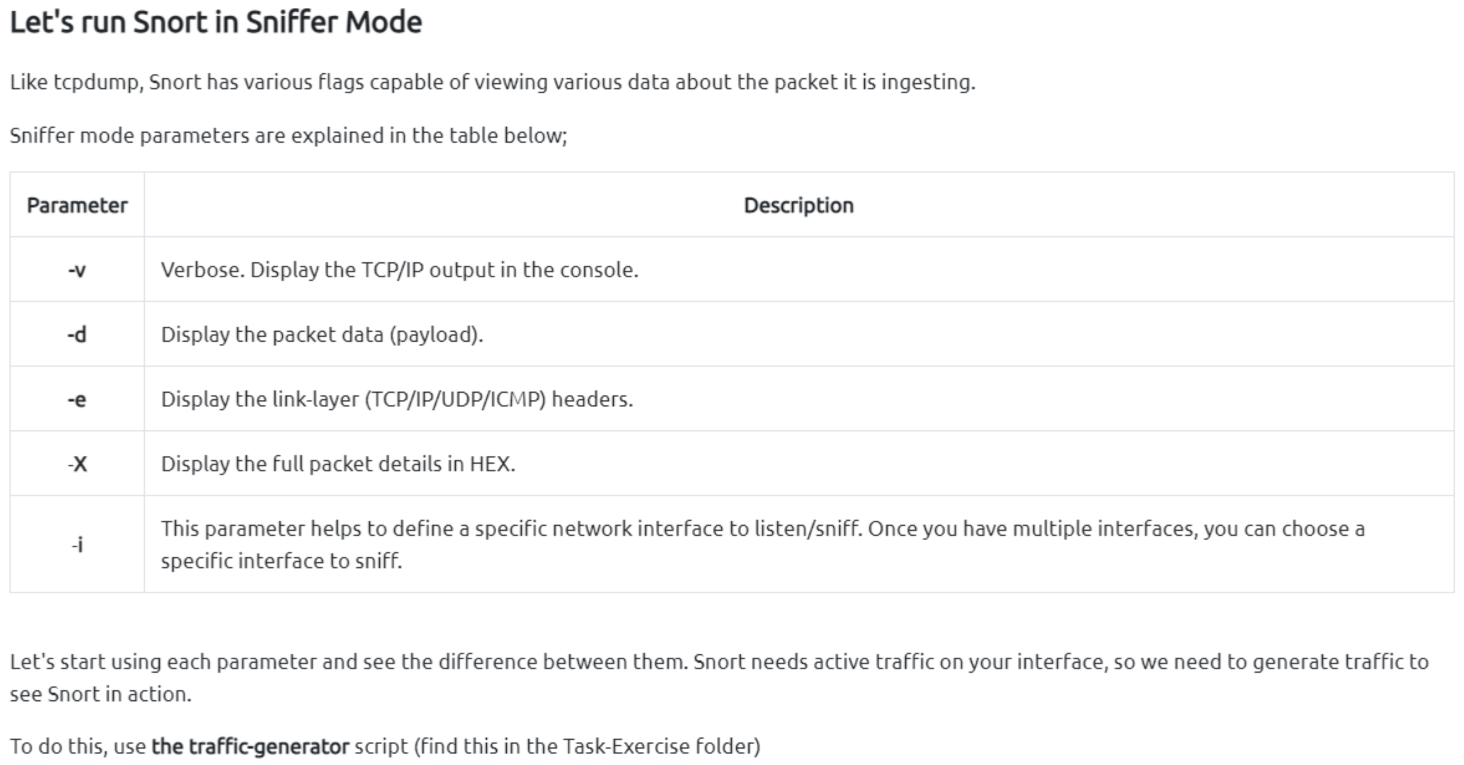
Snort

Snort has 4 modes:

1) Sniffer Mode:



Let's run Snort in Sniffer Mode

Like tcpdump, Snort has various flags capable of viewing various data about the packet it is ingesting.

Sniffer mode parameters are explained in the table below;

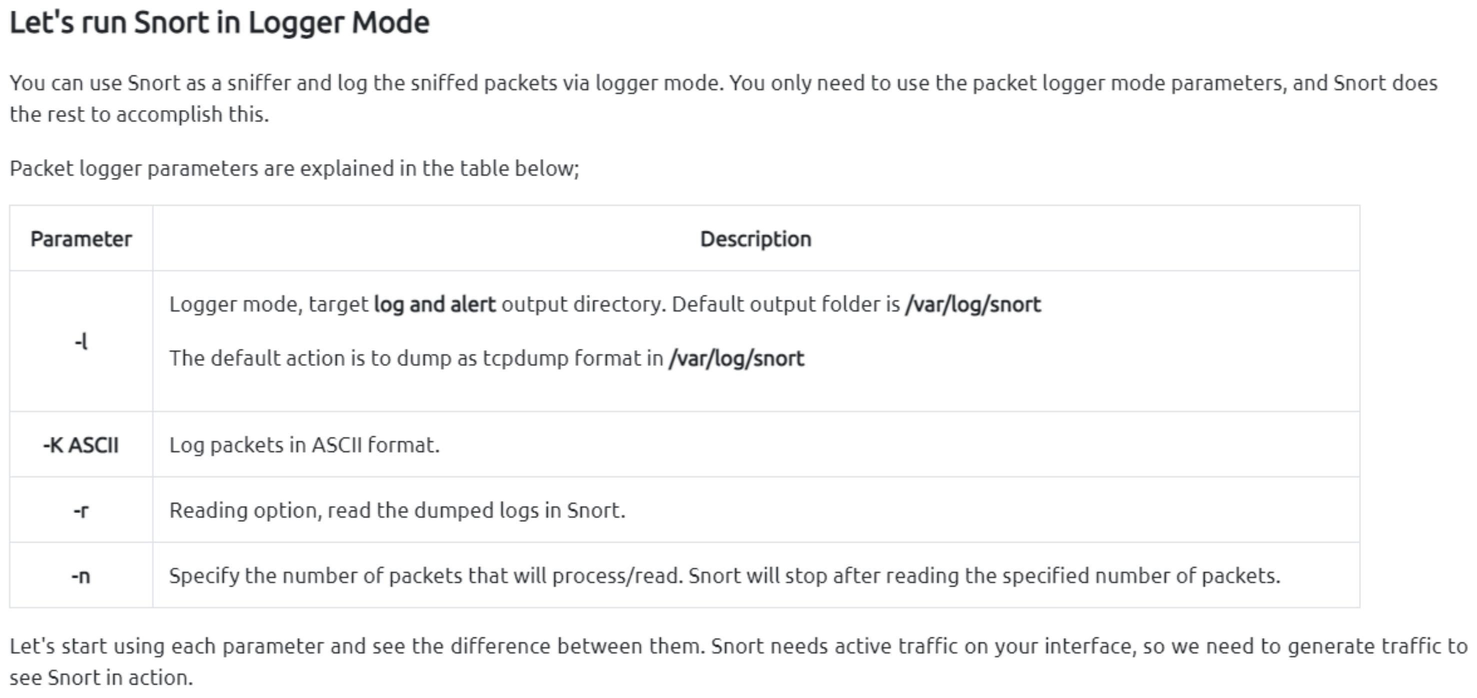
Parameter	Description
-v	Verbose. Display the TCP/IP output in the console.
-d	Display the packet data (payload).
-e	Display the link-layer (TCP/IP/UDP/ICMP) headers.
-**X**	Display the full packet details in HEX.
-i	This parameter helps to define a specific network interface to listen/sniff. Once you have multiple interfaces, you can choose a specific interface to sniff.

Let's start using each parameter and see the difference between them. Snort needs active traffic on your interface, so we need to generate traffic to see Snort in action.

To do this, use **the traffic-generator** script (find this in the Task-Exercise folder)

A note about -X : it is a **full packet dump mode (-X)**

2) Logger Mode:



Let's run Snort in Logger Mode

You can use Snort as a sniffer and log the sniffed packets via logger mode. You only need to use the packet logger mode parameters, and Snort does the rest to accomplish this.

Packet logger parameters are explained in the table below;

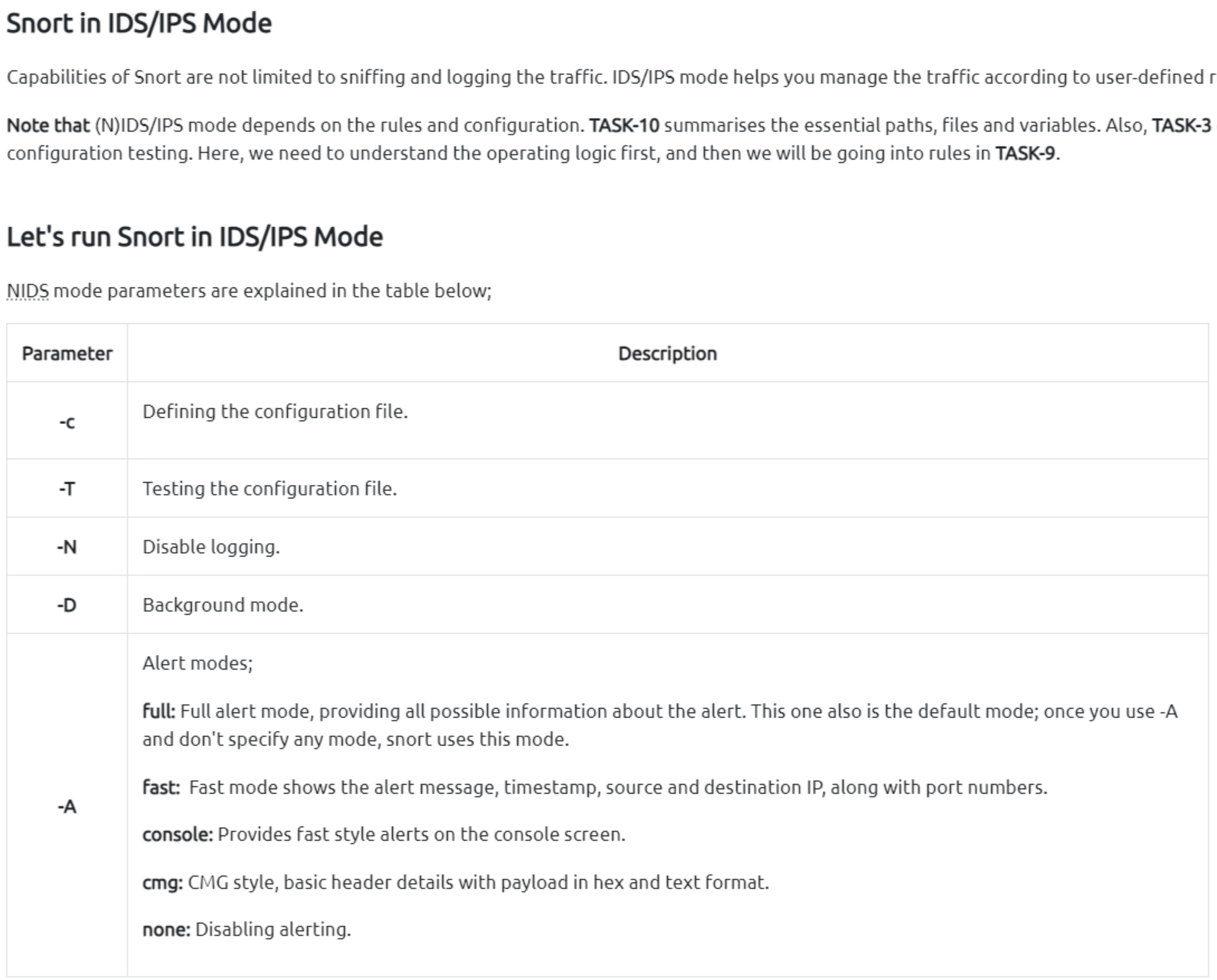
Parameter	Description
-l	Logger mode, target **log and alert** output directory. Default output folder is **/var/log/snort** The default action is to dump as tcpdump format in **/var/log/snort**
-K ASCII	Log packets in ASCII format.
-r	Reading option, read the dumped logs in Snort.
-n	Specify the number of packets that will process/read. Snort will stop after reading the specified number of packets.

Let's start using each parameter and see the difference between them. Snort needs active traffic on your interface, so we need to generate traffic to see Snort in action.

note#1: to save log files in a directory other than /var/log/snort ; you can either edit the snort.conf file or specify the path after the -l option.

note#2: When you use the -K ASCII option snort will no longer be able to process the generated files; you need to use the GUI to read them, but still snort will generate a binary log file that it will be able to process.

3) IDS/IPS Mode:



Snort in IDS/IPS Mode

Capabilities of Snort are not limited to sniffing and logging the traffic. IDS/IPS mode helps you manage the traffic according to user-defined rules.

Note that (N)IDS/IPS mode depends on the rules and configuration. **TASK-10** summarises the essential paths, files and variables. Also, **TASK-3** covers configuration testing. Here, we need to understand the operating logic first, and then we will be going into rules in **TASK-9**.

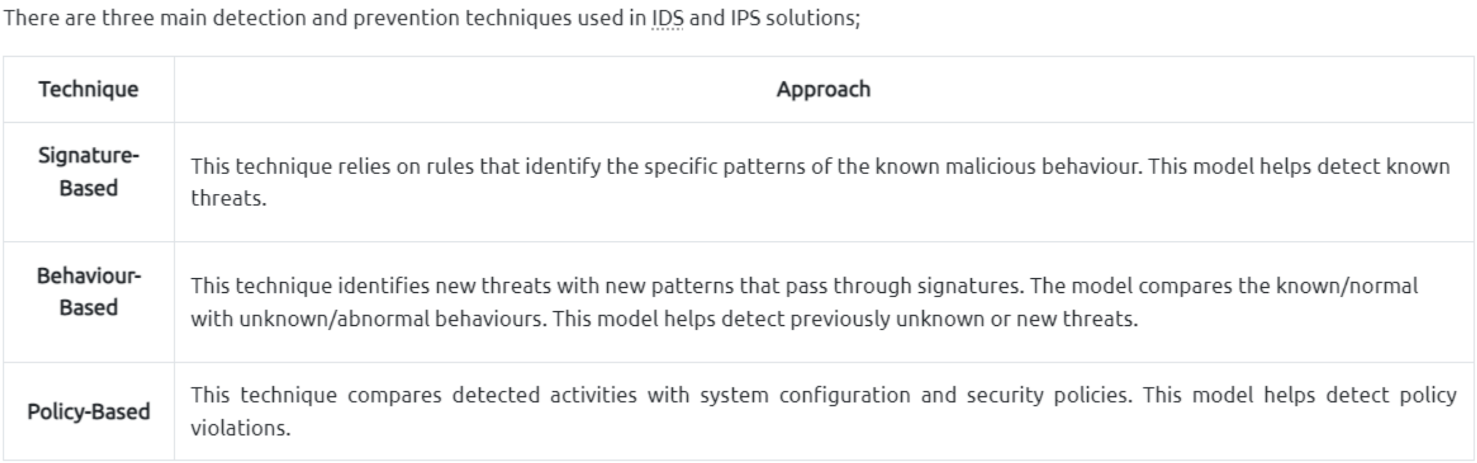
Let's run Snort in IDS/IPS Mode

NIDS mode parameters are explained in the table below;

Parameter	Description
-c	Defining the configuration file.
-T	Testing the configuration file.
-N	Disable logging.
-D	Background mode.
-A	Alert modes; **full:** Full alert mode, providing all possible information about the alert. This one also is the default mode; once you use -A and don't specify any mode, snort uses this mode. **fast:** Fast mode shows the alert message, timestamp, source and destination IP, along with port numbers. **console:** Provides fast style alerts on the console screen. **cmg:** CMG style, basic header details with payload in hex and text format. **none:** Disabling alerting.

*To run the snort in IPS mode use this command: `snort -c /etc/snort/snort.conf -q -Q --daq afpacket`
remember that snort needs root privileges to write to files(alerts and logs).

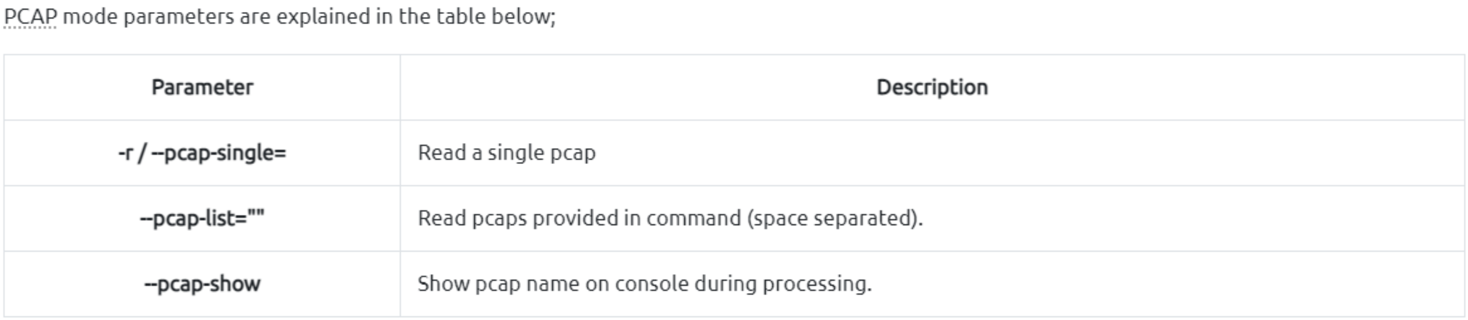
note: the default snort conf file is in /etc/snort/snort.conf ; this file can be modified so that the snort can prevent packets from getting in based on the signature you provide



There are three main detection and prevention techniques used in IDS and IPS solutions;

Technique	Approach
Signature-Based	This technique relies on rules that identify the specific patterns of the known malicious behaviour. This model helps detect known threats.
Behaviour-Based	This technique identifies new threats with new patterns that pass through signatures. The model compares the known/normal with unknown/abnormal behaviours. This model helps detect previously unknown or new threats.
Policy-Based	This technique compares detected activities with system configuration and security policies. This model helps detect policy violations.

4) Investigating pcaps Mode:



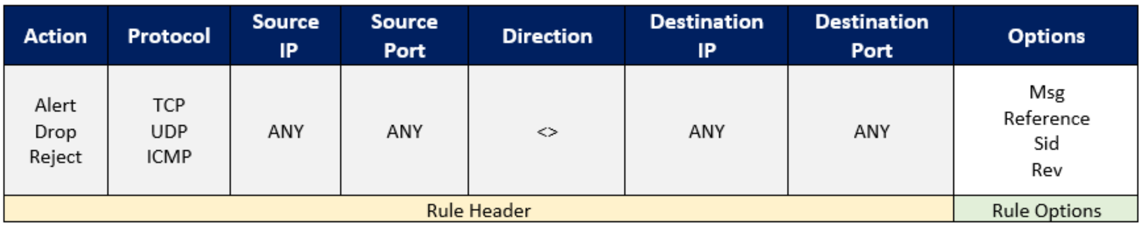
PCAP mode parameters are explained in the table below;

Parameter	Description
-r / --pcap-single=	Read a single pcap
--pcap-list=""	Read pcaps provided in command (space separated).
--pcap-show	Show pcap name on console during processing.

note: for the --pcap-list="" they should be space seperated.

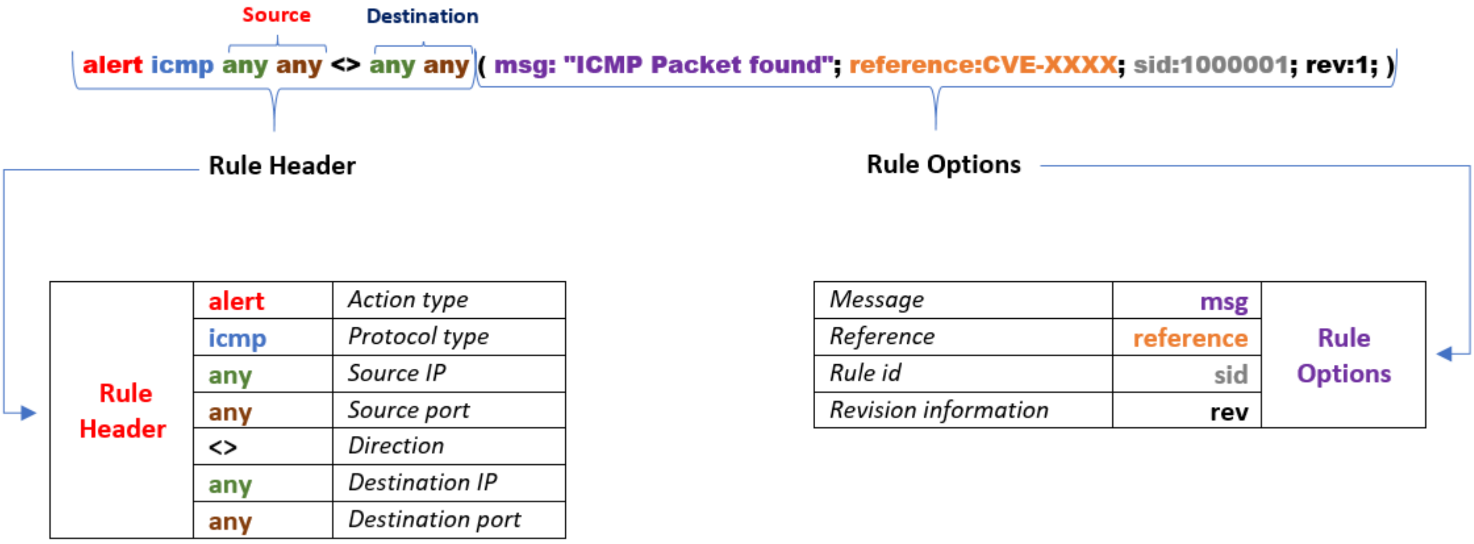
Snort Rules

Snort Rules structuer is shown below



Action	Protocol	Source IP	Source Port	Direction	Destination IP	Destination Port	Options
Alert Drop Reject	TCP UDP ICMP	ANY	ANY	<>	ANY	ANY	Msg Reference Sid Rev
Rule Header							Rule Options

➢ The following rule will generate an alert for each ICMP packet processed by Snort;



```
alert icmp any any <> any any ( msg: "ICMP Packet found"; reference:CVE-XXXX; sid:1000001; rev:1; )
```

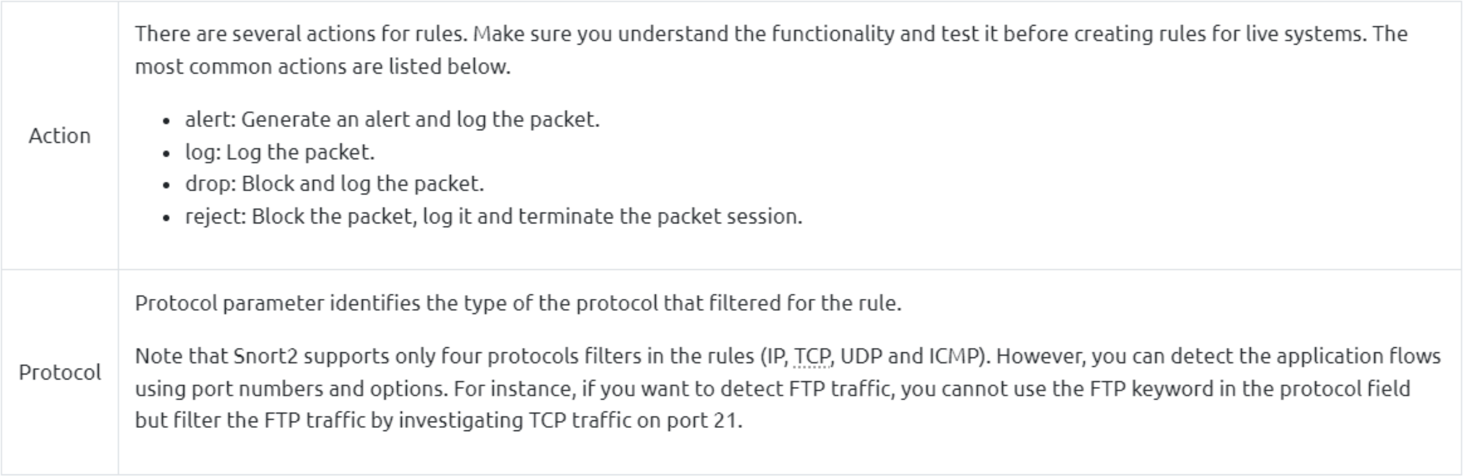
Rule Header	alert	*Action type*
	icmp	*Protocol type*
	any	*Source IP*
	any	*Source port*
	<>	*Direction*
	any	*Destination IP*
	any	*Destination port*

Message	msg	Rule Options
Reference	reference	
Rule id	sid	
Revision information	rev	

Remember, once you create a rule, it is a local rule and should be in your "local.rules" file. This file is located under "/etc/snort/rules/local.rules".

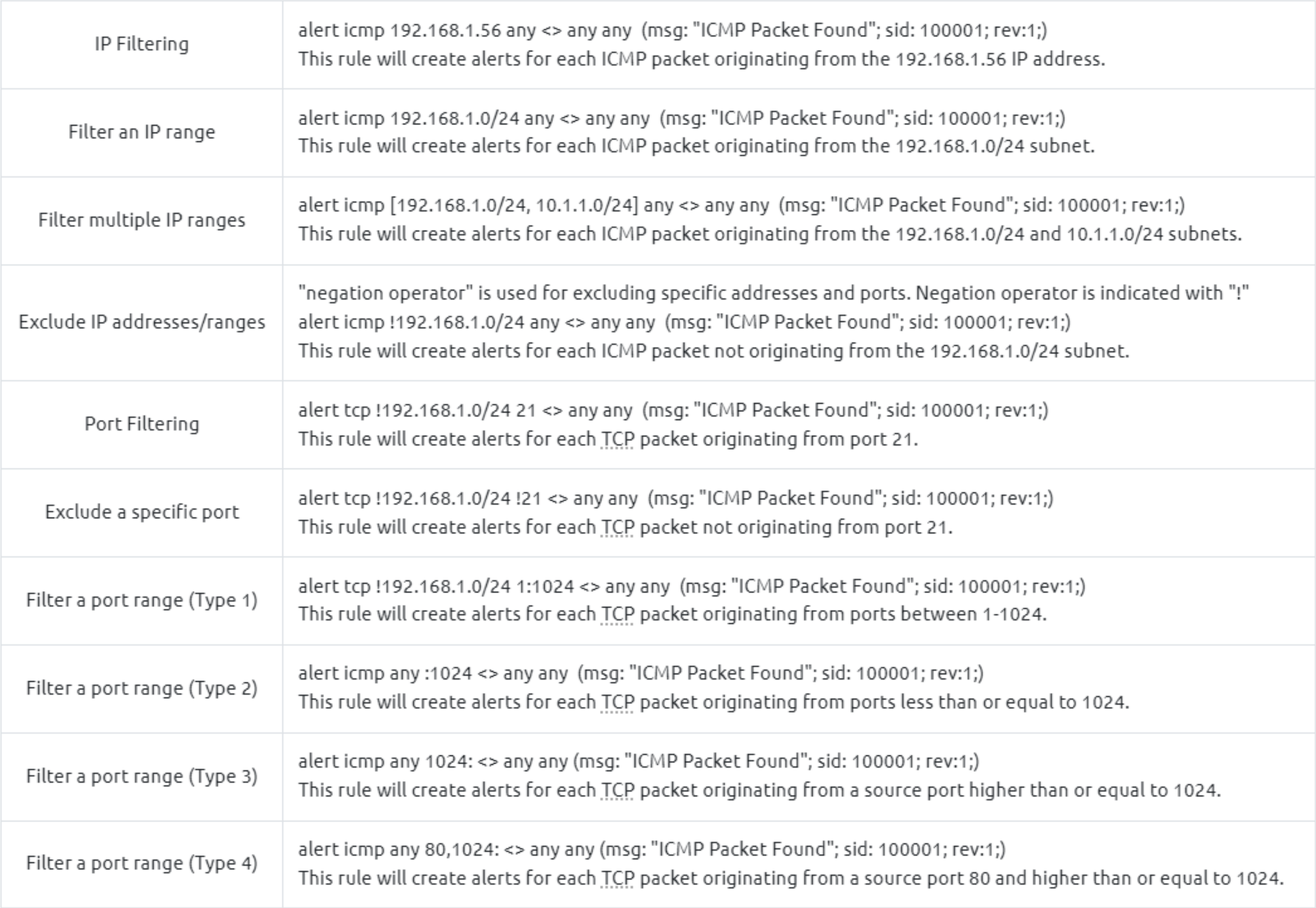
We will cover the basic rule structure in this room and help you take a step into snort rules. You can always advance your rule creation skills with different rule options by practising different use cases and studying rule option details in depth. We will focus on two actions; **"alert"** for IDS mode and **"reject"** for IPS m-ode.

Rules cannot be processed without a header. Rule options are "optional" parts. However, it is almost impossible to detect sophisticated attacks without using the rule options.



Action	There are several actions for rules. Make sure you understand the functionality and test it before creating rules for live systems. The most common actions are listed below. • alert: Generate an alert and log the packet. • log: Log the packet. • drop: Block and log the packet. • reject: Block the packet, log it and terminate the packet session.
Protocol	Protocol parameter identifies the type of the protocol that filtered for the rule. Note that Snort2 supports only four protocols filters in the rules (IP, TCP, UDP and ICMP). However, you can detect the application flows using port numbers and options. For instance, if you want to detect FTP traffic, you cannot use the FTP keyword in the protocol field but filter the FTP traffic by investigating TCP traffic on port 21.

IP and Port Numbers filtering:



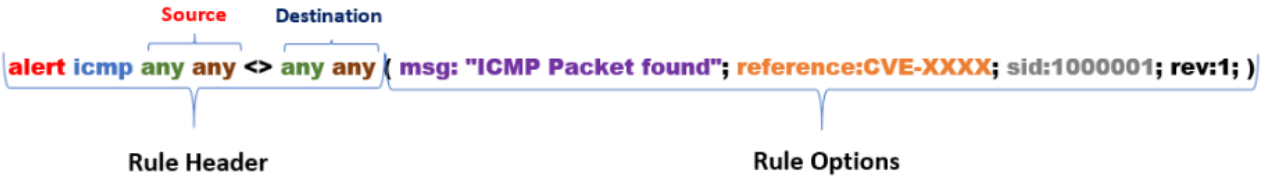
IP Filtering	alert icmp 192.168.1.56 any <> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;) This rule will create alerts for each ICMP packet originating from the 192.168.1.56 IP address.
Filter an IP range	alert icmp 192.168.1.0/24 any <> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;) This rule will create alerts for each ICMP packet originating from the 192.168.1.0/24 subnet.
Filter multiple IP ranges	alert icmp [192.168.1.0/24, 10.1.1.0/24] any <> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;) This rule will create alerts for each ICMP packet originating from the 192.168.1.0/24 and 10.1.1.0/24 subnets.
Exclude IP addresses/ranges	"negation operator" is used for excluding specific addresses and ports. Negation operator is indicated with "!" alert icmp !192.168.1.0/24 any <> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;) This rule will create alerts for each ICMP packet not originating from the 192.168.1.0/24 subnet.
Port Filtering	alert tcp !192.168.1.0/24 21 <> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;) This rule will create alerts for each TCP packet originating from port 21.
Exclude a specific port	alert tcp !192.168.1.0/24 !21 <> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;) This rule will create alerts for each TCP packet not originating from port 21.
Filter a port range (Type 1)	alert tcp !192.168.1.0/24 1:1024 <> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;) This rule will create alerts for each TCP packet originating from ports between 1-1024.
Filter a port range (Type 2)	alert icmp any :1024 <> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;) This rule will create alerts for each TCP packet originating from ports less than or equal to 1024.
Filter a port range (Type 3)	alert icmp any 1024: <> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;) This rule will create alerts for each TCP packet originating from a source port higher than or equal to 1024.
Filter a port range (Type 4)	alert icmp any 80,1024: <> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;) This rule will create alerts for each TCP packet originating from a source port 80 and higher than or equal to 1024.

Direction

The direction operator indicates the traffic flow to be filtered by Snort. The left side of the rule shows the source, and the right side shows the destination.

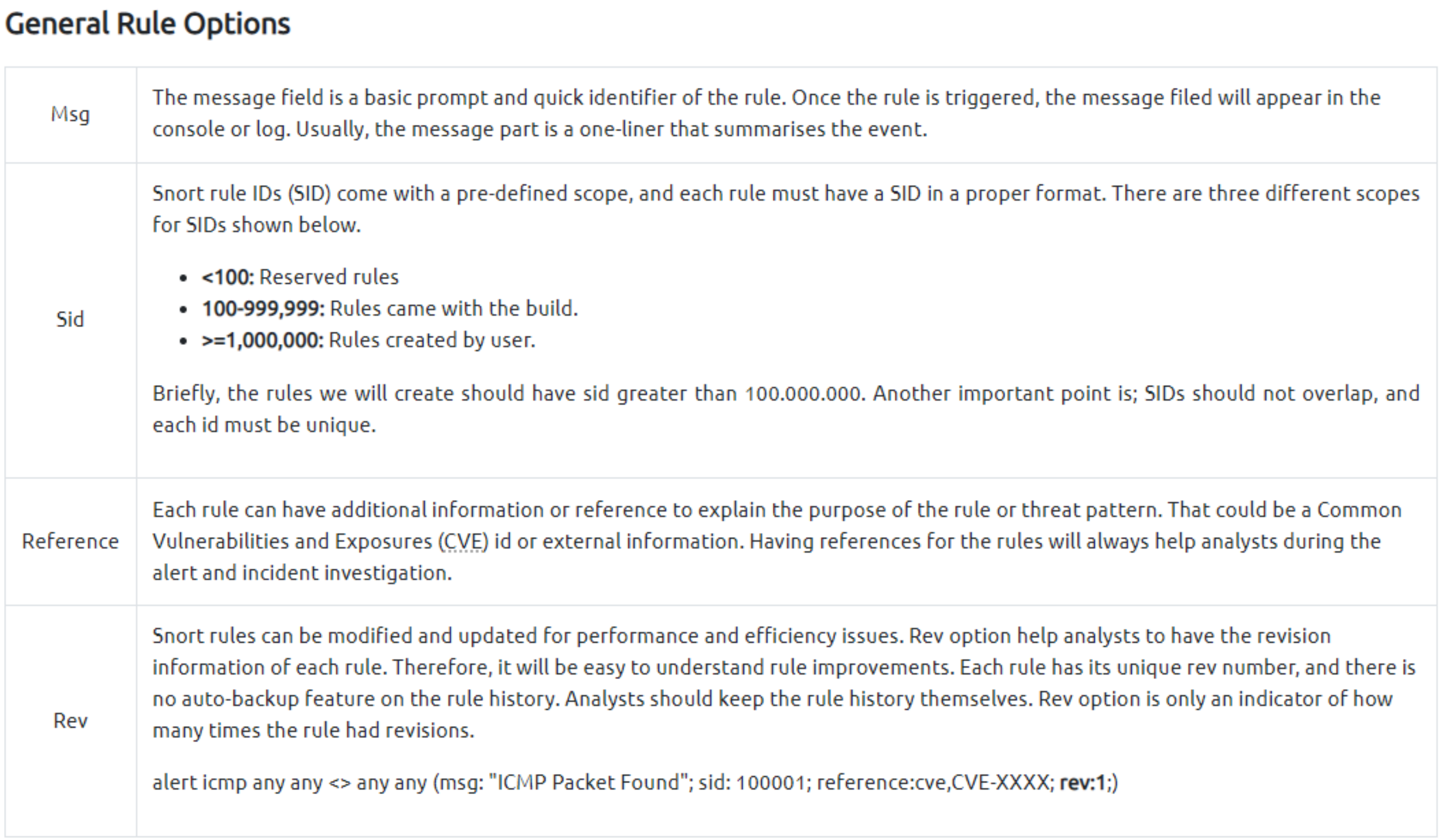
- **->** Source to destination flow.
- **<>** Bidirectional flow

Note that there is no "<-" operator in Snort.


Source
Destination
alert icmp any any <> any any (msg: "ICMP Packet found"; reference:CVE-XXXX; sid:1000001; rev:1;)
Rule Header
Rule Options

There are three main rule options in Snort;

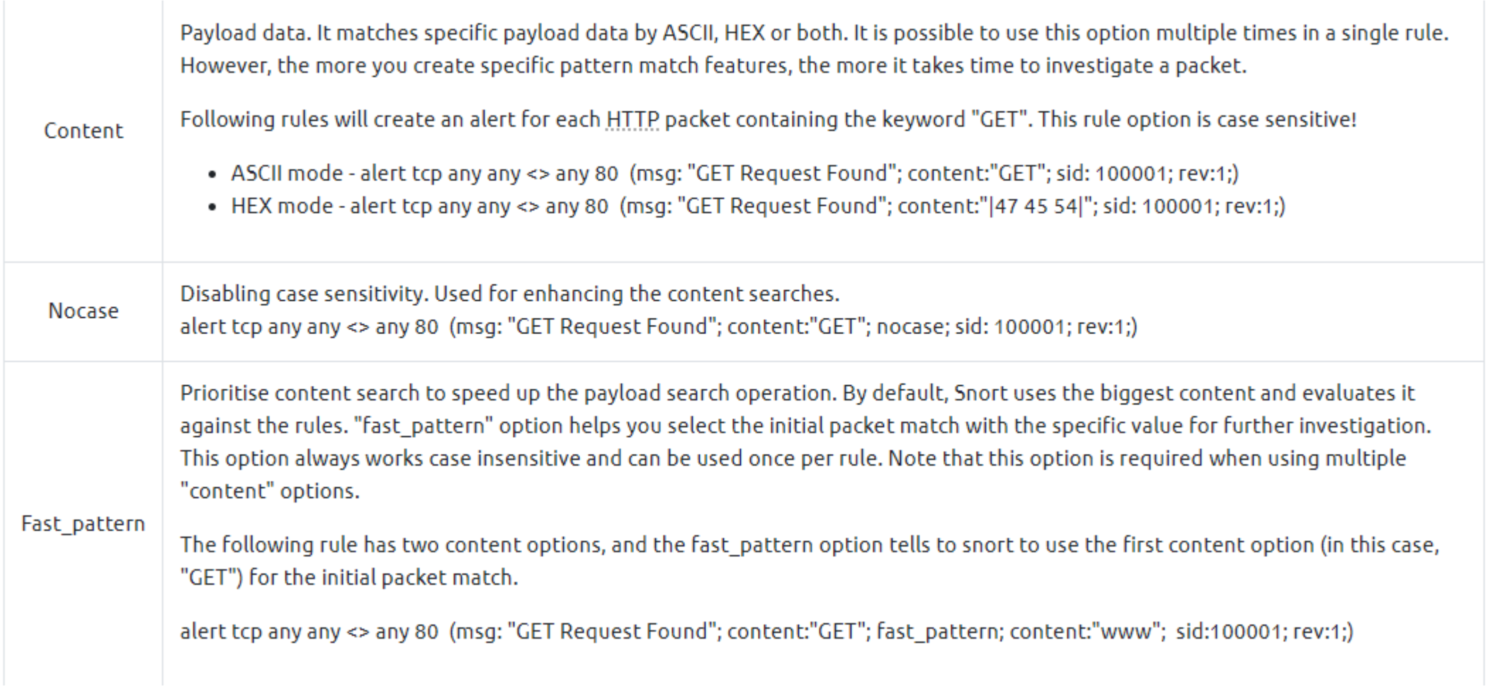
- **General Rule Options -** Fundamental rule options for Snort.
- **Payload Rule Options -** Rule options that help to investigate the payload data. These options are helpful to detect specific payload patterns.
- **Non-Payload Rule Options -** Rule options that focus on non-payload data. These options will help create specific patterns and identify network issues.



General Rule Options

Msg	The message field is a basic prompt and quick identifier of the rule. Once the rule is triggered, the message filed will appear in the console or log. Usually, the message part is a one-liner that summarises the event.
Sid	Snort rule IDs (SID) come with a pre-defined scope, and each rule must have a SID in a proper format. There are three different scopes for SIDs shown below. • **<100:** Reserved rules • **100-999,999:** Rules came with the build. • **>=1,000,000:** Rules created by user. Briefly, the rules we will create should have sid greater than 100.000.000. Another important point is; SIDs should not overlap, and each id must be unique.
Reference	Each rule can have additional information or reference to explain the purpose of the rule or threat pattern. That could be a Common Vulnerabilities and Exposures (CVE) id or external information. Having references for the rules will always help analysts during the alert and incident investigation.
Rev	Snort rules can be modified and updated for performance and efficiency issues. Rev option help analysts to have the revision information of each rule. Therefore, it will be easy to understand rule improvements. Each rule has its unique rev number, and there is no auto-backup feature on the rule history. Analysts should keep the rule history themselves. Rev option is only an indicator of how many times the rule had revisions. alert icmp any any <> any any (msg: "ICMP Packet Found"; sid: 100001; reference:cve,CVE-XXXX; **rev:1**;)

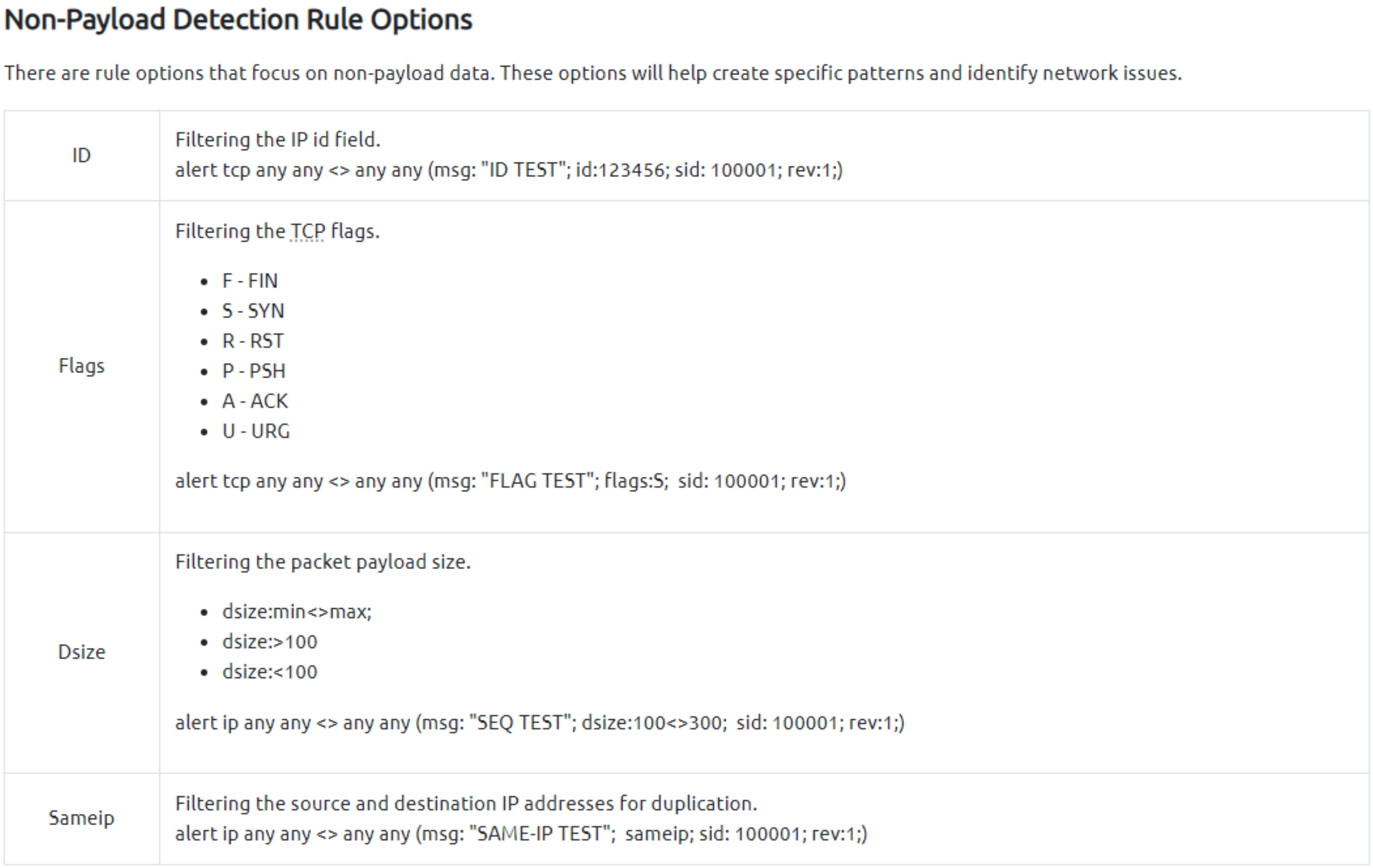
Payload Detection Rule Options



Content	Payload data. It matches specific payload data by ASCII, HEX or both. It is possible to use this option multiple times in a single rule. However, the more you create specific pattern match features, the more it takes time to investigate a packet. Following rules will create an alert for each HTTP packet containing the keyword "GET". This rule option is case sensitive! • ASCII mode - alert tcp any any <> any 80 (msg: "GET Request Found"; content:"GET"; sid: 100001; rev:1;) • HEX mode - alert tcp any any <> any 80 (msg: "GET Request Found"; content:"|47 45 54|"; sid: 100001; rev:1;)
Nocase	Disabling case sensitivity. Used for enhancing the content searches. alert tcp any any <> any 80 (msg: "GET Request Found"; content:"GET"; nocase; sid: 100001; rev:1;)
Fast_pattern	Prioritise content search to speed up the payload search operation. By default, Snort uses the biggest content and evaluates it against the rules. "fast_pattern" option helps you select the initial packet match with the specific value for further investigation. This option always works case insensitive and can be used once per rule. Note that this option is required when using multiple "content" options. The following rule has two content options, and the fast_pattern option tells to snort to use the first content option (in this case, "GET") for the initial packet match. alert tcp any any <> any 80 (msg: "GET Request Found"; content:"GET"; fast_pattern; content:"www"; sid:100001; rev:1;)

Non-Payload Detection Rule Options

There are rule options that focus on non-payload data. These options will help create specific patterns and identify network issues.



Non-Payload Detection Rule Options

There are rule options that focus on non-payload data. These options will help create specific patterns and identify network issues.

ID	Filtering the IP id field. alert tcp any any <> any any (msg: "ID TEST"; id:123456; sid: 100001; rev:1;)
Flags	Filtering the TCP flags. • F - FIN • S - SYN • R - RST • P - PSH • A - ACK • U - URG alert tcp any any <> any any (msg: "FLAG TEST"; flags:S; sid: 100001; rev:1;)
Dsize	Filtering the packet payload size. • dsize:min<>max; • dsize:>100 • dsize:<100 alert ip any any <> any any (msg: "SEQ TEST"; dsize:100<>300; sid: 100001; rev:1;)
Sameip	Filtering the source and destination IP addresses for duplication. alert ip any any <> any any (msg: "SAME-IP TEST"; sameip; sid: 100001; rev:1;)

points to remember

Points to Remember

Main Components of Snort

- **Packet Decoder -** Packet collector component of Snort. It collects and prepares the packets for pre-processing.
- **Pre-processors -** A component that arranges and modifies the packets for the detection engine.
- **Detection Engine -** The primary component that process, dissect and analyse the packets by applying the rules.
- **Logging and Alerting -** Log and alert generation component.
- **Outputs and Plugins -** Output integration modules (i.e. alerts to syslog/mysql) and additional plugin (rule management detection plugins) support is done with this component.

There are three types of rules available for snort

◇ **Community Rules -** Free ruleset under the GPLv2. Publicly accessible, no need for registration.
◇ **Registered Rules -** Free ruleset (requires registration). This ruleset contains subscriber rules with 30 days delay.
◇ **Subscriber Rules (Paid) -** Paid ruleset (requires subscription). This ruleset is the main ruleset and is updated twice a week (Tuesdays and Thursdays).

You can download and read more on the rules here.

Note: Once you install Snort2, it automatically creates the required directories and files. However, if you want to use the community or the paid rules, you need to indicate each rule in the **snort.conf** file.

Since it is a long, all-in-one configuration file, editing it without causing misconfiguration is troublesome for some users. **That is why Snort has several rule updating modules and integration**

tools. **To sum up, never replace your configured Snort configuration files; you must edit your configuration files manually or update your rules with additional tools and modules to not face any fail/crash or lack of feature.**

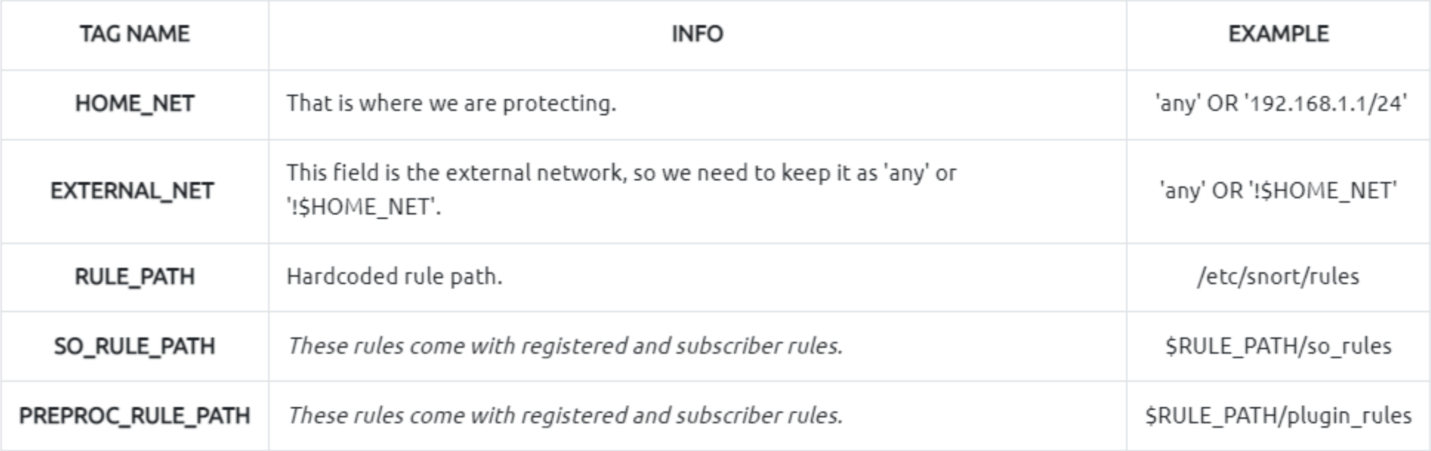
◇ **snort.conf:** *Main configuration file.*

◇ **local.rules:** *User-generated rules file.*

Let's start with overviewing the main configuration file (snort.conf) `sudo gedit /etc/snort/snort.conf`

Navigate to the "Step #1: Set the network variables." section.

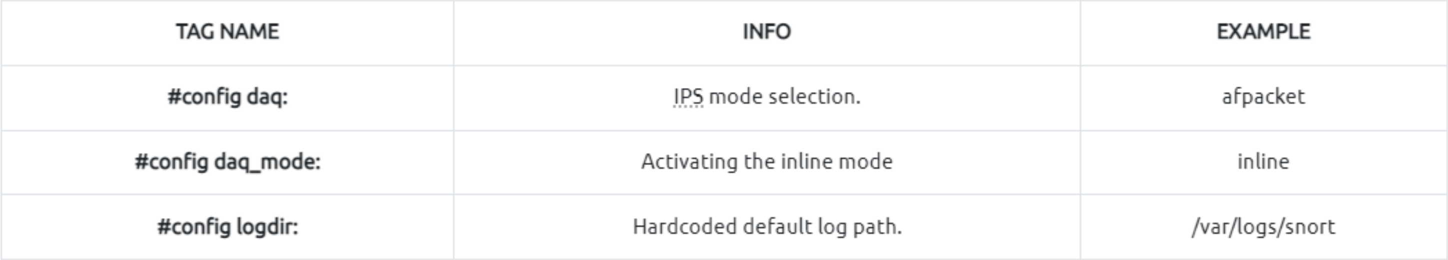
This section manages the scope of the detection and rule paths.



TAG NAME	INFO	EXAMPLE
HOME_NET	That is where we are protecting.	'any' OR '192.168.1.1/24'
EXTERNAL_NET	This field is the external network, so we need to keep it as 'any' or '!$HOME_NET'.	'any' OR '!$HOME_NET'
RULE_PATH	Hardcoded rule path.	/etc/snort/rules
SO_RULE_PATH	*These rules come with registered and subscriber rules.*	$RULE_PATH/so_rules
PREPROC_RULE_PATH	*These rules come with registered and subscriber rules.*	$RULE_PATH/plugin_rules

Navigate to the "Step #2: Configure the decoder." section.

In this section, you manage the IPS mode of snort. The single-node installation model IPS model works best with "afpacket" mode. You can enable this mode and run Snort in IPS.



TAG NAME	INFO	EXAMPLE
#config daq:	IPS mode selection.	afpacket
#config daq_mode:	Activating the inline mode	inline
#config logdir:	Hardcoded default log path.	/var/logs/snort

Data Acquisition Modules (DAQ) are specific libraries used for packet I/O, bringing flexibility to process packets. It is possible to select DAQ type and mode for different purposes.
There are six DAQ modules available in Snort;

- **Pcap:** Default mode, known as Sniffer mode.
- **Afpacket:** Inline mode, known as IPS mode.
- **Ipq:** Inline mode on Linux by using Netfilter. It replaces the snort_inline patch.
- **Nfq:** Inline mode on Linux.
- **Ipfw:** Inline on OpenBSD and FreeBSD by using divert sockets, with the pf and ipfw firewalls.

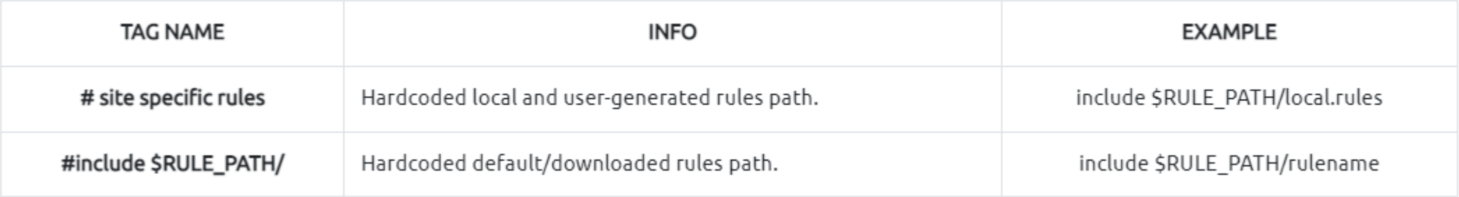
- **Dump:** Testing mode of inline and normalisation.

The most popular modes are the default (pcap) and inline/IPS (Afpacket).

Navigate to the "Step #6: Configure output plugins" section.

This section manages the outputs of the IDS/IPS actions, such as logging and alerting format details. The default action prompts everything in the console application, so configuring this part will help you use the Snort more efficiently.

Navigate to the "Step #7: Customise your ruleset" section.



TAG NAME	INFO	EXAMPLE
# site specific rules	Hardcoded local and user-generated rules path.	include $RULE_PATH/local.rules
#include $RULE_PATH/	Hardcoded default/downloaded rules path.	include $RULE_PATH/rulename

Note that "#" is commenting operator. You should uncomment a line to activate it.