Snort Installation on openSUSE Leap 42.2 64 bits

Boris A. Gómez
Universidad Tecnológica de Panamá
July 2017

About This Guide

This guide has been tested on openSUSE Leap 42.2, 64 bits, using DAQ 2.0.6 and Snort 2.9.9.0.

Software was installed in a virtual machine:

Virtual Machine Manager: VirtualBox 5.1.22 or KVM 1.4.0
HOST operating system: Windows 7 or openSUSE Leap 42.2
GUEST operating system: openSUSE Leap 42.2 (Snort will be installed here)

For clarity, the following color code was used:

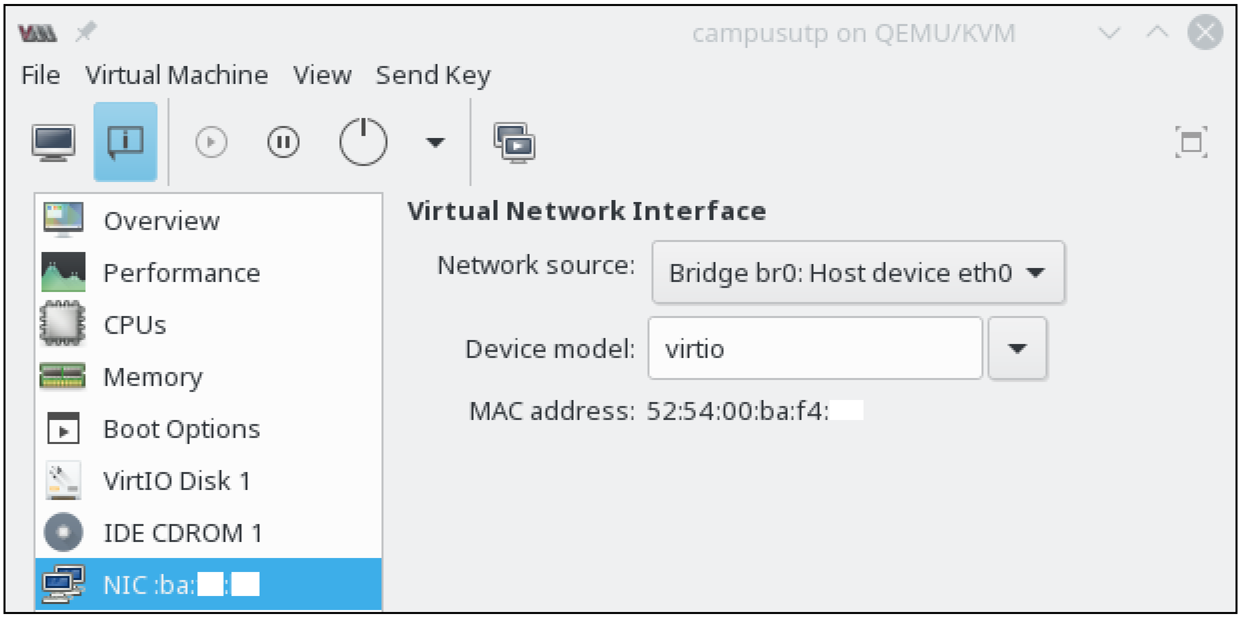
Orange – commands that the user types at the shell prompt.
Blue – text inside of configuration files.
Purple – text to focus your attention on.

This guide is based on the document "Snort 2.9.8.x on OpenSuSE 13x" by William Parker.

Network Card Configuration

Run **VirtualBox | KVM** manager and configure the network section of the **guest machine** to bridge mode.

KVM Manager:



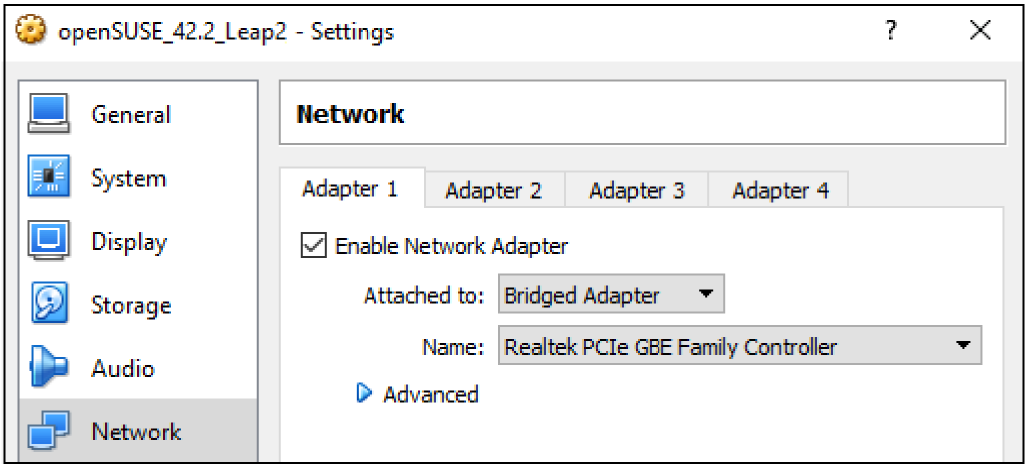
Virtual Network Interface

Network source: Bridge br0: Host device eth0

Device model: virtio

MAC address: 52:54:00:ba:f4:[redacted]

VirtualBox Manager:



Network

☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: Realtek PCIe GBE Family Controller

▷ Advanced

Guest Machine

Start your guest machine and set its network interface card to a static IP, for example 192.168.99.10, then check settings:

ifconfig

```
eth0   Link encap:Ethernet  HWaddr 08:00:27:50:CA:99
       inet addr: 192.168.99.10  Bcast:192.168.99.255  Mask:255.255.255.0
```

Verify that you can access Internet by accessing a web page, for example: https://snort.org

Before proceeding, it is advisable to update the system.

Required Packages

Use YAST to install the following packages:

gcc version 4.8.x (including libraries: **libgcc_s1** (5.3.1), **libgcc_s1-32bit**(5.3.1))
flex (2.5.37)
bison (2.7)
php5-zlib (5.5.14 including **zlib-devel** 1.2.8)
libpcap1 (1.8.1 including **libpcap-devel** 1.8.1) (versions must match)
libpcre1 (8.39 including **pcre-devel** 8.39 and **libpcre1-32bit** 8.39) (versions must match)
libdnet1 (1.12 including **libdnet-devel** 1.12) (versions must match)
tcpdump (4.5.1).

Installing DAQ and Snort

Download **DAQ** 2.0.6 and **Snort** 2.9.9.0:

```
wget -c https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
wget -c https://www.snort.org/downloads/snort/snort-2.9.9.0.tar.gz
```

note: files are saved in '/home/<user>/Downloads', where <user> is your username.

Open a Konsole terminal and switch to root:

```
su
```

```
Password:
```

enter root password.

Extract (untar) downloaded files:

```
cd /usr/local/src
tar -xzf /home/<user>/Downloads/daq-2.0.6.tar.gz
tar -xzf /home/<user>/Downloads/snort-2.9.9.0.tar.gz
```

Configure and install DAQ

```
cd /usr/local/src/daq-2.0.6
./configure
```

```
pc16:/usr/local/src/daq-2.0.6 # ./configure
configure: loading site script /usr/share/site/x86_64-unknown-
linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
```

The configure command must end with the following:

```
Build AFPacket DAQ module.. : yes
Build Dump DAQ module...... : yes
Build IPFW DAQ module...... : yes
Build IPQ DAQ module....... : no
Build NFQ DAQ module....... : no
Build PCAP DAQ module...... : yes
Build netmap DAQ module.... : no
pc16:/usr/local/src/daq-2.0.6 #
```

If it is different, check the config.log file:

```
tail -30 /usr/local/src/daq-2.0.6/config.log
```

```
#define HAVE_STRRCHR 1
#define HAVE_STRSTR 1
#define HAVE_STRTOUL 1

configure: exit 0
```

```
pc16:/usr/local/src/daq-2.0.6 #
```

Some errors may show up in the log but, in general, the final line = **exit 0**, indicates that the configuration went well.

make

```
pc16:/usr/local/src/daq-2.0.6 # make
make  all-recursive
make[1]: Entering directory '/usr/local/src/daq-2.0.6'
Making all in api
make[2]: Entering directory '/usr/local/src/daq-2.0.6/api'
/bin/sh ../libtool  --tag=CC   --mode=compile gcc -DHAVE_CONFIG_H -I. -I..
-I/usr/include  -g -O2 -fvisibility=hidden -Wall -Wwrite-strings -Wsign-compare -Wcast-
align -Wextra -Wformat -Wformat-security -Wno-unused-parameter -fno-strict-aliasing
-fdiagnost
```

...

```
make[2]: Leaving directory '/usr/local/src/daq-2.0.6/os-daq-modules'
make[2]: Entering directory '/usr/local/src/daq-2.0.6'
make[2]: Leaving directory '/usr/local/src/daq-2.0.6'
make[1]: Leaving directory '/usr/local/src/daq-2.0.6'
pc16:/usr/local/src/daq-2.0.6 #
```

make install

```
pc16:/usr/local/src/daq-2.0.6 # make install
Making install in api
make[1]: Entering directory '/usr/local/src/daq-2.0.6/api'
make[2]: Entering directory '/usr/local/src/daq-2.0.6/api'
 /usr/bin/mkdir -p '/usr/local/lib64'
 /bin/sh ../libtool   --mode=install /usr/bin/install -c   libdaq.la libdaq_static.la
'/usr/local/lib64'
libtool: install: /usr/bin/install -c .libs/libdaq.so.2.0.4 /usr/local/lib64/libdaq.so.2.0.4
```

...

```
make[2]: Entering directory '/usr/local/src/daq-2.0.6'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/local/src/daq-2.0.6'
make[1]: Leaving directory '/usr/local/src/daq-2.0.6'
pc16:/usr/local/src/daq-2.0.6 #
```

Configure and install Snort

cd /usr/local/src/snort-2.9.9.0

./configure --enable-sourcefire

```
pc16:/usr/local/src/snort-2.9.9.0 # ./configure --enable-sourcefire
configure: loading site script /usr/share/site/x86_64-unknown-linux-
gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
```

...

```
config.status: creating tools/file_server/Makefile
config.status: creating src/win32/Makefile
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands
pc16:/usr/local/src/snort-2.9.9.0 #
```

If it is different, check the config.log file:

tail -30 /usr/local/src/snort-2.9.9.0/config.log

```
#define HAVE_VISIBILITY 1
#define HAVE_ZLIB_H 1
#define HAVE_LIBZ 1
#define HAVE_YYLEX_DESTROY 1

configure: exit 0
```

Some errors may show up, but in general, the final line = **exit 0**, indicates that the configuration went well.

make

```
pc16:/usr/local/src/snort-2.9.9.0 # make
make all-recursive
make[1]: Entering directory '/usr/local/src/snort-2.9.9.0'
Making all in src
make[2]: Entering directory '/usr/local/src/snort-2.9.9.0/src'
Making all in sfutil
```

...

```
make[3]: Leaving directory '/usr/local/src/snort-2.9.9.0/tools'
make[2]: Leaving directory '/usr/local/src/snort-2.9.9.0/tools'
make[2]: Entering directory '/usr/local/src/snort-2.9.9.0'
make[2]: Leaving directory '/usr/local/src/snort-2.9.9.0'
make[1]: Leaving directory '/usr/local/src/snort-2.9.9.0'
pc16:/usr/local/src/snort-2.9.9.0 #
```

make install

```
 /usr/bin/mkdir -p '/usr/local/lib64/pkgconfig'
 /usr/bin/install -c -m 644 snort.pc '/usr/local/lib64/pkgconfig'
make[2]: Leaving directory '/usr/local/src/snort-2.9.9.0'
make[1]: Leaving directory '/usr/local/src/snort-2.9.9.0'
pc16:/usr/local/src/snort-2.9.9.0 #
```

Run “ldconfig –v” to create the necessary links and cache:
ldconfig -v /usr/local/lib64

```
        libzvbi-chains.so.0 -> libzvbi-chains.so.0.0.0
        libplds4.so -> libplds4.so
        libplc4.so -> libplc4.so
        libnspr4.so -> libnspr4.so
pc16:/usr/local/src/snort-2.9.9.0 #
```

Copy configuration files in **/usr/local/src/snort-2.9.9.0/etc** to **/etc/snort** directory:

cd /etc

mkdir snort

cd snort

cp /usr/local/src/snort-2.9.9.0/etc/* .   (there is a final dot in this command)

```
pc16:/etc/snort # ls -l
total 296
-rw-r--r--  1 root root   1281 May 14  19:41 attribute_table.dtd
-rw-r--r--  1 root root   3757 May 14  19:41
classification.config
-rw-r--r--  1 root root  23058 May 14  19:41 file_magic.conf
-rw-r--r--  1 root root  31971 May 14  19:41 gen-msg.map
-rw-r--r--  1 root root  13471 May 14  19:41 Makefile
-rw-r--r--  1 root root    190 May 14  19:41 Makefile.am
-rw-r--r--  1 root root  12306 May 14  19:41 Makefile.in
-rw-r--r--  1 root root    687 May 14  19:41 reference.config
-rw-r--r--  1 root root  26804 May 14  19:41 snort.conf
-rw-r--r--  1 root root   2335 May 14  19:41 threshold.conf
-rw-r--r--  1 root root 160606 May 14  19:41 unicode.map
```

Rules installation

To download Snort rules, you need an oinkcode. Once you register on the Snort website, you can find your oinkcode in your user account settings page.

wget -c https://www.snort.org/rules/snortrules-snapshot-2990.tar.gz?oinkcode=<oinkcode>

Untar the rules into **/etc/snort/** directory:
cd /etc/snort
tar -xzf /home/<user>/Downloads/snortrules-snapshot-2990.tar.gz

```
touch /etc/snort/rules/whitelist.rules /etc/snort/rules/blacklist.rules
touch /etc/snort/rules/snort.rules                    (a blank rules file for initial testings)
touch /etc/snort/rules/local.rules
```

Note: you may notice that some files do not contain rules and the rules are disabled in other files.

Create a Snort user account:

```
mkdir /var/log/snort
useradd snort -d /var/log/snort -s /bin/false -c SNORT_IDS
groupadd snort
```

Edit the Snort configuration file **/etc/snort/snort.conf**:

```
ipvar HOME_NET 192.168.99.0/24               (line 45)
                        (this is your internal network to be monitored)
ipvar EXTERNAL_NET !$HOME_NET
                        (your external network, from which attacks may initiate)
var RULE_PATH /etc/snort/rules               (line 104) (path to the Snort rules)
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

# path to dynamic preprocessor libraries              (line 246)
dynamicpreprocessor directory /usr/local/lib64/snort_dynamicpreprocessor/

# path to base preprocessor engine
dynamicengine /usr/local/lib64/snort_dynamicengine/libsf_engine.so

# path to dynamic rules libraries
dynamicdetection directory /usr/local/lib64/snort_dynamicrules
```

Now, jump to Reputation Preprocessor section:

```
whitelist $WHITE_LIST_PATH/iplists/white_list.rules, \     (line 511)
blacklist $BLACK_LIST_PATH/iplists/black_list.rules
```

Reputation Preprocessor section must look like:

```
# Reputation preprocessor. For more information see README.reputation
preprocessor reputation: \
   memcap 500, \
   priority whitelist, \
   nested_ip inner, \
   whitelist $WHITE_LIST_PATH/iplists/white_list.rules, \
   blacklist $BLACK_LIST_PATH/iplists/black_list.rules
```

Save the changes.

Now create an **iplists** directory to allow or deny IPs:

```
mkdir /etc/snort/rules/iplists
touch /etc/snort/rules/iplists/white_list.rules
touch /etc/snort/rules/iplists/black_list.rules
```

Use “sed” to comment out all the lines that have the text “include $RULE_PATH” in **/etc/snort/snort.conf** file.

```
cd /etc/snort                (if you are not here, already)
cp snort.conf snort.conf-orig
sed 's/include $RULE_PATH/#include $RULE_PATH/g' snort.conf-orig > snort.conf
```

Then add a line for **snort.rules** and re-enable the **local.rules** line in **/etc/snort/snort.conf** file:

```
include $RULE_PATH/snort.rules
include $RULE_PATH/local.rules
```

Save the changes.

snort.conf, must look like:

```
# site specific rules
include  $RULE_PATH/snort.rules
include  $RULE_PATH/local.rules
#include  $RULE_PATH/app-detect.rules
#include  $RULE_PATH/attack-responses.rules
 … etc.
```

Startup Script

Copy and paste the following script and save it as **/etc/init.d/snortd**. It is the script used to start, stop, restart and show the status of the Snort service).

```
----- 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
### 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=/bin:/usr/bin:/sbin:/usr/sbin
SNORT_BIN=/usr/local/bin/snort
SNORT_SOCKET=/var/log/snort/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 [ "$INTERFACE"X = "X" ]; then
  INTERFACE="-i eth0"
  HW_INTF="eth0"
else
  HW_INTF=$INTERFACE
  INTERFACE="-i $INTERFACE"
  SNORT_SOCKET=/var/run/snort_$HW_INTF.pid
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=10
fi

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

# Promiscuos mode
if [ $PROMISC = "YES" ]; then
    ip link set eth0 promisc on
else
    ip link set eth0 promisc off
fi

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

# Wait for the NIC to be up and ready, to avoid messages like:
# Can't start DAQ (-1) - eth0: That device is not up!
sleep $SECS

# 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
        # 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 -l $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/log/snort_$HW_INTF.* &&
        rm -f /var/log/snort/snort_$HW_INTF.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
    *)
        echo "Usage: $0 {start|stop|status|restart|reload}"
        exit 1      ;;
esac
rc_exit

----- CUT HERE -----
```

Now check the name of the network interface card (NIC) of your guest machine:

```
ifconfig
```

```
eth0   Link encap:Ethernet  HWaddr 52:54:00:xx:xx:xx
```

```
          inet addr:192.168.99.10  Bcast:192.168.99.255  Mask:255.255.255.0
          inet6 addr: 2001:1368:edf3:d2:5054:ff:ffff:ffff/64 Scope:Global
          inet6 addr: fe80::5054:ff:ffff:ffff/64 Scope:Link
```

In this example, eth0. Edit the snortd script and modify it if necessary.

Strengthen file permissions of the script:

```
chown snort:snort /etc/init.d/snortd
chmod 700 /etc/init.d/snortd
```

Execution parameters

Copy and paste the following script and save it as **/etc/sysconfig/snort**.

```
----- 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
PROMISC=NO

--- CUT HERE ---
```

Edit the snort script and modify the INTERFACE variable if necessary.

Strengthen file and directory permissions:

```
chown snort:snort /etc/sysconfig/snort
chmod 700 /etc/sysconfig/snort

cd /var/log
chown snort:snort snort
chmod 700 snort

cd /usr/local/lib64
mkdir snort_dynamicrules
chown -R snort:snort snort*
chown -R snort:snort pkgconfig
chmod -R 700 snort*
chmod -R 700 pkgconfig

cd /usr/local/bin
chown snort:snort daq-modules-config
chown snort:snort u2*
chmod 700 daq-modules-config
chmod 700 u2*

cd /etc
chown -R snort:snort snort
chmod -R 700 snort
```

What we have:

- Executable: /usr/local/bin/snort
- Startup Script: /etc/init.d/snortd
- Configuration parameters for Snort startup: /etc/sysconfig/snort
- Directory of rules and configuration files: /etc/snort
- Directory of dynamic libraries: /usr/local/lib64
- Directory of logs: /var/log/snort

Snort Test

Congratulations! Snort is installed and configured in your guest machine at this step. To test Snort, enter the commands:

```
date            (to know the exact time before the test)
/usr/local/bin/snort -T -i eth0 -u snort -g snort -c /etc/snort/snort.conf
```

If the test is successful, the following message will appear:

```
 Snort successfully validated the configuration!
Snort exiting
```

```
pc16:/etc #
```

Otherwise, check the system log:

```
journalctl --since hh:mm:ss
```

where **hh** is for hour, **mm** for minutes and **ss** for seconds, to filter the output of journalctl by specifying the starting log time.

You can now further check if Snort is working well by adding an ICMP rule (up to now, local.rules and snort.rules are the only active rules files, but both are blank).

Add the following line to **/etc/snort/rules/snort.rules**:

```
alert icmp any any -> any any (msg: "ICMP Packet found"; sid:2000001; rev:1;)
```

Now start Snort manually:

```
/usr/local/bin/snort -i eth0 -D -u snort -g snort -c /etc/snort/snort.conf
```

The system must show something similar to:

```
Spawning daemon child...
My daemon child 2185 lives...
Daemon parent exiting (0)
```

You can use “ps” anytime to verify if snort is running:

```
ps aux | grep snort
```

to get something like this:

```
snort    20872  0.0  2.7 453536 83652 ?       Ssl  11:04   0:00 /usr/local/bin/snort -i eth0 -D -u snort
-g snort -c /etc/snort/snort.conf
```

Then, from your HOST send ping packets to your GUEST 192.168.99.10:

```
ping 192.168.99.10
```

Check the log directory:

```
cd /var/log/snort
```

This directory should contain something like:

```
-rw-r--r-- 1 root  root  20560 Aug  8 10:20 alert
-rw------- 1 snort snort     5 Aug  8 10:16 snort_eth0.pid
-rw------- 1 snort snort     0 Aug  8 10:16 snort_eth0.pid.lck
-rw------- 1 snort snort 12584 Aug  8 10:20 snort.log.1470669370
```

To show the content of the alert file, do the following:

```
tail -20 alert
```

```
[**] [1:2000001:1] ICMP Packet found [**]
[Priority: 0]
05/05-11:13:16.055116 192.168.99.99 -> 192.168.99.10
ICMP TTL:64 TOS:0x0 ID:55760 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:9529   Seq:4  ECHO

[**] [1:2000001:1] ICMP Packet found [**]
[Priority: 0]
05/05-11:13:16.055155 192.168.99.10 -> 192.168.99.99
ICMP TTL:64 TOS:0x0 ID:24612 IpLen:20 DgmLen:84
Type:0  Code:0  ID:9529  Seq:4  ECHO REPLY
```

In this example, 192.168.99.99 is the HOST IP. You can check journalctl for more information in case the alert file is empty.

To stop Snort (manually executed), you must kill the process.

```
ps -ef |grep snort
kill <snort pid>
```

Snort as a Service

To start Snort at boot time, enable it as a service.

```
chkconfig -a snortd
```

```
Note: This output shows SysV services only and does not include native
systemd services. SysV configuration data might be overridden by native
systemd configuration.

If you want to list systemd services use 'systemctl list-unit-files'.
To see services enabled on particular target use
'systemctl list-dependencies [target]'.

snortd                  0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

Another way to enable the service is:

```
chkconfig snortd on
```

Note: as long as you modify the snortd script, you must reload systemd manager configuration:

```
systemctl daemon-reload
```

To manually start the service:

systemctl start snortd.service

You can verify Snort status:
systemctl status snortd.service

```
snortd.service - LSB: Start snort IDS
   Loaded: loaded (/etc/init.d/snortd; bad; vendor preset: disabled)
   Active: active (running) since Thu 2017-06-15 10:26:05 EST; 20h ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 2 (limit: 512)
   CGroup: /system.slice/snortd.service
           └─1601 /usr/local/bin/snort -A fast -b -d -D -i eth0 -u snort -g snort -c
```

(a note about "bad": systemctl shows Systemd Unit files status, and snortd.service is not a native service.)

Command syntax:
systemctl [start | stop | status | restart] [snort | snortd.service]

Sniff the network

To sniff the network, change the PROMISC variable to YES in **/etc/sysconfig/snort** file:
PROMISC=YES
and restart snortd service.

To verify that the NIC is in promiscuous mode, do the following:
netstat -i

```
Kernel Interface table
Iface    MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0    1500   0    11885      0    108      0      974      0      0      0 BMPRU
lo     65536   0       10      0      0      0       10      0      0      0 LRU
```

The "P" flag indicates promiscuous mode.

To verify the version of Snort, use the following command:
snort -V

```
pc16:/etc/snort # snort -V
   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.9.0 GRE (Build 56) x86_64
   ''''    By Martin Roesch & The Snort Team:
http://www.snort.org/contact#team
           Copyright (C) 2014-2016 Cisco and/or its affiliates. All rights
reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.8.1
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.8
```

For help, do the following:

snort --h

Finally, remember to activate other rules.

- - - This is the end - - -