Snorby on RHEL 5.4 (64-bit native) with Multiple Interfaces on Apache

With this document, you should have a working Snorby installation on RedHat Enterprise Linux 5.4 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 RHEL 5.4 (64-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.) Make sure to register with Snort.org and get an Oink Code! 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:

yum install mysql-server mysql-devel httpd httpd-devel automake autoconf gcc gcc-c++ gettext-devel libpcap libpcap-devel pcre-devel rpm-build flex bison

If at all possible install the 64-bit versions of the software. If you need to see what's available, just run a simple query as follows:

yum list |grep "package name"

The one that's 64-bit should be obvious. This will also tell you if it is already installed or not.

Now let's set up some temp stuff so we can get things done:

cd /tmp

mkdir mysql

mkdir snort

mkdir ruby

mkdir oinkmaster

mkdir git

First let's set up MySQL:

cd /tmp/mysql

wget http://dev.mysql.com/get/Downloads/ MySQL-5.0/MySQL-shared-compat-5.0.86-1.rhel5.x86_64.rpm/from/http://mirror.services.wisc.edu/mysql/

rpm –Uvh *.rpm

chkconfig mysqld on

service mysqld start

mysqladmin –u root password ‘new password’

Now we’ll set up the database for Snort/Snorby:

mysql –u root –p

<enter your password>

create database ‘database name’;

grant usage on *.* to ‘snorby user’@localhost identified by 'snorby password';

grant all privileges on snortdb.* to ‘snorby user’@localhost;

quit

Now let’s get Ruby up and running:

cd /tmp/ruby

wget ftp://ftp.ruby-lang.org/pub/ruby/stable/ruby-1.8.7-p173.tar.gz

wget http://rubyforge.org/frs/download.php/60718/rubygems-1.3.5.tgz

tar -xvf ruby-1.8.7-p173.tar.gz

tar -xvf rubygems-1.3.5.tgz

cd /ruby-1.8.7-p173

./configure

make

make install

cd ..

cd rubygems-1.3.5

ruby setup.rb

gem install rake prawn

gem install -v=2.3.2 rails

gem install dbd-mysql

gem install passenger

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

passenger-install-apache2-module

vi /etc/httpd/conf/httpd.conf (add the following lines in Modules section)

LoadModule passenger_module /usr/local/lib/ruby/gems/1.8/gems/passenger-2.2.5/ext/apache2/mod_passenger.so

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

PassengerRuby /usr/local/bin/ruby

save and exit

cd /usr/lib

ln –s /usr/lib64/libmysqlclient.so

cd /usr/bin

ln –s /usr/local/lib/ruby/gems/1.8/gems/rake-0.8.7/bin/rake

Now we'll work on Snort:

cd /tmp/snort

wget http://dl.snort.org/snort-current/snort-2.8.5.tar.gz

rpmbuild --with mysql -ta snort-2.8.5.tar.gz

cd /usr/src/redhat/RPMS/x86_64

rpm -Uvh snort*.rpm

Let's configure Snort so that it will start up correctly and work as we need it:

vi /etc/sysconfig/snort (edit the INTERFACE line and comment out the ALERTMODE line with # at the beginning of the line)

INTERFACE="eth1 eth2 eth3 eth4 eth5"

save and exit

vi /etc/snort/snort.conf (uncomment and edit the following line)

output database: log, mysql, user='snorby user' password='snorby password' dbname='snorby db name' host=localhost

var HOME_NET any (CAN NOT BE "ANY" IF YOU ARE USING EMERGING THREATS RULES)

save and exit

vi /etc/init.d/snortd (edit the following line so that it starts after mysqld after and stops before mysqld.

You can see where mysqld starts by typing: cat /etc/init.d/mysqld |grep chkconfig

The first set of numbers are run levels, second number is start order, third is stop order)

 # chkconfig: 2345 70 25

Set up the database:

mysql -u root -p 'snortdatabase' </usr/share/snort-2.8.5/schemas/create_mysql

Set up oinkmaster and the rules:

cd /tmp/oinkmaster

wget http://downloads.sourceforge.net/project/oinkmaster/oinkmaster/2.0/oinkmaster-2.0.tar.gz

tar -xvf oinkmaster-2.0.tar.gz

touch /etc/snort/rules/local.rules

cd oinkmaster-2.0

vi oinkmaster.conf (then add/modify the following lines)

url = http://www.snort.org/pub-bin/oinkmaster.cgi/'oink code'/snortrules-snapshot-CURRENT.tar.gz

tmpdir = /tmp/oinkmaster/

save and exit

cp oinkmaster.conf /etc/oinkmaster.conf

perl oinkmaster.pl -o /etc/snort/rules

Set up oinkmaster to run daily to get rule updates by creating a script:

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

#./bin/bash

perl /tmp/oinkmaster/oinkmaster-2.0/oinkmaster.pl -o /etc/snort/rules

save and exit

Or by adding the following lines to the crontab by running crontab -e:

30 2 * * * perl /tmp/oinkmaster/oinkmaster-2.0/oinkmaster.pl -o /etc/snort/rules -b /etc/snort/backup 2>&1

save and exit

Now would be a good time to edit any interfaces you want to start-up at boot time in /etc/sysconfig/network-scripts/

Setup git:

cd /tmp/git

wget http://www.kernel.org/pub/software/scm/git/git-1.6.4.4.tar.gz

wget http://curl.haxx.se/download/curl-7.19.6.tar.gz

tar -xvf curl-7.19.6.tar.gz

tar -xvf git-1.6.4.4.tar.gz

cd curl-7.19.6

./configure

make

make install

vi /etc/ld.so.conf (add the following line)

/usr/local/lib

save and exit

ldconfig

cd /tmp/git/git-1.6.4.4

./configure --with-curl=/usr/local

make

make install

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:

Useradd 'non-root user'

Now let's get and setup Snorby:

cd /var/www

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

cd Snorby/config/

cp database.yml.example database.yml

cp email.yml.example email.yml

vi database.yml (edit for your system)

vi email.yml (edit)

cd /var/www/Snorby

rake gems:install

rake snorby:setup RAILS_ENV=production

cd ..

chown -R 'non-root user' Snorby/

Now we'll set up Apache so it can start serving the web portion of Snorby, I know this is not the preferred way and it's cheating but I was in a hurry and it worked:

cd /etc/httpd/conf.d

vi welcome.conf (comment out all the lines and add the following ones 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

Check and make sure all your interfaces are configured correctly and up and running.

Make sure that httpd is set to start at boot with the following:

```
chkconfig httpd on
```

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

Just go to http://'server address'

Some notes:

I had issues with running this configuration with MySQL 5.1 in that it would just stop logging at 3AM every day.

I've also seen some issues running on Ruby 1.8.6 watching it chew up 12GB of RAM in less than a day.

You can also run the rule sets from emergingthreats.net by adding the following lines to /etc/oinkmaster.conf:

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

And add the following lines to /etc/snort/snortethx.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