snort -V (check snort version)

sudo snort -c /etc/snort/snort.conf -T

we should ensure our configuration file is valid.

Here "-T" is used for testing configuration, and "-c" is identifying the configuration file (snort.conf). Note that it is possible to use an additional configuration file by pointing it with "-c".

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| **-V / --version** | This parameter provides information about your instance version. |
| **-c** | Identifying the configuration file |
| **-T** | Snort's self-test parameter, you can test your setup with this parameter. |
| -**q** | Quiet mode prevents snort from displaying the default banner and initial information about your setup. |

Snort In Sniff 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. |

**Sniffing with parameter "-i"**

Start the Snort instance in **verbose mode (-v)** and **use the interface (-i)** "eth0"; sudo snort -v-i eth0

**Sniffing with parameter "-v"**

Start the Snort instance in **verbose mode (-v)**; sudo snort -v

Now run the traffic-generator script as sudo and start **ICMP/HTTP traffic**. Once the traffic is generated, snort will start showing the  packets in verbosity mode

**Sniffing with parameter "-d"**

Start the Snort instance in **dumping packet data mode (-d)**; sudo snort -d

Now run the traffic-generator script **as sudo** and start **ICMP/HTTP traffic**. Once the traffic is generated, snort will start showing the  packets in verbosity mode

**Sniffing with parameter "-de"**

Start the Snort instance in **dump (-d)** and **link-layer header grabbing (-e)** mode; snort -d -e

Now run the traffic-generator script **as sudo** and start **ICMP/HTTP traffic**. Once the traffic is generated, snort will start showing the  packets in verbosity mode

**Sniffing with parameter "-X"**

Start the Snort instance in**full packet dump mode (-X)**; sudo snort -X

Now run the traffic-generator script **as sudo** and start **ICMP/HTTP traffic**. Once the traffic is generated, snort will start showing the  packets in verbosity 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.

First, start the Snort instance in packet logger mode; sudo snort -dev -l .

Now start ICMP/HTTP traffic with the traffic-generator script.

Once the traffic is generated, Snort will start showing the packets and log them in the target directory. You can configure the default output directory in snort.config file. However, you can use the "-l" parameter to set a target directory. Identifying the default log directory is useful for continuous monitoring operations, and the "-l" parameter is much more useful for testing purposes.

The -l . part of the command creates the logs in the current directory. You will need to use this option to have the logs for each exercise in their folder.

sudo snort -dev -l . (Create log in current directory)

**Logging with parameter "-K ASCII"**

Start the Snort instance in packet logger mode; sudo snort -dev -K ASCII

Now run the traffic-generator script as sudo and start **ICMP/HTTP traffic**. Once the traffic is generated, Snort will start showing the  packets in verbosity mode

sudo snort -dev -K ASCII -l . (create packet ip in verbose mode)

**Reading generated logs with parameter "-r"**

Start the Snort instance in packet reader mode; sudo snort -r

sudo snort -r snort.log.1638459842

**"-r" parameter also allows users to filter the binary log files. You can filter the processed log to see specific packets with the "-r" parameter and Berkeley Packet Filters (BPF).**

* sudo snort -r logname.log -X
* sudo snort -r logname.log icmp
* sudo snort -r logname.log tcp
* sudo snort -r logname.log 'udp and port 53'

Investigate the traffic with the default configuration file **with ASCII mode.**

sudo snort -dev -K ASCII -l .

Execute the traffic generator script and choose **"TASK-6 Exercise"**. Wait until the traffic ends, then stop the Snort instance. Now analyse the output summary and answer the question.

sudo ./traffic-generator.sh

Now, you should have the logs in the current directory. Navigate to folder "**145.254.160.237**". What is the source port used to connect port 53?

53

Read the snort.log file with Snort; what is the IP ID of the 10th packet?

snort -r snort.log.1640048004 -n 10

**Let's run Snort in IDS/IPS Mode**

|  |  |
| --- | --- |
| **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. |

NIDS mode parameters are explained in the table below;

**IDS/IPS mode with parameter "-c and -T"**

Start the Snort instance and test the configuration file. sudo snort -c /etc/snort/snort.conf -T  This command will check your configuration file and prompt it if there is any misconfiguratioın in your current setting. You should be familiar with this command if you covered TASK3. If you don't remember the output of this command, **please revisit TASK4**.

**IDS/IPS mode with parameter "-N"**

Start the Snort instance and disable logging by running the following command: sudo snort -c /etc/snort/snort.conf -N

Now run the traffic-generator script **as sudo** and start **ICMP/HTTP traffic**. This command will disable logging mode. The rest of the other functions will still be available (if activated).

The command-line output will provide the information requested with the parameters. So, if you activate **verbosity (-v)** or **full packet dump (-X)** you will still have the output in the console, but there will be no logs in the log folder.

**IDS/IPS mode with parameter "-D"**

Start the Snort instance in background mode with the following command: sudo snort -c /etc/snort/snort.conf -D

**IDS/IPS mode with parameter "-A console"**

Console mode provides fast style alerts on the console screen. Start the Snort instance in **console alert mode (-A console ) with the following command**sudo snort -c /etc/snort/snort.conf -A console

Now run the traffic-generator script **as sudo** and start **ICMP/HTTP traffic**. Once the traffic is generated, snort will start generating alerts according to provided ruleset defined in the configuration file.

**IDS/IPS mode with parameter "-A cmg"**

Cmg mode provides basic header details with payload in hex and text format. Start the Snort instance in **cmg alert mode (-A cmg ) with the following command**sudo snort -c /etc/snort/snort.conf -A cmg

Now run the traffic-generator script **as sudo** and start **ICMP/HTTP traffic**. Once the traffic is generated, snort will start generating alerts according to provided ruleset defined in the configuration file.

**IDS/IPS mode with parameter "-A fast"**

Fast mode provides alert messages, timestamps, and source and destination IP addresses. **Remember, there is no console output in this mode.**Start the Snort instance in **fast alert mode (-A fast ) with the following command**sudo snort -c /etc/snort/snort.conf -A fast

Now run the traffic-generator script **as sudo** and start **ICMP/HTTP traffic**. Once the traffic is generated, snort will start generating alerts according to provided ruleset defined in the configuration file.

**IDS/IPS mode with parameter "-A full"**

Full alert mode provides all possible information about the alert. **Remember, there is no console output in this mode.**Start the Snort instance in **full alert mode (-A full ) with the following command**sudo snort -c /etc/snort/snort.conf -A full

Now run the traffic-generator script **as sudo** and start **ICMP/HTTP traffic**. Once the traffic is generated, snort will start generating alerts according to provided ruleset defined in the configuration file.

**IDS/IPS mode with parameter "-A none"**

Disable alerting. This mode doesn't create the alert file. However, it still logs the traffic and creates a log file in binary dump format. **Remember, there is no console output in this mode.**Start the Snort instance in **none alert mode (-A none) with the following command**sudo snort -c /etc/snort/snort.conf -A none

Now run the traffic-generator script **as sudo** and start **ICMP/HTTP traffic**. Once the traffic is generated, snort will start generating alerts according to provided ruleset defined in the configuration file.

**IPS mode and dropping packets**

Snort IPS mode activated with **-Q --daq afpacket** parameters. You can also activate this mode by editing snort.conf file. However, you don't need to edit snort.conf file in the scope of this room. Review the bonus task or snort manual for further information on daq and advanced configuration settings: -Q --daq afpacket

Activate the Data Acquisition (DAQ) modules and use the afpacket module to use snort as an IPS: -i eth0:eth1

Identifying interfaces note that Snort IPS require at least two interfaces to work. Now run the traffic-generator script **as sudo** and start **ICMP/HTTP traffic**.

Investigate the traffic with the default configuration file.

sudo snort -c /etc/snort/snort.conf -A full -l .

Execute the traffic generator script and choose **"TASK-7 Exercise"**. Wait until the traffic stops, then stop the Snort instance. Now analyse the output summary and answer the question.

sudo ./traffic-generator.sh

What is the number of the detected HTTP GET methods?

2

**Let's investigate PCAPs with Snort**

Capabilities of Snort are not limited to sniffing, logging and detecting/preventing the threats. PCAP read/investigate mode helps you work with pcap files. Once you have a pcap file and process it with Snort, you will receive default traffic statistics with alerts depending on your ruleset.

Reading a pcap without using any additional parameters we discussed before will only overview the packets and provide statistics about the file. In most cases, this is not very handy. We are investigating the pcap with Snort to benefit from the rules and speed up our investigation process by using the known patterns of threats.

**Note that** we are pretty close to starting to create rules. Therefore, you need to grasp the working mechanism of the Snort, learn the discussed parameters and begin combining the parameters for different purposes.

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. |

**Investigating single PCAP with parameter "-r"**

For test purposes, you can still test the default reading option with pcap by using the following command snort -r icmp-test.pcap

Let's investigate the pcap with our configuration file and see what will happen. sudo snort -c /etc/snort/snort.conf -q -r icmp-test.pcap -A console -n 10

**Investigating multiple PCAPs with parameter "--pcap-list"**

Let's investigate multiple pcaps with our configuration file and see what will happen. sudo snort -c /etc/snort/snort.conf -q --pcap-list="icmp-test.pcap http2.pcap" -A console -n 10

**Investigating multiple PCAPs with parameter "--pcap-show"**

Let's investigate multiple pcaps, distinguish each one, and see what will happen. sudo snort -c /etc/snort/snort.conf -q --pcap-list="icmp-test.pcap http2.pcap" -A console --pcap-show

Investigate the **mx-1.pcap** file with the default configuration file.

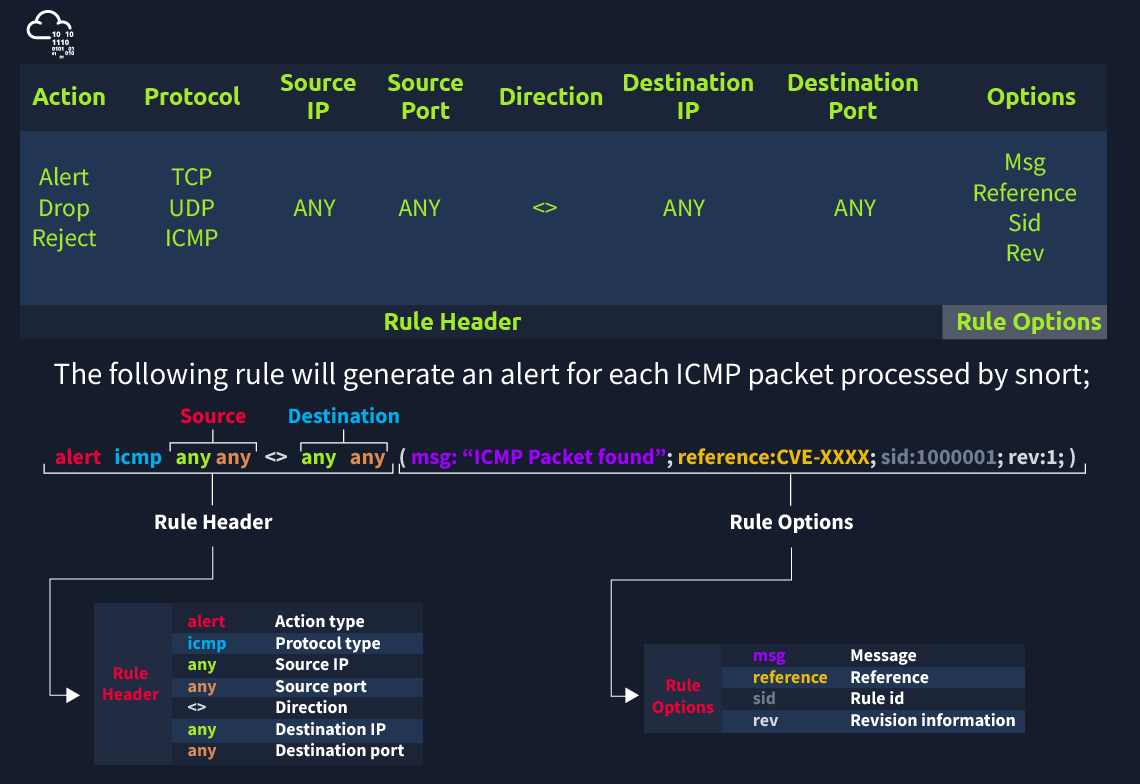
sudo snort -c /etc/snort/snort.conf -A full -l . -r mx-1.pcap

What is the number of the generated alerts?

170

**Let's Learn Snort Rules!**

Understanding the Snort rule format is essential for any blue and purple teamer.  The primary structure of the snort rule is shown below;



Each rule should have a type of action, protocol, source and destination IP, source and destination port and an option. Remember, Snort is in passive mode by default. So most of the time, you will use Snort as an IDS. You will need to start **"inline mode" to turn on IPS mode.** But before you start playing with inline mode, you should be familiar with Snort features and rules.

The Snort rule structure is easy to understand but difficult to produce. You should be familiar with rule options and related details to create efficient rules. It is recommended to practice Snort rules and option details for different use cases.

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 mode.

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**

These parameters identify the source and destination IP addresses and associated port numbers filtered for the rule.

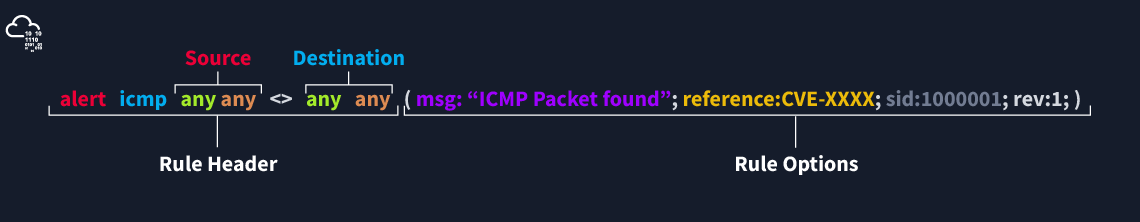
|  |  |
| --- | --- |
| IP Filtering | alert icmp 192.168.1.56 any <> any any  (msg: "ICMP Packet From "; sid: 100001; rev:1;)  This rule will create an alert for each ICMP packet originating from the 192.168.1.56 IP address. |
| Filter an IP range | alert icmp 192.168.1.0/24 any <> any any  (msg: "ICMP Packet Found"; sid: 100001; rev:1;)  This rule will create an alert 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 an alert 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 an alert for each ICMP packet not originating from the 192.168.1.0/24 subnet. |
| Port Filtering | alert tcp any any <> any 21  (msg: "FTP Port 21 Command Activity Detected"; sid: 100001; rev:1;)  This rule will create an alert for each TCP packet sent to port 21. |
| Exclude a specific port | alert tcp any any <> any !21  (msg: "Traffic Activity Without FTP Port 21 Command Channel"; sid: 100001; rev:1;)  This rule will create an alert for each TCP packet not sent to port 21. |
| Filter a port range (Type 1) | alert tcp any any <> any 1:1024   (msg: "TCP 1-1024 System Port Activity"; sid: 100001; rev:1;)  This rule will create an alert for each TCP packet sent to ports between 1-1024. |
| Filter a port range (Type 2) | alert tcp any any <> any :1024   (msg: "TCP 0-1024 System Port Activity"; sid: 100001; rev:1;)  This rule will create an alert for each TCP packet sent to ports less than or equal to 1024. |
| Filter a port range (Type 3) | alert tcp any any <> any 1025: (msg: "TCP Non-System Port Activity"; sid: 100001; rev:1;)  This rule will create an alert for each TCP packet sent to source port higher than or equal to 1025. |
| Filter a port range (Type 4) | alert tcp any any <> any [21,23] (msg: "FTP and Telnet Port 21-23 Activity Detected"; sid: 100001; rev:1;)  This rule will create an alert for each TCP packet sent to port 21 and 23. |

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.**



**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.

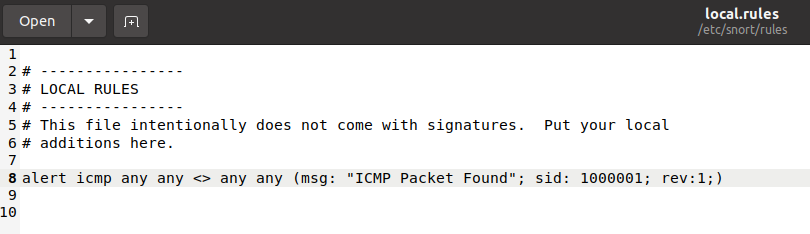
|  |  |
| --- | --- |
| ID | Filtering the IP id field.  alert tcp any any <> any any (msg: "ID TEST"; id:123456; sid: 100001; rev:1;) |
| Flags | Filtering the TCP flags.   * F - FIN * S - SYN * R - RST * P - PSH * A - ACK * U - URG   alert tcp any any <> any any (msg: "FLAG TEST"; flags:S;  sid: 100001; 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;) |

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". A quick reminder on how to edit your local rules is shown below.

modifying the local rules

user@ubuntu**$** sudo gedit /etc/snort/rules/local.rules

That is your "local.rules" file.



Note that there are some default rules activated with snort instance. These rules are deactivated to manage your rules and improve your exercise experience. For further information, please refer to the TASK-10 or [Snort manual](http://manual-snort-org.s3-website-us-east-1.amazonaws.com/).

By this point, we covered the primary structure of the Snort rules. Understanding and practicing the fundamentals is suggested before creating advanced rules and using additional options.

Wow! We have covered the fundamentals of the Snort rules!   Now, use the attached VM and navigate to the Task-Exercises/Exercise-Files/TASK-9 folder to answer the questions! Note that you can use the following command to create the logs in the c**urrent directory: -l .**