

Blue Team Scenarios

Team Names:

1. Mohamed Abdel-Moneam Mohamed
2. Toka Abdelgwad
3. Habiba Bastawe Mohamed
4. George Samir
5. Omar Mohamed Abo Elkasem

1. Recording Network Capture on a PCAP file

Mission: Recording Network Traffics

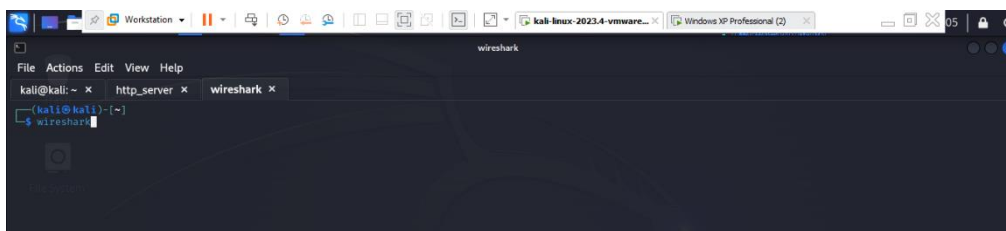
Tool: Wireshark

Wireshark: Wireshark is a powerful open-source network protocol analyzer used for capturing and inspecting the data traveling across a network in real-time.

Steps:

1. Open Wireshark on Linux

➤ Wireshark



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	93.126.105.202	192.168.128.130	TCP	4254	80 → 48530 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=2000
2	0.000000015	34.128.127.130	192.168.128.130	TCP	60	443 → 49542 [FIN, PSH, ACK] Seq=1 Ack=1 Win=64239 Len=0
3	0.000108140	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=4201 Win=65535 Len=0
4	0.000370598	192.168.128.130	34.128.127.130	TCP	54	49542 → 443 [ACK] Seq=1 Ack=2 Win=30660 Len=0
5	0.027864435	93.126.105.202	192.168.128.130	TCP	2854	80 → 48530 [PSH, ACK] Seq=4201 Ack=1 Win=64240 Len=2800
6	0.027977769	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=7001 Win=65535 Len=0
7	0.110260261	93.126.105.202	192.168.128.130	TCP	8454	80 → 48530 [PSH, ACK] Seq=7001 Ack=1 Win=64240 Len=8400
8	0.110424209	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=15401 Win=65535 Len=0
9	0.133980189	93.126.105.202	192.168.128.130	TCP	2854	80 → 48530 [PSH, ACK] Seq=15401 Ack=1 Win=64240 Len=2800
10	0.134076240	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=18201 Win=65535 Len=0
11	0.217089648	93.126.105.202	192.168.128.130	TCP	5654	80 → 48530 [PSH, ACK] Seq=18201 Ack=1 Win=64240 Len=5600
12	0.217140661	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=23801 Win=65535 Len=0
13	0.230517447	93.126.105.202	192.168.128.130	TCP	2854	80 → 48530 [PSH, ACK] Seq=23801 Ack=1 Win=64240 Len=2800
14	0.230597398	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=26601 Win=65535 Len=0
15	0.254082296	93.126.105.202	192.168.128.130	TCP	2854	80 → 48530 [PSH, ACK] Seq=26601 Ack=1 Win=64240 Len=2800
16	0.254159056	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=29401 Win=65535 Len=0
17	0.298328674	192.168.128.130	34.95.113.255	TCP	54	54574 → 443 [FIN, ACK] Seq=1 Ack=1 Win=31756 Len=0
18	0.299113277	34.95.113.255	192.168.128.130	TCP	60	443 → 54574 [ACK] Seq=1 Ack=2 Win=64239 Len=0
19	0.317571972	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=29401 Ack=1 Win=64240 Len=1400
20	0.317798084	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=30801 Win=65535 Len=0
21	0.318242205	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=30801 Ack=1 Win=64240 Len=1400
22	0.319313628	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=32201 Ack=1 Win=64240 Len=1400
23	0.319379069	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=33601 Win=65535 Len=0
24	0.320820628	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=33601 Ack=1 Win=64240 Len=1400
25	0.320941287	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=35001 Win=65535 Len=0
26	0.335216035	93.126.105.202	192.168.128.130	TCP	2854	80 → 48530 [PSH, ACK] Seq=35001 Ack=1 Win=64240 Len=2800
27	0.335438870	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=37801 Win=65535 Len=0
28	0.354095813	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=37801 Ack=1 Win=64240 Len=1400
29	0.354251281	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=39201 Win=65535 Len=0
30	0.354965562	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=39201 Ack=1 Win=64240 Len=1400
31	0.371130103	34.95.113.255	192.168.128.130	TCP	60	443 → 54574 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=0
32	0.371179315	192.168.128.130	34.95.113.255	TCP	54	54574 → 443 [ACK] Seq=2 Ack=2 Win=31756 Len=0
33	0.395923688	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=40601 Win=65535 Len=0
34	0.419055099	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=40601 Ack=1 Win=64240 Len=1400
35	0.419289508	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=42001 Win=65535 Len=0
36	0.422048897	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=42001 Ack=1 Win=64240 Len=1400
37	0.422258806	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=43401 Win=65535 Len=0
38	0.420085300	93.126.105.202	192.168.128.130	TCP	5654	80 → 48530 [PSH, ACK] Seq=43401 Ack=1 Win=64240 Len=5600

PCAP File Source:

https://drive.google.com/file/d/1EnJR_Plk0BSmWuNY3YLP63pOs-grR66Ea/view?usp=sharing

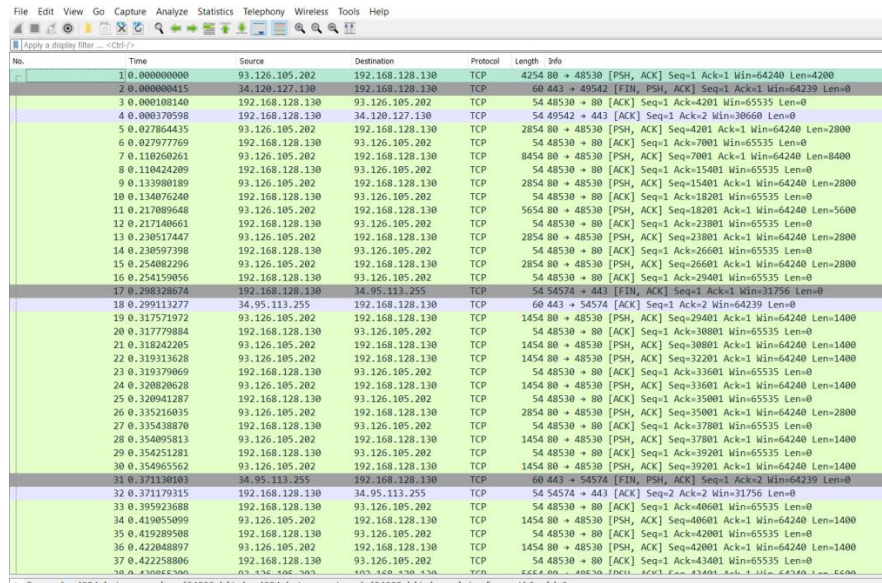
2. Investigating PCAP file to discover ongoing reconnaissance

Mission: Investigating PCAP file to discover reconnaissance on the Whole Network

Tool: Wireshark

Steps:

- Open PCAP file



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	93.126.105.202	192.168.128.130	TCP	4254	80 → 48530 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=4200
2	0.000000415	34.128.127.130	192.168.128.130	TCP	60	443 → 49542 [FIN, PSH, ACK] Seq=1 Ack=1 Win=64239 Len=0
3	0.000108140	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=4201 Win=65535 Len=0
4	0.000370598	192.168.128.130	34.128.127.130	TCP	54	49542 → 443 [ACK] Seq=1 Ack=2 Win=30600 Len=0
5	0.027864435	93.126.105.202	192.168.128.130	TCP	2854	80 → 48530 [PSH, ACK] Seq=4201 Ack=1 Win=64240 Len=2800
6	0.027977769	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=7001 Win=65535 Len=0
7	0.110260261	93.126.105.202	192.168.128.130	TCP	8454	80 → 48530 [PSH, ACK] Seq=7001 Ack=1 Win=64240 Len=8400
8	0.110424209	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=15401 Win=65535 Len=0
9	0.113900189	93.126.105.202	192.168.128.130	TCP	2854	80 → 48530 [PSH, ACK] Seq=15401 Ack=1 Win=64240 Len=2800
10	0.134076240	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=18201 Win=65535 Len=0
11	0.217080648	93.126.105.202	192.168.128.130	TCP	5654	80 → 48530 [PSH, ACK] Seq=18201 Ack=1 Win=64240 Len=5600
12	0.217140661	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=23801 Win=65535 Len=0
13	0.230517447	93.126.105.202	192.168.128.130	TCP	2854	80 → 48530 [PSH, ACK] Seq=23801 Ack=1 Win=64240 Len=2800
14	0.230597398	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=26601 Win=65535 Len=0
15	0.254082296	93.126.105.202	192.168.128.130	TCP	2854	80 → 48530 [PSH, ACK] Seq=26601 Ack=1 Win=64240 Len=2800
16	0.254159056	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=29401 Win=65535 Len=0
17	0.298320674	192.168.128.130	34.95.113.255	TCP	54	54574 → 443 [FIN, PSH, ACK] Seq=1 Ack=1 Win=31756 Len=0
18	0.299113277	34.95.113.255	192.168.128.130	TCP	60	443 → 54574 [ACK] Seq=1 Ack=2 Win=64239 Len=0
19	0.315751972	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=20401 Ack=1 Win=64240 Len=1400
20	0.317779884	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=30801 Win=65535 Len=0
21	0.318242205	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=30801 Ack=1 Win=64240 Len=1400
22	0.319313628	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=32201 Ack=1 Win=64240 Len=1400
23	0.319379069	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=33601 Win=65535 Len=0
24	0.320820628	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=33601 Ack=1 Win=64240 Len=1400
25	0.320941287	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=35001 Win=65535 Len=0
26	0.335216035	93.126.105.202	192.168.128.130	TCP	2854	80 → 48530 [PSH, ACK] Seq=35001 Ack=1 Win=64240 Len=2800
27	0.335438870	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=37801 Win=65535 Len=0
28	0.354095813	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=37801 Ack=1 Win=64240 Len=1400
29	0.354251281	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=39201 Win=65535 Len=0
30	0.354655562	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=39201 Ack=1 Win=64240 Len=1400
31	0.371130183	34.95.113.255	192.168.128.130	TCP	60	443 → 54574 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=0
32	0.371179315	192.168.128.130	34.95.113.255	TCP	54	54574 → 443 [ACK] Seq=2 Ack=2 Win=31756 Len=0
33	0.395923688	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=40601 Win=65535 Len=0
34	0.419055099	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=40601 Ack=1 Win=64240 Len=1400
35	0.419209580	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=42801 Win=65535 Len=0
36	0.422048897	93.126.105.202	192.168.128.130	TCP	1454	80 → 48530 [PSH, ACK] Seq=42801 Ack=1 Win=64240 Len=1400
37	0.422258806	192.168.128.130	93.126.105.202	TCP	54	48530 → 80 [ACK] Seq=1 Ack=43401 Win=65535 Len=0
38	0.430065300	93.126.105.202	192.168.128.130	TCP	54	48530 → 80 [ACK] Seq=1 Ack=43401 Win=65535 Len=0

- Investigation Output

- Received packets :159716
- IPV4: 159666
- ICMP :13
- UDP: 147
- TCP: 159506
- Alerts: 1111

There are TCP syn ,UDP and Fin scanning in the target machine

3. Downloading image by Mistake and Record ongoing traffic

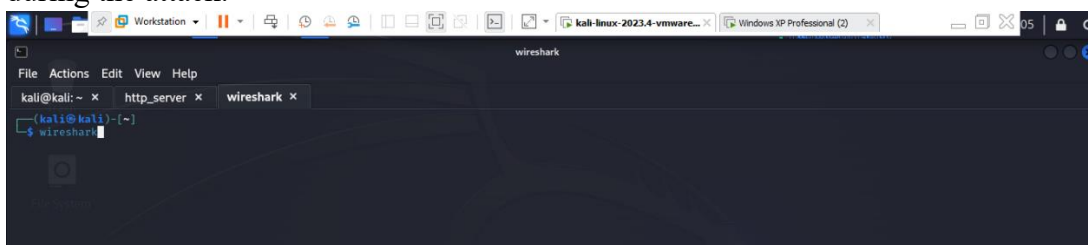
Mission: Downloading image by Mistake

Tool: Wireshark

Steps:

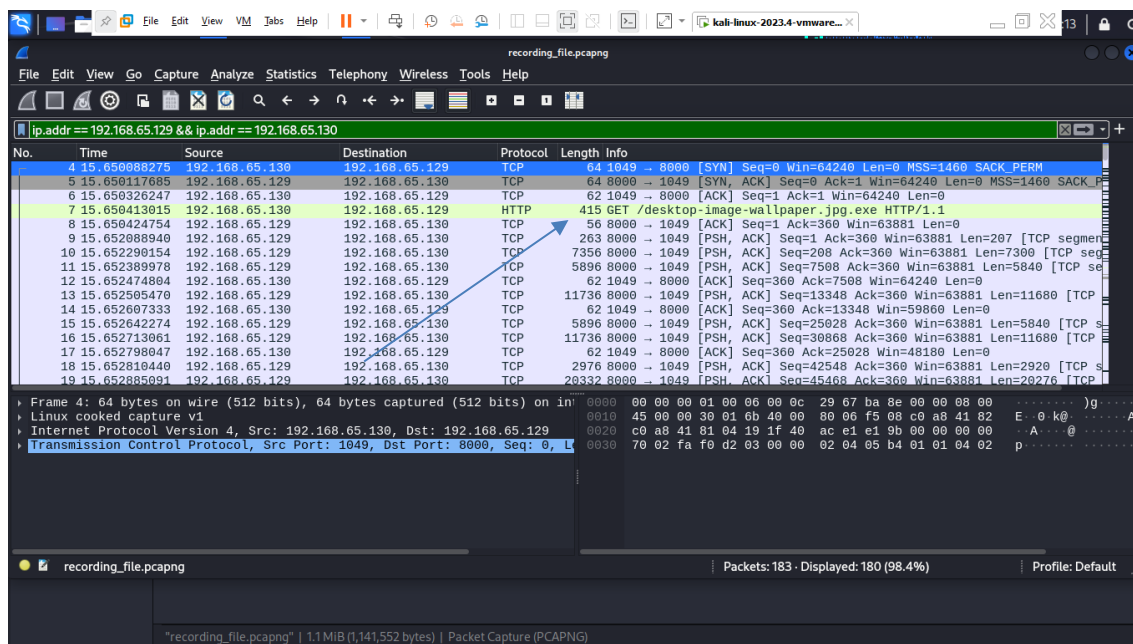
1. Network Traffic Capture:

- During the payload delivery and execution, network traffic was captured using Wireshark. This allowed for further analysis to track the exploitation process, and any data exchanged during the attack.



- Wireshark Filters Used:

`ip.addr == 192.168.65.129 && ip.addr == 192.168.65.130`



PCAP File Source:

<https://drive.google.com/file/d/1GaOZr0pqx2hR8Isc3gahzeb9h2vJ7T-h/view?usp=sharing>

4. Investigating the PCAP file for alert

Mission: investigating PCAP file

Tool: Snort

In this mission, we utilized a PCAP file recorded from a Windows XP machine, capturing reconnaissance activities conducted on that system.

Steps:

- `sudo snort -r Windows_xp.pcapng -c /etc/snort/snort.conf -A console`

```
File Actions Edit View Help
(kali@kali)-[~/project]
$ snort --version

--> Snort! <--
Version 2.9.7.0 GRE (Build 149)
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 1.10.5 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.3.1

(kali@kali)-[~/project]
$ sudo snort -r Windows_xp.pcapng -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 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 41
080 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 5600 ]
PortVar 'FILE_DATA_PORTS' defined : [ 80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 71
44:7145 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 3
4443:34444 41080 50002 55555 ]
PortVar 'GTP_PORTS' defined : [ 2123 2152 3386 ]
Detection:
  Search-Method = AC-Full-Q
  Split Any/Any group = enabled
  Search-Method-Optimizations = enabled
  Maximum pattern length = 20
Tagged Packet Limit: 256
Loading dynamic engine /usr/lib/snort_dynamicengine/libsengine.so... done
Loading all dynamic detection libs from /usr/lib/snort_dynamicrules...
WARNING: No dynamic libraries found in directory /usr/lib/snort_dynamicrules.
Finished Loading all dynamic detection libs from /usr/lib/snort_dynamicrules
Loading all dynamic preprocessor libs from /usr/lib/snort_dynamicpreprocessor/...
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libs_dns_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libs_ftptelnet_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libs_modbus_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libs_gtp_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libs_ssl_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libs_imap_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libs_ssh_preproc.so... done
Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libs_reputation_preproc.so... done
```

➤ The output:

Packet I/O Totals:	
Received:	159716
Analyzed:	159716 (100.000%)
Dropped:	0 (0.000%)
Filtered:	0 (0.000%)
Outstanding:	0 (0.000%)
Injected:	0

Breakdown by protocol (includes rebuilt packets):	
Eth:	159728 (100.000%)
VLAN:	0 (0.000%)
IP4:	159666 (99.961%)
Frag:	0 (0.000%)
ICMP:	13 (0.008%)
UDP:	167 (0.092%)
TCP:	159506 (99.861%)
IP6:	38 (0.024%)
IP6 Ext:	38 (0.024%)
IP6 Opts:	0 (0.000%)
Frag6:	0 (0.000%)
ICMP6:	5 (0.003%)
UDP6:	33 (0.021%)
TCP6:	0 (0.000%)
Teredo:	0 (0.000%)
ICMP-IP:	0 (0.000%)
IP4/IP4:	0 (0.000%)
IP4/IP6:	0 (0.000%)
IP6/IP4:	0 (0.000%)
IP6/IP6:	0 (0.000%)
GRE:	0 (0.000%)
GRE Eth:	0 (0.000%)
GRE VLAN:	0 (0.000%)
GRE IP4:	0 (0.000%)
GRE IP6:	0 (0.000%)
GRE IP6 Ext:	0 (0.000%)
GRE PPTP:	0 (0.000%)
GRE ARP:	0 (0.000%)
GRE IPX:	0 (0.000%)
GRE Loop:	0 (0.000%)
MPLS:	0 (0.000%)
ARP:	24 (0.015%)
IPX:	0 (0.000%)
Eth Loop:	0 (0.000%)
Eth Disc:	0 (0.000%)
IP4 Disc:	0 (0.000%)
IP6 Disc:	0 (0.000%)
TCP Disc:	0 (0.000%)
UDP Disc:	0 (0.000%)
ICMP Disc:	0 (0.000%)
All Discard:	0 (0.000%)
Other:	0 (0.000%)

Action Stats:	
Alerts:	1111 (0.696%)
Logged:	1111 (0.696%)
Passed:	0 (0.000%)

Limits:	
Match:	0
Queue:	0
Log:	0
Event:	0
Alert:	0

Verdicts:	
Allow:	159703 (99.992%)
Block:	0 (0.000%)
Replace:	0 (0.000%)
Whitelist:	13 (0.008%)
Blacklist:	0 (0.000%)
Ignore:	0 (0.000%)
Retry:	0 (0.000%)

Frag3 statistics:	
Total Fragments:	0
Frag Reassembled:	0
Discards:	0
Memory Faults:	0
Timeouts:	0
Overlaps:	0
Anomalies:	0
Alerts:	0
Drops:	0
FragTrackers Added:	0
FragTrackers Dumped:	0
FragTrackers Auto Freed:	0
Frag Nodes Inserted:	0
Frag Nodes Deleted:	0

Stream statistics:	
Total sessions:	73375
TCP sessions:	73334
UDP sessions:	41
ICMP sessions:	0
IP sessions:	0
TCP Prunes:	0
UDP Prunes:	0
ICMP Prunes:	0
IP Prunes:	0
TCP StreamTrackers Created:	73334
TCP StreamTrackers Deleted:	73334
TCP Timeouts:	0
TCP Overlaps:	0
TCP Segments Queued:	1177

HTTP Inspect - encodings (Note: stream-reassembled packets included):

```
POST methods: 0
GET methods: 0
HTTP Request Headers extracted: 0
HTTP Request Cookies extracted: 0
Post parameters extracted: 0
HTTP response Headers extracted: 1
HTTP Response Cookies extracted: 0
Unicode: 0
Double unicode: 0
Non-ASCII representable: 0
Directory traversals: 0
Extra slashes ("//"): 0
Self-referencing paths ("."): 0
HTTP Response Gzip packets extracted: 0
Gzip Compressed Data Processed: n/a
Gzip Decompressed Data Processed: n/a
Total packets processed: 4759
```

SMTP Preprocessor Statistics

```
Total sessions : 0
Max concurrent sessions : 0
```

dcerpc2 Preprocessor Statistics

```
Total sessions: 0
```

SSL Preprocessor:

```
SSL packets decoded: 11
  Client Hello: 0
  Server Hello: 2
  Certificate: 0
  Server Done: 0
Client Key Exchange: 0
Server Key Exchange: 0
Change Cipher: 4
  Finished: 0
Client Application: 6
Server Application: 3
  Alert: 0
Unrecognized records: 2
Completed handshakes: 0
  Bad handshakes: 0
  Sessions ignored: 1
Detection disabled: 1
```

SIP Preprocessor Statistics

```
Total sessions: 0
```

Snort exiting

5. Conclude a malicious activity and scan the device network

Mission : make scanning on machine

Tool: Nessus

Nessus: is a popular vulnerability scanning tool used to identify security weaknesses in systems

Steps:

1. Open Nessus from terminal

```
File Actions Edit View Help
kali@kali:~$ sudo systemctl start nessusd.service
[sudo] password for kali:
```

2. Set target Ip and start scanning

New Scan / Basic Network Scan

[← Back to Scan Templates](#)

Settings Credentials Plugins

BASIC

- General
- Schedule
- Notifications

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Name: windows_xp

Description:

Folder: My Scans

Targets: 192.168.28.133

Upload Targets Add File

Save Cancel

192.168.128.133



Vulnerabilities

Total: 31

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	-	34477	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING) (uncredentialed check)
CRITICAL	10.0	-	73182	Microsoft Windows XP Unsupported Installation Detection
CRITICAL	10.0	-	108797	Unsupported Windows OS (remote)
CRITICAL	10.0*	-	35362	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (uncredentialed check)
HIGH	8.1	-	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
HIGH	7.3	-	26920	SMB NULL Session Authentication
MEDIUM	5.3	-	57608	SMB Signing not required
LOW	2.1*	-	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	54615	Device Type

Nessus file:

<https://drive.google.com/file/d/1vZ5jIPdXSIBMWbZOze8VjcHnIciqUnys/view?usp=sharing>

6. After finding vulnerabilities, Launching an IPS to prevent the attack

Mission : after finding vulnerabilities and port scanning in network ,launching rules to prevent the attack

Tool: snort

Snort: Snort is an open-source network intrusion detection and prevention system (IDPS) . It is designed to monitor network traffic in real-time and analyze it for suspicious activities and potential threats.

Steps to prevent suspicious activity:

After analyzing the PCAP file using Snort and Wireshark, we observed TCP and UDP scanning activities on unknown ports, which may indicate potential malicious behavior. Additionally, we detected numerous packets associated with ping scans, suggesting reconnaissance efforts across the entire network. To address these security concerns, we will implement rules to alert and drop these suspicious packets.

ip.addr==192.168.128.133&&tcp						
No.	Time	Source	Destination	Protocol	Length	Info
1027	18.388810009	192.168.128.130	192.168.128.133	TCP	58	47590 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1028	18.388914754	192.168.128.130	192.168.128.133	TCP	58	47590 → 53 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1029	18.388978343	192.168.128.130	192.168.128.133	TCP	58	47590 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1030	18.389024025	192.168.128.130	192.168.128.133	TCP	58	47590 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1031	18.389115952	192.168.128.130	192.168.128.133	TCP	58	47590 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1032	18.389169532	192.168.128.130	192.168.128.133	TCP	58	47590 → 139 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1033	18.389212838	192.168.128.133	192.168.128.130	TCP	60	25 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1034	18.389234680	192.168.128.130	192.168.128.133	TCP	58	47590 → 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1035	18.389283045	192.168.128.133	192.168.128.130	TCP	60	53 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1036	18.389283067	192.168.128.130	192.168.128.133	TCP	58	47590 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1037	18.389283171	192.168.128.133	192.168.128.130	TCP	60	587 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1038	18.389331291	192.168.128.130	192.168.128.133	TCP	58	47590 → 1723 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1039	18.389431100	192.168.128.130	192.168.128.133	TCP	58	47590 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1040	18.389480555	192.168.128.133	192.168.128.130	TCP	60	445 → 47590 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
1041	18.389480793	192.168.128.133	192.168.128.130	TCP	60	995 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1042	18.389507727	192.168.128.130	192.168.128.133	TCP	54	47590 → 445 [RST] Seq=1 Win=0 Len=0
1043	18.389622772	192.168.128.133	192.168.128.130	TCP	60	139 → 47590 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
1044	18.389674598	192.168.128.130	192.168.128.133	TCP	54	47590 → 139 [RST] Seq=1 Win=0 Len=0
1045	18.389748934	192.168.128.133	192.168.128.130	TCP	60	1720 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1046	18.389876498	192.168.128.133	192.168.128.130	TCP	60	199 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1047	18.389876685	192.168.128.133	192.168.128.130	TCP	60	1723 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1048	18.389941680	192.168.128.133	192.168.128.130	TCP	60	80 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1049	18.390058123	192.168.128.130	192.168.128.133	TCP	58	47590 → 23 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1050	18.390152411	192.168.128.130	192.168.128.133	TCP	58	47590 → 8888 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1051	18.390189781	192.168.128.130	192.168.128.133	TCP	58	47590 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1052	18.390225729	192.168.128.130	192.168.128.133	TCP	58	47590 → 8080 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1053	18.390265889	192.168.128.130	192.168.128.133	TCP	58	47590 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1054	18.390295274	192.168.128.133	192.168.128.130	TCP	60	23 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1055	18.390305597	192.168.128.130	192.168.128.133	TCP	58	47590 → 111 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1056	18.390374522	192.168.128.130	192.168.128.133	TCP	58	47590 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1057	18.390398476	192.168.128.133	192.168.128.130	TCP	60	8888 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1058	18.390406448	192.168.128.130	192.168.128.133	TCP	58	47590 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1059	18.390435054	192.168.128.130	192.168.128.133	TCP	58	47590 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1060	18.390458599	192.168.128.133	192.168.128.130	TCP	60	113 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1061	18.390467766	192.168.128.130	192.168.128.133	TCP	58	47590 → 993 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1062	18.390496749	192.168.128.133	192.168.128.130	TCP	60	8080 → 47590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1063	18.390499887	192.168.128.130	192.168.128.133	TCP	58	47590 → 554 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1064	18.390500000	192.168.128.130	192.168.128.133	TCP	58	47590 → 1025 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Rules :

We will test these rules in real-time; however, we encountered some issues with (IPS) functionality in Snort, as it is not operating effectively on our machine. The required packages are incompatible, and despite our efforts to resolve these issues, we were unable to do so. As a result, we have opted to operate in (IDS) mode to identify and analyze the threats effectively.

1. Test TCP SYN Scan

```
(kali@kali)-[~]
$ sudo nmap -sS 192.168.128.133
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-21 16:07 EDT
Nmap scan report for 192.168.128.133 (192.168.128.133)
Host is up (0.00041s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 00:0C:29:61:FD:09 (VMware)
```

➤ Snort detected that activity and give us alert

```
10/21-16:07:59.139904  [**] [1:1000001:1] TCP SYN Scan Detected [**] [Priority: 0] {TCP} 192.168.128.130:34678 → 192.168.128.133:1723
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:07:59.140031  [**] [1:1000001:1] TCP SYN Scan Detected [**] [Priority: 0] {TCP} 192.168.128.130:34678 → 192.168.128.133:22
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:07:59.140158  [**] [1:1000001:1] TCP SYN Scan Detected [**] [Priority: 0] {TCP} 192.168.128.130:34678 → 192.168.128.133:1720
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:07:59.140276  [**] [1:1000001:1] TCP SYN Scan Detected [**] [Priority: 0] {TCP} 192.168.128.130:34678 → 192.168.128.133:12345
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
```

2. Test UDP scan

```
(kali㉿kali)-[~]
└─$ sudo nmap -sU 192.168.128.133
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-21 16:15 EDT
Nmap scan report for 192.168.128.133 (192.168.128.133)
Host is up (0.00040s latency).
Not shown: 993 closed udp ports (port-unreach)
PORT      STATE      SERVICE
123/udp    open       ntp
137/udp    open       netbios-ns
138/udp    open|filtered netbios-dgm
445/udp    open|filtered microsoft-ds
500/udp    open|filtered isakmp
1900/udp   open|filtered upnp
4500/udp   open|filtered nat-t-ike
MAC Address: 00:0C:29:61:FD:09 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.57 seconds
```

➤ Snort Detection

```
10/21-16:15:32.551316  [**] [1:1000003:1] UDP Scan Detected [**] [Priority: 0] {UDP} 192.168.128.130:42103 → 192.168.128.133:9876
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:15:32.551605  [**] [1:1000003:1] UDP Scan Detected [**] [Priority: 0] {UDP} 192.168.128.130:42103 → 192.168.128.133:434
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:15:32.551866  [**] [1:1000003:1] UDP Scan Detected [**] [Priority: 0] {UDP} 192.168.128.130:42103 → 192.168.128.133:3401
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:15:32.552112  [**] [1:1000003:1] UDP Scan Detected [**] [Priority: 0] {UDP} 192.168.128.130:42103 → 192.168.128.133:19956
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
```


3. Test Ping Scan

```
(kali㉿kali)-[~]
└─$ ping 192.168.128.133
PING 192.168.128.133 (192.168.128.133) 56(84) bytes of data:
64 bytes from 192.168.128.133: icmp_seq=1 ttl=128 time=1.26 ms
64 bytes from 192.168.128.133: icmp_seq=2 ttl=128 time=0.671 ms
64 bytes from 192.168.128.133: icmp_seq=3 ttl=128 time=0.770 ms
64 bytes from 192.168.128.133: icmp_seq=4 ttl=128 time=1.01 ms
^C

```

➤ Snort Detection

```
10/21-16:20:42.514074 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:43.538275 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:44.562464 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:45.586016 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:46.610358 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:47.634227 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:48.635405 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:49.650961 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:50.674166 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:51.697959 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:52.722156 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:53.746153 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:54.769920 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/21-16:20:55.770891 [**] [1:1000005:1] ICMP Ping Scan Detected [**] [Priority: 0] {ICMP} 192.168.128.130 → 192.168.128.133
WARNING: No preprocessors configured for policy 0.
```

9. Forensic Investigation Report on Backdoor Discovery

Mission: Perform Forensic Investigation on the Image and Get the Backdoor

Tool: Autopsy

Autopsy: Autopsy is the premier open-source forensics platform which is fast, easy-to-use, and capable of analyzing all types of mobile devices and digital media. Its plug-in architecture enables extensibility from community-developed or custom-built modules. Autopsy evolves to meet the needs of hundreds of thousands of professionals in law enforcement, national security, litigation support, and corporate investigation

Steps:

1. Introduction

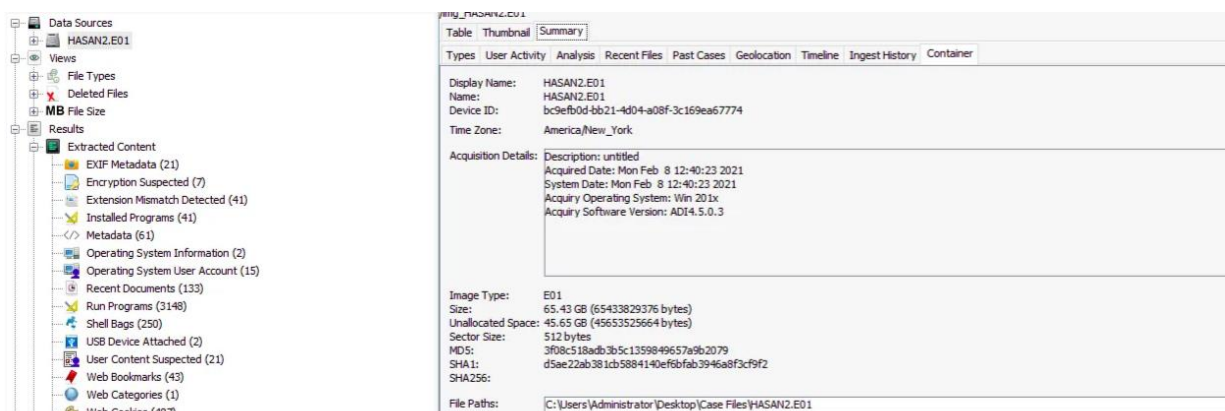
- **Purpose of the Investigation:**
 - To analyze the disk image for malicious activity and identify any backdoors present.
- **Scope of the Report:**
 - This report details the methodology used, findings, and conclusions drawn from the investigation.

2. Initial Findings

- **MD5 Hash of The Image**

We can find the hash of the image by selecting the appropriate data source in Autopsy and navigating to the Container tab under Summary.

3f08c518adb3b5c1359849657a9b2079



Types	User Activity	Analysis	Recent Files	Past Cases	Geolocation	Timeline	Ingest History	Container
Display Name: HASAN2.E01 Name: HASAN2.E01 Device ID: bc9efb0d-bb21-4d04-a08f-3c169ea67774 Time Zone: America/New_York								
Acquisition Details: Description: untitled Acquired Date: Mon Feb 8 12:40:23 2021 System Date: Mon Feb 8 12:40:23 2021 Acquired Operating System: Win 201x Acquired Software Version: ADI4.5.0.3								
Image Type: E01 Size: 65.43 GB (65433829376 bytes) Unallocated Space: 45.65 GB (45653525664 bytes) Sector Size: 512 bytes MD5: 3f08c518adb3b5c1359849657a9b2079 SHA1: d5ae22ab381cb5884140ef6fbab3946a8f3cf9f2 SHA256: File Paths: C:\Users\Administrator\Desktop\Case Files\HASAN2.E01								

What is the computer account name?

• User Accounts Identified

Just below the *Operating System Information* results, we see an option for *Operating System User Accounts*, we can get our answer from there.

Accounts are: H4S4N,joshwa,keshav,sandhya,shreya,sivapriya,srini,suba

Tryhackme - Autopsy 4.18.0
Case View Tools Window Help

Add Data Source Images/Videos Communications Geolocation Timeline Discovery Generate Report Close Case

Listing
Operating System User Account

Table Thumbnail Summary

Source File	S	C	O	User ID	Username	Count
SAM				S-1-5-21-3919888104-523186866-407859479-1005	keshav	2
SAM				S-1-5-21-3919888104-523186866-407859479-1006	sivapriya	2
SAM				S-1-5-21-3919888104-523186866-407859479-1007	sandhya	2
SAM				S-1-5-21-3919888104-523186866-407859479-1008	srini	2
SAM				S-1-5-21-3919888104-523186866-407859479-1001	H4S4N	2
SAM				S-1-5-21-3919888104-523186866-407859479-1002	joshwa	2
SAM				S-1-5-21-3919888104-523186866-407859479-500	Administrator	2
SAM				S-1-5-21-3919888104-523186866-407859479-1003	suba	2
SAM				S-1-5-21-3919888104-523186866-407859479-501	Guest	2
SAM				S-1-5-21-3919888104-523186866-407859479-1004	shreya	2
SAM				S-1-5-21-3919888104-523186866-407859479-503	DefaultAccount	2
SAM				S-1-5-21-3919888104-523186866-407859479-504	WDAGUtilityAccount	2
SOFTWARE				S-1-5-18	systemprofile	
SOFTWARE				S-1-5-19	LocalService	
SOFTWARE				S-1-5-20	NetworkService	

More accounts worked by Data Accessed

- **Last Logged Use** We can sort the User Accounts by “Date Accessed” and we see **sivapriya** was the last user who logged to the device

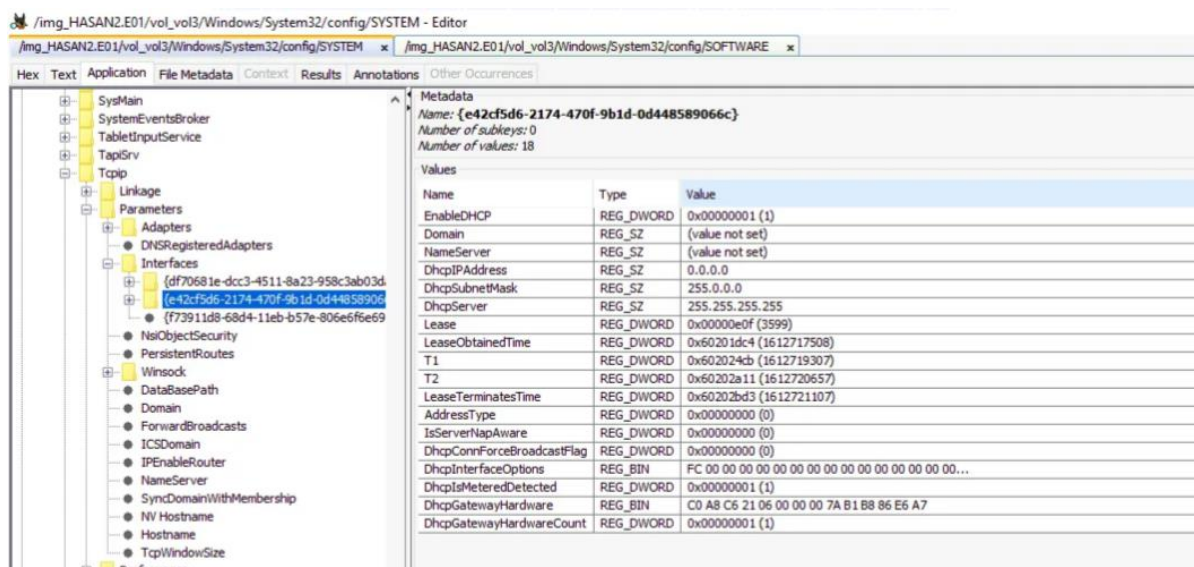
Listing
Operating System User Account

Table Thumbnail Summary

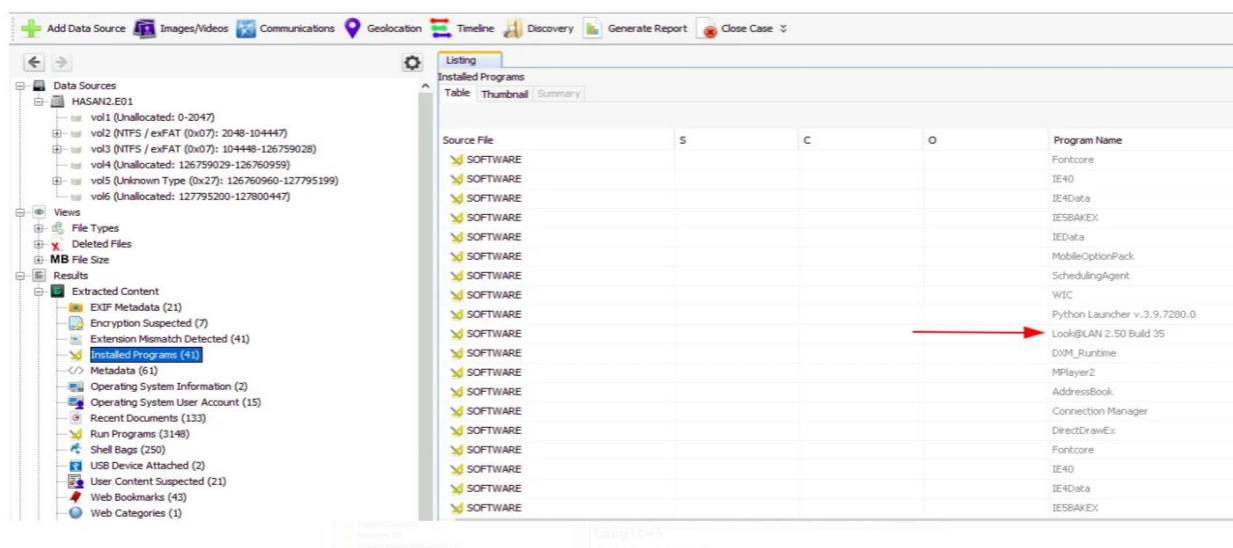
Source File	S	C	O	User ID	Username	Date Created	▼ Date Accessed	Count
SAM				S-1-5-21-3919888104-523186866-407859479-1006	sivapriya	2021-02-06 05:39:55 EST	2021-02-07 12:05:37 EST	10
SAM				S-1-5-21-3919888104-523186866-407859479-1001	H4S4N	2021-02-06 18:48:16 EST	2021-02-07 12:05:11 EST	24
SAM				S-1-5-21-3919888104-523186866-407859479-1004	shreya	2021-02-06 05:38:48 EST	2021-02-07 11:46:52 EST	13
SAM				S-1-5-21-3919888104-523186866-407859479-1003	suba	2021-02-06 05:38:22 EST	2021-02-07 11:46:01 EST	2
SAM				S-1-5-21-3919888104-523186866-407859479-1008	srini	2021-02-06 05:41:10 EST	2021-02-07 11:45:42 EST	2
SAM				S-1-5-21-3919888104-523186866-407859479-1007	sandhya	2021-02-06 05:40:42 EST	2021-02-07 11:45:11 EST	5
SAM				S-1-5-21-3919888104-523186866-407859479-1005	keshav	2021-02-06 05:39:20 EST	2021-02-07 11:45:00 EST	5
SAM				S-1-5-21-3919888104-523186866-407859479-1002	joshwa	2021-02-06 05:38:00 EST	2021-02-07 11:44:49 EST	5

• IP Address for the computer

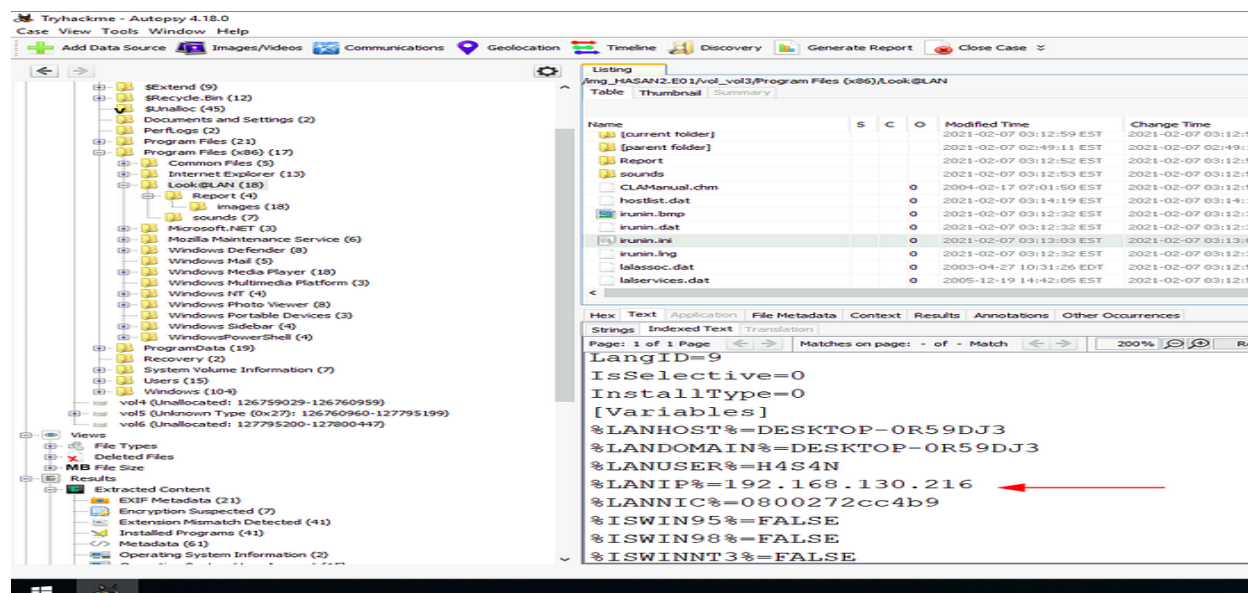
Since we're working with an image of a Windows machine, we can find the IP address associated with network adapters in the Windows Registry. We can even access the registry from within Autopsy.



No such luck, the IP address is listed as 0.0.0.0. We'll have to find it elsewhere, and while looking through Autopsy's findings, we notice an unusual application installed on the device.



Searching for the executable name tells us it is a network monitoring tool, so let's look for any logs it may have generated. We find its directory under *Program Files (x86)*. Among the files in the folder, only one stands out, a .ini file. We can view the file within Autopsy by selecting it. so the **IP Address** would be 192.168.130.216 and **MAC Address** would be 08-00-27-2c-c4-b9



The screenshot shows the Autopsy 4.18.0 interface. The left pane displays the file system tree, with 'Program Files (x86)' expanded. The right pane shows the details of the selected file, 'Look@LAN'. The file is an INI file located at 'Ang_JASAN2.E01/Vol_013/Program Files (x86)/Look@LAN'. The file's contents are displayed in the 'Text' tab, showing network configuration information. The IP address '192.168.130.216' and the MAC address '0800272cc4b9' are highlighted with red arrows.

Name	S	C	O	Modified Time	Change Time
[current folder]				2021-02-07 03:12:59 EST	2021-02-07 03:12:59
[parent folder]				2021-02-07 02:49:11 EST	2021-02-07 02:49:11
Report				2021-02-07 03:12:52 EST	2021-02-07 03:12:52
sounds				2021-02-07 03:12:53 EST	2021-02-07 03:12:53
CLAManual.chm			0	2004-02-17 07:01:50 EST	2021-02-07 03:12:55
hostlist.dat			0	2021-02-07 03:14:19 EST	2021-02-07 03:14:19
irunin.bmp			0	2021-02-07 03:12:32 EST	2021-02-07 03:12:32
irunin.dat			0	2021-02-07 03:12:32 EST	2021-02-07 03:12:32
irunin.ini			0	2021-02-07 03:13:00 EST	2021-02-07 03:13:00
irunin.log			0	2021-02-07 03:12:32 EST	2021-02-07 03:12:32
lalsoc.dat			0	2003-04-27 10:31:26 EDT	2021-02-07 03:12:52
lalservices.dat			0	2005-12-19 14:42:05 EST	2021-02-07 03:12:52

```

LangID=9
IsSelective=0
InstallType=0
[Variables]
%LANHOST%=DESKTOP-0R59DJ3
%LANDOMAIN%=DESKTOP-0R59DJ3
%LANUSER%=H4S4N
%LANIP%=192.168.130.216
%LANNIC%=0800272cc4b9
%ISWIN95%=FALSE
%ISWIN98%=FALSE
%ISWINNT3%=FALSE
  
```

3. File System Analysis

First, we identify unusual file names or extensions that may indicate malicious activity. After checking some of the user's Desktops, we locate a flag within the shreya user's Desktop directory. Now that we know the user, we'll check the PowerShell history for the account. There is also a PowerShell script on the user's desktop named exploit.

PowerShell command history is stored in
 APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
 As expected, we find the PowerShell history with a **flag{HarleyQuinnForQueen}**

The screenshot shows the Autopsy 4.18.0 interface. The left pane displays a file tree with the following structure:

- AppData (5)
 - Local (17)
 - LocalLow (4)
 - Roaming (5)
 - Adobe (3)
 - Microsoft (12)
 - Credentials (2)
 - Crypto (4)
 - Internet Explorer (4)
 - MMC (2)
 - Network (3)
 - Protect (4)
 - Spelling (3)
 - SystemCertificates (1)
 - Vault (2)
 - Windows (13)
 - AccountPictures
 - CloudStore (2)
 - Libraries (9)
 - Network Shortcuts
 - PowerShell (3)
 - PSReadLine
 - Printer Shortcut
 - Recent (25)
 - SendTo (9)
 - Start Menu (4)
 - Templates (2)
 - Themes (5)
- Application Data (2)
- Contacts (3)
- Cookies (2)
- Desktop (5)
- Documents (6)
- Downloads (5)
- Favorites (5)
- Links (5)
- Local Settings (2)
- Music (3)
- My Documents (2)
- NetHood (2)
- OneDrive (3)
- Pictures (5)

The right pane shows a file listing for the path `/img_HASAN2.E01/vol_vol3/Users/shreya/AppData/Roaming/Microsoft/Windows/PowerShell/PSReadLine`. The table below represents the data shown in the interface:

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)
[current folder]				2021-02-06 06:08:53 EST	2021-02-06 11:42:52 EST	2021-02-06 12:45:15 EST	2021-02-06 06:08:53 EST	288	Allocated
[parent folder]				2021-02-06 06:08:53 EST	2021-02-06 06:08:53 EST	2021-02-06 12:45:03 EST	2021-02-06 06:08:53 EST	256	Allocated
ConsoleHost_history.txt				2021-02-06 12:40:36 EST	2021-02-06 12:40:36 EST	2021-02-06 12:45:03 EST	2021-02-06 06:08:53 EST	421	Allocated

The bottom pane shows the content of the selected file, which is a PowerShell script:

```
cd .\Desktop\
exitcls
Add-Content .\shreya.txt 'flag{HarleyQuinnForQueen}'
Get-Content .\shreya.txt
Add-Content .\shreya.txt 'flag{HarleyQuinnForQueen}'
Get-Content .\shreya.txt
Set-Content .\shreya.txt 'flag{i_changed_it}'
exit
```

We noted a PowerShell script named exploit, so we'll go back and look at its contents now.

The screenshot shows the Autopsy 4.18.0 interface. The left pane displays a file tree with the following structure:

- AppData (5)
 - Application Data (2)
 - Contacts (3)
 - Cookies (2)
 - Desktop (5)
 - Documents (6)
 - Downloads (5)
 - Favorites (5)
 - Links (5)
 - Local Settings (2)
 - Music (3)
 - My Documents (2)
 - NetHood (2)
 - OneDrive (3)
 - Pictures (5)
 - PrintHood (2)
 - Recent (2)
 - Saved Games (3)
 - Searches (6)
 - SendTo (2)
 - Start Menu (2)
 - Templates (2)
 - Videos (3)
 - svagnya (34)
 - srini (33)
 - suba (33)
 - Windows (104)
 - addins (3)
 - appcompat (5)
 - appcache (13)
 - AppReadiness (2)
 - assembly (8)
 - bcastdrv (4)
 - Boot (9)
 - Branding (4)
 - CatTemp (2)
 - Containers (3)
 - Cursors (216)
 - debug (5)
 - diagnostics (5)

The right pane shows a file listing for the path `/img_HASAN2.E01/vol_vol3/Users/shreya/Desktop`. The table below represents the data shown in the interface:

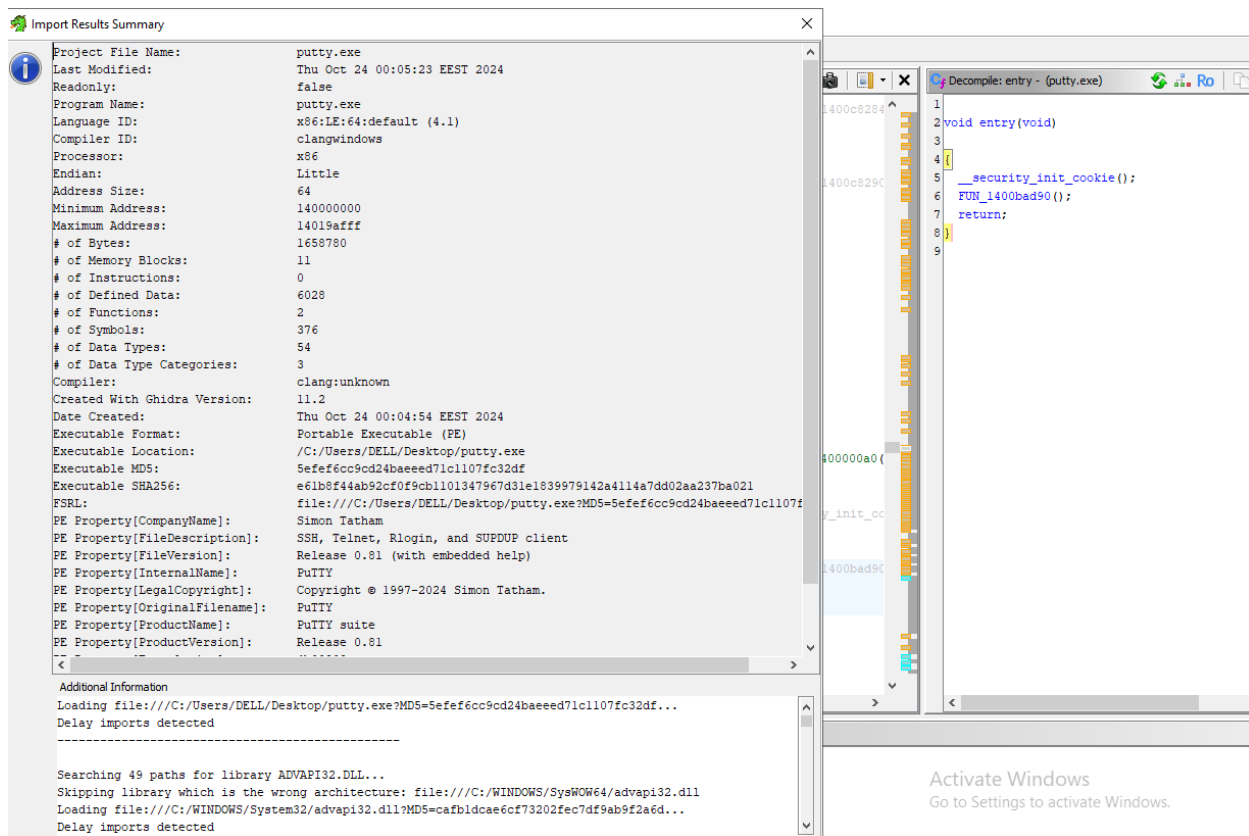
Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location
[current folder]				2021-02-06 06:51:42 EST	2021-02-06 06:51:42 EST	2021-02-07 11:48:09 EST	2021-02-06 05:41:55 EST	360	Allocated	Allocated	unknown	/img_HASAN2.E01/vol_vol3/Us
[parent folder]				2021-02-06 06:44:47 EST	2021-02-06 06:44:47 EST	2021-02-07 13:10:05 EST	2021-02-06 05:41:55 EST	256	Allocated	Allocated	unknown	/img_HASAN2.E01/vol_vol3/Us
desktop.ini			0	2021-02-06 06:41:58 EST	2021-02-06 06:12:21 EST	2021-02-07 13:10:05 EST	2021-02-06 05:41:55 EST	282	Allocated	Allocated	unknown	/img_HASAN2.E01/vol_vol3/Us
exploit.ps1			0	2021-02-06 06:53:29 EST	2021-02-06 06:53:29 EST	2021-02-07 03:01:54 EST	2021-02-06 06:06:22 EST	766	Allocated	Allocated	unknown	/img_HASAN2.E01/vol_vol3/Us
shreya.txt			0	2021-02-06 12:40:10 EST	2021-02-06 12:40:10 EST	2021-02-07 03:01:49 EST	2021-02-06 05:42:42 EST	20	Allocated	Allocated	unknown	/img_HASAN2.E01/vol_vol3/Us

The bottom pane shows the content of the selected file, which is a PowerShell script:

```
if(([System.Security.Principal.WindowsIdentity]::GetCurrent()).groups -
) {
    #Payload goes here
    #It'll run as Administrator
    New-Item "C:\Users\H4S4N\Desktop\hacked.txt"
    Add-Content C:\Users\H4S4N\Desktop\hacked.txt 'Flag{I-hacked-you}'
    ##### https://youtu.be/C9GfMfFjYI
```

10. Analysis of the Backdoor By Ghidra

Using Ghidra to analyze putty.exe allows you to reverse-engineer the executable and gain insights into its structure, functions, and potential behavior. Once you've loaded putty.exe into Ghidra and completed the analysis



Import Results Summary

Project File Name:	putty.exe
Last Modified:	Thu Oct 24 00:05:23 EEST 2024
Readonly:	false
Program Name:	putty.exe
Language ID:	x86:LE:64:default (4.1)
Compiler ID:	clangwindows
Processor:	x86
Endian:	Little
Address Size:	64
Minimum Address:	140000000
Maximum Address:	14019afff
# of Bytes:	1658780
# of Memory Blocks:	11
# of Instructions:	0
# of Defined Data:	6028
# of Functions:	2
# of Symbols:	376
# of Data Types:	54
# of Data Type Categories:	3
Compiler:	clang:unknown
Created With Ghidra Version:	11.2
Date Created:	Thu Oct 24 00:04:54 EEST 2024
Executable Format:	Portable Executable (PE)
Executable Location:	/C:/Users/DELL/Desktop/putty.exe
Executable MD5:	5efef6cc9cd24baeed71c1107fc32df
Executable SHA256:	e61b8f44ab92cf0f9cb1101347967d31e1839979142a4114a7dd02aa237ba021
FSRL:	file:///C:/Users/DELL/Desktop/putty.exe?MD5=5efef6cc9cd24baeed71c1107fc32df
PE Property[CompanyName]:	Simon Tatham
PE Property[FileDescription]:	SSH, Telnet, Rlogin, and SUPDUP client
PE Property[FileVersion]:	Release 0.81 (with embedded help)
PE Property[InternalName]:	PuTTY
PE Property[LegalCopyright]:	Copyright © 1997-2024 Simon Tatham.
PE Property[OriginalFilename]:	PuTTY
PE Property[ProductName]:	PuTTY suite
PE Property[ProductVersion]:	Release 0.81

Additional Information

Loading file:///C:/Users/DELL/Desktop/putty.exe?MD5=5efef6cc9cd24baeed71c1107fc32df...

Delay imports detected

Searching 49 paths for library ADVAPI32.DLL...

Skipping library which is the wrong architecture: file:///C:/WINDOWS/SysWOW64/advapi32.dll

Loading file:///C:/WINDOWS/System32/advapi32.dll?MD5=cafbdcae6cf73202fec7df9ab9f2a6d...

Delay imports detected

Decompile: entry - (putty.exe)

```

1 void entry(void)
2
3
4
5     __security_init_cookie();
6     FUN_1400bad90();
7     return;
8
9

```

Activate Windows
Go to Settings to activate Windows.