Global Information Assurance Certification Paper

Richard L. Greene Jr.
Version 1.2e
Using Snort v1.8 with SnortSnarf on a RedHat Linux system
July 25, 2001

I. Introduction

To effectively implement system and network security, a multi pronged approach should be used. Proper security policies, firewalls, proxy servers, properly complex passwords and intrusion detection systems layered together help form one of the bedrock principles called "Defense in Depth"[1]. The purpose of defense in depth is preventing inherent and unknown flaws in the technologies deployed from allowing unauthorized access into a system or server. If one layer fails there is another to protect the failed layer. The intrusion detection system's (IDS) job is to log attempts of unauthorized network access into the systems.

There are two basic types of IDS systems host-Based (HIDS) and network-based (NIDS). A host-based system would be on each and every host to be monitored. With the cacophony of different OS types and application vendor restrictions this can be difficult to deploy. A network based IDS monitors the network traffic and is not directly impacted by which OS types are installed. The OS mix is only important in deciding which rule sets to deploy. The IDS system log files along with system log files go a long way to implement another principle "Prevention is Ideal but detection is a must"[2]. But what does detection do if the data is buried deep within the IDS log files? This analysis concentrates on several ways of getting the log file information from an open source IDS system called Snort. The tool that is explored for that purpose is SnortSnarf.

II. What are Snort and SnortSnarf and how do they work

Snort - According to its creator, Marty Roesch, is a lightweight intrusion detection system [3]. Snort is a network based IDS that monitors the network traffic. Snort is the data source for SnortSnarf. Network data packets that match predefined rules will be logged. The –s and -l switches control the log file used. A Snort run with the –s indicates that logging data goes to the SYSLOG. Otherwise logging data will go to the Alert file. The –l switch indicates to which directory the default Snort log files will go. For additional information on Snort switches run snort with -?. For further information on Snort go to http://www.snort.org and for information from a previous paper on deploying snort visit http://www.sans.org/infosecFAQ/intrusion/snorth.htm.

SnortSnarf – SnortSnarf is a Perl script that uses the snort log files and processes them into a web viewable format [4]. The SnortSnarf web console was designed to minimize attacks whose purpose is to attempt the control of the console screen real estate. This control could be gained by deliberately creating spurious packets. These packets could be used either to hide the real attack or to desensitize the security personal because of the quantity of bogus packets [5]. There are several additional plug in modules for use with Snort and SnortSnarf combination. These include: Spade, Nmap2HTML and SISR.

SPADE is linked into Snort and is a **S**tatistical **P**acket **A**nomaly **D**etection

Engine. It is a Snort preprocessor. It monitors for anomalous packets and then reports them through the normal Snort log[6]. Its purpose is to correlate port scans and report them. SPADE is a work in progress. The monitor exists but at this time the correlation function is not operational.

Nmap2HTML hooks to SnortSnarf and converts nmap flat files to HTML formatted files. It creates a Web page for each IP address detected [7]. Nmap is a network port mapping program. Nmap uses raw IP packets to determine what hosts and open ports are available.

SISR is a SnortSnarf Incident Storage and Reporting tool ist purpose is as an incident management tool. It allows sets of alerts to be grouped into incidents. This allows for the creation of custom reports and it will then autofill the reports and email them [8].

III. Platform Configuration

My Analysis is based on the setup and configuration of a PC clone running RedHat LINUX v7.0 and SNORT v1.8. Snort v1.8 requires libpcap.0.4 and libpq.so.2.0. See the link below to download the necessary files.

- RedHat Linux i386 v7.0 can be downloaded from
  http://www.redhat.com/download/howto_download.html#download
  ftp://ftp.redhat.com/pub/redhat/linux/7.0/en/iso/i386/

- libpcap for LINUX i386 can be downloaded from
  http://rpmfind.net/linux/RPM/redhat/7.0/i386/libpcap-0.4-29.i386.html
  ftp://rpmfind.net/linux/redhat/7.0/en/os/i386/RedHat/RPMS/libpcap-0.4-29.i386.rpm

- libpq.so for LINUX i386 can be downloaded from
  http://rpmfind.net/linux/RPM/redhat/7.0/i386/postgresql-7.0.2-17.i386.html
  ftp://rpmfind.net/linux/redhat/7.0/en/os/i386/RedHat/RPMS/postgresql-7.0.2-17.i386.rpm

- Snort v1.8 can be downloaded from
  http://www.snort.org/snort-files.htm
  http://www.snort.org/Files/Snort-1.8p1-0.src.rpm
  http://www.snort.org/Files/Snort-1.8p1-0.i386.rpm

  /usr/sbin/snort - Snort Executable
  /etc/snort - Directory for Snort rule and configuration files
  /var/log/snort - Directory for Snort log files

IV. SNORT – Installation

With Linux there are two ways to install Snort. One with all RPM files and the alternate way requires compiling the source code. The second way will allow incorporating additional components such as SPADE. Both ways require a server level Linux system set up with libpq.so and Libpcap installed.

Install libpq.so

#>rpm –U install postgresql-7.0.2-17.i386.rpm

Install Libpcap
#>rpm –U install libpcap-0.4-29.i386.rpm

Install Snort - RPM method
#>rpm –U install Snort-1.8p1-0.i386.rpm

or

Install Snort - Compile method and incorporating SPADE
Copy Snort and Spade to a temopary directory. For this procedure it is assumed the directory is /snort-tmp.
Spade's location http://www.silicondefense.com/software/spice/Spade-011701.1.tar.gz
Install Snort source files and unpack Spade
#>gzip –d Spade-011701.1.tar.gz
#>tar xvf Spade-011701.1.tar
#> rpm –U Snort-1.8p1-0.src.rpm
#> cd /usr/src/redhat/SOURCES
#>gzip –d snort-1.8p1.tar.gz
#>tar xvf snort-1.8p1.tar

Move Spade files to proper locations [9].
#> cd /snort-tmp/Spade-011701.1
#>cp spp_anomsensor.c /usr/src/redhat/SOURCES/snort
#>cp spp_anomsensor.h /usr/src/redhat/SOURCES/snort

Edit plugbase.h add the following line.
#include "spp_anomsensor.h"

Edit plugbase.c add the following line to InitPreprocessrs() function.
SetupSpade();

Edit Makefile.am and add the following to snort_SOURCES line.
spp_anomsensor.c
spp_anomsensor.h

Change into the Snort source directory and compile the Snort code.
#>./configure
#>make
#>make install

Setup a directory for Spade to log files to.
#> mkdir /etc/spade

Edit snort.conf to enable Spade. Add the following lines.

preprocessor spade: 10.5 /etc/spade/spade.rcv /etc/spade/log.txt 3 50000

See the Spade usage file for what the values do and set them up correctly for your environment. http://www.silicondefense.com/software/spice/spiceusage.htm
Snort is compiled and installed with the Spade add-in.

V. SnortSnarf – Installation

The installation is fairly simple. Retrieve the installation and rules files for SnortSnarf. SnortSnarf is found at http://www.silicondefense.com/software/snortsnarf/SnortSnarf-052301.1.tar.gz. First step in after server and Snort is to unpack SnortSnarf and move the files to proper locations. The information that follows demonstrates the steps to accomplish this.

Uncompress and untar SnortSnarf software.

```
#>uncompress SnortSnarf-052301.1.tar.gz
#>tar xvf SnortSnarf-052301.1.tar.gz
```

Make and install the perl time modules

```
#>cd SnortSnarf-052301.1/Time-modules
#>perl Makefile.pl
#>make
#>make test
#>make install
```

Copy files to proper locations

```
#>cp SnortSnarf-052301.1/include/SnortSnarf \
        /usr/lib/perl5/site_perl/5.6.0/SnortSnarf
#>cp SnortSnarf-052301.1/ snortsnarf.pl   /etc/SnortSnarf
```

VI. Running Snort and SnortSnarf

The first task after installation is start up and testing. The Snort command will start Snort. The Perl SnortSnarf command will pull in the data from Snort alert file and put it in HTML format.

```
#>snort –c /etc/snort/snort.conf –D
    -c specifies the configuration file to use
    -D specifies to run in daemon mode
#>perl /etc/SnortSnarf/snortsnarf.pl –d /var/www/html/snortsnarf \
        /var/log/snort/alert
    -d specifies where to put html output files
```

Open your web viewer to address http://snort-server/snortsnarf/index.html. This will display the main SnortSnarf page. This occurs providing that Snort created an alert file at /var/log/snort called alert and there is a web server running and the root is located at /var/www/html.

The above commands will start Snort and run SnortSnarf one time. For

SnortSnarf to be effective it will need to run periodically. The frequency to run SnortSnarf will be dependant on the amount of data in the Snort logs, the power of the system it runs on, how often the html files are to be updated and how this all impacts Snort. Ideally SnortSnarf would be run every couple of minutes. This will ensure timely display and notification. If SnortSnarf was run that often the server would slowly ground to a halt. A large Snort log can take upwards of 5 to 20 minutes to process depending on the CPU. Snort will automatically start with Linux, providing it is at run level one through five. The default configuration is wrong for use with SnortSnarf. Edit the file /etc/rc.d/init.d/snortd. Find the line

daemon /usr/sbin/snort –u snort –g snort –s –d –D

Delete the –s. The -s sends the snort output to SYSLOG. SnortSnarf needs the data from the Snort logs. To automatically start SnortSnarf the following italicized information will need to go into a crontab for root. This will all be on a single line. Put the information into a file called SnortSnarf first.

****/30 * * * * root perl /etc/SnortSnarf/snortsnarf.pl –d /var/www/html/snortsnarf -refresh=30 /var/log/snort/alert***

The following commands are one method to get the necessary commands into a crontab for root.

```
#>cd /etc/cron.d
#>cat > SnortSnarf
*/30 * * * *  root  perl /etc/SnortSnarf/snortsnarf.pl –d /var/www/html/snortsnarf  -refresh=30 /var/log/snort/alert
<Ctrl>d
#>crontab –u root SnortSnarf
```

This will run SnortSnarf every 30 minutes and will force the web browser to refresh every 30 minutes.

VII. Usage of Snort and SnortSnarf

The proper usage of Snort and SnortSnarf rely on Snort being properly configured. Snort has a collection of rule files bound together with a configuration file. The default file is called snort.conf. To eliminate unnecessary noise edit the file and set up correctly your HOME_NET, EXTERNAL_NET and DNS_SERVERS. There are also variables to set up the SMTP, HTTP and SQL servers found on your network. These servers can be very chatty and can cause false alarms. If Spade was incorporated into Snort it will need to be enabled. The location of the Snort IDS system is a major item for proper configuration of the system. If a network is small and flat the IDS system can be placed on a hub port. Otherwise it will need to be on a choke point or points. Once Snort has been configured and placed properly, the SnortSnarf console can be used.

The main page of SnortSnarf http://your snort server name/snortsnarf/index.html will show the total number of alerts, the date range of the alerts, the source of the alerts and a summary screen of the various alerts. Shown on this summary section are: signature name, total number of alerts, number of sources and number of destination and a summary link for that signature type. On the summary screen there are a couple link points for further information. If there is additional information on a given signature there will a link. There will be a link for summary information of the

captured signatures.

For example: MISC Large ICMP Packet has a link to the web site http://www.whitehats.com/IDS/246. This will provide some additional information on what was received. The site will provide additional information on the likelihood that a good source address was captured. There is information on the possibility of false positives. On the page will be further information on CVE, Bugtraq and advICE attack numbers. These will be possible live click points for additional information. The page will have tabs at the top for Event, Protocol, Research and Signatures. Event is the initial screen. Under protocol tab is the protocol map for the signature. Research tab section contains additional information on the use of the packet. The signature tab lists various types of IDS detection signatures that are for the capture of this attack signature.

The various attack numbers are from different independent groups. The CVE numbers are from http://cve.mitre.org . CVE stands for Common Vulnerabilities and Events. The purpose of the CVE number is to standardize names for security problems.[10] The Bugtraq list is similar to that of whitehats, but includes how the exploit was generated and any known fixes for the attack http://www.securityfocus.com/bid. The advICE list is created by Network Ice http://advice.networkice.com/Advice/Intrusions/default.htm. It lists a summary of the attack, details of what was being attempted, whether it easy to spoof and other names the attack goes by. At the bottom of the list is additional link point for further exploration. The information provided by these sites are to help understand the type of attack and how to defend against it in the future. This information will also help to understand if a packet was captured as a false alarm.

The summary link point will link to a page that summarizes the activity to the various source and destination addresses that received this particular signature. The addresses are link points that contain a summary of the various signatures for the given IP address, as well as detailed data for each time the signature was received. The detailed data will show what type of packet was received, when the packet was received, what the source address was, what the destination address was and other statistical data. The page will also have links to various DNS and Whois lookup points. These lookup points have limited usage on addresses from large companies. It is not uncommon for several companies to be listed as having overlapping address ranges.

One of Features of Snort and SnortSnarf is to bring data together. The format is such that potential problems can be easily analyzed and researched. This analysis will verify if there was an incident. The point of Snort and SnortSnarf combination is the detection of any incursion. The Snort alert logs and system log files will provide data of what was possibly compromised. When a security incident occurs the research link points will allow the analyst to start looking for ways to prevent further incursions. This further research and analysis SnortSnarf data will help provide enough information to make a recovery plan. The analysis should help identify where your defense in depth plan failed. With this knowledge of what failed, where it failed and how it failed, plans can be made on how to prevent unauthorized access in the future.

VII. Conclusion

Snort and SnortSnarf work very well with Linux as an effective IDS solution. SnortSnarf brings research tools together in such a way as to allow a IT security analyst to gather the information to make informed decisions on the nature of the attack. The tools available via the link points will allow the research of possible methods of preventing further attacks.

[1] Michael H. Warfield, Internet Security Systems Inc, Securing Linux / Unix Systems: slide 29
URL: http://www.wittsend.com/mhw/1999/securing_linux/txt029.html (25-July-2001)
[2] Eric Cole, LevelOne SANS Security Essentials course 25-May-2001
[3] What is Snort?
URL: http://www.snort.org/what_is_snort.htm (24-July-2001)
[4] SnortSnarf
URL: http://www.silicondefense.com/software/snortsnarf/index.htm(24-July-2001)
[5] James A Hoagland, Stuart Staniford, Silicon Defense, Viewing IDS alerts: Lessons from SnortSnarf
URL: http://www.silicondefense.com/pptntext/snortsnarf-discex2.pdf (20-July-2001)
[6] James A Hoagland, Stuart Staniford, Silicon Defense, SPADE
URL: http://www.silicondefense.com/software/spice/index.htm(24-July-2001)
[7] James A Hoagland, Stuart Staniford, Silicon Defense, README.nmap2html
URL: http://www.silicondefense.com/software/snortsnarf/readme.nmap2html.shtml(24-July-2001)
[8] James A Hoagland, Stuart Staniford, Silicon Defense, README.SISR
URL: http://www.silicondefense.com/software/snortsnarf/readme.sisr.shtml(24-July-2001)
[9] James A Hoagland, Stuart Staniford, Silicon Defense, Installation file for the Spade v011701.1
URL: http://www.silicondefense.com/software/spice/spiceinstallation.htm(24-July-2001)
[10] About CVE
URL: http://cve.mitre.org/about/(24-July-2001)

Last Updated: May 29th, 2017

Upcoming Training

Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event