Build Securely Suricata with Sguil Sensor Step-by-Step Powered by Slackware 64-Bit Linux

By Guy Bruneau, GSE (GSEC, GCIA, GCIH, GCUX, GCFA, GPEN)

Version 8.0 – 17 June 2015

Introduction

This configuration process is used to deploy Snort sensors with the information managed through a Sguil console powered by the Slackware Linux (GNU) operating system. This setup was developed for sensors using IDE or SCSI drives. The full installation using this setup is ~1.4 GB in size and provides no remote services except through Secure Shell and Webmin for remote management of the sensor and the server. The Snort sensor logs are process via **Barnyard** backend processing.

This installation contains three separate and ready to use Sguil packages that contain all the necessary files to install Sguil as a sensor only (sensguil package), database only (sguildb package), or all-in-one systems (sguil package). Sguil contains some very useful analysis function such as the Security Analyst Network Connection Profiler (sancp) which collects statistical network traffic information, it has a script to log all the packets in pcap format, it uses tcpflow and p0f to get TCP session transcripts, it uses the Passive Asset Detection System (PADS) to collect banners on host services it sees to be used for correlation and uses Wireshark for in-depth packet analysis.

In addition, (if configured) this ISO has the ability to collect Netflow using NfSen a graphical web based front end for the nfdump netflow tools. Each sensor can generate netflow data using softflowd, see /etc/rc.d/rc.local to configure the application.

The last service (if configured), is the ability to collect all DNS queries viewed by the sensor using PassiveDNS and store the data into the passivedns database that can be queried using the custom Sguil client included in this ISO.

There is a sguil.pdf document that explains how to setup Sguil and the Sguil client on a Windows workstation. Additional information including console snapshots can be viewed at: http://sguil.sourceforge.net.

This installation has a web management interface called Webmin which is used to remotely manage the MySQL database and Snort sensor via a SSL enabled web browser. Additional information about Webmin is available at: http://www.webmin.com/. In addition, a Webmin Snort plugin to fully manage the Snort sensor (config file, plugins, oinkmaster, ruleset, etc.) to ease remote management of the sensor. See the Webmin section to correctly configure this package.

I recommend using **a minimum of 4GB of RAM** to function efficiently. If you have less than that, check the Snort documentation for the memory options in the Snort configuration file located in /usr/local/snort/etc/snort.external.conf to ensure the sensor function properly. The most common memory option used is either search-method ac-bnfa or lowmem for systems with 1 GB of RAM. **This package is built for a sensor that contains 2 NIC cards** (eth0 = control and eth1/bond0 = Snort).

In order to make Snort signature updates more flexible, I have included the oinkmaster script written by Andreas Östling available at http://oinkmaster.sourceforge.net/. The Snort sensor includes the Emerging Threats rules also updated daily via Oinkmaster. The rules are available at: http://www.emergingthreats.net/rules/. You should review the /usr/local/snort/oinkmaster.conf file for its configuration.

The Shadow/Snort ISO image powered by the Slackware Linux OS can be downloaded at: http://handlers.sans.edu/gbruneau/iso/shadow/shadow64.iso

The MD5 signature for the ISO image is available at: http://handlers.sans.edu/gbruneau/iso/shadow/shadow64.md5.

Important: Before you start, make sure you are disconnected from the network until the sensor has been securely configured.

The following is a summary on how to install a sensor, a database or a combination of both a sensor and a database on a single box. If you wish, you can also use the ISO and install the full system without turning on Sguil and use it to collect Netflow data from your router or firewalls that support it.

Detailed Installation, Configuration and Partitioning the Drive

Drive partitioning can be done in multiple ways. Of course the more disk you have available, you can size this way up to meet your need. This is an example that can be used if you have two drives and you are installing both the database and server on the same system:

Database and Sensor

Drive One	(Must have for basic installation)	
/	Minimum of 5 GB	
swap	Minimum of 1 GB	
/usr/local	Minimum of 10 GB	(Contains MySQL DB and Suricata)
/LOG	Remainder of drive	(Contains the tcpdump Sguil logs)

Drive Two		
/LOG/nfsen	Minimum of 20 GB	(i.e. collects NetFlow data)

- Boot on the system using the Slackware CD-ROM.

To partition the drive, login as root and run ***cfdisk /dev/hda*** (IDE drive), ***cfdisk /dev/sda*** (SCSI drive) or ***cfdisk /dev/cciss/c0d0*** (Raid drive). If this isn't a new drive, delete the old partitions before starting.

- sda1: / = 5 GB (Select new, select primary, size is 1000, beginning, bootable)
- sda2: SWAP = 1 GB or same amount as the RAM (Select Pri/Log Free Space, new,

primary, size is 512, beginning)
- Change hda2 to swap by selecting *type 82*
- sda3 (Select Pri/Log Free Space, new, primary, for NfSen logs)
- sda4 (Select Pri/Log Free Space, new, primary, for MySQL and Snort)
- *sdb1 (Select Pri/Log Free Space, new, primary, remainder of disk for pcap logs)*
- Select Write to save the new settings to disk
- Select *Quit* to exit

Note on MySQL Drive size estimate:

Recommend no more than about 250GB for the MySQL data. To keep ~500,000,000 rows of SANCP data and ~2,500,000 alerts, a MySQL database will need ~180GB of disk space.

Install the Software

Now that you have partitioned the drive, and saved your setting, you are ready to setup the Operating System.

- Run setup
- Select addswap with default
- Continue with installation: yes
- Check swap partitions for bad blocks (It’s a choice here but usually no): yes or no
- Swap partition configured
- Select Linux installation partition
    - /dev/sda1 (format, ext4 - default)
    - /dev/sda3 (format, ext4 - default)
    - /dev/sda4 (format, ext4 - default)
    - /dev/sdb1 (format, ext4 - default →only if using a second disk)
- Select mount point for /dev/sda3: /usr/local → **Needed for Suricata/MySQL**
- Select mount point for /dev/sda4: /LOG → **Needed for Sguil pcap logs**
- Select mount point for /dev/sdb1: /LOG/nfsen → **Needed for NfSen NetFlow**
    - Select Ok and continue with setup
- Select *1* to install from a Slackware CD-ROM
- Select *auto* to scan automatically for the CD or DVD drive
- Make sure the CD or DVD is in the CD-ROM drive and select OK
- Select *Yes* on continue
- Scan for the CD or DVD drive

- Install Snort with Sguil from the installation CD which only shows 8 package groups:

A, AP, D, L, N, TCL, X, Z

- Select *OK* to continue and go to the *INSTALL* section
- Select install everything (full) and wait until all packages are installed
- Make a boot disk for recovery (LILO) or skip it

- After the boot disk, choose continue with the configuration
- Install LILO and select *expert*
    - Select Begin, at the blank prompt press enter, select all *default*, install to *MBR* confirm location to install lilo (select default @/dev/hda, /dev/sda or /dev/cciss/c0d0) and none
    - Add Linux and choose the root partition (One with * i.e. /dev/hda1, /dev/sda1 or /dev/ccisss/c0d0p1)
    - Use *Linux* as a partition name
    - Install LILO
- Configure the network with your settings
- Confirm the network setup
- Confirm startup services you want to run (**Sensor only** → select Barnyard, Suricata, httpry, passivedns, sagan, barnsagan **Server only** → select sguild and MySQL or for **Sensor/Server** → select Sguil, passivedns, pdns2db, Barnyard, Suricata, httpry, passivedns, sagan, barnsagan and MySQL), select *Enter*
- Setup the hardware clock
- Setup the root password
- After the installation completed, at the Slackware Linux Setup screen, select <*Cancel*>
- Depending of type of installation: cd /cdrom/sguil (DB and sensor combo) or sguildb (DB only) or sensguil (sensor only)
- Run **pkgtool**
- Select Current and install the package
- Remove CD (**eject**)
- Reboot (reboot at the prompt)
- Manually eject the CD-ROM
- Log back into the sensor as root
- Delete residual mail *rm /var/spool/mail/root*
- Configure the sensor and/or database by running /root/scripts/configure_sguil.sh (Options 3,4 and 5). If you are configuring the database (Option 2), add a user (Option 4) and if configuring a sensor (Option 1) configure the HOME_NET (Option 5).
- Configure NfSen (Option #1 & 4) if you plan to collect Netflow data /root/scripts/install_nfsen_x64.sh
- Configure Webserver (Option #1 & 3) to access PassiveDNS records (if NfSen wasn't configured) /root/scripts/install_nfsen_x64.sh
- If this a sensor installation, configure stunnel to send data to passivedns database
- Configure Sendmail if you want to send email from the Sguil database

Sguil Client Configuration

I strongly recommend using the updated client that is configured with additional menus and queries (Passive DNS, Geo IP and httpry) located on the CD in the files\sguil-0.8.0\client directory.

The original version can be downloaded at or on CD files directory:
http://sourceforge.net/project/showfiles.php?group_id=71220

Unpack in C:\sguil-0.8.0

Download Windows Active TCL at: http://www.activestate.com/Products/ActiveTcl/
Install at c:\tcl

Download Windows TLS libraries at: http://tls.sourceforge.net
Unpack in C:\tcl\lib

Download Wireshark for Windows at: http://www.wireshark.org
Install at C:\Wireshark

If you prefer Firefox instead of Explorer, download at: http://www.mozilla.org/
Install at C:\Firefox

Sguil sguil.conf update

Edit c:\sguil-0.7.0\client\sguil.conf → Configure according to your Windows settings

```
# Change SERVERHOST to the correct IP or servername
set SERVERHOST 192.168.30.4

#  Enable Ext DNS
set EXT_DNS 1

# Define the external nameserver to use. OpenDNS list 208.67.222.222 and 208.67.220.220
set EXT_DNS_SERVER 208.67.222.222

# Define a list of space separated networks (xxx.xxx.xxx.xxx/yy) that you want to use the OS's
#resolution for.
set HOME_NET "192.168.0.0/16 10.0.0.0/8"

# Path to wireshark (ethereal) win32 example
set WIRESHARK_PATH "c:/wireshark/wireshark.exe"

# Where to save the temporary raw data files on the client system You need to remember to
# delete these yourself.
# win32 example
set WIRESHARK_STORE_DIR "c:/tmp"

# Favorite browser for looking at sig info on snort.org win32 example (IE)
set BROWSER_PATH c:/progra~1/intern~1/iexplore.exe              or
set BROWSER_PATH "c:/firefox/firefox.exe"

# Mailserver to use for emailing alerts
set MAILSERVER mail.example.com

# Default From: address for emailing
set EMAIL_FROM foo@example.com
```

Note: The TLS libraries are used to encrypt the session between the Windows client and the database server.

Create the C:\TMP directory to store Wireshark files if it doesn't already exist

Note: If you are planning in using the Passive DNS system, you will need to edit the client\lib\extdata.tcl script and modify the path where you Sguil database is located either with the correct name or IP address. The current URL to access the Passive DNS database is:

https://snort/passivedns/?query=

Where snort must be replaced by your own Sguil database URL or IP.

Client Access to Database

The client can access the database at this point by executing the sguil.tk. However, sguil.tk must be associated with the “wish application” before it will start.

c:\sguil-0.7.0\client\sguil.tk

TCP Wrapper - SSH

Configure SSH TCP Wrappers via Webmin or in the following way:

- vi /etc/hosts.allow (Webmin, Servers, TCP Wrappers)
- Add in the TCP Wrappers file which host(s) are allowed to connect to the sensor

```
            sshd: 192.168.3. \
                  192.168.2.6 \
                  .site.ca
```

- The /etc/hosts.deny has been configured to deny ALL (ALL: ALL) by default

IPTables Firewall

Configure iptables firewall (rc.firewall)

Note: You need a firewall (iptables) to allow the sensor to be as invisible as possible. You can use the firewall supplied with this installation or create your own.

O- Configure the firewall located in */etc/rc.d* directory or create your own
O- Edit the firewall script and change to variable according to your site
O- *chmod 755 /etc/rc.d/rc.firewall* to enable the firewall
O- To start the firewall at this time execute: */etc/rc.d/rc.firewall* at the prompt
O- Check the firewall policy with the following command: iptables -L
O- The firewall will start upon the next reboot

Mounting USB Drive

To mount a USB drive with this OS, plug in the USB drive and do the following:

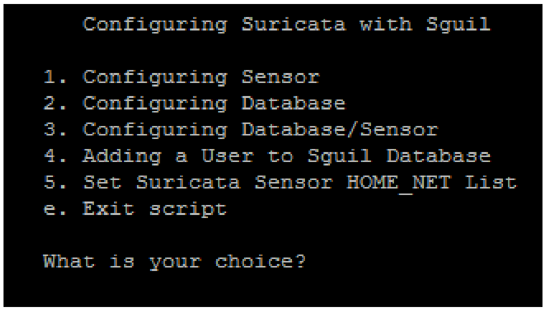
dmesg |grep sda, sdb, sdc or sdd	
mount /dev/sda? /mnt/hd	Where ? = the partition and usually 1
cd /mnt/hd	You can copy or move files from this directory
umount /usb	When done with the USB drive

Note: the device can be sda, sdb, sdc, sdd and usually partition 1 (i.e. sda1)

Final Installation Phase - Configure Database and Suricata Sensor

To configure all the scripts to get a sensor to report to the Sguil database, run the following script and this menu will appear:

/root/scripts/configure_sguil.sh



```
   Configuring Suricata with Sguil

1. Configuring Sensor
2. Configuring Database
3. Configuring Database/Sensor
4. Adding a User to Sguil Database
5. Set Suricata Sensor HOME_NET List
e. Exit script

What is your choice?
```

1. Is used to turn on Barnyard to report to the Sguil database and configure the Sensor name and the IP of the Sguil database
2. Is used to flush and reset all the database tables
3. Is used to turn on Barnyard to report to the Sguil dabase and to assign a sensor name for an all-in-one installation
4. Is used for adding a Sguil user account to the Sguil database for access
5. Is use to set a Snort sensor HOME_NET IP list

Configure suricata.yaml

Suricata has enabled by default several additional logs that use a lot of disk space. They can be useful but can quickly fill up /usr/local/suricata/log/suricata. Unless absolutely needed, I recommend disabling logging on the following log files:

eve.json, fast.log, http.log and stats.log

Edit the suricata.yaml log file and change each of the log file enabled: yes to a **no** to turn off logging

```
  # a line based log of HTTP requests (no alerts)
  - http-log:
    enabled: no
    filename: http.log
```

Using ArcSight CEF Formatted Logs

First step is to configure /etc/rsyslog.conf and add the following to the configuration file and restart /etc/rc.d/rc.syslog (change IP to your CEF receiver).

```
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
#*.* @@remote-host:514
local0.*        @192.168.25.7:514
```

Second step is to configure /usr/local/barnyard/etc/barnyard.conf and add the following in the alert_cef section and restart /etc/rc.d/rc.barnyard to activate:

```
output alert_cef: LOG_AUTH LOG_LOCAL0
```

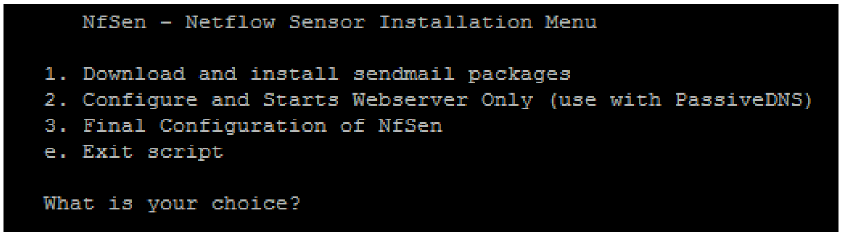
The /var/log/message log file will show this type of messages when Suricata alerts:

```
May 23 09:43:02 snort barnyard2[32362]: CEF:0|suricata|barnyard2|2.1.13|1.131:2003195:5|ET POLICY
Unusual number of DNS No Such Name Responses|9|src=64.71.255.198 dst=192.168.21.179 spt=53
dpt=28789 proto=UDP
```

Install NetFlow Sensor Collection on Primary Database

NfSen is a Netflow visualisation and investigation tool. NfSen is a graphical web based front end for the nfdump netflow tools. This ISO a preconfigured to collect netflow data using NfSen but before collection starts, you need to run the installation script that will download and configure the Sguil database to collect the netflow data. Run the following script and this menu will appear:

```
/root/scripts/install_nfsen_x64.sh
```



```
    NfSen - Netflow Sensor Installation Menu

 1. Download and install sendmail packages
 2. Configure and Starts Webserver Only (use with PassiveDNS)
 3. Final Configuration of NfSen
 e. Exit script

 What is your choice?
```

1. Is used to download and install the necessary Webserver packages and patch them
2. Is used to download and install the sendmail packages for email alerts
3. Configure and Starts Webserver Only (use with PassiveDNS)
4. Is used to complete the NfSen installation and configuration

Collect NetFlow using NfSen and softflowd

Each Sguil sensors have the ability to generate netflow data with the softflowd application. In order to start the application, you need to configure and edit the nfsen.conf file to set a listening port and name for the netflow data collected.

Using either Webmin (Servers, NfSen Controls), vi or nano, edit the following file:

```
vi /LOG/nfsen/etc/nfsen.conf
```

Find %sources and you should find something like this:

```
%sources = (
   'local'    => { 'port' => '9995', 'col' => '#0000ff', 'type' => 'netflow' },
   'peer1'       => { 'port' => '9996', 'col' => '#ff3300', 'type' => 'netflow' },
);
```

In this example, 2 listening ports are preconfigured but the name of the netflow sensor should be modified to match the sensor sending the data. If you have more sensor report, add them here with their own name, port and color and save the changes and exit.

The next step is the reconfigure and starts the netflow listener with the new information:

```
/LOG/nfsen/bin/nfsen reconfig
```

Last step. To generate netflow using softflowd, edit this file:

```
vi /etc/rc.d/rc.local
```

Find the following lines in the script and enable softflowd with the correct IP (either local or remote) and port:

```
echo "Starting SoftFlow..."
/usr/local/sbin/softflowd -i eth1 -t maxlife=60 -n 127.0.0.1:9996
```

Save the changes and either restart the sensor or type that command to start the service. This completes the setup. NfSen should now be receiving data which can be viewed on the database webserver.

Collect DNS Queries with PassiveDNS

The sensor now has the ability to collect all DNS queries viewed by the IDS interface (default eth1). If you enabled the rc.passivedns during the installation, the sensor upon reboot is already collecting all DNS queries which are saved in the /LOG/passivedns folder.

In order to load the queries into the passivedns database, you will need to enable rc.pdns2db service script.

Sensor Reporting to Sguil Server

If the installation is a standalone sensor, you need to configure and get stunnel configured to be able to send the data to the Sguil database. On the install CD there is a file named **stunnel.pdf**, follow these instructions before changing the rc.pdns2db service.

When the stunnel is setup between the Sguil database and sensor, proceed to change the startup script.

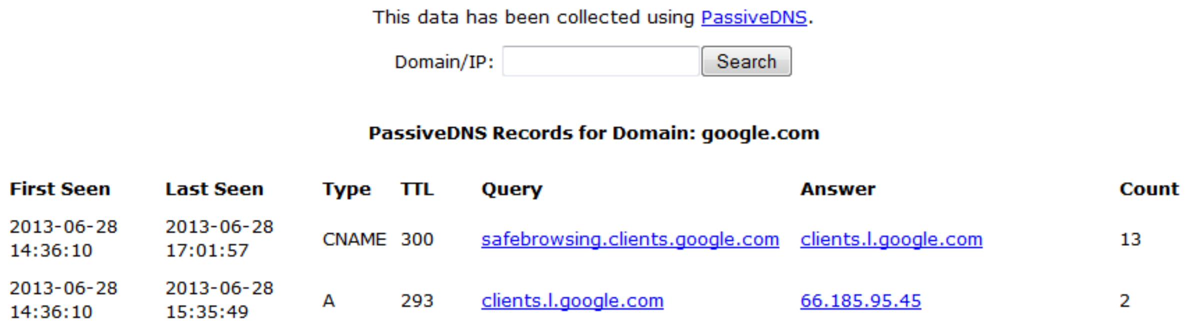
- chmod 755 /etc/rc.d/rc.pdns2db
- /etc/rc.d/rc.pdns2db start

All-in-Sguil Server/Sensor

For this configuration, just change the startup script to executable and start the service.

- chmod 755 /etc/rc.d/rc.pdns2db
- /etc/rc.d/rc.pdns2db start

Here is an example using the PassiveDNS database to query the domain google.com



This data has been collected using PassiveDNS.

PassiveDNS Records for Domain: google.com

First Seen	Last Seen	Type	TTL	Query	Answer	Count
2013-06-28 14:36:10	2013-06-28 17:01:57	CNAME	300	safebrowsing.clients.google.com	clients.l.google.com	13
2013-06-28 14:36:10	2013-06-28 15:35:49	A	293	clients.l.google.com	66.185.95.45	2

PassiveDNS Domain Exclusion

It is possible to exclude DNS domains from being inserted into the database by adding them into the configuration file located in /LOG/passivedns. There is a supplied example named skiplist.txt which takes a full domain as one line per entries. PassiveDNS is started with the /etc/rc.d/rc.passivedns script.

Note: Update lists as necessary.

Example – Single domain skiplist.txt

www.google.com
isc.sans.edu

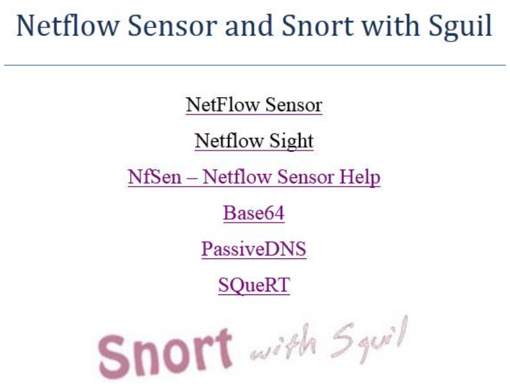
Example – Wildcard list with regex skiplist_pcre.txt

\.cloudfront.net$
\.doubleclick.net$
\.edgesuite.net$

Access Sguil Server Services

https://sguil_server

These web services will only be accessible after the installation of the NfSen sensor packages.



Netflow Sensor and Snort with Sguil

NetFlow Sensor

Netflow Sight

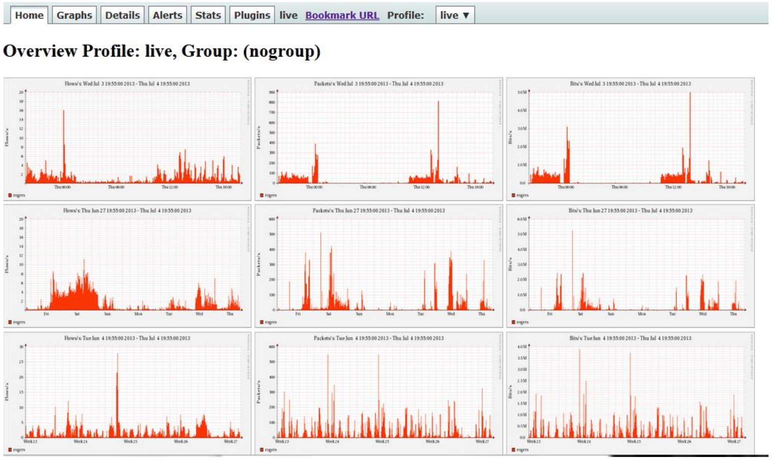
NfSen – Netflow Sensor Help

Base64

PassiveDNS

SQueRT

This is a Netflow (NfSen) display of the main dashboard. Author recommend using Firefox.



Overview Profile: live, Group: (nogroup)

Sagan - Realtime Log Analysis & Correlation Engine

“Sagan's structure and rules work similarly to the Sourcefire "Snort" IDS/IPS engine. This was intentionally done to maintain compatibility with rule management software (oinkmaster/pulledpork /etc) and allows Sagan to correlate log events with your Snort IDS/IPS system. Sagan can also write to Snort IDS/IPS databases via Unified2/Barnyard2. Sagan is compatible with all Snort "consoles".”[1]

Sagan is part of the default install and the services are started and configured during the system installation. The services can be enabled after the system has been installed if they are not running by executing pkgtool → Setup → Services (press spacebar) and select rc.sagan and rc.barnsagan and save the changes. Either reboot the sensor or execute the scripts to start them. (/etc/rc.d/rc.sagan start and /etc/rc.d/rc.barnsagan start).

The events processed and detected by Sagan are sent to the Sguil database.

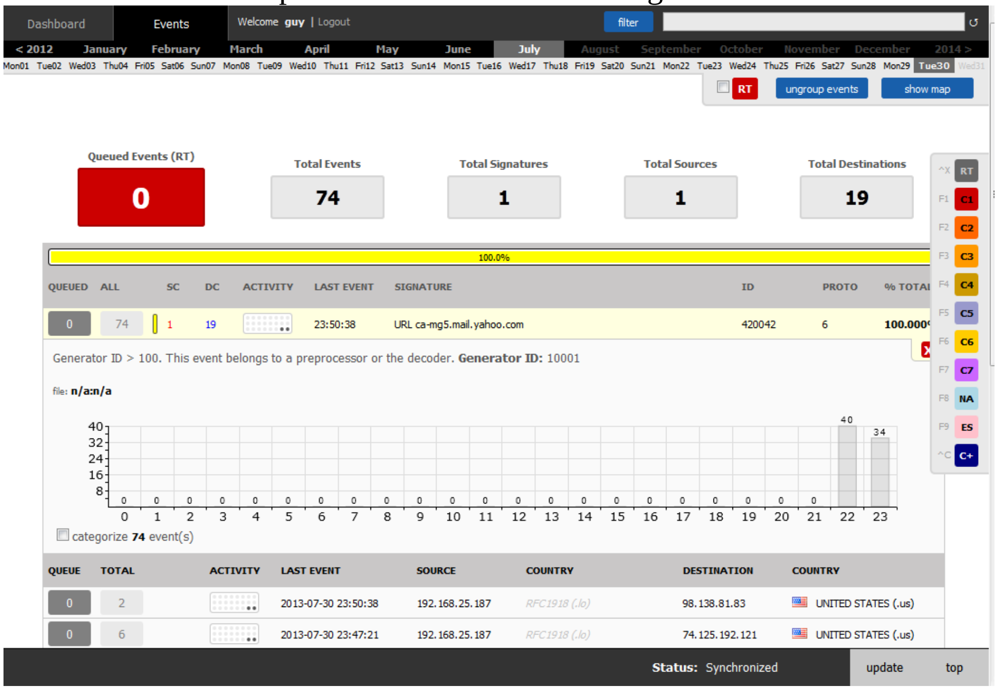
[1] http://sagan.quadrantsec.com

SQueRT – A Simple Query and Report Tool

"Squert is a web application that is used to query and view event data stored in a Sguil database (typically IDS alert data). Squert is a visual tool that attempts to provide additional context to events through the use of metadata, time series representations and weighted and logically grouped result sets."[2]

Squert is part of the default installation but access to its interface is only available after install the web components. See this section to install the webserver.

Here is an example of an output using Squert which gives graphs and summarized the data. The author has posted a list of all the existing screens which can be viewed here.



Queued Events (RT)	Total Events	Total Signatures	Total Sources	Total Destinations
0	74	1	1	19

QUEUED	ALL	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE	ID	PROTO	% TOTAL
0	74	1	19		23:50:38	URL ca-mg5.mail.yahoo.com	420042	6	100.000

Generator ID > 100. This event belongs to a preprocessor or the decoder. **Generator ID:** 10001

file: **n/a:n/a**

□ categorize **74** event(s)

QUEUE	TOTAL	ACTIVITY	LAST EVENT	SOURCE	COUNTRY	DESTINATION	COUNTRY
0	2		2013-07-30 23:50:38	192.168.25.187	RFC1918 (.lo)	98.138.81.83	UNITED STATES (.us)
0	6		2013-07-30 23:47:21	192.168.25.187	RFC1918 (.lo)	74.125.192.121	UNITED STATES (.us)

Status: Synchronized

[2] http://www.squertproject.org

Webmin Configuration

Webmin is a secure remote sensor and console manager. For example, the IDS can be remotely managed via an SSL enabled browser to manage MySQL, Sguil, the logs, the entire Suricata IDS including its plugins, the rules, the configuration files, the restart the sensor services, view a daily Sguil report, etc. It is quite versatile and very easy to use for those who prefer using a GUI to manage their sensor.

After you log into Webmin, to manage the server and the Snort sensor is located in the **Servers** section. You then are going to see Snort IDS Admin eth1 to eth4, etc. The eth1 and eth2 are the only two-interface preconfigured in Webmin.

Configuring Webmin

You need to change the default account password before making this system operational. The default account is **admin** and the default password is **admin**. Change the default admin account password the following manner. At the sensor command line console do:

```
/usr/local/webmin/changepass.pl /etc/webmin admin newpassword
```

The Webmin service can be started and stopped this way:

```
/etc/webmin/start
/etc/webmin/stop
```

Access is via SSL this way:

https://yourIPaddress:10000

A FAQ is available on the Webmin site at:

http://www.webmin.com/faq.html

Configuring Barnyard2 to send to Sguil database

Barnyard Version 2.1.13
13 May 2013 (build 327)

Barnyard unified logging output processor is used with this system to process all the data from the Suricata sensor. The default scripts are setup to process the unified logs to binary logs format and can also be configured to process MySQL database logging as well as syslog. In order to use Barnyard with MySQL, you must configure the output process as per the instructions below.

This Barnyard binary has been pre-built with the following options:

--with-tcl	Support for TCL
--enable-ipv6	IPv6 support
--enable-gre	GRE support
--enable-mpls	MPLS support
--enable-aruba	Aruba support
--with-mysql	MySQL support

Configure Barnyard to Forward Data to Sguil Database

The Suricata output processor is enable with **unified2** filename merged.log, limit 128 in the **suricata.yaml** to log to the /usr/local/suricata/log/suricata directory.

The default account is Sguil MySQL database account is *sguil* and the password is *password*. These can be changed if you desire but the whole sensor is configured to work with this account with the following minor configuration changes. These changes can be done at the command line or via Webmin in the Server tab. Configure Barnyard to forward data from a sensor to the Sguil database in the following manner:

```
vi /usr/local/barnyard/etc/barnyard.conf                (External interface)

# set the hostname (used for the sguil db output plugin)
config hostname: shadow

# Converts data from the dp_log plugin into standard pcap format
# Argument: <filename>

output log_pcap
```

Enable Data Insert into Sguil Database

```
# Use this configuration if using Sguil
```

```
output sguil: agent_port 7735 (eth1)
```

- Save the file and restart Barnyard

Note: Barnyard connects to the database via the suricata_agent startup scripts. The snort_agent configuration files are located in /etc/sguil in the suricata_agent_eth1.conf files in the *SET BY_PORT 7735* and require no other configurations.

```
/etc/rc.d/rc.barnyard restart
```

- Ensure the Barnyard processes are running

```
ps –aef |grep barnyard
```

Configure Sensor to report to a Sguil Server

If your configuration includes a sensor and a remote database server, you must configure the sensor to know where the database server is located. To allow the sensor to communicate with a Sguil database server, follow these steps:

- Log into Webmin
- Select the Server tab
- Select Sguil Sensor Controls
    - Select Snort Agent eth1 config file or Snort Agent bond0 config file
    - Select PADS Agent eth1 config file
    - Select PCAP Agent eth1 config file
    - Select SANCP Agent eth1 config file
    - Select httpry Agent eth1 config file

- Modify "**set SERVER_HOST 127.0.0.1**" and change the IP to the Sguil Server database IP and save changes

Configure httpry agent /etc/ httpry_agent.exclude

There is an exclusions file which can be used in one of two ways:

File httpry_agent_eth1.tcl located in /usr/local/sguil/bin (default set to 0)

1) If INVERT_MATCH is set to 0 in httpry_agent_eth1.tcl anything that matches an entry in httpry_agent.exclude will be ignored.

2) If INVERT_MATCH is set to 1 in httpry_agent_eth1.tcl anything that matches an entry in httpry_agent.exclude will be sent to Sguild.

Example 1: Match everything from the following TLD's (INVERT_MATCH set to 1)

```
*.ca
*.ru
*.cn
```

Example 2: Ignore everything from the following FQDN's (INVERT_MATCH set to 0)

```
*.facebook.com
*.dropbox.com
*.twitter.com
```

Notes

1) Unless you are doing extensive or very specific filtering (*.ca, *.com, *.net, *.org...) then you will want Sguild to autocat these events. These events are prefixed with "URL" so something like this will do:

```
none||ANY||ANY||ANY||ANY||ANY||ANY||%%REGEXP%%^URL||1
```

2) The events use a signature ID of 420042, event class "misc-activity"

- Restart **Restart snort_agent_eth1** or **Restart snort_agent_bond0**

Local sid Rule Mapping

Note: If you are going to create some local rules (i.e. local-eth1.rules) you MUST include the SID and the SID name in the /usr/local/snort/rules/local-sid.map and can be done via Webmin. These must match the information put in the rule file as follow:

```
9001 || Local task rule
9002 || TCP connections to TCP 3127
```

Procedures for Snort Oinkmaster Updates

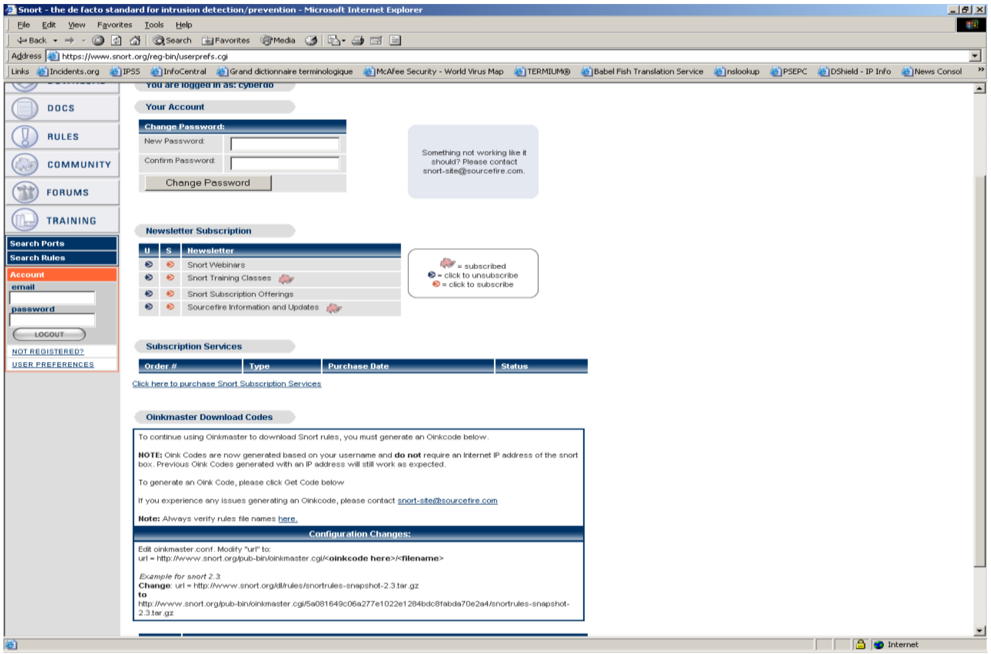
Oinkmaster

You should read the FAQ on how to configure oinkmaster to download the signature updates. http://oinkmaster.sourceforge.net/

Register with Snort to get an account

https://www.snort.org/downloads/#rule-downloads

You will receive your account information via e-mail
Login the site as per e-mail and password
Change your password if you want
In order for Oinkmaster to download the updated rules, you must generate a site string using the Get Code at the bottom of the page


Snort - the de facto standard for intrusion detection/prevention - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address https://www.snort.org/reg-bin/userprefs.cgi
Links Incidents.org IPSS InfoCentral Grand dictionnaire terminologique McAfee Security - World Virus Map TERMIUM® Babel Fish Translation Service nslookup PSEPC DShield - IP Info News Consol
DOCS
RULES
COMMUNITY
FORUMS
TRAINING
Search Ports
Search Rules
Account
email
password
LOGOUT
NOT REGISTERED?
USER PREFERENCES
Change Password
Internet

You are logged in as: cyberdo

Your Account

Change Password:
New Password:
Confirm Password:

Something not working like it should? Please contact snort-site@sourcefire.com.

Newsletter Subscription

U	S	Newsletter
		Snort Webinars
		Snort Training Classes
		Snort Subscription Offerings
		Sourcefire Information and Updates

= subscribed
= click to unsubscribe
= click to subscribe

Subscription Services

Order #	Type	Purchase Date	Status

Click here to purchase Snort Subscription Services

Oinkmaster Download Codes

To continue using Oinkmaster to download Snort rules, you must generate an Oinkcode below.

NOTE: Oink Codes are now generated based on your username and **do not** require an Internet IP address of the snort box. Previous Oink Codes generated with an IP address will still work as expected.

To generate an Oink Code, please click Get Code below

If you experience any issues generating an Oinkcode, please contact snort-site@sourcefire.com

Note: Always verify rules file names here.

Configuration Changes:

Edit oinkmaster.conf. Modify "url" to:
url = http://www.snort.org/pub-bin/oinkmaster.cgi/<**oinkcode here**>/<**filename**>

Example for snort 2.3
Change: url = http://www.snort.org/dl/rules/snortrules-snapshot-2.3.tar.gz
to
http://www.snort.org/pub-bin/oinkmaster.cgi/5a081649c06a277e1022e1284bdc8fabda70e2a4/snortrules-snapshot-2.3.tar.gz

As per the instructions at the bottom of the page, copy the new download path with your sting in it into oinkmaster.conf

Path (code is invalid, it is just an example):

url = http://www.snort.org/pub-bin/oinkmaster.cgi/**5a081649c06a277e1022e1284bdc8fabda70e2a4**/snortrules-snapshot-Current.tar.gz

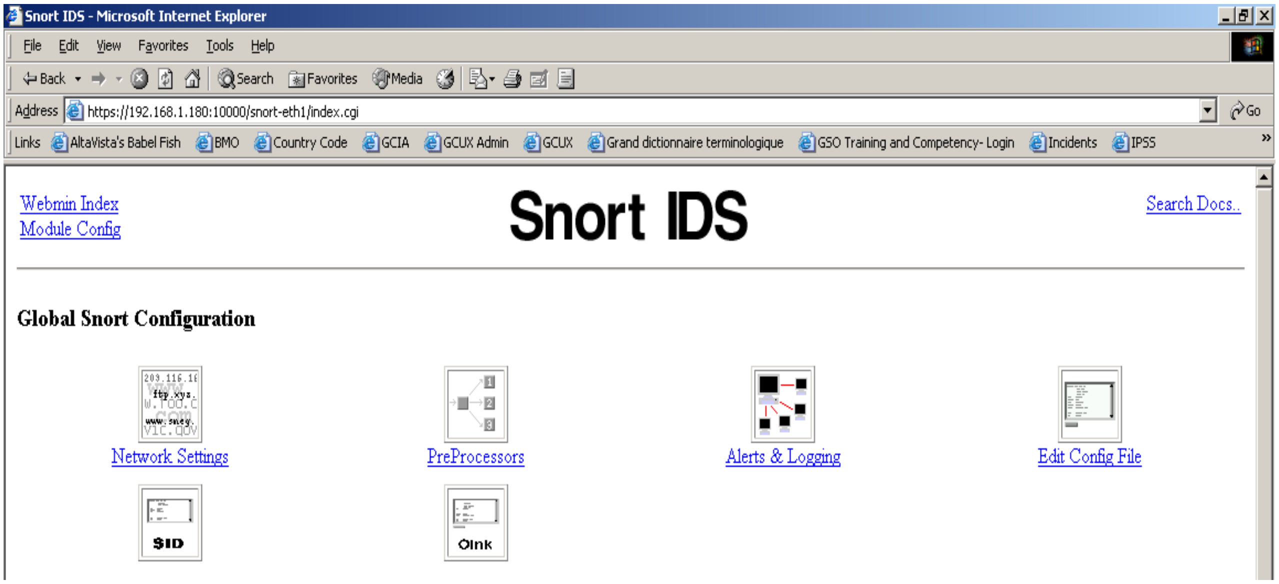
- The bold portion is where your code goes.
- The file is located at: /usr/local/snort
- The oinkmaster.conf file can be updated using vi or with Webmin
- If using Webmin, goto Servers, Snort IDS Admin eth1, select Oink
- Find url = http://www.snort.org/dl/rules/snortrules-snapshot-2.3.tar.gz

Change it to:

url = http://www.snort.org/pub-bin/oinkmaster.cgi/**INSERT YOUR ACCOUNT HERE**/snortrules-snapshot-Current.tar.gz

Webmin screenshots below

Snort IDS Main Screen

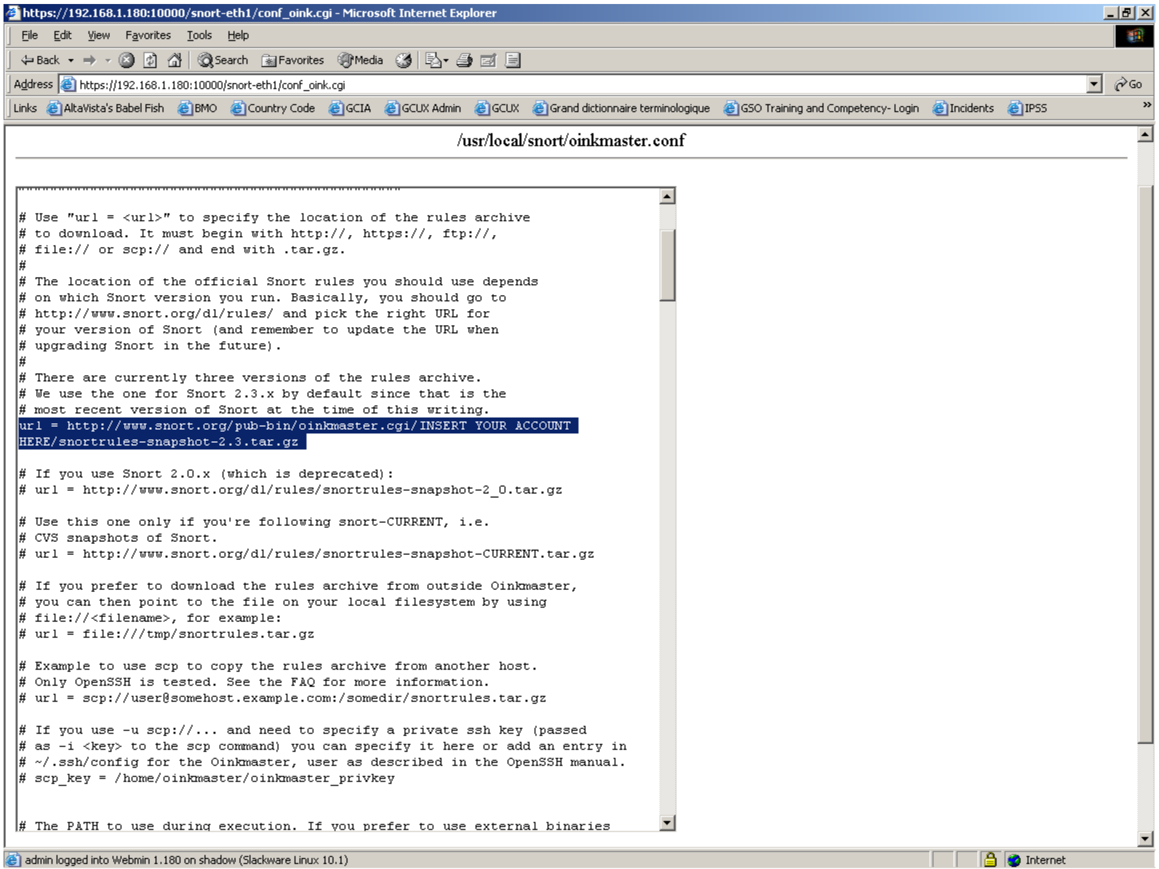

Snort IDS - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links AltaVista's Babel Fish BMO Country Code GCIA GCUX Admin GCUX Grand dictionnaire terminologique GSO Training and Competency- Login Incidents IPSS
Webmin Index
Module Config
Search Docs..
Network Settings
PreProcessors
Alerts & Logging
Edit Config File
$ID
Oink

Address https://192.168.1.180:10000/snort-eth1/index.cgi

Snort IDS

Global Snort Configuration

Edit Oinkmaster


https://192.168.1.180:10000/snort-eth1/conf_oink.cgi - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links AltaVista's Babel Fish BMO Country Code GCIA GCUX Admin GCUX Grand dictionnaire terminologique GSO Training and Competency- Login Incidents IPSS
Internet

Address https://192.168.1.180:10000/snort-eth1/conf_oink.cgi

/usr/local/snort/oinkmaster.conf

```
# Use "url = <url>" to specify the location of the rules archive
# to download. It must begin with http://, https://, ftp://,
# file:// or scp:// and end with .tar.gz.
#
# The location of the official Snort rules you should use depends
# on which Snort version you run. Basically, you should go to
# http://www.snort.org/dl/rules/ and pick the right URL for
# your version of Snort (and remember to update the URL when
# upgrading Snort in the future).
#
# There are currently three versions of the rules archive.
# We use the one for Snort 2.3.x by default since that is the
# most recent version of Snort at the time of this writing.
url = http://www.snort.org/pub-bin/oinkmaster.cgi/INSERT YOUR ACCOUNT
HERE/snortrules-snapshot-2.3.tar.gz

# If you use Snort 2.0.x (which is deprecated):
# url = http://www.snort.org/dl/rules/snortrules-snapshot-2_0.tar.gz

# Use this one only if you're following snort-CURRENT, i.e.
# CVS snapshots of Snort.
# url = http://www.snort.org/dl/rules/snortrules-snapshot-CURRENT.tar.gz

# If you prefer to download the rules archive from outside Oinkmaster,
# you can then point to the file on your local filesystem by using
# file://<filename>, for example:
# url = file:///tmp/snortrules.tar.gz

# Example to use scp to copy the rules archive from another host.
# Only OpenSSH is tested. See the FAQ for more information.
# url = scp://user@somehost.example.com:/somedir/snortrules.tar.gz

# If you use -u scp://... and need to specify a private ssh key (passed
# as -i <key> to the scp command) you can specify it here or add an entry in
# ~/.ssh/config for the Oinkmaster, user as described in the OpenSSH manual.
# scp_key = /home/oinkmaster/oinkmaster_privkey


# The PATH to use during execution. If you prefer to use external binaries
```

admin logged into Webmin 1.180 on shadow (Slackware Linux 10.1)

Suricata with Sguil Files and Scripts

/etc/rc.d	All system start/stop scripts
/etc/rc.d/rc.K	Kill all system script
/etc/rc.d/rc.S	Start up script for single-user mode
/etc/rc.d/rc.M	Start up script for multi-user mode
/etc/rc.d/rc.snort	Snort start/stop script
/etc/rc.d/rc.barnyard	Barnyard start/stop script
/etc/rc.d/rc.sguild	Sguil database start/stop script
/etc/rc.d/sancpd_eth1	sancp start/stop script for eth1
/etc/rc.d/pads_agent_eth1	Sguil PADS agent start/stop script to connect to database
/etc/rc.d/pcap_agent_eth1	Sguil pcap agent start/stop script to connect to database
/etc/rc.d/sancp_agent_eth1	Sguil sancp agent start/stop script to connect to database
/etc/rc.d/snort_agent_eth1	Sguil snort agent start/stop script to connect to database
/etc/rc.d/rc.mysqld	MySQL database script
/etc/rc.d/rc.local	Script for all other configuration
/etc/issue	Banner message
/etc/motd	Banner message
/etc/rc.d/rc.firewall	Setup firewall
/var/adm/messages	General log file
/var/adm/syslog	Syslog file
/var/run	Various server pid files
/usr/local/suricata	Suricata IDS
/usr/local/ suricata	suricata t directory files
/usr/local/ suricata /etc	suricata configuration scripts
/usr/local/ suricata /rules	suricata rules
/usr/local/ suricata /log	suricata log directory
/usr/local/ suricata /bin	suricata sensor binary
/usr/local/barnyard	Barnyard directory files (including Barnyard lock file)
/usr/local/barnyard/etc	Barnyard configuration scripts
/usr/local/barnyard/log	Barnyard log directory
/usr/local/barnyard/bin	Barnyard binary
/usr/local/etc/pads.conf	Pads configuration file
/etc/sguild	Sguil server configuration scripts
/etc/sguil	Sguil sensor agents configuration scripts.
/etc/sguild/incident_report.tcl	Daily Sguil web report output in Webmin/Servers
/usr/local/sguil/bin	Sguil logging scripts (log_packets_eth1.sh)
/usr/local/sguil/archive	Sguil pcap archives from sensors (delete regularly)
/LOG/external	Sguil log directory (eth1)
/LOG/external/dailylogs	Sguil daily logs collected by log_packets_eth1.sh
/LOG/external/sancp	sancp log dump directory
/LOG/external/pads	PADS log dump directory
/LOG/nfsen	Netflow configuration files and logs
/LOG/passivedns	Passive DNS logs and configuration file
/LOG/passivedns/skiplist.txt	Passive DNS list of skipped domains from DNS insertion

/LOG/pehunter	Used to extract Windows executables from packets
/LOG/internal	Sguil log directory (eth2)
/usr/local/mysql	MySQL database tables
/usr/local/SHADOW	Shadow directory files
/LOG/RAW/gmt	Shadow log files
/usr/local/NGREP	Network Grep sensor files
/root/sguil_pcap.sh	Sguil pcap files search script for /LOG/external/dailylogs
/root/epoch.pl	Convert Snort pcap files epoch time to YYYYMMDDHH

sancp custom queries

SID 1 – shadow
SID 2 – Snort1

Source and Destination Port Search

```
WHERE sancp.start_time > '2005-01-05' AND  (sancp.src_port  = '11768') OR
(sancp.dst_port = '11768')  LIMIT 500
```

Source and Destination Port Search for Only SID 2

```
WHERE sancp.sid= 2 AND sancp.start_time > '2005-01-20' AND  (sancp.src_port  =
'15118') OR (sancp.dst_port = '15118')  LIMIT 500
```

Single Day Source and Destination IP Search for SID 2 only

```
WHERE sancp.sid=2 AND sancp.start_time > '2005-01-15' AND sancp.end_time <
'2005-01-16' AND  (sancp.src_ip = INET_ATON('192.168.158.186') OR sancp.dst_ip =
INET_ATON('192.168.158.186'))  LIMIT 500
```

Specific Source IP

```
WHERE sancp.start_time > '2005-03-15' AND  (sancp.src_ip =
INET_ATON('192.168.8.89'))  LIMIT 500
```

Specific Sensor (sid=2) and Source Port

```
WHERE sancp.sid=2 AND sancp.start_time > '2005-03-30' AND  (sancp.src_port='53')
LIMIT 500
```

Suricata IDS configuration

Suricata Version 2.0.6
2 June 2015

Note about Sguil: You can install any of the 3 packages (sguil, sguildb and sensguil) available on the CD which has been expressly built to run on this sensor. Check the document called **sguil.pdf** in the rel_note on how to configure each of the packages. Snort is configured to automatically use Barnyard through its output log_unified2 function.

Requires libdnet 1.11 for this version. This Suricata sensor binary has been pre-built with the following options:

--enable-profiling	Enable performance profiling
--enable-unix-socket	Enable unix sockets
--enable-nfqueue	Enable NFQUEUE for inline IDP
--enable-nflog	Enable libnetfiler_log support
--enable-af-packet	Enable AF_PACKET support
--enable-lua	Enable Lua support
--enable-luajit	Enable Luajit support
--enable-geoip	Enable GeoIP support

The necessary files have been installed into the /usr/local/suricata directory.

Suricata configuration files is located in /usr/local/suricata/etc

Test the configuration by running the ./**check_suricata_eth1** script. If the this is successful, start the sensor

- **/etc/rc.d/rc.suricata start** script to starts the sensor
- suricata --list-runmodes shows all possible modes

Sguil Server Troubleshooting

Note: All these checks must be done as root

1- If you are unable to login the Sguil server via the Sguil client, check to see if the Sguil daemon are running on the server. Execute the following command:

```
ps –aef| grep tclsh
```

There should be 3 lines showing the services running. If you only see one, kill that service before restarting the Sguil daemon. Execute the following and repeat the previous command:

```
pkill tclsh
```

The Sguil database should have 2 ports open to provide access to the client and the sensor. Execute the following command:

```
netstat –an | grep 773*
```

You should see TCP port 7734 used by the client and TCP port 7736 used by the sensor (i.e. snort, sancp, pads, pcap agent service). If one is missing, restart the Sguil daemon:

```
/etc/rc.d/sguild stop
/etc/rc.d/sguild start
```

2- If the Sguild daemon won't start, check if the MySQL database is running:

```
ps –aef | grep mysqld
```

If the database isn't running, restart the database:

```
/etc/rc.d/mysqld stop
/etc/rc.d/mysqld start
```

3- If you suspect the data from a sensor is not going into the database, you can stop the Sguil service and manually observe the traffic. Do the following to watch the traffic in realtime:

```
/etc/rc.d/sguild stop
/etc/sguild/sguild
```

If there are any errors, Sguil is going to crash where the error occurs. If no errors are observed, use CTRL-C to stop and restart the service.

The database has a cronjob that should be cleaning the database logs on the first of each month at 1 am. If for some reasons that Sguil daemon does not restart, it is possible the database has not been cleaned and you may want to run the script to see if this will solve the issues. The script will stop the Sguil daemon if running, keep the past 30 days of logs and restart the Sguil Daemon. Run the following script and check to see if the Sguil daemon is running:

```
/etc/sguild/cleansguil.php
ps –aef | grep tclsh
```

However, if you get the following error:

```
root@sensor:/etc/sguild# ./sguild
pid(2182)  Loading access list: /etc/sguild/sguild.access
pid(2182)  Sensor access list set to ALLOW ANY.
pid(2182)  Client access list set to ALLOW ANY.
pid(2182)  Adding AutoCat Rule: ||ANY||ANY||ANY||ANY||ANY||ANY||tag:
Tagged Packet||1
pid(2182)  Connecting to 127.0.0.1 on 3306 as sguil
pid(2182)  MySQL Version: version 4.1.13-log
pid(2182)  SguilDB Version: 0.11
pid(2182)  Creating event MERGE table.
pid(2182)  Creating tcphdr MERGE table.
pid(2182)  Creating udphdr MERGE table.
pid(2182)  Creating icmphdr MERGE table.
pid(2182)  Creating data MERGE table.
ERROR: loaderd: You appear to be using an old version of the sguil Database schema
that does not support the MERGE sancp table. Please see the CHANGES document for
more information.
```

If you get the above error, you can execute this script:

```
/root/scripts/fix_mysql.sh
```

If the script doesn’t exist, execute the following commands:

```
/etc/rc.d/rc.sguil stop
mysql –usguil –psguil –h 127.0.0.1 –D sguildb
mysql> DROP TABLES sancp, event, tcphdr, udphdr, icmphdr, data;
```

That will remove the MERGE tables only. Your data will still be there
and the MERGE tables will get recreated when sguild starts.

```
/etc/rc.d/rc.sguil start
```

Login the client

Sguil Web Reports

The server containing the database creates by default a daily web report available through Webmin. Login Webmin in the report section https://server:10000/reports/ and look under Sguil_Reports. These reports are generated at 1200 daily on the traffic reported on the previous day. The reports are based on the events processed under the 7 incidents categories. Since all the events for the previous day won't be processed before the report is automatically generated, you can manually reprocess the previous' day report with the following command:

```
/etc/sguil/incident_report.tcl --start yesterday --end today
```

You can manually generate reports based on dates as well. Here is an example to create a monthly report for the previous month.

```
/etc/sguil/incident_report.tcl --start “2006-09-01” --end today “2006-10-01”
```

Setting bond0 networking

In order to setup eth1 and eth2 as bond0, you need to copy the following scripts:

This will enable all the bond0 scripts:
- cp /etc/rc.d/rc.local_bond0 /etc/rc.d/rc.local

- vi /usr/local/snort/etc/snort.external.nic
- Change the eth1 to bond0 and save the changes

- vi /usr/local/snort/etc/snort.internal.nic
- Delete eth2 from this file (dd and Shift ZZ). Save and exit

Change the log_packet script configuration in the root cron:

- crontab –e
- Commend out the log_packets script for eth1
- Uncomment the log_packets script for bond0

Reboot the system to enable bond0 interface

Setting bond0 networking for VLANs

This solution is used to remove the vlan IDs from the traffic and bound it together under one interface. This example removes the vlan tags and merges the traffic with regular traffic. It removes the VLAN tags from vlan 502 and 202 and merge the traffic with regular traffic. Use all the bond0 scripts instead of eth1 scripts for your sensor to work properly.

```
# Starting bond0

echo "Fire up bond0 now to start vlan collection and remove tags..."
/sbin/vconfig add eth1 502
/sbin/vconfig add eth1 202
/sbin/modprobe bonding
/sbin/ifconfig bond0 promisc -arp up
/sbin/ifenslave bond0 eth1 eth1.502 eth1.202
```

Setting up SoftFlow for Traffic Monitoring

The sensor contains the binary to monitor the network traffic using a tool called softflow. The sensor has the two binaries located in /usr/local/sbin

There is a good article on how to configure the software posted here:
http://wirewatcher.wordpress.com/2009/07/10/visualising-sguil-session-data-with-netflow/

To enable softflow on the sensor, edit the /etc/rc.d/rc.local script and enable where you want to send the softflow data. Here is an example how to startup softflow to send the data to a localhost. The IP can be changed to go to another system where the data will go.

```
/usr/local/sbin/softflowd -i eth1 -t maxlife=60 -n 127.0.0.1:9996
```

As suggested in the article, you can use ManageEngine’s NetFlow Analyzer to collect and report on the data. There is a few 30 days trial but after 30 days you can still monitor 2 interfaces with a few features disable.

http://www.manageengine.com/products/netflow/download.html

Background Information about this Setup

How to manually mount a CD-ROM or diskette

To manually mount the CD-ROM do:

```
mkdir /cdrom                          (create cdrom directory)
mount /dev/hdc /cdrom -t iso9660      (mount the cdrom)
umount /cdrom                         (un-mount the cdrom)
```

The cdrom maybe hdb, hdc or hdd depending where it has been installed in the computer. To find out which device is the CD-ROM, do *dmesg |more*

To manually mount the floppy do:

```
mkdir /floppy                         (create floppy directory)
mount /dev/fd0 /floppy -t vfat        (mount the floppy)
umount /floppy                        (un-mount the floppy)
```

Operating System Patches

The Slackware web site should be monitored for any new patches that should be applied to the sensor. The site is http://www.slackware.com

The security list is available at:

http://www.slackware.com/security/list.php?l=slackware-security&y=2012

Slackware Patch Maintenance Script

http://darklinux.net/slackupdate/

Patches can be maintained and downloaded by running the /root/slackupdate.sh script. This script will check for any package that are available for update and saves them in /tmp/slackupdate. To install the patch updates as follow:

```
telinit 1
cd /tmp/slackupdate
upgradepkg <patch>.txz
telinit 3
```

Note: The original script has been modified to work with 64-bits OS.

The message of the day changed to reflect more proactive security (Pre-configured with this installation):

- vi /etc/motd

```
*******************************************************************
This is a controlled access system.
This station is monitored at all times.
Only authorized users may connect
*******************************************************************
```

- cp /etc/motd /etc/issue

Update rc.local to start local applications:

O- vi /etc/rc.d/rc.local and add the following services

```
#!/bin/sh

# /etc/rc.d/rc.local:  Local system initialization script.

# Put any local setup commands in here:
# Starting eth1
echo "Fire up eth1 now to start Shadow collection..."
/sbin/ifconfig eth1 promisc -arp
/sbin/ifconfig eth1 up

# Turning off NIC offload features:
echo "Turning off NIC offload features before loading sensor..."
/usr/sbin/ethtool -K eth1 rx off
/usr/sbin/ethtool -K eth1 tx off
/usr/sbin/ethtool -K eth1 sg off
/usr/sbin/ethtool -K eth1 tso off
/usr/sbin/ethtool -K eth1 ufo off
/usr/sbin/ethtool -K eth1 gso off
/usr/sbin/ethtool -K eth1 gro off
/usr/sbin/ethtool -K eth1 lro off

# Starting eth2
#echo "Fire up eth2 now to start Shadow collection..."
#/sbin/ifconfig eth2 promisc -arp
#/sbin/ifconfig eth2 up

# This solution is used with network taps to agregate the
# date (send and receive) into one logical interface.
# Starting bond0
#echo "Fire up bond0 now to start Shadow collection..."
#/sbin/modprobe bonding
```

```
#/sbin/ifconfig bond0 promisc -arp up
#/sbin/ifenslave bond0 eth1
#/sbin/ifenslave bond0 eth2

# This solution is used to remove the vlan IDs from the
# traffic and bound it together under one interface.
# This example remove the vlan tags and merge the traffic
# with regular traffic.
# Starting bond0
#echo "Fire up bond0 now to start vlan collection and remove tags..."
#/sbin/vconfig add eth1 502
#/sbin/vconfig add eth1 202
#/sbin/modprobe bonding
#/sbin/ifconfig bond0 promisc -arp up
#/sbin/ifenslave bond0 eth1 eth1.502 eth1.202

#echo "Starting shadow sensor..."
#/usr/local/SHADOW/sensor/start_logger.pl gmt

# Uncomment any of these to start Network Grep sensor

#echo "Starting Network Grep sensor..."
#/usr/local/NGREP/sensor/start_ngrep.pl kazaa
#/usr/local/NGREP/sensor/start_ngrep.pl gnutella
#/usr/local/NGREP/sensor/start_ngrep.pl dir_c

echo "Starting Webmin..."
/etc/webmin/start

# echo "Starting Stunnel..."
#/usr/sbin/stunnel

echo "Starting firewall..."
/etc/rc.d/rc.firewall

echo "Starting Sguild server..."
/etc/rc.d/rc.sguild start

echo "Starting Sancp system for eth1..."
/etc/rc.d/sancpd_eth1 start

echo "Starting PADS for eth1..."
/usr/local/bin/pads -c /usr/local/etc/pads.conf -D

#echo "Starting Sancp system for bond0..."
#/etc/rc.d/sancpd_bond0 start
```

```
echo "Starting Sguil packet logger for eth1..."
/usr/local/sguil/bin/log_packets_eth1.sh start

#echo "Starting Sguil packet logger for bond0..."
#/usr/local/sguil/bin/log_packets_bond0.sh start

echo "Starting all Sguil services..."
/etc/rc.d/sancp_agent_eth1 start
/etc/rc.d/snort_agent_eth1 start
/etc/rc.d/pads_agent_eth1 start
/etc/rc.d/pcap_agent_eth1 start

echo "Starting Snort sensor and Barnyard output processor..."

if [ -x /etc/rc.d/rc.snort ]; then
  . /etc/rc.d/rc.snort start
  . /etc/rc.d/rc.barnyard start
fi

# All done.
```

Update cronjob to start various Sguil components, update time, and cut new logs each hour (Pre-configured on CD and relative of what is installed):

Crontab running as Root

```
# Internet date
17 23 * * * /usr/sbin/ntpdate time-a.nist.gov > /dev/null 2>1&
18 23 * * * /sbin/hwclock -- systohc > /dev/null 2>1&

# Clean and optimize Database Sguil Tables on the 1st of the month
0 2 1 1-12 * /etc/sguild/cleansguil.php > /dev/null 2>1&

# Restart snort every night at midnight after update Snort
# signatures have been downloaded

5 0 * * * /etc/rc.d/rc.snort stop > /dev/null 2>1&
5 0 * * * /etc/rc.d/rc.barnyard stop > /dev/null 2>1&
7 0 * * * /etc/rc.d/rc.snort start > /dev/null 2>1&
7 0 * * * /etc/rc.d/rc.barnyard start > /dev/null 2>1&

# Cut a new Sguil log on a hourly basis
# The log_packet_eth2.sh must be enabled if using two cards for the IDS
0 * * * * /usr/local/sguil/bin/log_packets_eth1.sh restart > /dev/null 2>1&
#0 * * * * /usr/local/sguil/bin/log_packets_eth2.sh restart > /dev/null 2>1&
#0 * * * * /usr/local/sguil/bin/log_packets_bond0.sh restart > /dev/null 2>1&
```

```
# Cut a new Sguil log on a hourly basis
# The log_packet_eth2.sh must be enabled if using two cards for the IDS
0 * * * * /usr/local/sguil/bin/log_packets_eth1.sh restart > /dev/null 2>1&
#0 * * * * /usr/local/sguil/bin/log_packets_eth2.sh restart > /dev/null 2>1&
#0 * * * * /usr/local/sguil/bin/log_packets_bond0.sh restart > /dev/null 2>1&

# Create a softlink to the logs in /LOG/external/dailylogs/$date
# to be used with the sguil_pcap.sh script
1 * * * * /root/epoch_eth1.pl >/dev/null 2>1&
#1 * * * * /root/epoch_eth2.pl >/dev/null 2>1&

# Sguil Daily incident report
# This report is saved in /usr/local/webmin/reports/Sguil_Reports
0 12 * * * /etc/sguild/incident_report.tcl --start yesterday --end today > /dev/null 2>1&

# Create Sguil Report web page
15 * * * * /usr/local/webmin/reports/ExplorerIndex.pl > /dev/null 2>1&
```

Crontab running as Snort

```
# This crontab runs at 1 am to updated the Snort signatures using the oinkmaster.pl
# script and merge the new rules into the rules directory.
#
0 0 * * * /usr/local/snort/rc.snortupdate
```

References

Snort IDS
http://www.snort.org

Webmin
http://www.webmin.com/