

Snort Installation Manual for Red Hat Enterprise 4.0

Snort on BASE

**Installed on
Red Hat Enterprise Linux ES 4.0 Update 3**



**September 2006
Version 2.10**

Bruce A. Westbrook

Table of Contents

Introduction:.....	1
Acknowledgments:	2
Comments or Corrections:	2
Revisions:.....	3
Other important reading:.....	5
Conceptual Topology:.....	6
Systems Requirements	7
Install, Update and Secure Red Hat	8
Installing and Configuring Red Hat Enterprise	8
Update the System	10
System Tweaking and Hardening	11
Install and Configure Snort Console.....	17
Copy Snort Installation Files	17
Install and Configure Prerequisites	19
Install Snort.....	21
Snort Startup Options.....	24
Setup MySQL Database.....	24
Test Snort	28
Install Prerequisites for BASE	30
Install BASE	30
Secure Apache	32
Install Prerequisite for Webmin	34
Install and Configure Webmin.....	35
Install and Automate PigSentry	36
Setup MySQL Database Dump and Backup.....	36
Update Snort Rules Automagically Using Oinkmaster	37
Watching the Watcher.....	39
Final Check	40
Adding Sensors	41
Install, Secure and Update Red Hat	41
Copy Snort Installation Files	42
Install Snort.....	42
Snort Startup Options.....	45
MySQL User for Sensor	46
IPTables Rule on Sensor	47
Test Snort	47
Install Prerequisite for Webmin	48
Install and Configure Webmin.....	49
Install and Automate PigSentry	50
Update Snort Rules Automagically Using Oinkmaster	50
Watching the Watcher.....	52
Final Sensor Tuning.....	53
Filtering Rules:.....	54

Snort Installation Manual v2.10

Snort on BASE

Installed on Red Hat Linux Enterprise ES v4.0 Update 3

v1.0 - Created by Patrick S. Harper, CISSP MCSE

v2.10 - Updated by Bruce A. Westbrook, CISSP MCSE CCNA

Introduction:

This document originated from Patrick S. Harper (<http://www.InternetSecurityGuru.com>), when a friend of his asked him to put together this procedure so that he could install Snort and Acid. It is pretty straightforward and can be used by both the Linux/Snort newbie, as well as the advanced guru who just needs to get this deployed. This is a “How in the hell do I get this installed and working” guide, including a security lockdown of your snort box(es). The purpose of this guide is to document the installation and configuration of a complete Snort implementation, based originally on Patrick’s document for Snort 2.0.4 and Red Hat 9.0. This guide contains all the necessary information for installing and securing your Snort IDS infrastructure, as well as add-ons for managing and keeping tabs on your Snort installation.

This document will walk through how to install a stand alone Snort server (good for consultant laptop sensors or SOHO sensors). Following that there will be a section for adding additional sensors that log back to your first Snort server (known as the Snort Console).

The information in this guide was written for implementing Snort 2.6.0 using Red Hat Enterprise Linux. You may find some discrepancies if you are installing different versions of Snort or using different versions of Linux.

While this guide can be used by the Linux/Snort newbie, it was written with the assumption that you understand what Snort is and have a basic understanding of Linux. This includes editing files, making directories, and understanding general *nix commands. This guide also explains some details on using and configuring Snort, although not in great detail. Links on where to obtain additional information can be found in the “Other Important Reading.”

Acknowledgments:

My thanks first goes to Patrick S. Harper for the original document from which I forked this document.

Thanks to Steven J. Scott and his documents “Snort Installation Manual – Snort, MySQL and ACID on Redhat 7.3” and “Snort Enterprise Implementation – Snort, MySQL, SnortCenter and ACID on Redhat 9.0” where I was able to understand and create the sections pertinent to making Webmin work with the Snort plugin.

Thanks to SANS for their excellent guide, “Securing Linux – A Survival Guide for Linux Security”, Center for Internet Security for their “Linux Benchmark” guide, and to MicroSolved Inc. (www.microsolved.com) for their review of this document from a security perspective.

Thanks to Terry Crow for his review and expert editing of the original final draft.

Thanks to the entire Information Technology group at Corporate One FCU for their feedback, much of which was included into the final original document.

Comments or Corrections:

Corrections should be submitted to Bruce Westbrook, bwestbrook@gmail.com.

Flames go to `/dev/null`

The latest version of this document and the files mentioned herein can be found at:

<http://www.thewestbrooks.com/downloads>

Revisions:

v2.10 – Revised Release – Bruce A. Westbrook

- Updated to Red Hat Enterprise Linux ES v4.0 Update 3
- Updated from Snort v2.4.3 to Snort v2.6.0
- Minor corrections throughout document

v2.00 – Revised Release – Bruce A. Westbrook

- Updated from Snort v2.3.3 to Snort v2.4.3
- Updated from ACID to BASE (a fork of ACID)
- Revised as a more “down and dirty” quick install of snort to get you up and running
- Removed tons of extraneous information that was unnecessary for a quick install guide
- Reformatted to include the use of my familiar procedures template
- Revised to use rpm based install instead of compiling everything from source in order to speed up deployment
- Removed Aanval

v1.60 – Revised Release – July 2004 – Bruce A. Westbrook

- Revised to be used with Red Hat Enterprise Linux 3.0 – Update 2
- Added Aanval, a new reporting tool

v1.52 – Revised Release – December 2003 – Bruce A. Westbrook

- Added script (test.sh) to check that snort is running, and alerts if it is not
- Added content and script (gooink) on obtaining, testing and updating Snort rules

v1.51 – Revised Release – December 2003 – Bruce A. Westbrook:

- Added the BPF filter section to the configuration file
- Added the BPF filter tweaking section
- Updated version for Snort from 2.0.2 to 2.1.0
- Updated version for Apache from 2.0.47 to 2.0.48
- Updated version for MySQL from 4.015 to 4.017
- Revised various instructions for clarification

v1.5 – Revised Release – September 2003 – Bruce A. Westbrook:

- Fixed the MySQL user rights
- Revised various instructions for clarification
- Revised various syntax strings for better use
- Updated versions of Snort, Apache, MySQL, and PHP
- Updated snort rules installation
- Addition of OpenSSL instructions to fix vulnerabilities
- Addition of the Webmin instructions (from Steven J. Scott)

- Addition of installation instructions of separate snort-console and snort-sensors (original document provided only single box)
- Addition of the .htaccess instructions to secure Apache
- Addition of MySQL database backup instructions
- Addition of promiscuous mode instructions on secondary NIC for the snort-sensors
- Addition of script troubleshooting instructions
- Addition of Pigsentry instructions for real-time alerting
- Addition of NTP setup for clock synching
- Addition of system tweaking and system hardening procedures (thanks to SANS and CIS for much of this information (80%), and thanks to too many misc. sources, myself included, for the other 20%)
- Split the installation instructions to encompass separate Snort Console and Snort Sensors

v1.0 – Initial Release - by Patrick S. Harper – reflects recommendations to draft versions and input from Nick Oliver.

Other important reading:

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/>

Snort CVS tree <http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/snort/snort/>

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>

BASE Home Page <http://secureideas.sourceforge.net/index.php>

PHP Home Page <http://www.php.net>

MySQL Home Page <http://www.mysql.com/>

Fedora Linux Home Page <http://fedora.redhat.com/>

Nessus Vulnerability Scanner <http://www.nessus.org/>

NMAP <http://www.nmap.org/>

Linux, Clocks, and Time <http://www.linuxsa.org.au/tips/time.html>

Incidents.org <http://www.incidents.org/>

Putty <http://www.chiark.greenend.org.uk/~sgtatham/putty>

Patrick S. Harper's website <http://www.internetsecurityguru.com>

The Snort Drinking Game http://www.theadamsfamily.net/~erek/snort/drinking_game.txt

Conceptual Topology:

There are six primary software packages that produce this topology. The Apache web server, MySQL database server, Webmin, BASE, Pigsentry, and of course Snort. This topology assumes you will be running a combined sensor, database and BASE console. To use multiple sensors or separate your sensor from your database and BASE console, you will still install the first combined Snort box, and then follow the instructions for installing additional sensors that log back to your first box.

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 sensor will be stored in a MySQL database.

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.

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 BASE.

Basic Analysis and Security Engine (BASE)

BASE provides a web front-end to query and analyze the alerts coming from our Snort IDS system. This is where all the sensor information is consolidated for viewing.

Webmin

Webmin is a package for managing your signatures and snort configuration files via a web-based GUI. While SnortCenter used to be the preferred method for managing multiple sensors, it currently has bugs that is preventing it from working properly with new Snort 2.0 rules. SnortCenter is a console that 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. In its' place I've developed scripts for updates as well as utilizing Webmin.

Pigsentry

Pig Sentry is a lightweight script which is run against a Snort alert log. BASE is very nice for after the fact perusal and historical reporting, but not for up-to-the minute information. Pig Sentry is for real-time alerts, without getting spammed. It is intended for a high volume of alerts (the site it was implemented for sees 200,000 to 300,000 alerts a day).

The way Pig Sentry works is it maintains a state table of recent alerts. It will send a notice if there is a new alert, or if there is an increase in the general trend or pattern of existing alerts. The trend notification is fairly configurable. It also expires alerts after there has been no activity with them for a while.

Systems Requirements

These particular installation procedures were originally designed to be used for a single-server installation. Single-server installs are good for portable laptop sensors or for where only a single sensor is needed.

For multi-server installations I've now included a section on adding additional sensors to this document, rather than splitting it out to a separate document. It simply makes it easier to update everything in one location. These multi-server installs are for use in larger environments where you want multiple segments watched, such as pre- and post-firewall, a DMZ, internal network, vendor connected networks, etc.

For a single-server installation, you will need the following:

- This instruction manual;
- Red Hat Enterprise Linux ES v4.0 Update 3 (4 CDs);
- My customized Snort Installation CD / files;
- A computer with 2 (two) NICs to run everything on (a single NIC can be used if you do not want to secure remote access to the box, such as a portable laptop sensor – however, these instructions are based on dual-NICs).

For multi-server installations:

- The only difference is the NIC requirement. For multi-servers you will definitely need dual-NICs in each sensor. The console can be either single or dual-NIC, depending on whether you are running it as a sensor as well. These instructions assume the console will also be a sensor, but you can easily disable Snort on the console.

Install, Update and Secure Red Hat

Installing and Configuring Red Hat Enterprise

√	Description
Install Red Hat	<p>We will install a minimal number of packages, sufficient for a usable system. After the install we'll turn off anything that is not needed. These installation instructions will build a system that is ideal as a dedicated IDS by hardening the OS and further securing the system.</p> <ol style="list-style-type: none"> 1. Boot with CD 1 of Red Hat 2. You can skip the CD-ROM test 3. Welcome to Red Hat Enterprise Linux ES – click Next 4. Select your language – click Next 5. Select your keyboard – click Next 6. For Installation type select Custom – click Next 7. For Disk Partitioning select Manually Partition with Disk Druid – click Next 8. Setup partitions as follows: <ol style="list-style-type: none"> a. Select your hard-drive (typically /dev/hda) and click Delete – this will delete all partitions on the drive. If this is server hardware, you'll want to delete any partitions individually and leave the utility partition. b. Set boot partition: <ol style="list-style-type: none"> i. Click the New button ii. For Mount Point, select /boot iii. For size (MB) enter 100 iv. Click the checkbox for Force to be a primary partition v. Click OK c. Set swap: <ol style="list-style-type: none"> i. Click the New button ii. Leave the Mount Point blank iii. For File System Type select Swap iv. For size (MB) enter the size of the swap partition (RAM times 1.5) in megabytes v. Click OK d. Set root: <ol style="list-style-type: none"> i. Click New ii. For Mount Point, pulldown and select / iii. For File System Type leave as ext3 iv. For size (MB) enter 8096 v. Click OK e. Set var: <ol style="list-style-type: none"> i. Click New ii. For Mount Point, pulldown and select /var iii. For File System Type leave as ext3 iv. For size (MB) ignore the field and instead select "Fill to maximum allowable size" v. Click OK f. Click Next

		<ol style="list-style-type: none">9. For Boot Loader Configuration click Next10. For the Network Devices screen, set your static IP, your FQDN hostname, gateway and DNS servers.11. Click Next12. For Firewall Configuration, select to Enable the firewall and then allow the following services:<ol style="list-style-type: none">a. Remote Login (SSH)b. Web Server (HTTP, HTTPS)13. Leave the Enable SELinux as Active14. Click Next15. For Additional Languages – click Next16. For Time Zone, set your time zone. Do <u>not</u> enable the system clock to use UTC – click Next17. Set your root password and click Next18. For Package Installation Defaults select Customize software packages to be installed19. Click Next20. Now make the following software <u>changes</u> at the Package Group Selection screen (unless mentioned here, keep all other package selections as they are):<ol style="list-style-type: none">a. Under Applications select the following:<ol style="list-style-type: none">i. Editors (keep defaults)ii. Graphical Internet (only)<ol style="list-style-type: none">1. Firefoxiii. Graphics (only)<ol style="list-style-type: none">1. Gimp2. Gimp-data-extras3. Gimp-print-pluginb. Under Servers select the following:<ol style="list-style-type: none">i. Web Server (only)<ol style="list-style-type: none">1. Mod_Auth_mysql2. Mod_perl3. Mod_ssl4. Php5. Php_mysqlii. MySQL Database (defaults plus the following)<ol style="list-style-type: none">1. mysql-server2. php-mysqlc. Under Development select the following:<ol style="list-style-type: none">i. Development Tools (defaults)d. Under system select the following:<ol style="list-style-type: none">i. System Tools (defaults plus the following)<ol style="list-style-type: none">1. ethereal-gnome2. nmap-frontentii. Deselect Printing Support21. Click Next and Next again to begin loading the system22. When the installation is complete, you will be prompted as such. Click the Reboot button
--	--	--

	<p>Post-installation Wizard</p>	<p>After the installation and the initial reboot you will be walked through a post-installation wizard.</p> <ol style="list-style-type: none"> 1. At the welcome screen – Next 2. Accept the license agreement – Next 3. Set the date/time – Next 4. Set your display as appropriate – Next 5. Configure your Red Hat Login as appropriate – Next 6. Create a console user account – Next 7. Test your audio device – Next 8. For Additional CDs, just click Next 9. Finish Setup – Next
	<p>Boot CLI</p>	<p>Let's now configure linux to boot up into text mode, not GUI. No reason to boot into the GUI by default on a server. To do this first login and then launch a terminal session (Applications => System Tools => Terminal), edit the <code>/etc/inittab</code> file and change the following line:</p> <pre>vi /etc/inittab id:5:initdefault change to id:3:initdefault</pre>

Update the System

√	Description	
	<p>Update System with Red Hat Up2date</p>	<p>Follow your organization's procedures for updating your system. I am not providing details on this since each organization is different and could be updating in various ways, with various policies and so forth.</p> <p>I'm also assuming this is an organization and not an individual since you're installing a stable, pay version of Red Hat, rather than a freebie distro like Fedora.</p>
	<p>Reboot</p>	<p>Reboot your box after updating. This will also now put you at a command prompt rather than the GUI.</p>

System Tweaking and Hardening

√	Description
User Account	<p>If you didn't create one at the end of the installation process, you should create a normal user account – typically I create a <code>console</code> account.</p> <pre>useradd console passwd console New password: password</pre> <p>After setting up the user you can hit <code>[Alt-F2]</code> and test the login.</p> <p>You will need this account to be able to SSH to the box, since we'll secure SSH to not allow root to login as a security measure.</p>
Date / Time	<p>If you setup an NTP server during the installation, you can check that it is running properly by issuing the command:</p> <pre>ntpq -p</pre> <p>The output should show your <code>*LOCAL</code> line plus one line for each of your configured NTP servers. The jitter column should show something other than 4000.00. A telltale sign that NTP synchronization is not working is a jitter of 4000.00. If this is the case, you can try to trace the problem with the following command: <code>ntptrace -vd NTP_server</code></p> <p>If you have no NTP servers setup you can set your local date and time as follows:</p> <ol style="list-style-type: none"> 1. Type <code>date</code> to check the current date/time 2. Change the date/time with the following syntax: <code>date -s "06/03/2004 09:36:00"</code> 3. Now sync the hardware clock <code>hwclock --systohc</code>
NumLock	<p>For workstations (you probably don't want to do this on a laptop) you can set the NumLock to enable on boot as follows:</p> <pre>vi /etc/rc.d/rc.local</pre> <p>Go to the end of the file and add:</p> <pre>INITTTY=/dev/tty[1-8] for tty in \$INITTTY; do settled5 -D +num <\$tty done</pre>

	<p>Disable CTRL+ALT+DEL</p>	<p>To disable the accidental rebooting of your linux box with your Microsoft happy fingers, perform the following:</p> <pre>vi /etc/inittab</pre> <pre>#ca::ctrlaltdel:/sbin/shutdown -t3 -r now</pre> <pre>ca:ctrlaltdel:/bin/echo "[CTRL]+[ALT]+[DEL]disabled"</pre> <p>After editing the /etc/inittab file you should execute the following:</p> <pre>/sbin/init q</pre> <p>This will reinitialize the inittab and include your new settings. Of course this will also occur on a reboot.</p>
	<p>Password Protect Single-user Mode</p>	<p>To add a level of protection to your box from being easily logged into as root by someone with physical access, perform the following:</p> <pre>vi /etc/inittab</pre> <pre>id:3:initdefault</pre> <pre>~~:S:wait:/sbin/sulogin</pre> <p>After editing the /etc/inittab file you should execute the following:</p> <pre>/sbin/init q</pre> <p>This will reinitialize the inittab and include your new settings. Or you can just reboot your box.</p>
	<p>Warning Banners - Local -</p>	<p>Edit the /etc/issue file to add whatever you'd like for a warning banner. An example follows:</p> <pre>vi /etc/issue</pre>

```
*****
*
*
*           This system is for authorized use only.
*
*           All activity is logged and monitored
*
*
*****
```

```
Red Hat Enterprise Linux ES release 4 (Nahant Update 3)
Kernel \r on an \m
```

	<p>Warning Banners - Remote -</p>	<p>Copy the <code>/etc/issue</code> file you just created to <code>/etc/issue.net</code>. Edit the file and remove the last two lines that identify the system, leaving only the warning banner itself.</p> <pre>cp /etc/issue /etc/issue.net vi /etc/issue.net</pre>
	<p>MOTD Banner</p>	<p>You can also edit the MOTD (Message Of The Day) file to display a message after a successful login:</p> <pre>vi /etc/motd</pre> <pre>Login authenticated and logged</pre>
	<p>Secure xinetd.d Services</p>	<p>Almost every old xinetd service has been replaced by newer and more secure programs. To see if you have any running that you really need, execute the following:</p> <pre>cd /etc/xinetd.d for file in * ; do chkconfig --list \$file ; done</pre> <p>You will see a list of services and whether they're on or off. If any are on, investigate why and determine another way to accomplish the task (such as SSH). Once you have done this, disable the entire xinetd service as follows:</p> <pre>chkconfig --del xinetd</pre>
	<p>Secure Standard Boot Services</p>	<p>Back to the understanding that every system daemon (service) that does not have a clear and defined purpose on the host should be disabled, let's disable daemons that you don't need or use.</p> <p>Here is a list of commonly started services that you can disable initially:</p> <pre><<line wrapped>> for file in anacron atd auditd avahi-daemon avahi- dnsconfd bluetooth cups cups-config-daemon dc_client dc_server diskdimp irda netdump rpcgssd rpcidmapd rpcsvcgssd vncserver; do chkconfig --del \$file ; done</pre> <p>You can then port a list of all your services to a file and browse through it to see what else you can disable. If you don't know what a service does this would be a great opportunity to do some research and understand what your system is running:</p> <pre>chkconfig --list > /root/services</pre>

	<p>Secure SSH</p>	<p>SSH should be configured to display your warning banner and allow only the more secure protocol 2. You should also not permit root logins or empty passwords. This ensures your remote root access is logged via a user account first. Find the following lines, unremark them and change them as shown:</p> <pre>vi /etc/ssh/sshd_config</pre> <pre> Protocol 2 PermitRootLogin no PermitEmptyPasswords no Banner=/etc/issue.net </pre> <p>After saving this file restart the SSH daemon:</p> <pre>service sshd restart</pre>
	<p>Secure Default Firewall Ruleset</p> <p>* Explanation *</p>	<p>As a brief explanation, the firewall rules for iptables are not really kept in any editable file. That is, the rules, once loaded, exist in memory and will overwrite the file they came from. So how do you configure iptables? And how does it load it's ruleset after a reboot?</p> <p>Well, one way is to make changes to the ruleset in memory, on the fly. You then tell iptables to save the rules in memory to a file. When the box reboots, iptables reads the rules from this saved file.</p> <p>So why can't you just change the actual rules in file? Because it's overwritten any time that you save the rules. And you can't delete rules by simply re-reading the file – the file will append to the rules in memory. Instead, you should create a file of your own with all your firewall rules and comments, run your file to add, delete or modify rules in memory, and then save the iptables memory to the /etc/sysconfig/iptables file. Whew!</p> <p>Rather than create a script to do our changes, we will perform the changes on the fly. We'll then save the memory to a file so they get removed permanently on reboot.</p> <p>To do this, we will perform the following:</p> <ol style="list-style-type: none"> 1. Delete (flush) all of the current rules 2. Define our chains/tables in memory 3. Add our "default" rules in memory 4. Add other rules in memory as needed 5. Save the new iptables from memory to the iptables file 6. Restart iptables to verify our changes

<p>Secure Default Firewall Ruleset</p> <p>* Steps *</p>	<p>** IMPORTANT NOTE: Making these changes via a remote SSH connection WILL lock you out almost immediately since command entered are applied in real-time. However, you can script everything in a file and then run the file over an SSH connection and stay connected.</p> <p>In order to stay consistent with our local firewall rules, we will remove any rules that came with the distribution and set up our own.</p> <p>First, let's remove anything that may currently exist in the iptables rules by "flushing" everything as follows:</p> <pre>iptables -F iptables -F INPUT iptables -F OUTPUT iptables -F FORWARD iptables -F -t mangle iptables -F -t nat iptables -X iptables -Z</pre> <p>Define the three chains, INPUT, FORWARD and OUTPUT, default actions. These three rules by themselves will drop all incoming packets and all forward packets. All packets initiated by the host will be allowed.</p> <pre>iptables -P INPUT DROP iptables -P FORWARD DROP iptables -P OUTPUT ACCEPT</pre> <p>Now let's allow some exceptions to our default of dropping all inbound packets. All of the following rules override the global DROP command that we started with.</p> <p>This rule will accept anything that originates from the local loopback interface and allow it to be used by user applications:</p> <pre>iptables -A INPUT -i lo -j ACCEPT</pre> <p>This rule allows connections that have already been established and are in the connection table maintained by the kernel, such as responses to our HTTP requests (line wrapped):</p> <pre>iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT</pre> <p>This rule allows any ICMP packets so we can perform pings and traceroutes as well as respond to pings:</p> <pre>iptables -A INPUT -p ICMP -j ACCEPT</pre> <p>This rule allows SSH connections to the box:</p> <pre>iptables -A INPUT -p tcp --dport 22 -j ACCEPT</pre> <p>This rule allows HTTP connections:</p> <pre>iptables -A INPUT -p tcp --dport 80 -j ACCEPT</pre> <p>This rule allow HTTPS connections (this is line wrapped):</p> <pre>iptables -A INPUT -p tcp --dport 443 -j ACCEPT</pre>
---	--

		<p>This rule will allow you to connect to Webmin on port 10000: <code>iptables -A INPUT -p tcp --dport 10000 -j ACCEPT</code></p> <p>The very last rule we will put in is to drop all remaining packets that didn't match any of our rules. This is simply good practice: <code>iptables -A INPUT -j DROP</code></p> <p>Finally, save your revised rules to a file, restart iptables and then verify your rules are all in place: <code>service iptables save</code> <code>service iptables restart</code> <code>iptables -L</code></p>	
--	--	---	--

Install and Configure Snort Console

Copy Snort Installation Files

✓	Description
<p>Copy Files</p>	<p>The original document (v1.0) had the download locations for all the necessary files. For this edition, the Snort File CD v2.00 files should be used, which contains all necessary files, scripts, docs, etc. I will leave the download locations here for informational purposes only – DO NOT download newer editions. The purpose to using the files from the CD is to establish a consistent installation across all snort installs.</p> <p>Place the Snort File CD v2.00 CD in the coffee cup holder or download all the files from: www.thewestbrooks.com/downloads/snort-rhel4u3.tar.gz</p> <p>Then copy all files to /root/snortinstall, as follows:</p> <pre>mount /dev/cdrom /mnt/cdrom mkdir /root/snortinstall cp -r -v /mnt/cdrom/* /root/snortinstall cd /root/snortinstall chmod -R +wr /root/snortinstall/* umount /mnt/cdrom</pre> <p>OR</p> <pre>cd /root wget www.thewestbrooks.com/downloads/snort-rhel4u3.tar.gz tar -zxvf snort.tar.gz cd /root/snortinstall</pre>
<p>File Locations</p>	<p>Packages are listed (in their order of use) to help establish a consistent baseline of applications for future revisions.</p> <p><u>Where/how these files were downloaded:</u> You can use wget (wget will place the file you're downloading into the directory where you're currently located) to download these files.</p> <p>To use wget, type <code>wget <URL_to_file></code> and it will begin the download to the directory that you are currently in. If you need to pass credentials for a proxy server, the syntax is <code>wget --http-user=username --http-passwd=password <URL_to_file></code></p> <p>If you want to use a Windows box and need an SSH client, then you can go to the PuTTY http://www.chiark.greenend.org.uk/~sgtatham/putty/ home page and download a free one. You can also get a scp (secure copy) and a sftp (Secure FTP) client for Windows there if you'd like.</p>

	<p><u>PCRE 5.0</u> http://easynews.dl.sourceforge.net/sourceforge/pcre/pcre-5.0.tar.gz Perl Compatible Regular Expressions – used in Snort v2.1.0 and above</p> <p><u>Snort 2.6.0</u> http://www.snort.org/dl/current/snort-2.6.0.tar.gz http://www.snort.org/dl/binaries/linux/snort-2.6.0-1.RHEL4.i386.rpm http://www.snort.org/dl/binaries/linux/snort-mysql-2.6.0-1.RHEL4.i386.rpm</p> <p><u>ADODB v4.62</u> http://easynews.dl.sourceforge.net/sourceforge/adodb/adodb462.tgz A graphics library dependency for BASE</p> <p><u>BASE 1.2.6</u> http://easynews.dl.sourceforge.net/sourceforge/secureideas/base-1.2.6-1.i386.rpm Basic Analysis and Security Engine</p> <p><u>NetSSLeay 1.23</u> http://www.webmin.com/download/Net_SSLeay.pm-1.23.tar.gz SSL implementation – used for Webmin</p> <p><u>Webmin 1.300</u> http://prdownloads.sourceforge.net/webadmin/webmin-1.300-1.noarch.rpm Web-based administration interface for Linux</p> <p><u>Snort Webmin Module 1.1</u> http://www.msbnetworks.net/snort/download/snort-1.1.wbm Webmin interface plugin for snort</p> <p><u>PigSentry</u> http://web.proetus.com/tools/pigsentry/pigsentry-1.2 Alerting tool for snort</p> <p><u>oinkmaster</u> http://oinkmaster.sourceforge.net/ Perl script used to automate the process of downloading and merging Snort rules</p>
--	---

Install and Configure Prerequisites

√	Description
Login	<p>If you are not logged in as root, then you will need to su to root (<code>su -</code> will load the environmental variables of root).</p> <p>Go to your download directory (<code>/root/snortinstall</code>) and start with the following procedures.</p>
Start Services	<pre>chkconfig httpd on chkconfig mysqld on service httpd start service mysqld start</pre> <p>If a message appears stating <code>httpd: could not determine the servers fully qualified domain name, using 127.0.0.1 for ServerName</code> when you start apache (httpd) then you need to edit the <code>/etc/hosts</code> to add the FQDN of the server (i.e. <code>host_name.domain_name</code>).</p>
Install PCRE	<pre>tar -zxvf pcre-5.0.tar.gz cd pcre-5.0 ./configure make make install cd ..</pre>
Test Apache and Verify PHP Functionality	<p>This procedure will test your default install of the Apache web server in <code>/var/www/</code>. This is the default installation location for Red Hat. This method will also test the PHP module.</p> <p>To test the PHP install, create a file called <code>phptest.php</code> in the <code>/var/www/html</code> directory.</p> <pre>vi /var/www/html/phptest.php</pre> <p>Place the following line in the file:</p> <pre><?php phpinfo(); ?></pre> <p>Now use a web browser (either use <code>lynx</code> locally, <code>startx</code> and use Firefox, or use another PC and browse to the snort IP address) to look at the file <code>http://localhost/testphp.php</code>. It should give you info on your system, Apache, and PHP. If it fails, then troubleshoot the failure notification – remember, google is your friend! ☺</p> <p>If you would like another PHP test and a cool little tool, try the Network Query Tool from http://shat.net/php/nqt/nqt.php.txt:</p> <pre>cp /root/snortinstall/scripts/nqt.php /var/www/html</pre> <p>Open the <code>nqt.php</code> file in a browser. It will look like the following:</p>

		<div style="text-align: center;"> <h3>Network Query Tool</h3> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #4a7ebb; color: white;"> <th style="text-align: left; padding: 5px;">Host Information</th> <th style="text-align: left; padding: 5px;">Host Connectivity</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;"><input type="radio"/> Resolve/Reverse Lookup</td> <td style="padding: 5px;"><input type="radio"/> Check port: <input style="width: 50px;" type="text" value="80"/></td> </tr> <tr> <td style="padding: 5px;"><input type="radio"/> Get DNS Records</td> <td style="padding: 5px;"><input type="radio"/> Ping host</td> </tr> <tr> <td style="padding: 5px;"><input type="radio"/> Whois (Web)</td> <td style="padding: 5px;"><input type="radio"/> Traceroute to host</td> </tr> <tr> <td style="padding: 5px;"><input type="radio"/> Whois (IP owner)</td> <td style="padding: 5px;"><input checked="" type="radio"/> Do it all</td> </tr> <tr style="background-color: #4a7ebb; color: white;"> <td colspan="2" style="padding: 5px; text-align: center;"> <input style="width: 150px;" type="text" value="Enter host or IP"/> <input type="button" value="Do It"/> </td> </tr> </tbody> </table>	Host Information	Host Connectivity	<input type="radio"/> Resolve/Reverse Lookup	<input type="radio"/> Check port: <input style="width: 50px;" type="text" value="80"/>	<input type="radio"/> Get DNS Records	<input type="radio"/> Ping host	<input type="radio"/> Whois (Web)	<input type="radio"/> Traceroute to host	<input type="radio"/> Whois (IP owner)	<input checked="" type="radio"/> Do it all	<input style="width: 150px;" type="text" value="Enter host or IP"/> <input type="button" value="Do It"/>	
Host Information	Host Connectivity													
<input type="radio"/> Resolve/Reverse Lookup	<input type="radio"/> Check port: <input style="width: 50px;" type="text" value="80"/>													
<input type="radio"/> Get DNS Records	<input type="radio"/> Ping host													
<input type="radio"/> Whois (Web)	<input type="radio"/> Traceroute to host													
<input type="radio"/> Whois (IP owner)	<input checked="" type="radio"/> Do it all													
<input style="width: 150px;" type="text" value="Enter host or IP"/> <input type="button" value="Do It"/>														
	<p>Configure SSL for Apache</p>	<p>Let's first create a new server certificate so it displays the server name and is setup to last longer then one year:</p> <p>Create the SSL certificates:</p> <pre>mkdir /var/www/certs cd /var/www/certs openssl genrsa -out server.key 1024</pre> <p><<line wrapped>></p> <pre>openssl req -new -key server.key -x509 -out server.crt -days 1095</pre> <p>Answer the various certificate questions.</p> <p>Edit your ssl.conf file to point to the cert you just created:</p> <pre>vi /etc/httpd/conf.d/ssl.conf</pre> <p>Locate the Server Certificate section. Change the SSLCertificateFile path to:</p> <pre>/var/www/certs/server.crt</pre> <p>...and now locate the Server Private Key section right underneath. Change the SSLCertificateKeyFile path to:</p> <pre>/var/www/certs/server.key</pre> <p>Now configure Apache to only allow SSL connections. In our case this will be easy, since we don't want to listen on port 80 at all – no rewrites, not redirection, nada. We'll simply turn off listening on port 80 all together.</p> <p>Edit your httpd.conf files, <code>vi /etc/httpd/conf/httpd.conf</code>, and locate the line Listen 80. Simply remark it out with a #, so it looks like this:</p> <pre>#Listen 80</pre> <p>Save and exit the file, then restart Apache.</p> <pre>service httpd restart</pre>												

	<p>Reconfigure Firewall</p>	<p>We should now disable unencrypted HTTP access to the server through the firewall.</p> <p>If you configured your iptables firewall based on your own organizations procedures, then you should know how to remove port 80 access in iptables. Otherwise, based on our previous configuration within these procedures, simply execute the following to remove the HTTP rule:</p> <pre>iptables -D INPUT -p tcp -dport 80 - j ACCEPT</pre> <p>Then save your changes restart iptables and then verify your rule is gone:</p> <pre>service iptables save service iptables restart iptables -vnL</pre> <p>You should only see port 22 and port 443 allowed through (other then local and established traffic, of course).</p> <p>Now open a browser and go to the server via HTTP://. You should get a “no connection” error as if the server doesn’t exist. Now use HTTPS:// and you should be prompted for authentication. Done!</p>
--	------------------------------------	--

Install Snort

√	Description	
	<p>Install Snort</p>	<pre>cd /root/snortinstall mkdir /etc/snort mkdir /var/snort mkdir /var/log/snort rpm -ivh snort-2.6.0-1.RHEL4.i386.rpm rpm -ivh snort-mysql-2.6.0-1.RHEL4.i386.rpm</pre>
	<p>Install Rules</p>	<p>We will use some pretty old rules to get snort up and running, but we’ll be updating the rules to the most current set in a later step. This is due to the registration process that we’ll go through when we setup Oinkmaster.</p> <pre><<line wrapped>> tar -zxvf snortules-snapshot-CURRENT.tar.gz -C /etc/snort</pre>

	Modify snort.conf	<p>Now let's modify our configuration file to reflect our network and needs:</p> <pre>vi /etc/snort/snort.conf</pre> <p>Change the internal network variable:</p> <pre>var HOME_NET 10.2.2.0/24</pre> <p>(make this whatever your internal or DMZ network is). For multiple networks, the syntax is: [10.2.2.0/24,192.168.1.0/24]</p> <p>Change the external network to mean everything except the internal networks defined above:</p> <pre>var EXTERNAL_NET !\$HOME_NET</pre> <p>Comment out the rule path variable with a # sign (Webmin cannot read the \$RULE_PATH variable – it takes it literally):</p> <pre>#var RULE_PATH /etc/snort/rules</pre> <p>Locate the <code>database</code> section and tell Snort to log to the mysql database (make sure this is all on one line). The password you create here you will need in a later step when setting up the Snort database:</p> <pre><<one big line wrap>> output database: log, mysql, user=snort password=your_password sensor_name=machine_name dbname=snort host=localhost</pre> <p>Remove all the \$RULE_PATH variables from rule paths at the end of the file and replace it with <code>rules</code>. Use the Find/Replace method as follows (type exactly as shown):</p> <pre>:%s@include \$RULE_PATH@include rules@g</pre> <p>This should change all the rule paths from:</p> <pre>include \$RULE_PATH/bad-traffic.rules</pre> <p>to this:</p> <pre>include rules/bad-traffic.rules</pre> <p>Save and close the file.</p>
--	--------------------------	---

Multiple Web Ports	<p>If you need to scan multiple ports for web hosts (say that you're running not only a public webserver on port 80, but also a server on port 8080 – you do not need to include HTTPS ports here like 443), then you need to use the following ugly hack. Snort (still) does not support port lists, so we'll have to run the web-rules once using the default of port 80, then re-define the HTTP_PORTS variable and run the web-rules again. Do this again for each additional port you may have.</p> <pre>vi /etc/snort/snort.conf</pre> <p>Page down to the bottom of the configuration file, where the rules are located. Find the group of rules that begin with <code>web-cgi.rules</code>. Now after the original 7 or so rule lines, change the HTTP_PORTS variable to your other web port, then copy and paste the same 7 or so rules again. You can repeat this as many times as necessary. For instance:</p> <pre>include rules/web-cgi.rules include rules/web-coldfusion.rules include rules/web-iis.rules include rules/web-frontpage.rules include rules/web-misc.rules include rules/web-client.rules include rules/web-php.rules #UGLY HACK for multiple HTTP ports - port 8080 var HTTP_PORTS 8080 include rules/web-cgi.rules include rules/web-coldfusion.rules include rules/web-iis.rules include rules/web-frontpage.rules include rules/web-misc.rules include rules/web-client.rules include rules/web-php.rules #UGLY HACK for multiple HTTP ports - port 8181 var HTTP_PORTS 8181 include rules/web-cgi.rules include rules/web-coldfusion.rules include rules/web-iis.rules include rules/web-frontpage.rules include rules/web-misc.rules include rules/web-client.rules include rules/web-php.rules</pre> <p>...and so on.</p> <p>Save and close the file.</p>
---------------------------	--

Snort Startup Options

√	Description
Edit Startup Options	<p>Let's set our startup options for Snort. The startup configuration file allows us to place various options within a file that we would in the past typically have put in the snort startup command. Open the startup configuration file for editing:</p> <pre>vi /etc/sysconfig/snort</pre> <p><u>Interface:</u> If you are using two interfaces, one for management and the other for Snort, ensure that the INTERFACE=ethx line is the Snort interface. INTERFACE=eth?</p> <p><u>Alert Mode:</u> When using BASE and/or PigSentry, the alertmode must be changed from the default fast to full. This ensures we log the full packet header information. ALERTMODE=full</p> <p><u>BPF Filter</u> The last section in this startup file has the Berkley Packet Filter file information. There may be times when you want to apply a filter in order to not alert on certain hosts and/or ports. Uncomment the BPFFILE=/etc/snort/bpf_file line and change it as follows: BPFFILE=/etc/snort/filters.bpf</p> <p>Save and close the file.</p> <p>Now, we need to create the filters.bpf file, or snort won't be able to start up. We don't need to actually have any filters yet, we just need to create an empty file. Do this with the touch command: touch /etc/snort/filters.bpf</p>

Setup MySQL Database

√	Description
Instructions	<p>Throughout the MySQL instruction, I will put a line with <code>mysql></code> in front of it so you will see what the output should be.</p> <p>Also note that in MySQL, a semi-colon <code>;</code> character is mandatory at the end of each input line – if you forget it, just type the <code>;</code> on the next line by itself.</p>

	<p>Create Database</p>	<p>Let's login to mysql (no password needed to start with) and set our local root password. Note that the root user in MySQL is not the same as the linux local root user.</p> <pre>mysql mysql>SET PASSWORD FOR root@localhost=PASSWORD('new_password'); >Query OK, 0 rows affected (0.25 sec) mysql>CREATE DATABASE snort; >Query OK, 2 rows affected (0.01 sec) mysql>EXIT</pre>
	<p>Delete Anonymous Logins</p>	<p>Now let's log back in with the password you just set:</p> <pre>mysql -p</pre> <p>Let's make sure we don't have other root users or unwanted users:</p> <pre>mysql>CONNECT mysql; >Current database: mysql mysql>SELECT user,host FROM user;</pre> <p>You will see something like this:</p> <pre>+-----+-----+ user host +-----+-----+ root localhost +-----+-----+ 2 rows in set (0.00 sec)</pre> <p>Uhoh! As seen above, mysql by default has blank user accounts – this means anyone (anonymous) can login. Let's fix this:</p> <pre>mysql>DELETE FROM user WHERE user=""; >Query OK, 2 rows affected (0.09 sec) mysql>DELETE FROM db WHERE user=""; >Query OK, 2 rows affected (0.10 sec) mysql>FLUSH PRIVILEGES; >Query OK, 2 rows affected (0.10 sec) mysql>SELECT user,host FROM user;</pre> <p>You should now see something like this:</p> <pre>+-----+-----+ user host +-----+-----+ root localhost +-----+-----+ 1 rows in set (0.00 sec)</pre> <p>Yeah! No anonymous users now.</p>

Delete Test Database	<p>Let's see what databases we have in our server:</p> <pre>mysql>SHOW DATABASES;</pre> <pre>+-----+ Database +-----+ mysql snort test +-----+</pre> <p>3 rows in set (0.00 sec)</p> <p>Egads! What's this? A test database? Hmmmm – not such a good thing to have on a production system. Let's delete the default test database that's installed by MySQL:</p> <pre>DROP DATABASE test;</pre> <p>Now check to see that the test database is gone:</p> <pre>mysql>SHOW DATABASES;</pre> <pre>+-----+ Database +-----+ mysql snort +-----+</pre> <p>2 rows in set (0.00 sec)</p> <p>You should now only see two databases, mysql and snort.</p> <pre>mysql>EXIT >Bye</pre>
-----------------------------	---

<p>Setup Snort Database</p>	<p>Now the database tables need to be set up. The setup scripts should be located in your <code>/usr/share/doc/snort-2.6.0/schemas</code> directory, however, at the time of this document the Snort RPM was not populating this directory with scripts – they are missing. So we’ll need to download the source file and get the scripts out of that instead. I’ve already done that and placed the database schema in the <code>./snortinstall/scripts</code> directory.</p> <pre>cd /root/snortinstall/scripts</pre> <p>Now execute the command to create the MySQL tables:</p> <pre>mysql -p < create_mysql snort</pre> <p>>Enter password:</p> <p>Now you need to check and make sure that the snort DB was created correctly:</p> <pre>mysql -p</pre> <p>>Enter password:</p> <p>Ok, next let’s make sure the tables are all there as well:</p> <pre>mysql>USE snort >Database changed mysql>SHOW TABLES;</pre> <pre>+-----+ Tables_in_snort +-----+ data detail encoding event icmphdr iphdr opt reference reference_system schema sensor sig_class sig_reference signature tcphdr udphdr +-----+</pre> <p>16 rows in set (0.00 sec)</p>
------------------------------------	---

<p>Setup Users</p>	<p>Now let's setup the user and password for remote connections from our snort sensor and for BASE deletions. Note that the password you use here is the <u>same one</u> you created and placed into the snort.conf file in an earlier step:</p> <pre>mysql>connect snort > Connection id: 44 > Current database: snort mysql>GRANT CREATE, INSERT, SELECT, DELETE, UPDATE ON snort.* TO snort@localhost IDENTIFIED BY 'password'; > Query OK, 0 rows affected (0.02 sec)</pre> <p>And here's the remote user and password that cannot delete alerts from the database and is used for querying via BASE or SAM only. You will need this username (console) and password for a later step when setting up BASE:</p> <pre>mysql>GRANT SELECT, INSERT, UPDATE ON snort.* TO console@localhost IDENTIFIED BY 'password';</pre> <p>It's always a good idea to flush privileges in order to re-read the grant tables when making user account changes. You don't necessarily need to do this step when using the grant command, but you do when updating the tables directly, when deleting users and when using the REVOKE command – so it's a good habit to be in whenever you make user changes:</p> <pre>>Query OK, 0 rows affected (0.00 sec) mysql>FLUSH PRIVILEGES; >Query OK, 0 rows affected (0.09 sec) mysql>EXIT >Bye</pre>
---------------------------	---

Test Snort

√	Description
<p>Verify Boot Startup</p>	<p>To check that snort is going to run at boot, issue the following command:</p> <pre>chkconfig --list snortd</pre> <p>You should see snort turned on at run-levels 2, 3, 4 and 5. If not, issued the command:</p> <pre>chkconfig snortd on</pre>
<p>Test Snort Configuration</p>	<p>At this point, the server is setup and ready to run Snort. To test the configuration file, simply <code>cd /etc/snort</code> and then issue the command <code>snort</code>. If there are any errors with starting snort, you will be able to see them on the screen. Otherwise, you should see an Initialization Complete notice and snort will be running. To quit, simply <code>[CTRL]+c</code>. (you can also execute <code>snort -T</code> to test the configuration file and immediately exit)</p>

		<p>If snort runs successfully, then delete the alert file that you created – because snort will not have access to the file you created just now running it manually while logged in as root.</p> <pre>rm -rf /var/log/snort/alert</pre> <p>If there are errors, begin troubleshooting. Google is your friend! 😊</p>
	<p>Test Snort Startup Script</p>	<p>If your configuration file works, then let's test Snort with the startup configuration file. To do this issue the command:</p> <pre>service snortd start.</pre> <p>Now check to see if snort stayed up after you launched it:</p> <pre>service snortd status</pre> <p>If it's running, great! But if you see something like <code>snort dead but subsys locked</code>, then you have an issue.</p> <p>If you get any errors, first make sure you deleted the <code>/var/log/snort/alert</code> file that you created when you manually ran snort for the first time. If you look at the file and see that it's owned by root, delete it and try running snort again.</p> <p>If you still get errors, then troubleshoot your <code>/etc/sysconfig/snort</code> file. Something you may find useful is to launch snort manually using all of the settings from the <code>/etc/sysconfig/snort</code> file. For instance, to test the alert mode with your configuration file, execute:</p> <pre>snort -A full -c /etc/snort/snort.conf</pre> <p>and see if there are errors. You can do this for all the settings in the startup script to see where the error is.</p> <p>Otherwise, if you have snort running successfully, go ahead and stop it:</p> <pre>service snortd stop</pre>

Install Prerequisites for BASE

√	Description
Install PHP-GD	<p>PHP-GD is used for creating and manipulating images with PHP.</p> <p>Run <code>up2date</code> to install php-gd, or if you haven't updated PHP and are still running PHP v4.3.9-3.9 from the installation CDs, then you can use the version of php-gd included in my snort installation package.</p> <pre>cd /root/snortinstall rpm -ivh php-gd-4.3.9-3.18.i386.rpm</pre>
Install ADODB	<p>Install the ADODB graphics library to the web directory:</p> <pre>cd /root/snortinstall tar -xvzf adodb462.tgz -C /var/www</pre>

Install BASE

√	Description
Install BASE	<p>Install the BASE installation into both the public web directory and a private web directory, then rename the directories from the version number to simply <code>base</code>:</p> <pre>cd /root/snortinstall mkdir /var/www/html/private tar -zxvf base-1.2.2.tar.gz -C /var/www/html/private tar -zxvf base-1.2.2.tar.gz -C /var/www/html cd /var/www/html mv base-1.2.6/ base cd /var/www/html/private mv base-1.2.6/ base</pre>
Configure BASE	<p>Using a web browser, let's first configure the private install of BASE: https://server_ip_address/private/base/setup</p> <p>You should get a message that says:</p> <p>Basic Analysis and Security Engine (BASE) Setup Program</p> <p>If there is an error about the config file not being writable, that's fine. We'll work around that. Click the <code>Continue</code> link.</p> <ol style="list-style-type: none"> 1. Select your <code>Language</code> as <code>English</code> (or whatever your preference is) 2. Set your <code>Path to ADODB</code> as <code>/var/www/adodb</code> 3. Click <code>Submit Query</code> 4. Select the <code>Database type</code> as <code>MySQL</code> 5. Set the <code>Database Name</code> to <code>snort</code> 6. Set the <code>Database Host</code> to <code>localhost</code> 7. Leave the <code>Database Port</code> blank

8. Set the Database User Name to `snort`
9. Set the Database Password to the `snort MySQL password` (this is the same password used in the snort config to connect to the database)
10. Leave the Use Archive Database de-selected
11. Click Submit Query
12. For the Use Authentication System, leave it unchecked and click Submit Query
13. Now click Create BASE AG
14. Verify all the statements in red are Successful, then click step 5
15. If your configuration is not writable by the web server (actually a good security measure) then copy the resulting configuration into a new `base_conf.php` file:

```
cd /var/www/html/private/base
vim base_conf.php
```

You're done with the private directory – now let's do the same for the public directory, only for the database user and password we'll use the console account so this install can only be used for viewing and items cannot be deleted.

16. Go to: `https://server_ip_address/base/setup`
17. Select your Language as `English` (or whatever your preference is)
18. Set your Path to ADODB as `/var/www/adodb`
19. Click Submit Query
20. Select the Database type as `MySQL`
21. Set the Database Name to `snort`
22. Set the Database Host to `localhost`
23. Leave the Database Port blank
24. Set the Database User Name to `console`
25. Set the Database Password to the `console MySQL password`
26. Leave the Use Archive Database de-selected
27. Click Submit Query
28. For the Use Authentication System, leave it unchecked and click Submit Query
29. Now click Create BASE AG
30. Verify the status shows DONE, then click step 5
31. If your configuration is not writable by the web server (actually a good security measure) then copy the resulting configuration into a new `base_conf.php` file.

```
cd /var/www/html/base
vim base_conf.php
```

Now when you go to `https://server_ip_address/base/` or `https://server_ip_address/private/base/` you should see the BASE homepage.

Secure Apache

√	Description
	<p>Explanation</p> <p>BASE actually comes with an authentication system for the application, however, there's little to no documentation on the roles and how they work. Instead, we'll use good old Apache .htaccess to protect the site.</p>
	<p>Password Protect Main Site</p> <p>Create the .htaccess that will protect the site.</p> <ol style="list-style-type: none"> Create a .htaccess file to control access <code>vi /var/www/html/base/.htaccess</code> Input the following information into the file <pre>AuthType Basic AuthName "BASE" AuthUserFile /var/www/.htpasswd require valid-user</pre> Create two users for the site by issuing the following command. Note that the "-c" is not used after the first user is creates (this switch initially creates the file): <pre><<line wrapped>> /usr/bin/htpasswd -c /var/www/.htpasswd snort > New password: (enter a password to use) /usr/bin/htpasswd /var/www/.htpasswd console > New password: (enter a password to use)</pre> Now we need to configure Apache to allow use of the .htaccess file: <code>vi /etc/httpd/conf/httpd.conf</code> Find the line <code><Directory /var/www/html></code>. About 20 lines after this is the line <code>AllowOverride None</code>. Change the <code>None</code> to <code>All</code>, as follows: <code>AllowOverride All</code> Let's also block index listings of the web server. The configuration line just above the <code>AllowOverride</code> is where you will see <code>Options Indexes FollowSymLinks</code>. Simply add a minus sign (-) in front of <code>Indexes</code> to forbid index listing, so it looks as follows: <code>Options -Indexes FollowSymLinks</code> Save and exit the file Restart Apache <code>service httpd restart</code>

<p>Password Protect Private BASE Directory</p>	<p>Now we'll secure the private directory where BASE has access to make deletions from the snort database. For this directory, we will only allow the snort user to login.</p> <pre><<line wrapped>> /usr/bin/htpasswd -c /var/www/.htpasswd-private snort >New password: (use the same snort password as for the main site)</pre> <p>Now, for you're learning pleasure, let's protect the private directory using the actual httpd.conf file rather than a .htaccess file.</p> <p>Edit the /etc/httpd/conf/httpd.conf file: <pre>vi /etc/httpd/conf/httpd.conf</pre></p> <p>Locate the section the section shown below:</p> <pre><Directory /> Options FollowSymLinks AllowOverride None </Directory></pre> <p>...and now add the following right below it:</p> <pre><Directory "/var/www/html/private"> AuthType Basic AuthName "Private BASE" AuthUserFile /var/www/.htpasswd-private require valid-user </Directory></pre> <p>Save and close the file.</p> <p>Now restart the Apache web server: <pre>service httpd restart</pre></p>
<p>Test Authentication</p>	<p>Using a browser or lynx, first go to: <pre>https://server_ip_address/base/</pre></p> <p>Are you prompted for credentials? Good! Try both sets – first the snort username & password, then close your browser (to clear the session authentication) and then try the console username & password.</p> <p>Next, go to <pre>https://server_ip_address/private/base/</pre></p> <p>When prompted for credentials use your snort username & password. We did not setup a console account for this private area where things can be deleted.</p>

	<p>Remove Default Page</p>	<p>Apache has a default index page that will be processed if you do not have an index page in the root directory. We really don't want to display this default page, so to remove this perform the following steps:</p> <pre>cd /etc/httpd/conf.d mv welcome.conf welcome.orig service httpd restart</pre> <p>You could also redirect the root to your public BASE install by placing an index.html file in the /var/www/html directory with the following content in the file:</p> <pre>vi /var/www/html/index.html</pre> <pre><meta http-equiv="REFRESH" content="0; URL=base/"></pre>
--	-----------------------------------	--

Install Prerequisite for Webmin

√	Description	
	<p>Install NetSSLeay</p>	<pre>cd /root/snortinstall tar -zxvf Net_SSLeay.pm-1.23.tar.gz cd Net_SSLeay.pm-1.23 unset LANG ./Makefile.PL -t</pre> <p>If you get an error stating: Warning: I could not locate your pod2man program. Please make sure, your pod2man program is in your PATH before you execute 'make'</p> <p>...then the <code>unset LANG</code> command did not work correctly. Try it again.</p> <p>Now, install it: <pre>make install</pre></p> <p>Test the install to ensure it works properly: <pre>perl -e 'use Net::SSLeay'</pre></p> <p>If there were no errors returned, then SSL has been setup properly for Webmin. <pre>cd ..</pre></p>

Install and Configure Webmin

√	Description
Install Webmin	<pre>cd /root/snortinstall rpm -ivh webmin-1.300-1.noarch.rpm</pre> <p>If there's an error "cannot identify OS", that is likely due to the /etc/issue being changed and this newer OS not being recognized. To work around this, add the line "Red Hat Linux release 9 (Shrike)" right before the Red Hat line in the /etc/issue file temporarily and run the rpm again. When the install completes remove the line.</p> <p>You should now be able to log (using root) into the Webmin console via a browser to https://server_ip_address:10000</p>
Configure Snort Plugin	<ol style="list-style-type: none"> 1. Open a browser and go to: https://snort_server:10000 2. Login as root 3. Select the Webmin Configuration icon 4. Select the Webmin Modules icon 5. Install the module from a local file <ol style="list-style-type: none"> a. <code>/root/snortinstall/snort-1.1.wbm</code> b. Click Install module 6. Select servers icon from the TOP of the web page 7. Select the Snort IDS Admin icon (it looks like a pig) 8. Select the Module Config tab in the left hand corner (if it doesn't come up automatically) 9. Set the configuration to match the following (lines are wrapped): <p>Full path to Snort executable (with options) = <code>/usr/sbin/snort -o -i eth1 -d -D -C -c /etc/snort/snort.conf</code></p> <p>Full path to Snort configuration file = <code>/etc/snort/snort.conf</code></p> <p>Full path to Snort rule files directory = <code>/etc/snort</code></p> <p>Full path to Snort PID file = <code>/var/run/snort_eth1.pid</code></p> <p>Command to start Snort (optional) = <code>/etc/rc.d/init.d/snortd start</code></p> <p>URL to ACID (optional) =</p> <p>When finished, click the save button and you're done!</p>

Install and Automate PigSentry

√	Description
Install PigSentry	<p>PigSentry is perl script that runs against the Snort alert log. It is used for real-time alerts, with a stable table of recent alerts to reduce the possibility of spamming yourself with emails. It will send a notice if there is a new alert, or if there is an increase in the general trend or pattern of existing alerts.</p> <p>To install PigSentry, we'll simply copy the perl script and initiate the proper startup script.</p> <pre><<line wrapped>> cp /root/snortinstall/scripts/pigsentry-1.2.pl /usr/local/bin/pigsentry <<line wrapped>> cp /root/snortinstall/scripts/gopigsentry /etc/rc.d/init.d chmod 755 /etc/rc.d/init.d/gopigsentry chkconfig --add gopigsentry</pre>
Configure PigSentry	<p>Now edit the <code>/etc/init.d/gopigsentry</code> file to change the email address as necessary:</p> <pre>vi /etc/init.d/gopigsentry</pre> <p>Locate the <code>your_email@your_domain.TLD</code> entry and change it to your email address.</p> <p>Save and close the file.</p>

Setup MySQL Database Dump and Backup

√	Description
MySQL Dump	<p>The MySQL snort database should be backed up in order to ensure the integrity of the data. As an example I've created a simple script that will backup, zip and copy a zipped copy of your MySQL snort database to a Windows server. It archives the prior 5 days worth of backups as well. Edit this, use it or don't use it as you see fit for your needs.</p> <p>To configure the nightly backup, perform the following:</p> <pre><<line wrapped>> cp /root/snortinstall/scripts/mysql_backup /usr/local/bin</pre> <p>Edit the <code>/usr/local/bin/mysql_backup</code> file and modify the six variables under the VARIABLES section.</p> <pre>vi /usr/local/bin/mysql_backup</pre> <p>Then edit the crontab file and add the following line to run it every night:</p> <pre>crontab -e 15 23 * * * /usr/local/bin/mysql_backup</pre>

Update Snort Rules Automagically Using Oinkmaster

√	Description	
Explanation	<p>We will use Oinkmaster to update and manage our rules. Oinkmaster is a perl script created to automate the process of downloading and merging Snort rules. Its homepage is http://oinkmaster.sourceforge.net/.</p> <p>Oinkmaster fetches Snort rules from the archive address specified in <code>oinkmaster.conf</code>, comments out the unwanted rules, and prints what rules have been changed since the last update. Unwanted rules are also specified in the <code>oinkmaster.conf</code> – this helps to specify rules that should never be included in the updated rulesets. It's a great way to automagically update your snort rules!</p> <p>The script can be run manually or as a cron job. We will set it up as a cron job, then verify the integrity of the rules (or rather, the proper syntax, since Snort will die if even one rule has the wrong syntax) by checking that Snort is still alive.</p>	
Obtain Oink Code	<p>As of March 2005 snort.org changed the way snort rules are distributed. You can still obtain the rules for free, but you must register and the rules will be released for free 5 days after paying subscribers can get them.</p> <ol style="list-style-type: none"> 1. Go to http://www.snort.org 2. Login to snort – if you don't have a registration, create one via the link 3. Once logged in, go to your User Preferences page 4. At the bottom of the page there will be a section labeled Oinkmaster Download Codes with an explanation on how to obtain and use the code with Oinkmaster 5. Click the Get Code button 6. An Oink Code will be generated for you. You will need this code configure Oinkmaster. 	
Install Oinkmaster	<pre>cd /root/snortinstall tar -xvzf oinkmaster-1.2.tar.gz cd oinkmaster-1.2 cp oinkmaster.pl /usr/local/bin cp oinkmaster.conf /usr/local/bin cd /root/snortinstall/scripts cp gooink /usr/local/bin cd /usr/local/bin chmod +x gooink</pre>	

	<p>Configure Oinkmaster</p>	<pre>vi oinkmaster.conf</pre> <p>First, under the General options section, locate the Example for Snort-current and unremark the url line below it that looks like this:</p> <pre>url = http://www.snort.org/pub-bin/oinkmaster.cgi/<oinkcode>/snortrules-snapshot-CURRENT.tar.gz</pre> <p>Replace the <oinkcode> with the Oink Code you were given when you registered at snort.org so it then looks like this (using an example oinkcode):</p> <pre>url = http://www.snort.org/pub-bin/oinkmaster.cgi/5a08f649c16a278e1012e1c84bdc8fab9a70e2a4/snortrules-snapshot-CURRENT.tar.gz</pre> <p>Finally, if there are any rules that you know you want disabled and not re-enabled from a new download, include them at the bottom of the file under disablesid – see the config file for syntax.</p> <p>Save and close the file.</p>
	<p>Configure gooink Script</p>	<p>The gooink script will fire off oinkmaster.pl to update your rules and test your rules, firing off an email to you for each phase. It will also email you a list of the rules so you can verify that they did indeed update from the timestamp.</p> <p>To set your email address, edit the <code>/usr/local/bin/gooink</code> script and set the EMAIL variable from <code>your_email@your_domain.TLD</code> to your real email address.</p> <pre>vi /usr/local/bin/gooink</pre>
	<p>Setup Cron Job</p>	<p>Setup the cron job to perform the updates:</p> <pre>crontab -e 00 12 * * * /usr/local/bin/gooink</pre>
	<p>Create Backup Directory and Test</p>	<p>Finally, create a new directory for the rules to be backed up to:</p> <pre>mkdir /etc/snort/old-rules</pre> <p>Oh, and you probably want to test it. Check your <code>/etc/snort/rules</code> directory for the current date on the rule files, then run <code>/usr/local/bin/gooink</code> and verify they changed.</p>

Watching the Watcher

√	Description	
	Explanation	<p>What happens if snort dies, whether through a server-side issue or through malicious intent? How will you know that it died, until days later when you run a report and find that there's a large gap where no alerts were logged?</p> <p>Well, we're smarter than the average bear! We'll configure a simple script to check if snort is alive. If it's not, we'll send an email and attempt to restart snort, check it again, and send a final email asking for help or stating that it's back up and running. Then we'll add the script to crontab and run it every 15 minutes.</p> <p>Well, lucky you, I've already created a script to do this. ☺</p>
	Install & Setup Cron Job	<pre>cp /root/snortinstall/scripts/test.sh /usr/local/bin</pre> <p>Set a cron job to run every 15 minutes to check snort</p> <pre>crontab -e 15 * * * * /usr/local/bin/test.sh</pre>
	Edit Configuration	<p>Now edit the script and change the EMAIL variable from <code>your_email@your_domain.TLD</code> to your real email address</p> <pre>vi /usr/local/bin/test.sh</pre>

Final Check

√	Description	
	Explanation	<p>Reboot your system and watch the boot process to make sure everything starts. When it comes up you can check to see if the various processes are running by issuing the command <code>ps -ef grep service</code>, where the service can be the process you're looking for, like mysql, httpd, snort, etc. To check all our important services at once, issue the following command:</p> <pre><<line wrapped>> ps -ef grep httpd && ps -ef grep mysql && ps -ef grep snort</pre> <p>Remember that you can always check snort itself by running it in interactive mode. If there are any errors with snort it will tell you immediately. Simply CD to the <code>/etc/snort</code> directory and run <code>snort</code>.</p>
	Attack!	<p>Now it's time to test your new snort box end to end. Use a scanner such as Nessus (http://www.nessus.org) and run it against the snort sensor box. Check BASE when you're done and it should have a bunch of alerts. If not, let the troubleshooting fun begin!</p> <p>Congratulations, you did it! You now have a fully functional IDS running and logging to a database and being viewed through a PHP script running on Apache.</p> <p>Good work and happy Snorting! ☺</p>

Adding Sensors

Install, Secure and Update Red Hat

√	Description
Install	<p>To add sensors, build and secure your linux boxes following the steps in the Install, Secure and Update Red Hat section, with the following caveats:</p> <p>During the package selection, make the following changes:</p> <ul style="list-style-type: none"> • <u>Do not</u> select the web server • For MySQL Database, accept <u>only</u> the defaults. We need only the client installed, not the server.
Secure	<p>For the security steps, do not open the ports for HTTP, HTTPS or Webmin. In other words, only do the following – for brevity’s sake, no explanations are given as they exist in the beginning of this document already:</p> <pre>iptables -F iptables -F INPUT iptables -F OUTPUT iptables -F FORWARD iptables -F -t mangle iptables -F -t nat iptables -X iptables -Z iptables -P INPUT DROP iptables -P FORWARD DROP iptables -P OUTPUT ACCEPT iptables -A INPUT -i lo -j ACCEPT iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT iptables -A INPUT -p ICMP -j ACCEPT iptables -A INPUT -p tcp --dport 22 -j ACCEPT iptables -A INPUT -p tcp --dport 10000 -j ACCEPT iptables -A INPUT -j DROP service iptables save service iptables restart iptables -L</pre>

Copy Snort Installation Files

√	Description
Copy Files	<p>Place the Snort File CD v2.00 CD in the coffee cup holder or download all the files from: www.thewestbrooks.com/downloads/snort-rhel4u3.tar.gz</p> <p>Then copy all files to /root/snortinstall, as follows:</p> <pre>mount /dev/cdrom /mnt/cdrom mkdir /root/snortinstall cp -r -v /mnt/cdrom/* /root/snortinstall cd /root/snortinstall chmod -R +wr /root/snortinstall/* umount /mnt/cdrom</pre> <p>OR</p> <pre>cd /root wget www.thewestbrooks.com/downloads/snort-rhel4u3.tar.gz tar -zxvf snort.tar.gz cd /root/snortinstall</pre>
Install PCRE	<pre>tar -zxvf pcre-5.0.tar.gz cd pcre-5.0 ./configure make make install cd ..</pre>

Install Snort

√	Description
Install Snort	<pre>cd /root/snortinstall mkdir /etc/snort mkdir /var/snort mkdir /var/log/snort rpm -ivh snort-2.6.0-1.RHEL4.i386.rpm rpm -ivh snort-mysql-2.6.0-1.RHEL4.i386.rpm</pre>
Install Rules	<p>We will use some pretty old rules to get snort up and running, but we'll be updating the rules to the most current set in a later step. This is due to the registration process that we'll go through when we setup Oinkmaster.</p> <pre><<line wrapped>> tar -zxvf snortules-snapshot-CURRENT.tar.gz -C /etc/snort</pre>

Modify snort.conf	<p>Now let's modify our configuration file to reflect our network and needs:</p> <pre>vi /etc/snort/snort.conf</pre> <p>Change the internal network variable:</p> <pre>var HOME_NET 10.2.2.0/24</pre> <p>(make this whatever your internal or DMZ network is). For multiple networks, the syntax is: [10.2.2.0/24,192.168.1.0/24]</p> <p>Change the external network to mean everything except the internal networks defined above:</p> <pre>var EXTERNAL_NET !\$HOME_NET</pre> <p>Comment out the rule path variable with a # sign (Webmin cannot read the \$RULE_PATH variable – it takes it literally):</p> <pre>#var RULE_PATH /etc/snort/rules</pre> <p>Locate the <code>database</code> section and tell Snort to log to the mysql database (make sure this is all on one line). The <code>host</code> is the Snort Console you've already built, and the <code>password</code> is the one you already use on your Snort Console box. The <code>sensor_name</code> should be something distinctive to this particular sensor so you can correlate reports properly – for instance, DMZ, Console, Vendor, Outside, etc.</p> <pre><<one big line wrap>> output database: log, mysql, user=snort password=your_password sensor_name=machine_name dbname=snort host=console_IP_address</pre> <p>Remove all the \$RULE_PATH variables from rule paths at the end of the file and replace it with <code>rules</code>. Use the Find/Replace method as follows (type exactly as shown):</p> <pre>:%s@include \$RULE_PATH@include rules@g</pre> <p>This should change all the rule paths from:</p> <pre>include \$RULE_PATH/bad-traffic.rules</pre> <p>to this:</p> <pre>include rules/bad-traffic.rules</pre> <p>Save and close the file.</p>
--------------------------	---

Multiple Web Ports	<p>If you need to scan multiple ports for web hosts (say that you're running not only a public webserver on port 80, but also a server on port 8080 – you do not need to include HTTPS ports here like 443), then you need to use the following ugly hack. Snort (still) does not support port lists, so we'll have to run the web-rules once using the default of port 80, then re-define the HTTP_PORTS variable and run the web-rules again. Do this again for each additional port you may have.</p> <pre>vi /etc/snort/snort.conf</pre> <p>Page down to the bottom of the configuration file, where the rules are located. Find the group of rules that begin with <code>web-cgi.rules</code>. Now after the original 7 or so rule lines, change the HTTP_PORTS variable to your other web port, then copy and paste the same 7 or so rules again. You can repeat this as many times as necessary. For instance:</p> <pre>include rules/web-cgi.rules include rules/web-coldfusion.rules include rules/web-iis.rules include rules/web-frontpage.rules include rules/web-misc.rules include rules/web-client.rules include rules/web-php.rules #UGLY HACK for multiple HTTP ports - port 8080 var HTTP_PORTS 8080 include rules/web-cgi.rules include rules/web-coldfusion.rules include rules/web-iis.rules include rules/web-frontpage.rules include rules/web-misc.rules include rules/web-client.rules include rules/web-php.rules #UGLY HACK for multiple HTTP ports - port 8181 var HTTP_PORTS 8181 include rules/web-cgi.rules include rules/web-coldfusion.rules include rules/web-iis.rules include rules/web-frontpage.rules include rules/web-misc.rules include rules/web-client.rules include rules/web-php.rules</pre> <p>...and so on.</p> <p>Save and close the file.</p>
---------------------------	--

Snort Startup Options

√	Description
<p>Edit Startup Options</p>	<p>Let's set our startup options for Snort. The startup configuration file allows us to place various options within a file that we would in the past typically have put in the snort startup command. Open the startup configuration file for editing:</p> <pre>vi /etc/sysconfig/snort</pre> <p><u>Interface:</u> If you are using two interfaces, one for management and the other for Snort, ensure that the INTERFACE=ethx line is the Snort interface.</p> <pre>INTERFACE=eth?</pre> <p><u>Alert Mode:</u> When using BASE and/or PigSentry, the alertmode must be changed from the default fast to full. This ensures we log the full packet header information.</p> <pre>ALERTMODE=full</pre> <p><u>BPF Filter</u> The last section in this startup file has the Berkley Packet Filter file information. There may be times when you want to apply a filter in order to not alert on certain hosts and/or ports. Uncomment the BPFFILE=/etc/snort/bpf_file line and change it as follows:</p> <pre>BPFFILE=/etc/snort/filters.bpf</pre> <p>Save and close the file.</p> <p>Now, we need to create the filters.bpf file, or snort won't be able to start up. We don't need to actually have any filters yet, we just need to create an empty file. Do this with the touch command:</p> <pre>touch /etc/snort/filters.bpf</pre>

MySQL User for Sensor

✓	Description
	<p data-bbox="285 275 444 302">Setup Users</p> <p data-bbox="488 275 1339 520">Now let's setup the user and password for remote connections from our various snort sensors. Note that the password you use here is the <u>same one</u> you created and placed into the snort.conf file in an earlier step. In fact, we're using the same username/password combination as we did when we originally built the Snort Console box, except now instead of <code>snort@localhost</code> we're setting up users for each individual sensor. We could open it up to the world, but we're security folks here, so we'll narrow it down to each individual sensor.</p> <p data-bbox="488 552 1273 611">Obviously to perform these steps you need to <u>go back to the Snort Console box</u> and login.</p> <pre data-bbox="583 674 1289 972">mysql -p >Enter password: mysql>connect snort > Connection id: 44 > Current database: snort mysql>GRANT CREATE, INSERT, SELECT, DELETE, UPDATE ON snort.* TO snort@sensor_IP_address IDENTIFIED BY 'password'; > Query OK, 0 rows affected (0.02 sec)</pre> <p data-bbox="488 1010 1317 1192">It's always a good idea to flush privileges in order to re-read the grant tables when making user account changes. You don't necessarily need to do this step when using the grant command, but you do when updating the tables directly, when deleting users and when using the REVOKE command – so it's a good habit to be in whenever you make user changes:</p> <pre data-bbox="583 1226 1174 1371">>Query OK, 0 rows affected (0.00 sec) mysql>FLUSH PRIVILEGES; >Query OK, 0 rows affected (0.09 sec) mysql>EXIT >Bye</pre>

IPTables Rule on Sensor

√	Description
Allow MySQL	<p>Ok, so now we setup the MySQL Server with an account that can login remotely from our new sensor. Now the Snort Console's iptables firewall needs to have a rule added to allow mysql connections through.</p> <p>Again, this step is obviously performed on the Snort Console box and is based on having setup iptables per these procedures.</p> <p>First, remove the last rule that drops everything that doesn't match any other rules:</p> <pre>iptables -D INPUT -j DROP</pre> <p>Next, add a rule to allow the MySQL port through:</p> <pre>iptables -A INPUT -p tcp --dport 3306 -j ACCEPT</pre> <p>Put the last drop rule back in:</p> <pre>iptables -A INPUT -j DROP</pre> <p>Finally, save and restart iptables, then check your rules:</p> <pre>service iptables save service iptables restart iptables -L</pre>

Test Snort

√	Description
Verify Boot Startup	<p>To check that snort is going to run at boot, issue the following command:</p> <pre>chkconfig --list snortd</pre> <p>You should see snort turned on at run-levels 2, 3, 4 and 5. If not, issued the command:</p> <pre>chkconfig snortd on</pre>
Test Snort Configuration	<p>At this point, the server is setup and ready to run Snort. To test the configuration file, simply <code>cd /etc/snort</code> and then issue the command <code>snort</code>. If there are any errors with starting snort, you will be able to see them on the screen. Otherwise, you should see an Initialization Complete notice and snort will be running. To quit, simply <code>[CTRL]+c</code>. (you can also execute <code>snort -T</code> to test the configuration file and immediately exit)</p> <p>If snort runs successfully, then delete the alert file that you created – because snort will not have access to the file you created just now running it manually while logged in as root.</p> <pre>rm -rf /var/log/snort/alert</pre> <p>If there are errors, begin troubleshooting. Google is your friend! ☺</p>

	<p>Test Snort Startup Script</p>	<p>If your configuration file works, then let's test Snort with the startup configuration file. To do this issue the command: <code>service snortd start.</code></p> <p>Now check to see if snort stayed up after you launched it: <code>service snortd status</code></p> <p>If it's running, great! But if you see something like <code>snort dead but subsys locked</code>, then you have an issue.</p> <p>If you get any errors, first make sure you deleted the <code>/var/log/snort/alert</code> file that you created when you manually ran snort for the first time. If you look at the file and see that it's owned by root, delete it and try running snort again.</p> <p>If you still get errors, then troubleshoot your <code>/etc/sysconfig/snort</code> file. Something you may find useful is to launch snort manually using all of the settings from the <code>/etc/sysconfig/snort</code> file. For instance, to test the alert mode with your configuration file, execute: <code>snort -A full -c /etc/snort/snort.conf</code> and see if there are errors. You can do this for all the settings in the startup script to see where the error is.</p> <p>Otherwise, if you have snort running successfully, go ahead and stop it: <code>service snortd stop</code></p>
--	---	--

Install Prerequisite for Webmin

√	Description	
	<p>Install NetSSLeay</p>	<pre>cd /root/snortinstall tar -zxvf Net_SSLeay.pm-1.23.tar.gz cd Net_SSLeay.pm-1.23 unset LANG ./Makefile.PL -t</pre> <p>Now, install it: <code>make install</code></p> <p>Test the install to ensure it works properly: <code>perl -e 'use Net::SSLeay'</code></p> <p>If there were no errors returned, then SSL has been setup properly for Webmin. <code>cd ..</code></p>

Install and Configure Webmin

√	Description
Install Webmin	<pre>cd /root/snortinstall rpm -ivh webmin-1.300-1.noarch.rpm</pre> <p>You should now be able to log (using root) into the Webmin console via a browser to <code>https://server_ip_address:10000</code></p>
Configure Snort Plugin	<ol style="list-style-type: none"> 1. Open a browser and go to: <code>https://snort_server:10000</code> 2. Login as <code>root</code> 3. Select the Webmin Configuration icon 4. Select the Webmin Modules icon 5. Install the module from a local file <ol style="list-style-type: none"> a. <code>/root/snortinstall/snort-1.1.wbm</code> b. Click Install module 6. Select servers icon from the TOP of the web page 7. Select the Snort IDS Admin icon (it looks like a pig) 8. Select the Module Config tab in the left hand corner (if it doesn't come up automatically) 9. Set the configuration to match the following (lines are wrapped): <p>Full path to Snort executable (with options) = <code>/usr/sbin/snort -o -i eth1 -d -D -C -c /etc/snort/snort.conf</code></p> <p>Full path to Snort configuration file = <code>/etc/snort/snort.conf</code></p> <p>Full path to Snort rule files directory = <code>/etc/snort</code></p> <p>Full path to Snort PID file = <code>/var/run/snort_eth1.pid</code></p> <p>Command to start Snort (optional) = <code>/etc/rc.d/init.d/snortd start</code></p> <p>URL to ACID (optional) =</p> <p>When finished, click the save button and you're done!</p>

Install and Automate PigSentry

√	Description
Install PigSentry	<p>PigSentry is perl script that runs against the Snort alert log. It is used for real-time alerts, with a stable table of recent alerts to reduce the possibility of spamming yourself with emails. It will send a notice if there is a new alert, or if there is an increase in the general trend or pattern of existing alerts.</p> <p>To install PigSentry, we'll simply copy the perl script and initiate the proper startup script.</p> <pre><<line wrapped>> cp /root/snortinstall/scripts/pigsentry-1.2.pl /usr/local/bin/pigsentry <<line wrapped>> cp /root/snortinstall/scripts/gopigsentry /etc/rc.d/init.d chmod 755 /etc/rc.d/init.d/gopigsentry chkconfig --add gopigsentry</pre>
Configure PigSentry	<p>Now edit the <code>/etc/init.d/gopigsentry</code> file to change the email address as necessary:</p> <pre>vi /etc/init.d/gopigsentry</pre> <p>Locate the <code>your_email@your_domain.TLD</code> entry and change it to your email address.</p> <p>Save and close the file.</p>

Update Snort Rules Automagically Using Oinkmaster

√	Description
Install Oinkmaster	<pre>cd /root/snortinstall tar -xvzf oinkmaster-1.2.tar.gz cd oinkmaster-1.2 cp oinkmaster.pl /usr/local/bin cp oinkmaster.conf /usr/local/bin cd /root/snortinstall/scripts cp goink /usr/local/bin cd /usr/local/bin chmod +x goink</pre>

	<p>Configure Oinkmaster</p>	<pre>vi oinkmaster.conf</pre> <p>First, under the General options section, locate the Example for Snort-current and unremark the url line below it that looks like this:</p> <pre>url = http://www.snort.org/pub-bin/oinkmaster.cgi/<oinkcode>/snortrules-snapshot-CURRENT.tar.gz</pre> <p>Replace the <oinkcode> with the Oink Code you were given when you registered at snort.org so it then looks like this (using an example oinkcode):</p> <pre>url = http://www.snort.org/pub-bin/oinkmaster.cgi/5a08f649c16a278e1012e1c84bdc8fab9a70e2a4/snortrules-snapshot-CURRENT.tar.gz</pre> <p>Finally, if there are any rules that you know you want disabled and not re-enabled from a new download, include them at the bottom of the file under disablesid – see the config file for syntax.</p> <p>Save and close the file.</p>
	<p>Configure gooink Script</p>	<p>The gooink script will fire off oinkmaster.pl to update your rules and test your rules, firing off an email to you for each phase. It will also email you a list of the rules so you can verify that they did indeed update from the timestamp.</p> <p>To set your email address, edit the <code>/usr/local/bin/gooink</code> script and set the EMAIL variable from <code>your_email@your_domain.TLD</code> to your real email address.</p> <pre>vi /usr/local/bin/gooink</pre>
	<p>Setup Cron Job</p>	<p>Setup the cron job to perform the updates – note that you need to stagger the times for all your sensors, as the code only allows you to pull updates once every X (not sure what X is, but it's minutes, not hours). I suggest setting your sensors about 1 hour apart. The syntax for the time is:</p> <pre>MIN(0-59) HOUR(0-23) DAY_OF_MONTH(1-31) MONTH(1-12) DAY_OF_WEEK(0-6 where 0=Sunday) Command to be executed</pre> <pre>crontab -e</pre> <pre>00 01 * * * /usr/local/bin/gooink</pre>
	<p>Create Backup Directory and Test</p>	<p>Finally, create a new directory for the rules to be backed up to:</p> <pre>mkdir /etc/snort/old-rules</pre> <p>Oh, and you probably want to test it. Check your <code>/etc/snort/rules</code> directory for the current date on the rule files, then run <code>/usr/local/bin/gooink</code> and verify they changed.</p>

Watching the Watcher

√		Description
	Install & Setup Cron Job	<pre>cp /root/snortinstall/scripts/test.sh /usr/local/bin</pre> <p>Set a cron job to run every 15 minutes to check snort</p> <pre>crontab -e 15 * * * * /usr/local/bin/test.sh</pre>
	Edit Configuration	<p>Now edit the script and change the EMAIL variable from your_email@your_domain.TLD to your real email address</p> <pre>vi /usr/local/bin/test.sh</pre>

Final Sensor Tuning

You will want to tune your IDS more specifically for your environment. This is an important step and you should know how to do this yourself, or bring in a consultant to help. While specifics on what to tune is outside the scope of this particular document, here is some basic information on performing the tuning process.

Using Webmin and the snort plugin makes it somewhat easier to tune the configuration file for snort. You do this by logging into the main Snort Console, making the necessary changes there, then pushing the files out to all the sensors, and restarting snort on all the sensors.

- `https://server_ip_address:10000`
- Select the “**Servers**” icon from the top of the screen
- Select the “**Snort**” icon (looks like a pig)

You will now be presented with a screen that allows you to control most aspects of your sensor. In the center of your screen you will see your rule files:

Rulesets

✓ = Enabled ✗ = Disabled

Rule Set	Status	Action	Rule Set	Status	Action	Rule Set	Status	Action
rules/attack-responses	✓	Disable	rules/misc	✓	Disable	rules/smtp	✓	Disable
rules/backdoor	✗	Enable	rules/multimedia	✗	Enable	rules/snmp	✓	Disable
rules/bad-traffic	✓	Disable	rules/mysql	✓	Disable	rules/sql	✓	Disable
rules/chat	✗	Enable	rules/netbios	✓	Disable	rules/telnet	✓	Disable
rules/ddos	✓	Disable	rules/nntp	✓	Disable	rules/tftp	✓	Disable
rules/dns	✓	Disable	rules/oracle	✓	Disable	rules/virus	✓	Disable
rules/dos	✓	Disable	rules/other-ids	✓	Disable	rules/web-attacks	✗	Enable
rules/experimental	✓	Disable	rules/p2p	✗	Enable	rules/web-cgi	✓	Disable
rules/exploit	✓	Disable	rules/policy	✗	Enable	rules/web-client	✓	Disable
rules/finger	✓	Disable	rules/pop2	✓	Disable	rules/web-coldfusion	✓	Disable
rules/ftp	✓	Disable	rules/pop3	✓	Disable	rules/web-frontpage	✓	Disable
rules/icmp	✓	Disable	rules/porn	✗	Enable	rules/web-iis	✓	Disable
rules/icmp-info	✗	Enable	rules/rpc	✓	Disable	rules/web-misc	✓	Disable
rules/imap	✓	Disable	rules/rservices	✓	Disable	rules/web-php	✓	Disable
rules/info	✗	Enable	rules/scan	✓	Disable	rules/x11	✓	Disable
rules/local	✓	Disable	rules/shellcode	✗	Enable	somefile	✗	Enable

Let’s take a look at the DNS rules first. Simply click on `rules/dns` it and you will see a screen like this where you can edit the various DNS rules:

Current Rules in /etc/snort/rules/dns.rules			
Rule	Signature	Status	Action
1	alert tcp \$EXTERNAL_NET any -> \$HOME_NET 53 (msg:"DNS zone transfer TCP"; flow:to_server,established; content:" 00 00 FC "; offset:15; reference:arachnids,212; reference:cve,1999-0532; reference:nessus,10595; classtype:attempted-recon; sid:255; rev:13;)	✓	Disable Edit Delete
2	alert udp \$EXTERNAL_NET any -> \$HOME_NET 53 (msg:"DNS zone transfer UDP"; content:" 00 00 FC "; offset:14; reference:arachnids,212; reference:cve,1999-0532; reference:nessus,10595; classtype:attempted-recon; sid:1948; rev:6;)	✓	Disable Edit Delete
3	alert tcp \$EXTERNAL_NET any -> \$HOME_NET 53 (msg:"DNS named authors attempt"; flow:to_server,established; content:" 07 authors"; offset:12; nocase; content:" 04 bind 00 "; offset:12; nocase; reference:arachnids,480; reference:nessus,10728; classtype:attempted-recon; sid:1435; rev:7;)	✓	Disable Edit Delete
4	alert udp \$EXTERNAL_NET any -> \$HOME_NET 53 (msg:"DNS named authors attempt"; content:" 07 authors"; offset:12; nocase; content:" 04 bind 00 "; offset:12; nocase; reference:arachnids,480; reference:nessus,10728; classtype:attempted-recon; sid:256; rev:6;)	✓	Disable Edit Delete
5	alert tcp \$EXTERNAL_NET any -> \$HOME_NET 53 (msg:"DNS named version attempt"; flow:to_server,established; content:" 07 version"; offset:12; nocase; content:" 04 bind 00 "; offset:12; nocase; reference:arachnids,278; reference:nessus,10028; classtype:attempted-recon; sid:257; rev:9;)	✓	Disable Edit Delete
6	alert udp \$EXTERNAL_NET any -> \$HOME_NET 53 (msg:"DNS named version attempt"; content:" 07 version"; offset:12; nocase; content:" 04 bind 00 "; offset:12; nocase; reference:arachnids,278; reference:nessus,10028; classtype:attempted-recon; sid:1616; rev:7;)	✓	Disable Edit Delete
7	alert udp \$EXTERNAL_NET 53 -> \$HOME_NET any (msg:"DNS SP00F query response PTR with TTL of 1 min. and no authority"; content:" 85 80 00 01 00 01 00 00 00 00 "; content:" 0C 0C 00 0C 00 01 00 00 00 <100 0F "; classtype:bad-unknown; sid:253; rev:4;)	✓	Disable Edit Delete
8	alert udp \$EXTERNAL_NET 53 -> \$HOME_NET any (msg:"DNS SP00F query response with TTL of 1 min. and no authority"; content:" 81 80 00 01 00 01 00 00 00 00 "; content:" 0C 0C 00 01 00 01 00 00 00 <100 04 "; classtype:bad-unknown; sid:254; rev:4;)	✓	Disable Edit Delete

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

1. **Rule:** Just the order in which the rule appears in the rule file;
2. **Signature:** This is what an actual snort signature looks like;
3. **Status:** Is the rule enabled or disabled?;
4. **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 that any time you make changes to rules, you will need to restart your snort daemon (`service snortd restart`) for the changes to take effect.

The most basic tuning of your sensor might be to simply disable all the rulesets that have nothing to do with what you're protecting. For instance, if you're not running a mail server, you could disable the pop2, pop3 and smtp rulesets.

Filtering Rules:

Filtering enables us to make exceptions to 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 #4 from above. 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:

```
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;)
```

Let's say on your sensors that it is normal for host 192.168.55.23 to perform DNS zone transfers with 192.168.12.5

Highlight the rule 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 you own signatures and for filtering, and it will not be overwritten when you download current rulesets from snort.org.

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	ftp	✓	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						

Once you're in the local rules file, paste the rule you just copied into the **Add Rule** box at the bottom of the screen.

BPF Filters:

Another way to perform filtering is to use a Berkley Packet Filter to drop packets at the BPF interface before they ever get to Snort. This saves on processing power and speeds up Snort as it never actually sees those packets.

During our installation, we created a line in the config file to use a BPF filter file. We configured it to look to `/etc/snort/filters.bpf`. To obtain filter syntax or to find various ways to use the filter, look for resources on snort.org. For our immediate tweaking, we can use the filter file to ignore particular hosts – helpful for ignoring internal assessment hosts that create a lot of activity and alerts that you will not respond to, as well as ignoring external scans from legitimate sources, such as Microsoft.

As an example, to ignore all traffic coming from 192.168.0.1 and 10.2.20.30, edit the `/etc/snort/filters.bpf` file and add the following line:

```
vi /etc/snort/filters.bpf
    not (host 192.168.0.1 or host 10.2.20.30)
```

Here is another example using multiple filters in the filters.bpf file:

```
not (host xxx.xxx.79.243 or host xxx.xxx.81.38 or host xxx.xxx.79.246)
and not (src host xxx.xxx.101.240 and dst host xxx.xxx.179.58)
and not (src host xxx.xxx.25.101 and dst port 21)
and not ((src host xxx.xxx.101.233 and src port 1521) and dst host
xxx.xxx.179.95)
and not (src host xxx.xxx.2.102 and dst port 161)
and not ((src host xxx.xxx.179.70) and (dst host xxx.xxx.239.50 and dst
port 135))
```