Building a Debian\Snort based IDS

Jason Weir – jason.weir@nhrs.org – 3/8/2011

This document is loosely based on a document by Andy Firman dated June 23, 2006, located at http://firmanix.com/deb-snort-howto.pdf - If you find any errors please let me know!

This document installs Debian 6.0 (Squeeze), Snort 2.9.0.5, Barnyard2-1.9, BASE 1.4.5 and the Emerging Threats rule set.

Table of Contents:

1. Install OS and base software

This document assumes 2 NIC cards with eth0 being the management interface and eth1 being the collector interface.

Get Debian here: http://www.debian.org/distrib/netinst. I used the i386 small CD version. Burn the iso image and boot the CD.

Choose the default options (or as appropriate for your site), when you get to the "Software Selection" screen, unselect all options to get a bare minimum install. After the install finishes, the CD ejects and the system will reboot.

 # apt-get update && apt-get install ssh – This is so we can connect via SSH and copy\paste to the terminal.

Dotdeb.org maintains current packages of mysql and php – we need to add their repositories so apt can use them
 # vi /etc/apt/sources.list

Add the following lines:

deb http://packages.dotdeb.org squeeze all
deb-src http://packages.dotdeb.org squeeze all

Install the dotdeb GnuPG key:

 # cd /usr/src
 # wget http://www.dotdeb.org/dotdeb.gpg
 # cat dotdeb.gpg | apt-key add -

Apt will require input – for example MySQL will ask for you to enter a “root” password for the MySQL server. Make it secure and don’t forget it.

**# apt-get update && apt-get install apache2 libapache2-mod-php5 libwww-perl mysql-server mysql-common mysql-client **
**php5-mysql libnet1 libnet1-dev libpcre3 libpcre3-dev autoconf libcrypt-ssleay-perl libmysqlclient-dev php5-gd php-pear **
libphp-adodb php5-cli libtool libssl-dev gcc-4.4 g++ automake gcc make flex bison apache2-doc ca-certificates vim

2. Install Snort pre-requisites - libpcap, libdnet, and DAQ

Install libpcap:
 # cd /usr/src
 # wget http://www.tcpdump.org/release/libpcap-1.1.1.tar.gz
 # tar -zxf libpcap-1.1.1.tar.gz && cd libpcap-1.1.1
 # ./configure --prefix=/usr --enable-shared
 # make && make install

Install libdnet:
 # cd /usr/src
 # wget http://libdnet.googlecode.com/files/libdnet-1.12.tgz
 # tar -zxf libdnet-1.12.tgz && cd libdnet-1.12
 # ./configure --prefix=/usr --enable-shared
 # make && make install

Install DAQ:
 # cd /usr/src
 # wget http://www.snort.org/dl/snort-current/daq-0.5.tar.gz
 # tar -zxf daq-0.5.tar.gz && cd daq-0.5

DAQ needs to be patched to properly recognize the buffer_size parameter.

 # vi /usr/src/daq-0.5/os-daq-modules/daq_pcap.c

on line 219 replace:
context->buffer_size = strtol(entry->key, NULL, 10);
with:
context->buffer_size = strtol(entry->value, NULL, 10);

 # ./configure
 # make && make install

Update the shared library path
 # echo >> /etc/ld.so.conf /usr/lib && ldconfig

3. Install, configure & start Snort

 # cd /usr/src
 # wget http://www.snort.org/dl/snort-current/snort-2.9.0.5.tar.gz -O snort-2.9.0.5.tar.gz
 # tar -zxf snort-2.9.0.5.tar.gz && cd snort-2.9.0.5
 # ./configure --with-mysql --enable-dynamicplugin --enable-perfprofiling --enable-ipv6 --enable-zlib --enable-reload
 # make && make install

 # mkdir /etc/snort /etc/snort/rules /var/log/snort /var/log/barnyard2 /usr/local/lib/snort_dynamicrules
 # groupadd snort && useradd -g snort snort
 # chown snort:snort /var/log/snort /var/log/barnyard2
 # cp /usr/src/snort-2.9.0.5/etc/*.conf* /etc/snort
 # cp /usr/src/snort-2.9.0.5/etc/*.map /etc/snort

 # vi /etc/snort/snort.conf

Change these lines:
Line #39 - **ipvar HOME_NET 192.168.1.0/24** – make this match your internal (friendly) network
Line #42 - **ipvar EXTERNAL_NET !$HOME_NET**
Line #80 - **var RULE_PATH ./rules** – this assumes /etc/snort/rules
Line #186-#190 comment out all of the preprocessor normalize_ lines
Line #366 - add this: **output unified2: filename snort.log, limit 128**
Line #395 - delete or comment out all of the “include $RULE_PATH” lines except “local.rules”

 # vi /etc/snort/rules/local.rules

Enter a simple rule like this for testing:
alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001;)

Now we can start and test snort.

 # /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0

Ping the management IP address from another machine, alerts should be printed to the console like this:

02/09-11:29:43.450236 [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 172.26.12.1 -> 172.26.12.2
02/09-11:29:43.450251 [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 172.26.12.2 -> 172.26.12.1
02/09-11:29:44.450949 [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 172.26.12.1 -> 172.26.12.2
02/09-11:29:44.450957 [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 172.26.12.2 -> 172.26.12.1

If so congrats – you have Snort working… Use ctrl-c to kill snort..

4. Setup the MySQL server

 # mysql -u root -p #You will be prompted to enter the password you created during installation.
mysql> create database snort;
mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to snort@localhost;
mysql> SET PASSWORD FOR snort@localhost=PASSWORD('mypassword'); # set user password
mysql> exit;

Now we have to import the database schema:
 # mysql -u root -p < /usr/src/snort-2.9.0.5/schemas/create_mysql snort # enter password again
 # mysql -u root -p # enter password again
mysql> use snort;
mysql> show tables; # you should see the list of new tables you just imported.
mysql> exit;

5. Install & configure barnyard2

 # cd /usr/src
 # wget http://www.securixlive.com/download/barnyard2/barnyard2-1.9.tar.gz
 # tar -zxf barnyard2-1.9.tar.gz && cd barnyard2-1.9
 # ./configure --with-mysql
 # make && make install
 # mv /usr/local/etc/barnyard2.conf /etc/snort
 # vi /etc/snort/barnyard2.conf
Line #215 change to **output alert_fast**

At the end of the file add this line:
output database: log, mysql, user=snort password=<mypassword> dbname=snort host=localhost

Now start snort and barnyard2 with these commands:
 # /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0 &
**# /usr/local/bin/barnyard2 -c /etc/snort/barnyard2.conf **
**-d /var/log/snort -f snort.log -w /etc/snort/bylog.waldo **
**-G /etc/snort/gen-msg.map -S /etc/snort/sid-msg.map **
-C /etc/snort/classification.config &

This command shows that barnyard is correctly inserting events into the database:
 # mysql -uroot -p -D snort -e "select count(*) from event" # enter password again

6. Configure Apache2 & PHP

 # cp /etc/apache2/sites-available/default-ssl /etc/apache2/sites-enabled
 # vi /etc/php5/apache2/php.ini
Line #514 – change line to read - **error_reporting = E_ALL & ~E_NOTICE**
 # a2enmod ssl
 # pear config-set preferred_state alpha
 # pear install Image_Color Image_Canvas Image_Graph
 # /etc/init.d/apache2 restart

7. Install and configure BASE

 # cd /usr/src
 # wget http://sourceforge.net/projects/secureideas/files/BASE/base-1.4.5/base-1.4.5.tar.gz
 # tar -zxf base-1.4.5.tar.gz
 # cp -r base-1.4.5 /var/www/base
 # chmod 777 /var/www/base (just for now)

Open a browser and go to: https://192.168.1.13/base (or whatever the management IP is) .

Click Continue, choose English
Path to adodb: /usr/share/php/adodb
Click Continue
Database Name: snort
Database Host: localhost
Database Port: leave blank
Database User Name: snort
Database Password: mypass

Put in values for the authentication system and click submit.
Click "create baseag" which extends the DB to support BASE.

Continue to step 5 to login.
You should see a number next to unique alerts – click on that and you should see alerts like this:

Snort Alert [1:10000001:0] – the test rule we created above

If you see alerts in BASE – Congrats – everything is working as it should be..

8. Startup script for snort & barnyard

 # vi /etc/init.d/snortbarn
Paste the following into the file:

```
#! /bin/sh
#
### BEGIN INIT INFO
# Provides:          snortbarn
# Required-Start:    $remote_fs $syslog mysql
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# X-Interactive: true
# Short-Description: Start Snort and Barnyard
### END INIT INFO

. /lib/init/vars.sh
. /lib/lsb/init-functions

mysqld_get_param() {
        /usr/sbin/mysqld --print-defaults \
                | tr " " "\n" \
                | grep -- "--$1" \
                | tail -n 1 \
                | cut -d= -f2
}

do_start()
{
        log_daemon_msg "Starting Snort and Barnyard" ""
        # Make sure mysql has finished starting
        ps_alive=0
        while [  $ps_alive -lt 1 ];
        do
        pidfile=`mysqld_get_param pid-file`
        if [ -f "$pidfile" ] && ps `cat $pidfile` >/dev/null 2>&1; then ps_alive=1; fi
        sleep 1
        done

        /sbin/ifconfig eth1 up
        /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth1 &
        /usr/local/bin/barnyard2 -q -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.log -w /etc/snort/bylog.waldo \
        -G /etc/snort/gen-msg.map -S /etc/snort/sid-msg.map -C /etc/snort/classification.config 2> /dev/nul &

        log_end_msg 0
        return 0
}

do_stop()
{
        log_daemon_msg "Stopping Snort and Barnyard" ""
        kill $(pidof snort) 2> /dev/nul
        kill $(pidof barnyard2) 2> /dev/nul
        log_end_msg 0
        return 0
}

case "$1" in
  start)
        do_start
        ;;
  stop)
        do_stop
        ;;
  restart)
        do_stop
        do_start
```

```
    ;;
  *)
    echo "Usage: snort-barn {start|stop|restart}" >&2
    exit 3
    ;;
esac
exit 0
```

Make it executable and create the startup symlinks:

 # chmod +x /etc/init.d/snortbarn
 # insserv -f -v snortbarn

Snort & Barnyard will now start automatically at boot.

9. Keep your rules up to date with pulledpork

 # cd /usr/src
 # wget http://pulledpork.googlecode.com/files/pulledpork-0.5.0.tar.gz
 # tar -zxf pulledpork-0.5.0.tar.gz && cd pulledpork-0.5.0
 # cp pulledpork.pl /usr/local/bin && cp etc/*.conf /etc/snort

For simplicity we're starting with the Emerging Threats open rule set. I encourage you to look at the professional rules available at http://www.emergingthreatspro.com/ and http://www.snort.org/snort-rules/

 # vi /etc/snort/pulledpork.conf

Comment out line 20 & 24

Line 56: change to: **rule_path=/etc/snort/rules/snort.rules**
Line 64: change to: **local_rules =/etc/snort/rules/local.rules**
Line 67: change to: **sid_msg=/etc/snort/sid-msg.map**
Line 90: change to: **config_path=/etc/snort/snort.conf**
Line 101: change to: **distro=Debian-Lenny**
Line 133: Uncomment and change to: **snort_version=2.9.0.5**
Line 137: Uncomment and change to: **enablesid=/etc/snort/enablesid.conf**
Line 139: Uncomment and change to: **disablesid=/etc/snort/disablesid.conf**
Line 140: Uncomment and change to: **modifysid=/etc/snort/modifysid.conf**

 # echo pcre:fwsam >> **/etc/snort/disablesid.conf** # disables all block (fwsam) rules
 # vi /etc/snort/modifysid.conf # last line – change to 302,429,1821 "$EXTERNAL_NET" "$HOME_NET" (typo in the file I think)

Run pulledpork

 # /usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf -T -l

You should now see local.rules and snort.rules in /etc/snort/rules.

Clean Up:

 # rm /var/www/index.html
 # chmod 755 /var/www/base
 # pkill snort && pkill barnyard2
 # rm -rf /var/log/snort/* /var/log/barnyard2/*
 # vi /etc/snort/rules/local.rules – Comment out the test rule
 # vi /etc/snort/snort.conf – Line 394: add: **include $RULE_PATH/snort.rules**

Plug a span port or tap into eth1 and restart snort

 # /etc/init.d/snortbarn restart

10. What I left out, building it was the easy part..

- How to use VI – sorry notepad users
- Hardening the sensor – for example, not allowing ssh root login.
- Tuning the sensor – read snort.conf, disabledsid.conf, enablesid.conf and modifysid.conf.
- Scheduling rule updates – running pulled pork daily via /etc/cron.daily works good.
- Restarting Snort after rule updates – I don't like running snort as root so pulledpork doesn't work.
- Setting up a span port or ethernet tap – where to place the sensor and how to get packets to it.
- Rule writing – hardest thing to master, join the Emerging Threats and Snort signature mailing lists.
- What to do with the data.