

Getting SNORT working in OpenSuSE 13.x 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 5.0.1x)
OpenSuSE 13.x (32 or 64 bit) as the GUEST operating system (which runs SNORT)
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 **OpenSuSE 13.x Virtual Machine**, you will need to set the **NETWORK** section to **BRIDGED** mode to allow the assignment of a static IP to your OpenSuSE 13.x VM (if you are using a standalone system running OpenSuSE 13.x you can ignore this step).

Configure your **Static IP**, **Network Mask**, **DNS**, and **Gateway** in **YAST | SYSTEM | Network Settings** for OpenSuSE 13.x (in my case, I used **ethernet 0 (eth0)** 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 OpenSuSE 13.x (www.snort.org) would be a good site to visit (shameless plug here).

Make sure the following packages are installed in your OpenSuSE 13.x system via YAST: **gcc** version 4.8.x (including libraries), **flex** (2.5.37), **bison** (2.7), **zlib** (1.2.8 including **zlib-devel**), **libpcap** (1.3.x including **libpcap-devel**), **pcre** (8.33 including **pcre-devel**), **libdnet** (1.12 including **libdnet-devel**) and **tcpdump** (4.4.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.

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).

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.

To obtain the OpenSuSE 13.x (32/64-bit) versions of **libpcap-devel** and **libdnet-devel**, you can go to the following URL which is maintained by OpenSuSE for direct download (for some strange reason, these were not included in the installation DVD I downloaded for OpenSuSE 13.x). They can be installed (as 'root') using the 'rpm -i' command:

<http://download.opensuse.org/distribution/13.x/repo/oss/suse/i586/> (32-bit)

<http://download.opensuse.org/distribution/13.x/repo/oss/suse/x86-64/> (64-bit)

Obtain **SNORT** (version 2.9.8.x), **DAQ** (version 2.0.x), and snort rules from www.snort.org and download them to your OpenSuSE 13.x 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 OpenSuSE 13.x box.

First, unpack the source code for DAQ 2.0.x and Snort 2.9.8.x:

```
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 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 and also check the file 'config.log' which is generated from the 'configure' line above.

Update the dynamic linker run-time bindings with the commands below:

```
ldconfig -v /usr/local/lib <enter>
ldconfig -v /usr/local/lib64 <enter> (for 64 bit OpenSuSE installs)
```

Do the following to compile SNORT 2.9.8.x:

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

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

Update the dynamic linker run-time bindings with the commands below:

```
ldconfig -v /usr/local/lib <enter>
ldconfig -v /usr/local/lib64 <enter> (for 64 bit OpenSuSE installs)
```

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-2.9.8.x/etc/* . <enter>
tar -zxvf <path to>snortrules-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 a user and group for snort in your system (using the commands below) or use YAST:

```
useradd snort -d /var/log/snort -s /bin/false -c SNORT_IDS <enter>
groupadd snort <enter>
```

Use the commands below to take ownership of all files in `/etc/snort`:

```
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
```

Special Note: On 64-bit OpenSuSE installs, the library files for DAQ-2.0.x and Snort-2.9.8.x (preprocessors, etc) **may be installed** in `/usr/local/lib64` rather than `/usr/local/lib`, if you get errors during testing (with `-T` option) being unable to load various preprocessors or on snort startup in daemon mode, check the date/timestamps for DAQ/SNORT libraries in `/usr/local/lib` and `/usr/local/lib64`, and adjust path values for these libraries in `snort.conf` accordingly (if this is needed).

Also, at www.snort.org/docs there are a set of initialization scripts which are available for various operating systems, including [OpenSuSE 13.x](#). 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 OpenSuSE 13.x box:

```
----- CUT HERE -----
#!/bin/sh
#
# /etc/init.d/snortd
# and its symbolic link
# /usr/sbin/rcsnortd
#
###
### adapted to openSUSE 11.0 by hans @ www.kriyayoga.com
### December 13 2008
### this script also works on OpenSUSE 12.x as written
### use as is - use at your own risk
### report bugs in THIS snortd init-script to hans@kriyayoga.com
###
### BEGIN INIT INFO
# Provides:      snort
# Required-Start:  $syslog $remote_fs
# Required-Stop:  $syslog $remote_fs
# Default-Start:  3 5
# Default-Stop:   0 1 2 6
# Short-Description: Start snort
# Description:    Start snort IDS
### END INIT INFO
PATH=/usr/sbin:/usr/bin:/usr/sbin:/sbin:/usr/sbin:/usr/bin:/bin
SNORT_BIN=/usr/sbin/snort
SNORT_SOCKET=/var/run/snort_eth0.pid

test -x $SNORT_BIN || { echo "$SNORT_BIN not installed";
    if [ "$1" = "stop" ]; then exit 0;
    else exit 5; fi; }

# Check for existence of needed config file and read it
SNORT_CONFIG=/etc/snort/snort.conf
test -r $SNORT_CONFIG || { echo "$SNORT_CONFIG not existing";
    if [ "$1" = "stop" ]; then exit 0;
    else exit 6; fi; }

./etc/rc.status
```

```

# Shell functions sourced from /etc/rc.status:
# rc_check      check and set local and overall rc status
# rc_status     check and set local and overall rc status
# rc_status -v  ditto but be verbose in local rc status
# rc_status -v -r ditto and clear the local rc status
# rc_failed     set local and overall rc status to failed
# rc_reset      clear local rc status (overall remains)
# rc_exit       exit appropriate to overall rc status

# First reset status of this service

# Reset status of this service
rc_reset

# Source the local configuration file
SNORTD_SYSCONFIG=/etc/sysconfig/snort
test -r $SNORTD_SYSCONFIG || exit 6
. $SNORTD_SYSCONFIG

#. /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 eth0"
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)
        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
                chmod -R 700 $LOGDIR
                /sbin/startproc -p $SNORT_SOCKET $SNORT_BIN $ALERTMODE
                $BINARY_LOG $LINK_LAYER $NO_PACKET_LOG $DUMP_APP -D
                $PRINT_INTERFACE -i $i -u $USER -g $GROUP $CONF -l $LOGDIR/$i
                $PASS_FIRST $BPFFILE $BPF > /dev/null 2>&1
                # Remember status and be verbose
                rc_status -v
            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
                    chmod -R 700 $LOGDIR
                    /sbin/startproc -p $SNORT_SOCKET $SNORT_BIN $ALERTMODE
                    $BINARY_LOG $LINK_LAYER $NO_PACKET_LOG $DUMP_APP -D
                    $PRINT_INTERFACE -i $i -u $USER -g $GROUP $CONF -l $LOGDIR/$i
                    $PASS_FIRST $BPFFILE $BPF > /dev/null 2>&1
                done
            fi
        fi
    *)
        echo "Usage: $0 {start}"
        exit 1
esac
```



```

# Remember status and be verbose
rc_status -v
done
else
    # Run with a single interface (default)
    /sbin/startproc -p $SNORT_SOCKET $SNORT_BIN $ALERTMODE
$BINARY_LOG $LINK_LAYER $NO_PACKET_LOG $DUMP_APP -D
$PRINT_INTERFACE $INTERFACE -u $USER -g $GROUP $CONF -i $LOGDIR
$PASS_FIRST $BPFFILE $BPF > /dev/null 2>&1
    # Remember status and be verbose
    rc_status -v
    fi
fi
;;
stop)
    echo -n "Shutting down snort "
    /sbin/killproc $SNORT_BIN > /dev/null 2>&1
    chown -R $USER:$GROUP /var/run/snort_eth0.* &&
    rm -f /var/run/snort_eth0.pi*
    rc_status -v
    ;;
restart)
    $0 stop
    echo -n "starting snort - moment please "
    i=60
    while [ -e $SNORT_SOCKET ] && [ $i -gt 0 ]; do
        sleep 1
        i=$((i-1))
        echo -n "."
    done
    echo "."
    $0 start
    ;;
reload)
    echo "Sorry, not implemented yet"
    ;;
status)
    echo -n "Checking for service snort "
    /sbin/checkproc $SNORT_BIN
    rc_status -v
    ;;
## Check status with checkproc(8), if process is running
## checkproc will return with exit status 0.

# Status has a slightly different for the status command:

```

```
# 0 - service running
# 1 - service dead, but /var/run/pid file exists
# 2 - service dead, but /var/lock/lock file exists
# 3 - service not running
```

```
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
    grep -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|status|try-restart|restart|force-reload|reload|probe}"
    exit 1
    ;;
esac
rc_exit
----- 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 file below should be named 'snort' and placed into the `/etc/sysconfig` directory on your OpenSuSE 13.x 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  
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'

Below is the snort.service file for use in OpenSuSE 13.x, it should be placed in the directory `‘/usr/lib/systemd/system’` and named `‘snort.service’`:

--- CUT HERE ---

```
# the systemd script user/group snort, 700 rights
```

```
# This is the service file for systemd, place it in
```

```
# /usr/lib/systemd/system/snort.service
```

```
#
```

```
[Unit]
```

```
Description=Snort NIDS Daemon
```

```
After=syslog.target network.target
```

```
[Service]
```

```
Type=simple
```

```
ExecStart=/usr/local/bin/snort -A fast -b -d -i eth0 -u snort -g snort -c /etc/snort/snort.conf
```

```
-l /var/log/snort
```

```
[Install]
```

```
WantedBy=multi-user.target
```

--- 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 `‘root’` (permissions should be 700):

```
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>
mkdir snort_dynamicrules <enter>
chown -R snort:snort snort_* <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

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 -i eth0 -D -u snort -g snort -c /etc/snort/snort.conf <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:

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)

Columns 45-53 (the assigned shell or login 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