



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials Bootcamp Style (Security 401)"
at <http://www.giac.org/registration/gsec>

Deploying Snort, a Lightweight Network Intrusion Detection System

David G. Sullivan

November 20, 2000

I. Introduction

Information security is best applied using the defense-in-depth approach. The defense-in-depth concept relies on the ability to protect, detect, and react. Only through the use of a diverse protection scheme can this ability be achieved [1]. A single security solution should not be relied upon, but rather a combination of perimeter, network and host-based protection and detection systems. This paper concentrates on only one part of the solution for the defense-in-depth concept; an inexpensive network intrusion system named Snort.

II. What is SNORT

According to Marty Roesch, the creator of the software, "Snort is a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks." [2]. Snort is basically a network sniffer, based upon the packet capturing (libpcap) library, that is able to monitor, log and/or alarm on network traffic that matches an easily defined rule-set [3]. Snort is termed a "lightweight" network intrusion detection system (NIDS) due to its small size and the fact that it combines the functionality of a sensor and reporting station that are used in most commercially available NIDS software.

III. Considerations in Deploying Snort

Careful consideration should be taken when deciding on a location to install Snort. As with any network sniffer device, Snort will only listen to what is on the network segment on which it is connected. If Snort is being installed in a switched environment, it must be connected to the spanning port. The traffic load must also be considered when running Snort in this manner, as the aggregate speed of the traffic off of the spanning port might cause Snort to drop packets [4].

IV. Obtaining, Compiling and Installing Snort

There are two software packages needed to compile and install Snort.

The source distribution of Snort: <http://www.snort.org/Files/snort-1.6.3.tar.gz>

The libpcap library: <ftp://ftp.ee.lbl.gov/libpcap.tar.Z>

First uncompress and untar the libpcap library.

```
#> uncompress libpcap.tar.Z
```

```
#> tar xvf libpcap.tar
```

Change directories into the libpcap directory. Run the **configure** script, and then the **make** command. Provided that the libpcap library compiles without errors, run the **make install** command, and then the **make install-incl** command to complete the installation [5].

```
#> ./configure
```

```
#> make
```

```
#> make install
```

```
#> make install-incl
```

Uncompress and untar the Snort source file.

```
#> gzip -d snort-1.6.3.tar.gz
```

```
#> tar xvf snort-1.6.3.tar
```

Change directories into the Snort directory. Run the **configure** script and then run the **make** command. If Snort

compiled without errors, run the **make install** command [6].

```
#> ./configure
#> make
#> make install
```

Snort has now been successfully compiled and installed.

V. Configuring and Running Snort

The configuration for Snort is relatively easy. First, create a directory for the Snort signature libraries, and copy the library files from the Snort source directory.

```
#> mkdir /usr/local/snort
#> cp *-lib /usr/local/snort/
```

Edit the **snort-lib** library and set the two variables, HOME_NET and DNS_SERVER, used by Snort. Change the HOME_NET variable to use the system's network range, and set the DNS_SERVER variable to the local DNS server of the system. When setting these variables, Snort requires the use of CIDR notation for the network range.

```
#> vi /usr/local/snort/snort-lib
HOME_NET=192.168.17.0/24
DNS_SERVER=192.168.17.5/32
```

Create the directory for the Snort logs, and set the permissions.

```
#> mkdir /var/log/snort
#> chmod 700 /var/log/snort
```

Use the following command to run Snort as a daemon in full alert mode with the snort-lib library.

```
#> snort -A full -D -c /usr/local/snort/snort-lib -d
```

Snort should now be up and running.

VI. Monitoring SNORT logs

Snort performs several different types of logging depending upon the alert and the way in which Snort is configured. By default, Snort logs any alerts to the **snort.alert** file in the **/var/log/snort** directory. The alerts in this file are comprised of the alert message, date and time stamp, source IP address and port, destination IP address and port, protocol, and IP flags. Below is an example of a logged traceroute.

```
[**] Traceroute [**]
11/21-12:17:00.209624 192.168.210.219:42342 -> 192.168.210.255:42342
UDP TTL:1 TOS:0x0 ID:10726 DF
Len: 84
```

Snort will also log more detailed information under a directory named with the source's IP address in a file name derived from the protocol used, source and destination ports. When run with the "-d" flag, as performed above, it will dump the application into this log. Below is an example of the detail log from the same traceroute.

```
#> cat UDP:42342-42342

[**] Traceroute [**]
11/21-12:17:00.209624 192.168.210.219:42342 -> 192.168.210.255:42342
UDP TTL:1 TOS:0x0 ID:10726 DF
Len: 84
00 00 00 4C 3A 00 00 00 00 00 0C 44 53 41 4D ...L:.....DSAM
```

```
65 73 73 61 67 65 00 00 00 00 00 0C 00 00 00 10 essage.....
4F 52 42 65 6C 69 6E 65 20 32 2E 30 00 00 00 00 ORBeline 2.0....
00 00 00 01 39 DE 2C 24 3A 00 00 00 00 00 00 04 ....9,$.:.....
3A 3A 00 64 00 00 00 04 00 00 00 00 ::d.....
```

Snort also has a portscan preprocessor that will log detected portscans to the **snort.alert** file. Below is an example of the messages in **snort.alert** related to a detected portscan.

```
[**] spp_portscan: PORTSCAN DETECTED from 192.168.210.200 (THRESHOLD 4 connections
exceeded in 0 seconds) [**]
```

```
11/22-10:17:11.935897
```

```
[**] spp_portscan: portscan status from 192.168.210.200: 25 connections across 1 hosts: TCP(25),
UDP(0) [**]
```

```
11/22-10:17:15.059373
```

```
[**] spp_portscan: End of portscan from 192.168.210.200: TOTAL time(0s) hosts(1) TCP(25) UDP(0) [**]
```

```
11/22-10:17:19.393349
```

The portscan preprocessor also creates a more detailed log, **/var/log/portscan.log**, which tells what ports were hit and what IP flags were used. Below is an example from a brief SYN portscan.

```
Nov 22 10:17:11 165.27.210.200:34017 -> 165.27.210.216:24 SYN **S*****
```

```
Nov 22 10:17:11 165.27.210.200:34018 -> 165.27.210.216:8 SYN **S*****
```

```
Nov 22 10:17:11 165.27.210.200:34019 -> 165.27.210.216:22 SYN **S*****
```

```
Nov 22 10:17:11 165.27.210.200:34020 -> 165.27.210.216:11 SYN **S*****
```

```
Nov 22 10:17:11 165.27.210.200:34021 -> 165.27.210.216:21 SYN **S*****
```

```
Nov 22 10:17:11 165.27.210.200:34022 -> 165.27.210.216:17 SYN **S*****
```

```
Nov 22 10:17:11 165.27.210.200:34023 -> 165.27.210.216:23 SYN **S*****
```

```
Nov 22 10:17:11 165.27.210.200:34024 -> 165.27.210.216:13 SYN **S*****
```

The logging subsystem of Snort is extremely powerful, and yet provides for relatively easy analysis of alerts generated. The Snort logs provide the user the necessary means to solve two of the aspects of defense-in-depth, to detect and react to incidents in real-time.

VII. Conclusion

A network intrusion detection system can be an integral part of the defense-in-depth strategy.

Network intrusion detection systems can give the user the ability to solve two of the critical aspects of the defense-in-depth strategy, to detect and react. Snort is a wonderful low to no cost solution for businesses looking to deploy a workable NIDS solution. Due to its small size and low overhead, Snort can be installed on existing systems in a network without the purchase of dedicated systems. Snort, written in C, can compile and run on a variety of different UNIX based operating systems with minimal configuration, and has even been ported to Windows 95/NT/2000. This makes Snort one of the easiest, most flexible and inexpensive NIDS to deploy.

References

[1] Clarence A. Robinson, Jr. Info Security 2000: Defense in Depth.
URL: <http://www.faircount.co.uk/web04/yic/info.html> (18 Nov. 2000).

[2] What is Snort?
URL: http://www.snort.org/what_is_snort.htm (20 Nov. 2000).

[3] Martin Roesch. Snort – Lightweight Intrusion Detection for Networks
URL: <http://www.clark.net/~roesch/lisapaper.txt> (18 Nov. 2000).

[4] Mark Cooper. An Overview of Intrusion Detection. Autumn 2000.
URL: http://www.xinetica.com/tech_explained/general/ids/wp_ids.html (18 Nov. 2000)

[5] Lawrence Berkeley nation Laboratory, Network Research Group. INSTALL. 20 Mar. 1998.
URL: <ftp://ftp.ee.lbl.gov/libpcap.tar.Z> (18 Nov. 2000).

[6] INSTALL. 17 Mar. 2000.
URL: <http://www.snort.org/Files/snort-1.6.3.tar.gz> (18 Nov. 2000)

© SANS Institute 2000 - 2005, Author retains full rights

Upcoming Training

Click Here to
{Get CERTIFIED!}



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
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event