

# **Lab 10**

## **Log and Malware Analysis**

Mina Salehi

University of Maryland Baltimore County

CYBR 642 Introduction to Digital Forensics

Presented to: Gina Scaldaferri

04/30/2025

## Introduction

In this exercise, students assume the role of a cybersecurity analyst tasked with investigating a potential data breach at Bob's Dry Cleaners. The business, which retains sensitive credit card and contact information for its Platinum Dry Cleaning customers—many of whom are executives—relies on strict data protection. The internal environment consists of segmented networks (192.168.30.0/24 for internal operations, 10.30.30.0/24 for the DMZ, and 172.30.1.0/24 representing the external "Internet".) The organization's security team proactively collects operating system logs from Linux servers, Windows workstations, and Cisco ASA firewalls. These logs are centrally managed using a syslog server (192.168.30.30) running rsyslogd. For this lab, investigators are provided with three primary log files—auth.log, workstations.log, and firewall.log—capturing authentication events, user activities, and network traffic flows during the suspected incident.

The primary objective is to analyze these logs to determine whether unauthorized access occurred and assess the risk of confidential customer data exfiltration. Students are required to substantiate their findings with detailed evidence and articulate the potential impacts of the compromise if data theft is confirmed.

## Managing the Splunk Container

### 1. Log Analysis

Started the lab by opening the auth.log file in Splunk. The brute-force activity begins at 18:56:50 and is targeted at baboon-srv, targeting both root and bob accounts. Many password attempts is a strong indicator that the threat actor is using a password-guessing attack.

```
auth.log
x
file:///home/sansforensics/Desktop/cases/Evidence/Log%20Analysis/auth.log
2011-04-26T18:48:21.75545-06:00 cheetah-srv su[11794]: user=root; rhost=/dev/tty2; ruser=root; command=/bin/su
2011-04-26T18:48:21.748871-06:00 cheetah-srv su[11794]: Successful su for root by root
2011-04-26T18:48:21.749444-06:00 cheetah-srv su[11794]: + /dev/tty2 root:root
2011-04-26T18:48:21.751316-06:00 cheetah-srv su[11794]: pam_unix(su:session): session opened for user root by
user1(uid=0)
2011-04-26T18:56:50-06:00 baboon-srv sshd[6423]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=172.30.1.77 user=root
2011-04-26T18:56:53-06:00 baboon-srv sshd[6423]: Failed password for root from 172.30.1.77 port 60372 ssh2
2011-04-26T18:56:56-06:00 baboon-srv sshd[6423]: last message repeated 2 times
2011-04-26T18:56:56-06:00 baboon-srv sshd[6423]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh
ruser= rhost=172.30.1.77 user=root
2011-04-26T18:56:57-06:00 baboon-srv sshd[6425]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=172.30.1.77 user=root
2011-04-26T18:56:59-06:00 baboon-srv sshd[6425]: Failed password for root from 172.30.1.77 port 60373 ssh2
2011-04-26T18:57:02-06:00 baboon-srv sshd[6425]: last message repeated 2 times
2011-04-26T18:57:02-06:00 baboon-srv sshd[6425]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh
ruser= rhost=172.30.1.77 user=root
2011-04-26T19:00:56-06:00 baboon-srv sshd[6503]: Failed password for root from 172.30.1.77 port 49185 ssh2
2011-04-26T19:00:57-06:00 baboon-srv sshd[6505]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=172.30.1.77 user=bob
2011-04-26T19:00:59-06:00 baboon-srv sshd[6505]: Failed password for bob from 172.30.1.77 port 49186 ssh2
2011-04-26T19:01:03-06:00 baboon-srv sshd[6505]: last message repeated 2 times
2011-04-26T19:01:03-06:00 baboon-srv sshd[6505]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh
ruser= rhost=172.30.1.77 user=bob
2011-04-26T19:01:04-06:00 baboon-srv sshd[6507]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0
```

At 19:05:10, an unauthorized actor using the external IP address 172.30.1.77 successfully compromised Bob's user account. The attacker initially attempted to escalate privileges using the sudo command to modify the auth.log file via the vi text editor. Although the first attempt to gain elevated access failed, subsequent attempts were successful, allowing the adversary to execute privileged commands. Evidence indicates that the attacker edited the authentication logs on baboon-srv to obscure their presence and activity. Additionally, the threat actor utilized the tcpdump utility, suggesting network monitoring or data collection, and later installed Nmap, a reconnaissance tool commonly used to perform port scanning and network enumeration (Scarfone & Mell, 2007).

```
2011-04-26T19:05:10-06:00 baboon-srv sudo: pam_unix(sudo:auth): authentication failure; logname=bob uid=0
euid=0 tty=/dev/pts/0 ruser= rhost= user=bob
2011-04-26T19:05:18-06:00 baboon-srv sudo:      bob : TTY=pts/0 ; PWD=/home/bob ; USER=root ;
COMMAND=/usr/bin/vi /var/log/auth.log
2011-04-26T19:05:34-06:00 baboon-srv sudo:      bob : TTY=pts/0 ; PWD=/home/bob ; USER=root ;
COMMAND=/usr/sbin/tcpdump -nni eth0
2011-04-26T19:07:03-06:00 baboon-srv sudo:      bob : TTY=pts/0 ; PWD=/home/bob ; USER=root ;
COMMAND=/usr/bin/apt-get update
2011-04-26T19:07:15-06:00 baboon-srv sudo:      bob : TTY=pts/0 ; PWD=/home/bob ; USER=root ;
COMMAND=/usr/bin/apt-get install nmap
2011-04-26T19:14:53-06:00 baboon-srv sshd[6632]: pam_unix(sshd:session): session closed for user bob
```

## **2. Dynamic Malware Analysis**

In the ever-evolving threat landscape, cyber adversaries continue to leverage advanced tactics such as social engineering and client-side exploits to infiltrate high-value targets. This lab investigates a fictional dynamic malware analysis scenario involving a targeted spear phishing attack against a lead developer at SaucyCorp, a company known for its proprietary "Secret Sauce" formula. The attacker, Ann Dercover, uses reconnaissance and known vulnerabilities in Internet Explorer to deliver a malicious payload via an enticing phishing email.

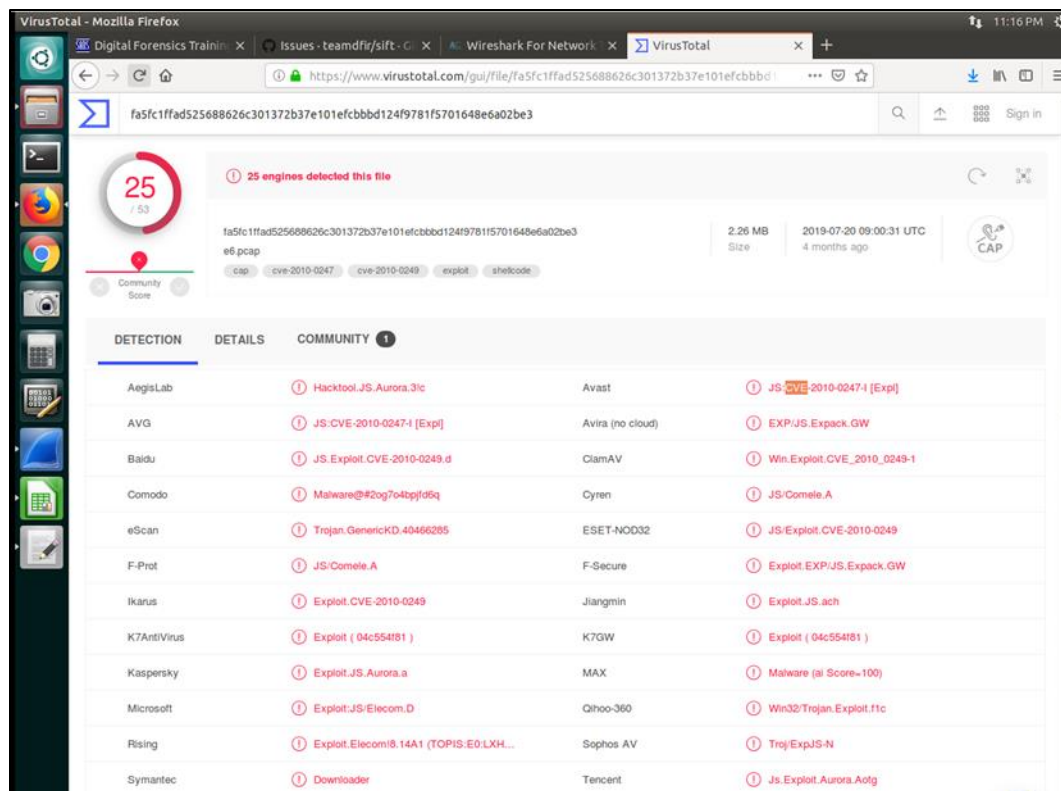
The targeted user, Vick Timmes, unknowingly activates the exploit while accessing a link promising career-enhancing recipe tips. As a result, malicious activity is triggered on his system (10.10.10.70), potentially allowing remote access to SaucyCorp's internal resources.

Fortunately, due to his awareness of cyber risks, Vick had implemented packet capture logging on his home network. Upon noticing unusual traffic behavior, he provides these network captures to investigators.

The primary objective of this dynamic malware analysis lab is to examine the packet capture files to identify the presence and behavior of malware, assess the technique used for initial infection, and evaluate the scope of compromise.

### **Analysis**

The evidence-malware.pcap file captures network traffic originating from Vick Timmes's home network. Upon uploading this file to VirusTotal, the analysis results indicated the presence of malware-associated activity.

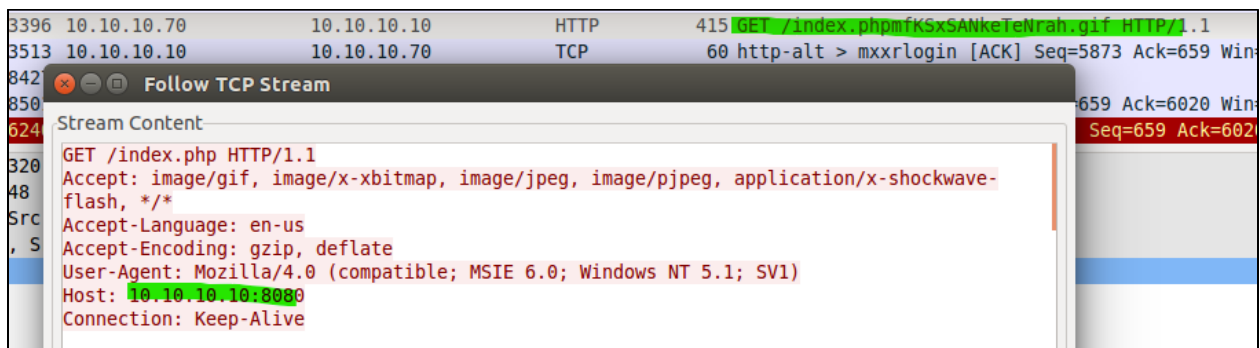


The network traffic captured in the evidence-malware.pcap file revealed that the compromise originated from IP address 10.10.10.10 communicating over TCP port 8080. Immediately following the receipt of a .gif file, a sequence of TCP packets was observed between 10.10.10.10 (external attacker) and 10.10.10.70 (Vick Timmes's system). The communication included the typical pattern of a TCP three-way handshake, evidenced by the exchange of SYN, SYN-ACK, and ACK packets.

This handshake is a foundational mechanism in the Transmission Control Protocol (TCP), used to establish a reliable connection between a client and a server. The handshake process is as follows:

1. SYN (synchronize): The client initiates the connection by sending a SYN packet to the server, indicating a request to start communication and proposing an initial sequence number.
2. SYN-ACK (synchronize-acknowledge): The server acknowledges the request and replies with a SYN-ACK packet, confirming the receipt of the client's SYN and sending its own sequence number.
3. ACK (acknowledge): The client sends an ACK packet to acknowledge receipt of the server's SYN-ACK, thereby completing the handshake and establishing a full-duplex connection.

This three-step exchange ensures that both endpoints agree on initial sequence numbers and are prepared to transmit data. In this scenario, the immediate handshake activity following the suspicious file delivery indicates that the attacker used the image as a vehicle for a client-side exploit, which, once triggered, opened a communication channel with the attacker's host via port 8080—commonly used for web or proxy traffic but also frequently leveraged in command-and-control (C2) communications (Scarfone & Mell, 2007).



The image shows a Wireshark packet capture. The top pane displays a list of packets. Packet 3396 is an HTTP GET request from 10.10.10.70 to 10.10.10.10. Packet 3513 is a TCP ACK from 10.10.10.10 to 10.10.10.70. Packet 842 is an HTTP GET request from 10.10.10.10 to 10.10.10.70. Packet 850 is a TCP ACK from 10.10.10.70 to 10.10.10.10. Packet 624 is an HTTP GET request from 10.10.10.10 to 10.10.10.70. The bottom pane shows the 'Follow TCP Stream' window for the connection between 10.10.10.10 and 10.10.10.70. The stream content shows an HTTP GET request for /index.php HTTP/1.1, followed by various headers including Accept, Accept-Language, Accept-Encoding, User-Agent, Host, and Connection.

```
3396 10.10.10.70 10.10.10.10 HTTP 415 GET /index.php HTTP/1.1
3513 10.10.10.10 10.10.10.70 TCP 60 http-alt > mxrlogin [ACK] Seq=5873 Ack=659 Win=
842 10.10.10.10 10.10.10.70 HTTP 415 GET /index.php HTTP/1.1
850 10.10.10.70 10.10.10.10 TCP 60 http-alt > mxrlogin [ACK] Seq=5873 Ack=659 Win=
624 10.10.10.10 10.10.10.70 HTTP 415 GET /index.php HTTP/1.1
Stream Content
GET /index.php HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-
flash, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: 10.10.10.10:8080
Connection: Keep-Alive
```

|      |            |                 |             |             |      |  |
|------|------------|-----------------|-------------|-------------|------|--|
| 1    | 2010-04-28 | 23:39:59.311284 | 10.10.10.70 | 10.10.10.10 | HTTP | 351 GET /index.php HTTP/1.1                                    |
| 2    | 2010-04-28 | 23:39:59.311382 | 10.10.10.10 | 10.10.10.70 | TCP  | 60 http-alt > mxrlogin [ACK] Seq=1 Ack=298 Win=8576 Len=0      |
| 3    | 2010-04-28 | 23:39:59.656689 | 10.10.10.10 | 10.10.10.70 | TCP  | 1514 [TCP segment of a reassembled PDU]                        |
| 4    | 2010-04-28 | 23:39:59.656768 | 10.10.10.10 | 10.10.10.70 | TCP  | 1514 [TCP segment of a reassembled PDU]                        |
| 5    | 2010-04-28 | 23:39:59.656892 | 10.10.10.10 | 10.10.10.70 | TCP  | 1514 [TCP segment of a reassembled PDU]                        |
| 6    | 2010-04-28 | 23:39:59.657013 | 10.10.10.10 | 10.10.10.70 | TCP  | 1514 [TCP segment of a reassembled PDU]                        |
| 7    | 2010-04-28 | 23:39:59.657108 | 10.10.10.10 | 10.10.10.70 | TCP  | 60 mxrlogin > http-alt [ACK] Seq=298 Ack=5841 Win=65535 Len=0  |
| 8    | 2010-04-28 | 23:39:59.657213 | 10.10.10.10 | 10.10.10.70 | HTTP | 86 HTTP/1.1 200 OK (text/html)                                 |
| 9    | 2010-04-28 | 23:39:59.773396 | 10.10.10.10 | 10.10.10.10 | HTTP | 415 GET /index.phpmfKsXsANkeTetrah.gif HTTP/1.1                |
| 10   | 2010-04-28 | 23:39:59.773513 | 10.10.10.10 | 10.10.10.70 | TCP  | 60 http-alt > mxrlogin [ACK] Seq=5873 Ack=659 Win=9648 Len=0   |
| 11   | 2010-04-28 | 23:39:59.878427 | 10.10.10.10 | 10.10.10.70 | HTTP | 201 HTTP/1.1 200 OK (GIF89a)                                   |
| 12   | 2010-04-28 | 23:40:00.048501 | 10.10.10.70 | 10.10.10.10 | TCP  | 60 mxrlogin > http-alt [ACK] Seq=659 Ack=6020 Win=65356 Len=0  |
| 1343 | 2010-04-28 | 23:41:04.876240 | 10.10.10.70 | 10.10.10.10 | TCP  | 60 mxrlogin > http-alt [RST, ACK] Seq=659 Ack=6020 Win=0 Len=0 |

## Recovering Malware from packet Capture

The header "MZ" indicates that the file is a Windows executable (PE file). The "MZ" signature comes from the initials of Mark Zbikowski, one of the developers of MS-DOS. It is the magic number found at the beginning of all DOS and Windows executables (EXE, DLL, SYS, etc.). The presence of "MZ" and the message "This program cannot be run in DOS mode" strongly suggests that the payload being transferred over the TCP stream is a Windows PE executable, not an image or text file (Sikorski & Honig, 2012.)

|    |            |                 |             |             |     |  |
|----|------------|-----------------|-------------|-------------|-----|--|
| 16 | 2010-04-28 | 23:40:00.837623 | 10.10.10.10 | 10.10.10.70 | TCP | 60 krb524 > nsstp [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=4    |
| 17 | 2010-04-28 | 23:40:00.841061 | 10.10.10.10 | 10.10.10.70 | TCP | 1514 krb524 > nsstp [ACK] Seq=5 Ack=1 Win=5840 Len=1460    |
| 18 | 2010-04-28 | 23:40:00.841140 | 10.10.10.10 | 10.10.10.70 | TCP | 1514 krb524 > nsstp [ACK] Seq=1465 Ack=1 Win=5840 Len=1460 |
| 19 | 2010-04-28 | 23:40:00.841462 | 10.10.10.10 | 10.10.10.10 | TCP | 60 nsstp > krb524 [ACK] Seq=1 Ack=2925 Win=65535 Len=0     |
| 20 | 2010-04-28 | 23:40:00.841541 | 10.10.10.10 | 10.10.10.70 | TCP | 60 nsstp > krb524 [ACK] Seq=1 Ack=2925 Win=65535 Len=0     |
| 21 | 2010-04-28 | 23:40:00.841620 | 10.10.10.10 | 10.10.10.70 | TCP | 60 nsstp > krb524 [ACK] Seq=1 Ack=2925 Win=65535 Len=0     |
| 22 | 2010-04-28 | 23:40:00.841699 | 10.10.10.10 | 10.10.10.70 | TCP | 60 nsstp > krb524 [ACK] Seq=1 Ack=2925 Win=65535 Len=0     |
| 23 | 2010-04-28 | 23:40:00.841778 | 10.10.10.10 | 10.10.10.70 | TCP | 60 nsstp > krb524 [ACK] Seq=1 Ack=2925 Win=65535 Len=0     |
| 24 | 2010-04-28 | 23:40:00.841857 | 10.10.10.10 | 10.10.10.70 | TCP | 60 nsstp > krb524 [ACK] Seq=1 Ack=2925 Win=65535 Len=0     |
| 25 | 2010-04-28 | 23:40:00.841936 | 10.10.10.10 | 10.10.10.70 | TCP | 60 nsstp > krb524 [ACK] Seq=1 Ack=2925 Win=65535 Len=0     |

|   |  |
|---|--|
| Follow TCP Stream                                   |  |
| Stream Content:                                     |  |
| .j.MZ.....[REU.....Wh...P..h...Vh...P.....!..L!This |  |
| program cannot be run in DOS mode.                  |  |



To gain deeper insight into the nature of the suspicious network traffic, the pcap file was uploaded to NetworkTotal.com, an online analysis platform that integrates Suricata and Snort rule sets for malware detection. The analysis results revealed that the capture contains multiple signatures associated with malicious activity, including Meterpreter payloads, Metasploit shellcode, and common Trojan behaviors.

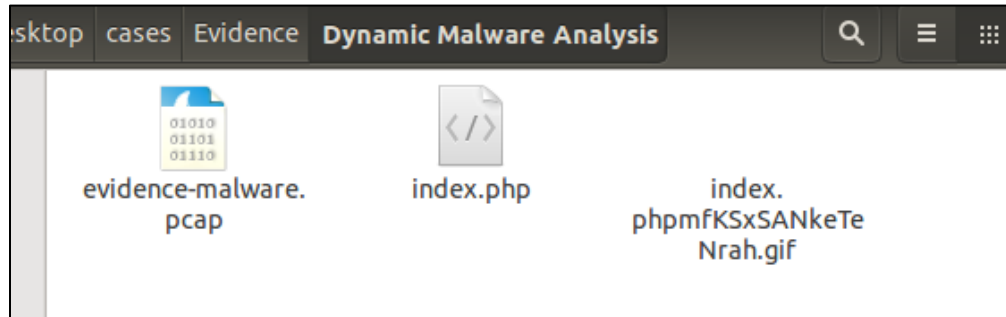
Each detection entry is associated with a Signature ID (SID), which corresponds to specific intrusion detection rules that matched patterns in the network traffic. For instance, SID 1:2812092:1 and 1:2812093:1 indicate that the traffic includes communication typically associated with Win32/Meterpreter sessions, a hallmark of post-exploitation frameworks used for remote control. Other SIDs identified NOOP sleds (1:2101390:6) and Metasploit bind payloads (1:2025644:1), suggesting that this was a sophisticated attack leveraging crafted shellcode to establish unauthorized remote access.



| Events:                         |  |               |   |
|---------------------------------|--|---------------|---|
| Date                            | MD5  | sid           | msg   |
| Thu, 29 Apr 2010 01:41:04 +0000 | 364e75e7214945e7c10504303a2e8cac<br><a href="#">[VT]</a> | [1:2210041:1] | SURICATA STREAM RST recv but no session                                       |
| Thu, 29 Apr 2010 01:40:01 +0000 | 364e75e7214945e7c10504303a2e8cac<br><a href="#">[VT]</a> | [1:2101390:6] | GPL SHELLCODE x86 inc ebx NOOP  |
| Thu, 29 Apr 2010 01:40:01 +0000 | 364e75e7214945e7c10504303a2e8cac<br><a href="#">[VT]</a> | [1:2812092:1] | ETPRO TROJAN Win32/Meterpreter Receiving Meterpreter M1                       |
| Thu, 29 Apr 2010 01:40:00 +0000 | 364e75e7214945e7c10504303a2e8cac<br><a href="#">[VT]</a> | [1:2012088:3] | ET SHELLCODE Possible Call with No Offset TCP Shellcode                       |
| Thu, 29 Apr 2010 01:40:01 +0000 | 364e75e7214945e7c10504303a2e8cac<br><a href="#">[VT]</a> | [1:2025644:1] | ET TROJAN Possible Metasploit Payload Common Construct Bind API (from server) |
| Thu, 29 Apr 2010 01:40:01 +0000 | 364e75e7214945e7c10504303a2e8cac<br><a href="#">[VT]</a> | [1:2812093:1] | ETPRO TROJAN Win32/Meterpreter Receiving Meterpreter M2                       |

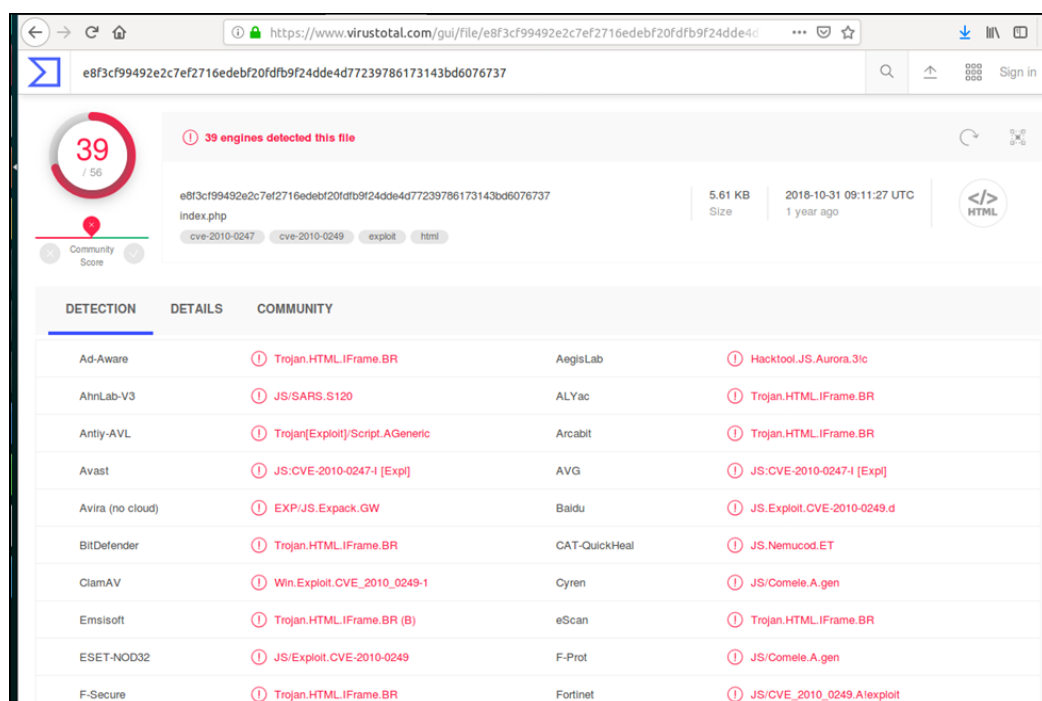
To extract potentially malicious files from a PCAP file, begin by launching Wireshark and opening the capture file. Next, identify suspicious or relevant traffic within the packet list. Once

located, navigate to File > Export Objects, select the appropriate protocol (e.g., HTTP), and save the extracted files for further analysis.



Generate the MD5 hash of the index.php file using the md5sum command to obtain its unique fingerprint. Then, upload the file to VirusTotal.com for analysis. The results confirm that the file is malicious, matching a known malware signature.

```
sansforensics@siftworkstation:~$ md5sum '/home/sansforensics/Desktop/cases/Evidence/Dynamic Malware Analysis/index.php'
03fae1d2f7a0bec010398e58138a493c  /home/sansforensics/Desktop/cases/Evidence/Dynamic Malware Analysis/index.php
sansforensics@siftworkstation:~$
```



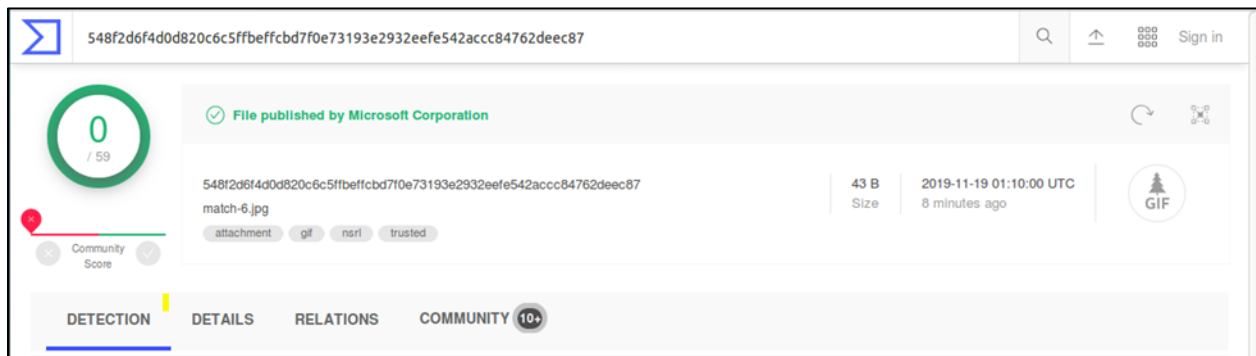


Repeated the same process for the second file index.phpmfKSxSANkeTeNrah.gif

```
sansforensics@siftworkstation:~$ md5sum '/home/sansforensics/Desktop/cases/Evidence/Dynamic Malware Analysis/index.phpmfKSxSANkeTeNrah.gif'
df3e567d6f16d040326c7a0ea29a4f41  /home/sansforensics/Desktop/cases/Evidence/Dynamic Malware Analysis/index.phpmfKSxSANkeTeNrah.gif
sansforensics@siftworkstation:~$
```

## Conclusion

The analysis of the evidence-malware.pcap file revealed a targeted client-side attack in which a disguised executable was delivered over TCP port 8080. The file, falsely presented as a .gif, was confirmed to be malware based on its "MZ" header and VirusTotal results. NetworkTotal analysis further identified multiple intrusion detection signatures, including Meterpreter and shellcode indicators. One extracted file was confirmed as malicious, while another was deemed clean. This case underscores the importance of traffic analysis, file inspection, and external threat intelligence in detecting and validating network-based compromises.



## **Glossary**

**Authentication Log (auth.log):** A record of login attempts, user authentications, and privileged command usage typically gathered from Linux-based systems.

**Cisco ASA Firewall:** A hardware device that filters and manages incoming and outgoing network traffic based on an organization's security policies.

**Data Exfiltration:** The unauthorized transfer of data from a computer or network, often carried out by malicious actors.

**DMZ (Demilitarized Zone):** A physical or logical subnetwork that separates an internal local area network (LAN) from untrusted external networks, such as the internet.

**Log Analysis:** The process of reviewing and interpreting log entries to detect abnormal or suspicious behavior that could indicate a security incident.

**Nmap (Network Mapper):** A free and open-source utility used for network discovery and security auditing, often employed in the reconnaissance phase of cyberattacks.

**Rsyslogd:** An open-source software utility used for forwarding log messages in an IP network.

**sudo:** A command-line program that allows a permitted user to execute a command as the superuser or another user.

**tcpdump:** A network packet analyzer that captures and displays TCP/IP and other packets transmitted or received over a network.

TCP Port 8080: A common port for web proxy and alternative HTTP services, often exploited for backdoor or C2 connections.

Workstation Logs (workstations.log): Files capturing system events, errors, and user activities from Windows desktop computers.

vi: A command-line text editor used in Unix-like systems for editing files.

VirusTotal: An online service that aggregates multiple antivirus engines to analyze files and URLs for potential malware and threats.

Privilege Escalation: The act of exploiting a bug, design flaw, or configuration oversight in an operating system or software to gain elevated access to resources.

## References

Rapid7. (n.d.). *Meterpreter*. Metasploit Documentation. Retrieved April 29, 2025, from <https://docs.metasploit.com/docs/using-metasploit/advanced/meterpreter/meterpreter.html>

Rapid7. (n.d.). *About post-exploitation*. Metasploit Documentation. Retrieved April 29, 2025, from <https://docs.rapid7.com/metasploit/about-post-exploitation>

Scarfone, K., & Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS) (NIST Special Publication 800-94). National Institute of Standards and Technology.  
<https://doi.org/10.6028/NIST.SP.800-94>

Sikorski, M., & Honig, A. (2012). *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press.