

## Practical: 1

Date: 14/12/2022

**Aim:** Perform 5 different types of (port) scanning using nmap on a single port and capture the packets using wireshark and analyze the output.

### Theory:

#### 1. Nmap:

- Nmap is a free and open-source network scanner created by Gordon Lyon. Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses.
- Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection.
- These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features.
- Nmap can adapt to network conditions including latency and congestion during a scan.
- Nmap started as a Linux utility and was ported to other systems including Windows, macOS, and BSD. It is most popular on Linux, followed by Windows.

#### 2. Wireshark:

- Wireshark is a free and open-source packet analyser.
- It is used for network troubleshooting, analysis, software and communications protocol development, and education.
- Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues.
- Wireshark is cross-platform, using the Qt widget toolkit in current releases to implement its user interface, and using pcap to capture packets; it runs on Linux, macOS, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows.
- There is also a terminal-based (non-GUI) version called TShark. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of version 2 of the GNU General Public License.

#### 3. open:

- An application is actively accepting TCP connections, UDP datagrams or SCTP associations on this port.
- Finding these is often the primary goal of port scanning. Security-minded people know that each open port is an avenue for attack.

- Attackers and pen-testers want to exploit the open ports, while administrators try to close or protect them with firewalls without thwarting legitimate users.
- Open ports are also interesting for non-security scans because they show services available for use on the network.

#### 4. closed

- A closed port is accessible (it receives and responds to Nmap probe packets), but there is no application listening on it.
- They can be helpful in showing that a host is up on an IP address (host discovery, or ping scanning), and as part of OS detection. Because closed ports are reachable, it may be worth scanning later in case some open up.
- Administrators may want to consider blocking such ports with a firewall. Then they would appear in the filtered state, discussed next.

#### 5. filtered

- Nmap cannot determine whether the port is open because packet filtering prevents its probes from reaching the port.
- The filtering could be from a dedicated firewall device, router rules, or host-based firewall software.
- These ports frustrate attackers because they provide so little information. Sometimes they respond with ICMP error messages such as type 3 code 13 (destination unreachable: communication administratively prohibited), but filters that simply drop probes without responding are far more common.
- This forces Nmap to retry several times just in case the probe was dropped due to network congestion rather than filtering. This slows down the scan dramatically.

#### 6. Unfiltered

- The unfiltered state means that a port is accessible, but Nmap is unable to determine whether it is open or closed.
- Only the ACK scan, which is used to map firewall rulesets, classifies ports into this state.
- Scanning unfiltered ports with other scan types such as Window scan, SYN scan, or FIN scan, may help resolve whether the port is open.

#### 7. open|filtered

- Nmap places ports in this state when it is unable to determine whether a port is open or filtered. This occurs for scan types in which open ports give no response.

- The lack of response could also mean that a packet filter dropped the probe or any response it elicited.
- So Nmap does not know for sure whether the port is open or being filtered. The UDP, IP protocol, FIN, NULL, and Xmas scans classify ports this way.

## 8. closed|filtered

- This state is used when Nmap is unable to determine whether a port is closed or filtered.
- It is only used for the IP ID idle scan.

### Implementation:

- There are two kinds of ports on each computer – TCP, and UDP – and 65,536 of each.
- The first 1024 TCP ports are the well-known ports like FTP(21), HTTP(80), or SSH(22).
- Anything above 1024 is available for use by services or applications.
- To scan Nmap ports on a remote system, enter the following in the terminal:

#### 1. TCP Scan :

TCP scan will scan for TCP port like port 22, 21, 23, 445 etc and ensure for listening port through 3-way handshake connection between the source and destination port. If the port is open then source made request with SYN packet, a response destination sent SYN, ACK packet and then source sent ACK packets, at last source again sent RST, ACK packets.

Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
(kali㉿kali)-[~]
└─$ nmap -sT -p 445 192.168.17.206
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-03 04:39 EST
Nmap scan report for 192.168.17.206
Host is up (0.0016s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 2.91 seconds
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.11.128	192.168.17.206	TCP	74	42666 → 80 [SYN] Seq=0 Win=1460 SACK_PERM TS...
2	0.000229579	192.168.11.128	192.168.17.206	TCP	74	37628 → 443 [SYN] Seq=0 Win=1460 SACK_PERM TS...
3	0.001279264	192.168.17.206	192.168.11.128	TCP	60	80 → 42666 [SYN, ACK] Seq=0 Ack=1 Win=1460 SACK...
4	0.001348657	192.168.11.128	192.168.17.206	TCP	54	42666 → 80 [ACK] Seq=1 Ack=1 Win=1460 SACK_PERM T...
5	0.001514041	192.168.11.128	192.168.17.206	TCP	54	42666 → 80 [RST, ACK] Seq=1 Ack=1 Win=1460 SACK_P...
6	0.002460665	192.168.11.128	192.168.11.2	DNS	87	Standard query 0x5969 PTR 206.17.168.192.in-addr.ap...
7	0.035134365	192.168.17.206	192.168.11.128	TCP	60	443 → 37628 [RST, ACK] Seq=1 Ack=1 Win=1460 SACK_P...
8	2.819732117	192.168.11.2	192.168.11.128	DNS	87	Standard query response 0x5969 No such name PTR 206.17...
9	2.820251011	192.168.11.128	192.168.17.206	TCP	74	40594 → 445 [SYN] Seq=0 Win=1460 SACK_PERM TS...
10	2.821826307	192.168.17.206	192.168.11.128	TCP	60	445 → 40594 [SYN, ACK] Seq=0 Ack=1 Win=1460 SACK_P...

## 2. Stealth Scan:

SYN scan is the default and most popular scan option for good reasons. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by restrictive firewalls. It is also relatively typical and stealthy since it never completes TCP connections.

```
(kali㉿kali)-[~]
└─$ sudo nmap -sS -p 22 192.168.17.206
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-03 05:05 EST
Nmap scan report for 192.168.17.206
Host is up (0.0012s latency).

PORT      STATE      SERVICE
22/tcp     filtered  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds

```

No.	Source	Destination	Protocol	Length	Info
00000	192.168.11.128	192.168.17.206	TCP	74	42666 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TS...
29579	192.168.11.128	192.168.17.206	TCP	74	37628 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TS...
79264	192.168.17.206	192.168.11.128	TCP	60	80 → 42666 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK...
848657	192.168.11.128	192.168.17.206	TCP	54	42666 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0 MSS=1460 SACK_P...
14941	192.168.11.128	192.168.17.206	TCP	54	42666 → 80 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0 MSS=1460 SACK...
160665	192.168.11.128	192.168.11.2	DNS	87	Standard query 0x5969 PTR 206.17.168.192.in-addr.ap...
34365	192.168.17.206	192.168.11.128	TCP	60	443 → 37628 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0 MSS=1460 SACK_P...
32117	192.168.11.2	192.168.11.128	DNS	87	Standard query response 0x5969 No such name PTR 206.17.168.19...
51011	192.168.11.128	192.168.17.206	TCP	74	40594 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TS...
126307	192.168.17.206	192.168.11.128	TCP	60	445 → 40594 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_P...

### 3. Fin Scan:

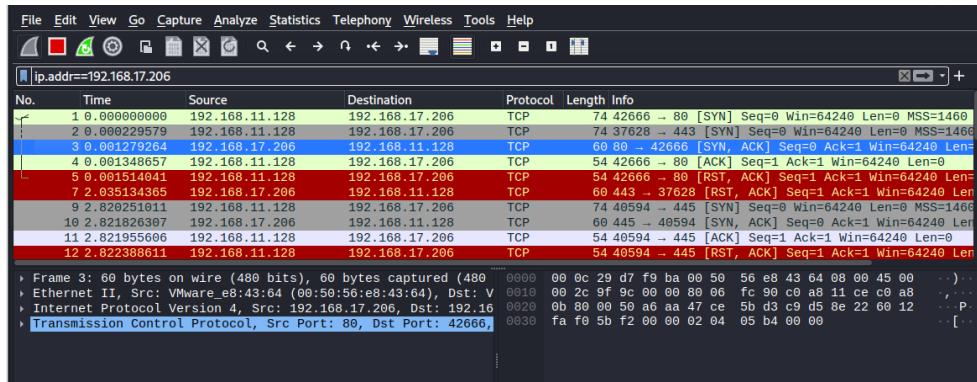
FIN packet is used to terminate the TCP connection between the source and destination port typically after the data transfer is complete. In the place of an SYN packet, Nmap starts a FIN scan by using a FIN packet. If the port is open then no response will come from destination port when FIN packet is sent through source port.

Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
(kali㉿kali)-[~]
└─$ sudo nmap -sF -p 22 192.168.17.206
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-03 05:09 EST 42666
Nmap scan report for 192.168.17.206
Host is up (0.00094s latency).

PORT      STATE      SERVICE
22/tcp    open|filtered ssh

Nmap done: 1 IP address (1 host up) scanned in 0.67 seconds
```



### 4. Null Scan:

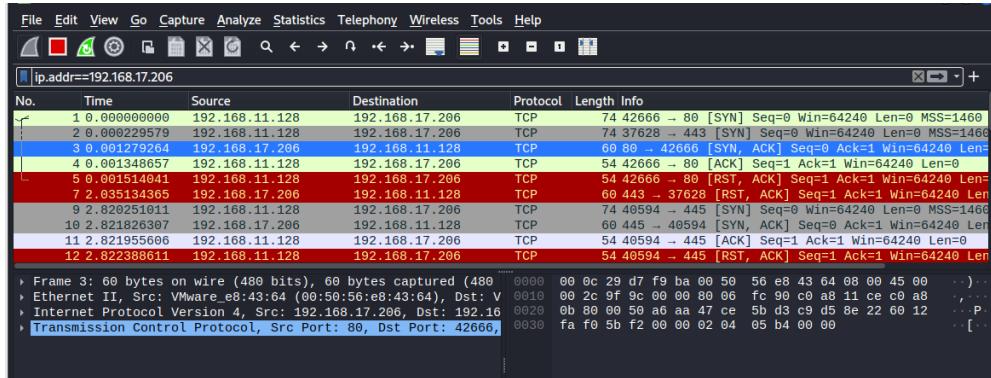
Null Scan is a series of TCP packets which hold a sequence number of “zeros” (00000000) and since there are none flags set, the destination will not know how to reply the request. It will discard the packet and no reply will be sent, which indicate that the port is open.

Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
(kali㉿kali)-[~]
└─$ sudo nmap -sN -p 22 192.168.17.206
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-03 05:07 EST
Nmap scan report for 192.168.17.206
Host is up (0.0011s latency).

PORT      STATE      SERVICE
22/tcp    open|filtered  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.46 seconds
```



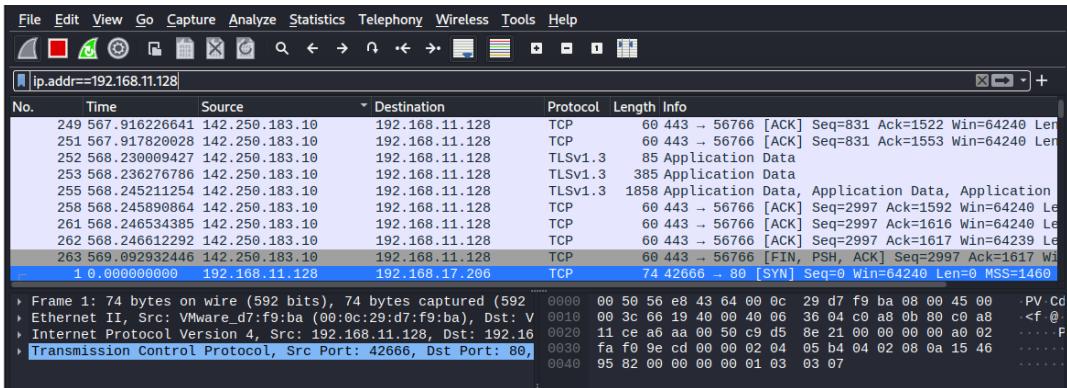
## 5. UDP Scan :

UDP scan works by sending a UDP packet to every destination port; it is a connectionless protocol. For some common ports such as 53 and 161, a protocol-specific payload is sent to increase the response rate, a service will respond with a UDP packet, proving that it is open. If no response is received after retransmissions, the port is classified as open|filtered. This means that the port could be open, or perhaps packet filters are blocking the communication.

```
(kali㉿kali)-[~]
└─$ sudo nmap -sU -p 161 192.168.11.128
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-03 05:13 EST
Nmap scan report for 192.168.11.128
Host is up (0.00022s latency).

PORT      STATE      SERVICE
161/udp   closed    snmp

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```



### Conclusion/Summary:

Student Signature & Date	Marks	Evaluator Signature & Date
--------------------------	-------	----------------------------

## Practical 2

Date: 21/12/2022

**Aim:** Perform a Vulnerability Scan on a system within the Local Area Network and Submit the report.

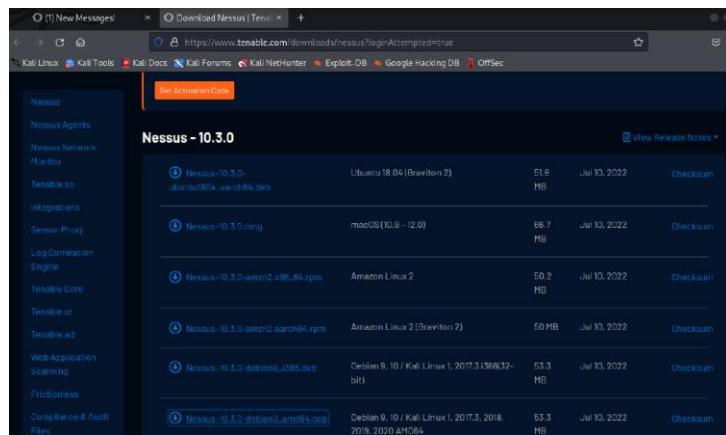
### Theory:

#### Nessus Essentials:

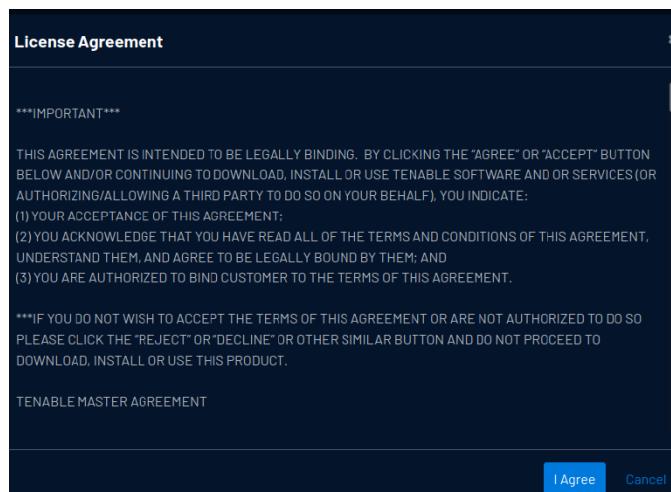
- Nessus Essentials is a free vulnerability assessment solution for up to 16 IPs that provides an entry point into the Tenable ecosystem.
- Backed by market leading functionality from Nessus Professional, Nessus Essentials gives you the accuracy and speed you need to discover, prioritize and remediate vulnerabilities.

#### Implementation:

- Firstly, Nessus Essential is not pre-installed. Hence, we need to download it.



- Accept the Agreement



- Now, unpack the package

```
(kali㉿kali)-[~/Downloads]
└─$ ls
Nessus-10.3.0-debian9_amd64.deb

(kali㉿kali)-[~/Downloads]
└─$ sudo dpkg -i Nessus-10.3.0-debian9_amd64.deb

Selecting previously unselected package nessus.
(Reading database ... 298561 files and directories currently installed.)
Preparing to unpack Nessus-10.3.0-debian9_amd64.deb ...
Unpacking nessus (10.3.0) ...
Setting up nessus (10.3.0) ...
Unpacking Nessus Scanner Core Components ...

- You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
- Then go to https://kali:8834/ to configure your scanner
```

- Now, enter the following commands

```
(kali㉿kali)-[~/Downloads]
└─$ ls
Nessus-10.3.0-debian9_amd64.deb

(kali㉿kali)-[~/Downloads]
└─$ sudo dpkg -i Nessus-10.3.0-debian9_amd64.deb
[sudo] password for kali:
(Reading database ... 298810 files and directories currently installed.)
Preparing to unpack Nessus-10.3.0-debian9_amd64.deb ...
Unpacking nessus (10.3.0) over (10.3.0) ...
Setting up nessus (10.3.0) ...
Unpacking Nessus Scanner Core Components ...

- You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
- Then go to https://kali:8834/ to configure your scanner

[21:52:23]
(kali㉿kali)-[~/Downloads]
└─$ sudo systemctl start nessusd.service
```

- Check the status

```
(kali㉿kali)-[~/Downloads]
└─$ sudo systemctl status nessusd.service
● nessusd.service - The Nessus Vulnerability Scanner [21:23]
   Loaded: loaded (/lib/systemd/system/nessusd.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2022-07-27 13:51:48 EDT; 12s ago
     Main PID: 22007 (nessus-service)
        Tasks: 13 (limit: 2264)
       Memory: 127.7M
          CPU: 10.874s
        CGroup: /system.slice/nessusd.service
                  └─22007 /opt/nessus/sbin/nessus-service -q

Jul 27 13:51:48 kali systemd[1]: Started The Nessus Vulnerability Scanner.
Jul 27 13:51:49 kali nessus-service[22009]: Cached 0 plugin libs in 0msec
Jul 27 13:51:49 kali nessus-service[22009]: Cached 0 plugin libs in 0msec
```

- Go to the link provided and proceed further. (<https://kali:8834>)

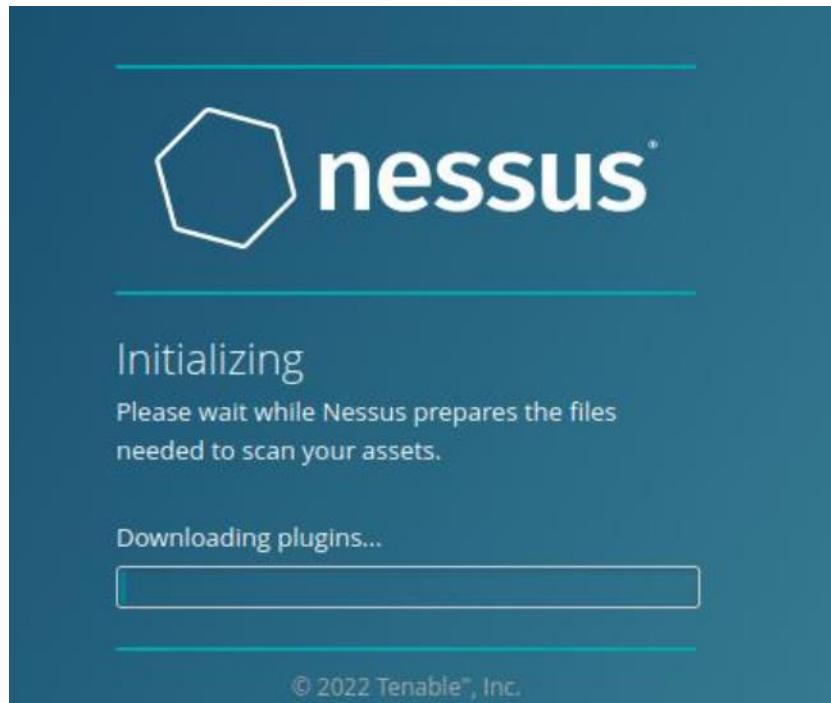
- The installation page will arrive



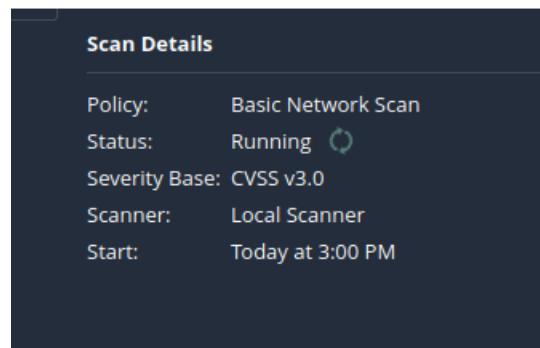
- Apply the activation code.



- After providing the user\_name and password, download process will begin.



- Once, all plugins are installed, it will prompt you to enter the details of hosts that you want to check for.
- After that, it will start the scanning.



- Once completed, following information will be shown.

The screenshot shows the Tenable.io interface for a scan named 'try'. On the left, there's a sidebar with 'Folders' (My Scans, All Scans, Trash), 'Resources' (Policies, Plugin Rules, Terrascan), and 'Tenable News' (NETGEAR Nighthawk WiFi6 Router Network Misconfigur...). The main area displays a table of vulnerabilities with columns: Sev, Score, Name, Family, and Count. A search bar at the top right says '10 Vulnerabilities'. To the right, there's a 'Scan Details' panel showing the policy is 'Basic Network Scan', status is 'Running', severity base is 'CVSS v3.0', scanner is 'Local Scanner', and start time is 'Today at 3:27 AM'. Below that is a 'Vulnerabilities' section with a pie chart showing the distribution of severity levels: Critical (red), High (orange), Medium (yellow), Low (light blue), and Info (blue).

Sev	Score	Name	Family	Count
MIXED	...	SSL (Multiple Issues)	General	6
INFO	...	HTTP (Multiple Issues)	Web Servers	3
INFO	...	TLS (Multiple Issues)	Service detection	2
INFO		Service Detection	Service detection	3
INFO		Nessus SYN scanner	Port scanners	2
INFO		Web Server No 404 Error Cod...	Web Servers	2
INFO		Host Fully Qualified Domain ...	General	1

This screenshot shows a detailed list of vulnerabilities from the previous scan. The table has columns: Sev, Score, Name, Family, Count, and actions (edit, delete). The data includes various findings like SMB Signing not required, Microsoft Windows (Multiple Issues), and Nessus SYN scanner.

Sev	Score	Name	Family	Count	Action
MEDIUM	5.3	SMB Signing not required	Misc.	1	
INFO	...	SMB (Multiple Issues)	Windows	6	
INFO	...	Microsoft Windows (Multiple Issues)	Windows	2	
INFO		DCE Services Enumeration	Windows	8	
INFO		Nessus SYN scanner	Port scanners	6	
INFO		Service Detection	Service detection	2	
INFO		VMware ESX/GSX Server detection	Service detection	2	
INFO		Common Platform Enumeration (CPE)	General	1	
INFO		Device Type	General	1	
INFO		Nessus Scan Information	Settings	1	
INFO		OS Identification	General	1	

This screenshot shows the 'VPR Top Threats' section. It displays a large green checkmark icon and the text 'Assessed Threat Level: None'. Below that, it states 'No vulnerabilities have been found as prioritized by Tenable's patented Vulnerability Priority Rating (VPR) system.' and provides a link to 'Predictive Prioritization'. At the bottom, it says 'No prioritized vulnerabilities found.'

**Conclusion/Summary:**

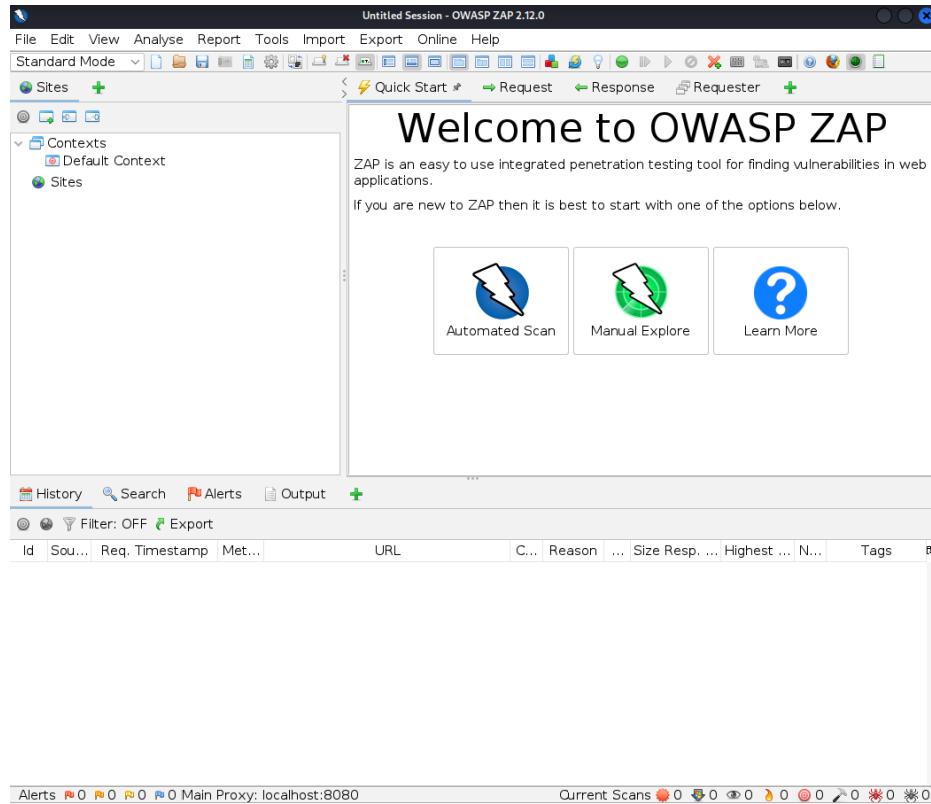
<b>Student Signature &amp; Date</b>	<b>Marks</b>	<b>Evaluator Signature &amp; Date</b>
-------------------------------------	--------------	---------------------------------------

## Practical: 3

Date: 28/12/2022
<b>Aim:</b> Implementation to identify web vulnerabilities, using OWASP project
<b>Theory:</b> <ul style="list-style-type: none"><li>• OWASP stands for “Open Web Application Security Project”.</li><li>• It is an open, online community that creates methodologies, tools, technologies and guidance on how to deliver secure web applications.</li><li>• OWASP ZAP (ZAP) is one of the world’s most popular free security tools and is actively maintained by hundreds of international volunteers. It can help to find security vulnerabilities in web applications. It’s also a great tool for experienced pen testers and beginners.</li><li>• ZAP is what is known as a “man-in-the-middle proxy.” It stands between the browser and the web application. While you navigate through all the features of the website, it captures all actions. Then it attacks the website with known techniques to find security vulnerabilities.</li><li>• It is one of the most active Open Web Application Security Project (OWASP) projects and has been given Flagship status.</li><li>• When used as a proxy server it allows the user to manipulate all of the traffic that passes through it, including traffic using https.</li><li>• It can also run in a daemon mode which is then controlled via a REST API.</li><li>• ZAP was added to the ThoughtWorks Technology Radar in May 2015 in the Trial ring.</li><li>• ZAP was originally forked from Paros, another pentesting proxy. Simon Bennetts, the project lead, stated in 2014 that only 20% of ZAP’s source code was still from Paros.</li></ul>

## Implementation:

- Starting ZAP.
  - Once setup you can start ZAP by clicking the ZAP icon on your Windows desktop or from the start menu.

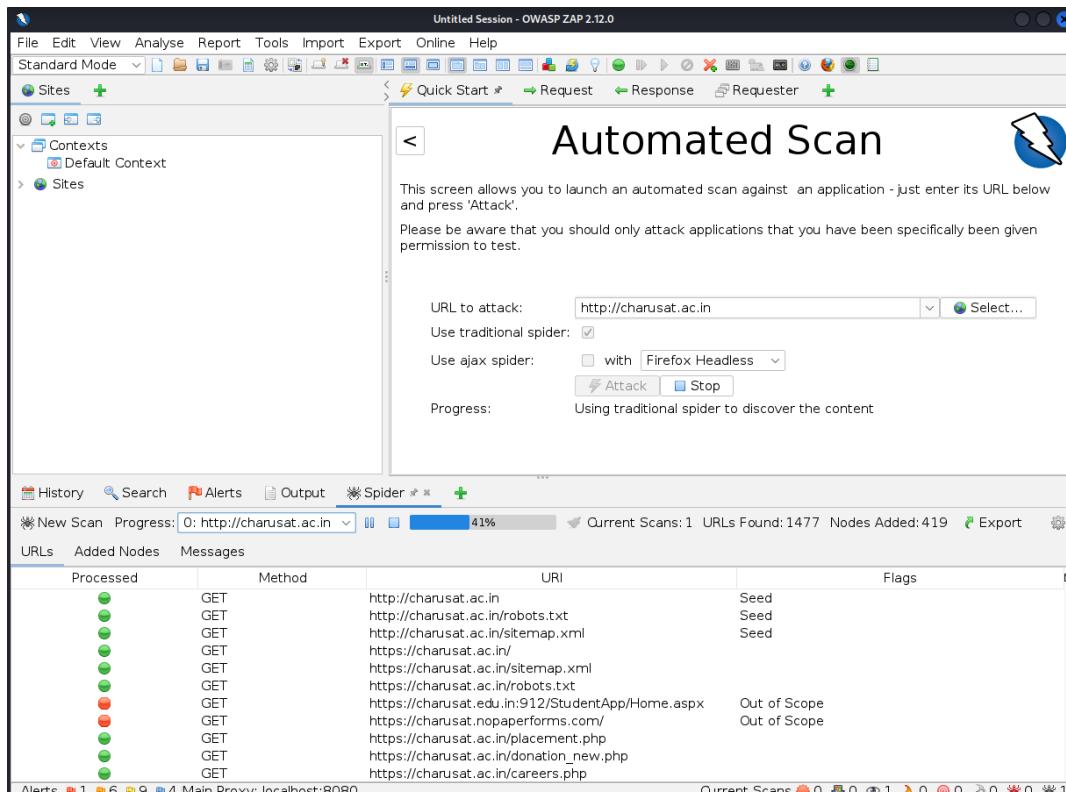


## Glimpses Of ZAP Console

- Spidering the web application
  - Spidering a web application means crawling all the links and getting the structure of the application. ZAP provides two spiders for crawling web applications;
  - The traditional ZAP spider discovers links by examining the HTML in responses from the web application. This spider is fast, but it is not always effective when exploring an AJAX web application.
  - This is more likely to be effective for AJAX applications. This spider explores the web application by invoking browsers which then follow the links that have been generated. The AJAX spider is slower than the traditional spider.

### Automated Scan:

- This option allows you to launch an automated scan against an application just by entering the URL. If you are new to ZAP, it is best to start with Automated Scan mode.
- To run a Quick Start Automated Scan:
  1. Start Zap and click the large ‘Automated Scan’ button in the ‘Quick Start’ tab.
  2. Enter the full URL of the web application you want to attack in the ‘URL to attack’ text box.
  3. Click the ‘Attack’ button.

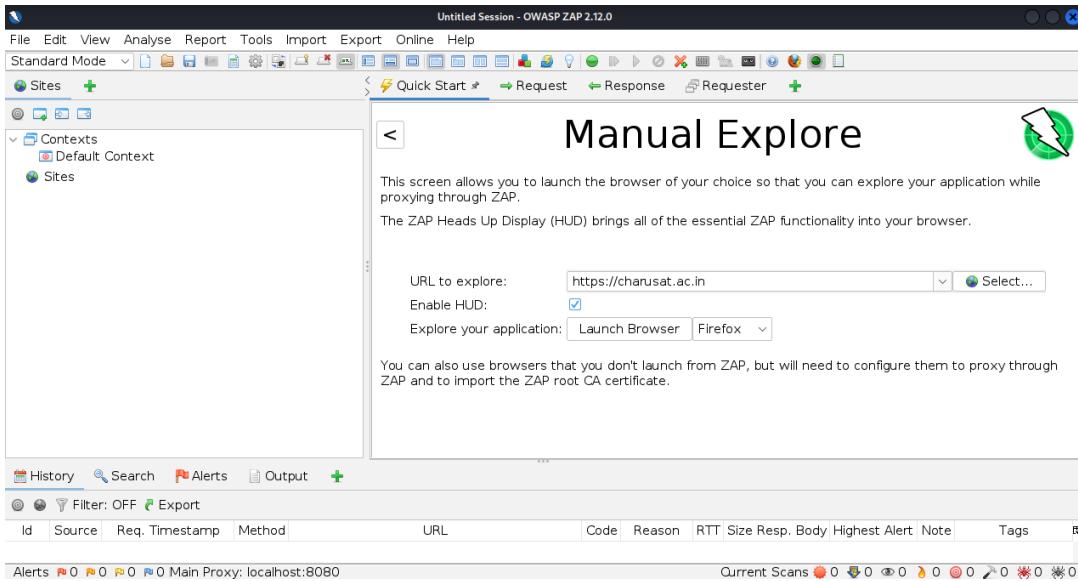


Crawling started

- Once you click the ‘Attack’ button, ZAP will start crawling the web application with its spider and passively scan each page it finds. Then ZAP will use the active scanner to attack all of the discovered pages, functionality and parameters.
- Exploring the web application manually
- Spiders are a great way to explore the basic site, but they should be combined with manual exploration to be more effective. This functionality is very useful when your web application needs a login or contains things like registration forms, etc.

- You can launch browsers that are pre-configured to proxy through ZAP via the Quick Start tab. Browsers launched in this way will also ignore any certificate validation warnings that would otherwise be reported.

### Manual Explore:



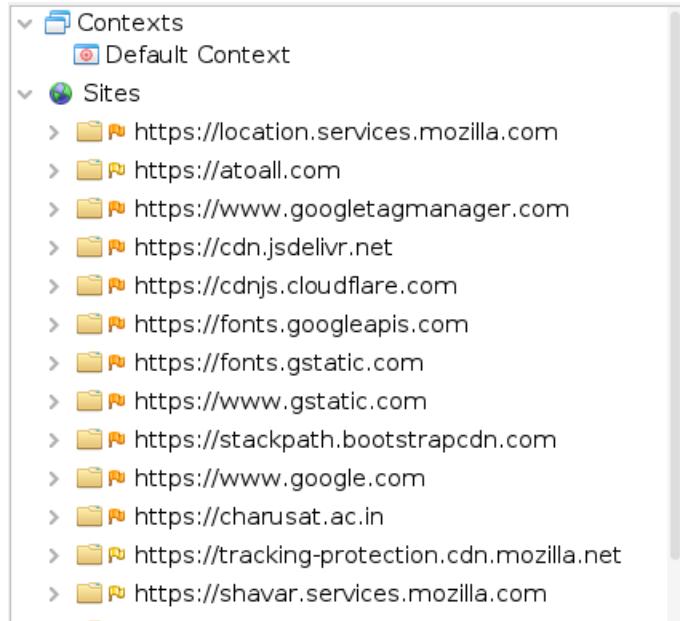
- To Manually Explore the web application.
- Start ZAP and click on the large “Manual Explore” button in the Quick Start tab.
- Enter the full URL of the web application to be explored in the ‘URL to explore’ text box.
- Select the browser you would like to use and click the ‘Launch Browser’ button.
- This will launch the selected browser with a new profile. Now explore all of the targeted web applications through this browser.
- ZAP passively scans all the requests and responses made during your exploration for vulnerabilities, continues to build the site tree, and records alerts for potential vulnerabilities found during the exploration.

### What is passive scanning?

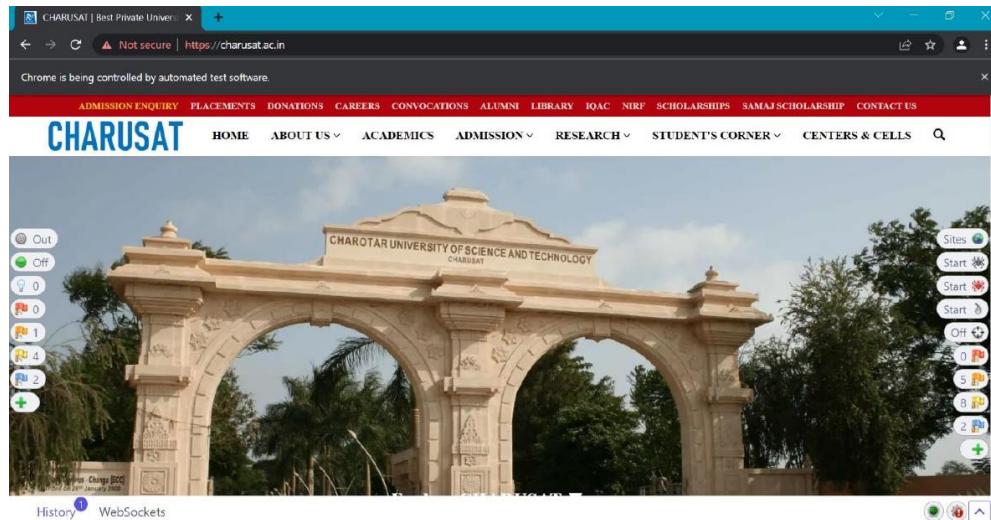
- Passive Scans only scan the web application responses without altering them.
- It does not attack or insert malicious scripts to the web application, so this is a safe scan; you can use it if you are new to security testing.
- Passive scanning is good at finding some vulnerabilities and as a way to get a feel for the basic security of a web application.

## What is active scanning?

- Active scan attacks the web application using known techniques to find vulnerabilities. This is a real attack that attempts to modify data and insert malicious scripts in the web application.



### Sites Updates



Manual scanning initiated

- We will not proceed as we are not allowed to scan the website.

**Conclusion/Summary:**

<b>Student Signature &amp; Date</b>	<b>Marks</b>	<b>Evaluator Signature &amp; Date</b>
-------------------------------------	--------------	---------------------------------------

## Practical 4

Date: / /2023

**Aim:** Perform log analysis of machine data using Splunk software in windows/Linux. This machine data can come from web applications, sensors, devices, or any data created by the user.

### Theory:

#### Splunk

Splunk (the product) captures, indexes, and correlates real-time data in a searchable repository from which it can generate graphs, reports, alerts, dashboards, and visualizations.

#### Features offered by Splunk Enterprise: -

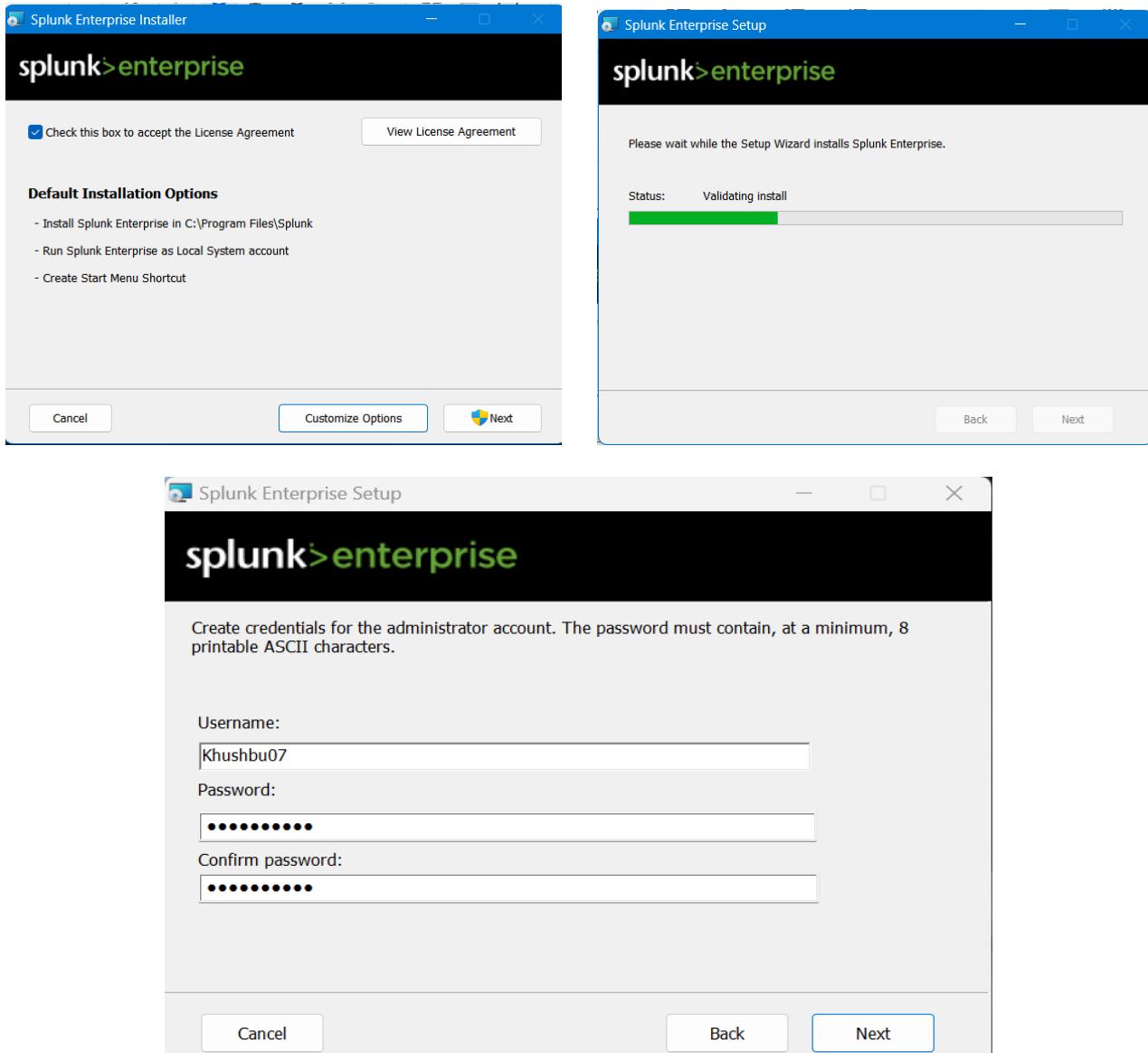
1. Data searching: – Searching in Splunk involves the pattern of creating metrics or indexes on Dashboards.
2. Data ingestion: – Splunk ingest data in various formats like XML, JSON, and unstructured machine data such as logs of CPU running on web servers.
3. Data Indexing: – Splunk auto index the ingested data of various machines for the faster searching on various conditions
4. Alerts: – Splunk alert used for triggering emails or other feeds when some unusual suspicious activity found in data is being analyzed.
5. Dashboards: – It shows the search results in the form of pivots, area mapping, pie charts, reports, etc.

#### Uses of Splunk: -

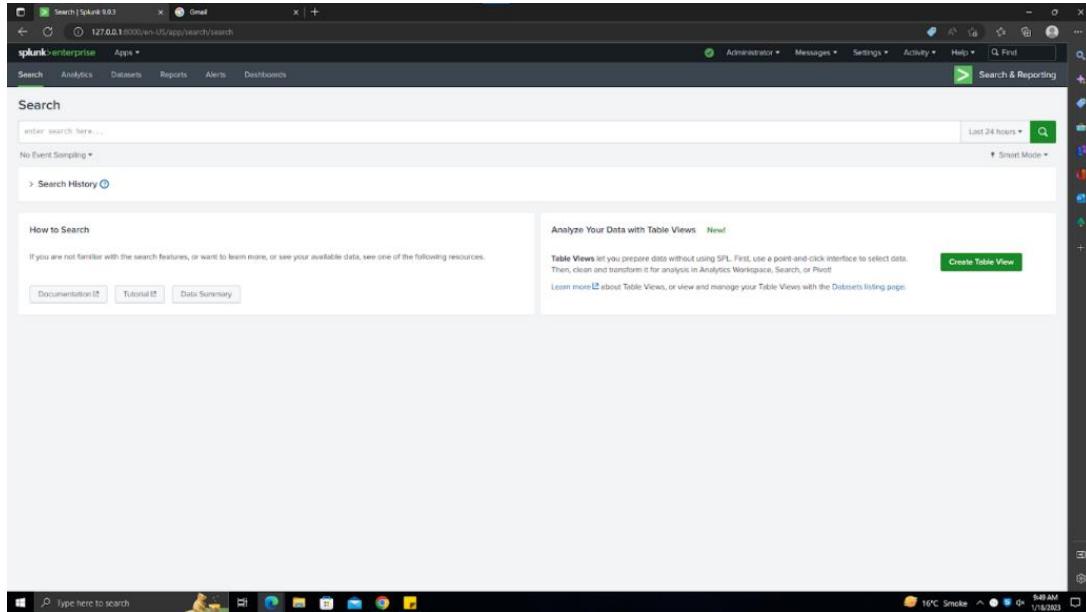
- Splunk is used for monitoring and searching through big data.
- It indexes and correlates information in a container that makes it searchable, and makes it possible to generate alerts, reports and visualizations.
- In the context of security, Splunk is essentially used as a log analysis engine. “It is used to correlate security events, which allows you to identify where your breaches are coming from,”

## Implementation:

### Installation: -



## Step 1: Launch the Splunk Enterprise Application



## Step 2: Searching a Windows event

Type the following command in the filter or search section

`source='WinEventLog:Application'`

The screenshot shows the search results for the query `source='WinEventLog:Application'`. The results table has columns for Time, Event, and host. There are 111 events listed. The first few events are as follows:

Time	Event	host
1/18/23 9:48:27 AM	LogName=Application EventCode=1034 EventType=4 ComputerName=LAD3-8 Show all 12 lines	LAD3-8
1/18/23 9:48:27.000 AM	LogName=Application EventCode=1034 EventType=4 ComputerName=LAD3-8 Show all 12 lines	LAD3-8
1/18/23 9:48:27.500 AM	LogName=Application EventCode=1034 EventType=4 ComputerName=LAD3-8 Show all 12 lines	LAD3-8
1/18/23 9:48:28.000 AM	LogName=Application EventCode=1034 EventType=4 ComputerName=LAD3-8 Show all 12 lines	LAD3-8
1/18/23 9:48:28.500 AM	LogName=Application EventCode=1034 EventType=4 ComputerName=LAD3-8 Show all 12 lines	LAD3-8

### Step 3: Now expand any one Windows Event and observe the details

The screenshot shows a Splunk search results page. A single event is selected, providing a detailed breakdown of its fields. Key fields include:

- Type:** Event
- host:** LAB3-B
- source:** WinEventLog Application
- sourceType:** WinEventLog Application
- EventCode:** 10707
- EventID:** 4
- Keywords:** Classic
- LogName:** Application
- Message:** Product: Splunk Enterprise -- Installation completed successfully.
- OpCode:** Info
- RecordNumber:** 14320
- Sid:** S-1-5-21-3909406397-3429262321-204087700-500
- SitType:** 0
- SourceName:** MsiInstaller

### Step 4: Searching for any kind of Failure/error in any Windows Event

Type the following command in the filter or search section

source='WinEventLog:Application' type='error'

The screenshot shows a Splunk search results page for errors in the WinEventLog:Application source. Two events are listed:

- Event 1:** Host: LAB3-B, Source: WinEventLog Application, Type: Error. Message: "Message=Failed to create restore point (Process = C:\Windows\system32\msiexec.exe /V; Description = Installed Splunk Enterprise; Error = 0x80070422)."
- Event 2:** Host: LAB3-B, Source: WinEventLog Application, Type: Error. Message: "Message=Failed to create restore point (Process = C:\Windows\system32\msiexec.exe /V; Description = Installed Splunk Enterprise; Error = 0x80070422)."

## Step 5: Expand any one event of your choice and observe the details

The screenshot shows the Splunk interface with an event expanded. The expanded event details are as follows:

- Type:** host = LAB3-B
- source:** WINEVENTLOG.APPLICATION
- sourcetype:** WINEVENTLOG.APPLICATION
- Event:**
  - ComputerName = LAB3-B
  - EventCode = 8193
  - EventType = 2
  - Keywords = Classic
  - LogName = Application
  - Message = Failed to create restore point (Process = C:\Windows\system32\msinexec.exe !V; Description = Installed Splunk Enterprise; Error = 0x80070422).
  - OpCode = Info
  - RecordNumber = 14317
  - SourceName = System Restore
  - TaskCategory = None
  - Type = Error
- Time:** 2023-01-18T09:34:31.000Z
- Default:** index = main
- Imeasure:** 12
- punct:** { \_raw \_index \_score \_type }

## Step 6: Creating an Index

Hover on setting then under Data column click on Indexes

The screenshot shows the Splunk dashboard with the 'Settings' menu open. Under the 'DATA' section, the 'Indexes' option is highlighted.

Now name the index as per your wish

The screenshot shows the 'New Index' configuration dialog. The 'General Settings' tab is selected, displaying the following configuration:

- Index Name:** 202304
- Index Data Type:** Events
- Home Path:** \$SPLUNK\_DB/confgreker
- Cold Path:** \$SPLUNK\_DB/confgreker/cold
- Thawed Path:** \$SPLUNK\_DB/confgreker/thawed
- Data Integrity Check:** Enable
- Max Size of Entire Index:** 500 GB
- Max Size of Hot/Warm/Cold Bucket:** auto
- Frozen Path:** \$SPLUNK\_DB/confgreker/frozen
- App:** Search & Reporting
- Storage Optimization:** Tard Retention Policy, Enable Reduction, Disable Reduction

### Step 2: In Data Inputs select Files & Directories option.

The screenshot shows the 'Data inputs' page in Splunk Enterprise. The 'Local inputs' section is displayed, listing five types of inputs:

Type	Inputs	Actions
Local event log collection Collect event logs from this machine.	-	Edit
Remote event log collections Collect event logs from remote hosts. Note: this uses WMI and requires a domain account.	1	+ Add new
Files & Directories Index a local file or monitor an entire directory.	13	+ Add new
Local performance monitoring Collect performance data from local machine.	0	+ Add new
Remote performance monitoring Collect performance and event information from remote hosts. Requires domain credentials.	0	+ Add new

The screenshot shows the 'Files & directories' page in Splunk Enterprise. It displays a list of 13 configured items, each with details like full path, source type, index, number of files, app, and status. A green button labeled 'New Local File & Directory' is visible at the top right.

Full path to your data	Set host	Source type	Index	Number of files	App	Status	Actions
\$SPLUNK_HOME/var/log/python_upgrade_readiness_app	Constant Value	python_upgrade_readiness_app	_internal	1	python_upgrade_readiness_app	Enabled   Disable	
\$SPLUNK_HOME/etc/splunk/version	Constant Value	splunk_version	_internal	1	system	Enabled   Disable	
\$SPLUNK_HOME/var/log/introspection	Constant Value	Automatic	_introspection	4	introspection_generator_addon	Enabled   Disable	
\$SPLUNK_HOME/var/log/splunk	Constant Value	Automatic	_internal	75	system	Enabled   Disable	
\$SPLUNK_HOME/var/log/splunk/configuration_change.log	Constant Value	Automatic	_configtracker	1	system	Enabled   Disable	
\$SPLUNK_HOME/var/log/splunk/license_usage_summary.log	Constant Value	Automatic	_telemetry	1	system	Enabled   Disable	

### Step 3: Select the Source as shown below.

The screenshot shows the 'Add Data' wizard in Splunk Enterprise, currently on the 'Select Source' step. The 'Files & Directories' option is selected. On the right, configuration options are shown for monitoring a file named 'SystemRestore' located at 'C:\Windows\Logs\SystemRestore'. Fields for 'Includelist' and 'Excludelist' are also present.

#### Step 4: Select the Input Settings as shown below.

The screenshot shows the 'Input Settings' step of the 'Add Data' wizard in Splunk Enterprise. The 'Source type' is set to 'WindowsRestore\_1'. The 'Source Type Category' is set to 'Custom'. The 'Source Type Description' field is empty. The 'App context' is set to 'Search & Reporting (search)'. The top navigation bar shows 'Administrator', 'Messages', 'Settings', 'Activity', 'Help', and a search bar.

#### Step 5: Click on Submit Button.

The screenshot shows a success message: 'File input has been created successfully.' It includes a link to 'Configure your inputs by going to Settings > Data Inputs'. Below this, there are five buttons: 'Start Searching', 'Extract Fields', 'Add More Data', 'Download Apps', and 'Build Dashboards', each with a corresponding description and link.

#### Forward Data

The screenshot shows the 'Forward data' configuration page. It lists a single host entry: 'localhost:9997' with 'Automatic Load Balancing' set to 'Enabled'. The status is 'Enabled'. There are 'Clone' and 'Delete' actions available. At the top right, there is a green button labeled 'New Forwarding Host'.

## Receive Data

The screenshot shows the Splunk Enterprise interface with the title 'Receive Data'. A message at the top says 'Successfully saved "9997".' Below is a table with one row:

Listen on this port	Status	Actions
9997	Enabled   Disable	Delete

## Step 7: Applying Search on the data by writing:

index="audit" | stats count by source, sourcetype.

The screenshot shows the Splunk Enterprise interface with the title 'New Search'. The search bar contains the query 'index=\_audit' | stats count by source, sourcetype. The results table shows:

source	sourcetype	count
audittrail	audittrail	10165

## Step 8: Visualization of logs of the selected file.

The screenshot shows the Splunk Enterprise interface with the title 'New Search'. The search bar contains the same query as before. The visualization section is active, showing a column chart titled 'audittrail' with one data point:

source	count
audittrail	10165

<b>Conclusion/Summary:</b>		
<b>Student Signature &amp; Date</b>	<b>Marks:</b>	<b>Evaluator Signature &amp; Date</b>

## Practical 5

<b>Date:</b> / /2023
----------------------

<b>Aim:</b> Monitor the traffic in real time and issue alerts to users when it discovers potentially malicious packets or threats on Internet Protocol network using SNORT.
---

### **Theory:**

SNORT is an open-source, rule-based Network Intrusion Detection and Prevention System. It was developed and still maintained by Martin Rocher, open-source contributors, and the CISCO talos team.

Snort is the foremost open-source Intrusion Prevention System in the world. Snort IPS uses a series of rules that helps define malicious network activity and uses those rules to find packets that match against them and generate alerts for users.

### **Intrusion Detection System (IDS)**

IDS is a passive monitoring solution for detecting possible malicious activity/patterns, abnormal incidents and policy violations. It is responsible for generating alerts for each suspicious event.

There are two main types of IDS systems:

- **Network-based IDS (NIDS)** – NIDS monitors the traffic flow from various areas of the network. The main aim is to investigate the traffic on the entire subnet. If a signature is identified, an alert is created.
- **Host-based IDS (HIDS)** – HIDS monitors the traffic flow from a single endpoint device. The aim is to investigate the traffic on a particular device. If a signature is identified, an alert is created.

### **Intrusion Prevention System (IPS)**

IPS is an active protecting solution for preventing possible malicious activity/patterns, abnormal incidents and policy violations. It is responsible for stopping/preventing/terminating the suspicious event as soon as the detection is performed.

There are four main types of IPS systems:

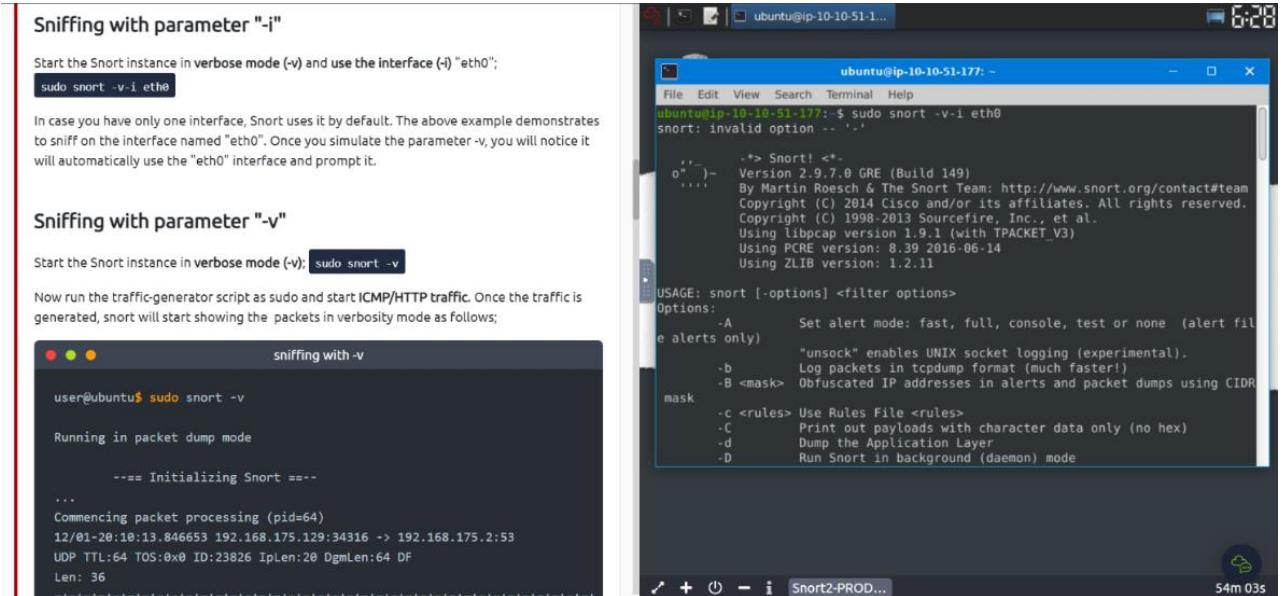
- **Network IPS (NIPS)** – NIPS monitors the traffic flow from various areas of the network. The aim is to protect the traffic on the entire subnet. If a signature is identified, the connection is terminated.
- **Network Behaviour-based IPS (NBA)** – It monitors the traffic flow from various area of the network. The aim is to protect the traffic on the entire subnet. If a signature is identified, the connection is terminated.

- **Wireless IPS (WIPS)** – It monitors the traffic flow from wireless network. The aim is to protect the wireless traffic and stop possible attacks launched from there. If a signature is identified, the connection is terminated.
- **Host-based IPS (HIPS)** – It actively protects the traffic flow from a single endpoint device. The aim is to investigate the traffic on a particular device. If a signature is identified, the connection is terminated.

## Implementation:

### Step-1: Running Snort in sniffer mode

Sniffing with parameters “-i” verbose mode (-v) and the interface (-i).



**Sniffing with parameter "-i"**

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

In case you have only one interface, Snort uses it by default. The above example demonstrates to sniff on the interface named "eth0". Once you simulate the parameter -v, you will notice it will automatically use the "eth0" interface and prompt it.

**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 as follows;

```
sniffing with -v
user@ubuntu$ sudo snort -v
Running in packet dump mode
    === Initializing Snort ===
...
Commencing packet processing (pid=64)
12/01-20:10:13.846653 192.168.175.129:34316 -> 192.168.175.2:53
  UDP TTL:64 TOS:0x0 ID:23826 Iplen:20 DgmLen:64 DF
  Len: 36
```

ubuntu@ip-10-10-51-177: ~

File Edit View Search Terminal Help

ubuntu@ip-10-10-51-177: ~ \$ sudo snort -v -i eth0

snort: invalid option -- '-'

'-'--> Snort! <\*-  
...- Version 2.9.7.0 GRE (Build 149)  
... By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using libpcap version 1.9.1 (with TPACKET\_V3)  
Using PCRE version: 8.39 2016-06-14  
Using ZLIB version: 1.2.11

USAGE: snort [-options] <filter options>  
Options:  
-A Set alert mode: fast, full, console, test or none (alert file  
e alerts only)  
 "unsock" enables UNIX socket logging (experimental).  
-B Log packets in tcpdump format (much faster!)  
-B <mask> Obfuscated IP addresses in alerts and packet dumps using CIDR  
mask  
-C <rules> Use Rules File <rules>  
-D Print out payloads with character data only (no hex)  
-d Dump the Application Layer  
-D Run Snort in background (daemon) mode

## Step 2: Sniffing with parameter "-v" verbose mode (-v)

**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 as follows;

```

user@ubuntu$ sudo snort -v
Running in packet dump mode
==== Initializing Snort ====
...
Commencing packet processing (pid=64)
12/01-20:10:13.846653 192.168.175.129:34316 -> 192.168.175.2:53
UDP TTL:64 TOS:0x0 ID:23826 IpLen:20 DgmLen:64 DF
Len: 36
=====
12/01-20:10:13.846794 192.168.175.129:38655 -> 192.168.175.2:53
UDP TTL:64 TOS:0x0 ID:23827 IpLen:20 DgmLen:64 DF
Len: 36
=====
Snort exiting
  
```

As you can see in the given output, verbosity mode provides tcpdump like output information.

ubuntu@ip-10-10-51-1: ~

ubuntu@ip-10-10-51-177: ~

File Edit View Search Terminal Help

03/15-06:30:01.068373 10.100.1.202:51044 -> 10.10.51.177:80  
TCP TTL:64 TOS:0x0 ID:11172 Iplen:20 DgmLen:84 DF  
\*\*\*AP\*\*\* Seq: 0xD325E092 Ack: 0x1882B7C5 Win: 0x12D8 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 2636422025 3983149927  
=====

WARNING: No preprocessors configured for policy 0.  
03/15-06:30:01.068469 10.100.1.202:51044 -> 10.10.51.177:80  
TCP TTL:64 TOS:0x0 ID:11173 Iplen:20 DgmLen:84 DF  
\*\*\*AP\*\*\* Seq: 0xD325E092 Ack: 0x1882B7C5 Win: 0x12D8 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 2636422025 3983149927  
=====

03/15-06:30:01.068481 10.100.1.202:51044 -> 10.10.1.202:51044  
TCP TTL:64 TOS:0x0 ID:32730 Iplen:20 DgmLen:52 DF  
\*\*\*AP\*\*\* Seq: 0x1882B7C5 Ack: 0xD325E002 Win: 0x1CB TcpLen: 32  
TCP Options (3) => NOP NOP TS: 3983149927 2636422025  
=====

WARNING: No preprocessors configured for policy 0.  
03/15-06:30:01.068515 10.100.1.202:51044 -> 10.10.51.177:80  
TCP TTL:64 TOS:0x0 ID:11174 Iplen:20 DgmLen:68 DF  
\*\*\*AP\*\*\* Seq: 0xD325E002 Ack: 0x1882B7C5 Win: 0x12D8 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 2636422025 3983149927  
=====

ubuntu@ip-10-10-51-1: ~

ubuntu@ip-10-10-51-177: ~

File Edit View Search Terminal Help

51m 11s

## Step 3: Sniffing with parameter "-d" dumping packet data mode (-d)

**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 as follows;

```

user@ubuntu$ sudo snort -d
Running in packet dump mode
==== Initializing Snort ====
...
Commencing packet processing (pid=67)
12/01-20:45:42.068675 192.168.175.129:37820 -> 192.168.175.2:53
UDP TTL:64 TOS:0x0 ID:53099 IpLen:20 DgmLen:56 DF
Len: 28
99 A5 01 00 00 01 00 00 00 00 00 00 06 67 6F 6F .....gle.com....
67 6C 65 03 63 6F 6D 00 00 01 00 01 .....gle.com....
=====
WARNING: No preprocessors configured for policy 0.
12/01-20:45:42.070742 192.168.175.2:53 -> 192.168.175.129:44947
UDP TTL:128 TOS:0x0 ID:63307 Iplen:20 DgmLen:72
Len: 44
cc c1 01 00 00 01 00 00 00 00 00 00 00 00 00 00
  
```

ubuntu@ip-10-10-51-1: ~

ubuntu@ip-10-10-51-177: ~

File Edit View Search Terminal Help

31 A0 50 83 85 58 AC 7E 1C 83 14 23 68 C1 0D 34 1.P.X.-...#h..  
C4 E2 0D B9 B8 88 87 E6 40 43 0C 00 00 00 FF FF F2 2F 80 00 .....  
FF 00 00 00 00 00 00 00 00 00 FF FF F2 2F 80 00 .....  
00 80 00 00 01 01 01 .....

=====

WARNING: No preprocessors configured for policy 0.  
03/15-06:31:23.977868 10.100.1.202:51044 -> 10.10.51.177:80  
TCP TTL:64 TOS:0x0 ID:16027 Iplen:20 DgmLen:52 DF  
\*\*\*AP\*\*\* Seq: 0xD326E09A Ack: 0x19164A5C Win: 0x12D8 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 2636504931 3983232836  
=====

WARNING: No preprocessors configured for policy 0.  
03/15-06:31:23.978834 10.100.1.202:51044 -> 10.10.51.177:80  
TCP TTL:64 TOS:0x0 ID:16028 Iplen:20 DgmLen:52 DF  
\*\*\*AP\*\*\* Seq: 0xD326E09A Ack: 0x19164CE3 Win: 0x12D8 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 2636504932 3983232836  
=====

ubuntu@ip-10-10-51-1: ~

ubuntu@ip-10-10-51-177: ~

File Edit View Search Terminal Help

50m 42s

#### **Step 4: Sniffing with parameter "-de"**

dump (-d) and link-layer header grabbing (-e)

## **Step 5: Sniffing with parameter "-X" full packet dump mode (-X)**

**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 as follows;



```
ubuntu@ip-10-10-51-1:~
```

File Edit View Search Terminal Help

0x0100: 0A 6F 3B 7B 0B 60 8D CA CA C8 C6 0E 94 72 7B 84 ..o{. ....rp.

0x0109: EA E2 74 02 32 9C 05 91 A4 84 5C 40 01 8C 33 D0 ..t.2. ....\@..3.

0x01F0: 0D 6B 2B 9C 57 B7 87 A7 97 1B 93 37 03 83 DC 19 ..h \. ....?....

0x0200: 7B 29 06 1F 5F 46 2E 16 3F A0 D7 FC 19 83 C8 8E p)...F..?....

0x0210: 47 48 31 30 04 72 07 81 8D 9C 96 00 21 14 AA 83 GH10. ....!....

0x0220: 4A 1E 08 00 00 FF FF 0B 00 00 00 00 00 00 00 FF J.....

0x0230: FF FF 2B 8F 00 00 00 00 00 00 01 01 01 ..

=====

WARNING: No preprocessors configured for policy 0.

03/15/06 13:34:36.828196 10.100.1.202:51044 -> 10.10.51.177:80

TCP TTL:64 TOS:0x0 ID:21761 Iplen:20 DgmLen:52 DF

\*\*\*A\*\*\*\* Seq: 0x032803FC Ack: 0x19BEC178 Win: 0x13AA TcpLen: 32

TCP Options (3) => NOP NOP TS: 2636697775 3983425687

0x0000: 02 33 98 AC F3 CF 02 C8 85 B5 5A 0A 00 45 00 ..Z.....E.

0x0010: 00 C4 BE DD 00 00 01 11 9A A7 C0 AB AF 01 EF FF ..

0x0020: FF FA E5 02 07 6C 00 B8 85 AE 4D 2D 53 45 41 52 ..1....M-SEAR

0x0030: 43 48 2B 2A 2B 48 54 54 50 2F 31 2E 31 0D 0A 48 CH \* HTTP/1.1.H

0x0040: 4F 53 54 3A 20 32 33 39 2E 32 35 2E 32 35 35 OST: 239.255.255

0x0050: 2E 32 35 30 3A 31 39 30 30 0A 4D 41 4E 3A 20 ..250:1900..MAN:

0x0060: 22 73 73 64 70 3A 64 69 73 63 6F 76 65 72 22 0D "ssdp:discover".

0x0070: 0A 4D 5B 3A 20 31 0D 0A 53 54 3A 2B 75 72 6E 3A ..MX: 1..ST: urn:

0x0080: 64 69 61 6C 2D 6D 75 76 69 73 63 72 65 66 ddi-multiscreen

0x0090: 2D 6F 72 67 3A 73 65 72 76 69 63 65 3A 64 69 61 ..org:service:dia

=====

48M 15s

## Packet Logger Mode

### Step 6: Logging with parameter "-l"

**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 / 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

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.

**LogFile Ownership**

Before generating logs and investigating them, we must remember the Linux file ownership and permissions. No need to deep dive into user types and permissions. The fundamental file ownership rule; whoever creates a file becomes the owner of the corresponding file.

```
ubuntu@ip-10-10-51-17:~$ sudo snort -dev -l
snort: option requires an argument -- 'l'

-> Snort! <-
o"-")- Version 2.9.7.0 GRE (Build 249)
By Martin Roesch & The Snort Team: http://www.snort.org/contact@team
Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.9.1 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11

USAGE: snort [-options] <filter options>
Options:
-A Set alert mode: fast, full, console, test or none (alert file
e alerts only) "unsock" enables UNIX socket logging (experimental).
-B <mask> Obfuscated IP addresses in alerts and packet dumps using CIDR
mask
-C <rules> Use Rules File <rules>
-D Print out payloads with character data only (no hex)
-d Dump the Application Layer
-D Run Snort in background (daemon) mode
```

### Step 7: Reading log file

```
Total allocated space (uordblks): 678144
Total free space (fordblk): 108288
Topmost releasable block (keepcost): 102304
=====
Packet I/O Totals:
Received: 4327
Analyzed: 4327 (100.000%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 0 ( 0.000%)
Injected: 0
=====
Breakdown by protocol (includes rebuilt packets):
Eth: 4327 (100.000%)
VLAN: 0 ( 0.000%)
IP4: 4325 ( 99.954%)
Frag: 0 ( 0.000%)
ICMP: 68 ( 1.572%)
UDP: 2 ( 0.046%)
TCP: 3629 ( 83.869%)
IP6: 0 ( 0.000%)
IP6 Ext: 0 ( 0.000%)
IP6 Opts: 0 ( 0.000%)
Frag6: 0 ( 0.000%)
```

## Step 8: Log Packet in ASCII mode

**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 as follows;
```

The screenshot shows a terminal window titled "logging with -K ASCII" and a file manager window titled "ubuntu - File Manager".

**Terminal Output:**

```
logging with -K ASCII
user@ubuntu$ sudo snort -dev -K ASCII -l .
Running in packet logging mode
==== Initializing Snort ====
Initializing Output Plugins!
Log directory = /var/log/snort
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

==== Initialization Complete ====
...
Commencing packet processing (pid=2679)
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
```

**File Manager:**

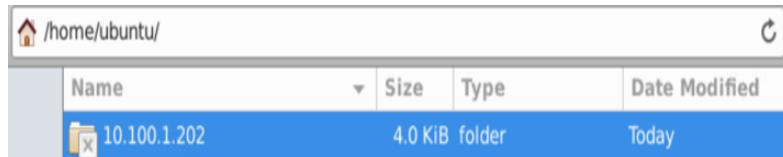
Name	Size	Type	Date Modified
10.100.1.202	4.0 KiB	folder	Today
Desktop	4.0 KiB	folder	01/10/22
Documents	4.0 KiB	folder	07/12/21
Downloads	4.0 KiB	folder	02/04/22
Music	4.0 KiB	folder	07/12/21
Pictures	4.0 KiB	folder	07/13/21
Public	4.0 KiB	folder	07/12/21
Templates	4.0 KiB	folder	07/12/21
Videos	4.0 KiB	folder	07/12/21
PACKET_NONIP	0 bytes	plain text document	Today

The terminal also shows the command "ls" output for the folder "10.100.1.202":

```
Snort exiting
ubuntu@ip-10-10-51-177:~$ ls
10.100.1.202  Documents  Music      Pictures  Templates
Desktop       Downloads  PACKET_NONIP  Public    Videos
ubuntu@ip-10-10-51-177:~$
```

## In Code View & Folder View

```
Snort exiting
ubuntu@ip-10-10-51-177:~$ ls
10.100.1.202  Documents  Music      Pictures  Templates
Desktop       Downloads  PACKET_NONIP  Public    Videos
ubuntu@ip-10-10-51-177:~$
```



## IDS & IPS Mode

### Step 1: Starting snort instance and testing conf file

#### 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 misconfiguration 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
```

Now run the traffic-generator script as sudo and start ICMP/HTTP traffic. Once the traffic is generated, snort will start processing the packets and accomplish the given task with additional parameters.

```
ubuntu@ip-10-10-51-177:~$ sudo snort -c /etc/snort/snort.conf -T
Running in Test mode

--- Initializing Snort ---
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP PORTS' defined : [ 80:81 311 383 591 593 901 1220 1414 1741 1838
2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777
7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300
8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002
55555 ]
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 1024:65535 ]
PortVar 'SSH_PORTS' defined : [ 22 ]
PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined : [ 5060:5061 5680 ]
PortVar 'FILE_DATA_PORTS' defined : [ 80:81 110 143 311 383 591 593 901 1220 14
14 1741 1830 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:71
45 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 82
43 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444
41080 50002 55555 ]
```

### Step 2: Starting snort instance in background mode and checking process with ps command

#### 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
```

Now run the traffic-generator script as sudo and start ICMP/HTTP traffic. Once the traffic is generated, snort will start processing the packets and accomplish the given task with additional parameters.

```
running in background mode
user@ubuntu$ sudo snort -c /etc/snort/snort.conf -D
Spawning daemon child...
My daemon child 1996 lives...
Daemon parent exiting (0)
```

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

Once you start the background mode and want to check the corresponding process, you can easily use the "ps" command as shown below;

```
running in background mode
user@ubuntu$ ps -ef | grep snort
root      2898  1706  0 05:53 ?        00:00:00 snort -c
/etc/snort/snort.conf -D
```

```
ubuntu@ip-10-10-51-177:~$ sudo snort -c /etc/snort/snort.conf -D
Spawning daemon child...
My daemon child 1996 lives...
Daemon parent exiting (0)
ubuntu@ip-10-10-51-177:~$ ps -ef | grep snort
root      1996  1 0 06:48 ?        00:00:00 snort -c /etc/snort/snort.conf -D
ubuntu     2000 1722  0 06:49 pts/0    00:00:00 grep --color=auto snort
ubuntu@ip-10-10-51-177:~$
```

### **Step 3: Console alert mode**

**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.

```
user@ubuntu$ sudo snort -c /etc/snort/snort.conf -A console
Running in IDS mode

--== Initializing Snort ==
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
...
Commencing packet processing (pid=3743)
12/12/02:08:27.577495 [**] [1:366:7] ICMP PING *NIX [**] [Classification: activity] [Priority: 3] [ICMP] 192.168.175.129 -> 142.250.187.110
12/12-02:08:27.577495 [**] [1:10000001:0] ICMP Packet found [**] [Priority: 3] [ICMP] 192.168.175.129 -> 142.250.187.110
12/12-02:08:27.577495 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] [ICMP] 192.168.175.129 -> 142.250.187.110
12/12-02:08:27.608919 [**] [1:10000001:0] ICMP Packet found [**] [Priority: 3] [ICMP] 192.168.175.129 -> 142.250.187.110
```

```
ubuntu@ip-10-10-51-177: ~
File Edit View Search Terminal Help
ubuntu@ip-10-10-51-177: $ sudo snort -c /etc/snort/snort.conf -A console
Running in IDS mode

--== Initializing Snort ==
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP PORTS' defined : [ 80:81 311 383 591 593 901 1220 1414 1741 1830
2301 2381 2809 3037 3128 3702 4434 4848 5256 6988 7000:7001 7144:7145 7510 7777
7779 8000 8008 8014 8028 8008 8088 8090 8118 8123 8180:8181 8243 8280 8300
8800 8888 8899 9000 9060 9088 9090:9091 9443 9999 11371 34443:34444 41080 50000
55555 ]
PortVar 'SHELLCODE PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE PORTS' defined : [ 1024:65535 ]
PortVar 'SSH PORTS' defined : [ 22 ]
PortVar 'FTP PORTS' defined : [ 21 2100 3535 ]
PortVar 'SIP PORTS' defined : [ 5060:5061 5600 ]
PortVar 'FILE DATA PORTS' defined : [ 80:81 110 143 311 383 591 593 901 1220 14
14 1741 1830 2301 2809 3037 3128 3702 4434 4848 5256 6988 7000:7001 7144:7145
7510 7777 7779 8000 8008 8014 8028 8088 8090 8118 8123 8180:8181 8182
43 8280 8300 8800 8888 8899 9000 9060 9088 9090:9091 9443 9999 11371 34443:34444
41080 50000 55555 ]
```

```
+-----[Rule Port Counts]-----  
|          tcp    udp   icmp     ip  
|  src    151     18      0      0  
|  dst    3306    126     0      0  
|  any    383     48     146     22  
|  nc     27      8      95     20  
|  s+d    12      5      0      0  
+-----[detection-filter-config]-----
```

**Step 4: cmg mode provides basic header details with payload in hex and text format**

**Step 5: Fast mode provides alert messages, timestamps, source, and destination IP address**

**Step 6: -A none does not create the alert file but creates log 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.

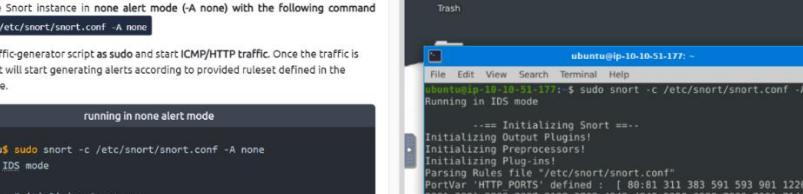
**running in none alert mode**

```
user@ubuntu$ sudo snort -c /etc/snort/snort.conf -A none
Running in IDS mode

--= Initializing Snort --=
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
...
Commencing packet processing (pid=3745)
=====
```

As you can see in the picture below, there is no alert file. Snort only generated the log file.

**Snort2-PROD...**



ubuntu@ip-10-10-51-177: ~ \$ sudo snort -c /etc/snort/snort.conf -A none  
Running in IDS mode  
--= Initializing Snort --=  
Initializing Output Plugins!  
Initializing Preprocessors!  
Initializing Plug-ins!  
Parsing Rules file "/etc/snort/snort.conf"  
...  
Commencing packet processing (pid=3745)  
=====

Name Size  
snort.log 1638307205  
0.1M

## **Created Log Files**

Name	Size	Type	Date Mod
alert	36.2 KiB	plain text document	Today
snort.log	198 bytes	application log	Today
snort.log.1677243122	0 bytes	plain text document	Today
snort.log.1677243619	0 bytes	plain text document	Today
snort.log.1677243859	0 bytes	plain text document	Today

**Alert File**

```
Open ▾ + alert [Read-Only]
/var/log/snort
1 [**] [1:1000001:1] ICMP Packet Found [**]
2 [Priority: 0]
3 02/24-12:46:37.052334 fe80::5e:f4ff:fe3b:2563 -> ff02::2
4 IPV6-ICMP TTL:255 TOS:0x0 ID:0 IpLen:40 DgmLen:56
5
6 [**] [1:366:7] ICMP PING *NIX [**]
7 [Classification: Misc activity] [Priority: 3]
8 02/24-12:47:46.055586 192.168.175.129 -> 142.250.187.110
9 ICMP TTL:64 TOS:0x0 ID:682 IpLen:20 DgmLen:84 DF
0 Type:8 Code:0 ID:12 Seq:1 ECHO
1
2 [**] [1:1000001:1] ICMP Packet Found [**]
3 [Priority: 0]
4 02/24-12:47:46.055586 192.168.175.129 -> 142.250.187.110
```

**Conclusion/Summary:**

<b>Student Signature &amp; Date</b>	<b>Marks:</b>	<b>Evaluator Signature &amp; Date</b>
-------------------------------------	---------------	---------------------------------------

## Practical 6

**Date:** / /2023

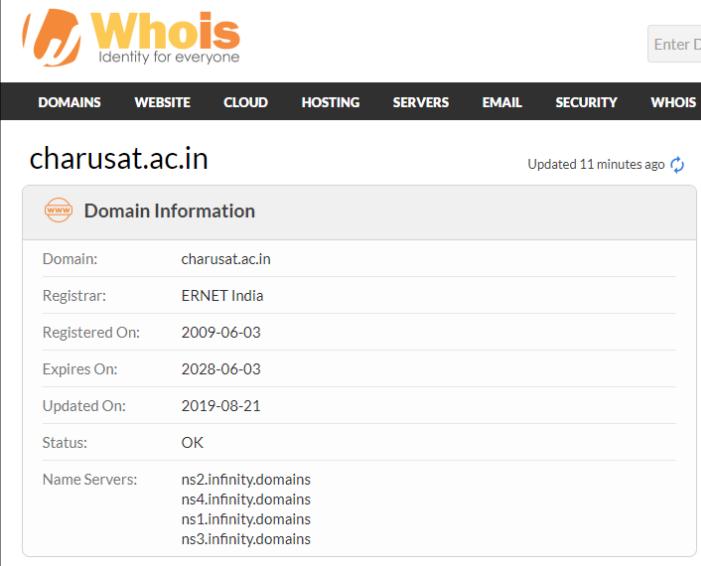
**Aim:** Implementation to gather information from any PC's connected to the LAN using whois, port scanners, network scanning, IP scanners etc.

### Solution:

#### 1) IP scanner- whois:

'Whois' is a widely used Internet record listing that identifies who owns a domain and how to get in contact with them. Here for example we have searched for domain name charusat.ac.in  
<https://www.charusat.ac.in/>

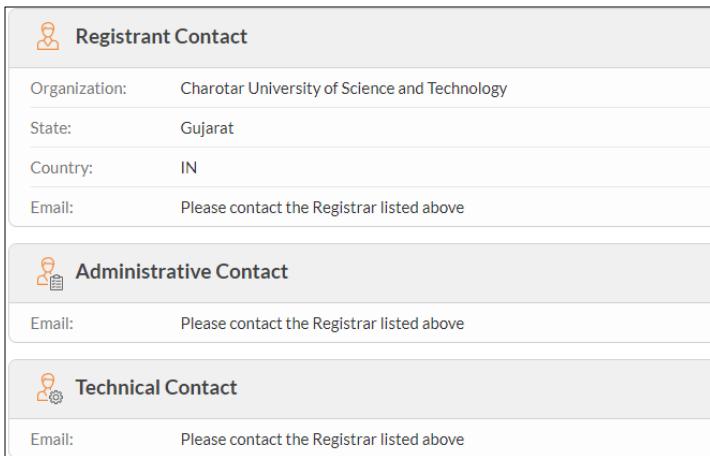
It displays domain information like the register date, update date, registrar provider etc.



The screenshot shows the Whois search results for the domain `charusat.ac.in`. The results are displayed in a table format under the heading "Domain Information".

Attribute	Value
Domain:	charusat.ac.in
Registrar:	ERNET India
Registered On:	2009-06-03
Expires On:	2028-06-03
Updated On:	2019-08-21
Status:	OK
Name Servers:	ns2.infinity.domains ns4.infinity.domains ns1.infinity.domains ns3.infinity.domains

At the top right of the results page, it says "Updated 11 minutes ago" with a refresh icon.



The screenshot shows the Whois search results for the domain `charusat.ac.in`, continuing from the previous page. It displays contact information for three roles: Registrant Contact, Administrative Contact, and Technical Contact.

Contact Type	Organization	Address
Registrant Contact	Charotar University of Science and Technology	State: Gujarat, Country: IN, Email: Please contact the Registrar listed above
Administrative Contact	Charotar University of Science and Technology	Email: Please contact the Registrar listed above
Technical Contact	Charotar University of Science and Technology	Email: Please contact the Registrar listed above

## 2) Port scanners

Port Scanner is an application that is used to determine the open ports on the network. Port scanning is performed to get information about open ports that are ready to receive information.

Port Scanning is a five-step process as described below.

**Step 1:** For port scanning, there is a need for active hosts. Active hosts can be discovered using the network scanning process.

**Step 2:** These active hosts are mapped to their IP addresses.

**Step 3:** Now we have active hosts and thus port scanning process is performed. In this process, packets are sent to specific ports on a host.

**Step 4:** Here responses will get analyzed.

**Step 5:** Through this analysis, information about running services will be learned and potential vulnerabilities will be identified.

### 2.1) Nmap Online port scanner:

Here we have searched for open ports for amazon.in using it's IP address.

The screenshot shows a web-based Nmap scan interface. At the top, the IP address '52.95.116.115' is entered. Below it is a teal button labeled 'QUICK NMAP SCAN'. The main area displays the scan results:

```
Starting Nmap 7.40 ( https://nmap.org ) at 2023-01-04 04:31 UTC
Nmap scan report for 52.95.116.115
Host is up (0.080s latency).

PORT      STATE    SERVICE
21/tcp    closed   ftp
22/tcp    closed   ssh
23/tcp    closed   telnet
80/tcp    open     http
110/tcp   closed   pop3
143/tcp   closed   imap
443/tcp   open     https
3389/tcp  closed   ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
```

### 3) Nikto:

Nikto is an open source web server and web application scanner. Nikto can perform comprehensive tests against web servers for multiple security threats, including over 6700 potentially dangerous files/programs. Nikto can also perform checks for outdated web servers software, and version-specific problems.

To perform a simple domain scan, use the `-h` (host) flag:

```
nikto -h scanme.nmap.org
```

```
$ nikto -h scanme.nmap.org
- Nikto v2.1.6
[+] Target IP: 45.33.32.156
[+] Target Hostname: scanme.nmap.org
[+] Target Port: 80
[+] Start Time: 2023-01-04 10:25:46 (GMT5.5)
[+] Server: Apache/2.4.7 (Ubuntu) 1 354 000+ 記事
[+] Retrieved via header: HTTP/1.1 forward.http.proxy:3128
[+] The anti-clickjacking X-Frame-Options header is not present.
[+] The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS 2 477 000+ articles
[+] The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
```

For domains with HTTPS enabled, you have to specify the `-ssl` flag to scan p:

Here we have scanned Wikipedia.org

```
$ nikto -h https://www.wikipedia.org/ -ssl
- Nikto v2.1.6
[+] Target IP: 103.102.166.224
[+] Target Hostname: www.wikipedia.org
[+] Target Port: 443
[+] SSL Info: Subject: /C=US/ST=California/L=San Francisco/O=Wikimedia Foundation, Inc./CN=*.wikipedia.org
Ciphers: TLS_AES_256_GCM_SHA384
Issuer: /C=US/O=DigiCert Inc/CN=DigiCert TLS Hybrid ECC SHA384 2020 CA1
[+] Start Time: 2023-01-04 10:30:36 (GMT5.5) 6 585 000+ articles
[+] Server: ATS/9.1.3
[+] Cookie GeoIP created without the httponly flag
[+] IP address found in the 'x-client-ip' header. The IP is "136.233.130.146".
[+] The anti-clickjacking X-Frame-Options header is not present.
[+] The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
[+] Uncommon header 'x-cache' found, with contents: cp5024 hit, cp5024 hit/1800 45
[+] Uncommon header 'accept-ch' found, with contents: Sec-CH-UA-Arch,Sec-CH-UA-Bitness,Sec-CH-UA-Full-Version-List,Sec-CH-UA-Model,Sec-CH-UA-Platform-Version 1 785 000+ voc
[+] Uncommon header 'report-to' found, with contents: { "group": "wm_nel", "max_age": 86400, "endpoints": [ { "url": "https://intake-logging.wikimedia.org/v1/events?stream=w3c.reportingapi.network_error&schema_uri=/w3c/reportingapi/network_error/1.0.0" } ] }
[+] Uncommon header 'permissions-policy' found, with contents: interest-cohort=(),ch-ua-arch=(self "intake-analytics.wikimedia.org"),ch-ua-bitness=(self "intake-analytics.wikimedia.org"),ch-ua-full-version-list=(self "intake-analytic
```

#### 4) Zenmap:

Zenmap is the official Nmap Security Scanner GUI. It is a multi-platform (Linux, Windows, Mac OS X, BSD, etc.) free and open source application which aims to make Nmap easy for beginners to use while providing advanced features for experienced Nmap users. Frequently used scans can be saved as profiles to make them easy to run repeatedly. A command creator allows interactive creation of Nmap command lines. Scan results can be saved and viewed later. Saved scan results can be compared with one another to see how they differ.

We have used it to scan for network and ip addresses

```

Target: 142.250.192.36
Command: nmap -T4 -A -v 142.250.192.36
Profile: Intense scan

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans
OS Host bom12s15-in-f4.1e
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:52
Completed NSE at 10:52, 0.00s elapsed
Initiating NSE at 10:52
Completed NSE at 10:52, 0.00s elapsed
Initiating NSE at 10:52
Completed NSE at 10:52, 0.00s elapsed
Initiating NSE at 10:52
Completed NSE at 10:52, 0.00s elapsed
Scanning 142.250.192.36 [4 ports]
Completed Ping Scan at 10:52, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:52
Completed Parallel DNS resolution of 1 host. at 10:52, 0.01s elapsed
Initiating SYN Stealth Scan at 10:52
Scanning 142.250.192.36 [44 services on 142.250.192.36] [1000 ports]
Discovered open port 443/tcp on 142.250.192.36
Discovered open port 80/tcp on 142.250.192.36
Completed SYN Stealth Scan at 10:52, 4.78s elapsed (1000 total ports)
Initiating Service scan on 10:52
Scanning 2 services on bom12s15-in-f4.1e100.net (142.250.192.36)
Service scan.Timing: About 50.00% done; ETC: 10:54 (0:00:39 remaining)
Completed Service scan at 10:54, 62.14s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against bom12s15-in-f4.1e100.net (142.250.192.36)
Initiating Traceroute at 10:54
Completed Traceroute at 10:54, 0.02s elapsed
NSE: Script scanning 142.250.192.36.
Initiating NSE at 10:54
Completed NSE at 10:54, 19.73s elapsed
Initiating NSE at 10:54
Completed NSE at 10:54, 1.88s elapsed
Initiating NSE at 10:54
Completed NSE at 10:54, 0.00s elapsed
Nmap scan report for bom12s15-in-f4.1e100.net (142.250.192.36)
Host is up (0.0000s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http
|_fingerprint-strings:
|   HTTPOptions:
|     HTTP/1.1 200 OK
|     Date: Wed, 04 Jan 2023 05:23:09 GMT
|     Cache-Control: no-cache
|     Pragma: no-cache
|     Content-Type: text/html; charset=UTF-8"
|     Content-Length: 714
|     Via: HTTP/1.1 forward.http.proxy:3128
|     Connection: close
|     <HTML><HEAD>
|     <meta http-equiv=pragma content=nocache>
|     <META HTTP-EQUIV=Expires CONTENT=>1>
|     <SCRIPT>
|     DOMParser().parseFromString('<a></a>', 'text/xml').documentElement.textContent;
|     window.onload=function() {

```

On Scanning we found two open ports and their information

Port1

Port 2

## Once the Scan is over

```

Target: 142.250.192.36
Command: nmap -T4 -A -v 142.250.192.36

Hosts: OS: bom12s15-in-f4.1e
      142.250.192.36 [open]

      OS: Android 7.X, Linux 3.X
      OS_CPE: cpe:/o:google:android:7.1.2 cpe:/o:linux:linux_kernel:3.10
      OS_details: Android 7.1.2 (Linux 3.10)
      Network Distance: 1 hop
      TCP Sequence Prediction: Difficulty=262 (Good luck!)
      IP ID Sequence Generation: All zeros

      TRACEROUTE (using port 443/tcp)
      HOP RTT ADDRESS
      1  0.00 ms bom12s15-in-f4.1e[0].net (142.250.192.36)

      NSE: Script Post-scanning.
      Initiating NSE at 10:54
      Completed NSE at 10:54, 0.00s elapsed
      Initiating NSE at 10:54
      Completed NSE at 10:54, 0.00s elapsed
      Initiating NSE at 10:54
      Completed NSE at 10:54, 0.00s elapsed
      Read data files from: C:\Program Files (x86)\Nmap
      OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
      Nmap done: 1 IP address (1 host up) scanned in 91.63 seconds
      Raw packets sent: 2060 (93.208KB) | Rcvd: 34 (1.576KB)
  
```

## Conclusion/Summary:

Student Signature & Date	Marks:	Evaluator Signature & Date
--------------------------	--------	----------------------------

## Practical 7

<b>Date:</b> / /2023
----------------------

**Aim:** Set up a Virtual lab environment with Windows XP (SP1), Metasploitable OS, and BRICKS/DVWA web server and an Attacker machine (KALI/BT) in virtual machines (network in NAT mode). Now carry out Vulnerability assessment in environment

### a. Network VA/PT

- i. Find the open ports in domain.
- ii. Find out the hosts in domains.
- iii. Find out the services running on domains and their versions.
- iv. Banner Grabbing of server.
- v. Find out default vulnerabilities in Services.
- vi. Exploit the vulnerabilities.
- vii. Deploy and maintain the backdoor.

### b. Web VA/PT

- i. Find the domain information.
- ii. Find the details of server and its default vulnerabilities.
- iii. Perform automated testing using BurpSuite or ZAP proxies.

Tools: nmap, netcat, netcraft, nslookup, whois, dig, ping, Nessus, Metasploit, FOCA.

## Theory:

### METASPLOIT:

- Metasploit is one of the best penetration testing frameworks that help a business find out and shore up vulnerabilities in their systems before exploitation by hackers. To put it simply, Metasploit allows hacking with permission.
- A Metasploit penetration test begins with the information gathering phase, wherein Metasploit integrates with various reconnaissance tools like Nmap, SNMP scanning, and Windows patch enumeration, and Nessus to find the vulnerable spot in your system.
- Once the weakness is identified, choose an exploit and payload to penetrate the chink in the armor.
- If the exploit is successful, the payload gets executed at the target, and the user gets a shell to interact with the payload. One of the most popular payloads to attack Windows systems is Meterpreter – an in-memory-only interactive shell.

Once on the target machine, Metasploit offers various exploitation tools for privilege escalation, packet sniffing, pass the hash, keyloggers, screen capture, plus pivoting tools. Users can also set up a persistent backdoor if the target machine gets rebooted.

## Implementation:

### Step1: Start metasploit

```

Press ENTER to size up the situation

xxxxxxxxxxxxxxxxxxxxxxxxxxxx Date: April 25, 1848 xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx Weather: It's always cool in the lab xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx Health: Overweight xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx Caffeine: 12975 mg xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx Hacked: All the things xxxxxxxxxxxxxxxxxxxxxxx

Press SPACE BAR to continue

-[ metasploit v6.2.9-dev
+ --=[ 2230 exploits - 1177 auxiliary - 398 post
+ --=[ 867 payloads - 45 encoders - 11 nops
+ --=[ 9 evasion

Metasploit tip: To save all commands executed since start up
to a file, use the makerc command

msf6 > s

```

**Step 2:** Find the vulnerability using nessus tool in windows xp.

**Step 3:** Search the vulnerability. Command: search ms04-007

```

msf6 > search ms04-007
Matching Modules

#  Name                               Disclosure Date   Rank   Check  Description
#  exploit/windows/smb/ms04_007_killbill  2004-02-10     Low   No    MS04-007 Microsoft ASN.1 Library Bitstring Head Overflow

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms04_007_killbill

```

**Step 4:** Now, use the path of exploit.

Command: use exploit/windows/smb/ms04\_007\_killbill

```

msf6 > use exploit/windows/smb/ms04_007_killbill
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms04_007_killbill) >

```

**Step 5:** List out the option. Command: show options

```
msf6 exploit(windows/smb/ms04_007_killbill) > show options

Module options (exploit/windows/smb/ms04_007_killbill):

Name      Current Setting  Required  Description
PROTO     smb                yes       Which protocol to use (Accepted: smb, http)
RHOSTS    192.168.2.15        yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     445                yes       The SMB service port (TCP)

Payload options (windows/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.2.15        yes       The listen address (an interface may be specified)
LPORT     4444                yes       The listen port

Exploit target:

Id  Name
-- 
#  Windows 2000 SP2-SP4 + Windows XP SP0-SP1
```

**Step 6:** Set the RHOSTS by using the IP of windows.

Command: set RHOSTS IP\_address

```
msf6 exploit(windows/smb/ms04_007_killbill) > set RHOSTS 192.168.200.237
RHOSTS => 192.168.200.237
```

**Step 7:** Now, to set payload, we have find the index of payload using the following command.  
Command: show payloads

```
msf6 exploit(windows/smb/ms04_007_killbill) > show payloads

Available Payloads:
#  Name
0  payload/generic/cobalt
1  payload/generic/devnull
2  payload/generic/evilbin_tcp
3  payload/generic/evilbin_udp
4  payload/generic/evilhttp
5  payload/generic/evilhttp2
6  payload/windows/adbapi
7  payload/windows/ellipticcurve_dsa_eccdsa_spkac_tcp
8  payload/windows/ellipticcurve_dsa_eccdsa_spkac_udp
9  payload/windows/ellipticcurve_dsa_spkac_tcp
10  payload/windows/ellipticcurve_dsa_spkac_udp
11  payload/windows/ellipticcurve_ecdsa_spkac
12  payload/windows/ellipticcurve_ecdsa_spkac_tcp
13  payload/windows/ellipticcurve_ecdsa_spkac_udp
14  payload/windows/ellipticcurve_ecdsa_spkac_x509
15  payload/windows/ellipticcurve_ecdsa_spkac_x509_tcp
16  payload/windows/ellipticcurve_ecdsa_spkac_x509_udp
17  payload/windows/ellipticcurve_ecdsa_spkac_x509_x509
18  payload/windows/ellipticcurve_ecdsa_spkac_x509_x509_tcp
19  payload/windows/ellipticcurve_ecdsa_spkac_x509_x509_udp
20  payload/windows/ellipticcurve_ecdsa_x509
21  payload/windows/ellipticcurve_ecdsa_x509_tcp
22  payload/windows/ellipticcurve_ecdsa_x509_udp
23  payload/windows/ellipticcurve_ecdsa_x509_x509
24  payload/windows/ellipticcurve_ecdsa_x509_x509_tcp
25  payload/windows/ellipticcurve_ecdsa_x509_x509_udp
26  payload/windows/ellipticcurve_ecdsa_x509_x509_x509
27  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_tcp
28  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_udp
29  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509
30  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_tcp
31  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_udp
32  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_x509
33  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_x509_tcp
34  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_x509_udp
35  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_x509_x509
36  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_x509_x509_tcp
37  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_x509_x509_udp
38  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_x509_x509_x509
39  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_x509_x509_x509_tcp
40  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_x509_x509_x509_udp
41  payload/windows/ellipticcurve_ecdsa_x509_x509_x509_x509_x509_x509_x509_x509
42  payload/windows/filecat_all
43  payload/windows/filecat_tcp
44  payload/windows/messages
45  payload/windows/meterpreter/reverse_tcp
46  payload/windows/meterpreter/reverse_tcp_x509
47  payload/windows/meterpreter/reverse_tcp_x509_tcp
48  payload/windows/meterpreter/reverse_tcp_x509_x509
49  payload/windows/meterpreter/reverse_tcp_x509_x509_tcp
50  payload/windows/meterpreter/reverse_tcp_x509_x509_x509
51  payload/windows/meterpreter/reverse_tcp_x509_x509_x509_tcp
52  payload/windows/meterpreter/reverse_tcp_x509_x509_x509_x509
```

**Step 8:** Setting Payload

Command: set payload 106

```
msf6 exploit(windows/sub/ms06_007_killbill) > set payload 106
payload => windows/peinject/bind_tcp_rc4
msf6 exploit(windows/sub/ms06_007_killbill) > 
```

**Step 9:** Final step is to perform exploit.

Command: exploit

```
[+] 192.168.200.237:445 - Exploit failed: no implicit conversion of nil into String
[*] Exploit completed, but no session was created.
msf6 exploit(windows/sub/ms06_007_killbill) > 
```

**b. Web VA/PT****Step 1:** Download Damn vulnerable web application (DVWA)

```
└$ sudo git clone https://github.com/digininja/DVWA
[sudo] password for jinalkotadia:
Cloning into 'DVWA' ...
remote: Enumerating objects: 4119, done.
remote: Counting objects: 100% (133/133), done.
remote: Compressing objects: 100% (78/78), done.
remote: Total 4119 (delta 66), reused 112 (delta 54), pack-reused 3986
Receiving objects: 100% (4119/4119), 1.86 MiB | 1.00 MiB/s, done.
Resolving deltas: 100% (1924/1924), done.
```

**Step 2:** Configure DVWA using flowing commands.

1. Sudo chmod –R 777 dvwa/
2. cd dvwa/config
3. sudo cp config.inc.php.dist config.inc.php
4. sudo nano config.inc.php

The screenshot shows a terminal window titled 'PS> rushikrathod@kali: /home/rushikrathod/var/www/html/dvwa/config'. The window displays the contents of the config.inc.php file. The file contains PHP code for DVWA database configuration, including variables for DBMS, database, user, password, port, and ReCAPTCHA settings. The terminal also shows the nano editor's status bar at the bottom, indicating it has read 66 lines from DOS format.

```

File Actions Edit View Help
GNU nano 6.4 ./config.inc.php
<?php

# If you are having problems connecting to the MySQL database and all of the variables below are correct
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
#   Thanks to @digininja for the fix.

# Database management system to use
$DBMS = 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
#   WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
#   Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'dvwa';
$_DVWA[ 'db_password' ] = 'p@ssw0rd';
$_DVWA[ 'db_port' ] = '3306';

# ReCAPTCHA settings
#   Used for the 'Insecure CAPTCHA' module
#   You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
$_DVWA[ 'recaptcha_public_key' ] = '';
$_DVWA[ 'recaptcha_private_key' ] = '';

# Default security level
#   Default value for the security level with each session.

```

**Step 3:** Install Mysql on Kali in new cmd then start mysql.

The screenshot shows two terminal commands. The first command is 'sudo apt install default-mysql-server', which installs the MySQL server. The second command is 'sudo service mysql start', which starts the MySQL service. Both commands are run in a terminal window.

```

$ sudo apt install default-mysql-server
[sudo] password for jinalkotadia:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
default-mysql-server is already the newest version (1.0.8).
0 upgraded, 0 newly installed, 0 to remove and 648 not upgraded.

$ sudo service mysql start

```

#### Step 4: Configure MySql database

```

└$ systemctl status mysql
● mariadb.service - MariaDB 10.6.10 database server
  Loaded: loaded (/lib/systemd/system/mariadb.service; disabled; preset: disabled)
  Active: active (running) since Wed 2023-02-15 02:16:09 EST; 7min ago
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
  Process: 52760 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=>
  Process: 52762 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=>
  Process: 52764 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR=`cd >
  Process: 52810 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (cod>
  Process: 52812 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
 Main PID: 52793 (mariadb)
   Status: "Taking your SQL requests now..."
     Tasks: 8 (limit: 4584)
    Memory: 99.7M
      CPU: 1.328s
     CGroup: /system.slice/mariadb.service
             └─52793 /usr/sbin/mariadb

Feb 15 02:16:09 kali mariadb[52793]: 2023-02-15  2:16:09 0 [Note] InnoDB: Loading buffer pool(s) >
Feb 15 02:16:09 kali mariadb[52793]: 2023-02-15  2:16:09 0 [Warning] You need to use --log-bin to >
Feb 15 02:16:09 kali mariadb[52793]: 2023-02-15  2:16:09 0 [Note] InnoDB: Buffer pool(s) load com>
Feb 15 02:16:09 kali mariadb[52793]: 2023-02-15  2:16:09 0 [Note] Server socket created on IP: '1>
Feb 15 02:16:09 kali mariadb[52793]: 2023-02-15  2:16:09 0 [Note] /usr/sbin/mariadb: ready for c>
Feb 15 02:16:09 kali mariadb[52793]: Version: '10.6.10-MariaDB-1+b1'  socket: '/run/mysqld/mysqld>
Feb 15 02:16:09 kali systemd[1]: Started MariaDB 10.6.10 database server.
Feb 15 02:16:09 kali /etc/mysql/debian-start[52814]: Upgrading MySQL tables if necessary.
Feb 15 02:16:09 kali /etc/mysql/debian-start[52826]: Checking for insecure root accounts.
Feb 15 02:16:09 kali /etc/mysql/debian-start[52830]: Triggering myisam-recover for all MyISAM tabl>
lines 1-28/28 (END)
```

```

└$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-1+b1 Debian n/a

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create user 'user'@'127.0.0.1' identified by 'pass';
Query OK, 0 rows affected (0.009 sec)

MariaDB [(none)]> grant all privileges on dvwa.* to 'user'@'127.0.0.1' identified by 'pass';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> exit
Bye
```

**Step 5:** Install php and extensions

```
$ sudo apt -y install lsb-release apt-transport-https ca-certificates
[sudo] password for jinalkotadia:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lsb-release is already the newest version (12.0-1).
lsb-release set to manually installed.
ca-certificates is already the newest version (20211016).
The following additional packages will be installed:
  apt apt-utils libapt-pkg6.0
Suggested packages:
  apt-doc aptitude | synaptic | wajig
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  apt apt-utils libapt-pkg6.0
3 upgraded, 1 newly installed, 0 to remove and 1338 not upgraded.
Need to get 2,594 kB of archives.
After this operation, 28.7 kB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 libapt-pkg6.0 amd64 2.5.5 [903 kB]
```

```
$ sudo wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
--2023-02-15 02:50:01-- https://packages.sury.org/php/apt.gpg
Resolving packages.sury.org (packages.sury.org)... 185.40.106.117
Connecting to packages.sury.org (packages.sury.org)|185.40.106.117|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1769 (1.7K) [application/octet-stream]
Saving to: '/etc/apt/trusted.gpg.d/php.gpg'

/etc/apt/trusted.gpg.d/p 100%[=====] 1.73K --.-KB/s in 0s

2023-02-15 02:50:02 (19.5 MB/s) - '/etc/apt/trusted.gpg.d/php.gpg' saved [1769/1769]
```

```
$ sudo apt install php8.1 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be upgraded:
  php8.1
1 upgraded, 0 newly installed, 0 to remove and 1346 not upgraded.
Need to get 31.2 kB of archives.
After this operation, 5,120 B of additional disk space will be used.
Get:1 https://packages.sury.org/php buster/main amd64 php8.1 all 8.1.16+repack-1+0~20230214.36+debian10~1.gpbp38498 [31.2 kB]
Fetched 31.2 kB in 2s (20.5 kB/s)
(Reading database ... 393836 files and directories currently installed.)
Preparing to unpack .../php8.1_8.1.16+repack-1+0~20230214.36+debian10~1.gpbp38498_all.deb ...
Unpacking php8.1 (8.1.16+repack-1+0~20230214.36+debian10~1.gpbp38498) over (8.1.12-1) ...
Setting up php8.1 (8.1.16+repack-1+0~20230214.36+debian10~1.gpbp38498) ...
```

```
(kali㉿kali)-[~]
$ sudo apt install php7.4-{cli,json,imap,bcmath,bz2,intl,gd,mbstring,mysql,zip}
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
php7.4-cli is already the newest version (7.4.21-1+deb11u1).
```

**Step 6:** Configure in apache server and check status

```
; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; http://php.net/allow-url-fopen
allow_url_fopen = On

; Whether to allow include/require to open URLs (like http:// or ftp://) as files
; http://php.net/allow-url-include
allow_url_include = On
```

```
└$ systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: disabled)
  Active: active (running) since Wed 2023-02-15 03:12:27 EST; 15s ago
    Docs: https://httpd.apache.org/docs/2.4/
   Process: 67914 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 67935 (apache2)
   Tasks: 6 (limit: 4584)
  Memory: 18.8M
     CPU: 172ms
    CGroup: /system.slice/apache2.service
            └─67935 /usr/sbin/apache2 -k start
              ├─67937 /usr/sbin/apache2 -k start
              ├─67938 /usr/sbin/apache2 -k start
              ├─67939 /usr/sbin/apache2 -k start
              ├─67940 /usr/sbin/apache2 -k start
              └─67941 /usr/sbin/apache2 -k start

Feb 15 03:12:27 kali systemd[1]: Starting The Apache HTTP Server ...
Feb 15 03:12:27 kali apachectl[67934]: AH00558: apache2: Could not reliably determine the server's >
Feb 15 03:12:27 kali systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)
```

**Step 7:** Access DVWA on Your Browser

Open firefox and search: <http://127.0.0.1/dvwa/setup.php> which will open home page for dvwa.

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

### General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerability with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

**Conclusion/Summary:**

<b>Student Signature &amp; Date</b>	<b>Marks:</b>	<b>Evaluator Signature &amp; Date</b>
-------------------------------------	---------------	---------------------------------------

## Practical 8

Date: / /2023

**Aim:** Gather information of any domain/website/IP address using following Information Gathering Tools.

1. Samspade
2. Nslookup
3. Whois
4. Tracert

**Solution:**

### 1) Samspade

Systems and security administrators have a number of useful tools at their disposal to obtain information about computers attached to other networks on the Internet, as well as information about the Internet itself. Ping, traceroute, whois and nslookup are among the essential utilities for even rudimentary maintenance and testing. But the native Windows environment includes only a few of these tools and they are, by and large, individual command line utilities and one has to go to third parties to obtain many of the missing utilities. Sam Spade is a nice piece of software that combines many of these common tools -- and several more uncommon ones -- into a single, integrated, Windows-compatible package.

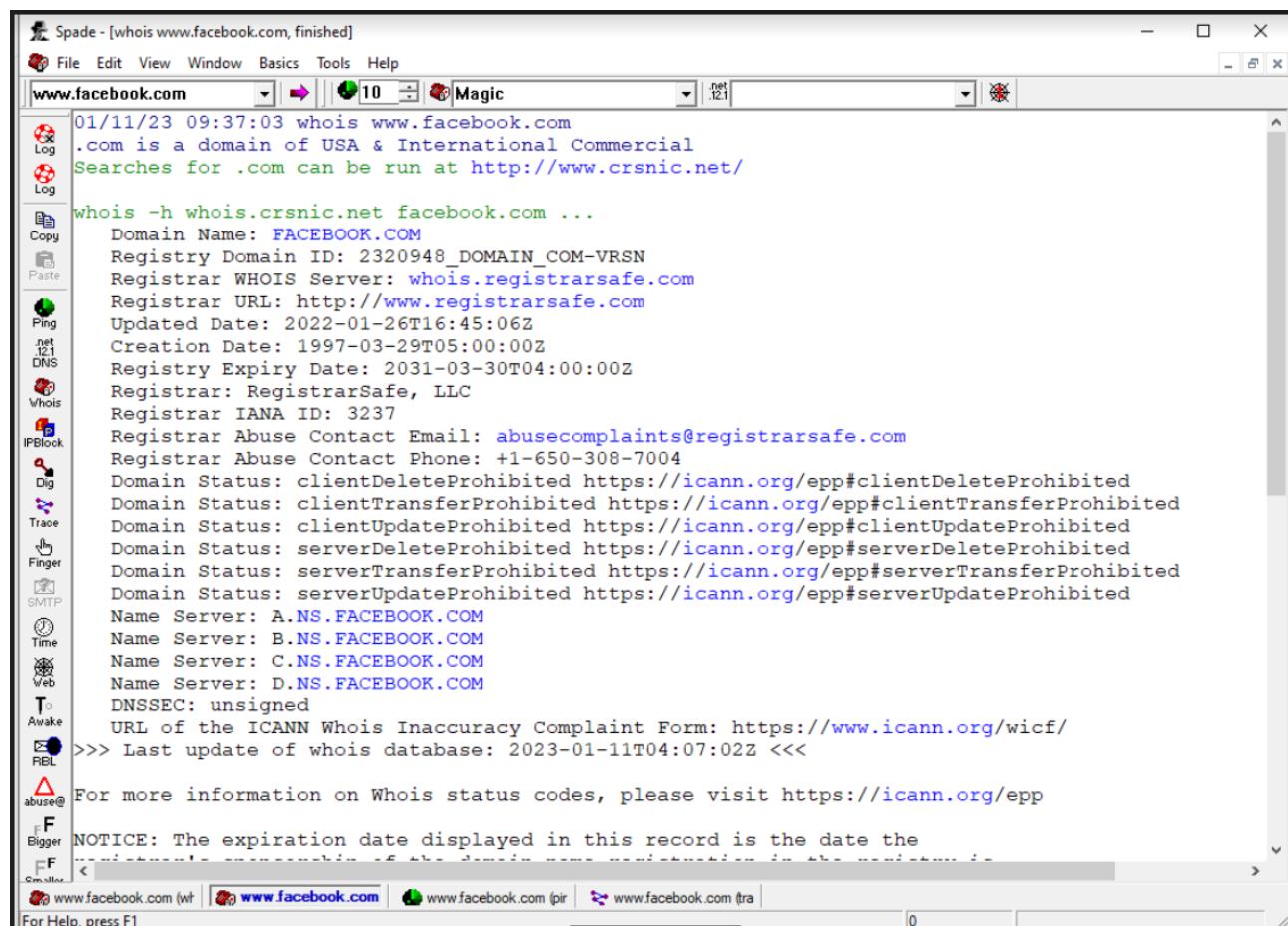
Sam Spade's utilities allow the user to look up information about a remote host or domain, generally for the purpose of initial reconnaissance or forensic analysis:

- *Ping* sends a series of packets to the indicated host to determine if that system is reachable via the network and provides an estimate of the round trip packet time.
- *Traceroute* traces the route that packets take from the user's system to the specified target host address, listing all intermediate routers and showing a graph of the hop-by-hop delay times. Fast and slow traceroute differ only in the number of attempts made to learn the route.
- *Nslookup* and *Decode URL* display the IP address and name of a specified host. This can help an investigator learn about the owner of a system from the domain name or obtain an IP address with which to further investigate the geographic location of a system.
- *Whois* provides ownership and contact information for the specified host's domain. This tool is increasingly convenient as the number of domain name registrars grows. When Network Solutions was the sole registrar for .com, for example, their whois database was the only one

you needed to search. With about 100 accredited registrars today, you have to do a search just to find out which registrar to lookup. Sam Spade's whois function does this for you.

- *IP Block* indicates the owner of the IP address block to which the specified host belongs. By identifying the owner of an address block, you can start to narrow down where a host is geographically located and/or learn about the host's upstream Internet service provider (ISP).
- *DIG (Domain Internet Groper)*, like nslookup, looks up DNS information. Sam Spade's DIG function returns all DNS records associated with a specified host or domain, including the start of authority (SOA), mail exchange (MX) and name server (NS) records. This information allows the user to determine where to send e-mail to a host's domain and how to access the manager of the domain's name space.
- *Zone Transfer* is used to request that a DNS server send all of the information that it has about a given domain. Properly configured DNS servers will not comply with this request as a security precaution, but it will work surprisingly often. This is a great way to test your own name servers.
- *Finger* obtains host/user information from a system running the finger daemon (TCP port 79). Finger is generally (or should be) disabled at a host because it can give an attacker a lot of information about users and/or the host itself, but it isn't always turned off.

## Samspade – who.is



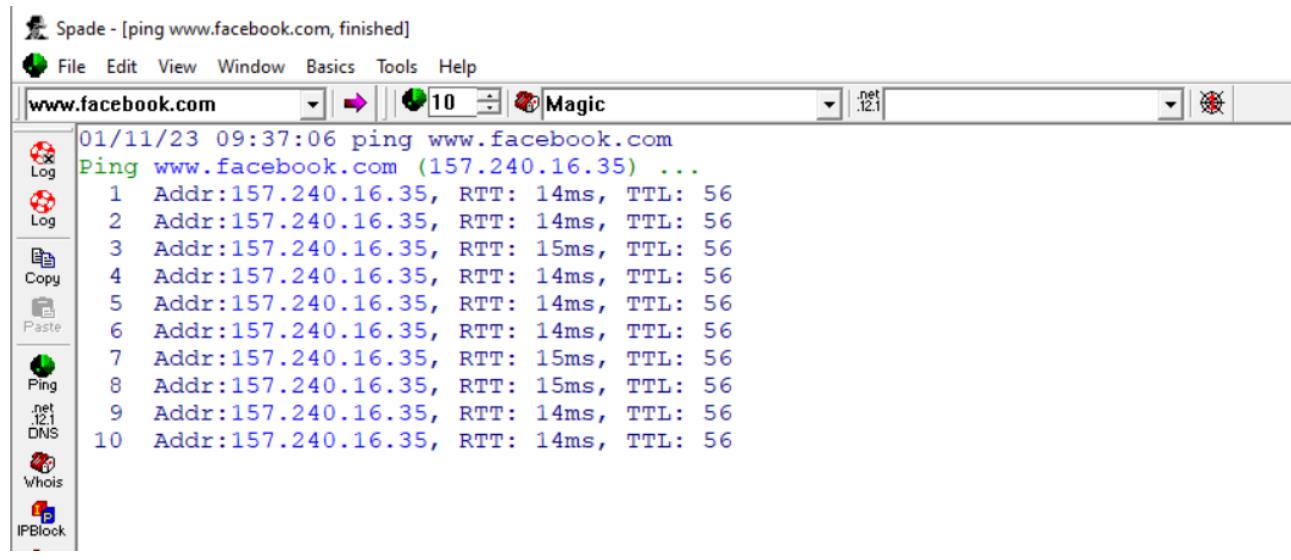
Spade - [whois www.facebook.com, finished]

File Edit View Window Basics Tools Help

www.facebook.com | 10 | Magic | net | .12.1 | X

```
01/11/23 09:37:03 whois www.facebook.com
.com is a domain of USA & International Commercial
Searches for .com can be run at http://www.crsnic.net/
whois -h whois.crsnic.net facebook.com ...
Domain Name: FACEBOOK.COM
Registry Domain ID: 2320948_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.registrarsafe.com
Registrar URL: http://www.registrarsafe.com
Updated Date: 2022-01-26T16:45:06Z
Creation Date: 1997-03-29T05:00:00Z
Registry Expiry Date: 2031-03-30T04:00:00Z
Registrar: RegistrarSafe, LLC
Registrar IANA ID: 3237
Registrar Abuse Contact Email: abusecomplaints@registrarsafe.com
Registrar Abuse Contact Phone: +1-650-308-7004
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: A.NS.FACEBOOK.COM
Name Server: B.NS.FACEBOOK.COM
Name Server: C.NS.FACEBOOK.COM
Name Server: D.NS.FACEBOOK.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2023-01-11T04:07:02Z <<<
For more information on Whois status codes, please visit https://icann.org/epp
NOTICE: The expiration date displayed in this record is the date the
domain's registration was last updated in the database by its
registrar. The domain administrator is responsible for maintaining the
record's accuracy. We do not guarantee its completeness or timeliness.
For Help, press F1
```

## Samspade – ping



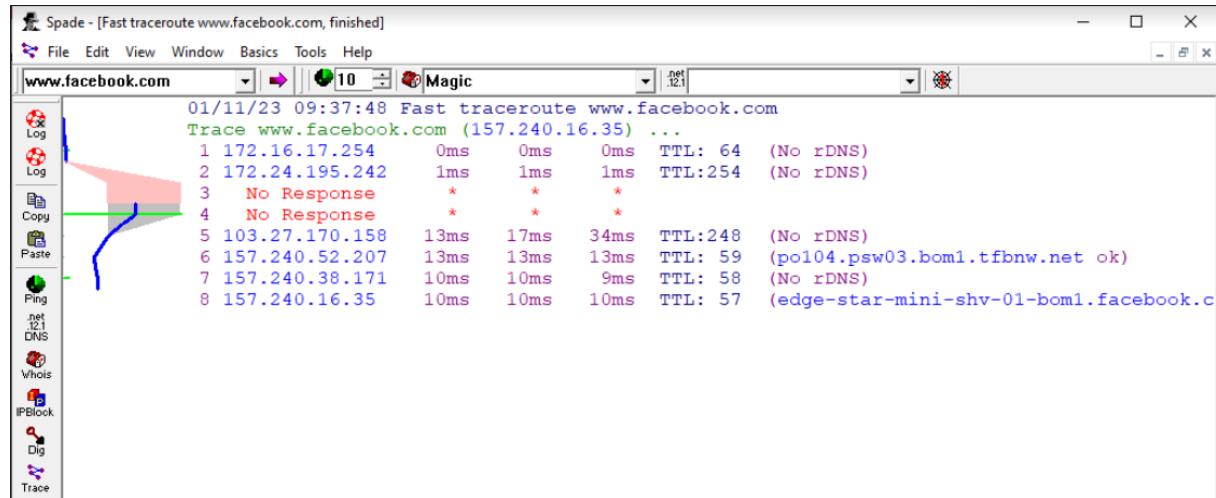
Spade - [ping www.facebook.com, finished]

File Edit View Window Basics Tools Help

www.facebook.com | 10 | Magic | net | .12.1 | X

```
01/11/23 09:37:06 ping www.facebook.com
Ping www.facebook.com (157.240.16.35) ...
1 Addr:157.240.16.35, RTT: 14ms, TTL: 56
2 Addr:157.240.16.35, RTT: 14ms, TTL: 56
3 Addr:157.240.16.35, RTT: 15ms, TTL: 56
4 Addr:157.240.16.35, RTT: 14ms, TTL: 56
5 Addr:157.240.16.35, RTT: 14ms, TTL: 56
6 Addr:157.240.16.35, RTT: 14ms, TTL: 56
7 Addr:157.240.16.35, RTT: 15ms, TTL: 56
8 Addr:157.240.16.35, RTT: 15ms, TTL: 56
9 Addr:157.240.16.35, RTT: 14ms, TTL: 56
10 Addr:157.240.16.35, RTT: 14ms, TTL: 56
```

### Samsrade – tracer



### Samespade – addresses



## 2) NsLookup

nslookup (from name server lookup) is a network administration command-line tool for querying the Domain Name System (DNS) to obtain the mapping between domain name and IP address, or other DNS records.

DNS records for **www.amazon.in**

Cloudflare Google DNS OpenDNS Authoritative Local DNS 

The Cloudflare DNS server responded with these DNS records. Cloudflare will serve these records for as long as the time to live (TTL) has not expired. After this period, Cloudflare will update its cache by querying one of the authoritative name servers.

**A records**

IPv4 address	Revalidate in
a 108.138.240.25	
> For CNAME d1elgm1ww0dfwo.cloudflare.net ~ tp.c95e7ef602-frontier.amazon.in.	1m

**AAAA records**

IPv6 address	Revalidate in
a 2600:9000:2146:6800:8:b109:e12:2281	
> For CNAME d1elgm1ww0dfwo.cloudflare.net ~ tp.c95e7ef602-frontier.amazon.in.	8s
a 2600:9000:2146:fa00:8:b109:e12:2281	
> For CNAME d1elgm1ww0dfwo.cloudflare.net ~ tp.c95e7ef602-frontier.amazon.in.	8s
a 2600:9000:2146:e800:8:b109:e12:2281	
> For CNAME d1elgm1ww0dfwo.cloudflare.net ~ tp.c95e7ef602-frontier.amazon.in.	8s
a 2600:9000:2146:ce00:8:b109:e12:2281	
> For CNAME d1elgm1ww0dfwo.cloudflare.net ~ tp.c95e7ef602-frontier.amazon.in.	8s
a 2600:9000:2146:7e00:8:b109:e12:2281	

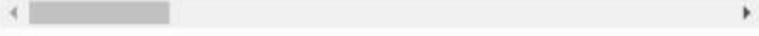


Acrobat Pro DC  
Acrobat's got it.  
Try free

**Whois IP 172.16.17.4**

Updated 6 days ago

```
#  
# ARIN WHOIS data and services are subject to the Terms of Use  
# available at: https://www.arin.net/resources/registry/whois/tou/  
#  
# If you see inaccuracies in the results, please report at  
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/  
#  
# Copyright 1997-2023, American Registry for Internet Numbers, Ltd.  
#  
  
NetRange:      172.16.0.0 - 172.31.255.255  
CIDR:         172.16.0.0/12  
NetName:       PRIVATE-ADDRESS-BBLK-RFC1918-IANA-RESERVED  
NetHandle:     NET-172-16-0-0-1  
Parent:        NET172 (NET-172-0-0-0-0)  
NetType:       IANA Special Use  
OriginAS:  
Organization:  Internet Assigned Numbers Authority (IANA)  
RegDate:      1994-03-15  
Updated:       2013-08-30  
Comment:       These addresses are in use by many millions of independently  
Comment:       These addresses can be used by anyone without any need to coo  
Comment:       These addresses were assigned by the IETF, the organization t  
Comment:       http://datatracker.ietf.org/doc/rfc1918  
Comment:       Ref: https://rdap.arin.net/registry/ip/172.16.0.0  
  
  
OrgName:      Internet Assigned Numbers Authority  
OrgId:        IANA  
Address:      12025 Waterfront Drive  
Address:      Suite 300  
City:          Los Angeles  
StateProv:    CA  
PostalCode:   90292  
Country:      US  
RegDate:  
Updated:      2012-08-31  
Ref:          https://rdap.arin.net/registry/entity/IANA  
  
  
OrgAbuseHandle: IANA-IP-ARIN  
OrgAbuseName:  ICANN  
OrgAbusePhone: +1-310-301-5820  
OrgAbuseEmail: abuse@iana.org  
OrgAbuseRef:   https://rdap.arin.net/registry/entity/IANA-IP-ARIN  
  
OrgTechHandle: IANA-IP-ARIN  
OrgTechName:  ICANN  
OrgTechPhone: +1-310-301-5820  
OrgTechEmail: abuse@iana.org  
OrgTechRef:   https://rdap.arin.net/registry/entity/IANA-IP-ARIN  
  
#  
# ARIN WHOIS data and services are subject to the Terms of Use  
# available at: https://www.arin.net/resources/registry/whois/tou/  
#  
# If you see inaccuracies in the results, please report at  
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/  
#  
# Copyright 1997-2023, American Registry for Internet Numbers, Ltd.  
#
```



### 3) whois

Website Information - Search the whois database, look up domain and IP owner information, and check out dozens of other statistics.

On Demand Domain Data- Get all the data you need about a domain and everything associated with that domain anytime with a single search.

Domain names – Register your favorite domain name

136.233.130.146 address profile

Whois Diagnostics

IP Whois cache expires in 23 hours, 41 minutes and 21 seconds

```
NetRange: 136.232.0.0 - 136.233.255.255
CIDR: 136.232.0.0/15
NetName: APNIC
NetHandle: NET-136-232-0-0-1
Parent: NET136 (NET-136-0-0-0-0)
NetType: Early Registrations, Transferred to APNIC
OriginAS:
Organization: Asia Pacific Network Information Centre (APNIC)
RegDate: 2016-11-01
Updated: 2016-11-01
Ref: https://rdap.arin.net/registry/ip/136.232.0.0

ResourceLink: http://wq.apnic.net/whois-search/static/search.html
ResourceLink: whois://whois.apnic.net

OrgName: Asia Pacific Network Information Centre
OrgId: APNIC
Address: PO Box 3646
City: South Brisbane
StateProv: QLD
PostalCode: 4101
Country: AU
RegDate:
Updated: 2012-01-24
Ref: https://rdap.arin.net/registry/entity/APNIC

ReferralServer: whois://whois.apnic.net
ResourceLink: http://wq.apnic.net/whois-search/static/search.html

OrgAbuseHandle: AWC12-ARIN
OrgAbuseName: APNIC Whois Contact
OrgAbusePhone: +61 7 3858 3188
OrgAbuseEmail: search-apnic-not-arin@apnic.net
OrgAbuseRef: https://rdap.arin.net/registry/entity/AWC12-ARIN

OrgTechHandle: AWC12-ARIN
OrgTechName: APNIC Whois Contact
OrgTechPhone: +61 7 3858 3188
OrgTechEmail: search-apnic-not-arin@apnic.net
OrgTechRef: https://rdap.arin.net/registry/entity/AWC12-ARIN
```

The screenshot shows a diagnostic tool interface for the IP address 136.233.130.146. The interface includes tabs for Whois and Diagnostics, with Diagnostics selected. Under the Diagnostics tab, there are two sections: Ping and Traceroute.

**Ping:**

```
PING 136.233.130.146 (136.233.130.146) 56(84) bytes of data.
--- 136.233.130.146 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4081ms
```

**Traceroute:**

```
traceroute to 136.233.130.146 (136.233.130.146), 30 hops max, 60 byte packets
1 ip-10-0-0-14.ec2.internal (10.0.0.14) 0.736 ms 0.721 ms 0.728 ms
2 216.182.229.174 (216.182.229.174) 8.076 ms 216.182.238.159 (216.182.238.159) 4.456 ms 216.182.231.48 (216.182.231.48) 20.701 ms
3 100.65.83.224 (100.65.83.224) 7.489 ms 100.66.9.226 (100.66.9.226) 19.381 ms 100.65.128.64 (100.65.128.64) 8.882 ms
4 100.66.38.4 (100.66.38.4) 13.036 ms 100.66.14.146 (100.66.14.146) 19.132 ms 100.66.11.198 (100.66.11.198) 18.060 ms
5 100.66.63.140 (100.66.63.140) 229.948 ms 100.66.42.176 (100.66.42.176) 15.141 ms 241.0.4.211 (241.0.4.211) 1.413 ms
6 248.0.40.19 (248.0.40.19) 1.371 ms 241.0.4.212 (241.0.4.212) 1.174 ms 248.0.40.27 (248.0.40.27) 1.211 ms
7 248.0.40.27 (248.0.40.27) 1.230 ms 248.0.40.16 (248.0.40.16) 1.247 ms 248.0.40.28 (248.0.40.28) 1.228 ms
8 242.0.171.145 (242.0.171.145) 5.374 ms 242.0.171.1 (242.0.171.1) 1.114 ms 242.0.171.17 (242.0.171.17) 8.591 ms
9 52.93.28.187 (52.93.28.187) 1.819 ms 242.0.171.145 (242.0.171.145) 1.879 ms 242.0.170.145 (242.0.170.145) 2.109 ms
```

**4) Tracert**

- The TRACERT diagnostic utility determines the route to a destination by sending Internet Control Message Protocol (ICMP) echo packets to the destination. In these packets, TRACERT uses varying IP Time-To-Live (TTL) values. Because each router along the path is required to decrement the packet's TTL by at least 1 before forwarding the packet, the TTL is effectively a hop counter. When the TTL on a packet reaches zero (0), the router sends an ICMP "Time Exceeded" message back to the source computer.
- TRACERT sends the first echo packet with a TTL of 1 and increments the TTL by 1 on each subsequent transmission, until the destination responds or until the maximum TTL is reached. The ICMP "Time Exceeded" messages that intermediate routers send back show the route. Note however that some routers silently drop packets that have expired TTLs, and these packets are invisible to TRACERT.

```
C:\Users\Administrator>tracert 172.16.17.4
Tracing route to host.docker.internal [172.16.17.4]
over a maximum of 30 hops:
1 <1 ms <1 ms <1 ms host.docker.internal [172.16.17.4]

Trace complete.

C:\Users\Administrator>tracert www.amazon.in
Tracing route to d1elgm1ww0d6wo.cloudfront.net [18.66.54.214]
over a maximum of 30 hops:
1 <1 ms <1 ms <1 ms 172.16.17.254
2 4 ms 3 ms 3 ms 136.232.113.77.static.jio.com [136.232.113.77]
3 6 ms 6 ms 8 ms 10.67.49.2
4 5 ms 4 ms 4 ms 172.18.32.130
5 10 ms 9 ms 12 ms 10.63.94.141
6 6 ms 6 ms 6 ms 10.81.155.111
7 9 ms 10 ms 13 ms 10.81.155.108
8 6 ms 6 ms 6 ms 10.67.59.163
9 10 ms 6 ms 5 ms 10.83.244.197
10 5 ms 5 ms 4 ms 172.29.203.69
11 * * * Request timed out.
12 * * * Request timed out.
13 * * * Request timed out.
14 * * * Request timed out.
15 13 ms 13 ms 13 ms server-18-66-54-214.bom78.r.cloudfront.net [18.66.54.214]
```

**Conclusion/Summary:**

<b>Student Signature &amp; Date</b>	<b>Marks:</b>	<b>Evaluator Signature &amp; Date</b>
-------------------------------------	---------------	---------------------------------------

## Practical 9

**Date:** / /2023

**Aim:** Identify any 5 online web portals for information gathering. Scan an IP address/URL for gathering information. Prepare a report.

**Solution:**

**1) HostedScan**

Online tool for scanning networks, servers, and websites for security risks. To try for free type in the url and your email address, and a detailed report will be sent to your email address.

The screenshot shows the HostedScan homepage. At the top, there's a navigation bar with 'HostedScan' logo, 'Scan Types', 'Pricing', 'Docs', 'Log in', and 'Create Account'. Below the header, there's a large banner with the text 'Online vulnerability scanners, for less' and a subtext 'Scan networks, servers, and websites for security risks. Manage your risks via dashboards, reporting, and alerts.' A 'Try for free' button is visible. In the center, there are three main dashboard panels labeled 'Risks Breakdown', 'Most Recent Risks', and 'Exposure Window'. At the bottom, there's a search bar with 'amazon.in' and a 'Scan Now' button.

Nmap Scan for amazon.in

The screenshot shows an Nmap Scan Report for the IP address 162.219.225.220. The report title is 'Nmap Scan Report - Scanned at Wed Jan 4 05:05:08 2023'. It includes a 'Scan Summary' section with arguments used: 'nmap -v -oX-- --host-timeout=28800s -Pn -T4 -sT --webxml --max-retries=1 --open -p-65355 www.amazon.in'. Verbosity is set to 1. The scan summary states: 'Nmap 7.40 was initiated at Wed Jan 4 05:05:08 2023 with these arguments: nmap -v -oX-- --host-timeout=28800s -Pn -T4 -sT --webxml --max-retries=1 --open -p-65355 www.amazon.in'. The scan took 89.62 seconds. The report then lists the target address '162.219.225.220 / www.amazon.in' and provides details about hostnames ('www.amazon.in (user)'), ports ('Ports'), and a table of open ports.

Port	State (toggle closed [0]   filtered [0])	Service	Reason	Product	Version	Extra info
80	tcp open	http	syn-ack			
443	tcp open	https	syn-ack			

## Report summary

### Report Summary

This report contains information on all of the risks found from your vulnerability scans. Each risk is assigned a threat level (high, medium, or low).

#### Total Risks

Total number of risks found by severity.

0  
High

2  
Medium

0  
Low

0  
Accepted

### Open TCP Ports

The NMAP TCP port scan discovers open ports on with a complete scan of ports 0 to 65535.

#### Total Risks

Total number of risks found by the TCP port scan.

0  
High

2  
Medium

0  
Low

0  
Accepted

#### Open Ports Summary

Summary of detected open ports.

Threat Level	Title	Open Count	Accepted Count
MEDIUM	<a href="#">Open TCP Port: 443</a>	1	0
MEDIUM	<a href="#">Open TCP Port: 80</a>	1	0

## 2) Port scanners

Port Scanner is an application that is used to determine the open ports on the network. Port scanning is performed to get information about open ports that are ready to receive information.

Port Scanning is a five-step process as described below.

Step 1: For port scanning, there is a need for active hosts. Active hosts can be discovered using the network scanning process.

Step 2: These active hosts are mapped to their IP addresses.

Step 3: Now we have active hosts and thus port scanning process is performed. In this process, packets are sent to specific ports on a host.

Step 4: Here responses will get analyzed.

Step 5: Through this analysis, information about running services will be learned and potential vulnerabilities will be identified.

### 2.1) Nmap Online port scanner:

Here we have searched for open ports for amazon.in using its IP address.

The screenshot shows a web-based Nmap scan interface. At the top, the IP address "52.95.116.115" is entered. Below it is a teal button labeled "QUICK NMAP SCAN". The main area displays the scan results:

```
Starting Nmap 7.40 ( https://nmap.org ) at 2023-01-04 04:31 UTC
Nmap scan report for 52.95.116.115
Host is up (0.080s latency).

PORT      STATE SERVICE
21/tcp    closed  ftp
22/tcp    closed  ssh
23/tcp    closed  telnet
80/tcp    open   http
110/tcp   closed pop3
143/tcp   closed imap
443/tcp   open   https
3389/tcp closed ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
```

### 3) Zenmap:

- Zenmap is the official Nmap Security Scanner GUI. It is a multi-platform (Linux, Windows, Mac OS X, BSD, etc.) free and open source application which aims to make Nmap easy for beginners to use while providing advanced features for experienced Nmap users. Frequently used scans can be saved as profiles to make them easy to run repeatedly. A command creator allows interactive creation of Nmap command lines. Scan results can be saved and viewed later. Saved scan results can be compared with one another to see how they differ. The results of recent scans are stored in a searchable database.

The screenshot shows the Zenmap application window. At the top, there is a target field containing "142.250.192.36" and a command field containing "nmap -T4 -A -v 142.250.192.36". To the right of the command field are buttons for "Profile" and "Intense scan". Below the command field is a toolbar with tabs: Hosts (which is selected), Services, Nmap Output, Ports / Hosts, Topology, Host Details, and Scans. Under the Hosts tab, there is a tree view showing "OS < Host" and "bom12s15-in-f4.1e". The main pane displays the Nmap scan output. The output starts with "Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-04 10:52 India Standard Time" and continues with various log messages related to the scan process, including NSE script execution, service detection, and port scanning details. The output concludes with a detailed service fingerprint for port 80/tcp, showing the HTTP response headers and body.

```

NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:52
Completed NSE at 10:52, 0.00s elapsed
Initiating NSE at 10:52
Completed NSE at 10:52, 0.00s elapsed
Initiating NSE at 10:52
Completed NSE at 10:52, 0.00s elapsed
Initiating Ping Scan at 10:52
Scanning 142.250.192.36 [4 ports]
Completed Ping Scan at 10:52, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:52
Completed Parallel DNS resolution of 1 host. at 10:52, 0.01s elapsed
Initiating SYN Stealth Scan at 10:52
Scanning bom12s15-in-f4.1e100.net (142.250.192.36) [1000 ports]
Discovered open port 443/tcp on 142.250.192.36
Discovered open port 80/tcp on 142.250.192.36
Completed SYN Stealth Scan at 10:52, 4.78s elapsed (1000 total ports)
Initiating Service scan at 10:52
Scanning 2 services on bom12s15-in-f4.1e100.net (142.250.192.36)
Service scan Timing: About 50.00% done; ETC: 10:54 (0:00:39 remaining)
Completed Service scan at 10:54, 62.14s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against bom12s15-in-f4.1e100.net (142.250.192.36)
Initiating Traceroute at 10:54
Completed Traceroute at 10:54, 0.02s elapsed
NSE: Script scanning 142.250.192.36.
Initiating NSE at 10:54
Completed NSE at 10:54, 19.73s elapsed
Initiating NSE at 10:54
Completed NSE at 10:54, 1.88s elapsed
Initiating NSE at 10:54
Completed NSE at 10:54, 0.08s elapsed
Nmap scan report for bom12s15-in-f4.1e100.net (142.250.192.36)
Host is up (0.00087s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http
|_fingerprint-strings:
|   HTTPOptions:
|     HTTP/1.1 200 OK
|     Date: Wed, 04 Jan 2023 05:23:09 GMT
|     Cache-Control: no-cache
|     Pragma: no-cache
|     Content-Type: text/html; charset=UTF-8"
|     Content-Length: 714
|     Via: HTTP/1.1 forward.http.proxy:3128
|     Connection: close
|     <HTML><HEAD>
|     <meta http-equiv=pragma content=nocache>
|     <META HTTP-EQUIV=Expires CONTENT=-1>
|     <SCRIPT>
|     DOMParser().parseFromString('<a></a>', 'text/xml').documentElement.textContent;
|     window.onload=function() {

```

On Scanning we found two open ports and their information

Port1

## Port 2

Once the Scan is over

4) [whatismyip.com](http://whatismyip.com)

An online tool used to find Ip address of any URL or your own Ip address.

Domain URL: <https://in.pinterest.com/>

**Lookup**

IPv4 Address for in.pinterest.com

✓ Domain Server IP: [23.203.100.196](#)

### 5) IP scanner- whois:

Whois is a widely used Internet record listing that identifies who owns a domain and how to get in contact with them. Here for example we have searched for domain name <https://www.charusat.ac.in/>. It displays domain information like the register date, update date, registrar provider etc..

The screenshot shows the Whois search interface with the domain `charusat.ac.in` entered. The results are displayed under the **Domain Information** section.

Attribute	Value
Domain:	<code>charusat.ac.in</code>
Registrar:	ERNET India
Registered On:	2009-06-03
Expires On:	2028-06-03
Updated On:	2019-08-21
Status:	OK
Name Servers:	<ul style="list-style-type: none"> <li>ns2.infinity.domains</li> <li>ns4.infinity.domains</li> <li>ns1.infinity.domains</li> <li>ns3.infinity.domains</li> </ul>

The screenshot shows the Whois search interface with the domain `charusat.ac.in` entered. Below the domain information, there are three sections: Registrant Contact, Administrative Contact, and Technical Contact, each with an icon and a table.

Contact Type	Organization	Email
Registrant Contact	Charotar University of Science and Technology	Please contact the Registrar listed above
Administrative Contact		Please contact the Registrar listed above
Technical Contact		Please contact the Registrar listed above

**Conclusion/Summary:**

<b>Student Signature &amp; Date</b>	<b>Marks:</b>	<b>Evaluator Signature &amp; Date</b>
-------------------------------------	---------------	---------------------------------------

## Practical 10

Date: / /2023

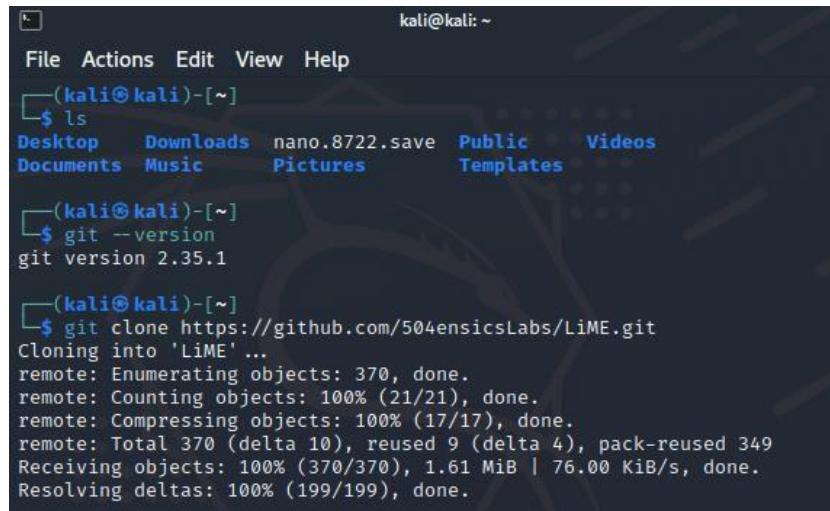
**Aim:** Perform Live / Memory Analysis on a Linux OS and prepare a detailed report.

### Theory:

- Memory forensics is a way to find and extract this valuable information from memory. Volatility is an open-source tool that uses plugins to process this type of information. However, there's a problem: Before you can process this information, you must dump the physical memory into a file, and Volatility does not have this ability.

### Implementation:

**Step 1:** Download from <https://github.com/504ensicsLabs/LiME>

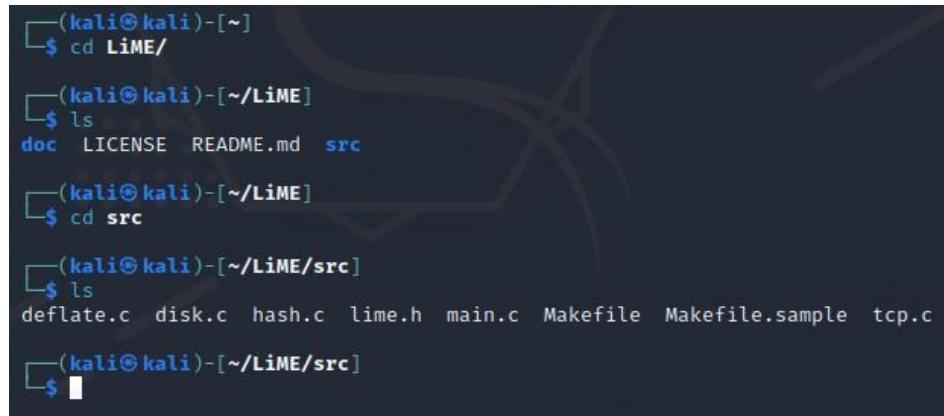


```
kali@kali: ~
File Actions Edit View Help
[(kali㉿kali)-[~]]
$ ls
Desktop Downloads nano.8722.save Public Videos
Documents Music Pictures Templates

[(kali㉿kali)-[~]]
$ git --version
git version 2.35.1

[(kali㉿kali)-[~]]
$ git clone https://github.com/504ensicsLabs/LiME.git
Cloning into 'LiME' ...
remote: Enumerating objects: 370, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 370 (delta 10), reused 9 (delta 4), pack-reused 349
Receiving objects: 100% (370/370), 1.61 MiB | 76.00 KiB/s, done.
Resolving deltas: 100% (199/199), done.
```

**Step 2:** Then, install openssh-server



```
[(kali㉿kali)-[~]]
$ cd LiME/
[(kali㉿kali)-[~/LiME]]
$ ls
doc LICENSE README.md src

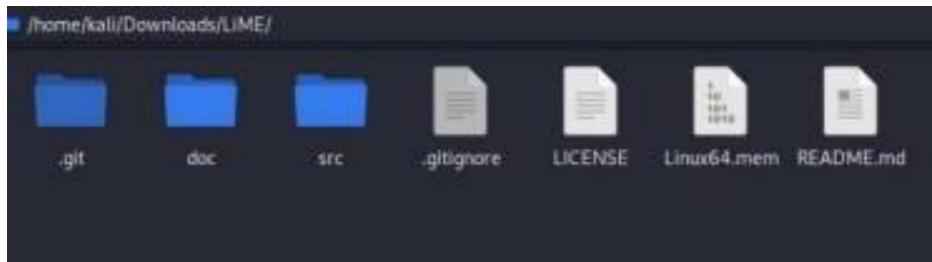
[(kali㉿kali)-[~/LiME]]
$ cd src
[(kali㉿kali)-[~/LiME/src]]
$ ls
deflate.c disk.c hash.c lime.h main.c Makefile Makefile.sample tcp.c

[(kali㉿kali)-[~/LiME/src]]
```

**Step 3:** Start the ssh service and server and Check for the service status

```
(kali㉿kali)-[~/LiME/src]
$ sudo insmod ./lime-5.5.0-kali2-amd64.ko "path= ../Linux64.mem format=raw"
[sudo] password for kali:
```

**Step 4:** Navigate to ssh folder to check the details of the connection.



**Conclusion/Summary:**

Student Signature & Date	Marks:	Evaluator Signature & Date
--------------------------	--------	----------------------------