Global Information Assurance Certification Paper

Building and maintaining a NIDS cluster using FreeBSD and Snort.

Michael Boman
30 August 2001

In today's world what you don't know can hurt you. Internet is becoming a really ugly place, and the attacks that are lunched are becoming more sophisticated and easier to use. This is a harsh reality, but it's true.

You are no longer safe just buying a firewall: most firewalls are not intelligent and usually badly configured - besides the firewall code can also have flaws, just like any other software. I don't say that getting a firewall is a bad idea, but trusting your firewall to protect you against everything is a bad thing.

What I am going to show you here is how to build a NIDS cluster with central logging and maintenance facilities. Hopefully this will help you take more control over your environment so you actually know what is happening on your network, and by knowing that you can take appropriate counter measures to remove the threat. This can include everything to automatically tearing down the TCP connection to reconfigure the firewall(s) to block the offending packets to enter your network in the first place.

It has a few requirements; you need a few PC's you can spare to do the task, a couple of hours time to set the whole thing up and then an hour or so per day to digest logs and maintaining the cluster.

Installation of a NIDS sensor

First of all, you need to install FreeBSD on the PC's. I am not going to tell you how to do it step by step, instead please refer to http://www.freebsd.org/handbook/install.html. Please remember that you shouldn't install more then you need, so leave out all software that's nice to have but not necessary to do the work (XFree86 for an example is something nice, but not necessary to have for a cluster sensor). You will need the following thought: Base install, kernel and system source code and ports collection.

As the latest version of FreeBSD that you can go and buy is 4.3-RELEASE we need to update it - there is a few security issues will the 4.3-RELEASE code. This is the same step as you should do with any system: apply the latest patches (servicepack / hotfix) to make sure that no known security issues is applicable on your production system. You do this by first installing cvsup. If you don't have fast internet access I recommend you to install it from the CDROM. To do that you first mount your CDROM using the command:

```
mount /cdrom
```

and then you install cvsup by executing

```
pkg_add /cdrom/packages/net/cvsup-bin-16.1.tgz
```

If you have installed over network you need to copy the cvsup-bin-16.1.tgz file using FTP, SSH or NFS mounted disk and then run the pkg_add(1) command on it. You can also use pkg_add to retrieve and install the file over internet. For that I suggest you take a look at the manual page for pkg_add.

Once you have cvsup installed we need to update our sources. This has to be done over the internet as we want the latest version of both the kernel and the software. You need to run the following command, changing *cvsup-mirror* to a local FreeBSD cvsup mirror. See http://www.freebsd.org/handbook/cvsup.html#CVSUP-MIRRORS for a list.

```
/usr/local/bin/cvsup -h cvsup-mirror /usr/share/examples/cvsup/ports-supfile
/usr/local/bin/cvsup -h cvsup-mirror /usr/share/examples/cvsup/stable-supfile
```

Once updated we need to re-build the system with the latest sources. Now, this will take quite a long time so I suggest you do it over night or something. You definitely don't need to baby-sit this step.

First, check your kernel configuration. How to configure your kernel is well documented at http://www.freebsd.org/handbook/kernelconfig.html. If you for some reason don't want to re-configure your kernel the GENERIC configuration will do.

The following steps is required to build your new kernel and system binaries (Replace *MYKERNEL* with the name of your custom-configured kernel or GENERIC if your are using the standard configuration):

```
cd /usr/src
make buildworld
make buildkernel KERNCONF=MYKERNEL
```

You can also combine them to a single line so you don't need to baby-sit this step:

```
cd /usr/src; make buildworld; make buildkernel KERNCONF=MYKERNEL
```

Next you need to install your newly built kernel and system binaries. It's recommended that you perform this step in single user mode. To enter single user mode execute

```
init 1
```

Once in single user mode you continue the installation (again, replace *MYKERNEL* with your kernel configuration):

```
cd /usr/src
make installworld
```

```
make installkernel KERNCONF=MYKERNEL
```

and last we need to migrate the new system configuration. This is done with mergemaster(8). Just execute mergemaster and check your configuration, if unsure you can leave the configuration for later by pressing enter at the choice. Reboot the system and now you should have a fully updated FreeBSD system (at this point of time it's 4.4-RC).

Next thing we need to do is to install and configure Snort, an open source Network Intrusion Detection System (NIDS). In FreeBSD this is fairly simple:

```
cd /usr/ports/security/snort
make -DWITH_MYSQL -DWITH_FLEXRESP all install
```

This will install Snort with MySQL database logging and Flexible Response functionality. Because of the intelligent ports system in FreeBSD the MySQL client library will also be downloaded and installed automatically.

You have to decide if you want to copy or synchronize configuration files and if this should be done using SSH (or RSH but I do not recommend this). If you decide you want to synchronize configuration files you need to install rsync by running:

```
cd /usr/ports/net/rsync
make all install clean
```

We also need to install wget:

```
cd /usr/ports/ftp/wget
make all install clean
```

SSH is installed by default in FreeBSD so no extra steps are necessary to use that. Next step is to close down ports we are not going to use:

For RSH you need to have following services enabled in /etc/inetd.conf: telnet and shell (telnet for accessing the system remotely from your workstation). It is however considered a bad idea to use any of these services as they are sending username and password in clear text, and is very easy to break in to a system using those services, so I will not cover that in this paper. All other services can be disabled.

For SSH you don't need to run inetd at all (you can turn it off by setting inetd_enable="NO" in your /etc/rc.conf file), just the SSH service. To enable SSH put sshd_enable="YES" in your /etc/rc.conf file. You also need to permit root logins (or install sudo) by setting "PermitRootLogin yes" in /etc/ssh/sshd_config. Reboot to apply the changes.

Installation of NIDS master

After you have repeated all steps up to here a couple of times (once for each IDS node) it's time to glue the NIDS together. First you need a master server, the system that will handle the configuration for the rest of the systems and be responsible to collect the alerts generated by Snort. It can either be one of the nodes, or a separate server. Please remember to protect it properly thought, as it will run a MySQL database server and a web server. First, install Analyst Console for Intrusion Detection (ACID) on the master. ACID is a nice web based front-end for managing snort alerts written in PHP. To install it you run:

```
cd /usr/ports/security/acid
make all install clean
```

Once that is done it already has installed Apache and MySQL server for you as well. Next step is to create the snort database in MySQL. Execute

```
echo "create database snort;" | mysql
```

to create the database itself, and then

```
cat /usr/ports/security/snort/work/snort-*/contrib/create_mysql | mysql snort
```

to create the relevant tables. Unfortunately the standard configuration for MySQL in FreeBSD is a bit too open, so we need to close it down a bit. We do that by assigning a password for root and create users for ACID and snort. In MySQL run the following queries:

```
use mysql;

UPDATE user SET password = password("new_mysql_root_password")
WHERE User = 'root';

GRANT INSERT, UPDATE, SELECT, DELETE ON snort.* TO acid@localhost
IDENTIFIED BY 'acid_password';
```

And repeat the query below for each snort sensor you have installed (replace *sensor* with the fully qualified hostname or IP address of the cluster node).

```
GRANT SELECT,INSERT ON snort.* TO snort@sensor IDENTIFIED BY
'snort_password';
```

Restart your MySQL database by running

```
mysqladmin flush-privileges
```

Now you will need to enter a password every time you want to access MySQL (and that's a good thing).

Back to the ACID installation, we now need to teach ACID about the username, password and location of the MySQL server. You do this by edit /usr/local/share/doc/apache/acid/acid_conf.php. You need to set the following variables:

```
$DBtype = "mysql";
$alert_dbname = "snort";
$alert_host = "localhost";
$alert_port = "3306";
$alert_user = "acid";
$alert_password = "acid_password";
```

Now we have the Analyst Console up, and it can be accessed by any system. This means that you can sit at your workstation and look at all those trespassers and vandals that wants to get into your network as well as detecting strange traffic (more about this later).

Using SSH to authenticate with the sensors

If you are using ssh you will asked for your password every time you want to copy configuration files over or restart snort on the sensor, which is good but sometimes very annoying. You can solve this in two ways: using an ssh key file with blank password (another bad idea, but not as bad as using clear text communication)

```
ssh-keygen -f $HOME/.ssh/sensor_key -N ""
```

or a ssh key file with a password

```
ssh-keygen -f $HOME/.ssh/sensor_key -N "sensor key password"
```

and use it together with ssh-agent(1)

```
ssh-agent /bin/sh
ssh-add $HOME/.ssh/sensor_key
```

Once the key has been created you need to copy/append the public part of the key to the $HOME/.ssh/authorized_keys file on the sensor, so the sensor know that it can trust this key.

Either way will authenticate you to the sensor using the ssh key instead of a password. As you noticed we ssh as root, and it's another bad idea (in what is now a history of bad ideas). If we use sudo instead (found in /usr/ports/security/sudo) we could execute the

things we needed as root and the rest of it as a normal user. It's a little more trickier to setup and will not be covered here today.

Placement of sensors

You should place a NIDS node in each segment of the network you want to monitor. For the NIDS to see all the traffic it needs to sit either on a hub, or on a managed switch with a SPAN port. The only switch with SPAN port I know is Cisco's Catalyst 19xx and 29xx series, but I am sure there are other brands as well[1].

NIST[2] has published a paper on IDS that includes network diagrams etc. on where to place your IDS. The paper can be found at http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf.

Managing the NIDS cluster

Managing the sensors will be done by using ssh and scp (or rsync over ssh)[3]. By creating a directory structure like ~/nidsmanager/sensors/*sensorname* a script can easily be created that goes trough the list of directories (that is actually the name of each sensor) and copies the snort configuration and rules over to the sensor with scp or with rsync over ssh.

The script should look something like this:

```
for dir in ~/nidsmanager/sensors/*; do
        scp $dir/* root@`echo $dir | \
       sed s/.*sensors//g`:~snort/etc/;
done
```

Applying the changes is done by using SSH in rsh mode to first kill and then restart snort. That script would look something like this:

```
for dir in ~/nidsmanager/sensors/*; do
        ssh root@`echo $dir | sed s/.*sensors//g` \
       /bin/kill 'cat /var/run/snort_$if`;
        ssh root@`echo $dir | sed s/.*sensors//g` \
       /usr/local/bin/snort $snortoptions;
done
```

Another script that downloads the updated distribution rule set can be coded and then placed in crontab. This will make sure that you always have the latest rules available.

---

1 Where I am working we only use CISCO equipment, so I have had little chance to look at other vendors.

2 NIST, National Institute of Standards and Technology. Http://www.nist.gov

3 I was planning to complete a set of scripts for this but I was running shot on time so the scripts are not included in this paper, but keep an eye on www.snort.org as they will be published there once they are completed.

```
wget --quiet\
   http://snort.sourcefire.com/downloads/snortrules.tar.gz
tar xvzf snortrules.tar.gz
rm. snortrules.tar.gz
```

A script that migrate the distribution rule set with what the user has currently configured, and keep a unwanted rule list file - just like what the arachnids_upd[4] is doing - is a nice addition. Modification of distribution rules should be done by copying the rule to local.rules and disable it in it's original file. That way everything that is custom-made is in local.rules, even if it's just a change in the original rule set.

Conclusion

A distributed cluster of NIDS will help you detecting intrusion, and sometimes even buy you some time to fix security holes - but do not let it be your only line of defense. There are numerous ways you can fool a NIDS system[5].

By automating the maintenance of a large number of installations you lower the time you need to spend on managing your IDS sensors and have more things to do other things, like applying the latest patches to the rest of your servers or actually get some work done (unless managing IDS sensors and patching servers is the only work you have to do, and then you have more time over to browse sites like http://www.securityfocus.com, http://www.microsoft.com/security and my personal favorite: http://www.userfriendly.org, not security related but it's a cool site for a good laugh).

References

1. The FreeBSD Project. "FreeBSD Handbook" April 21, 2001. URL: http://www.freebsd.org/handbook/
2. Base, Rebecca and Mell, Peter. "Intrusion Detection Systems" August, 2001. URL: http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf
3. Roesch, Marty. "Snort Users Manual" August 13, 2001. URL: http://snort.sourcefire.com/docs/writing_rules/
4. OpenBSD Project. OpenSSH website. July 25, 2001. URL: http://www.openssh.org
5. Fox, Brian and Ramey, Chet. "GNU Bash-2.05". March 5, 2001. bash(1) manual page alt. URL: http://www.FreeBSD.org/cgi/man.cgi?query=bash&apropos=0&sektion=0&manpath=FreeBSD+Ports&format=html

4 http://nitzer.dhs.org/arachnids_upd/
5 See http://all.net/journal/netsec/9712.html

Last Updated: October 24th, 2017

Upcoming Training

San Diego Fall 2017 - SEC401: Security Essentials Bootcamp Style	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, United Arab Emirates	Nov 04, 2017 - Nov 16, 2017	Live Event
Community SANS Colorado Springs SEC401~	Colorado Springs, CO	Nov 06, 2017 - Nov 11, 2017	Community SANS
SANS Miami 2017	Miami, FL	Nov 06, 2017 - Nov 11, 2017	Live Event
Community SANS Vancouver SEC401^	Vancouver, BC	Nov 06, 2017 - Nov 11, 2017	Community SANS
SANS Sydney 2017	Sydney, Australia	Nov 13, 2017 - Nov 25, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS St. Louis SEC401	St Louis, MO	Nov 27, 2017 - Dec 02, 2017	Community SANS
Community SANS Portland SEC401	Portland, OR	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS San Francisco Winter 2017	San Francisco, CA	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS London November 2017	London, United Kingdom	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS Khobar 2017	Khobar, Saudi Arabia	Dec 02, 2017 - Dec 07, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Dec 04, 2017 - Dec 09, 2017	Community SANS
SANS Austin Winter 2017	Austin, TX	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201712,	Dec 11, 2017 - Jan 24, 2018	vLive
SANS Bangalore 2017	Bangalore, India	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Cyber Defense Initiative 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Dec 14, 2017 - Dec 19, 2017	vLive
Community SANS Nashville SEC401^	Nashville, TN	Jan 08, 2018 - Jan 13, 2018	Community SANS
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
Community SANS Hawaii SEC401	Honolulu, HI	Jan 08, 2018 - Jan 13, 2018	Community SANS
Mentor Session - SEC401	Memphis, TN	Jan 09, 2018 - Mar 13, 2018	Mentor
Northern VA Winter - Reston 2018	Reston, VA	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Amsterdam January 2018	Amsterdam, Netherlands	Jan 15, 2018 - Jan 20, 2018	Live Event
Mentor Session - SEC401	Minneapolis, MN	Jan 16, 2018 - Feb 27, 2018	Mentor
Las Vegas 2018 - SEC401: Security Essentials Bootcamp Style	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	vLive
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS Miami 2018	Miami, FL	Jan 29, 2018 - Feb 03, 2018	Live Event