How to setup and secure Snort, MySQL and Acid on FreeBSD 4.7 Release

by Keith Tokash

Purpose of document

This document will help a user install FreeBSD 4.7 Release, Snort 1.9.0, MySQL 3.23.53, and ACID-0.9.6b21. It will also guide the user through the process of securing the machine and getting the snort sensor(s) to log to a central database over stunnel. The intention is to give users that are new to any of the software the opportunity to build an enterprise-class system based completely on free, open-source tools. Following the instructions in this document will get you the following:

- Multiple FreeBSD boxes, one running the Windowmaker desktop. I chose Windowmaker because the intention of this tutorial is to create dedicated Snort machines. In other words Gnome and KDE are overkill for what we are doing here (and it looks nice).
- Locked-down machines (C2 in 2002!). I tried to be responsible with the securing of these boxes, but this is not a definitive guide to securing FreeBSD; there are several links to those at the end. If I have missed something obvious, feel free to point it out (nicely please) with your suggestion on exactly how to fix it.
- Multiple Snort sensors logging to a central MySql server/viewing station.
- An easy method of updating your software via the ports collection.
- The fastest NIDS for your money.

Assumptions

This document assumes the following:

- The user has at least a little (a few months) experience with a Unix-like operating system, such as any Linux distribution, Solaris etc. This isn't necessary but will make life easier.
- The user has the installation ISO image on CD. This is worth the effort as a time-saver, especially since you will need more than one FreeBSD box.
- These machines will be dedicated Snort boxes.
- The user has the patience to work through the entire document. Much of this document may be repeated in two spots (once for the viewing station setup and again for the sensor); I believe in being redundant for clarity.

I am not an ultra-mega expert on FreeBSD, Snort, MySQL or Apache, nor do I play one on TV, so any contributions are welcome. Please send all suggestions or comments to twigles at yahoo dt com. When following this document remember that it is based on the ports, so if a port gets updated, something may break. Please let me know if this happens but don't get angry about it; it happens. *Please don't ask me how to bend this tutorial to fit Linux – I simply don't know.*

Setting up the viewing station

File system layout
This machine shouldn't need a huge hard drive. An 18-gig SCSI would be fine assuming you eventually delete alerts or move them to some other box. The recommended amount of swap space can be a point of contention; the old Unix standard was 2 x RAM, but nowadays systems can have gigabytes of RAM. I have heard many say that over 2GB RAM, swap space should be 1 x RAM. This is how I would lay out a 9-gig hard drive.

/	100m
swap	2000m
/usr	2000m
/tmp	200
/var	rest

It's important to have at least 1.5 gigs for /usr on the viewing station because we'll be installing X and source code in case we have to patch it later. The large /var partition is for mysql, which logs to /var/db/mysql.

So let's do it.

Other installation details

- Burn the ISO image (the one called "Install") on to a cd. It is well worth the time.
- Choose the X-user install, then scroll down a few lines, hit Custom and choose to install all of the source.
- Install the ports collection.
- Don't enable inetd, we won't need it.
- Don't install the Linux compatibility.
- Say no to most everything (NFS, FTP etc.). Accept the "Moderate" security.
- Configure X; I can't tell you how to do this because it's hardware dependent (know your video card and look up your monitor's horizontal and vertical sync is my advice). My experience with xf86cfg so far is that it will tell you that the setup failed even if it succeeded, so trial and error….
- You have the option during the initial installation to choose the packages installed on your system. I usually do this later in case I mess up the installation and have to start over.
- You probably want to make the default shell /bin/csh rather than /bin/sh for any users you add. The FreeBSD folks replaced csh with tcsh so you can do convenient things like hit the up-arrow for previously used commands and hit [tab] to finish commands or file-names.
- Put your normal user in the group "wheel", otherwise you won't be able to use the "su" command to get to root.
- Use different passwords for your normal user and your root account. If someone figures out one but not the other, they still can't log in as root remotely.

On the initial boot

Log in as root. We are going to do several things to the box before installing Snort and Acid. Since some of it has to do with X, you'll want to be at the console.

Updating your ports collection

The recommended way to install software in FreeBSD is via the ports collection. Therefore it's important to keep your ports up to date. This is easily accomplished by issuing the following commands:

1. cd /usr/ports/net/cvsupit
2. make install clean

The make install took me over 30 minutes. Now you'll get to choose which branch of ports you want to follow. Since we want to follow the current ports, scroll down a few choices and choose ".". You can then tab your way to the OK button, hit enter, then say yes to track the FreeBSD ports collection. Say yes to follow docs, choose a nearby server, and say no when it asks you if you want to run CVS now. Change directory to /etc and comment out the line that says "src-all" in the "cvsupfile" file.

3. rehash
4. cvsup -g -L 2 /etc/cvsupfile

The first time you run CVS, it will likely take another 15 minutes, but after that it's pretty quick. To update your ports collection daily (very recommended), create the following script as /usr/local/etc/cvsup:

```
#!/bin/sh

# updating ports daily
/usr/local/bin/cvsup -g -L 2 /etc/cvsupfile
```

Now install it as a cronjob by issuing the "crontab –e" command and putting the following into the file:

```
0 13 * * *      /usr/local/etc/cvsup
```

Note that the default editor is vi, so hit the 'i' key to enter insert mode, then just type. After you type the command in, hit escape, then "ZZ" to exit and save. Make the file executable with the following command:

- chmod 755 /usr/local/etc/cvsup

This crontab will run cvsup every day at 1pm (to make it run at 1am, change the 13 to a 1). For more details type "man 5 crontab". Note that running CVS does not update your ports, it updates the source in your ports collection so you can rebuild the ports yourself. For more information about the ports collection, goto http://www.freebsd.org/ports/.

X-windows

Don’t start X-windows as root, if you do all the terminals that you pop up will pop up as root. There’s also the X-server issue, which we’ll also mitigate via ipfw. To start X as a mere mortal we’ll install wrapper.

1. cd /usr/ports/x11/wrapper
2. make install clean
3. cd /usr/ports/x11-wm/windowmaker
4. make install clean
5. touch /usr/home/[username]/.xinitrc
6. echo “wmaker” > /usr/home/[username]/.xinitrc

From here you want to change to a normal user to test your X. My non-root user is twigles, so I do the following:

7. su twigles
8. xinit

If your X configuration is set up correctly, you’ll have a beautiful new Windowmaker session staring at you. If not, you’ll get something less beautiful, in which case you just hit [control] + [alt] + [backspace] to kill the X session. Then you can type “/stand/sysinstall”, go to Configure -> XF86Server and try to configure X again.

Installing and setting up PGP

Several high profile break-ins lately have reinforced how important it is to verify the authenticity of whatever it is you are installing. This is not the time to get lazy.

1. cd /usr/ports/security/pgp5
2. make install clean
3. rehash

Setting up pgp is easy if a bit cryptic. The following setup can be for a throwaway key; one that you aren’t really interested in using except to establish a keyring to import the keys of the FreeBSD team. If you don’t understand public-key encryption and are planning on making a living as a security professional, run, don’t walk, to a resource on this topic (there are hundreds) and learn it.

1. pgpk –g
2. pgpk -a http://www.freebsd.org/doc/pgpkeyring.txt

Patching BSD

As of this writing (12-10-02) there are no vulnerabilities in FreeBSD 4.7 Release that affect this guide. However due diligence is every administrator’s responsibility and I’ve included a section here demonstrating how to download, check the validity of, and install a patch.

There are several ways to stay current on patching FreeBSD. The easiest is to subscribe to the security list, which is described at the end. Alternately (or initially) you can go to http://www.freebsd.org/releases/4.7R/errata.html and read up on all of the announcements. Yet another way is to go through the mail archives here: http://docs.freebsd.org/mail/archive/2002/freebsd-security-notifications/.

This is an official patch announcement with instructions (this patch does not need to be applied to 4.7 Release, it is simply an example).

```
a) Download the relevant patch from the location below, and verify the
detached PGP signature using your PGP utility.

# fetch ftp://ftp.FreeBSD.org/pub/FreeBSD/CERT/patches/SA-02:42/resolv.patch
# fetch ftp://ftp.FreeBSD.org/pub/FreeBSD/CERT/patches/SA-02:42/resolv.patch.asc
```

“Resolv.patch” is the patch, the .asc file is the PGP signature of it.

Verifying the PGP signature of the patch

I downloaded the above patch and verified it in the following way (note that I have not explicitly trusted this key so it throws up a warning. This is normal.):

```
L# pgpv resolv.patch.asc
This signature applies to another message
File to check signature against [resolv.patch]:
Good signature made 2002-11-12 18:29 GMT by key:
  1024 bits, Key ID 73D288A5, Created 1996-04-22
   "FreeBSD Security Officer <security-officer@freebsd.org>"

WARNING: The signing key is not trusted to belong to:
FreeBSD Security Officer <security-officer@freebsd.org>
```

Now that you are sure this patch is what it claims to be, you can finish following the installation instructions (sent from the FreeBSD team).

```
b) Execute the following commands as root:

# cd /usr/src
# patch < /path/to/patch

c) Recompile the operating system as described in
<URL:http://www.freebsd.org/doc/handbook/makeworld.html>.
```

Post-installation cleanup
FreeBSD is secure by default but we can do some work. Note that this is not a definitive guide to securing FreeBSD; I have added links to more exhaustive resources at the end.

- Go into /etc/rc.conf and change
  sendmail_enable="YES"
  to
  sendmail_enable="NONE"
  We won't be emailing alerts so Sendmail is unnecessary. If you want to email anything from this box, leave Sendmail alone or research another MTA.
- Also in /etc/rc.conf, edit the following:
  usbd_enable="NO"
  inetd_enable="NO"
  Unless, of course, you are using a USB device.
- Copy /etc/motd to /etc/motd.default and edit /etc/motd to say something nasty about unauthorized access. Leave two blank lines at the top of the file or they will be overwritten. I have an example motd at the end of this document.
- Edit /etc/ssh/sshd_config by changing the line that reads:
  ```
  #Protocol 2,1
  ```
  To:
  ```
  Protocol 2
  ```
  For anyone who thinks SSH version 1 is good enough (like some clue-deficient decision maker at Cisco), take a look at Ettercap (ettercap.sourceforge.net).
- Go into /etc/ttys and change:
  ```
  console none                    unknown off secure
  ```
  To:
  ```
  console none                    unknown off insecure
  ```
  This will make the BSD ask for a password when rebooting into single-user mode. Note that booting into single-user mode is the primary way to recover your root password, so if you do this don't forget root's password.

Installing the necessary ports
The ports collection is housed by default in /usr/ports. Ports require internet connectivity. Several programs are conspicuously missing from the following ports (Apache, MySQL etc). These are installed as required as dependencies. If you ever meet a ports maintainer, buy them a Newcastle. Do the following:

1. Change directory to /usr/ports/sysutils/idled
2. make install clean
3. Change directory to /usr/local/etc
4. cp idled.cf.template idled.cf
5. Open idled.cf in your text editor and add "exempt tty ttyv0 all". This should stop idled from killing your X session.
6. Examine the idled.cf file and change what you want, it's pretty self-explanatory. If necesssary, see "man 5 idled.cf".

7. After installing idled and making the above change, I received the following message when I let my SSH session sit there for an hour:

```
Thu Jul 11 13:57:08
This terminal has been idle 60 minutes. If it remains idle
for 5 more minutes it will be logged out by the system.
```

# Installing Mozilla sometimes takes over an hour.

1. Change directory to /usr/ports/www/mozilla
2. make install clean

1. Change directory to /usr/ports/ftp/wget
2. make install clean

1. Change directory to /usr/ports/graphics/phplot
2. make WITH_X11=yes
3. When presented with a menu of options to configure into phplot, choose GD 2, then hit ok.
4. make install clean

1. Change directory to /usr/ports/databases/adodb
2. make install clean

1. Change directory to /usr/ports/security/stunnel
2. make install clean

1. Change directory to /usr/ports/security/snort
2. make -DWITH_MYSQL -DWITH_FLEXRESP ; make install
3. cp /usr/ports/security/snort/work/snort-1.9.0/contrib/create_mysql /tmp

1. Change directory to /usr/ports/security/acid
2. make install clean

- Type “rehash” so the C shell can find all of the new programs installed.

Adding a user for Snort

Even though this box should not be running Snort, we can run the SQL queries as our standard, non-priveleged Snort user.

```
Enter username [^[a-z0-9_][a-z0-9_-]*$]: snortman
Enter full name [ ]:
Enter shell csh date no sh tcsh [csh]: no
Enter home directory (full path) [/home/snortman]:
Uid [1000]:
Enter login class: default [ ]:
```

```
Login group snortman [snortman]:
Login group is ``snortman". Invite snortman into other groups: guest no
[no]:
Enter password [ ]:
Enter password again [ ]:
```

It's a good idea to use this user exclusively to run Snort so it doesn't need a shell.

Editing the necessary files

Several files need to be edited/customized for this to work. I will not go into optimizing snort.conf here since it is covered in wonderful detail in the Snort User's Manual. However I will tell you what to change to log to MySQL and some other tips to make it work. Also, some other files need to be changed for Apache and Acid to work correctly.

*** Note: If you edit the rules files in Notepad you MUST go into vi and remove the "^M"s

For Snort

1. This machine won't be running Snort. If you decide to run Snort on this machine (which I discourage), then follow the directions in the sensor setup section.

For Apache

1. Go into /etc/hosts and define your host there. For example, my test system is named "sensor01.com", so my /etc/hosts file looks like this:
   - 127.0.0.1 sensor01.com localhost
2. Go into /usr/local/etc/apache/httpd.conf.
3. Change the following lines:
   - DocumentRoot "/usr/local/www/data" to
   - DocumentRoot "/usr/local/www/acid"

   - <Directory "/usr/local/www/data"> to
   - <Directory "/usr/local/www/acid">

For ACID

When using the port, Acid is extremely easy to set up. Just don't use the root user to access the snort database.

1. chmod 644 /usr/local/www/acid/acid_conf.php
2. Go into /usr/local/www/acid/acid_conf.php
3. Edit the following lines:
   - $alert_dbname = "snort";
   - $archive_user = "snortman";
   - $alert_password = [yourPassword];
   - $ChartLib_path = "/usr/local/lib/php/phplot";
   - $portscan_file = "/var/log/snort/portscan.log";

Setting up intial MySQL functionality

Run the following commands as root:

- /usr/local/bin/mysql_install_db
- /usr/local/etc/rc.d/mysql-server.sh start

Now that the server is started:

- Do a "netstat –an". It should show port 3306 listening.
- Do a "ps –aux". You should see a line that says something like "/usr/local/libexec/mysqld --basedir=/usr/local --datadir=/var/db/mysql --user=mysql --pid-file=/"

Kill mysqld process – "/usr/local/etc/rc.d/mysql-server.sh stop"
Change directory to /usr/local/share/mysql. Look through the four different .cnf files to see which one matches your situation best. Personally I used "my-large.cnf" so I did the following:

- cp /usr/local/share/mysql/my-large.cnf /etc/my.cnf

We will now restart mysql with the startup script:

- /usr/local/etc/rc.d/mysql-server.sh start

Now we need to set a password for the root user in mysql. Note that the root user in mysql is *not* the same root user as the FreeBSD root. You can log into mysql as its root user even if you are a mere mortal within FreeBSD. So to set up a password for mysql's root we issue the following commands:

- /usr/local/bin/mysql –u root
- SET PASSWORD FOR root@localhost=PASSWORD('snortman');
- FLUSH PRIVILEGES;
- exit

These commands log you into mysql as root, set the password to "snortman", reset the privileges so the changes take affect, and quit mysql. Don't forget the semi-colon at the end of almost every command in MySQL except quit or exit.

Now type "/usr/local/bin/mysql" and you should get rejected.
Type "/usr/local/bin/mysql -p" and enter the password when prompted.

Setting up MySQL to accept data from Snort

Due to the way stunnel interacts with mysql (using a TCP port instead of a socket) we have to grant permissions on both 127.0.0.1 *and* localhost. This can be a tricky concept, but since stunnel will forward remote traffic via 127.0.0.1, any permissions given to snortman@127.0.0.1 will be given to remote stations. Any permissions given to snortman@localhost will be local only.
Set up the mysql database for snort with the included scripts:

1. As root at the shell type "echo "CREATE DATABASE snort;" | /usr/local/bin/mysql -u root -p"
2. Log into mysql and type "grant INSERT,SELECT on snort.* to snortman@127.0.0.1 IDENTIFIED BY 'snortman';"
3. Now grant snort rights on localhost as well – “grant INSERT,SELECT,CREATE,DELETE on snort.* to snortman@127.0.0.1 IDENTIFIED BY 'snortman';"
4. quit
5. /usr/local/bin/mysql -p < /tmp/create_mysql snort
6. mkdir /var/log/snort
7. chown snortman:snortman /var/log/snort
8. Edit the startup script (/usr/local/etc/rc.d/mysql-server.sh) so that the line that begins with “/usr/local/bin/safe_mysqld --user=mysql” includes the parameter “--bind-address=127.0.0.1”

Log into mysql and verify your snort tables:

```
system02# /usr/local/bin/mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7 to server version: 3.23.53-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> grant INSERT,SELECT on snort.* to snortman@127.0.0.1;
Query OK, 0 rows affected (0.00 sec)

mysql> use snort
Database changed
mysql> show tables;
+------------------+
| Tables_in_snort |
+------------------+
| data           |
| detail          |
…
…
```

Setting up Stunnel

Stunnel is a great little program that allows us to tunnel mysql (and many many other things) over an encrypted session. The port installs a user/group stunnel so we don’t have to.

1. cd /usr/local/etc/stunnel
2. cp stunnel.conf-sample stunnel.conf
3. chmod 644 stunnel.conf

4. openssl req -new -out mail.pem -keyout mail.pem -nodes -x509 -days 365
5. chown stunnel:stunnel mail.pem
6. chmod 600 mail.pem
7. cp /usr/local/etc/rc.d/stunnel.sh.sample /usr/local/etc/rc.d/stunnel.sh
8. mkdir /var/tmp/stunnel
9. chown stunnel:stunnel /var/tmp/stunnel/

This is what the stunnel.conf should look like (edit your own IP in the “accept” line):

```
master# vi stunnel.conf
# Sample stunnel configuration file
# Copyright by Michal Trojnara 2002

# Comment it out on Win32
cert = /usr/local/etc/stunnel/mail.pem
chroot = /var/tmp/stunnel
# PID is created inside chroot jail
pid = /stunnel.pid
setuid = stunnel
setgid = stunnel

# Authentication stuff
#verify = 2
# don't forget about c_rehash CApath
# it is located inside chroot jail:
#CApath = /certs
# or simply use CAfile instead:
#CAfile = /usr/local/etc/stunnel/certs.pem

# Some debugging stuff
#debug = 7
#output = stunnel.log

# Use it for client mode
#client = yes

#foreground = yes

# Service-level configuration

[3307]
accept = 192.168.1.6:3307
connect = 127.0.0.1:3306
```

Start stunnel with the startup script – “/usr/local/etc/rc.d/stunnel.sh start”. If stunnel doesn’t start or some unknown thing is broken and you suspect stunnel might yield useful information, uncomment the “debug = 7” and “foreground = yes” lines in stunnel.conf and restart it.

Mozilla

Point Mozilla to http://localhost/acid_main.php and run the Setup. You should see four lines telling you you successfully created the tables. I recommend making this your home page.

Preparing your firewall boot options

In /etc/rc.conf, add the following. We will start with an “OPEN” firewall, then close this hole after we verify that the box works.

```
#required for ipfw support
firewall_enable="YES"
firewall_script="/etc/rc.firewall"
firewall_type="OPEN"
firewall_quiet="YES"
firewall_logging_enable="YES"

#extra firewalling options
log_in_vain="YES"
tcp_drop_synfin="NO”
tcp_restrict_rst="YES"
icmp_drop_redirect="YES"
```

Kernel configuration

The default kernel (“GENERIC”) does not have the firewall built in. We will need to change this.

- cd /sys/i386/conf
- cp GENERIC FW

Add the following lines to FW. By default you should have the ‘vi’ and ‘ee’ text editors. ‘ee’ is a easier if you’ve never used either (It stands for easy editor).

```
#To enable IPFW with default deny all packets
options   IPFIREWALL
options   IPFIREWALL_VERBOSE
options   IPFIREWALL_VERBOSE_LIMIT=10
```

After adding and subtracting what you want from your kernel, build it with the following commands. Note that it is easy to mess up kernels so don’t get frustrated if yours doesn’t compile and run the first time. Just go back and try changing less things.

1. cd /usr/src
2. make buildkernel KERNCONF=FW

3. make installkernel KERNCONF=FW
4. shutdown –r now

Setting up rules for IPFW

Obviously our packet filter isn’t doing anything right now. This was deliberate since we want to make sure everything works before adding the complexity of a firewall. We can change this by doing the following:

- Change directory to /etc
- Open rc.firewall in your text editor. Scroll down to the line that says [Cc][Ll][Ii][Ee][Nn][Tt]. This is the setup for a client, meaning that this ruleset is designed to protect this box, not the whole network.
- Set the “net”, “netmask”, and “ip” variables to your network’s values.
- Comment out the following unecessary lines:

```
${fwcmd} add pass all from ${ip} to ${net}:${mask}
${fwcmd} add pass all from ${net}:${mask} to ${ip}

${fwcmd} add pass tcp from any to ${ip} 25 setup
```

- Since I’m running this box in a test lab on a 192.168.10.x network, I added the following lines right below the line we comment out for email to allow me to SSH into the box from inside this network:

```
 # Allow incoming SSH
 ${fwcmd} add pass tcp from 192.168.10.0/24 to ${ip} 22 setup
```

- This rule allows the mysql over stunnel in. Note that this rule isn’t particularly tight; an advanced ruleset is beyond the scope of this paper.

```
# Allow incoming mysql over stunnel
${fwcmd} add pass tcp from 192.168.10.0/24 to ${ip} 3307 setup
```

- The order of rules is important so don’t permit something after a line that blocks it.
- Open rc.conf in your text editor and change “firewall_type="OPEN"” to “firewall_type="CLIENT"”
- Reboot with “shutdown –r now”
- Test the rules by pinging the box and then SSHing to it. The ping should fail and the SSH should work.

Setting up the sensor(s)

File system layout

This machine shouldn't need a huge hard drive. An 9-gig SCSI would be more than enough assuming you clean out /var/log/snort/alert. The recommended amount of swap space can be a point of contention; the old Unix standard was 2 x RAM, but nowadays systems can have gigabytes of RAM. I have heard many say that over 2GB RAM, swap space should be 1 x RAM. This is how I would lay out a 9-gig hard drive.

/	100m
swap	3000m
/tmp	200m
/var	100m
/usr	rest

It's important to have a reasonably large /usr because we'll be installing source code in case we have to patch it later.

Other installation details

- Choose the minimum install, then scroll down a few lines, hit Custom and choose to install all of the source, then select the ports too.
- Don't install inetd, we won't need it.
- Don't install the Linux compatibility.
- Say no to most everything (NFS, FTP etc.). Accept the "Moderate" security.
- You have the option during the initial installation to choose the packages installed on your system. I usually do this later in case I mess up the installation and have to start over.
- You probably want to make the default shell /bin/csh rather than /bin/sh for any users you add. The FreeBSD folks replaced csh with tcsh so you can do convenient things like hit the up-arrow for previously used commands and hit [tab] to finish commands or file-names.
- Put your normal user in the group "wheel", otherwise you won't be able to use the "su" command to get to root.
- Use different passwords for your normal user and your root account. If someone figures out one but not the other, they still can't log in as root remotely.

On the initial boot

Log in as root. We are going to do several things to the box before installing Snort and Acid. Feel free to SSH into the box for this part since the sensors should be completely manageable via remote access.

Updating your ports collection

Please refer to the viewing station process, it is identical.

Patching BSD

Please refer to the viewing station process, it is identical.

Post-installation cleanup

- This step also follows the same process as the viewing station with the exception of bringing up the sniffing interface. Since this box should have two NICs (one maintanence and one sniffing), add a line in /etc/rc.conf to bring the second NIC up at boot. Since I have an Intel Pro card, I added the following:

  ifconfig_fxp0="up"

  This brings the sniffing interface up without an IP address.

Installing the necessary ports

The ports collection is housed by default in /usr/ports. Ports require internet connectivity. Several programs are conspicuously missing from the following ports (Apache, MySQL etc). These are installed as required as dependencies. If you ever meet a ports maintainer, buy them a Newcastle. Do the following:

1. Change directory to /usr/ports/sysutils/idled
2. make install clean
3. Change directory to /usr/local/etc
4. cp idled.cf.template idled.cf
5. Examine the idled.cf file and change what you want, it's pretty self-explanatory. If necesssary, see "man 5 idled.cf".
6. After installing idled and making the above change, I received the following message when I let my SSH session sit there for an hour:

```
Thu Jul 11 13:57:08
This terminal has been idle 60 minutes. If it remains idle
for 5 more minutes it will be logged out by the system.
```

1. Change directory to /usr/ports/ftp/wget
2. make install clean

1. Change directory to /usr/ports/security/stunnel
2. make install clean

1. Change directory to /usr/ports/security/snort
2. make -DWITH_MYSQL -DWITH_FLEXRESP ; make install

- Type "rehash" so the C shell can find all of the new programs installed.

Adding a user for Snort

It is never a good idea to run anything as root (or Local System <snicker>) if you don't have to.

```
Enter username [^[a-z0-9_][a-z0-9_-]*$]: snortman
Enter full name [ ]:
Enter shell csh date no sh tcsh [csh]: no
Enter home directory (full path) [/home/snortman]:
Uid [1000]:
Enter login class: default [ ]:
Login group snortman [snortman]:
Login group is ``snortman". Invite snortman into other groups: guest no
[no]:
Enter password [ ]:
Enter password again [ ]:
```

It’s a good idea to use this user exclusively to run Snort so it doesn’t need a shell.

Editing the necessary files

Several files need to be edited/customized for this to work. I will not go into optimizing snort.conf here since it is covered in wonderful detail in the Snort User’s Manual. However some other files need to be changed for Snort and Stunnel to work correctly.

> *** Note: If you edit the rules files in Notepad you MUST go into vi and remove the "^M"s

For Snort

1. Change directory to /usr/local/etc and issue the command “cp snort.conf-sample snort.conf”.
2. chmod 644 snort.conf
3. Edit the variables in section one to fit your network.
4. In Section 3 add the following line in the database section (edited to fit your own username/password information obviously):
    a. "output database: log, mysql, user=snortman password=snortman dbname=snort host=127.0.0.1 sensor_name=sensor01"
5. cp /usr/local/share/snort/classification.config-sample /usr/local/etc/classification.config
6. mkdir /var/log/snort
7. chown -R snortman:snortman /var/log/snort

Snort startup script

I’ve never won any awards for my scripting but this one gets snort started at boot. I do receive an error regarding MySQL, but Snort still works so investigating that error is a lower priority than it would normally be. Also, the “sleep 10” was necessary on my lab machine to allow Snort to drop privileges without failing to start…also a mystery. Put this script in /usr/local/etc/rc.d. I called mine snort.sh (it must end in “.sh”) and don’t forget to chmod 750 it. Also you must have a MySQL server database for Snort to log into upon boot or it may die; that means boot your viewing station first.

```
#!/bin/sh
```

```
sleep 10

case "$1" in
        start)
                if [ -x /usr/local/bin/snort ]; then
                        /usr/local/bin/snort -c /usr/local/etc/snort.conf -i rl0 -u snortman -g
snortman -D > /dev/null & echo -n ' snort'
                echo ""
                fi
                ;;
        stop)
                /usr/bin/killall snort > /dev/null 2>&1 && echo -n ' snort'
                echo ""
                ;;
        *)
                echo ""
                echo "Usage: `basename $0` { start | stop }"
                echo ""
                exit 64
                ;;
esac
```

Setting up Stunnel
Stunnel is a great little program that allows us to tunnel mysql (and many many other things) over an encrypted session. The port installs a user/group stunnel so we don't have to.

1. cd /usr/local/etc/stunnel
2. cp stunnel.conf-sample stunnel.conf
3. chmod 644 stunnel.conf
4. openssl req -new -out mail.pem -keyout mail.pem -nodes -x509 -days 365
5. chown stunnel:stunnel mail.pem
6. chmod 600 mail.pem
7. cp /usr/local/etc/rc.d/stunnel.sh.sample /usr/local/etc/rc.d/stunnel.sh
8. mkdir /var/tmp/stunnel
9. chown stunnel:stunnel /var/tmp/stunnel/

This is what the stunnel.conf should look like (edit your own IP in the "accept" line). Note that this is not identical to the viewing station's file:

```
sensor01# more stunnel.conf
# Sample stunnel configuration file
# Copyright by Michal Trojnara 2002

# Comment it out on Win32
cert = /usr/local/etc/stunnel/mail.pem
```

```
chroot = /var/tmp/stunnel
# PID is created inside chroot jail
pid = /stunnel.pid
setuid = stunnel
setgid = stunnel

# Authentication stuff
#verify = 2
# don't forget about c_rehash CApath
# it is located inside chroot jail:
#CApath = /certs
# or simply use CAfile instead:
#CAfile = /usr/local/etc/stunnel/certs.pem

# Some debugging stuff
#debug = 7
#output = stunnel.log

# Use it for client mode
client = yes

#foreground = yes

# Service-level configuration

[3306]
accept = 127.0.0.1:3306
connect = 192.168.1.6:3307
```

Preparing your firewall boot options

Please refer to the viewing station process, it is identical.

Kernel configuration

Please refer to the viewing station process, it is identical.

Setting up rules for IPFW

Please refer to the viewing station process, it is identical with the exception that you don’t need to allow TCP connections inbound on port 3307.

Miscellaneous

Be careful using a Cisco 29xx or 35xx switch to sniff from. Due to the way shared memory is allocated in those switches, if the sniffing port is highly utilized (>50%) it can drag performance down from other ports. You also can not sniff more than one VLAN on those switches or have the sniffer and its target both in port-protected mode.

Some hubs are also tricky when it comes to sniffing. If the hub is 10/100 instead of all 10Mb or all 100Mb there is a chance it is switching or bridging between the two speeds and some traffic may not reach your sensor.

Things that the administrator should do on his/her own

There are several things the administrator should do that are not included in this guide.

- Set up and use NTP. The entire network should be running the exact same time. This subject is well documented and not hard to do.
- Place the sensor in a location where it will see all the traffic you want it to.
- Use good password policies. Specifically don't use a password on this box that is used on other, less-secure machines.
- Tune ruleset.

To-do list

- Script most of this garbage!!
- Use Blowfish for passwords
- Incorporate secure_levels
- Add archive database in Acid
- Rsync for configuration and maybe rulesets (not in clear)
- CPU/memory statistics for monitoring box
- Updating procedure via ports
- Change timezone to UTC (GMT to FreeBSD)
- Integrate management tool(s)

Example motd

This motd looks much better in raw text. Give it a try.

Unauthorized access is prohibited.

```
          ))))
           ))))
::::        ))))
::::         ))))
     ----      ))))
     ----      ))))
::::         ))))
::::        ))))
           ))))
          ))))
```

Lists to subscribe to

During the course of your BSD box's lifetime, it will become vulnerable to exploits. The easiest way to deal with this is to receive timely news regarding FreeBSD's vulnerabilities. This can be done by subscribing to the security@freebsd.org list, which has a lot of knowledgeable folks discussing BSD security. Alternatively, you can subscribe to freebsd-security-notifications@freebsd.org, which has a *lot* less traffic (only official notifications sent out by the development team). To subscribe to the latter, send an email to majordomo@freebsd.org with "subscribe freebsd-security-notifications" in the body of the email.

Resources

I drew from quite a few resources to put this together; the top ones are as follows:

www.freebsd.org/doc/handbook/
www.onlamp.com/bsd/
http://docs.freebsd.org/mail/
"FreeBSD Unleashed" by Michael Urban and Brian Tiemann
The security@freebsd.org mailing list
The stunnel mailing list
The mysql mailing list
Google and multiple archives
Snort Installation Manual - Snort, MySQL, Redhat 7.3 by Steven J Scott (He shamed me into adding a table of contents and other nice things;-).
More man pages than I care to admit

Further reading on FreeBSD security
http://sddi.net/FBSDSecCheckList.html
http://people.freebsd.org/~jkb/howto.html
http://www.freebsd.org/security/
http://www.daemonnews.org/200108/security-howto.html