Snort Installation Manual

Snort, MySQL and ACID on Redhat 7.3

August, 2002

Version 1.5

Prepared by Steven J. Scott
sjscott007@yahoo.com
http://home.earthlink.net/~sjscott007/

Table of Contents

ACKNOWLEDGMENTS **4**

COMMENTS & CORRECTIONS **4**

WHERE TO GET THE LATEST VERSION OF THIS GUIDE **4**

INTRODUCTION **4**

REQUIRED SOFTWARE **4**

CONCEPTUAL TOPOLOGY **5**

SENSOR PLACEMENT MODEL **6**

HOW TO USE THIS GUIDE **8**

REDHAT 7.3 INSTALLATION **8**

POST REDHAT INSTALLATION **10**

SNORT INSTALLATION **10**

WEBMIN INSTALLATION **12**

ACID CONSOLE & CENTRALIZED MYSQL DATABASE **18**

ACCESSING THE ACID CONSOLE **21**

SENSOR TUNING **23**

TIME ZONES **28**

NETWORK TIME PROTOCOL (NTP) **29**

MAINTENANCE **29**

SENSOR CHARACTERISTICS **33**

ADDITIONAL INFORMATION.....35

CHANGE LOG.....35

Acknowledgments

I would like to thank the following people for their help in creating this guide, and backing the project that helped create it.

Fred Beste
His aptitude for empowering and complementing his skills with that of his people will only contribute to his continued success. I cannot begin to explain the great things that can be accomplished when you have control over your own destiny. It just shows how great leaders let their people lead, and share the wealth with those that perform.

Bob Kaelin
For painstakingly help roll out the many sensors while ensuring that the documentation flowed throughout the entire process. Magnificent!!

Comments & Corrections

If you find any errors or would like make comments please send them to sjscott007@yahoo.com.

Where to get the latest version of this Guide

The latest version of this guide can be found at http://home.earthlink.net/~sjscott007/.

You can also find it mirrored at http://www.snort.org.

Introduction

The purpose of this guide is to document the installation and configuration of a complete Snort implementation. This guide contains all the necessary information for installing and understanding the architectural layout of the implementation.

The information in this guide was written for implementing Snort 1.8 using Redhat 7.3. You may find some discrepancies if you are installing different versions of Snort or using different versions of Redhat.

This guide was written with the assumption that you understand how to run Snort and have a basic understanding of Linux. This includes editing files, making directories, compiling software and understanding general Unix commands. This guide does not explain how to use or configure Snort, but information on where to obtain this information can be found in the "Additional Information" section.

Required Software

The following is a list of required software and the versions that were used:

Redhat 7.3	ftp://ftp.redhat.com
Snort v1.8.7	http://www.snort.org/dl/
MySQL v3.23.52	http://www.mysql.com/downloads/mysql-3.23.html
Webmin v.99	http://www.webmin.com/
NetSSLeay v1.20	http://symlabs.com/Net_SSLeay/
ACID 0.9.6B21	http://acidlab.sourceforge.net/
PHP v4.1.*	ftp://updates.redhat.com/7.3/en/os/i386/

ADODB v2.31	http://php.weblogs.com/adodb
PHPLOT v4.4.6	http://www.phplot.com/
GD v1.8.4	http://www.boutell.com/gd/
Snortd file	http://home.earthlink.net/~sjscott007/snortd
Mozilla	http://www.mozilla.org/
Snort Webmin Module v1.1	http://msbnetworks.net/snort/

Conceptual Topology

There are five primary software packages that produce this topology. The Apache web server, MySQL database server, Webmin, ACID and Snort. This topology assumes you will be running your sensors on dedicated hardware separate from your database and ACID console. Below is a brief description of each of the packages and their purpose in the topology.

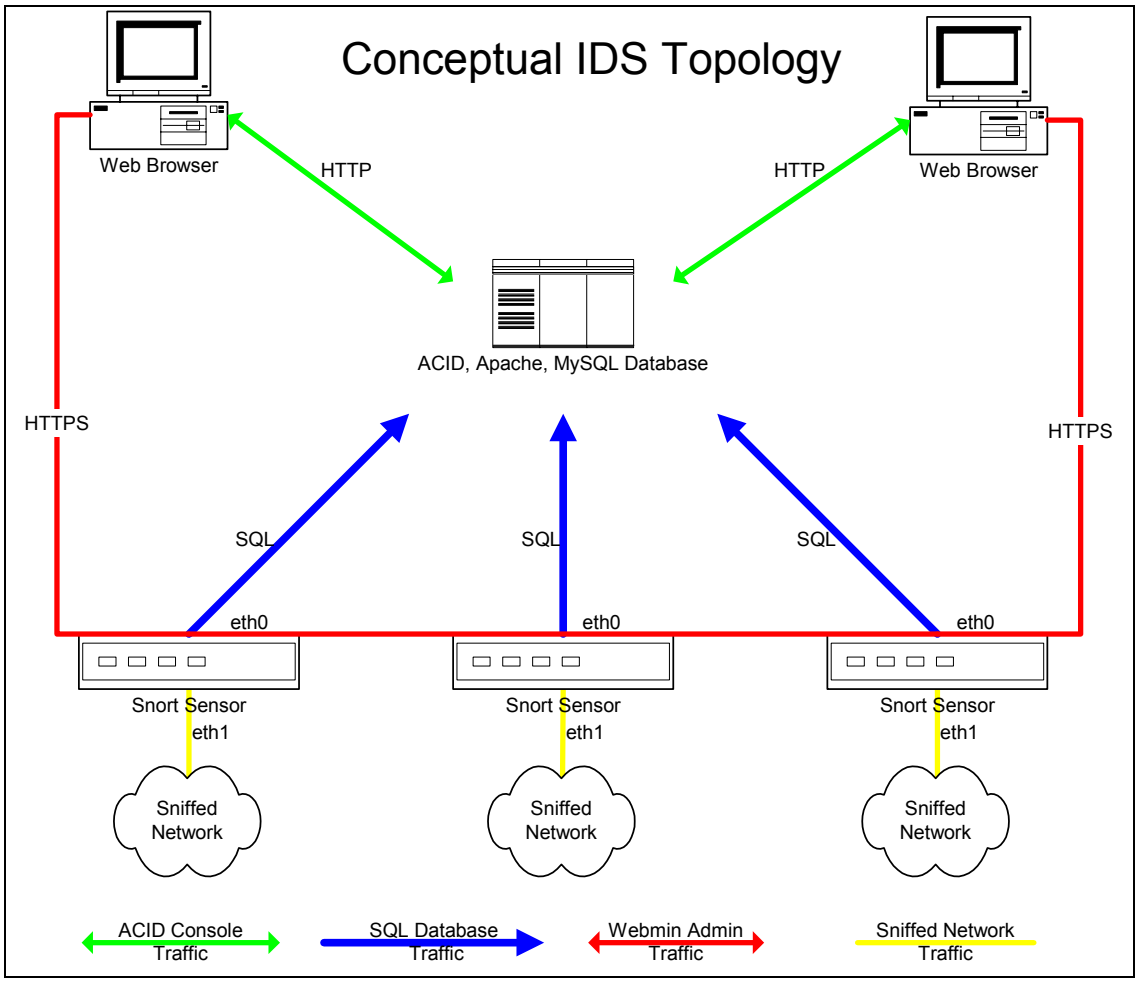
Apache Web Server
This is the web server of choice for the majority of websites that are accessed on the Internet. The sole purpose of Apache is for hosting the ACID web-based console.

MySQL Server
MySQL is a SQL based database server for a variety of platforms and is the most supported platform for storing Snort alerts. All of the IDS alerts that are triggered from our sensors are stored in the MySQL database.

Webmin
Webmin is a web-based interface for administrating Unix based servers. It provides a graphical interface to most of the services and configuration options that are available at the shell level. Webmin is written in Perl and new modules (plugins for administrating services. E.g. DNS, users & groups) are being created all the time. There is also a snort module that is installed which allows you to graphically administer Snort.

Analysis Console for Intrusion Databases (ACID)
ACID is a web-based application for viewing firewall logs and/or IDS alerts. This is where all the sensor information is consolidated for viewing.

Snort
Snort is a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. This is the software package that is used to gather information from the network.

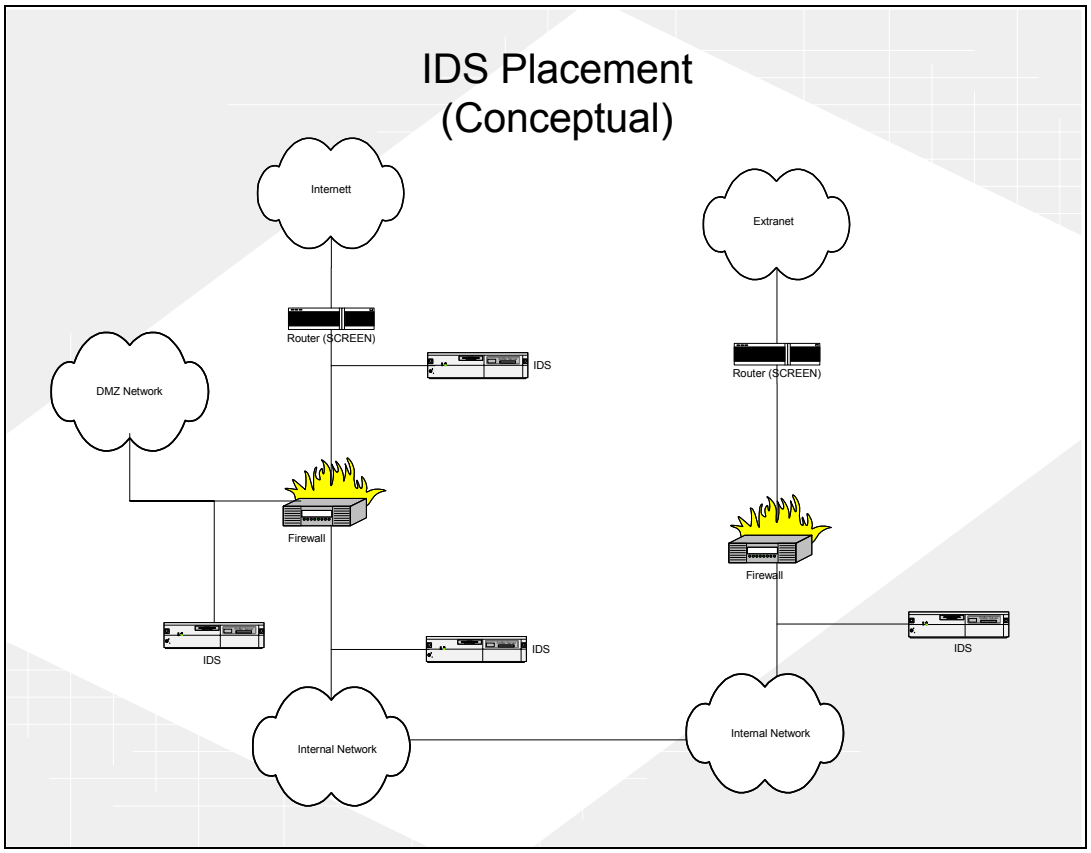


Sensor Placement Model

Internet (Public Services / Outgoing Traffic)

The most practiced and standard way of deploying your sensors is before and after a firewall. This accomplishes three goals:

- Knowing of any attempts that are being made before any packet filtering is done (Pre-firewall – External)
- Knowing that an attempt was successful or blocked by the firewall (Post-Firewall – Internal)
- Verifying the configuration of your firewalls.



It always good to know if someone is attempting to break into your network. This is why we put an Intrusion Detection System (IDS) before the first firewall (external side). You can compare this to having a camera monitoring your front door, without this camera you would never know who even attempted to pick your lock unsuccessfully.

Knowing that an attempt was successful in passing through your firewall can let you focus on real threats and help you cut down on false positives. The other benefit is in environments that use Network Address Translation (NAT). This will allow to you get the real source address by correlating the events between the IDS systems before and after the firewall.

This topology will allow you to verify that your firewall baselines are being followed, or that someone didn't make a mistake when changing a firewall rule. If you know that your firewall baselines outlaw the use of ftp and your post-firewall IDS system is showing ftp alerts, then you know that the firewall is not blocking FTP traffic. This is just a side effect and should not be the only way you verify compliance with your baselines.

Extranet

Extranet connections are monitored with one IDS system placed on the internal side of the firewall or router. The reasons we do not monitor the external side of the extranet is that the rules for this private connection should be extremely tight and access should be limited to only the resources (servers) that are needed for the business relationship.

How to use this Guide

The easiest way to use this guide is to build your MySQL and ACID server first. This can be achieved by reading the following sections in the guide: Redhat 7.3 Installation, Post Redhat Installation, ACID Console & Centralized MySQL Database.

The sensors can be created with the following sections: Redhat 7.3 Installation, Post Redhat Installation, Snort Installation, Webmin Installation.

Redhat 7.3 Installation

1. English language
2. Keyboard Configuration
    a. *Next*
3. Mouse Configuration
    a. *Next*
4. Welcome Screen
    a. *Next*
5. Install Options
    a. *Custom* → *Next*
6. Partitioning Strategy

    There are two partitioning strategies noted below. Follow the one for the Snort sensor or the one for Database / Acid Console. These configurations are based on an 18gig hard drive.

    Snort Sensor
    a. Select, "*Manually partition with Disk Druid*" → *Next*
    b. Select *New*
        i. Mount point: */boot*
        ii. Size (MB): 40
        iii. Select "*OK*"
    c. Select *New*
        i. Filesystem: *swap*
        ii. Size (MB): 512
        iii. Select "*OK*"
    d. Select *New*
        i. Mount point: */var*
        ii. Size (MB): 4000
        iii. Select "*OK*"
    e. Select *New*
        i. Mount point: */*
        ii. Check, "*Fill to maximum allowable size*"
        iii. Select "*OK*"
    f. Select Next

    MySQL Database / Acid Console
    a. Select, "*Manually partition with Disk Druid*" → *Next*
    b. Select *New*
        i. Mount point: */boot*
        ii. Size (MB): 40
        iii. Select "*OK*"
    c. Select *New*
        i. Filesystem: *swap*

   ii. Size (MB): 512
   iii. Select "*OK*"
d. Select *New*
   i. Mount point: */*
   ii. Size (MB): 4000
   iii. Select "*OK*"
e. Select *New*
   i. Mount point: */var*
   ii. Check, "*Fill to maximum allowable size*"
   iii. Select "*OK*"
f. Select Next

2. Boot Loader
   a. Next
3. Grub Password
   a. Next
4. Network Configuration
   a. Setup the IP address information for Eth0
      i. Unselect, "*Configure Using DHCP option*"
   b. Select *eth1* tab
      i. Select, "*Activate at boot*"
   c. Next

   **Note that eth0 is your internal interface and eth1 is your sniffing interface. You should never assign an IP address to the sniffing interface(eth1).
5. Firewall Configuration
   a. *No Firewall → Next*
6. Language Support
   a. Next
7. Time Zone Selection
   a. Set UTC to the proper offset
   b. Use daylight savings time option if appropriate
   c. Check the box "System clock uses UTC"
   d. Next
8. Account Configuration
   a. Set root password
   b. Create individual accounts
   c. Next
9. Authentication Configuration
   a. Next
10. Select Package Groups
    a. Select the following packages for installation:
       - ❑ Printing Support
       - ❑ Classic X Windows System
       - ❑ X Windows System
       - ❑ Gnome
       - ❑ Network Support
       - ❑ Messaging and Web Tools
       - ❑ Network Managed Workstation
       - ❑ Authoring and Publishing
       - ❑ Emacs
       - ❑ Utilities
       - ❑ Software Development

    b. Next
11. Video Configuration
    a. Select your installed video card
12. About to Install

a. Next
13. When prompted insert Redhat CD 2
14. When prompted for Boot disk creation, choose *Skip* → *Next*
15. Monitor Selection
    a. Choose the appropriate model → *Next*
16. Custom X Configuration
    a. Choose color depth and resolution
    b. Choose, "*Text*" for your login type
    c. *Next*
    d. *Exit*

Post Redhat Installation

1. Install all relevant Redhat updates and patches
    e. http://www.redhat.com/support/errata/rh72-errata.html
2. Turn off the PortMapper service
    a. *chkconfig portmap off*

Snort Installation

The first thing we need to do is install the MySQL dependences for snort. They can be downloaded from http://www.mysql.com/

```
# rpm –ivh MySQL-client-*.**.**-*.rpm
# rpm –ivh MySQL-devel-*.**.**-*.rpm
```

Next download the snort tar package from http://www.snort.org/dl. It will be called something like snort-1.8.*.tar.gz. Download the latest version and compile it.

```
# cp snort-1.8.*.tar.gz  /usr/src/redhat/SOURCES
# cd /usr/src/redhat/SOURCES
# tar –zxvf snort-1.8.*.tar.gz
# cd /usr/src/redhat/SOURCES/snort-1.8.*
# ./configure --with-mysql
# make
# make install
```

Download and install the latest rules. Download them from http://www.snort.org/dl/signatures/, Make sure you download the **snortrules.tar.gz** and **NOT** the snortrules-current.tar.gz.

```
# mkdir /etc/snort
# cp snortrules.tar.gz  /etc/snort
# cd /etc/snort
# tar –zxvf snortrule.tar.gz
```

After you have untared the rules file it will have created a rules directory under */etc/snort*. We need to move all of the rules files in to the */etc/snort* directory. The reason we have to do this is because of Webmin and the *$RULE_PATH* variable. For some reason the Webmin module for snort does not like the *$RULE_PATH* variable and hinders you from editing your rules.

```
# cd /etc/snort/rules
```

```
# mv * ../
# cd ..
# rmdir rules
# vi snort.conf
```

Edit the following lines in the *snort.conf* file. Replace the xxxx with the appropriate password for the snort account. The host variable should be set to your ACID / MySQL server IP.

#output database: log, mysql, user=root password=test dbname=db host=localhost

to

output database: log, mysql, user=snort password=snort dbname=snort host=000.000.000.000

Comment out the $RULE_PATH variable:

var RULE_PATH ../rules

to

#var RULE_PATH ../rules

Remove all the *$RULE_PATH* variables from each of the following lines. E.g. make the first rule look like this: **include bad-traffic.rules**

```
#==========================================
# Include all relevant rulesets here
#
# shellcode, policy, info, backdoor, and virus rulesets are
# disabled by default.  These require tuning and maintance.
# Please read the included specific file for more information.
#==========================================

include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/sql.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/icmp.rules
```

```
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/attack-responses.rules
# include $RULE_PATH/backdoor.rules
# include $RULE_PATH/shellcode.rules
# include $RULE_PATH/policy.rules
# include $RULE_PATH/porn.rules
# include $RULE_PATH/info.rules
# include $RULE_PATH/icmp-info.rules
# include $RULE_PATH/virus.rules
include $RULE_PATH/chat.rules
include $RULE_PATH/p2p.rules
include $RULE_PATH/multimedia.rules
include $RULE_PATH/experimental.rules
include $RULE_PATH/local.rules
```

Create the logging directory for snort. Port scan information is put here. Also, if you're doing packet logging or are not populating a database, then that information is placed here.

```
# mkdir /var/log/snort
```

Install the Snort automated startup script. You can download the script from http://home.earthlink.net/~sjscott007/snortd. If you get errors when trying to execute the file after downloading it make sure it was transfer ASCII not binary. The best way to insure this is to cut and copy into a text file.

```
# cp snortd  /etc/rc.d/init.d
# cd /etc/rc.d/init.d
# chmod 755 snortd
# chkconfig --level 2345 snortd on
```

The –u parameter records all times in UTC. The –o parameter changes the default rule order from Alert->Pass->Log to Pass->Alert->Log. This allows Snort to ignore false positives by using the local.rules file with the "pass" option for filtering noisy machines.

Lets test our snort configuration

```
# /etc/rc.d/init.d/snortd start
```

First make sure that the process is running by issuing a *ps –ef* command. Look for snort to be running. Generate some illegal traffic on the monitored segment (like an NMAP scan). Your Acid console should now display the results. You should also see the sensor count on the main ACID page increment. Note that your sensor will not be displayed in ACID until an alert is generated (but the sensor count in ACID gets incriminated).

When done testing run the following to stop Snort from running

```
# /etc/rc.d/init.d/snortd stop
```

Webmin Installation

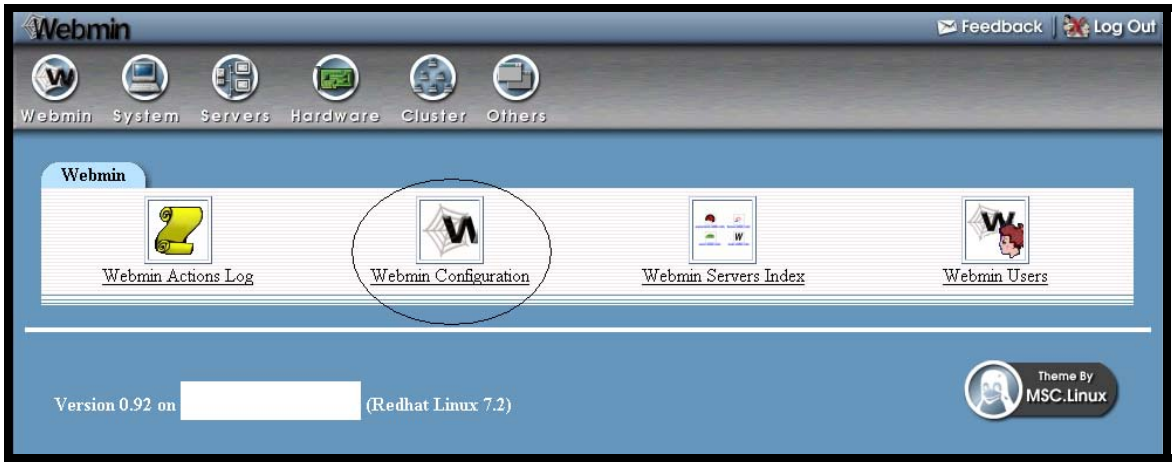
Install dependencies for using SSL connections with Webmin. You can download Net_SSLeay from http://symlabs.com/Net_SSLeay/.

```
# cp Net_SSLeayrpm-*-**.tar.gz  /usr/src/redhat/SOURCES
# cd /usr/src/redhat/SOURCES
#  tar –zxvf Net_SSLeay.rpm-*-**.tar.gz
# cd Net_*
# perl Makefile.PL
# make install
```

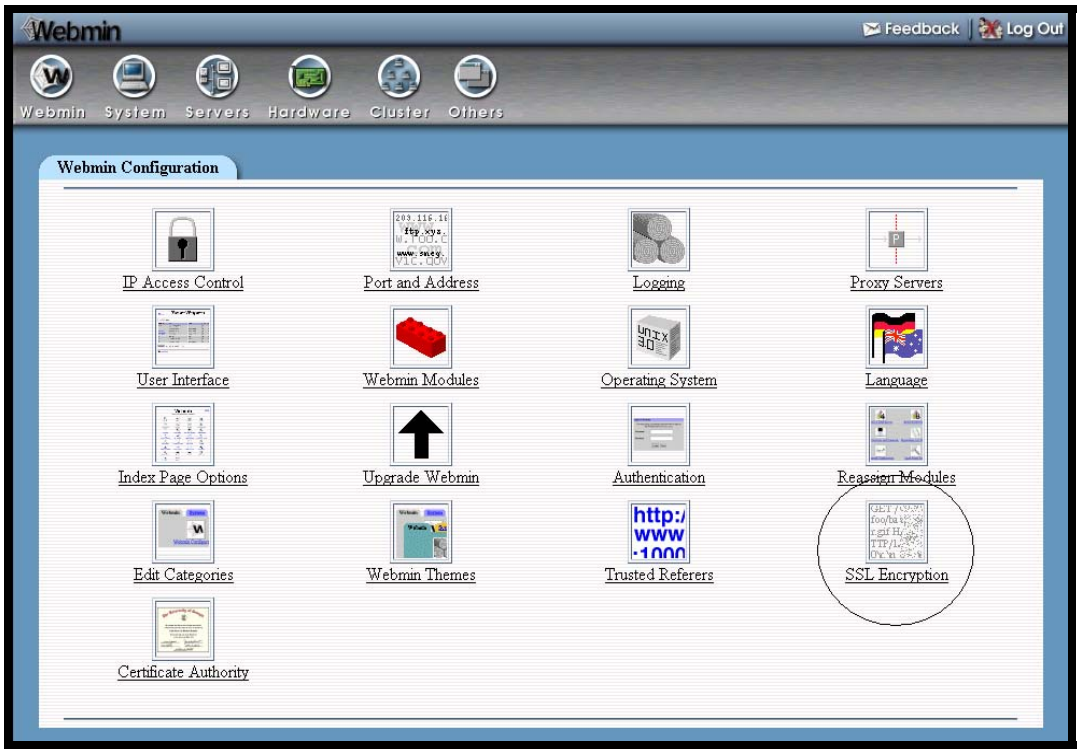
Install the Webmin RPM. Download from http://www.webmin.com/

```
# rpm –ivh webmin-0.99.-1.noarch.rpm
```

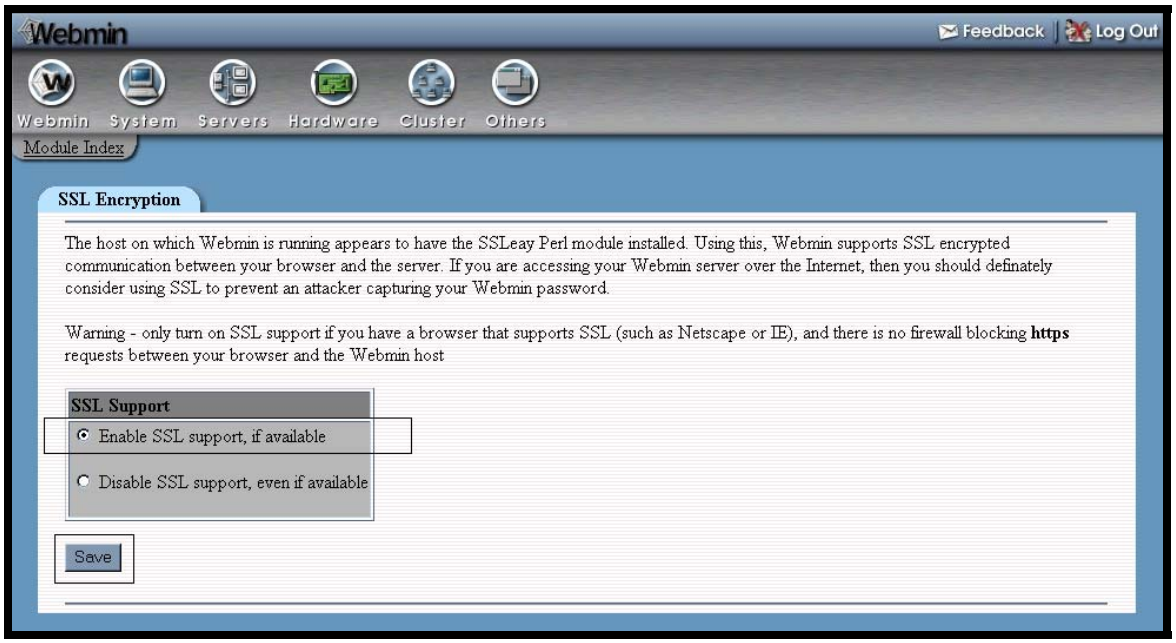
1. Configure SSL
    a. Open Mozilla browser and go to address: http://127.0.0.1:10000
    b. Login as ROOT
    c. Select, "*Webmin Configuration*" icon



    d. Select, "*SSL Encryption*" icon

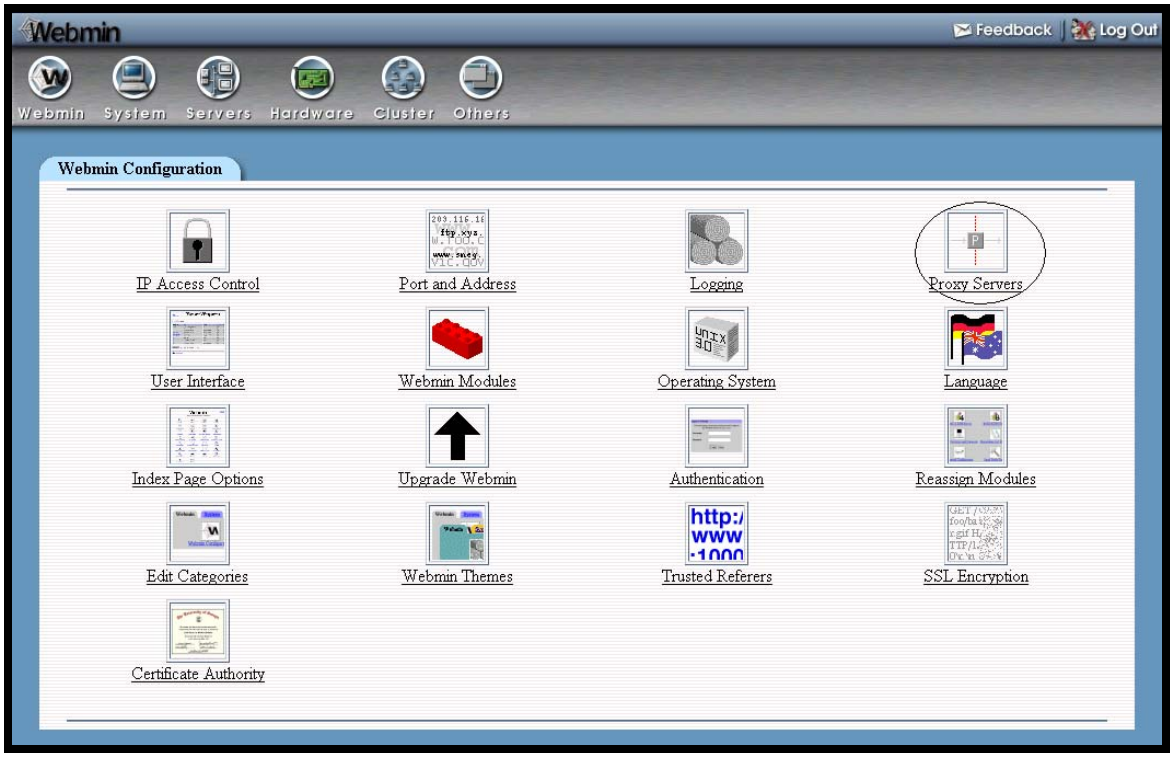


e. Select, "*Enable SSL support if available*" and click the "Save" button

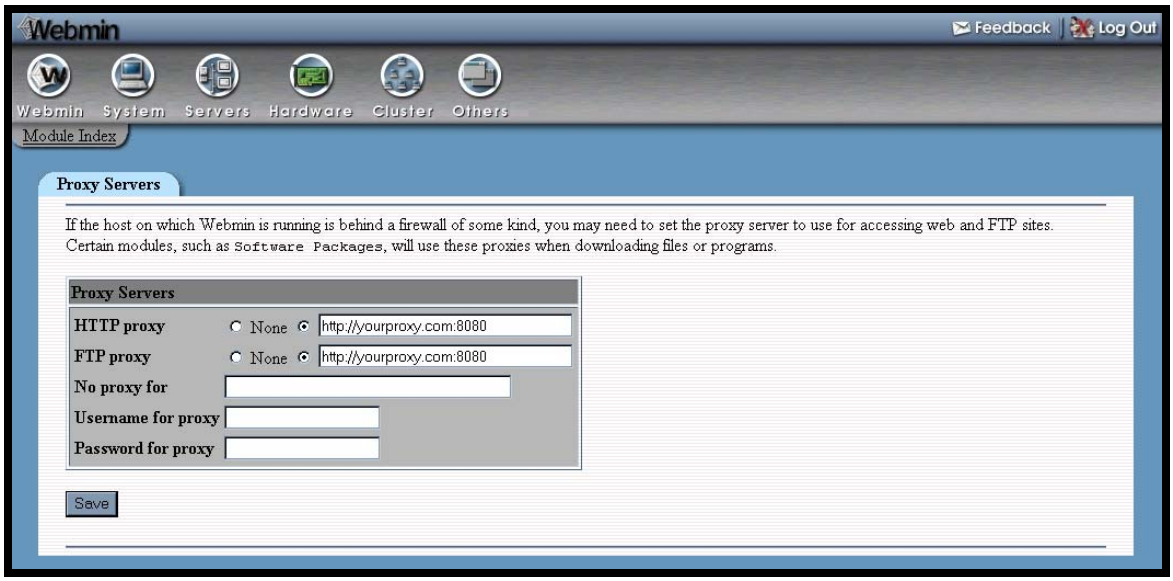


2. Configure Proxy if you are behind a firewall
   a. Select, "*Webmin Configuration*" icon

b. Select, "*Proxy Servers*" icon

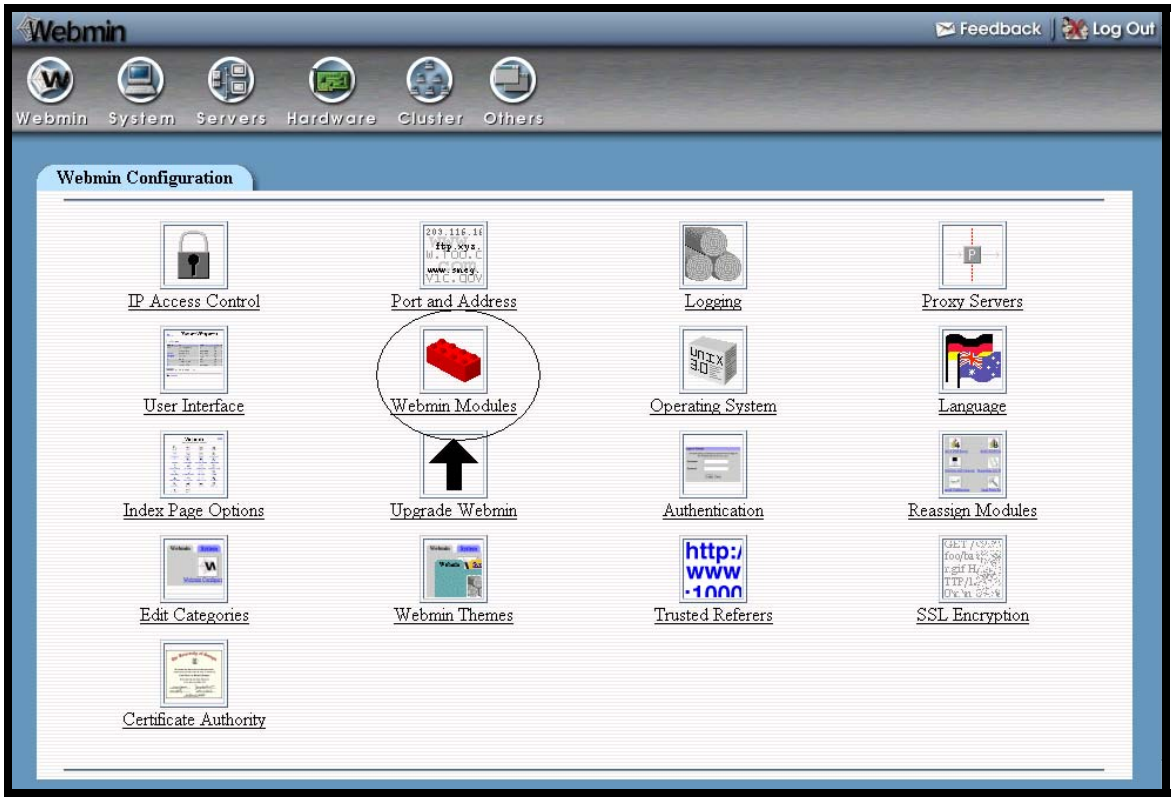


c. Enter your proxy information and click the "SAVE" button

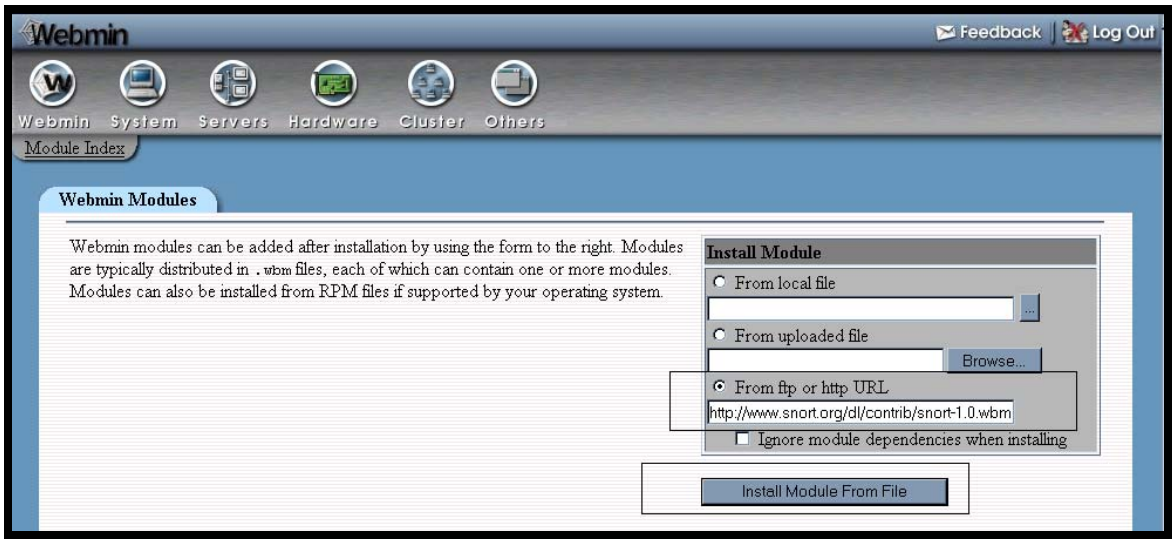


3. Install Snort Webmin plugin
   a. Select, "*Webmin Configuration*" icon

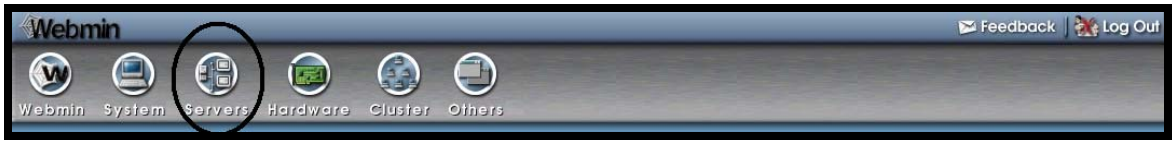
b. Select, “Webmin Modules” icon



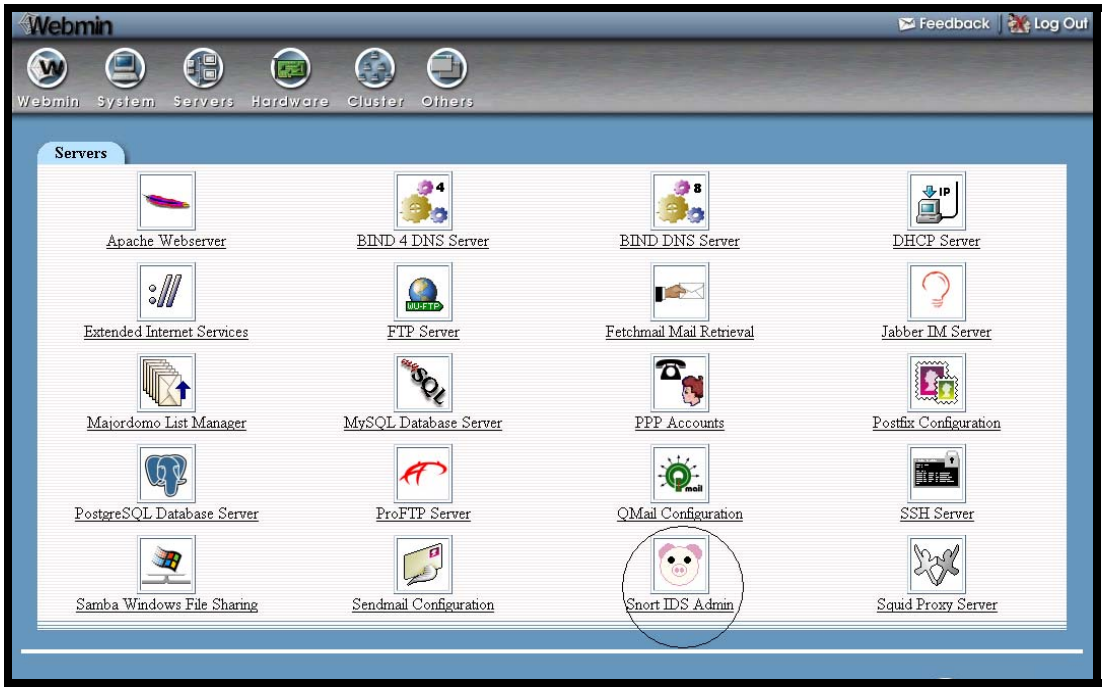
c. Install module from url: http://www.snort.org/dl/contrib/front_ends/webmin_plugin/snort-1.0.wbm and click “Install”



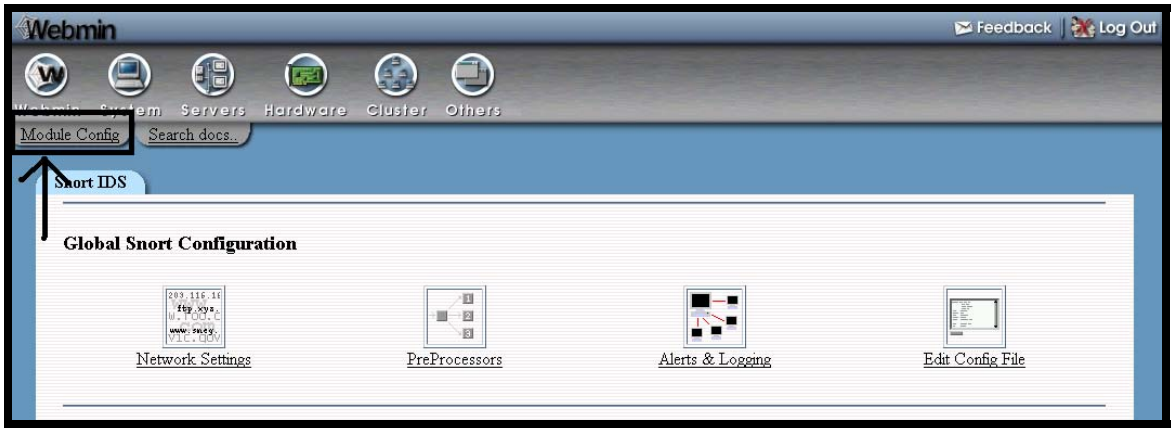
4. Configure Snort Plugin
    a. Select, “Servers” icon from the TOP of the web page.



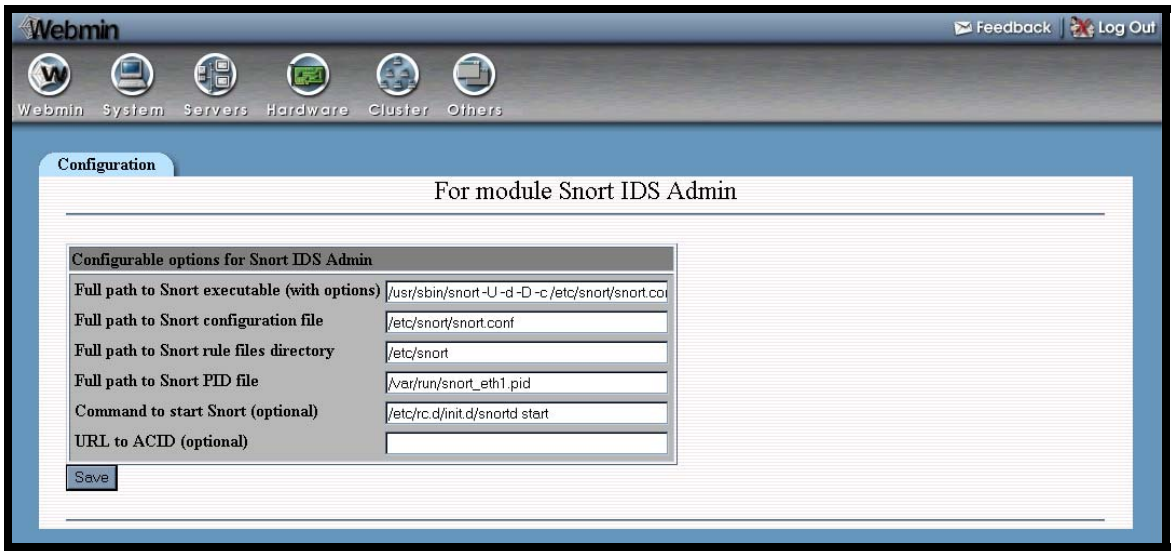
    b. Select, “Snort IDS Admin” icon (Looks like a pig!)



    c. Select the “Module Config” tab in the left hand corner.



You should now see a screen like this:



Your configuration should match the following:

Full path to Snort executable (with options) =	/usr/local/bin/snort -U -d -D -c /etc/snort/snort.conf
Full path to Snort configuration file =	/etc/snort/snort.conf
Full path to Snort rule files directory =	/etc/snort
Full path to Snort PID file =	/var/run/snort_eth1.pid
Command to start Snort (optional) =	/etc/rc.d/init.d/snortd start

When finished click the "Save" button. You're done!

Acid Console & Centralized MySQL Database

The first thing we need to do is install the Apache web server so that ACID has a home. The latest RPM for Apache can be found at ftp://updates.redhat.com/7.3/en/os/i386/

```
# rpm -ivh apache-1.3.X-X.i386.rpm
# chkconfig --level 2345 httpd on
# /etc/rc.d/init.d/httpd start
```

Next we install and configure the MySQL database. Download it from http://www.mysql.com/.

```
# rpm -ivh MySQL-3.23.X-X.i386.rpm
# rpm -ivh MySQL-client-3.23.X-X.i386.rpm
# rpm -ivh MySQL-shared--3.23.X-X.i386.rpm
# mysql -u root
mysql> set password for 'root'@'localhost' = password('yourpassword');
mysql> create database snort;
```

```
mysql>exit
```

NOTE: For some odd reason the MySQL-3.23.56.i386.rpm doesn't start the mysql service on run level 3. Do the following to correct the problem.

```
# chkconfig –level 3 mysql on
```

Note that after you set the root password above you need to login using a password to access the database with root. E.g. # mysql –u root -p

The database tables need to be set up. We accomplish this by running the *create_mysql* script. This can be found in the CVS tree at http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/snort/snort/contrib/.

If the file is not located in the directory from which the *mysql* program was run from, add the path to the source statement. E.g. **mysql> source /home/john/create_mysql**

```
# mysql –u root -p
mysql> connect snort
mysql>source create_mysql
mysql>grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to snort;
```

So you can connect locally with this account

```
mysql>grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to snort@localhost;
```

Creates a user that cannot delete alerts from database: may only need the local account

```
mysql> grant CREATE, INSERT, SELECT, UPDATE on snort.* to acidviewer;
```

So you can connect locally with this account

```
mysql>grant CREATE, INSERT, SELECT,  UPDATE on snort.* to acidviewer@localhost;
```

Set the passwords for the MySQL accounts.

```
mysql>connect mysql
mysql> set password for 'snort'@'localhost' = password('yourpassword');
mysql> set password for 'snort'@'%' = password('yourpassword');
mysql> set password for 'acidviewer'@'localhost' = password('yourpassword');
mysql> set password for 'acidviewer'@'%' = password('yourpassword');
mysql> flush privileges;
mysql> exit
```

Acid requires the installation of PHP and the supporting Mysql module. Download from ftp://updates.redhat.com/7.3/en/os/i386/.

```
# rpm -ivh php-4.1.*-*.i386.rpm
# rpm -ivh php-mysql-4.1.*-*.i386.rpm
```

Now its time to install ACID. You can download all the files from:

ACID 0.9.6B21	http://acidlab.sourceforge.net/
ADODB v2.31	http://php.weblogs.com/adodb

PHPLOT v4.4.6	http://www.phplot.com/
GD v1.8.4	http://www.boutell.com/gd/

Once there files have been downloaded untar the following files to ***/var/www/html***.

```
# tar -zxvf acid-0.9.*.tar.gz -C /var/www/html
# tar -zxvf adodb231.tgz -C /var/www/html
# tar -zxvf gd-1.8.4.tar.gz -C /var/www/html
# tar -zxvf phplot-4.4.6.tar.gz -C /var/www/html
```

*** Important: Remove the version number from the directory names (e.g. **mv gd-1.8.4 to gd and mv phplot-4.4.6 phplot**)

Lets configure the ACID configuration file:

```
# cd /var/www/html/acid
# vi acid_conf.php
```

Once you're in the *acid_conf.php* file modify the following variables. Change the *xxxx* to reflect the password you've chosen for the *snort* account.

```
$DBlib_path="../adodb";
$alert_dbname="snort";
$alert_user="snort";
$alert_password="xxxx";
$Chartlib_path="../phplot";
```

Next we want to setup the view only ACID portal (NO deleting of events). This is good for people who only need to view alerts. Copy the /var/www/html/acid to /var/www/html/acidviewer (view only acid)

```
# cp –R /var/www/html/acid /var/www/html/acidviewer
# cd /var/www/html/acidviewer
# vi acid_conf.php
```

Change the following variables in /var/html/www/acidviewer/acid_conf.php. Again, Change the *xxxx* to reflect the password you've chosen for the *acidviewer* account.

```
$alert_user="acidviewer";
$alert_password="xxxx";
```

Now we secure both of the ACID websites with Apache. Setup the two accounts for accessing the ACID website. When prompted enter your password for that web account. Be careful not to include the *–c* option in the third line!

```
# mkdir /usr/lib/apache/passwords
# htpasswd -c /usr/lib/apache/passwords/passwords admin
# htpasswd  /usr/lib/apache/passwords/passwords acidviewer
```

Add the following lines to /etc/httpd/conf/httpd.conf in the DIRECTORY section. Section means the general area when you see the other Directory formats.

```
<Directory "/var/www/html/acid">
        AuthType Basic
        AuthName "yourcompany"
        AuthUserFile /usr/lib/apache/passwords/passwords
        Require user admin
        AllowOverride None
</Directory>

<Directory "/var/www/html/acidviewer">
        AuthType Basic
        AuthName "yourcompany"
        AuthUserFile /usr/lib/apache/passwords/passwords
        Require user acidviewer
        AllowOverride None
</Directory>
```

Reboot the server.

```
# reboot
```

Accessing the ACID Console

You now have two websites for the ACID console:

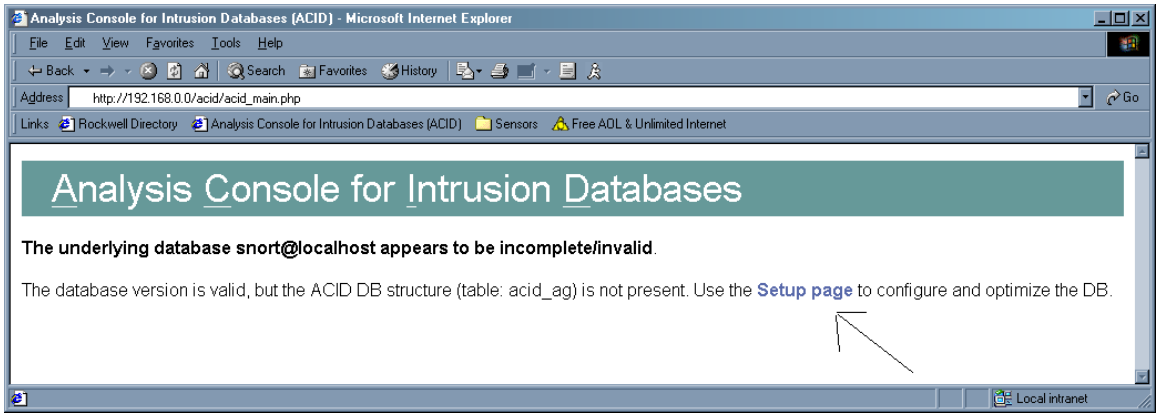
1) http://youracidhost/acid/index.html

This site is for the administrator and can be access using the ADMIN account you created earlier. You can delete events using this site.

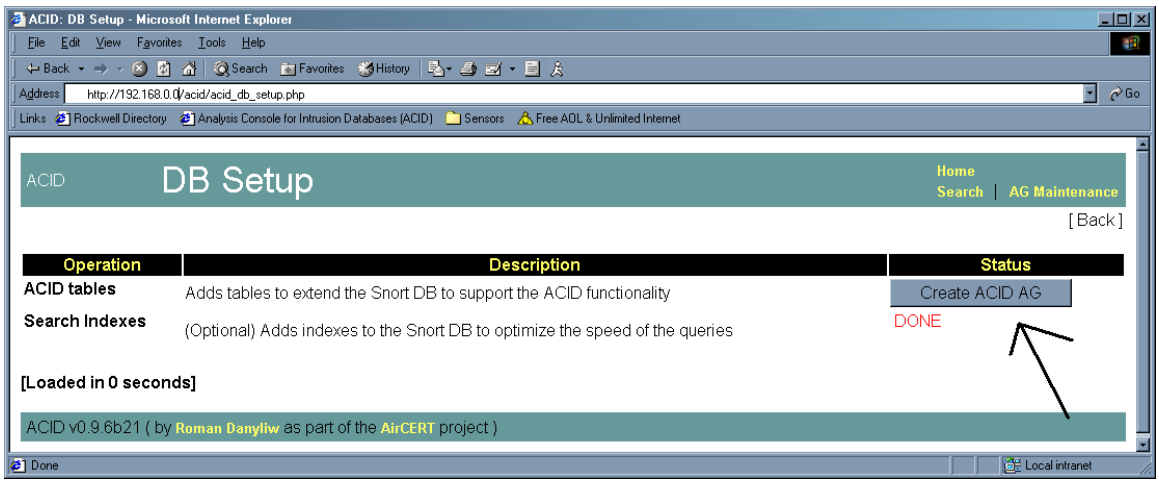
http://youracidhost/acidviewer/index.html

This site is for anyone who requires read access to the events and can be access using the ACIDVIEWER account you created earlier. Users of this site cannot delete events

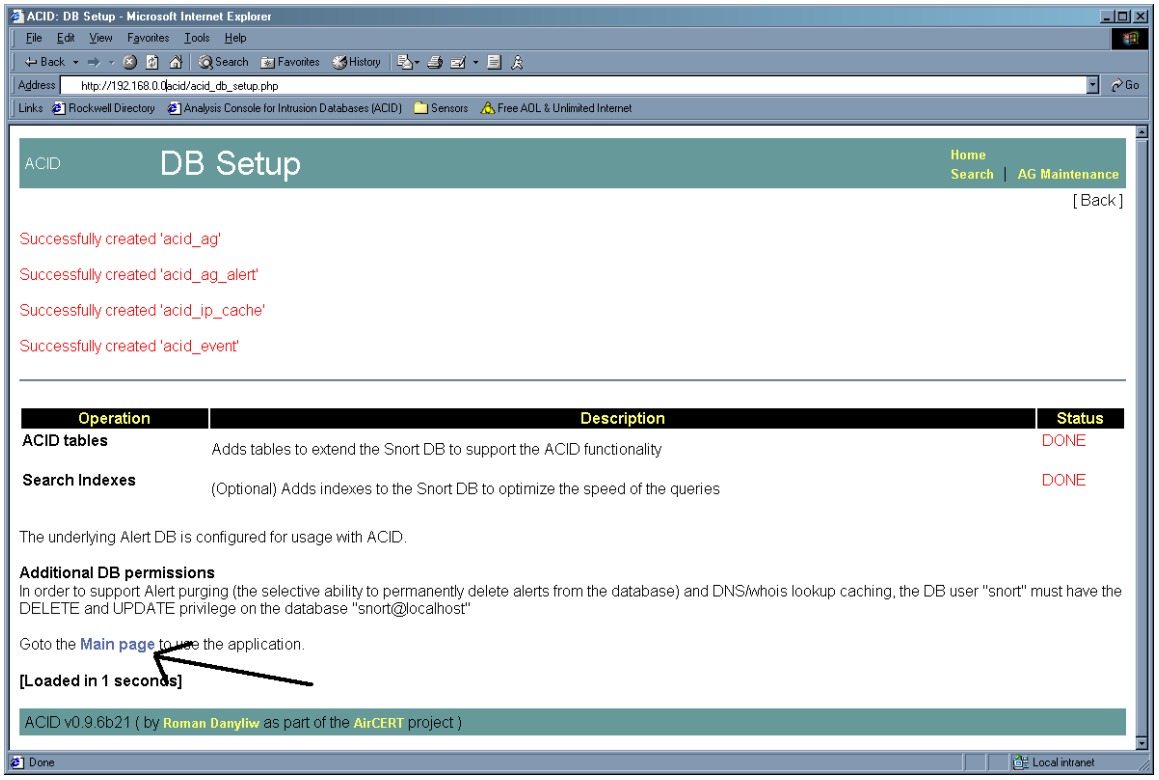
The first time you connect to the ACID website you will see a display like this. Click <setup page>.



Once your on the setup page click "Create ACID AG".



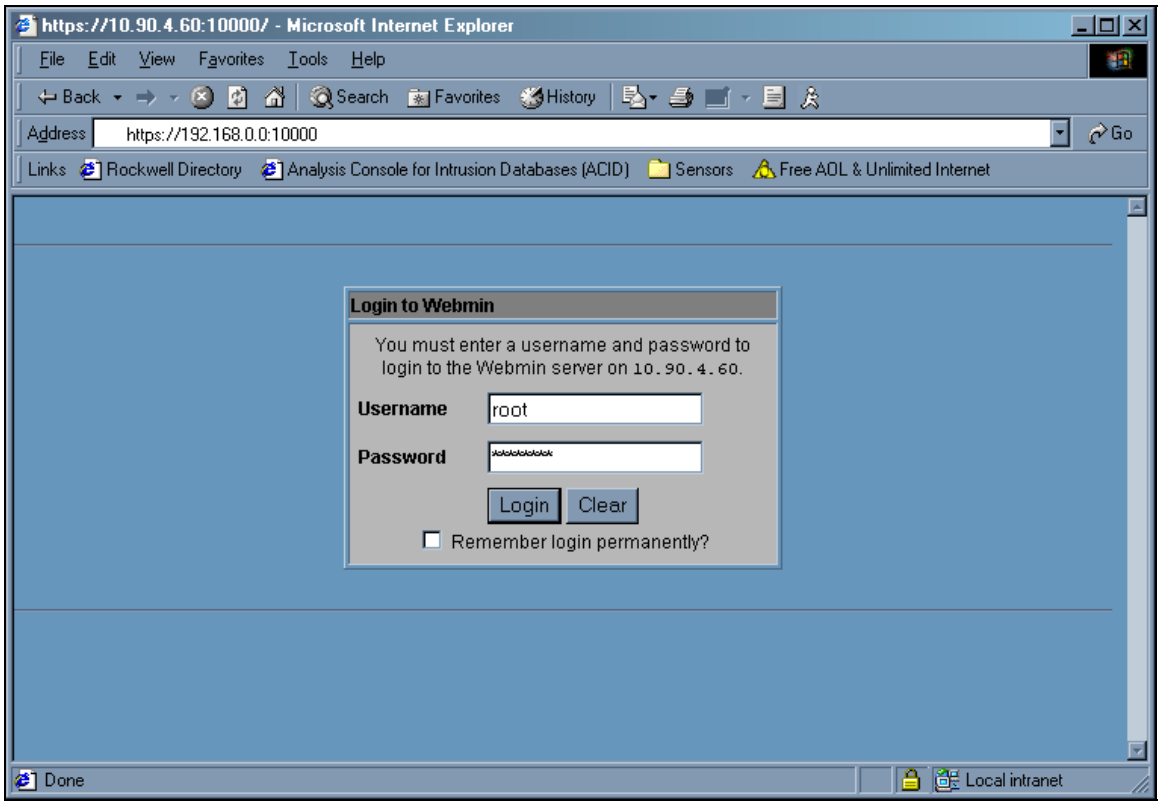
Once it completes click <Main Page> and your done!



Sensor Tuning

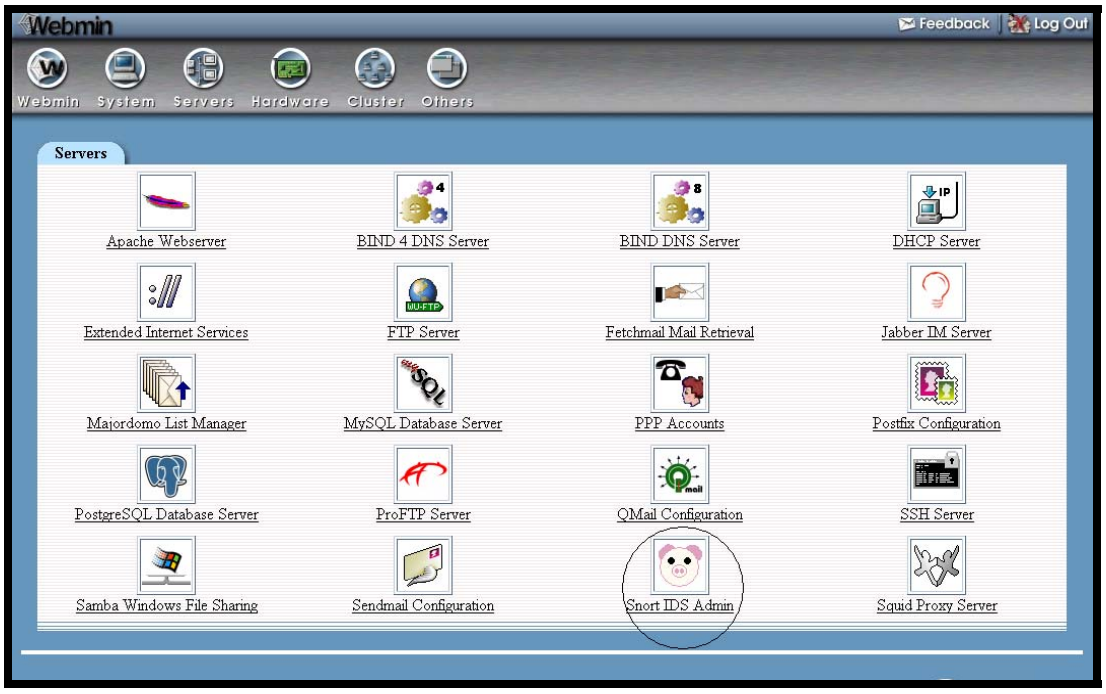
Using Webmin and the snort plugin you can easily tune your sensors. The following will demonstrate one way of managing your rules using Webmin.

The first thing we need to do is login to one of our sensors, which can be accessed via https://yoursensor:10000. Login using you root password as show below.



You will then be presented with a screen like the one below. Select the Sever Icon from the top of the screen.

Next select the Snort icon (P.S. it looks like a pig).



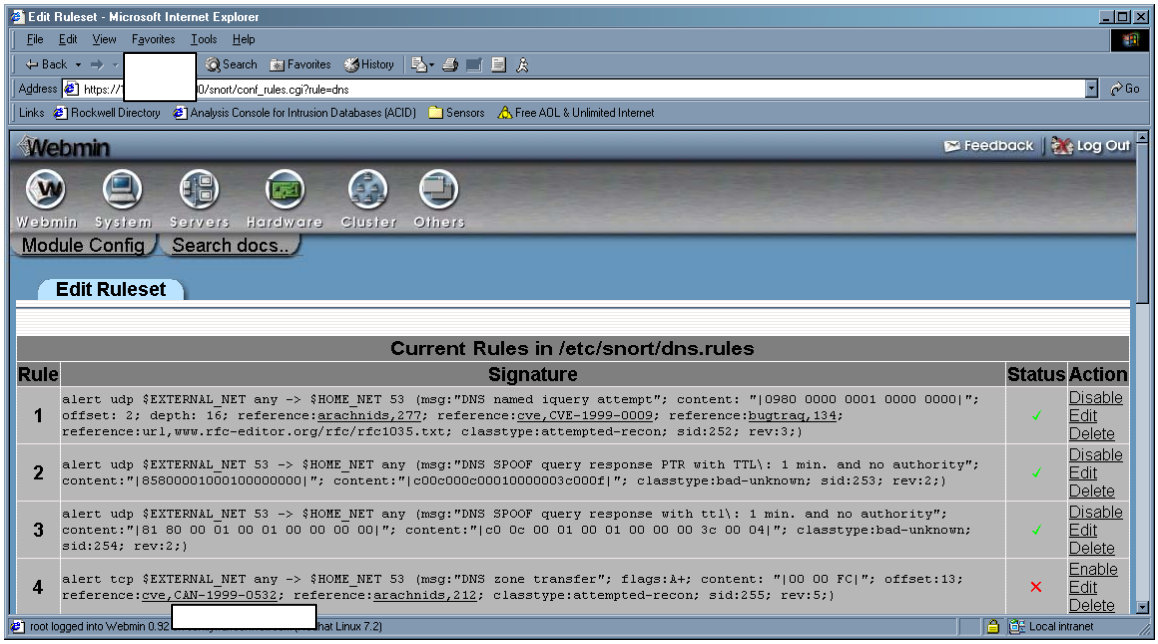
You will now be presented with the screen that allows you control most aspects of your sensor. In the center of screen you will see all your rule files. Note yours may look different. Pay special attention to the local rules. This is where we put our filters in.

Rulesets

✓ = Enabled ✗ = Disabled

Rule Set	Status	Action	Rule Set	Status	Action	Rule Set	Status	Action
$RULE_PATH/backdoor	✗	Enable	dos	✓	Disable	smtp	✓	Disable
$RULE_PATH/experimental	✗	Enable	exploit	✓	Disable	sql	✓	Disable
$RULE_PATH/icmp-info	✗	Enable	finger	✓	Disable	telnet	✓	Disable
$RULE_PATH/info	✗	Enable	ftp	✓	Disable	tftp	✓	Disable
$RULE_PATH/policy	✗	Enable	icmp	✓	Disable	web-attacks	✓	Disable
$RULE_PATH/porn	✗	Enable	local	✓	Disable	web-cgi	✓	Disable
$RULE_PATH/shellcode	✗	Enable	misc	✓	Disable	web-coldfusion	✓	Disable
$RULE_PATH/virus	✗	Enable	netbios	✗	Enable	web-frontpage	✓	Disable
attack-responses	✓	Disable	rpc	✓	Disable	web-iis	✓	Disable
bad-traffic	✓	Disable	rservices	✓	Disable	web-misc	✓	Disable
ddos	✓	Disable	scan	✓	Disable	x11	✓	Disable
dns	✓	Disable						

Lets take a look at the DNS rules file. Simply click on it and you see a screen like this.



As you can see there are four columns that make up the rule file.

Rule : Just the order in which the rule the rule appears in the rule file.

Signature: This is what an actual snort signature looks like.

Status: Is the rule enabled or disabled?

Action: These are the actions that you can perform on that given rule.

It should be apparent that you can enable, disable, change, and add rules from this screen. Remember after you make changes that you need to restart the snort daemon for the changes to take effect. You can find the button to restart the service on the main Snort plugin page at the bottom of the screen.

Filtering Rules

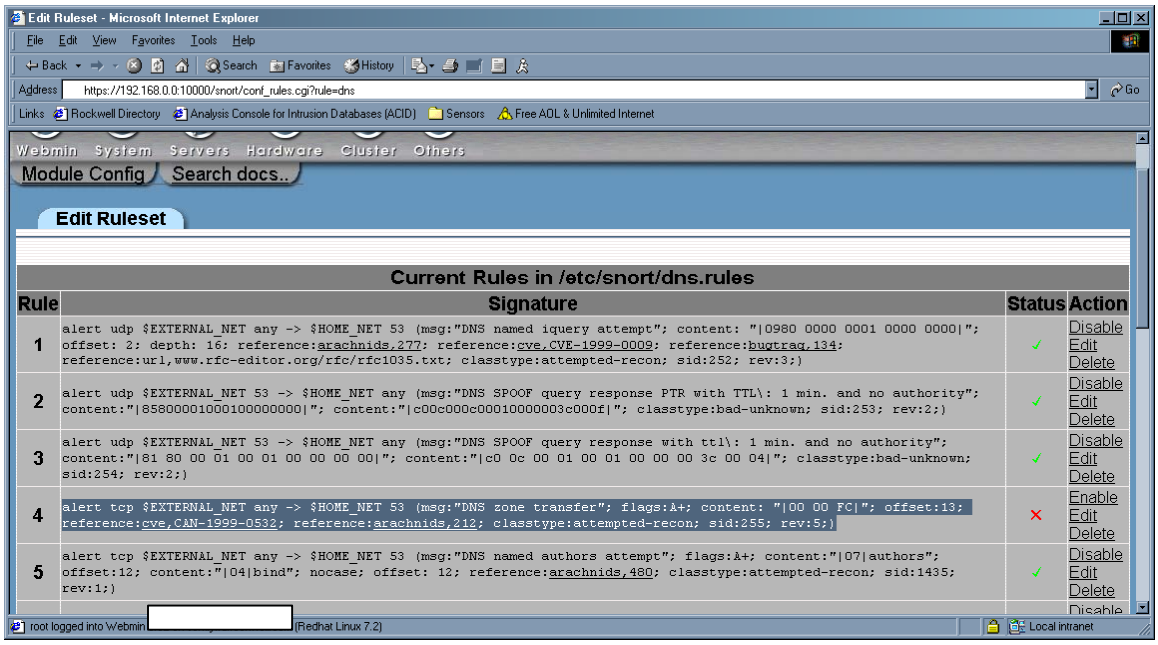
Filtering enables us to make exceptions to the rules without completely disabling the rule. As you progress with your IDS systems you find that some signatures are rather noisy and require tuning. Filtering is one way of accomplishing this.

For this example we are going to take rule number #4 from the above example. This rule is used to detect DNS zone transfers. There are many cases where this is legal and we don't want to be alerted on it when it is performed from expected hosts. Here's what Rule #4 looks like.

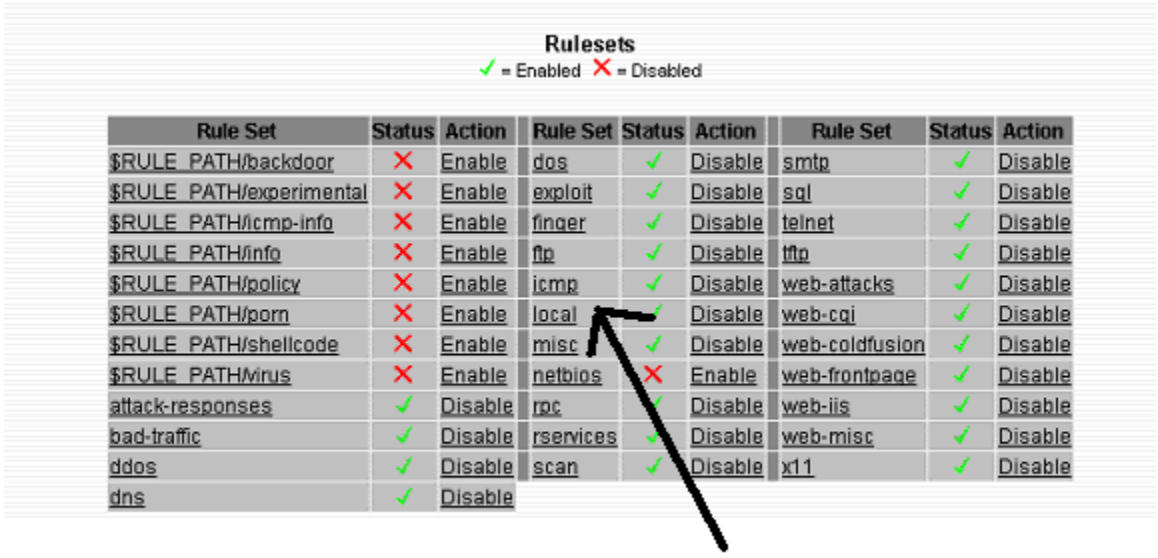
RULE #4: `alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS zone transfer"; flags:A+; content: "|00 00 FC|"; offset:13; reference:cve,CAN-1999-0532; reference:arachnids,212; classtype:attempted-recon; sid:255; rev:5;)`

Lets say on this sensor that it is normal for host 192.168.55.23 to perform DNS zone transfers with 192.168.12.5.

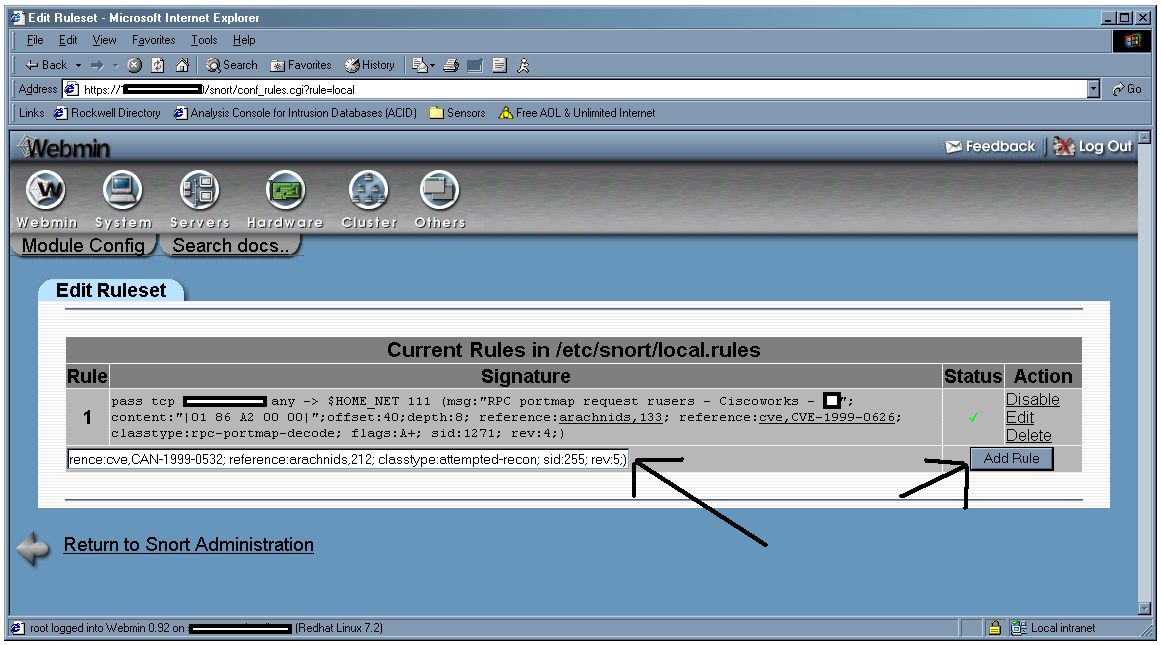
Highlight the rule and shown below and copy it.



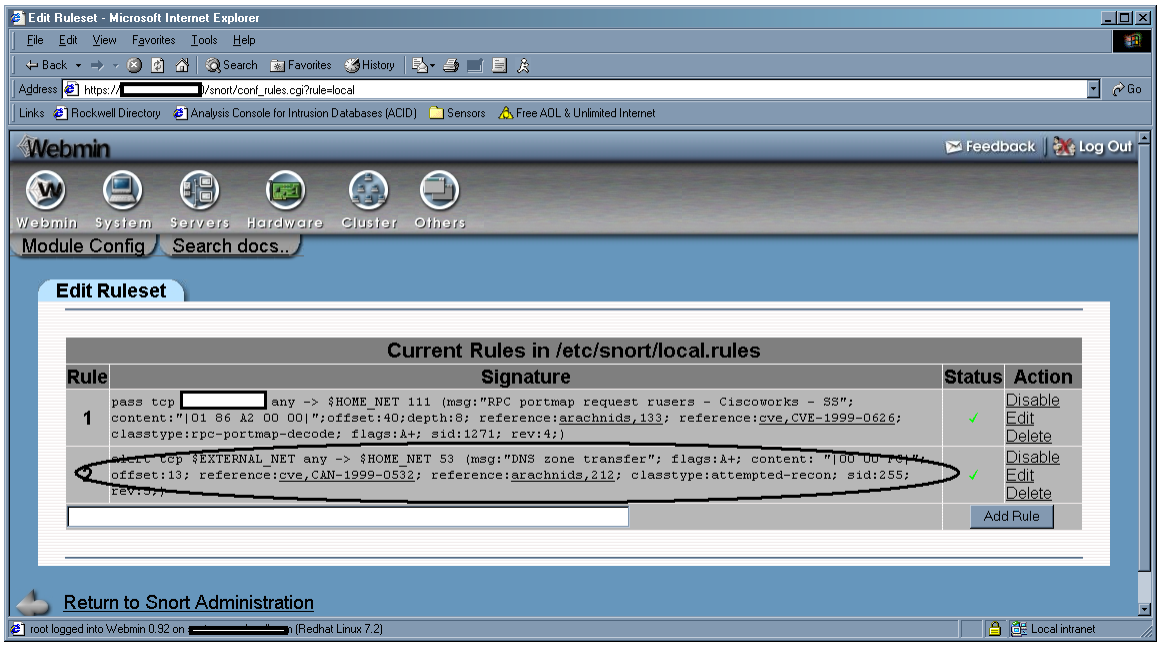
Then select the back button and go back to the main snort plugin screen. Click on the local rules file. The local rules file is used for your own rules. You can use this file for your own signatures and for filtering.



Once your in the local rules file paste the rule you copied into the add rule box at the bottom of the screen.



Then click add and your rule should appear like below.

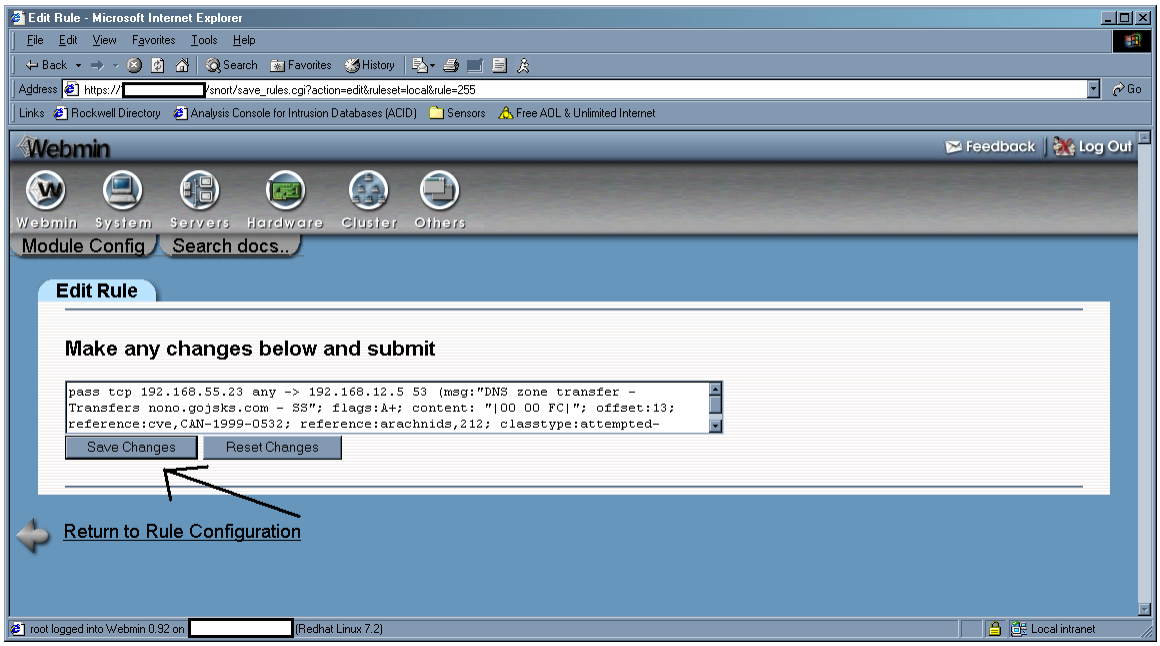


Now select edit on the rule. We are going to customize the rule for filtering out are expected DNS zone transfers. In this case we are going to modify the rules action, source, destination and message field.

1) The action field is going to be <pass>.
2) The source is 192.168.55.23
3) The destination is 192.168.12.5

4) The message field describes the signature. I keep the signature description, But I add a comment to describe why we're filtering this event and I add my initials to show who created it.

See below.



Now click save. As you can see your new rule now appears in the file. Now just restart snort from the main snort plugin page and your filter takes effects.

The reason that this works is because of the snort startup option –o.

Time Zones

You many be deploying your sensors in different time zones. So it is very important to set the time correctly. Therefore, we need to set the proper time zone and make sure all time is recorded in the UTC standard (formally Greenwich Mean Time).

The easiest way to accomplish this is to set the hardware clock (BIOS) to UTC. This can be accomplished during the Redhat install or after the installation is completed. A good tutorial on setting the time can be found at http://www.linuxsa.org.au/tips/time.html. The following is how to set time after the installation has been completed.

The actual time zone files are stored in the */usr/share/zoneinfo* directory. To select a time zone, copy the appropriate file to the */etc* directory and name it *localtime*. I don't know why Redhat doesn't use a symbolic link here.

For central time:

```
# cp /usr/share/zoneinfo/America/Chicago /etc/localtime
```

or

```
# ln -sf /usr/share/zoneinfo/America/Chicago /etc/localtime
```

Edit the */etc/sysconfig/clock* file and change *UTC* variable equal to true.

```
UTC=true
```

Now set the system clock. The example given is for March 25, 2002 at 12:30pm CST. Time is set in 24 hour mode using **your local time** (not UTC time). See man page for more information: *man date*

```
# date 032512302002
```

Set the hardware clock to the system clock.

```
# hwclock --systohc --utc
```

Network Time Protocol (NTP)

There is a need to keep accurate time on the sensors without having to manually set the clocks. The easiest way to keep your sensors in sync is using the Network Time Protocol (NTP).

Edit the */etc/ntp.conf* file. Change the server entry to reflect you timeserver and comment out the entry starting with fudge. See below.

```
# is never used for synchronization, unless no other other
# synchronization source is available. In case the local host is
# controlled by some external source, such as an external oscillator or
# another protocol, the prefer keyword would cause the local host to
# disregard all other synchronization sources, unless the kernel
# modifications are in use and declare an unsynchronized condition.
#
server      yourtimeserver.com
#fudge      127.127.1.0 stratum 10
```

Next start the *ntpd* daemon and make it run at startup.

```
# /etc/rc.d/init.d/ntpd start
# chkconfig ntpd on
```

Maintenance

Using the Redhat Network

If you are setting up your servers for the first time you need to register it first. Issue the following command and follow the prompts.

```
# rhn_register
```

There are two scenarios where packages will not be automatically upgraded. The first is kernel upgrades and the second is RPM's that modify configuration files. Make sure you know what packages your updating before making the following changes.

Once registered login into https://rhn.redhat.com/ and establish the entitlement for your new server. Then launch an upgrade from the Redhat Network.

Kernel upgrades

Run the following command:

```
# export display=
# up2date –nox –configure
```

Edit line 23 or 24 depending on which version of up2date you are using. The line should contain the variable <pkgSkipList>. Clear this variable out by type the line number and then type a CAPITAL 'C' to clear the entry.

Press enter to exit up2date.

Run the following command to download the kernel upgrades:

```
# rhn_check
```

After it completes reboot the machine. When the machine comes back up run the following command to verify the success of the upgrade. In the event that machine does not come back from the reboot, you will have to manual select the old kernel from the grub boot screen.

After a successful kernel upgrade, we can now cleanup the old kernel. Edit the *grub.conf* file in the */etc* directory.

```
# vi /etc/grub.conf
```

Remove the last 4 lines of the file that refer to the old kernel version.

Next we need to clean up all the files that reference the old kernel. These are located in the */boot* directory. Delete the following files that match the old kernel version numbers. The files I list have have '*' representing the old version numbers.

```
# rm initrd-*.*.*-*.img
# rm module-info-*.*.*-*
# rm system.map-*.*.*-*
#rm vmlinux-*.*.*-*
```

Run the following command:

```
# up2date –nox –configure
```

Edit line 23 or 24 depending on which version of up2date you are using. The line should contain the variable <pkgSkipList>. Change the able out by typing the line number and then type a 'kernel*'. This stops the kernel from being automatically upgraded.

Press enter to exit. That's it!

RPM's that modify configuration files

Run the following command:

```
# export DISPLAY=
# up2date –nox --configure
```

Edit line 19. The line should contain the variable <noReplaceConfig>. Change the viable from 'Yes' to 'No'.

Press enter to exit up2date.

Proceed with update by running the following command:

```
# rhn_check
```

Once complete go back in to the up2date configuration screen:

```
# up2date –nox –configure
```

Edit 19 again and change the value back to 'Yes'.

Press enter to exit.

That's it!

Synchronizing your Redhat Profile

If you manually update RPM's or some how get out of sync with the Redhat Network you will need to upload your profile again. Run the following command to get back in sync:

```
# export DISPLAY=
# up2date -p
```

Manually update your Redhat packages (without the redhat network)

The best way to update your Redhat servers that are in remote locations is to SSH in and run the following commands:

```
# export DISPLAY=
# up2date --nox -u
```

You should now see the command line version of up2date running. Once the up2date exits all your rpm's have been updated.

How to completely remove a sensor from the MySQL database

Go into ACID and delete all the events associate with that sensor. This may take a while depending on the number of events to be deleted and the type of hardware your running the database on. Be patient, your browser may even time out while waiting for it to finish. Use top to watch the mysqld service. When I was testing on a slow box, I had to go in multiple times and keep deleting the events. I had upwards of 60000 events and multiple sensors. I also had to keep exiting the sensor screen and then re entering it to make the deletes work because It kept giving me an "unsuccessful delete".

Next remove the sensor completely from the database. This will correct the sensor count on the main ACID web page.

```
# mysql -u root -p
mysql> connect snort
mysql> select * from sensor;
```

Look for the sid number of sensor you wish to delete. eg.. mysql> delete from sensor where sid=2;

```
mysql> delete from sensor where sid=<number>;
```

Sensor Characteristics

The purpose of having sensor characteristics is to document and understand the traffic that transverses the link where the sensor is located. You can use this information to cut down on your false positives, tune your sensors, and eventually find anomalies in the traffic. Below is the format to use when populating the fields.

Fields	Description
Sensor	DNS Name of your sensor
IP	IP address of the management interface
Mask	Subnet mask for the above IP
GW	Default Gateway for the above IP
Network Placement	Internet / Pre-Firewall / (External) Internet / Post-Firewall / (Internal) Extranet / Post-Firewall / (Internal)
Source Address Category	External Internet Address Internal Address Extranet Address Proxy Firewall
Destination Address Category	External Internet Address Internal Address Extranet Address Proxy Firewall
Relationship to other sensors	This field is used to show relations between sensors. For example, a sensor before and after a proxy. If you see an alert on the IDS system after the proxy and want the real address of source, you will need reference the sensor before the proxy.
Comments	Comments regarding any special circumstances
Contact	Information on who to contact
Allowed Protocol Flow	This should contain all the allowed protocols that cross the link.
Public Servers	Any servers that are accessible to the public

Example Template

Sensor: Coco23	**IP:** 127.2.44.2	**Mask:** 255.255.255.0	**GW:** 127.2.44.1
Network Placement: Internet / Pre-Firewall / (External)		**Source Address Category:** External Internet Address	
Destination Address Category: Proxy (10.77.3.4)			
Relationship to other sensors: Momo44 – To find the real destination address correlate events with Momo44 sensor.			
Contact:			
Comments:			

Allowable Protocols			
Source Address	**Direction (→ or ←)**	**Destination**	**Protocol**
Any	→	10.77.3.4	FTP
Any	←	10.77.0.0/16	HTTP

Public Servers		
Source Address	**Running Services**	**Contact**
10.77.3.4	FTP	Jimmy John (444)-555-1111

Additional Information

Snort Home Page	http://www.snort.org/
Snort FAQ	http://www.snort.org/docs/faq.html
Snort Users Manual	http://www.snort.org/docs/writing_rules/
Snort-Setup for Statistics	http://www.linuxdoc.org/HOWTO/Snort-Statistics-HOWTO/
Man Page	http://www.dpo.uab.edu/~andrewb/snort/manpage.html
Usenet Groups	
Snort-announce	http://lists.sourceforge.net/mailman/listinfo/snort-announce
Snort-users	http://lists.sourceforge.net/mailman/listinfo/snort-users
Snort-sigs	http://lists.sourceforge.net/mailman/listinfo/snort-sigs
Snort-devel	http://lists.sourceforge.net/mailman/listinfo/snort-devel
Snort-cvsinfo	http://lists.sourceforge.net/mailman/listinfo/snort-cvsinfo
Snort CVS tree	http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/snort/snort/
ACID Home Page	http://acidlab.sourceforge.net/
MySQL Home Page	http://www.mysql.com/
Webmin Home Page	http://www.webmin.com/
Redhat Home Page	http://www.redhat.com/
Redhat 7.2 Reference Books	http://www.redhat.com/support/resources/howto/rhl73.html
Redhat 7.2 Updates / Patches	http://www.redhat.com/support/errata/rh73-errata.html
Redhat Network Guide	https://rhn.redhat.com/help/basic/
Compaq Linux	http://www.compaq.com/products/software/linux/
Nessus Vulnerability Scanner	http://www.nessus.org/
Linux, Clocks, and Time	http://www.linuxsa.org.au/tips/time.html

Change Log

V1.0 May, 2002
Initial document

V1.5 August 2002
Redone for Redhat 7.3
Error Corrections
Sensor tuning section was added
Changlog section was added
Accessing the ACID Console section was added