

Complete Sguil-0.6.0p1 Installation Guide on OpenBSD 3.8 with Mysql 5 [20060113] - Written by geek00L<geek00L[at]gmail.com>

About the Sguil

Sguil is the framework that based on Network Security Monitoring[NSM] model which not only relies totally on the capabilities of Intrusion Detection System but utilizes the abilities of security analyst. It is a must for security analyst to fully understand what is happening in the network as well as performing network forensic and investigation perfectly by combining all the components such as snort, sancp, tcpflow, p0f and etc. To understand more about sguil, visits <http://sguil.sf.net>. To get a picture of what Sguil is, you can view it via <http://sguil.sf.net/diagram.txt>.

Understanding variables

/usr/local/src - Define as directory that stores all the softwares source file.

/usr/local/stow - Define as directory that store all the softwares installed path.

\$\$SGUIL - Define where sguil source been installed, it is installed in /usr/local/stow/sguil-0.6.0p1 here.

\$\$SENSORNAME - Define as sensor monitoring interface[sguil in single box] or sensor hostname[sguil sensor in standalone box].

/nsm/sguild_data/rules/\$\$SENSORNAME - Rule directory for the reference of Sguil Server(sguild.conf)

/usr/local/snortrules-\$\$SENSORNAME - Rule directory for snort instance of each sensor(snort.conf)

Note: Creating symlink for /usr/local/snortrules-\$\$SENSORNAME to /nsm/sguild_data/rules/\$\$SENSOR if it is sguil in single box.

shell>ln -s /usr/local/snortrules-\$\$SENSORNAME /nsm/sguild_data/rules/\$\$SENSORNAME

\$\$NSM - Define as directory that stores all the nsm data - /nsm

\$\$NSM/snort_data/\$\$SENSORNAME/dailylogs - Define as a directory that populates dailylogs for each sguil sensor instance

\$\$NSM/snort_data/\$\$SENSORNAME/portscans - Define as a directory that populates portscans for each sguil sensor instance

\$\$NSM/snort_data/\$\$SENSORNAME/sancp/today - Define as directory that populates sancp data for each sensor instance

**Note: Creating symlink for \$NSM/snort-logs/\$SENSORNAME to /var/log/snort-\$SENSORNAME for convenience if it is sguil in single box.*

shell>ln -s \$NSM/snort-logs/\$SENSORNAME /var/log/snort-\$SENSORNAME

/etc/sguil - Define as directory that stores all the sguil sensor configurations. It contains the following,

*sancp.conf
sensor_agent-\$SENSORNAME.conf
barnyard-\$SENSORNAME.conf*

/etc/sguild - Define as directory that stores sguil server configurations. It contains the following,

Note: Since all the sensors share same sancp configuration, it is considered global.

/var/run/sguil - Define as directory that hold the PID files. Skip it if the system has has a sguil server or sensor configured on it.

**Note: Before installing anything, we should know what are the components needed to install Sguil Server, Sguil Sensor and Mysql DB for Sguil. If you choose to install all of them in single box, you can just download all the components mentioned below into single box. You may need wget for all the boxes to fetch components easily.*

Mysql Database Box

Ports/Packages

- wget-1.10.2.tgz
- mysql-client-4.1.15p0.tgz
- gmake-3.80p1.tgz

Sources

- mysql-5.0.18.tar.gz
- sguil-0.6.0p1.tar.gz

Sguil Server Box

Ports/Packages

- wget-1.10.2.tgz
- mysql-client-4.1.15p0(Not necessary if you decide to compile mysqltcl using Mysql 5 source)
- p0f-2.0.5
- tcpflow-0.21

Sources

- sguil-0.6.0p1.tar.gz(server only)
- mysql-5.0.18.tgz

TCL suites

- tcl8.4.11.tar.gz
- tcllib-1.8.tar.gz
- tclx-8.4.tar.gz
- tcl-1.5.tar.gz
- mysqltcl-2.51.tar.gz/mysqltcl-3.01.tar.gz(If you build mysqltcl using OpenBSD mysql client port/package, you can go for 3.01, however if you build mysqltcl using mysql 5, you should use 2.51)

Sguil Sensor Box

Ports/Packages

- wget-1.10.2.tgz
- libnet-1.0.2ap1.tgz(To compile snort with flex-resp flag)
- autoconf-2.59.tgz
- automake -1.9.6p0.tgz

Note: Autoconf and and automake are needed to get barnyard compiled.

Libraries

- libpcap-0.9.4.tar.gz
- pcre-6.4.tar.gz

Sources

- sguil-0.6.0p1.tar.gz(sensor only)
- snort-2.4.3.tar.gz
- barnyard-0.2.0.tar.gz
- sancp-1.6.1.tar.gz
- oinkmaster-1.1.tar.gz(only needed if you want automatically updates for snort rules)

Tcl Suite

- tcl-8.4.11.tar.gz
- tclx-8.4.tar.gz

Installing All the necessary ports/packages

Pre-installed ports/packages before deploying Sguil, that's the only ports/packages that I have on my Sguil system. You can choose to install them in one shot or install them when needed. I myself prefer the latter since I want to know what is needed by certain applications, anyway it's up to your preference.

Setting up PKG_PATH to ease your OpenBSD port/package installation

```
shell>echo "PKG_PATH=ftp://ftp.openbsd.org/pub/OpenBSD/snapshots/packages/i386/"
>> /root/.profile
```

```
shell>echo "export PKG_PATH" >> /root/.profile
```

Logout and relogin as root and the changes will be applied.

To install all the packages in one shoot,

```
shell>pkg_add ${PKG_PATH}autoconf-2.59.tgz automake-1.9.6p0.tgz bison-2.1p0.tgz
expat-1.95.6p1.tgz gmake-3.80p1.tgz wget-1.10.2.tgz p0f-2.0.5.tgz tcpflow-0.21.tgz libnet-1.
0.2ap1.tgz mysql-client-4.1.15p0.tgz
```

autoconf-2.59 automatically configure source code on many Un*x platforms
automake-1.9.6p0 GNU standards-compliant Makefile generator
bison-2.1p0 GNU parser generator
expat-1.95.6p1 XML 1.0 parser written in C
gettext-0.14.5p0 GNU gettext
gmake-3.80p1 GNU make
libconv-1.9.2p3 character set conversion library
libnet-1.0.2ap1 raw IP packet construction library
metaauto-0.5 wrapper for gnu auto*
p0f-2.0.5 passive OS fingerprinting tool
stow-1.3.3p0 manages software package installations with symlinks
tcpflow-0.21 tool for capturing data from TCP connections
wget-1.10.2 retrieve files from the 'net via HTTP and FTPol
mysql-client-4.1.15p0 multithreaded SQL database (client)

**Note: Stow is important since it is the Software Management System that will ease your system/application upgrades. Wget is used to fetch all the sources needed to build sgul. Mysql client is installed to get mysqltcl to compiled, it is not necessary if you choose to compiled mysqltcl using Mysql 5 client librabry.*

Mysql 5 Installation on OpenBSD-3.8

You need gmake and bison to compile Mysql 5 properly. Just install them via ports/packages.

```
shell>pkg_add ${PKG_PATH}gmake-3.80p1.tgz bison-2.1p0.tgz
```

Configuring, compiling and installing Mysql5

```
shell>cd /usr/local/src
```

```
shell>wget http://dev.mysql.com/get/Downloads/MySQL-5.0/mysql-5.0.18.tar.gz/from/http://mirror.mysql-partners-jp.biz/
```

```
shell>tar xzf mysql-5.0.18.tar.gz
```

```
shell>CC=gcc
```

```
shell>CFLAGS="-O3 -fno-strength-reduce"
```

```
shell>CXX=gcc
```

```
shell>CXXFLAGS="-O3 -fno-rtti -fno-exceptions -felide-constructors -fno-strength-reduce"
```

```
shell>export CC CFLAGS CXX CXXFLAGS
```

```
shell>./configure --prefix=/usr/local/stow/mysql-5.0.18
```

```
shell>make
```

```
shell>make install
```

To stow Mysql 5

```
shell>cd /usr/local/stow
```

```
shell>stow mysql-5.0.18
```

Mysql 5 is installed now and you need to set it up properly

```
shell>groupadd -g 5000 mysql
```

```
shell>useradd -u 5000 -g 5000 -d /home/mysql -s /bin/ksh -c "MySQL User" mysql
```

```
shell>mkdir /nsm/mysql
```

```
shell>chown -R mysql:mysql /nsm/mysql
```

```
shell>chmod 755 /nsm/mysql
```

Copy mysql configuration file to /etc,

```
shell>cp /usr/local/src/mysql-5.0.18/support-files/my-medium.cnf /etc/my.cnf
```

Installing mysql initial database.

```
shell>/usr/local/bin/mysql_install_db --basedir=/usr/local/stow/mysql-5.0.18 --user=mysql --  
ldata=/nsm/mysql
```

Running Mysql 5 at the very first time

```
shell>/usr/local/bin/mysqld_safe --user=mysql --datadir=/nsm/mysql &
```

Checking if it works,

```
shell>/usr/local/bin/mysqladmin ping  
mysqld is alive
```

Now you have Mysql 5 working proprly and it's time to setup Sguil Database.

Securing Mysql 5 Server and installing Sguil Database

Log in as root and update the root password for mysql database.

```
shell>mysql -u root mysql
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 2 to server version: 5.0.18-log
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> update user set Password=OLD_PASSWORD("r00t") where User = "root";  
Query OK, 2 rows affected (0.08 sec)  
Rows matched: 2  Changed: 2  Warnings: 0
```

```
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> exit  
Bye
```

Relogin again as root and now setting up user sguil and sguil@localhost.

```
shell>mysql -u root -p mysql
```

```
Enter password:
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 3 to server version: 5.0.18-log
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> GRANT ALL PRIVILEGES ON sguildb.* TO sguil@localhost IDENTIFIED BY  
"sguil" WITH GRANT OPTION;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT FILE ON *.* to sguil@localhost;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> update user set Password = OLD_PASSWORD("sguil") where User = "sguil";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

Importing Sguil database from sguil source.

```
shell>mysql -u sguil -p -e "CREATE DATABASE sguildb"
```

```
shell>mysql -u sguil -p -D sguildb \
< /usr/local/src/sguil-0.6.p1/server/sql_scripts/create_sguildb.sql
```

```
shell>mysql -u sguil -p -D sguildb -e "show tables"
```

Checking whethere Sguil database is loaded properly,

```
shell>mysql -u sguil -p -D sguildb -e "show tables"
```

Enter password:

```
+-----+
| Tables_in_sguildb |
+-----+
| history           |
| nessus            |
| nessus_data       |
| portscan          |
| sensor            |
| sessions          |
| status            |
| user_info         |
| version           |
+-----+
```

Note: You may find this part familiar since I have stolen it from Hanashi's instantNSM guide. I have to do it for the completeness of the documentation :P

FETCHING AND COMPILING ALL SOURCES

Since I have uploaded all the sources that needed by Sguil to the central location - http://www.dissectible.org/anonymous/Sguil_OBSD/source, it has already simplified your works to fetch all of them to /usr/local/src.

Remember to download **sguil-0.6.0p1** and **untar** it to **/usr/local/stow/** since **sguil** doesn't require installation.

Installing Libpcap-0.9.4 from source

```
shell>cd /usr/local/src
```

```
shell>tar xzf libpcap-0.9.4.tar.gz
```

```
shell>cd libpcap-0.9.4
```

```
shell>./configure --prefix=/usr/local/stow/libpcap-0.9.4
```

```
shell>make && make install
```

To stow libpcap,

```
shell>cd /usr/local/stow
```

```
shell>stow libpcap-0.9.4
```

Creating a symlink to **/usr/local/libpcap** manually.

```
shell>ln -s /usr/local/stow/libpcap-0.9.4 /usr/local/libpcap
```

Installing Pcre-6.4 from source

```
shell>tar xzf pcre-6.4.tar.gz
```

```
shell>./configure --prefix=/usr/local/stow/pcre-6.4
```

```
shell>make && make install
```

To stow pcre,

```
shell>cd /usr/local/stow
```

```
shell>stow pcre-6.4
```

Installing Libnet-1.0.2 using package/port

Installing via package

```
shell>pkg_add ${PKG_PATH}libnet-1.0.2ap1.tgz
```

Installing via port,

```
shell>cd /usr/ports/net/libnet/1.0
```

```
shell>make && make install
```

To solve the error of which libnet-config not found when compiling snort with flex-resp,

```
shell>ln -s /usr/local/bin/libnet-config-1.0 /usr/local/bin/libnet-config
```

Installing Snort-2.4.3 from source

```
shell>tar xvzf snort-2.4.3.tar.gz
```

```
shell>cd snort-2.4.3
```

```
shell>./configure --with-libnet-includes=/usr/local/include/libnet-1.0/ \  
--with-libnet-libraries=/usr/local/lib/libnet-1.0/ --enable-flexresp \  
--enable-permonitor --prefix=/usr/local/stow/snort-2.4.3
```

```
shell>make && make install
```

To stow snort,

```
shell>cd /usr/local/stow
```

```
shell>stow snort-2.4.3
```

Installing Barnyard-0.2.0 from source

**Note: Get autoconf and automake before installing barnyard*

```
shell>pkg_add ${PKG_PATH}autoconf-2.59.tgz
```

```
shell>pkg_add ${PKG_PATH}automake-1.9.6p0.tgz
```

Patching barnyard

```
shell>tar xzf barnyard-0.2.0.tar.gz
```

```
shell>cd barnyard-0.2.0
```

```
shell>export SGUIL=/usr/local/stow/sguil-0.6.0p1
```

```
shell>cp $SGUIL/sensor/barnyard_mods/op_sgUIL.* src/output_plugins
```

```
shell>cp $SGUIL/sensor/barnyard_mods/configure.in .
```

```
shell>cd src/output_plugins
```

```
shell>patch -p0 < $SGUIL/sensor/barnyard_mods/op_plugbase.c.patch
```

```
shell>export AUTOCONF_VERSION=2.59
```

```
shell>export AUTOMAKE_VERSION=1.9
```

```
shell>cd ../..
```

```
shell>autoreconf -f
```

Installing barnyard

```
shell>./configure --prefix=/usr/local/stow/barnyard-0.2.0 \  
--enable-tcl --with-tcl=/usr/local/lib/
```

```
shell>make
```

I get an error below,

```
op_sgUIL.c:53:17: tcl.h: No such file or directory and bla bla ...
```

Then I navigate op_sgUIL.c, and the line 53 ...

```
#include <tcl.h>
```

So i edit it to(depends on your tcl.h path)

```
#include "/usr/local/include/tcl.h"
```

Rerun make and it works now.

```
shell>make install
```

```
shell>cp -r ./etc /usr/local/stow/barnyard-0.2.0/etc
```

To stow barnyard,

```
shell>cd /usr/local/stow
```

```
shell>stow barnyard-0.2.0
```

Installing Sancp-1.6.1 from source

**Note: Remember to download it's patches to /usr/local/src as well.*

```
shell>tar xzf sancp-1.6.1.tar.gz
```

```
shell>cp sancp-1.6.1.fix200511.* ./sancp-1.6.1
```

```
shell>cd sancp-1.6.1
```

```
shell>patch -p1 < sancp-1.6.1.fix200511.a.patch
```

```
shell>patch -p1 < sancp-1.6.1.fix200511.b.patch
```

Edit the Makefile

```
CFLAGS = -O3 -s -I/usr/local/include -L/usr/local/lib
```

Then start to compile, you will find sancp binary installed in the current directory.

```
shell>make
```

```
shell>mkdir -p /usr/local/stow/sancp-1.6.1/bin
```

```
shell>cp sancp /usr/local/stow/sancp-1.6.1/bin
```

To stow sancp,

```
shell>cd /usr/local/stow/
```

```
shell>stow sancp-1.6.1
```

Installing p0f and tcpflow from ports

```
shell>pkg_add ${PKG_PATH}p0f-2.0.5.tgz tcpflow-0.21.tgz
```

p0f will be installed in /usr/local/sbin and tcpflow will be installed in /usr/local/bin

To standardize it so that all binaries under /usr/local/bin,

```
shell>cp /usr/local/sbin/p0f /usr/local/bin/p0f
```

Installing Tcl-8.4.11 from source

```
shell>tar xzf tcl8.4.11-src.tar.gz
```

```
shell>cd tcl8.4.11/unix
```

```
shell>./configure --prefix=/usr/local/stow/tcl-8.4.11
```

```
shell>make && make install
```

To stow tcl,

```
shell>cd /usr/local/stow
```

```
shell>stow tcl-8.4.11
```

Creating Symlink

```
shell>ln -s /usr/local/bin/tclsh8.4 /usr/local/bin/tclsh
```

Installing tcllib-1.8 from source

```
shell>tar xvzf tcllib-1.8.tar.gz
```

```
shell>cd tcllib-1.8
```

```
shell>./configure --prefix=/usr/local/stow/tcllib-1.8
```

```
shell>make && make install
```

To stow tcllib,

```
shell>cd /usr/local/stow
```

```
shell>stow tcllib-1.8
```

Installing tclx-8.4 from source

```
shell>tar xvzf tclx-8.4.tar.gz
```

```
shell>cd tclx8.4
```

```
shell>./configure --prefix=/usr/local/stow/tclx-8.4 --enable-tk=NO
```

```
shell>make && make install
```

To stow tclx,

```
shell>cd /usr/local/stow
```

```
shell>stow tclx-8.4
```

Installing tcltls-1.5 from source

```
shell>tar xzf tls1.5.0-src.tar.gz
```

```
shell>cd tls1.5
```

```
shell>./configure --prefix=/usr/local/stow/tls-1.5 --with-ssl-dir=/usr/
```

```
shell>make && make install
```

To stow tcltls,

```
shell>cd /usr/local/stow
```

```
shell>stow tls-1.5
```

Installing mysqltcl-2.51/3.01 from source

**Note: This is a little bit tricky, if you want to compile mysqltcl-2.51 and mysqltcl-3.01 perfectly, you have to install Mysql client from port/package and using it's client library. If you are comfortable with just mysql-2.51, then you can use Mysql 5 client library to compile. Both work for me, however you will get into trouble of which object can't be loaded if you compiled mysqltcl-3.01 using Mysql 5 source.*

IMPORTANT!!!!: You'll have to unstow mysql-5.0.18 before installing mysql client from port/package , this is where we see the power of stow, after destroying the symlinks by unstowing it, we can install mysql client from port/package now and restow Mysql 5 later. However we need to remain libmysqlclient.so.14.0 so that mysqltcl won't break.

Compiling mysqltcl-2.51 against Mysql 5 source

```
shell>ln -s /usr/local/lib/mysql/libmysqlclient.so.15.0 \  
/usr/local/stow/lib/mysql/libmysqlclient.so
```

```
shell>tar xzf mysqltcl-2.51.tar.gz
```

```
shell>cd mysqltcl-2.51
```

```
shell>env CC=gcc ./configure --prefix=/usr/local/stow/mysqltcl-2.51 \  
\
```

```
--with-mysql-include=/usr/local/include/mysql \  
--with-mysql-lib=/usr/local/lib/mysql --with-tcl=/usr/local/lib \  
--with-tclinclude=/usr/local/include
```

```
shell>make && make install
```

To stow myqltcl,

```
shell>cd /usr/local/stow
```

```
shell>stow myqltcl-2.51
```

Compiling myqltcl-2.51/3.01 against Mysql Client from port/package

**Note: We don't have to unstow if Mysql 5 server is installed in separated box, this is applied to Sguil in single box.*

To unstow Mysql 5,

```
shell>cd /usr/local/stow
```

```
shell>stow -D mysql-5.0.18
```

We can install mysql client from port/package by now without conflict.

Installing mysql client,

```
shell>pkg_add ${PKG_PATH}mysql-client-4.1.15p0
```

Creating symlink

```
shell>ln -s /usr/local/lib/libmysqlclient.so.14.0 /usr/local/lib/libmysqlclient.so
```

**Note: If you wish to install myqltcl-3.01 instead of 2.51, then just change the version below since both works and people always prefer latest release.*

```
shell>tar xzf myqltcl-2.51.tar.gz
```

```
shell>env CC=gcc ./configure --prefix=/usr/local/stow/myqltcl-2.51 \  
--with-mysql-include=/usr/local/include/mysql \  
--with-mysql-lib=/usr/local/lib/mysql --with-tcl=/usr/local/lib \  
--with-tclinclude=/usr/local/include
```

```
shell>make && make install
```

To stow myqltcl,

```
shell>cd /usr/local/stow
```

```
shell>stow mysqltcl-2.51
```

*Note: To avoid conflict , you have to remove mysql client package before restowing Mysql 5, however since we don't want libmysqlclient.so.14.0 to be removed, we have to rename it first, and recreate the symlink for it later after restow Mysql 5.

Destroy symlink before renaming libmysqlclient.so.14.0

```
shell>rm /usr/local/lib/libmysqlclient.so
```

Rename it,

```
shell>mv /usr/local/lib/libmysqlclient.so.14.0 /usr/local/lib/libmysqlclient.so.14.0.orig
```

Uninstall mysql client packages,

```
shell>pkg_delete mysql-client-4.1.15p0
```

Now we can restow Mysql 5,

```
shell>cd /usr/local/stow
```

```
shell>stow -R mysql-5.0.18
```

After removing mysql client package and restowing Mysql 5, we can now reinvoke back the libmysqlclient.so.14.0

```
shell>mv /usr/local/lib/libmysqlclient.so.14.0.orig /usr/local/lib/libmysqlclient.so.14.0
```

Recreate the symlink for libmysqlclient so that mysqltcl won't break.

```
shell>ln -s /usr/local/lib/libmysqlclient.so.14.0 /usr/local/lib/libmysqlclient.so
```

Sometimes you will have the errors below of which package can't be found. You can invoke it back.

```
shell>tclsh
```

```
% package require mysqltcl  
can't find package mysqltcl
```

To invoke it,

```
% package provide mysqltcl 2.51
```

Creating index with pkg_mkIndex by scanning given directory,

```
% pkg_mkIndex /usr/local/mysqltcl-2.51/lib/mysqltcl-2.51/ *.so
```

Package can be used now,

```
% package require mysqltcl  
2.51
```

Configuring Sguil Server

Creating sguil user and it's group on the system

```
shell>groupadd -g 5555 sguil
```

```
shell>useradd -u 5555 -g 5555 -md /home/sguil -c "SGUIL User" sguil
```

Creating sguil local storage

```
shell>mkdir -p /nsm/sguild_data/archive
```

```
shell>mkdir /nsm/sguild_data/rules
```

```
shell>chown -R sguil:sguil /nsm/sguild_data
```

Installing sguil to the central path,

```
shell>cd /usr/local/src
```

```
shell>tar xzf sguil-0.6.0p1.tar.gz
```

```
shell>cp -fR /usr/local/src/sguil-0.6.0p1 /usr/local/stow/
```

```
shell>cd /usr/local/stow/sguil-0.6.0p1/server/
```

Creating sguil server configuration directory

```
shell>mkdir /etc/sguild
```

Copy all the config files to /etc/sguild

```
shell>cp sguild.conf /etc/sguild/
```

```
shell>cp autocat.conf /etc/sguild/
```

```
shell>cp sguild.users /etc/sguild/
```

```
shell>cp sguild.queries /etc/sguild
```

```
shell>cp sguild.access /etc/sguild
```

```
shell>cp sguild.email /etc/sguild
```

```
shell>cp sguild.reports /etc/sguild
```

Changing file owner to sguild

```
shell>chown -R sguild:sguild /etc/sguild
```

Add user for sguild, this is important since it is required to connect to sguild to launch analyst console. It will be added to it's own database called sguild.users.

```
shell>/usr/local/stow/sguil-0.6.0p1/server/sguild -adduser sguild
```

Configuring Sguil Sensor

```
shell>mkdir -p /nsm/snort-logs/$SENSORNAME/OLD
```

```
shell>mkdir -p /nsm/snort_data/$SENSORNAME/dailylogs
```

```
shell>mkdir -p /nsm/snort_data/$SENSORNAME/sanctp/today
```

```
shell>mkdir -p /nsm/snort_data/$SENSORNAME/portscans
```

```
shell>chown -R sguild:sguild /nsm/snort-logs /nsm/snort_data
```

```
shell>ln -s /nsm/snort-logs/$SENSORNAME /var/log/snort-$SENSORNAME
```

```
shell>mkdir /var/run/sguild
```

```
shell>chown sguild:sguild /var/run/sguild
```

```
shell>mkdir /usr/local/snortrules-$SENSOR
```

```
shell>chown -R root:wheel /usr/local/snortrules-$SENSORNAME
```

**Note: Copy all the File for patching before move to proper directory, I choose to use Richard's patch to configure all the configurations file including sguild server, sensor, snort and etc. It is much more easier to use Richard's patch instead of configuring it by hand manually.*

```
shell>mkdir /usr/local/stow/sguil-0.6.0p1/richard_patch
```

```
shell>cd /usr/local/stow/sguil-0.6.0p1/richard_patch
shell>cp /usr/local/stow/sguil-0.6.0p1/server/sguild.conf ./
shell>cp /usr/local/stow/sguil-0.6.0p1/sensor/sanct/sanct.conf ./
shell>cp /usr/local/stow/sguil-0.6.0p1/sensor/sensor_agent.conf ./
shell>cp /usr/local/stow/barnyard-0.2.0/etc/barnyard.conf ./
shell>cp /usr/local/stow/sguil-0.6.0p1/sensor/log_packets.sh ./
# Fetch Richard's patches
shell>wget http://www.bejtlich.net/sensor_agent.conf.patch
shell>wget http://www.bejtlich.net/sguild.conf.patch
shell>wget http://www.bejtlich.net/snort.conf.patch
shell>wget http://www.bejtlich.net/barnyard.conf.patch
shell>wget http://www.bejtlich.net/sanct.conf.patch
shell>wget http://www.bejtlich.net/log_packets.sh.patch
shell>wget http://www.bejtlich.net/log_packets.sh.crontab
```

Before Applying Patches, you should tweak the patch to feed your need, else you can configure it manually by hand to suit your environment.

Setting barnyard.conf

```
# my barnyard config file is /etc/sguil/barnyard- $\$$ SENSOR.conf
# Set the sensor name or interface, my sensor interface is pcn1
config hostname: pcn1
# Ignore the interface name that used by acid
config interface:
# Ignore acid db filter
config filter:
# Set output plugin, comment out unused feature and add the one used by sguil
```

#out log_dump, agent_port for which port you want barnyard connect to the sensor agent

output sguil: sensor_name pcn1, agent_port 7740

Setting sancp.conf

configure your own HOME_NET

var HOME_NET 0.0.0.0

Setting sguild.conf

set SGUILD_LIB_PATH /usr/local/stow/sguil-0.6.0p1/server/lib

set DEBUG 0

set BIND_SENSOR_IP_ADDR 192.168.0.151

set RULES_DIR /nsm/sguild_data/rules

set TMPDATADIR /tmp

Set DBNAME sguildb

set DBPASS "sguil"

set DBHOST localhost

set DBPORT 3306

set DBUSER sguil

Storing Archive raw file when xscriptds are requested

set LOCAL_LOG_DIR /nsm/sguild_data/archive

set TCPFLOW "/usr/local/bin/tcpflow"

set P0F_PATH "/usr/local/bin/p0f"

Setting sensor_agent.conf

set SERVER_HOST localhost

set SERVER_PORT 7736

Port sensor_agent listens for barnyard to connect

set BY_PORT 7740

set HOSTNAME pcn1

```
set LOG_DIR /nsm/snort_data
set S4_KEEP_STATS 0
```

```
set PORTSCAN_DIR ${LOG_DIR}/${HOSTNAME}/portscans
```

Setting snort.conf

```
RULE_PATH /usr/local/snortrules-$SENSORNAME/
```

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

Setting log_packets.sh

```
HOSTNAME="pcn1"
```

```
SNORT_PATH="/usr/local/bin/snort"
```

```
LOG_DIR="/nsm/snort_data"
```

```
INTERFACE="pcn1"
```

```
OPTIONS="-u sguil -g sguil -m 122"
```

```
PID="/var/run/snort_log-${HOSTNAME}.pid"
```

*Note: If you choose to apply patches, remember to edit it to feed your needs

```
shell>patch -p0 < sensor_agent.conf.patch
```

```
shell>patch -p0 < sguil.conf.patch
```

```
shell>patch -p0 < snort.conf.patch
```

```
shell>patch -p0 < barnyard.conf.patch
```

```
shell>patch -p0 < sancp.conf.patch
```

```
shell>patch -p0 < log_packets.sh.patch
```

Copy them to the proper directory after applying patches

```
shell>cp snort.conf /nsm/sguild_data/rules/pcn1
```

```
shell>cp sguil.conf /etc/sguild/sguild.conf
```

```
shell>cp sancp.conf /etc/sguil/sancp.conf
```

```
shell>cp sensor_agent.conf /etc/sguil/sensor_agent-$$SENSORNAME.conf
```

```
shell>cp barnyard.conf /etc/sguil/barnyard-$$SENSORNAME.conf
```

```
shell>cp log_packets.sh /usr/local/bin/log_packets-$$SENSORNAME.sh
```

Edit log_packets.sh.crontab

```
00 0-23/1 * * * /usr/local/bin/log_packets-pcn1.sh restart
```

Installing log_packets crontab

```
shell>crontab -u root log_packets.sh.crontab
```

Installing Snort Rules

Create directory to hold snort rules,

```
shell>mkdir /usr/local/src/snort-rules
```

```
shell>cd /usr/local/src/snort-rules
```

Downloading snort rules

```
shell>wget http://www.snort.org/pub-bin/downloads.cgi/Download/vrt_os/snortrules-snapshot-2.4.tar.gz
```

```
shell>tar xvzf snortrules-snapshot-2.4.tar.gz
```

Copy the rules to the sgul server rules reference directory

```
shell>cp /usr/local/src/snort-rules/rules/* /nsm/sguild_data/rules/$$SENSORNAME
```

Launching SGUIL - It's time to rock!!!!

All the Startup scripts required to run sgul in single box.

Under /root directory,

mysql_start.sh

snort_start.sh

sancp_start.sh

Under /home/sguil directory,
sguild_start.sh
sensor_agent_start.sh
barnyard_start.sh

Sguil running in Order

Log in as root,

```
shell>./mysql_start.sh
```

```
shell>./snort_start.sh
```

```
shell>./sancp_start.sh
```

Log in as sguil,

```
shell>./sguild_start.sh
```

```
shell>./sensor_agent_start.sh
```

```
shell>./barnyard_start.sh
```

The content of all the scripts

/root/mysql_start.sh

```
#!/bin/sh  
# Mysql 5 DB  
# As a daemon  
/usr/local/bin/mysqld_safe --user=mysql --datadir=/nsm/mysql &  
  
# In foreground  
#/usr/local/bin/mysqld_safe --user=mysql --datadir=/nsm/mysql
```

/root/snort_start.sh

```
#!/bin/sh  
SENSOR=pcn1  
INTERFACE=pcn1  
  
# As a daemon  
snort -u sguil -g sguil -c /nsm/sguild_data/rules/$SENSOR/snort.conf -U -l  
/nsm/snort_data/$SENSOR/ -m 122 -A none -i $INTERFACE -D
```

```
# In foreground
#snort -u sguil -g sguil -c /nsm/sguild_data/rules/$SENSOR/snort.conf -U -l
/nsm/snort_data/$SENSOR/ -m 122 -A none -i $INTERFACE
```

/root/sanccp_start.sh

```
#!/bin/sh
SENSOR=pcn1
INTERFACE=pcn1
```

```
# As a daemon
sanccp -D -d /nsm/snort_data/$SENSOR/sanccp/ -i $INTERFACE -u sguil -g sguil -c
/etc/sguil/sanccp.conf > /var/log/sanccp.log
```

```
# In foreground
#sanccp -d /nsm/snort_data/$SENSOR/sanccp/ -i $INTERFACE -u sguil -g sguil -c
/etc/sguil/sanccp.conf > /var/log/sanccp.log
```

/home/sguil/sguild_start.sh

```
#!/bin/sh
SGUIL=sguil-0.6.0p1
# As a daemon
/usr/local/stow/$SGUIL/server/sguild -c /etc/sguild/sguild.conf -u /etc/sguild/sguil.users -D
```

```
# In foreground
#/usr/local/stow/$SGUIL/server/sguild -c /etc/sguild/sguild.conf -u /etc/sguild/sguild.users
```

/home/sguil/sensor_agent_start.sh

```
#!/bin/sh
SGUIL=sguil-0.6.0p1
cd /usr/local/stow/$SGUIL/sensor
# As a daemon
./sensor_agent.tcl -c /etc/sguil/sensor_agent-pcn1.conf -D
```

```
# In foreground
#./sensor_agent.tcl -c /etc/sguil/sensor_agent-pcn1.conf
```

/home/sguil/barnyard_start.sh

```
#!/bin/sh
```

```
# As a daemon
barnyard -D -c /etc/sguil/barnyard-pcn1.conf -d /nsm/snort_data/pcn1 -g /usr/local/snortrules-
pcn1/gen-msg.map -s /usr/local/snortrules-pcn1/sid-msg.map -f snort.log -w
/nsm/snort_data/pcn1/waldo.file
```

```
# In foreground
#barnyard -c /etc/sguil/barnyard-pcn1.conf -d /nsm/snort_data/pcn1 -g /usr/local/snortrules-
pcn1/gen-msg.map -s /usr/local/snortrules-pcn1/sid-msg.map -f snort.log -w
/nsm/snort_data/pcn1/waldo.file
```

Sguil Client

If you want to install Sguil client on OpenBSD 3.8, there's a installation script that works on OpenBSD-3.8, just grab it at http://www.dissectible.org/anonymous/Sguil_OpenBSD.

Basic Info of the Sguil System

System User

Username: root
password: r00t

Username: sguil
password: sguilNSM

Mysql Database

Username: root
password: r00t

Username: sguil
password: sguil

Sguild User

Username: sguil
password: sguil

Special thanks to All the lamerz in #snort-gui especially qru, hanashi, helevius, transporter and transzorp, without them producing this Sguil-0.6.0p1 installation guide would be much more harder than it is. Coincidentally my previous guide for Sguil-0.5.3 was released on 13th of Jan as well, and just right after one year the second one is borned. For anyone who use this guide, please do feedback to help improving this documentation, thanks again!!!!

Reference

1. *InstantNSM - Sguil_on_RedHAT_install.pdf* by David J.Bianco who known as Hanashi.
2. *Sguil_install_v0.2.sh* by Richard Bejtlich who known as Helevius.