

Snort Lab

Installation:

1. Download snort from <http://www.snort.org/dl/>
2. # `rpm -ivh snort-1.9.0-1snort.i386.rpm`

Usage:

There are three main modes in which Snort can be configured:

Sniffer, packet logger, and network intrusion detection system.

Sniffer mode simply reads the packets off of the network and displays them for you in a continuous stream on the console.

Packet logger mode logs the packets to the disk.

Network intrusion detection mode is the most complex and configurable configurations, allowing Snort to analyze network traffic for matches against a user defined rule set and perform several actions based upon what it sees.

Sniffer Mode

First, let's start with the basics. If you just want to print out the TCP/IP packet headers to the screen (i.e. sniffer mode), try this:

```
./snort -v
```

This command will run Snort and just show the IP and TCP/UDP/ICMP headers, nothing else. If you want to see the application data in transit, try the following:

```
./snort -vd
```

This instructs Snort to display the packet data as well as the headers. If you want an even more descriptive display, showing the data link layer headers do this:

```
./snort -vde
```

(As an aside, these switches may be divided up or smashed together in any combination. The last command could also be typed out as:

```
./snort -d -v -e
```

and it would do the same thing.)

Packet Logger Mode

OK, all of these commands are pretty cool, but if you want to record the packets to the disk, you need to specify a logging directory and Snort will automatically know to go into packet logger mode:

```
./snort -dev -l ./log
```

Of course, this assumes you have a directory named "log" in the current directory. If you don't, Snort will exit with an error message. When Snort runs in this mode, it collects every packet it sees and places it in a directory hierarchy based upon the IP address of one of the hosts in the datagram.

If you just specify a plain "-l" switch, you may notice that Snort sometimes uses the address of the remote computer as the directory in which it places packets, and sometimes it uses the local host address. In order to log relative to the home network, you need to tell Snort which network is the home network:

```
./snort -dev -l ./log -h 192.168.1.0/24
```

This rule tells Snort that you want to print out the data link and TCP/IP headers as well as application data into the directory ./log, and you want to log the packets relative to the 192.168.1.0 class C network. All incoming packets will be recorded into subdirectories of the log directory, with the directory names being based on the address of the remote (non-192.168.1) host. Note that if both hosts are on the home network, then they are recorded based upon the higher of the two's port numbers, or in the case of a tie, the source address.

If you're on a high speed network or you want to log the packets into a more compact form for later analysis you should consider logging in "binary mode". Binary mode logs the packets in "tcpdump format" to a single binary file in the logging directory:

```
./snort -l ./log -b
```

Note the command line changes here. We don't need to specify a home network any longer because binary mode logs everything into a single file, which eliminates the need to tell it how to format the output directory structure. Additionally, you don't need to run in verbose mode or specify the -d or -e switches because in binary mode the entire packet is logged, not just sections of it. All that is really required to place Snort into logger mode is the specification of a logging directory at the command line with the -l switch, the -b binary logging switch merely provides a modifier to tell it to log the packets in something other than the default output format of plain ASCII text.

Once the packets have been logged to the binary file, you can read the packets back out of the file with any sniffer that supports the tcpdump binary format such as tcpdump or Ethereal. Snort can also read the packets back by using the -r switch, which puts it into playback mode. Packets from any tcpdump-formatted file can be processed through Snort in any of its run modes. For example, if you wanted to run a binary log file through Snort in sniffer mode to dump the packets to the screen, you can try something like this:

```
./snort -dv -r packet.log
```

Network Intrusion Detection Mode

To enable network intrusion detection (NIDS) mode (so that you don't record every single packet sent down the wire), try this:

```
./snort -dev -l ./log -h 192.168.1.0/24 -c snort.conf
```

Where snort.conf is the name of your rules file. This will apply the rules set in the snort.conf file to each packet to decide if an action based upon the rule type in the file should be taken. If you don't specify an output directory for the program, it will default to /var/log/snort.

One thing to note about the last command line is that if Snort is going to be used in a long-term way as IDS, the "-v" switch should be left off the command line for the sake of speed. The screen is a slow place to write data to, and packets can be dropped while writing to the display. It's also not necessary to record the data link headers for most applications, so it's not necessary to specify the -e switch either.

```
./snort -d -h 192.168.1.0/24 -l ./log -c snort.conf
```

This will configure Snort to run in it's most basic NIDS form, logging packets that the rules tell it to in plain ASCII to a hierarchical directory structure (just like packet logger mode).

NIDS Mode Output Options

There are a number of ways to configure the output of Snort in NIDS mode. The default logging and alerting mechanisms are to log in decoded ASCII format and use "full" alerts. The full alert mechanism prints out the alert message in addition to the full packet headers. There are several other alert output modes available at the command line, as well as two logging facilities.

Alert modes are somewhat more complex. There are six alert modes available at the command line, full, fast, socket, syslog, smb (WinPopup), and none. Four of these modes are accessed with the -A command line switch. The four options are:

-A fast

fast alert mode, write the alert in a simple format with a timestamp, alert message, source and destination IPs/ports

-A full

this is also the default alert mode, so if you specify nothing this will automatically be used

-A unsock

send alerts to a UNIX socket that another program can listen on

-A none

turn off alerting

Packets can be logged to their default decoded ASCII format or to a binary log file via the -b command line switch. If you wish to disable packet logging all together, use the -N command line switch.

For output modes available through the configuration file, note that command line logging options override any output options specified in the configuration file. This allows debugging of configuration issues quickly via the command line.

To send alerts to syslog, use the "-s" switch. The default facilities for the syslog alerting mechanism are LOG_AUTHPRIV and LOG_ALERT. If you want to configure other facilities for syslog output, use the output plugin directives in the rules files.

Finally, there is the SMB alerting mechanism. This allows Snort to make calls to the smbclient that comes with Samba and send WinPopup alert messages to Windows machines. To use this alerting mode, you must configure Snort to use it at configure time with the -enable-smbalerts switch.

Here are some output configuration examples:

- Log to default (decoded ASCII) facility and send alerts to syslog

```
./snort -c snort.conf -l ./log -h 192.168.1.0/24
```

- Log to the default facility in /var/log/snort and send alerts to a fast alert file:

```
./snort -c snort.conf -s -h 192.168.1.0/24
```

- Log to a binary file and send alerts to Windows workstation:

```
./snort -c snort.conf -b -M WORKSTATIONS
```

High Performance Configuration

If you want Snort to go *fast* (like keep up with a 100 Mbps net fast) use the "-b" and "-A fast" or "-s" (syslog) options. This will log packets in tcpdump format and produce minimal alerts. For example:

```
./snort -b -A fast -c snort.conf
```

In this configuration, Snort has been able to log multiple simultaneous probes and attacks on a 100 Mbps LAN running at saturation level of approximately 80 Mbps. In this configuration, the logs are written in binary format to the snort.log tcpdump-formatted file. To read this file back and break out the data in the familiar Snort format, just rerun Snort on the data file with the "-r" option and the other options you would normally use. For example:

```
./snort -d -c snort.conf -l ./log -h 192.168.1.0/24 -r snort.log
```

Once this is done running, all of the data will be sitting in the log directory in its normal decoded format.

Changing Alert Order

Some people don't like the default way in which Snort applies its rules to packets. The Alert rules applied first, then the Pass rules, and finally the Log rules. This sequence is somewhat counterintuitive, but it's a more foolproof method than allowing the user to write a hundred alert rules and then disable them all with an errant pass rule.

For people who know what they're doing, the "-o" switch has been provided to change the default rule application behavior to Pass rules, then Alert, then Log:

```
./snort -d -h 192.168.1.0/24 -l ./log -c snort.conf -o
```

Miscellaneous

If you are willing to run snort in "daemon" mode, you can add -D switch to any combination above. Please NOTICE that if you want to be able to restart snort by sending SIGHUP signal to the daemon, you will need to use full path to snort binary, when you start it, i.g.:

```
/usr/local/bin/snort -d -h 192.168.1.0/24 -l \
/var/log/snortlogs -c /usr/local/etc/snort.conf -s -D
```

Relative paths are not supported due to security concerns.

If you're going to be posting packet logs to public mailing lists you might want to try out the -O switch. This switch "obfuscates" your the IP addresses in the packet printouts. This is handy if you don't want the people on the mailing list to know the IP addresses involved. You can also combine the -O switch with the -h switch to only obfuscate the IP addresses of hosts on the home network. This is useful if you don't care who sees the address of the attacking host. For example:

```
./snort -d -v -r snort.log -O -h 192.168.1.0/24
```

This will read the packets from a log file and dump the packets to the screen, obfuscating only the addresses from the 192.168.1.0/24 class C network.