

Hands-On Workshop

Snort Intrusion Detection System

Using Snort as IDS and Compliance Tool

Rafeeq Rehman

@rafeeq_rehman

rafeeqrehman.com



Module 1 - Introduction

Module Outline

- Software List for the course
- Pre-Requisites
- Reference Material
- Intrusion Detection and Intrusion Prevention Systems
- Types of Intrusion Detection Systems (IDS)
- Rule and Anomaly based Intrusion Detection Methods
- Inline and Passive Mode IDS
- Placement of IDS in Network

Software/Manual for This Workshop

- Following is a high level software list for this workshop:
 - Linux Operating System
 - Snort
 - Snort Rules
 - Snort DAQ Library
 - Compilers and different utilities
 - Libraries for linking with snort binaries
- Snort Manual

Pre-Requisites

- Familiarity with Linux Operating Systems is necessary. You don't need to be expert in Linux but need to know simple commands to operate Linux.
- Linux file system structure
- A laptop/desktop with Virtual Box installed to run Linux. You will receive a virtual machine
- Access to the Internet

Reference Material

- Snort – www.snort.org
- Snort Rules
 - Sourcefire – <http://www.snort.org/snort-rules/>
 - Emerging Threats – <http://rules.emergingthreats.net>
- OSSEC – <http://www.ossec.net>
- OSSEC Reference Manual – <http://www.ossec.net/doc/>
- Splunk – www.splunk.com
- Splunk Base – <http://splunk-base.splunk.com/apps/>
- Splunk OSSEC –
<http://splunk-base.splunk.com/apps/22285/splunk-for-ossec-splunk-v4-version>
- Splunk Snort –
<http://splunk-base.splunk.com/apps/22369/splunk-for-snort-splunk-4x>

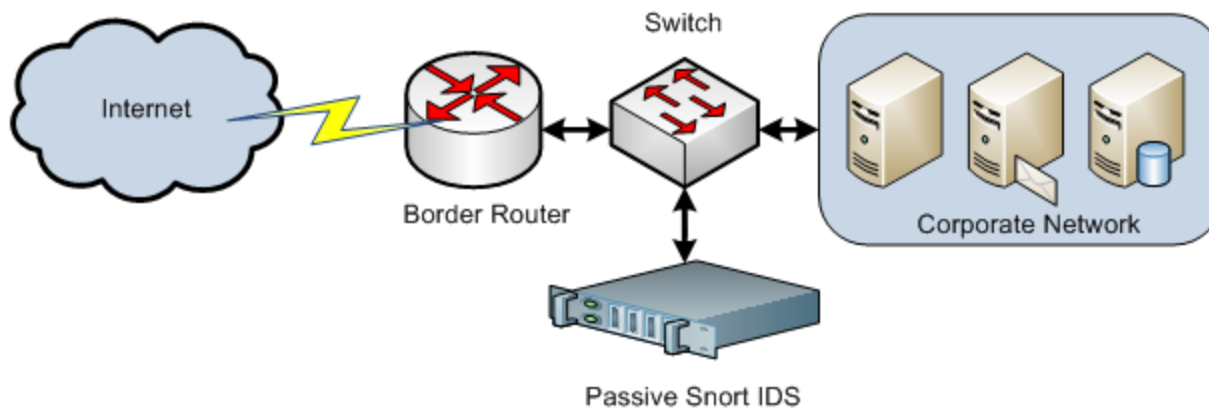
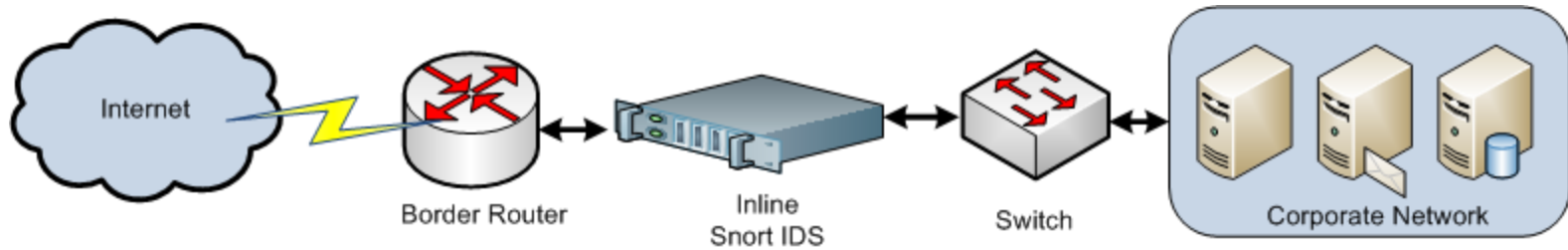
Some Initial Thoughts

- Snort is a tool, not a solution to a problem. It is up to a user how to use this tool.
- Don't look for something specific, just learn how to use the tool. If you are told to look for specific, you don't see anything else!
- Ask questions: How do you know that?
- You are not going to be expert after this training. The objective is to get you started on a journey.
- **ASK QUESTIONS!** If I don't know the answer, I will find it for you. It will help me get better.

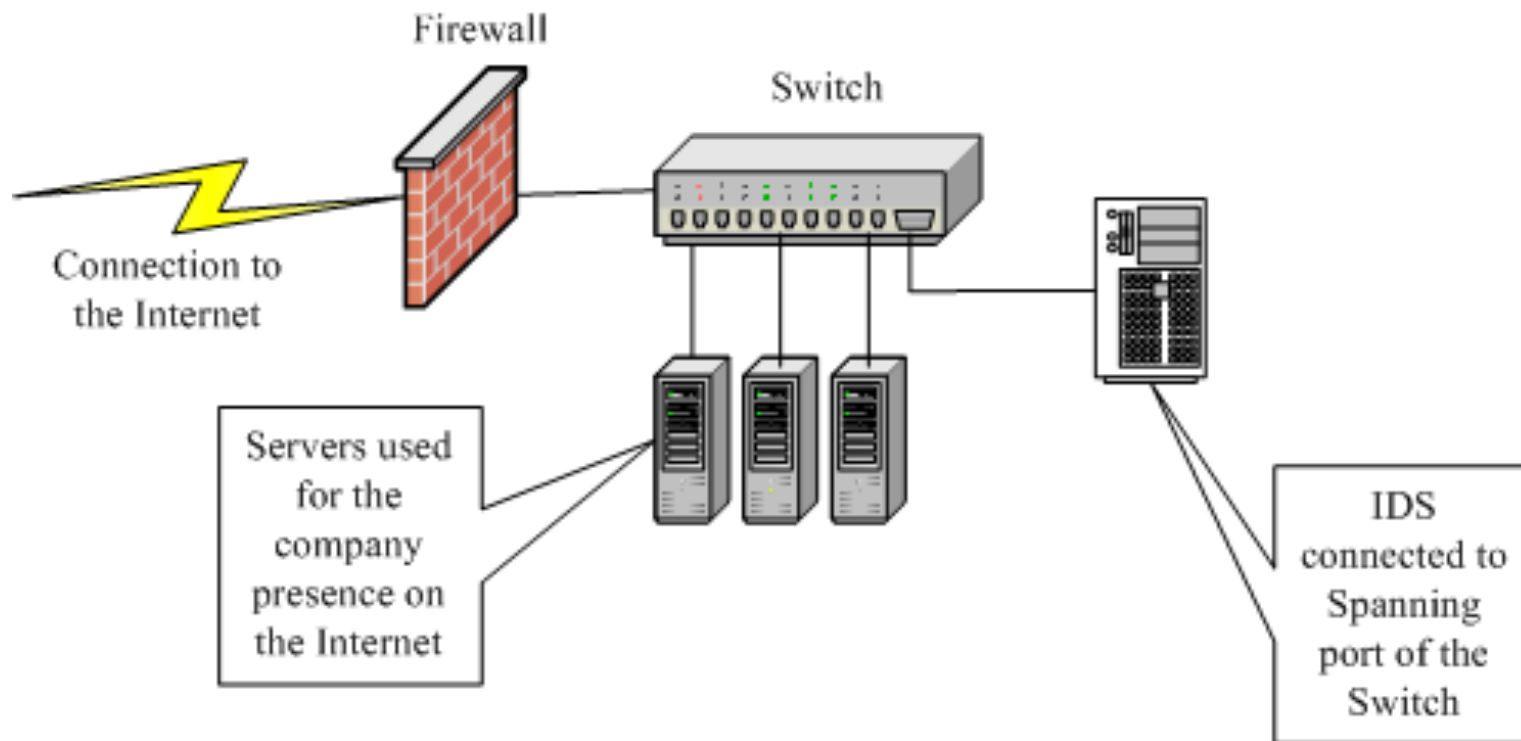
What is an IDS and IPS?

- Intrusion Detection System will create alerts
- Intrusion Prevention System will stop activity based upon some criteria
- Network Intrusion Detection System - Checks for specific activities or anomalies at the network layer and
- Types of IDS
 - Host Based (HIDS)
 - Network Based (NIDS)
- Method of Detection
 - Rule Based
 - Anomaly Based
 - Hybrid
- Snort is NIDS but can be used “inline” to work as IPS

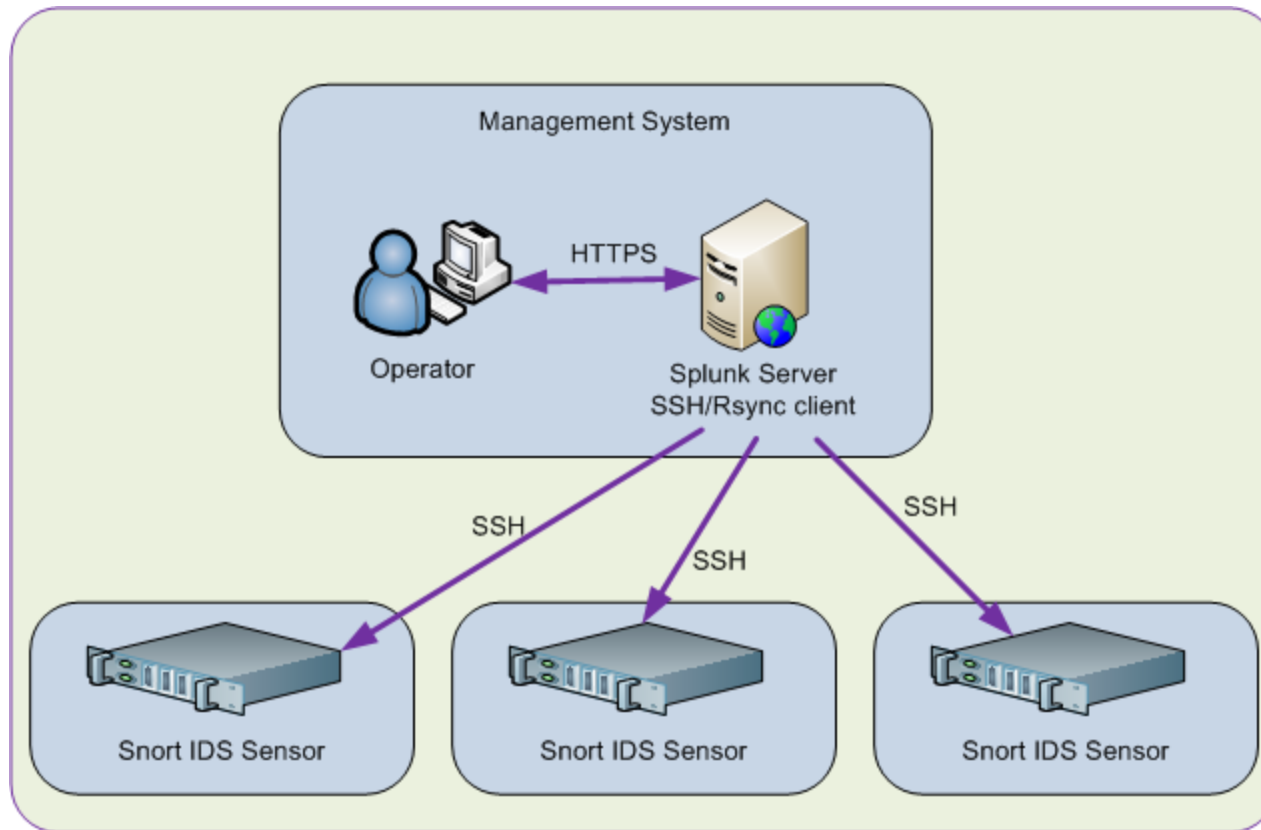
Inline and Passive IDS/IPS



Placement of NIDS



Enterprise IDS with Multiple Sensors

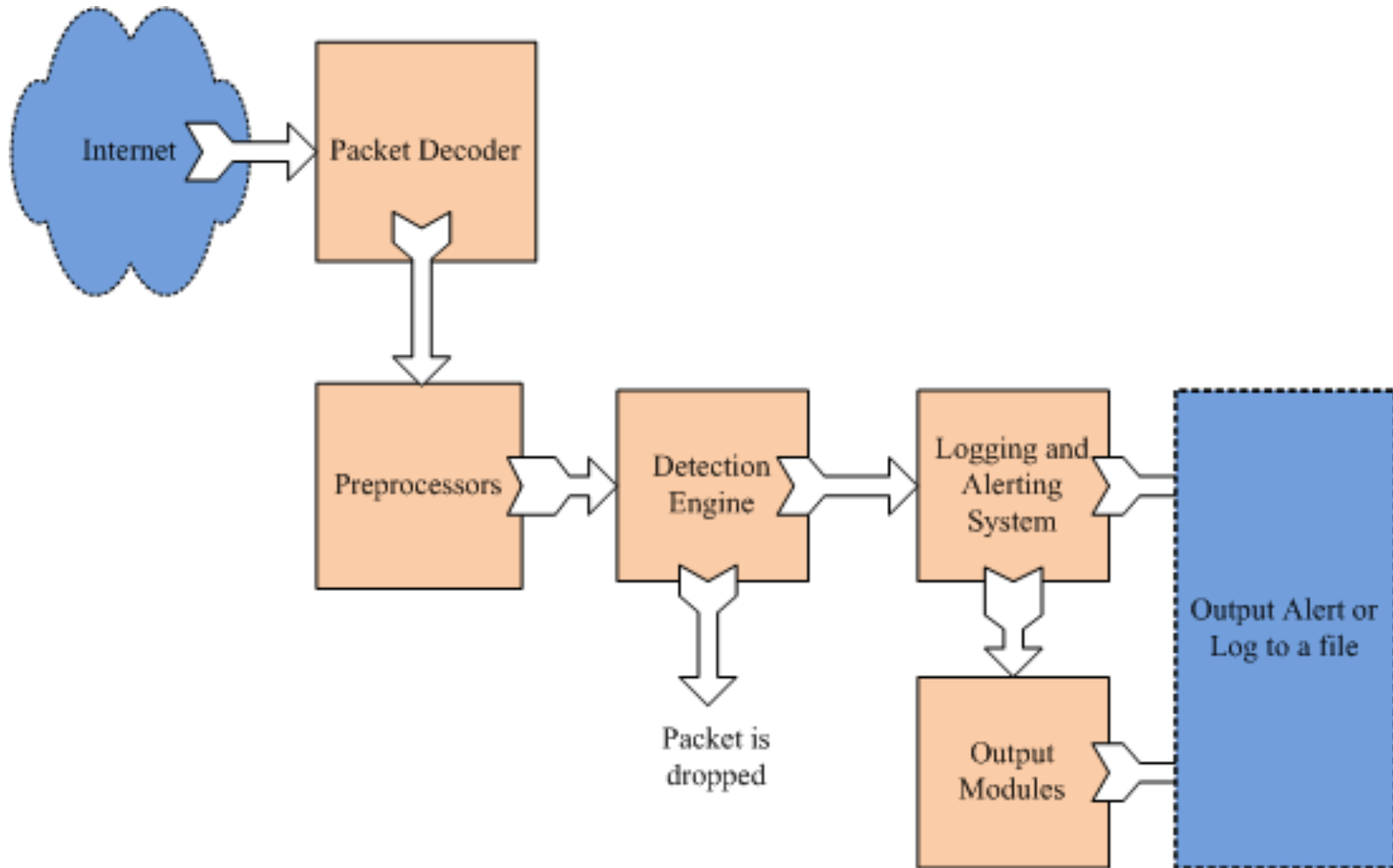


Host Based IDS

- Detect host based attacks
 - System file changes
 - Registry changes
 - Log file parsing
 - File Integrity Checks
- Alerting
- Rules
- Centralized Management

- Snort is not a host-based IDS. It is better if you use something like OSSEC or AIDE

High Level Snort Architecture





Introduction to Snort

Tasks for Getting Snort Up

1. Installation
 - Source Code
 - RPM
 - Yum (or other tools)
2. Installing Snort rules
3. Initial Configuration
 - Mainly configuring snort.conf file
4. Managing alerts

Snort Preprocessors

- Preprocessors handle data before it is handed over to detection engine and after packet decoding.
- Major preprocessors include:
 - Frag 3 - IP defragmentation
 - Streams 5 - TCP stream reassembly
 - sfPortscan - Detect reconnaissance
 - RPC Decode
 - Performance Monitor
 - HTTP Inspect - Find and normalize fields
 - SMTP - Find SMTP commands and responses
 - FTP and Telnet Preprocessors - FTP/Telnet commands and responses
 - SSH - Detects SSH protocol exploits
 - DNS - Detects DNS exploits by looking and DNS queries
 - ARP Spoof detection
- You can write your own preprocessors

Configuring Preprocessors

- Preprocessors are configured through Snort configuration file `snort.conf`

*preprocessor sensitive_data: alert_threshold 25 \
mask_output \
ssn_file ssn_groups_Jan10.csv*

- Preprocessors can take files as input. An example is reputation preprocessor that can read black list IP address file.

Logging and Alerting

- Difference of Log and Alert Files/Destinations
- Writing rules for logging and alerting
- Logging and Alerting
 - You can only log, alert, or both
- Logging and Alerting Mechanisms
 - Storing Snort data in files using Full and Fast alerting
 - Syslog
 - Unix Socket
 - Database
 - CSV
 - TCP dump logging
- You can create your own output modules

Configuring Output Modules

- Output modules are configured through snort.conf file
- Setting log limits for files

```
output alert_syslog: host=192.168.2.10:514, <facility> <priority>  
<options>
```

```
output alert_syslog: host=192.168.2.10:514, log_auth log_alert  
log_ndelay
```

Output Modules

- Syslog
- Alert and Log to file
- CSV
- Database
- Tcpdump

Snort Directory Structure

- The bin directory
- The lib directory
- The etc directory
- Rules directories

Sensor Directory Structure

- All directories are under /opt/snort directory

```
.  
|-- admin  
|-- bin  
|-- etc  
|-- lib  
|-- preproc_rules  
|-- rules  
|-- share  
|-- so_rules  
`-- src
```

Logs and Alerts directory

- Logs and alerts go under logs directory but can be configured to any place

```
/opt/snort
|-- etc
|-- logs
|   `-- 192.168.144.154
|       `-- snort
|-- preproc_rules
|-- rules
|-- scripts
|-- so_rules
`-- temp
```

- Each sensor has a directory under /opt/snort/logs directory

Summary

Exercise



Snort Installation

Module Outline

- The installation can be done in two basic ways:
 - Using Snort rpm which are pre-built packages
 - From the source code
- RPM method is easy
- Advanced users should use source to compile the options they need.
- Compiling from source code has a number of benefits:
 - You can include the options that you want
 - You can choose the location of installation files
 - The pre-installed Snort versions usually lack many features
- Installing from source code needs more knowledge of Linux
- We will cover both methods.

Steps for Installing Snort from Source Code

- Download Snort and DAQ Libraries
- Unpack and Install DAQ
 - Use tar command to unpack
 - Use configure to prepare for compilation
 - Use make; make install to compile and install DAQ
- Install Dependencies (libdnet, pcre, etc)
- Unpack and Install Snort
 - Use tar command to unpack
 - Use configure command to prepare for installation (discussed on next slide)
 - Use make; make install to install Snort.
- Install Snort rules files
- Edit snort.conf file

Compiling From Source Code

- Download source code file *snort-2.9.4.tar.gz*
- Download *daq-2.0.0.tar.gz*
- Install DAQ library (`tar zxvf daq-2.0.0.tar.gz`, `configure`, `make`, `make install`)
- Unpack source code `tar -zxvf snort-2.9.4.tar.gz` which will create directory *snort-2.9.4*
- Go to directory using command “`cd snort-2.9.4`”
- Run configure command: “`./configure --prefix=/opt/snort --enable-normalizer --enable-reload --enable-dynamicplugin --enable-zlib --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable-profile`”
- Most probably this command will fail due to dependencies.

Compiling From Source Code (Continued)

- You will need to install at least the following libraries and header files:
 - libpcap (pcap-devel)
 - pcre (pcre-devel)
 - libnet
 - libdnet (<http://code.google.com/p/libdnet/>)
- You may also need to have following tools if you have not already installed. The configure command will show you what you need to install
 - bison
 - flex or lex
- Continue running configure command until you succeed. Each time you will install any missing software needed for compilation.

Compiling From Source Code (Continued)

- After *configure* script succeeds, run “*make*” command.
- Once “*make*” is complete, run “*make install*” command
- The above step will create */opt/snort* directory (from *--prefix* command line parameter of *configure* script).
- Download snort rules file from snort.org after registering. This file is *snortrules-snapshot-2941.tar.gz*
- Run “*tar zxvf snortrules-snapshot-2941.tar.gz*” command that will extract rules by creating multiple directories. Copy these directories in */opt/snort* directory.
- Edit */opt/snort/etc/snort.conf* file to configure location of rule files and other directories (set location of different files and directories)
- Create log directory using command “*mkdir /opt/snort/logs*”
- Run “*/opt/snort/bin/snort -c /opt/snort/etc/snort.conf -l /opt/snort/logs*”
- If everything goes well, you will be running a working snort now.

A Word About Snort Rules

- Snort Rules filenames contain Snort version.
- Download the file that is relevant to your Snort installation.
- You have to register to download rules.

Registered User Release

The Registered User Release makes Sourcefire VRT Certified Rules updates available to registered users of Snort.org free of charge 30-days after the initial release to subscribers.

Documentation

[Rule Documentation \(opensource.gz\)](#)

MD5 - 05 Dec, 2012

Snort v2.9

[snortrules-snapshot-2940.tar.gz](#)

MD5 - 20 Dec, 2012

[snortrules-snapshot-2923.tar.gz](#)

MD5 - 20 Dec, 2012

[snortrules-snapshot-2930.tar.gz](#)

MD5 - 20 Dec, 2012

[snortrules-snapshot-2931.tar.gz](#)

MD5 - 20 Dec, 2012

Starting and Stopping Snort

- Testing Snort “`/opt/snort/bin/snort -dev -i eth0`”
- The above command will run snort in packet dump mode.
- Run Snort using command “`/opt/snort/bin/snort -c /opt/snort/etc/snort.conf -l /opt/snort/logs`”
 - The `-c` flag is used to specify configuration file
 - The `-l` flag is used to specify location of log files
 - If location of log files are not specified, the logs go to `/var/log/snort` directory by default. You need to create this directory manually.
- If you want Snort to listen on a specific adapter, you need to use `-i` command line switch. An example would be `-i eth1` (or `-i p2p1`)

Gotchas

- Log Directory
 - Snort log directory may not exist and you may have to create it.
 - Default log directory is `/var/log/snort`
 - You can use a different log directory using command line switch or `snort.conf` file.
- Ethernet Device Name
 - The Ethernet device name may differ on different Linux versions/distributions.
 - To check device name, run `ifconfig` command.
 - On CentOS, RedHat, Old Fedora, the device name may be `eth0` or `eth1`. On new Fedora distributions, it may be `p2p1` etc.
- Initially disable reputation preprocessor in `snort.conf` file to make Snort work.

Automatic Start and Stop

- Create a script `/etc/init.d/snort`
- Set up run levels and order of start/stop
- Use `chkconfig --add snort` to create links in sequencer directories
- Use `/etc/init.d/snort start` to start Snort
- Use `/etc/init.d/snort stop` to stop Snort
- Use “`ps -ef | grep snort`” to command to check if Snort is running.
- Use “`tail -f /opt/snort/logs/alerts`” to see new alerts in real time

IP Reputation – Black and White Lists

- Reputation Plugin
- Download files from emerging threats:
- <http://rules.emergingthreats.net/open-nogpl/snort-2.9.0/rules/rbn-ips.txt>
- <http://rules.emergingthreats.net/open-nogpl/snort-2.9.0/rules/rbn-malvertisers-ips.txt>
- <http://rules.emergingthreats.net/open-nogpl/snort-2.9.0/rules/compromised-ips.txt>
- <http://rules.emergingthreats.net/open-nogpl/snort-2.9.0/rules/emerging-compromised.rules>

Summary

- Installing Snort using:
 - RPM
 - From Source Code
- Copying Rules Files
- Editing snort.conf file
- Snort startup and Shutdown
- Automatic startup/shutdown scripts

Exercise

- Build Snort and Run it.





Installing Snort Rules

Summary

- Downloading Sourcefire Snort Rules and Installation
- Editing snort.conf to select different types of rules
- Creating local rules

Snort Directory Structure

- The snort directory structure is as follows:

```
.  
├── bin  
├── etc  
├── include  
├── lib  
├── logs  
├── preproc_rules  
├── rules  
├── share  
├── so_rules  
└── src
```

- Rules are divided into rule files which are present under “rules” directory.
- Rule files are included in snort.conf file.

Configuring Snort and Installing Rules

- Get latest rules from either snort.org (needs registration) or Emerging threats web site
- Emerging Threats Snort Rules
<http://rules.emergingthreats.net/open-nogpl/snort-2.9.0/>
<http://rules.emergingthreats.net/open/snort-2.9.0/emerging.rules.tar.gz>
- Create/Edit main configuration file snort.conf to include rules files.
- Create automated startup/shutdown scripts
- Start Snort and test creation of alerts (usually a simple ping will generate some alerts)

Download Rules

- Download rules from snort.org. The filename should match your current Snort version. For example for version 2.9.4.1, the rules file will contain a number 2941 in the name.

- Unpack rules files

```
tar ztvf snortrules-snapshot-2941.tar.gz
```

- This will create multiple directories, copy these directories to the `/opt/snort` directory (or untar file inside `/opt/snort` directory)

- Directories are `etc`, `preproc_rules`, `so_rules`, `rules`

Snort Rule Files

- There are many rules file under rules directory.
- You can place rule files anywhere on file system, just need to configure snort.conf properly.

```
[root@localhost rules]# ls
app-detect.rules          file-executable.rules
attack-responses.rules   file-flash.rules
backdoor.rules           file-identify.rules
bad-traffic.rules        file-image.rules
blacklist.rules          file-multimedia.rules
botnet-cnc.rules         file-office.rules
browser-chrome.rules     file-other.rules
browser-firefox.rules    file-pdf.rules
browser-ie.rules         finger.rules
browser-other.rules      ftp.rules
browser-webkit.rules     icmp-info.rules
```

Configure snort.conf Enable Rules

- Rules can be directly placed inside snort.conf file.
- All rules files are “included” through snort.conf file (better way to organize rules).
- Using rule files, specific rule types can be enabled/ disabled
- Set Variables (Absolute or relative paths)
 - var RULE_PATH ../rules
- Include Rule files
 - include \$RULE_PATH/local.rules
 - include \$RULE_PATH/app-detect.rules
 - include \$RULE_PATH/attack-responses.rules
 - include \$RULE_PATH/backdoor.rules
 - include \$RULE_PATH/bad-traffic.rules
 - include \$RULE_PATH/blacklist.rules
 - include \$RULE_PATH/botnet-cnc.rules

Adding Emerging Threats Rules

- Emerging Threats is a public repositories of Snort rules.
- Some rules are similar to Sourcefire but there are many additional rules.
- Download Emerging Threats Snort Rules
 - <http://rules.emergingthreats.net/open-nogpl/snort-2.9.0/>
 - <http://rules.emergingthreats.net/open/snort-2.9.0/emerging.rules.tar.gz>
- Basic principles are the same:
 - Download tar files
 - Unpack and move rule files to Snort rules directories (Can be anywhere as long as snort.conf is configured properly)
 - Edit snort.conf file to include these files

Snort Rule Anatomy

- Snort rules consist of two major parts:
 - Rule Header
 - Rule Options

- A sample rule will be as follows:

action protocol src_addr src_port direction dst_addr dst_port Options

- A real rule looks like this:

alert tcp any any -> any 21 (msg: "FTP Traffic");

- The red part is *header* and the green part is *options*

Testing Snort Rules

- The local.rules is a special file to put your own rules in it.
- Create a simple rule that sends alerts for each ICMP packet.

```
alert icmp any any -> any any (msg: "test rule"; sid:10001;)
```

- Restart Snort
- Use ping command to ping any host. You should start seeing alerts in /opt/snort/logs/alert file.

Starting Snort

- Packet Sniffing Mode
 - `bin/snort -vd -i eth0`
- Packet Logging Mode
 - `bin/snort -l /var/log/snort -i eth0`
- Starting Snort in IDS Mode
 - Use `-c <config file location>` on the command line to start Snort in IDS mode.
 - Use `-D` to start as background process.

Exercise

- Build Snort and Run it.



Snort Configuration File

Summary

- The snort.conf File (Can be any name and any location)
 - Variables
 - Decoders
 - Detection Engine Parameters
 - Dynamic Libraries and Preprocessors
 - Preprocessors and Output Plugins
 - Include Files
- Sample snort.conf file
- The classification.conf File
- The reference.conf File
- The threshold.conf file
- Snort Rule Files
- Reloading Snort Configuration

Snort Directory Structure

- The snort directory structure is as follows:

```
.  
├── bin  
├── etc  
├── include  
├── lib  
├── logs  
├── preproc_rules  
├── rules  
├── share  
├── so_rules  
└── src
```

- Rules are divided into rule files which are present under “rules” directory.
- Rule files are included in snort.conf file.

The snort.conf File

- The main configuration file – snort.conf
- It can be placed anywhere and referenced on the command line when starting Snort.
- The typical location of snort.conf file is in /opt/snort/etc directory.
- A number of other configuration files are used by “including” them in snort.conf file.
- The snort.conf file has nine sections.

Sections in snort.conf

1. Set the network variables.
2. Configure the decoder
3. Configure the base detection engine
4. Configure dynamic loaded libraries
5. Configure preprocessors
6. Configure output plugins
7. Customize your rule set
8. Customize preprocessor and decoder rule set
9. Customize shared object rule set

part 1: Variables – ipvar and portvar

- The ipvar Variables
 - ipvar HOME_NET any
 - ipvar EXTERNAL_NET any
 - ipvar DNS_SERVERS \$HOME_NET
 - ipvar SMTP_SERVERS \$HOME_NET
- The portvar Variables
 - portvar HTTP_PORTS [80,81,311,591,593,901,1220,1414,1741,1830,2301,2381,2809,3128,3702,4343,4848,5250,7001,7145,7510,7777,7779,8000,8008,8014,8028,8080,8088,8090,8118,8123,8180,8181,8243,8280,8800,8888,8899,9000,9080,9090,9091,9443,9999,11371,55555]
 - portvar SHELLCODE_PORTS !80
 - portvar ORACLE_PORTS 1024:
 - portvar SSH_PORTS 22
 - portvar FTP_PORTS [21,2100,3535]
- Regular Variables - var RULE_PATH ../rules

Part 2: Configure Decoders

- # Stop generic decode events:
`config disable_decode_alerts`
- # Stop Alerts on experimental TCP options
`config disable_tcpopt_experimental_alerts`
- # Stop Alerts on T/TCP alerts
 - `config disable_tcpopt_ttcp_alerts`
- # Stop Alerts on all other TCPOption type events:
`config disable_tcpopt_alerts`
- # Stop Alerts on invalid ip options
`config disable_ipopt_alerts`

Part 3: Configure Detection Engine

```
config pcre_match_limit: 3500
```

```
config pcre_match_limit_recursion: 1500
```

- # Configure the detection engine See the Snort Manual, [Configuring Snort - Includes - Config](#)

```
config detection: search-method ac-split search-optimize max-pattern-len 20
```

- # Configure the event queue. For more information, see [README.event_queue](#)

```
config event_queue: max_queue 8 log 3 order_events content_length
```

Part 7: Rules Files

- include \$RULE_PATH/local.rules
- include \$RULE_PATH/app-detect.rules
- include \$RULE_PATH/attack-responses.rules
- include \$RULE_PATH/backdoor.rules
- include \$RULE_PATH/bad-traffic.rules
- include \$RULE_PATH/blacklist.rules
- include \$RULE_PATH/botnet-cnc.rules

The classification.config file

- Used to set priority of alerts. Rules can override default priority

```
alert TCP any any -> any 25 (msg:"SMTP expn root"; flags:A+; \  
  content:"expn root"; nocase; classtype:attempted-recon;)
```

- This is included inside snort.conf file.
- Format: *config classification:shortname,short description,priority*
- config classification: not-suspicious,Not Suspicious Traffic,3
- config classification: unknown,Unknown Traffic,3
- config classification: bad-unknown,Potentially Bad Traffic, 2
- config classification: attempted-recon,Attempted Information Leak,2
- config classification: successful-recon-limited,Information Leak,2

The reference.config File

- Included in snort.conf file
- Provides reference URLs for rules.

config reference: bugtraq <http://www.securityfocus.com/bid/>

config reference: cve <http://cve.mitre.org/cgi-bin/cvename.cgi?name=>

config reference: arachNIDS <http://www.whitehats.com/info/IDS>

config reference: osvdb <http://osvdb.org/show/osvdb/>

Snort rule Files

- Usually placed under rules directory but can be placed anywhere.
- Similar rules are grouped in the same file. Example:
 - DDoS rules are grouped in the one file.
 - DNS attack detection rules are grouped in a different file.
- Comments – All lines starting with `#` sign are comments
- Snort my reload rule files after making any change.
 - Restart Snort
 - Send HUP signal if snort is build with `--enable-reload` option.

Testing Snort Config File

- Test snort.conf file by using `-T` option

```
/opt/snort/bin/snort -T -c /opt/snort/etc/snort.conf
```

- Good option to verify where Snort may be failing during startup.



Snort Preprocessors

Module Outline

- frag3
- stream5
- http_inspect
- ftp_telnet
- smtp
- ssh
- dns
- ssl
- Reputation

Preprocessor – sensitive_data

- Step 1: Configure Preprocessor

```
preprocessor sensitive_data: alert_threshold 1
```

```
preprocessor sensitive_data: alert_threshold 25 mask_output
```

```
ssn_file ssn_groups_Jan10.csv
```

- Step 2: Create Rules

```
alert tcp any any -> any $SMTP_PORTS (msg:"Credit Card numbers sent over email"; gid:138; sid:10001; rev:1; sd_pattern:1,credit_card;)
```

```
alert tcp any any -> any any (msg:"Email address detected"; gid:138; sid:10002; rev:1; sd_pattern:1,email;)
```

- Step 3: Test (login to Wifi Router with email address)

Sensitive Date Alert

- Alert from Rule

*[**] [138:10002:1] "Email address detected" [**]*

[Priority: 0]

11/03-12:31:08.567117 192.168.97.1:80 -> 10.0.2.15:35674

TCP TTL:64 TOS:0x0 ID:23510 IpLen:20 DgmLen:236

****AP*** Seq: 0x299D5202 Ack: 0x76A20896 Win: 0xFFFF TcpLen: 20*

- Alert from Preprocessor

*[**] [139:1:1] (spp_sdf) SDF Combination Alert [**]*

[Classification: Sensitive Data was Transmitted Across the Network] [Priority:

2] 11/03-15:09:01.465595 192.168.97.1 -> 10.0.2.15PROTO:254

TTL:64 TOS:0x0 ID:8320 IpLen:20 DgmLen:52

Preprocessor – reputation

Step 1: Enable Preprocessor

```
preprocessor reputation: \  
blacklist /etc/snort/default.blacklist, \  
whitelist /etc/snort/default.whitelist
```

Step 2: Create blacklist and whitelist files

In file "default.blacklist"

```
# These two entries will match all ipv4 addresses 1.0.0.0/1  
128.0.0.0/1
```

In file "default.whitelist" `68.177.102.22 # sourcefire.com`
`74.125.93.104 # google.com`

Step 3: Enable preprocessor rules in snort.conf

Preprocessor – http_inspect

- Step 1: Enable preprocessor (enabled by default)
- Step 2: Create/Enables Rules
- Step 3: Test (Use simple test like with wifi router)
 - Use wget for Wifi router
 - Use web browser for Wifi router
- Warning – Can be noisy

Validating snort.conf File

- Why Validate?

- Test before loading

- Command Line to Validate

/opt/snort/bin/snort -T -c /opt/snort/etc/snort.conf

Preprocessor - sfPortscan

- Step 1: Configure/Enable

```
preprocessor sfportscan: proto { all } scan_type { all } sense_level { high }
```

- Step 2: Test with nmap (port scan wifi router)

```
[**] [122:5:1] (portscan) TCP Filtered Portscan [**][Classification:  
Attempted Information Leak] [Priority: 2] 11/03-12:46:49.312320  
10.0.2.15 -> 192.168.97.1PROTO:255 TTL:42 TOS:0x0 ID:9128 IpLen:  
20 DgmLen:158
```




Output Modules

Introduction

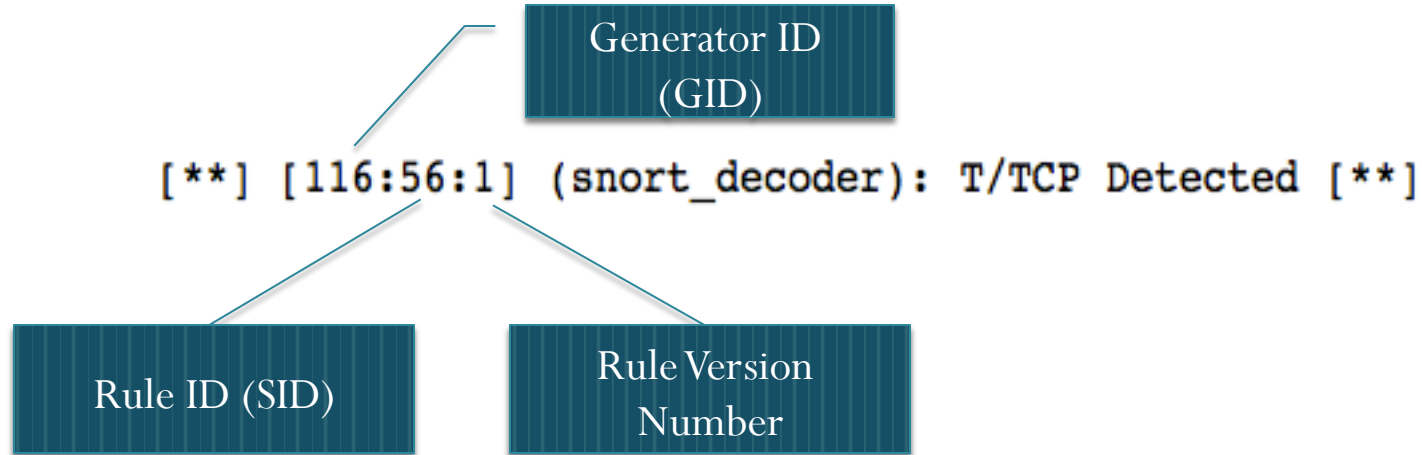
- Output modules provide interface to alerts and logs.
- Multiple destinations can be used for output:
 - Syslog
 - Local files
 - Unified
- Output modules are configured in `snort.conf` file.
- Same alerts can be sent to multiple locations (e.g. local files and syslog).
- Additional software packages can be used as user interface (MySQL, Base, Splunk, etc).
- Syslog may be easiest method for integration in existing centralized logging system.

Alerts and Logs to Files

- alert_full
- alert_fast
- alert_syslog
- alert_database
- Limiting size of log files

File based output

- Standard output files are created under `/var/log/snort`.
- You can change location by using `-l` command line option.
- A typical alert looks like the following:



- Timestamp, src/dst IP addresses, Message, Additional info

Output - Syslog

- Configuring in snort.conf
 - *output alert_syslog: LOG_AUTH LOG_ALERT*
- Alerts in /var/log/messages
 - *Nov 3 17:03:15 localhost snort[6570]: [138:10002:1] "Email address detected" {TCP} 216.92.2.158:80 -> 192.168.97.105:45321*
 - *Nov 3 17:03:15 localhost snort[6570]: [120:3:1] (http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE [Classification: Unknown Traffic] [Priority: 3] {TCP} 216.92.2.158:80 -> 192.168.97.105:45321*
 - *Nov 3 17:03:33 localhost snort[6570]: [139:1:1] (spp_sdf) SDF Combination Alert [Classification: Sensitive Data was Transmitted Across the Network] [Priority: 2] {PROTO:254} 216.92.2.158 -> 192.168.97.105*

MySQL + Barnyard

- Multiple sensors, centralized database
- Install Barnyard2
 - Untar barnyard2 and run autogen.sh
 - Run ./configure
 - Run make
 - Run make install
 - Create/Edit config file
 - Create Database Schema
 - Run barnyard2
- Run Apache and Base for user interface

Rsync + SSH + Splunk

- Use in a multi-sensor environment.
- Configure SSH with key based authentication.
- Write script to synchronize all sensors to a centralized server for analysis.
- Run Splunk with Snort application for centralized visualization.

Splunk Installation

- Download from splunk.com and install using rpm
`rpm -i --prefix=/opt <splunk rpm file>`
- Add splunk user and groups
`groupadd splunk`
`useradd -g splunk splunk`
- Create startup scripts (you will need to accept license)
`/opt/splunk/bin/splunk enable boot-start -user splunk`
- Change owner and group permissions of /opt/splunk
`chown -R splunk.splunk /opt/splunk`
- Disable SELinux
- Start splunk for the first time
`/etc/init.d/splunk start --accept-license`

Splunk Installation

1. Download from splunk.com and install using rpm
`rpm -i --prefix=/opt <splunk rpm file>`
2. Create startup scripts (you will need to accept license)
`/opt/splunk/bin/splunk enable boot-start`
3. Disable SELinux (Edit `/etc/sysconfig/selinux`)
4. Start splunk for the first time
`/etc/init.d/splunk start --accept-license`
5. Start splunk afterwards
`/etc/init.d/splunk start`

Installing Splunk

RPM Installation

```
[root@localhost opt]# rpm -i --prefix=/opt /home/rafeeq/Downloads/splunk-4.1.6-89596
splunk-4.1.6-89596.i386.rpm                splunk-4.1.6-89596-Linux-x86_64.tgz
[root@localhost opt]# rpm -i --prefix=/opt /home/rafeeq/Downloads/splunk-4.1.6-89596.i386.rpm
warning: /home/rafeeq/Downloads/splunk-4.1.6-89596.i386.rpm: Header V3 DSA/SHA1
Signature, key ID 653fb112: NOKEY
-----
Splunk has been installed in:
    /opt/splunk

To start Splunk, run the command:
    /opt/splunk/bin/splunk start

To use the Splunk Web interface, point your browser at:
    http://localhost.localdomain:8000

Complete documentation is at http://www.splunk.com/r/docs
-----
[root@localhost opt]# █
```

Installing Splunk

- Creating Init Script

```
/opt/splunk/etc/auth/distServerKeys/private.pem
/opt/splunk/etc/auth/distServerKeys/trusted.pem
['openssl', 'genrsa', '-out', '/opt/splunk/etc/auth/distServerKeys/private.pem', '1024']
/opt/splunk/etc/auth/distServerKeys/private.pem generated.
/opt/splunk/etc/auth/distServerKeys/public.pem generated.
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
writing RSA key

This appears to be your first time running this version of Splunk.
Moving '/opt/splunk/share/splunk/search_mrsparkle/modules.new' to '/opt/splunk/share/splunk/search_mrsparkle/modules'
'.
  Creating: /opt/splunk/var/lib
  Creating: /opt/splunk/var/run/splunk
  Creating: /opt/splunk/var/run/splunk/upload
  Creating: /opt/splunk/var/spool/splunk
  Creating: /opt/splunk/var/spool/dirmoncache
  Creating: /opt/splunk/var/lib/splunk/authDb
  Creating: /opt/splunk/var/lib/splunk/hashDb
  Checking databases...
  Validated databases: _audit, _blocksignature, _internal, _thefishbucket, history, main, sample, summary

Init script installed at /etc/init.d/splunk.
Init script is configured to run at boot.
[root@localhost opt]# █
```

Installing Splunk. Starting it First Time

```
[root@localhost opt]# splunk/bin/splunk start --accept-license

Splunk> Australian for grep.

Checking prerequisites...
  Checking http port [8000]: open
  Checking mgmt port [8089]: open
  Checking configuration... Done.
  Checking index directory... Done.
  Checking databases...
  Validated databases: _audit, _blocksignature, _internal, _thefishbucket, history, main, sample, summary
  Checking for SELinux.
All preliminary checks passed.

                                     [ OK ]
Starting splunk server daemon (splunkd)... Done.Starting splunkweb... /opt/splunk/share/splunk/certs does not exist.
Will create
Generating certs for splunkweb server
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'privkeySecure.pem'
-----
Signature ok
subject=/CN=localhost.localdomain/O=SplunkUser
Getting CA Private Key
writing RSA key

                                     [ OK ]

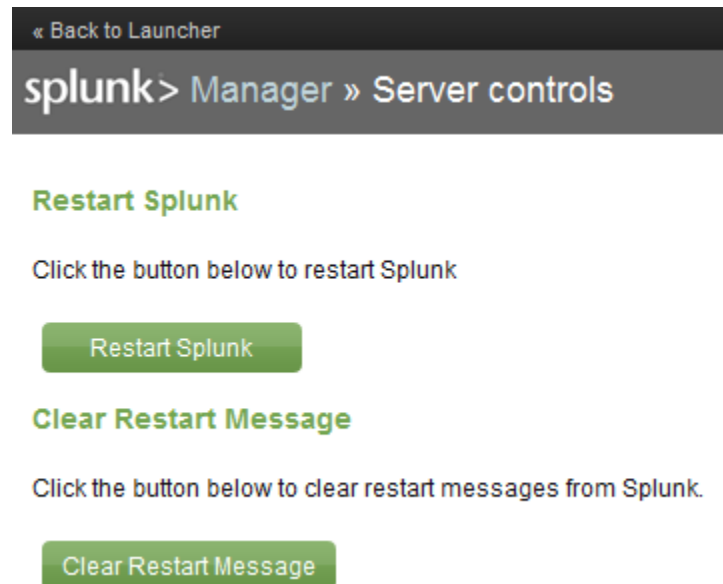
Done.
If you get stuck, we're here to help.
Look for answers here: http://www.splunk.com/base/Documentation

The Splunk web interface is at http://192.168.144.135:8000

[root@localhost opt]# █
```

Installing Snort Application

- Go to /opt/splunk/etc/apps folder
- Run tar zxvf <Snort App File Name>
- Restart Splunk by going to Manager->Server controls



Add Snort Log Files to Splunk

- Add a new data input file

« Back to Launcher

splunk> Manager » Data inputs

Data Inputs

Set up data inputs from files and directories, network ports, and scripted inputs. If you want to set up forwarding and receiving between two Splunk instances, go to [Forwarding and receiving](#).

Type	Inputs	Actions
Files & Directories <i>Upload a file, index a local file, or monitor an entire directory.</i>	5	Add new
TCP <i>Listen on a TCP port for incoming data, e.g. syslog.</i>	0	Add new
UDP <i>Listen on a UDP port for incoming data, e.g. syslog.</i>	0	Add new
Scripts <i>Run custom scripts to collect or generate more data.</i>	0	Add new

Add Snort Logs

[« Back to Launcher](#)

splunk > Manager » Data inputs » Files & Directories » /opt/snort/logs

Host

Set host field for all events from this source.

Set host

constant value

Specify method for getting host field for events coming from this source.

Host field value

localhost.localdomain

Source type

Set sourcetype field for all events from this source.

Set sourcetype

Manual

When this is set to automatic, Splunk classifies and assigns the sourcetype automatically, and gives unknown sourcetypes placeholder names.

Source type (optional)

snort

Index

Set the destination index for this source.

Index

default

Advanced options

Whitelist (optional)

Specify a regex that files from this source must match to be monitored by Splunk.

Blacklist (optional)

Specify a regex that files from this source must NOT match to be monitored by Splunk.

Cancel

Save

Data Input File

- You should see something like this after adding Snort Log Directory

« Back to Launcher Logged in as admin | Jobs | Logout

splunk> Manager » Data inputs » Files & Directories Help for this page

Search:

Data inputs (files)

Showing 1-5 of 5 items Results per page: 25 ▼

Full path on server ↕	Set host ↕	Source type ↕	Index ↕	Number of files ↕	App ↕	Status ↕	Actions
\$SPLUNK_HOME/etc/apps/sample_app/logs	Constant Value	sendmail	sample	3	sample_app	Enabled	Disable Clone
\$SPLUNK_HOME/etc/splunk.version	Constant Value	splunk_version	_internal	1	system	Enabled	Disable Clone
\$SPLUNK_HOME/var/log/splunk	Constant Value	Automatic	_internal	16	system	Enabled	Disable Clone
\$SPLUNK_HOME/var/spool/splunk	Constant Value	Automatic	default	1	system	Enabled	Disable Clone
/opt/snort/logs	Constant Value	snort	default	5	SplunkforSnort	Enabled	Disable Clone Delete


Splunk Dashboard

splunk> Launcher Logged in as admin | App ▾ | Manager | Jobs | Logout


Welcome Your apps (4) Browse more apps

Your installed apps


Below you'll find Splunk apps to get the most out of your Splunk experience.

 ***NIX** [Enable](#)


*This is a useful app for helping monitor, manage, and troubleshoot *nix platforms. This app comes with set of scripted inputs for collecting CPU, disk, I/O, memory, log, configuration, and user info. It also provides convenient dashboards, form searches, and alerts to make getting started with Splunk a breeze.*

 **Getting started**

Get started with Splunk. This app introduces you to many of Splunk's features. You'll learn how to use Splunk to index data, search and investigate, add knowledge, monitor and alert, report and analyze.

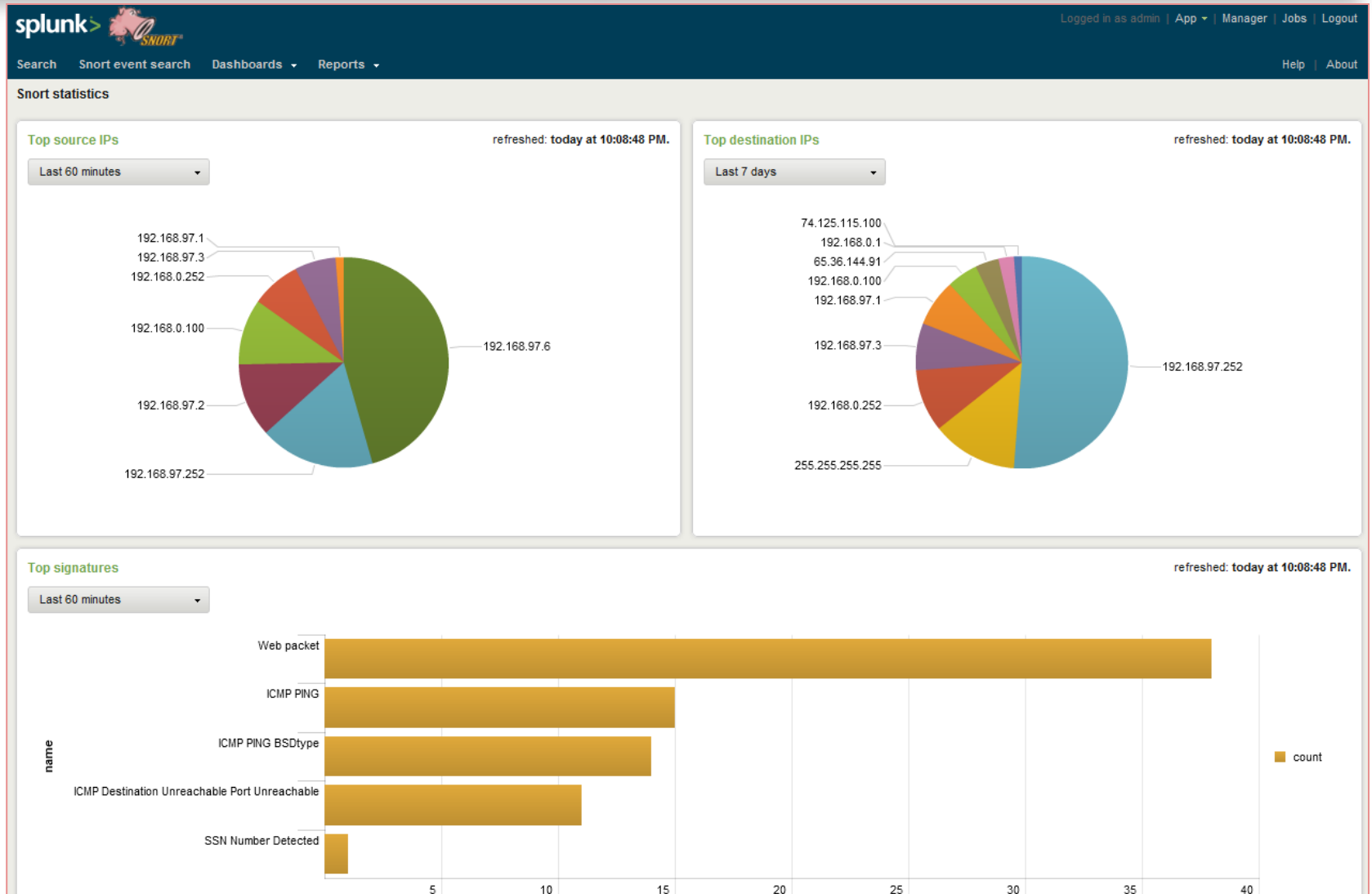
 **Search**

The Search app is Splunk's default interface for searching and analyzing IT data. It allows you to index data into Splunk, add knowledge, build reports, and create alerts. The Search app can be used across many areas of IT including application management, operations management, security, and compliance.

 **Splunk for Snort**

Splunk for Snort provides field extractions for Snort alert logs (fast and full) as well as dashboards, saved searches, event types, tags and event search interfaces.

Snort Dashboard in Splunk



Splunk Reports

The screenshot shows the Splunk web interface. At the top, the Splunk logo and 'SNORT' branding are visible. The user is logged in as 'admin'. The search bar contains the query 'sourcetype="snort" | top dest_ip'. The search results show 84 matching events. A timeline view is displayed, showing a period from 10:00 PM to 10:50 PM on Thursday, March 3, 2011. On the left, a list of 52 fields is shown, including 'host', 'source', 'sourcetype', and various other fields. The main results section shows 9 results over all time, with a table listing the top 10 destination IPs by count and percentage.

Search: Top 10 destination IPs | Actions

sourcetype="snort" | top dest_ip

All time

84 matching events

Save search Show report

Timeline: zoom in zoom out Scale: linear log

100 100
50 50

10:00 PM Thu Mar 3 2011 10:10 PM 10:20 PM 10:30 PM 10:40 PM 10:50 PM

52 fields | Pick fields

Selected fields (3)
host (1)
source (1)
sourcetype (1)

Other interesting fields (34)
Ack (n) (30)
bytes_in (n) (16)
dest_ip (9)
dest_port (n) (13)
DgmLen (n) (16)
dgmlen (n) (16)
eventtype (1)
generator_id (n) (2)
id (n) (71)
ID (n) (71)

9 results over all time

Options... Results per page 10

Overlay: None

	dest_ip ↕	count ↕	percent ↕
1	192.168.97.252	43	51.190476
2	255.255.255.255	11	13.095238
3	192.168.0.252	8	9.523810
4	192.168.97.3	6	7.142857
5	192.168.97.1	6	7.142857
6	192.168.0.100	4	4.761905
7	65.36.144.91	3	3.571429
8	192.168.0.1	2	2.380952
9	74.125.115.100	1	1.190476



Writing Snort Rules

Module Outline

- Anatomy of a Snort Rule
- Rule Headers
- Rule Options
- Getting Snort Rules from different sources
- Running Snort with Default Rule Set
- General Rule Options
- Rule Options for Dealing with Payload
- Rule Options for Non-payload Detection
- Rule Options for Taking Actions
- Rule Optimization

Snort Rule Anatomy

- Snort rules consist of two major parts:
 - Rule Header
 - Rule Options

- A sample rule will be as follows:

action protocol src_addr src_port direction dst_addr dst_port Options

- A real rule looks like this:

alert tcp any any -> any 21 (msg: "FTP Traffic");

- The red part is *header* and the green part is *options*

Snort Rule Header

- Rule Header has following parts:
 - Action
 - Protocol
 - IP Address
 - Port
 - Direction

`alert tcp any any -> any 21 (msg: "FTP Traffic");`

Rule Header: Actions

- Actions use the following keywords:
 - alert
 - log
 - pass
 - activate
 - dynamic
 - drop
 - reject
 - sdrop (Silently drop)

alert tcp any any -> any 21 (msg: "FTP Traffic");

Rule Header: Custom Actions

- In addition to default list of actions, you can create custom actions as well:

```
ruletype suspicious
```

```
{
```

```
  type log
```

```
  output log_tcpdump: suspicious.log
```

```
}
```

Rule Header: Protocol

- Following Protocols are supported in rules at this point:
 - TCP
 - UDP
 - IP
 - ICMP
- Future versions will support other protocols as well like:
 - IGMP
 - GRE
 - IPSec
- Selection of the protocol field impacts options part as well

Rule Header: IP Address

- Source and destination IP Addresses can be in multiple ways
 - Single IP address
 - CIDR
 - 192.168.2.0/24

- IP addresses can be negated using ! sign

```
alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 \  
(content:"|00 01 86 a5|";msg:"external mountd access");)
```

- List of IP addresses are used with comma separation

```
alert tcp ![192.168.1.0/24,10.1.1.0/24] any -> \  
[192.168.1.0/24,10.1.1.0/24] 111 (content:"|00 01 86 a5|"; \  
msg:"external mountd access");)
```

Rule Header: Ports

- Ports can be specified in multiple ways:
 - Single Port
 - Port Range 1:1024
- Ports can be negated using ! sign just like IP addresses
- Port ranges can have missing starting or ending range. If start is missing, it will be considered as 0 and if end is missing it will be considered as 65535
- Example :5000

log udp any any -> 192.168.1.0/24 1:1024

log tcp any :1024 -> 192.168.1.0/24 500:

Rule Header: Direction

- Direction Operator
 - Left to Right: ->
 - Both Ways: <>
 - There is no <- operator

log tcp any any -> 192.168.1.0/24 !6000:6010

log tcp !192.168.1.0/24 any <> 192.168.1.0/24 23

Rule Options

- Options have following four types:
 - General
 - Payload
 - Non-payload
 - Post-detection

General Rule Options: msg, reference

- The msg option is used to add text to rule output:

```
msg: "<message text>";
```

- Reference rule option adds reference to different sources

```
reference: <id system>, <id>; [reference: <id system>, <id>;]
```

- A real rule looks like the following:

```
alert tcp any any -> any 7070 (msg: "IDS411 / dos-realaudio"; \  
flags: AP; content: "\xffff4 fffd 06 | "; reference: arachnids, IDS411;)
```

General Rule Options: gid and sid

- GID – Used for specific systems and preprocessors (1 is default)

alert tcp any any -> any 80 (content:"BOB"; gid:1000001; sid:1; rev:1;)

- For custom rules, use gid anything about 1000000

- SID – Snort rule ID

- <100 Reserved for future use
- 100-999,999 Rules included with the Snort distribution
- >=1,000,000 Used for local rules

alert tcp any any -> any 80 (content:"BOB"; sid:1000983; rev:1;)

General Rule Options: classtype

- Represent classtype for grouping of alerts:

```
alert tcp any any -> any 25 (msg:"SMTP expn root"; flags:A+; \  
content:"expn root"; nocase; classtype:attempted-recon;)
```

- The file classification.conf in Snort is used to set up classifications for Snort rules

General Rule Options: metadata

- Used to put Key-Value pair in alerts:

```
alert tcp any any -> any 80 (msg:"HTTP Service Rule Example"; \  
metadata:service http;)
```

- Multiple key-value pairs can be used

```
metadata:key1 value1, key2 value2;
```

Payload Rule Options: content modifiers

- nocase

```
alert tcp any any -> any 21 (msg:"FTP ROOT"; content:"USER root";  
                               nocase;)
```

- rawbytes

```
alert tcp any any -> any 21 (msg:"Telnet NOP"; content:" | FF F1 | ";  
                               rawbytes;)
```

- depth - shows how far Snort should look into packet. Maxim is 64K (65535) bytes

- offset - where to start searching

```
alert tcp any any -> any 80 (content:"cgi-bin/phf"; offset:4; depth:20;)
```

Payload Rule Options: content modifiers

- distance - how far go before start searching for next pattern

alert tcp any any -> any any (content:"ABC"; content:"DEF"; distance:1;)

- within - following example search within 10 bytes distance

alert tcp any any -> any any (content:"ABC"; content:"EFG"; within:10;)

Payload Rule Options: content modifiers

- http client body

```
alert tcp any any -> any 80 (content:"ABC"; content:"EFG";  
http_client_body;)
```

- http cookie
- http raw cookie
- Depends upon enable_cookie key option.

```
alert tcp any any -> any 80 (content:"ABC"; content:"EFG";  
http_cookie;)
```

- http header
- http raw header
- http method

Payload Rule Options: content modifiers

- http uri
- http raw uri
- http stat code

```
alert tcp any any -> any 80 (content:"ABC"; content:"200";  
http_stat_code;)
```

- http stat msg

```
alert tcp any any -> any 80 (content:"ABC"; content:"Not Found";  
http_stat_msg;)
```

- fast pattern

Payload Rule Options: pcre

- PCRE (www.pcre.org) - Perl Compatible Regular Expressions

```
pcre:[!]"(/<regex>/ | m<delim><regex><delim>)  
[ismxAEGRUBPHMCOIDKYS]";
```

```
alert ip any any -> any any (pcre:"/BLAH/i");
```

Non Payload Options - ttl

- Time to Live - ttl
 - ttl:[<, >, =, <=, >=]<number>;
 - ttl:[<number>]-[<number>;
 - ttl:<=5;
 - ttl:>=5;
 - ttl:=5;

Non Payload Options (Cont.)

- TOS
 - tos:!4;
- ID
 - id:31337;

Non Payload Options (Cont.)

- IP Options
 - **rr** - Record Route
 - **eol** - End of list
 - **nop** - No Op
 - **ts** - Time Stamp
 - **sec** - IP Security
 - **esec** - IP Extended Security
 - **lsrr** - Loose Source Routing
 - **lsrre** - Loose Source Routing (For MS99-038 and CVE-1999-0909)
 - **ssrr** - Strict Source Routing
 - **satid** - Stream identifier
 - **any** - any IP options are set
- Example: *ipopts:lsrr;*

Non Payload Options (Cont.)

- The fragbits keyword is used to check if fragmentation and reserved bits are set in the IP header.
- The following bits may be checked:
 - M - More Fragments
 - D - Don't Fragment
 - R - Reserved Bit
- The following modifiers can be set to change the match criteria:
 - + match on the specified bits, plus any others
 - * match if any of the specified bits are set
 - ! match if the specified bits are not set
- This example checks if the More Fragments bit and the Do not Fragment bit are set.

fragbits:MD+;

Non Payload Options (Cont.)

- The `dsize` keyword is used to test the packet payload size. This may be used to check for abnormally sized packets. In many cases, it is useful for detecting buffer overflows.

- **Format**

`dsize:min<>max;`

`dsize:[< | >]<number>;`

- **Example**

- This example looks for a `dsize` that is between 300 and 400 bytes.

`dsize:300<>400;`

Non Payload Options (Cont.)

- TCP Flags
 - **F** - FIN - Finish (LSB in TCP Flags byte)
 - **S** - SYN - Synchronize sequence numbers
 - **R** - RST - Reset
 - **P** - PSH - Push
 - **A** - ACK - Acknowledgment
 - **U** - URG - Urgent
 - **1** - CWR - Congestion Window Reduced (MSB in TCP Flags byte)
 - **2** - ECE - ECN-Echo (If SYN, then ECN capable. Else, CE flag in IP header is set)
 - **0** - No TCP Flags Set

alert tcp any any -> any any (flags:SF;)

Non Payload Options (Cont.)

- Flow Checking

```
alert tcp !$HOME_NET any -> $HOME_NET 21 (msg:"cd incoming detected"; flow:from_client; content:"CWD incoming"; nocase;)
```

Non Payload Options (Cont.)

- Other TCP Parameters:
 - Sequence Number: *seq:0*;
 - Acknowledge Number: *ack:0*;
 - Window size: *window:55808*;

- ICMP Parameters Examples
 - *itype:>30*;
 - *icode:>30*;
 - *icmp_id:0*;
 - *icmp_seq:0*;

Non Payload Options (Cont.)

- IP Protocol Examples:

- *alert ip any any -> any any (ip_proto:igmp;)*
- *alert ip any any -> any any (sameip;)*

Non Payload Options (Cont.)

- The stream reassemble keyword allows a rule to enable or disable TCP stream reassembly on matching traffic.

```
alert tcp any 80 -> any any (flow:to_client, established; content:"200 OK";  
stream_reassemble:disable,client,noalert;)
```

- The stream size keyword allows a rule to match traffic according to the number of bytes observed, as determined by the TCP sequence numbers.

```
alert tcp any any -> any any (stream_size:client,<,6;)
```

Post Detection Options - logto

- The logto is used to log data to a file
- Useful to capture data from HTTP, nmap etc

logto:"filename";

Post Detection Options - session

- The following example logs all printable strings in a telnet packet.

```
log tcp any any <> any 23 (session:printable;)
```

- Given an FTP data session on port 12345, this example logs the payload bytes in binary form.

```
log tcp any any <> any 12345 (metadata:service ftp-data; session:binary;)
```

Snort Rules

- Emerging Threats Snort Rules

<http://rules.emergingthreats.net/open-nogpl/snort-2.9.0/>

Using Snort as PII Data Detection Tool

- Use Sensitive Data preprocessor or write specific rules.

- Enable Preprocessor

```
preprocessor sensitive_data: alert_threshold 10
```

- Write Rules

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $SMTP_PORTS \  
(msg:"SSN sent via email"; gid:138; sid:5001; rev:2 ; sd_pattern:1,us_social;)
```

- Mark Output

```
preprocessor sensitive_data: alert_threshold 10 mask_output
```

- Use pcre to write custom rules for detecting other PII data.

Use Snort Reputational IPS

- Use white list and black list functionality
- Black list and whitelist files are included in snort.conf file
- Sources of Black List - dshield Block List
 - <http://feeds.dshield.org/block.txt>
 - <http://feeds.dshield.org/block.txt>

*preprocessor reputation: *

*blacklist /etc/snort/blacklist.list, *

whitelist /etc/snort/whitelist.list

Using Snort for Insecure Protocols Detection

- PCI Compliance
 - FTP
 - Telnet
 - POP3 - Default
 - IMAP – Default
 - Other protocols detection rules
- Other compliance needs to detect information on the Internet
- Compliance Evidence Tool

Thank You

<http://rafeeqrehman.com>

Twitter: @rafeeq_rehman

Email: rafeeq.rehman@gmail.com