

Getting SNORT working in Fedora Core 17/18/19 and VirtualBox 5.x.x.

Last revised on December 24, 2015

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-2.9.8.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 compiling and installing SNORT 2.9.8.x and DAQ 2.0.x using the Hardware and Operating System(s) listed below:

Microsoft Windows 7 Professional Edition w/SP1 as the HOST operating system

VirtualBox 5.x.x with Oracle Extension Pack 5.x.x (I use version 5.x.1x)

Fedora Core 17 or 18 or 19 (64-bit version) as the GUEST operating system

SNORT 2.9.8.x, DAQ 2.0.x, and a set of snort rules (www.snort.org)

The hardware in the HOST system listed above is a Turion TL-58 processor (AMD) @ 2.0Ghz, 4GB of 667Mhz SO-DIMM RAM, and a onboard Marvel Yukon PCIe Gigabit Ethernet 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/5.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 **Fedora Core 17/18/19 Virtual Machine**, you will need to set the **NETWORK** section to **BRIDGED** mode to allow the assignment of a static IP to your Fedora Core 17/18/19 VM (if you are using a standalone system running Fedora Core 17/18/19 you can ignore this step).

Configure your **Static IP**, **Network Mask**, **DNS**, and **Gateway** in **Network Settings** for Fedora Core 17/18/19 or by editing `/etc/sysconfig/network-scripts/ifcfg-<interface>` (in my case, I used **ethernet 0 (p2p1)** as the port to monitor traffic on).

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 Fedora Core 17/18/19 (www.snort.org) would be a good site to visit (shameless plug here).

Make sure the following packages are installed in your [Fedora Core 17/18/19](#) system via [rpm](#) or online updates: [gcc](#) version 4.7.x (including libraries), [flex](#) (2.5.x), [bison](#) (2.5.x/2.6.x), [zlib](#) (1.2.x including [zlib-devel](#)), [libpcap](#) (1.2.1/1.3.0/1.4.0 including [libpcap-devel](#)), [pcre](#) (8.2x/8.3x including [pcre-devel](#)), [libdnet](#) (1.12 including [libdnet-devel](#)) and [tcpdump](#) (4.2.x). 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 the upgrade is installed.

Note: The steps in this document should apply to compiling [DAQ-2.0.x](#) and [SNORT 2.9.7.x](#) without any changes in actual configuration or makefiles (except the paths to the actual source files, etc).

You will need the following packages to complete the installation of [DAQ-2.0.x](#) and [Snort 2.9.8.x](#) on [Fedora Core 17/18/19](#) (the ISO I downloaded lacked the necessary development packages even after the install was completed):

If you are installing Fedora Core 17, the following packages should be downloaded:

[libpcap-1.2.1-2.fc17.x86_64.rpm](#)
[libpcap-devel-1.2.1-2.fc17.x86_64.rpm](#)
[pcre-8.21-3.fc17.x86_64.rpm](#)
[pcre-devel-8.21-3.fc17.x86_64.rpm](#)
[libdnet-1.12-8.fc17.x86_64.rpm](#)
[libdnet-devel-1.12-8.fc17.x86_64.rpm](#)

If you are installing Fedora Core 18, the following packages should be downloaded:

[libpcap-1.3.0-2.fc18.x86_64.rpm](#)
[libpcap-devel-1.3.0-2.fc18.x86_64.rpm](#)
[pcre-8.31-2.fc18.x86_64.rpm](#)
[pcre-devel-8.31-2.fc18.x86_64.rpm](#)
[libdnet-1.12-9.fc18.x86_64.rpm](#)
[libdnet-devel-1.12-9.fc18.x86_64.rpm](#)

If you are installing Fedora Core 19, the following packages should be downloaded:

```
libpcap-1.4.0-1.fc19.x86_64.rpm  
libpcap-devel-1.4.0-1.fc19.x86_64.rpm  
pcre-8.32-7.fc19.x86_64.rpm  
pcre-devel-8.32-7.fc19.x86_64.rpm  
libdnet-1.12-11.fc19.x86_64.rpm  
libdnet-devel-1.12-11.fc19.x86_64.rpm
```

These can be found via the 'rpm.pbone.net' site, or you can do a google search for the above filenames (or via online updates/add or remove software).

Before starting use the following commands to see if the necessary packages are installed:

```
rpm -qa | grep -i "libpcap" <enter>  
rpm -qa | grep -i "pcre" <enter>  
rpm -qa | grep -i "dnet" <enter>  
rpm -qa | grep -i "flex" <enter>  
rpm -qa | grep -i "bison" <enter>  
rpm -qa | grep -i "zlib" <enter>  
rpm -qa | grep -i "tcpdump" <enter>
```

If any of the packages are missing (which it will tell you), search for the packages via the 'rpm.pbone.net' site, google search, or online updating via [Fedora Core 17/18/19](#), and then install the downloaded packages via: `rpm -i <name of the package>`

If you added the packages above via '`rpm -i`' or '[online update](#)', make sure you run the command below:

```
ldconfig -v /usr/lib <enter>
```

Obtain **SNORT** (version 2.9.8.x), **DAQ** (version 2.0.x), and snort rules from www.snort.org and download them to your Fedora Core 17/18/19 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 [Fedora Core 17/18/19](#) system.

```
cd /usr/local/src <enter>  
tar -zxvf <path to>daq-2.0.x.tar.gz <enter>  
tar -zxvf <path to>snort-2.9.8.x.tar.gz <enter>
```

Do the following to compile DAQ on your Fedora Core 17/18/19 system::

```
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.

Do the following to compile SNORT on your Fedora Core 17/18/19 system:

```
cd /usr/local/src/snort-2.9.8.x <enter>
./configure --enable-sourcefire <enter> (Note: Joel Esler at Sourcefire rcommends this)
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.

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 that 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-2.9.8.x/etc/* . <enter>
tar -zxvf <path to>snorrules-snapshot-<nnnn>.tar.gz <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 2.9.8.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 the following directory to /usr/local/lib:

```
cd /usr/local/lib <enter>
mkdir snort_dynamicrules <enter>
```

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

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

Also, at www.snort.org/docs there are a set of initialization scripts which are available for various operating systems, including [Fedora 17/18/19](#). 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.

Place the shell script below into the /etc/init.d directory on your Fedora Core 17/18/19 box:

```
----- CUT HERE -----
#!/bin/sh
# $Id$
#
# snortd      Start/Stop the snort IDS daemon.
#
# chkconfig: 2345 40 60
# description: snort is a lightweight network intrusion detection tool that
#              currently detects more than 1100 host and network
#              vulnerabilities, portscans, backdoors, and more.
#
# Source function library.
. /etc/rc.d/init.d/functions

# Source the local configuration file
. /etc/sysconfig/snort

# 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 p2p1"
else
    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

#####
# Now to the real heart of the matter:

# See how we were called.
case "$1" in
    start)
        echo -n "Starting snort: "
        cd $LOGDIR
        if [ "$INTERFACE" = "-i ALL" ]; then
            for i in `cat /proc/net/dev|grep eth|awk -F ":" '{ print $1; }'`
            do
                mkdir -p "$LOGDIR/$i"
                chown -R $USER:$GROUP $LOGDIR
                daemon /usr/sbin/snort $ALERTMODE $BINARY_LOG $LINK_LAYER
                $NO_PACKET_LOG $DUMP_APP -D $PRINT_INTERFACE -i $i -u $USER -g
                $GROUP $CONF -I $LOGDIR/$i $PASS_FIRST $BPFFILE $BPF
            done
        else
            # check if more than one interface is given
            if [ `echo $INTERFACE|wc -w` -gt 2 ]; then
                for i in `echo $INTERFACE | sed s/"-i "/`
                do
                    mkdir -p "$LOGDIR/$i"
                    chown -R $USER:$GROUP $LOGDIR
                    daemon /usr/sbin/snort $ALERTMODE $BINARY_LOG $LINK_LAYER
                    $NO_PACKET_LOG $DUMP_APP -D $PRINT_INTERFACE -i $i -u $USER -g
                    $GROUP $CONF -I $LOGDIR/$i $PASS_FIRST $BPFFILE $BPF
                done
            else
                # Run with a single interface (default)

```

```

        daemon /usr/sbin/snort $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
    fi
    fi
    touch /var/lock/subsys/snort
    echo
    ;;
stop)
    echo -n "Stopping snort: "
    killproc snort
    rm -f /var/lock/subsys/snort
    echo
    ;;
reload)
    echo "Sorry, not implemented yet"
    ;;
restart)
    $0 stop
    $0 start
    ;;
condrestart)
    [ -e /var/lock/subsys/snort ] && $0 restart
    ;;
status)
    status snort
    ;;
stats)
    TC=125                # Trailing context to grep
    SNORTNAME='snort'    # Process name to look for

    if [ ! -x "/sbin/pidof" ]; then
        echo "/sbin/pidof not present, sorry, I cannot go on like this!"
        exit 1
    fi

    #Grab Snort's PID
    PID=`pidof -o $$ -o $PPID -o %PPID -x ${SNORTNAME}`

    if [ ! -n "$PID" ]; then    # if we got no PID then:
        echo "No PID found: ${SNORTNAME} must not running."
        exit 2
    fi

    echo ""
    echo "*****"

```

```

echo "WARNING: This feature is EXPERIMENTAL - please report errors!"
echo "*****"
echo ""
echo "You can also run: $0 stats [long | opt]"
echo ""
echo "Dumping ${SNORTNAME}'s ($PID) statistics"
echo "please wait..."

# Get the date and tell Snort to dump stats as close together in
# time as possible--not 100%, but it seems to work.
startdate=`date '+%b %e %H:%M:%S'`

# This causes the stats to be dumped to syslog
kill -USR1 $PID

# Sleep for $SECS secs to give syslog a chance to catch up
# May need to be adjusted for slow/busy systems
sleep $SECS

if [ "$2" = "long" ]; then      # Long format
    egrep -B 3 -A $TC "^$startdate .* snort.*:={79}" $SYSLOG | \
        grep snort.*:
elif [ "$2" = "opt" ]; then    # OPTimize format
    # Just show stuff useful for optimizing Snort
    egrep -B 3 -A $TC "^$startdate .* snort.*:={79}" $SYSLOG | \
        egrep "snort.*: Snort analyzed |snort.*: dropping|emory .aults:"
else                            # Default format
    egrep -B 3 -A $TC "^$startdate .* snort.*:={79}" $SYSLOG | \
        grep snort.*: | cut -d: -f4-
fi
;;
*)
echo "Usage: $0 {start|stop|reload|restart|condrestart|status|stats (long|opt)}"
exit 2
esac

exit 0
----- CUT HERE -----

```

Note - On the above script, I made a symlink in [/usr/sbin](#) to point to where the actual SNORT binary was compiled on my system (you could also copy the snort binary to [/usr/sbin](#) as well).

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

```
cd /usr/sbin <enter>
ln -s /usr/local/bin/snort snort <enter>
chmod 700 snort <enter>
```

The script file below should be named 'snort' and placed into the [/etc/sysconfig](#) directory on your Fedora Core 17/18/19 system:

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

#### General Configuration

INTERFACE=p2p1
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
LINK_LAYER=0
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 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 p2p1 -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

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 -i p2p1 -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:

```
snort  1212  0.0  16.5 461924 126328 ?    Ssl  21:26   0:00 /usr/sbin/snort -A fast -b -d
-D -i p2p1 -u snort -g snort -c /etc/snort/snort.conf -l /var/log/snort
```

Tips to improve the security of SNORT while running on Fedora Core Linux:

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:1001:1000:SNORT_IDS:/var/log/snort:/bin/false
```

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 9-12 (the user id (UID) 1001)

Columns 14-17 (the group id (GID) 1000, in this case the group is 'snort')

Columns 19-27 (the full name of the user, in this case 'SNORT_IDS')

Columns 29-43 (the default directory for this user)

The /bin/false 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