Snorby on Ubuntu Server 9.04 (32-bit) with Multiple Interfaces on Apache

With this document, you should have a working Snorby installation on Ubuntu Server 9.04 up and running in as little as a couple hours depending on connection speeds and how fast you can type. This document assumes a very basic install of Ubuntu Server 9.04 (32-bit) already up and running. (If you need help with that or want me to append this document to include the base install just let me know.) If you need to a write-up with installing the current version of Snort, let me know and I'll get working on that. As always, if anything needs changing or I forgot something just let me know. Jonathan Krautter (secretmoose at gmail dot com)

Let's start with the essentials:

sudo apt-get install mysql-server libmysqlclient15-dev ruby rubygems libmysql-ruby ruby1.8-dev build-essential debian-keyring git-core libopenssl-ruby apache2 apache2-threaded-dev

First let's set up MySQL by setting the root password during the install then we'll set up the database for Snort/Snorby:

mysql –u root –p

<enter your password>

create database databasename;

grant usage on *.* to snorbyuser@localhost identified by snorbypassword;

grant all privileges on snortdb.* to snorbyuser@localhost;

quit

Now let's get Ruby up and running:

sudo gem install rake prawn

sudo gem install -v=2.3.2 rails

sudo gem install dbd-mysql

sudo gem install passenger

Let's hook passenger into Apache and edit the configs so things will work:

sudo /var/lib/gems/1.8/gems/passenger-2.2.5/bin/passenger-install-apache2-module

sudo vi /etc/apache2/mods-enabled/env.load (add the lines below)

LoadModule passenger_module /usr/local/lib/ruby/gems/1.8/gems/passenger-2.2.5/ext/apache2/mod_passenger.so (all on one line)

PassengerRoot /usr/local/lib/ruby/gems/1.8/gems/passenger-2.2.5

PassengerRuby /usr/local/bin/ruby

save and exit

cd /usr/bin

sudo ln –s /var/lib/gems/1.8/bin/rake

Now we'll work on Snort, make sure to answer No for setting up the databse:

sudo apt-get install snort-mysql snort-doc

sudo vi snort.conf (edit the following lines)

output log_tcpdump: tcpdump.log (comment this line out with a # at the beginning)

output database: log, mysql, user=snorbyuser password=snorbypassword dbname=snorbydbname host=localhost (remove the # at the beginning and edit accordingly)

save and exit

sudo vi /etc/oinkmaster.conf (comment out the download URL)

save and exit

sudo rm db-pending-config

sudo vi /etc/default/snort (edit the following line shown below)

ALLOW_UNAVAILABLE= "no" (change to yes)

Save and exit

Set up the database:

cd /usr/share/doc/snort-mysql

zcat create_mysql.gz |mysql -u root -p snorbydbname<enter password>

Finish setting up Snort:

sudo dpkg --configure --pending

sudo dpkg-reconfigure snort-mysql (answer the questions according to your environment)

Now we're ready for Snorby, first let's create a non-root user to take ownership of the Snorby files once we're done (you may skip this part step if you plan to use your local account as the non root user account to chown Snorby later):

sudo useradd nonrootuser

Now let's get and setup Snorby:

cd /var/www

sudo git clone git://github.com/mephux/Snorby.git

cd Snorby/config/

sudo cp database.yml.example database.yml

sudo cp email.yml.example email.yml

sudo vi database.yml (edit for your system)

sudo vi email.yml (edit)

cd /var/www/Snorby

sudo rake gems:install

sudo rake snorby:setup RAILS_ENV=production

cd ..

sudo chown -R nonrootuser Snorby/ (or your local account if you so choose)

Now we'll set up Apache so it can start serving the web portion of Snorby:

cd /etc/apache2/sites-available

sudo vi sitename (add the following lines listed below)

<VirtualHost *:80>

ServerAdmin name@name.com

ServerName (servername)

ServerAlias alias.something.tld

DocumentRoot /var/www/Snorby/public

RailsBaseURI /

```
        <directory "/var/www/Snorby/public">
                AllowOverride All
                Order deny,allow
                Allow from all
        </directory>
        </VirtualHost>
```

save and exit

sudo a2ensite sitename

Plug in all your connections and reboot your server and you should be good to go!

Just go to http://'server address'

Some notes:

By default apt installs oinkmaster, however the installed version of snort is older than current from snort.org so unless you feel like hand editing all your rules, I'd just accept whatever updates come from Ubuntu. You can also run the rule sets from emergingthreats.net by adding the following line to /etc/oinkmaster.conf:

url = http://www.emergingthreats.net/rules/emerging.rules.tar.gz

Now we'll add the script to cron.daily so we can stay current:

sudo vi /etc/cron.daily/ruleupdates (then add the following lines)

```
        #./bin/bash
        oinkmaster -o /etc/snort/rules
```

save and exit

And add the following lines to /etc/snort/snort.conf and edit to taste:

 # Below are the rule sets and configs for Emerging Threats.

 # This var is required for several sigs in the POLICY ruleset. It is plural because you can do a range of ports

var SSH_PORTS 22

#do this:

#include $RULE_PATH/emerging-all.rules

#or these

include $RULE_PATH/emerging-attack_response.rules

include $RULE_PATH/emerging-dos.rules

include $RULE_PATH/emerging-exploit.rules

include $RULE_PATH/emerging-game.rules

include $RULE_PATH/emerging-inappropriate.rules

include $RULE_PATH/emerging-malware.rules

include $RULE_PATH/emerging-p2p.rules

include $RULE_PATH/emerging-policy.rules

include $RULE_PATH/emerging-scan.rules

include $RULE_PATH/emerging-virus.rules

include $RULE_PATH/emerging-voip.rules

include $RULE_PATH/emerging-web.rules

include $RULE_PATH/emerging-web_client.rules

include $RULE_PATH/emerging-web_server.rules

include $RULE_PATH/emerging-web_specific_apps.rules

include $RULE_PATH/emerging-user_agents.rules

include $RULE_PATH/emerging-current_events.rules

#Now choose which of the below you want. These are very specific. Use only what you need of these, not all

#There are general sigs in the web ruleset that cover much of this

#include $RULE_PATH/emerging-web_sql_injection.rules

#These are specific blocking and alerting

#Do not run these unless you update often. These are updated at least daily

#Those with a -BLOCK are preconfigured with Snortsam

(http://www.snortsam.net) block statements. Run one or the other, not both

#include $RULE_PATH/emerging-botcc-BLOCK.rules

include $RULE_PATH/emerging-botcc.rules

#include $RULE_PATH/emerging-compromised-BLOCK.rules

include $RULE_PATH/emerging-compromised.rules

#include $RULE_PATH/emerging-drop-BLOCK.rules

include $RULE_PATH/emerging-drop.rules

#include $RULE_PATH/emerging-dshield-BLOCK.rules

include $RULE_PATH/emerging-dshield.rules

#include $RULE_PATH/emerging-rbn-BLOCK.rules

include $RULE_PATH/emerging-rbn.rules

#include $RULE_PATH/emerging-tor-BLOCK.rules

include $RULE_PATH/emerging-tor.rules