux server (or VM) with FreeNAC installed
- a Cisco Switch
- a Windows PC to run the Windows Administration User Interface
  (this software can also run from a Windows share, and does not need to be actually 'installed').

FreeNAC server hardware requirements

Example: a site running with ~2'000 active end-devices. The server is rarely loaded (CPU or I/O). The slowest part is the Windows GUI with its complex SQL queries - not the VMPS back-end.

- A PC server which is compatible with the brand of Linux you choose.
  The FreeNAC core team use Ubuntu now (so we recommend the latest ubuntu stable server release), but Suse, Fedora, Gentoo, Rehat/CentOS have been used too. Solaris has also been used. Ironically, it can be more difficult to get the non-free, Commercial Enterprise versions of Linux working (RedHat, Suse) since the software is older and not all graphics libraries may be available.
- Disk
  - A hard-disk of at least 20G, recommended 100G: the more the better for backups.
  - Have a second disk for backups or RAID, for additional data security. Ensure the RAID system is Linux compatible.
  - IDE is fine, but SCSI or SATA are often more reliable, faster and you can boot from each disk without changing cables.
- Network Interfaces: Have two if possible.
  The second interface is useful for a dedicated switch management, or backup network. 100Mb speed is fine (Gb is not necessary).
- CPU is not a bottleneck, 2GHz should be fine.
- Memory is always good for databases. 1GB will work fine, but the more the better.
- A dual, hot plug power supply improves reliability.
- A dual, hot plug fan also improves reliability.
- Remote console management should be included in the Server Bios (e.g. HP Lights out, Sun LOM, Dell DRAC5), for easier remote installation, troubleshooting and monitoring of hardware status.

Its recommended to have at least 2, and perhaps 3 servers. If you are used to Virtual Machines, do the Master as a VM, and one or more replicas as 'real' machines. Point the switches at the replicas, and use the master for serving GUI requests, scanning and polling switches/routers and processing syslogs.

By doing the master as a VM, snapshots can be used before system upgrades, and roll backs are easier.

Quick install: using a Virtual Machine (VM)

Introduction

FreeNAC is quite complex, and needs a machine with GNU/Linux and many OpenSource source tools installed.

To get you started quickly with FreeNAC, we have built a Virtual Appliance (also called a Virtual Machine, VM) with Linux, the modules needed, and FreeNAC installed in a 'demo mode'.

The VM is ideal if you just want to test FreeNAC, or feel daunted by the length installation documentation for FreeNAC.

On the other hand, the VM is only updated every few months, and its slow to download.

There are two datasets included with the VM: "nacdemo" contains an example that should help to understand the GUI and what information FreeNAC stores, whereas "opennac" is an empty dataset ready for productive use.

The FreeNAC version included in the VM is the latest stable branch at the time the VM was created, from the Subversion repository.

Installing the VM

1. Ensure you have a Vmware product to start VMs.
   i.e. VMware Workstation or the free VMware Player or Vmware Server . For production use, consider Vmware ESX.
2. Download the file FreeNAC_VM_from SourceForge (note: its a large file, several hundred MB).
3. Uncompress it into a folder where your VMs are normally stored
4. The VM is configured to use "bridged mode", which is fine to be actively running in your network since it needs to receive packets on a dedicated IP address.
5. Start the VM and ignore errors you might receive on startup, and log in as **freenac**:**freenac** for 2.2RC4 or **root**:**freenac** for prior releases of the VM.
6. The VM will try to get an IP address via DHCP, which you can see with

```
ifconfig -a
```

Optional changes to the VM:

1. If you need to configure a static IP address or change the network settings, execute 'yast network' (Suse) or 'vi /etc/network/interfaces (Ubuntu).
2. The keyboard layout in the VM is Swiss German, to set it to your preference,
   Suse: *YaST -> Hardware -> Keyboard Settings* .
   Ubuntu: sudo dpkg-reconfigure -plow console-setup
3. Timezone

```
dpkg-reconfigure tzdata
```

4. To get the latest revision (bug fixes) from the release, in the directory */opt/nac* just type 'svn update'.

Connect to the Web Interface

1. To connect to the Web GUI, open your favorite web browser and type in the address bar the IP of your VM (which you seen with 'ifconfig' above).
2. If the web page is displayed correctly, then the VM is running, on the network, and Apache is running fine too.
3. On the web page there should be a summary of documentation, links to the web gui and a copy of instructions on this page.
4. Click on the "Web GUI" link. Its features are described in the User Guide .
   Note that this interface is accessing the "opennac" database, which is empty after a fresh VM install.

Install the Windows Interface

The Windows GUI is the primary, recommended, method of administering FreeNAC .

Please follow the instructios in the section Installing the Windows Interface.

See the Users Guide for a description of how to use the Windows interface.

Connecting Switches to FreeNAC

Point your switch(es) to the FreeNAC VM (see the technical Guide (http://freenac.net/en/techguide).

You may also need to define your SNMP RO/RW communities in your switches. Some scripts (namely cron_restart_port.php) require to know these communities in order to operate properly. Once you've defined your SNMP communities on your switches, edit as root the file */opt/nac/etc/config.inc*, find the lines:

```
## SNMP communities
$snmp_ro="PASSWORD2";            # maybe public
$snmp_rw="PASSWORD3";            # very private
$router_ro="PASSWORD4";           #For router_mac_ip script. If empty, use snmp_ro
defined above
```

and change PASSWORD2, PASSWORD3 and PASSWORD4 accordingly.

Once you've done that, check for the existence of the file */opt/nac/bin/cron_restart_port.pid* and if it exists, delete it

```
rm /opt/nac/bin/cron_restart_port.pid
```

Observe syslog and play with the GUIs.

Using FreeNAC in 802.1x mode

to do

1. Linux platform Installation

This section describes how to install Linux and the components needed for NAC.

Overview

The steps involved in preparing the Linux servers are:

1. Install Linux
2. Install additional key components: such as PHP, MySQL etc.
3. Install optional components: for 802.1x support Samba and FreeRadius are also needed.
4. Harden: disable unneeded services.

5. Configure linux: configure email, syslog, time synchronisation, DNS, shell profile, system monitoring scripts

Once Linux is installed, the FreeNAC software needs to be installed, and the Linux components & FreeNAC configured.

Linux

The exact name of required packages is distribution specific. The following is the required package list for FreeNAC, extracted from the Ubuntu package. Look for the equivalencies to those packages according to your distribution.

libwrap0, apache2, mysql-client-5.0, libapache2-mod-php5, apache2.2-common, apache2-utils, php5-common, ucf, libaprutil1, php5-mysql, libdbi-perl, libmysqlclient15off, libplrpc-perl, mysql-server, libdbd-mysql-perl, mysql-server-5.0, libnet-daemon-perl, libapr1, libexpat1, libxml2, libpcre3, libpq5, apache2-mpm-prefork, mysql-common, flex, python-dev, apt-file, libsnmp-base, libsnmp9-dev, mailx, nmap, openssh-server, zip, unzip, ncurses-dev, libfreetype6-dev, libjpeg-dev, libpng12-dev, apache2-prefork-dev, php-pear, php5-snmp, libxml2-dev, graphviz, subversion, php5-sybase

Basically you need LAMP, some graphics libraries for the Web GUI and SNMP.

For 802.1x support Samba and Freeradius is needed.

For connections to MS-SQL DBs, such as ePO or Wsus, FreeTDS is needed.

Linux Installation notes: Suse

A. Introduction

This section describes Suse (version 9.3) specific commands.

- this section has not been updated since V2.1 in summer '06 (The core team now uses ubuntu).
- Suse 9.3 is old, you should be using a more recent release.
- A nice, c**omplete step-by-step installation guide for FreeNAC v2.1 and Suse v10** has been contributed in Word format, see http://opennac.svn.sourceforge.net/viewvc/*checkout*/opennac/branches/2....

B. Installing Suse packages

Packages to install: rcs xntp sharutils tcpdump iptraf whois nmap automake gcc ethereal rsync lynx links pin scanlogd rsync uudeview ltrace smartmontools zip unzip pcre net-snmp ntop arpwatch perl-dbi flex pytn python-dev

a) via the nework
Yast -> Network services -> proxy
http://YOUR.PROXY.COM:80/

Set Patch source 9.x in Switzerland
http://mirror.switch.ch/ftp/mirror/SuSE/suse/
Install source 9.x:
http://sunsite.cnlab-switch.ch//ftp/mirror/suse/suse/i386/9.3/
sunsite.cnlab-switch.ch /ftp/mirror/suse/suse/i386/9.3/

yast -i
yast online_update

b) or, if you have no internet access,
by downloading the Suse 9.3 ISO images to /opt/install/suse9.3
and then mounting/unmounting a CD as needed:
umount /mnt/cd
mount -o loop -t iso9660 /opt/install/suse9.3/cd1.iso /mnt/cd

In Yast, set the install source to the local directory "/mnt/cd".

C. Linux preparation

Create /etc/mods (documentation of system changes) and "chmod 600" it

/etc/hosts : timehost, loghost, mailhost

rcSuSEfirewall2 stop
chkconfig SuSEfirewall2 off
chkconfig SuSEfirewall2_init off
chkconfig SuSEfirewall2_setup off

rcportmap stop
chkconfig nfs off
chkconfig nfsboot off
chkconfig portmap off

chkconfig mdnsd off
rcmdnsd stop

optional
vi /etc/snmpd.conf [enable a read-only community if you want SNMP monitoring]
rcsnmpd start
chkconfig snmpd on

Disable powersaving on servers and especially VMs:
/etc/sysconfig/powersave/cpufreq
POWERSAVE_CPUFREQD_MODULE="off"

Email
Yast -> Network services -> mail transfer agent
Outgoing mail server = [YOUR_OUTBOUND_SERVER]

vi /etc/aliases, and set "root" alias to the sysadmin
newaliases

Test email:
echo test | mailx -s "test" root

Time sync
cp /etc/localtime /etc/localtime.orig
cp /usr/share/zoneinfo/Europe/Zurich /etc/localtime [Switzerland]
cron:
0,30 7-20 * * 1-5 /usr/sbin/ntpdate -s A.B.C.D X.Y.Z.Z; /sbin/hwclock --systohc

Setup syslog for centralised logging to the master server:
In /etc/hosts, add an entry for each NAC server
XX vmps1
YY vmps2

On the Master, enable the syslog server:
vi /etc/syslog-ng/syslog-ng.conf.in

uncomment to process log messages from network:
#
udp(ip("0.0.0.0") port(514));

SuSEconfig
rcsyslog restart

Slave: syslog client:
/etc/syslog-ng/syslog-ng.conf.in
Forward *.info to loghost
filter f_info { level(info) ; };
destination network { udp("loghost" port(514)); };
log { source(src); filter(f_info); destination(network); };

add loghost to the vmps2 line in /etc/hosts

SuSEconfig
rcsyslog restart

change the root GECOS field in /etc/passwd to "root MACHINE"

Also check: /root/.ssh/authorized_keys

naming:
vi /etc/resolv.conf

If you use DNS domains with ".local", then replace dns library since Suse does not like domains ending int ".local". Backup libresolv.so.2 and create a new /lib/libresolv.so.2.orig that is not so brain dead:
cd /lib cp libresolv.so.2 libresolv.so.2.orig
cat libresolv.so.2.orig |sed 's/local/lokal/g' > libresolv.so.2.NO_LOCAL
cp libresolv.so.2.NO_LOCAL libresolv.so.2

If SSH logins seem very slow, you might have to replace LOCAL with 127.0.0.1 in /etc/hosts.allow for the sshd entry.

D. additional extras

create /secure check_disk, monitor_processes, secure.conf
ln -s /usr/bin/perl /bin/perl

Environment
copy /etc/profile.local from another machine
. /etc/profile.local

Setup filewatch
mkdir -p /var/filewatcher/archive
copy /usr/local/bin/filewatcher from another machine
copy /etc/filewatcher.conf from another machine
filewatcher -c /etc/filewatcher.conf
Setup Cron entry:
2 6-18 * * 1-5 /usr/local/bin/filewatcher -c /etc/filewatcher.conf

check_disk in root cron
*/3 * * * * /secure/check_disk 90 800

Linux Installation notes: Ubuntu

Introduction

There is a Ubuntu package which takes care of all dependency packages. You don't need to install or compile any other package.

Procedure

Install Ubuntu, enable the SSH server, ensure that the network connectivity is OK and update with the latest patches.

Login via SSH, and do an sudo to root. If you do not have a root shell, then all commands in the installation will need to be prefixed with 'sudo'.
Note -if you are using the FreeNAC virtual machine: The password sudo will ask for is the password of the logged on user, in the case of the VM - freenac.

Modify the installation sources by un-commenting the lines starting with deb from the /etc/apt/sources.list file and comment out the lines with deb cdrom.

```
vi /etc/apt/sources.list
apt-get update
```

Installing the Ubuntu/Debian package

Get the package from the downloads section.

Since the package is a simple deb and not embedded in a repository, dependency handling can not be done by apt. The dependency list must be extracted from the deb and fed to apt manually to install all needed packages.
NOTE: Even if you want to get FreeNAC from Subversion, rather than installing the Ubuntu package, the package can still be used to nicely install all dependencies.

```
$ sudo dpkg -f freenac_*deb depends | sed -e 's/,//g' | xargs sudo apt-get -qy
install
```

That command installed MySQL and all other packages you need. Nice!

Now install the freenac package:

```
$ sudo dpkg -i freenac_*deb
```

There are several packages that may be useful for system administration, these are not required for FreeNAC though: see also the Packages section below.

```
$ sudo apt-get install rcs iptraf whois links uudeview arpwatch screen zip unzip

Finally, insure that the latest version of package are installed:

$ sudo apt-get upgrade
```

You may now skip to the next chapter in the Installation Guide.

Notes

From the installation, you should have set your time zone properly. In case you haven't, copy from the */usr/share/zoneinfo* directory the file that best suits your timezone.

In our case:

```
sudo cp /etc/localtime /etc/localtime.orig ; #create a backup of the original
timezone
sudo cp /usr/share/zoneinfo/Europe/Zurich /etc/localtime ; # timezone of
Switzerland
```

Packages used by FreeNAC

As part of the FreeNAC installation, the following packages are **required,** (see also ./contrib/package_files/control):.

> libwrap0 apache2 apache2-prefork-dev
> libapache2-mod-php5 apache2.2-common apache2-utils php5-common ucf libaprutil1 php5-mysql libdbi-perl libmysqlclient15off libplrpc-perl
> php-pear php5-snmp libxml2-dev php5-sybase
> mysql-client-5.0 mysql-server libdbd-mysql-perl mysql-server-5.0 mysql-common
> libnet-daemon-perl libapr1 libexpat1 libxml2 libpcre3 libpq5
> apache2-mpm-prefork flex python-dev
> apt-file mailx nmap openssh-server zip unzip subversion
> libsnmp-base libsnmp9-dev ncurses-dev libfreetype6-dev libjpeg-dev libpng12-dev graphviz

The following packages, are **recommended:**

- freetds-dev (for MS-SQL connections: Wsus, Epo..)
- syslog-ng (for syslog/primary servers: offers more fine grained control)

The following packages, are **optional:**

- rcs (for file level revision control)
- tcpdump (troubleshooting)
- iptraf (troubleshooting)
- whois (troubleshooting)
- rsync (backups)
- lynx (downloads)
- links (downloads, web gui testing)
- uudeview (uuencode can be useful)
- ltrace (system level debugging)
- arpwatch (optional network monitoring Layer 2)
- smartmontools (hard disk monitoring)
- traceroute (troubleshooting)
- sharutils
- screen (multiple logon sessions)
- zip
- unzip

Compiling key non-FreeNAC components from sources

Introduction

This document explains how to compile key components from source, if needed. It is recommended to use the packages that are included with your distribution if possible, since automated updates will be easier.

It was last updated in Mar'07, and refers to versions available on that date.

Download

You'll need to download the packages, always use the latest releases, the following are example URLs.

http://mirror.switch.ch/ftp/mirror/apache/dist/httpd/httpd-2.2.2.tar.gz
http://mirror.switch.ch/ftp/mirror/mysql/Downloads/MySQL-5.0/mysql-5.0.2...
ftp://fr.rpmfind.net/pub/libxml/libxml2-2.6.23.tar.gz
http://www.ibiblio.org/pub/Linux/ALPHA/freetds/stable/release_candidates...
http://ch2.php.net/get/php-5.2.0.tar.bz2/from/this/mirror

Apache

cd /opt/install
tar xvzf httpd-2.2.2.tar.gz
cd httpd-2.2.2
./configure --prefix=/usr/local/apache2 --enable-so
make install
ln -s /usr/local/apache2 /usr/local/apache
ln -s /usr/local/apache2/bin/apachectl /etc/init.d/apache2
ln -s /usr/local/apache2/bin/apachectl /sbin/rcapache2

Actually start apache if you intend using the web interfaces, see below:
chkconfig apache2 on
/etc/init.d/apache2 start

MYSQL 5

Prerequisites: ncurses-devel gcc-c++

cd /opt/install
tar xvzf mysql-5.0.27.tar.gz
cd mysql-5.0.27
./configure --prefix=/usr/local/mysql-5.0.27 --localstatedir=/mysqldata --with-unix-socket-path=/var/lib/mysql/mysql.sock
make install

cd /usr/local
mv mysql mysql.$$ [in case you have a link already]
ln -s mysql-5.0.27 mysql
ln -s /usr/local/mysql/bin/mysqld_safe /usr/local/mysql/bin/mysql

Create a mysql user:
groupadd mysql
useradd -g mysql mysql

Create an empty database:
cd /usr/local/mysql
bin/mysql_install_db --user=mysql
mv data /var/lib/mysql

ln -s /var/lib/mysql data
ln -s /var/lib/mysql /mysqldata
ln -s /var/lib/mysql/mysql.sock /tmp/mysql.sock

Set permissions:
chown -R mysql:mysql /mysqldata /var/lib/mysql

libxml

cd /opt/install
tar xvzf libxml2-2.6.24.tar.gz
cd libxml2-2.6.24/
./configure --prefix=/opt/libxml2 && make install

FreeTDS

If you need to access MS-SQL or Sybase Enterprise databases.

cd /opt/install
wget http://www.ibiblio.org/pub/Linux/ALPHA/freetds/stable/release_candidates...
tar xvzf freetds-0.64RC2.tar.gz
cd freetds-0.64RC2
./configure --prefix=/opt/freetds --enable-msdblib
make install
vi /opt/freetds/etc/freetds.conf and add a definition to a DB to test:
[sms] <-- alias name
host = MyServer.mydomain.com <-- sever name/IP
port = 1433
tds version = 4.2
dump file = /var/log/freetds.log
dump file append = yes
#debug level = 10
debug level = 3

Try connectivity:
/opt/freetds/bin/tsql -S [alias] -U [user] -P [password]

PHP5

Install first the prerequisites packages, PHP is built with many options enabled: gd-devel freetype2-devel zlib-devel libpng-devel libjpeg-devel
net-snmp net-snmp-devel tcpd-devel rpm-devel
openssl openssl-devel openldap2-devel graphviz

cd /opt/install;
tar xBf php-5.2.0.tar.bz2
cd php-5.2.0

If you need MS-SQL (its best to assume you do - FreeTDS was compiled above)
./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql=/usr/local/mysql --with-mysql-sock=/var/lib/mysql/mysql.sock --prefix=/opt/php-5.2.0 --with-xml --with-libxml-dir=/opt/libxml2 --enable-pcntl --enable-force-cgi-redirect --with-mssql=/opt/freetds --with-gd --with-zlib-dir --with-ttf --with-freetype-dir --with-snmp=/usr --enable-ucd-snmp-hack --with-ldap

make install

Disable any current php binaries, and enable the new ones:

mv /usr/bin/php /usr/bin/php.$$
mv /opt/php5 /opt/php5.$$
ln -s /opt/php-5.2.0 /opt/php5
ln -s /opt/php5/bin/php /usr/bin/php

Test PHP:
php -v

Note:

- On Suse Linux, PHP may complain about MySQL libraries, it may be best to compile MySQL from source (see above), rather than using binary packages.
- If MS-SQL support (via TDS) is not needed in PHP, the build line is simpler: ./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql=/usr/local/mysql --prefix=/opt/php-5.2.0 --with-xml --with-libxml-dir=/opt/libxml2 --enable-pcntl --enable-force-cgi-redirect --with-gd --with-zlib-dir --with-ttf --with-freetype-dir --with-ldap
- To allow easier upgrading, rollback or testing new PHP modules, we install into a version specific directory like '/opt/php5.2.0' above, and create links to this directory.

Enable PHP in apache:

Edit your httpd.conf (e.g. /usr/local/apache/conf/httpd.conf) to load the PHP module

```
LoadModule php5_module modules/libphp5.so
```

The path on the right hand side of the LoadModule statement must point to the path of the PHP module on your system. Then "make install" from above may have already added this for you, but be sure to check. Also, tell Apache to parse certain extensions as PHP in httpd.conf
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps

2. Installing the FreeNAC software

This page discusses the different ways to obtain and unpack (if it applies) the FreeNAC software. If you want to install/configure the software, please see http://freenac.net/en/installguide/server

Install the software from a 'tarball'

Download the latest tarball from the downloads section to the /opt/ directory. If such a directory doesn't exist, create it. To uncompress this file, in the command line type the following:

```
cd /opt
tar xvzf freenac-X.Y.rcZ.tar.gz
```

This should create a directory and a symbolic link pointing to this directory. The directory contains all software required to perform a successful installation/configuration.

Install the software from Subversion

Theck FreeNAC out directly from Subversion, see the instructions from the download page. See also the SVN tips on configuring ~/.subversion/servers if you are behind a proxying firewall. For example:

mkdir /opt/nac
svn co https://opennac.svn.sourceforge.net/svnroot/opennac/branches/3.0 /opt/nac

Installing the Ubuntu/Debian package

See the Linux Installation notes: Ubuntu section.

3. FreeNAC Server - initial configuration

This sections describes the instllation of the master server components.

0. MySQL configuration

MySQL settings

General

Ensure that mysql starts automatically (e.g. 'chkconfig mysql on' on RedHat/Suse systems or 'update-rc.d mysql defaults' on Debian based systems).

Add the path to 'mysql' to your PATH for ease of use.

Set a softlink "/mysqldata" to point to the mysql database directory, for example '/var/lib/mysql'. In most of the documentation we refer to /mysqldata for brevity.

```
ln -s /var/lib/mysql /mysqldata
```

my.cnf

Compare your /etc/my.cnf (or /etc/mysql/my.cnf) with /opt/nac/contrib/etc/my.cnf, for parameters that may need to be set in the [mysqld] section.

The most important parameters to check are:

log-bin and report-host to include hostname. On the master this might be vmps1, on secondaries vmps2/3 etc.:

```
log-bin = vmps1-bin
log-warnings
report-host = vmps1
server-id       = 10                    [10 for master, 20 for slave1, 20 for
slave 2 etc..]
relay-log=vmps1-relay-bin
replicate-do-db= opennac
replicate-wild-ignore-table= opennac.vmpsauth%
```

On Ubuntu 7.10, **log-bin** is configured with the full path, and should include the hostname. It may also be called log_bin, not log-bin:

```
log-bin = /var/log/mysql/vmps1-bin.log
```

Consider increasing the connection **timeouts** to avoid spurious deconnection on low traffic networks, add the following:

```
interactive_timeout = 604800
wait_timeout = 604800
```

MySQL needs to be listening to the network on port 3306, but it might be bound only to **localhost** (e.g. Ubuntu default). Check the parameter *bind-address* and comment it out:

```
#bind-address = 127.0.0.1
```

Each server can insert data locally, changes are replicated to other servers and the changes do not conflict. Datasets must be configured with autoincrement keys, and the autoincrement value set differently on each server - thus avoiding replication conflicts. An auto_increment_increment value of 5 allows a maximun of 5 servers. **Each server must have a different auto_increment_offset** (1 for the first, or main server, 2 for the second, etc.)

```
auto_increment_increment= 5
auto_increment_offset   = 1    [1 for vmps1, 2 for vmps2, 3 for vmps3 ...]
```

Permissions

Ensure the mysql user can write to the database files (this is usually the case).

```
chown -R mysql /mysqldata /var/lib/mysql
```

Restart

Ensure that /etc/init.d/mysql exists, and automatic start is enabled. Finally, restart mysql in order to take into account the modifications you made to my.cnf:

```
/etc/init.d/mysql restart
```

You can check that mysql is running by looking at netstat, and verify that mysqld is now bound on 0.0.0.0 and not 127.0.0.1 only:

```
$ netstat -anp|grep mysql
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
5666/mysqld
```

Initial FreeNAC data-set

Extract the SQL scripts

```
cd /mysqldata
cp /opt/nac/contrib/opennac_db.tar.gz .
tar xvzf opennac_db.tar.gz
```

Create an empty dataset (new masters *only*)

For a new master server: Install an initial set of empty FreeNAC tables for the 'opennac' database, backing up the existing tables first (**Note:** You may need to prefix each command with sudo, depending on the permissions of the directory. And during the first install, you do not have an opennac db to backup :-)):

```
cd /mysqldata
cp -R opennac opennac.$$

mysql -u root -p -e "create database opennac;"
mysql -u root -p opennac < tables.sql
mysql -u root -p opennac < values.sql
```

Configuring database permissions

As of v2.2 RC3, we provide a permissions.sql file, so you don't have to worry about setting permissions by hand.

```
cd /mysqldata
mysql -u root opennac < permissions.sql
```

check /mysqldata/localhost.err for errors. (or whereever your log file resides, i.e. /var/log/mysql.err or syslog - 'grep mysqld /var/log/syslog')

Login to sql to check connectivity:

```
mysql opennac
          show tables;
          select * from port;
```

Configure mysql users for local PHP scripts (IMPORTANT)

By default the permissions script above, and the default config.inc use the password 'PASSWORD2' to connect to the database and thus be able to run the daemons.

It is **important** for security to change the passwords from the default values.

Connect first as root to the mysql database:

```
mysql -u root -p mysql
```

Then execute the following commands to change the passwords:

```
SET PASSWORD FOR inventwrite@localhost=PASSWORD('NEW_PASSWORD2');
SET PASSWORD FOR inventwrite@'%'=PASSWORD('NEW_PASSWORD1');
```

NEW_PASSWORD2 is the password you'll use in your config.inc file and NEW_PASSWORD1 will be used by the Windows GUI.

Regular housekeeping with cron

The cron tool is where all regular tasks are done to keep the system healthy. The following are recommended regular tasks.

The following crontab entries are for FreeNAC v3.0. For versions prior to this one, you don't need to include the .php extension at the end of the script name.

Master server: Remove 'unknowns' from the DB, that were never authorised and are very old:

```
0 1 * * 1              /opt/nac/bin/purge_unknowns.php
```

Clean mysql logs on the 1st per month. In this example, the absolute path of the mysql binary file is the one defined below. Please adjust the path according to your system.

```
0 6 30  * 1     /usr/bin/mysql -uroot -e "PURGE MASTER LOGS BEFORE DATE_SUB( NOW(
), INTERVAL 30 DAY);"
```

Optional: The following are scripts to backup the system in different ways to the second internal disk. These are highly system specifc, make sure you understand, tune and test them (e.g. you will need a '/disk2' partition). Remember to adjust any path according to your system.

```
0   3    * * 1-5 /opt/nac/bin/dump_ports.php
0   3    * * 1   /usr/bin/mysqlhotcopy --allowold --keepold --regexp=".+"
/disk2/backups/mysql 2>&1 | logger
```

Adapt the MySQL path to your distribution

Database rights [Old: for versions prior to 2.2 RC3]

The following has to be done in the event that you don't have a permissions file (releases prior to 2.2RC3)

There are 3 mysql users needed for accessing the database.

A. Local daemon user for PHP scripts: inventwrite@localhost
B. A user for the remote Delphi Windows GUI: inventwrite@'%'
C. Root is used by the sysadmin for local configuration. By default root is only allowed from localhost, and has no password. Its is recommended that you set a root password for mysql root, if the NAC server login is accessible to several users.

Local daemon user for PHP scripts (set the user/pw in /opt/nac/config.inc):

```
grant SELECT,INSERT,UPDATE        ON opennac.*       to inventwrite@localhost
IDENTIFIED by 'PASSWORD2';
SET PASSWORD FOR inventwrite@localhost = OLD_PASSWORD('PASSWORD2');
grant SELECT,INSERT,UPDATE,DELETE ON opennac.systems to inventwrite@localhost;
grant CREATE TEMPORARY TABLES     ON opennac.*        to inventwrite@localhost;
grant ALL                         ON opennac.vmpsauth to inventwrite@localhost;
```

Remote delphi Windows GUI user. See also the vmps.ini file on the Windows client.

```
grant SELECT,INSERT               ON opennac.*       to inventwrite@'%' IDENTIFIED
by 'PASSWORD1';
  SET PASSWORD FOR inventwrite@'%' = OLD_PASSWORD('PASSWORD1');

  grant SELECT,UPDATE               ON opennac.oper    to inventwrite@'%' ;
  grant SELECT,UPDATE               ON opennac.config  to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.building to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.location to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.port    to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.switch  to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.vlan    to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.systems to inventwrite@'%';
  grant SELECT,INSERT,UPDATE,DELETE  ON opennac.users   to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.patchcable to inventwrite@'%';
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.vlanswitch to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.cabletype  to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.sys_class to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.sys_class2 to inventwrite@'%' ;
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.sys_os        to inventwrite@'%';
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.sys_os1       to inventwrite@'%';
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.sys_os2       to inventwrite@'%';
  grant SELECT,INSERT,UPDATE,DELETE ON opennac.sys_os3       to inventwrite@'%';
```

Changing the mysql root password

We normally leave a blank password and expect a dedicated server to be used for FreeNAC. Scripts also expect a balnk password.

Optional: If the NAC server is not exclusively used by one administrator, you may want to set a local root password for mysql. This make administratig more difficult though, and some cron scripts will need to be adapted to provide a password.

```
mysqladmin -u root password 'new-password'
mysqladmin -u root -h MYHOST password 'new-password'
```

1. MySQL replication

Introduction

This document explains how to setup MySQL replication between master and slaves.

References: See also http://dev.mysql.com/doc/refman/5.0/en/replication-howto.html

Since FreeNAC Version 3.0.1, the MySQL database is configured to run in a so called 'multiple-master' scenario, meaning that each server is both a master and a slave in MySQL terminology. So each server queries updates from others (a slave), and makes any updates which were made to its dataset available to other servers (master).

Therefore a replication must be setup in each direction, for each server. Lets assume we have two servers vmps1 (our 'main' or primary server) and vmps2.
It is possible to have more than two servers (using the mysql relay_log), but this has not been tested or documented in FreeNACA yet.

The procedure is basically as follows:

First get vmps1 (the main server) running, with actual data.

A) configure vmps1 to share its data, copy an initial dataset to vmps2, configure vmps2 to retrieve updates via replication

B) configure vmps2 to share its updates, and vmps1 to retrieve these via replication

Replace the following in the examples below:

```
SERVER2.DOMAIN       the FQDN of your slave
repl                 Replication username
REPL_PASSWD          Replication password
opennac              Name of your database (this was 'inventory' prior to NAC
v2.2).
```

A. Initial vmps1 --> vmps2 replication

Initialisation

"vmps2" is a MySQL slave, and "vmps1" is a MySQL master.

Allow vmps2 the right to get replication updates from vmps1.
Note: it is important the master name corresponds to the DNS name in the GRANT statement below, otherwise use its IP address. Check /mysqldata/mysqld.log for errors.

```
GRANT SELECT, PROCESS, FILE, SUPER, REPLICATION CLIENT, REPLICATION SLAVE ON *.*
TO 'repl'@'vmps2' IDENTIFIED BY 'REPL_PASSWD';

SHOW MASTER STATUS;
```

Purge unneeded logs on the master:

```
PURGE MASTER LOGS TO 'SERVER-bin.NUMBER'
```

[the exact name comes from the File field in the 'show master status' above]

Copy initial data-set

0) On the slave, vmps2

```
stop slave;
```

1) On the master, vmps1: Lock the tables, note log position, restart

```
mysql> FLUSH TABLES WITH READ LOCK;

vmps1:$ cd /mysqldata; tar cvf opennac.tar opennac

mysql> SHOW MASTER STATUS;
+------------------------+----------+--------------+------------------+
| File                   | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------------+----------+--------------+------------------+
| vmps1-bin.000027       | 12717436 |              |                  |
```

==> take note of the position

```
mysql> UNLOCK TABLES;
```

2) Slave vmps2:

Stop mysql

```
/etc/init.d/mysql stop
```

Copy DB tar file from master & extract:

```
cd /mysqldata && mv opennac opennac.$$
scp vmps1:/mysql/opennac.tar .
tar xvf opennac.tar
chmod 770 opennac; chmod g+s opennac; chown -R mysql:mysql opennac;
```

Configure slave: start daemon with slave off

```
 /usr/sbin/mysqld --skip-slave-start --log-warnings &
```

Start replication

Start mysql client (on vmps2):

```
mysql> reset slave;
```

CHANGE MASTER: replace XXXX, YYYY, ZZZZ and 'FILE_NAME' with the values from the 'show master' above:

```
mysql> CHANGE MASTER TO MASTER_HOST='vmps1', MASTER_USER='repl',
MASTER_PASSWORD='YYYY', MASTER_LOG_FILE='FILE_NAME', MASTER_LOG_POS=ZZZ;
```

Start replication:

```
START SLAVE;
show slave status \G;
```

Check the log position with that on the master:

```
show master status;
```

Empty the vmpsauth table, which is the only local table:

```
DELETE FROM opennac.vmpsauth;
```

Also check the slave mysql log (or syslog) for errors.

If all looks fine, stop the slave:

```
/etc/init.d/mysql stop

Check with 'ps' to make sure mysql is dead, other use 'kill' with the PID of the
mysqlprocess.
Then start mysql normally

/etc/init.d/mysql start
```

If vmps is configured already, restart that too. If this is a first time installation, wait.

```
/etc/init.d/vmps restart;
/etc/init.d/postconnect restart;
tail -f /var/log/messages | grep vmpsd_external
```

B. Initial vmps1 <-- vmps2 replication

Initialisation

"vmps1" is a MySQL slave, and "vmps2" is a MySQL master.
Note: it is important the master name corresponds to the DNS name in the GRANT statement below, otherwise use its IP address. Check /mysqldata/mysqld.log for errors.

On the vmps**2** mysql prompt:

GRANT SELECT, PROCESS, FILE, SUPER, REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl'@'**vmps1**' IDENTIFIED BY '**REPL_PASSWD**';

SHOW MASTER STATUS;

Examining logged SQL queries/updates on vmps2

Now vmps2 is already humming along (due to the procedure in section A.) with a copy of vmps1's data, and is retrieving vmps1 update via replication. Since that there may have been updated to vmps2 though.

To see what logs vmps2 has, the name of the current log and position:

```
show binary logs;
```

```
show master status;
```

Now lets look at the updates in the current log:

```
show binlog events limit 20;
```

This will show the most recent 100 SQL statements that are pending, allowing you to verify that they make sense.

Enable replication client on vmps1

Start the mysql client and tell the replication to start at the initial position of the log on vmps2 (see also the output from the *show master status* on vmps2)

mysql> reset slave;

mysql> CHANGE MASTER TO MASTER_HOST='**vmps2**', MASTER_USER='repl', MASTER_PASSWORD='**REPL_PASSWD**', MASTER_LOG_FILE='**vmps2-bin.000001**', MASTER_LOG_POS=**1**;

Start replication:

```
start slave;
    show slave status \G;
```

Verify that the master log position is correct, Slave_IO_Running: Yes and Slave_SQL_Running: Yes. Last_Error should be empty.

Check the log position with that on vmps2:

```
    show master status;
```

Check the mysql log (/mysqldata/mysql.log or syslog) for errors.

Double check replication: on vmps2, insert some data

```
insert into naclog set what='test2';
select * from naclog order by id desc limit 10;
```

on vmps2, see if it appears as expected:

```
select * from naclog order by id desc limit 10;
```

The id of the inserted row should have an increment offset of 2.

Notes: Fixing a replication problem

It has happened to us that replication stops due to an invalid query.
Replication is OK on a slave if

```
   show slave status \G;
```

reports that the master log position is correct, Slave_IO_Running: Yes and Slave_SQL_Running: Yes. Last_Error should be empty.

For example, lets say Slave_SQL_Running was 'No'. To see why examine the Last_Error entry which may list the SQL entry causing the problem and then the mysql log (/mysqldata/mysql.log or syslog).

Lets assume that you understand the SQL statement, decide its not a big problem and just want to ignore that statement. So we fix it, by stopping the
slave and skipping the SQL Query causing the problem:

```
  stop slave;
  SET GLOBAL SQL_SLAVE_SKIP_COUNTER=1;
  start slave;
  show slave status \G;
```

It now skips to the next error, for example:

```
  Slave_SQL_Running: No
  Last_Error: Error 'Unknown table 'opennac.v_1'' on query. Default database:
'opennac'. Query: 'DROP   VIEW v_1'
```

Pending log events can also be examined:

```
show binlog events limit 100;
show warnings;
```

To get through these difficult queries, it may be necessary to repeat the above.

More reading:
http://dev.mysql.com/doc/refman/5.0/en/set-global-sql-slave-skip-counter...
http://dev.mysql.com/doc/refman/5.1/en/replication-options.html

4. Monitoring replication

Activate monitor_mysql_slave - call it from cron on all servers (since all servers are slaves), e.g. every 5 minutes during office hours:

```
*/5  7-18 * * 1-5 /opt/nac/bin/monitor_mysql_slave
```

2. Sysadmin: Syslog, Email, time sync, etc.

Syslog: Main server

The NAC main server needs to have a syslog server to collect messages locally, and from any secondary servers. The secondaries are configured to send a copy.

Its also useful, though not mandatory, for switches to send a copy of their events via syslog too.

Configure the syslog daemon to listen to the network interface for messages, e.g. by starting with the "-r" option. Syslog-ng needs a directive for the network interface. Some examples are:

- On Suse Linux, set SYSLOGD_PARAMS="-r" in /etc/sysconfig/syslog, and possibly also "udp(ip("0.0.0.0") port(514));" in 'source src' of /etc/syslog-ng/syslog-ng.conf .
- With Ubuntu, the default is sysklogd. For sysklogd, modify the line

  ```
  SYSLOGD=""
  ```

  for

  ```
  SYSLOGD="-u syslog -r"
  ```

  in */etc/init.d/sysklogd*. If you install syslog-ng in Ubuntu, locate the line

  ```
  # udp();
  ```

  in *syslog-ng.conf* and uncomment it.

  Create a symlink pointing to your syslog startup file, so if for example you are using sysklogd, do:

  ```
  ln -s /etc/init.d/sysklogd /etc/init.d/syslog
  ```

  or in case you are using syslog-ng

  ```
  ln -s /etc/init.d/syslog-ng /etc/init.d/syslog
  ```

  We assume that /etc/init.d/syslog is a valid link to your syslog, for the rotate scripts mentioned below.

Syslog: Secondary servers

Configure a syslog client to send a copy of messages to the server.

First add a 'loghost' alias to /etc/hosts, then configure syslog:

1. Classical syslog: Add the following to the bottom of /etc/syslog.conf, note that there is a tab (not a space) between the two fields.

```
*.info  @loghost
```

2. Syslog-ng: Add the following to /etc/syslog-ng/syslog-ng.conf (example for Ubuntu 7.10)

```
## Forward *.info to loghost
filter f_info        { level(info) ; };
destination network  { udp("loghost" port(514)); };
log { source(s_all); filter(f_info); destination(network); };
```

Syslog: rotation & testing

Secondary servers: use the default Linux log rotation mechanisms, or optionally, the mechanisms below.

Main server: Its important to ensure logs are regularly archived, rotated and that syslog is working as expected.

Syslog-ng: Add the following to /etc/syslog-ng/syslog-ng.conf (example for Ubuntu 7.10)

Find the following section in syslog-ng.conf

```
# all messages of info, notice, or warn priority not coming form the auth,
# authpriv, cron, daemon, mail, and news facilities
filter f_messages {
level(info,notice,warn)
and not facility(auth,authpriv,cron,daemon,mai
};
```

and change it as follows

```
# all messages of info, notice, or warn priority not coming form the auth,
# authpriv, cron, daemon, mail, and news facilities
filter f_messages {
level(info..emerg)
and not facility(auth,authpriv,cron,mail,news);
}
```

and comment out the following sections:

```
log {
source(s_all);
filter(f_syslog);
destination(df_syslog);
};

log {
source(s_all);
filter(f_daemon);
destination(df_daemon);
};
```

A log pruning configuration file is provided with FreeNAC, which focusses on /var/log/messages. This file is rotated weekly, archived for one year, two scripts /opt/nac/logcheck/logcheck.sh and /opt/nac/bin/monitor_allows_count.sh are run before-hand and all FreeNAC daemons restarted afterwards.

So, review the logrotate config file, copy it to /etc, activate and test:

a) syslog-ng (preferred)

```
mv /etc/logrotate.d/syslog-ng /etc/syslog-ng.$$
cp /opt/nac/contrib/logrotate.d/syslog-ng /etc/logrotate.d/syslog-ng
/usr/sbin/logrotate -d --force /etc/logrotate.conf
```

b) classical syslog

```
mv /etc/logrotate.d/syslog /etc/syslog.$$
cp /opt/nac/contrib/logrotate.d/syslog /etc/logrotate.d/syslog
/usr/sbin/logrotate -d --force /etc/logrotate.conf
```

Add a cron entry to prune syslog, for example on weekday mornings:

```
# Force Log pruning check each morning

0 6 * * 1-5            /usr/sbin/logrotate /etc/logrotate.conf | logger
```

Activate the syslog configuration above, and test:

```
/etc/init.d/syslog restart


echo test_syslog | logger -p local3.info -t daemon


grep test_syslog /var/log/messages
```

Email

FreeNAC tools send notifications by default to the 'nac' and 'root' user. In the file */etc/aliases* there should be an alias for nac and root that point to a sysadmin.

To delivery email beyond a local user, the mail daemon (postfix, exim, sendmail, etc.) will have to be configured to send emails via your gateways.
On Ubuntu 7.10 for example, we deinstall exim and install postfix (personal preference) and configure it:

```
apt-get install postfix
dpkg-reconfigure postfix    (Internet site with mailhost is probably what you
want)
```

To test email delivery:

```
echo test | mailx -v -s "test" root
tail /var/log/mail.info
```

Change the root GECOS field in */etc/passwd* to "root MACHINE", this makes email from headers easier to read.

Time Sync

Adjust the root crontab to update the current time from an NTP server

```
0,30 * * * * /usr/sbin/ntpdate -s A.B.C.D X.Y.Z.Z; /sbin/hwclock --systohc
```

where A.B.C.D X.Y.Z.Z are NTP servers to synchronize from.

Other optional sysadmin settings

Create the file */etc/mods* where you'll store the changes made to your system

```
touch /etc/mods
chmod 600 /etc/mods
```

Setup SSH trusts if needed: */root/.ssh/authorized_keys*

3. PHP settings

Server side programming is in PHP, so PHP5 cli (command line interface) module must be installed - see also the linux installation section.

PHP setting are stored in 'php.ini' the default that comes with your distribution should be enough (assuming its recent and up to date).

If installing PHP from the sources use the 'php.ini-recommended' file included in the distribution, or the contrib/etc_php5_cli/php.ini of FreeNAC
Recommended variables set in your /etc/php5/cli/php.ini file are as follows,

Memory

```
memory_limit = 256M      ; Maximum amount of memory a script may consume (128MB)
```

Error_reporting on productive servers: show all errors, except for notice. Log errors to syslog:

```
error_reporting = E_ALL & ~E_NOTICE
log_errors = on
display_errors = off
error_log = syslog
```

On test / development servers where you wish to see all software warnings:

```
error_reporting = E_ALL | E_STRICT
display_errors = on
error_log = syslog
```

4. FreeNAC daemons

Install the software

The Freenac software should already have been installed in the section Installing the FreeNAC software .

Create group and user

You need to create a freenac username and group. This is handy if you want the configuration file to be accesible by other daemons (e.g. Apache, Radius).

```
groupadd freenac && useradd freenac -r -g freenac
```

Configure FreeNAC

Master server: Create a config.inc from a template and set the DB connection parameters (this should correspond to the password you set in the MySQL configuration):

```
  cp /opt/nac/etc/config.inc.template /opt/nac/etc/config.inc
vi /opt/nac/etc/config.inc
```

Slave servers: copy /opt/nac/etc/config.inc from the master

Change the group the *config.inc* file and the lib directory belong to and its permissions

```
chgrp freenac /opt/nac/etc/config.inc
chgrp freenac /opt/nac/lib
chmod 640 /opt/nac/etc/config.inc
chmod -R 640 /opt/nac/lib
```

For version 2.2 RC2 and earlier: import the config file into the database. To do so:

```
cd /opt/nac/contrib
./config2db ../etc/config.inc
```

For v2.2 RC3, V3.0 and later, all settings are in the 'config' table in the database. Only usernames and passwords are in config.inc. The 'config' paramets are set via the Windows GUI (see the Administration -> config tab).

Policy (v3.0 and later)

A substantial change in FreeNAC v3.0 is the introduction of an OO (objected oriented) policy interface, which provides greater flexibility and encapsulation of individual decisions regarding access to the network.

You need to specify a policy file to use. We provide some sample policy files in the *etc* directory. In the Technical Guide there are several policy chapters, please example at least the Sample Policies .

A policy file that would be useful to many sites is the *etc/policy5.php* file, lets assume you wish to use that. Now create a link from policy file to the default policy file name 'policy.inc.php':

```
cd /opt/nac/etc
ln -s policy5.php policy.inc.php
```

Masters and slaves normally have the same policy.

Start the vmps daemon

Creating a start-up file and start the service:

```
cp /opt/nac/contrib/startup_init.d/vmps /etc/init.d/vmps
chmod 750 /etc/init.d/vmps
vi /etc/init.d/vmps       [adapt IP address on vmpsd start line, if have more than
one interface]
```

And activate it to start automatically according to your distro

```
chkconfig vmps on [SuSE]
update-rc.d vmps defaults [Ubuntu/Debian based distros]
```

Start and watch syslog for events:

```
/etc/init.d/vmps start

ps -ef | grep vmps

tail -f /var/log/messages

If vmpsd does not start, see the troubleshooting section  of the Users Guide.
```

Start the postconnect (v2.x: vmps_lastseen) daemon

On V3:

```
cp /opt/nac/contrib/startup_init.d/postconnect /etc/init.d/postconnect
chmod 750 /etc/init.d/postconnect
```

And activate it to start automatically according to your distro

```
chkconfig postconnect on     [SuSE]
update-rc.d postconnect defaults    [Ubuntu/Debian based distros]
```

Start and watch syslog for events:

```
/etc/init.d/postconnect start
tail -f /var/log/messages
```

On V2.2 and earlier:

```
cp /opt/nac/contrib/startup_init.d/vmps_lastseen /etc/init.d/vmps_lastseen
chmod 750 /etc/init.d/vmps_lastseen
```

And activate it to start automatically according to your distro

```
chkconfig vmps_lastseen on [SuSE]
update-rc.d vmps_lastseen defaults [Ubuntu/Debian based distros]
```

Testing

Watch syslog for events:

```
tail -f /var/log/messages
```

If vmpsd does not start, see the troubleshooting section of the Users Guide.
See the Policy testing chapter of the Technical Guide, to explain how to read the syslog messages.

5. Monitoring

Introduction

In production environments, monitoring and alerting of the FreeNAC server is recommended . This section discusses several such tools included in FreeNAC. You may choose to use these scripts, do nothing, or do similar monitoring with your own tools.

The monitoring scripts that need to be individually tested and enabled in root cron. In Previous chapters other cron scripts were mentioned, for example for monitoring MySQL. This section, which is addressed at the Linux adept, covers other housekeeping scripts that inducate to the system administrator if the FreeNAC system is behaving properly, or if specific switches or ports are having issues.

All of these are focussed on the **main server**, some such as process monitoring may also be used on secondary servers.

Monitoring syslog

Monitoring syslog for unusual events, is done by the logcheck which basically does a grep on the logs. See also the syslog configuration chapter.

```
0 8,12  * * 1-5        /opt/nac/logcheck/logcheck.sh
```

The following two check that a minimum number of devices are being regularly allowed onto the network (i.e. FreeNAC is actually seeing and VMPS authenticating end-devices), and that a port is not flapping between several vlans.

```
30   6-22 * * 1-5 /opt/nac/bin/monitor_allows.sh
*/4 *     * * *   /opt/nac/bin/flap_detect.php
```

Are there any 'MAC-NOT-RECONFIRMED' from switches or vmps requests with MAC 000000 that might indicate communication problems between switches and the NAC server?

```
*/10 7-18 * * 1-5 /opt/nac/bin/monitor2.sh
```

FreeNAC Updates

Notify if there are updates to NAC (query FreeNAC.net and report if there is a new version)

```
0 0 * * 0        /opt/nac/bin/updates.php
```

Process monitoring

Monitor_processes just does a grep on the process list and send an email alert if a process dies. This tool is run regularly from cron.

```
*/20 7-18 * * 1-5 /opt/nac/bin/monitor_processes.pl proctst vmpsd_external
postconnect
```

In 802.1x mode, check samba & free radius too.

```
*/20 7-18 * * 1-5 /opt/nac/bin/monitor_processes.pl winbindd smbd nmbd radiusd
```

proctst: There is an aletrnative process monitoing with the *proctst* daemon. proctst (as opposed to monitor_processes) is a daemon: it does not need cron, and not just alerts when a process dies but **also restarts** it.
With proctst there are also unexpected side effects: if you should shutdown a daemon manually, because you want to do some debugging or so, proctst immediately restarts it. e.g. You shutdown mysql, want to do backups or maintenance, in fact mysql was immediately restarted by proctst and you may not realise it is running.
So consider using proctst when there is an actual problem with a daemon dying, or for production servers where everything has to be as automated as possible.

Configuring proctst:

- Copy the example configuration contrib/etc/proctst.conf to /etc, review and adapt.
- Copy the startup file from contrib/startup_init.d/proctst to /etc/init.d.
- Start the daemon:
  /etc/init.d/proctst start
- Test that it works as expected; stop daemons, watch syslog etc.
- Then enable proc to automatically start when the system is rebooted:
  chkconfig proctst on [Suse/Redhat]
  update-rc.d proctst defaults [Debian/Redhat]

Other tools

To do: check_disk watches system load and disk space usage.

0 8-18 * * 1-5 /opt/nac/bin/check_disk 90 800

4. Installing the Windows Interface

Installation

1. Download the Windows GUI (vmps.exe and vmps.xml) and save a copy to a folder, for example called *FreeNAC*.
2. The GUI configuration file is *vmps.xml*, open this in your favourite text editor (vim, notepad, ...)

Configuration: NAC server IP address and Database name

1. Configure *vmps.xml* and set the 'mysql server' parameter to the IP address of the NAC master server.
2. By default, the GUI expects to connect to the 'opennac' dataset, which is initially empty. It will be used when FreeNAC talks to switches and routers in your environment.
   To select this database, no changes are needed to vmps.xml.

3. There is also a **sample** database "**nacdemo**" (see the contrib directory), if you wish the GUI to point to this:
   Configure *vmps.xml* and set the 'mysql database' to 'nacdemo'.
4. Optional (v3.0.1): a 'server2' parameter can be added to the mysql section to specify a secondary server. This enables a second 'connect' button in the GUI that alllows connections to the secondary server if the primary fails - i.e. to allow the GUI to continue to work if one of the servers dies.

Configuration: GUI user rights

There are two levels of authentication/authorisation:
A. MySQL authentication & authorisation: the windows GUI uses a specific user & password to connect to the DB. We call this the 'mysql user'
B. Windows GUI identification and authorisation: the GUI takes your currently windows logged-in user to identify you, and uses the value in the nac_rights field for this user to control what you can do (client-side enforcement). we call this the 'NAC user'

A. MySQL user

The MySQL user is created as part of the mysql configuration and given rights to access certain tables remotely. This user is usually called 'inventwrite' (for historical reasons). The password chosen for this user now needs to be encrypted and stored in the windows configuration file.

To verify that the inventwrite user exists, try the following SQL command:

```
select * from user where user='inventwrite';
```

If you are using the Demo Virtual Machine, the password is PASSWORD1. This, of course, should be changed in a productive environment!

Now, inform the windows GUI about which username/password it should use. The username and password is stored in an encrypted string called 'auth' in the vmps.xml configuration file.

- Starting the GUI vmps.exe
- Select Admin -> Encrypt User
- Fill in the Username and Password, and click on Generate
- Copy the value of the 'generated key' field to the 'auth' field in vmps.xml
- Then, quit the GUI

B. NAC user

The GUI takes your currently windows logged-in user to identify you, to the server. It also sends the Windows domain to the server too.

Depending on the rights of this login name in the 'users' table, the GUI will grant you access or refuse to work.

So the windows username must also exist in the NAC user table, and the user must also have a permission value set. **The permission is in the nac_rights field and can have three values** (1=readonly, 2=write, 99=administrator).

- Read-only: Users can view, but not make any changes to the GUI.
- Write: Users can make changes to the Overview and Edit tabs, but not to the Switch, Ports, Logs or Administration tabs (config, vlan)

Example: to add a user called 'smith' to the users table, with administrator permissions, the following SQL command will needed to be executed:

```
insert into opennac.users (username, Surname, GivenName, nac_rights) values
('jsmith', 'John', 'Smith', 99);

Other examples:

update users set nac_rights=1 where username='JOE';
update users set nac_rights=2 where username='BILL';
update users set nac_rights=99 where username='SUSAN';
```

Once users have been added, their permissions and other details can be changed in the GUI itself. (Administration -> Users).

Demo mode:

For demonstration purposes, there is a 'demo mode' which is enabled if the field 'DemoMode' is set to '1' in the config table on the server.

If DemoMode is=1, and the DEMO company is set in vmps.xml, then all Windows users are given administrator access, which is fine for initial testing, but must be changed afterwards.

To disable, do the following as root on the MySQL prompt:

```
update opennac.config set value='0' WHERE name='DemoMode';
```

Verification of the windows domain

- The GUI can also be restricted to a specific window domain, if the 'guidomain' field in the config table on the server is set.
- If this is set to 'MYDOMAIN' for example, then the GUI will only allow users to connect who are logged onto that domain.

Using the NAC Windows GUI

Start the GUI and press 'connect'.

See the Users Guide for a description of how to use the Windows interface.

There is also a pending bug/fixes list for the Windows GUI that you may wish to consult.

Connect to the online FreeNAC demo database

A sample database is available to play around and try out the GUI.

1. Download the Windows GUI (vmps.exe) and save a copy to a folder, for example called *FreeNACdemo*.
2. Download the demo config file vmps.xml
3. Start vmps.exe

This will try to connect via the Internet to the FreeNAC demo database, which is re-initiailised automatically every hour.
Note: this will not run behind a corporate proxying firewall, port 3306/mysql needs to be open outgoing.

Other vmps.xml parameters

The mysql-inv stanza if for the Microsoft SQL inreface to a static inventory system, if one exists. This has been used for custom installed and is not documented yet. Basically the inventory key in the systems table is used to lookup and display information from the static inventory DB and show it in the Edit tab.

Since v3, parameters for enabling modules such as StaticInvEnabled, NmapEnabled, AntiVirusEnabled, PatchCableEnabled, as no longer needed - these are now set in the config table on the server.

5. Router integration

Introduction

Routers can be queried regularly to discover the IP addresses & DNS names attributed to MAC addresses. This is an important part of the "auto discovery" of end devices.

If router_mac_ip_discoverall=true in the config table, the router_mac_ip module will document all MAC/IP pairs it finds on the network, not just those actively managed with the vmps protocol. End-devices found in this way are marked with the status "unmanaged" (see for example the overview page in the windows GUI).

See also the Installation Guide -> Learning Mode .

Configuration

Settiings are configured in two places, etc/config.inc and the 'config' mysql table. In V3 and later all settings except passwords are iun the config table.

Configuration: config.inc

This file, created from config.inc.template contains sensitive data such as passwords. For this module, set the SNMP community string for querying router settings:

```
$snmp_ro
```

Configuration: 'config' table

The 'config' table can be managed either from the mysql command line (use 'describe config' and 'select * from config' if you are at ease with SQL), or more easily from the Windows GUI (Users Guide -> Windows GUI -> Administration tab).
Note: In V2.2 and earlier, settings are in config.inc only.

There are several configuration variable that must be set.

What are the IP addresses of routers from which ARP tables are to be queried?

```
core_routers=192.168.245.3 192.168.245.6 192.168.245.30
```

Should all new IP addresses be documented, or just those already in the systems table?

```
router_mac_ip_discoverall=true
```

What IP and MAC addresses are to be ignored when querying?

```
router_mac_ip_ignore_ip= /^(127.0.0|192.168.|193.5.238)/
```

```
router_mac_ip_ignore_mac= /^(00d0.0064.d000|0008.02a1.a3b3)/
```

Should IP addresses be translated into names from DNS and updated?

```
router_mac_ip_update_from_dns=true
```

Names can also be updated from NMB (Windows naming), as opposed to DNS. Most sites should stick with DNS.

```
router_mac_ip_update_from_nmb=false
```

Installation

Adapt the settings above, then try /opt/nac/bin/router_mac_ip.php from the command line, initially increasing the debug level from 0 to 3.

```
$logger->setDebugLevel(3);
```

Check the messages tagged 'router_mac_ip' in syslog to understand behaviour. Make sure that the router queries are working, and are fast (e.g. 20 secs.). Are end-devices being added to the systems table, are they visible in the GUI?

When its works as expected, then add an entry to the root cron, for example to query the routers every 6 minutes:

```
*/6 *    * * *   /opt/nac/bin/router_mac_ip
```

6. Switch integration

Introduction

Switches are integrated in five ways:

1. 'passive' scanning of mac addresses visible on all ports
2. Querying of port status
3. Port control (stop/start/restart/set static vlan/set vmps mode)
4. Answering of vmps queries.
5. Answering of 802.1x/radius queries.

1. Passive scanning of MAC tables via SNMP

FreeNAC (v2.2 RC2 and later) includes the snmp_scan.php tool which queries the information from switches:

- Switch's hardware, software version
- Discover new Ports.
- Update port names, status (up or down), auth profile (static vlan, dynamic/vmps or trunk)
- Update the last vlan on a port, for static ports
- For each MAC address found its LastSeen, LastVlan and LastPort fields are updated, or for new MACs a new entry is added with the name 'unknown'.

The script only scans switches which have the flag *scan*=1 in the *switch* table.

Settings are configured in two places, etc/config.inc and the 'config' DB table.

Configuration: switch

SNMP needs to be enabled, and ACLs set so that queries are allowed from the FreeNAC server IP address. If there is a firewall between FreeNAC and the switches, the SNMP port (udp/161) needs to be open.

Configuration: config.inc

This file, created from config.inc.template and stored in /opt/nac, contains sensitive data such as passwords. Please set the SNMP community string for reading switch settings:

```
$snmp_ro
```

Configuration: Config table

The parameters in the 'config' table can be set either from the mysql command line (use 'describe config' and 'select * from config' if you are at ease with SQL), **or more easily, from the Windows GUI** (See Users Guide -> Windows GUI -> Administration tab and the Windows GUI installation page).

The *snmp_dryrun* setting should be false (=0).

Configuration: switch table

You need to first declare which switches are going to be scanned, either via Windows GUI (Users Guide-> Windows GUI-> Switches) or via the MySQL commandline:

```
insert into switch set ip='1.2.3.4', name='swXX', location='1';
```

For a switch to be automatically scanned, set the 'scan' flag to 1:

update switch set scan='1' where ip='1.2.3.4';

Change the values according to your system.

Activating snmp_scan

Once configured, run it from the command line to test:

```
cd /opt/nac/bin
./snmp_scan.php
```

Look at syslog to see how the tool is progressing. This tool can take some time, depending on the number of switches of your network and their age. If it times-out or take a long long time it probably mean that SNMP is not correctly configured on the switch, or the community is not correct in config.inc.

To run regularly, for example at 11:05 daily, add an entry to the root cron:

```
3 11 * * 1-5     /opt/nac/bin/snmp_scan.php | logger
```

Note: scheduling of such scans cannot be configured yet from the Windows GUI.

2. Querying of switch port status

As of FreeNAC v3.0, we introduced the tool ping_switch.php which queries the switch port status (up/down), that can be seen in the GUI.

To activate, add the switch to freenac, set the 'scan' flag to 1 (see the previous section), and add an entry to root crontab, to scan every hour for example:

```
10   8-17   * * 1-5 /opt/nac/bin/ping_switch.php 2>&1   | logger -t
ping_switch.php
```

Note: scheduling of such scans cannot be configured yet from the Windows GUI.

3. Port control

The active programming of certain parameters is possible from the Windows GUI. These parameters are stored in the database, and then written to switches by the tool cron_restart_port.php on the server.

For each port the parameters that can be set are as follows, see also the Switches section of the Windows GUI User Guide:

- restart

- clear_mac
- shutdown
- static or dynamic vlan attribution
- if static, the valn can be defined

Configuration: Set the snmp write community ($snmp_rw) in config.inc. Test cron_restart_port.php on the command line and verify results by reviewing the syslog and the 'server log' in the Windows GUI. Then activate in the root crontab e.g. every minute:

* * * * * /opt/nac/bin/cron_restart_port.php

Configuring clear_mac (new in V3.0.2): Refer to the tech guide for a description this feature. It is needed as a supplement to port_restart for Switches with more recent IOS versions. You need to do the following to activate it.

1. Configure the values for $sw_user, $sw_pass and $sw_en_pass in *config.inc*.
   These variables are needed to access the switch via telnet and call the IOS command to clean the MAC address from the switch.
   The commands to the switch are send in clear-text (via) telnet. If a dedicated switch management network is not available, this may increase security risks.
   For additional security, one can restrict the commands that the sw_user can perform on the switch e.g. only to the "*clear mac*" command. Such restricted user access is typically done on a TACACS server such as Cisco ACS.
   If you don't want to create a dedicated restricted user, your admin username should work, as well as the enable password.
2. Activate this feature: Set the value of the *config* variable *check_clear_mac* from false to true (via the Windows GUI of the mysql command line).
3. Select the IOS switches where clear_mac will be used.
   This can be done from
   A) the Windows GUI: Switch configuration page, set the'switch_type' to '1'.
   B) or in MySQL by issuing the following query:

```
mysql> UPDATE switch SET switch_type='1' WHERE
name='mysuperswitch';
```

4. Answering of vmps queries.

The main feature of FreeNAC was to originally answer VMPS requests, and answer with an ALLOW or DENY (with an associated Vlan). The answering is done by the *vmpsd_external* daemon in accordance with the configured policy (see the policy chapters of the Technical Guide).

Although individual ports can be set to static or dynamic (vmps) mode, from the Windows GUI (see previous section), **key VMPS parameters such as the following must be programmed directly on each switch manually** (via telnet or SSH).

- vmps server IP addresses
- timeouts
- reconfirmation intervals

These Parameters and how to set them for Cisco CatOS and IOS switches is covered in the FreeNAC Techincal Guide, 'Configuring Network Switches ' chapter.

5. Answering of 802.1x/radius queries

the 802.1x answers requests to authenticate end-devices based on the 802.1x protocol, typically either a User's Windows Domain logon, or a Certificate.

This involved the FreeRadius and Samba modules, and also requires manual port programming (via SSH or telnet) on the switches.

Please refer to the 802.1x section of the Technical Guide.

7. Additional modules

See the child pages for more information

Deleting old users from database

To have a more accurate users' information, the delete_old_users.php script will delete users not seen in the central directory for more than 'delete_users_threshold' days that have no systems assigned. If systems are assigned, send a warning that they need to be reassigned to someone else.

In order to install this feature, you first need to add its configuration variable in the config table.

```
INSERT INTO config SET type='integer', name='delete_users_threshold', value='365',
comment='Delete users not seen in the central directory for more than XX days';
```

Adjust the value according to your needs. This example is meant to delete users which haven't been seen in the central directory for one year.

Add the following entry in crontab

```
#Delete old users
0 1 * * 1       /opt/nac/bin/delete_old_users.php
```

This will cause this scrpt to run once a week (Monday at 1:00 AM).

If there are users to be deleted, you'll receive an e-mail summarizing the deleted users. or information of systems assigned to those users, in order to be assigned to someone else.

IP scanning with the NMAP module

How to install:

You'll need nmap v4 or later, if you don't have it, download it from www.insecure.org/nmap, or your local package source.

For versions prior to 2.2RC3, rename port_scan.inc.template to port_scan.inc

```
cp /opt/nac/etc/port_scan.inc.template /opt/nac/etc/port_scan.inc
```

and modify the configuration settings according to your needs, especially the nmap path. If you are using release 2.2 RC2 or prior, you need to import this configuration file into the database. As of release V2.2 RC2, port_scan takes all the variables from the config table (except for $debug_flag1), so the port_scan.inc file has to be also imported into the database.

Do the following from the /opt/nac/contrib directory (in case you are using 2.2 RC2):

```
./config2db ../etc/port_scan.inc
```

If you need to redefine some of these settings, you can do so through the Windows GUI.

Next, create the following directory in case it doesn't exist

```
mkdir /opt/nac/scan
```

Now you need to define the networks you'd like to scan. This can be done with Administration > NmapSubnets in the WindowsGUI, Or, in MySQL:

```
insert into subnets set ip_address='192.168.1.0', ip_netmask='24', scan='1';
```

This will add the subnetwork 192.168.1.0/24 to your subnets table and with 'scan=1' we are saying that this subnetwork can be scanned. You need to do this for every subnet you want to take into account.

If there are specific hosts you'd like to ignore, add a CSV list of the IP addresses of systems you don't want to scan.

```
update subnets set dontscan='192.168.1.1, 192.168.1.127, 192.168.1.254' where
ip_address='192.168.1.0' and ip_netmask='24' and scan='1';
```

Other settings (Administration > Config) are *scan_for_hours, time_threshold, whats_units_time, nmap_flags* and *which_nmap*. The only one which typically needs changing is scan_for_hours: only systems seen on layer 3 in this number of hours (default 3) are scanned, others are ignored. This is to make scans quicker.

Next, schedule the automatic scans. Add the following entries to the root crontab.

```
# FreeNAC: port_scanning of systems
*/5 8-18 * * 1-5       /opt/nac/bin/port_scan.php --scannow
# Active scanning: nmap monday
0 11 * * 1             /opt/nac/bin/port_scan.php
```

The first entry runs every five minutes, and it is the main mechanism to control immediate scans requested from the GUI. With the GUI you set the flag 'scannow=1' and then this cronjob will scan the systems which have this flag set.

The second entry scans all devices present in the systems table every Monday at 11:00 AM

See also the the technical guide section on the nmap module.

FreeTDS

FreeTDS component

At the time of this writing, some distributions (Ubuntu 7.10) lack a component needed for the Enterprise version of FreeNAC. Such a component is the FreeTDS client (tsql binary) needed to test connectivity with MSSQL databases.

If your distribution doesn't have this package, we **highly** recommend you to get the latest version from http://www.freetds.org/ and compile and build it yourself.

In case you don't have the expertise to perform such a task, we provide a binary (v0.64RC2) and a configuration file that will run on i386 platforms. Proceed to our downloads page in Sourceforge and get these components and install the binary under */usr/bin*

Important note: The config file must be placed under */opt/freetds/etc* in order to be found by the binary we provide.

Known issues:

Setting *connect_timeout* values for *mssql_connect()* in PHP using Ubuntu 7.10

There are some parameters in *php.ini* which allow to manipulate the timeouts (connect_timeout, query_timeout) but those are for mssql specific. In Ubuntu 7.10 there is no package *php5-mssql*.

If you do a

```
php -i | grep mssql
```

you can see that such a package is not installed.

The replacement for *php5-mssql* in Ubuntu 7.10 is *php5-sybase*, but the *php.ini* configuration for Sybase, doesn't specify any timeouts. Trying setting the values of *mssql.connect_timeout* and *mssql.timeout* doesn't work.

There is a bug report for *php5-mssql*
http://bugs.launchpad.net/ubuntu/+source/php5/+bug/8706
and the package we'll be created for the next release of Ubuntu (Hedgy). If that's the case,

then *freetds* will be needed, since *php5-mssql* will be used to perform queries to MS SQL databases.

McAfee EPO synchronization

Introduction

In an enterprise environment, McAfee "EPO server" is often used to manage client PCs, pushing anti-virus updates and ensuring that client AVs stay up to date.

Epo has an MS-SQL server in its core and the tables in this database were examined to see what information was in there that could be useful to FreeNAC users.

The FreeNAC "EPO module" queries information from the Epo regularly (e.g. each night) and stores it in dedicated tables in the FreeNAC database.

Installation

Pre-requisites: In order to install this feature, you need FreeTDS installed and of course an Epo server.

Configuration: Epo server

Create an SQL user, which FreeNAC will use to query Epo. Give this user SELECT rights on the Epo database.

Configuration: FreeNAC

1. The first step is to configure 'FreeTDS' so that SQL queries to Epo actually work.

Add the following instance to *freetds.conf* for your EPO server. Adjust the host and perhaps port for your Epo server:

```
[epo_alias]
host = server.domain.com
port = 1433
tds version = 4.2
dump file = /var/log/freetds_inv.log
dump file append = yes
debug level = 1
```

2. In the *config_en.inc* file, set up your username and password in the variables *$epo_dbuser* and *$epo_dbpass*. These corresponds to the user name and password configured on the Epo SQL server.

3. Set *epo_dbalias* and *epo_db* in the mysql *opennac.config* table

```
update config set value='epo_alias' where name='epo_dbalias';
update config set value='epo_db' where name='epo_db';
```

Here *epo_alias* is the alias you declared in your *freetds.conf* file and *epo_db* is the database that holds the information regarding the McAfee antivirus.

4. Enable the Epo module in FreeNAC:

```
update config set value='true' where name='epo_enabled';
```

If such a field doesn't exist in your config table, create it as follows:

```
insert into config set type='boolean', name='epo_enabled', value='true',
comment='Enable or disable the McAfee Epo module';
```

This flag also enables Epo related features in the FreeNAC Windows GUI.

5. Ensure that the local FreeNAC MySQL user can update the local Epo tables. Grant permissions to the EpoComputerProperties table:

```
grant SELECT,UPDATE,DELETE ON opennac.EpoComputerProperties to
inventwrite@'localhost';
```

Testing

Run the test script *epo_test.php*. If everything went fine, you'll see the output of the SQL query "SELECT TOP 5 ParentID, ComputerName, IPHostName, DomainName, IPAddress, OSType, OSVersion, OSServicePackVer, NetAddress, UserName, TheTimestamp, TheHiddenTimestamp, Description FROM ComputerProperties".

If this did not work, verify the above setting, there is probably a configuration or connectivity problem.

Next, try the EPO sync script to synchronise Epo information to the local tables.

Watch stdout and syslog for errors.

```
log |grep -i epo &
```

Edit the epo_sync.php script, and ensure that the variable

```
$EPO_VERSION = 3;              // either 3 or 4
```

represents the right EPO version you are using. Change this value accordingly.

start the sync

```
./epo_sync.php
```

Syslog entries like the following should appear:

```
Aug 27 11:34:11 freenac epo_sync[31844]: Update AV status 00114336D065
0011.4336.D065 20070827103729, 5102.0000, USER1
Aug 27 11:34:11 freenac epo_sync[31844]: Update AV status 0015C54CC15D
0015.C54C.C15D 20070827111501, 5102.0000, USER2
Aug 27 11:34:11 freenac epo_sync[31844]: Update AV status 0019D139EB34
0019.D139.EB34 20070802153610, 5087.0000, USER3
```

Now, in the FreeNAC Windows GUI, Epo information should be visible in

Reports -> AV, and for individual end devices in Edit->Anti-Virus.

Operations

If the previous steps went ok, add it to the root cron for regular Epo synchronisation.

```
30  3    * * 1-5 /opt/nac/bin/epo_sync.php
```

In Windows GUI, updated Epo information should be visible in

Reports -> AV, and for individual end devices in Edit->Anti-Virus.

Microsoft SMS

Introduction

In an enterprise environment, Microsofts Systems Management Server (SMS) may be used to manage client PCs Software packages, pushing SW package updates, querying inventory and ensuring that client packages stay up to date.

MS-SMS has an MS-SQL server in its core certain information in there, such as MAC addresses and system names may be useful to FreeNAC.

The FreeNAC SMS class "SMSEndDevice.php" is usually called by "postconnect", and it queries information from its local SMS copy, to find out if the MAC address of an unknown device currently trying to connect to the LAN is in the SMS database.

postconnect then allows the device, if information is found, and updates the FreeNAC tables with the information from SMS, such as logged on user, system name etc.

If you plan to use this module, you need to modify the postconnect section of your policy file accordingly. A sample snippet is shown below:

```
 /**
* The postconnect method is used by the postconnect daemon.
* It updates information for PORTS and HOSTS
* This method writes to the database, so it shouldn't be called from a slave
server.
* @param object $REQUEST     A SyslogRequest object
*/
public function postconnect($REQUEST)
{
# SMS module
$SMS_HOST=new CallWrapper(new SMSEndDevice($REQUEST));
#Insert End device if unknown
$SMS_HOST->insertIfUnknown();
#Insert a switch or port if unknown
$REQUEST->switch_port->insertIfUnknown();
#Update port information
$REQUEST->switch_port->update();
# Update host lastseen timestamp
$REQUEST->host->update();
}
```

Installation

Pre-requisites: In order to install this feature, you need FreeTDS installed and of course an MS-SMS server.

Configuration: MS-SMS server

An SQL 'view' will have to be created on the MS-SMS server with the correct fields. The view should have these fields:

1. Name0: end-device name
2. User_Name0: the user last logged onto that end-device
3. Operating_System_Name_and0: The operating system detected on the en-device
4. MACAddress0: The MAC associated with the end-device.

Create an SQL user, which FreeNAC will use to query MS-SQL. Give this user SELECT rights on the above view.

You also need to create the SMS table (nac_sms_1) in the FreeNAC database. To do so, go to */opt/nac/contrib* and run the following command:

```
mysql opennac < sms_tables
```

Configuration: FreeNAC

1. The first step is to configure 'FreeTDS' so that SQL queries to MS-SQL actually work.

Add the following instance to *freetds.conf*. Adjust the host and perhaps port for your MS-SQL server:

```
[epo_alias]
host = server.domain.com
port = 1433
tds version = 4.2
dump file = /var/log/freetds_sms.log
dump file append = yes
debug level = 1
```

2. In the *config_en.inc* file, set up your username and password in the variables *$sms_dbuser* and *$sms_dbpass*. These corresponds to the user name and password configured on the Epo SQL server.

3. Set *sms_dbalias* and *sms_db* in the mysql *opennac.config* table

```
update config set value='sms_alias' where name='sms_dbalias';
update config set value='sms_db' where name='sms_db';
```

Here *sms_alias* is the alias you declared in your *freetds.conf* file and *sms_db* is the database that holds the information regarding the McAfee antivirus.

4. Enable the Epo module in FreeNAC:

```
update config set value='true' where name='sms_enabled';
```

If such a field doesn't exist in your config table, create it as follows:

```
insert into config set type='boolean', name='sms_enabled', value='true',
comment='Enable or disable the SMS module';
```

This flag also enables SMS related features in the FreeNAC Windows GUI.

Operations

Configure the MS-SMS module in the config table.

- lastseen_sms: set to 'true' to enable the SMS module.
- lastseen_sms_notify: set to 'true' to enable Email alerts
- sms_device: Text to display in the email alert when an SMS-known end-device connects to the LAN.
- lastseen_sms_restart: set to 'true' to restart the switch port when an SMS-known end-device connects to the LAN.
- lastseen_sms_vlan: Which vlan (index number) is to be assgned to SMS-known end-devices

These can be assigned in the FreeNAC Windows GUI under Administration->config

Synchronization

Run the sync script from /opt/nac/bin

```
./sms_getinfo
```

and watch syslog. It could take some time to complete. If everything went fine, you should see in syslog a message like the following:

```
Dec 19 09:23:47 freenac sms_getinfo[16087]: SMS synchronization was successful.
```

Now that your nac_sms_1 table is populated, you need to restart postconnect in order to load your policy and the changes you made to the SMS config variables.

```
/etc/init.d/postconnect restart
```

Add an entry in cron to perform the synchronization automatically according to your needs.

```
0 2 * * 1       /opt/nac/bin/sms_getinfo
```

WSUS synchronization

Introduction

In an enterprise environment, Microsft's"WSUS server" is often used to manage client PCs patches, pushing Windows and Office updates and ensuring that Windows client PCs stay up to date.

Wsus has an MS-SQL server in its core and the tables in this database were examined to see what information was in there that could be useful to FreeNAC users.

The FreeNAC "WSUS module" queries information from the WSUS regularly (e.g. each night) and stores it in dedicated tables in the FreeNAC database.

The script that performs the WSUS synchronization is *wsus_sync.php*.

Installation

Pre-requisites:

- FreeTDS.
- Wsus must be installed with a full MS-SQL server (not the "light" MSDE), and this module has only been tested with Wsus version 3.0.

Configuration: Wsus

Create an SQL user, which FreeNAC will use to query Wsus. Give this user SELECT rights on the Wsus database.

Enusre that network connectivity to the MS-SQL engine is enabled.

Configuration: FreeNAC

1. The first step is to configure 'FreeTDS' so that SQL queries to Wsus actually work.

Add the following instance to *freetds.conf*. Adjust the host and perhaps port for your Wsus server:

```
[wsus_alias]
host = server.domain.com
port = 1433
tds version = 4.2
dump file = /var/log/freetds_inv.log
dump file append = yes
debug level = 1
```

2. In the *config_en.inc* file, set up your username and password in the variables *$wsus_dbuser* and *$wsus_dbpass*. These corresponds to the user name and password configured on the MS-SQL server.

3. Set *wsus_dbalias* and *wsus_db* in the mysql *opennac.config* table

```
update config set value='wsus_dbalias' where name='wsus_dbalias';
update config set value='wsus_db' where name='wsus_db';
```

Here *wsus_dbalias* is the alias you declared in your *freetds.conf* file and *wsus_db* is the database that holds the information regarding the WSUS patches information.

4. Enable the Wsus module in FreeNAC:

```
update config set value='true' where name='wsus_enabled';
```

If such a field doesn't exist in your config table, create it as follows:

```
insert into config set type='boolean', name='wsus_enabled', value='true',
comment='Enable or disable the WSUS module';
```

This flag also enables Wsus related features in the FreeNAC Windows GUI.

Synchronization

Once you have done the above, running the script is very straightforward.

```
/opt/nac/bin/wsus_sync.php
```

If you don't receive any errors, that means that all parameters have been properly set.

Operations

Add the following entry in crontab in order to update patch information every day

```
0   4    * * 1-5 /opt/nac/bin/wsus_sync.php
```

In this way, it'll run from Monday to Friday at 4:00AM.

Now, in the FreeNAC Windows GUI, Wsus information should be visible in Reports -> Wsus, and for individual end devices in Edit->Wsus.

8. Web interface

Description

The Web GUI is an alternative to the Windows GUI allowing control of some parts of the FreeNAC system. The main method of configuring and monitoring FreeNAC remains the Windows GUI however.

Basic installation

Install Apache & libraries for graphics support:

- Install Apache
- Graphviz for the switch view
- JPGraph for statistics (also required freetype & MS core fonts)
- GD devel libraries

These libraries are normally provided by your distribution and should have been installed in previous (Linux) steps of this installation.

One exception is JPGraph, where the standard package may not work (e.g. on Ubuntu 7.10). If the graphs are not showing, or are completely blank, install from sources as follows. Download the latest jpgraph sources (this example uses the version 2.3) from http://www.aditus.nu/jpgraph/jpdownload.php and untar the file to /opt

```
cd /opt/
tar xvzf jpgraph-2.3.tar.gz
ln -s jpgraph-2.3 jpgraph
```

Next, install the MS TTF fonts that are used by jpgraph, if the msttfcorefonts package is not already on your system.

```
sudo apt-get install msttfcorefonts
```

JPGraph expects to find these fonts at /usr/X11R6/lib/X11/fonts/truetype, so create a link from the actual install location to there, for example on Ubuntu:

```
ln -s /usr/share/fonts/truetype/msttcorefonts/ /usr/X11R6/lib/X11/fonts/truetype
```

Configure the WebGUI

There are several options in the 'config' table of FreeNAC that may need setting, which can be configured using the windows GUI (Administration -> config), or on the SQL command line (e.g. update config set value='NEW VALUE' where name='VARIABLE NAME').

- web_jpgraph: set the path to jpgraph library (default /opt/jpgraph/src/)
- web_dotcmd: set the graphviz binary (default /usr/bin/neato)
- web_lastdays: show devices seen in the last XX days (default 14) in the switch-port diagram of the GUI.

- web_logtail_file: what syslog file is to be shown under Monitoring > Syslog Messages log (default /var/log/messages)
- web_logdebug_file: what syslog file is to be shown under Monitoring > Syslog Debug log (default /var/log/debug)
- web_logtail_length: how many lines of the above log are to be shown (default 100)?
- web_showdhcp: Enable the showing of the fields related to the DHCP management module (in beta status, default=false)
- web_showdns: Enable the showing of the fields related to the DNS management module (in beta status, default=false)

Optional: Excel export

If you want to use the Excel export function you also need the following PEAR Module: *Spreadsheet_Excel_Writer*. To install Spreadsheet_Excel_Writer invoque the following command on your shell:

```
pear install --alldeps -f Spreadsheet_Excel_Writer
```

File permissions

Allow the apache user to read and write key files, for the WebGUI to function correctly.

> Add the apache user to the *freenac* unix group (e.g. on Ubuntu the apache user is www-data)

```
usermod -a -G freenac www-data
```

> Ensure that the freenac group can read the configuration file:

```
chgrp freenac config.inc
chmod 640 config.inc
```

> Change the owner of the */opt/nac/web/tmp* directory to the apache user (e.g. 'www-data' on Ubuntu). This directory is used to write data when graphing.

```
chown www-data /opt/nac/web/tmp
```

The web interface can display the last lines of a given logfile (see 'Monitor > Syslog message log'. By default it shows the last 100 lines of /var/log/messages and /var/log/debug (on Ubuntu). These files needs to be readable by the webserver :

```
chmod 644 /var/log/messages
chmod 644 /var/log/debug
```

Apache: Enable the FreeNAC WebGUI

The Web GUI is located in */opt/nac/web*, so we'll create a virtual directory in Apache pointing to this directory.

Locate your Apache main configuration file (it is distribution dependant) and add the definition of this virtual directory as follows. For example on Ubuntu, create /etc/apache2/sites-available/nac:

```
Alias /nac /opt/nac/web
<Directory /opt/nac/web/>
Options None
Order deny,allow
Allow from all
</Directory>
<LocationMatch "\/nac.*\.inc\.*">
Deny from all
</LocationMatch>
```

The *LocationMatch* stanza protects from reading all include files that you could contain within your */opt/nac/web* directory. This is really important since your *config.inc* file, contains sensitive information such as usernames and passwords.

To make the GUI the default webpage on the webserver, add to /etc/apache2/httpd.conf (on Ubuntu):

```
DocumentRoot /opt/nac/web
```

Enable the /etc/apache2/sites-available/nac configuration above.

```
a2ensite nac
```

Restart apache (/etc/init.d/apache2 restart)

Apache: Restrict Access to the GUI by IP address

The basic configuration above doesn't restrict the use of this interface to anyone. The FreeNAC GUI can be configured to use either AD (active Directory) or no authentication.The AD configuration is discussed in the next section.

If not doing any authentication in the FreeNAC GUI, then $anon_auth=true (and $ad_auth=false) must be set in web/web1.config.inc.

Then access to the GUI probably needs to be limited by a network firewall, or by limiting allowed source addresses in the webserver. To restrict access only to certain IP addresses, adapt the 'nac' Virtual host definition able as follows:

```
Deny from all
Allow from 192.168.0.1 192.168.0.2
```

Apache: Restrict Access by apache login

Alternatively, user accounts can be maintained on apache, and a logon forced to limit access. Once again, set $anon_auth=true (and $ad_auth=false) in web/web1.config.inc.

```
AuthType Basic
AuthName name
AuthBasicProvider  file
# local file
AuthUserFile .htpasswd
Require valid-user
```

Apache: Authentication against Active Directory

The FreeNAC GUI can be configured to use either AD (active Directory) or no authentication (see above). For AD authentication, Apache must be configured (see below) and web/web1.config.inc set as follows: $anon_auth=false, $ad_auth=true.

Using AD authentication allows assignment of rights per user, as each user is individually identified. Rights such as readonly/edit/admin are assigned via the 'guirights' field for that user (see examples further below).

To configure Apache to authenticate users against AD, use the module mod_authnz_ldap. Check if in the list of compiled in modules there is an entry like mod_authnz_ldap.c (running a2enmod without any parameters should list available modules).

Then enable the module:

a2enmod authnz_ldap

If the module is enabled, we are ready to start configuring Apache and the Web interface. If not, install this module.

In your Apache configuration (see above) you have already defined a VirtualHost entry for /nac. To perform AD authentication, you need to modify that entry as follows:

```
 Alias /nac /opt/nac/web
<Directory "/opt/nac/web/">
Options All ExecCGI -Indexes
Order deny,allow
Allow from all

AuthzLDAPAuthoritative off
AuthType Basic
AuthBasicProvider ldap
AuthUserFile /dev/null
AuthName "Sensitive Zone"
AuthLDAPBindDN cn=Administrator,cn=Users,dc=domain,dc=com
AuthLDAPBindPassword password
AuthLDAPURL "ldap://server.domain.com/?userPrincipalName?sub?(&
(objectClass=person)(objectClass=user))"
require valid-user
</Directory>
<LocationMatch "\/nac.*\.inc\.*">
Deny from all
</LocationMatch>
```

AuthLDAPBindDN is an optional DN used to bind to the server when searching for entries. If not provided, mod_authnz_ldap will use an anonymous bind.
AuthLDAPBindPassword is a bind password to use in conjunction with the bind DN.
AuthLDAPBindDN and AuthLDAPBindDN should only be used if no anonymous bind is allowed.
AuthzLDAPAuthoritative prevents other authentication modules from authenticating the user if this one fails. Set to off if this module should let other authentication modules attempt to authenticate the user, should authentication with this module fail.

If you have more than one domain, you should be using global catalog. Global catalog uses port 3268. Global Catalog is a read only copy of selected attributes of all the Active Directory servers within the Active Directory forest. Querying the Global Catalog allows all the domains to be queried in a single query, without the query spanning servers over potentially slow links.

To use the Global Catalog, you just need to substitute the line

```
AuthLDAPURL "ldap://server.domain.com/?userPrincipalName?sub?(&
(objectClass=person)(objectClass=user))"
```

for

```
AuthLDAPURL "ldap://server.domain.com:3268/?userPrincipalName?sub?(&
(objectClass=person)(objectClass=user))"
```

To distinguish users between domains, an identifier called a User Principal Name (UPN) can be added to a user's entry in the directory. This UPN usually takes the form of the user's account name, followed by the domain components of the particular domain, for example

somebody@nz.somedomain.com

For more information about mod_authnz_ldap please see
http://httpd.apache.org/docs/2.2/mod/mod_authnz_ldap.html

Once you are done with this, restart Apache and let's start configuring the Web interface.

Edit your file */opt/nac/etc/config.inc* (or */opt/nac/web/config.inc* if you are using v2.x) and adjust the following variables:

```
$ad_server
$ad_port
$ad_user
$ad_password
$ad_base
$ad_auth
```

- ad_server: The Domain controller where the AD is queried.
- ad_port: Make sure it matches what you have defined in your Apache main configuration file. If you are using a Global Catalog set it to 3268, 389 otherwise.

- ad_user: This is the DN of a user with sufficient privileges to read the necessary information from AD. The possible values for this setting should be in the form 'cn=User,cn=users,dc=domain,dc=com';
- ad_password: The password for ad_user
- ad_base: The base DN (Distinguished Name) where users' information is stored. The possible values for this setting should be in the form 'cn=users,dc=test,dc=com'
- ad_port: Set it to true to active AD authentication in the Web Interface, to false otherwise.

This interface reuses the credentials supplied to Apache to identify the user and do access control. A read-only and edit mode is available, which can be decided on a per-user basis.

Currently the rights can only be assigned in the 'Windows GUI' (Administration > Users > NAC Gui Rights) or on the SQL command line:

```
update users set nac_rights=1 where username='JOE';
update users set nac_rights=2 where username='BILL';
update users set nac_rights=99 where username='SUSAN';
```

This allow the user with the name Joe read-only access, Edit access for Bill and Admin access for Susan. It is really only in the Windows interface that the power of the admin access comes into play.

Starting the WebGUI

After the above configuration, reload/restart apache

```
/etc/init.d/apache2 restart
```

Finally, point your web browser to http://YOURSERVER/nac and you should see the web interface.

See also the User guide documentation.

For troubleshooting, check:

```
         tail -f /var/log/debug   (syslog debug)
tail -f /var/log/message (syslog 'normal' messages)
tail -f /var/log/apache2/error.log  (Apache)
```

The naclog and guilog tables, both of which are visible from the Windows and Web GUIs.

9. Installing 802.1X authentication

Introduction

FreeNAC uses 'vmps' by default for identifying network devices, based on their MAC address. **For stronger authentication look at 802.1x** (which although not bullet proof, is a cryptographic authentication and more difficult to break).

This step is optional, if you already have the standard FreeNAC running, and don't need the additional security.

For an a technical discussion of 802.1x, see the techguide chapter. This section covers some concrete use cases and how to get up and running with them:

- Basic FreeRadius installation
- Authentication Users in a Windows Domain
- Using Cisco's 'Mac-auth-bypass'

Install FreeRadius

For 802.1X support you need to have a RADIUS server installed. The one we use is FreeRadius because it provides a host of features that others don't.

There are several ways to get freeradius running. If you have the FreeNAC Vm it is included, or you can compile from a tarball, or install the binary packages as follows.

Ubuntu:

```
apt-get install freeradius
```

SuSE:

```
yast -i freeradius
```

By default, FreeRadius comes with a sample configuration file (radiusd.conf) which allows you to run your RADIUS server out-of-the-box. The location of the different configuration files is distribution dependant.

To test that you have a working server, as root type:

```
radiusd -xX
```

If in the end you see the following line

```
Info: Ready to process requests.
```

then your Radius server is working. Press Ctrl+C to stop the radius server and activate FreeRadius. Bear in mind that the start up filename is distribution dependant.

Ubuntu:

```
update-rc.d freeradius defaults
```

SuSE:

```
chkconfig freeradius on
```

Basic FreeRadius configuration

You need to define the switches that'll be contacting your Radius server. To do so, create a backup of your clients.conf file and edit it. The configuration files are probably in /etc/freeradius or /usr/local/etc/raddb or /etc/raddb.

```
cp clients.conf clients.conf.$$
vi clients.conf
```

Then add an entry for every switch you want to contact your server in the following form.

```
client 192.168.1.1 {
secret                  = whatever
shortname               = my_switch
nastype                 = cisco
}
```

where *secret* is the string shared between your switch and the Radius server, used to encrypt and sign packets. It doesn't have to be same for all your switches. *shortname* is your switch's identifier and is used for logging and *nastype* is the type of switch your are using and this field is optional.

Create a backup of your radiusd.conf file and proceed to configure your radius server.

```
cp radiusd.conf radiusd.conf.$$
```

802.1x in specific use cases.

Below you'll find two common configuration scenarios for your Radius server.

Authenticating users in Active Directory

0. Introduction

This section describes how to configure FreeRadius to authenticate users in Active Directory.

1. Software required

To authenticate users in Active Directory, make sure you have the following installed in your system:

- Kerberos development libraries
- OpenLDAP development libraries
- Samba
- Winbind
- FreeRadius

2. Configure & test Kerberos & Samba

Create or modify the file *smb.conf* to include the following minimum configuration (change for your Windows environment)

```
[global]
workgroup = domain
security = ads
idmap uid = 16777216-33554431
idmap gid = 16777216-33554431
template shell = /bin/bash
winbind use default domain = no
password server = ads.domain.com
realm = domain.com
[homes]
comment = Home Directories
browseable = No
writable = yes
```

Presumably you already have a functioning Active Directory domain, and know how to run it. AD is very dependent on DNS (domain name system) so first you'll need to add an entry for your server in your DNS.

Once you've added this entry, we need to configure kerberos. Edit the file *krb5.conf* and add in the realms section info concerning your domain.
Your *krb5.conf* file should look like

```
[libdefaults]
default_realm = DOMAIN.COM
dns_lookup_real = false
dns_lookup_kdc = false
[realms]
DOMAIN.COM = {
default_domain = domain.com
kdc = ads.domain.com
admin_server = ads.domain.com
}
[logging]
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmin.log
default = FILE:/var/log/krb5lib.log
```

Modify the lines

```
default_realm = DOMAIN.COM
DOMAIN.COM = {
default_domain = domain.com
```

And change domain.com for your domain. Mind the case.
For the lines

```
kdc = ad.domain.com
admin_server = ad.domain.com
```

You specifiy your Active Directory domain server.

Clock synchronization is so important in the security of the Kerberos protocol.
If clocks are not synchronized within a reasonable window, Kerberos will report fatal errors and refuse to function.
Clients attempting to authenticate from a machine with an inaccurate clock will be failed by the KDC in authentication attempts due to the time difference with the KDC's clock.
Ensure you have your clock properly configured. If you want to use an external source to synchronize your server use ntp.

The Network Time Protocol (NTP) is available for the time synchronization of servers. Add an entry in your crontab to synchronize the clock of your computer with an external time source adding the next entry.

```
#Time synchronization
0 0 * * *        /usr/sbin/ntpdate server > /dev/null 2>&1
```

Save your changes. This entry will synchronize every midnight your clock with the one of server.

Even if your DNS servers are perfect in every way, it is a good idea to add important servers to your local /etc/hosts file. It speeds up lookups and provides a fallback in case the DNS servers go down. So, add the entry for your Active Directory domain server in /etc/hosts.

```
192.168.1.1        ad.domain.com    ad
```

Check that you get no error from typing

```
 kinit Administrator
```

or with other user who has enough privileges to read from Active Directory

This will ask you for the user's password. Make sure you know it beforehand.
Possible causes of error are:

- Clocks not properly synchronized
- DNS resolution

Edit the file */etc/nsswitch.conf* and add winbind at the end of the following lines:

```
passwd:
group:
protocols:
services:
netgroup:
automount:
```

If everything went ok, start Samba and verify that it started. Check for errors in the Samba log. If errors are present, check again your *smb.conf* file. Once Samba has started correctly, do:

```
net join
```

This will join you to the domain. You can verify that your computer has joined the domain by typing

```
klist
```

This should display valid Kerberos tickets and one of those is from your Active Directory domain server.
Now, change the group that the lock *winbindd_privileged* belongs to

```
chgrp radiusd /opt/samba/var/locks/winbindd_privileged
```

Ajust the path and the group according to your system.

This is done because we'll run freeradius as the radiusd user, and this user needs to access this lock in order to perform the authentication against the Windows Domain

Now change its permissions

```
chmod 750 /opt/samba/var/locks/winbindd_privileged
```

And start winbind

```
/etc/init.d/winbindd start
```

Verify that windbind is working. This command pulls a list of users from AD

```
wbinfo -u
```

And check for errors in the winbindd log. If it started successfully, it will create another log file called log.wb-DOMAIN. Once we have winbindd running, activate both smbd and winbindd as services if they are not already activated.

All this has been done in order to get ntlm_auth to run. Try to auth with NTLM

```
ntlm_auth --request-nt-key --domain=domain.com --username=Administrator
password:
NT_STATUS_OK: Success (0x0)
```

This success message indicates that Samba is properly configured to authenticate users against AD, which is what we need for FreeRadius.

3. Configure Freeradius

Below are the relevant sections to configure in radiusd.conf

modules:

```
mschap {
authtype = MS-CHAP
use_mppe = yes
require_encryption = yes
require_strong = yes
with_ntdomain_hack = yes
ntlm_auth = "/opt/samba/bin/ntlm_auth --request-nt-key --domain=%{mschap:NT-
Domain:-DOMAIN} --username=%{mschap:User-Name} --challenge=%{mschap:Challenge:-00}
--nt-response=%{mschap:NT-Response:-00}"
}
```

authorize:

```
mschap
eap
```

authenticate:

```
Auth-Type MS-CHAP {
mschap
}
eap
```

MAC-Auth-bypass

As a requirement for this section you need to have installed and working FreeRadius and FreeNAC.

Before starting configuring your Radius server, there is a step you need to perform. You need to add the user required to run your Radius server to the freenac group. In that way, it can access the configuration file located at */opt/nac/etc/config.inc*. In our system, the FreeRadius server runs under the *radiusd* user, so:

```
usermod -a -G freenac radiusd
```

Find in your */opt/nac/etc/config.inc* file the variable *$vmps_servers* and put there the IP addresses or hostnames of your FreeNAC servers like follows:

```
##Binding with FreeRadius
$vmps_servers = "freenac01, 192.168.201.201,freenac03";
```

Below are the relevant sections to MAC-Authentication bypass to configure in *radiusd.conf*:

modules:

```
perl check_mac {
module = "/opt/nac/bin/rad2vmps"
}
```

authorize:

```
 # Enable MAC lookup via VMPS: collect request data from radius
check_mac
```

authenticate:

```
Auth-Type MAC-Auth-Bypass {
check_mac
}
```

post-auth

```
# Enable MAC lookup via VMPS: query vmps and assign vlan or deny
check_mac
```

Learning mode

Passive Device Detection

If router_mac_ip_discoverall=true in the config table, the router_mac_ip module will document all MAC/IP pairs it finds on the network, not just those actively managed with the vmps protocol (see below). End-devices found in this way are marked with the status "unmanaged" (see for example the overview page in the windows GUI).

See also Installation Guide -> Router Integration .

In this mode FreeNAC can be connected "passively" to the network and an automated list of all end devices on your network is collected for you!

If the nmap scanning module is enabled, even more information will be automatically collected on each end-device.

Active device detection

Starting with a test switch on specific ports, NAC is enabled by configuring ports to use dynamic vlan assignment (“switchport access vlan dynamic” in IOS).

The switch performs a VLAN assignment requests for each new connection and will regularly re-confirm existing connections.

1. If the MAC Address of the connecting system is in the database, the switch will assign the attributed VLAN.
2. If the MAC Address of the connecting system is not in the database, meaning that this is an *unknown* system (new, unmanaged or something else), the switch will assign the *default vlan*.
   The MAC Address will also be inserted into the NAC database so that it can be later edited and activated.

During this *learning mode*, all ports are open to all hosts. This is meant to avoid disrupting the network during the initialisation of the NAC service.

As noted above, new MAC addresses will be inserted into the database as *unknown* hosts. The NAC system will auto-discover the IP & DNS names of these systems (via the *router_mac_ip* program called from cron).

The NAC administrator(s) or the super-users, will need to edit those hosts in the NAC and designate their VLAN and optionally document the end-device details and assign it to a user.

Troubleshooting

See the users guide http://freenac.net/en/community?q=en/usersguide