SANS | GIAC CERTIFICATIONS

WHITE PAPER

Using Snort For a Distributed Intrusion Detection System

Michael Brennan

Using Snort For a Distributed Intrusion Detection System
Version 1.3
Michael P. Brennan

Abstract

Intrusion detection has become an extremely important feature of the defense-in-depth strategy. The thought used to be that if you had a firewall protecting your network you were secure. This is no longer the case. A firewall is an essential and important part of network security but it does not have the ability to detect hostile intent. Unlike a firewall, an intrusion detection system has the ability to evaluate solitary packets and generate an alarm if it detects a packet with hostile potential. This document will provide an option for setting up a distributed network intrusion detection system using open source tools including the intrusion detection software Snort. Through the use of open source tools and spare hardware an intrusion detection system can be setup with minimal financial burden.

Why Snort?

At this point you may be wondering why Snort was used when there are so many quality commercial intrusion detection systems available today. Many companies that are looking for some type of intrusion detection simply cannot afford one of these commercial systems. Snort does not cost a thing but this by no means infers that it cannot perform as an intrusion detection device.

Snort is currently the most popular free network intrusion detection software. The advantages of Snort are numerous. According to the snort web site, "It can perform protocol analysis, content searching/matching, and can be used to detect a variety of attacks and probes, such as buffer overflow, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more" (Caswell). One of the advantages of Snort is its ease of configuration. Rules are very flexible, easily written, and easily inserted into the rule base. If a new exploit or attack is found a rule for the attack can be added to the rule base in a matter of seconds. Another advantage of snort is that it allows for raw packet data analysis. This allows for examination of a packet down to the payload to determine what caused the alert, why the something caused the alert, and whether action needs to be taken. Snort's flexibility, ease of configuration, and raw packet analysis make it a powerful intrusion detection device.

Hardware

Hardware requirements for this system are dependent upon the size of your network and volume of traffic. Three types of machines are discussed in this document, a remote sensor, a management server, and a monitoring machine.

The remote sensor discussed in this document was installed on SPARC Ultra 5 machines utilizing 128MB of RAM, 400MHz SPARC2I processor, and a 2GIG hard drive. This hardware far exceeds the requirements necessary to run snort on an average network. The NSS Group lists the absolute minimum requirements for a remote sensor as, "Pentium 133 with 32MB of RAM" (The NSS Group). You should not need any more than 2GIG of hard drive space on your remote sensor machines if you rotate your log files biweekly and archive them on another machine.

For the management server a SPARC Ultra 5 was used with the same basic specs as the remote sensor except another 128MB of memory was used as well as two 8GIG hard drives. One hard drive was used solely for the Snort database. This once again probably exceeds the

hardware requirements for most networks. The NSS Group lists the absolute minimum requirements for a management server as, "300MHz Celeron with 64 MB of RAM" (The NSS Group). Your hard drive requirements may be a slight bit higher if you are dealing with a large number of sensors and a lot of traffic. To ensure hard drive space the database information is archived regularly and stored on a different media.

The monitoring machines do not require any specific hardware, however they must be able to run a Web Browser and communicate through SSH.

Required Software

This system requires a number of different software packages. Every piece of software used in this system is open source software except for the operating system. The operating system used for this particular setup is Solaris 8. Any Unix variety of operating system will work for this type of setup with some minor tinkering depending on the OS. You could use Linux as your operating system as well. There is a document available at http://www.sans.org/newlook/resources/IDFAQ/open_source.htm, written by TJ Vanderpoel, which does a pretty good job of explaining a Linux setup of a similar system to the one discussed here. In fact the concept for the system developed here was derived from the system explained in TJ's paper. Solaris was chosen for this particular install because a number of SPARC machines were available and Solaris was the familiar operating system. You can use whatever operating system is most convenient for you. You will of course want to make sure that you fully secure your operating system. For Solaris, information on securing the operating system is provided at http://www.sans.org/newlook/resources/hard_solaris.htm. Be sure to apply any patches or upgrades that are currently available.

The versions of the software used in this example setup may be older versions than are currently available. These are the versions that worked properly at the time of the original install. If a newer version is available and you can get it to work properly then use it. The following are the software packages used for the system.

Snort version 1.8.3 is the intrusion detection software that is used for this system. You can download Snort at http://www.snort.org/downloads.html.

Libpcap version 0.6.2 is the next piece of software. Libpcap is the library that Snort depends on to capture packets from the network interface. It can be downloaded from http://www.tcpdump.org.

OpenSSH version 2.9p2 is required for this system. OpenSSH is the free version of the Secure Shell software. It provides a secure way to connect to the machines remotely for administrative purposes. OpenSSH can be downloaded from http://www.openssh.com.

MySQL version 3.23.41 is used for this system. MySQL is the most popular open source SQL database. The MySQL database is where all of the alert data from snort is stored. The remote machines use the MySQL libraries to communicate with the database located on the management server. You can find the software at http://www.mysql.org.

Apache Web Server version 1.3.20 is the next software package necessary for this setup. Apache is the most used web server on the Internet. It provides the secure web server that is used to display the analysis console. You can download the web server software from http://httpd.apache.org.

PHP version 4.0.6 is also required for this system. PHP is an open source scripting language similar to Perl. The PHP parsing engine is required to run the Analysis Console. PHP is available for download at http://www.php.net/downloads.php.

Mod_SSL version 2.8.4-1.3.20 is necessary for our use of Apache in this setup. Mod_SSL provides strong cryptography for the Apache 1.3 web server via the Secure Sockets Layer and Transport Layer Security protocols by the help of the Open Source SSL/TLS toolkit OpenSSL (Engelschall). You can find it at http://www.modssl.org.

OpenSSL version 0.9.6b is the next piece of software required for this system. OpenSSL is a library that provides cryptographic functionality to applications such as secure web servers. You can download it at http://www.openssl.org.

Stunnel version 3.20 is used for this setup. It is a program that allows you to encrypt arbitrary TCP connections inside SSL (Hatch). Stunnel works in conjunction with OpenSSL to send encrypted data to the management server. You can find Stunnel at http://www.stunnel.org/download.

A.C.I.D. (Analysis Console for Intrusion Detection) version 0.9.6b17 is the final piece of software for this system. A.C.I.D. was developed by Roman Danyliw at the CERT Coordination Center to be used as a part of the AIRCERT project. ACID has a query-builder and search interface to find alerts matching information such as a signature, and IP address, a certain port, a packet payload, or a certain sensor machine. It includes a packet viewer that graphically displays packet information for all logged alerts. This allows for packet analysis down to the packet payload to determine what the attacker was trying to accomplish and whether it was a real attack. A.C.I.D. allows for alert management through the grouping of alerts to create incidents, removing of false positives or handled alerts, and for archiving alerts for later use or database transfer (Danyliw). A.C.I.D. can be downloaded from http://www.cert.org/kb/acid/.

System Setup

The first step in setting up the intrusion detection system is to install and secure the operating system on the remote sensor and the management server. When installing on the management server be sure to create a large partition to store the snort database. A GNU make compiler and GCC compiler collection is necessary on the remote sensor and management server. They are both available at http://www.gnu.org/software/. For this system Solaris packages were obtained from http://www.sunfreeware.com to ease installation.

Management Server

The first machine that must be setup for this system is the management server. The management server requires eight software packages to be installed for proper function. The first piece of software to install is OpenSSL. The first step is to untar the source and change to the source files directory:

```
tar zxf openssl-0.9.6b.tar.gz
cd openssl-0.9.6b
```

The next step is to configure for compiling using the configure script:

```
./configure
```

Then you will compile and install the software with these two commands:

```
make
make install
```

If the compile and install run without any errors you can move on to the next application. Stunnel can be installed now that OpenSSL is set up. First you will need to untar the source and change to the source files directory:

```
tar zxf stunnel-3.20.tar.gz
```

cd stunnel-3.20

Now you just need to run the configure script without any options:

./configure

Finally complete the install by compiling and installing with make and make install just like you did above for OpenSSL. In order for Stunnel to work properly when we start the service we will need to create a service name in /etc/services:

echo "mysqls 3307/tcp" >> /etc/services

This completes the Stunnel setup so you can move on to installing OpenSSH. The first step is to untar the source and change to the source file directory:

tar zxf openssh-2.9p2.tar.gz
cd openssh-2.9p2

Next you need to run the configure script without any options:

./configure

Complete the install by compiling and installing using make and make install. Once OpenSSL, Stunnel, and OpenSSH are installed properly you can move on to installing MySQL. Before compiling MySQL you will need to untar the source file for Apache because MySQL needs the apache source libraries to configure properly for our purposes. The source files for mod_SSL and PHP will also be untared at this time because they need the apache source libraries to configure properly as well. The following commands will execute this:

tar zxf mysql-3.23.41.tar.gz
tar zxf php-4.0.6.tar.gz
tar zxf mod_ssl-2.8.4-1.3.20.tar.gz
tar zxf apache_1.3.20.tar.gz

After untaring the source files you need to change to the MySQL source directory and run the configure script with the following option:

cd mysql-3.23.41
./configure --with-apache=/path/to/apache/source/dir

If the configure runs properly you can complete the MySQL install by building and installing the software with make and make install. The next application you need to install is mod_SSL. Change to the mod_SSL source directory and then run the configure script with the following option:

cd ../mod_ssl-2.8.4-1.3.20
./configure --with-apache=/path/to/apache/source/dir

This completes the mod_SSL setup and you can now move on to PHP. To install PHP you need to change to the source files directory and run the configure script with the following options:

cd ../php-4.0.6
./configure --with-mysql=/path/to/mysql/dir --with-zlib --enable-track-vars --enable-bcmath --with-apache=/path/to/apache/source/dir

Complete the install by building and installing the software by running make and make install. Now you can install Apache Web Server. Change directory to the Apache source files directory:

cd ../apache_1.3.20

Now you need to run the configure script. This one is a little different because you need to define SSL_BASE which the Apache configure script uses to find the libraries of OpenSSL. You will need to change your shell to run this properly. The command should look like this:

sh

SSL_BASE=/path/to/OpenSSL/source/files ./configure --enable-module=ssl --enable-shared=ssl --prefix=/usr/local/apache --enable-rule=EAPI --activate-module=/src/modules/php4/lipphp4.a

The SSL_BASE and ./configure are part of one big command. Next you can compile the software by running make. Before you run make install you will need to run:

make certificate

This command generates the secure certificate used by the secure web server. Now complete the install by running make install. Now that Apache is installed you will need to start the MySQL database and create the users. The following commands will take care of this:

```
Shell> mysql --user=root mysql
mysql> GRANT ALL PRIVILEGES ON *.* TO snort@localhost
       IDENTIFIED BY 'some_pass' WITH GRANT OPTION;
mysql> GRANT ALL PRIVILEGES ON *.* TO snort@"%"
       IDENTIFIED BY 'some_pass' WITH GRANT OPTION;
```

Now that the database and users are created you will need to configure Apache. First of all you need to create a .htaccess file in /usr/local/apache/htdocs with the following:

```
AuthUserFile /usr/local/apache/.htpasswd
AuthGroupFile /dev/null
AuthName "IDS Login"
AuthType basic
require valid-user
```

The .htaccess file is the file that can modify server configurations. Adding a .htaccess with the above information will turn on password protection for the server. Then you have to create the password database (.htpasswd). This is a flat text file modified by the htpasswd command that comes with Apache. To create the file and add a new user to it, run the command:

/usr/local/apache/bin/htpasswd -c /usr/local/apache/.htpasswd username

You will be prompted to enter a password and then will be prompted to verify that password. You can add more users later by running /usr/local/apache/bin/htpasswd /usr/local/apache/.htpasswd username2. Be sure to place this file outside of the /usr/local/apache/htdocs directory so that it is not accessible via a web browser. This completes the Apache configurations. A.C.I.D. is the last software application that you need to install. First you will need to untar the source files:

tar zxf acid-0.9.6b

Next you need to create a directory for the acid files:

mkdir /usr/local/apache/htdocs/acid

Now change directory to the acid source files and copy the necessary files to the directory that you just created:

```
cd acid-0.9.6b
cp *.php /usr/local/apache/htdocs/acid
cp *.html /usr/local/apache/htdocs/acid
cp *.css /usr/local/apache/htdocs/acid
```

Then change the permissions on the directory and files:

```
chmod 755 /usr/local/apache/htdocs/acid
chmod 644 /usr/local/apache/htdocs/acid/*
```

This is all that you need to do to install the A.C.I.D. The final step in setting up the management server is to create the snort data directory and the databases. First you create the database:

mysql -p create snortdb

Now you need to move the data over to the partition that was created to store the snort database and create a symlink to that partition so that all data is sent there from now on:

mv /usr/local/mysql/data/snortdb /partition_for_the_db/snortdb
ln -s /db/snortdb /usr/local/mysql/data/snortdb

Now run your Create table commands

mysql -p snortdb < snort_mysql_tables
mysql -p snortdb < acid_mysql_tables

This should complete the setup of the management server. To test it you need to start Apache:

/usr/local/apache/bin/startssl

You will now be asked to insert the password that you used when you created the certificate. Then open a web browser on a remote console to https://management_server_IP/acid. You should be prompted for a username and password and when entered correctly you should see the Analysis for Intrusion Databases page. Now you can start the rest of your services:

/usr/local/sbin/sshd (starts the SSH daemon)
/usr/local/stunnel/sbin/stunnel -d mysqls -r 127.0.0.1:3306 -p \ /usr/local/etc/stunnel.pem -s stunnel -g stunnel&

The -d mysqls option activates Stunnel in daemon mode for the mysqls service we created earlier. The -r 127.0.0.1:3306 specifies the remote service where connections to the mysqls port will be forwarded. In this case the TCP port 3306 is used. The -p /usr/local/etc/stunnel.pem gives the location of the Stunnel private key. The -s stunnel option runs a setuid() to user stunnel and the -g stunnel option runs a setgid() to group stunnel (Chan, p.22). Be sure to include all of these commands in start up scripts so the services start if for some reason the machine is rebooted.

Remote Sensor

Now that the management server is setup and operational you can move on to the remote sensor. The remote sensor requires six software packages in order to function properly. To start the setup you will need to install OpenSSL. Install the software with the exact same procedure that was used for the management server. When you finish installing OpenSSL you will need to install OpenSSH with the same procedures as the management server as well. Now you can move on to install Stunnel. The compiling and installing of Stunnel can also be done using the same procedures that you used for the management server. However, you do not need to create a service name in /etc/services. The next step is to setup MySQL. For this install we only need to install the MySQL client. The process begins by untaring the source file and changing to the source directory:

tar zxf mysql-3.23.41
cd mysql-3.23.41

Next you will need to run the configure script and then run make and make install. The configure line that you will use looks like this:

./configure --without-server --prefix=/usr/local/mysql

If you do not get any errors during the compile this will complete the MySQL setup. The next step is to install the libpcap libraries so that Snort works properly. For this system a package from http://www.sunfreeware.com was used, however, the source file can be used by untaring the file and then compiling with configure, make, and make install. The next process in creating

the sensor is to install and configure Snort. The first step is to untar the source file and change to the source files directory:

```
tar zxf snort-1.8.3.tar.gz
cd snort-1.8.3
```

Now run the configure script with the following option:

```
./configure --with-mysql=/usr/local/mysql
```

Complete the install by running make and make install. If it compiled without any errors you can move on to the second step, which is to create the snort data directories and copy the snort configuration files to the configuration directory. First you will need to create a snort user and as that user create the data directories. These commands will cover it:

```
mkdir /export/home/snort/conf
mkdir /export/home/snort/logs
mkdir /export/home/snort/bin
```

When the directories are setup you will need to setup the permissions like so:

```
chmod –R 700 /export/home/snort/*
```

Now you need to change the directory to your snort source directory, for example:

```
cd /usr/src/snort-1.8.3/
```

Then copy the configuration files over to the snort configuration directory that you created above like this:

```
cp *.rules /export/home/snort/conf/
cp snort.conf /export/home/snort/conf/
cp classification.config /export/home/snort/conf/
```

The final step in creating the remote sensor is to configure snort for the network that this machine will be sniffing. To do this you will need to edit the snort.conf file. This configuration file contains all the information that Snort uses to properly monitor a network. Set var HOME_NET to the network that you wish to monitor. Set var EXTERNAL_NET to anything other than the HOME_NET, for example: !$HOME_NET. Set var DNS_SERVERS to any DNS servers that you are using. You will also want to set any preprocessors that you want to use. Next you will need to configure the output that Snort uses. This is the line that you need to use:

```
output database: alert, mysql, host=127.0.0.1 dbname=snortdb user=snort passw
ord=yourpassword sensor_name=Name_of_Sensor, encoding=ascii
```

The last part of the snort.conf file is the rule sets. Here you choose which rule sets you want to use. If do not want to use a particular rule set just comment it out. For a more in depth description of how to setup snort read the manuals provided with the software or check the snort web site (http://www.snort.org) for complete documentation. This completes the setup of the remote sensor. Now you can start your services:

```
/usr/local/bin/sshd
/usr/local/stunnel/sbin/stunnel -c -d 127.0.0.1:3306 -r \ Management_Server_IP:3307 -s
stunnel -g stunnel&
```

The -c option is used to configure Stunnel to act as a client.

```
/usr/local/snort/bin/snort -c /export/home/snort/conf/snort.conf –l /export/home/snort/logs
-d -i (interface) -u snort -g snort&
```

The -c /export/home/snort/conf/snort.conf option tells snort to use snort.conf as the configuration file. The -l /export/home/snort/logs option tells snort were to send its logs. The -d option dumps the application layer data. The -i (interface) option tells snort which interface it will use to monitor traffic. Finally the -u and -g options set the uid and gid for the user and group. Be sure

to include these commands in start up scripts so the services start if for some reason the machine is rebooted.

In order to ease the installation of future remote sensors, Solaris JumpStart was used. According to Sun Microsystems:

> Solaris JumpStart software is an automated system that can install and set up a Solaris system anywhere on your network without any user interaction… With Solaris JumpStart software, the Solaris Operating Environment and application software can be placed on centralized install servers and the install process can be customized…" (Sun Microsystems)

Packages were created for each of the pieces of software on the sensor and then installed with Jumpstart. This option is only available if you have identical hardware for each of your sensors. Another possible option for installing remote sensors could be by mirroring the hard drives from one sensor to another or using some form of an installation script. The key here is to find a way to quickly reproduce sensors without having to go through the complete installation process again.

Network Placement

The placement of the remote sensors is one of the most important parts of this system. If the remote sensors are misplaced packets may not be seen by the detection system. Andrew Baker explains it best when he remarks, "In order for Snort to be most effective, it needs to be positioned where it will see the most traffic possible" (Baker).

Before going into the network connections of the remote sensors we need to explain the settings of the interfaces. The "sniffer" interface or the interface Snort uses to monitor traffic must be set to promiscuous mode. This allows Snort to examine all traffic that passes by this interface while also ensuring that no connections can be made to the remote sensor from this interface. The other interface on the remote sensor will be used for administration purposes through the use of SSH. This administration interface should be connected behind your firewall preferably on its own private subnet. This is also the interface used to send information to the management server.

The first example of the remote sensor placement is if you have a high-speed connection to the Internet. You will want to monitor traffic coming from and going to that connection. The best way to achieve this would be to place a hub between the border router and your firewall. Connect your "sniffer" interface to that hub and you will be able to monitor all traffic going out of you network to the internet as well as the traffic coming into your network from the internet. Another example of remote sensor placement would be to monitor your DMZ network. The best setup for this type of network would be to place a hub between your firewall and DMZ machines then connect the "sniffer" interface to this hub. This allows you to monitor all traffic to and from your DMZ. The final configuration for remote sensor placement would be to monitor your internal or private network. Much like the DMZ network you will probably want to connect your "sniffer" interface to a hub between your firewall and the private network. One other thing to note about the "sniffer" interface is that if it is connected to a switch it must be connected to a spanning port. This means that it must be connected to a port that sends all of the traffic that goes through the switch through that port.

The placement of the management server is much easier. The management server should be placed on its own network segment behind your firewall. Only traffic from your remote sensors and your monitoring machines should be allowed to the server.

Monitoring

Now that the things are setup and running properly you can start to monitor your networks and make necessary configuration changes on your remote sensors. The alerts may seem a little overwhelming at first but once your sensors are properly configured things will make more sense. Soon you will be able to use the alerts from your intrusion detection system along with your firewall logs and system logs to find potential attacks before a real problem occurs. You will also want to make sure that you keep your machines updated including your Snort rule sets. New rule sets can be obtained from http://www.snort.org.

Conclusion

An intrusion detection system can be setup for little to no cost through the use of open source tools. This document provided an inexpensive solution for setting up a distributed network intrusion detection system with Snort and other open source tools. Hopefully you can use this setup to better protect your network and stop potential attacks before they cause any damage.

References

Caswell, Brian. "Snort - The Open Source Network IDS : More info about Snort" URL: http://www.snort.org/about.html

The NSS Group "Snort 1.8.1. Questionnaire" 25 November 2001 URL: http://www.nss.co.uk/ids/snort/snort_questionnaire.htm

Engelschall, Ralf S. "mod_ssl: The Apache Interface to OpenSSL" 16 October 2001. URL: http://www.modssl.org

Hatch, Brian. "Stunnel.org" 11 November 2001. URL: http://www.stunnel.org

Danyliw, Roman. "AIR-CERT - Analysis Console for Intrusion Databases (ACID)" 11 January 2002. URL: http://www.cert.org/kb/acid

Chan, Jason. "Distributed Intrusion Detection with Open Source Tools" SysAdmin Volume 10 Number 8 August 2001(2001): 20-25.

Sun Microsystems "Solaris[tm] Product Line: Datasheets" 18 October 2001. URL: http://www.sun.com/software/solaris/ds/ds-webstart/

Andrew R. Baker "Deploying Snort" 17 April 2000. URL: http://www.dpo.uab.edu/~andrewb/snort/deploying.html