Snort

What is Snort?

- Network intrusion detection system
- Created by Martin Roesch
- First developed in 1998 as a basic sniffer
- Small ~4.1M source download
- Fast and Portable
- Configurable – Open rules language
- Free – Open Source Software/GPL

Getting Started with Snort

- http://www.snort.org
  - Download in
    - Source tarball
    - RPM/DEB
    - Win32 Self extracting executable
    - Available for multiple platforms
  - Install from source.

Snort Rules

- Downloaded separately
- Four sets to choose from:
  - Subscription: Paid service offering Sourcefire rules
  - Registered: All rules including Sourcefire rules but 30 days behind
  - Unregistered: Rule set at time of point release
  - Bleedingsnort: Community-driven rules.
- Install rules by un-taring in the parent Snort Directory.

Operation Modes

- Snort can run in three different modes:
  - Sniffer
  - Packet Logger
  - Network Intrusion Detection System (NIDS)
- Runtime modes is determined by command line switches.

Sniffer Mode

- Sniff network packets and dump to screen
- Runtime switches to use:
  - **-v:** verbose mode
  - **-d:** dump packet payload
  - **-X:** dumps entire packet in hex
  - **-e:** display link layer

- Example:
  #> snort –dve

Sniffer Mode Output

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
02/13-09:43:30.354767 0:C:29:D9:79:E9 -> 0:50:56:E9:D2:6C type:0x800
len:0x45 172.16.210.129:32768 -> 172.16.210.2:53 UDP TTL:64
TOS:0x0 ID:19210 IpLen:20 DgmLen:55 DF Len: 27
D2 91 01 00 00 01 00 00 00 00 00 06 67 6F 6F ............goo
67 6C 65 02 63 61 00 00 01 00 01                    gle.ca.....
```

Output Explained....

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/13-09:43:30.354767 0:C:29:D9:79:E9 -> 0:50:56:E9:D2:6C type:0x800

len:0x45 172.16.210.129:32768 -> 172.16.210.2:53 UDP TTL:64

TOS:0x0 ID:19210 IpLen:20 DgmLen:55 DF Len: 27

D2 91 01 00 00 01 00 00 00 00 00 00 06 67 6F 6F .............goo

67 6C 65 02 63 61 00 00 01 00 01                    gle.ca.....
```

Logger Mode

- Log network packets to a log file
- Runtime switches to use:

  -K ascii –l <logdir>: log packet in ASCII to <logdir>

  -l <logdir>: Log packets in tcpdump format; default format

  Example:

  #> snort –K ascii –l /var/log/snort *#Logs in ASCII*

  #> snort –l /var/log/snort *#Logs in binary*

Logging to Directories

-K ascii –l /var/log/snort

Log directory (/var/log/snort) content

subdirectory: 172.16.210.129:
ICMP_ECHO UDP:32768-53
subdirectory: 64.233.187.104:
ICMP_ECHO_REPLY

NIDS Mode

- Network Intrusion Detection Mode
- Start Snort with your set of rules, setup packet analysis tools and let it do it's work
- Snort will analyze packets in real time, generate alerts and log offending packets.

NIDS Mode (Continued)

- Simply specify a configuration file

  -c <conf. File>

  Example:

  #> snort –c /etc/snort/snort.conf

- Test configuration and rules for syntax

  -T –c <conf. File>

  Example:

  #> snort –T –c </etc/snort/snort.conf>

  Snort will succeed or fail and exit.

NIDS Configuration

- Default configuration for snort
  - *Output directory:* ***/var/log/snort***
  - Alert mode: ***Full***
  - Logging mode: ***tcpdump binary format***
  - ***/etc/snort.conf*** if none specified

NIDS Command-line Options

- Alternate alert mode:
  - ***-A <mode>: full, fast, none, console and unsock***
  - **-s:** alert to syslog
  - **-v:** prints alerts and packet information
  - ***-M <workstation>:*** send SMB alerts
- Alternate logging directory:
  - ***-l <log directory>***
- Alternate logging mode:
  - ***-K:*** *ASCII log mode*
  - **-N:** No logs (disable logging)!!!!!

Examples of NIDS Mode

- Binary mode

  #> snort –c /etc/snort/snort.conf –l /var/log/snort –A fast

- Log in ASCII to a directory

  #> snort –c /etc/snort/snort.conf -K ascii –l /var/log/snort

- No logging!!!

  #> snort –c /etc/snort/snort.conf –s –N

  -N will turn off logging

  -s will log to syslog

Snort Configuration

snort.conf is the default configuration file

- All variables are set in configuration file to define network
- Configure pre-processors.
- Configure output plug-ins.
- Configure and tune rule sets.

Configuration: Variables

- Variables are extremely useful in configuration settings; set it once and use many times.
  - Increases modularity and portability of rules
  - variables are specified in config file or on command line using ***–S*** option.

  Examples:

  var HOME_NET 172.16.210.0/24
- Global variables will always be initialized to the IP address/mask of the specified interface

  Examples:

  var HOME_NET $eth1_ADDRESS

Configuration: Variables Cont.

- Variables definition have the following format:
  var <variable_name> <value>
- *Precede variable name with '$' before using*

  Examples:
  var HOME_NET 172.16.210.0/24

  alert tcp any any -> $HOME_NET 9999 (msg: "It works...."; sid:100001;)

- NOTE: snort will die at run time if a variable is called without being defined.

Configuration: Variables Cont.

- Variables can be overridden on command line by using ***–S*** option

Examples:

#> snort –c /etc/snort/snort.conf –S HOME_NET=172.16.210.0/24

#> snort –c /etc/snort/snort.conf –S HOME_NET=172.16.210.0/24 –S SMTP_SERVER=172.16.210.3

Configuration: Includes

- Includes are used to combine file content into current configuration file.
- Must use absolute path when including files unless include file resides in the same directory.
- Format of include statements

  include <filename>

Example:

include my-cool. rules

include /etc/snort/rules/more-cool.rules

Note: first rule resides in same directory as conf file

Configuration: Preprocessors

- Preprocessors are used to either examine or manipulate packets.
- Packets will be examined by many pre-defined preprocessors
- Format for specifying preprocessors:
  preprocessor <name_of_preproc>:
  <config_options>
- Preproc. arguments are unique to each one; read comments in snort.conf

Example:

preprocessor sfPortscan: proto { all }

Configuration: output plug-ins

- Output used to specify type of alerts and logging to be used.
- Output plug-ins can be stacked and can be overridden with command line options.
- Plug-ins are configured with directives; arguments are unique per plug-ins; check comments in conf. File

Example:

output log_tcpdump: tcpdump.log

output database: log, mysql, user=sqluser password=snort dbname=snortdb host=localhost

Configuration: All steps

1. Set the variables for your network
2. Configure dynamic loaded libraries
3. Configure pre-processors
4. Configure output plug-ins
5. Add any runtime configuration directives
6. Customize your rule set

Snort Rules

- Simple format with great flexibility
- Rules will define what will snort look for and who/what considered to be unusual activities
- Rules can be used to inspect packet header, packet payload or both.
- Rules have two parts:
  - Rule header (**required**)
  - Rule options
- Long rules running over multiple lines must have '\' continuation character.

Snort Rules: Header

alert tcp !172.16.210.0/24 any -> 172.16.210.0/24 any (flags: SF; msg: "SYN-FIN scan"; sid:100001;)

Rule Header
Rule Options

Rule Header details (who):

- Action (alert)
- Protocol (tcp)
- Source IP and port (!172.16.210.0/24, any)
- Dest. IP and port (172.16.210.0/24, any)
- Direction of traffic (->)

Rule Header: Action

alert tcp !172.16.210.0/24 any -> 172.16.210.0/24 any

- The action field instruct snort on what to do
- Values for action field:
  - ***alert*** (alert and log packet)
  - ***log*** (only log packet, no alerts)
  - ***pass*** (drop packet)
  - user defined actions

    Example of custom rule type:

```
ruletype myaction {
    type log
    output log_tcpdump:myrule.log
}
```

Rule Header: Src. IP & Port

alert tcp !172.16.210.0/24 any -> 172.16.210.0/24 any

- It defines where traffic is coming from in CIDR notation.
- '!' special keyword indicates negation.
- ***any*** defines the source port number and in this case non-specified (any port number will match)
- port can be specified as a number or a range.

CIDR (Classless Inter Domain Routing) format

172.0.0.0/8	Class A
172.16.0.0/16	Class B
172.16.210.0/24	Class C
172.16.210.129/32	Host address

Rule Header: Traffic direction

alert tcp !172.16.210.0/24 any -> 172.16.210.0/24 any

- Traffic direction defines packet matching direction.
- Only two direction types:
  - **->** (traffic flows from source to destination)
  - **<>** (traffic is bi-direction)

Rule Header: Dest. IP & Port

alert tcp !172.16.210.0/24 any -> 172.16.210.0/24 any

- It defines where traffic is going to in CIDR notation.
- Any ip on the 172.16.210.0/24 subnet will match.
- ***any*** defines the destination port number and in this case non-specified (any port number will match)
- port can be specified as a number or a range.

Snort Rules: Options

alert tcp !172.16.210.0/24 any -> 172.16.210.0/24 any (flags: SF; msg: "SYN-FIN scan"; sid:100001;)

Rule Header
Rule Options

Rule Options define (what to do):

- Packet attributes to inspect.
- Packet attributes must be separated by semi colon.
- Each attribute is composed of keyword ***(flags:)*** and a value ***(SF;)***
- Options must be enclosed in parenthesis.
- Ultimately forming a signature for a specific attack.

Snort Rules Options

- Rules Options are the driving force behind snort
- Options create a unique signature for almost any traffic of interest
- Rules options can be defined in many different groups
  - Event notification to provide info about the event that triggered ***(msg:;sid:;rev:;reference:;classification:;priority:;)***
  - IP header specific inspections ***(sameip:;ip_proto:;id:;tos:;...)***
  - ICMP header specific inspections ***(itype:;icode:;....)***
  - TCP header specific inspections ***(flags:;seq:;ack:;...)***
  - Content specific inspection (payload inspection using text or hex matching)

Snort References

- Snort - http://www.snort.org
- Snort Docs - http://www.snort.org/docs/
- SnortSnarf - http://www.snort.org/dl/contrib/data_analysis/snortsnarf
- BASE - http://base.secureideas.net/
- SnortCenter - http://users.telenet.be/larc/
- SnortSam - http://www.snortsam.net/
- Sguil - http://sguil.sourceforge.net/

Lab

This lab assumes that snort executable is in your path, if not make sure you run snort using the absolute path. Start the lab by booting into your CNSnort virtual machine and login as root/root (BAD!!!)

Please note that running as root is dangerous and should not be done in real/production environment.

Section 1:
Goal: Run snort to obtain version and usage information. Change directory to /root/snort_handson/lab1

#> snort -? Or snort -h

Find out the snort option to use to obtain the version number.

Version:

Lab Cont.

Section 2:
Goal: Check Snort configuration file for syntax errors. The command option to use can be found by asking snort for command line options.

Change directory to /**root/snort_handson/lab1** and use snort.conf to test.

What is the command line option to use?

Specify the number of rules that snort reported.

What is the log directory that snort will use?

Lab Cont.

Section 3:
Goal: Reading packets from a pcap capture file.
#> snort –vd –r snort.pcap

The ***–vd*** instruct snort to be verbose and to dump the application layer data ***–r*** tells snort to read from a capture file; snort.pcap.

Note: packet count can be restricted by using ***-n*** option.

Run snort to read from snort.pcap, display verbose application layer and show one packet. Indicate the following:

Destination IP/port numbers:
Number of TCP options:
TCP flags:
TTL:

Lab Cont.

Section 4:

Goal: Logging output to directory instead of screen. First you have to create the log directory (logdir) then execute snort to read from snort.pcap file and log in ASCII to the log directory.

Note: snort logs in binary by default, use ***–K ascii*** to log in ASCII

List the source ip that generated "EVIL Traffic" packet content?

What is the IP ID of the same traffic:

Lab Cont.

Section 5:

Goal: Snort can be instructed to look for one protocol using tcpdump filters. A tcpdump filter can be given at the end of the command line. Run snort to read from snort.pcap file looking for all **UDP** traffic only and then view the results using snort.

Note: snort should be run in quit mode, use ***–q*** option.

What is the name and type of log file snort generated?

After analysing the newly generated log file using snort, how many UDP packets were found?

How many UDP packets were destined to port 80?

Lab Cont.

Section 6:

Goal: Snort in Network Intrusion Detection System mode using a simplified snort.conf file that contains one rule. In your working directory you will find a file called ***snort.conf*** with one rule that alert on traffic generated from ***any*** destined to ***any*** looking for content ***"Hello"***

Note: Create a new directory to log all NIDS traffic then run:
#>snort –c snort.conf –dq –l <logdir> –r snort.pcap

What files were created by snort?

Look inside the alert file and indicate the message associated with this alert.