Global Information Assurance Certification Paper

Brian A. Hughes

GIAC Security Essentials Certification

Practical Assignment Version 1.4c

Option 1 – Research on Topics in Information Security

11 March 2005

Snort, Barnyard, MySQL and SGUIL on Fedora Core 3: A Rookie's Cookbook

Abstract

Snort is the most popular Open Source Network Intrusion Detection System (NIDS) in existence today. Snort is widely deployed and has a tremendous following. While much has been written about Snort and many sources of information exist, there are few concise guides available aimed at an audience new to Snort. Since I too am relatively new to snort I can hopefully bring my inexperience to the table in order to fill that niche, providing one stop shopping for information on installation and configuration.

In keeping with the open source flavor that helped lead to Snort's popularity I will concentrate on utilizing only open source software in this guide. In order to take advantage of the most current features of both Snort and the underlying OS this guide will focus on the latest software available at the time of this writing, namely Snort 2.3.2, Barnyard 0.2.0, MySQL 4.1.10, Squil 0.5.3, and Fedora Core 3.

Style

I should also note that it is my belief that the primary reason to write is to communicate ideas to other people as clearly as possible. I will use a concise simple style, so you won't have to read this with a dictionary by your side while I try and impress you with all the multi-syllable words that I can muster. Nor will I inflict long passages of flowery prose on you when fewer words will do. My goal is to save you some time installing your NIDS not to put you in a comma.

I should also warn you that there may be a few isolated pieces of humor that have slipped into this document. If your philosophy is that humor has no place in a technical document and you find it offensive, I'm certain that a well piloted magic marker could strike the offending sentences and still leave the remainder of the document fairly useful.

Notation

Throughout this document you will see some reoccurring strings that will act as notation they are:

shell$ - denotes a regular bash shell
shell# - denotes a root shell
mysql> - denotes a MySQL prompt

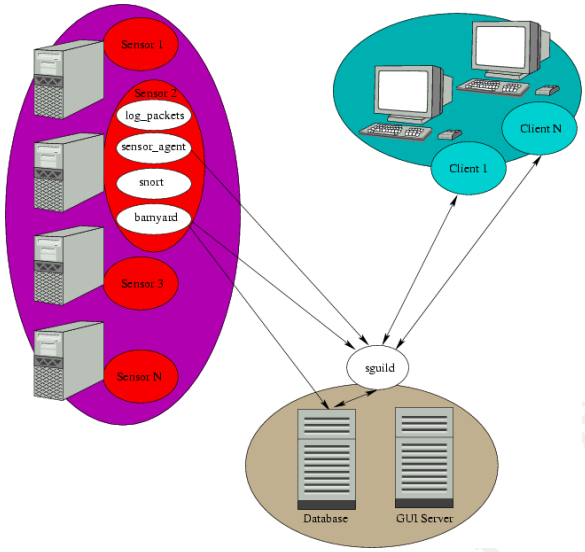
These stings are not part of the commands and should not be typed. They are merely there to show the context in which the commands are entered. Commands that are longer then the width of the page will have a \ on the end and be continued on the next line. When you utilize these commands you my omit the \ and keep typing on a single line or type then as they appear whichever you prefer. Some of the long lines that contain very long strings without spaces simply wrap, but hopefully by noting where the next shell$ or shell# prompt appears you will be able to determine where one command stops and another begins. You may also notice a symbolic link here and there. While these may look innocent enough please pay close attention to them as they are necessary for packages that follow to compile cleanly.

A quick word about the implementation

I work in a university environment and while by some standards it is a smaller institution with approximately 15,000 students and 2,000 faculty and staff, I was concerned that my implementation would be able to properly scale.

One of the most important aspects of intrusion detection is to see all the packets without dropping any of them because the sensor is too busy doing something else. Something that could easily slow the sensor down is outputting alerts. Jack Koziol notes in his book Intrusion Detection with Snort, "The unified output plugin is designed specifically for speed. It is the fastest possible method of outputting Snort intrusion data." (67). Since I'm interesting in Snort spending as little time as possible outputting the alerts and as much time as possible watching traffic this sounds like a very good thing, but what do we do with the data after the unified plugin gets a hold of it? Here is where Barnyard comes into play. In Snort 2.1: Intrusion Detection the authors note that "Barnyard was developed to separate the various output-processing tasks from the more time critical task of monitoring network traffic." (Baker et al. 531)

I was also concerned that as I added more sensors to watch the multiple points of interest in my network that ability of the software to keep up would be overrun, so I decided to break things out into separate hosts. It was around that time that I found the following diagram (sguil_arch.png) describing the SGUIL architecture which seemed to fit the bill perfectly.



Utilizing this architecture I could run each of my sensors on a separate machine, a MySQL server on a different machine, a SGUIL server on still another machine, and finally a client on yet a fourth machine.

I can almost hear people asking, “Why SGUIL and not ACID?” Yes, it’s true that you can also break the functions into separate platforms and still use ACID, but in my limited trials I found that ACID response times became unacceptable as the MySQL database grew in size. This could have been a matter of tuning on my part, since I’m fairly new to this, but the database definitely had a tendency to grow faster then I could keep up. Besides the allure of performance remaining fairly stable even with larger databases SGUIL also held the promise of combining several other tools into one console.

Those Who Have Gone Before

Since this is a cookbook of sorts I should note that many good recipes are the result of personalizing someone’s existing recipe and adapting it to one’s own tastes; such is the case here. Richard Bejtlich has done a fine job in his online installation guide, but it’s a little BSD’ish for my taste, so while I will reference his installation guide as a jumping off point I will focus on a Fedora Core 3 platform and the idiosyncrasies I've found to make things work there.

In addition to a focus on Fedora Core 3 I was also unable to find any installation guides that used Fedora Core 3 and SGUIL. I was also unable to find an installation guide that left SELinux enabled. Finally I was unable to find any single guide that would carry me through the entire process, so I hope that gathering all the pieces into one place will save others some time.

A Word About Machine Names

I'm going to invoke my vivid imagination and name these four machines Sensor, MySQLServer, SGUILServer and SGUILClient. (It's probably just as well I don't have children. Kid1, Kid2 etc. would probably not cut it.) You may of course select any host names that you prefer and then modify the commands as appropriate.

Installing Fedora Core 3

With the exception of the SGUIL client I'm going to install Fedora the same on all of the other machines. The only reason the SGUIL client differs is that in addition to being a SGUIL client you may want to use this machine for other things. In my case the box running the SGUIL client is my main workstation, so I will not be stripping it down to basics like the boxes we are configuring below, but instead I choose to install it as a workstation and leave the default software alone.

Some of the machines in my configuration are rack mounts that share a switch box and a HP TFT5600 RKM monitor and for some reason the graphical install of Fedora has issues with this monitor, so all of my instructions are going to be based on the text based install of Fedora Core 3. To get started boot from the Fedora Core 3 disk 1 CD and begin the text based installation by typing:

linux text

The first screen you will see asks if you would like to test the CD Media. If you are using untested CDs I would highly recommend that you take the extra time to test the media and replace any CDs that fail. While this may initially seem like a waste of time in the long run you'll be far ahead. Troubleshooting a bizarre problem down the road that was caused by a corrupt package is guaranteed to take longer then the media tests. If you have already tested the media select:

Skip

You'll now see the "Welcome to Fedora Core!" screen. Now select:

OK

The next screen displayed is "Language Selection". Here I'm going to select:

English

OK (to select OK press the tab key)

"Keyboard Selection" is now displayed. Select:

us

OK

You may or may not see a "System to Upgrade" screen. This will only be displayed on machines that Fedora finds a previous installation on. If this screen does appear a fresh installation may be accomplished by selecting:

Reinstall System

OK

"Installation Type" is the next screen to display. On this screen select:

Personal Desktop

OK

Now “Disk Partitioning Setup” will appear, so select:

Autopartition

“Automatic Partitioning” now asks how to handle any existing partitions. I am assuming that this is a dedicated machine, so select:

Remove all partitions on this system

OK

If you are sure respond Yes to the “Warning” window.

You will now see a window titled “Partitioning” accept the default file system layout by choosing:

OK

“Boot Loader Configuration” now asks which boot loader to use. Accept the default of GRUB by selecting:

OK

“Boot Loader Configuration” asks of any special options need to be passed to the kernel at boot time. Typically this is not the case so select:

OK

“Boot Loader Configuration” now asks if a GRUB Password is desired. Since my configuration has these machines in a physically secured environment I am going to opt for no GRUB password by simply selecting:

OK

“Boot Loader Configuration” offers the option of booting other operating systems in addition to Fedora, but since this is a dedicated system again the default may be selected by choosing:

OK

“Boot Loader Configuration” now asks where the boot loader should be installed. Most people will select the default Master Boot Record by selecting:

OK

Next “Network Configuration for eth0” is displayed. By default it wants to “Configure using DHCP” and “Activate on boot”. While activate on boot is desirable, configure using DHCP is not, so remove the asterisk from DHCP by pressing the space bar. Once the asterisk is removed you will then be able to manually configure “IP Address” and “Netmask” by tabbing to those fields. Save your entries by selecting:

OK

This is probably also a good time to mention that for machines configured as sensors I like to run dual Ethernet NICs. One of them I configure with an address like we just did for eth0, but the other I will leave in a down state. In other words I will uncheck both “Configure using DHCP” and “Activate on boot”. The reason for this is that I’ll use the interface with the IP address on a secure net to communicate with the sensor and I’ll use the NIC without an IP address as my monitor port.

Since we opted for manual configuration of the network the next section of entries “Miscellaneous Network Settings” needs to be supplied. Once the

parameters have been supplied save your selections by choosing:

OK

The “Hostname Configuration” screen asks if we would like hostname configuration via DHCP or manual configuration. Leave this field at the default setting of manual and supply the fully qualified host name (example host.your.domain.) Save your settings by selecting:

OK

The “Firewall” screen now asks if whether to enable the firewall or not. While we will latter need to do some additional adjusting of the firewall settings on some of these machines for now select:

Customize

The “Customize Firewall Configuration” screen displays. Tab to “Remote Login (SSH)” Enable SSH by pressing the space bar so that an asterisk appears. Save your settings by selecting:

OK

You are now returned to the main “Firewall” screen. Make sure that “Enable firewall” has an asterisk in front of it and then select:

OK

“Security Enhanced Linux” also known as SELinux is the next window. This will need to be adjusted on the MySQL Server latter, but it is desirable to have these systems as hardened as possible, so for now let’s leave this at the default of “Active” by simply choosing:

OK

The next window displayed is “Language Support” which we will leave at the default of “English (USA)” by choosing:

OK

“Time Zone Selection” displays next. Ultimately it is best to have our sensors and servers use UTC. This avoids the two times per year that we switch from Daylight Savings Time to Standard Time and visa versa. To date I have been unable to configure this during installation but latter in the document I will cover how to insure that your machine is using UTC. For now place an asterisk in front of “System clock uses UTC” and select whatever time zone is appropriate for your location. Save your settings by choosing:

OK

Next “Choose Root Password” screen is displayed. Remember to use the generally agreed upon standard of at least 8 characters. Utilize upper and lower case letters, numbers, and special characters and insure that your password does not contain any words found in the dictionaries of any language. Once you have supplied the password and confirmed it save your selection by choosing:

OK

“Package Defaults” is the next screen displayed. Here we have the chance to customize the software selection that will be installed on our system. Take advantage of this opportunity by placing an asterisk in front of “Customize software selection.” Save the setting by selecting:

OK

The “Package Group Selection” window is now displayed. Remove the

asterisks from in front of the following packages: (Remember this is done for the Sensor, MySQLServer and SGUILServer, but not for the SGUILClient.)

X Window System
Gnome Desktop Environment
Graphical Internet
Office/Productivity
Sound and Video
Graphics
Games and Entertainment
Printing Support (unless for some reason you need this)

To make sure that we have a compiler to build some of the packages we'll need latter you will need to place an asterisk in front of:

Development Tools

Save your settings and continue by choosing:

OK

The "Installation to begin" window now appears reminding us that we can find a complete log of the installation process in /root/install.log. Continue by selecting:

OK

A window titled "Required Install Media" tells us that we will need Fedora Core 3 CD's 1-3. Proceed by selecting:

Continue

"Change CDROM" screens will then prompt you to supply Fedora Core disc 2 and disc 3 as the installation progresses. The next screen you see will be titled "Complete" and will ask you to remove any installation media and press enter to reboot.

After a successful reboot you should now be at a "login:" prompt. Type "root" and press enter. At the "Password:" prompt supply the password that you selected earlier in the installation. To insure that the system is up to date run the Yellowdog Updater Modified also known as yum. In order to run the updates in text mode using yum, one first must issue the command:

shell# rpm --import /usr/share/doc/fedora-release-3/RPM-GPG-KEY*

Now we can apply any updates by typing:

shell# yum update

After some time yum will determine all the packages that need to be updated, any dependencies to resolve, and finally any other transactions such as obsolete packages to remove. You will then be asked if it is ok to proceed. Respond "y" for yes. Yum will now download all of the necessary packages (198 of them at the time of this writing). The time it takes will of course vary depending on available bandwidth. The last line displayed will read "Complete!" followed by a root shell prompt.

Now that the OS has been updated the next step is to create a user account. Some of the software that we will ultimately be running on these systems does not need to run as root, but instead will run as this user. We will also secure the system so that the user root cannot ssh into the system directly, so this user will be used for both of those purposes. We will call our new user

"sguil." We will add the new user and assign it a password with the following commands:
shell# useradd sguil
shell# passwd sguil (remember out discussion about strong passwords)

Since multiple levels of security are always desirable, it's probably a good idea to configure SSH so that someone cannot ssh directly in as the root user. Patrick Harper's Snort Install Manual contains the following directions on securing SSH:

> In the /etc/ssh/sshd_config file change the following lines (if it is commented out remove the #)
> Protocol 2
> PermitRootLogin no
> PermitEmptyPasswords no
> Save the file and type "service sshd restart". ssh (sic) will restart, enacting these changes. (You will need to SSH into the box with the user account you created after this, as root will no longer be accepted. Just "su -" to the root account) (9).

Time

While a shift in the time space continuum might be a great basis for a Star Trek episode, it's not too great when you are trying to correlate events. In Insrusion Detection with Snort, Koziol notes:

> Large organizations, particularly ISPs, can have a tremendous number of users or customers making user of the same pool of IP addresses. These addresses can be reassigned to different persons several times in the space of a minute. If your IDS has not accurately synced time with a reliable external time server, the external party cannot positively determine who was using the offending IP address at the requested time.
>
> Inaccurate time can make it impossible to link an exact person to the attacking IP address. Although you mat be able to determine that it was very likely that your attacker was in possession of the attacking IP address, you introduce inconsistencies in your evidence if you do not have perfectly accurate time. This could be problematic if you were to bring legal action against the suspected attacker (107).

To that I can only add that inaccurate time also damages your credibility with other ISPs. If you send them reports with inaccurate time they will be less likely to take future reports seriously.

The answer to accurate time is Network Time Protocol NTP. NTP is installed by default, but is not running by default. If you do nothing to alter the configuration once NTP is started it will sync with time servers at ntp.org which is fine, but if your institution has designated time servers you may want to alter the configuration (see /etc/ntp.conf and the files in the /etc/ntp directory) to sync

with those instead. To start NTP type:
shell# /etc/init.d/ntpd start
To insure that NTP starts whenever the machine is booted type:
shell# chkconfig --level 2345 ntpd on

The other hot topic related to time is time zone. With the exception of Arizona the United States switches between Standard Time and Daylight Savings Time. This twice per year switch plays havoc with intrusion logs, so why not switch to Universal Time Coordinated also know as UTC? If there is not a way to select UTC during the Fedora installation there certainly should be, but I was unable to find it. With the aid of my best research friend Google I was able to find http://wiki.ehow.com/Change-the-Timezone-in-Linux (Derouin).After reading this pages I was able to determine that with the following commands will allow you to switch your time zone to UTC:
shell# rm -rf /etc/localtime
shell# ln -s /usr/share/zoneinfo/UTC /etc/localtime
I would highly recommend setting up all the machines in your NIDS group to use NTP and use UTC as their time zone.

Setting up the Sensor

First assemble all the programs the sensor will need. All of the necessary files will be stored in /usr/local on the sensor machine, so before starting the download process I'm going to navigate there by typing cd /usr/local.

We're looking for MySQL 4.1.10-0 client programs, MySQL 4.1.10-0 libraries and header files, Perl Compatible Regular Expressions, Snort, Barnyard, Security Analyst Connection Profiler, and finally the SGUIL sensor software. These programs may be found by issuing the respective wget commands that appear below:
shell# wget http://mysql.osuosl.org/Downloads/MySQL-4.1/MySQL-client-4.1.10-0.i386.rpm
shell# wget http://mysql.osuosl.org/Downloads/MySQL-4.1/MySQL-devel-4.1.10-0.i386.rpm
shell# wget http://easynews.dl.sourceforge.net/sourceforge/pcre/pcre-5.0.tar.gz
shell# wget http://www.snort.org/dl/current/snort-2.3.2.tar.gz
shell# wget http://www.snort.org/dl/barnyard/barnyard-0.2.0.tar.gz
shell# wget http://easynews.dl.sourceforge.net/sourceforge/tcl/tcl8.4.9-src.tar.gz
shell# wget http://www.metre.net/files/sancp-1.6.1.tar.gz
shell# wget http://easynews.dl.sourceforge.net/sourceforge/sguil/sguil-sensor-0.5.3.tar.gz

If for some reason you would like the ability to select different mirrors or would like to navigate to each of the separate sites with your browser you may instead use the following URLs:
http://dev.mysql.com/downloads/mysql/4.1.html (MySQL).
http://prdownloads.sourceforge.net/pcre/pcre-5.0.tar.gz (PCRE).
http://www.snort.org/dl/ (Snort downloads).
http://prdownloads.sourceforge.net/barnyard/barnyard-0.2.0.tar.gz (Barnyard).

http://www.metre.net/sancp.html (Curry).
http://sguil.sourceforge.net/index.php?page=download (Visscher).

Unpack and install the sensor software. (Again this assumes a current working directory of /usr/local and that it is the recipient directory for all of the downloaded packages.) We will start with MySQL client programs, libraries and header files.

```
shell# rpm -Uvh MySQL-client-4.1.10-0.i386.rpm
shell# rpm -Uvh MySQL-devel-4.1.10-0.i386.rpm
```

Next we'll install the Perl Compatible Regular Expressions

```
shell# tar -zxvf pcre-5.0.tar.gz
shell# cd pcre-5.0
shell#./configure
shell# make
shell# make install
shell# cd /usr/local
```

Snort, the heart of our project is next.

```
shell# tar -zxvf snort-2.3.2.tar.gz
shell# cd snort-2.3.2
shell#./configure
shell# make
shell# make install
shell# cd /usr/local
```

Now comes Barnyard which will off load database communications so that Snort will be free to watch packets.

```
shell# tar -zxvf barnyard-0.2.0.tar.gz
shell# cd barnyard-0.2.0
shell#./configure -enable-mysql
shell# make
shell# make install
shell# cd /usr/local
```

Next come the Tool Command Language also known as Tcl

```
shell# tar -zxvf tcl8.4.9-src.tar.gz
shell# cd tcl8.4.9/unix
shell#./configure
shell# make
shell# make install
shell# ln -s /usr/local/bin/tclsh8.4 /usr/local/bin/tclsh
shell# cd /usr/local
```

The Security Analyst Connection Profiler comes next.

```
shell# tar -zxvf sancp-1.6.1.tar.gz
shell# cd sancp-1.6.1
shell# make
shell#./install.sh
shell# cd /usr/local
```

And finally the sensor portion of SGUIL.

```
shell# tar -zxvf sguil-sensor-0.5.3.tar.gz
```

shell# mv *.gz *.rpm src

Configuration of the Sensor

Now that the Sensor's software is compiled a bit of configuration is in order. A directory structure to contain data and configuration files is necessary. In Sguil Installation Guide v 0.5.3 02 Richard Bejtlich likes putting everything under a /nsm directory, but notes that the SGUIL developers use a /snort_data naming scheme which is what I'm going to use. So we'll use the mkdir command to make our new directory structure:

```
shell# mkdir /snort_data
shell# mkdir /snort_data/sensor
shell# mkdir /snort_data/sensor/dailylogs
shell# mkdir /snort_data/sensor/portscans
shell# mkdir /snort_data/sensor/sancp
shell# mkdir /snort_data/sensor/rules
shell# mkdir /var/log/snort
shell# mkdir /usr/local/etc/snort
```

Next we will make the sguil user owner of the newly created directory structure.

```
shell# chown -R sguil:sguil /snort_data /var/log/snort /usr/local/etc/snort
```

We'll need to configure snort, so begin by copying the configuration file that is supplied in the distribution to a place in our newly created directory structure:

```
shell# cp /usr/local/snort-2.3.2/etc/snort.conf /usr/local/etc/snort
```

The configuration file needs a few modifications. I like to use vi as my editor, but feel free to use the editor of your choice.

```
shell# vi /usr/local/etc/snort/snort.conf
```

Modify lines in the snort.conf file to read:

```
var HOME_NET specify-your-net-here
var EXTERNAL_NET !$HOME_NET
var RULE_PATH /snort_data/sensor/rules
```

Uncomment the output line to read

```
output log_unified: filename snort.log, limit 128
```

Save the new configuration file and then copy the rules into their proper place with the following command:

```
shell# cp /usr/local/snort-2.3.2/rules/* /snort_data/sensor/rules
```

We also need to copy some miscellaneous files that will be used by both Snort and Barnyard by issuing the following commands:

```
shell# cd /usr/local/snort-2.3.2/etc
shell# cp *.map *.config threshold.conf /snort_data/sensor/rules
```

Bejtlich also recommends symbolic links from /usr/local/etc/snort to the actual files in the /nsm/sensor/rules directory which we'll do with the following set of commands (remember we decided to go with /snort_data instead of /nsm):

```
shell# cd /usr/local/etc/snort
shell# ln -s /snort_data/sensor/rules/classification.config classification.config
```

shell# ln -s /snort_data/sensor/rules/gen-msg.map gen-msg.map
shell# ln -s /snort_data/sensor/rules/reference.config reference.config
shell# ln -s /snort_data/sensor/rules/sid-msg.map sid-msg.map
shell# ln -s /snort_data/sensor/rules/threshold.conf threshold.conf
shell# ln -s /snort_data/sensor/rules/unicode.map unicode.map

Now let's configure Barnyard. First we'll copy the configuration file supplied in the distribution over to /usr/local/etc/snort by issuing the command:
shell# cp /usr/local/barnyard-0.2.0/etc/barnyard.conf /usr/local/etc/snort
And now we can can edit our new configuration file.
shell# vi /usr/local/etc/snort/ barnyard.conf
Make the following changes to the configuration file:
config hostname: Sensor
output sguil: mysql, sensor_id 0, database sguildb, server mysqlserver, \
user sguil, password YourDatabasePassword, sguild_host sguilserver, \
sguild_port 7736

SANCP also requires a small amount of configuration first we'll copy the include configuration file over to /etc/local/etc/snort by issuing the command:
shell# cp /usr/local/sancp-1.6.1/etc/sancp/sancp.conf /usr/local/etc/snort
Then we can modify the newly created configuration file's HOME_NET variable to match the network that we're interested in watching. The HOME_NET variable is not present in the original configuration file, so it has to be added.

The SGUIL Sensor Agent needs to be configured next.
shell# vi /usr/local/sguil-0.5.3/sensor/sensor_agent.conf
Make the following changes:
set SERVERHOST SGUILserver
set HOSTNAME Sensor
set S4_KEEP_STATS 0
set SANCP 1
(Bejtlich).

Finally a few changes to log_packets.sh
shell# vi /usr/local/sguil-0.5.3/sensor/log_packets.sh
Make the following changes
HOSTNAME=”Sensor”
LOG_DIR=”/snort_data”INTERFACE=”eth0” (This is the interface you're listening on, so adjust accordingly.)
OPTIONS=”-u sguil -g sguil -m122”
(Bejtlich).

Setting up the MySQL Server

Just as we did with the sensor our first order of business with the MySQL Server is to download the necessary programs. Again I'm going to use /usr/local as my staging area, so first navigate there by typing:
shell# cd /usr/local.
To install the MySQLServer host we'll need the MySQL server and the SGUIL server software. The latter is only needed for a couple of scripts to create the

database as we will run the actual SGUIL server on a separate host.
You may download MySQL server and the SGUIL server by using the following command:
shell# wget http://mysql.osuosl.org/Downloads/MySQL-4.1/MySQL-server-4.1.10-0.i386.rpm
shell# wget http://mysql.osuosl.org/Downloads/MySQL-4.1/MySQL-client-4.1.10-0.i386.rpm
shell# wget http://easynews.dl.sourceforge.net/sourceforge/sguil/sguil-server-0.5.3.tar.gz

If you prefer to choose your own mirror and/or not use wget you may also find the necessary files at http://dev.mysql.com/downloads/mysql/4.1.html (MySQL). and
http://sguil.sourceforge.net/index.php?page=download. (Visscher).

Next let's install the software that we just downloaded. Prior to installation of MySQL you will need to allow MySQL transactions in SELinux. Again my friend Google came through and I found a Web Page with the necessary clues to determine how to accomplish this. Edit the /etc/selinux/targeted/booleans file:
shell# vi /etc/selinux/targeted/booleans
and add a line at the end that reads: mysqld_disable_trans=1
Then restart the system by typing:
shell# shutdown -r now (Mitchell).
Another thing that will come in quite handy is the ability to answer on the MySQL port. To enable that port run the firewall configuration utility:
shell# /usr/bin/system-config-securitylevel-tui
Then select customize and then in addition to the ssh that is already checked enter “mysql:tcp” in the Other field and save your settings by selecting:
OK

MySQL and SGUIL software installation is accomplished via the following commands:
shell# rpm -Uvh MySQL*.rpm
shell# tar -zxvf sguil-server-0.5.3.tar.gz
In preparation for the next section let’s actually start the MySQL server:
shell# /etc/init.d/mysql start
Insure that the MySQL server starts whenever the system is rebooted by issuing the following comman:
shell# chkconfig --level 2345 mysql on

Creating the Database

The next step is database creation and password setup. here we’ll follow the instructions that Robert “Bamm” Visscher provides in the install.txt document included in the SGUIL package. The first thing Visscher notes is that “Some operating systems install mysql-server with a ‘root’ admin account and NO passwd.” This is certainly the case here, so the correct that situation:

shell# mysql –u root mysql

```
mysql> UPDATE user SET \
Password=PASSWORD('PickRootPass') WHERE user='root';
mysql> FLUSH PRIVILEGES;
mysql> quit
```

Visscher also recommends creating a sguil user in MySQL and to use that account for the installation:

```
shell# mysql -u root -p mysql
Enter password: (This matches the PickRootPass above)
mysql> CREATE DATABASE sguildb;
mysql> GRANT ALL PRIVILEGES ON sguildb.* to \
sguil@localhost IDENTIFIED BY 'PickSguilPass' with GRANT \
OPTION;
```

If you choose to utilize separate machines as I have in this installation you will also need to allow your sensor to write to this database with the following command:

```
mysql> GRANT ALL PRIVILEGES ON sguildb.* to sguil@Sensor.your.domain \
IDENTIFIED BY ' PickSguilPass ' with GRANT OPTION;
```

Again if you decide to utilize separate machines you will also need to allow your SGUILServer access with the following command:

```
mysql> GRANT ALL PRIVILEGES ON sguildb.* to sguil@eeyore.isu.edu \
IDENTIFIED BY ' PickSguilPass ' with GRANT OPTION;
```

Finally, prior to actual creation of the data base Visscher recommends flushing privileges and exiting from MySQL:

```
mysql> FLUSH PRIVILEGES;
mysql> quit
```

For the actual creation of the data base use the command

```
shell# mysql -u sguil -p sguildb \
< /usr/local/sguil-0.5.3/server/sql_scripts/create_sguildb.sql
```

Now test to make sure that the tables are really there with the command

```
shell# mysql -u root -p -D sguildb -e "show tables"
```

If you have good karma, have lived a good life, and probably most importantly have not skipped any steps, you will be rewarded with the following output:

```
+------------------------+
| Tables_in_sguildb |
+------------------------+
| data                   |
| event                  |
| history                |
| icmphdr                |
| nessus                 |
| nessus_data            |
| portscan               |
| sancp                  |
| sensor                 |
| sessions               |
| status                 |
```

```
| tcphdr                |
| udphdr                |
| user_info             |
| version               |
+-----------------------+
```

Setting up the SGUIL Server

Now we are gathering the ingredients for the Sguild Server. All of the necessary programs my be downloaded using the following commands:

```
shell# wget http://mysql.osuosl.org/Downloads/MySQL-4.1/MySQL-devel-4.1.10-0.i386.rpm
shell# wget http://mysql.osuosl.org/Downloads/MySQL-4.1/MySQL-shared-4.1.10-0.i386.rpm
shell# wget http://easynews.dl.sourceforge.net/sourceforge/tcl/tcl8.4.9-src.tar.gz
shell# wget http://easynews.dl.sourceforge.net/sourceforge/tls/tls1.5.0-src.tar.gz
shell# wget http://easynews.dl.sourceforge.net/sourceforge/tcllib/tcllib-1.7.tar.gz
shell# wget http://easynews.dl.sourceforge.net/sourceforge/tclx/tclx8.3.5-src.tar.gz
shell# wget http://www.xdobry.de/mysqltcl/mysqltcl-3.01.tar.gz
shell# wget http://lcamtuf.coredump.cx/p0f.tgz
shell# wget http://www.circlemud.org/pub/jelson/tcpflow/tcpflow-0.21.tar.gz
shell# wget http://easynews.dl.sourceforge.net/sourceforge/sguil/sguil-server-0.5.3.tar.gz
```

If you prefer to choose your own mirror and/or if you were bitten by a wget as a child and prefer not use it you may also find the necessary files at:
http://dev.mysql.com/downloads/mysql/4.1.html (MySQL).
http://www.tcl.tk/software/tcltk/downloadnow84.html (Tcl Developer Xchange).
http://prdownloads.sourceforge.net/tls/tls1.5.0-linux-x86.tar.gz?download (TLS OpenSSL Tcl Extension).
http://prdownloads.sourceforge.net/tcllib/tcllib-1.7.tar.gz?download (Tcllib).
http://prdownloads.sourceforge.net/tclx/tclx8.3.5-src.tar.gz?download (Tclx).
http://www.xdobry.de/mysqltcl/mysqltcl-3.01.tar.gz (Soderlund).
http://lcamtuf.coredump.cx/p0f.tgz (Zalewski).
http://sguil.sourceforge.net/index.php?page=download (Visscher).

Remember we already created a sguil user at OS installation time and assigned the user a password. Next let's create a couple directories and give the sguil user ownership:

```
shell# mkdir /snort_data
shell# mkdir /snort_data/archive
shell# mkdir /snort_data/rules
shell# mkdir /snort_data/rules/Sensor
shell# chown -R sguil:sguil /snort_data
```

Then install the MySQL Libraries and header files as well as the dynamic

client libraries that will be needed later for MySQLTcl to compile. This is accomplished with the following command:

```
shell# rpm -Uvh MySQL*.rpm
```

Now we will install TCL by issuing the commands:

```
shell# tar -zxvf tcl8.4.9-src.tar.gz
shell# cd tcl8.4.9/unix
shell# ./configure
shell# make
shell# make install
shell# ln -s /usr/local/bin/tclsh8.4 /usr/local/bin/tclsh
shell# cd /usr/local
```

Next we'll unpack and compile the TLS OpenSSL extension to Tcl by issuing the following commands:

```
shell# tar -zxvf tls1.5.0-src.tar.gz
shell# ln -s /usr/share/ssl /usr/local/ssl
shell# ln -s /usr/include /usr/local/ssl/include
shell# cd tls1.5
shell# ./configure
shell# make
shell# make install
shell# cd /usr/local
```

On to tcllib which we will unpack and compile with the following commands:

```
shell# tar -zxvf tcllib-1.7.tar.gz
shell# cd tcllib-1.7
shell# ./configure
shell# make
shell# make install
shell# cd /usr/local
```

The TclX extension to Tcl comes next, so here are the commands to get it going:

```
shell# tar -zxvf tclx8.3.5-src.tar.gz
shell# cd tclx8.3.5/unix
shell# ./configure --enable-tk=NO
shell# make
shell# make install
shell# cd /usr/local
```

MySQLTcl is next on the docket and the commands that will unpack and compile this program are:

```
shell# tar -zxvf mysqltcl-3.01.tar.gz
shell# cd mysqltcl-3.01
shell# ./configure
shell# make
shell# make install
shell# cd /usr/local
```

The passive OS fingerprinting tool comes next and the commands to

install it are:
shell# ln -s /usr/include/pcap-bpf.h /usr/include/net/bpf.h
shell# tar -zxvf p0f.tgz
shell# cd p0f
shell# cp mk/Linux Makefile
shell# make
shell# make install
shell# cd /usr/local

Now time for the TCP flow recorder, tcpflow the commands to unpack and compile are:
shell# tar -zxvf tcpflow-0.21.tar.gz
shell# cd tcpflow-0.21
shell# ./configure
shell# make
shell# make install
shell# cd /usr/local

Finally we will install the SGUIL server software with the following commands:
shell# tar -zxvf sguil-server-0.5.3.tar.gz
shell# cd sguil-0.5.3/server
shell# vi sguild.conf
Make these mofifications
set RULESDIR /snort_data/rules
set DBPASS PasswordForSguilUser (this was selected when you setup the MySQL server)
set DBUSER sguil
set LOCAL_LOG_DIR /snort_data/archive
set TCPFLOW "/usr/local/bin/tcpflow"
set P0F_PATH "/usr/sbin/p0f"
Save the modifications.
Create a sguild user by issuing the following command:
shell# ./sguild -c sguild.conf -u sguild.users -adduser sguil

Since we've gone to all the trouble to set up a SGUIL server it would probably be nice to allow it to answer on the ports that it's expecting to communicate on namely tcp ports 7734 and 7736. Do this by issuing the command:
shell# /usr/bin/system-config-securitylevel-tui and then select customize. In addition to the ssh field that is already checked enter "7734:tcp 7736:tcp" in the other field and save your changes by selecting:
OK

Encrypting Communication between SGUIL Server and Clients

Visscher has included a file called /usr/local/sguil-0.5.3/doc/OPENSSL.README and Bejtlich also has an "Encryption" section in his Installation Guide we will utilize a combination of the instructions found in

those two places to configure OpenSSL and the OpenSSL Tcl Extension. First we will create a directory that will hold our certificates and then we will make our sguil user the owner of that directory.

```
shell# mkdir /usr/local/etc/snort
shell# chown sguil:sguil /usr/local/etc/snort
shell# su - sguil
```

Generate a Certificate Authority.

```
sguil$ openssl req -out CA.pem -new -x509
```

Enter PEM pass phrase when prompted.
Enter certificate information when prompted.

Generate a certificate key pair.

```
sguil$ openssl genrsa -out sguild.key 1024
sguil$ openssl req -key sguild.key -new -out sguild.req
```

Enter certificate information when prompted, but I choose to leave challenge password and company name both blank.

```
sguil$ vi file.sr1
```

Pick some two digit number out of the air; stick it in here and save the file.

```
sguil$ openssl x509 -req -in sguild.req -CA CA.pem -CAkey privkey.pem \
-CAserial file.sr1 -out sguild.pem
```

Enter a pass phrase when prompted.

```
squil$ mv sguild.key sguild.pem /usr/local/etc/snort
```

Setting up the SGUIL Client

Now last but certainly not least we'll get the client ready to go. The first thing we can do is to uninstall a few items that we don't need since that were installed with Fedora. rpm -e tclx-8.3.5-4 tcl-8.4.7-2 tk-8.4.7-2 emacspeak-17.0-7.i386. Next we can gather the packages that we will need.

```
shell# wget
http://downloads.activestate.com/ActiveTcl/Linux/8.4.9/ActiveTcl8.4.9.0.121397-linux-ix86.tar.gz
shell# wget http://easynews.dl.sourceforge.net/sourceforge/tls/tls1.5.0-src.tar.gz
shell# wget http://easynews.dl.sourceforge.net/sourceforge/sguil/sguil-client-0.5.3.tar.gz
```

If you would prefer to use a browser to navigate to the individual sites they URLs for the packages above are:

http://activestate.com/Products/ActiveTcl/
http://tls.sourceforge.net/
http://sguil.sourceforge.net/index.php?page=download

We're ready to install ActiveTcl. Bejtlich notes in his Sguil Installation Guide "The Sguil client requires the most addition of new applications on the analyst workstation." By untilzing ActiveTcl instead of compiling everything from source we dramatically cut down the number of steps involved. To install ActiveTCL:

```
shell# tar -zxvf ActiveTcl8.4.9.0.121397-linux-ix86.tar.gz
shell# cd ActiveTcl8.4.9.0.121397-linux-ix86
shell# ./install.sh
```

The installation will prompt you to agree to the license agreement it will also ask you where you would like to install. Respond that you would like to install the software in /usr/local and you'll avoid having to mess with altering your path later on.

```
shell# cd /usr/local
```

Next we'll unpack and compile the TLS OpenSSL extension to Tcl by issuing the following commands:

```
shell# ln -s /usr/share/ssl /usr/local/ssl
shell# ln -s /usr/include /usr/local/ssl/include
shell# tar -zxvf tls1.5.0-src.tar.gz
shell# cd tls1.5
shell# ./configure
shell# make
shell# make install
shell# cd /usr/local
```

It would be a bit difficult to have a SGUIL client without the SGUIL client software installed. The commands to accomplish that are:shell# tar -zxvf sguil-client-0.5.3.tar.gz

Modifiy the file /usr/local/sguil-0.5.3/client/sguil.conf to read "set SERVERHOST SGUILServer."

Analysts Start Your Engines or at Least Test Them

Bejtlich recommends the following sequence of commands to test all of the pieces:

On SGUILServer as the user squil type:

```
shell$ cd /usr/local/sguil-0.5.3/server
shell$ ./sguild -c sguild.conf -u sguild.users -O \
/usr/local/lib/libtls1.50.so -C /usr/local/etc/snort
```

On Sensor as user sguil type:

```
shell$ cd /usr/local/etc/snort
shell$ barnyard -c barnyard.conf -d /snort_data/piglet -g \
gen-msg.map -s sid-msg.map -f snort.log -w waldo.file
```

On Sensor as user root type:

```
shell# ifconfig eth0 -arp up
shell# snort -u sguil -g sguil -c /usr/local/etc/snort/snort.conf -U -l \
/snort_data/piglet -m 122 -A none -i eth0
```

On Sensor as user root type:

```
shell# /usr/local/bin/sancp -d /snort_data/piglet/sancp -i eth0 -u \
sguil -g sguil -c /usr/local/etc/snort/sancp.conf > /var/log/sancp.log
```

On Sensor as user sguil type:

```
shell$ cd /usr/local/sguil-0.5.3/sensor
shell$ ./sensor_agent.tcl
```

On Sensor as user root type:

```
shell# cd /usr/local/sguil-0.5.3/sensor
shell# ./log_packets.sh start
```

On the SGUIL Client as user root type:

```
shell# cd /usr/local/sguil-0.5.3/client
shell# ./sguil.tk
```

If all has gone well you should now see a login box. Use the name and password created under “Setting up the SGUIL Server” above. After successful authentication occurs, choose the sensor you would like to monitor which at this point is the only sensor. Finally select start SGUIL. Now you should be rewarded with the SGUIL GUI!

Conclusion

As you can see installation of Snort, Barnyard, MySQL and SGUIL on Fedora Core 3 while not exactly rocket science is also not a trivial task. It is my sincere hope that this guide will help you get started quickly with as little pain as possible. While this guide should take you through the installation it is not the definitive source for these packages, so I would strongly suggest that you make good use of the “Works Consulted” section.

May your false positives be few and your false negatives be fewer; happy detecting.

Works Consulted

ActiveTCL. 6 Mar. 2005
< http://activestate.com/Products/ActiveTcl/>.

Ampatt, Praveen D. "Tutorial on Installing and configuring Snort on Fedora core2: An Intrusion Analyst's, developer's & a researcher's perspective." GIAC 8 Dec. 2004. SANS Institute. 24 Jan. 2005 <http://www.giac.org/practical/GSEC/Praveen_Ampatt_GSEC.pdf>.

Baker, Andrew R., Brian Caswell, Mike Poor, Raven Alder, Jacob Babbin, Jay Beale, Adam Doxtater, James C. Foster, Toby Kohlenberg, Michael Rash. Snort 2.1: Intrusion Detection. 2nd ed. Rockland: Syngress, 2004.

Barnyard. Project Admin. Andrew Baker. 2005. Sourceforge. 25 Jan. 2005 <http://prdownloads.sourceforge.net/barnyard/barnyard-0.2.0.tar.gz?download>.

Boman, Michael. "Network Security Analysis with SGUIL." BOSECO Internet Security Solutions 7 May 2004. Jensen Consulting Pte Ltd. 24 Jan. 2005 <http://www.boseco.com/presentations/sguil-2004-1/?meid=18>.

Bejtlich, Richard. "Squil Installation Guide v 0.5.3_02." Tao Security 7 Nov. 2004. Identity Vector Solutions, LLC. 23 Jan. 2005 <http://sguil.sourceforge.net/sguil_guide_latest.txt>.

---. The Tao of Network Security Monitoring: Beyond Intrusion Detection. Boston: Addison-Wesley, 2005.

Caswell, Brian, Jeremy Hewlett. "Snort Users Manual 2.2.0." Snort 10 Aug 2004. Sourcefire. 23 Jan. 2005 <http://www.snort.org/docs/snort_manual.pdf>.

Curry, John. “Security Analyst Network Connection Profiler.” 2 Nov. 2004. Metre Networks. 25 Jan. 2005 <http://www.metre.net/files/sancp-1.6.1.tar.gz>.

Derouin, Travis. “How to Change the Timezone in Linux.” wikiHow. Ed. Anonymous.18 Jan. 2005 <http://wiki.ehow.com/Change-the-Timezone-in-Linux>.

Elson, Jeremy. “tcpflow -- A TCP Flow Recorder.” 7 Aug. 2003. www.circlemud.org. 25 Jan. 2005 <http://www.circlemud.org/pub/jelson/tcpflow/tcpflow-0.21.tar.gz>.

Harper, Patrick. “Snort Install Manual Version 8.” Internet Security Guru 10 Nov. 2004. InternetSecurityGuru.com. 23 Jan. 2005 <http://www.internetsecurityguru.com/documents/Snort_SSL_FC2.pdf>.

“incr Tcl.” Project Admins. David Gravereaux, et al. 13 Sep. 2002. Sourceforge. 25 Jan. 2005 <http://sourceforge.net/project/showfiles.php?group_id=13244>.

Koziol, Jack. Intrusion Detection with Snort. Indianapolis: Sams, 2003.

Mitchell, Justin. PHP 5 and MySQL 4 on Fedora Core 3, from RPMs. Ed. Marion Bates. 21 Feb. 2005. 7 Mar. 2005 <http://www.whoopis.com/howtos/php5-mysql4-FC3-rpm.html>.

MySQL 4.1 Downloads. 2005. 25 Jan. 2005 <http://dev.mysql.com/downloads/mysql/4.1.html>.

Nasrawi, Ghaith. “Snort Installation Guide with BASE and MySQL support ver 0.4g.” 27 Dec. 2004. Sourceforge 23 Jan. 2005 <http://drider.sourceforge.net/docs/snort/snort_install_guide_FC3_2.txt>.

PCRE. Project Admin. Andrew Ho. 2005. 25 Jan. 2005. Sourceforge. <http://prdownloads.sourceforge.net/pcre/pcre-5.0.tar.gz?download>.

Reining, Christopher J. "The State of Intrusion Detection." GIAC 30 Mar. 2004. SANS Institute. 24 Jan. 2005. <http://www.giac.org/practical/GCIA/Christopher_Reining_GCIA.pdf>.

sguil_arch.png. n.d. SGUIL - The Analyst Console for Network Security Monitoring. 7 Dec. 2004. Sourceforge.net 24 Jan. 2005. <http://sguil.sourceforge.net/sguil_arch.png>.

Snort downloads center. 2005. Sourcefire. 11 Mar. 2005 <http://www.snort.org/dl/snort-2.3.tar.gz>.

Soderlund, Hakan, Gregory Gulik, Tobias Ritzau, Paolo Brutti, Artur Trzewik. mysqltcl - Tcl Mysql Interface (Version 3.01). 28 Dec 2004. 25 Jan. 2005 <http://www.xdobry.de/mysqltcl/mysqltcl-3.01.tar.gz>.

Tcl Developer Xchange. 7 Dec 2004. Tcl Developer Xchange. 25 Jan. 2005 <http://www.tcl.tk/software/tcltk/downloadnow84.html>.

Tcllib. 6 Oct. 2004. tcllib.sourceforge.net. 25 Jan. 2005 <http://prdownloads.sourceforge.net/tcllib/tcllib-1.7.tar.gz?download>.

Tclx. 31 Oct 2002. tclx.sourceforge.net. 25 Jan. 2005 <http://prdownloads.sourceforge.net/tclx/tclx8.3.5-src.tar.gz?download>.

TLS OpenSSL Tcl Extension. 17 Feb. 2004. tls.sourceforge.net 25 Jan. 2005 <http://prdownloads.sourceforge.net/tls/tls1.5.0-linux-x86.tar.gz?download>.

Visscher, Robert "Bamm". "Install v 1.17." 10 Aug. 2004. SGUIL - The Analyst

Console for Network Security Monitoring Sourceforge 23 Jan. 2005 <http://sguil.sourceforge.net/install.txt>.

---. "Project: Sguil: File List." 7 Dec. 2004. SGUIL - The Analyst Console for Network Security Monitoring. Sourceforge 25 Jan. 2005 <http://sourceforge.net/project/showfiles.php?group_id=71220&package_id=70722&release_id=288260>.

Zalewski, Michal. the new p0f: 2.0.5. 2004. 25 Jan. 2005 <http://lcamtuf.coredump.cx/p0f.tgz>.