Snort 2.9.14.1 on Centos 7

Milad Rezaei
R&D, Shatel group of companies
September 4, 2019

Install

In this tutorial, we show how to install and configure snort 2.9.14.1 & Barnyard2 & Base & Pulledpork on CentOS 7.6.

First, update the OS:

```
yum update -y

yum install epel-release -y
```

(We can install snort from source or install it using precompiled package exists in snort.org)

Snort provides rpm package for CentOS 7, which can be install simply with the below command:

```
yum install https://www.snort.org/downloads/snort/snort-2.9.14.1-1.centos7.x86_64.rpm
```

Installing from the source

Install necessary packages:

```
yum install gcc gcc-c++ libnetfilter_queue libnetfilter_queue-devel git flex bison zlib zlib-devel pcre pcre-devel libdnet libdnet-devel tcpdump libnghttp2 wget xz-devel lzma mysql-devel* -y
```

We will download and store source files in the following folder:

```
mkdir ~/snort_src

cd ~/snort_src
```

Snort requires Libpcap and DAQ and we need to install them before installing snort:

Install libpcap

```
wget http://www.tcpdump.org/release/libpcap-1.8.1.tar.gz
tar xzvf libpcap-1.8.1.tar.gz
cd libpcap-1.8.1
./configure && make && make install
yum install libpcap-devel -y
cd ..
```

Install DAQ

```
wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
tar xvfz daq-2.0.6.tar.gz
cd daq-2.0.6
./configure && make && make install
cd ..
```

We need to install LuaJIT package:

```
wget http://luajit.org/download/LuaJIT-2.0.5.tar.gz
tar xvf LuaJIT-2.0.5.tar.gz
 cd LuaJIT-2.0.5/
make && make install
```

Install Snort:

```
wget https://www.snort.org/downloads/snort/snort-2.9.14.1.tar.gz
tar -xvzf snort-2.9.14.1.tar.gz
cd snort-2.9.14.1
./configure --enable-sourcefire && make && make install
```

Configuration

Now we need to edit some configuration files, download the rules from snort.org and take snort for a test run.

First, we will update shared library:

```
ldconfig
```

Snort on CentOS is installed in /usr/local/bin/snort directory, it is a good practice to create a symbolic link to /usr/sbin/snort.

(If you installed Snort with 'yum' you can skip this command.)

```
ln -s /usr/local/bin/snort /usr/sbin/snort
```

To verify the installation of snort use the command below:

```
snort -v
```

If you get error while loading shared libdnet.1 libraries, create the following link and try again.

```
ln -s /usr/lib64/libdnet.so.1.0.1 /usr/lib64/libdnet.1
```

To run Snort on CentOS safely without root access, we should create a new unprivileged user and a new user group for the daemon.

(If you installed Snort with 'yum' you can skip this command.)

```
groupadd snort
useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort
```

Then create the folder structure to keep the Snort configuration, use commands below. If you installed Snort using yum these directories should have already been added at install, check to make sure.

```
mkdir -p /etc/snort/rules
mkdir /var/log/snort
mkdir /usr/local/lib/snort_dynamicrules
```

Set the permissions for the new directories:

```
chmod -R 5775 /etc/snort
chmod -R 5775 /var/log/snort
chmod -R 5775 /usr/local/lib/snort_dynamicrules
chmod -R 5775 /usr/local/lib/snort_dynamicrules
chown -R snort:snort /var/log/snort
chown -R snort:snort /usr/local/lib/snort_dynamicrules
```

Create new files:

```
touch /etc/snort/rules/white_list.rules
touch /etc/snort/rules/black_list.rules
touch /etc/snort/rules/local.rules
```

copy gen-msg.map to snort folder:

```
cd ~/snort_src/snort-2.9.14.1/etc/
cp  ~/snort_src/snort-2.9.14.1/etc/ gen-msg.map   /etc/snort/
```

Install Pulledpork

Pulled_Pork is tool written in perl for managing Snort rule sets. Pulled_Pork features include:

Automatic rule downloads using your Oinkcode

MD5 verification prior to downloading new rulesets

Full handling of Shared Object (SO) rules

Generation of so_rule stub files

Modification of ruleset state (disabling rules, etc)

The project is run by JJ Cummings

Install necessary packages:

```
yum install perl-libwww-perl perl-core "perl(Crypt::SSLeay)" perl-LWP-Protocol-https
```

Download Pulledpork from Git and install:

```
git clone https://github.com/shirkdog/pulledpork.git
cd pulledpork/
cp pulledpork.pl /usr/local/bin
chmod +x /usr/local/bin/pulledpork.pl
cp etc/*.conf /etc/snort
mkdir /etc/snort/rules/iplists
touch /etc/snort/rules/iplists/default.blacklist
```

To verify the installation of Pulledpork use the command below:

```
pulledpork.pl -V
```

Run these commands to change rules path on snort.conf and make some files:

```
echo "include \$RULE_PATH/so_rules.rules" >> /etc/snort/snort.conf
echo "include \$RULE_PATH/snort.rules" >> /etc/snort/snort.conf
touch /etc/snort/rules/so_rules.rules
touch /etc/snort/rules/snort.rules
```

Then make change to Pulledpork config file like below: replace your oinkcode

```
vi /etc/snort/pulledpork.conf
rule_url=https://www.snort.org/reg-rules/|snortrules-snapshot.tar.gz| <oinkcode>
line 21 comment it
line 74 change to:rule_path=/etc/snort/rules/snort.rules
line 89 change to:local_rules=/etc/snort/rules/local.rules
line 92 change to:sid_msg=/etc/snort/sid-msg.map
line 119 change to:config_path=/etc/snort/snort.conf
line 136 change to:distro=Centos-7
line 144 change to:ack_list=/etc/snort/rules/iplists/default.blacklist
line 153 change to:IPRVersion=/etc/snort/rules/iplists
line 202 uncomment and change to:enablesid=/etc/snort/enablesid.conf
line 203 uncomment and change to:dropsid=/etc/snort/dropsid.conf
line 204 uncomment and change to:disablesid=/etc/snort/disablesid.conf
line 205 uncomment and change to:modifysid=/etc/snort/modifysid.conf
```

Save and run these commands:

```
mkdir -p /usr/local/etc/snort/rules/iplists/
touch /usr/local/etc/snort/rules/iplists/default
```

Runing Pulledpork:

```
pulledpork.pl -c /etc/snort/pulledpork.conf
```

Then to make Pulledpork run every day to check new signature and rules:

```
crontab –e
0 0 * * * root /usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf
```

Configuring the network and rule sets

Edit snort.conf file to modify a few parameters:

```
vi /etc/snort/snort.conf
```

Change parameter as the example below:

```
ipvar HOME_NET 192.168.1.0/24
ipvar EXTERNAL_NET !$HOME_NET
var RULE_PATH /etc/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
```

In this tutorial, we use Barnyard2 to copy log from snort log folder to the databases. For that reason, we need to setup snort output log to unified as below:

```
output unified2: filename snort.log, limit 1024
```

Finally test snort configuration file by the following command:

```
snort -T -c /etc/snort/snort.conf
```

If you get success message everything is correct.

To test Snort we add rules to local. Rules:

```
vi /etc/snort/rules/local.rules

alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:001;)
```

These rules make alerts for all icmp messages being sent to $HOME_NET (snort alert structure could be found in snort.org)

Running snort as a Daemon

To run snort on CentOS as a service in the background you should copy following script to /etc/init.d/

vi /etc/init.d/snortd

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

if [ "$INTERFACE"X = "X" ]; then
  INTERFACE="-i ens33"
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 $NO_PACKET_LOG $DUMP_APP -D $PRINT_INTERFACE -i
$i -u $USER -g $GROUP $CONF -l $LOGDIR/$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 $NO_PACKET_LOG $DUMP_APP -D $PRINT_INTERFACE -i
$i -u $USER -g $GROUP $CONF -l $LOGDIR/$i $PASS_FIRST $BPFFILE $BPF
          done
        else
          # Run with a single interface (default)
          daemon /usr/sbin/snort $ALERTMODE $BINARY_LOG $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
```

Then:

```
vi /etc/sysconfig/snort
```

And copy this script to that file (replace interface name) :

```
# /etc/sysconfig/snort

# $Id$

# All of these options with the exception of -c, which tells Snort where

# the configuration file is, may be specified in that configuration file as

# well as the command line. Both the command line and config file options

# are listed here for reference.

#### General Configuration


# What interface should snort listen on?  [Pick only 1 of the next 3!]

# This is -i {interface} on the command line

# This is the snort.conf config interface: {interface} directive

INTERFACE=ens33

#

# The following two options are not directly supported on the command line

# or in the conf file and assume the same Snort configuration for all

# instances

#

# To listen on all interfaces use this:

#INTERFACE=ALL

#

# To listen only on given interfaces use this:

#INTERFACE="eth1 eth2 eth3 eth4 eth5"

# Where is Snort's configuration file?

# -c {/path/to/snort.conf}

CONF=/etc/snort/snort.conf
```

```
# What user and group should Snort drop to after starting? This user and
# group should have very few privileges.
# -u {user} -g {group}
# config set_uid: user
# config set_gid: group
USER=snort
GROUP=snort


# Should Snort change the order in which the rules are applied to packets.
# Instead of being applied in the standard Alert->Pass->Log order, this will
# apply them in Pass->Alert->Log order.
# -o
# config order: {actions in order}
# e.g. config order: log alert pass activation dynamic suspicious redalert
PASS_FIRST=0



#### Logging & Alerting


# NOTE: NO_PACKET_LOG and BINARY_LOG, ALERTMODE, etc. are mutually
# exclusive. Use either NO_PACKET_LOG or any/all of the other logging
# options. But the more logging options use you, the slower Snort will run.
# Where should Snort log?
# -l {/path/to/logdir}
# config logdir: {/path/to/logdir}
LOGDIR=/var/log/snort
# How should Snort alert? Valid alert modes include fast, full, none, and
# unsock.  Fast writes alerts to the default "alert" file in a single-line,
# syslog style alert message.  Full writes the alert to the "alert" file
# with the full decoded header as well as the alert message.  None turns off
# alerting. Unsock is an experimental mode that sends the alert information
# out over a UNIX socket to another process that attaches to that socket.
# -A {alert-mode}
# output alert_{type}: {options}
#ALERTMODE=full
```

```
# Should Snort dump the application layer data when displaying packets in
# verbose or packet logging mode.
# -d
# config dump_payload
#DUMP_APP=1

# Should Snort keep binary (AKA pcap, AKA tcpdump) logs also? This is
# recommended as it provides very useful information for investigations.
# -b
# output log_tcpdump: {log name}
#BINARY_LOG=0

# Should Snort turn off packet logging?  The program still generates
# alerts normally.
# -N
# config nolog
NO_PACKET_LOG=0

# Print out the receiving interface name in alerts.
# -I
# config alert_with_interface_name
PRINT_INTERFACE=0
# When dumping the stats, what log file should we look in
SYSLOG=/var/log/messages

# When dumping the stats, how long to wait to make sure that syslog can
# flush data to disk
SECS=5
# To add a BPF filter to the command line uncomment the following variable
# syntax corresponds to tcpdump(8)
#BPF="not host 192.168.1.1"
# To use an external BPF filter file uncomment the following variable
# syntax corresponds to tcpdump(8)
# -F {/path/to/bpf_file}
# config bpf_file: /path/to/bpf_file
#BPFFILE=/etc/snort/bpf_file
```

If you install Snort using yum, you should already have the startup script configured. Start the service as described below.

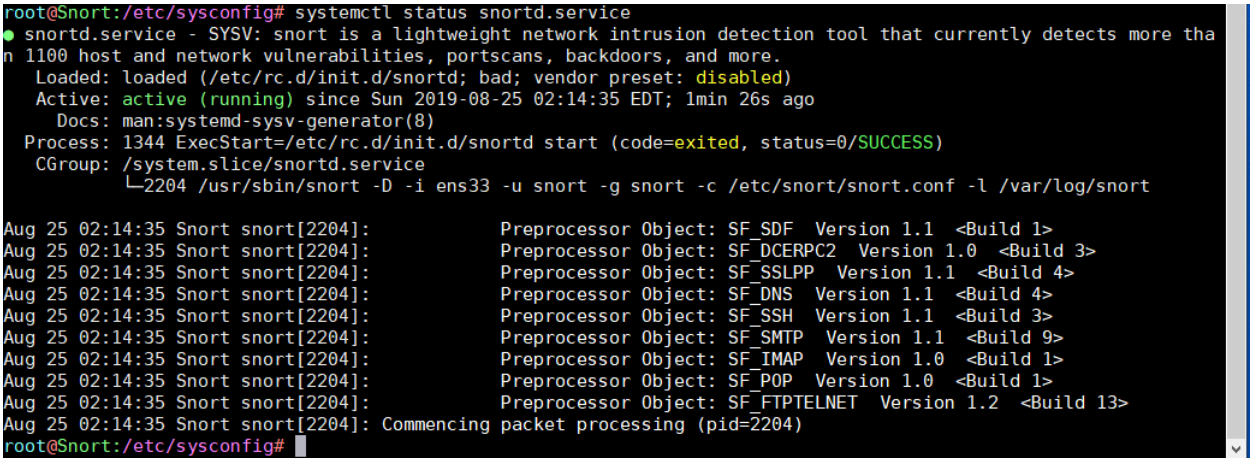
```
chmod 755  /etc/init.d/snortd

systemctl daemon-reload

systemctl start snortd

systemctl enable snortd
```

If we use systemctl status snortd.service we should see output like below:



In /etc/sysconfig/snort file we can determine how snort starts and sniffs on which interface or determine how to save output logs. If we store logs to “barnyard2”, we need to comment some variables in this file like:

```
vi /etc/sysconfig/snort
```

Comment this variable:

```
BINARY_LOG=0

DUMP_APP=1

ALERTMODE=full
```

Save and exit. Now, snort has been installed and ready to use in Nids mode.

Barnyard2

Barnyard2 provides the following enhancements to the original snort:

Parsing of the new unified2 log files.

Maintaining majority of the command syntaxes.

Addressing all associated bug reports and feature requests arising since barnyard-0.2.0.

Completely rewritten code based on the GPLv2 Snort making it entirely GPLv2.

SnortSam functionality

To install and configure Barnyard2:

```
yum install git unzip  libtool mariadb-server -y

cd  ~/snort_src

git clone https://github.com/firnsy/barnyard2.git

cd barnyard2-master/

./autogen.sh && ./configure --with-mysql --with-mysql-libraries=/usr/lib64/mysql &&
make && make install

cp -v etc/barnyard2.conf /etc/snort/
```

Make some changes to barnyard2 config file:

```
vi /etc/snort/barnyard2.conf
```

Change variable:

```
config logdir: /var/log/barnyard2

config hostname:   localhost

config interface:  ens33

config waldo_file: /var/log/barnyard2/barnyard2.waldo
```

Make Barnyard2 log folder:

```
mkdir /var/log/barnyard2

chmod 744 /var/log/barnyard2 && chown snort.snort /var/log/barnyard2

touch /var/log/barnyard2/barnyard2.waldo && chown snort.snort
/var/log/barnyard2/barnyard2.waldo
```

to test Barnyard we use the command below: (It runs Barnyard in Batch mode)

```
cd /var/log/snort

barnyard2 -c /etc/snort/barnyard2.conf -o snort.log
```

IF every thing is ok, Barnyard2.waldo file size increases.

To run Barnyard2 in continuous mode use this command:

```
cd /var/log/snort && barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.log
-w /var/log/barnyard2/barnyard2.waldo
```

To prepare Mysql for using with Barnyard2 and Base:

```
systemctl start mariadb && systemctl enable mariadb

mysql_secure_installation
```

Set password for ‘root’ and answer to all questions with “Y”

Then connect to mysql and create database and make snort user: (choose a strong password)

```
mysql -u root -p
create database snort;
grant all privileges on snort.* to snort@'localhost' identified by '123456';
grant all privileges on snort.* to snort@'127.0.0.1' identified by '123456';
flush privileges;
exit
```

Then use Barnayrd2 schemas:

```
cd  ~/snort_src/barnyard2-master/schemas/
mysql -u root -p snort < create_mysql
```

config Barnyard2 to connect mysql:

```
vi /etc/snort/barnyard2.conf
Line 227 comment ---> #output alert_fast: stdout
Line 351 output database: log, mysql, user=snort password=123456 dbname=snort
host=localhost
```

Now everything is ok and we can use them.

We can see data in mysql by using this command:

```
mysql -u root -p
use snort
show tables;
select * from data;
```

To use Barnyard2 as service in CentOS 7:

```
cd  /etc/systemd/system
vi barnyard2.service
```

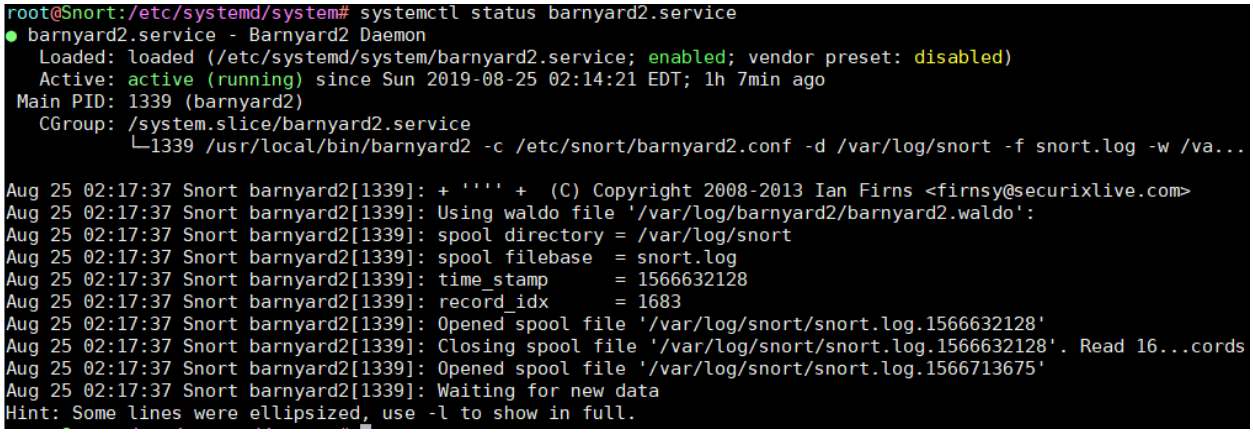
Copy and paste the command below:

```
[Unit]
        Description=Barnyard2 Daemon
        After=syslog.target network.target

[Service]
        Type=simple
        User=snort
        Group=snort
        ExecStart=/usr/local/bin/barnyard2 -c /etc/snort/barnyard2.conf -d
/var/log/snort -f snort.log -w /var/log/barnyard2/barnyard2.waldo

[Install]
    WantedBy=multi-user.target
```

```
chmod 644 barnyard2.service
systemctl start barnyard2
systemctl enable barnyard2
systemctl status barnyard2
```



Finally, Baryard is ready to use.

Install Base

BASE is the Basic Analysis and Security Engine. It is based on the code from the Analysis Console for Intrusion Databases (ACID) project. This application provides a web front-end to query and analyze the alerts coming from a SNORT IDS system.

To install Base, first install required packages:

```
yum install httpd php php-pear php-gd php-mysql -y
systemctl start httpd && systemctl enable httpd
```

You should add http service in your firewall; You can disable your firewall (Not recommended).

Download adodb (Database Abstraction Layer for PHP)

https://sourceforge.net/projects/adodb/files/latest/download

Download adodb-5.20.14.zip and copy to /var/www

```
cp adodb-5.20.14.zip  /var/www/
cd /var/www
unzip adodb-5.20.14.zip
mv adodb5 adodb
chown apache:apache adodb
```

Download Base

https://sourceforge.net/projects/secureideas/files/latest/download

Copy base to /var/www/html

```
cd /var/www/html &&  tar xzvf base-1.4.5.tar.gz
mv base-1.4.5 base && chown apache:apache base
cd base && mv base_conf.php.dist base_conf.php
```

Make the following changes to base_conf.php to connect to mysql:

```
vi base_conf.php
$BASE_urlpath = '/base';
$DBlib_path = '/var/www/adodb';
$alert_dbname   = 'snort';
$alert_host     = 'localhost';
$alert_port     = '';
$alert_user     = 'snort';
$alert_password = '123456';
```

To set timezone for php :

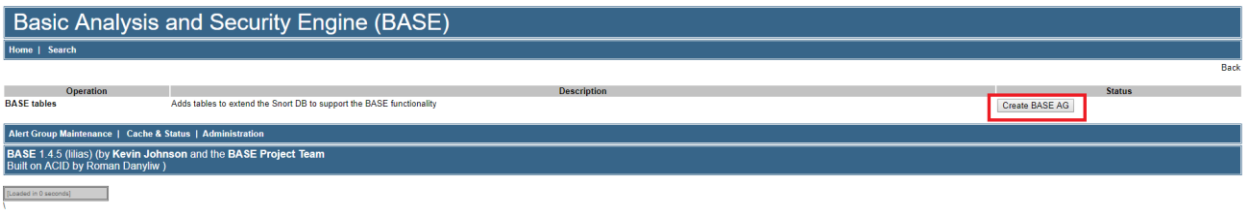
```
vi /etc/php.ini
line 878 date.timezone = America/Toronto
```

You can connect to Base by this URL:

http://yourmachineipaddress/base

After that you should see page like this, then click on Setup page:

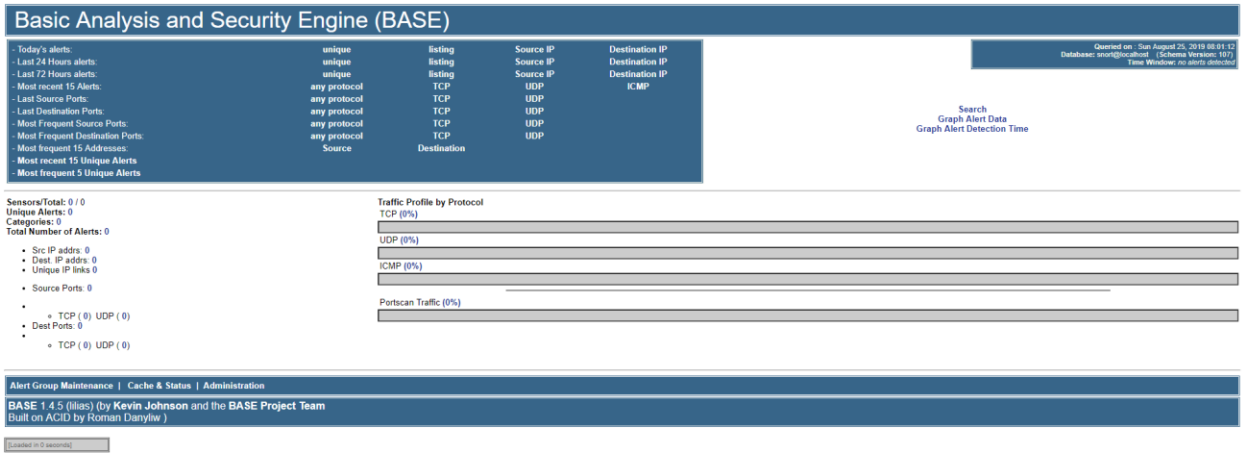
Then click on Create BASE AG



After that you should see output like this:

Then click on Main page:

The base is ready:



You can see all Snort log Alert in base.