EncryptEdge Labs

Cybersecurity Analyst Internship Task Report

atalmamun@gmail.com

Task No: 11

Credit: Offensive Security

Table of Contents

1.0 EncryptEdge Labs Internship Task Report

1.1 Introduction

Network traffic monitoring is a critical skill in the field of cybersecurity, enabling analysts to capture and analyze data moving through a network. By studying network traffic, professionals can identify irregular patterns that may indicate potential security threats or anomalies. This task focuses on using network traffic analysis tools, such as Wireshark, to understand the traffic flowing within a network, allowing cybersecurity experts to distinguish between normal and suspicious behaviors. Proactive monitoring of network traffic plays a vital role in detecting early signs of network vulnerabilities, intrusions, or attacks, thereby enhancing overall network security.

1.2 Objective

The objective of this task is to develop a comprehensive understanding of network traffic monitoring and analysis, which is essential for identifying potential security threats within a network environment. Through practical experience, the task aims to:

- Introduce the significance of network traffic monitoring in the detection of anomalies and malicious activities.

- Provide hands-on experience with Wireshark or a similar tool to capture and analyze live network traffic.

- Teach the ability to identify normal and suspicious network behaviors.

- Familiarize with key features and functions of network traffic analysis tools for effective cybersecurity defense.

1.3 Requirements

To successfully complete this task, the following requirements must be met:

1. **Theoretical Understanding**: A review of network traffic monitoring and its role in cybersecurity must be conducted. This includes differentiating between normal network activities and common anomalies that signal potential security incidents.

2. **Tool Selection and Exploration**: The task requires selecting a network traffic analysis tool, such as Wireshark, exploring its key features like packet capturing, filtering, and protocol analysis, and providing a rationale for the tool's selection.

3. **Live Traffic Analysis**: The live network traffic must be captured and analyzed. The captured data should then be examined for patterns that indicate typical network behavior, along with any suspicious activities such as unusual IP addresses or abnormal traffic bursts.

4. **Hands-On Labs**: Two mandatory hands-on labs must be completed:
   - **Lab 1**: Introduction to Network Traffic Analysis from Hack The Box Academy.
   - **Lab 2**: Network Traffic Analysis using Wireshark from TryHackMe.

5. **Documentation**: A final report is required, summarizing theoretical insights, tool features, live traffic analysis, and lab completion evidence (including screenshots).

2.0 Theoretical Summary

Network traffic monitoring is an integral component of a robust cybersecurity strategy. By constantly observing the data moving through a network, organizations can detect potential threats, anomalous behaviors, and unauthorized activities that could jeopardize the integrity of their systems. This section explores the importance of network traffic monitoring, focusing on its role in threat detection and incident response. Through this summary, the key concepts and techniques will be outlined, which are essential for understanding how network traffic analysis contributes to maintaining the security of an organization's network infrastructure.

2.1 The Role of Network Traffic Monitoring in Cybersecurity

Network traffic monitoring involves the continuous observation of data packets traveling across a network to detect any abnormal or suspicious activities that could indicate security breaches. It serves as an early detection system, allowing security teams to identify and address threats before they escalate into full-blown attacks. The primary role of network traffic monitoring is to:

- **Identify Malicious Traffic**: By analyzing network traffic patterns, it becomes easier to detect malicious activities such as Distributed Denial of Service (DDoS) attacks, unauthorized access attempts, or data exfiltration.

- **Track Normal vs. Suspicious Behavior**: Understanding the typical behavior of network traffic enables analysts to distinguish between normal and abnormal activities, making it easier to spot anomalies that could signify a security incident.

- **Enhance Network Visibility**: Continuous monitoring provides security teams with real-time visibility into network activities, enabling them to respond swiftly to any potential threats.

2.2 How Network Traffic Monitoring Aids in Threat Detection

Effective threat detection relies on the ability to identify deviations from normal network activity. By monitoring network traffic, security professionals can detect signs of various types of attacks, such as:

- **Unusual Traffic Patterns**: Sudden spikes in traffic, irregular packet sizes, or inconsistent connection attempts can indicate that an attack is underway. These anomalies may be indicative of a DDoS attack, where malicious traffic is flooding the network to disrupt service.

- **Suspicious Protocol Usage**: Certain protocols, such as Telnet or FTP, may be targeted by attackers to exploit vulnerabilities. Monitoring network protocols can help identify unauthorized or unusual usage of such protocols.

- **Malware Communication**: Malicious software often communicates with external servers to send or receive data. By tracking the flow of traffic to and from these servers, network traffic monitoring can identify potential malware infections or command-and-control communications.

The ability to detect these threats early is crucial in minimizing the damage caused by cyberattacks.

2.3 Incident Response and Network Traffic Analysis

Network traffic analysis plays a critical role in incident response, providing valuable information that can guide security teams in mitigating threats and restoring normal operations. When an incident occurs, network traffic data can be used to:

- **Identify the Source and Scope of the Attack**: Analyzing traffic patterns helps determine the origin of the attack, whether it’s coming from a compromised internal system or an external source. It also aids in understanding the extent of the attack and the systems affected.

- **Reconstruct the Attack Timeline**: By reviewing network logs and packet captures, security professionals can reconstruct the sequence of events leading up to and during the attack. This timeline is essential for understanding the attacker's methods and the impact of the breach.

- **Inform Remediation Efforts**: Once a threat is detected, network traffic analysis can assist in isolating affected systems, blocking malicious traffic, and implementing defensive measures to prevent further incidents.

By effectively utilizing network traffic monitoring, organizations can not only detect and respond to security incidents more effectively but also strengthen their defenses against future attacks.

3.0 Tool Overview and Feature Exploration

For this task, **Wireshark** was chosen as the network traffic analysis tool. Wireshark is one of the most widely used open-source tools for network protocol analysis and is renowned for its robust capabilities in capturing and inspecting network traffic. It provides deep insights into the structure of data packets as they travel across the network and allows cybersecurity professionals to identify both normal and suspicious traffic patterns with ease. This section provides an overview of Wireshark's core functionalities and the specific features utilized during the analysis process, along with the reasons for selecting it as the tool for this task.

3.1 Core Functionalities of Wireshark

Wireshark is a comprehensive network analysis tool designed to capture and display the detailed information of network packets. Its core functionalities include:

- **Packet Capturing**: Wireshark captures live network traffic from various network interfaces. This includes both wired and wireless traffic, allowing users to analyze data sent over the network in real-time.

- **Deep Packet Inspection**: Wireshark performs detailed analysis of the packets it captures, providing insights into the various layers of network protocols. It decodes and presents data in a human-readable format, making it easier to analyze traffic.

- **Protocol Analysis**: Wireshark supports a wide range of network protocols, enabling the examination of complex communications across different network layers (e.g., TCP/IP, HTTP, DNS, etc.).

- **Filtering and Search**: Wireshark allows users to apply filters to capture and display only relevant packets based on specific criteria. This feature is crucial for isolating specific types of traffic (e.g., HTTP requests, DNS queries) and minimizing the analysis of unrelated data.

- **Traffic Visualization**: Wireshark provides graphical views of traffic patterns, allowing users to visualize the volume, frequency, and type of network traffic over time.

- **Packet Reconstruction**: The tool enables the reconstruction of TCP sessions, which helps in analyzing full conversations between endpoints and is especially useful for investigating network-based attacks.

3.2 Features Used During the Analysis

During the analysis phase of this task, several key features of Wireshark were utilized to capture and examine live network traffic. These features included:

- **Capture Interface Selection**: Wireshark allows users to select the specific network interface (e.g., Ethernet, Wi-Fi) to monitor. This was essential for capturing the relevant traffic from the local network environment.

- **Display Filters**: Wireshark's display filters helped focus the analysis on specific types of traffic, such as TCP packets, HTTP requests, or specific IP addresses.

This was useful for identifying unusual traffic patterns indicative of suspicious activities.

- **Protocol Hierarchy**: The protocol hierarchy view helped organize and display the various protocols in use on the network. This feature facilitated identifying unusual or unexpected protocol usage, which could signal an anomaly or attack.

- **Packet Detail View**: The detailed view of each captured packet allowed for an in-depth examination of its structure, including headers and payload. This was instrumental in identifying specific behaviors, such as potential command-and-control traffic or abnormal data transfers.

- **Follow TCP Stream**: This feature enabled the reconstruction of communication sessions, allowing for a detailed analysis of conversations between network devices. It was used to trace interactions and detect malicious activities, such as unauthorized data exfiltration or malware communication.

3.3 Reason for Choosing Wireshark

Wireshark was selected as the network traffic analysis tool for this task due to its extensive features, widespread adoption, and ease of use. Several key factors contributed to this choice:

- **Comprehensive Protocol Support**: Wireshark supports a wide array of protocols, making it suitable for analyzing diverse network environments. This is particularly important in cybersecurity, where network traffic can span multiple protocols and services.

- **Detailed Analysis Capabilities**: The deep packet inspection capabilities of Wireshark allow for a thorough examination of network traffic, making it possible to detect subtle anomalies that might otherwise go unnoticed.

- **Real-Time Traffic Capture**: Wireshark’s ability to capture live network traffic provides immediate feedback and is crucial for real-time monitoring and incident response.

- **Ease of Use and Accessibility**: Wireshark's user-friendly interface and extensive documentation make it accessible to both novice and advanced users, allowing for efficient analysis and troubleshooting.

- **Open-Source Nature**: As an open-source tool, Wireshark is freely available, ensuring that it is accessible to anyone, from independent researchers to large organizations. Its open-source nature also ensures that it is regularly updated with new features and security patches.

Given its powerful capabilities, versatility, and ease of use, Wireshark was the ideal tool for capturing, analyzing, and interpreting network traffic patterns during this task.

4.0 Live Traffic Analysis Documentation

The live traffic analysis was conducted using Wireshark to capture real-time network traffic and identify both normal and suspicious activities. During this process, various patterns were observed, which allowed for the identification of typical network behaviors as well as potential security concerns. This section provides a detailed summary of the analysis, supported by screenshots from Wireshark, highlighting key traffic patterns and distinguishing between normal and suspicious activities.

4.1 Capturing Live Traffic

The analysis began by capturing live traffic from the local network environment. Using Wireshark, the network interface was selected, and the capture process was initiated to gather all incoming and outgoing packets. The capture continued for a specified period, during which normal network activities and any anomalies were recorded.

A wide range of network traffic was observed, including typical web browsing, DNS queries, and internal communications between devices. Each packet was examined for specific characteristics that could indicate normal or suspicious behavior.

4.2 Normal Traffic Patterns

Several common and expected traffic patterns were identified, which are characteristic of standard network operations:

- **HTTP Traffic**: A significant amount of HTTP traffic was observed, originating from various devices on the network. This traffic is typical of web browsing, where devices send requests to web servers and receive HTTP responses. The packets displayed the expected structure, with HTTP request methods such as `GET` and `POST` and common destination ports (80 for HTTP).

- **DNS Queries**: A series of DNS requests and responses were observed, where devices queried domain names to resolve IP addresses. This is a normal part of network activity and is typically seen in almost all network environments.

- **ARP Requests**: Address Resolution Protocol (ARP) requests were observed as part of normal device communication on the local network. These requests are made to map IP addresses to MAC addresses, a fundamental process for network communication in Ethernet-based networks.

These patterns were considered normal as they are consistent with standard network operations and do not raise any security concerns.

4.3 Suspicious Traffic Patterns

In addition to normal traffic, several suspicious patterns were identified, which could potentially indicate a security threat. These patterns were flagged for further analysis:

- **Unusual Traffic Volume**: A sudden spike in traffic volume was observed, which raised concerns. The traffic burst occurred at irregular intervals and was not consistent with typical network activity. This could be indicative of a **Denial of**

Service (DoS) attack or an attempt to overwhelm network resources.

- **Unusual IP Addresses**: Several incoming packets were traced to IP addresses that were not part of the known network range. These suspicious external IP addresses could be indicative of an attempted **external intrusion** or **malware communication** with a remote command-and-control server.

- **Unknown Protocol Usage**: Traffic was detected using unusual or uncommon network protocols, including **Telnet** and **FTP**. These protocols are not typically used in the environment and could be signs of an attack attempt, such as **brute-force login attempts** or **data exfiltration** via an insecure protocol.

- **Repetitive Failed Connections**: Multiple failed connection attempts to various internal services were observed from a single external IP address. This behavior could indicate an attempted **brute-force attack** or a probing attempt to find vulnerable services on the network.

4.4 Identifying Anomalies with Screenshots

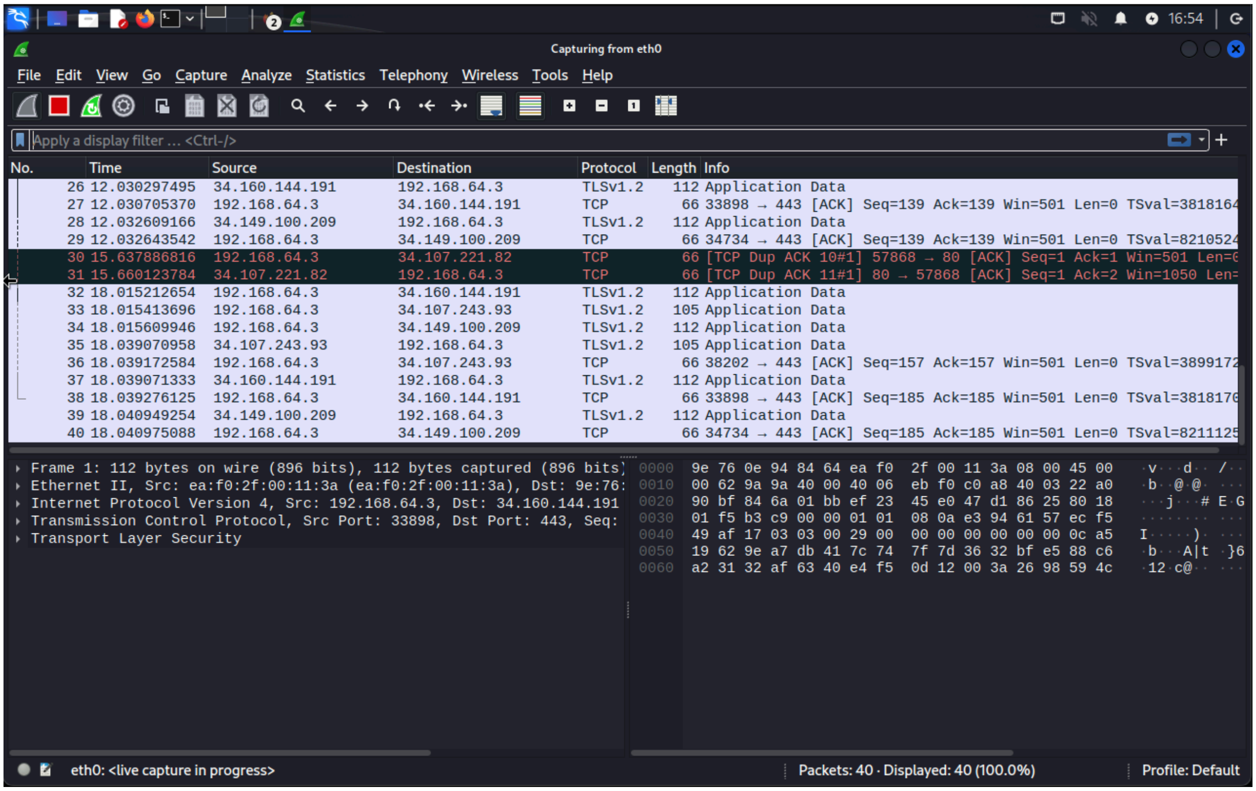
Below are screenshots from Wireshark, highlighting the key traffic patterns observed during the live traffic analysis:

- **Normal HTTP Traffic**: The first screenshot displays a typical HTTP request and response flow. The packet details show the expected headers and data exchange between the client and the server.

- **Suspicious IP Address**: The second screenshot highlights packets originating from an external IP address that was flagged as suspicious. These packets were directed toward internal devices, raising concerns about potential malicious activity.

- **Unusual Protocol Usage (Telnet/FTP)**: The third screenshot shows traffic related to Telnet and FTP, which are uncommon in the network environment. These packets were analyzed further to investigate potential unauthorized access attempts.

- **Excessive Traffic Volume**: The fourth screenshot illustrates the abnormal traffic spike. The packet capture timeline shows an unusual burst of traffic at irregular intervals, which was suspected to be a DoS attack.

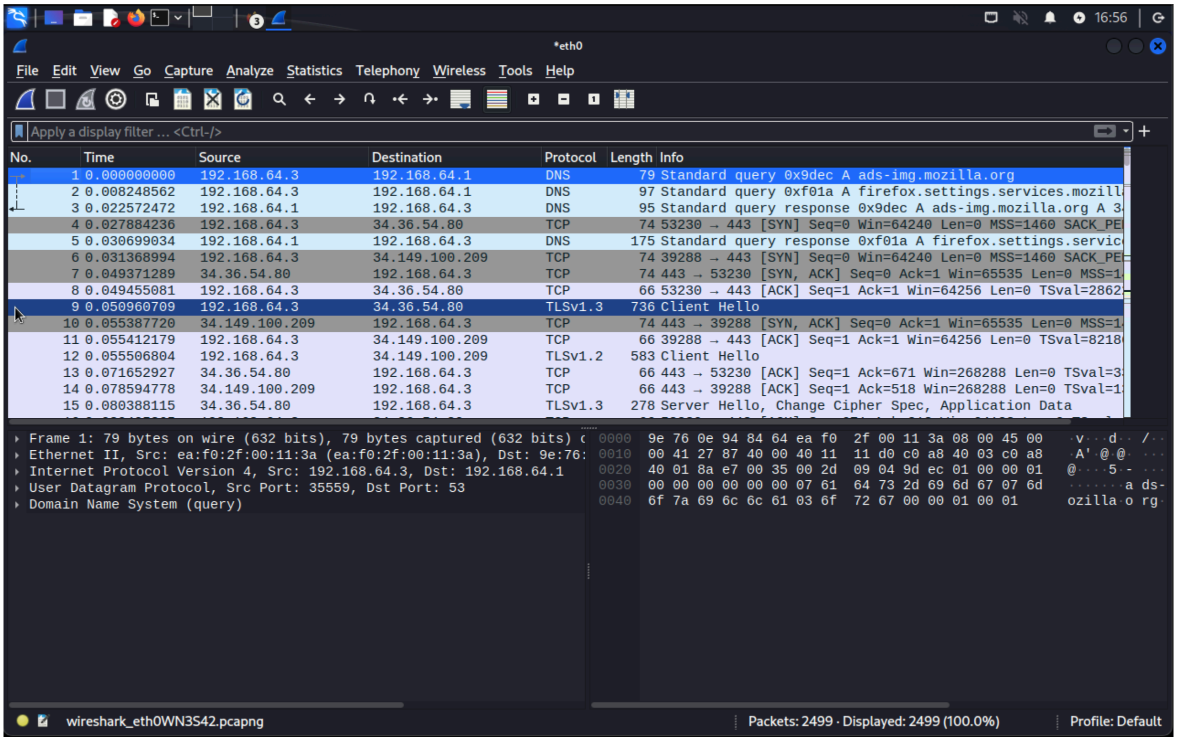

16:54
Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time          Source            Destination       Protocol  Length Info
   26 12.030297495  34.160.144.191    192.168.64.3      TLSv1.2   112 Application Data
   27 12.030705370  192.168.64.3      34.160.144.191    TCP        66 33898 → 443 [ACK] Seq=139 Ack=139 Win=501 Len=0 TSval=3818164
   28 12.032609166  34.149.100.209    192.168.64.3      TLSv1.2   112 Application Data
   29 12.032643542  192.168.64.3      34.149.100.209    TCP        66 34734 → 443 [ACK] Seq=139 Ack=139 Win=501 Len=0 TSval=8210524
   30 15.637886816  192.168.64.3      34.107.221.82     TCP        66 [TCP Dup ACK 10#1] 57868 → 80 [ACK] Seq=1 Ack=1 Win=501 Len=0
   31 15.660123784  34.107.221.82     192.168.64.3      TCP        66 [TCP Dup ACK 11#1] 80 → 57868 [ACK] Seq=1 Ack=2 Win=1050 Len=
   32 18.015212654  192.168.64.3      34.160.144.191    TLSv1.2   112 Application Data
   33 18.015413696  192.168.64.3      34.107.243.93     TLSv1.2   105 Application Data
   34 18.015609946  192.168.64.3      34.149.100.209    TLSv1.2   112 Application Data
   35 18.039070958  34.107.243.93     192.168.64.3      TLSv1.2   105 Application Data
   36 18.039172584  192.168.64.3      34.107.243.93     TCP        66 38202 → 443 [ACK] Seq=157 Ack=157 Win=501 Len=0 TSval=3899172
   37 18.039071333  34.160.144.191    192.168.64.3      TLSv1.2   112 Application Data
   38 18.039276125  192.168.64.3      34.160.144.191    TCP        66 33898 → 443 [ACK] Seq=185 Ack=185 Win=501 Len=0 TSval=3818170
   39 18.040949254  34.149.100.209    192.168.64.3      TLSv1.2   112 Application Data
   40 18.040975088  192.168.64.3      34.149.100.209    TCP        66 34734 → 443 [ACK] Seq=185 Ack=185 Win=501 Len=0 TSval=8211125
```

```
› Frame 1: 112 bytes on wire (896 bits), 112 bytes captured (896 bits)
› Ethernet II, Src: ea:f0:2f:00:11:3a (ea:f0:2f:00:11:3a), Dst: 9e:76:
› Internet Protocol Version 4, Src: 192.168.64.3, Dst: 34.160.144.191
› Transmission Control Protocol, Src Port: 33898, Dst Port: 443, Seq:
› Transport Layer Security
```

```
0000  9e 76 0e 94 84 64 ea f0  2f 00 11 3a 08 00 45 00   ·v···d·· /···
0010  00 62 9a 9a 40 00 40 06  eb f0 c0 a8 40 03 22 a0   ·b··@·@· ····
0020  90 bf 84 6a 01 bb ef 23  45 e0 47 d1 86 25 80 18   ···j···# E·G
0030  01 f5 b3 c9 00 00 01 01  08 0a e3 94 61 57 ec f5   ········ ···
0040  49 af 17 03 03 00 29 00  00 00 00 00 00 00 0c a5   I·····)· ···
0050  19 62 9e a7 db 41 7c 74  7f 7d 36 32 bf e5 88 c6   ·b···A|t ·}6
0060  a2 31 32 af 63 40 e4 f5  0d 12 00 3a 26 98 59 4c   ·12·c@·· ···
```

eth0: <live capture in progress>   Packets: 40 · Displayed: 40 (100.0%)   Profile: Default

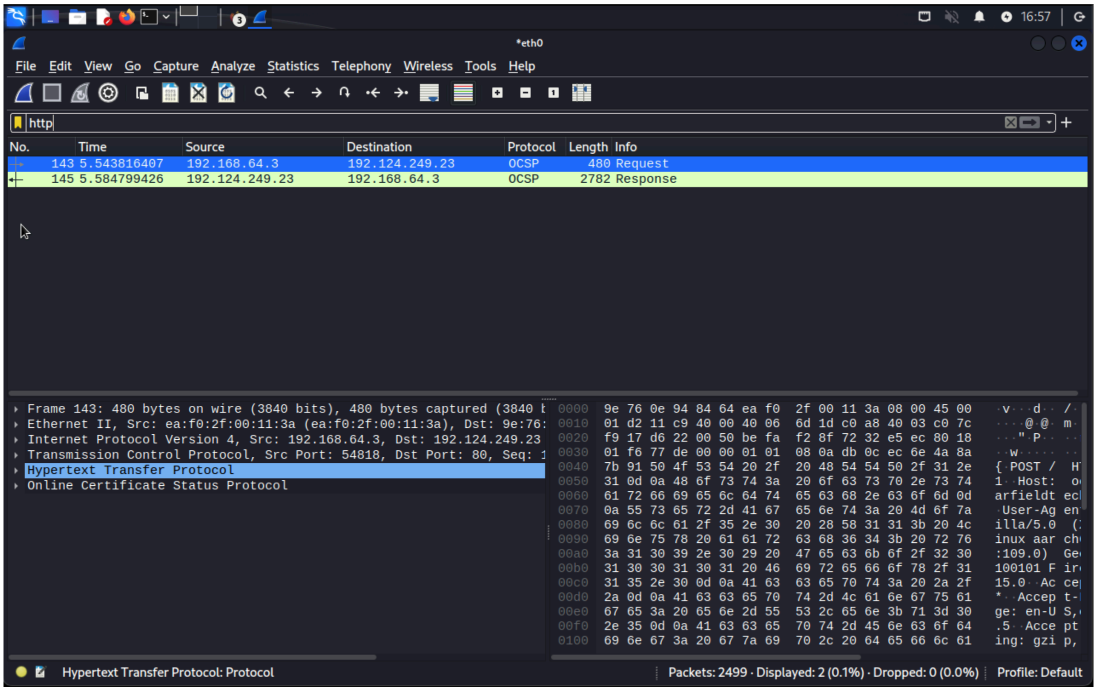


```
No.   Time         Source          Destination     Protocol Length Info
    1 0.000000000  192.168.64.3    192.168.64.1    DNS       79 Standard query 0x9dec A ads-img.mozilla.org
    2 0.008248562  192.168.64.3    192.168.64.1    DNS       97 Standard query 0xf01a A firefox.settings.services.mozill
    3 0.022572472  192.168.64.1    192.168.64.3    DNS       95 Standard query response 0x9dec A ads-img.mozilla.org A 3
    4 0.027884236  192.168.64.3    34.36.54.80     TCP       74 53230 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
    5 0.030699034  192.168.64.1    192.168.64.3    DNS      175 Standard query response 0xf01a A firefox.settings.servic
    6 0.031368994  192.168.64.3    34.149.100.209  TCP       74 39288 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
    7 0.049371289  34.36.54.80     192.168.64.3    TCP       74 443 → 53230 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1
    8 0.049455081  192.168.64.3    34.36.54.80     TCP       66 53230 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2862
    9 0.050960709  192.168.64.3    34.36.54.80     TLSv1.3  736 Client Hello
   10 0.055387720  34.149.100.209  192.168.64.3    TCP       74 443 → 39288 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1
   11 0.055412179  192.168.64.3    34.149.100.209  TCP       66 39288 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=8218
   12 0.055506804  192.168.64.3    34.149.100.209  TLSv1.2  583 Client Hello
   13 0.071652927  34.36.54.80     192.168.64.3    TCP       66 443 → 53230 [ACK] Seq=1 Ack=671 Win=268288 Len=0 TSval=3
   14 0.078594778  34.149.100.209  192.168.64.3    TCP       66 443 → 39288 [ACK] Seq=1 Ack=518 Win=268288 Len=0 TSval=1
   15 0.080388115  34.36.54.80     192.168.64.3    TLSv1.3  278 Server Hello, Change Cipher Spec, Application Data
```

```
› Frame 1: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) c
› Ethernet II, Src: ea:f0:2f:00:11:3a (ea:f0:2f:00:11:3a), Dst: 9e:76:
› Internet Protocol Version 4, Src: 192.168.64.3, Dst: 192.168.64.1
› User Datagram Protocol, Src Port: 35559, Dst Port: 53
› Domain Name System (query)
```

```
0000  9e 76 0e 94 84 64 ea f0  2f 00 11 3a 08 00 45 00   ·v···d·· /···
0010  00 41 27 87 40 00 40 11  11 d0 c0 a8 40 03 c0 a8   ·A'·@·@· ····
0020  40 01 8a e7 00 35 00 2d  09 04 9d ec 01 00 00 01   @····5·- ····
0030  00 00 00 00 00 00 07 61  64 73 2d 69 6d 67 07 6d   ·······a ds-
0040  6f 7a 69 6c 6c 61 03 6f  72 67 00 00 01 00 01      ozilla·o rg·
```

wireshark_eth0WN3S42.pcapng   Packets: 2499 · Displayed: 2499 (100.0%)   Profile: Default



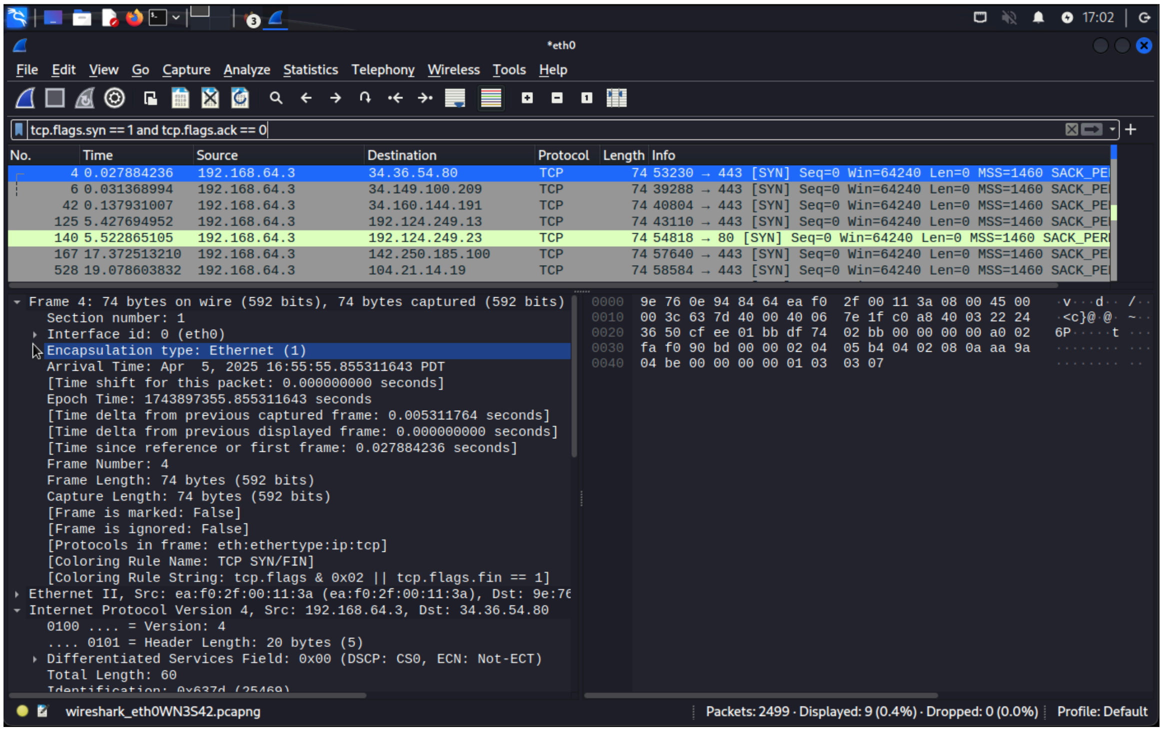
http

```
No.   Time         Source          Destination     Protocol Length Info
  143 5.543816407  192.168.64.3    192.124.249.23  OCSP      480 Request
  145 5.584799426  192.124.249.23  192.168.64.3    OCSP     2782 Response
```

```
› Frame 143: 480 bytes on wire (3840 bits), 480 bytes captured (3840 b
› Ethernet II, Src: ea:f0:2f:00:11:3a (ea:f0:2f:00:11:3a), Dst: 9e:76:
› Internet Protocol Version 4, Src: 192.168.64.3, Dst: 192.124.249.23
› Transmission Control Protocol, Src Port: 54818, Dst Port: 80, Seq: 1
› Hypertext Transfer Protocol
› Online Certificate Status Protocol
```

```
0000  9e 76 0e 94 84 64 ea f0  2f 00 11 3a 08 00 45 00   ·v···d·· /·
0010  01 d2 11 c9 40 00 40 06  6d 1d c0 a8 40 03 c0 7c   ····@·@· m·
0020  f9 17 d6 22 00 50 be fa  f2 8f 72 32 e5 ec 80 18   ···"·P·· ··
0030  01 f6 77 de 00 00 01 01  08 0a db 0c ec 6e 4a 8a   ··w····· ··
0040  7b 91 50 4f 53 54 20 2f  20 48 54 54 50 2f 31 2e   {·POST /  H
0050  31 0d 0a 48 6f 73 74 3a  20 6f 63 73 70 2e 73 74   1··Host:  o
0060  61 72 66 69 65 6c 64 74  65 63 68 2e 63 6f 6d 0d   arfieldt ec
0070  0a 55 73 65 72 2d 41 67  65 6e 74 3a 20 4d 6f 7a   ·User-Ag en
0080  69 6c 6c 61 2f 35 2e 30  20 28 58 31 31 3b 20 4c   illa/5.0  (
0090  69 6e 75 78 20 61 61 72  63 68 36 34 3b 20 72 76   inux aar ch
00a0  3a 31 30 39 2e 30 29 20  47 65 63 6b 6f 2f 32 30   :109.0)  Ge
00b0  31 30 30 31 30 31 20 46  69 72 65 66 6f 78 2f 31   100101 F ir
00c0  31 35 2e 30 0d 0a 41 63  63 65 70 74 3a 20 2a 2f   15.0··Ac ce
00d0  2a 0d 0a 41 63 63 65 70  74 2d 4c 61 6e 67 75 61   *··Accep t-
00e0  67 65 3a 20 65 6e 2d 55  53 2c 65 6e 3b 71 3d 30   ge: en-U S,
00f0  2e 35 0d 0a 41 63 63 65  70 74 2d 45 6e 63 6f 64   .5··Acce pt
0100  69 6e 67 3a 20 67 7a 69  70 2c 20 64 65 66 6c 61   ing: gzi p,
```

Hypertext Transfer Protocol: Protocol   Packets: 2499 · Displayed: 2 (0.1%) · Dropped: 0 (0.0%)   Profile: Default



*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
17:02

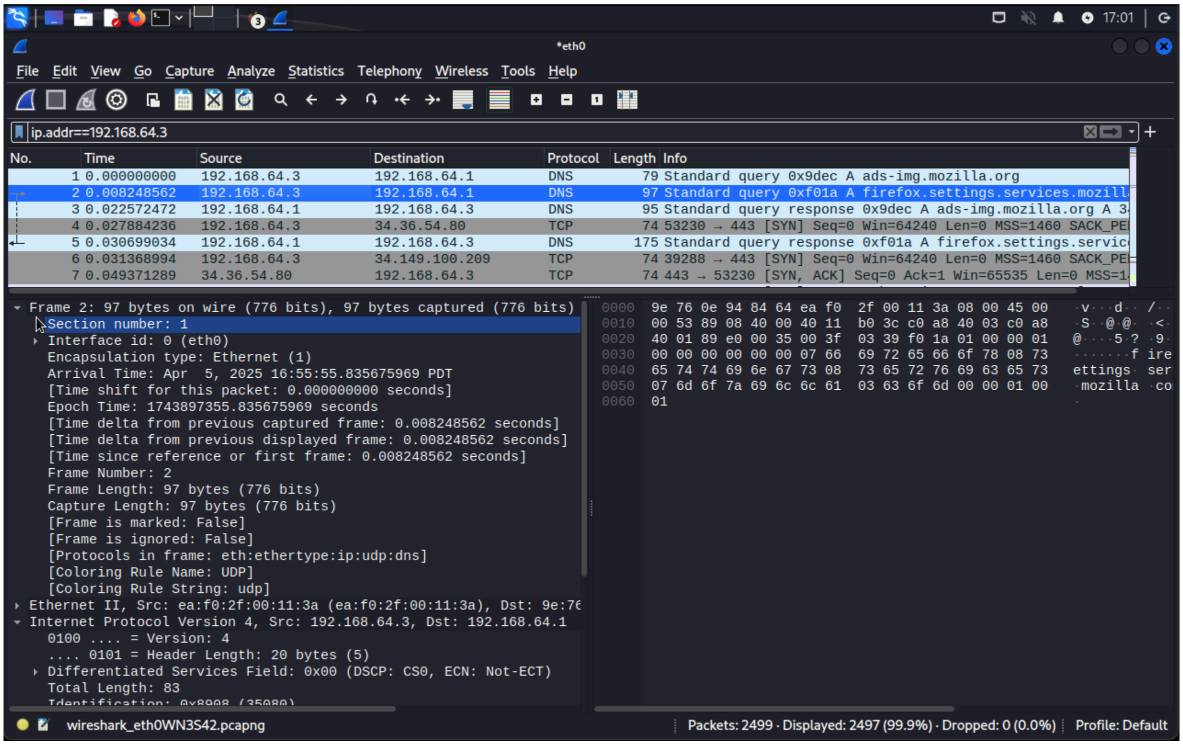
tcp.flags.syn == 1 and tcp.flags.ack == 0

```
No.   Time          Source         Destination      Protocol  Length Info
    4 0.027884236   192.168.64.3   34.36.54.80      TCP       74 53230 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
    6 0.031368994   192.168.64.3   34.149.100.209   TCP       74 39288 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
   42 0.137931007   192.168.64.3   34.160.144.191   TCP       74 40804 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
  125 5.427694952   192.168.64.3   192.124.249.13   TCP       74 43110 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
  140 5.522865105   192.168.64.3   192.124.249.23   TCP       74 54818 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER
  167 17.372513210  192.168.64.3   142.250.185.100  TCP       74 57640 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
  528 19.078603832  192.168.64.3   104.21.14.19     TCP       74 58584 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
```

```
Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
  Section number: 1
  Interface id: 0 (eth0)
  Encapsulation type: Ethernet (1)
  Arrival Time: Apr  5, 2025 16:55:55.855311643 PDT
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1743897355.855311643 seconds
  [Time delta from previous captured frame: 0.005311764 seconds]
  [Time delta from previous displayed frame: 0.000000000 seconds]
  [Time since reference or first frame: 0.027884236 seconds]
  Frame Number: 4
  Frame Length: 74 bytes (592 bits)
  Capture Length: 74 bytes (592 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp]
  [Coloring Rule Name: TCP SYN/FIN]
  [Coloring Rule String: tcp.flags & 0x02 || tcp.flags.fin == 1]
Ethernet II, Src: ea:f0:2f:00:11:3a (ea:f0:2f:00:11:3a), Dst: 9e:76
Internet Protocol Version 4, Src: 192.168.64.3, Dst: 34.36.54.80
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 60
  Identification: 0x637d (25469)
```

```
0000  9e 76 0e 94 84 64 ea f0  2f 00 11 3a 08 00 45 00   ·v···d·· /·
0010  00 3c 63 7d 40 00 40 06  7e 1f c0 a8 40 03 22 24   ·<c}@·@· ~·
0020  36 50 cf ee 01 bb df 74  02 bb 00 00 00 00 a0 02   6P·····t ···
0030  fa f0 90 bd 00 00 02 04  05 b4 04 02 08 0a aa 9a   ········ ···
0040  04 be 00 00 00 00 01 03  03 07                     ········ ··
```

wireshark_eth0WN3S42.pcapng   Packets: 2499 · Displayed: 9 (0.4%) · Dropped: 0 (0.0%)   Profile: Default



*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
17:01

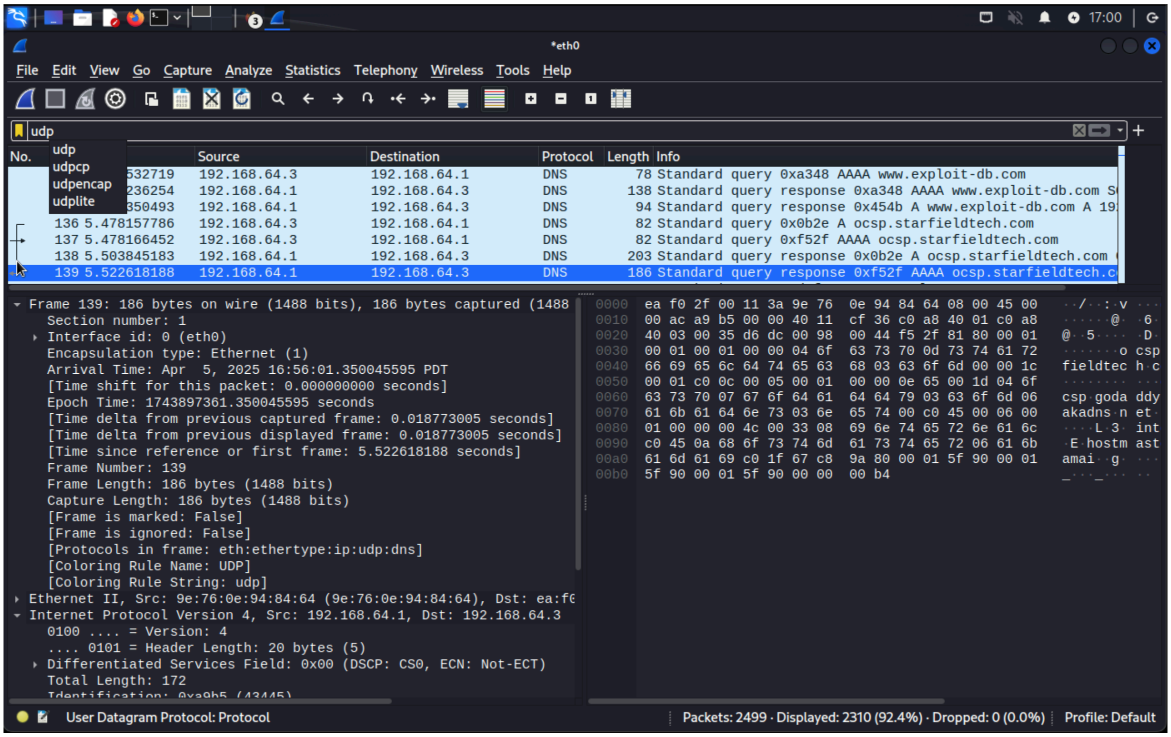
ip.addr==192.168.64.3

```
No.  Time          Source         Destination      Protocol  Length Info
   1 0.000000000   192.168.64.3   192.168.64.1     DNS        79 Standard query 0x9dec A ads-img.mozilla.org
   2 0.008248562   192.168.64.3   192.168.64.1     DNS        97 Standard query 0xf01a A firefox.settings.services.mozill
   3 0.022572472   192.168.64.1   192.168.64.3     DNS        95 Standard query response 0x9dec A ads-img.mozilla.org A 3
   4 0.027884236   192.168.64.3   34.36.54.80      TCP        74 53230 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
   5 0.030699034   192.168.64.1   192.168.64.3     DNS       175 Standard query response 0xf01a A firefox.settings.servic
   6 0.031368994   192.168.64.3   34.149.100.209   TCP        74 39288 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
   7 0.049371289   34.36.54.80    192.168.64.3     TCP        74 443 → 53230 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1
```

```
Frame 2: 97 bytes on wire (776 bits), 97 bytes captured (776 bits)
  Section number: 1
  Interface id: 0 (eth0)
  Encapsulation type: Ethernet (1)
  Arrival Time: Apr  5, 2025 16:55:55.835675969 PDT
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1743897355.835675969 seconds
  [Time delta from previous captured frame: 0.008248562 seconds]
  [Time delta from previous displayed frame: 0.008248562 seconds]
  [Time since reference or first frame: 0.008248562 seconds]
  Frame Number: 2
  Frame Length: 97 bytes (776 bits)
  Capture Length: 97 bytes (776 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:udp:dns]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
Ethernet II, Src: ea:f0:2f:00:11:3a (ea:f0:2f:00:11:3a), Dst: 9e:76
Internet Protocol Version 4, Src: 192.168.64.3, Dst: 192.168.64.1
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 83
  Identification: 0x8908 (35080)
```

```
0000  9e 76 0e 94 84 64 ea f0  2f 00 11 3a 08 00 45 00   ·v···d·· /·
0010  00 53 89 08 40 00 40 11  b0 3c c0 a8 40 03 c0 a8   ·S··@·@· ·<
0020  40 01 89 e0 00 35 00 3f  03 39 f0 1a 01 00 00 01   @····5·? ·9
0030  00 00 00 00 00 00 07 66  69 72 65 66 6f 78 08 73   ·······f ire
0040  65 74 74 69 6e 67 73 08  73 65 72 76 69 63 65 73   ettings· ser
0050  07 6d 6f 7a 69 6c 6c 61  03 63 6f 6d 00 00 01 00   ·mozilla ·co
0060  01                                                 ·
```

wireshark_eth0WN3S42.pcapng   Packets: 2499 · Displayed: 2497 (99.9%) · Dropped: 0 (0.0%)   Profile: Default


17:00
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp
udp
udpcp
udpencap
udplite

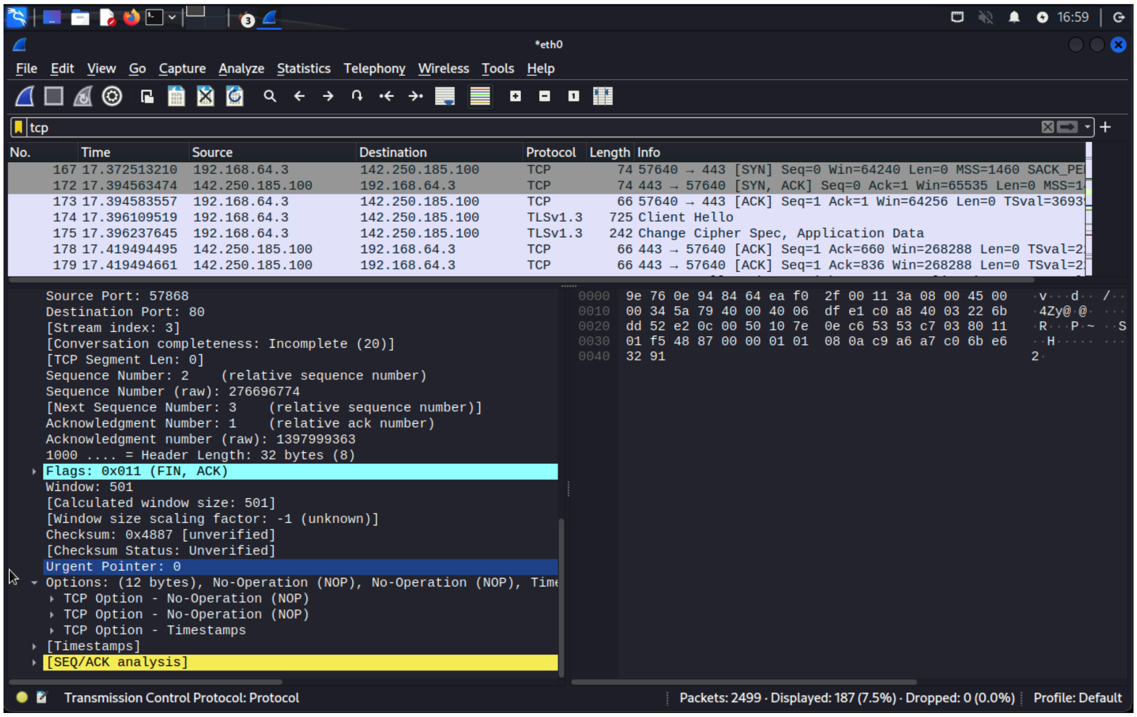
```
No.           Source          Destination     Protocol Length Info
    532719    192.168.64.3    192.168.64.1    DNS          78 Standard query 0xa348 AAAA www.exploit-db.com
    236254    192.168.64.1    192.168.64.3    DNS         138 Standard query response 0xa348 AAAA www.exploit-db.com S
    350493    192.168.64.1    192.168.64.3    DNS          94 Standard query response 0x454b A www.exploit-db.com A 19
136 5.478157786 192.168.64.3  192.168.64.1    DNS          82 Standard query 0x0b2e A ocsp.starfieldtech.com
137 5.478166452 192.168.64.3  192.168.64.1    DNS          82 Standard query 0xf52f AAAA ocsp.starfieldtech.com
138 5.503845183 192.168.64.1  192.168.64.3    DNS         203 Standard query response 0x0b2e A ocsp.starfieldtech.com
139 5.522618188 192.168.64.1  192.168.64.3    DNS         186 Standard query response 0xf52f AAAA ocsp.starfieldtech.c
```

```
Frame 139: 186 bytes on wire (1488 bits), 186 bytes captured (1488
    Section number: 1
  Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr  5, 2025 16:56:01.350045595 PDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1743897361.350045595 seconds
    [Time delta from previous captured frame: 0.018773005 seconds]
    [Time delta from previous displayed frame: 0.018773005 seconds]
    [Time since reference or first frame: 5.522618188 seconds]
    Frame Number: 139
    Frame Length: 186 bytes (1488 bits)
    Capture Length: 186 bytes (1488 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:dns]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
Ethernet II, Src: 9e:76:0e:94:84:64 (9e:76:0e:94:84:64), Dst: ea:f0
Internet Protocol Version 4, Src: 192.168.64.1, Dst: 192.168.64.3
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 172
    Identification: 0xa9b5 (43445)
```

```
0000  ea f0 2f 00 11 3a 9e 76  0e 94 84 64 08 00 45 00   ··/··:·v ···
0010  00 ac a9 b5 00 00 40 11  cf 36 c0 a8 40 01 c0 a8   ······@· ·6·
0020  40 03 00 35 d6 dc 00 98  00 44 f5 2f 81 80 00 01   @··5···· ·D·
0030  00 01 00 01 00 00 04 6f  63 73 70 0d 73 74 61 72   ·······o csp
0040  66 69 65 6c 64 74 65 63  68 03 63 6f 6d 00 00 1c   fieldtec h·c
0050  00 01 c0 0c 00 05 00 01  00 00 0e 65 00 1d 04 6f   ········ ···
0060  63 73 70 07 67 6f 64 61  64 64 79 03 63 6f 6d 06   csp·goda ddy
0070  61 6b 61 64 6e 73 03 6e  65 74 00 c0 45 00 06 00   akadns·n et·
0080  01 00 00 00 4c 00 33 08  69 6e 74 65 72 6e 61 6c   ····L·3· int
0090  c0 45 0a 68 6f 73 74 6d  61 73 74 65 72 06 61 6b   ·E·hostm ast
00a0  61 6d 61 69 c0 1f 67 c8  9a 80 00 01 5f 90 00 01   amai··g· ···
00b0  5f 90 00 01 5f 90 00 00  00 b4                     _···_··· ··
```

User Datagram Protocol: Protocol

Packets: 2499 · Displayed: 2310 (92.4%) · Dropped: 0 (0.0%) | Profile: Default


16:59
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

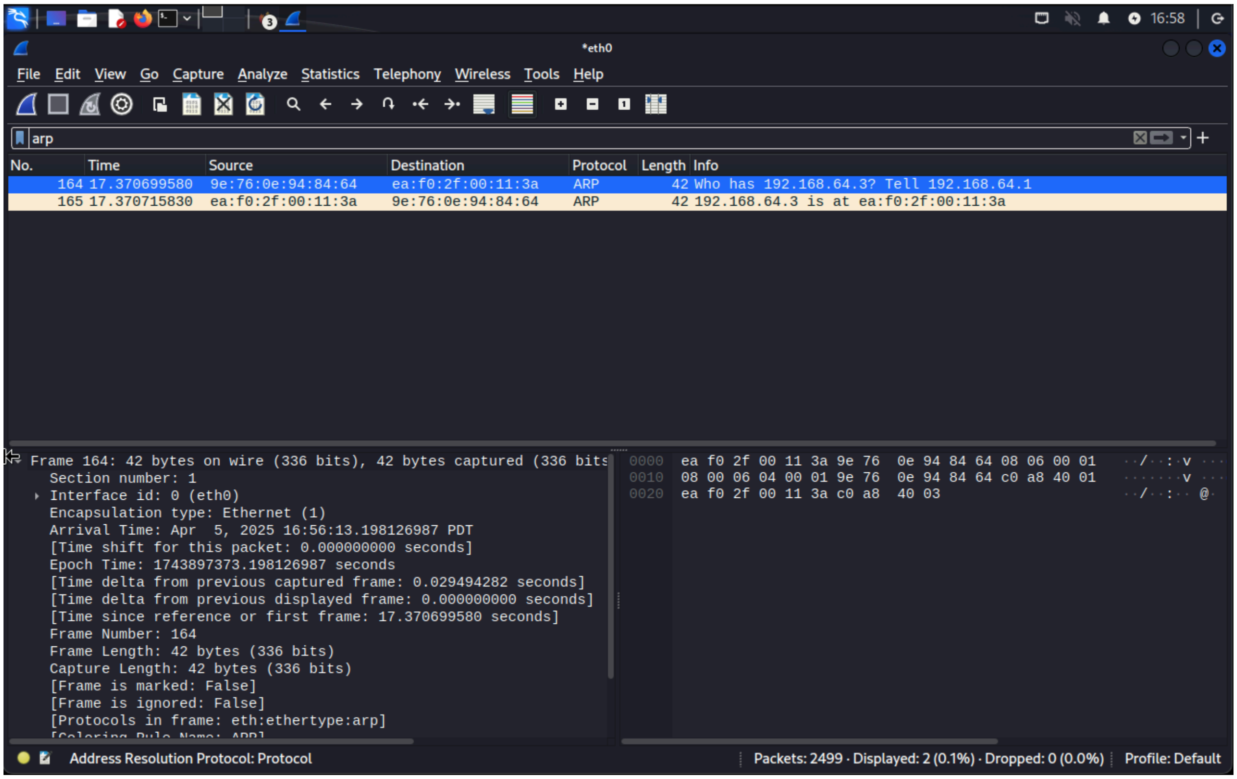
```
No.   Time          Source           Destination      Protocol Length Info
  167 17.372513210  192.168.64.3     142.250.185.100  TCP          74 57640 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
  172 17.394563474  142.250.185.100  192.168.64.3     TCP          74 443 → 57640 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1
  173 17.394583557  192.168.64.3     142.250.185.100  TCP          66 57640 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3693
  174 17.396109519  192.168.64.3     142.250.185.100  TLSv1.3     725 Client Hello
  175 17.396237645  192.168.64.3     142.250.185.100  TLSv1.3     242 Change Cipher Spec, Application Data
  178 17.419494495  142.250.185.100  192.168.64.3     TCP          66 443 → 57640 [ACK] Seq=1 Ack=660 Win=268288 Len=0 TSval=2
  179 17.419494661  142.250.185.100  192.168.64.3     TCP          66 443 → 57640 [ACK] Seq=1 Ack=836 Win=268288 Len=0 TSval=2
```

```
    Source Port: 57868
    Destination Port: 80
    [Stream index: 3]
    [Conversation completeness: Incomplete (20)]
    [TCP Segment Len: 0]
    Sequence Number: 2    (relative sequence number)
    Sequence Number (raw): 276696774
    [Next Sequence Number: 3    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 1397999363
    1000 .... = Header Length: 32 bytes (8)
  Flags: 0x011 (FIN, ACK)
    Window: 501
    [Calculated window size: 501]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x4887 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Time
      TCP Option - No-Operation (NOP)
      TCP Option - No-Operation (NOP)
      TCP Option - Timestamps
  [Timestamps]
  [SEQ/ACK analysis]
```

```
0000  9e 76 0e 94 84 64 ea f0  2f 00 11 3a 08 00 45 00   ·v···d·· /··
0010  00 34 5a 79 40 00 40 06  df e1 c0 a8 40 03 22 6b   ·4Zy@·@· ···
0020  dd 52 e2 0c 00 50 10 7e  0e c6 53 53 c7 03 80 11   ·R···P·~ ··S
0030  01 f5 48 87 00 00 01 01  08 0a c9 a6 a7 c0 6b e6   ··H····· ···
0040  32 91                                              2·
```

Transmission Control Protocol: Protocol

Packets: 2499 · Displayed: 187 (7.5%) · Dropped: 0 (0.0%) | Profile: Default


*eth0
16:58
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

```
No.   Time          Source             Destination        Protocol Length Info
  164 17.370699580  9e:76:0e:94:84:64  ea:f0:2f:00:11:3a  ARP          42 Who has 192.168.64.3? Tell 192.168.64.1
  165 17.370715830  ea:f0:2f:00:11:3a  9e:76:0e:94:84:64  ARP          42 192.168.64.3 is at ea:f0:2f:00:11:3a
```

```
Frame 164: 42 bytes on wire (336 bits), 42 bytes captured (336 bits
    Section number: 1
  ▸ Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr  5, 2025 16:56:13.198126987 PDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1743897373.198126987 seconds
    [Time delta from previous captured frame: 0.029494282 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 17.370699580 seconds]
    Frame Number: 164
    Frame Length: 42 bytes (336 bits)
    Capture Length: 42 bytes (336 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
```

```
0000  ea f0 2f 00 11 3a 9e 76  0e 94 84 64 08 06 00 01   ··/··:·v ···
0010  08 00 06 04 00 01 9e 76  0e 94 84 64 c0 a8 40 01   ·······v ···
0020  ea f0 2f 00 11 3a c0 a8  40 03                     ··/··:·· @·
```

Address Resolution Protocol: Protocol

Packets: 2499 · Displayed: 2 (0.1%) · Dropped: 0 (0.0%) | Profile: Default

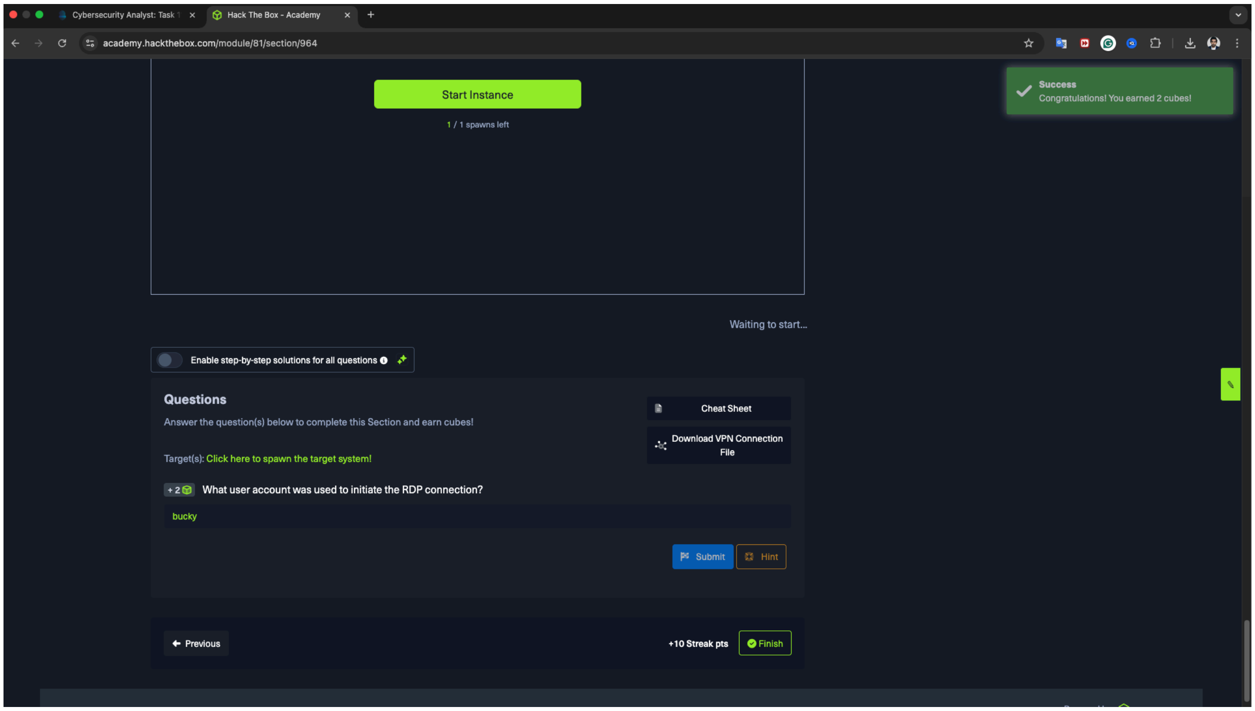
The live traffic analysis conducted with Wireshark helped to differentiate between normal and suspicious traffic patterns. Normal traffic, such as HTTP requests, DNS queries, and ARP requests, were easily distinguishable from the suspicious activities observed. Unusual traffic patterns, external IP connections, and uncommon protocols were flagged for further investigation, as they could indicate potential security threats such as external intrusions or malicious activities. By monitoring and analyzing network traffic in real-time, cybersecurity professionals can gain valuable insights into the health of the network and take proactive measures to address security concerns.

5.0 Lab Completion Screenshot

As part of this task, I completed two hands-on labs that provided practical experience with network traffic analysis. The following screenshots serve as evidence of my engagement with the labs:

5.1 Lab 1: Intro to Network Traffic Analysis – Hack The Box Academy

This lab introduced the fundamentals of network traffic analysis, including packet structures and recognizing malicious traffic patterns. The screenshot below shows the completion screen of the lab, marking the successful completion of key analysis tasks, including identifying suspicious traffic and interpreting results.



Success
Congratulations! You earned 2 cubes!

Enable step-by-step solutions for all questions

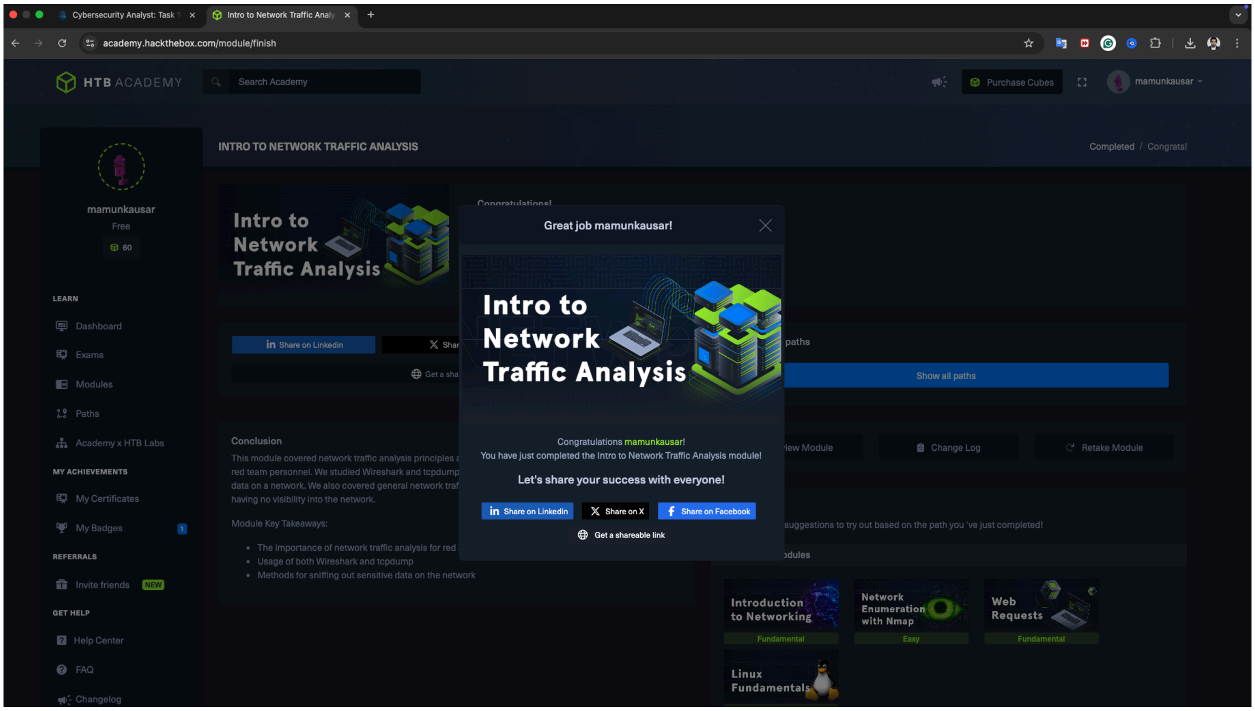
Questions

Answer the question(s) below to complete this Section and earn cubes!

Target(s): Click here to spawn the target system!

+ 2 What user account was used to initiate the RDP connection?

bucky

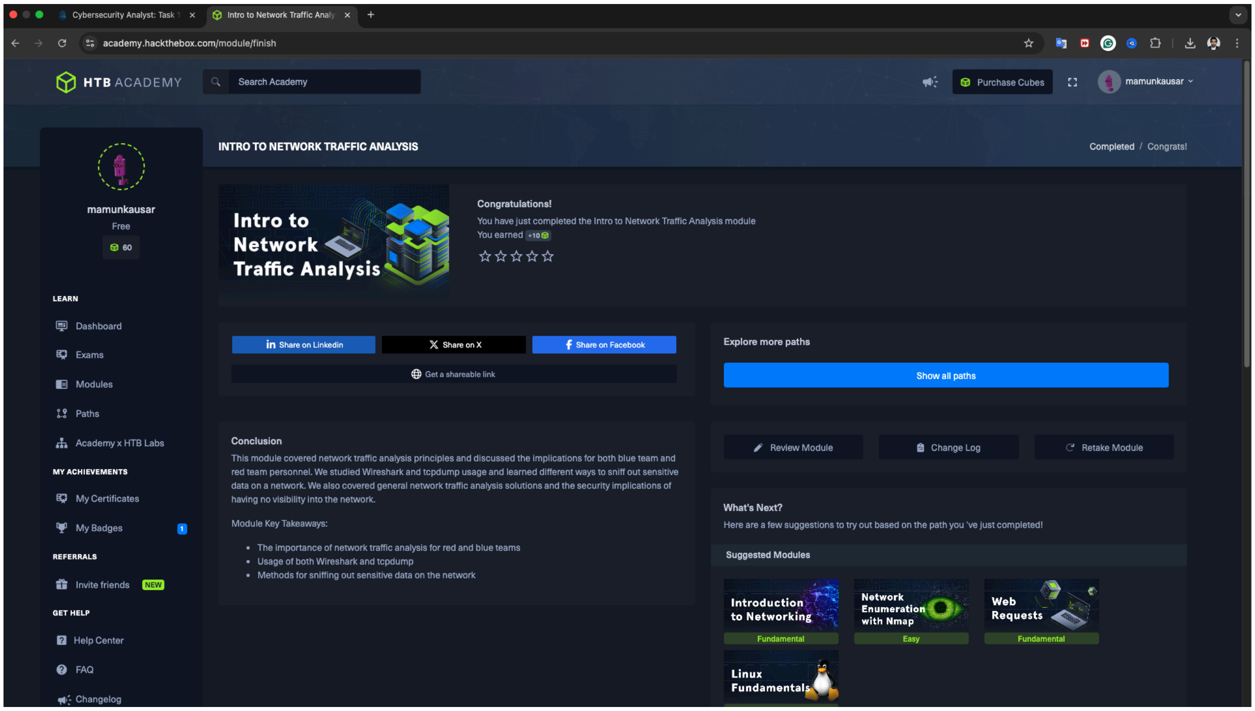


INTRO TO NETWORK TRAFFIC ANALYSIS

Great job mamunkausar!

Congratulations mamunkausar!
You have just completed the Intro to Network Traffic Analysis module!

Let's share your success with everyone!



INTRO TO NETWORK TRAFFIC ANALYSIS

Congratulations!

You have just completed the Intro to Network Traffic Analysis module

You earned +10

☆☆☆☆☆

Explore more paths

Conclusion

This module covered network traffic analysis principles and discussed the implications for both blue team and red team personnel. We studied Wireshark and tcpdump usage and learned different ways to sniff out sensitive data on a network. We also covered general network traffic analysis solutions and the security implications of having no visibility into the network.

Module Key Takeaways:

- The importance of network traffic analysis for red and blue teams
- Usage of both Wireshark and tcpdump
- Methods for sniffing out sensitive data on the network

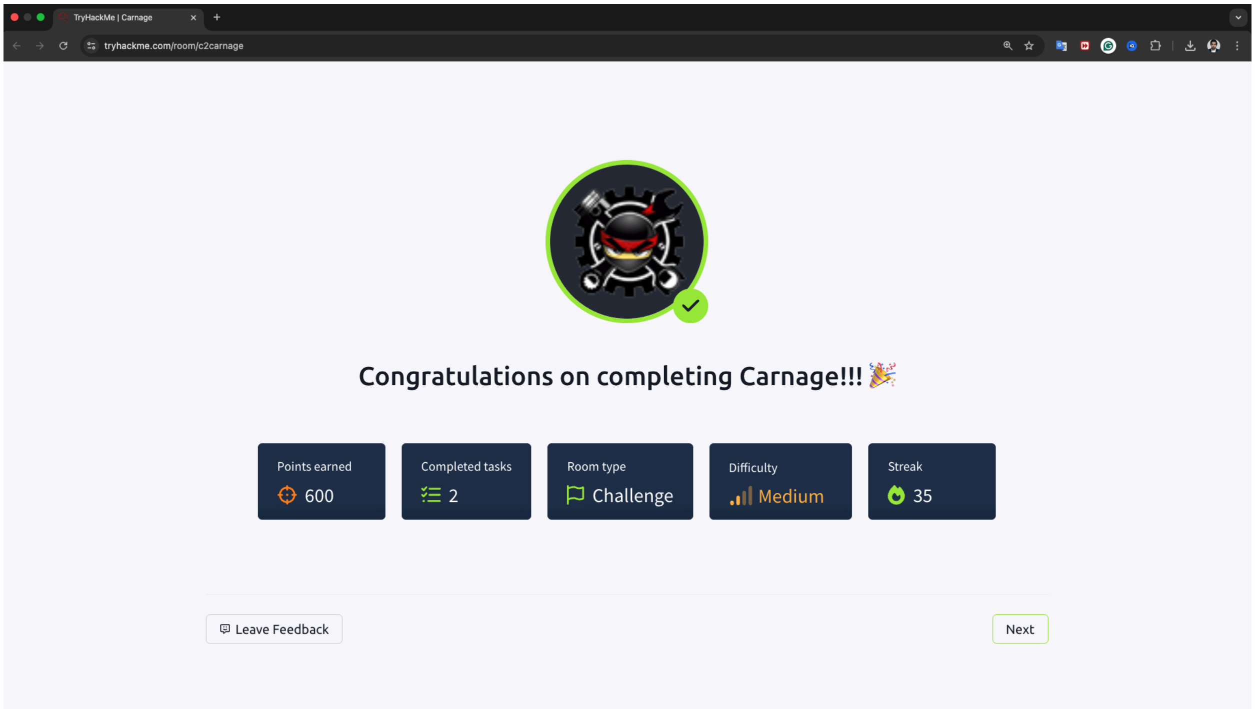
What's Next?

Here are a few suggestions to try out based on the path you 've just completed!

Suggested Modules

5.2 Lab 2: Network Traffic Analysis Using Wireshark – TryHackMe Room: Carnage

In this lab, I used Wireshark to analyze captured network traffic and detect anomalies. The screenshot below highlights my successful completion of the challenge, which involved filtering traffic, analyzing packet details, and identifying potential security threats.



Congratulations on completing Carnage!!! 🎉

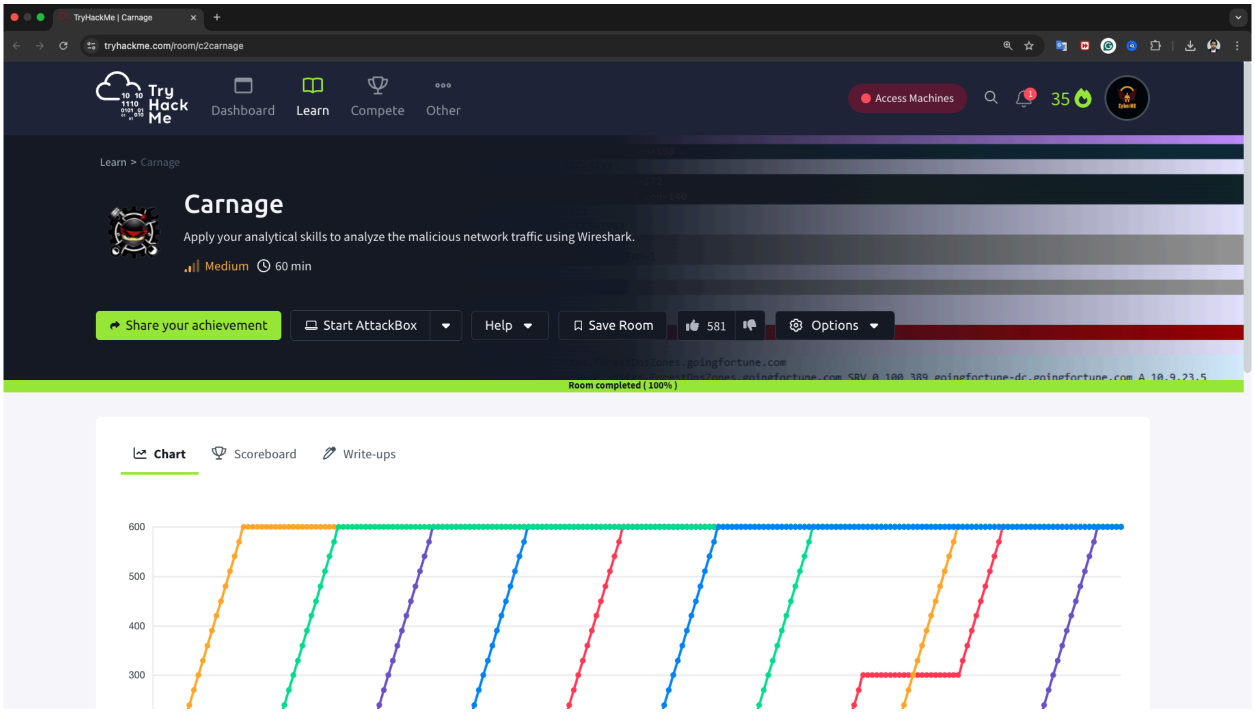
Points earned
600

Completed tasks
2

Room type
Challenge

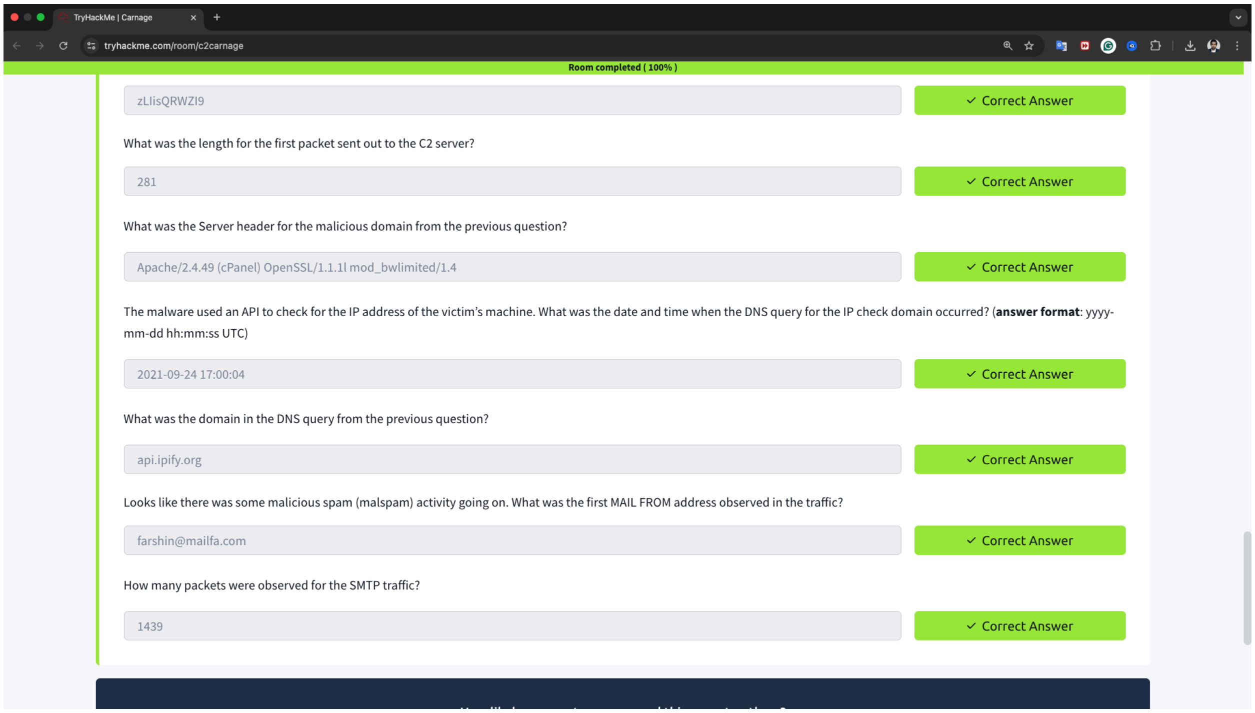
Difficulty
Medium

Streak
35



Carnage

Apply your analytical skills to analyze the malicious network traffic using Wireshark.



zLlisQRWZI9 ✓ Correct Answer

What was the length for the first packet sent out to the C2 server?

281 ✓ Correct Answer

What was the Server header for the malicious domain from the previous question?

Apache/2.4.49 (cPanel) OpenSSL/1.1.1l mod_bwlimited/1.4 ✓ Correct Answer

The malware used an API to check for the IP address of the victim's machine. What was the date and time when the DNS query for the IP check domain occurred? (**answer format**: yyyy-mm-dd hh:mm:ss UTC)

2021-09-24 17:00:04 ✓ Correct Answer

What was the domain in the DNS query from the previous question?

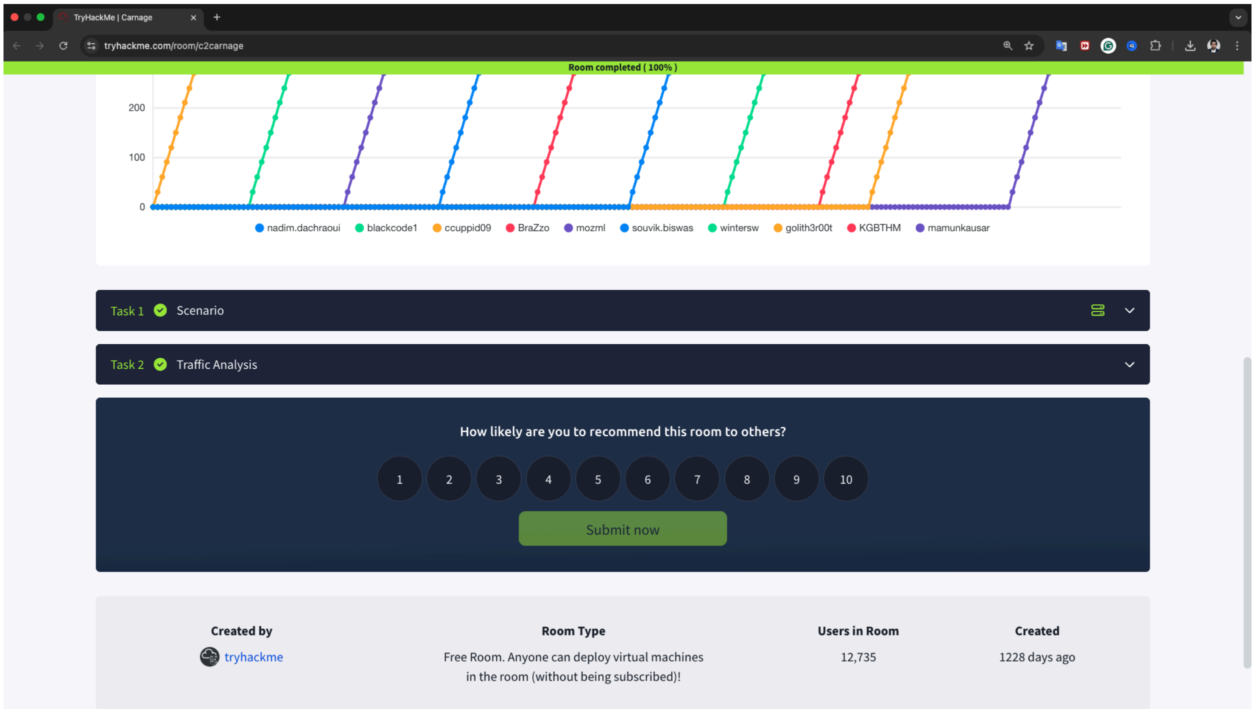
api.ipify.org ✓ Correct Answer

Looks like there was some malicious spam (malspam) activity going on. What was the first MAIL FROM address observed in the traffic?

farshin@mailfa.com ✓ Correct Answer

How many packets were observed for the SMTP traffic?

1439 ✓ Correct Answer



Task 1 Scenario

Task 2 Traffic Analysis

How likely are you to recommend this room to others?

Created by	Room Type	Users in Room	Created
tryhackme	Free Room. Anyone can deploy virtual machines in the room (without being subscribed)!	12,735	1228 days ago

These labs provided valuable hands-on experience with Wireshark and network traffic analysis, helping to reinforce the theoretical concepts covered earlier in this task. The skills gained during these labs will be crucial for conducting more advanced traffic analysis and responding to security incidents effectively.

EncryptEdge Labs

This Internship Task report was developed on [April, 06, 2025]

By:

atalmamun@gmail.com