Network and Memory Forensics

HAIDER MUSTAFA

Table of Contents

NETWORK FORENSICS

INTRODUCTION

This report details the findings of a network forensics investigation following an Intrusion Detection System (IDS) alert within the company network. The objective of this analysis was to determine the root cause of the infection, identify the responsible malware, and gather critical Indicators of Compromise (IOCs) from the provided Packet Capture (pcap) file, named malware-traffic-analysis. The scope of the investigation was limited to the provided pcap file, which is suspected of containing Windows-based malware activity.

The analysis employed a systematic methodology focusing on network traffic patterns, payload delivery, Command and Control (C2) communication, and passive host fingerprinting. The key deliverables of this investigation include identifying the malware family, associated C2 server IP addresses, ports, malicious files and their hashes, the infected host name, and the compromised user account.

LAB PREPARATION

Forensic Environment

Due to the nature of the infection being a suspected Windows-based malware, the analysis was conducted entirely within a non-Windows environment to mitigate any risk of accidental execution or compromise of the forensic workstation. Kali Linux was exclusively used as the operating system for all tools and analysis commands.

Tools Used

The following network analysis and forensic tools were utilized to process and analyse the provided packet capture file (malware-traffic-analysis.pcap):

- **Wireshark**: Used for deep packet inspection, filtering (e.g., http, tls.handshake.type==11), statistical analysis, flow tracking, and extracting the embedded malicious object from the HTTP stream.
- **NetworkMiner:** Employed for passive operating system and user fingerprinting, which successfully identified the infected Windows host name and the associated user account

- **Linux Command Line Utilities (Kali):**
  File: Used to determine the true file type of the extracted payload, confirming it was a DLL executable despite having a .png extension.
  Sha256sum: Used to generate a cryptographic hash of the extracted file for identification and threat intelligence matching.
- **Online Threat Intelligence:**
  VirusTotal: https://www.virustotal.com
  Malware Bazaar: https://bazaar.abuse.ch/

INVESTIGATION

Using wireshark, I opened the pcap file, the first step was to identify the affected host(s). Clicking on statistics>Endpoints on wireshark and selecting IPv4 gave a list of all IPv4 Ips in the network.

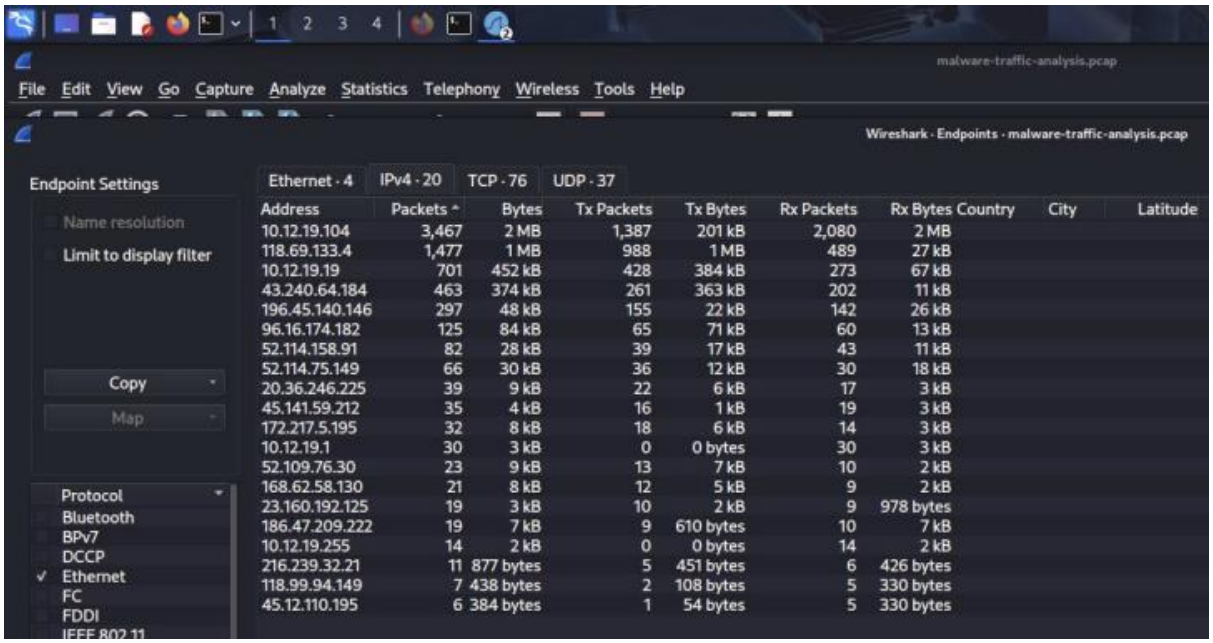

Figure 1 IPv4 address on the network

We are only interested in the private Ips (10.0.0.0 – 10.255.255.255, 172.16.0.0 – 172.31.255.255, 192.168.0.0 – 192.168.255.255) to locate the affected host. This network has the following private Ips:

10.12.19.104

10.12.19.19

10.12.19.1

10.12.19.255

From the list, 10.12.19.104 seems to have a higher amount of activity with 3,467 packets so I decided to take a closer look at it with the http filter. The results are in the Figure 2. 2 HTTP filter on host 10.12.19.104 below.

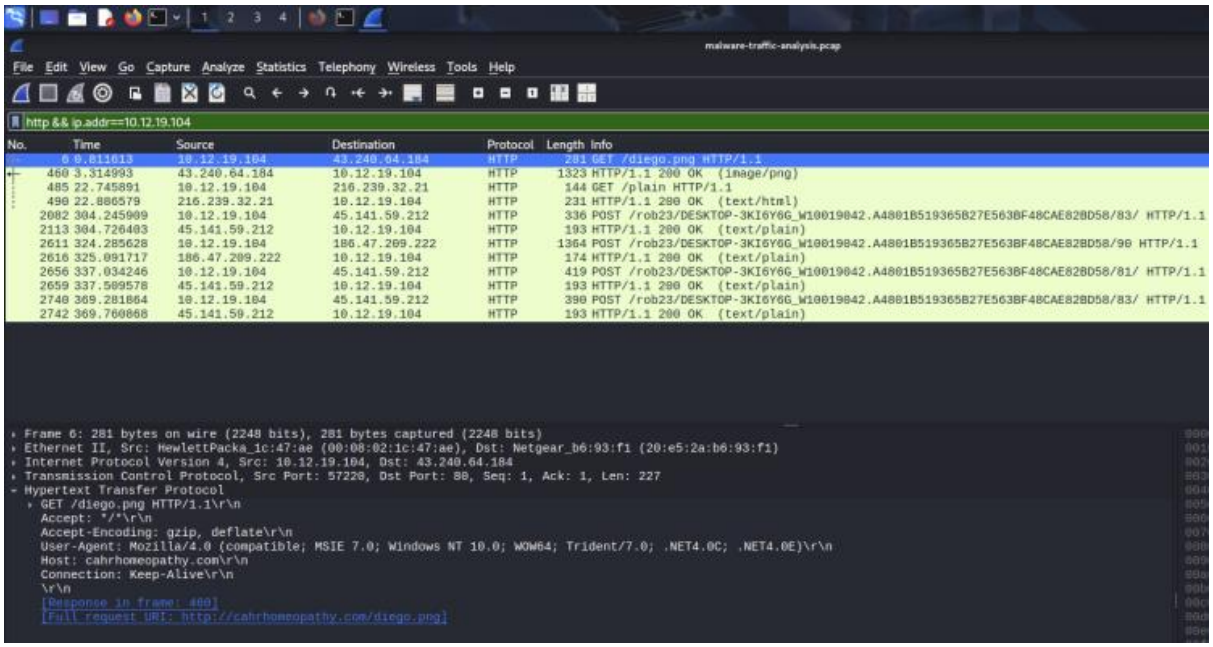

Figure 2 HTTP filter on host 10.12.19.104

I immediately noticed that this host (10.12.19.104) sent an HTTP request to a website cahrhomeopathy.com with IP address 43.240.64.184 and the URI of the request was cahrhomeopathy.com/deigo.png, date and time of this was Dec. 19th, 2020, at 03:54:27 UTC.

Using virus total, I scanned the website.

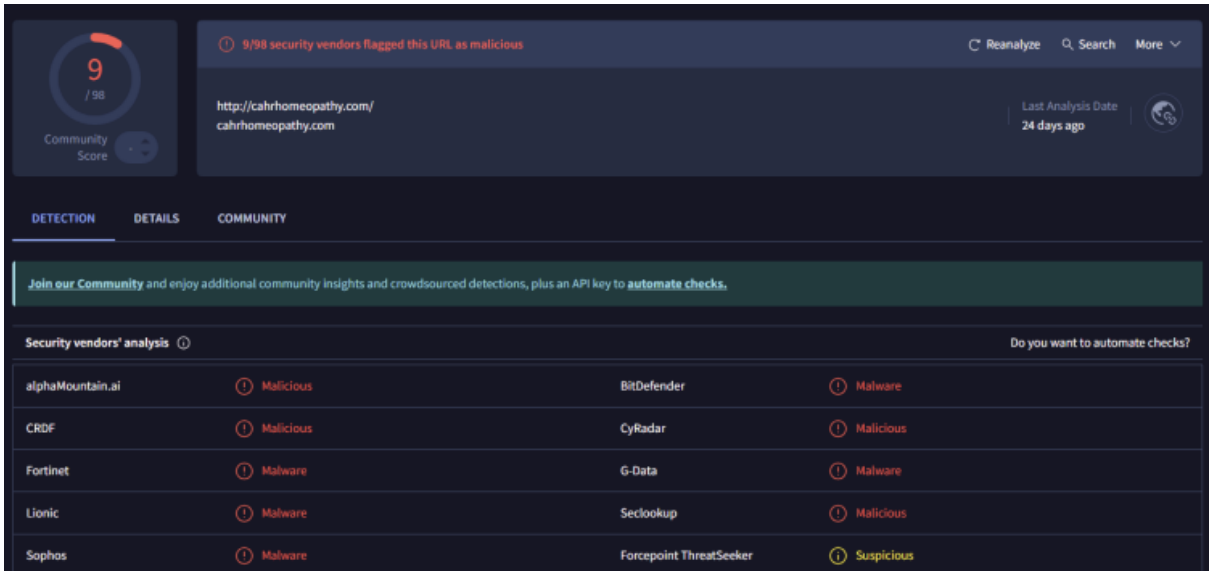

Figure 3 Virustotal results for cahrhomeopathy.com

https://www.virustotal.com/gui/url/7cef4f741850b6fbe17d76d848b4ab2893a83ae4492bdf481606ca758c92f6d8

This website has been flagged by 9 vendors as malicious, indicating malware. So, I extracted the diego.png object in the HTTP request to the malicious website by going to **file>Export Objects>HTTP**, selecting the malicious HTTP request and clicking save.

After extracting the diego.png object, I examined it using File function on Kali and discovered it was not a PNG file but a DLL file, I also extracted the hash of the file (da1ae69acf1b97bfac587addc9266155342bf8f2a7a80e0d09df9a577c39f7f9) using sha256 for further investigation.

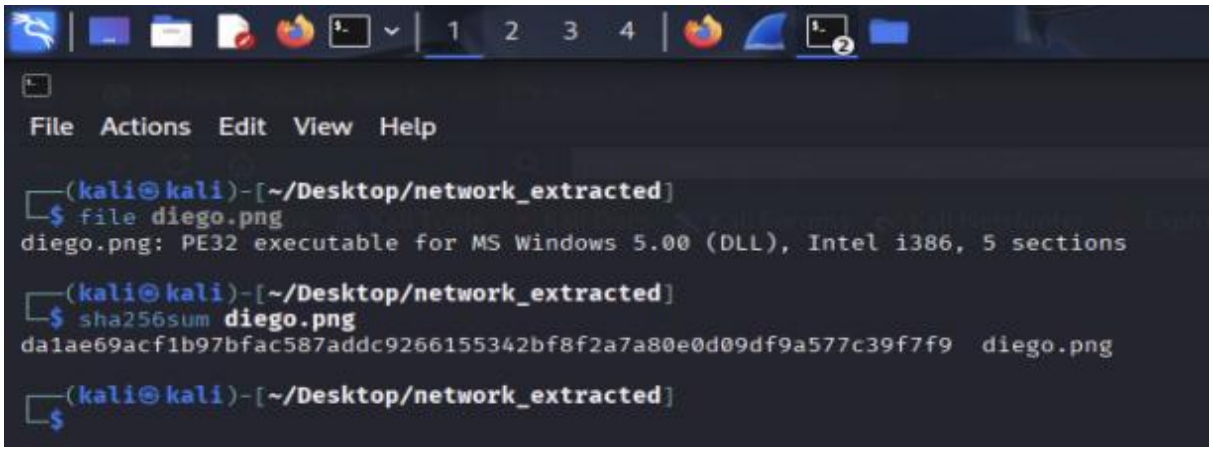

Figure 4 Diego.png file type and hash results

Using Malware Bazaar and Virustotal, I was able to confirm that diego.dll hidden as diego.png is a malicious DLL file that is a TrickBot belonging to the Trojan family, in accordance with (cisa.gov) that TrickBots are Trojans. This also confirms that 10.12.19.104 is the affected host.

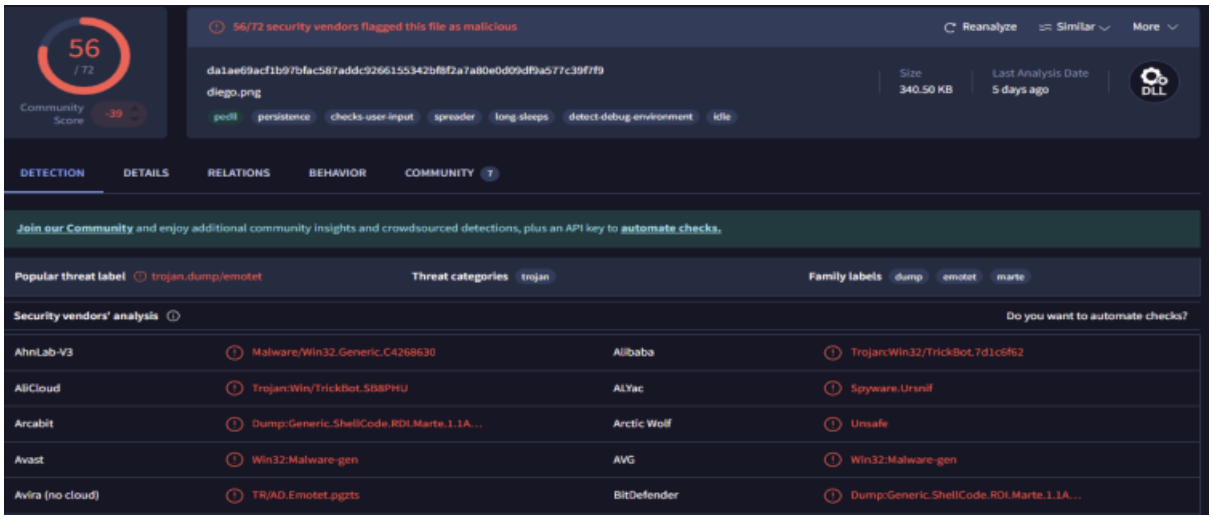

Figure 5 Snippet of Diego.png(dll) hash results on VirusTotal

https://www.virustotal.com/gui/file/da1ae69acf1b97bfac587addc9266155342bf8f2a7a80e0d09df9a577c39f7f9

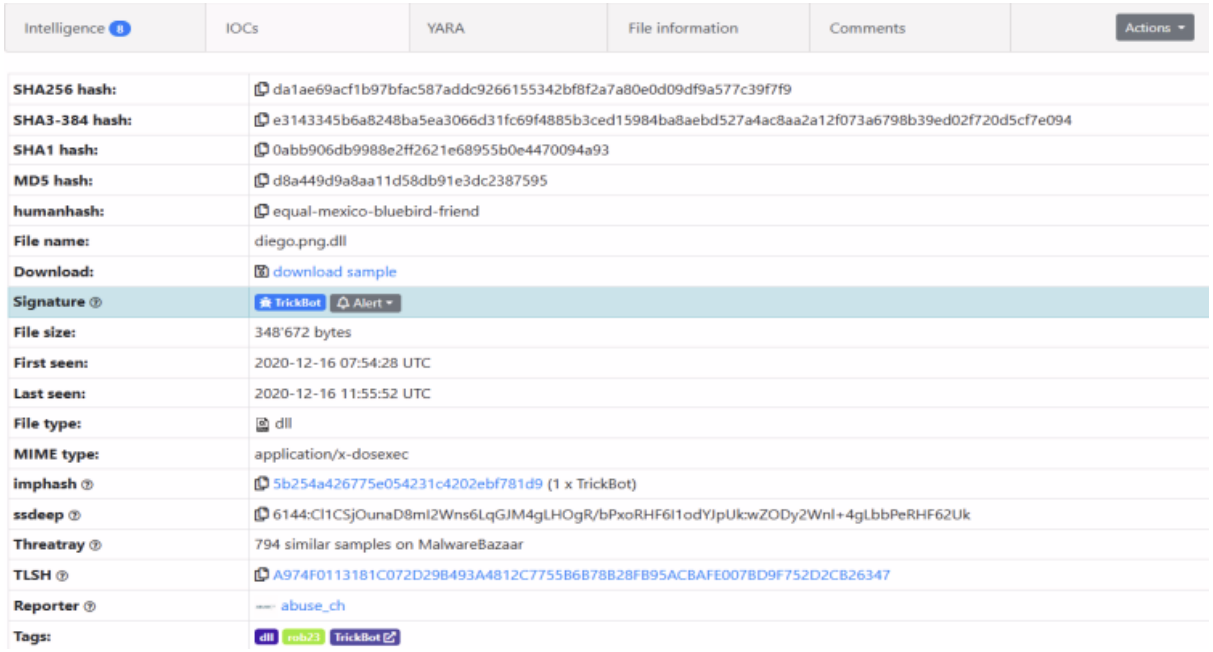

Figure 6 Snippet of Diego.png(dll) hash results on Malware Balzaar

https://bazaar.abuse.ch/sample/da1ae69acf1b97bfac587addc9266155342bf8f2a7a80e0d09df9a577c39f7f9/

According to Rossouw (2025), most C2 servers exchange certificate with the host domain so I used **tls.handshake.type==11** filter to see the IP addresses that had a tls handshake in the network. Results are in **Error! Reference source not found**.

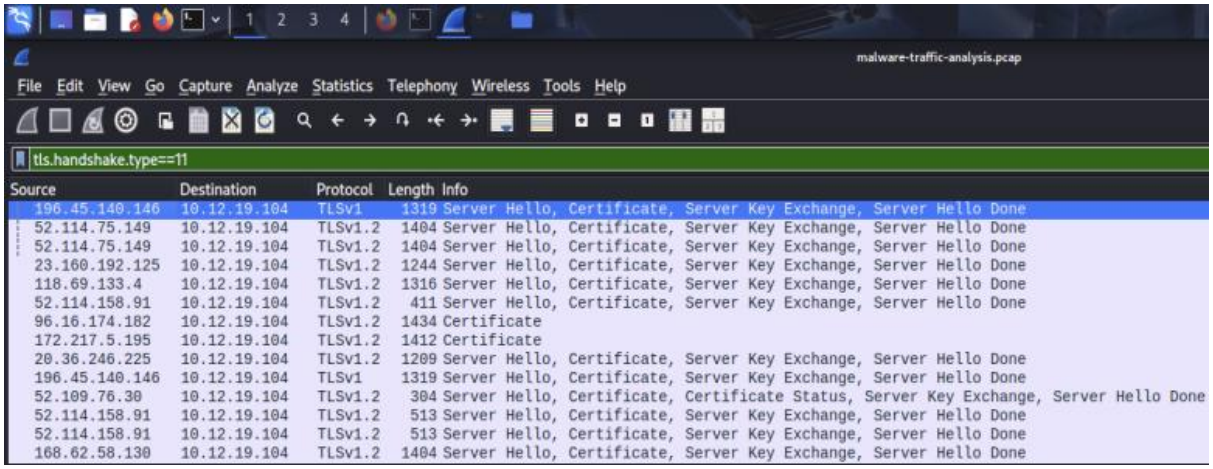

Figure 7 TLS handshake filter result

From the filter results, a total of 10 IP addresses had a TLS handshake in the network, and all were with our affected host (10.12.19.104), so I analysed the IP addresses by looking at the issuer of their certificate in the Transport Layer Security session of each packet. Out of the 10 IP addresses, 3 of them had suspicious issuer organization name and the rest IP addresses were mostly issued by Microsoft, one by Cyber Trust and one by GlobalSign.

The 3 addresses with a suspicious certificate issuer are analysed below:

1. **196.45.140.146**
   Certificate issuer: Internet Widgits Pty
   Port 449
   Communication Protocols used: TCP, TLSV1
   Host communicated with: Only the affected host (10.12.19.104)
   Total packets: 297

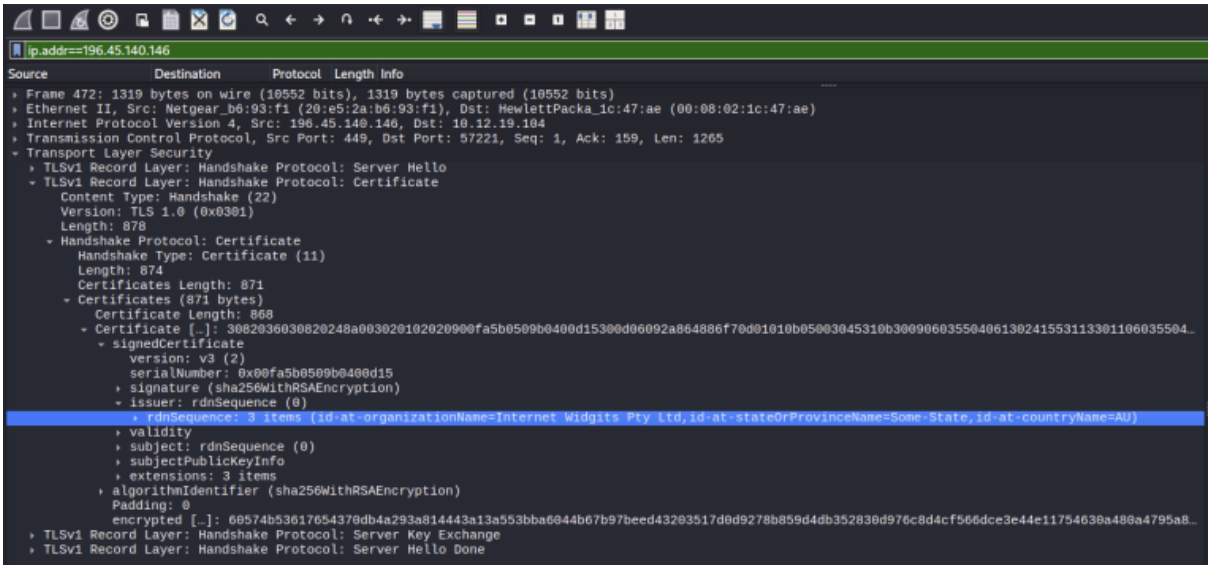

Figure 8 Suspected C2 server (1) certificate

2. 23.160.192.125
   Certificate Issuer: Qjvoobim
   Port: 447
   Communication Protocols used: TCP, TLSV1.2
   Host communicated with: Only the affected host (10.12.19.104)
   Total packets: 19

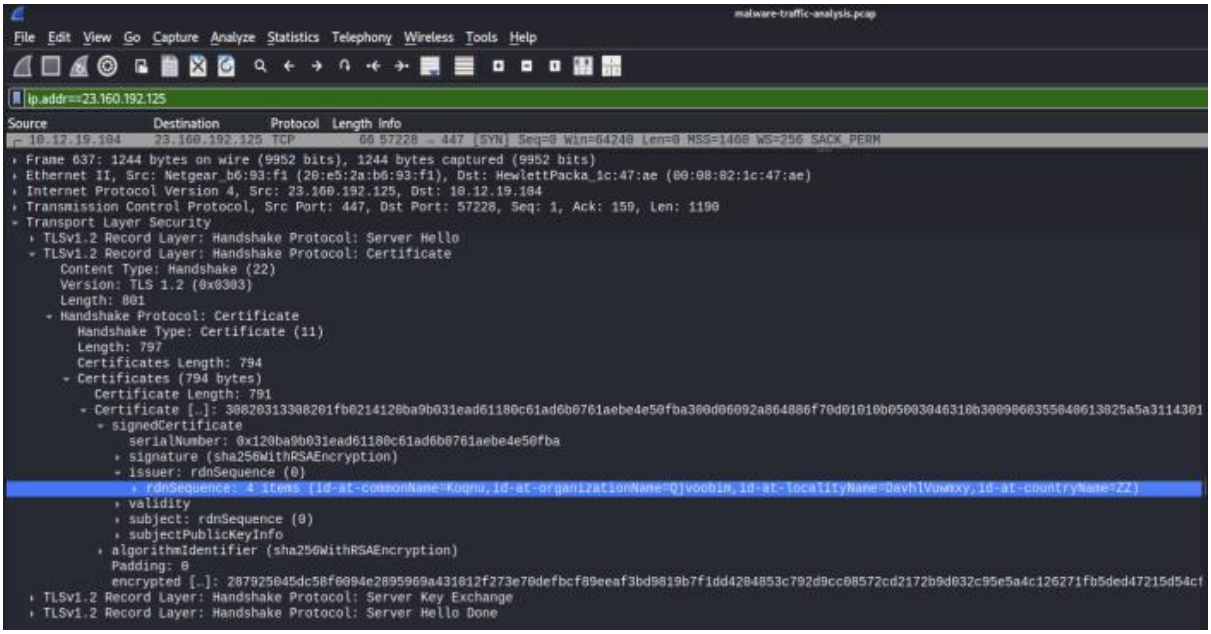

Figure 9 Suspected C2 server (2) certificate

3. **118.69.133.4**
   Certificate Issuer: Internet Widgits Pty Ltd
   Port: 447
   Communication Protocols used: TCP, TLSV1.2
   Host communicated with: Only the affected host (10.12.19.104)
   Total packets: 1,477

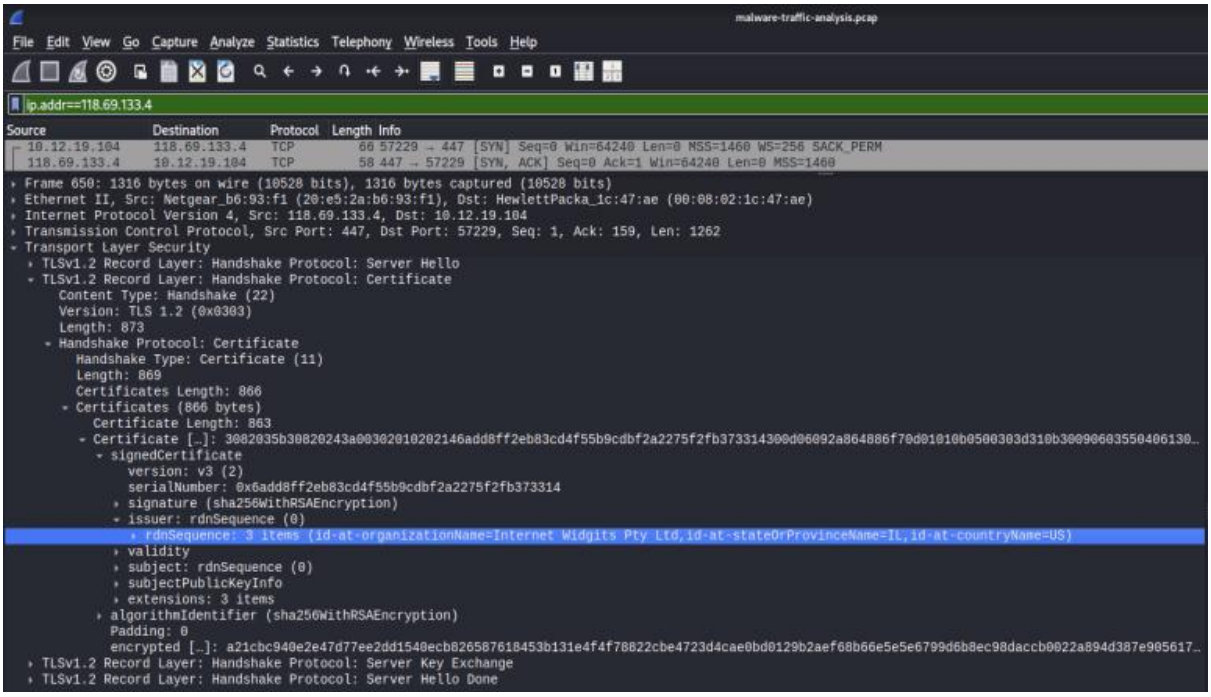

Figure 10 Suspected C2 server (3) certificate

According to cisa.gov (2021), Trickbot detection signatures include certificate field having Default City name or Default Company Ltd. We can confirm that the IP addresses with organization name Internet Widgits Pty Ltd (196.45.140.146 (1) **and** 118.69.133.4 (3)) are C2 servers having confirmed that Internet Widgits Pty Ltd is the default name given to a certificate when created (Mokbel, 2021).

For the other suspected C2 server with IP address 23.160.192.125 (2) and certificate issuer name Qjvoobim and locality name Davhlvuwmxy, I could not find any organization with that name or locality name which means it might have been randomly generated and given to the certificate. A search of the IP address on Virustotal gave a clean result as it had not been flagged by any vendors but there is a community note on the IP linking it to a trickbot.

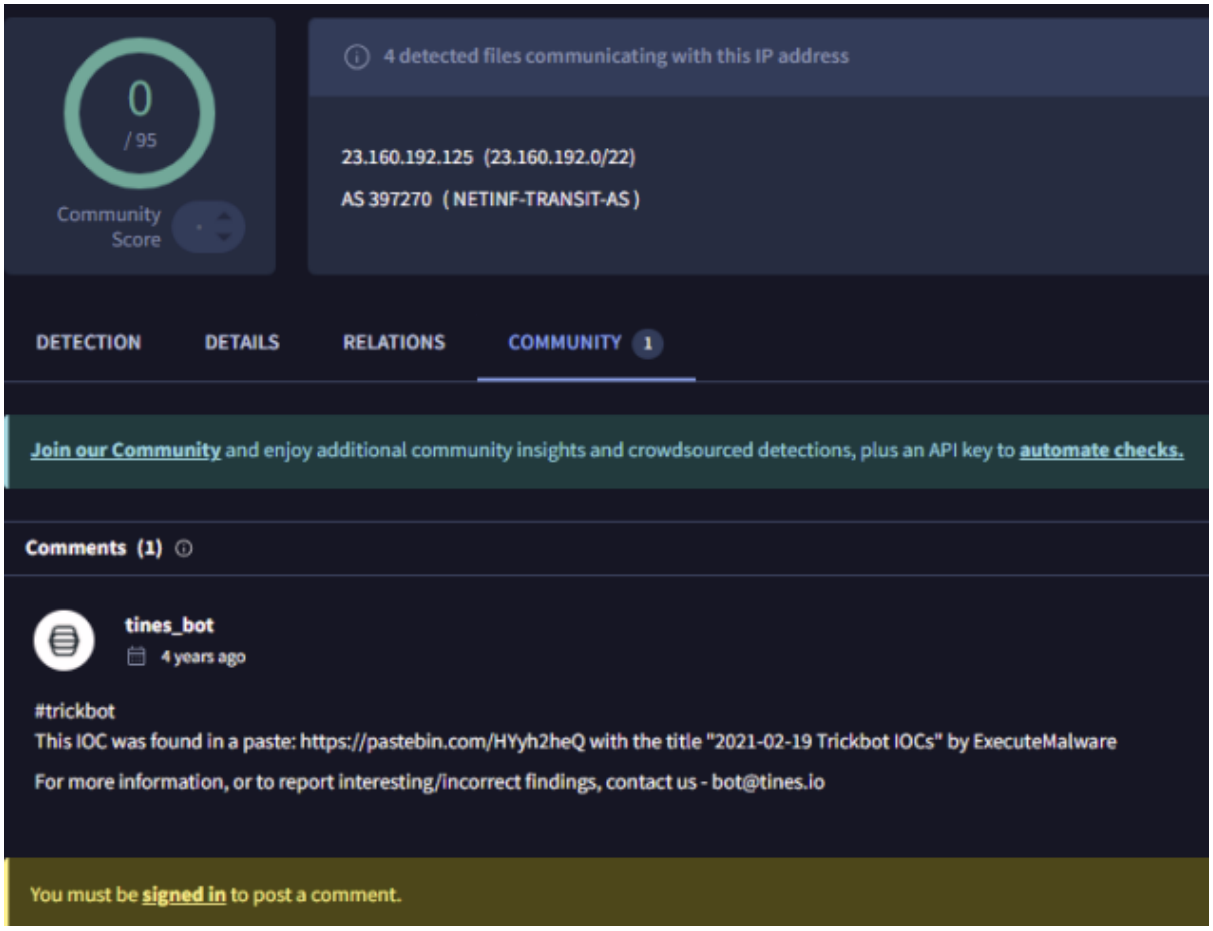

Figure 11 Virustotal report on suspected C2 server (2)

With this information and the fact that the server's certificate was issued by an unknown entity, I would classify 23.160.192.125 as a C2 server.

Also looking at the IP address that delivered the diego.dll payload.
cahrhomeopathy.com - 43.240.64.184
Port: 80
Communication Protocols used: TCP, HTTP
Host communicated with: Only the affected host (10.12.19.104)
Total packets: 463

I opened the pcap file on Network Miner and discovered the following:

1. The network has 2 windows hosts 10.12.19.19 and our affected host 10.12.19.104.
2. Name of affected host is DESKTOP-3kI6Y6G and username is smalls.hammish.

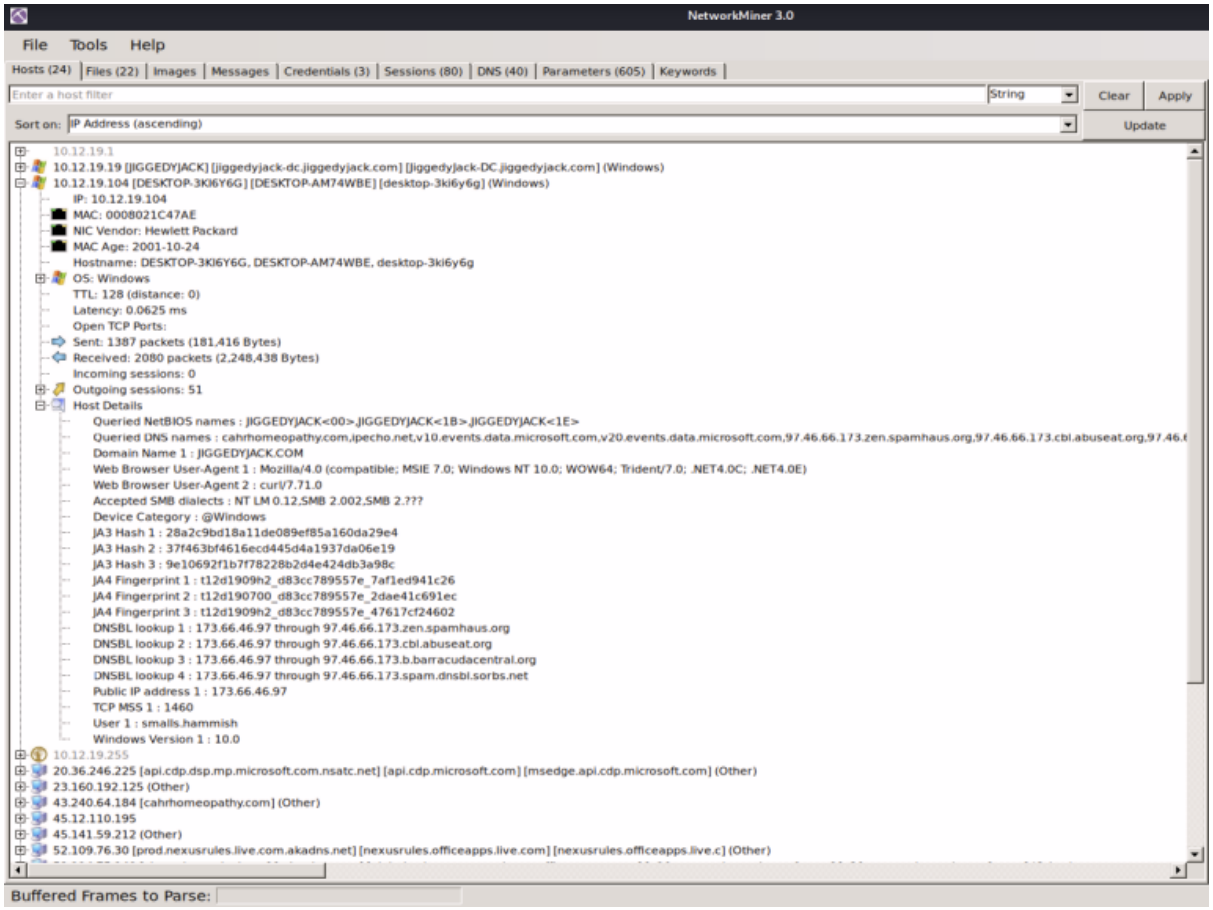

Figure 12 Network Miner result of the pcap file

Just to confirm the other host (10.12.19.19) was not affected as well, I filtered the host with http, nbns and smb and did not find anything suspicious, it is also known that this host did not have communication with any of the identified C2 servers.

SUMMARY

MALWARE FAMILY

1. **Trickbot (Trojan)**, identified by scanning the sha256 hash of the dll file (diego.png) recovered from the network on Malware Balzaar and VirusTotal.
2. **IP ADDRESS(ES) OF THE COMMAND-AND-CONTROL (C2) SERVER**
   The following IP addresses were identified as the active Command-and-Control infrastructure by analysing the certificates of servers that had a tls handshake in the network:

- 196.45.140.146 (Certificate Issuer: Internet Widgits Pty Ltd)
- 118.69.133.4 (Certificate Issuer: Internet Widgits Pty Ltd)
- 23.160.192.125 (Certificate Issuer: Qjvoobim)
  (Note: IP 43.240.64.184 was identified as the Distribution/Delivery Server rather than a C2 server, as it hosted the initial malware payload)

3. **THE MALICIOUS FILE AND ITS HASH**

- **File Name:** diego.png
- **Actual File Type:** PE32 executable (DLL) masked as a PNG image
- **SHA256 Hash:** da1ae69acf1b97bfac587addc9266155342bf8f2a7a80e0d09df9a577c39f7f9

4. **PORTS USED**

The malware utilized the following ports for its operations:

- **Port 80 (TCP):** Used for the initial HTTP GET request to download the malware payload from IP 43.240.64.184 and other TCP communication.
- **Port 449 (TCP):** Used for C2 communication with 196.45.140.146.
- **Port 447 (TCP):** Used for C2 communication with 118.69.133.4 and 23.160.192.125.

5. **COMMUNICATIONS BETWEEN THE SERVERS AND THE HOSTS**

The infected host (10.12.19.104) established two types of communication: **Payload Delivery** and **Command & Control**.

A. Payload Delivery Communication

Server: 43.240.64.184 (cahrhomeopathy.com)
Protocol: HTTP and TCP over Port 80.
Activity: The host sent a GET request for /diego.png . The server responded by delivering the malicious DLL file, other TCP requests were also noticed between the host and this IP.
Traffic Volume: 463 packets exchanged between host and payload server.

B. Command & Control (C2) Communication

After infection, the host communicated with three C2 servers:

1. **Server** 196.45.140.146
   Port 449
   Communication Protocols used: TCP, TLSV1
   Traffic Volume: 297 packets exchanged
2. **Server** 23.160.192.125
   Port: 447
   Communication Protocols used: TCP, TLSV1.2
   Traffic Volume: 19 packets exchanged
3. **Server** 118.69.133.4
   Port: 447
   Communication Protocols used: TCP, TLSV1.2
   Traffic Volume: 1,477 packets exchanged.

6. **INFECTED WINDOWS HOST NAME:**

DESKTOP-3kI6Y6G - Identified by NetworkMiner.

7. **USER ACCOUNT NAME:**
   smalls.hammish - Identified by NetworkMiner.

TABLE OF FIGURES

REFERENCES

Bejtlich, R. (2013) The Practice of Network Security Monitoring: Understanding Incident Detection and Response. San Francisco: No Starch Press.

Bejtlich, R. (2021) Applied Network Security Monitoring. 2nd edn. Cambridge: O'Reilly Media.

Beverly, R. and Bauer, S. (2011) 'The Spoofer Project: Inferring the Extent of Source Address Filtering on the Internet', USENIX SRUTI, 11, pp. 53–59.

Casey, E. (2011) Digital Evidence and Computer Crime. 3rd edn. Amsterdam: Elsevier Academic Press.

Cichonski, P., Millar, T., Grance, T. and Scarfone, K. (2012) Computer Security Incident Handling Guide (SP 800-61 Rev.2). Gaithersburg: National Institute of Standards and Technology.

CISA (2025) TrickBot Malware. Available at: https://www.cisa.gov.

Conti, G. (2007) Security Data Visualization. Boston: Addison-Wesley.

Foreman, P. (2016) The Basics of Digital Forensics. 2nd edn. Waltham: Syngress.

Garcia, S., Grill, M., Stiborek, J. and Zunino, A. (2014) ‘An Empirical Comparison of Botnet Detection Methods’, Computers & Security, 45, pp. 100–123.

Kent, K., Chevalier, S., Grance, T. and Dang, H. (2006) Guide to Integrating Forensic Techniques into Incident Response (SP 800-86). Gaithersburg: National Institute of Standards and Technology.

Ligh, M., Adair, S., Hartstein, B. and Richard, M. (2014) Malware Analyst’s Cookbook. 2nd edn. Indianapolis: Wiley.

Mandiant (2025) TrickBot: A Technical Analysis. Available at: https://www.mandiant.com.

Mokbel, M. (2021) ‘Understanding Self-Signed Certificates in Malware Command-and-Control Infrastructure’, SANS Digital Forensics Whitepaper. Available at: https://www.sans.org.

Open Information Security Foundation (2025) Suricata User Guide. Available at: https://suricata.io.

Paxson, V. (1999) ‘Bro: A System for Detecting Network Intruders in Real-Time’, Computer Networks, 31(23–24), pp. 2435–2463.

Rossow, C. (2014) ‘Prudent Practices for Designing Malware Experiments’, IEEE Security & Privacy, 12(2), pp. 56–63.

Scarfone, K. and Mell, P. (2012) Guide to Intrusion Detection and Prevention Systems (SP 800-94). Gaithersburg: National Institute of Standards and Technology.

Shanmugam, S. and Behl, A. (2021) Cybersecurity and Cyberwar: What Everyone Needs to Know. Oxford: Oxford University Press.

Sikorski, M. and Honig, A. (2012) Practical Malware Analysis. San Francisco: No Starch Press.

VirusTotal (2025) Online Malware and URL Scanning Service. Available at: https://www.virustotal.com.

Wireshark Foundation (2025) Wireshark User’s Guide. Available at: https://www.wireshark.org/docs.

Zanero, S. (2013) ‘Behavioral Malware Detection’, Journal in Computer Virology, 9(1), pp. 1–4.

BAD PDF CASE (MEMORY FORESNICS)

Disclaimer
Network-based evidence can strongly indicate that a system has been compromised; however, it cannot conclusively demonstrate what executed on the host or what actions were performed internally.

To extend the investigation beyond external indicators and demonstrate how such findings are validated at the host level, a second, independent case study was analysed. This case represents a comparable post-compromise scenario in which a memory image was captured following the detection of suspicious activity, allowing for deeper examination of execution, persistence, and attacker intent.

Disclaimer

INTRODUCTION

This forensic investigation was initiated at the request of Best Finance (BF), a financial institution that recently detected suspicious activity. The primary investigator, Ali, a renowned banking system forensics expert, was contacted by BF following an internal security incident.

Incident Summary:
A Best Finance employee reported receiving an email from a coworker containing a PDF attachment. Upon opening the file, the employee did not observe any immediate anomalies; however, subsequent unusual activity was detected in the employee's bank account. To facilitate the investigation, BF's security team captured a memory image of the employee's virtual machine immediately following the suspected infection.

Objective:
Due to current caseload demands, Ali has assigned this analysis to the junior forensic analyst. The objective is to analyze the provided virtual memory image (BF.vmem) to determine the root cause of the compromise, identify malicious processes, map network connections, and uncover the scope of the potential banking fraud.

LAB PREPARATION

The analysis focused on memory forensics such as running processes and active network connections.

Forensic Environment

- **Operating System:** The analysis was conducted on a **Kali Linux** workstation (same as before).
- **Evidence File:** The target of the analysis was a raw memory dump file named **BF.vmem**.

Tools and Utilities

The following tools were utilized to extract and analyse data from the memory image:

- Volatility2
  Language: The framework was executed using Python 2 (python2 vol.py)
  Profile Identification: The imageinfo plugin was used to identify the operating system profile. The system was identified as **WinXPSP2x86**
- **Strings & Grep**: Native Linux command-line utilities used to search for specific text patterns (such as URLs and protocol identifiers like "http/https") within dumped memory files.
- **VirusTotal** (https://www.virustotal.com/): An external threat intelligence service used to cross-reference and confirm the malicious nature of extracted executables and hash values.
- **IpInfo**(https://ipinfo.io/): To get information about IP addresses

Volatility Plugins Deployed

The following specific Volatility plugins were employed to answer the investigation questions:

- **pslist:** To list running processes and identify hierarchy anomalies.
- **malfind**: To detect hidden or injected code and dump malicious memory segments.
- **connscan**: To scan for active network connections and open sockets
- **memdump**: To extract the full memory address space of suspicious processes for string analysis.
- **hashdump**: To extract password hashes from the registry stored in memory.

INVESTIGATION

Using volatility2 on my kali, I used imageinfo to get the suggested profile to work on, volatility2 picked and instantiated with profile WinXPSP2x86 so there would be no need to call the profile I want to work on when using volatility2

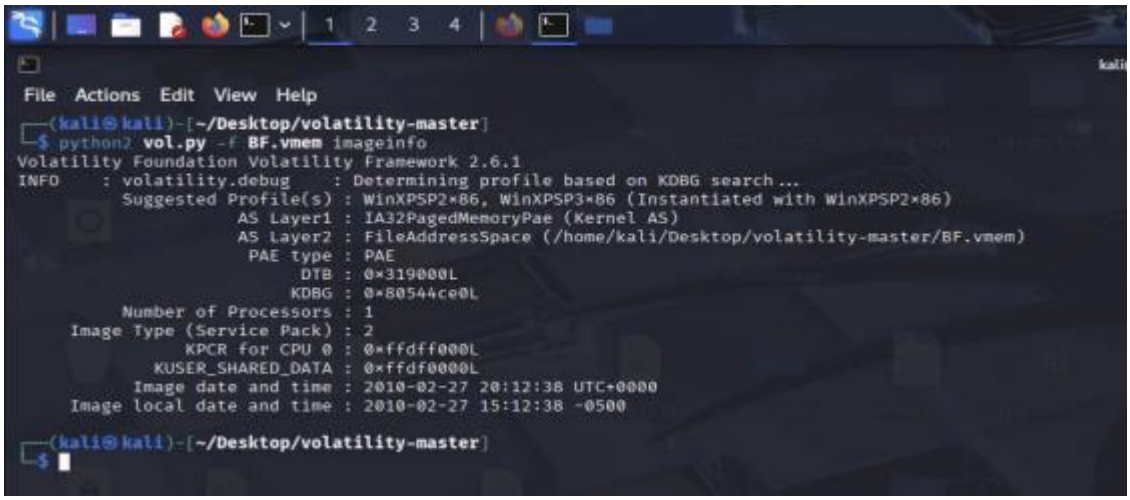

Figure 13 Imageinfo Result

Pslist was used to get running processes on the machine for examination to get suspicious processes.

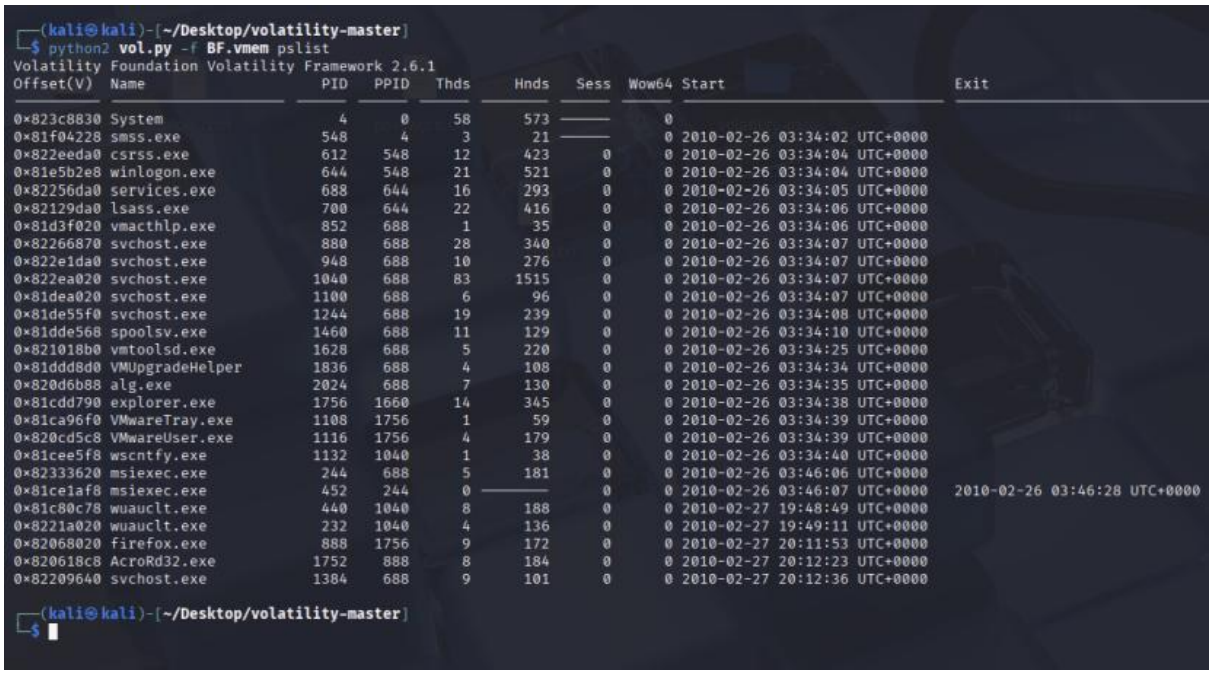

Figure 14 Pslist list result

From the list, the svchost.exe with PID 1384 is out of sync with other system processes and it started at 20:12:36 UTC, just 3 seconds after process AcroRd32.exe with PID 1752

started which suggests process AcroRd32.exe might have caused the system process to start which is unusual. To confirm if there was malware in any of these processes, I used **malfind** on both processes and on the parent of the AcroRd32.exe, firefox.exe with PID 888.

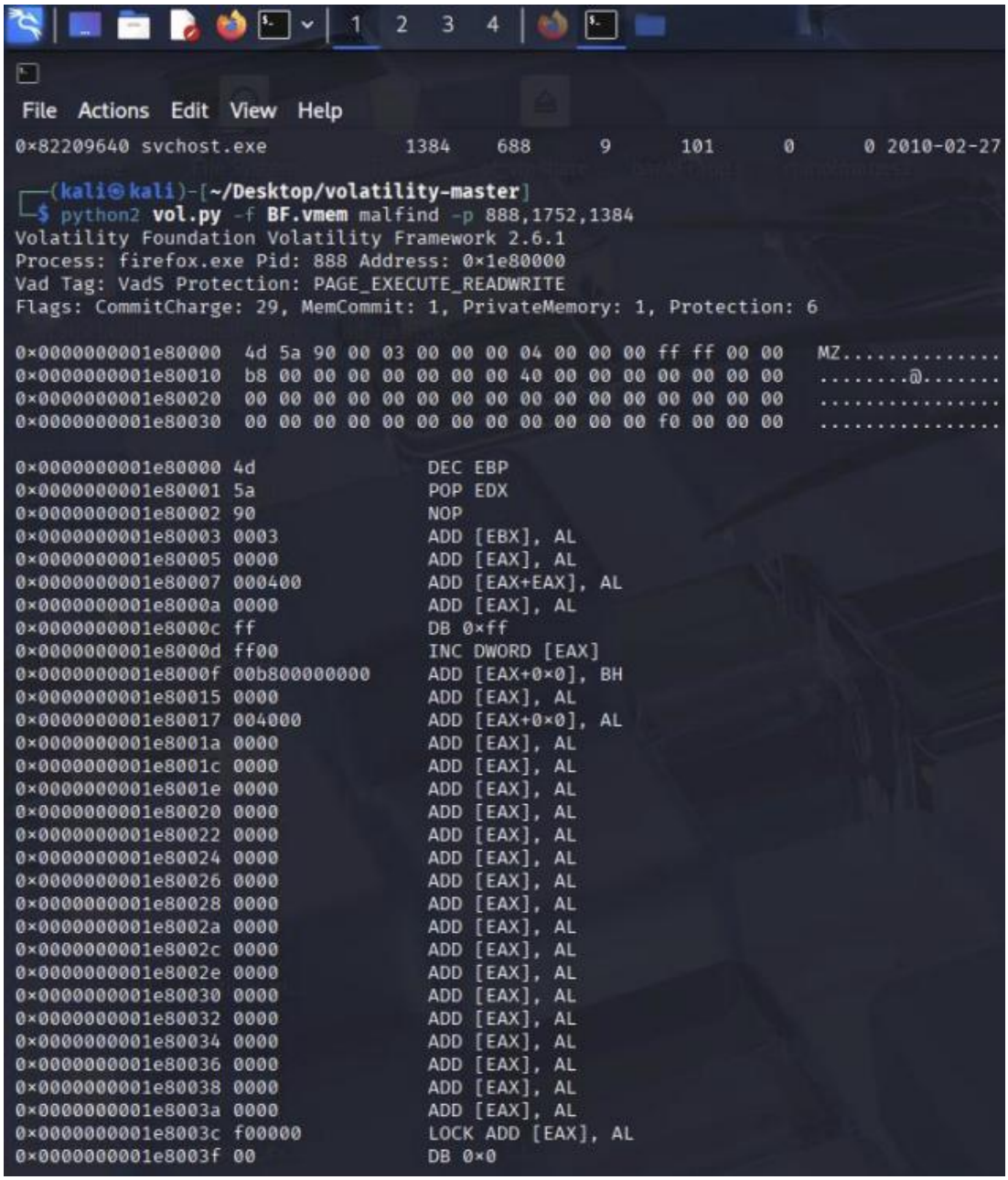

Figure 15 Malfind on firefox.exe (PID 888)

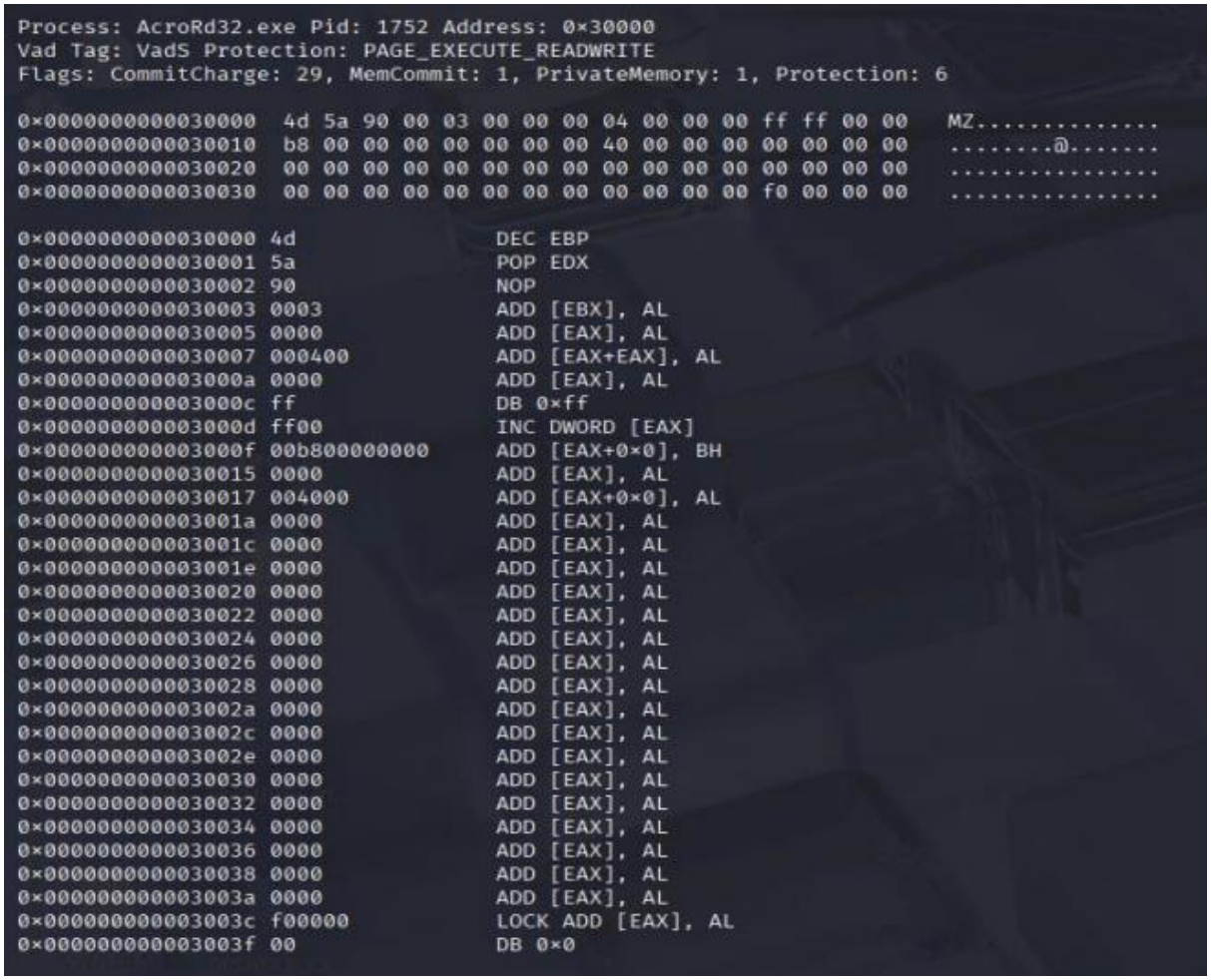

Figure 16 Malfind on AcroRd32.exe (PID 1752)

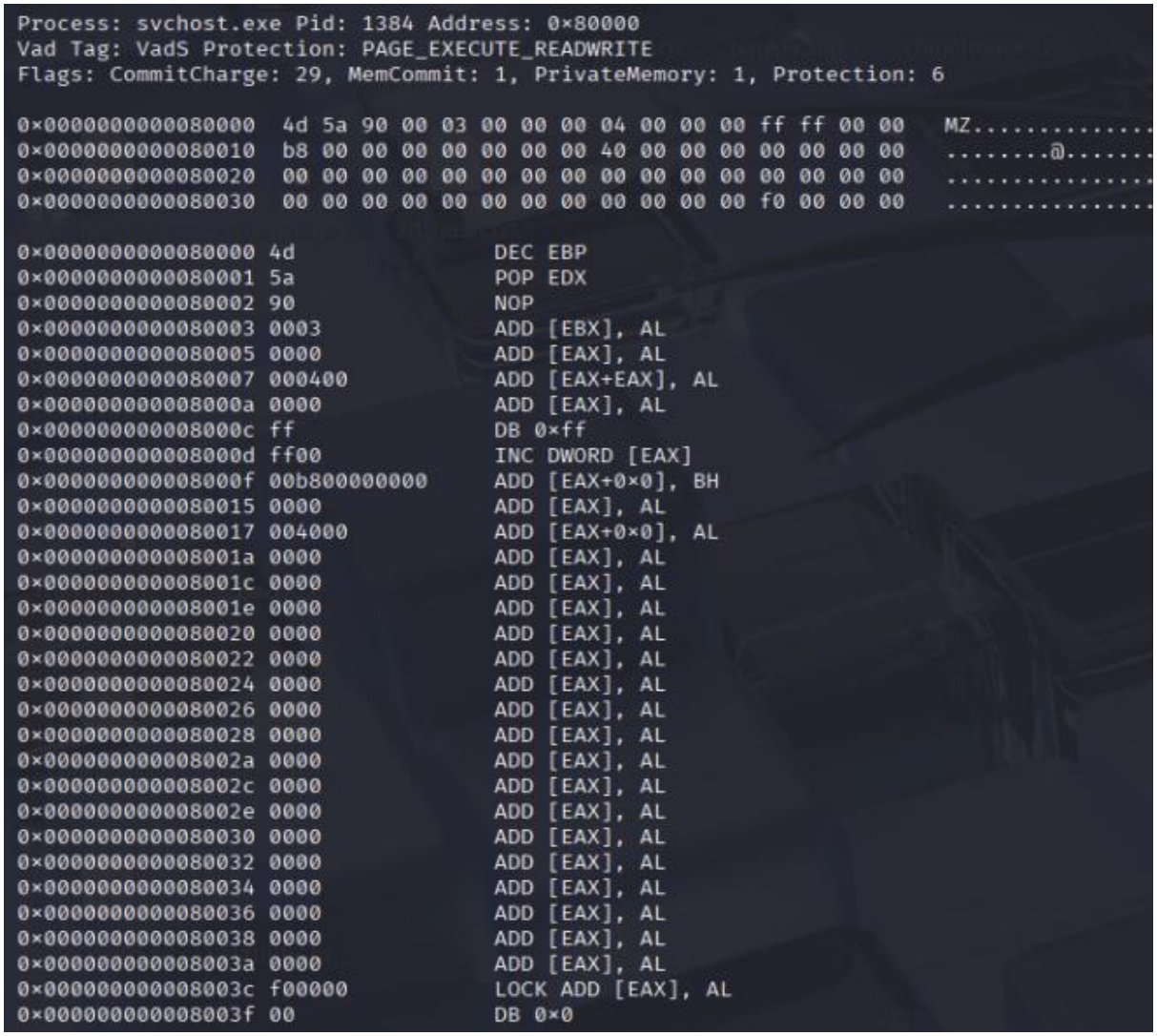

Figure 17 Malfind on svchost.exe (PID 1384)

All three processes had Read/Write privileges and they all contain an MZ file which is a DOS executable file.

Using **python2 vol.py -f BF.vmem malfind -p 888,1752,1384 --dump-dir**.

I dumped the MZ files on my kali's memory and saved them as mal888.dmp, mal1752.dmp and 1384.dmp. Using file function on kali, I was able to confirm they are PE32 executable for MS Windows. I also scanned them on VirusTotal and it was confirmed that the 3 MZ files on the processes were malicious.

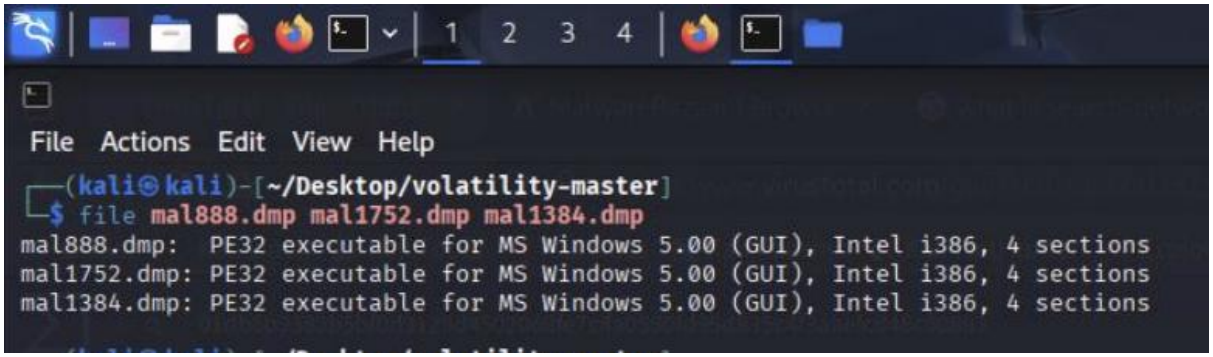

Figure 18 File Result on dumped MZ files

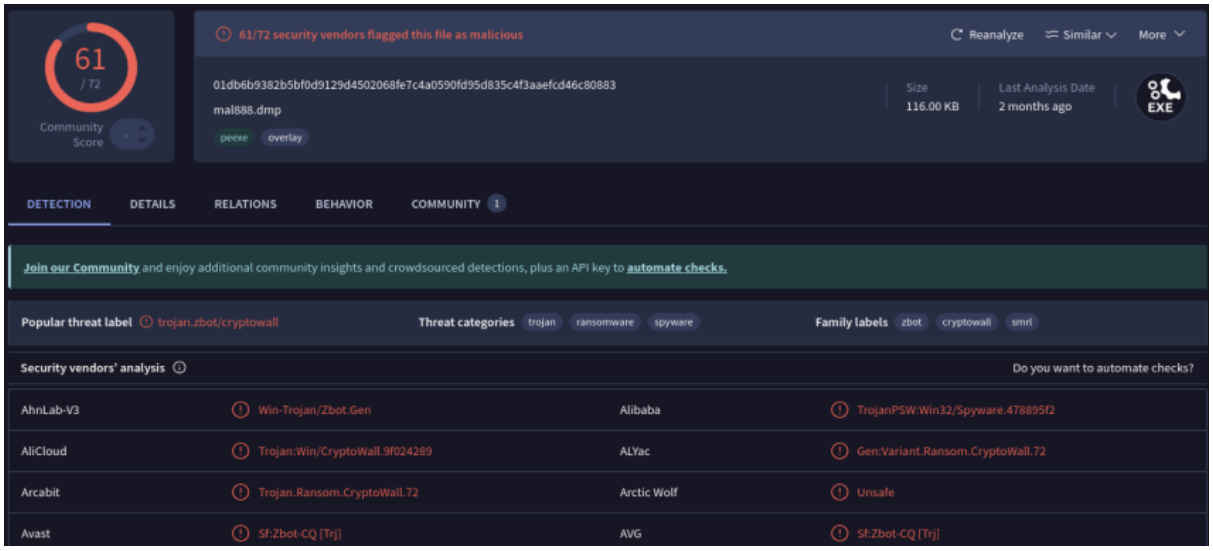

Figure 19 MZ file on PID 888 VirusTotal Result

https://www.virustotal.com/gui/file/01db6b9382b5bf0d9129d4502068fe7c4a0590fd95d835c4f3aaefcd46c80883

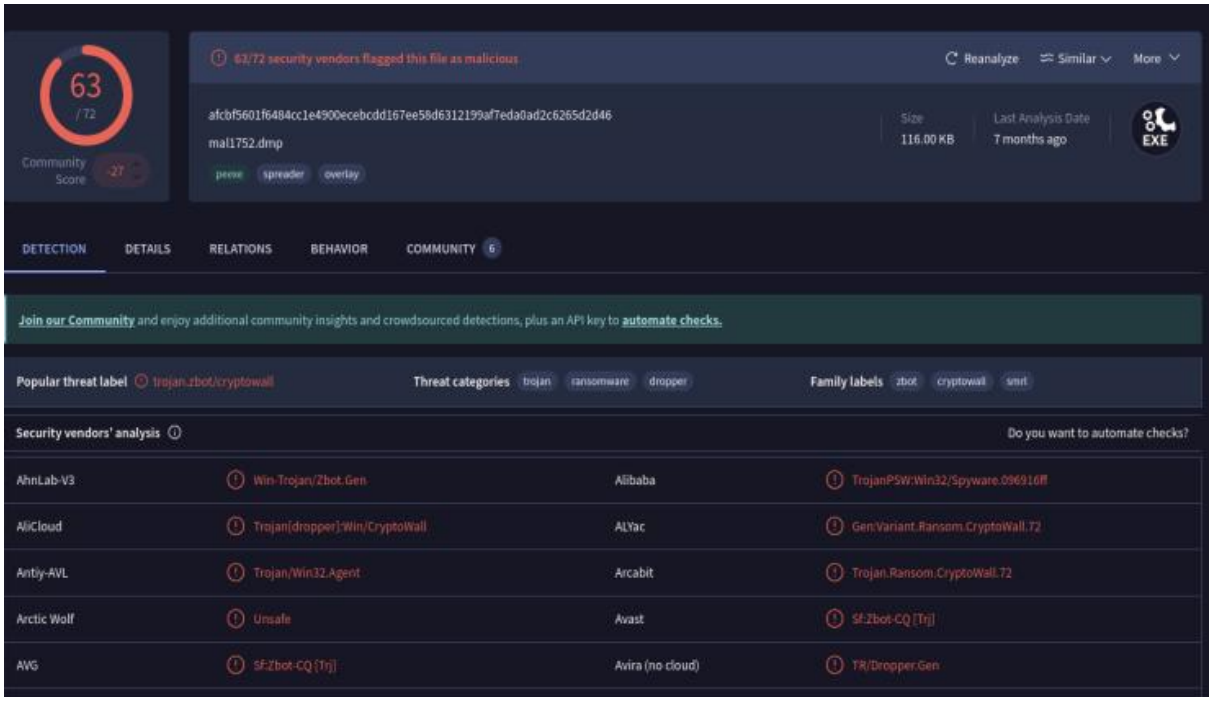

Figure 20 MZ file on PID 1752 VirusTotal Result

https://www.virustotal.com/gui/file/afcbf5601f6484cc1e4900ecebcdd167ee58d6312199af7eda0ad2c6265d2d46

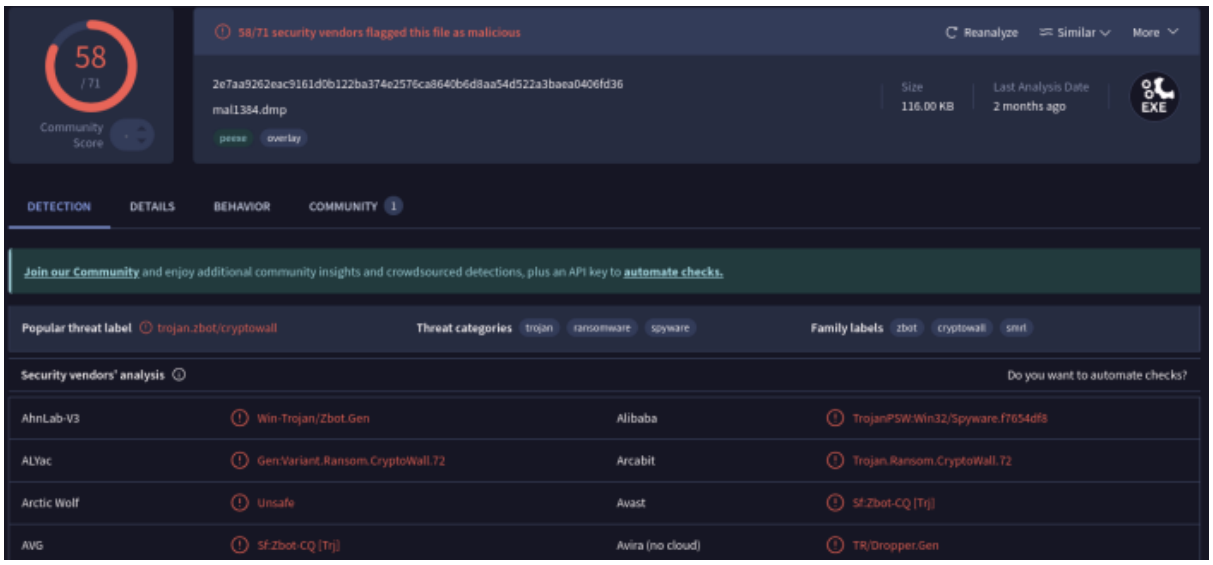

Figure 21 MZ file on PID 1384 VirusTotal Result

https://www.virustotal.com/gui/file/2e7aa9262eac9161d0b122ba374e2576ca8640b6d8aa54d522a3baea0406fd36

With suspicious processes identified as AcroRd32.exe (PID 1752), firefox.exe (PID 888) and svchost.exe (PID 1384), the next step was to check if any of these processes had a connection to the internet. Using connscan, I was able to achieve this.

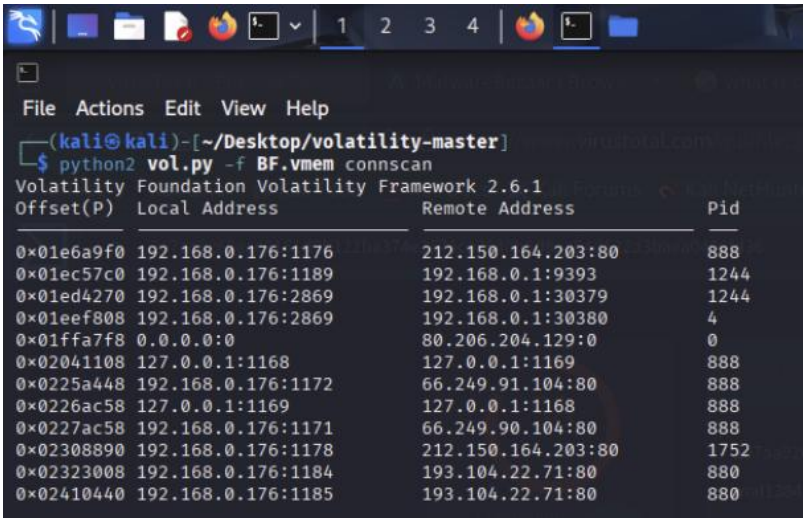

Figure 22 Connscan results

NO	Local Address	Remote Address	PID	Action
1	192.168.0.176:1176	212.150.164.203:80	888	Investigate
2	192.168.0.176:1189	192.168.0.1:9393	1244	Ignore
3	192.168.0.176:2869	192.168.0.1:30379	1244	Ignore
4	192.168.0.176:2869	192.168.0.1:30380	4	Ignore
5	0.0.0.0:0	80.206.204.129:0	0	Ignore
6	127.0.0.1:1168	127.0.0.1:1169	888	Ignore
7	192.168.0.176:1172	66.249.91.104:80	888	Ignore
8	127.0.0.1:1169	127.0.0.1:1168	888	Ignore
9	192.168.0.176:1171	66.249.90.104:80	888	Ignore
10	192.168.0.176:1178	212.150.164.203:80	1752	Investigate
11	192.168.0.176:1184	193.104.22.71:80	880	Investigate
12	192.168.0.176:1185	193.104.22.71:80	880	Investigate

Table 1 List of active connections and actions

Using **Figure 22 Connscan results**, I created a table of network traffic and suggested actions to follow. Traffic on NO 2,3,4,6 and 8 are ignored because they seem to be traffic between host machines/profiles. NO 5 suggests a terminated connection, NO 7 and 9 are safe traffic to Google related IP addresses (https://ipinfo.io/).

No 1,10,11,12 are suggested for investigation because they are connected to not know Remote Addresses, out of the 4 processes to investigate, 2 of them have a connection

to suspected malware carrying processes PID 888 (NO 1) and PID 1752 (NO 10), while the other two NO 11 and 12 were connected to an unsuspected svchost.exe process with PID 880.

VirusTotal did not provide any evidence of malware to the unknown IP addresses in the suspected traffic, so I dumped the three processes with PIDs 888,1752 and 880 using memdump;

python2 vol.py -f BF.vmem memdump -p 888 --dump-dir=

python2 vol.py -f BF.vmem memdump -p 1752 --dump-dir=

python2 vol.py -f BF.vmem memdump -p 880 --dump-dir=

Files were saved as 888.dmp, 1752.dmp and 880.dmp.

The next step was to use Strings function to search the dumped files for the websites/servers they visited using "http/https" as the filter terms, using the commands;

strings -e l 888.dmp | grep -Ei "http|https"

This was to search PID 888 for http/https matches, looked through the results and did not find any suspicious URLs or IP addresses

strings -e l 1752.dmp | grep -Ei "http|https"

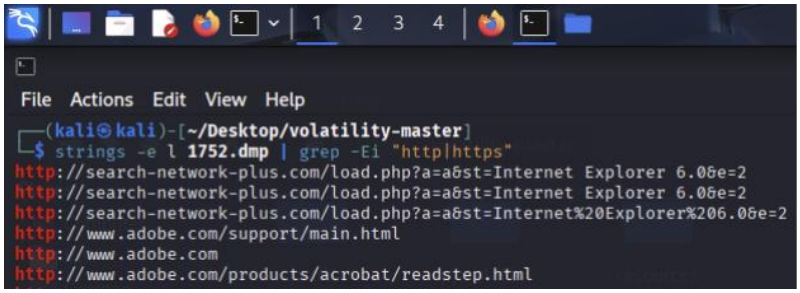

Figure 23 Snippet of String Result on PID 1752 showing suspicious URL

The search on PID 1752 using the "http|https" revealed a suspicious URL that was visited thrice by process 1752.

http://search-network-plus.com/load.php?a=a&st=Internet Explorer 6.0&e=2

http://search-network-plus.com/load.php?a=a&st=Internet Explorer 6.0&e=2

http://search-network-plus.com/load.php?a=a&st=Internet%20Explorer%206.0&e=2

strings -e l 880.dmp | grep -Ei "http|https"

Figure 24 Snippet of String Result on PID 1752 showing suspicious URL

The Strings search on PID 880, the unsuspicious process using the “http|https” filter also revealed a suspicious URL and IP address.
http://193.104.22.71/~produkt/983745213424/34650798253
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php

Using hashdump on the BF.vmem file, I was able to extract the hashes in memory

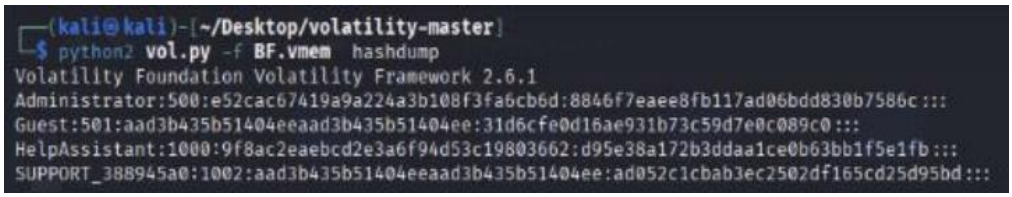

Figure 25 Hashdump results

The hashes recovered are:

Administrator:
500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c

Guest:
501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0

HelpAssistant:
1000:9f8ac2eaebcd2e3a6f94d53c19803662:d95e38a172b3ddaa1ce0b63bb1f5e1fb

SUPPORT_388945a0:
1002:aad3b435b51404eeaad3b435b51404ee:ad052c1cbab3ec2502df165cd25d95bd

SUMMARY

1. **WHAT PROCESSES WERE RUNNING ON THE VICTIM'S MACHINE. WHICH PROCESS WAS MOST LIKELY RESPONSIBLE FOR THE INITIAL EXPLOIT?**

   According to the pslist results, the following processes were running on the machine:

- System (PID 4)
- smss.exe (PID 548)
- csrss.exe (PID 612)
- winlogon.exe (PID 644)
- services.exe (PID 688)
- lsass.exe (PID 700)
- vmacthlp.exe (PID 852)
- svchost.exe (PIDs 880, 948, 1040, 1100, 1244, 1384)
- spoolsv.exe (PID 1468)
- vmtoolsd.exe (PID 1628)
- VMUpgradeHelper (PID 1836)
- alg.exe (PID 2024)
- explorer.exe (PID 1756)
- VMwareTray.exe (PID 1108)
- VMwareUser.exe (PID 1116)
- wscntfy.exe (PID 1132)
- msiexec.exe (PIDs 244, 452)
- wuauclt.exe (PIDs 440, 232)
- firefox.exe (PID 888)
- AcroRd32.exe (PID 1752)

   Process Responsible for Initial Exploit
   The process most likely responsible for the initial exploit is AcroRd32.exe (PID 1752).

- **Reasoning:** The analysis noted that svchost.exe (PID 1384) started only 3 seconds after AcroRd32.exe (PID 1752). The parent of AcroRd32.exe was firefox.exe (PID 888), indicating the user likely downloaded and opened the malicious PDF via Firefox. malfind analysis confirmed both the Adobe Reader (PID 1752), svchost (PID 1348) and the firefox (PID8 88) processes contained malicious files. All three processes PIDs 888, 1752 and 1348 are now considered suspicious.

2. **WHICH SOCKETS WERE OPEN ON THE VICTIM'S MACHINE DURING INFECTION.**
   ARE THERE ANY SUSPICIOUS PROCESSES THAT HAVE SOCKETS OPEN?

Open sockets:

The connscan plugin revealed the following active connections:

NO	Local Address	Remote Address	PID
1	192.168.0.176:1176	212.150.164.203:80	888
2	192.168.0.176:1189	192.168.0.1:9393	1244
3	192.168.0.176:2869	192.168.0.1:30379	1244
4	192.168.0.176:2869	192.168.0.1:30380	4
5	0.0.0.0:0	80.206.204.129:0	0
6	127.0.0.1:1168	127.0.0.1:1169	888
7	192.168.0.176:1172	66.249.91.104:80	888
8	127.0.0.1:1169	127.0.0.1:1168	888
9	192.168.0.176:1171	66.249.90.104:80	888
10	192.168.0.176:1178	212.150.164.203:80	1752
11	192.168.0.176:1184	193.104.22.71:80	880
12	192.168.0.176:1185	193.104.22.71:80	880

Table 2 List of open sockets

Suspicious processes with open Sockets:

Two suspicious processes out of the three we have already established have a suspicious connection to an unknown IP.

1. **Firefox.exe (PID 888):** Connected to 212.150.164.203.
2. **AcroRd32.exe (PID 1752):** Connected to 212.150.164.203.

3. **LIST ANY SUSPICIOUS URLS AND IP ADDRESSES THAT MAY BE ASSOCIATED WITH THE PROCESSES.**

Suspicious IP Addresses:

212.150.164.203: Associated with firefox.exe (PID 888) and AcroRd32.exe (PID 1752).

Suspicious URLs:
http://search-networkplus.com/load.php?a=a&st=Internet%20Explorer%206.0&e=2
Associated with AcroRd32.exe (PID 1752)

4. ARE THERE ANY OTHER PROCESSES THAT CONTAIN URLS THAT MAY POINT TO BANKING TROUBLES? IF SO, WHAT ARE THESE PROCESSES AND WHAT ARE THE URLS?

Yes, **svchost.exe (PID 880**) contained URLs that indicate malware communication. Although initially thought to be an unsuspecting process, the network scan revealed connections to a suspicious IP, and memory string dumps revealed deep links to a suspicious php file and numeric directories.

The URLs found in PID 880 are:

http://193.104.22.71/~produkt/983745213424/34650798253.

http://193.104.22.71/~produkt/9j856f_4m9y8urb.php.

5. **WHAT IS THE PURPOSE AND INTENT OF THE SUSPECTED FILES OR PROCESSES?**

The purpose of the suspected processes appears to be data theft and financial fraud, utilizing a banking trojan/spyware. The purpose of the suspected processes appears to be data theft and financial fraud, utilizing a banking trojan/spyware.

- **Malware Identification**: VirusTotal analysis labelled the dumped memory files from malfind (PIDs 888, 1752, 1384) as **"Trojan/Zbot"**, **"Spyware",** and **"Ransom.CryptoWall".**
- **Zbot (Zeus):** The presence of "Zbot" tags in the VirusTotal results specifically points to the Zeus Trojan, which is a notorious piece of malware designed to steal banking credentials and financial information (Man-in-the-Browser attacks).
- **Intent:** The malicious PDF (AcroRd32.exe) acted as a dropper/exploit to compromise the system, injecting code into svchost.exe processes to maintain persistence and communicate with Command-and-Control servers (the URLs identified in question 3 & 4) to exfiltrate data or download further ransomware/spyware components.

6. **FIND POSSIBLE HASHES FOR THE ADMINISTRATOR PASSWORD.**

Using the hashdump command on the memory image, the following hash was recovered for the Administrator account:

Administrator:
500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c

TABLE OF FIGURES (For the entire doc)

TABLE OF TABLES