INT 301 Academic Task 03

Using Open Source Software to Audit :-

Network, System software and Physical security of your system

By-Gurpreet Singh(11903687)

gurpreet.11903687@gmail.com

Introduction

Organizations confront a rising number of security risks that may compromise their networks, software, and physical devices in today's quickly changing digital environment. Maintaining a strong security posture is essential for organisations to safeguard their vital assets and data from potential breaches and assaults. This in-depth security report has been created to examine the organization's present network security, system software security, and physical security while offering takeaways and suggestions for improvement.

Project Objective

This project's main goal is to assess the security architecture of the company, spot gaps and weaknesses, and offer concise, doable advice for improving the overall security posture. The undertaking seeks to

1. Fully evaluate the network security of the firm, finding any potential weaknesses and related with network topology, unrestricted ports, services, and intrusion prevention measures.
2. In order to safeguard the organization's software inventory from known vulnerabilities, malware, and other threats, the system software security should be evaluated. This should be done by performing antivirus scanning, vulnerability assessments, intrusion detection, and system hardening.
3. Determine the system's physical security, making sure that the hardware, add-ons, and data

storage locations are sufficiently shielded from illegal access, theft, or manipulation.

Description

A thorough analysis of the organization's security architecture across network security, system software security, and physical security is provided in the complete security report. The study uses a variety of open-source technologies and security best practises to pinpoint gaps, risks, and vulnerabilities. The results are then organised into a systematic style that provides insightful explanations and doable suggestions for improvement.

Scope

The following topics are covered by this in-depth security report: Network Security:

1. Examining the network infrastructure of the company, including its open ports, services, and intrusion detection and prevention systems.
2. System Software Security: Evaluation of the organization's software inventory, including operating systems, programmes, and other software parts, with an emphasis on antivirus and malware protection, vulnerability analyses, intrusion detection, and system hardening measures.
3. Security controls such as access control systems, video surveillance, safe storage, and other pertinent security measures are evaluated as part of the organization's physical security.

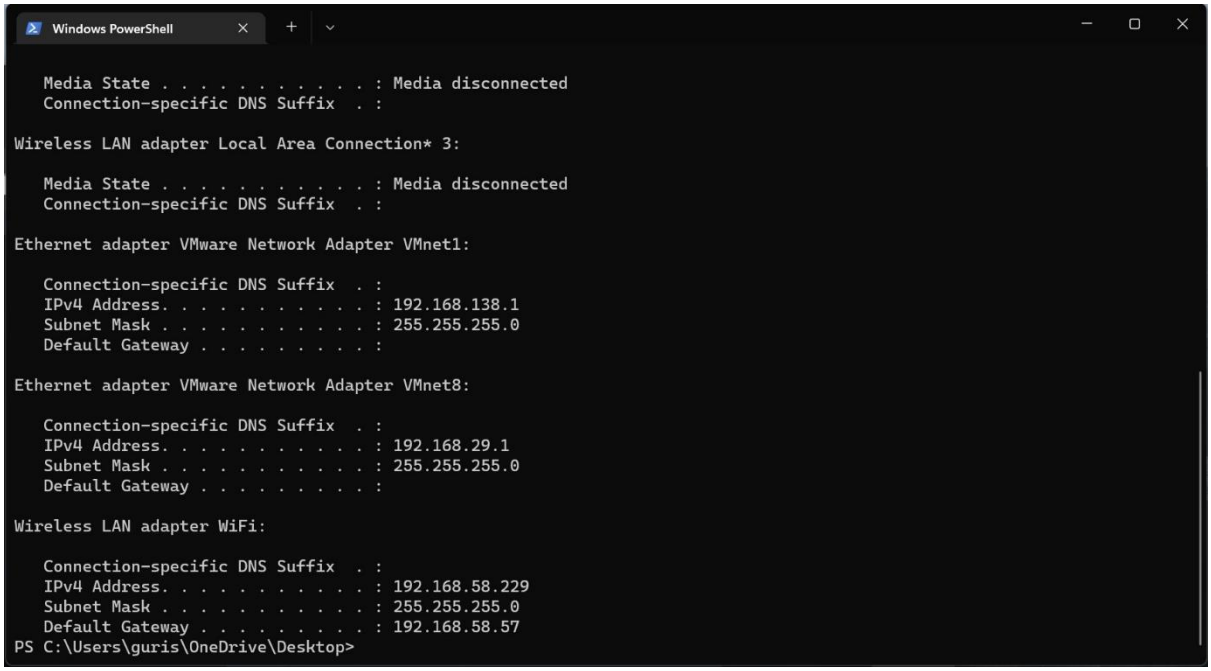
The firm may enhance overall security, better safeguard crucial assets, and reduce the risk of cyber-attacks, data breaches, and physical security incidents by addressing the vulnerabilities and putting the suggested solutions within the purview of this study into practise.

System Description

The organization's network infrastructure, system software, and physical security measures are all included in the target system for the complete security report. Servers, routers, switches, firewalls, and other network equipment make up the network infrastructure. Operating systems, programmes, databases, and other types of software are all included in system software. Access control systems, video monitoring, safe storage, and other pertinent security controls are included in the physical security measures.

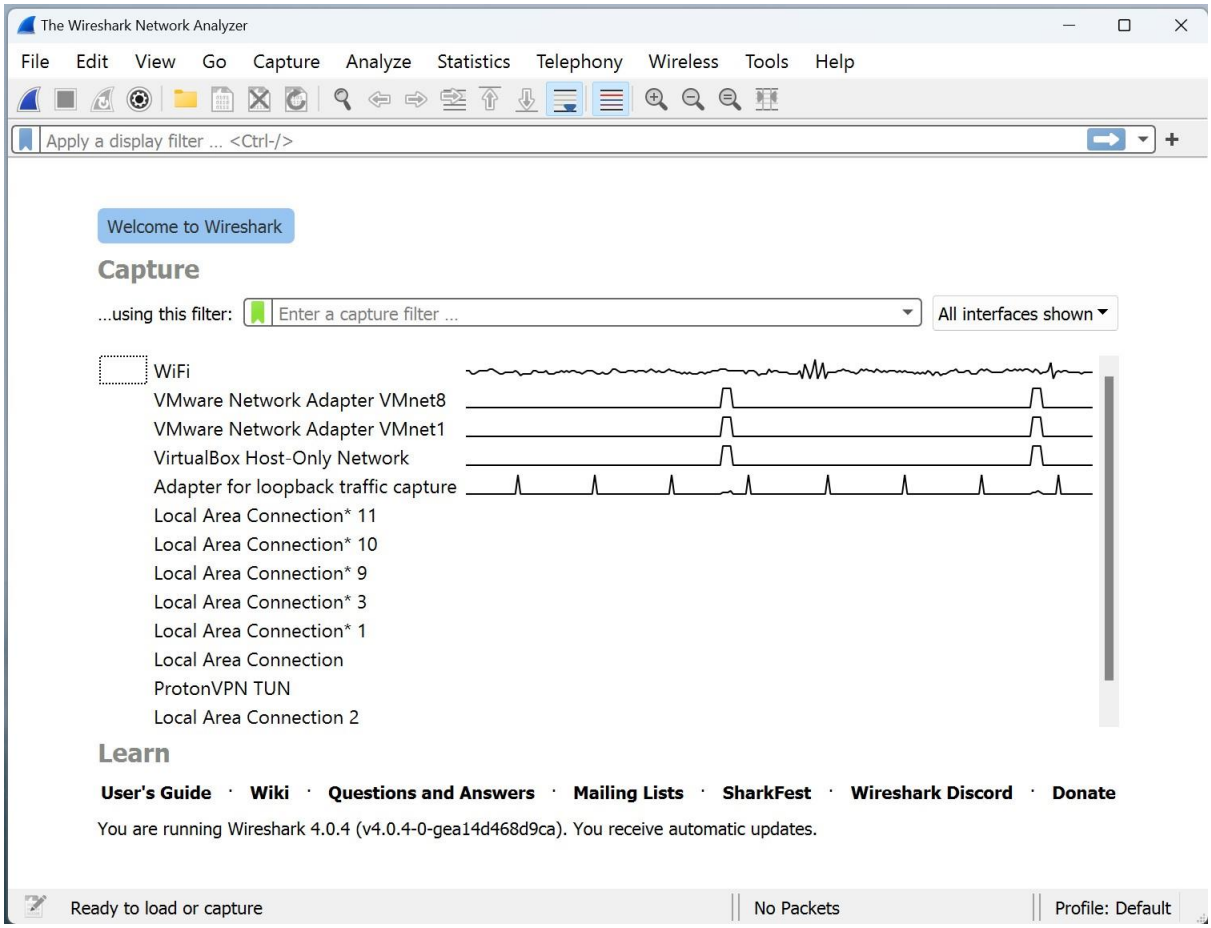
Analysis Report

System snapshots

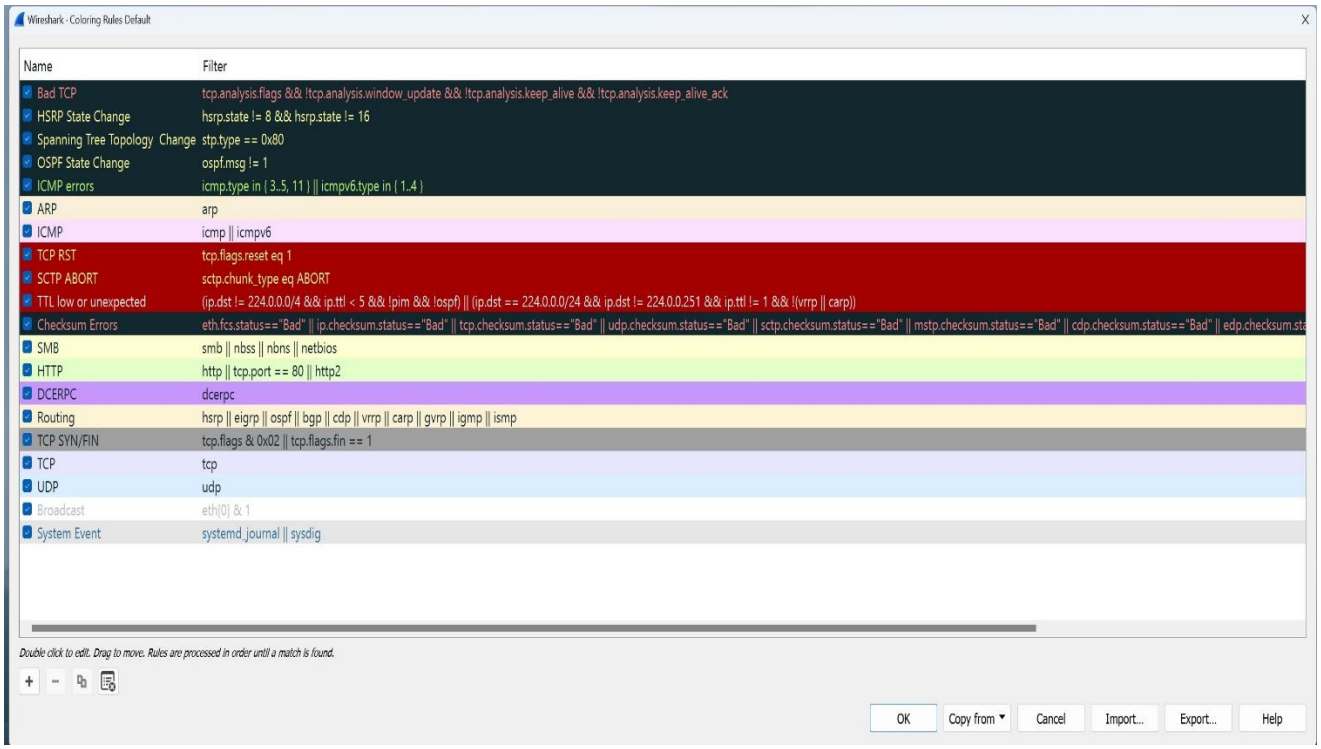


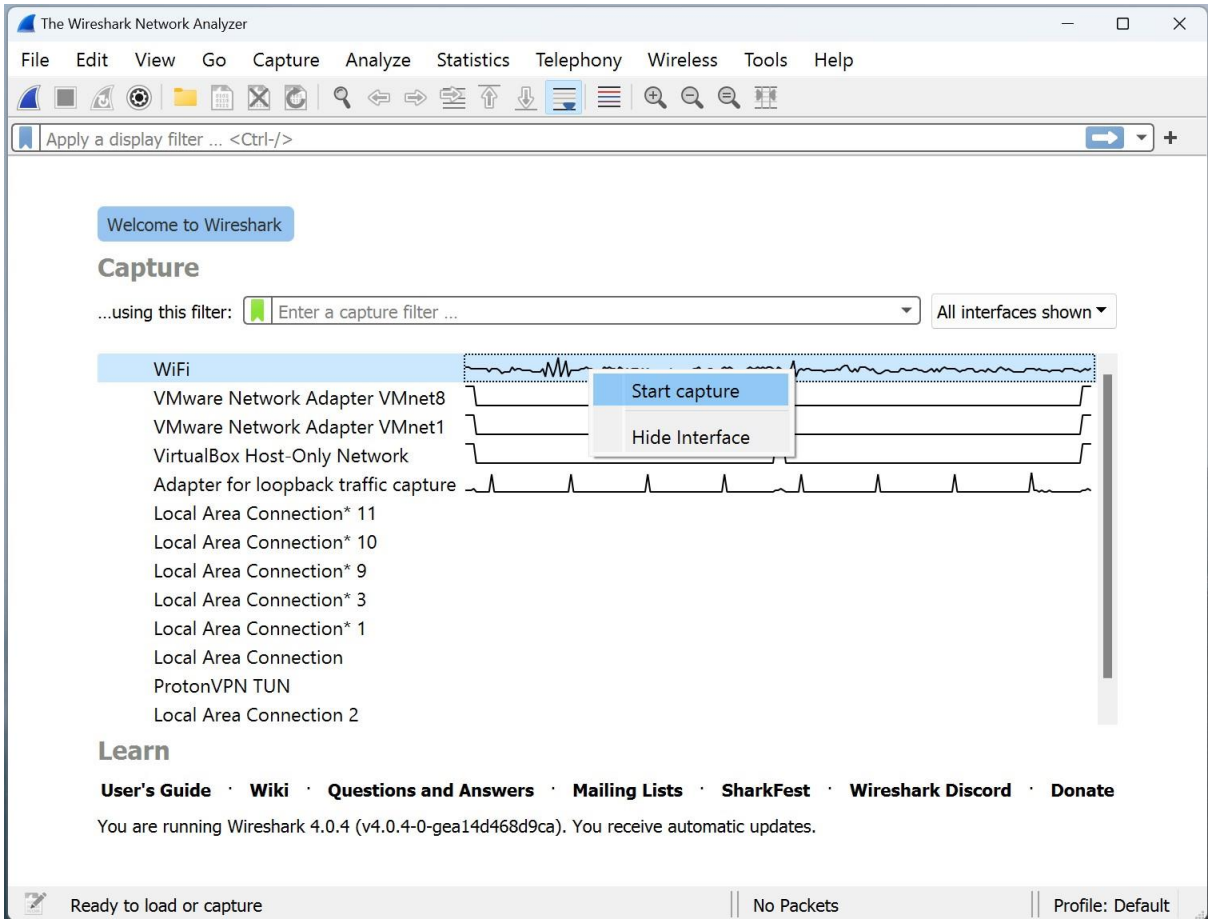
Network security

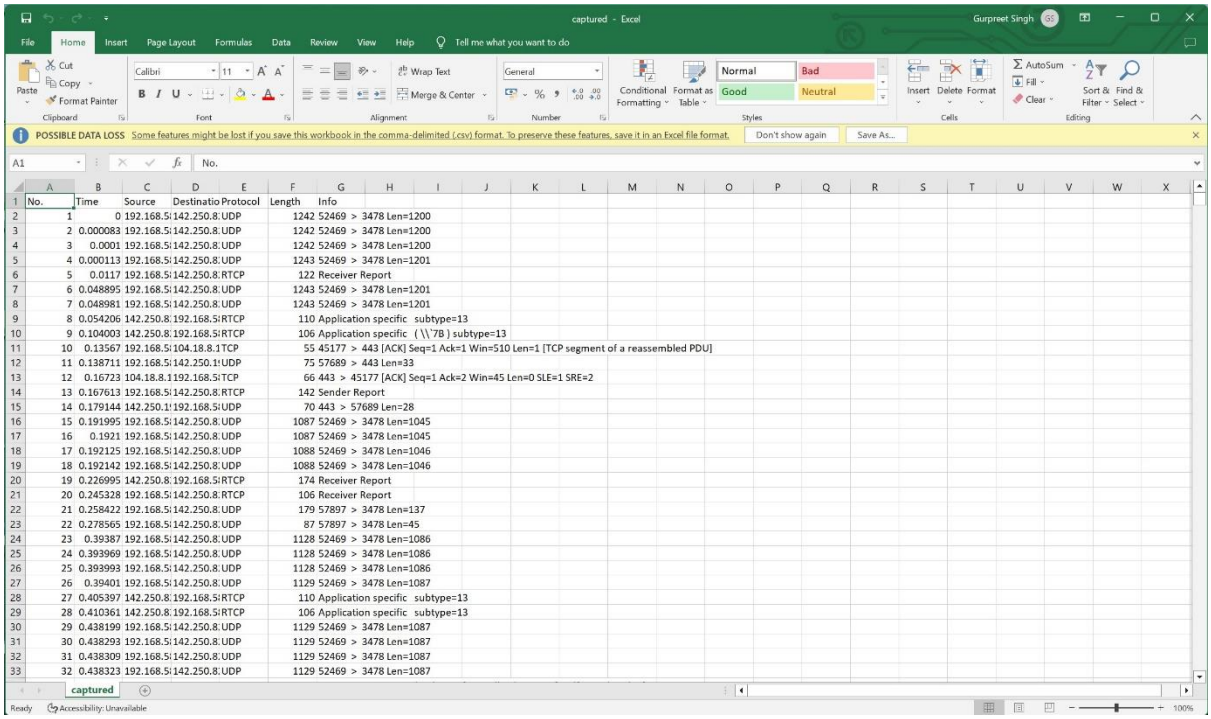
In order to create a report in Wireshark, network traffic must normally be captured, filters must be applied, and data must be exported in a certain format.



Open Wireshark, then choose the network interface from which you wish to collect traffic (e.g., Ethernet, Wi-Fi).







captured - Excel

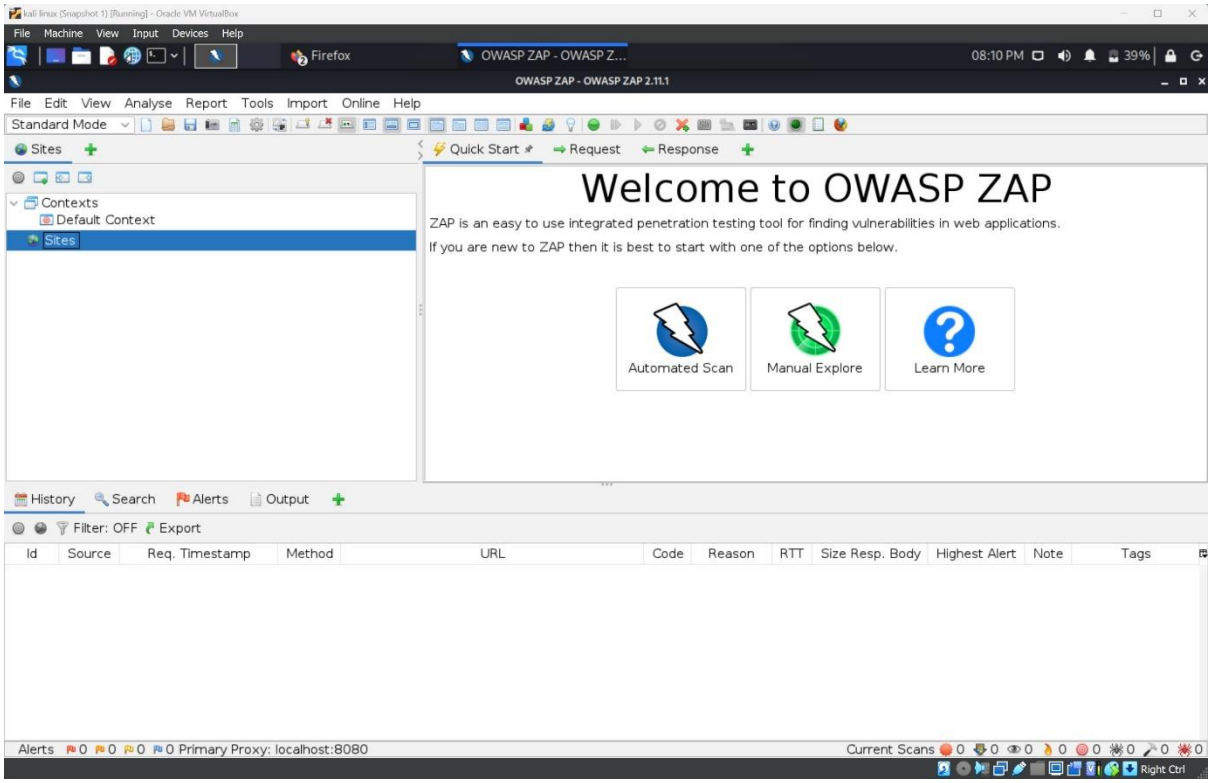
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

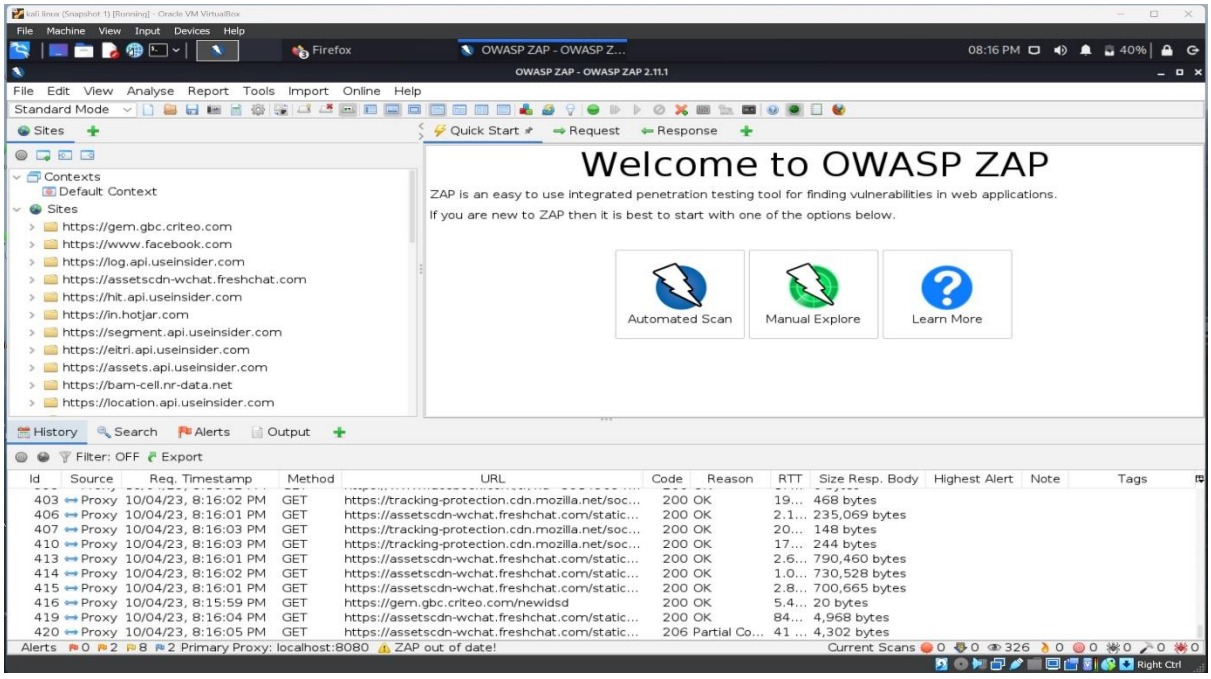
No.	Time	Source	Destinatio	Protocol	Length	Info
1	0	192.168.5	142.250.8	UDP	1242	52469 > 3478 Len=1200
2	0.000083	192.168.5	142.250.8	UDP	1242	52469 > 3478 Len=1200
3	0.0001	192.168.5	142.250.8	UDP	1242	52469 > 3478 Len=1200
4	0.000113	192.168.5	142.250.8	UDP	1243	52469 > 3478 Len=1201
5	0.0117	192.168.5	142.250.8	RTCP	122	Receiver Report
6	0.048895	192.168.5	142.250.8	UDP	1243	52469 > 3478 Len=1201
7	0.048981	192.168.5	142.250.8	UDP	1243	52469 > 3478 Len=1201
8	0.054206	142.250.8	192.168.5	RTCP	110	Application specific subtype=13
9	0.104003	142.250.8	192.168.5	RTCP	106	Application specific (\\`7B) subtype=13
10	0.13567	192.168.5	104.18.8.1	TCP	55	45177 > 443 [ACK] Seq=1 Ack=1 Win=510 Len=1 [TCP segment of a reassembled PDU]
11	0.138711	192.168.5	142.250.1	UDP	75	57689 > 443 Len=33
12	0.16723	104.18.8.1	192.168.5	TCP	66	443 > 45177 [ACK] Seq=1 Ack=2 Win=45 Len=0 SLE=1 SRE=2
13	0.167613	192.168.5	142.250.8	RTCP	142	Sender Report
14	0.179144	142.250.1	192.168.5	UDP	70	443 > 57689 Len=28
15	0.191995	192.168.5	142.250.8	UDP	1087	52469 > 3478 Len=1045
16	0.1921	192.168.5	142.250.8	UDP	1087	52469 > 3478 Len=1045
17	0.192125	192.168.5	142.250.8	UDP	1088	52469 > 3478 Len=1046
18	0.192142	192.168.5	142.250.8	UDP	1088	52469 > 3478 Len=1046
19	0.226995	142.250.8	192.168.5	RTCP	174	Receiver Report
20	0.245328	192.168.5	142.250.8	RTCP	106	Receiver Report
21	0.258422	192.168.5	142.250.8	UDP	179	57897 > 3478 Len=137
22	0.278565	192.168.5	142.250.8	UDP	87	57897 > 3478 Len=45
23	0.39387	192.168.5	142.250.8	UDP	1128	52469 > 3478 Len=1086
24	0.393969	192.168.5	142.250.8	UDP	1128	52469 > 3478 Len=1086
25	0.393993	192.168.5	142.250.8	UDP	1128	52469 > 3478 Len=1086
26	0.39401	192.168.5	142.250.8	UDP	1129	52469 > 3478 Len=1087
27	0.405397	142.250.8	192.168.5	RTCP	110	Application specific subtype=13
28	0.410361	142.250.8	192.168.5	RTCP	106	Application specific subtype=13
29	0.438199	192.168.5	142.250.8	UDP	1129	52469 > 3478 Len=1087
30	0.438293	192.168.5	142.250.8	UDP	1129	52469 > 3478 Len=1087
31	0.438309	192.168.5	142.250.8	UDP	1129	52469 > 3478 Len=1087
32	0.438323	192.168.5	142.250.8	UDP	1129	52469 > 3478 Len=1087

Go to "File" > "Export Packet Dissections" in the main menu to create a report. The following are your choices for exporting the data:

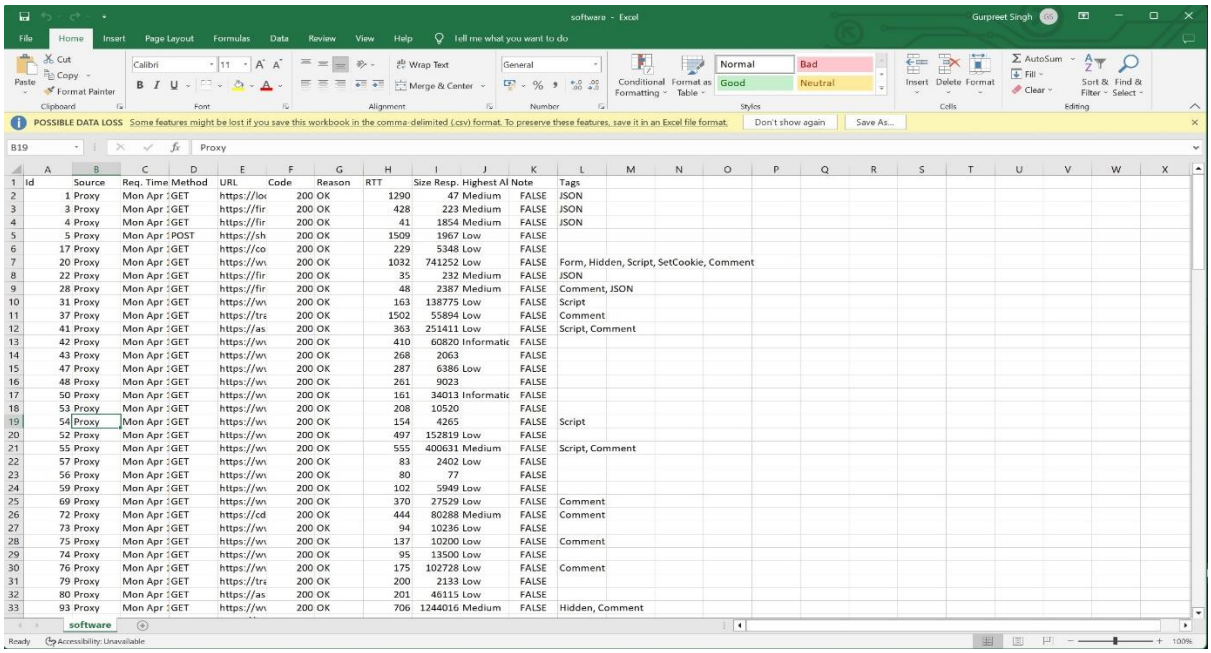
1. Export the data as a plain text file by selecting "As Plain Text."
2. By selecting "As CSV," you may export the data in CSV format.
3. Export the data as C-style arrays by selecting "As C Arrays".
4. Export the data in JSON format by selecting "As JSON."
5. Export the data in Packet Details Markup Language (PDML) format by selecting "As PDML."
6. Export the data in Packet Summary Markup Language (PSML) format by selecting "As PSML."
7. Provide the file name and location for the exported report's file and the chosen export format.
8. To produce the report, click "Save."

Software security



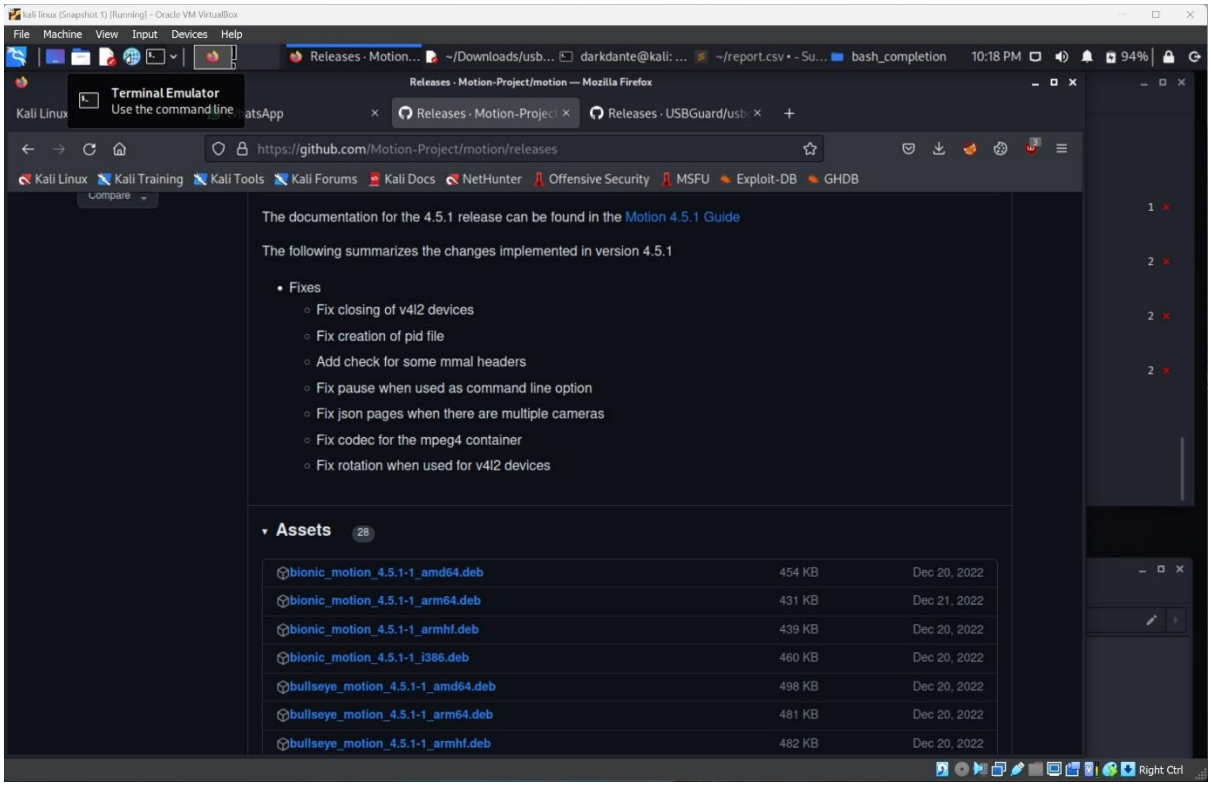


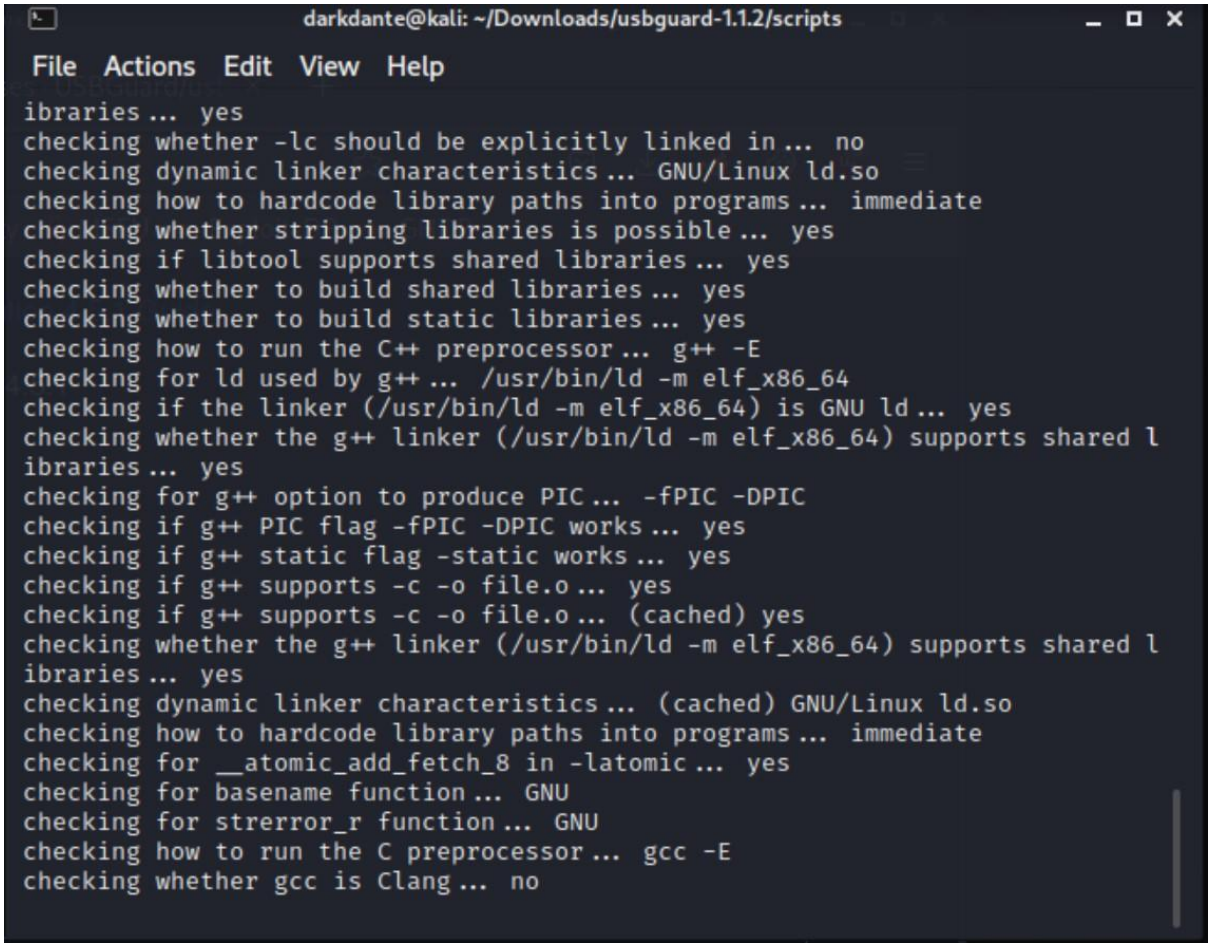
1. Generating a report in OWASP Zed Attack Proxy (ZAP) is straightforward.
2. Run ZAP, then set the target web application up for scanning.
3. Run the target web application through the specified scan (passive or active).
4. The findings of the scanning procedure are presented in the "Alerts" tab once it is finished.
5. Go to the main menu and select "Report" > "Create HTML Report" or "Generate XML Report" to create a report. Instead, you may select "Create MD Report" to generate a report in Markdown format or "Generate JSON Report" to generate a report in JSON format.
6. Give the report file a name and specify the place where you want to save it in the "Save" dialogue box.
7. To produce the report, click "Save."



Id	Source	Req. Time	Method	URL	Code	Reason	RTT	Size Resp.	Highest Al	Note	Tags
1	Proxy	Mon Apr	GET	https://lo	200	OK	1290	47	Medium	FALSE	JSON
3	Proxy	Mon Apr	GET	https://fir	200	OK	428	223	Medium	FALSE	JSON
4	Proxy	Mon Apr	GET	https://fir	200	OK	41	1854	Medium	FALSE	JSON
5	Proxy	Mon Apr	POST	https://sh	200	OK	1509	1967	Low	FALSE	
17	Proxy	Mon Apr	GET	https://co	200	OK	229	5348	Low	FALSE	
20	Proxy	Mon Apr	GET	https://wv	200	OK	1032	741252	Low	FALSE	Form, Hidden, Script, SetCookie, Comment
22	Proxy	Mon Apr	GET	https://fir	200	OK	35	232	Medium	FALSE	JSON
28	Proxy	Mon Apr	GET	https://fir	200	OK	48	2387	Medium	FALSE	Comment, JSON
31	Proxy	Mon Apr	GET	https://wv	200	OK	163	138775	Low	FALSE	Script
37	Proxy	Mon Apr	GET	https://tra	200	OK	1502	55894	Low	FALSE	Comment
41	Proxy	Mon Apr	GET	https://as	200	OK	363	251411	Low	FALSE	Script, Comment
42	Proxy	Mon Apr	GET	https://wv	200	OK	410	60820	Informatic	FALSE	
43	Proxy	Mon Apr	GET	https://wv	200	OK	268	2063		FALSE	
47	Proxy	Mon Apr	GET	https://wv	200	OK	287	6386	Low	FALSE	
48	Proxy	Mon Apr	GET	https://wv	200	OK	261	9023		FALSE	
50	Proxy	Mon Apr	GET	https://wv	200	OK	161	34013	Informatic	FALSE	
53	Proxy	Mon Apr	GET	https://wv	200	OK	208	10520		FALSE	
54	Proxy	Mon Apr	GET	https://wv	200	OK	154	4265		FALSE	Script
52	Proxy	Mon Apr	GET	https://wv	200	OK	497	152819	Low	FALSE	
55	Proxy	Mon Apr	GET	https://wv	200	OK	555	400631	Medium	FALSE	Script, Comment
57	Proxy	Mon Apr	GET	https://wv	200	OK	83	2402	Low	FALSE	
56	Proxy	Mon Apr	GET	https://wv	200	OK	80	77		FALSE	
59	Proxy	Mon Apr	GET	https://wv	200	OK	102	5949	Low	FALSE	
69	Proxy	Mon Apr	GET	https://wv	200	OK	370	27529	Low	FALSE	Comment
72	Proxy	Mon Apr	GET	https://cd	200	OK	444	80288	Medium	FALSE	Comment
73	Proxy	Mon Apr	GET	https://wv	200	OK	94	10236	Low	FALSE	
75	Proxy	Mon Apr	GET	https://wv	200	OK	137	10200	Low	FALSE	Comment
74	Proxy	Mon Apr	GET	https://wv	200	OK	95	13500	Low	FALSE	
76	Proxy	Mon Apr	GET	https://wv	200	OK	175	102728	Low	FALSE	Comment
79	Proxy	Mon Apr	GET	https://tra	200	OK	200	2133	Low	FALSE	
80	Proxy	Mon Apr	GET	https://as	200	OK	201	46115	Low	FALSE	
93	Proxy	Mon Apr	GET	https://wv	200	OK	706	1244016	Medium	FALSE	Hidden, Comment

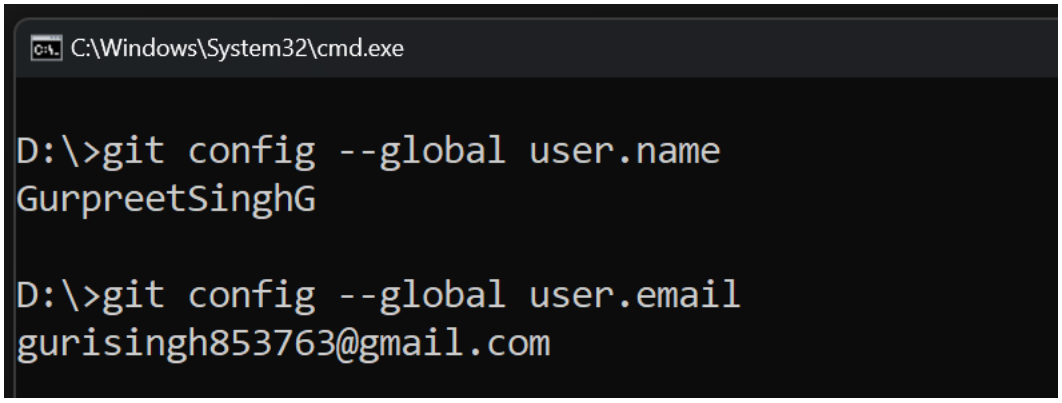
Physical security



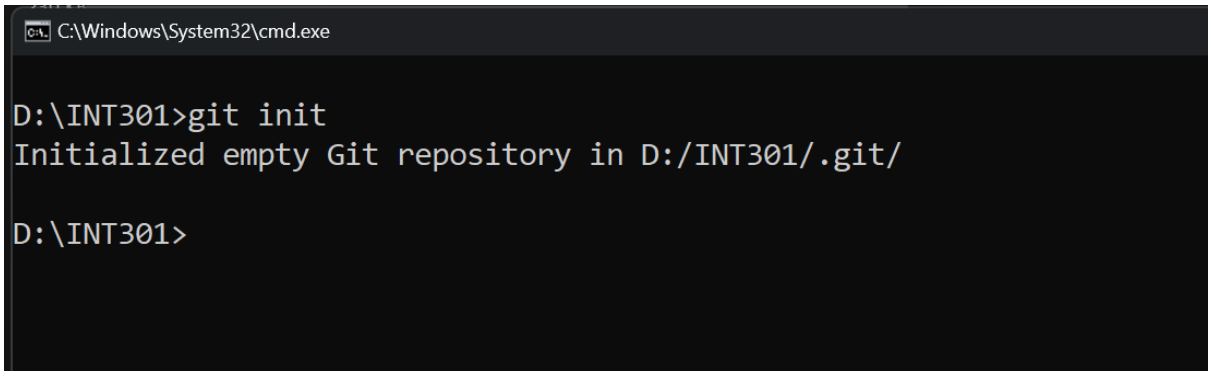


GitHub Snapshots

Setting the global username and email

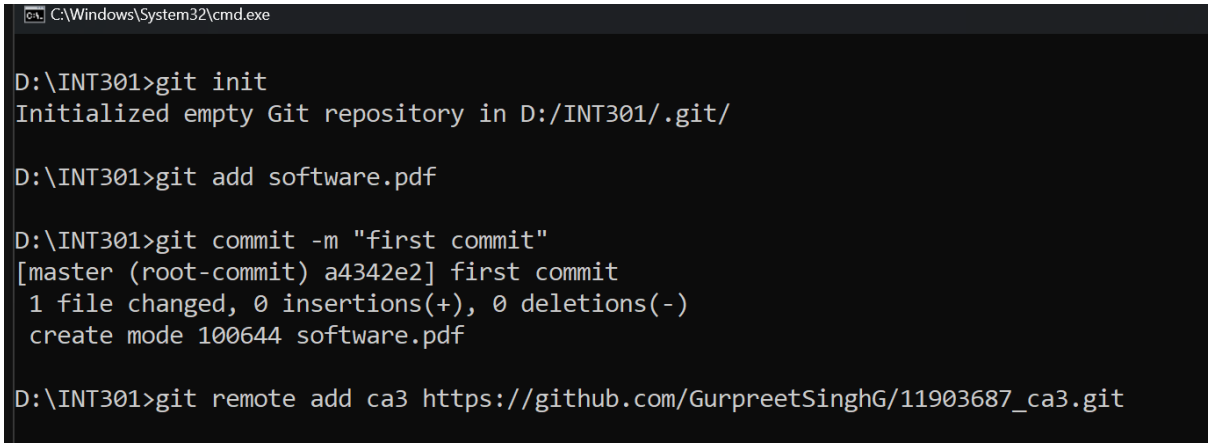


Intializing the repository



Adding the allthe revisions , committing them and pushing the reports to the github

Revision 1



```
D:\INT301>git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 478.61 KiB | 21.75 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/GurpreetSinghG/11903687_Gurpreet.git
   a4342e2..059fccf  main -> main
```

References

https://motion-project.github.io/

https://usbguard.github.io/

https://nvd.nist.gov/

https://www.wireshark.org/

https://www.zaproxy.org/download/

https://www.zaproxy.org/

GitHub Link

https://github.com/GurpreetSinghG/11903687_ca3/