Snort 2.9.7.x on Ubuntu 12 and 14

with Barnyard2, PulledPork, and BASE

Noah Dietrich
`Noah@SublimeRobots.com`

January 14, 2015

Contents

1 Introduction

This guide will walk you through installing Snort as a NIDS (network intrusion detection system), with three pieces of additional software to improve the functionality of Snort. This guide is written with the Snort host as a VMware ESXi virtual machine, but can be easily used to install Snort on a physical machine or as a virtual machine on another platform.

The latest version of this guide plus additional notes can be found at SublimeRobots.com.

This installer guide has been tested on the following versions of Ubuntu running on VMware ESXi 5:

- Ubuntu 12.04.5 LTS x86
- Ubuntu 12.04.5 LTS x64
- Ubuntu 14.10 Server LTS x86
- Ubuntu 14.10 Server LTS x64

While you can choose to install Snort without any supporting software and it will work just fine, it becomes much more useful with a few additional software packages. These packages are:

Barnyard2:
Software that takes Snort output and writes to a SQL database, which reduces load on the system.

PulledPork:
Automatically downloads the latest Snort definitions files.

BASE:
A web-based graphical interface for viewing and clearing Snort events.

If you just want to setup Snort on a Ubuntu system without going through the work in this document, there is a project called Autosnort (`https://github.com/da667/Autosnort`) that will install all the same software as this guide with a script. Additinally, you could use a fully configured LiveCD like EasyIDS (`http://www.skynet-solutions.net/About-EasyIDS`). The benefit of this guide over *Autosnort* or *EasyIDS* is that this guide walks you through installing each component, explaining the steps as you go along. This will give you a better understanding of the software components that make up Snort, and will allow you to configure Snort for your own needs.

Note: while this guide focuses on current software, if one wanted this guide will most likely work to install Snort 2.9.6.x, and could be used to install Snort on Ubuntu 13, if desired.

2 About This Guide

Passwords: This guide chooses to use simplistic passwords to make it obvious as to what is being done. You should select your own passwords in place of these passwords.

Software Package Versions: This guide is written to install with the latest version of all software available, except where noted for compatibility reasons. This guide should work with slightly newer or older versions of all software packages, but ensuring compatibility is up to the individual user. If you have issues with a newer package, please first try with the version this guide uses to determine if the difficulty is with the specific version or another issue. Additionally, this guide tries to use software from official Ubuntu repositories as much as possible, downloading specific tarballs from trusted 3rd party sites (such as `snort.org` only when no package is available from official repositories.

Software versions used in this guide:

- Snort 2.9.7.0
- Barnyard2 2-1.13
- PulledPork 0.7.0
- BASE 1.4.5

Administrator Accounts: This guide assumes that you are logged into the system as a normal user, and will run all administrative commands with `sudo`. This helps to identify what commands require administrative credentials, and which do not. We will also create a non-administrative user named `snort` to run our software under, following current best practices.

3 Enviornment

As stated above, this guide was written with an eye towards installing Snort on virtual machine running on an VMware ESXi 5 hypervisor. The ESXi hypervisor is a free product from vMware: http://www.vmware.com/products/ESXi-hypervisor, and which I highly recommend for testing software due to the ability to create snapshots. If you choose to install Snort outside of a virtual machine, the steps below should be the same, except for a few VMware specific steps that should be fairly obvious.

4 VMware Virtual Machine Configuration

If you are using VMware ESXi to host your Snort virtual machine, when creating the virtual machine, make sure to select the **VMXNET 3** network adapter (not the default adapter) when creating the client virtual machine, as it works better for Snort[1] [2].

This guide assumes that you have created a virtual machine with a single network adapter that will be used for both administrative control (over SSH) as well as for Snort to listen on for traffic. You can easily add more adapters when setting up the system or at a later date, you just need to make sure to specify the correct adapter Snort should listen on at runtime (this should be fairly obvious).

5 Installing Ubuntu

This guide will assume that you have installed one of the supported versions of Ubuntu with all the default settings, and that you have selected "install security updates automatically" during the configuration.

We need an IP address assigned to eth0 (the adapter that Snort will listen to traffic on). by default Ubuntu will use DHCP to auto-configure an address, if this is the case, you can verify your ip address by running `ifconfig eth0`. if you do not have an IP address, configure one manually. You will need internet connectivity in order to download packages and software tarballs.

Once you have logged in for the first time and verified internet connectivity, make sure the system is up to date, and install openssh-server (so we can remotely-manage the system). Reboot after installation to make sure all patches are applied.

[1]https://isc.sans.edu/diary/Running+Snort+on+VMWare+ESXi/15899
[2]http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1001805

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install -y openssh-server
sudo reboot
```

6 Network Card Configuration

From http://manual.snort.org/node7.html:

> Some network cards have features named “Large Receive Offload” (lro) and “Generic Receive Offload” (gro). With these features enabled, the network card performs packet reassembly before they’re processed by the kernel. By default, Snort will truncate packets larger than the default snaplen of 1518 bytes. In addition, LRO and GRO may cause issues with Stream5 target-based reassembly. We recommend that you turn off LRO and GRO.

Disable LRO and GRO:

```
sudo apt-get install -y ethtool
sudo ethtool -K eth0 gro off
sudo ethtool -K eth0 lro off
```

7 Installing the Snort Pre-Requisites

Snort has four main pre-requisites:

pcap	(libpcap-dev)	available from the Ubuntu repository
PCRE	(libpcre3-dev)	available from the Ubuntu repository
Libdnet	(libdumbnet-dev)	available from the Ubuntu repository
DAQ	(http://www.snort.org/downloads/)	compiled from source

First we want to install all the tools required for building software. The `build-essentials` package does this for us:

```
sudo apt-get install -y build-essential
```

Once our build tools are installed, we install all Snort pre-requisites that are available from the Ubuntu repositories[3]:

```
sudo apt-get install -y libpcap-dev libpcre3-dev libdumbnet-dev
```

In this guide, we will be downloading a number of tarbals for various software packages. We will create a folder called `snort_src` to keep them all in one place:

```
mkdir ~/snort_src
cd ~/snort_src
```

The Snort DAQ (Data AcQuisition library)has a few pre-requisites that need to be installed:

[3]Many guides that install Snort on Ubuntu have you download libdnet from its homepage http://http://libdnet.sourceforge.net/. This is possible and will work fine. However, the `libdumbnet-dev` Ubuntu package provides the same software (do *not* install the libdnet package from Ubuntu archives, as it is an un-related package and does not provide the required libdent libraries). If you want to compile the libdent libraries from source and you are running a 64-bit version Ubuntu, use the `-fPIC` flag during the ’configure’ stage.

```
sudo apt-get install -y bison flex
```

Download and install the latest version of DAQ from the Snort website (please check the website to ensure you are getting the latest version). The steps below use wget to download version 2.0.4 of DAQ, which is the latest version at the time of writing this guide.

```
wget https://www.snort.org/downloads/snort/daq-2.0.4.tar.gz
tar -xvzf daq-2.0.4.tar.gz
cd daq-2.0.4
./configure
make
sudo make install
```

8 Installing Snort

To install Snort on Ubuntu, there is one other additional Snort pre-requisite that needs to be installed that is not mentioned in the documentation, `zlibg`, a compression library:

```
sudo apt-get install -y zlib1g-dev
```

We are now ready to download the Snort source tarball, compile, and then install. The `--enable-sourcefire` option gives Packet Performance Monitoring (PPM)[4][5], which lets us do performance monitoring for rules and pre-processors, and builds Snort the same way that the Snort team does:

```
cd ~/snort_src
wget https://www.snort.org/downloads/snort/snort-2.9.7.0.tar.gz
tar -xvzf snort-2.9.7.0.tar.gz
cd snort-2.9.7.0
./configure --enable-sourcefire
make
sudo make install
```

Run the following command to update shared libraries (you'll get an error when you try to run Snort if you skip this step):

```
sudo ldconfig
```

Place a symlink to the Snort binary in /usr/sbin:

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

Test Snort by running the binary as a regular user, passing it the `-V` flag (which tells Snort to verify itself and any configuration files passed to it). You should see output similar to what is shown below (although exact version numbers may be slightly different).

[4] `--enable-sourcefire`: http://blog.snort.org/2011/09/snort-291-installation-guide-for-centos.html
[5] PPM: http://manual.snort.org/node221.html

```
user@snortserver:~$ snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.6.2
           Using PCRE version: 8.35 2014-04-04
           Using ZLIB version: 1.2.8


user@snortserver:~$
```

9 Configuring Snort to Run in NIDS Mode

Since we don't want Snort to run as root, we need to setup an unprivileged account and group for the daemon to run under `(snort:snort)`. We will also create a number of files and directories required by Snort, and set permissions on those files. Snort will keep all configurations and rule files in `/etc/snort`, and all alerts will be written to `/var/log/snort`.

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

sudo mkdir /etc/snort
sudo mkdir /etc/snort/rules
sudo mkdir /etc/snort/preproc_rules
sudo touch /etc/snort/rules/white_list.rules /etc/snort/rules/black_list.rules /etc/snort/rules/local.rules

sudo mkdir /var/log/snort
sudo mkdir /usr/local/lib/snort_dynamicrules

sudo chmod -R 5775 /etc/snort
sudo chmod -R 5775 /var/log/snort
sudo chmod -R 5775 /usr/local/lib/snort_dynamicrules

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

Snort needs the following configuration files copied from the Snort tarball into the `/etc/snort` folder:

- classification.config
- reference.config
- snort.conf
- threshold.conf
- gen-msg.map
- unicode.map

To copy the configuration files, run the following commands:

```
sudo cp ~/snort_src/snort-2.9.7.0/etc/*.conf* /etc/snort
sudo cp ~/snort_src/snort-2.9.7.0/etc/*.map /etc/snort
```

We now have the following directory layout and file locations:

Snort binary file	`/usr/local/bin/snort`	(linked to /sbin/snort)
Snort configuration file	`/etc/snort/snort.conf)`	
Snort log data directory	`/var/log/snort`	
Snort rules directories	`/etc/snort/rules`	
	`/usr/local/lib/snort_dynamicrules`	

Our Snort directory listing looks like this:

```
user@snortserver:~$ tree /etc/snort
/etc/snort
|-- classification.config
|-- file_magic.conf
|-- gen-msg.map
|-- preproc_rules
|-- reference.config
|-- rules
|   |-- black_list.rules
|   |-- local.rules
|   |-- white_list.rules
|-- snort.conf
|-- threshold.conf
|-- unicode.map
```

We now need to edit Snort's main configuration file, `/etc/snort/snort.conf`. When we run Snort with this file as an argument, it tells Snort to run in NIDS mode.
Before we run Snort in NIDS mode, we need to make a few edits to the default configuration file. We need to comment out all of the individual rule files that are referenced in the Snort configuration file, since instead of downloading each file individually, we will use PulledPork to manage our rulesets, which combines all the rules into a single file. The following line will comment out all rulesets in our `snort.conf` file:

```
sudo sed -i 's/include \$RULE\_PATH/#include \$RULE\_PATH/' /etc/snort/snort.conf
```

We will now manually change some settings in the `snort.conf` file, using your favorite editor:

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

Change the following lines to meet your environment:

Line 45, `HOME_NET` should match your internal (friendly) network, and line 48, `EXTERNAL_NET` should be all other networks. In the below example our HOME_NET is 10.0.0.0 with a 24-bit subnet mask (255.255.255.0)[6]:

```
ipvar HOME_NET 10.0.0.0/24
ipvar EXTERNAL_NET !$HOME_NET
```

Set the following file paths, beginning at line 104:

```
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
```

[6] http://books.gigatux.nl/mirror/snortids/0596006616/snortids-CHP-5-SECT-1.html

In order to make testing snort easy, we want to enable the `local.rules` file, where we can add rules that Snort can alert on, which is good for testing. Un-comment (remove the hash symbol) from line 545:

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

Once the configuration file is ready, we will have Snort verify that it is a valid file, and all necessary files it references are correct. We use the `-T` flag to test the configuration file, and we use the `-c` flag to tell Snort which configuration file to use. Run the following command and look for the following output:

```
user@snortserver:~$ sudo snort -T -c /etc/snort/snort.conf
    (...)
    Snort successfully validated the configuration!
    Snort exiting
user@snortserver:~$
```

At this stage, Snort does not have any rules loaded (our rule files referenced in `snort.conf` are empty). You can verify that Snort has not loaded any rules if you scroll up through the output from the previous command. To test Snort, let's create a simple rule that will cause Snort to generate an alert whenever Snort sees an ICMP "Echo request" or "Echo reply" messages, which is easy to generate with the ubiquitus `ping` utility (this makes for easy testing of the rule).
Paste the following line into the empty local rules file: `/etc/snort/rules/local.rules`:

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

When you un-commented line 545 above (`include $RULE_PATH/local.rules`) you were telling Snort that the `local.rules` file should be loaded by Snort. When Snort loads that file on start-up, it will see the rule you created, and use that rule on all traffic the interface sees. In this case, when we created the rule, we told Snort that it should generate an alert when it sees an ICMP ping. Since we made changes to the Snort configuration, we should test the configuration file again:

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

This time if you scroll up through the output, you will find that one rule (the one we created in `local.rules`, and loaded by the `include` in `snort.conf`) has been loaded:

```
        +-------------------[Rule Port Counts]---------------------------------------
        |             tcp     udp    icmp      ip
        |     src       0       0       0       0
        |     dst       0       0       0       0
        |     any       0       0       1       0
        |      nc       0       0       1       0
        |     s+d       0       0       0       0
        +----------------------------------------------------------------------------
```

Now that we know that Snort correctly loads our rule and our configuration, we can start snort in NIDS mode, and tell it to output any alerts right to the console. We will run Snort from the command line, using the following flags:

-A console	The 'console' option prints fast mode alerts to stdout
-q	Quiet mode. Don't show banner and status report.
-u snort	Run Snort as the following user after startup
-g snort	Run Snort as the following group after startup
-c /etc/snort/snort.conf	The path to our `snort.conf` file
-i eth0	The interface to listen on

```
$ sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

Snort is now running, processing all packets that arrive on eth0, comparing them to the rules it has loaded (in this case our single ICMP Ping rule), and sending all alerts generated when a packet matches our rule to the console. From another computer, ping the IP address of eth0 on the Snort computer (or alternately ping from the Snort host to another machine, or to its own eth0, but not loopback interface), and you should see console output similar to what is displayed below (in the below example, the Snort server is listening on eth0 with and IP address of 10.0.0.218, and the computer generating the ping is 10.0.0.169).

```
01/01-16:03:28.608782  [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 10.0.0.169 -> 10.0.0.218
01/01-16:03:28.608824  [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 10.0.0.218 -> 10.0.0.169
01/01-16:03:29.610094  [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 10.0.0.169 -> 10.0.0.218
01/01-16:03:29.610123  [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 10.0.0.218 -> 10.0.0.169
01/01-16:03:30.611143  [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 10.0.0.169 -> 10.0.0.218
01/01-16:03:30.611173  [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 10.0.0.218 -> 10.0.0.169
01/01-16:03:31.612174  [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 10.0.0.169 -> 10.0.0.218
01/01-16:03:31.612202  [**] [1:10000001:0] ICMP test [**] [Priority: 0] {ICMP} 10.0.0.218 -> 10.0.0.169
^C*** Caught Int-Signal
```

Use `ctrl-c` to stop Snort from running. Note that Snort has saved a copy of this information in `/var/log/snort`, with the name `snort.log.nnnnnnnnn` (the numbers may be different). At this point Snort is running correctly in NIDS mode and generating alerts.

10 Installing Barnyard2

It is resource intensive for Snort to write events in human-readable mode, either to the console or to text files, as done above. Ideally, we would like Snort to write events to a MySQL database so we can view, search, and profile the events. To efficiently get Snort events into a MySQL database, we use Barnyard2. Snort outputs events in binary form to a folder, and then Barnyard2 reads those events asynchronously and copies them to our MySQL database.

First install the Barnyard2 pre-requisites:

```
sudo apt-get install -y mysql-server libmysqlclient-dev mysql-client autoconf libtool
```

The install will prompt you to create a root mysql user password. For the examples below, we will use MYSQLROOTPASSWORD. You should choose something different and more secure, and store it safely. We will also be creating a Snort MySQL user account, and the password for that account will be MYSQLSNORTPASSWORD, please note the difference between these two passwords.

We need to tell snort that it should output it's alerts in a binary format that Barnyard2 can process. To do that, edit the `/etc/snort/snort.conf` file, and after line 520 (the commented line starting with the hash sign) add the following line:

```
output unified2: filename snort.u2, limit 128
```

So that lines 520 and 521 now looks like:

```
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types}
output unified2: filename snort.u2, limit 128
```

Now download and install Barnyard2:

```
cd ~/snort_src
wget https://github.com/firnsy/barnyard2/archive/master.tar.gz -O barnyard2-2-1.13.tar.gz
tar zxvf barnyard2-2-1.13.tar.gz
cd barnyard2-master
autoreconf -fvi -I ./m4
```

Depending on your OS version (x86 or x86_64, you need to point the install to the correct MySQL library. Run *one* of the following two lines to configure the build process, depending on your architecture:

```
./configure --with-mysql --with-mysql-libraries=/usr/lib/x86_64-linux-gnu
./configure --with-mysql --with-mysql-libraries=/usr/lib/i386-linux-gnu
```

Now complete the build and install Barnyard2 to `/usr/local/bin/barnyard2`:

```
make
sudo make install
```

We need to copy and create some files for Barnyard2 to run

```
cd ~/snort_src/barnyard2-master
sudo cp etc/barnyard2.conf /etc/snort
sudo mkdir /var/log/barnyard2
sudo chown snort.snort /var/log/barnyard2

sudo touch /var/log/snort/barnyard2.waldo
sudo chown snort.snort /var/log/snort/barnyard2.waldo
sudo touch /etc/snort/sid-msg.map
```

Since Barnyard2 saves alerts to our MySQL database, we need to create that database, as well as a 'snort' MySQL user to access that database. Run the following commands to create the dabase and MySQL user. When prompted for a password, use the MYSQLROOTPASSWORD. You will also be setting the MySQL snort user password in the 2nd command (to MYSQLSNORTPASSWORD), so change it there as well.

```
echo "create database snort;" | mysql -u root -p
mysql -u root -p -D snort < ~/snort_src/barnyard2-master/schemas/create_mysql
echo "grant create, insert, select, delete, update on snort.* to \
 snort@localhost identified by 'MYSQLSNORTPASSWORD'" | mysql -u root -p
```

We need to tell Barnyard2 how to connect to the MySQL database. Edit `/etc/snort/barnyard2.conf`, and at the end of the file add this line (changing password as required):

```
output database: log, mysql, user=snort password=MYSQLSNORTPASSWORD dbname=snort host=localhost
```

Since the password is stored in cleartext in the `barnyard2.conf` file, we should prevent other users from reading it:

```
sudo chmod o-r  /etc/snort/barnyard2.conf
```

Now we want to test that Snort is writing events to the correct binary log file, and that Barnyard2 is reading those logs and writing the events to our MySQL database. We could just start both programs up in daemon mode and generate some events by pinging the interface (triggering the rule we created earlier), but it's better to test one portion at a time.

Run Snort in alert mode (the command we run below is how Snort will normally be run when we set it up as a daemon, except we aren't using the `-D` flag which causes it to run as a daemon).

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

Ping the interface eth0 from another computer, you won't see any output on the screen because Snort wasn't started with the `-A console` flag like before. Once the ping stops, type ctrl-c to stop Snort. you should see a new file in the `/var/log/snort` directory with following name (the numbers will be different because they are based on the current time. The snort.log is the output file we created before):

```
user@snortserver:/var/log/snort$  ls -l /var/log/snort/
total 8
-rw-r--r-- 1 snort snort    0 Nov 11 14:07 barnyard2.waldo
-rw------- 1 snort snort  744 Nov 11 13:49 snort.log.1415710140
-rw------- 1 snort snort 1360 Nov 11 14:10 snort.u2.1415711432
```

We now run Barnyard2 and tell it to look at these events and load into the Snort database. We use the following flags with Barnyard2:

-c /etc/snort/barnyard2.conf	The path to the `barnyard2.conf` file
-d /var/log/snort	The folder to look for Snort output files
-f snort.u2	The Filename to look for in the above directory (snort.u2.nnnnnnnnnn)
-w /var/log/snort/barnyard2.waldo	The location of the waldo file (bookmark file)
-u snort	Run Barnyard2 as the following user after startup
-g snort	Run Barnyard2 as the following group after startup

Run Barnyard2 with the following command:

```
sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo \
-g snort -u snort
```

Barnyard2 will start up (be patient, it can take some time), and then it will process the alerts in the `/var/log/snort/snort.u2.nnnnnnnnnn` file, write them to both the screen and the database, and then wait for more events to appear in the `/var/log/snort directory`. use `Ctrl-c` to stop the process. You should see output similar to:

```
(...)
Opened spool file '/var/log/snort/snort.u2.1389532785'
Closing spool file '/var/log/snort/snort.u2.1389532785'. Read 8 records
Opened spool file '/var/log/snort/snort.u2.1389535513'
01/12-19:35:17.878473  [**] [1:10000001:1] ICMP test [**] [Classification ID: 0] [Priority ID: 0] {ICMP} 10.0.0.169 -> 10.0.0.218
01/12-19:35:18.848924  [**] [1:10000001:1] ICMP test [**] [Classification ID: 0] [Priority ID: 0] {ICMP} 10.0.0.169 -> 10.0.0.218
01/12-19:35:19.849956  [**] [1:10000001:1] ICMP test [**] [Classification ID: 0] [Priority ID: 0] {ICMP} 10.0.0.169 -> 10.0.0.218
01/12-19:35:20.859685  [**] [1:10000001:1] ICMP test [**] [Classification ID: 0] [Priority ID: 0] {ICMP} 10.0.0.169 -> 10.0.0.218
Waiting for new data

^C*** Caught Int-Signal
```

We now want to check the MySQL database to see if Barnyard2 wrote the events. Run the following command to query the MySQL database, you will be prompted for the MySQL Snort user password: MYSQLSNORTPASSWORD (not the MySQL root password):

```
mysql -u snort -p -D snort -e "select count(*) from event"
```

If sucessful, you will then get the following output, showing the 8 events written to the database from the ICMP request and reply packets (when you ping from a windows system, it will by default send 4 ICMP messages. If you pinged from another system the count could be different):

```
+----------+
| count(*) |
+----------+
|       8  |
+----------+
```

Congratulations, if you have similar output (count greater than 0) as above, then Snort and Barnyard2 are properly installed and configured. We will create startup scripts later to launch both applications as daemons automatically on boot up.

11 Installing PulledPork

PulledPork is a perl script that will download, combine, and install/update snort rulesets from various locations for use by Snort. if you would rather install rulesets manually, see Apendix: Installing Snort Rules Manually.

Install the PulledPork pre-requisites:

```
sudo apt-get install -y libcrypt-ssleay-perl liblwp-useragent-determined-perl
```

Download and install the PulledPork perl script and configuration files:

```
cd ~/snort_src
wget https://pulledpork.googlecode.com/files/pulledpork-0.7.0.tar.gz
tar xvfvz pulledpork-0.7.0.tar.gz
cd pulledpork-0.7.0/
sudo cp pulledpork.pl /usr/local/bin
sudo chmod +x /usr/local/bin/pulledpork.pl
sudo cp etc/*.conf /etc/snort
```

The last command will copy the following files from the PulledPork directory into the Snort configuration directory:
  disablesid.conf
  dropsid.conf
  enablesid.conf
  modifysid.conf
  pulledpork.conf

Create some files and directories that PulledPork expects:

```
sudo mkdir /etc/snort/rules/iplists
sudo touch /etc/snort/rules/iplists/default.blacklist
```

Check that PulledPork runs by checking the version, using the `-V` flag:

```
user@snortserver:~$ /usr/local/bin/pulledpork.pl -V
     PulledPork v0.7.0 - Swine Flu!
user@snortserver:~$
```

12 Configuring PulledPork to Download Rulesets

You need to create an account on `http://snort.org` in order to get a unique Oinkcode that will allow you to download updated rulesets.

Configure PulledPork by editing `/etc/snort/pulledpork.conf` with the following command:

```
sudo vi /etc/snort/pulledpork.conf
```

Anywhere you see <oinkcode>enter your oinkcode from snort.org account:

```
Line 19 & 26: enter your oinkcode where appropriate
Line 27 & 30 : leave alone (uncommented) to use the Emerging Threats rule set

Line 72: change to: rule_path=/etc/snort/rules/snort.rules
Line 87: change to: local_rules=/etc/snort/rules/local.rules
Line 90: change to: sid_msg=/etc/snort/sid-msg.map
Line 117: change to: config_path=/etc/snort/snort.conf

Line 131: change to: distro=Ubuntu-10-4

Line 139: change to: black_list=/etc/snort/rules/iplists/default.blacklist
Line 148: change to: IPRVersion=/etc/snort/rules/iplists

Line 194: Uncomment and change to: enablesid=/etc/snort/enablesid.conf
Line 195: Uncomment and change to: dropsid=/etc/snort/dropsid.conf
Line 196: Uncomment and change to: disablesid=/etc/snort/disablesid.conf
Line 197: Uncomment and change to: modifysid=/etc/snort/modifysid.conf
```

We want to run PulledPork manually this one time to make sure it works. The following flags are used with PulledPork:

-l	Write detailed logs to /var/log
-c /etc/snort/snort.conf	The path to our pulledpork.conf file

Run the following command:

```
sudo /usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf -l
```

PulledPork should finish with output similar to the below (showing the new rules downloaded, in the example below there are over 22,000 new rules downloaded):

```

(...)
Rule Stats...
        New:-------22764
        Deleted:---0
        Enabled Rules:----6353
        Dropped Rules:----0
        Disabled Rules:---16410
        Total Rules:------22763
IP Blacklist Stats...
        Total IPs:-----9374

Done
user@snortserver:~$
```

When PulledPork completes successfully as above, You should now see `snort.rules` in `/etc/snort/rules`, and .so rules in `/usr/local/lib/snort_dynamicrules`.

Pulled Pork combines all the rulesets into one file. You need to make sure to add the line: `include $RULE_PATH/snort.rules` to the `snort.conf file`, or the PulledPork rules will never be read into memory when Snort starts.

Edit `/etc/snort/snort.conf`, and append to the end of the file (on a new line):

`include $RULE_PATH/snort.rules`

Since we've modified the Snort configuration file (via the loaded ruleset files), we should test the Snort configuration file. This will also check the new `snort.rules` file that PulledPork created:

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

Once that is sucessfull, we want to set PulledPork to run daily. To do this, we add the PulledPork script to root's crontab:

```
sudo crontab -e
```

Append the follwoing line in crontab:

```
01 04 * * * /usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf -l
```

13 Creating Startup Scripts

We will use Upstart rather than SystemV init scrips to run both Snort and Barnyard2. First we need to create the Snort startup script:

```
sudo vi /etc/init/snort.conf
```

With the following content (note that we are using the same flags as when we tested above, except for the addition of the `-D` flag, which tells Snort to run as a daemon):

```
description "Snort NIDS Service"
stop on runlevel [!2345]
start on runlevel [2345]
script
     exec /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0 -D
end script
```

Now make the script executable, and tell Upstart that the script exists, and then verify that it is visible:

```
user@snortserver:~$ sudo chmod +x /etc/init/snort.conf
user@snortserver:~$ initctl list | grep snort
snort stop/waiting
user@snortserver:~$
```

Do the same for our Barnyard2 script (note that the exec command should be one one line):

```
sudo vi /etc/init/barnyard2.conf
```

With the following content:

```
description "Barnyard2 service"
stop on runlevel [!2345]
start on runlevel [2345]
script
    exec /usr/local/bin/barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort
     /barnyard2.waldo -g snort -u snort -D
end script
```

Make the script executable and check to see that it installed correctly:

```
user@snortserver:~$ sudo chmod +x /etc/init/barnyard2.conf
user@snortserver:~$ initctl list | grep barnyard
barnyard2 stop/waiting
user@snortserver:~$
```

Reboot the computer and check that both services are started:

```
user@snortserver:~$ service snort status
snort start/running, process 1116
user@snortserver:~$ service barnyard2 status
barnyard2 start/running, process 1109
user@snortserver:~$
```

If Barnyard2 does not startup, you may need to delete then re-create the Snort database. Follow the instructions in Apendix: Re-Creating the Snort Database if this is needed.

14 Installing BASE

BASE (http://base.professionallyevil.com/) is the Basic Analysis and Security Engine, which provides a web front-end to the database of Snort events.

First, install the BASE pre-requisites:

```
sudo apt-get install -y apache2 libapache2-mod-php5 php5 php5-mysql php5-common \
 php5-gd php5-cli php-pear
```

Install pear Image_Graph, with the following commands.

```
sudo pear install -f Image_Graph
```

when installation completes, you can ignore the warnings, but make sure you see the three 'install ok' lines at the end:

```
(...)
                WARNING: failed to download pear.php.net/Image_Graph within preferred state "stable", will
     instead download version 0.8.0, stability "alpha"
                WARNING: failed to download pear.php.net/Image_Canvas within preferred state "stable", will
     instead download version 0.3.5, stability "alpha"

                ...done: 9,501 bytes
                install ok: channel://pear.php.net/Image_Color-1.0.4
                install ok: channel://pear.php.net/Image_Canvas-0.3.5
                install ok: channel://pear.php.net/Image_Graph-0.8.0
user@snortserver:~$
```

Download and install ADODB 5.18 (not ADODB 5.19, it is not compatible with the current version of BASE):

```
cd ~/snort_src
wget http://sourceforge.net/projects/adodb/files/adodb-php5-only/adodb-518-for-php5/adodb518a.tgz/download \
-O adodb518.tgz
tar -xvzf adodb518.tgz
sudo mv adodb5 /var/adodb
```

Download and install BASE

```
cd ~/snort_src
wget http://sourceforge.net/projects/secureideas/files/BASE/base-1.4.5/base-1.4.5.tar.gz
tar -zxvf base-1.4.5.tar.gz
```

Ubuntu 12 and 13 use Apache 2.2 while Ubuntu 14 uses Apache 2.4. The difference determines the folder location to install BASE:

For Ubuntu 12 and 13 running Apache 2.2, install BASE into `/var/www/base/`, and then configure:

```
sudo mv base-1.4.5 /var/www/base/
cd /var/www/base
sudo cp base_conf.php.dist base_conf.php
```

For Ubuntu 14 running Apache 2.4, install BASE into `/var/www/html/base/`, and then configure:

```
sudo mv base-1.4.5 /var/www/html/base/
cd /var/www/html/base
sudo cp base_conf.php.dist base_conf.php
```

Open the BASE configuration file for editing as listed below:

```
% (Ubuntu 12 and 13):
    sudo vi /var/www/base/base_conf.php
% (Ubuntu 14):
    sudo vi /var/www/html/base/base_conf.php
```

Make the following edits to the `base_conf.php` configuration file:

```
$BASE_urlpath = '/base';                 # line 50
$DBlib_path = '/var/adodb/';             # line 80
$alert_dbname   = 'snort';               # line 102
$alert_host     = 'localhost';
$alert_port     = '';
$alert_user     = 'snort';
$alert_password = 'MYSQLSNORTPASSWORD';  # line 106
```

Set permissions on the BASE folder

```
% (Ubuntu 12 and 13):
     sudo chown -R www-data:www-data /var/www/base
% (Ubuntu 14):
     sudo chown -R www-data:www-data /var/www/html/base
```

Since the MySQL password is stored in plaintext in the `base_conf.php` file, we should prevent other users from reading it:

```
% (Ubuntu 12 and 13):
     sudo chmod o-r /var/www/base/base_conf.php
% (Ubuntu 14):
     sudo chmod o-r /var/www/html/base/base_conf.php
```

BASE is now configured work with our database, restart Apapche:

```
sudo service apache2 restart
```

The last step in configuring BASE is done through its web page:
1. Browse to `http://snort_ip_address/base/index.php` and click on “`setup page`” link.
2. Click on “`Create BASE AG`” button on the upper right of the page
3. Click on the “`Main page`” line.

BASE is now configured, you should see the events that we generated with our pings earlier, and can view packet detail, delete, and other tasks.

15 ESXi and Snort in Promiscuous Mode

Often you want your Snort NIDS to listen on an adapter that receives all traffic for a switch. VMware calls this “Promiscuous Mode”, while Cisco calls this a “Mirror Port”. To configure your ESXi server to mirror all traffic to an interface on a Virtual Machine (such as the interface for our Snort VM), follow the steps below, from VMware’s website[7]:

1. Log into the ESXi/ESX host or vCenter Server using the ESXi Client.
2. Select the ESXi/ESX host (the VMware Server) in the inventory.
3. Click the Configuration tab.
4. In the Hardware section, click Networking.
5. Click Properties of the virtual switch (the switch that Snort has its listening interface on) for which you want to enable promiscuous mode.
6. Select the virtual switch or portgroup you wish to modify and click Edit.
7. Click the Security tab.
8. From the Promiscuous Mode dropdown menu, click Accept.

Once VMware is configured to permit promiscuous mode, you then need to configure the interface that Snort is listening on for promiscuous mode. to do this, edit `/etc/network/interfaces`:

```
sudo vi /etc/network/interfaces
```

and make modifications similar to the following, depending on the configuration of your system:

```
# The primary network interface
auto eth0
iface eth0 inet dhcp

# Interface that Snort listens on
auto eth1
iface eth1 inet manual
        up ifconfig $IFACE 0.0.0.0 up
        up ip link set $IFACE promisc on
        down ip link set $IFACE promisc off
        down ifconfig $IFACE down
```

In the above example, Snort will listen on `eth1` (remember that this also has to be changed in the Snort daemon script (the interface referenced by the `-i` flag in `/etc/init/snort.conf`). We choose not to set an IP addresson the interface that Snort will listen on, since this helps to protect the system from exploits. Management in the above example will be through `eth0` which is configured for DHCP. The command `up ip link set $IFACE promisc on` is what configures the interface for promiscuous mode, and is how the system knows to process all traffic the interface sees, not just traffic that is specifically for the adapter.

To test this configuration, restart networking (or restart the system) and ensure that Snort has started, and is listening on the correct interface. Ping between two hosts on the subnet (two hosts that are not the Snort server) and you should see the events logged.

An easy way to see if this is working is to stop the Snort Daemon (with `sudo service snort stop`), then run the following (change the interface as needed):

```
$ sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth1
```

When you ping between the two hosts that aren’t the Snort server, but which are on the same subnet as the Snort server, you should see the events written to the screen. Use `ctr-c` to stop Snort.

[7] http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1004099

A Apendix: Installing Snort Rules Manually

If you just want to test Snort manually, and want to use the rules from snort without setting up PulledPork, follow the instructions below. You will need a Oinkcode (free with an account from snort.org)

We need to un-comment all the #include lines in snort.conf, as the downloaded rules will be a series of rule files, rather than the one that PulledPork creates:

```
sudo sed -i 's/\#include \$RULE\_PATH/include \$RULE\_PATH/' /etc/snort/snort.conf
```

Download the rules, replacing <oinkcode>with your personal Snort code. you might also want to get a newer version of the rules:

```
cd ~/snort_src
wget https://www.snort.org/reg-rules/snortrules-snapshot-2956.tar.gz/<SNORTCODE> -O snortrules-snapshot-2956.
    tar.gz
sudo tar xvfvz snortrules-snapshot-2956.tar.gz -C /etc/snort
```

Move all new files from `/etc/snort/etc` to `/etc/snort` (and get rid of `/etc/snort/etc` folder that was copied as well):

```
sudo cp ./*.conf* ../
sudo cp ./*.map ../
cd /etc/snort
sudo rm -Rf /etc/snort/etc
```

Now modify `/etc/snort/snort.conf` with any changes from the original `snort.conf`.
We want the new `snort.conf` in case it references any new rulesets.

Test the configuration file with Snort:

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

You can now run snort as you normally would (with a startup script or manually).

B Apendix: Re-Creating the Snort Database

Occasionally you will have issues with Barnyard2 not starting, and sometimes deleting the Snort database can help.

Stop Snort and Barnyard2 service:

```
sudo service snort stop
sudo service barnyard2 stop
```

Delete the Snort satabase (use MySQL root password when prompted):

```
mysql -u root -p
>  drop database snort;
Query OK, 16 rows affected (0.04 sec)
> exit
```

Recreate the Snort database as we did originally when installing Barnyard2. Enter the MySQL root password (MYSQLROOTPASSWORD) when prompted, and change MYSQLSNORTPASSWORD to your MySQL Snort user password.

```
echo "create database snort;" | mysql -u root -p
mysql -u root -p -D snort < ~/snort_src/barnyard2-master/schemas/create_mysql
echo "grant create, insert, select, delete, update on snort.* to \
 snort@localhost identified by 'MYSQLSNORTPASSWORD'" | mysql -u root -p
```

And finally you will need to re-configure BASE from the web page:
1. Browse to `http://snort_ip_address/base/index.php` and click on “`setup page`” link.
2. Click on “`Create BASE AG`” button on the upper right of the page
3. Click on the “`Main page`” line.

Reboot and see if the services start again:

```
sudo reboot
```

Additionally you can delete all files in `/var/log/snort` as well, to see if that helps (you also need to then recreate an empty waldo file for barnyard2 to run correctly):

```
sudo rm /var/log/snort/*
sudo touch /var/log/snort/barnyard2.waldo
```