Getting SNORT working in CentOS 6.3/6.4 and VirtualBox 4.x.x

Last Revised on November 1, 2014

The document below uses the following color codes for items/steps the user should be aware of during the configuration and installation of DAQ-2.0.x and Snort-2.9.7.x:

Blue - informational messages and comments
Orange – These are commands that the user types at the shell prompt
Red – **Read carefully before proceeding**.

This document describes the configuration, compiling, and installation of DAQ 2.0.x and SNORT 2.9.7.x using the Hardware and Operating System(s) listed below:

Microsoft Windows 7 Ultimate Edition w/SP1 as the HOST operating system
VirtualBox 4.x.x with Oracle Extension Pack 4.x.x (I use version 4.3.18)
CentOS 6.3/6.4 (64-bit version) as the GUEST operating system (which runs SNORT)
SNORT 2.9.7.x, DAQ 2.0.x, and a set of snort rules (www.snort.org)

The hardware in the HOST system listed above is a quad-core processor (AMD) @ 2.8Ghz, 8GB of DDR2 1066Mhz RAM, and a onboard Realtek PCIe Gigabit Ethernet Family Controller.

*** NOTE ***

Before replacing a WORKING production copy of Snort with a new version of Snort and updated Snort rules, it is STRONGLY recommended that users set up a test environment to install the latest versions of DAQ and Snort (along with updated Snort rule snapshots) and to fully test any potential modifications in this environment.

I prefer to use a Virtual Machine inside of VirtualBox 4.x.x when installing and/or upgrading Snort, so if something goes wrong, I can simply remove the virtual machine and reload the operating environment from scratch, without damaging any production systems that may be running Snort or other critical services.

*** NOTE ***

In the CentOS 6.3/6.4 Virtual Machine, you will need to set the NETWORK section to BRIDGED mode to allow the assignment of a static IP to your CentOS 6.3/6.4 VM (if you are using a standalone system running CentOS 6.3/6.4 you can ignore this step).

Configure your Static IP, Network Mask, DNS, and Gateway on the desktop by clicking System | Preferences | Network Connections (requires root access) for CentOS 6.3/6.4 (in my case, I used ethernet 0 (eth0) as the port to monitor traffic on with an assigned IP address of 192.168.1.90).

After completing the step above, ensure your network connectivity is working (try ping www.cisco.com, you should get a response), also try surfing a few web pages from CentOS 6.3/6.4, www.snort.org would be a good site to visit (shameless plug here).

Make sure the following packages are installed in your CentOS 6.3/6.4 system via System | Administration | Add/Remove Software (requires '**root**' privileges): **gcc** version (4.4.6 including libraries), **flex (**2.5.35), **bison** (2.4.1), **zlib** (1.2.3 including **zlib-devel**), **libpcap** (1.0.0 including **libpcap-devel**), **pcre** (7.84 including **pcre-devel**), **libdnet** (1.11 or 1.12 including **libdnet-devel**) and **tcpdump** (4.1.0). Versions of these packages already installed may be newer than what is listed here, but should NOT cause any issues when compiling DAQ and/or SNORT.

When upgrading to the newest version of SNORT, it is **strongly recommended** to **back up local.rules, snort.conf, threshold.conf**, **white_list.rules**, and **black_list.rules** before any snort upgrade is installed.

Note: The steps in this document should apply to compiling DAQ 2.0.x and SNORT 2.9.6.x without any changes in actual configuration or makefiles (except the paths to the actual source files, etc).

To obtain the CentOS 6.3/6.4 (64-bit) versions of **libpcap-devel**, **libdnet,** and **libdnet-devel**, the filenames I used for the packages (via a google search) were:

libpcap-devel-1.0.0-6.20091201git117cb5.el6.x86_64.rpm
libdnet-devel-1.12-6.el6.x86_64.rpm
libdnet-debuginfo-1.12-6.choon.centos6.x86_64.rpm

using 'rpm –i' to install the 'libpcap' and 'libdnet-debuginfo' RPM's

The libdnet-devel package **failed to install** due to **dependency issues**, so I downloaded libdnet-1.11.tar.gz and used it to build working dnet libraries and header files (more on this below).

Obtain **SNORT** (version 2.9.7.x), **DAQ** (version 2.0.x), and snort rules from www.snort.org and download them to your CentOS 6.3/6.4 box.

The steps below will require '**root**' access and terminal/console access in order to successfully complete the compilation, installation, and running of SNORT on your CentOS 6.3/6.4 box.

Type the following commands in a terminal window (if you were able to find a suitable libdnet-devel RPM for CentOS 6.3/6.4, skip unpacking of the libdnet tarball below):

cd /usr/local/src <enter>
tar -zxvf <path to>libdnet-1.11.tar.gz <enter>
tar -zxvf <path to>daq-2.0.x.tar.gz <enter>
tar -zxvf <path to>snort-2.9.7.x.tar.gz <enter>

If you were able to find a suitable libdnet-devel RPM for CentOS 6.3/6.4, skip the section below and go to the next section to configure/compile/install DAQ-2.0.x.

First, let's configure, compile, and install libdnet:

cd /usr/local/src/libdnet-1.11 <enter>
./configure –with-pic <enter>
make <enter>
make install <enter>

Note any errors which may cause the 'configure' step to abort, also, you can check the file '**config.log**' which is generated from the 'configure' line above.

cd /usr/local/lib <enter>
ldconfig –v /usr/local/lib <enter>

Now we will configure, compile, and install DAQ-2.0.x:

cd /usr/local/src/daq-2.0.x <enter>
./configure <enter>
make <enter>
make install <enter>

Note any errors which may cause the 'configure' step to abort, also, you can check the file '**config.log**' which is generated from the 'configure' line above.

cd /usr/local/lib <enter>
ldconfig –v /usr/local/lib <enter>

Do the following to compile SNORT 2.9.7.x on your CentOS 6.3/6.4 system:

Note: Joel Esler at Sourcefire rcommends the use of the --enable-sourcefire option

cd /usr/local/src/snort-2.9.7.x <enter>
./configure --enable-sourcefire <enter>
make <enter>
make install <enter>

cd /usr/local/lib <enter>
ldconfig –v /usr/local/lib <enter>

Note any errors which may cause the 'configure' step to abort, also, you can check the file '**config.log**' which is generated from the 'configure' line above.

In order to download snort rules from www.snort.org, you must be a **registered user** or have a **paid subscription** to download rule sets or VRT rules. Information can be found at www.snort.org on how to become a **registered user**. **Registered users** will be able to download rule sets which are **approximately one month behind** what is available to paid subscription holders.

Issue the commands below:

cd /etc <enter>
mkdir -p snort <enter>
cd snort <enter>
cp /usr/local/src/snort-2.9.7.x/etc/* . <enter>
tar -zvxf <path to>snortrules-snapshot-<nnnn>.tar.gz <enter>
cp ./etc/* . <enter>
touch /etc/snort/rules/white_list.rules /etc/snort/rules/black_list.rules <enter>

Note - this will place the configuration files from the snort 2.9.7.x unpack and the rules snapshot under the /etc/snort directory. If the rules snapshot file is newer, this is not an issue (since rules are updated on a periodic basis by the snort team).

Also, the configuration files (e,g, - snort.conf, threshold.conf, etc) are residing in /etc/snort/ and the rules files will be in /etc/snort/rules and for the so_ and preprocessor rules, these will be located in /etc/snort

Add a user and group for snort in your system (using the commands below):

groupadd -g 40000 snort <enter>
useradd snort -u 40000 -d /var/log/snort -s /sbin/nologin -c SNORT_IDS –g snort <enter>
cd /etc/snort <enter>
chown -R snort:snort * <enter>
chown -R snort:snort /var/log/snort <enter>

Locate and modify the following variables in your snort.conf file
(in directory /etc/snort) as follows (usually between lines 40 and 120):

This assumes the network you are going to monitor is 192.168.1.0/24

var RULE_PATH /etc/snort/rules
ipvar HOME_NET 192.168.1.0/24
ipvar EXTERNAL_NET !$HOME_NET
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules
var WHITE_LIST_PATH /etc/snort/rules
var BLACK_LIST_PATH /etc/snort/rules

The following commands should be used to take ownership of directories and change file permissions that are related to SNORT and/or DAQ.

cd /usr/local/src <enter>
chown -R snort:snort daq-2.0.x <enter>
chmod -R 700 daq-2.0.x <enter>
chown -R snort:snort snort-2.9.7.x <enter>
chmod -R 700 snort-2.9.7.x <enter>
chown -R snort:snort snort_dynamicsrc <enter>
chmod -R 700 snort_dynamicsrc <enter>

The snort initialization script on the next page is something which was put together from an existing script in CentOS 6.3/6.4's /etc/init.d directory. It is still a work in progress, but it will allow you to start, stop, restart, and give the status of snort on your system. As improvements are made to the script, it will be updated in this document. Also, if anyone has improvements to the script they would like to have incorporated into this document, please email me at the address at the bottom of this document.

Note – some users have reported problems with the script below, but fortunately, the snort-2.9.7.x/2.9.6.x source code tree has a directory called 'RPM' which has a shell script called 'snortd' which can be copied to /etc/init.d and named 'snort' (with appropriate permissions, of course) which will allow snort to be started from /etc/init.d.

Also, at www.snort.org/docs there are a set of initialization scripts which are available for various operating systems, including CentOS 6.3/6.4. These scripts are available due to the fact that some users have reported problems copying and pasting the script below when it is in the form of a PDF document.

This script can also be added to the existing scripts CentOS 6.3/6.4 knows about via the 'chkconfig' command, to do so issue the command below:

chkconfig --add snort <enter>

Doing this will set automatic startup in runlevels 2, 3, 4, and 5 on your CentOS 6.3/6.4 system.

Place the shell script below into the /etc/init.d directory on your CentOS 6.3/6.4 box:

```
----- CUT HERE -----
#!/bin/bash
#
# snort            Start up the SNORT Intrusion Detection System daemon
#
# chkconfig: 2345 55 25
# description: SNORT is a Open Source Intrusion Detection System
#              This service starts up the snort daemon.
#
# processname: snort
# pidfile: /var/run/snort_eth0.pid

### BEGIN INIT INFO
# Provides: snort
# Required-Start: $local_fs $network $syslog
# Required-Stop: $local_fs $syslog
# Should-Start: $syslog
# Should-Stop: $network $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Start up the SNORT Intrusion Detection System daemon
# Description:       SNORT is an application for Open Source Intrusion Detection.
#                    This service starts up the Snort IDS daemon.
### END INIT INFO

# source function library
. /etc/rc.d/init.d/functions

# pull in sysconfig settings
[ -f /etc/sysconfig/snort ] && . /etc/sysconfig/snort

RETVAL=0
prog="snort"
lockfile=/var/lock/subsys/$prog

# Some functions to make the below more readable
SNORTD=/usr/local/bin/snort
#OPTIONS="-A fast -b -d -D -i eth0 -u snort -g snort -c /etc/snort/snort.conf -l
/var/log/snort"
#PID_FILE=/var/run/snort_eth0.pid

# Convert the /etc/sysconfig/snort settings to something snort can
# use on the startup line.
```

```
if [ "$ALERTMODE"X = "X" ]; then
  ALERTMODE=""
else
  ALERTMODE="-A $ALERTMODE"
fi

if [ "$USER"X = "X" ]; then
  USER="snort"
fi

if [ "$GROUP"X = "X" ]; then
  GROUP="snort"
fi

if [ "$BINARY_LOG"X = "1X" ]; then
  BINARY_LOG="-b"
else
  BINARY_LOG=""
fi

if [ "$LINK_LAYER"X = "1X" ]; then
  LINK_LAYER="-e"
else
  LINK_LAYER=""
fi

if [ "$CONF"X = "X" ]; then
  CONF="-c /etc/snort/snort.conf"
else
  CONF="-c $CONF"
fi

if [ "$INTERFACE"X = "X" ]; then
  INTERFACE="-i eth0"
  PID_FILE="/var/run/snort_eth0.pid"
else
  PID_FILE="/var/run/snort_$INTERFACE.pid"
  INTERFACE="-i $INTERFACE"
fi

if [ "$DUMP_APP"X = "1X" ]; then
  DUMP_APP="-d"
else
  DUMP_APP=""
fi
```

```
if [ "$NO_PACKET_LOG"X = "1X" ]; then
  NO_PACKET_LOG="-N"
else
  NO_PACKET_LOG=""
fi

if [ "$PRINT_INTERFACE"X = "1X" ]; then
  PRINT_INTERFACE="-I"
else
  PRINT_INTERFACE=""
fi

if [ "$PASS_FIRST"X = "1X" ]; then
  PASS_FIRST="-o"
else
  PASS_FIRST=""
fi

if [ "$LOGDIR"X = "X" ]; then
  LOGDIR=/var/log/snort
fi

# These are used by the 'stats' option
if [ "$SYSLOG"X = "X" ]; then
  SYSLOG=/var/log/messages
fi

if [ "$SECS"X = "X" ]; then
  SECS=5
fi

if [ ! "$BPFFILE"X = "X" ]; then
  BPFFILE="-F $BPFFILE"
fi

runlevel=$(set -- $(runlevel); eval "echo \$$#" )

start()
{
        [ -x $SNORTD ] || exit 5

        echo -n $"Starting $prog: "
        daemon --pidfile=$PID_FILE $SNORTD $ALERTMODE $BINARY_LOG
$LINK_LAYER $NO_PACKET_LOG $DUMP_APP -D $PRINT_INTERFACE
```

```
$INTERFACE -u $USER -g $GROUP $CONF -l $LOGDIR $PASS_FIRST $BPFFILE
$BPF && success || failure
        RETVAL=$?
        [ $RETVAL -eq 0 ] && touch $lockfile
        echo
        return $RETVAL
}

stop()
{
        echo -n $"Stopping $prog: "
        killproc $SNORTD
        if [ -e $PID_FILE ]; then
           chown -R $USER:$GROUP /var/run/snort_eth0.* &&
           rm -f /var/run/snort_eth0.pi*
        fi
        RETVAL=$?
        # if we are in halt or reboot runlevel kill all running sessions
        # so the TCP connections are closed cleanly
        if [ "x$runlevel" = x0 -o "x$runlevel" = x6 ] ; then
           trap " TERM
           killall $prog 2>/dev/null
           trap TERM
        fi
        [ $RETVAL -eq 0 ] && rm -f $lockfile
        echo
        return $RETVAL
}

restart() {
        stop
        start
}

rh_status() {
        status -p $PID_FILE $SNORTD
}

rh_status_q() {
        rh_status >/dev/null 2>&1
}

case "$1" in
        start)
                rh_status_q && exit 0
```

```
                start
                ;;
        stop)
                if ! rh_status_q; then
                        rm -f $lockfile
                        exit 0
                fi
                stop
                ;;
        restart)
                restart
                ;;
        status)
                rh_status
                RETVAL=$?
                if [ $RETVAL -eq 3 -a -f $lockfile ] ; then
                        RETVAL=2
                fi
                ;;
        *)
                echo $"Usage: $0 {start|stop|restart|status}"
                RETVAL=2
esac
exit $RETVAL
```

----- CUT HERE -----

To make the symbolic link (symlink) for snort, issue the commands below:

cd /usr/sbin <enter>
ln -s /usr/local/bin/snort snort <enter>

The file below should be named 'snort' and placed into the /etc/sysconfig directory on your CentOS 6.3/6.4 system:

----- CUT HERE -----
/etc/sysconfig/snort
$Id: snort.sysconfig,v 1.8 2003/09/19 05:18:12 dwittenb Exp $

General Configuration

INTERFACE=eth0
CONF=/etc/snort/snort.conf
USER=snort
GROUP=snort
PASS_FIRST=0

Logging & Alerting

LOGDIR=/var/log/snort
ALERTMODE=fast
DUMP_APP=1
BINARY_LOG=1
NO_PACKET_LOG=0
PRINT_INTERFACE=0

----- CUT HERE -----

Note: The above file should be owned by user/group 'snort' with permissions '700'

If the directory '/var/log/snort' does not exist on your system, issue the following commands as the 'root' user (permissions should be 700), the commands below will also change the ownership of the directories and files to user 'snort' and group 'snort'.

cd /var/log <enter>
mkdir snort <enter>
chmod 700 snort <enter>
chown -R snort:snort snort <enter>
cd /usr/local/lib <enter>
chown -R snort:snort snort* <enter>
chown -R snort:snort snort_dynamic* <enter>
chown -R snort:snort pkgconfig <enter>
chmod -R 700 snort* <enter>
chmod -R 700 pkgconfig <enter>
cd /usr/local/bin <enter>
chown -R snort:snort daq-modules-config <enter>
chown -R snort:snort u2* <enter>
chmod -R 700 daq-modules-config <enter>
chmod 700 u2* <enter>
cd /etc <enter>
chown -R snort:snort snort <enter>
chmod -R 700 snort <enter>

At this point, you should be ready to do some testing of SNORT to see if it actually starts up and reads in the rules (you can check /var/log/messages to catch any fatal errors or crashes).

If you want to test SNORT startup, issue the following commands:

cd /usr/local/bin <enter>
./snort -T -i eth0 -u snort -g snort -c /etc/snort/snort.conf <enter>

The above command will cause SNORT to start up in self-test mode, checking all the supplied command line switches and rules files that are passed to it and indicating that everything is ready to proceed. If all the tests are passed, you should see the following:

Snort successfully validated the configuration!
Snort exiting

Here are some **common errors** that **snort may return** when running on CentOS 6.x:

ERROR: snort.conf(253) Could not stat dynamic module path
"/usr/local/lib/snort_dynamicrules": No such file or directory.
Fatal Error, Quitting.

Solution below:

mkdir -p /usr/local/lib/snort_dynamicrules <enter>
chown -R snort:snort /usr/local/lib/snort_dynamicrules <enter>
chmod -R 700 /usr/local/lib/snort_dynamicrules <enter>

Copy any dynamic rulesets you have or are using to the above directory.

Another method would be to comment out that line in snort.conf if you have no dynamic rules in use.

ERROR: /etc/snort/rules/web-misc.rules(555) Cannot use the fast_pattern content modifier for a lone http cookie/http raw uri /http raw header /http raw cookie /status code / status msg /http method buffer content.
Fatal Error, Quitting.

Solution below:

The fast_pattern option cannot be used with the http_method string. Edit the web-misc.rules file and remove it from the snort rule. Do a search for "2010-0388" and remove the alert option fast_pattern from the alert rule.

ERROR: /etc/snort/snort.conf(244) => 'compress_depth' and 'decompress_depth' should be set to max in the default policy to enable 'unlimited_decompress'
Fatal Error, Quitting.

Solution below:

Edit the /etc/snort/snort.conf file and set the http_inspect compress_depth and decompress_depth to 65535 from 20480.

If no errors are returned, proceed with the steps below (otherwise check /var/log/messages for more information):

To manually start snort, issue the following commands:

cd /usr/local/bin <enter> (if you are already in this directory, skip this command)
./snort -A fast -b -d -D -i eth0 -u snort -g snort -c /etc/snort/snort.conf -l /var/log/snort <enter>

Make sure that snort initializes properly before proceeding below, you can check /var/log/messages for more information in the event of an error in initialization.

To see if snort is actually running on your system, issue the following command:

ps aux | grep -i "snort" <enter>

If snort is working, it should return something that looks like the output below:

19235 ? Ssl 0:06 /usr/sbin/snort -A fast -b -d -D -i eth0 -u snort -g snort -c /etc/snort/snort.conf -l /var/log/snort

Tips to improve the security of SNORT while running on Linux (all flavors):

Here are some suggestions to lessen the impact that a vulnerability discovered in SNORT would give potential unauthorized access to a privileged account:

1. When running SNORT in daemon (-D) mode, the '-u' (user) and '-g' (group) switches should be used. This will allow SNORT to run as a given user and group after it is initialized. Typically, most system administrators prefer to add the 'snort' user and group to their systems, and that the 'snort' user should be unable initiate a login or shell privileges. Here is an example of a 'snort' user on a Linux system:

snort:x:501:501:SNORT_IDS:/var/log/snort:/sbin/nologin

In the above example, the line is broken down as follows:

Columns 1-5 (the username, in this case 'snort')
Column 7 (the 'x' indicates that the password is encrypted)
Columns 9-11 (the user id (UID) 501)
Columns 13-15 (the group id (GID) 501, in this case the group is 'snort')
Columns 17-26 (the full name of the user, in this case 'SNORT_IDS')
Columns 28-41 (the default directory for this user, in this case '/var/log/snort')

The /sbin/nologin at the end of the line shows that logins are disabled for the 'snort' user on this system.

2. The source code for SNORT/DAQ, binaries, logging directories, shared/static libraries, and configuration files should all be owned by the 'snort' user and group with appropriate permissions (mode 700 is preferred).

3. All binaries which are produced by the compiling and installation process of SNORT and DAQ should be verified using a hash function (i.e. - MD5, SHA-1, etc) and the output stored on removable media. A cron job could be used to run this process on a regular basis with results emailed to a system administrator. Another alternative would be the use of a utility called 'tripwire' for auditing installed software on a given computer.

I have separated the information for mirroring and/or copying packets from a home router to a snort sensor to a separate document located at the following URL:

www.snort.org/docs

Under the section marked 'Deployment Guides' and the link is marked:

How to make some home routers mirror traffic to Snort

Finally, if you have SNORT working in test mode (-T option), try starting SNORT with /etc/init.d/snort start (you should get a running message if all is well). If there is a problem, check the output in /var/log/messages for additional details as to why snort failed to start.

Also, you can check the status of snort by issuing the command below (while still in /etc/init.d):

./snort status <enter>

If it's working, you should see the output below:

Checking for service snort running

Next, change directory to /var/log/snort and issue the command 'ls -al' if everything is working properly, you should see two (or more) files, one marked 'alert' and 'snort.*' files (which are binary captures which can be read with tcpdump or wireshark). If you use 'tail -f alert' in your terminal/console window, you should see alerts coming into your snort IDS (as they occur).

If you have any questions, comments, or suggestions, please email me at:

wp02855@gmail.com (wp02855 at gmail dot com)

Bill Parker