Snort Enterprise Implementation

Snort, MySQL, SnortCenter and ACID on Redhat 7.3

October, 2002

Version 2.0

Prepared by Steven J. Scott
sjscott007@yahoo.com
http://www.superhac.com/snort

Table of Contents

ACKNOWLEDGMENTS **4**

COMMENTS & CORRECTIONS **4**

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

INTRODUCTION **4**

REQUIRED SOFTWARE **5**

CONCEPTUAL TOPOLOGY **5**

SENSOR PLACEMENT MODEL **6**

HOW TO USE THIS GUIDE **8**

REDHAT 7.3 INSTALLATION **8**

POST REDHAT INSTALLATION **10**

APACHE INSTALLATION **10**

MYSQL DATABASE INSTALLATION **11**

ACID CONSOLE INSTALLATION **12**

SNORTCENTER CONSOLE INSTALLATION **13**

ACCESSING THE ACID CONSOLE **14**

ACCESSING THE SNORTCENTER CONSOLE **15**

SNORT SENSOR INSTALLATION **18**

SNORTCENTER AGENT INSTALLATION **18**

ADDING SENSORS TO THE SNORTCENTER CONSOLE **19**

USING SNORTCENTER 21

FILTERING EVENTS WITH SNORTCENTER 22

TIME ZONES 26

NETWORK TIME PROTOCOL (NTP) 27

MAINTENANCE 27

SENSOR CHARACTERISTICS 30

ADDITIONAL INFORMATION 32

APPENDIX A – IMPORTANT FILES, DIRECTORY'S AND COMMANDS 33

APPENDIX B – PHYSICAL IDS PLACEMENT DRAWING 34

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
Bob was the original tester of this document. He used the document to roll out the many sensors we have in production today.

Stefan Dens
Stefan is the author of SnortCenter, which lets security guys like me manage multiple sensors with minimal effort. He has also given me a lot of insight on how his software works and answered the many questions that I had. This software will definitely expedite the acceptance of Snort in enterprise environments. Great work Stefan!!!

T. Brian Granier
Brian took the time to explain how to make the document more functional, and more intuitive for the reader. He also wrote the "How to Use this Guide" section. Thanks Brian!

I would also like to thank the following beta testers: John Hall and Richard N. Smith.

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://www.superhac.com/snort.

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 1.9 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 v1.9.0	http://www.snort.org/dl/
create_mysql	http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/snort/snort/contrib/.
MySQL v3.23.52	http://www.mysql.com/downloads/mysql-3.23.html
MySQL-client-3.23.X-X.i386.rpm	
MySQL-shared--3.23.X-X.i386.rpm	
MySQL-devel-*.**.**-*.rpm	
ACID 0.9.6B22	http://acidlab.sourceforge.net/
PHP v4.1.*	ftp://updates.redhat.com/7.3/en/os/i386/
php-mysql-4.1.*-*	ftp://updates.redhat.com/7.3/en/os/i386/
ADODB v2.42	http://php.weblogs.com/adodb
JPgraph v1.9.1	http://www.aditus.nu/jpgraph/jpdownload.php
GD v1.8.4	http://www.boutell.com/gd/
SnortCenter v0.9.6-RC2	http://users.pandora.be/larc/download/
Snoertcenter-v0.9.*	
Snortcenter-agent-v0.9.6*	
NetSSLeay v1.20	http://symlabs.com/Net_SSLeay/
Apache 1.3.x	ftp://updates.redhat.com/7.3/en/os/i386/

Conceptual Topology

There are five primary software packages that produce this topology. The Apache web server, MySQL database server, SnortCenter, 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.

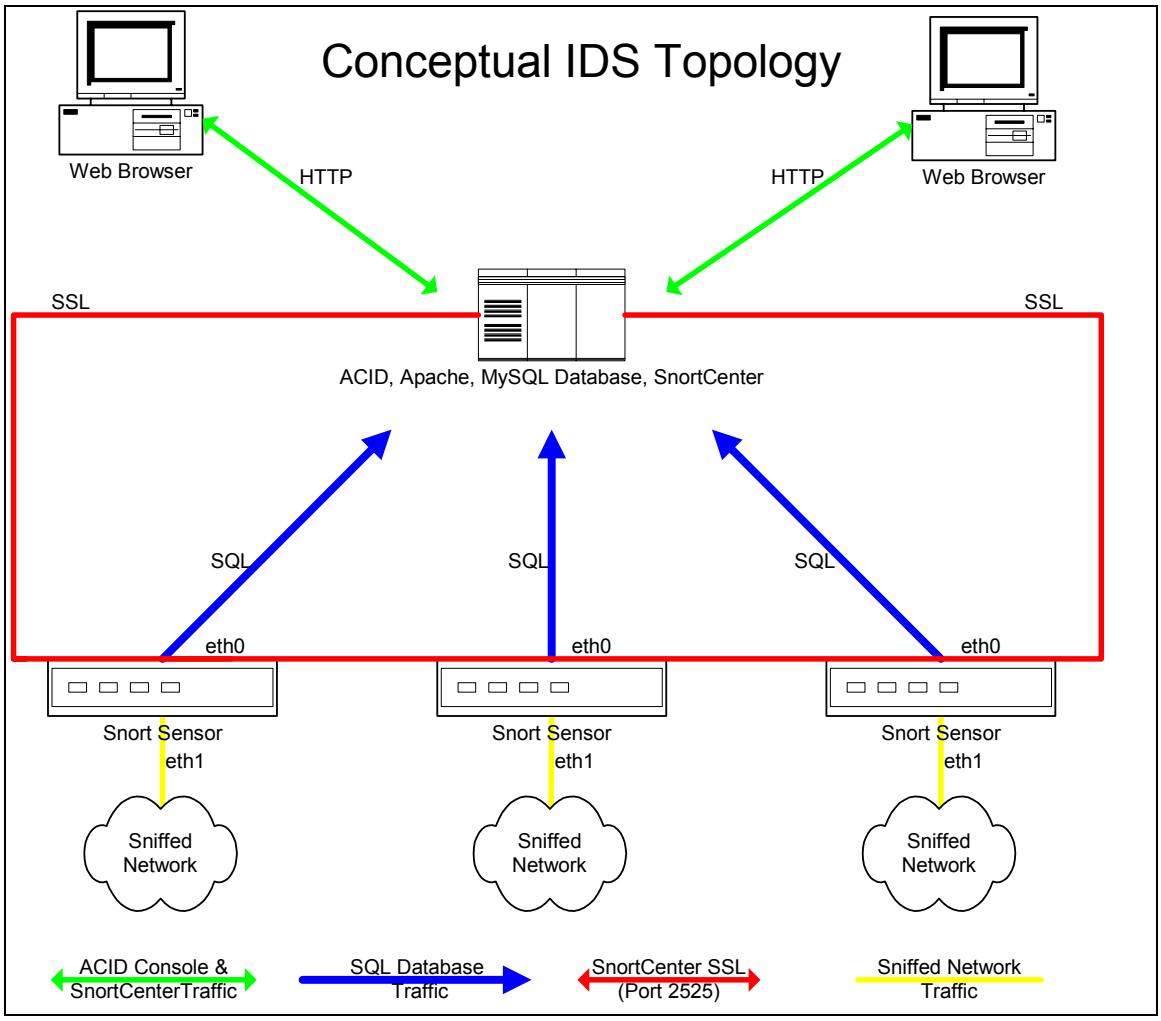
Apache Web Server
This is the web server of choice for the majority of websites that are accessed on the Internet. The sole purpose of Apache is for hosting the ACID web-based console.

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.

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.

SnortCenter
SnortCenter is a package for centrally managing your signatures and snort configuration files. The console is web-based with agents installed on each sensors communicating via SSL. This eliminates the need to update each sensor directly and track signature changes.

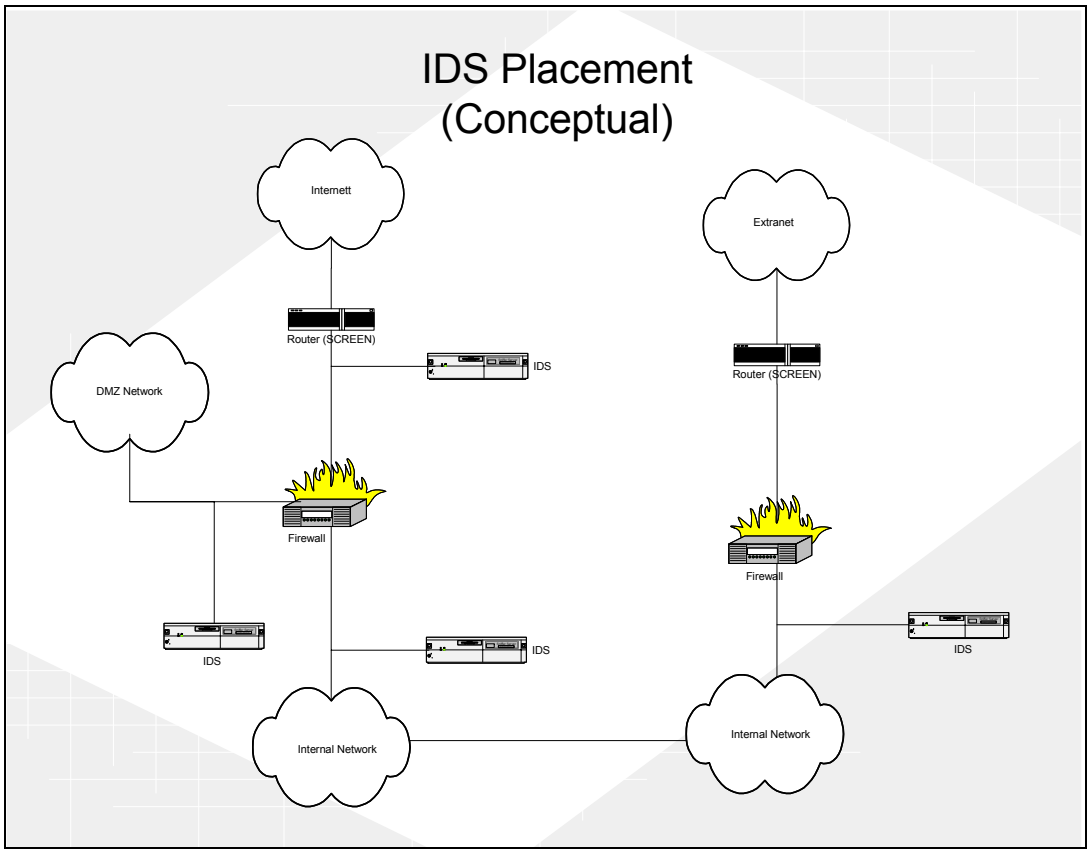


Sensor Placement Model

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 firewall(s).



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/SnortCenter/ACID server, then build your sensor, and then complete your SnortCenter configuration. When this is done your installation is functionally complete. After you are comfortable with this setup, it is recommended to further your understanding of the implementation and to proceed with maintenance and cleanup of your setup. I recommend the following approach:

Phase I - MySQL/SnortCenter/ACID server

- ❑ Redhat 7.3 Installation
- ❑ Post Redhat Installation
- ❑ Apache Installation
- ❑ MySQL Database Installation
- ❑ ACID Console Installation
- ❑ SnortCenter Console Installation

Phase II - Snort sensor(s) installation

- ❑ Redhat 7.3 Installation
- ❑ Post Redhat Installation
- ❑ Snort Sensor Installation
- ❑ SnortCenter Agent Installation

Phase III - SnortCenter completion

- ❑ Add sensors to the SnortCenter Console
- ❑ Accessing the ACID Console
- ❑ Accessing the SnortCenter Console

Phase IV - Learn SnortCenter

- ❑ Read "Using SnortCenter"

Phase V - Maintenance and cleanup

- ❑ Setup Time Synchronization
- ❑ Maintenance - Redhat Network

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

7. Boot Loader
   a. Next
8. Grub Password
   a. Next
9. 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).
10. Firewall Configuration
    a. *No Firewall* → *Next*
11. Language Support
    a. Next
12. 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
13. Account Configuration
   a. Set root password
   b. Create individual accounts
   c. Next
14. Authentication Configuration
   a. Next
15. 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
16. Video Configuration
   a. Select your installed video card
17. About to Install
   a. Next
18. When prompted insert Redhat CD 2
19. When prompted for Boot disk creation, choose *Skip* → *Next*
20. Monitor Selection
   a. Choose the appropriate model → *Next*
21. 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
   a. http://www.redhat.com/support/errata/rh73-errata.html (refer to the maintenance section)
2. Turn off the PortMapper service
   a. # chkconfig portmap off

Apache Installation

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
```

MySQL Database Installation

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
```

Acid Console Installation

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

ACID 0.9.6B22	http://acidlab.sourceforge.net/
ADODB v2.42	http://php.weblogs.com/adodb
JPgraph v1.9.1	http://www.aditus.nu/jpgraph/jpdownload.php
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 adodb242.tgz -C /var/www/html
# tar -zxvf gd-1.8.4.tar.gz -C /var/www/html
# tar -zxvf jpgraph-1.9.1.tar.gz
# cd jpgraph-1.9.1/src
# mkdir /var/www/html/jpgraph
# cp –R * /var/www/html/jpgraph
```

Important: Remove the version number from the directory names

```
# cd /var/www/html
# mv gd-1.8.4 gd
# 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="../jpgraph";
```

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>
```

Restart the httpd service.

```
# service httpd restart
```

SnortCenter Console Installation

Download and install the SnortCenter console. You can find it at http://users.pandora.be/larc/download/.

```
# tar -zxvf  snortcenter-v0.9.6*.tar.gz
# cd www
```

```
# mkdir /var/www/html/snortcenter
# cp –R * /var/www/html/snortcenter
# cd /var/www/html/snortcenter
# vi config.php
```

Edit the following lines in config.php. The $DB_password should be the root password on the database and the $hidden_key_num should just be a random number (its used in the authentication system to encrypt a value in the cookie).

```
$DBlib_path = "../adodb"
$DB_user = "root"
$DB_password="XXXX"
$hidden_key_num = "XXXXXXX"
```

Now we need to create the SnortCenter database:

```
# mysql –u root –p
mysql> CREATE DATABASE snortcenter;
mysql> exit
```

Accessing the ACID Console

You now have two websites for the ACID console:

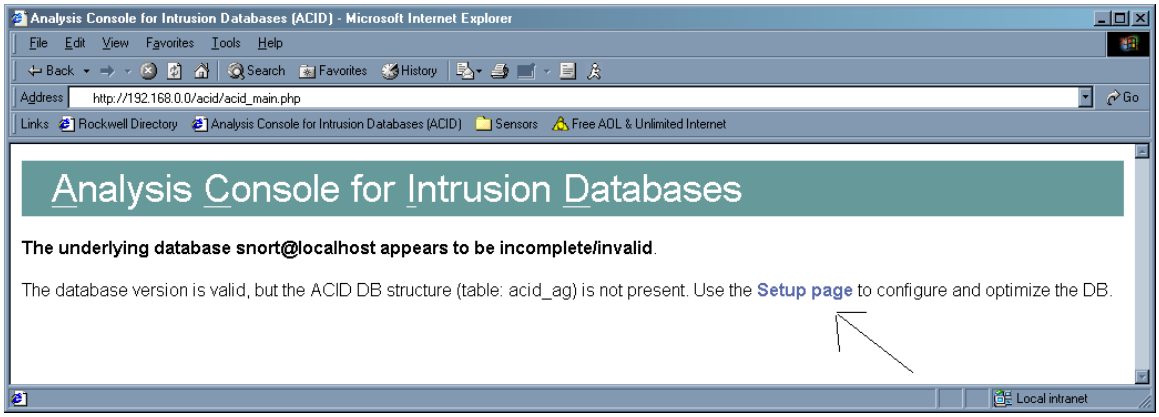
1) http://<youracidhost>/acid/index.html

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

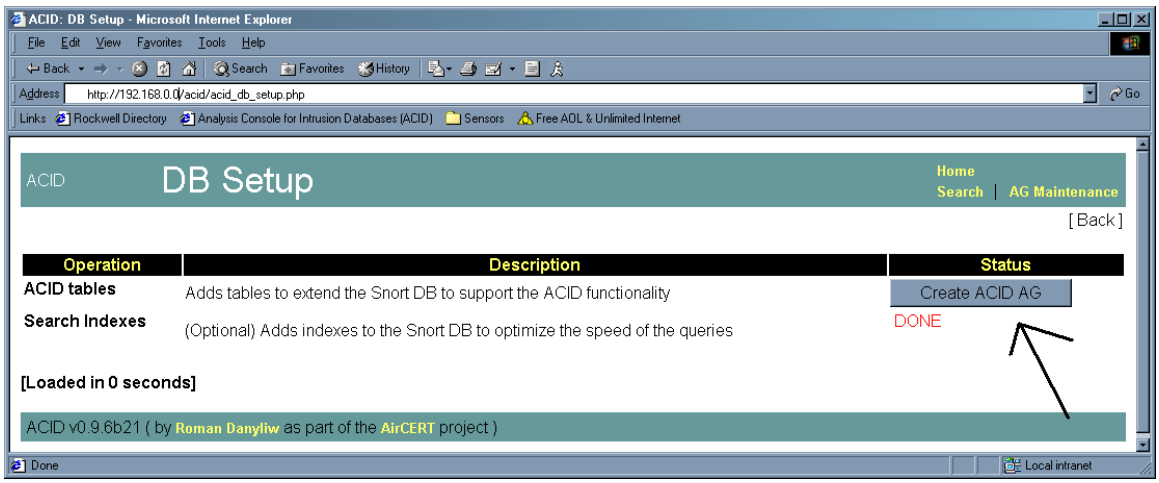
2) http://<youracidhost>/acidviewer/index.html

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

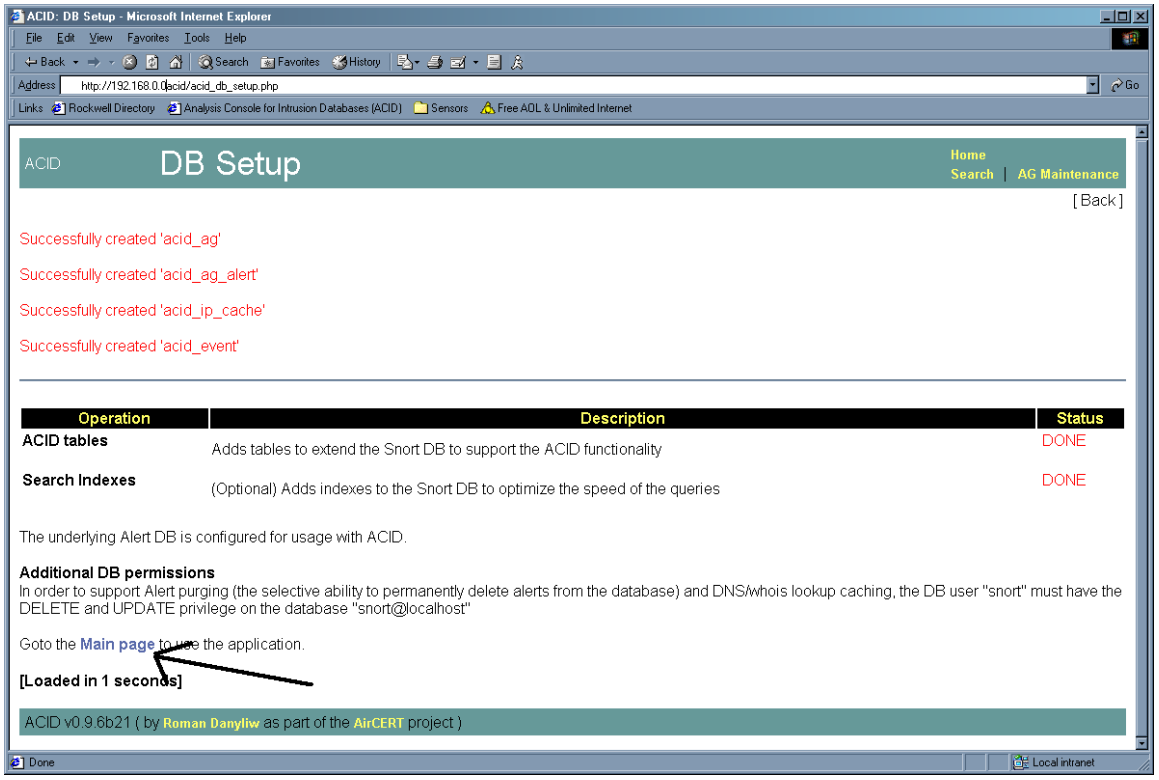
The first time you connect to the ACID website you will see a display like this. Click <setup page>.



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



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



Accessing the SnortCenter Console

You can access the SnortCenter console at

http://youracidhost/snortcenter/

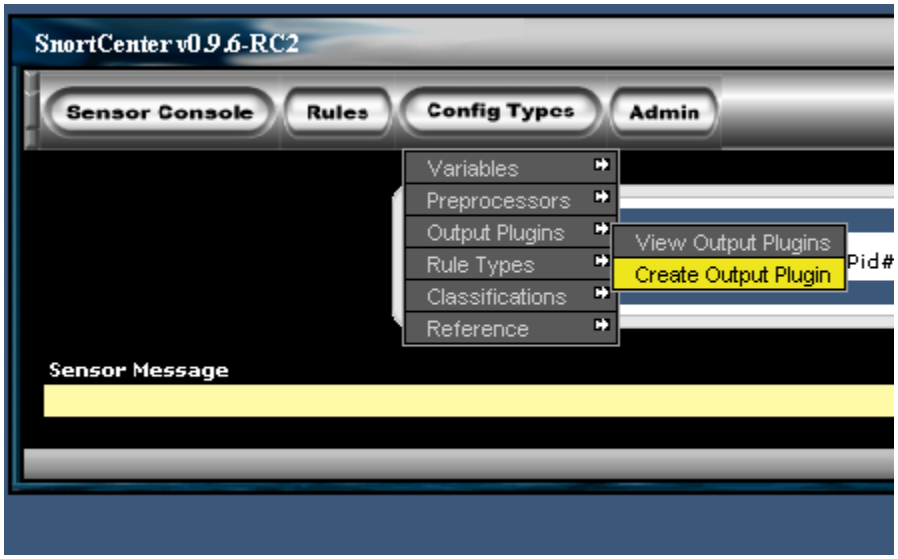
The default account is “admin” with the password “change”.

Follow these steps your first time:

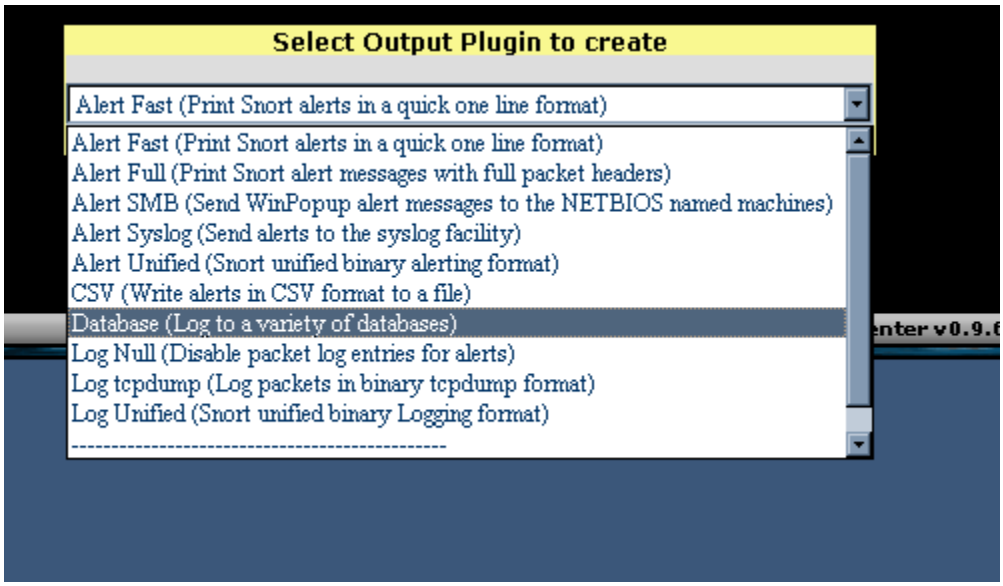
1. Select ADMIN -> Create DB Tables
2. Select RULES -> IMPORT / UPDATE RULES -> UPDATE FROM INTERNET
3. Select ADMIN -> ACTIVATE DEFAULT SNORT RULES
4. Change your ADMIN password
    a. ADMIN -> USER ADMINISTRATOR -> VIEW USERS
        i. Click the edit icon(Looks like an open book)

Next we need to configure the default output plugin for all your sensors:

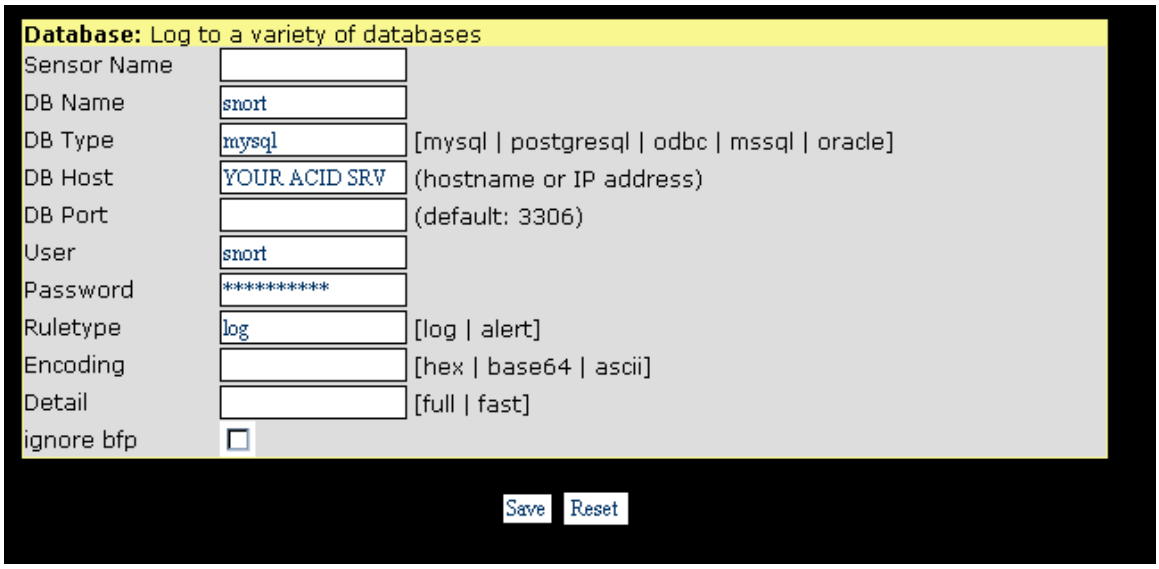
1. Select CONFIG TYPES -> OUTPUTPLUGINS -> CREATE OUTPUT PLUGIN (Shown Below)



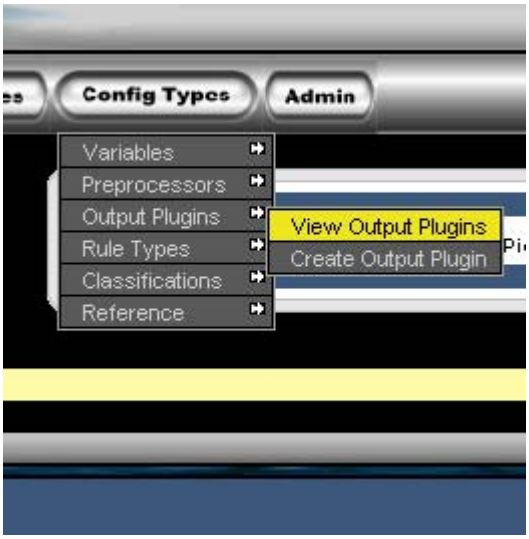
Now select the database option:



Configure the database options like below. **Then Click "SAVE"**



Now we need to activate it on the default server (more on this later).



Now you'll see a screen like this. Click on the red square to make it green!

This will now become your default output plugin for all your sensors.

Snort Sensor 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-1.9.*.tar.gz. Download the latest version and compile it.

```
# cp snort-1.9.*.tar.gz  /usr/src/redhat/SOURCES
# cd /usr/src/redhat/SOURCES
# tar –zxvf snort-1.9.*.tar.gz
# cd /usr/src/redhat/SOURCES/snort-1.9.*
# ./configure --with-mysql
# make
# make install
```

Create a directory for your snort configuration files:

```
# mkdir /etc/snort
```

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 as well.

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

SnortCenter Agent Installation

Install dependencies for using SSL connections with SnortCenter. 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
```

Start the Snortcenter agent install.

```
# mkdir /opt/snortagent/
# cp snortcenter-agent-v0.1.6*.tar.gz /opt/snortagent
#  cd /opt/snortagent
#  tar -zxvf snortcenter-agent-v0.1.6*.tar.gz
#  cd sensor
# ./setup.sh
```

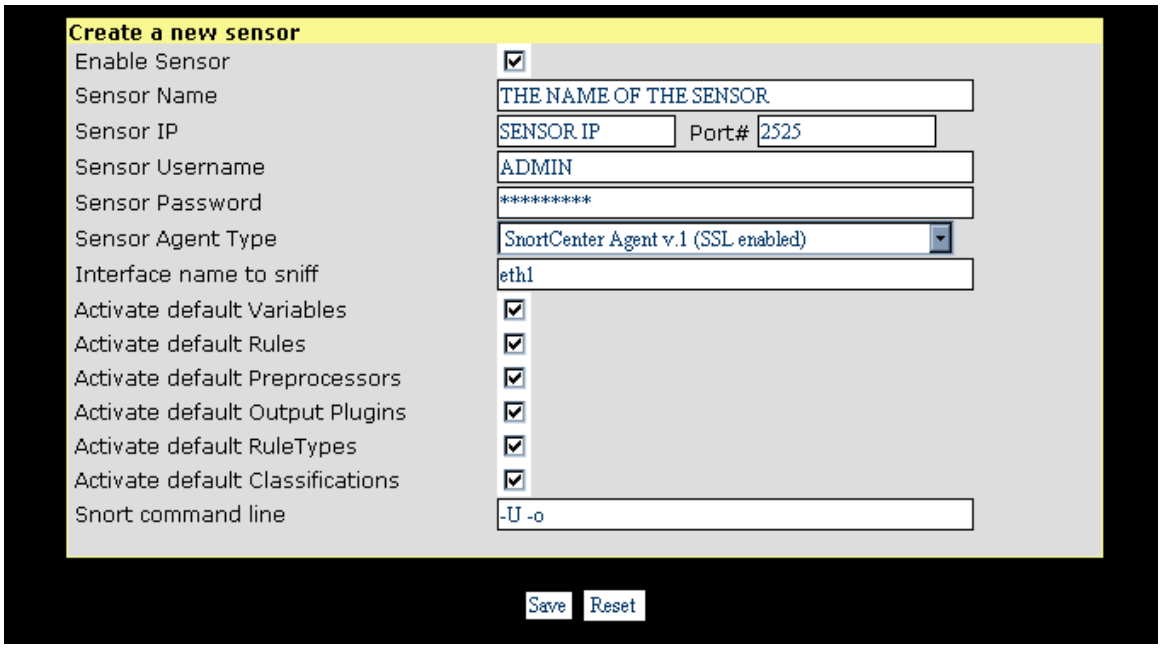
You will then be prompted with a series of questions:

Config File Directory	= /etc/snort
Log File Directory	= /var/log/snort
Perl	= <ENTER>
Snort	= <ENTER>
Snort Rule Config File	= /etc/snort
Port	= <ENTER>
IP Address	= (Your sensors management IP (eth0))
Login Name	= <ENTER>
Password	= (Password that the consoles uses to access the sensor)
Confirm Password	= (Same as above)
Host	= <ENTER>
SSL	= Y
Allow IP	= (This is the IP address of you SnortCenter Server)
Start on Boot	= Y

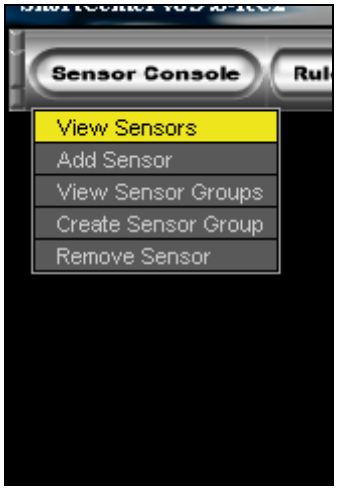
Adding Sensors to the SnortCenter Console

Once you have the SnortCenter agent installed you need to tell the SnortCenter console about it. Access the SnortCenter website you setup and add a new sensor:

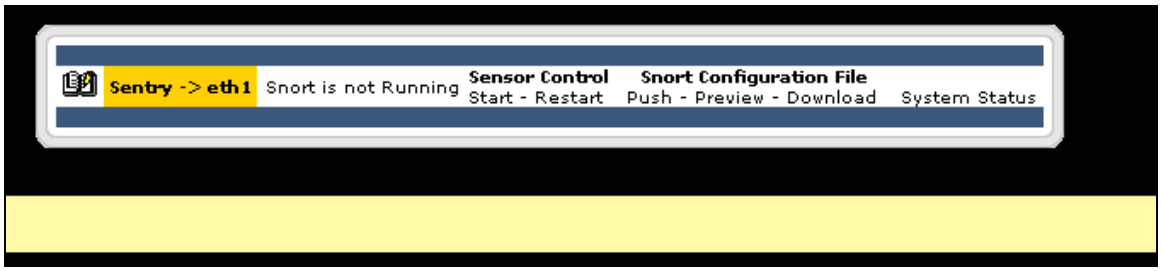
You should then fill in the following:



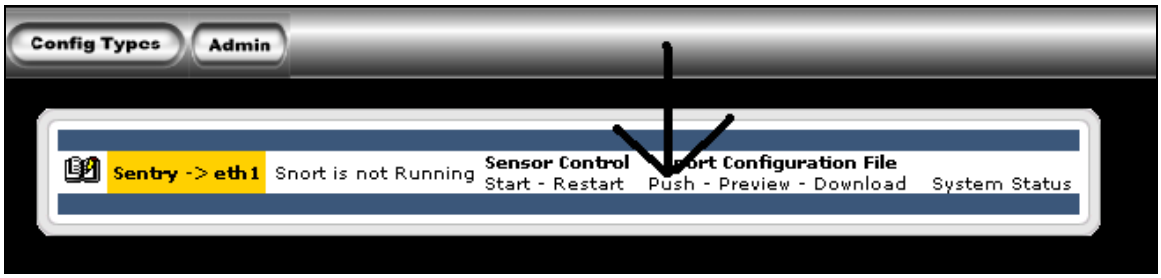
Click save and go to sensor view:



You sensor should appear like this.

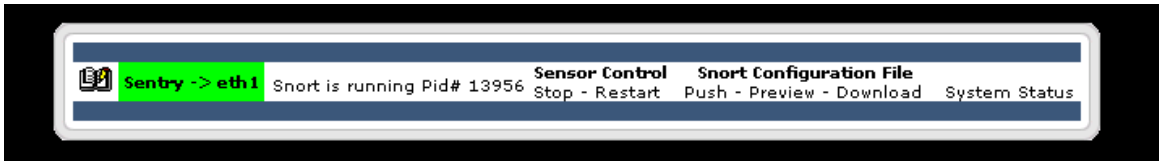


Now we need to push our defaults rules and settings to the sensor. Click on PUSH.



If everything goes right you shouldn't see any messages. Next lets start our sensor.

Finally it should turn green like this:



The sensor is now running the default rules.

Using SnortCenter

SnortCenter is an easy to use tool for centrally managing multiple Snort sensors. The system manages your Snort signatures, the Snort configuration file, and monitors the sensors availability all from a central server. This section is intended to give you an overview of how SnortCenter operates.

The first thing we need to understand is the way SnortCenter manages the rules and configuration files for the sensors. SnortCenter uses the term scope to define what will be affected by the change or operation. The main scope or global scope is "Default sensor". Any changes made while this scope is selected affects all the **NEW** sensors that are added. You should only use this scope for two reasons:

- To set all the default configurations of newly added sensors. E.g. setting up configuration parameters like database options, preprocessor plugins, etc.

- When adding filters to the local.rules category. The "Save as New" option only shows up under the scope option of default server.

The other scopes that are available are directly related to sensors that you have SnortCenter managing. As you add sensors you will see that the scope menu expand to display your sensors as a scope. Any time that an operation is performed with a sensor selected as the scope, that operation only affects that specific sensor. This is demonstrated in the filtering section.

Now that we understand scope there's really only one more thing to know. Normally when you're running Snort individually each snort sensor has its own local.rules file. This file is used for adding your own rules. SnortCenter uses one local.rules file for all your sensors. Now you may be thinking that every rule you want to add or filter will have to be included with every sensor. Wrong! Remember that each configuration option is effected by the selected scope. By using the scope option on the local.rules file, you are able to control which rules are active with respect to each sensor. It also has the benefit of reusing the same filter or rule on multiple sensors!

SnortCenter's interface provides a consistent look and feel throughout the numerous configuration screens. Almost everything within the interface is self-explanatory; so don't be afraid to click on something. Let go over some the common Icons and the color references you will be dealing with.

This represents that the item can be edited, like rules or a sensor's configuration.

The color green represents that everything is operational or the item is active.

The color orange means that the Snort service is not running or that something is only partly selected.

The color red means that agent communication has failed or an item is deactivated.

If you're having problems starting or pushing a policy to your sensors refer to the "View sensor" screen and look at the sensor messages window at the bottom of the screen. All operations will report their output in this window.

The next version of SnortCenter will incorporate policy-based management. This will allow you to put the same configuration to preexisting sensors. Currently if you want a change to affect all sensors that are currently in the system, you have to apply the change to each of them.

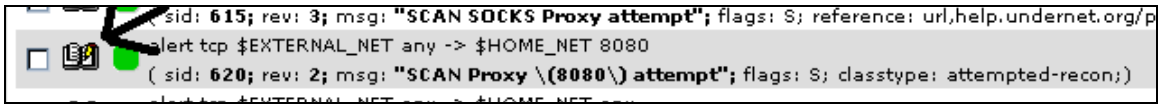
Filtering events with SnortCenter

Filtering events in SnortCenter is fast and efficient. The key is to remember that all the sensors share the same local.rules file. The way you control which sensors are affected by the filter is through the *SENSOR SCOPE* option. So lets start with a simple example of filtering out proxy attempts made to our legitimate proxy server.

Let's first enter rules screen:

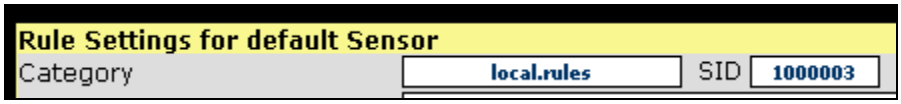
You should now be at the main rules screen. Next navigate to the *RULES SCOPE* option and select <scan.rules>. Notice we left the *SENSOR SCOPE* to <default sensor>.

Now all the rules we are seeing are the ones in the <scan.rules> file. Look for this line and click the edit option (the open book)

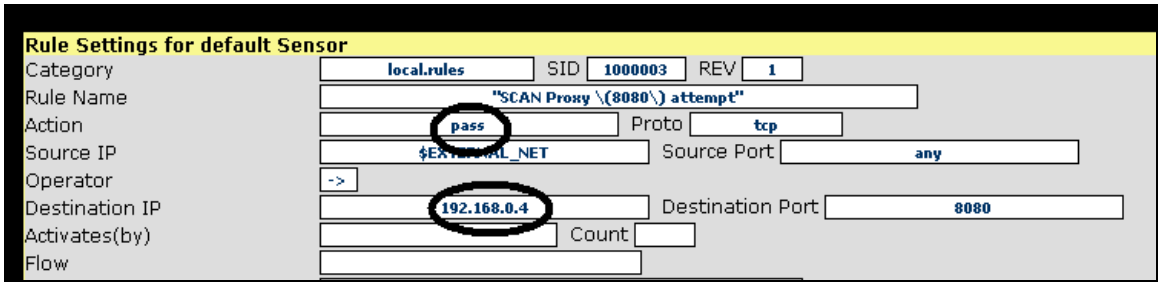


You then a see the edit screen with all the fields that makes up a signature definition. At the bottom of the screen there is an option to "Save as New". Select it.

Take notice to the changes that have occurred. The category field has changed to local.rules and a new SID number has been created as noted below.



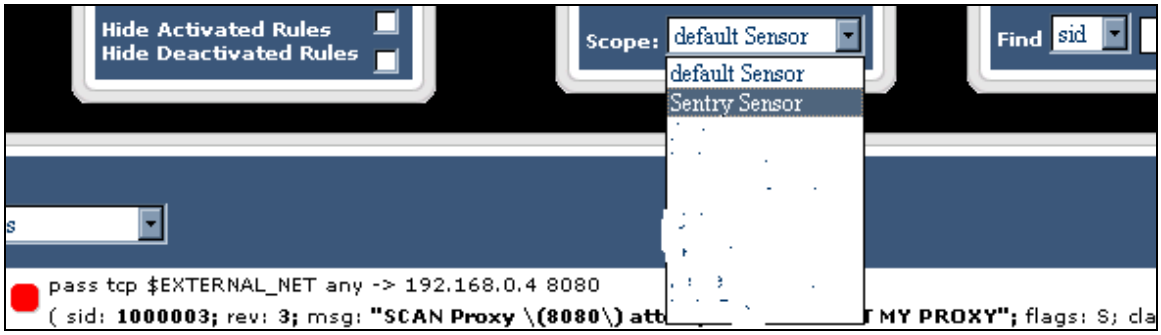
For this example we are going to filter out our real proxy host at 192.168.0.4. So we need to change the ACTION field to pass and the destination field 192.168.0.4. Additionally, you could edit the Rule name and provide some information about what the filter is filtering.



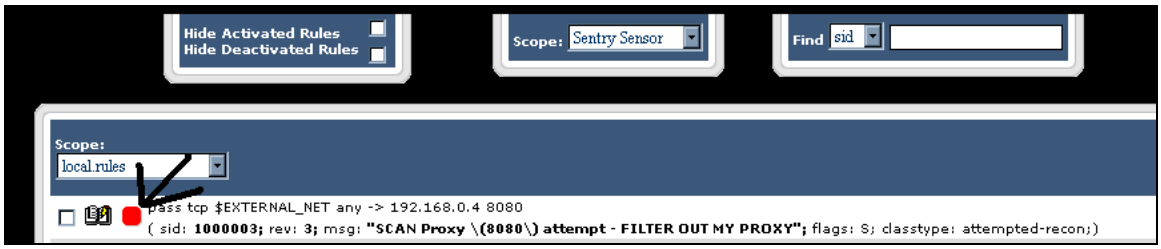
Now click the Update button.

What we have now done is create a new rule in the <local.rules> file. The rule is currently inactive, so our next step is to active it for the appropriate sensor. Lets go back to the main rules screen.

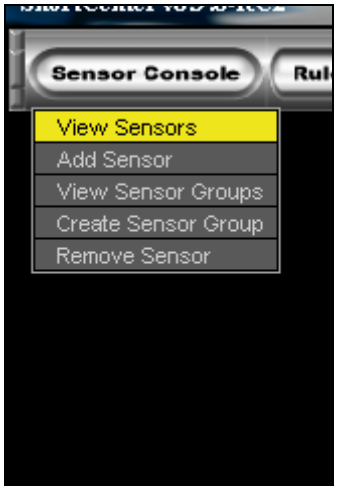
The *RULE SCOPE* should default to <local.rules>. If not just change it. Next we need to change the *SENSOR SCOPE* as noted below to the sensor that the filter is to be placed on. In case it's the Sentry sensor.



Now lets activate the rule by clicking on the RED SQUARE.



It should now be green. The last thing to do is push the new policy to the sensor and restart it. Go back to the main sensor view and select Push and then Restart.



That's it!

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 your 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 manually 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/
Redhat Home Page	http://www.redhat.com/
Redhat 7.3 Reference Books	http://www.redhat.com/support/resources/howto/rhl73.html
Redhat 7.3 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
SnortCenter	http://users.pandora.be/larc/index.html
Incidents.og	http://www.incidents.org/

Appendix A – Important Files, Directory's and Commands

SnortCenter Agent

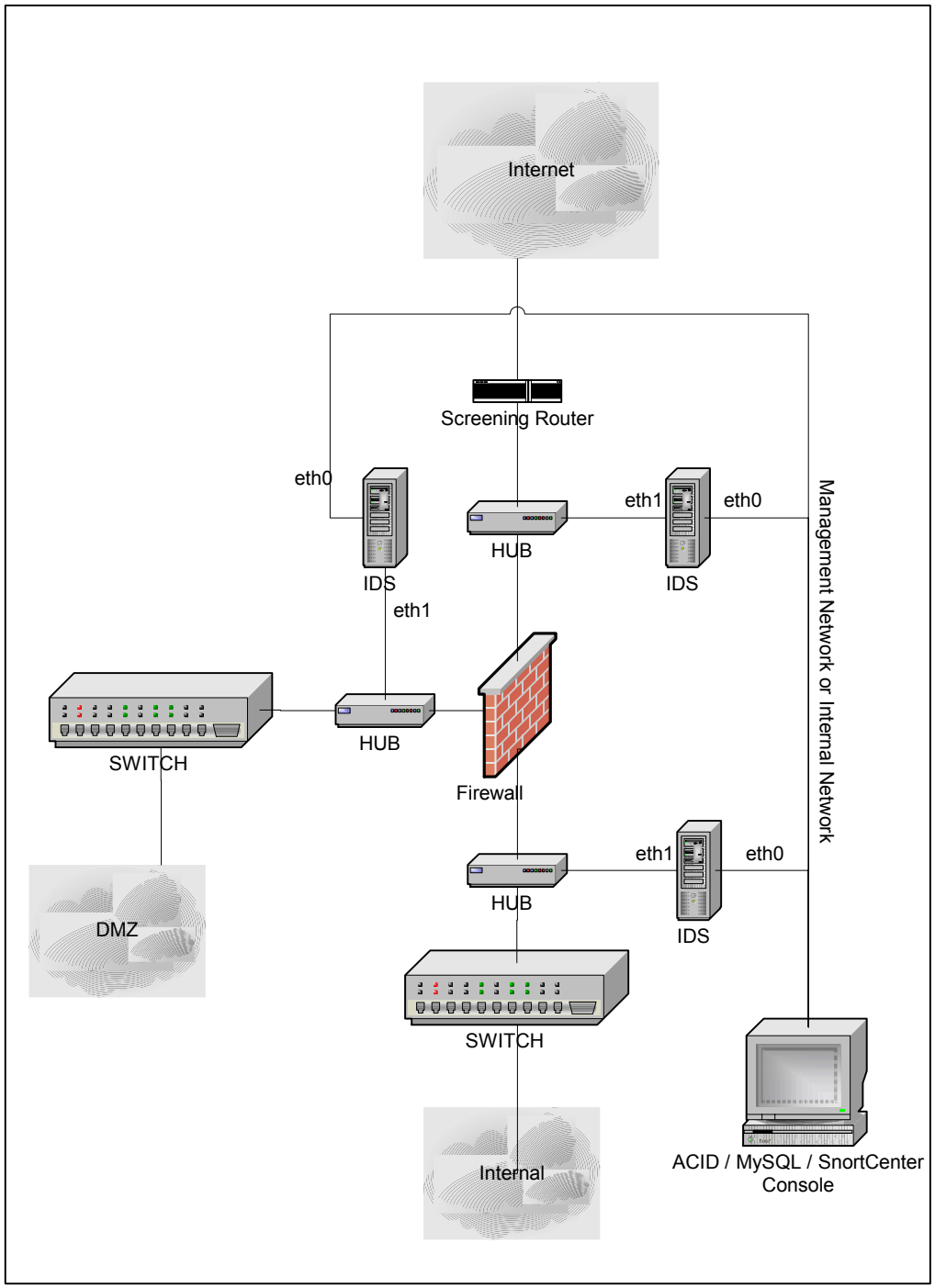
SnortCenter has two files that can be edited if necessary, and most likely will only need to be edited if you made a mistake during the install or your configuration changes.

/etc/snort/config	holds the agent path information among other things.
/etc/snort/miniserv.conf	contains most of the variables that you answered during the install

You can also start and stop SnortCenter agent by using the *service* command in Linux.

Start the agent	# service sensor start
Stop the agent	# service sensor stop
Restart the agent	# service sensor restart

Appendix B – Physical IDS Placement Drawing



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

V2.0 Octomber, 2002
Document layout and formatting changes
SnortCenter section was added
Sensor Tuning with SnortCenter was added
Appendix A – Important Files and Directory's was added
Appendix B – Physical Placement Diagram was added
Removed all references to Webmin and the Snort plugin
How to section was revamped
Document name changed
Error corrections