EncryptEdge Labs

Cybersecurity Analyst Internship

Task Report

atalmamun@gmail.com

Task No: 02

Credit: Offensive Security

Table of Contents

1.0 EncryptEdge Labs Internship Task Report

1.1 Introduction

Networking is a fundamental component of cybersecurity, as it underpins how data is transmitted and secured across systems. This report focuses on key networking concepts such as the OSI model, IP addressing, subnetting, and common network protocols like HTTP, HTTPS, FTP, SNMP, and DNS. The task also involved practical exercises, including protocol interaction using Netcat and packet analysis using Wireshark. These activities provide essential insights into network communication and security vulnerabilities, equipping cybersecurity professionals with the necessary skills to analyze and protect networked environments.

1.2 Objective

The objective of this task was to develop a foundational understanding of networking principles essential for cybersecurity. The key goals included:

- Exploring the OSI model and understanding how data flows through different network layers.
- Learning about IP addressing and subnetting to grasp network segmentation and communication.
- Analyzing common network protocols such as HTTP, HTTPS, FTP, SNMP, and DNS, and their security implications.
- Gaining hands-on experience with packet analysis using Wireshark to inspect and interpret network traffic.
- Using Netcat to experiment with network protocols and understand their interactions.
- Strengthening the ability to identify, analyze, and secure network traffic, a crucial skill for cybersecurity professionals. This includes exploring the OSI model, understanding IP addressing and subnetting, analyzing common network protocols, and gaining hands-on experience with packet analysis using Wireshark.

By completing this task, I aimed to enhance my ability to identify, analyze, and secure network traffic, a crucial skill for protecting modern digital infrastructures.

1.3 Requirements

For this task, the following tools and resources were used:

Tools & Software:

- **Wireshark** – For capturing and analyzing network traffic.
- **Netcat** – For interacting with network protocols and testing connectivity.
- **Kali Linux (or another Linux-based OS)** – Used as the primary environment for running networking tools.

Networking Concepts & Protocols:

- **OSI Model** – Understanding the seven layers of network communication.
- **IP Addressing & Subnetting** – Learning how networks are structured and segmented.
- **Network Protocols** – Researching and analyzing HTTP, HTTPS, FTP, SNMP, and DNS.

Practical Exercises:

- Using **Netcat** to simulate and test network communications.
- Capturing and analyzing **packets with Wireshark**.
- Completing the **"Intro to Networking" lab** as a hands-on learning experience.

2.0 High-Level Summary

This task provided foundational knowledge of networking principles essential for cybersecurity. It covered key concepts such as the OSI model, IP addressing, subnetting, and network protocols. Hands-on exercises included using Netcat to test network protocol interactions and Wireshark to capture and analyze network traffic. These activities helped in understanding network communication, identifying security vulnerabilities, and improving network defense strategies.

2.1 Recommendations

- **Use Secure Protocols**: Replace insecure protocols like FTP with secure alternatives such as SFTP or FTPS.
- **Implement Strong Network Monitoring**: Regularly capture and analyze network traffic with tools like Wireshark to detect anomalies.
- **Enhance DNS Security**: Use DNS security extensions (DNSSEC) to prevent DNS spoofing attacks.
- **Limit Open Ports**: Restrict unnecessary open ports using firewall rules to reduce attack surfaces.
- **Train Employees on Phishing Awareness**: Since DNS and HTTP are frequently targeted in phishing attacks, user awareness training is critical.

2.2 Risk Exposure over Time

- **Unencrypted Protocols**: Continued use of unencrypted protocols like HTTP and FTP can expose sensitive data to attackers through sniffing attacks.
- **Lack of Network Monitoring**: Without regular traffic analysis, malicious activity such as unauthorized access or data exfiltration may go undetected.
- **Poor Subnetting Practices**: Inefficient subnetting can lead to increased broadcast traffic, making networks more vulnerable to congestion and security threats.
- **DNS Vulnerabilities**: If DNS security measures are not implemented, attackers can exploit vulnerabilities for DNS spoofing or cache poisoning attacks.
- **Delayed Incident Response**: Without real-time packet analysis, organizations may struggle to detect and mitigate cyber threats effectively.

3.0 Methodologies

This section outlines the approaches and techniques used to complete the networking task, including information gathering, network protocol analysis, and packet capture with Wireshark.

- **Research on OSI Model**: Studied the seven layers of the OSI model to understand how data flows through a network.
- **IP Addressing and Subnetting**: Explored how IP addresses and subnet masks organize and segment networked devices.
- **Routing Concepts**: Learned how data moves between networks using gateways and routing tables.

3.1 Summary on OSI Model

The **OSI (Open Systems Interconnection) model** is a conceptual framework that standardizes network communication by dividing it into seven layers. Each layer has a specific function in processing and transmitting data across networks.

The Seven Layers of the OSI Model

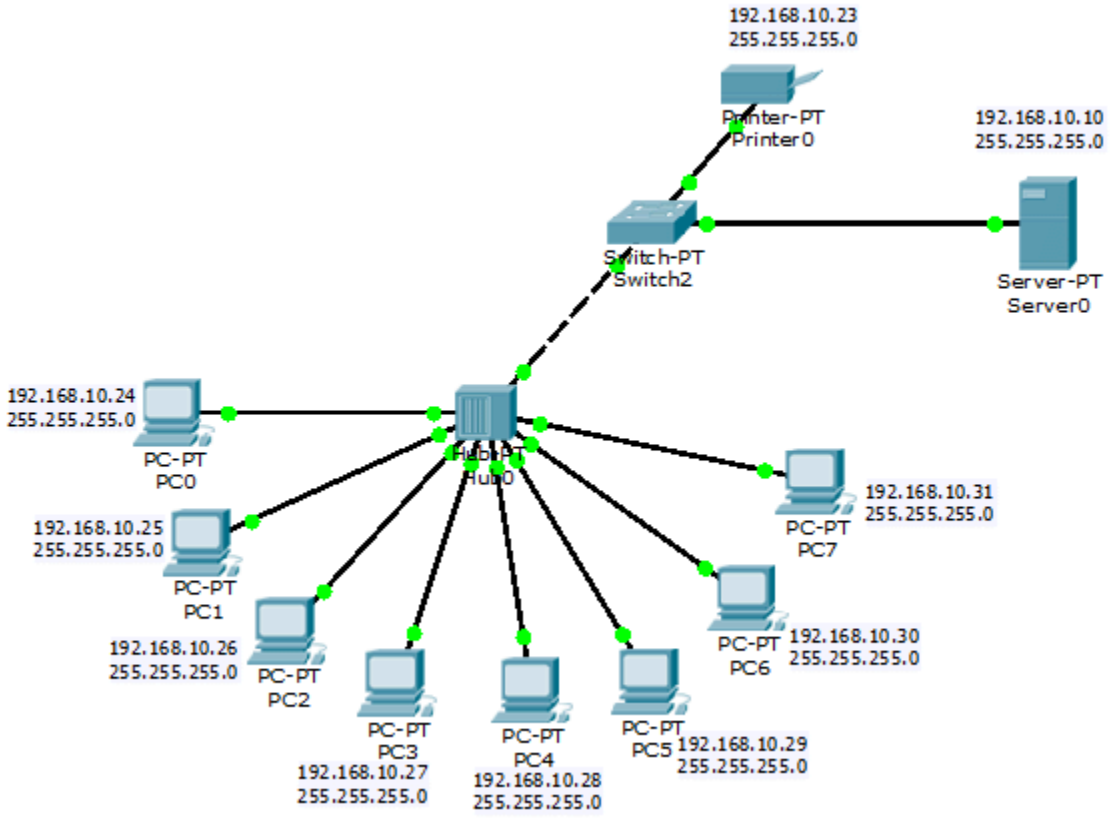
1. **Physical Layer (Layer 1)**
    - Deals with raw data transmission through physical media (cables, radio signals).
    - Converts binary data (1s and 0s) into electrical, optical, or radio signals.
    - Example: Ethernet cables, Wi-Fi signals.
2. **Data Link Layer (Layer 2)**
    - Manages node-to-node data transfer and error detection.
    - Uses MAC (Media Access Control) addresses to identify devices on a network.
    - Example: Ethernet frames, MAC addresses.
3. **Network Layer (Layer 3)**
    - Handles routing and addressing using IP (Internet Protocol).
    - Determines the best path for data to travel between devices.

- Example: IP addresses, routers, IPv4 & IPv6.

4. **Transport Layer (Layer 4)**
   - Ensures complete and reliable data transfer between devices.
   - Uses protocols like TCP (reliable, connection-based) and UDP (faster, connectionless).
   - Example: TCP handshake, UDP streaming.
5. **Session Layer (Layer 5)**
   - Manages and maintains communication sessions between applications.
   - Ensures sessions remain open and can recover in case of failure.
   - Example: Remote desktop connections, authentication protocols.
6. **Presentation Layer (Layer 6)**
   - Translates, encrypts, and compresses data for the application layer.
   - Converts data into a format readable by different systems.
   - Example: SSL/TLS encryption, JPEG image compression.
7. **Application Layer (Layer 7)**
   - Provides network services to end-users and applications.
   - Interfaces with software like web browsers and email clients.
   - Example: HTTP, HTTPS, FTP, DNS.

3.2 IP Addressing & Subnetting

The diagram represents a simple **local area network (LAN)** setup where multiple devices are connected using a **switch**. This network includes **PCs, a printer, and a server**, all assigned **unique IP addresses** within the same subnet. Below is a breakdown of key components and their roles:



Network Devices and Their Roles

- **Switch** (Center of the Network)
    - Acts as the central hub, allowing communication between connected devices.
    - Ensures efficient data transfer within the network.
- **PCs (End Devices)**
    - Multiple PCs (PC0 to PC7) are connected to the switch.
    - Each PC has an assigned **IP address in the 192.168.10.0/24 subnet** with a **255.255.255.0 subnet mask**.
    - This subnet allows up to **254 usable IP addresses**, meaning all devices can communicate within the same network.
- **Server (192.168.10.10)**

  - Provides network services, such as hosting applications or file sharing.
  - Connected directly to the switch for network-wide access.
- **Printer (192.168.10.23)**
  - A shared network printer accessible to all PCs.
  - Connected via the switch, ensuring centralized printing access.

IP Addressing and Subnetting

- The **IP range (192.168.10.0/24)** is used for all devices, meaning they belong to the same subnet.
- The **subnet mask (255.255.255.0)** ensures that all devices can directly communicate without needing a router.
- Each device has a **unique IP address**, preventing conflicts and ensuring smooth data flow.

Importance of This Network Structure

- **Efficient Communication**: All devices can exchange data seamlessly without extra routing configurations.
- **Security & Management**: Subnetting ensures network segmentation, which enhances security and performance.
- **Scalability**: New devices can be easily added within the available IP range.

3.3 Network Protocols

Network protocols are essential for secure communication between devices. This section explores **HTTP, HTTPS, FTP, SNMP, and DNS**, discussing their functionality, security implications, and real-world applications. Additionally, Netcat is used to analyze protocol behavior in a networked environment.

3.3.1 Protocol Reference Guide

Hypertext Transfer Protocol (HTTP)

- ***Purpose***: *HTTP is a client-server protocol used for transmitting hypertext over the web.*
- ***Functionality***: *Operates on **port 80**, facilitating data transfer between web browsers and servers.*
- ***Security Considerations***: *HTTP is **unencrypted**, making it vulnerable to **man-in-the-middle attacks (MITM)** and **data interception**.*
- ***Use Cases***: *Used for **loading web pages** and retrieving online resources when security is not a priority.*

Hypertext Transfer Protocol Secure (HTTPS)

- ***Purpose***: *Secure version of HTTP, ensuring encrypted communication over the internet.*
- ***Functionality***: *Uses **TLS/SSL encryption** and operates on **port 443** to protect data integrity and confidentiality.*
- ***Security Considerations***: *Prevents **eavesdropping**, **data tampering**, and **spoofing** attacks.*
- ***Use Cases***: *Used in **secure online transactions**, **login pages**, and any site handling **sensitive data**.*

File Transfer Protocol (FTP)

- ***Purpose***: *Used for **transferring files** between computers over a network.*
- ***Functionality***: *Works in **active and passive modes**, operating on **port 21** (control) and **port 20** (data transfer).*
- ***Security Considerations***: *Plain FTP sends data **in cleartext**, making it vulnerable to **packet sniffing** and **MITM attacks**. Secure alternatives include **SFTP (SSH File Transfer Protocol)** and **FTPS (FTP Secure with TLS/SSL)**.*
- ***Use Cases***: *Used for **uploading/downloading files**, **web server management**, and **backup storage**.*

Simple Network Management Protocol (SNMP)

- ***Purpose***: *Monitors and manages network devices like **routers, switches, and servers**.*
- ***Functionality***: *Uses UDP **ports 161 and 162** to exchange management information. SNMP agents on devices collect data and send it to a network management system (NMS).*
- ***Security Considerations***: *Older versions like **SNMPv1 and SNMPv2** lack encryption, making them vulnerable to attacks. **SNMPv3** offers authentication and encryption.*
- ***Use Cases***: *Used for **network performance monitoring, device management, and fault detection**.*

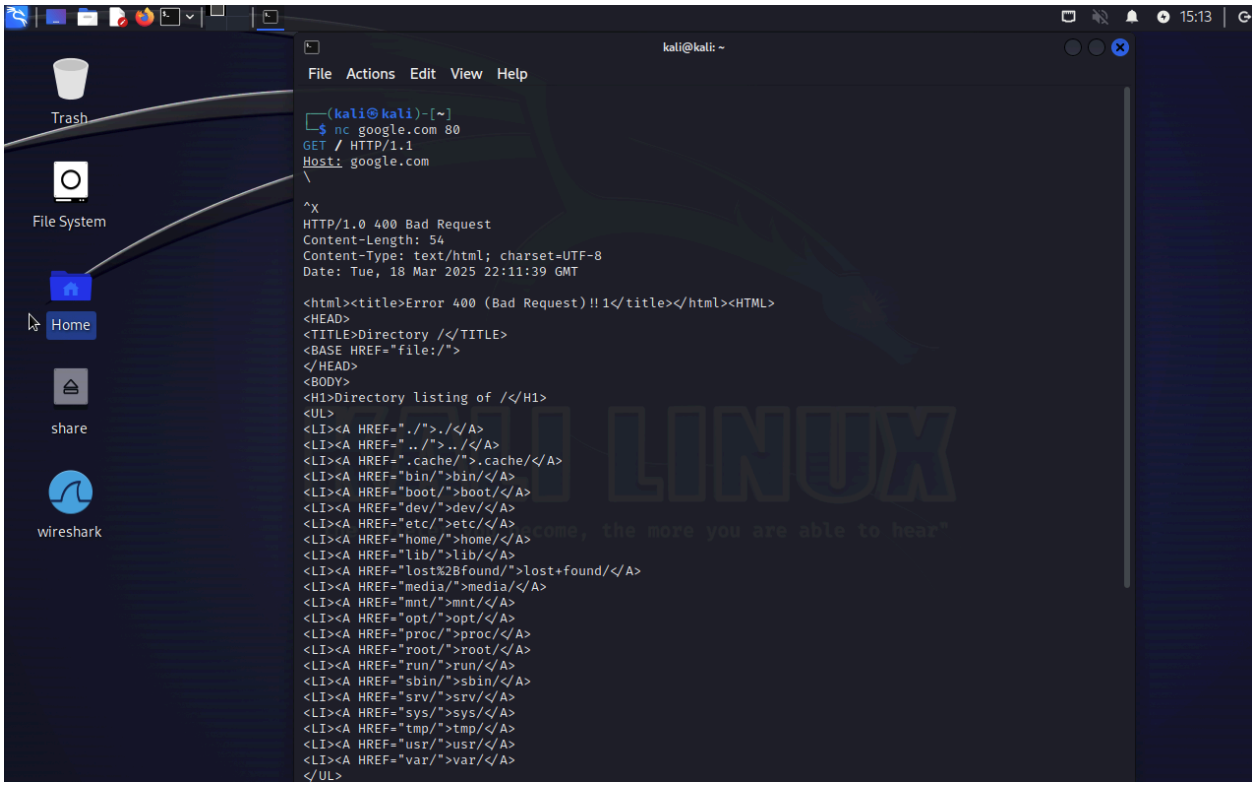
Domain Name System (DNS)

- ***Purpose***: *Translates **human-readable domain names (e.g., google.com)** into **IP addresses (e.g., 142.250.190.46)**.*
- ***Functionality***: *Operates on **UDP port 53**, using a distributed hierarchy of servers to resolve domain names.*
- ***Security Considerations***: *DNS is susceptible to **spoofing, cache poisoning, and DDoS attacks**. **DNSSEC (Domain Name System Security Extensions)** enhances security by verifying authenticity.*
- ***Use Cases***: *Essential for **internet navigation**, allowing users to access websites by name instead of IP addresses.*

3.3.2 Hands-on Practice with Netcat

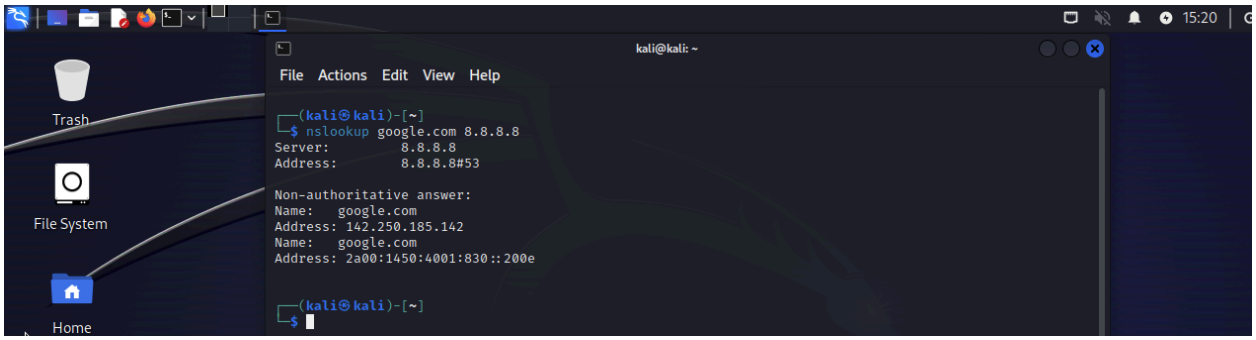
Netcat (**nc**) is a command-line tool used for **testing network connections and protocols**. Below are some Netcat commands demonstrating protocol functionality.

HTTP Request using Netcat



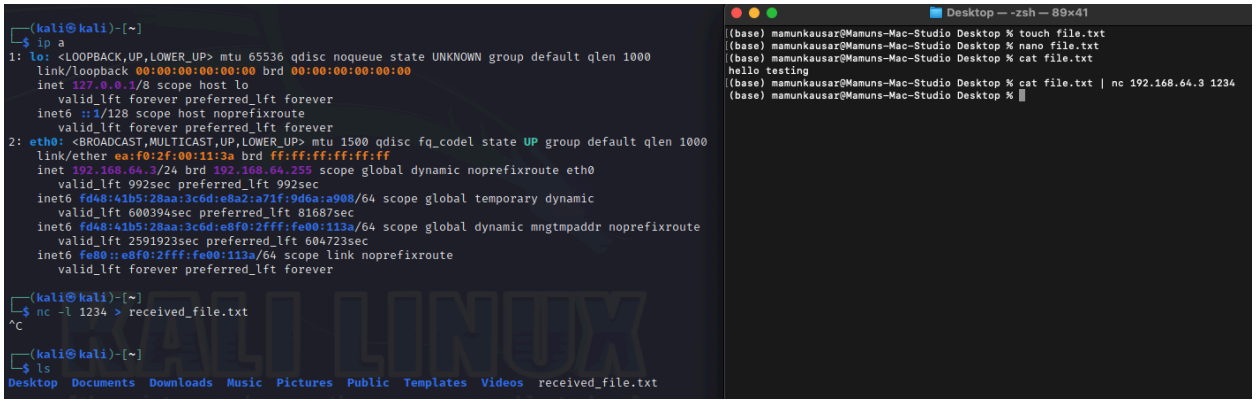
Observation: The response includes the HTTP headers and the webpage source code.

Checking DNS Resolution with Netcat



Observation: Sending DNS queries to Google's public DNS server (**8.8.8.8**) resolves domain names to IP addresses.

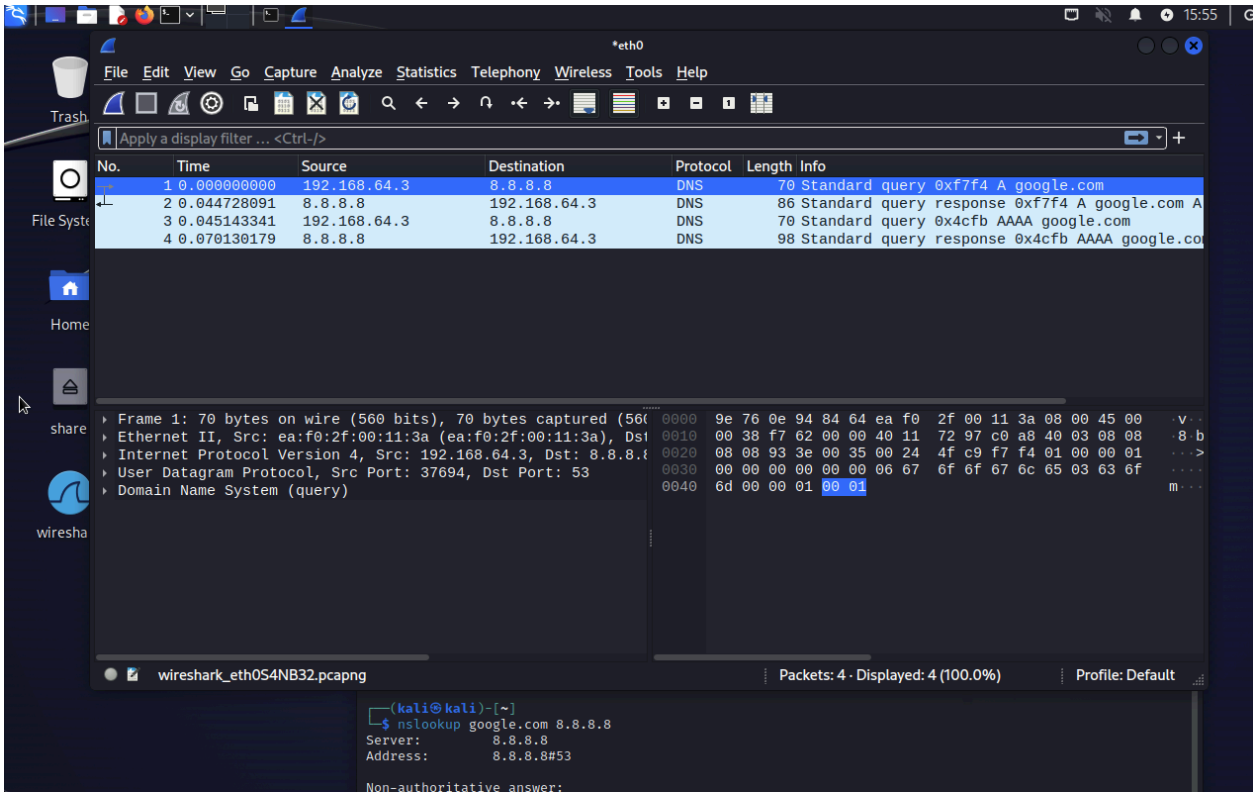
Transferring a File Using Netcat



Observation: The file is transferred from one machine to another using Netcat.

3.4 Packet Analysis Basics with Wireshark

Captured DNS Query and Response



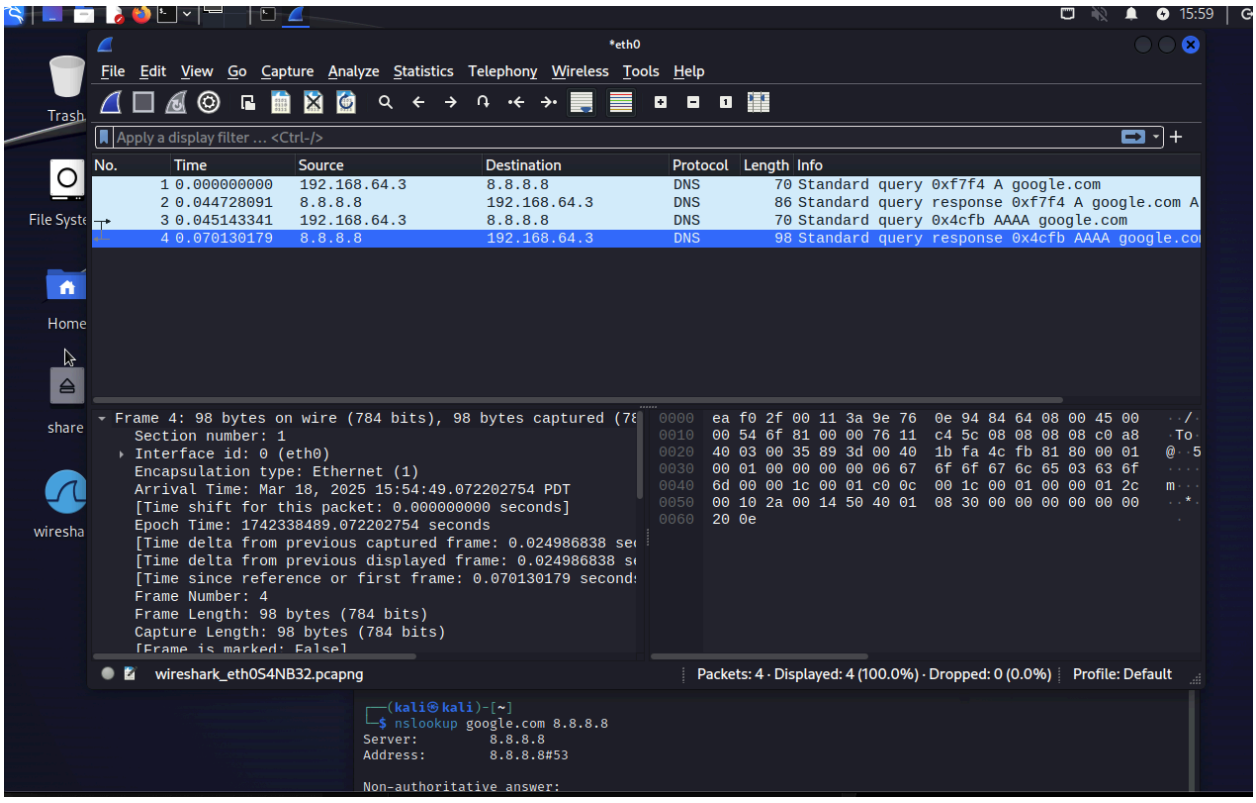
The screenshot shows a DNS query and response process between the local machine (`192.168.64.3`) and Google's public DNS server (`8.8.8.8`).

- **Packet 1:** The local machine sends a DNS query to `8.8.8.8`, requesting the IP address (A record) of `google.com`.
- **Packet 2:** The DNS server responds with the resolved IP address for `google.com`.
- **Packet 3:** Another query is sent for the AAAA (IPv6) record of `google.com`.
- **Packet 4:** The DNS server returns the AAAA record response.

3. Packet Breakdown

Each packet contains multiple layers of network information:

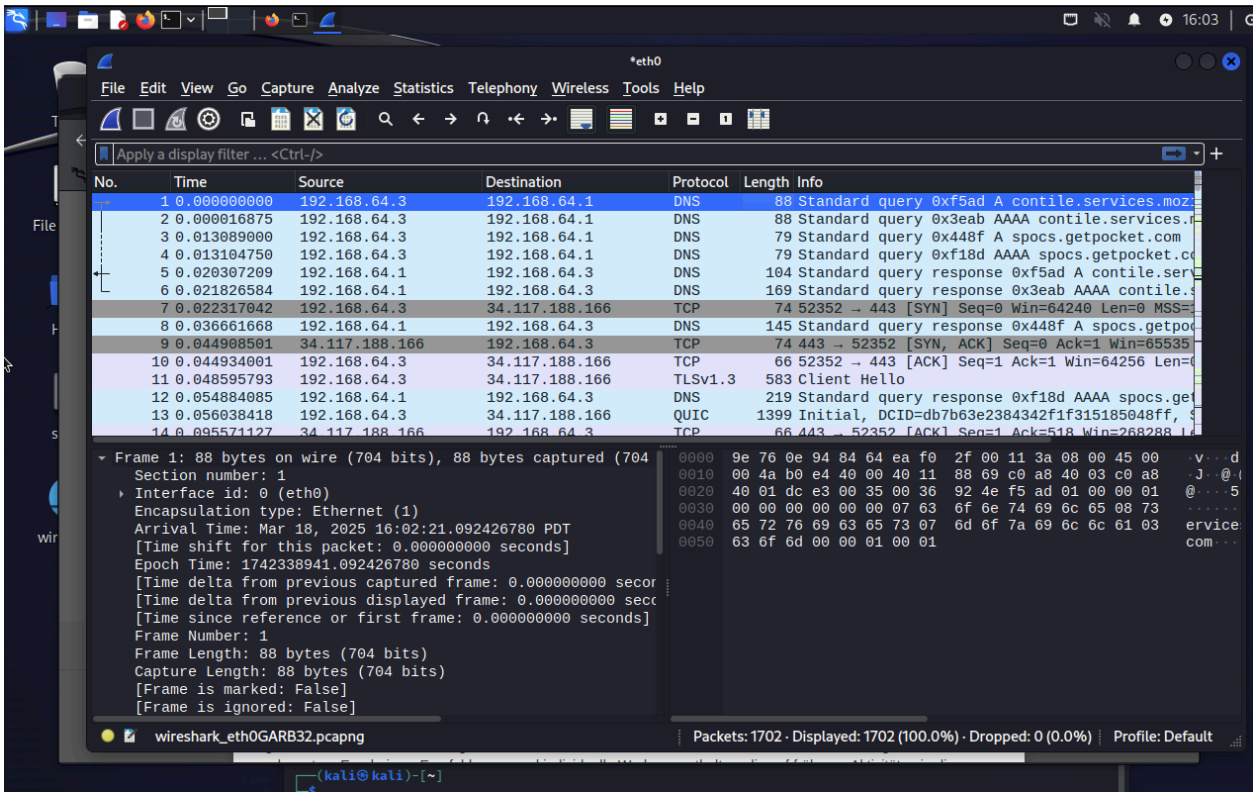
- **Ethernet Layer:** Displays the MAC addresses of source and destination devices.
- **IP Layer:** Shows the source and destination IP addresses (`192.168.64.3` → `8.8.8.8`).
- **UDP Layer:** Indicates that the query was sent over UDP port `53` (DNS service port).
- **DNS Layer:** Contains the actual DNS request, asking for `google.com` (highlighted in hexadecimal).



4. Significance in Network Analysis

- DNS queries and responses help resolve domain names, allowing internet browsing.
- Attackers can exploit DNS traffic for exfiltrating data (DNS tunneling).
- Using Wireshark, cybersecurity professionals can detect anomalies in DNS traffic, such as unauthorized queries or responses.

Wireshark Packet Capture Analysis - DNS and TCP Traffic



Key Observations

1. **DNS Queries and Responses:**
    - The first few packets show DNS queries from `192.168.64.3` (local machine) to `192.168.64.1` (local DNS server) requesting name resolution for domains such as `contile.services.mozilla.com` and `getpocket.com`.
    - The responses provide the corresponding IP addresses or AAAA records.
2. **TCP Handshake:**
    - Packets 7 to 9 illustrate a **TCP three-way handshake** between the local machine (`192.168.64.3`) and `34.117.188.166`.

- **SYN (Packet 7):** The client initiates a connection to port 443 (HTTPS) on the server.
- **SYN-ACK (Packet 9):** The server acknowledges the request.
- **ACK (Packet 10):** The client confirms, establishing the connection.

3. **TLS Handshake:**
   - A **TLS Client Hello** message is observed in Packet 11, initiating secure communication.
   - This indicates that an HTTPS session is being set up, likely for encrypted web browsing.
4. **QUIC Protocol Traffic:**
   - The presence of QUIC packets suggests modern encrypted web communication, often used by Google and other major services to improve performance over traditional TCP.

4.0 Hands-on Labs – Networking Fundamentals & Wireshark

4.1 Intro to Networking Lab

The **Intro to Networking** room provides foundational knowledge of networking, including key concepts such as:

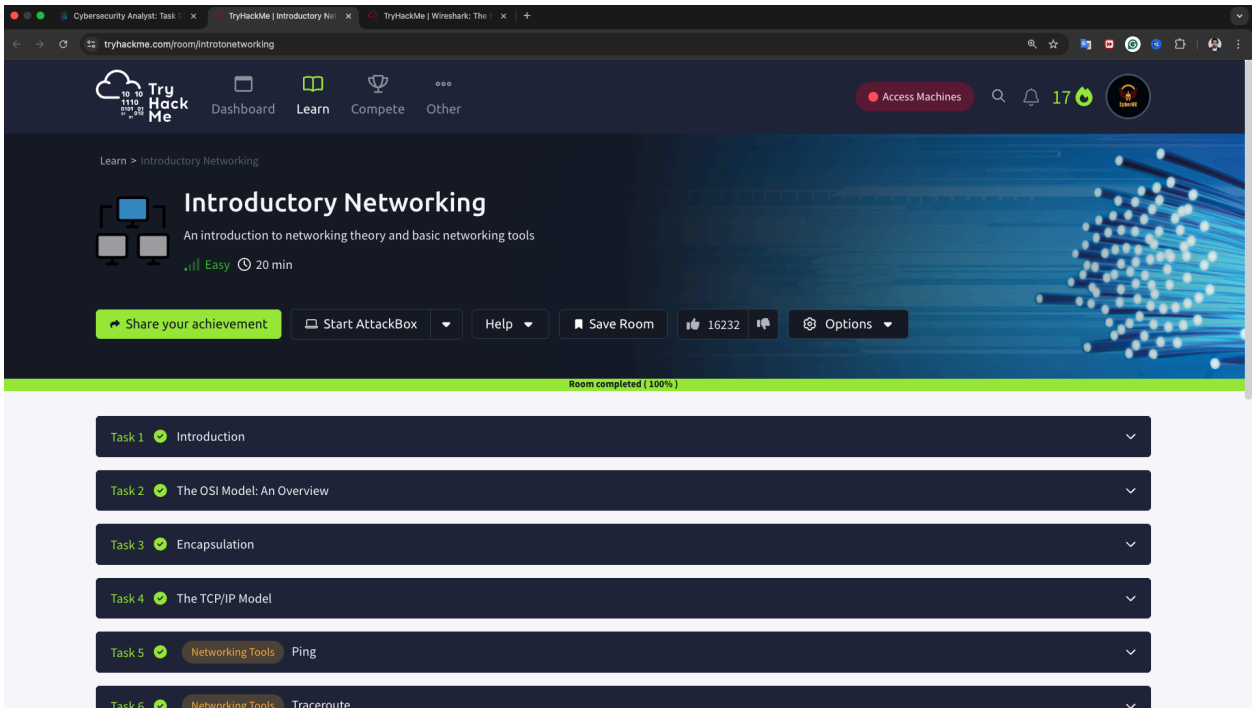
- **IP Addressing** – Understanding IPv4 and IPv6, subnetting, and addressing schemes.
- **Network Protocols** – Covering common protocols such as HTTP, DNS, and TCP/IP.
- **Basic Network Troubleshooting** – Using tools like `ping`, `traceroute`, and `netstat` to diagnose connectivity issues.

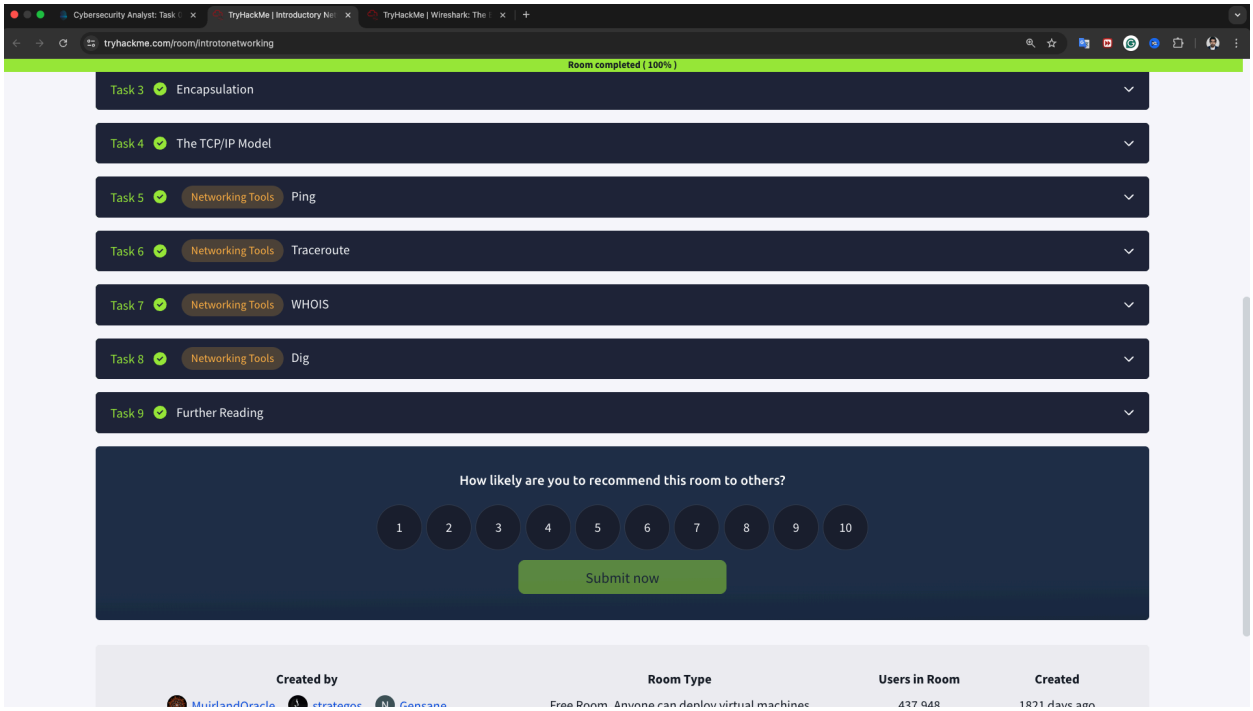
Hands-on Activities & Findings

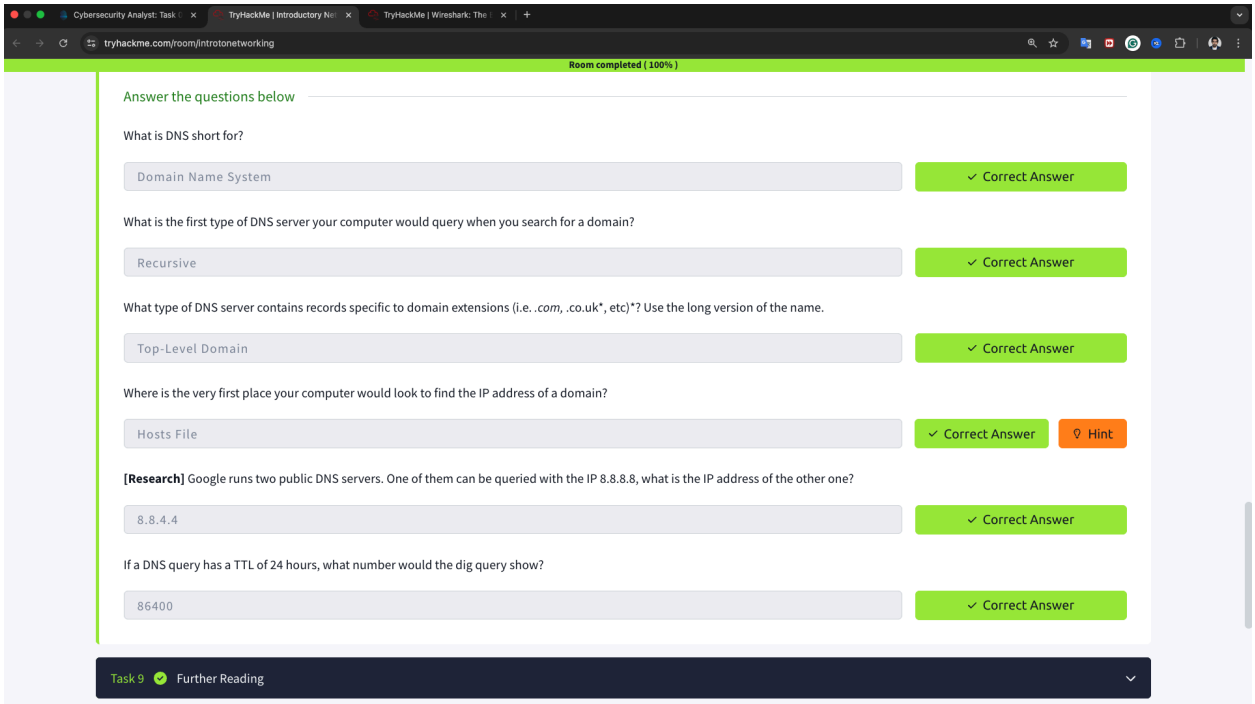
During this lab, I performed the following actions:

- Configured and analyzed IP addresses and subnet masks.
- Practiced troubleshooting connectivity issues using command-line tools.
- Examined network packet flow between devices.Screenshots

Screenshots




tryhackme.com/room/introtonetworking
Room completed (100%)
Task 3 Encapsulation
Task 4 The TCP/IP Model
Task 5 Networking Tools Ping
Task 6 Networking Tools Traceroute
Task 7 Networking Tools WHOIS
Task 8 Networking Tools Dig
Task 9 Further Reading
How likely are you to recommend this room to others?
1 2 3 4 5 6 7 8 9 10
Submit now
Created by
Room Type
Users in Room
Created
Free Room. Anyone can deploy virtual machines
437,948
1821 days ago


tryhackme.com/room/introtonetworking
Room completed (100%)
Answer the questions below
What is DNS short for?
Domain Name System
✓ Correct Answer
What is the first type of DNS server your computer would query when you search for a domain?
Recursive
✓ Correct Answer
What type of DNS server contains records specific to domain extensions (i.e. .com, .co.uk*, etc)*? Use the long version of the name.
Top-Level Domain
✓ Correct Answer
Where is the very first place your computer would look to find the IP address of a domain?
Hosts File
✓ Correct Answer
Hint
[Research] Google runs two public DNS servers. One of them can be queried with the IP 8.8.8.8, what is the IP address of the other one?
8.8.4.4
✓ Correct Answer
If a DNS query has a TTL of 24 hours, what number would the dig query show?
86400
✓ Correct Answer
Task 9 Further Reading

4.2 Wireshark Basics Lab (Optional, Paid)

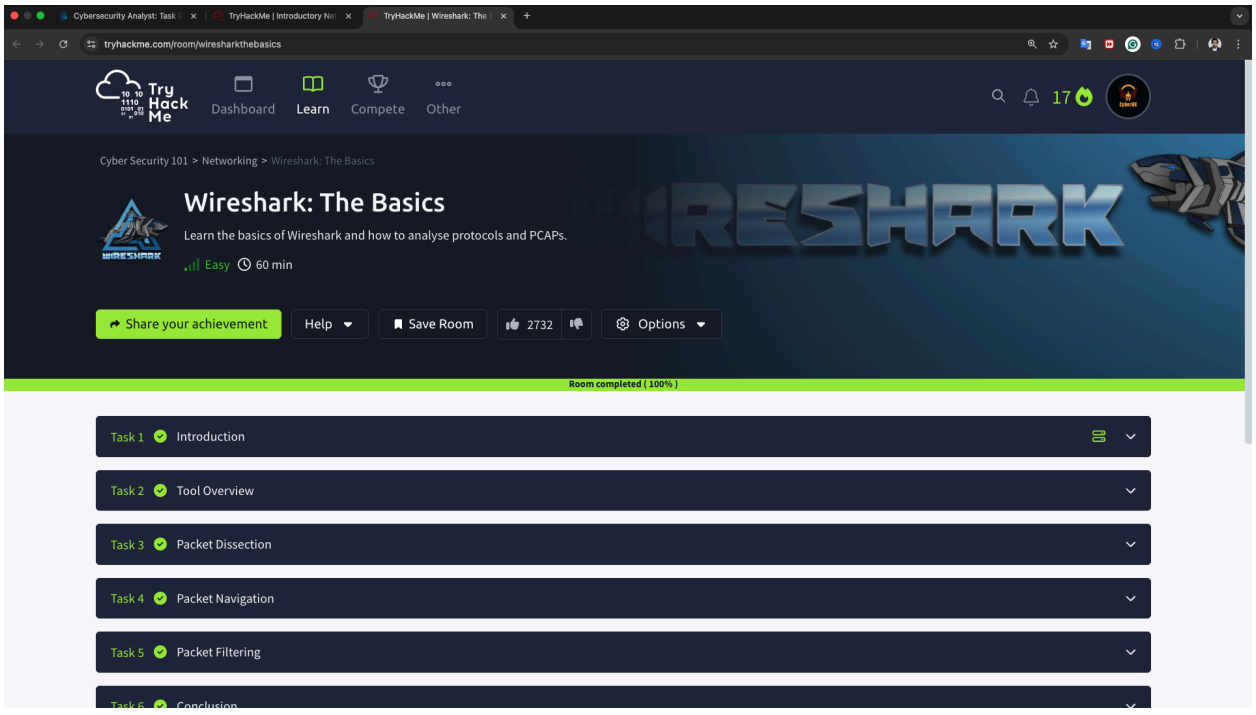
The **Wireshark Basics** room introduces the fundamentals of **Wireshark**, a powerful tool for network traffic analysis. It covers:

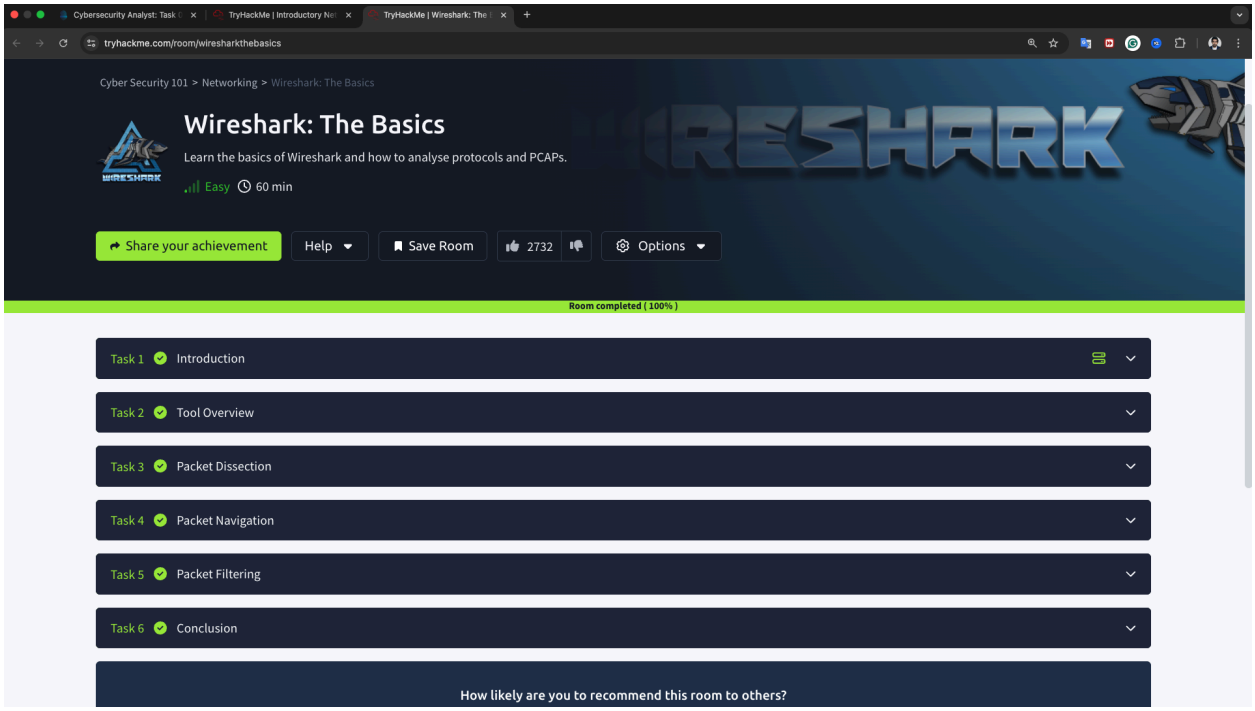
- Capturing network packets in real-time.
- Analyzing protocol behavior such as DNS, TCP, and HTTP.
- Identifying network anomalies and security vulnerabilities.

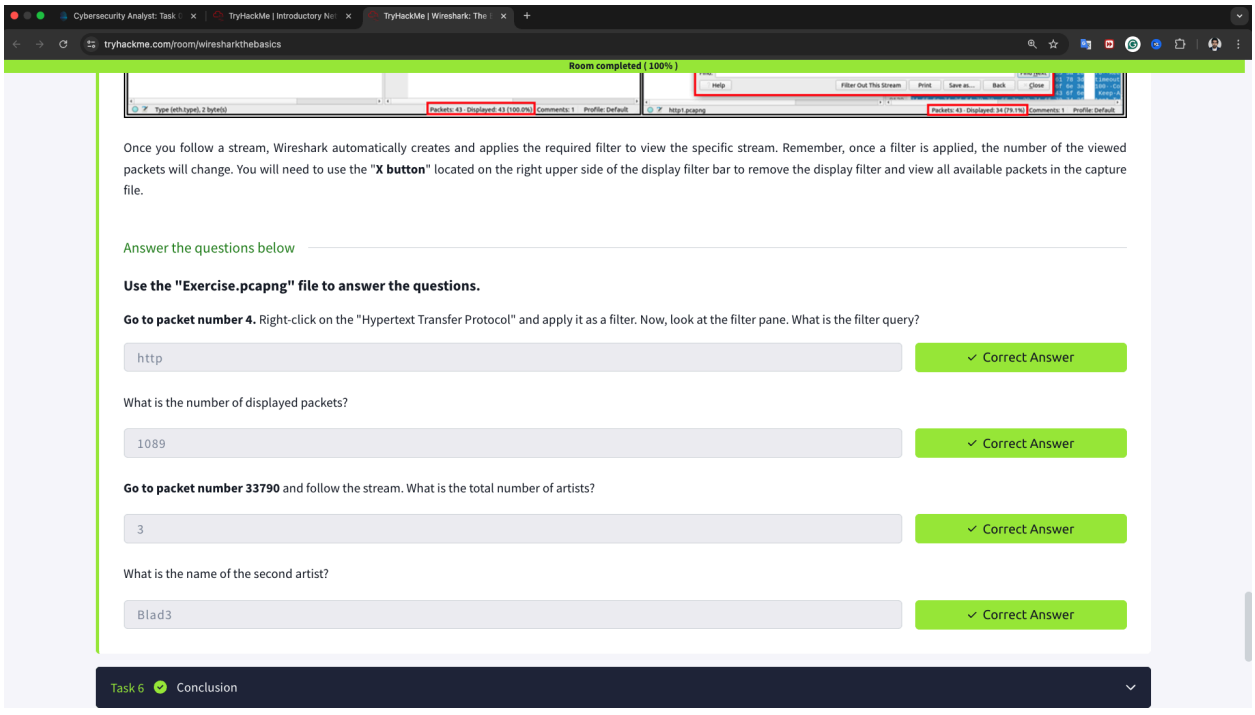
Hands-on Activities & Findings

- Captured live network traffic and examined packet structures.
- Analyzed DNS queries, TCP handshakes, and HTTP requests using Wireshark.
- Identified potential network vulnerabilities and unencrypted traffic.

Screenshots




tryhackme.com/room/wiresharkthebasics
Cyber Security 101 > Networking > Wireshark: The Basics
Wireshark: The Basics
Learn the basics of Wireshark and how to analyse protocols and PCAPs.
Easy
60 min
Share your achievement
Help
Save Room
2732
Options
Room completed (100%)
Task 1 Introduction
Task 2 Tool Overview
Task 3 Packet Dissection
Task 4 Packet Navigation
Task 5 Packet Filtering
Task 6 Conclusion
How likely are you to recommend this room to others?


tryhackme.com/room/wiresharkthebasics
Room completed (100%)
Once you follow a stream, Wireshark automatically creates and applies the required filter to view the specific stream. Remember, once a filter is applied, the number of the viewed packets will change. You will need to use the "X button" located on the right upper side of the display filter bar to remove the display filter and view all available packets in the capture file.
Answer the questions below
Use the "Exercise.pcapng" file to answer the questions.
Go to packet number 4. Right-click on the "Hypertext Transfer Protocol" and apply it as a filter. Now, look at the filter pane. What is the filter query?
http
Correct Answer
What is the number of displayed packets?
1089
Correct Answer
Go to packet number 33790 and follow the stream. What is the total number of artists?
3
Correct Answer
What is the name of the second artist?
Blad3
Correct Answer
Task 6 Conclusion

EncryptEdge Labs

By completing these labs, I have gained valuable practical skills in networking and network traffic analysis. The Intro to Networking lab solidified my understanding of network fundamentals, while the Wireshark Basics lab provided hands-on experience in analyzing real network traffic. These skills are essential for identifying security threats, troubleshooting network issues, and ensuring secure communication within an organization.

This Internship Task report was developed on [Mar, 19, 2025]

By:

atalmamun@gmail.com