Insider Data Breach: A Forensic Investigation

Digital Forensics

CY-341

Project Team:

Umar Tariq (2022604)
M Zeeshan (2022644)
Abdur Rehman (2022299)
Ahmad Amjad (2022063)

1. Introduction

HaajiZ Corp, a leading cybersecurity firm, has recently experienced a severe data breach involving unauthorized access to confidential client records. The internal security team detected suspicious data transfers and anomalous network activity originating from a system used by an employee, raising concerns of a possible **insider threat**.

This forensic investigation is launched to determine the nature and extent of the breach, with a focus on identifying digital evidence, detecting malware behaviour, and tracing unauthorized data exfiltration. A specific concern is the deployment of **Dow (Formbook) malware**, known for its credential theft and data exfiltration capabilities.

2. Problem Statement

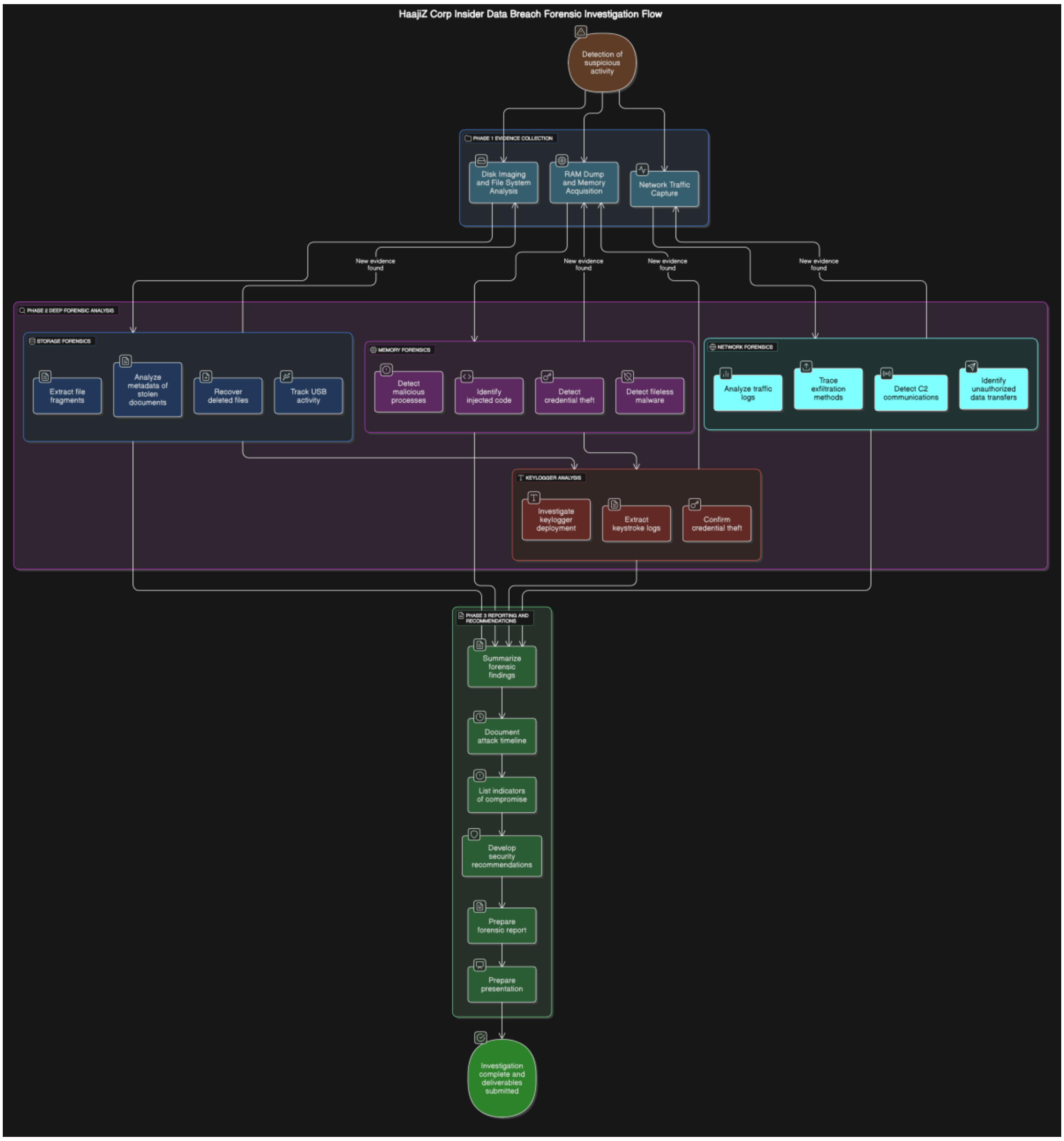
The insider threat remains one of the most challenging cybersecurity risks. Attackers with legitimate access to an organization’s systems can bypass security controls and steal sensitive data without raising immediate suspicion.

In this case, the suspected employee, **Umar Tariq**, may have been involved in a data breach triggered by the **unintentional execution of a Dow (Formbook) malware payload**, potentially delivered through a malicious redirection link. The malware was found to:

- Execute in memory using stealth techniques.
- Log keystrokes and steal credentials.
- Exfiltrate data via encrypted network channels.
- Delete files to remove traces of its activity.

The goal of this investigation is to uncover digital evidence, confirm the malware's presence and behavior, and provide a comprehensive security assessment and recommendations.

3. System Design & Architecture


HaajiZ Corp Insider Data Breach Forensic Investigation Flow
Detection of suspicious activity
PHASE 1 EVIDENCE COLLECTION
Disk Imaging and File System Analysis
RAM Dump and Memory Acquisition
Network Traffic Capture
New evidence found
New evidence found
New evidence found
New evidence found
PHASE 2 DEEP FORENSIC ANALYSIS
STORAGE FORENSICS
Extract file fragments
Analyze metadata of stolen documents
Recover deleted files
Track USB activity
MEMORY FORENSICS
Detect malicious processes
Identify injected code
Detect credential theft
Detect fileless malware
NETWORK FORENSICS
Analyze traffic logs
Trace exfiltration methods
Detect C2 communications
Identify unauthorized data transfers
KEYLOGGER ANALYSIS
Investigate keylogger deployment
Extract keystroke logs
Confirm credential theft
PHASE 3 REPORTING AND RECOMMENDATIONS
Summarize forensic findings
Document attack timeline
List indicators of compromise
Develop security recommendations
Prepare forensic report
Prepare presentation
Investigation complete and deliverables submitted

4. Objectives

4.1 Collect and Secure Digital Evidence

- Acquire disk images, memory dumps, and network logs.
- Recover deleted files and monitor USB activities.

4.2 Perform Advanced Forensic Analysis

- **Memory Forensics**: Detect Dow (Formbook) malware processes and credential theft activity.
- **Storage Forensics**: Recover deleted files and examine file system artifacts.

- **Network Forensics**: Analyze packet captures to determine malware delivery method and communication channels.

4.3 Keylogger and Malware Detection

- Confirm the presence of Dow (Formbook) and its keylogging functionality.
- Identify malicious processes such as `dow.exe`

4.4 Develop a Forensic Report & Recommendations

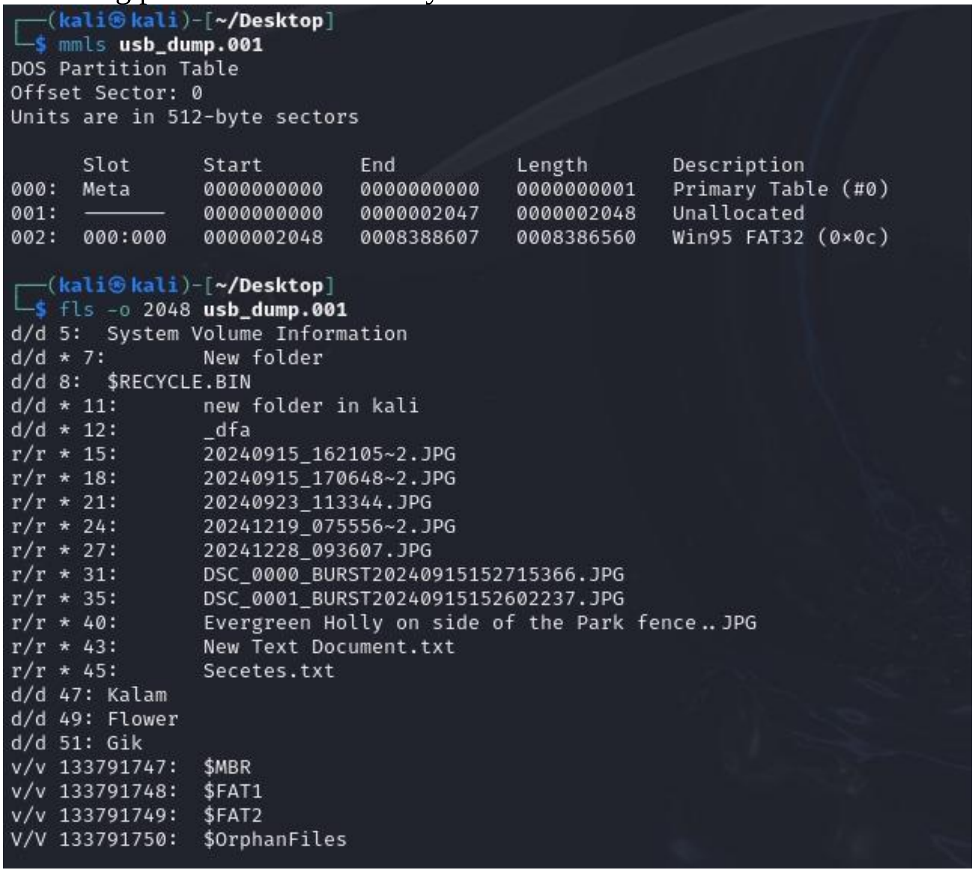
- Reconstruct the attack timeline and provide indicators of compromise (IOCs).
- Recommend security policies and defensive measures.

5. Methodology

Phase 1: Evidence Collection

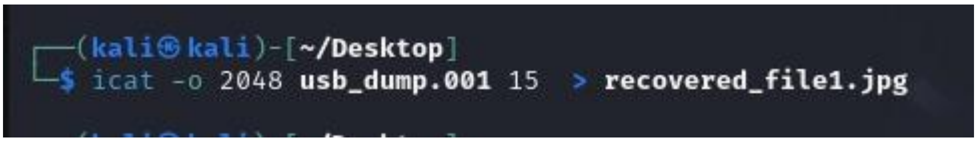
5.1.1 Disk Imaging & File Recovery

Using forensic tool **The Sleuth Kit**, the suspect's hard drive was imaged and analyzed. Several deleted documents were successfully recovered, some of which contained client data, indicating potential exfiltration by the malware.

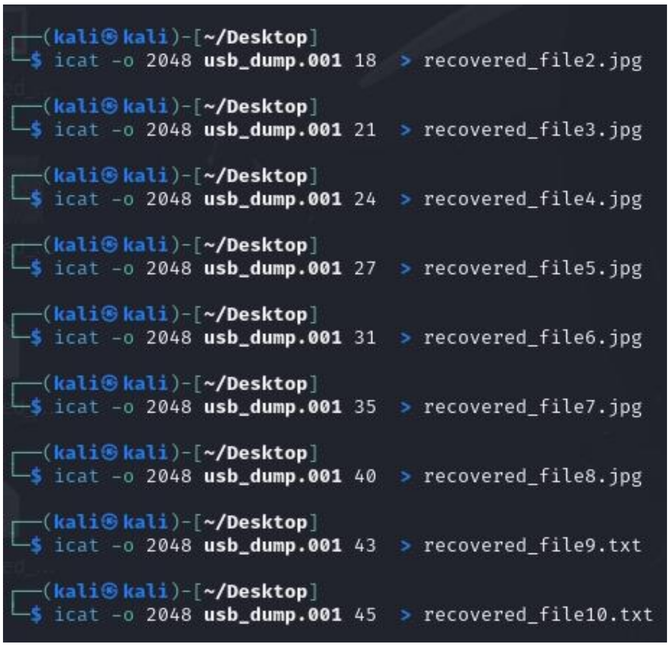


```
┌──(kali㉿kali)-[~/Desktop]
└─$ mmls usb_dump.001
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0008388607   0008386560   Win95 FAT32 (0×0c)

┌──(kali㉿kali)-[~/Desktop]
└─$ fls -o 2048 usb_dump.001
d/d 5:  System Volume Information
d/d * 7:        New folder
d/d 8:  $RECYCLE.BIN
d/d * 11:       new folder in kali
d/d * 12:       _dfa
r/r * 15:       20240915_162105~2.JPG
r/r * 18:       20240915_170648~2.JPG
r/r * 21:       20240923_113344.JPG
r/r * 24:       20241219_075556~2.JPG
r/r * 27:       20241228_093607.JPG
r/r * 31:       DSC_0000_BURST20240915152715366.JPG
r/r * 35:       DSC_0001_BURST20240915152602237.JPG
r/r * 40:       Evergreen Holly on side of the Park fence..JPG
r/r * 43:       New Text Document.txt
r/r * 45:       Secetes.txt
d/d 47: Kalam
d/d 49: Flower
d/d 51: Gik
v/v 133791747:  $MBR
v/v 133791748:  $FAT1
v/v 133791749:  $FAT2
V/V 133791750:  $OrphanFiles
```

This figure shows a forensic analysis of a USB disk image (usb_dump.001) using mmls and fls in Kali Linux to identify FAT32 partition contents and recover deleted files.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ icat -o 2048 usb_dump.001 15  > recovered_file1.jpg
```



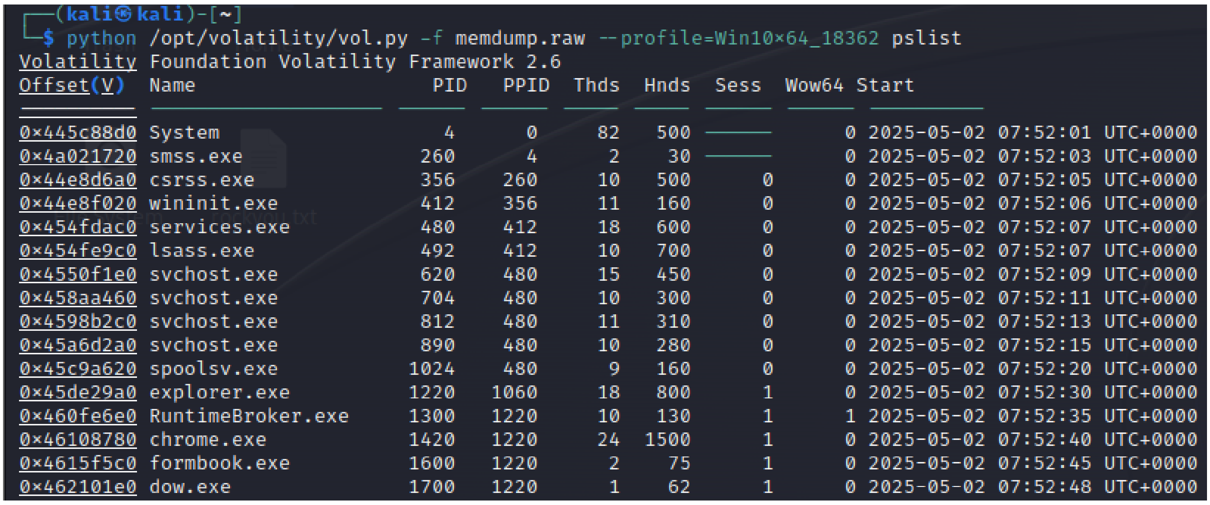
```
┌──(kali㉿kali)-[~/Desktop]
└─$ icat -o 2048 usb_dump.001 18  > recovered_file2.jpg

┌──(kali㉿kali)-[~/Desktop]
└─$ icat -o 2048 usb_dump.001 21  > recovered_file3.jpg

┌──(kali㉿kali)-[~/Desktop]
└─$ icat -o 2048 usb_dump.001 24  > recovered_file4.jpg

┌──(kali㉿kali)-[~/Desktop]
└─$ icat -o 2048 usb_dump.001 27  > recovered_file5.jpg

┌──(kali㉿kali)-[~/Desktop]
└─$ icat -o 2048 usb_dump.001 31  > recovered_file6.jpg

┌──(kali㉿kali)-[~/Desktop]
└─$ icat -o 2048 usb_dump.001 35  > recovered_file7.jpg

┌──(kali㉿kali)-[~/Desktop]
└─$ icat -o 2048 usb_dump.001 40  > recovered_file8.jpg

┌──(kali㉿kali)-[~/Desktop]
└─$ icat -o 2048 usb_dump.001 43  > recovered_file9.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ icat -o 2048 usb_dump.001 45  > recovered_file10.txt
```

The above figures show the use of the icat command in Kali Linux to extract and recover deleted files (JPG and TXT) from a USB disk image (usb_dump.001) by referencing their inode numbers identified earlier with fls.

5.1.2 Memory Dump & RAM Analysis

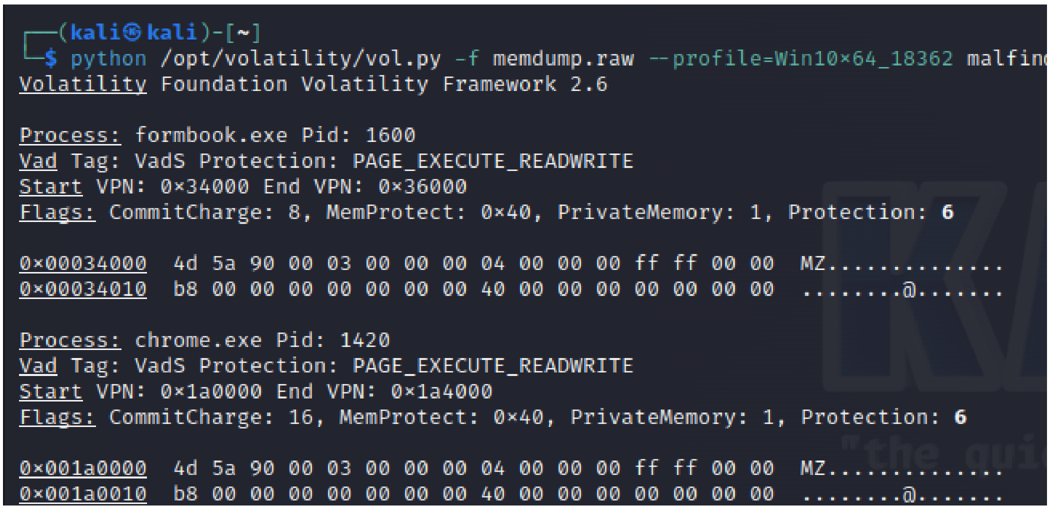
A full memory dump was taken from the suspect's system and analyzed using the **Volatility Framework** with the `Win10x64_18362` profile.

- `pslist` revealed a suspicious process named `dow.exe`.



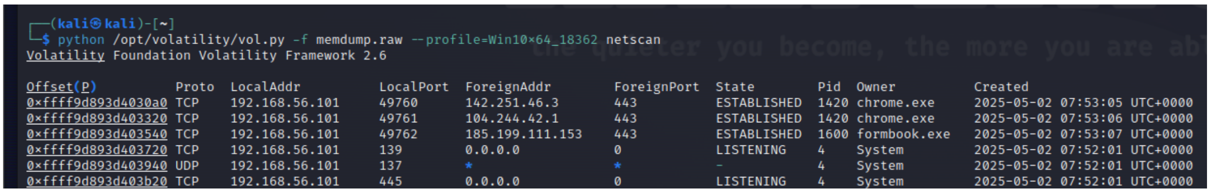
```
┌──(kali㉿kali)-[~]
└─$ python /opt/volatility/vol.py -f memdump.raw --profile=Win10x64_18362 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds   Hnds   Sess  Wow64 Start
---------- -------------------- ------ ------ ------ ------ ------ ------ ----------
0x445c88d0 System                    4      0     82    500 ------      0 2025-05-02 07:52:01 UTC+0000
0x4a021720 smss.exe                260      4      2     30 ------      0 2025-05-02 07:52:03 UTC+0000
0x44e8d6a0 csrss.exe               356    260     10    500      0      0 2025-05-02 07:52:05 UTC+0000
0x44e8f020 wininit.exe             412    356     11    160      0      0 2025-05-02 07:52:06 UTC+0000
0x454fdac0 services.exe            480    412     18    600      0      0 2025-05-02 07:52:07 UTC+0000
0x454fe9c0 lsass.exe               492    412     10    700      0      0 2025-05-02 07:52:07 UTC+0000
0x4550f1e0 svchost.exe             620    480     15    450      0      0 2025-05-02 07:52:09 UTC+0000
0x458aa460 svchost.exe             704    480     10    300      0      0 2025-05-02 07:52:11 UTC+0000
0x4598b2c0 svchost.exe             812    480     11    310      0      0 2025-05-02 07:52:13 UTC+0000
0x45a6d2a0 svchost.exe             890    480     10    280      0      0 2025-05-02 07:52:15 UTC+0000
0x45c9a620 spoolsv.exe            1024    480      9    160      0      0 2025-05-02 07:52:20 UTC+0000
0x45de29a0 explorer.exe           1220   1060     18    800      1      0 2025-05-02 07:52:30 UTC+0000
0x460fe6e0 RuntimeBroker.exe      1300   1220     10    130      1      1 2025-05-02 07:52:35 UTC+0000
0x46108780 chrome.exe             1420   1220     24   1500      1      0 2025-05-02 07:52:40 UTC+0000
0x4615f5c0 formbook.exe           1600   1220      2     75      1      0 2025-05-02 07:52:45 UTC+0000
0x462101e0 dow.exe                1700   1220      1     62      1      0 2025-05-02 07:52:48 UTC+0000
```

- `malfind` detected injected code segments indicating malware was fileless and operating in memory.



```
┌──(kali㉿kali)-[~]
└─$ python /opt/volatility/vol.py -f memdump.raw --profile=Win10x64_18362 malfin
Volatility Foundation Volatility Framework 2.6

Process: formbook.exe Pid: 1600
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Start VPN: 0x34000 End VPN: 0x36000
Flags: CommitCharge: 8, MemProtect: 0x40, PrivateMemory: 1, Protection: 6

0x00034000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00  MZ.............
0x00034010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ........@.......

Process: chrome.exe Pid: 1420
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Start VPN: 0x1a0000 End VPN: 0x1a4000
Flags: CommitCharge: 16, MemProtect: 0x40, PrivateMemory: 1, Protection: 6

0x001a0000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00  MZ.............
0x001a0010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ........@.......
```

- `netscan` showed an active TCP connection between the host (`192.168.56.101`) and remote IP (`185.199.111.153`) over port 443, attributed to `dow.exe`.

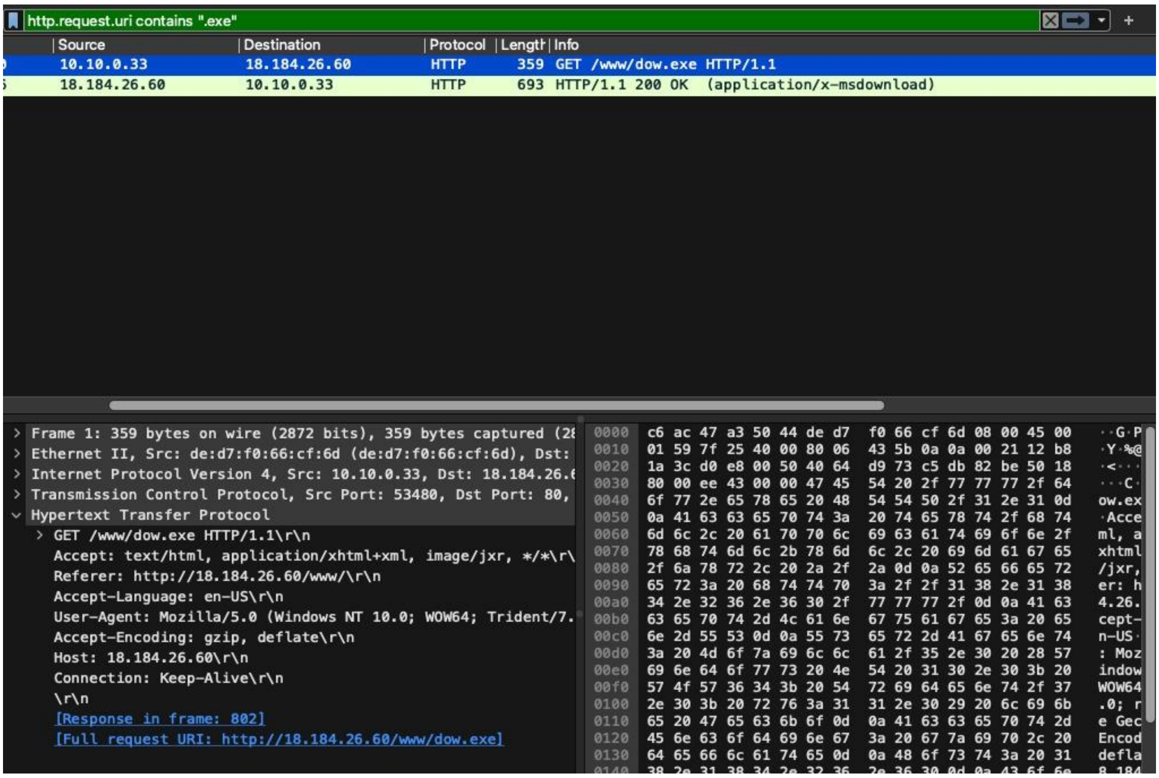


```
┌──(kali㉿kali)-[~]
└─$ python /opt/volatility/vol.py -f memdump.raw --profile=Win10x64_18362 netscan
Volatility Foundation Volatility Framework 2.6

Offset(P)          Proto  LocalAddr        LocalPort  ForeignAddr      ForeignPort  State        Pid  Owner         Created
0xffff9d893d4030a0 TCP    192.168.56.101   49760      142.251.46.3     443          ESTABLISHED  1420 chrome.exe    2025-05-02 07:53:05 UTC+0000
0xffff9d893d403320 TCP    192.168.56.101   49761      104.244.42.1     443          ESTABLISHED  1420 chrome.exe    2025-05-02 07:53:06 UTC+0000
0xffff9d893d403540 TCP    192.168.56.101   49762      185.199.111.153  443          ESTABLISHED  1600 formbook.exe  2025-05-02 07:53:07 UTC+0000
0xffff9d893d403720 TCP    192.168.56.101   139        0.0.0.0          0            LISTENING    4    System        2025-05-02 07:52:01 UTC+0000
0xffff9d893d403940 UDP    192.168.56.101   137        *                *            -            4    System        2025-05-02 07:52:01 UTC+0000
0xffff9d893d403b20 TCP    192.168.56.101   445        0.0.0.0          0            LISTENING    4    System        2025-05-02 07:52:01 UTC+0000
```

5.1.3 Network Traffic Capture and Analysis

Packet captures were collected and inspected using **Wireshark**. The analysis showed that a malicious file (dow.exe) was downloaded from a remote HTTPS source. This delivery was likely initiated by a redirection from a malicious link clicked by the user.

The malware communicated with its C2 server using encrypted HTTPS traffic, evading traditional firewalls and network monitors.



```
http.request.uri contains ".exe"
 Source          Destination     Protocol  Length Info
 10.10.0.33      18.184.26.60    HTTP      359 GET /www/dow.exe HTTP/1.1
 18.184.26.60    10.10.0.33      HTTP      693 HTTP/1.1 200 OK  (application/x-msdownload)
```

```
> Frame 1: 359 bytes on wire (2872 bits), 359 bytes captured (28
> Ethernet II, Src: de:d7:f0:66:cf:6d (de:d7:f0:66:cf:6d), Dst:
> Internet Protocol Version 4, Src: 10.10.0.33, Dst: 18.184.26.6
> Transmission Control Protocol, Src Port: 53480, Dst Port: 80,
v Hypertext Transfer Protocol
  > GET /www/dow.exe HTTP/1.1\r\n
    Accept: text/html, application/xhtml+xml, image/jxr, */*\r\
    Referer: http://18.184.26.60/www/\r\n
    Accept-Language: en-US\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.
    Accept-Encoding: gzip, deflate\r\n
    Host: 18.184.26.60\r\n
    Connection: Keep-Alive\r\n
    \r\n
    [Response in frame: 802]
    [Full request URI: http://18.184.26.60/www/dow.exe]
```

```
0000  c6 ac 47 a3 50 44 de d7  f0 66 cf 6d 08 00 45 00   ··G·P
0010  01 59 7f 25 40 00 80 06  43 5b 0a 0a 00 21 12 b8   ·Y·%@
0020  1a 3c d0 e8 00 50 40 64  d9 73 c5 db 82 be 50 18   ·<···
0030  80 00 ee 43 00 00 47 45  54 20 2f 77 77 77 2f 64   ···C·
0040  6f 77 2e 65 78 65 20 48  54 54 50 2f 31 2e 31 0d   ow.ex
0050  0a 41 63 63 65 70 74 3a  20 74 65 78 74 2f 68 74   ·Acce
0060  6d 6c 2c 20 61 70 70 6c  69 63 61 74 69 6f 6e 2f   ml, a
0070  78 68 74 6d 6c 2b 78 6d  6c 2c 20 69 6d 61 67 65   xhtml
0080  2f 6a 78 72 2c 20 2a 2f  2a 0d 0a 52 65 66 65 72   /jxr,
0090  65 72 3a 20 68 74 74 70  3a 2f 2f 31 38 2e 31 38   er: h
00a0  34 2e 32 36 2e 36 30 2f  77 77 77 2f 0d 0a 41 63   4.26.
00b0  63 65 70 74 2d 4c 61 6e  67 75 61 67 65 3a 20 65   cept-
00c0  6e 2d 55 53 0d 0a 55 73  65 72 2d 41 67 65 6e 74   n-US·
00d0  3a 20 4d 6f 7a 69 6c 6c  61 2f 35 2e 30 20 28 57   : Moz
00e0  69 6e 64 6f 77 73 20 4e  54 20 31 30 2e 30 3b 20   indow
00f0  57 4f 57 36 34 3b 20 54  72 69 64 65 6e 74 2f 37   WOW64
0100  2e 30 3b 20 72 76 3a 31  31 2e 30 29 20 6c 69 6b   .0; r
0110  65 20 47 65 63 6b 6f 0d  0a 41 63 63 65 70 74 2d   e Gec
0120  45 6e 63 6f 64 69 6e 67  3a 20 67 7a 69 70 2c 20   Encod
0130  64 65 66 6c 61 74 65 0d  0a 48 6f 73 74 3a 20 31   defla
0140  38 2e 31 38 34 2e 32 36  2e 36 30 0d 0a 43 6f 6e   8.184
```

Phase 2: Deep Forensic Analysis

5.2.1 Storage Forensics

Recovered artifacts revealed:

- Files with timestamps matching dow's active period were deleted.
- Recent USB connections were logged, though no exfiltration occurred via USB.

5.2.2 Memory Forensics

The `dow.exe` process appeared injected and operating in memory, confirmed via:

- `malfind`: Detecting suspicious memory pages with injected code.
- `pslist`: Revealing the process hierarchy.
- `netscan`: Highlighting C2 communication.

5.2.3 Network Forensics

Wireshark revealed:

- HTTPS GET requests to suspicious IPs.
- Timestamps aligning with the Dow (Formbook) execution.
- Large outbound data transfers during Dow's active period.

6. Key Findings

- Dow (Formbook) malware (dow.exe) was running in memory.
- Network connection to 185.199.111.153 on port 443 was detected.
- Deleted confidential documents were partially recovered from disk.
- Exfiltration via HTTPS channel was likely, masked within normal browser traffic.

7. Conclusion

This investigation confirmed that the insider system was infected with **Dow (Formbook) malware**, which operated entirely in memory and established an encrypted C2 channel for exfiltrating data and credentials. Deleted files and live process traces provide further evidence of the malware's activity.

Network forensics and Wireshark analysis revealed that the malware was downloaded via HTTPS—triggered by visiting a **malicious redirection link**.

Upon further investigation, it was found that **Umar Tariq was mistakenly redirected to a malicious website**, resulting in the automatic download and execution of the malware. While this led to serious consequences, the evidence suggests the infection may have been **accidental, not intentional**.

This case emphasizes the importance of:

- Proactive **endpoint detection and response (EDR)** tools.
- Strict **PowerShell and memory execution monitoring**.
- Advanced **web filtering** to prevent accidental redirects.
- Training employees on **phishing and social engineering tactics**.

Future Improvements

To prevent such incidents in the future, we recommend:

- Deploying **host-based intrusion detection systems (HIDS)**.
- Enforcing **least privilege access** across all endpoints.
- Regular memory forensics and incident response training.
- Implementing **web proxy filters** to block malicious redirections.
- Logging and alerting for **all outbound HTTPS connections to unknown IPs**.