

Cyber Defense Strategies for

# Combating C2 Based Attacks



## About CyberWarFare Labs :

CW Labs is a renowned UK based Ed-tech company specializing in cybersecurity cyber range labs. They provide on-demand educational services and recognize the need for continuous adaptation to evolving threats and client requirements. The company has two primary divisions :

**1. Cyber Range Labs**

**2. Up-Skilling Platform**



# INFINITE LEARNING EXPERIENCE

## **About Speaker :**

### **Harisuthan S** **(Senior Security Engineer)**

Is a Blue Team Security researcher, bringing over 3+ years of experience in cyber defence. possesses a deep understanding of Blue Team methodologies including investigation and detection over cyber attacks,



## Agenda

- Working of C2
- Investigating C2 targeted attacks
- JA3 Fingerprinting
- Certified Cyber Defense Analyst : CCDA
- Certification Procedure

# Working of Command & Control

# Working of **Command & Control**

**Phase 01**

**Reconnaissance**

**Weaponization**

**Phase 02**

**Delivery**

**Exploitation**

**Installation**

**Phase 03**

**Command and Control (C2)**

# General Working Overview

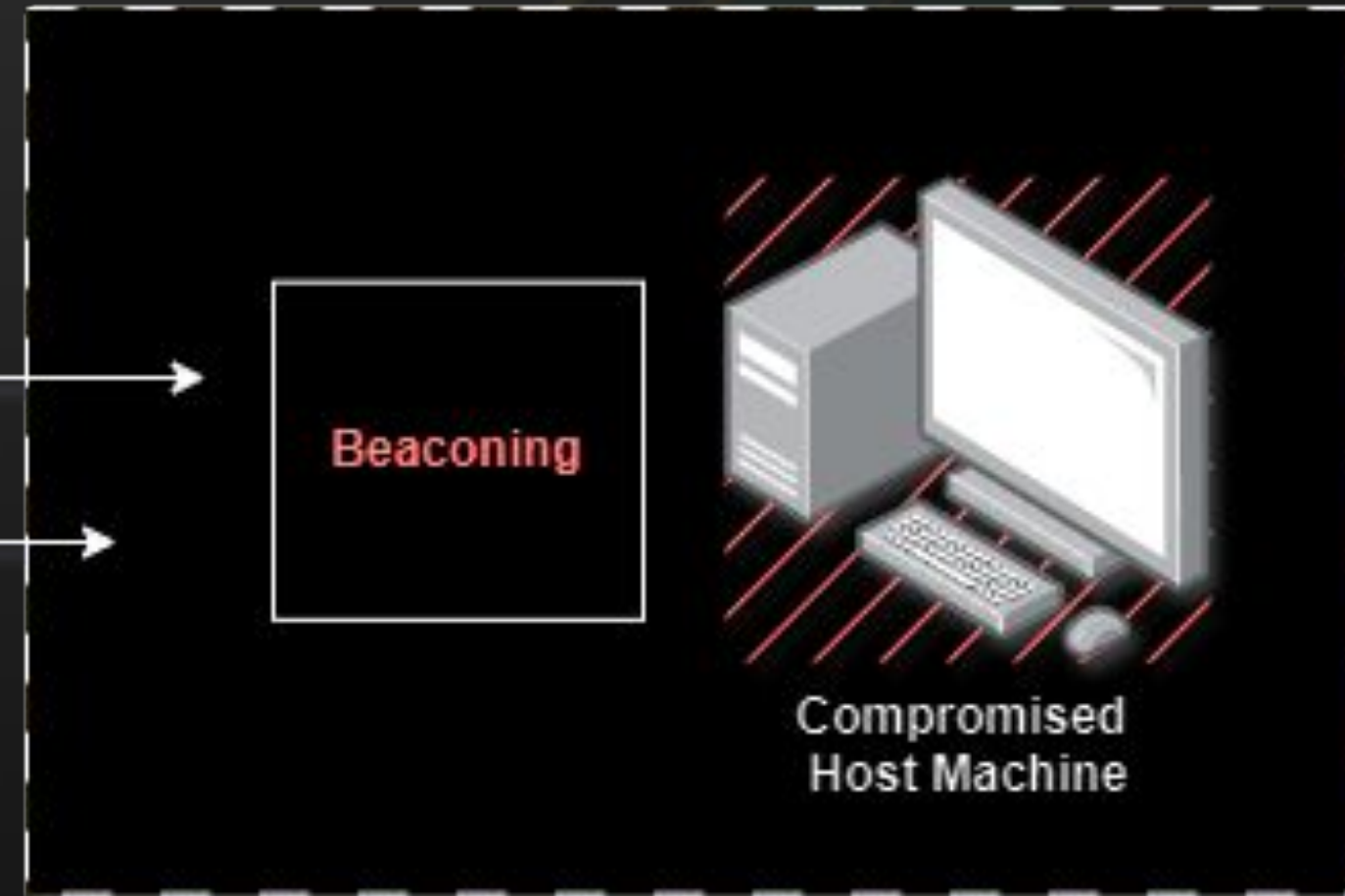


C2 Server

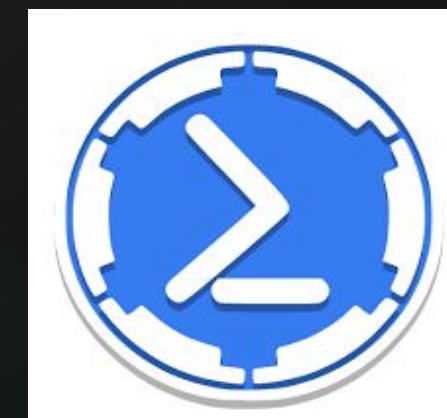
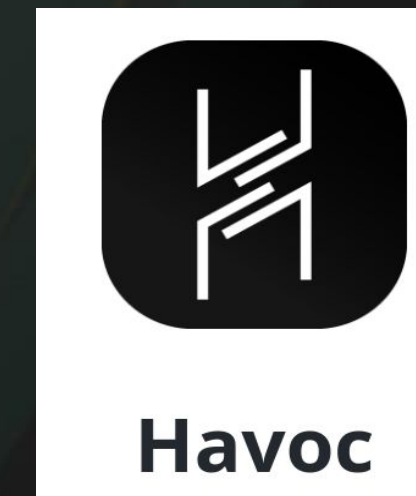
Listener  
Session



Internal Network



# Common **C2** Services





## Working Overview

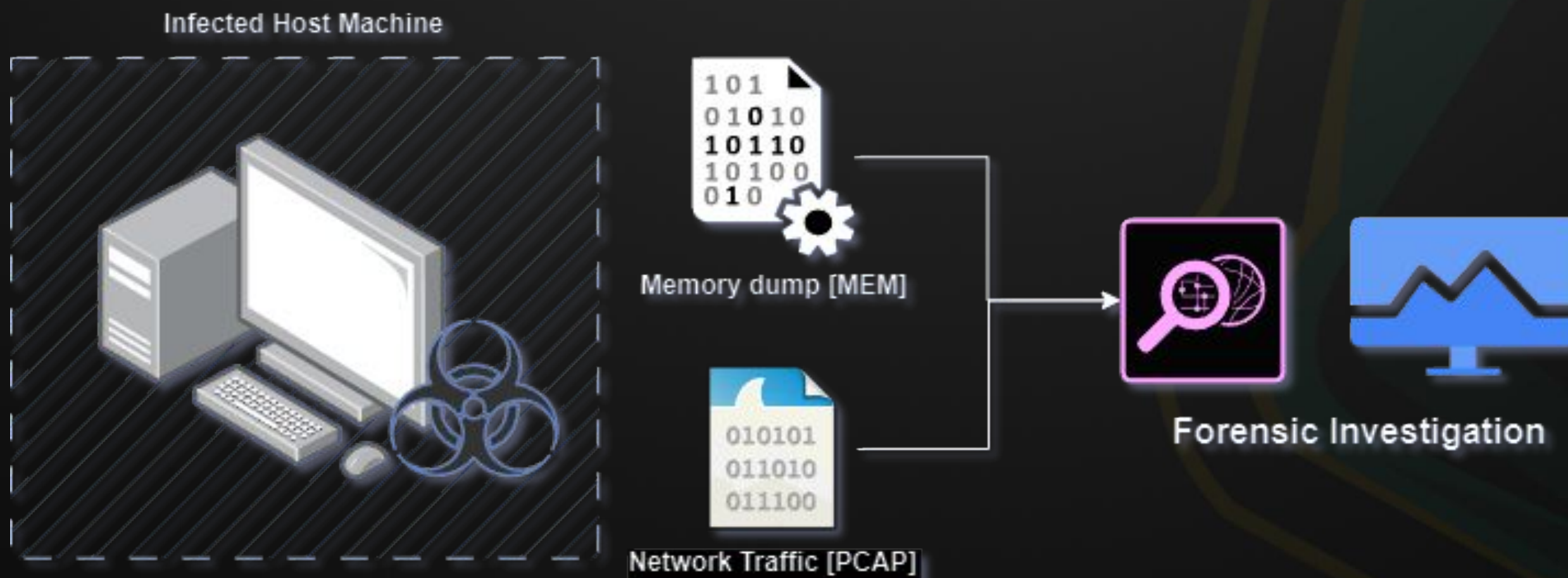
The overall overview of Cyber Defence has been grouped into three categories

- Malicious File Drop
- Initial Communication
- Handshake
- Command Transmission
- Data Exfiltration
- Beaconing



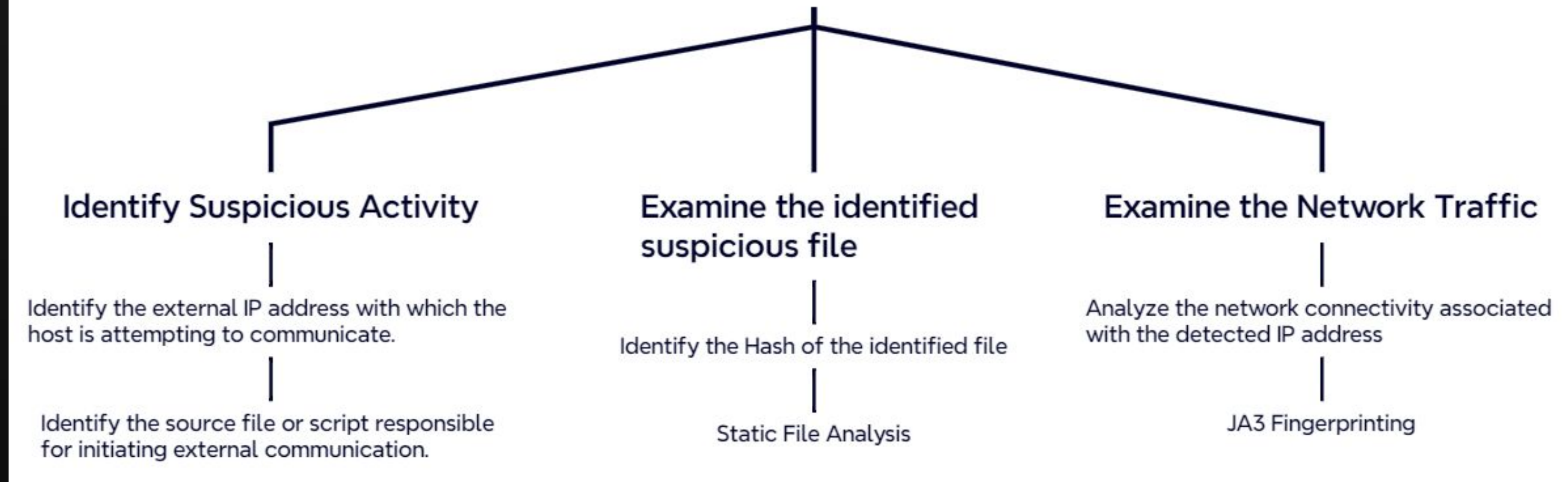
## Investigating C2 targeted attacks

The primary objective is to conduct a thorough investigation into a suspected memory dump [mem] and network dump [PCAP] with the goal of identifying the source, scope, and impact of the attack.



# Investigative Mind Map

## Command & Control Investigation Mind Map



## Gather the information

The first step of investigation begins by analysing the basic information such as the operating system version, architecture, and system configuration can aid in accurately identifying the system being analysed.

Use the following command to obtain the basic information for the detected image dump.

```
sudo python3 vol.py -f <file_path> windows.info.Info
```

# Identify the external IP address and Suspicious file

After gathering the basic information the next step is to identify the the IP which is trying to communicate over the reported port **8888**, Additionally we also identified that the file **AMATEUR\_TOOTHB** is been associated with the same activity

```
(kali@kali)-[~/Desktop/New Folder/volatility3-2.4.1]
└─$ sudo python3 vol.py -f /home/kali/Downloads/memdump.mem windows.netstat.NetStat
Volatility 3 Framework 2.4.1
Progress: 100.00
PDB scanning finished
Offset Proto LocalAddr LocalPort ForeignAddr ForeignPort State PID Owner Created
0xc18d521ee4e0 TCPv4 192.168.14.238 50355 152.199.39.108 443 CLOSE_WAIT 7636 WWAHost.exe 2024-04-24 14:06:23.000000
0xc18d4c04f4a0 TCPv4 192.168.14.238 50550 192.168.14.202 8888 ESTABLISHED 6236 AMATEUR_TOOTHB 2024-04-24 14:23:20.000000
0xc18d4a8c2460 TCPv4 192.168.14.238 49747 20.198.119.143 443 ESTABLISHED 2944 svchost.exe 2024-04-25 02:17:09.000000
0xc18d4f2b0010 TCPv4 192.168.14.238 50554 20.189.173.1 443 ESTABLISHED 3524 msedge.exe 2024-04-24 14:26:08.000000
0xc18d51dd29b0 TCPv4 192.168.14.238 50356 152.199.39.108 443 CLOSE_WAIT 7636 WWAHost.exe 2024-04-24 14:06:23.000000
0xc18d51ff5b20 TCPv4 192.168.14.238 50353 152.199.39.108 443 CLOSE_WAIT 7636 WWAHost.exe 2024-04-24 14:06:23.000000
0xc18d4f4862a0 TCPv4 192.168.14.238 50555 20.44.10.123 443 ESTABLISHED 3524 msedge.exe 2024-04-24 14:26:09.000000
0xc18d4fbcea20 TCPv4 192.168.14.238 50350 152.195.38.76 80 CLOSE_WAIT 7636 WWAHost.exe 2024-04-24 14:06:23.000000
```

## Examine the identified suspicious file

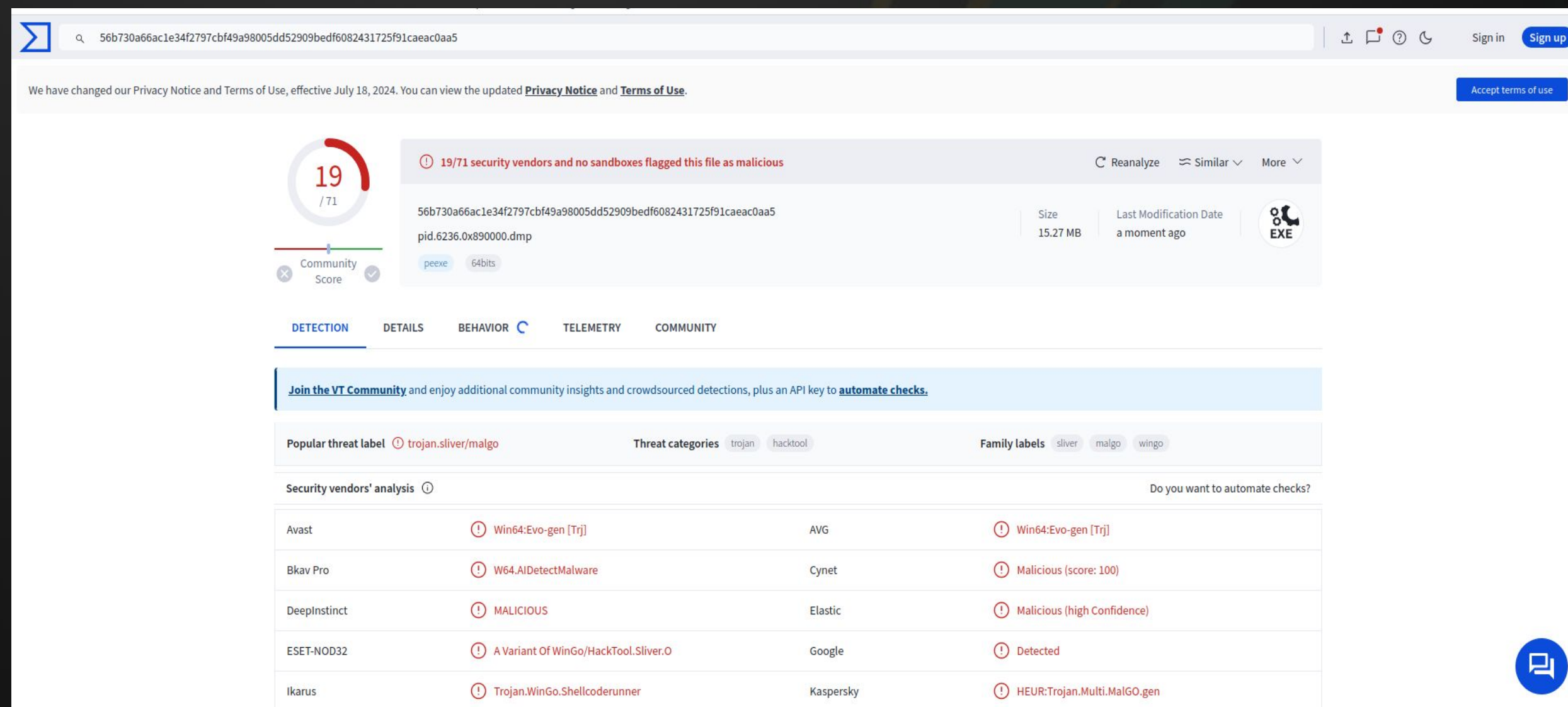
Next step of the investigation is to download the detected suspicious file for further investigation to, using our raw image file we can easily retrieve the file using the PID value associate with it, execute the below mentioned commands and observed the result

```
sudo python3 vol.py -f <file_path>/suspected.raw windows.pslist --pid 6236 --dump
```

```
(kali@kali)-[~/Desktop/New Folder/volatility3-2.4.1]
└─$ sudo python3 vol.py -f /home/kali/Downloads/memdump.mem windows.pslist --pid 6236 --dump
Volatility 3 Framework 2.4.1
Progress: 100.00      PDB scanning finished
PID      PPID      ImageFileName      Offset(V)      Threads  Handles  SessionId      Wow64  CreateTime      ExitTime      File output
6236     4444     AMATEUR_TOOTHB     0xc18d53262340  9        -         1              False  2024-04-24 14:23:20.000000  N/A      pid.6236.0x890000.dmp
```

# Determine the File Repudiation

The next step is to analyse the dump via virustotal, upload the extracted dump directly into the virustotal for further analysis



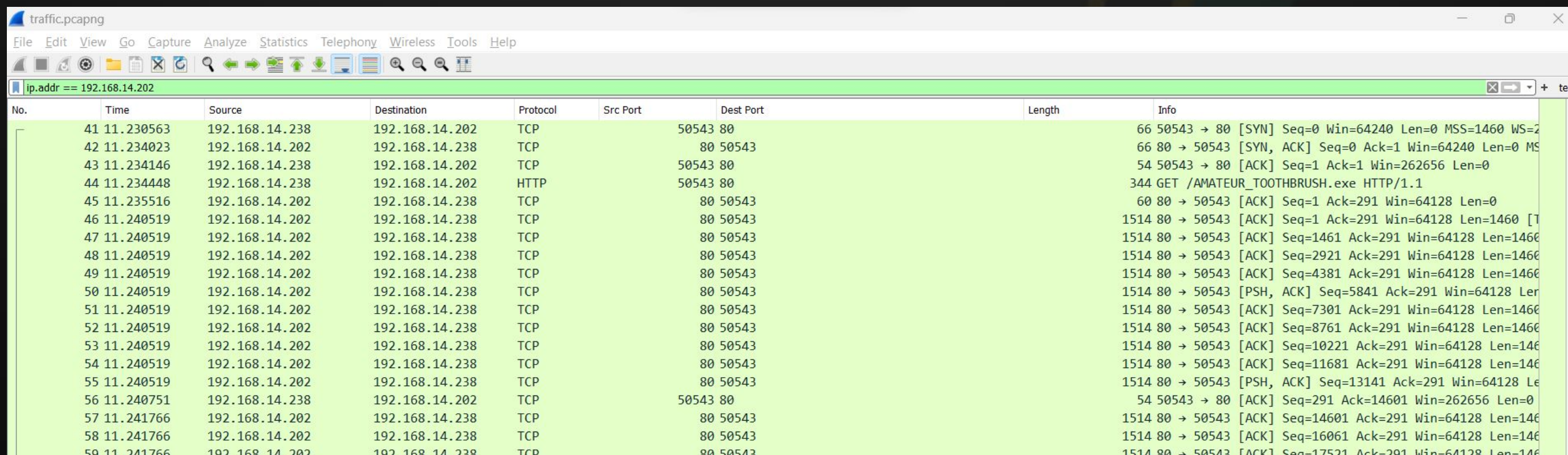
The screenshot shows the VirusTotal analysis page for a file. The file name is `pid.6236.0x890000.dmp` and its size is 15.27 MB. The analysis shows that 19 out of 71 security vendors have flagged the file as malicious. The file is categorized as a trojan, specifically `trojan.sliver/malgo`. The security vendors' analysis table is as follows:

Vendor	Detection	Vendor	Detection
Avast	Win64:Evo-gen [Trj]	AVG	Win64:Evo-gen [Trj]
Bkav Pro	W64.AIDetectMalware	Cynet	Malicious (score: 100)
DeepInstinct	MALICIOUS	Elastic	Malicious (high Confidence)
ESET-NOD32	A Variant Of WinGo/HackTool.Sliver.O	Google	Detected
Ikarus	Trojan.WinGo.Shellcoderunner	Kaspersky	HEUR:Trojan.Multi.MalGO.gen

# Examine the network traffic

From our previous investigation we identified the IP : **192.168.14.202** which is been associated with the activity, execute the below query to specifically retrieve the activity associated with the IP

```
ip.addr == <IP_Address>
```



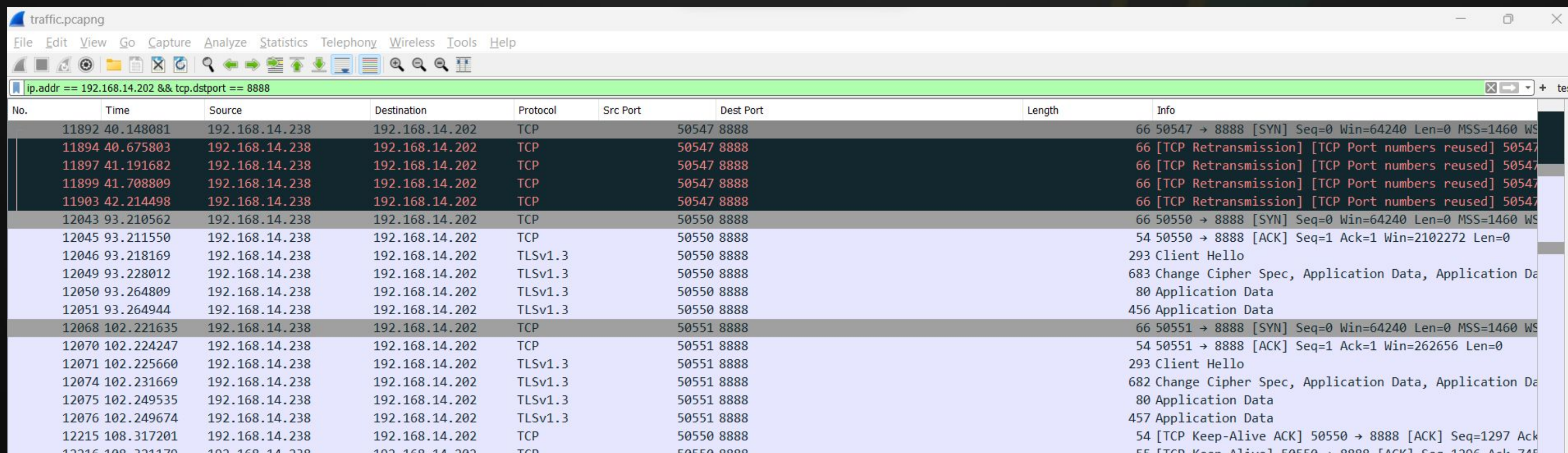
No.	Time	Source	Destination	Protocol	Src Port	Dest Port	Length	Info
41	11.230563	192.168.14.238	192.168.14.202	TCP	50543	80	66	66 50543 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2
42	11.234023	192.168.14.202	192.168.14.238	TCP	80	50543	66	66 80 → 50543 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MS
43	11.234146	192.168.14.238	192.168.14.202	TCP	50543	80	54	54 50543 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
44	11.234448	192.168.14.238	192.168.14.202	HTTP	50543	80	344	344 GET /AMATEUR_TOOTHBRUSH.exe HTTP/1.1
45	11.235516	192.168.14.202	192.168.14.238	TCP	80	50543	60	60 80 → 50543 [ACK] Seq=1 Ack=291 Win=64128 Len=0
46	11.240519	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=1 Ack=291 Win=64128 Len=1460 [T
47	11.240519	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=1461 Ack=291 Win=64128 Len=1460
48	11.240519	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=2921 Ack=291 Win=64128 Len=1460
49	11.240519	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=4381 Ack=291 Win=64128 Len=1460
50	11.240519	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [PSH, ACK] Seq=5841 Ack=291 Win=64128 Len=1460
51	11.240519	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=7301 Ack=291 Win=64128 Len=1460
52	11.240519	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=8761 Ack=291 Win=64128 Len=1460
53	11.240519	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=10221 Ack=291 Win=64128 Len=1460
54	11.240519	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=11681 Ack=291 Win=64128 Len=1460
55	11.240519	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [PSH, ACK] Seq=13141 Ack=291 Win=64128 Len=1460
56	11.240751	192.168.14.238	192.168.14.202	TCP	50543	80	54	54 50543 → 80 [ACK] Seq=291 Ack=14601 Win=262656 Len=0
57	11.241766	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=14601 Ack=291 Win=64128 Len=1460
58	11.241766	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=16061 Ack=291 Win=64128 Len=1460
59	11.241766	192.168.14.202	192.168.14.238	TCP	80	50543	1514	1514 80 → 50543 [ACK] Seq=17521 Ack=291 Win=64128 Len=1460



# Examine the network traffic

After identifying the malicious download request our next step is to retrieve the activity associated with port reported port **8888**, execute the below query and observe the result

```
ip.addr == <IP_Address> && tcp.dstport == <Port>
```



No.	Time	Source	Destination	Protocol	Src Port	Dest Port	Length	Info
11892	40.148081	192.168.14.238	192.168.14.202	TCP	50547	8888	66	50547 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
11894	40.675803	192.168.14.238	192.168.14.202	TCP	50547	8888	66	[TCP Retransmission] [TCP Port numbers reused] 50547
11897	41.191682	192.168.14.238	192.168.14.202	TCP	50547	8888	66	[TCP Retransmission] [TCP Port numbers reused] 50547
11899	41.708809	192.168.14.238	192.168.14.202	TCP	50547	8888	66	[TCP Retransmission] [TCP Port numbers reused] 50547
11903	42.214498	192.168.14.238	192.168.14.202	TCP	50547	8888	66	[TCP Retransmission] [TCP Port numbers reused] 50547
12043	93.210562	192.168.14.238	192.168.14.202	TCP	50550	8888	66	50550 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
12045	93.211550	192.168.14.238	192.168.14.202	TCP	50550	8888	54	50550 → 8888 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
12046	93.218169	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	293	Client Hello
12049	93.228012	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	683	Change Cipher Spec, Application Data, Application Da
12050	93.264809	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	80	Application Data
12051	93.264944	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	456	Application Data
12068	102.221635	192.168.14.238	192.168.14.202	TCP	50551	8888	66	50551 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
12070	102.224247	192.168.14.238	192.168.14.202	TCP	50551	8888	54	50551 → 8888 [ACK] Seq=1 Ack=1 Win=262656 Len=0
12071	102.225660	192.168.14.238	192.168.14.202	TLSv1.3	50551	8888	293	Client Hello
12074	102.231669	192.168.14.238	192.168.14.202	TLSv1.3	50551	8888	682	Change Cipher Spec, Application Data, Application Da
12075	102.249535	192.168.14.238	192.168.14.202	TLSv1.3	50551	8888	80	Application Data
12076	102.249674	192.168.14.238	192.168.14.202	TLSv1.3	50551	8888	457	Application Data
12215	108.317201	192.168.14.238	192.168.14.202	TCP	50550	8888	54	[TCP Keep-Alive ACK] 50550 → 8888 [ACK] Seq=1297 Ack
12216	108.321170	192.168.14.238	192.168.14.202	TCP	50550	8888	55	[TCP Keep-Alive] 50550 → 8888 [ACK] Seq=1296 Ack=745

# Examine the network traffic

Based on the findings, we've detected TLS communication between the compromised host and the external network. Initially, the requests and responses followed the typical pattern, but in our case, we're observing multiple occurrences of client hello and application data exchanges, which deviate from the norm.

12046	93.218169	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	293 Client Hello
12049	93.228012	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	683 Change Cipher Spec, Application Data, Application Data, Application Data
12050	93.264809	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	80 Application Data
12051	93.264944	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	456 Application Data
12068	102.221635	192.168.14.238	192.168.14.202	TCP	50551	8888	66 50551 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
12070	102.224247	192.168.14.238	192.168.14.202	TCP	50551	8888	54 50551 → 8888 [ACK] Seq=1 Ack=1 Win=262656 Len=0
12071	102.225660	192.168.14.238	192.168.14.202	TLSv1.3	50551	8888	293 Client Hello
12074	102.231669	192.168.14.238	192.168.14.202	TLSv1.3	50551	8888	682 Change Cipher Spec, Application Data, Application Data, Application Data
12075	102.249535	192.168.14.238	192.168.14.202	TLSv1.3	50551	8888	80 Application Data
12076	102.249674	192.168.14.238	192.168.14.202	TLSv1.3	50551	8888	457 Application Data
12215	108.317201	192.168.14.238	192.168.14.202	TCP	50550	8888	54 [TCP Keep-Alive ACK] 50550 → 8888 [ACK] Seq=1297 Ack=745 Win=2101504 Len=0
12216	108.321179	192.168.14.238	192.168.14.202	TCP	50550	8888	55 [TCP Keep-Alive] 50550 → 8888 [ACK] Seq=1296 Ack=745 Win=2101504 Len=1
12269	117.277616	192.168.14.238	192.168.14.202	TCP	50551	8888	54 [TCP Keep-Alive ACK] 50551 → 8888 [ACK] Seq=1297 Ack=744 Win=261888 Len=0
12270	117.321054	192.168.14.238	192.168.14.202	TCP	50551	8888	55 [TCP Keep-Alive] 50551 → 8888 [ACK] Seq=1296 Ack=744 Win=261888 Len=1
12277	122.402975	192.168.14.238	192.168.14.202	TCP	50550	8888	54 50550 → 8888 [ACK] Seq=1297 Ack=816 Win=2101504 Len=0
12278	122.404923	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	80 Application Data
12279	122.405047	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	111 Application Data
12285	127.256792	192.168.14.238	192.168.14.202	TCP	50550	8888	54 50550 → 8888 [ACK] Seq=1380 Ack=891 Win=2101504 Len=0
12286	127.358363	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	80 Application Data
12287	127.358510	192.168.14.238	192.168.14.202	TLSv1.3	50550	8888	1648 Application Data
12291	132.332387	192.168.14.238	192.168.14.202	TCP	50551	8888	55 [TCP Keep-Alive] 50551 → 8888 [ACK] Seq=1296 Ack=744 Win=261888 Len=1
12298	142.366824	192.168.14.238	192.168.14.202	TCP	50550	8888	55 [TCP Keep-Alive] 50550 → 8888 [ACK] Seq=2999 Ack=891 Win=2101504 Len=1
12300	142.368574	192.168.14.238	192.168.14.202	TCP	50550	8888	54 [TCP Keep-Alive ACK] 50550 → 8888 [ACK] Seq=3000 Ack=891 Win=2101504 Len=0

## Examine the network traffic

Typically, these patterns are observed in **C2 communication**. The C2 sends **client hello** requests to check if the targeted host is active and accessible. When the attacker attempts to push malicious commands, they are sent as application data. It's worth noting that even if you attempt to read the packets, the information inside is encrypted and not in a readable format.

```
Wireshark - Packet 12286 - traffic.pcapng
> Frame 12286: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface \Device\NPF_{17C33C3C-2E4D-4BC1
> Ethernet II, Src: PcsCompu_ab:7d:97 (08:00:27:ab:7d:97), Dst: PcsCompu_73:6e:ba (08:00:27:73:6e:ba)
> Internet Protocol Version 4, Src: 192.168.14.238, Dst: 192.168.14.202
> Transmission Control Protocol, Src Port: 50550, Dst Port: 8888, Seq: 1380, Ack: 891, Len: 26
> Transport Layer Security

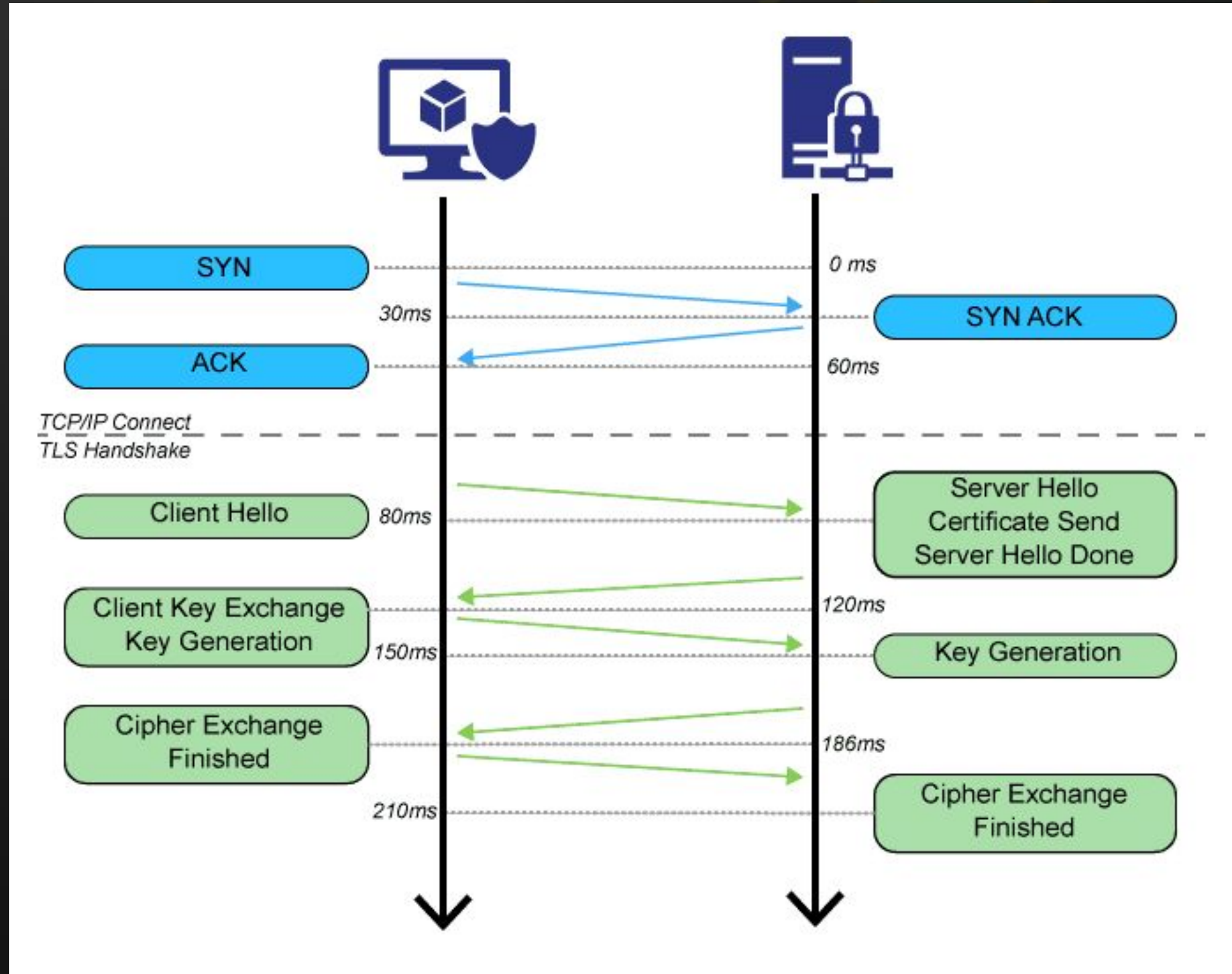
0000  08 00 27 73 6e ba 08 00 27 ab 7d 97 08 00 45 00  ...'sn... '...}...E.
0010  00 42 7b c9 40 00 80 06 00 00 c0 a8 0e ee c0 a8  -B{.@... ..
0020  0e ca c5 76 22 b8 0b 63 1d 93 95 4a b1 e5 50 18  ...v"...c ...J..P.
0030  20 11 9f 3d 00 00 17 03 03 00 15 2e 08 c7 89 76  ...=.....v
0040  63 c1 ee bc ab 07 d1 9c 51 13 f1 40 39 88 c4 c2  c.....Q..@9...
```

## JA3 Fingerprinting

JA3 correlation methodology will help us in identifying and categorizing different types of software or libraries based on their unique fingerprints generated during the handshake process, JA3 typically get generated based on their cryptographic characteristics of the SSL/TLS handshake.

Each unique SSL/TLS handshake will result in unique JA3 fingerprinting.

# Working of TLS Handshake



## Working of JA3

JA3 fingerprinting value is calculated by collecting the decimal values of the bytes for the following fields.

- Version
- Accepted Ciphers
- List of Extensions
- Elliptic Curves
- Elliptic Curve Formats

The collected decimal values are then hash to MD5 format and resulted with 32 character fingerprints.

```
▼ Transport Layer Security
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 512
  ▼ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 508
    Version: TLS 1.2 (0x0303)
    > Random: f960dc3adc8edd6630139f558a1a0bf8259392ab92243a64716b61bd6b83cca9
    Session ID Length: 32
    Session ID: dd6c51fa66ac1f3d511c185198f4b8520cbf6a7712016f19ef49bcfcff942aad
    Cipher Suites Length: 36
    > Cipher Suites (18 suites)
    Compression Methods Length: 1
    > Compression Methods (1 method)
    Extensions Length: 399
    > Extension: server name (len=41)
    > Extension: ec_point_formats (len=4)
    > Extension: supported_groups (len=22)
    > Extension: session_ticket (len=0)
    > Extension: application_layer_protocol_negotiation (len=11)
    > Extension: encrypt_then_mac (len=0)
    > Extension: extended_master_secret (len=0)
    > Extension: post_handshake_auth (len=0)
    > Extension: signature_algorithms (len=42)
    > Extension: supported_versions (len=5)
    > Extension: psk_key_exchange_modes (len=2)
    > Extension: key_share (len=38)
```

# Co-relating the JA3

On the Client Hello request we can identify the JA3 section under the Transport Layer Security

This generated JA3 value can be further searched over internet to determine the whether its been associated with any other malicious activity

```
▼ Transport Layer Security
  ▼ TLSv1.3 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 234
  ▼ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 230
    Version: TLS 1.2 (0x0303)
    Random: 9b13a36a903efce16a8405ae74f79e5670457d8f7883bc5939331c34ffe46c44
    Session ID Length: 32
    Session ID: 63cd29a7aaab9231b041de3d981999bc705842b2fda591d061b29754eefc5c1a
    Cipher Suites Length: 38
  > Cipher Suites (19 suites)
  > Compression Methods Length: 1
  > Compression Methods (1 method)
  > Extensions Length: 119
  > Extension: status_request (len=5)
  > Extension: supported_groups (len=10)
  > Extension: ec_point_formats (len=2)
  > Extension: signature_algorithms (len=26)
  > Extension: renegotiation_info (len=1)
  > Extension: signed_certificate_timestamp (len=0)
  > Extension: supported_versions (len=5)
  > Extension: key_share (len=38)
  [JA3 Fullstring: 771,49195-49199-49196-49200-52393-52392-49161-49171-49162-49172-156-157-47-53-49170-10-4865-4866-4867,5-10-11-13-65281-18-43-51,29-23-24-25,0]
  [JA3: 19e29534fd49dd27d09234e639c4057e]
```

## JA3 Value

**19e29534fd49dd27d09234e639c4057e**

## Cyber Defence Analyst : CCDA

The Certified Cyber Defence Analyst (CCDA) training offers an investigative approach to Blue Teaming. It's designed to equip participants with the necessary knowledge and skills to become effective in threat detection and investigation as part of a Blue Team.

Threat Detection & Its Investigation

Enhance the real time investigation skills

Hands-on investigations

Cyber Defence Labs

Multiple Investigative mind map

Incident Response Strategies



**Cyber Defense  
Analyst  
[CCDA]**



# BTF Lab Overview



**Cyber Defense Team**

# Web based Investigation and Analysis



**Email Header Analysis**



**Credential Phishing Investigation**



**Suspicious Attachment Phishing Investigation**



**Suspicious Macros Phishing Investigation**

# Web based Investigation and Analysis



**Admin Page access detected**



**Subdomain Enumeration**



**SQL-Map activity detected**



**File Inclusion activity  
detected**



**Command Injection activity  
detected**



**Vulnerability Enumeration activity  
detected**

# Network based Investigation and Analysis



**NMAP Detection**



**DOS investigation**



**Suspected Data exfiltration detected**



**Netcat activity detected**

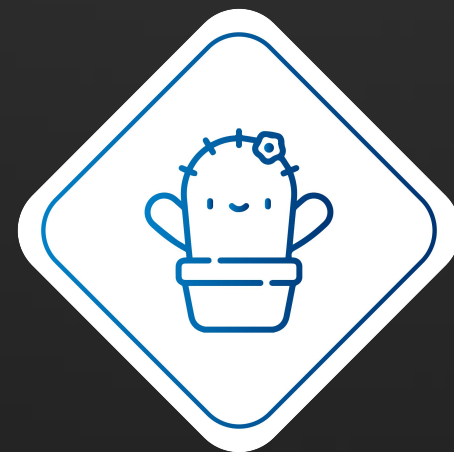


**Service brute forcing**

# Host Based Attack Investigation Challenges



**Suspicious File activity detected**



**Suspected .SH file detected**



**Suspicious Scheduled task detected**



**Suspicious Linpeas activity detected**



**Multiple Remote  
Failed Login Detected**



**Suspicious PowerShell activity  
detected**

# Certification Procedure

*Enroll in CCDA  
On-Demand Course*



*Complete Study Materials  
[Videos + PDF]*



*Schedule 30 Days  
Lab Access*



*Take 24hrs  
Hands-on Exam*



*Share the exam report [PDF]  
in next 24 Hrs*



*Clear 70% Passing Criteria  
Earn Accredible Badge*

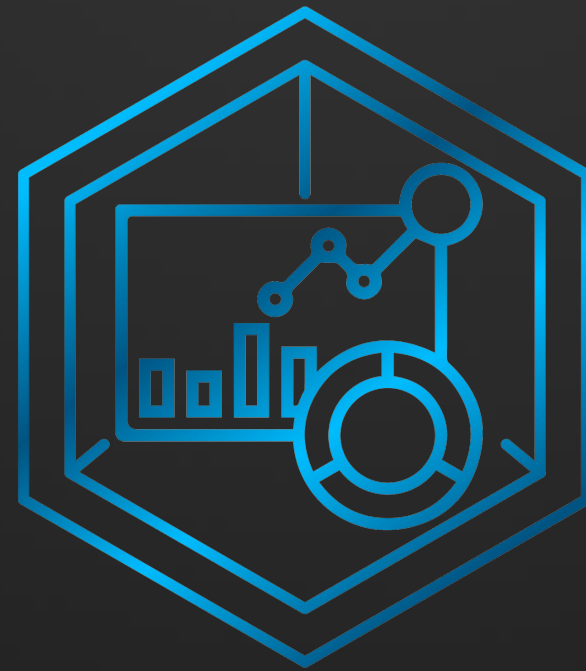


## Giveaway **Alert**

### 5 Certified Cyber Defence Analyst | CCDA

We're giving away Latest Launch "Cyber Defence Analyst [CCDA]"





# Thank You

For Professional Red Team / Blue Team / Purple Team / Cloud Cyber Range labs /  
Trainings, please contact

**support@cyberwarfare.live**

To know more about our offerings, please visit: <https://cyberwarfare.live>