Network and Memory Forensics Casework

CHRISTIAN IGBINOSA

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 analyze 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.

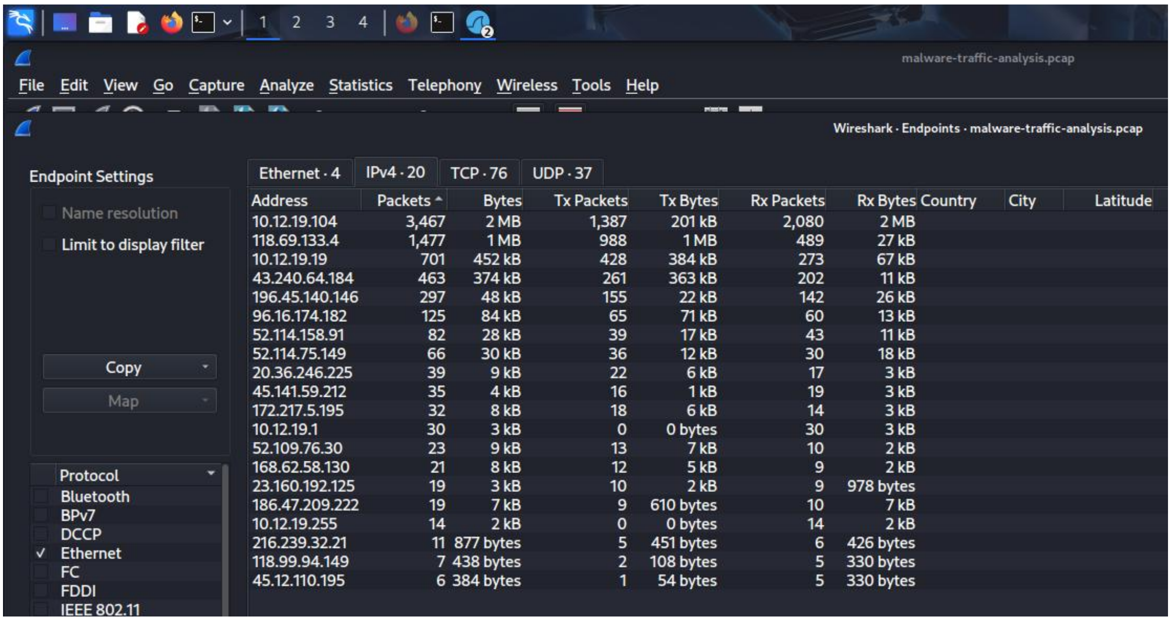
- **Online Threat Intelligence:**

  VirusTotal: https://www.virustotal.com

  Malware Bazaarhttps://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.



malware-traffic-analysis.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Endpoints · malware-traffic-analysis.pcap
Endpoint Settings
Copy
Map
Ethernet · 4
IPv4 · 20
TCP · 76
UDP · 37

□ Name resolution

□ Limit to display filter

Protocol
□ Bluetooth
□ BPv7
□ DCCP
✓ Ethernet
□ FC
□ FDDI
□ IEEE 802.11

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	Latitude
10.12.19.104	3,467	2 MB	1,387	201 kB	2,080	2 MB			
118.69.133.4	1,477	1 MB	988	1 MB	489	27 kB			
10.12.19.19	701	452 kB	428	384 kB	273	67 kB			
43.240.64.184	463	374 kB	261	363 kB	202	11 kB			
196.45.140.146	297	48 kB	155	22 kB	142	26 kB			
96.16.174.182	125	84 kB	65	71 kB	60	13 kB			
52.114.158.91	82	28 kB	39	17 kB	43	11 kB			
52.114.75.149	66	30 kB	36	12 kB	30	18 kB			
20.36.246.225	39	9 kB	22	6 kB	17	3 kB			
45.141.59.212	35	4 kB	16	1 kB	19	3 kB			
172.217.5.195	32	8 kB	18	6 kB	14	3 kB			
10.12.19.1	30	3 kB	0	0 bytes	30	3 kB			
52.109.76.30	23	9 kB	13	7 kB	10	2 kB			
168.62.58.130	21	8 kB	12	5 kB	9	2 kB			
23.160.192.125	19	3 kB	10	2 kB	9	978 bytes			
186.47.209.222	19	7 kB	9	610 bytes	10	7 kB			
10.12.19.255	14	2 kB	0	0 bytes	14	2 kB			
216.239.32.21	11	877 bytes	5	451 bytes	6	426 bytes			
118.99.94.149	7	438 bytes	2	108 bytes	5	330 bytes			
45.12.110.195	6	384 bytes	1	54 bytes	5	330 bytes			

Figure 2. 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:

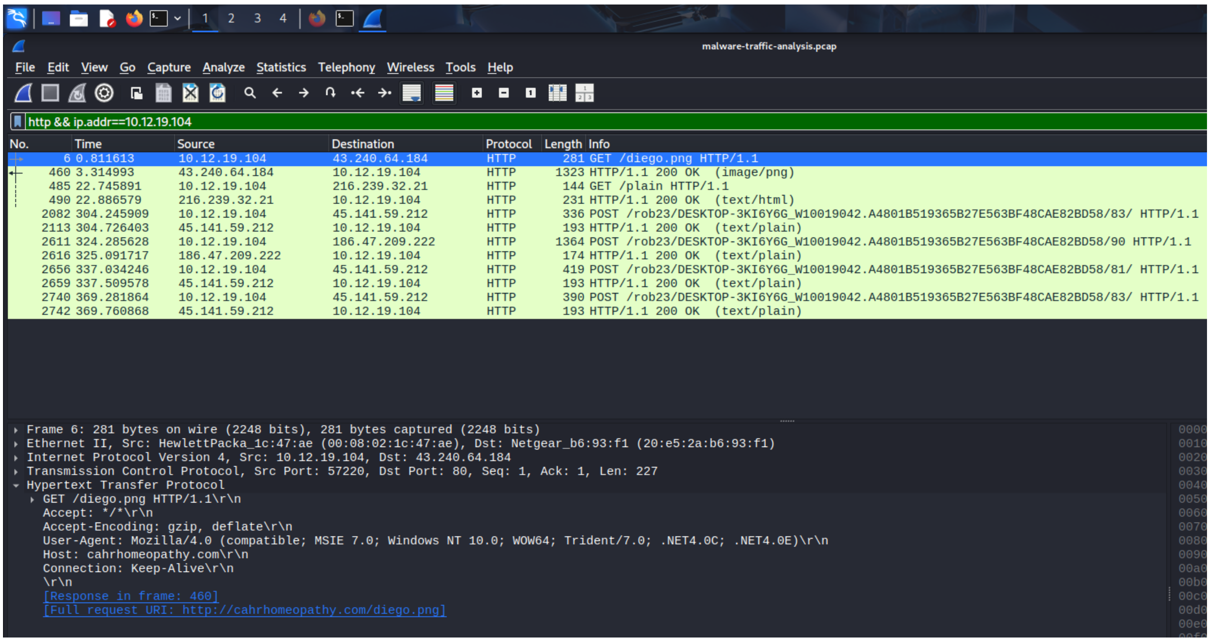
10.12.19.104

10.12.19.19

10.12.19.1

10.12.19.255

From the list, 10.12.19.104 seems to have a lot 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 10.12.19.104below.



malware-traffic-analysis.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http && ip.addr==10.12.19.104
```

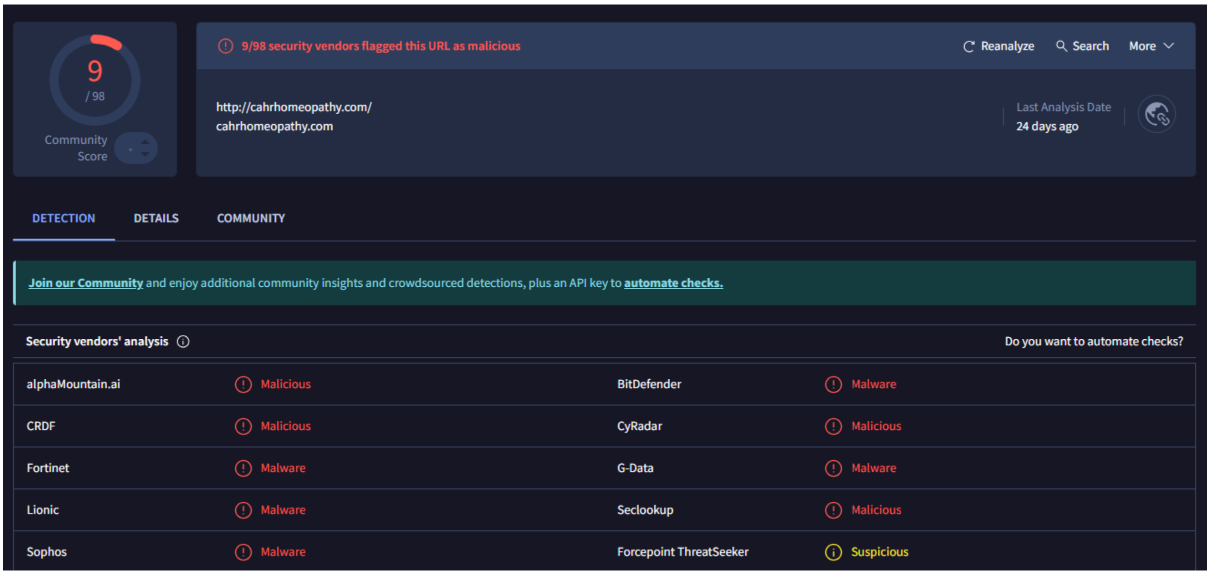
```
No.    Time         Source            Destination       Protocol  Length Info
     6 0.811613     10.12.19.104      43.240.64.184     HTTP       281 GET /diego.png HTTP/1.1
   460 3.314993     43.240.64.184     10.12.19.104      HTTP      1323 HTTP/1.1 200 OK  (image/png)
   485 22.745891    10.12.19.104      216.239.32.21     HTTP       144 GET /plain HTTP/1.1
   490 22.886579    216.239.32.21     10.12.19.104      HTTP       231 HTTP/1.1 200 OK  (text/html)
  2082 304.245909   10.12.19.104      45.141.59.212     HTTP       336 POST /rob23/DESKTOP-3KI6Y6G_W10019042.A4801B519365B27E563BF48CAE82BD58/83/ HTTP/1.1
  2113 304.726403   45.141.59.212     10.12.19.104      HTTP       193 HTTP/1.1 200 OK  (text/plain)
  2611 324.285628   10.12.19.104      186.47.209.222    HTTP      1364 POST /rob23/DESKTOP-3KI6Y6G_W10019042.A4801B519365B27E563BF48CAE82BD58/90 HTTP/1.1
  2616 325.091717   186.47.209.222    10.12.19.104      HTTP       174 HTTP/1.1 200 OK  (text/plain)
  2656 337.034246   10.12.19.104      45.141.59.212     HTTP       419 POST /rob23/DESKTOP-3KI6Y6G_W10019042.A4801B519365B27E563BF48CAE82BD58/81/ HTTP/1.1
  2659 337.509578   45.141.59.212     10.12.19.104      HTTP       193 HTTP/1.1 200 OK  (text/plain)
  2740 369.281864   10.12.19.104      45.141.59.212     HTTP       390 POST /rob23/DESKTOP-3KI6Y6G_W10019042.A4801B519365B27E563BF48CAE82BD58/83/ HTTP/1.1
  2742 369.760868   45.141.59.212     10.12.19.104      HTTP       193 HTTP/1.1 200 OK  (text/plain)
```

```
Frame 6: 281 bytes on wire (2248 bits), 281 bytes captured (2248 bits)
Ethernet II, Src: HewlettPacka_1c:47:ae (00:08:02:1c:47:ae), Dst: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1)
Internet Protocol Version 4, Src: 10.12.19.104, Dst: 43.240.64.184
Transmission Control Protocol, Src Port: 57220, Dst Port: 80, Seq: 1, Ack: 1, Len: 227
Hypertext Transfer Protocol
  GET /diego.png HTTP/1.1\r\n
  Accept: */*\r\n
  Accept-Encoding: gzip, deflate\r\n
  User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E)\r\n
  Host: cahrhomeopathy.com\r\n
  Connection: Keep-Alive\r\n
  \r\n
  [Response in frame: 460]
  [Full request URI: http://cahrhomeopathy.com/diego.png]
```

Figure 2. 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 virustotal, I scanned the website.



9/98 security vendors flagged this URL as malicious

http://cahrhomeopathy.com/
cahrhomeopathy.com

Last Analysis Date
24 days ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis

Do you want to automate checks?

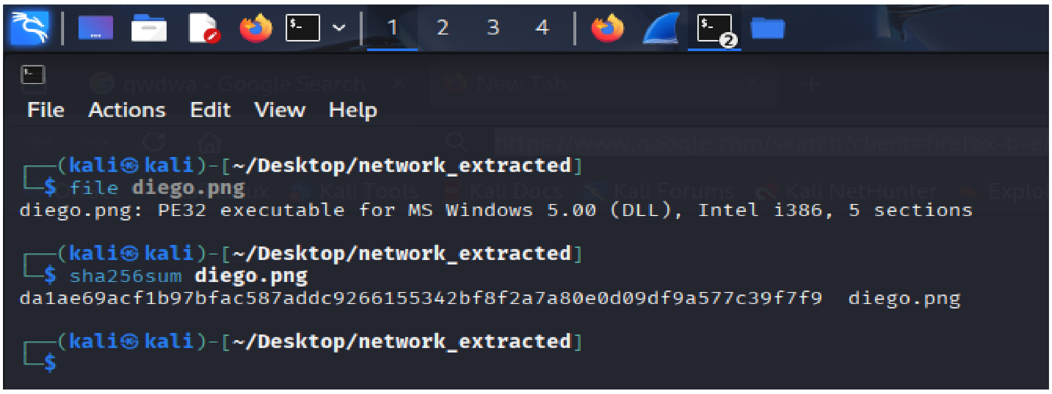
alphaMountain.ai	Malicious	BitDefender	Malware
CRDF	Malicious	CyRadar	Malicious
Fortinet	Malware	G-Data	Malware
Lionic	Malware	Seclookup	Malicious
Sophos	Malware	Forcepoint ThreatSeeker	Suspicious

Figure 2. 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.

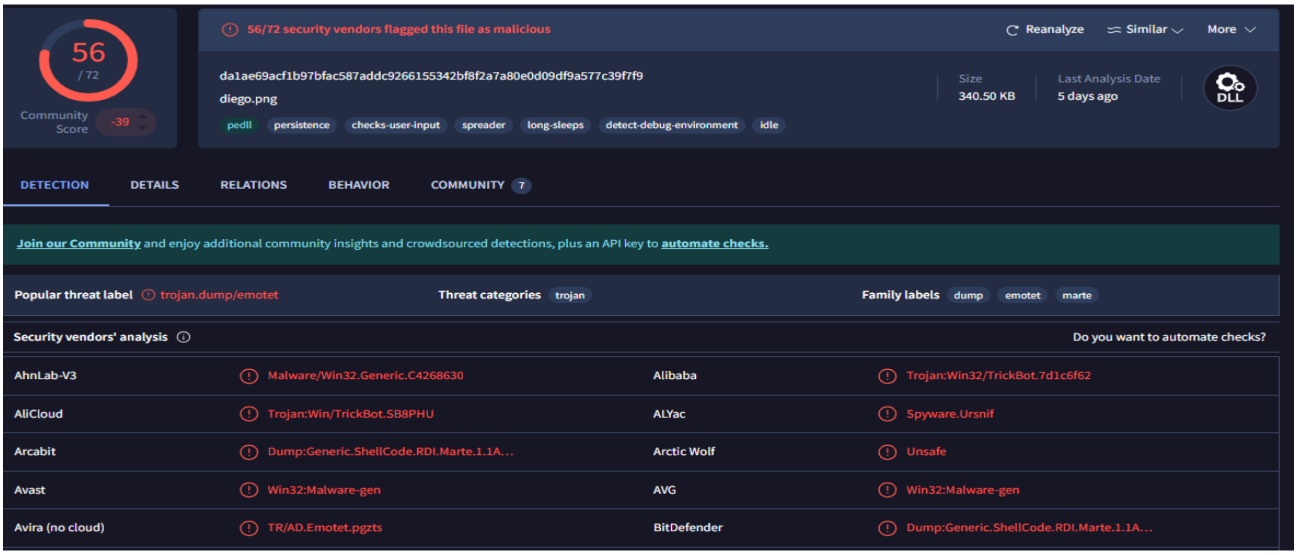
After extracting the diego.png object, I examined it using File function on Kali and discovered it was not a PNG file but a DLL file, I also extracted the hash of the file (da1ae69acf1b97bfac587addc9266155342bf8f2a7a80e0d09df9a577c39f7f9) using sha256 for further investigation.



```
┌──(kali㉿kali)-[~/Desktop/network_extracted]
└─$ file diego.png
diego.png: PE32 executable for MS Windows 5.00 (DLL), Intel i386, 5 sections

┌──(kali㉿kali)-[~/Desktop/network_extracted]
└─$ sha256sum diego.png
da1ae69acf1b97bfac587addc9266155342bf8f2a7a80e0d09df9a577c39f7f9  diego.png

┌──(kali㉿kali)-[~/Desktop/network_extracted]
└─$
```

Figure 2. 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.



56/72 security vendors flagged this file as malicious

da1ae69acf1b97bfac587addc9266155342bf8f2a7a80e0d09df9a577c39f7f9
diego.png

pedll · persistence · checks-user-input · spreader · long-sleeps · detect-debug-environment · idle

Size: 340.50 KB
Last Analysis Date: 5 days ago

DETECTION · DETAILS · RELATIONS · BEHAVIOR · COMMUNITY 7

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan.dump/emotet
Threat categories: trojan
Family labels: dump · emotet · marte

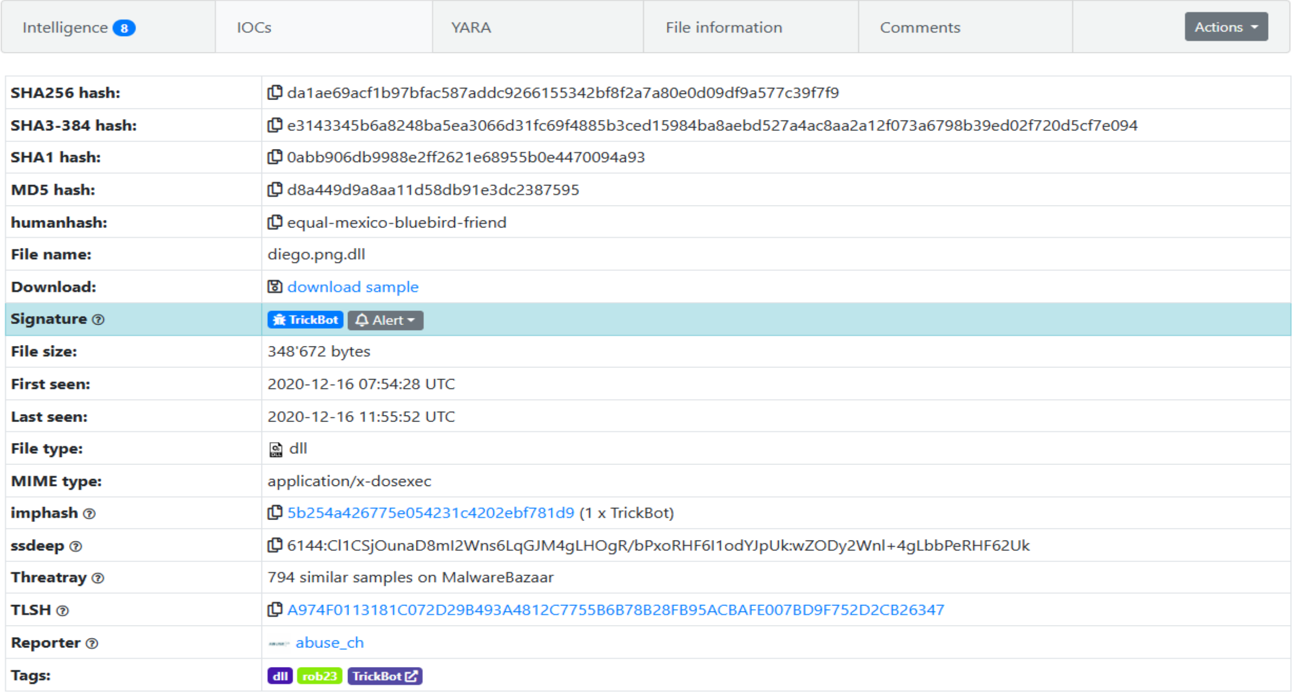
Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Malware/Win32.Generic.C4268630	Alibaba	Trojan:Win32/TrickBot.7d1c6f62
AliCloud	Trojan:Win/TrickBot.SB8PHU	ALYac	Spyware.Ursnif
Arcabit	Dump:Generic.ShellCode.RDI.Marte.1.1A...	Arctic Wolf	Unsafe
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
Avira (no cloud)	TR/AD.Emotet.pgzts	BitDefender	Dump:Generic.ShellCode.RDI.Marte.1.1A...

Figure 2. 5 Snippet of Diego.png(dll) hash results on VirusTotal

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

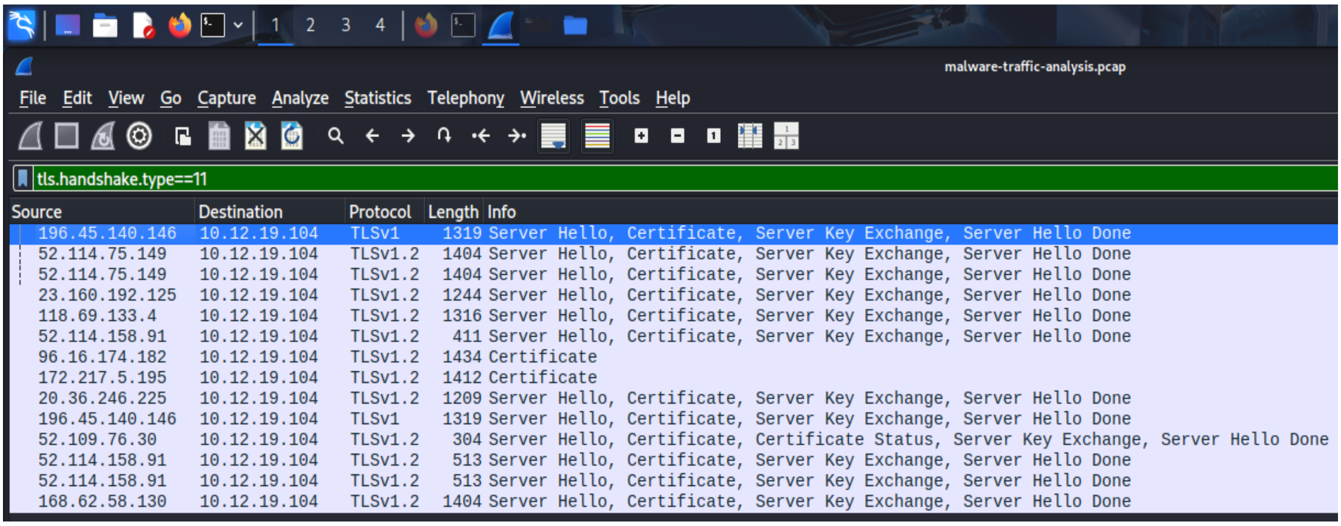


SHA256 hash:	da1ae69acf1b97bfac587addc9266155342bf8f2a7a80e0d09df9a577c39f7f9
SHA3-384 hash:	e3143345b6a8248ba5ea3066d31fc69f4885b3ced15984ba8aebd527a4ac8aa2a12f073a6798b39ed02f720d5cf7e094
SHA1 hash:	0abb906db9988e2ff2621e68955b0e4470094a93
MD5 hash:	d8a449d9a8aa11d58db91e3dc2387595
humanhash:	equal-mexico-bluebird-friend
File name:	diego.png.dll
Download:	download sample
Signature	TrickBot Alert
File size:	348'672 bytes
First seen:	2020-12-16 07:54:28 UTC
Last seen:	2020-12-16 11:55:52 UTC
File type:	dll
MIME type:	application/x-dosexec
imphash	5b254a426775e054231c4202ebf781d9 (1 x TrickBot)
ssdeep	6144:Cl1CSjOunaD8mI2Wns6LqGJM4gLHOgR/bPxoRHF6I1odYJpUk:wZODy2Wnl+4gLbbPeRHF62Uk
Threatray	794 similar samples on MalwareBazaar
TLSH	A974F0113181C072D29B493A4812C7755B6B78B28FB95ACBAFE007BD9F752D2CB26347
Reporter	abuse_ch
Tags:	dll rob23 TrickBot

Figure 2. 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.**.


malware-traffic-analysis.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tls.handshake.type==11
```

```
Source          Destination    Protocol Length Info
196.45.140.146  10.12.19.104   TLSv1    1319 Server Hello, Certificate, Server Key Exchange, Server Hello Done
52.114.75.149   10.12.19.104   TLSv1.2  1404 Server Hello, Certificate, Server Key Exchange, Server Hello Done
52.114.75.149   10.12.19.104   TLSv1.2  1404 Server Hello, Certificate, Server Key Exchange, Server Hello Done
23.160.192.125  10.12.19.104   TLSv1.2  1244 Server Hello, Certificate, Server Key Exchange, Server Hello Done
118.69.133.4    10.12.19.104   TLSv1.2  1316 Server Hello, Certificate, Server Key Exchange, Server Hello Done
52.114.158.91   10.12.19.104   TLSv1.2   411 Server Hello, Certificate, Server Key Exchange, Server Hello Done
96.16.174.182   10.12.19.104   TLSv1.2  1434 Certificate
172.217.5.195   10.12.19.104   TLSv1.2  1412 Certificate
20.36.246.225   10.12.19.104   TLSv1.2  1209 Server Hello, Certificate, Server Key Exchange, Server Hello Done
196.45.140.146  10.12.19.104   TLSv1    1319 Server Hello, Certificate, Server Key Exchange, Server Hello Done
52.109.76.30    10.12.19.104   TLSv1.2   304 Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
52.114.158.91   10.12.19.104   TLSv1.2   513 Server Hello, Certificate, Server Key Exchange, Server Hello Done
52.114.158.91   10.12.19.104   TLSv1.2   513 Server Hello, Certificate, Server Key Exchange, Server Hello Done
168.62.58.130   10.12.19.104   TLSv1.2  1404 Server Hello, Certificate, Server Key Exchange, Server Hello Done
```

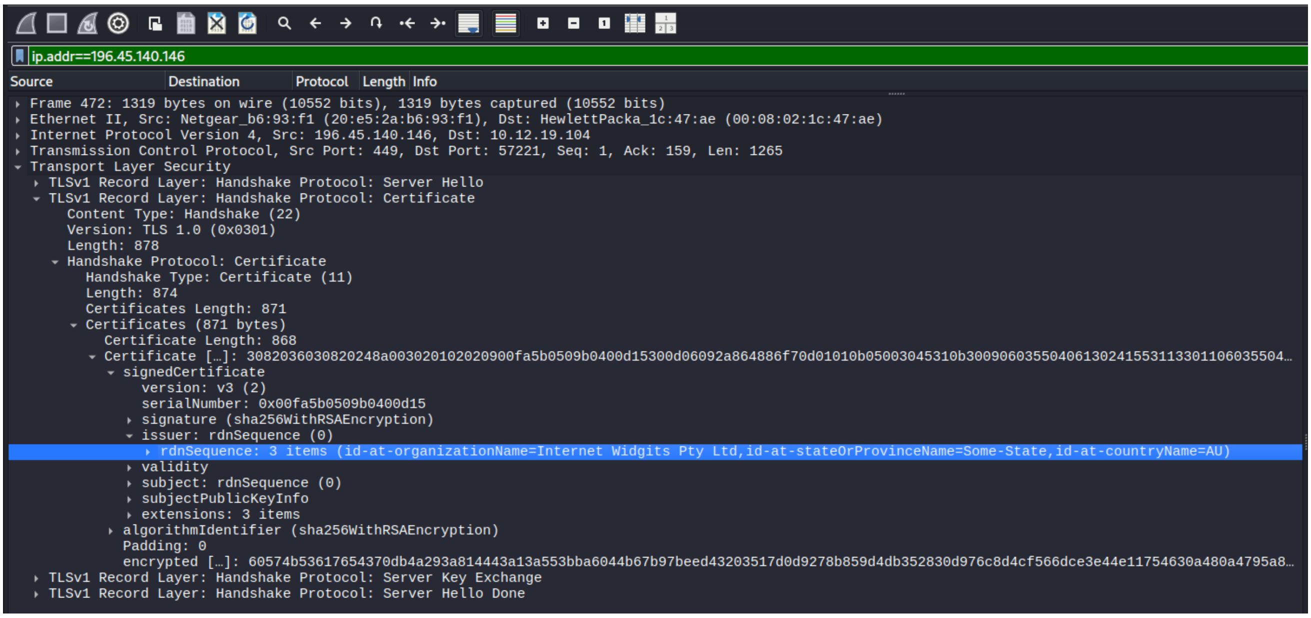
Figure 2. 7 TLS handshake filter result

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 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



```
▸ Frame 472: 1319 bytes on wire (10552 bits), 1319 bytes captured (10552 bits)
▸ Ethernet II, Src: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1), Dst: HewlettPacka_1c:47:ae (00:08:02:1c:47:ae)
▸ Internet Protocol Version 4, Src: 196.45.140.146, Dst: 10.12.19.104
▸ Transmission Control Protocol, Src Port: 449, Dst Port: 57221, Seq: 1, Ack: 159, Len: 1265
▾ Transport Layer Security
  ▸ TLSv1 Record Layer: Handshake Protocol: Server Hello
  ▾ TLSv1 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 878
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 874
        Certificates Length: 871
      ▾ Certificates (871 bytes)
          Certificate Length: 868
        ▾ Certificate […]: 3082036030820248a003020102020900fa5b0509b0400d15300d06092a864886f70d01010b05003045310b30090603550406130241553113301106035504…
          ▾ signedCertificate
              version: v3 (2)
              serialNumber: 0x00fa5b0509b0400d15
            ▸ signature (sha256WithRSAEncryption)
            ▾ issuer: rdnSequence (0)
              ▸ rdnSequence: 3 items (id-at-organizationName=Internet Widgits Pty Ltd,id-at-stateOrProvinceName=Some-State,id-at-countryName=AU)
            ▸ validity
            ▸ subject: rdnSequence (0)
            ▸ subjectPublicKeyInfo
            ▸ extensions: 3 items
          ▸ algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted […]: 60574b53617654370db4a293a814443a13a553bba6044b67b97beed43203517d0d9278b859d4db352830d976c8d4cf566dce3e44e11754630a480a4795a8…
  ▸ TLSv1 Record Layer: Handshake Protocol: Server Key Exchange
  ▸ TLSv1 Record Layer: Handshake Protocol: Server Hello Done
```

Figure 2. 8 Suspected C2 server (1) certificate

2. 23.160.192.125

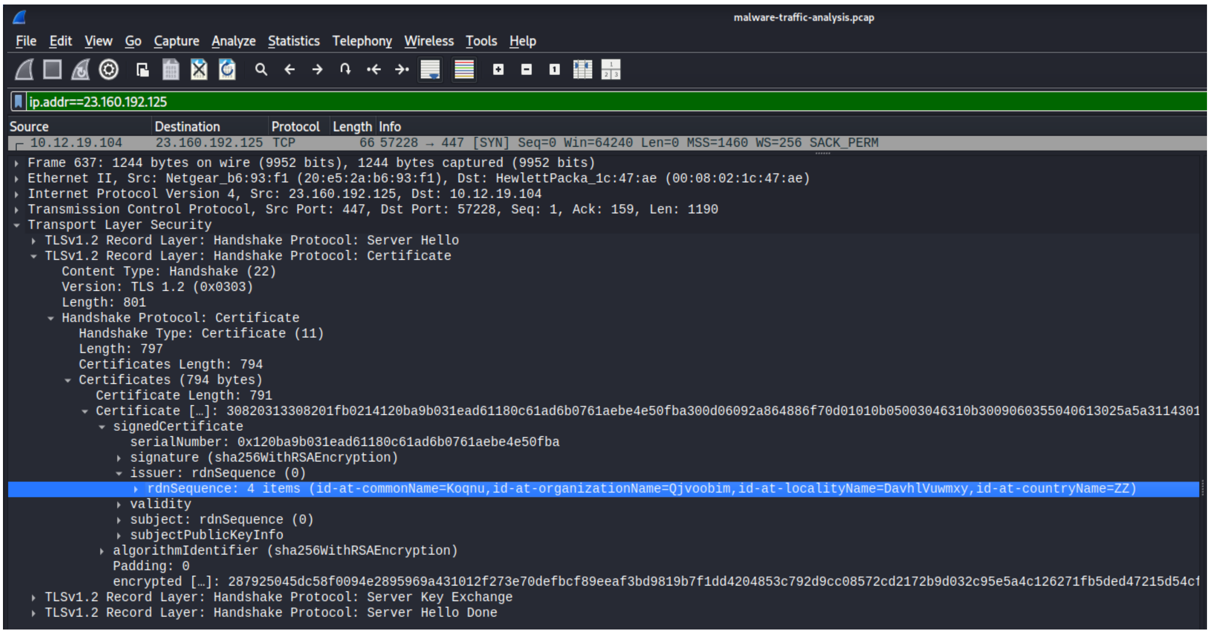
Certificate Issuer: Qjvoobim

Port: 447

Communication Protocols used: TCP, TLSV1.2

Host communicated with: Only the affected host (10.12.19.104)

Total packets: 19

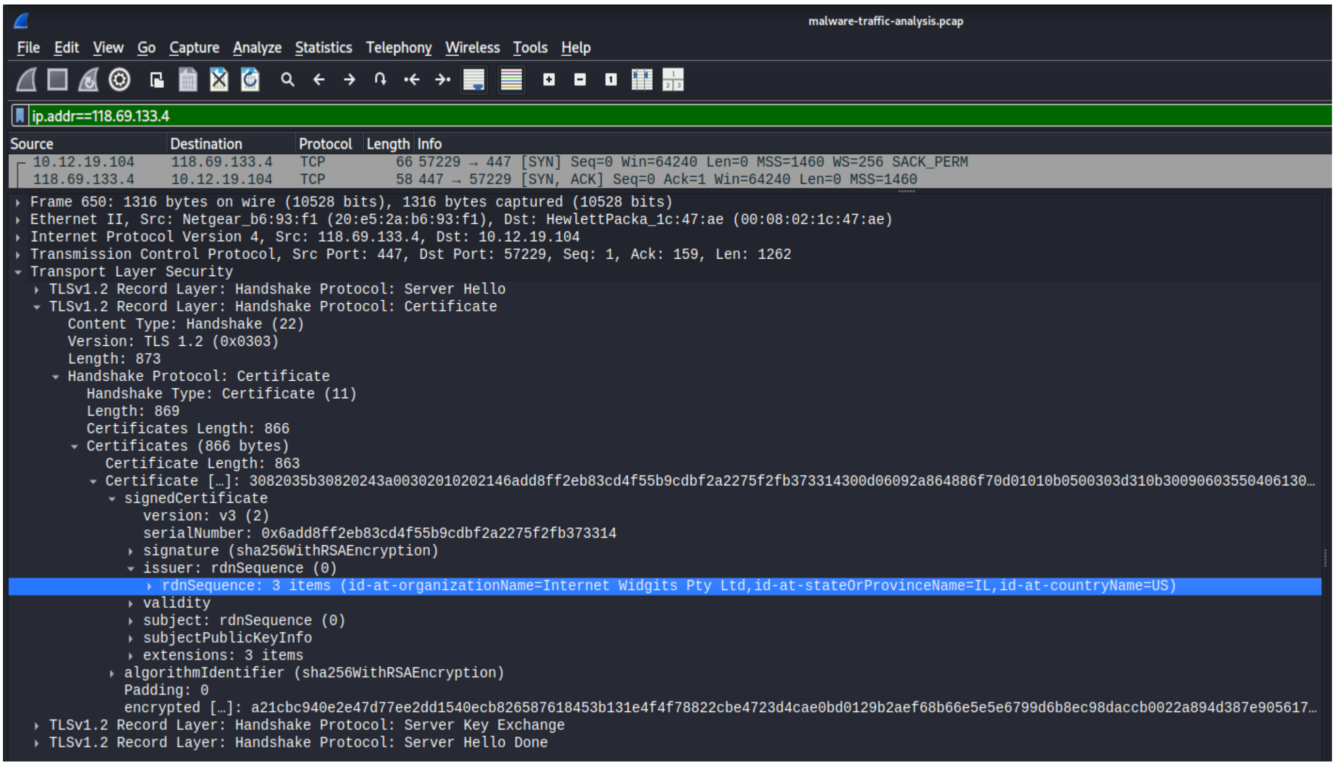


```
┌ 10.12.19.104    23.160.192.125  TCP        66 57228 → 447 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
▸ Frame 637: 1244 bytes on wire (9952 bits), 1244 bytes captured (9952 bits)
▸ Ethernet II, Src: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1), Dst: HewlettPacka_1c:47:ae (00:08:02:1c:47:ae)
▸ Internet Protocol Version 4, Src: 23.160.192.125, Dst: 10.12.19.104
▸ Transmission Control Protocol, Src Port: 447, Dst Port: 57228, Seq: 1, Ack: 159, Len: 1190
▾ Transport Layer Security
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 801
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 797
        Certificates Length: 794
      ▾ Certificates (794 bytes)
          Certificate Length: 791
        ▾ Certificate […]: 30820313308201fb0214120ba9b031ead61180c61ad6b0761aebe4e50fba300d06092a864886f70d01010b05003046310b300906035504061302 5a5a3114301
          ▾ signedCertificate
              serialNumber: 0x120ba9b031ead61180c61ad6b0761aebe4e50fba
            ▸ signature (sha256WithRSAEncryption)
            ▾ issuer: rdnSequence (0)
              ▸ rdnSequence: 4 items (id-at-commonName=Koqnu,id-at-organizationName=Qjvoobim,id-at-localityName=DavhlVuwmxy,id-at-countryName=ZZ)
            ▸ validity
            ▸ subject: rdnSequence (0)
            ▸ subjectPublicKeyInfo
          ▸ algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted […]: 287925045dc58f0094e2895969a431012f273e70defbcf89eeaf3bd9819b7f1dd4204853c792d9cc08572cd2172b9d032c95e5a4c126271fb5ded47215d54cf
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
```

Figure 2. 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

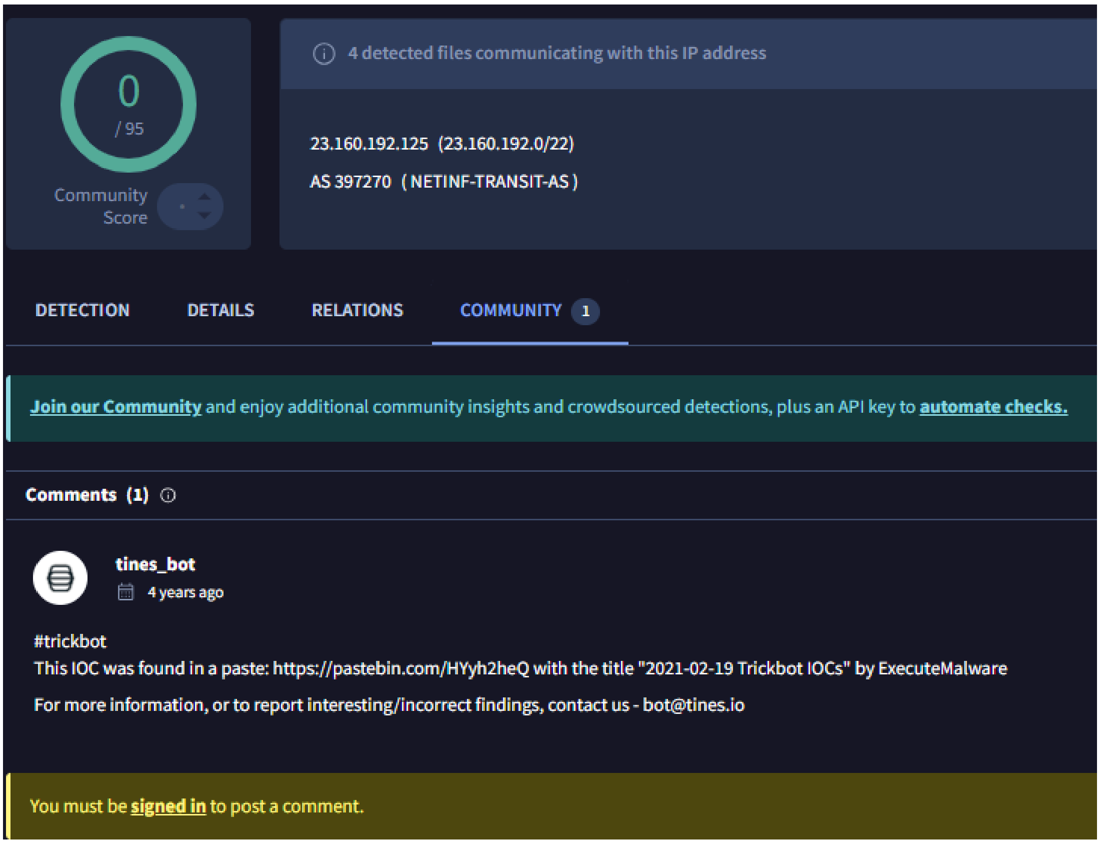

malware-traffic-analysis.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.addr==118.69.133.4

Source          Destination     Protocol  Length Info
 10.12.19.104   118.69.133.4    TCP           66 57229 → 447 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
 118.69.133.4   10.12.19.104    TCP           58 447 → 57229 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

› Frame 650: 1316 bytes on wire (10528 bits), 1316 bytes captured (10528 bits)
› Ethernet II, Src: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1), Dst: HewlettPacka_1c:47:ae (00:08:02:1c:47:ae)
› Internet Protocol Version 4, Src: 118.69.133.4, Dst: 10.12.19.104
› Transmission Control Protocol, Src Port: 447, Dst Port: 57229, Seq: 1, Ack: 159, Len: 1262
˅ Transport Layer Security
  › TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  ˅ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 873
    ˅ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 869
        Certificates Length: 866
      ˅ Certificates (866 bytes)
          Certificate Length: 863
        ˅ Certificate […]: 3082035b30820243a00302010202146add8ff2eb83cd4f55b9cdbf2a2275f2fb373314300d06092a864886f70d01010b0500303d310b30090603550406130…
          ˅ signedCertificate
              version: v3 (2)
              serialNumber: 0x6add8ff2eb83cd4f55b9cdbf2a2275f2fb373314
            › signature (sha256WithRSAEncryption)
            ˅ issuer: rdnSequence (0)
              › rdnSequence: 3 items (id-at-organizationName=Internet Widgits Pty Ltd,id-at-stateOrProvinceName=IL,id-at-countryName=US)
            › validity
            › subject: rdnSequence (0)
            › subjectPublicKeyInfo
            › extensions: 3 items
          › algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted […]: a21cbc940e2e47d77ee2dd1540ecb826587618453b131e4f4f78822cbe4723d4cae0bd0129b2aef68b66e5e5e6799d6b8ec98daccb0022a894d387e905617…
  › TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
  › TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
```

Figure 2. 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.



4 detected files communicating with this IP address

23.160.192.125 (23.160.192.0/22)
AS 397270 (NETINF-TRANSIT-AS)

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Comments (1)

tines_bot
4 years ago

#trickbot
This IOC was found in a paste: https://pastebin.com/HYyh2heQ with the title "2021-02-19 Trickbot IOCs" by ExecuteMalware
For more information, or to report interesting/incorrect findings, contact us - bot@tines.io

You must be signed in to post a comment.

Figure 2. 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

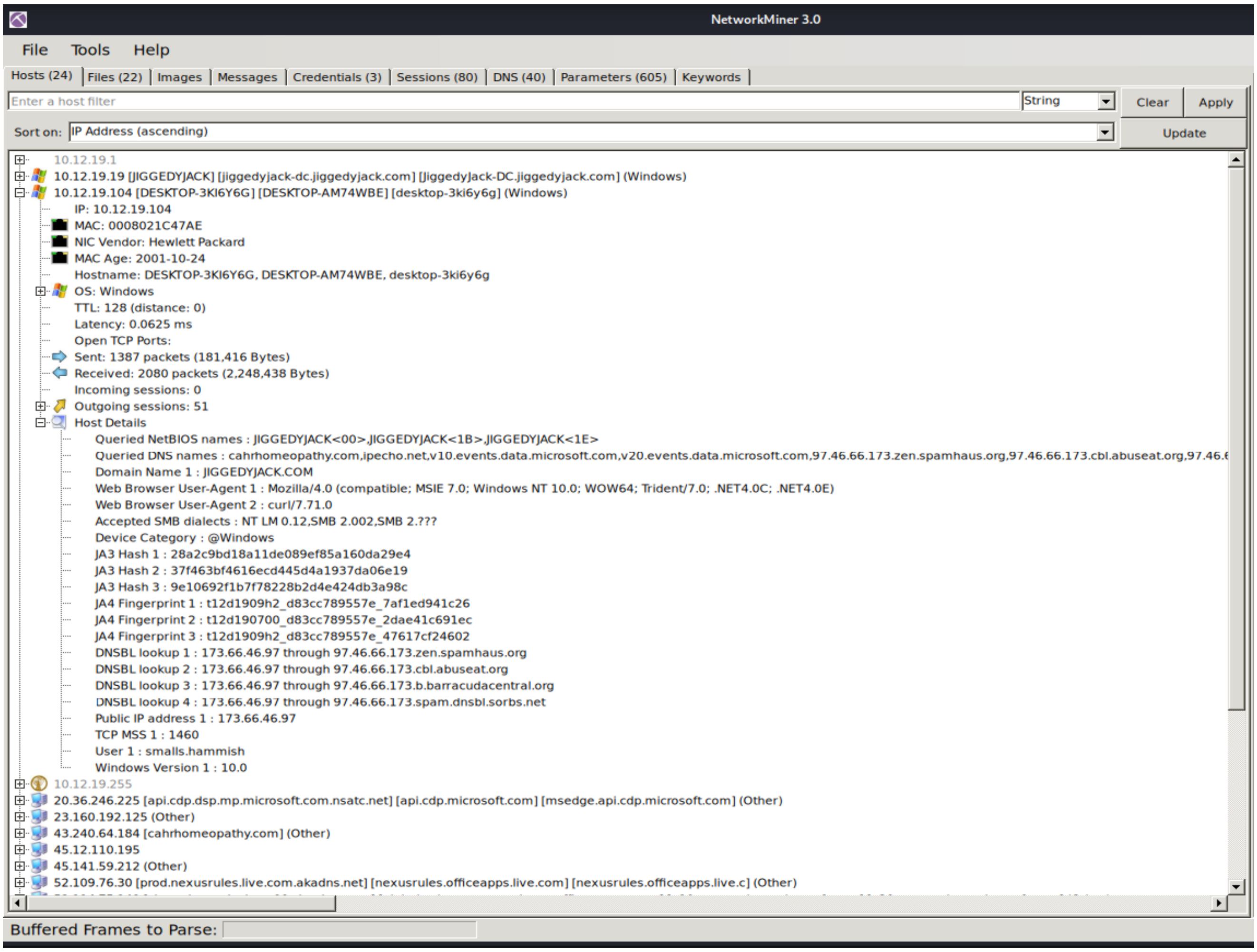
Communication Protocols used: TCP, HTTP

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.



```
10.12.19.1
10.12.19.19 [JIGGEDYJACK] [jiggedyjack-dc.jiggedyjack.com] [JiggedyJack-DC.jiggedyjack.com] (Windows)
10.12.19.104 [DESKTOP-3KI6Y6G] [DESKTOP-AM74WBE] [desktop-3ki6y6g] (Windows)
    IP: 10.12.19.104
    MAC: 0008021C47AE
    NIC Vendor: Hewlett Packard
    MAC Age: 2001-10-24
    Hostname: DESKTOP-3KI6Y6G, DESKTOP-AM74WBE, desktop-3ki6y6g
    OS: Windows
    TTL: 128 (distance: 0)
    Latency: 0.0625 ms
    Open TCP Ports:
    Sent: 1387 packets (181,416 Bytes)
    Received: 2080 packets (2,248,438 Bytes)
    Incoming sessions: 0
    Outgoing sessions: 51
    Host Details
        Queried NetBIOS names : JIGGEDYJACK<00>,JIGGEDYJACK<1B>,JIGGEDYJACK<1E>
        Queried DNS names : cahrhomeopathy.com,ipecho.net,v10.events.data.microsoft.com,v20.events.data.microsoft.com,97.46.66.173.zen.spamhaus.org,97.46.66.173.cbl.abuseat.org,97.46.6
        Domain Name 1 : JIGGEDYJACK.COM
        Web Browser User-Agent 1 : Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E)
        Web Browser User-Agent 2 : curl/7.71.0
        Accepted SMB dialects : NT LM 0.12,SMB 2.002,SMB 2.???
        Device Category : @Windows
        JA3 Hash 1 : 28a2c9bd18a11de089ef85a160da29e4
        JA3 Hash 2 : 37f463bf4616ecd445d4a1937da06e19
        JA3 Hash 3 : 9e10692f1b7f78228b2d4e424db3a98c
        JA4 Fingerprint 1 : t12d1909h2_d83cc789557e_7af1ed941c26
        JA4 Fingerprint 2 : t12d190700_d83cc789557e_2dae41c691ec
        JA4 Fingerprint 3 : t12d1909h2_d83cc789557e_47617cf24602
        DNSBL lookup 1 : 173.66.46.97 through 97.46.66.173.zen.spamhaus.org
        DNSBL lookup 2 : 173.66.46.97 through 97.46.66.173.cbl.abuseat.org
        DNSBL lookup 3 : 173.66.46.97 through 97.46.66.173.b.barracudacentral.org
        DNSBL lookup 4 : 173.66.46.97 through 97.46.66.173.spam.dnsbl.sorbs.net
        Public IP address 1 : 173.66.46.97
        TCP MSS 1 : 1460
        User 1 : smalls.hammish
        Windows Version 1 : 10.0
10.12.19.255
20.36.246.225 [api.cdp.dsp.mp.microsoft.com.nsatc.net] [api.cdp.microsoft.com] [msedge.api.cdp.microsoft.com] (Other)
23.160.192.125 (Other)
43.240.64.184 [cahrhomeopathy.com] (Other)
45.12.110.195
45.141.59.212 (Other)
52.109.76.30 [prod.nexusrules.live.com.akadns.net] [nexusrules.officeapps.live.com] [nexusrules.officeapps.live.c] (Other)
```

Figure 2. 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

1. **MALWARE FAMILY**

   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

1. **Rossouw.** (2025). *Threat hunting C2 over HTTPS connections using the TLS certificate*. Active Countermeasures. https://www.activecountermeasures.com/threat-hunting-c2-over-https-connections-using-the-tls-certificate/

2. Cybersecurity and Infrastructure Security Agency. (2021). *TrickBot malware* (AA21-076A). https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-076a

3. **Mokbel.** (2021). *Analyzing SSL/TLS certificates used by malware*. Trend Micro. https://www.trendmicro.com/en_gb/research/21/i/analyzing-ssl-tls-certificates-used-by-malware.html

BAD PDF CASE (MEMORY FORENSICS)

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 co-worker 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.
- **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 analyze 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.

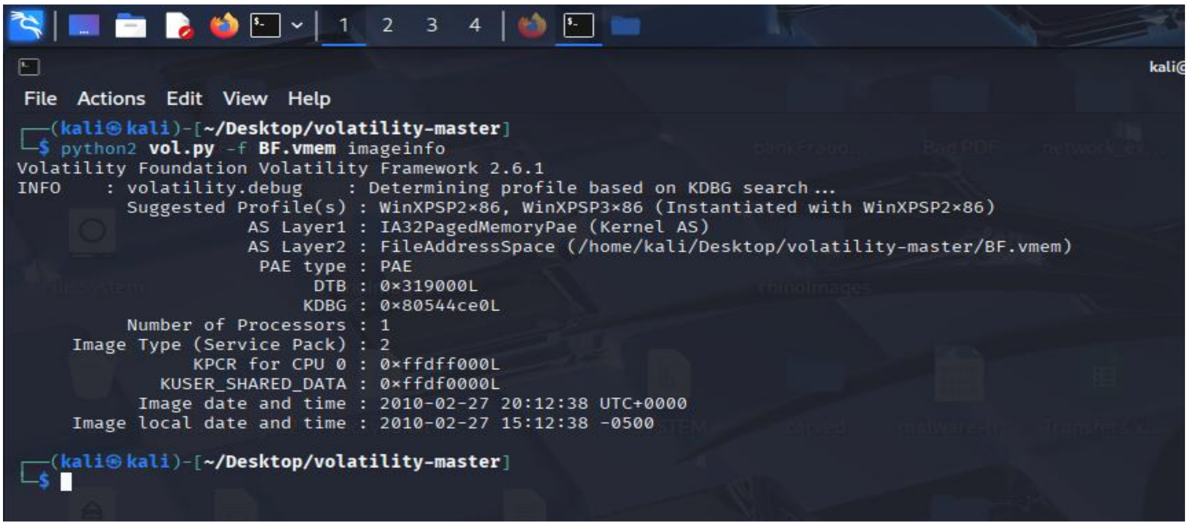
Volatility Plugins Deployed

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

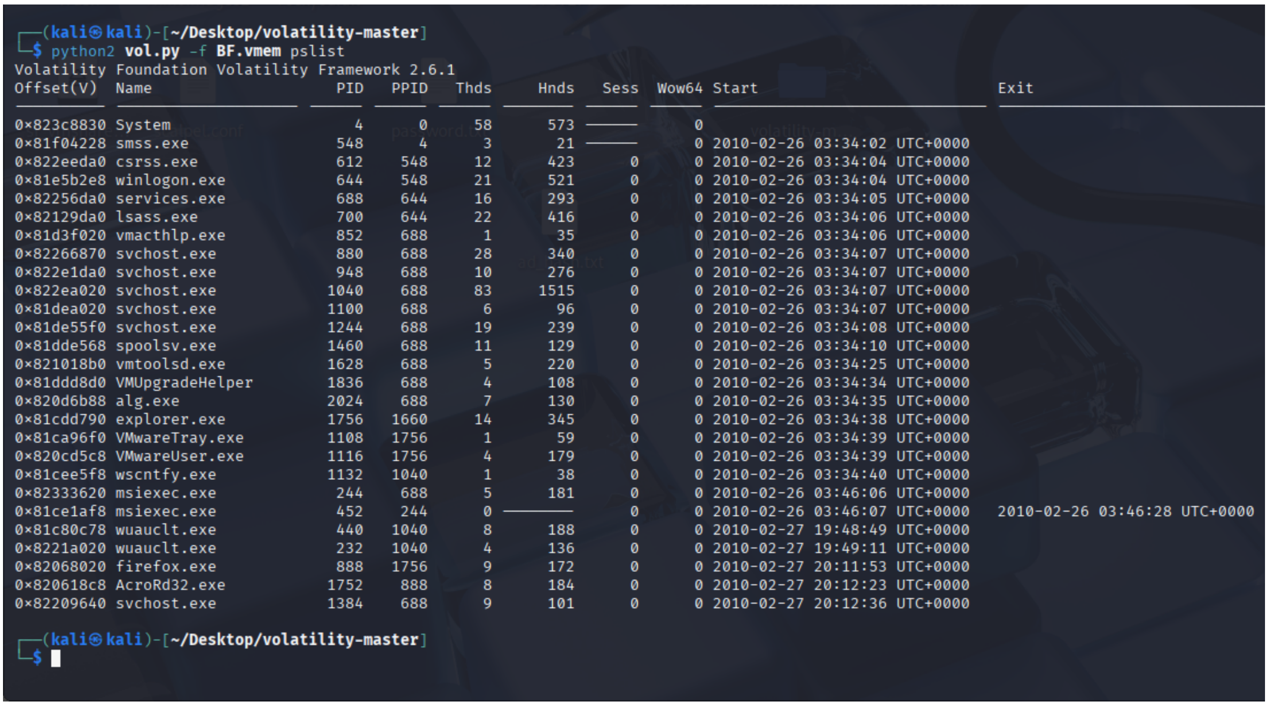
Using volatility2 on my kali, I used **imageinfo** to get the suggested profile to work on, volatility2 picked and instantiated with profile WinXPSP2x86 so there would be no need to call the profile I want to work on when using volatility2.



```
┌──(kali㉿kali)-[~/Desktop/volatility-master]
└─$ python2 vol.py -f BF.vmem imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search ...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/kali/Desktop/volatility-master/BF.vmem)
                      PAE type : PAE
                           DTB : 0x319000L
                          KDBG : 0x80544ce0L
          Number of Processors : 1
     Image Type (Service Pack) : 2
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2010-02-27 20:12:38 UTC+0000
     Image local date and time : 2010-02-27 15:12:38 -0500

┌──(kali㉿kali)-[~/Desktop/volatility-master]
└─$ █
```

Figure 3. 1 Imageinfo Result

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

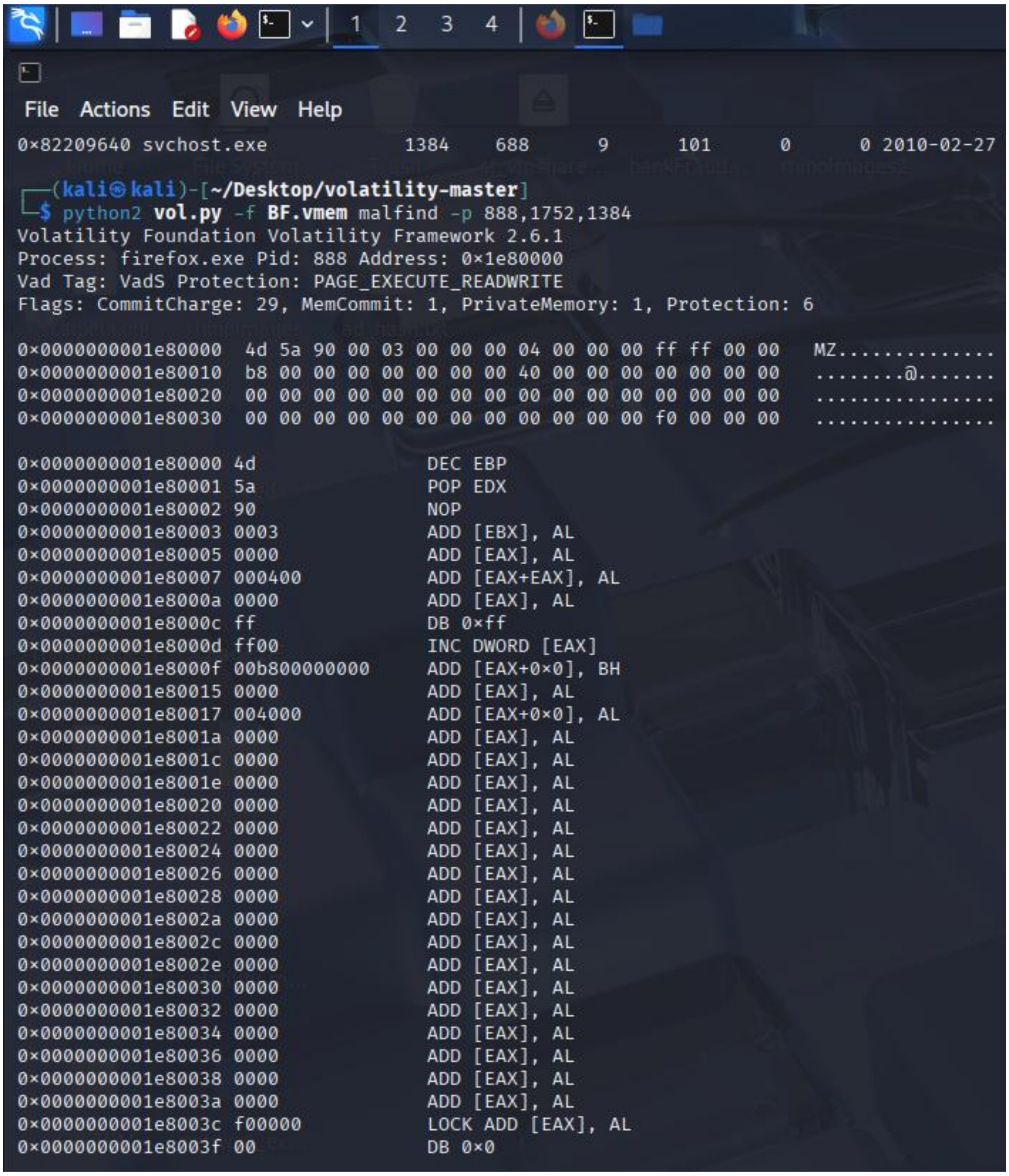


```
┌──(kali㉿kali)-[~/Desktop/volatility-master]
└─$ python2 vol.py -f BF.vmem pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x823c8830 System                    4      0     58      573 ------      0
0x81f04228 smss.exe                548      4      3       21 ------      0 2010-02-26 03:34:02 UTC+0000
0x822eeda0 csrss.exe               612    548     12      423      0      0 2010-02-26 03:34:04 UTC+0000
0x81e5b2e8 winlogon.exe            644    548     21      521      0      0 2010-02-26 03:34:04 UTC+0000
0x82256da0 services.exe            688    644     16      293      0      0 2010-02-26 03:34:05 UTC+0000
0x82129da0 lsass.exe               700    644     22      416      0      0 2010-02-26 03:34:06 UTC+0000
0x81d3f020 vmacthlp.exe            852    688      1       35      0      0 2010-02-26 03:34:06 UTC+0000
0x82266870 svchost.exe             880    688     28      340      0      0 2010-02-26 03:34:07 UTC+0000
0x822e1da0 svchost.exe             948    688     10      276      0      0 2010-02-26 03:34:07 UTC+0000
0x822ea020 svchost.exe            1040    688     83     1515      0      0 2010-02-26 03:34:07 UTC+0000
0x81dea020 svchost.exe            1100    688      6       96      0      0 2010-02-26 03:34:07 UTC+0000
0x81de55f0 svchost.exe            1244    688     19      239      0      0 2010-02-26 03:34:08 UTC+0000
0x81dde568 spoolsv.exe            1460    688     11      129      0      0 2010-02-26 03:34:10 UTC+0000
0x821018b0 vmtoolsd.exe           1628    688      5      220      0      0 2010-02-26 03:34:25 UTC+0000
0x81ddd8d0 VMUpgradeHelper        1836    688      4      108      0      0 2010-02-26 03:34:34 UTC+0000
0x820d6b88 alg.exe                2024    688      7      130      0      0 2010-02-26 03:34:35 UTC+0000
0x81cdd790 explorer.exe           1756   1660     14      345      0      0 2010-02-26 03:34:38 UTC+0000
0x81ca96f0 VMwareTray.exe         1108   1756      1       59      0      0 2010-02-26 03:34:39 UTC+0000
0x820cd5c8 VMwareUser.exe         1116   1756      4      179      0      0 2010-02-26 03:34:39 UTC+0000
0x81cee5f8 wscntfy.exe            1132   1040      1       38      0      0 2010-02-26 03:34:40 UTC+0000
0x82333620 msiexec.exe             244    688      5      181      0      0 2010-02-26 03:46:06 UTC+0000
0x81ce1af8 msiexec.exe             452    244      0 --------      0      0 2010-02-26 03:46:07 UTC+0000   2010-02-26 03:46:28 UTC+0000
0x81c80c78 wuauclt.exe             440   1040      8      188      0      0 2010-02-27 19:48:49 UTC+0000
0x8221a020 wuauclt.exe             232   1040      4      136      0      0 2010-02-27 19:49:11 UTC+0000
0x82068020 firefox.exe             888   1756      9      172      0      0 2010-02-27 20:11:53 UTC+0000
0x820618c8 AcroRd32.exe           1752    888      8      184      0      0 2010-02-27 20:12:23 UTC+0000
0x82209640 svchost.exe            1384    688      9      101      0      0 2010-02-27 20:12:36 UTC+0000

┌──(kali㉿kali)-[~/Desktop/volatility-master]
└─$ █
```

Figure 3. 2 Pslist list result

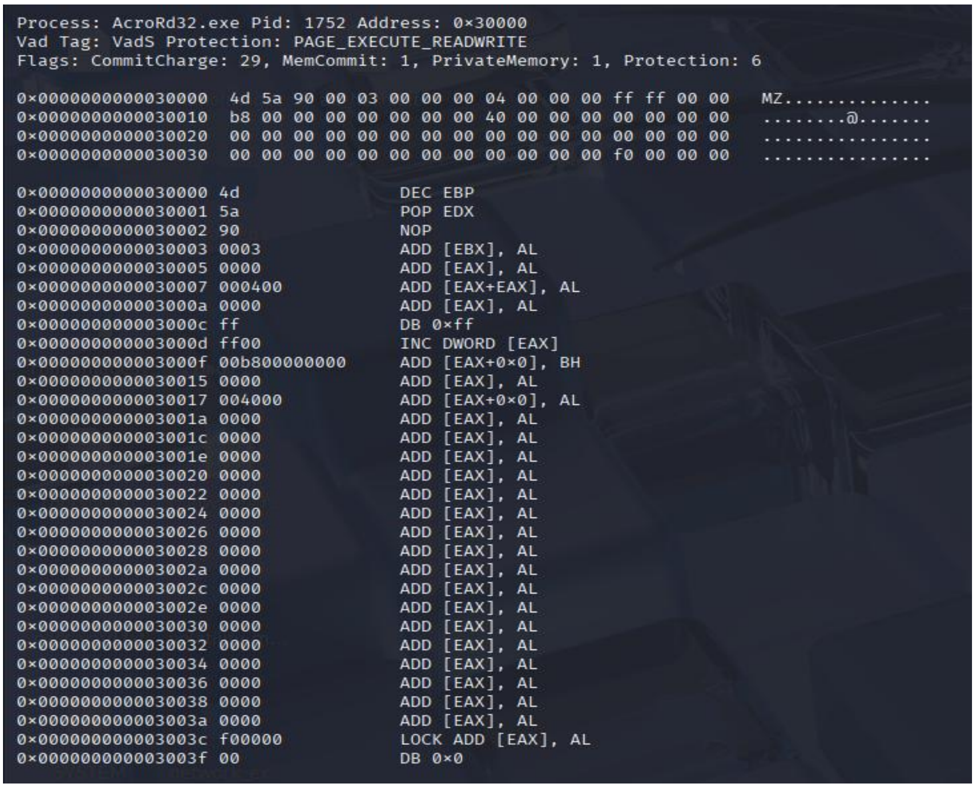
From the list, the svchost.exe with PID 1384 is out of sync with other system processes and it started at 20:12:36 UTC, just 3 seconds after process AcroRd32.exe with PID 1752 started which suggests process AcroRd32.exe might have caused the system process to start which is unusual.

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.



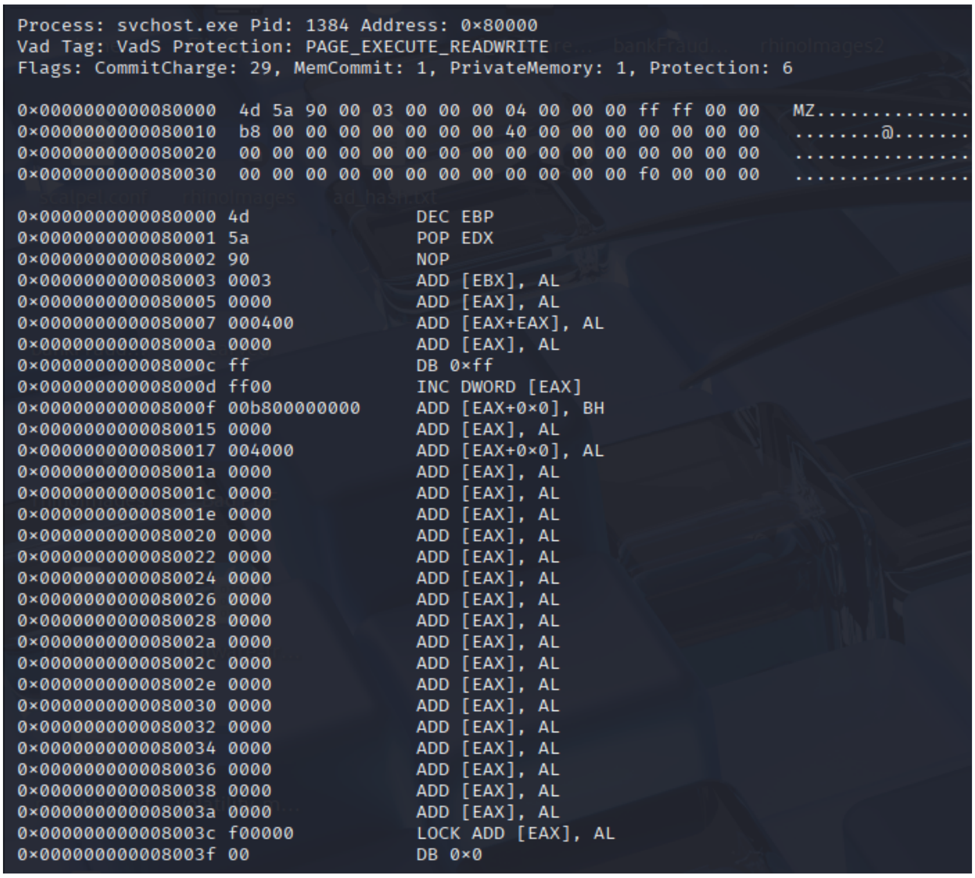
```
0x82209640 svchost.exe            1384    688       9      101      0       0 2010-02-27

┌──(kali㉿kali)-[~/Desktop/volatility-master]
└─$ python2 vol.py -f BF.vmem malfind -p 888,1752,1384
Volatility Foundation Volatility Framework 2.6.1
Process: firefox.exe Pid: 888 Address: 0x1e80000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 29, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0000000001e80000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x0000000001e80010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x0000000001e80020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0000000001e80030  00 00 00 00 00 00 00 00 00 00 00 00 f0 00 00 00   ................

0x0000000001e80000 4d               DEC EBP
0x0000000001e80001 5a               POP EDX
0x0000000001e80002 90               NOP
0x0000000001e80003 0003             ADD [EBX], AL
0x0000000001e80005 0000             ADD [EAX], AL
0x0000000001e80007 000400           ADD [EAX+EAX], AL
0x0000000001e8000a 0000             ADD [EAX], AL
0x0000000001e8000c ff               DB 0xff
0x0000000001e8000d ff00             INC DWORD [EAX]
0x0000000001e8000f 00b800000000     ADD [EAX+0x0], BH
0x0000000001e80015 0000             ADD [EAX], AL
0x0000000001e80017 004000           ADD [EAX+0x0], AL
0x0000000001e8001a 0000             ADD [EAX], AL
0x0000000001e8001c 0000             ADD [EAX], AL
0x0000000001e8001e 0000             ADD [EAX], AL
0x0000000001e80020 0000             ADD [EAX], AL
0x0000000001e80022 0000             ADD [EAX], AL
0x0000000001e80024 0000             ADD [EAX], AL
0x0000000001e80026 0000             ADD [EAX], AL
0x0000000001e80028 0000             ADD [EAX], AL
0x0000000001e8002a 0000             ADD [EAX], AL
0x0000000001e8002c 0000             ADD [EAX], AL
0x0000000001e8002e 0000             ADD [EAX], AL
0x0000000001e80030 0000             ADD [EAX], AL
0x0000000001e80032 0000             ADD [EAX], AL
0x0000000001e80034 0000             ADD [EAX], AL
0x0000000001e80036 0000             ADD [EAX], AL
0x0000000001e80038 0000             ADD [EAX], AL
0x0000000001e8003a 0000             ADD [EAX], AL
0x0000000001e8003c f00000           LOCK ADD [EAX], AL
0x0000000001e8003f 00               DB 0x0
```

Figure 3. 3 Malfind on firefox.exe (PID 888)



```
Process: AcroRd32.exe Pid: 1752 Address: 0×30000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 29, MemCommit: 1, PrivateMemory: 1, Protection: 6

0×0000000000030000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0×0000000000030010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0×0000000000030020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0×0000000000030030  00 00 00 00 00 00 00 00 00 00 00 00 f0 00 00 00   ................

0×0000000000030000 4d                DEC EBP
0×0000000000030001 5a                POP EDX
0×0000000000030002 90                NOP
0×0000000000030003 0003              ADD [EBX], AL
0×0000000000030005 0000              ADD [EAX], AL
0×0000000000030007 000400            ADD [EAX+EAX], AL
0×000000000003000a 0000              ADD [EAX], AL
0×000000000003000c ff                DB 0×ff
0×000000000003000d ff00              INC DWORD [EAX]
0×000000000003000f 00b800000000      ADD [EAX+0×0], BH
0×0000000000030015 0000              ADD [EAX], AL
0×0000000000030017 004000            ADD [EAX+0×0], AL
0×000000000003001a 0000              ADD [EAX], AL
0×000000000003001c 0000              ADD [EAX], AL
0×000000000003001e 0000              ADD [EAX], AL
0×0000000000030020 0000              ADD [EAX], AL
0×0000000000030022 0000              ADD [EAX], AL
0×0000000000030024 0000              ADD [EAX], AL
0×0000000000030026 0000              ADD [EAX], AL
0×0000000000030028 0000              ADD [EAX], AL
0×000000000003002a 0000              ADD [EAX], AL
0×000000000003002c 0000              ADD [EAX], AL
0×000000000003002e 0000              ADD [EAX], AL
0×0000000000030030 0000              ADD [EAX], AL
0×0000000000030032 0000              ADD [EAX], AL
0×0000000000030034 0000              ADD [EAX], AL
0×0000000000030036 0000              ADD [EAX], AL
0×0000000000030038 0000              ADD [EAX], AL
0×000000000003003a 0000              ADD [EAX], AL
0×000000000003003c f00000            LOCK ADD [EAX], AL
0×000000000003003f 00                DB 0×0
```

Figure 3. 4 Malfind on AcroRd32.exe (PID 1752)



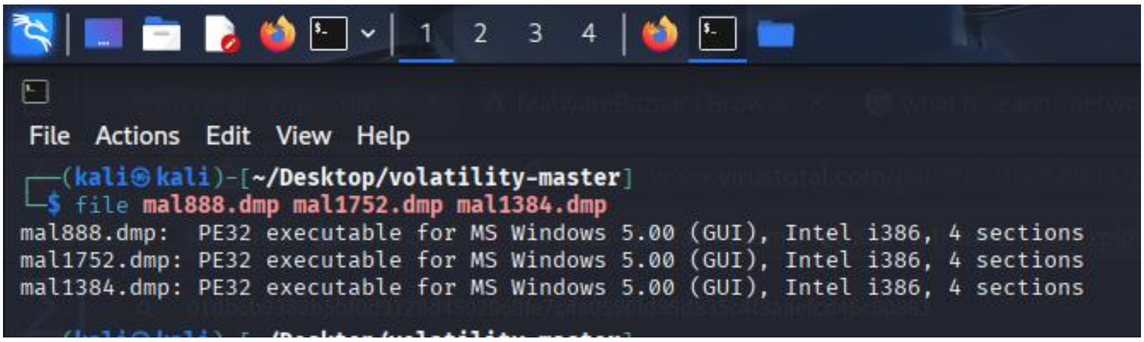
```
Process: svchost.exe Pid: 1384 Address: 0×80000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 29, MemCommit: 1, PrivateMemory: 1, Protection: 6

0×0000000000080000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0×0000000000080010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0×0000000000080020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0×0000000000080030  00 00 00 00 00 00 00 00 00 00 00 00 f0 00 00 00   ................

0×0000000000080000 4d                DEC EBP
0×0000000000080001 5a                POP EDX
0×0000000000080002 90                NOP
0×0000000000080003 0003              ADD [EBX], AL
0×0000000000080005 0000              ADD [EAX], AL
0×0000000000080007 000400            ADD [EAX+EAX], AL
0×000000000008000a 0000              ADD [EAX], AL
0×000000000008000c ff                DB 0×ff
0×000000000008000d ff00              INC DWORD [EAX]
0×000000000008000f 00b800000000      ADD [EAX+0×0], BH
0×0000000000080015 0000              ADD [EAX], AL
0×0000000000080017 004000            ADD [EAX+0×0], AL
0×000000000008001a 0000              ADD [EAX], AL
0×000000000008001c 0000              ADD [EAX], AL
0×000000000008001e 0000              ADD [EAX], AL
0×0000000000080020 0000              ADD [EAX], AL
0×0000000000080022 0000              ADD [EAX], AL
0×0000000000080024 0000              ADD [EAX], AL
0×0000000000080026 0000              ADD [EAX], AL
0×0000000000080028 0000              ADD [EAX], AL
0×000000000008002a 0000              ADD [EAX], AL
0×000000000008002c 0000              ADD [EAX], AL
0×000000000008002e 0000              ADD [EAX], AL
0×0000000000080030 0000              ADD [EAX], AL
0×0000000000080032 0000              ADD [EAX], AL
0×0000000000080034 0000              ADD [EAX], AL
0×0000000000080036 0000              ADD [EAX], AL
0×0000000000080038 0000              ADD [EAX], AL
0×000000000008003a 0000              ADD [EAX], AL
0×000000000008003c f00000            LOCK ADD [EAX], AL
0×000000000008003f 00                DB 0×0
```

Figure 3. 5 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.

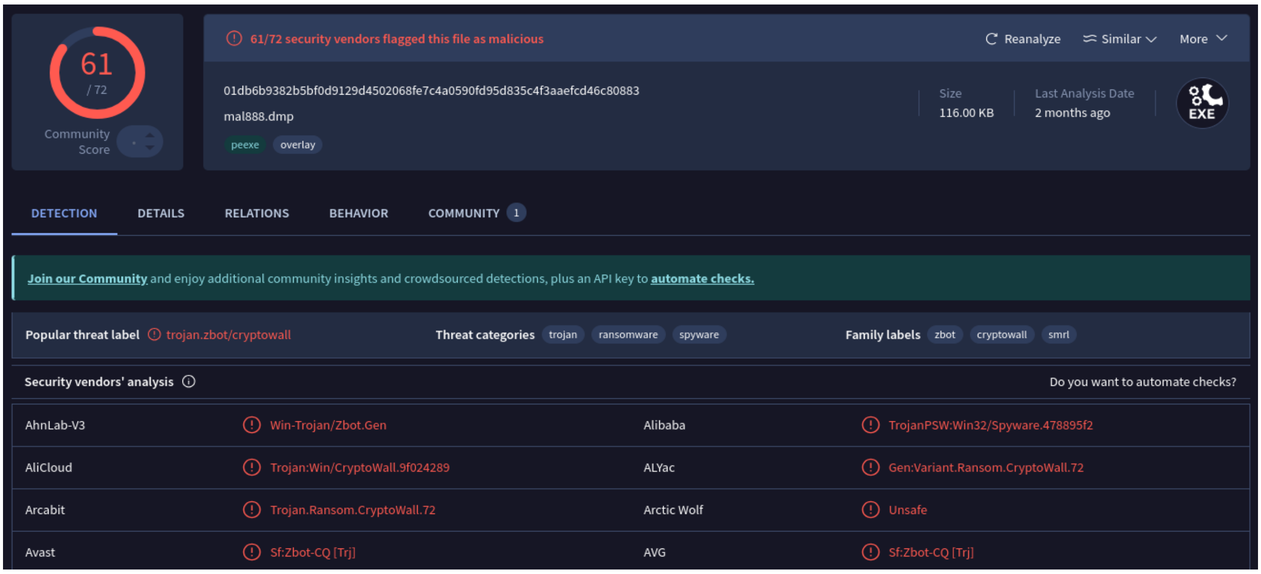
Using **python2 vol.py -f BF.vmem malfind -p 888,1752,1384 --dump-dir .**

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.



```
┌──(kali㉿kali)-[~/Desktop/volatility-master]
└─$ file mal888.dmp mal1752.dmp mal1384.dmp
mal888.dmp:  PE32 executable for MS Windows 5.00 (GUI), Intel i386, 4 sections
mal1752.dmp: PE32 executable for MS Windows 5.00 (GUI), Intel i386, 4 sections
mal1384.dmp: PE32 executable for MS Windows 5.00 (GUI), Intel i386, 4 sections
```

Figure 3. 6 File Result on dumped MZ files



61 / 72 Community Score

61/72 security vendors flagged this file as malicious

01db6b9382b5bf0d9129d4502068fe7c4a0590fd95d835c4f3aaefcd46c80883

mal888.dmp

peexe overlay

Size 116.00 KB

Last Analysis Date 2 months ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.zbot/cryptowall Threat categories trojan ransomware spyware Family labels zbot cryptowall smrl

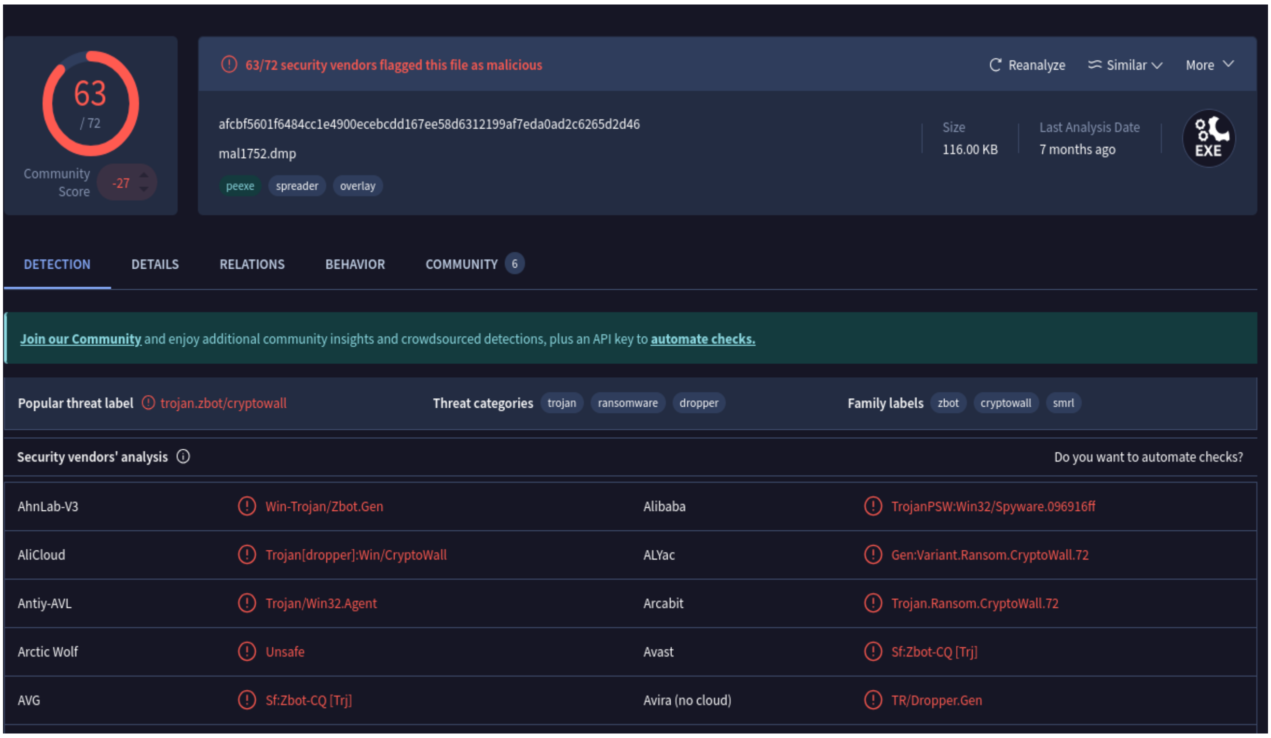
Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Win-Trojan/Zbot.Gen	Alibaba	TrojanPSW:Win32/Spyware.478895f2
AliCloud	Trojan:Win/CryptoWall.9f024289	ALYac	Gen:Variant.Ransom.CryptoWall.72
Arcabit	Trojan.Ransom.CryptoWall.72	Arctic Wolf	Unsafe
Avast	Sf:Zbot-CQ [Trj]	AVG	Sf:Zbot-CQ [Trj]

Figure 3. 7 MZ file on PID 888 VirusTotal Result

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



63/72 security vendors flagged this file as malicious

afcbf5601f6484cc1e4900ecebcdd167ee58d6312199af7eda0ad2c6265d2d46

mal1752.dmp

peexe spreader overlay

Size 116.00 KB

Last Analysis Date 7 months ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.zbot/cryptowall

Threat categories trojan ransomware dropper

Family labels zbot cryptowall smrl

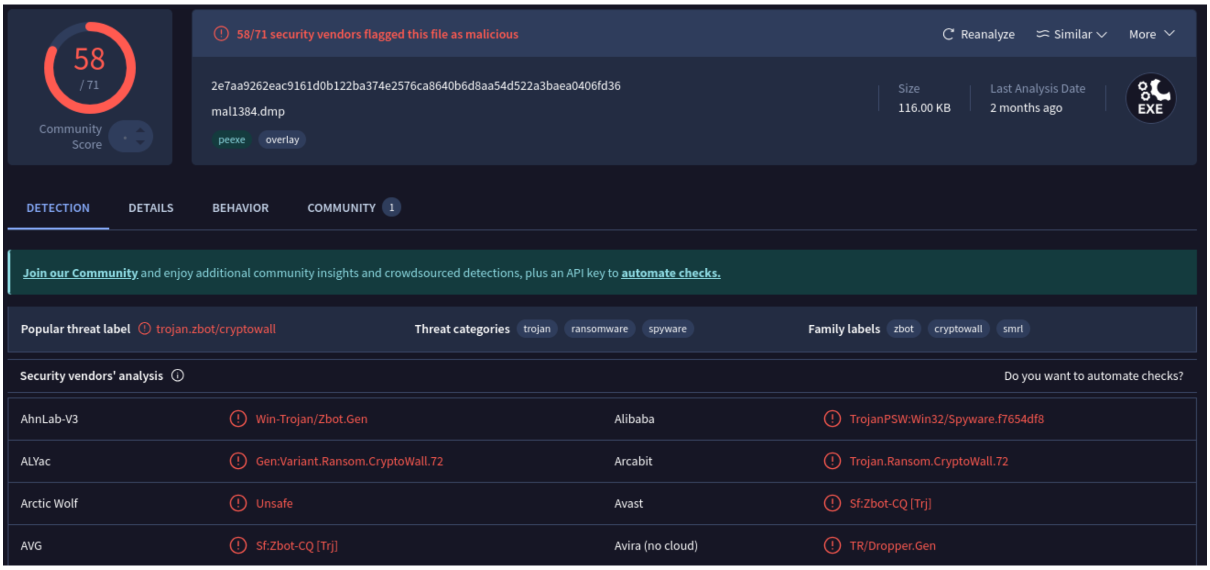
Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Win-Trojan/Zbot.Gen	Alibaba	TrojanPSW:Win32/Spyware.096916ff
AliCloud	Trojan[dropper]:Win/CryptoWall	ALYac	Gen:Variant.Ransom.CryptoWall.72
Antiy-AVL	Trojan/Win32.Agent	Arcabit	Trojan.Ransom.CryptoWall.72
Arctic Wolf	Unsafe	Avast	Sf:Zbot-CQ [Trj]
AVG	Sf:Zbot-CQ [Trj]	Avira (no cloud)	TR/Dropper.Gen

Figure 3. 8 MZ file on PID 1752 VirusTotal Result

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



58/71 security vendors flagged this file as malicious

2e7aa9262eac9161d0b122ba374e2576ca8640b6d8aa54d522a3baea0406fd36

mal1384.dmp

peexe overlay

Size 116.00 KB

Last Analysis Date 2 months ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.zbot/cryptowall

Threat categories trojan ransomware spyware

Family labels zbot cryptowall smrl

Security vendors' analysis

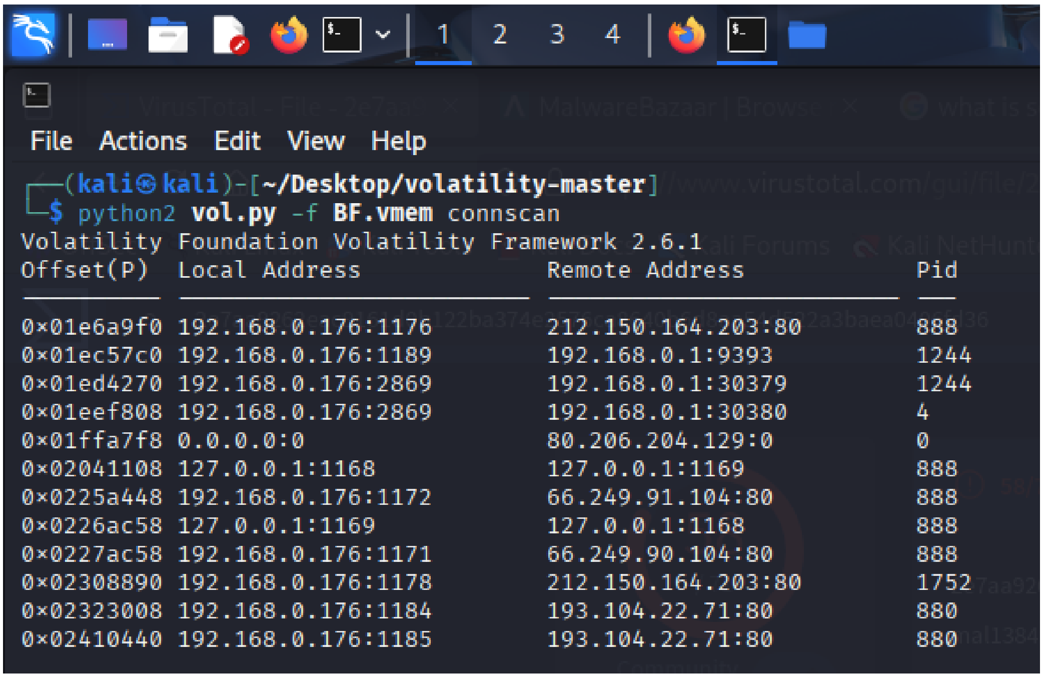
Do you want to automate checks?

AhnLab-V3	Win-Trojan/Zbot.Gen	Alibaba	TrojanPSW:Win32/Spyware.f7654df8
ALYac	Gen:Variant.Ransom.CryptoWall.72	Arcabit	Trojan.Ransom.CryptoWall.72
Arctic Wolf	Unsafe	Avast	Sf:Zbot-CQ [Trj]
AVG	Sf:Zbot-CQ [Trj]	Avira (no cloud)	TR/Dropper.Gen

Figure 3. 9 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.



```
┌──(kali㉿kali)-[~/Desktop/volatility-master]
└─$ python2 vol.py -f BF.vmem connscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
0x01e6a9f0 192.168.0.176:1176        212.150.164.203:80        888
0x01ec57c0 192.168.0.176:1189        192.168.0.1:9393          1244
0x01ed4270 192.168.0.176:2869        192.168.0.1:30379         1244
0x01eef808 192.168.0.176:2869        192.168.0.1:30380         4
0x01ffa7f8 0.0.0.0:0                 80.206.204.129:0          0
0x02041108 127.0.0.1:1168            127.0.0.1:1169            888
0x0225a448 192.168.0.176:1172        66.249.91.104:80          888
0x0226ac58 127.0.0.1:1169            127.0.0.1:1168            888
0x0227ac58 192.168.0.176:1171        66.249.90.104:80          888
0x02308890 192.168.0.176:1178        212.150.164.203:80        1752
0x02323008 192.168.0.176:1184        193.104.22.71:80          880
0x02410440 192.168.0.176:1185        193.104.22.71:80          880
```

Figure 3. 10 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 3. 1 List of active connections and actions

Using **Figure 3. 10** 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=.

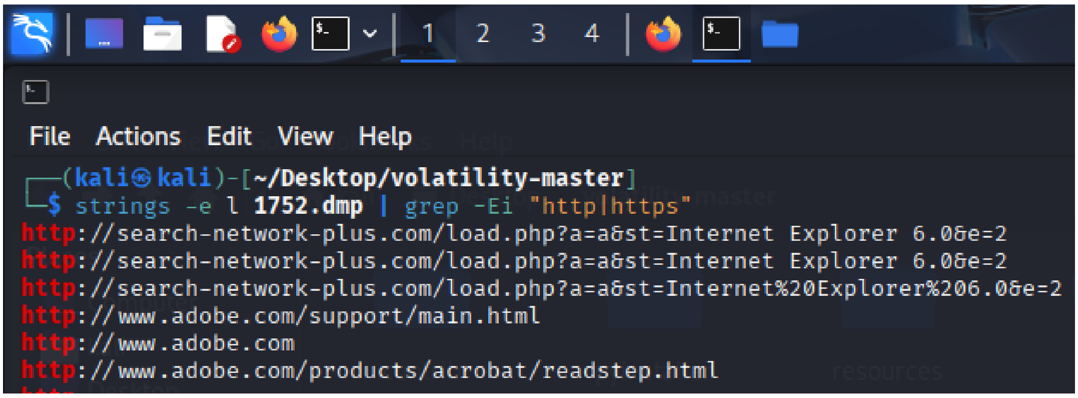
Files were saved as 888.dmp, 1752.dmp and 880.dmp.

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"



```
┌──(kali㉿kali)-[~/Desktop/volatility-master]
└─$ strings -e l 1752.dmp | grep -Ei "http|https"
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
http://www.adobe.com/support/main.html
http://www.adobe.com
http://www.adobe.com/products/acrobat/readstep.html
```

Figure 3. 11 Snippet of String Result on PID 1752 showing suspicious URL

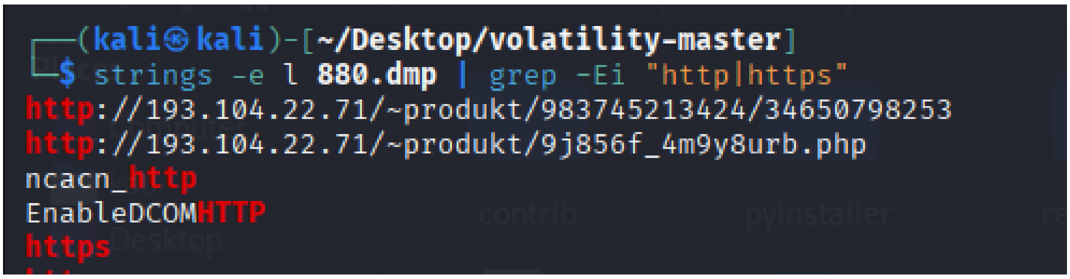
The search on PID 1752 using the “http|https” revealed a suspicious URL that was visited thrice by process 1752.

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"



```
┌──(kali㉿kali)-[~/Desktop/volatility-master]
└─$ strings -e l 880.dmp | grep -Ei "http|https"
http://193.104.22.71/~produkt/983745213424/34650798253
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php
ncacn_http
EnableDCOMHTTP
https
```

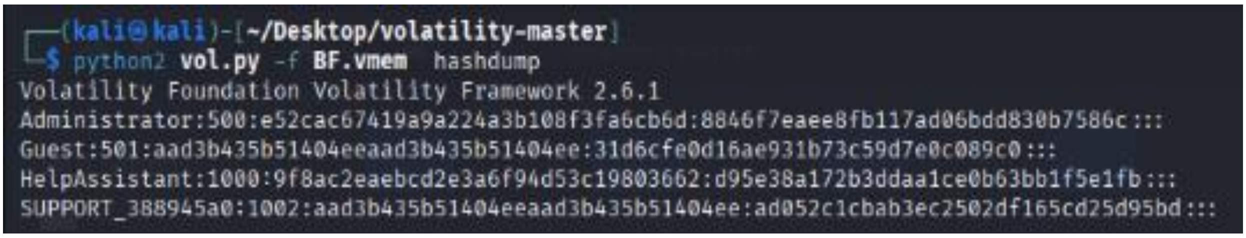
Figure 3. 12 Snippet of String Result on PID 1752 showing suspicious URL

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.



```
┌──(kali㉿kali)-[~/Desktop/volatility-master]
└─$ python2 vol.py -f BF.vmem  hashdump
Volatility Foundation Volatility Framework 2.6.1
Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:9f8ac2eaebcd2e3a6f94d53c19803662:d95e38a172b3ddaa1ce0b63bb1f5e1fb:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:ad052c1cbab3ec2502df165cd25d95bd:::
```

Figure 3. 13 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. LIST THE PROCESSES THAT WERE RUNNING ON THE VICTIM'S MACHINE. WHICH PROCESS WAS MOST LIKELY RESPONSIBLE FOR THE INITIAL EXPLOIT?

Running Processes:

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. LIST THE SOCKETS THAT 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 3. 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:

The strings analysis revealed the following URL:

- **Associated with AcroRd32.exe (PID 1752):**

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

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.

- **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

TABLE OF TABLES