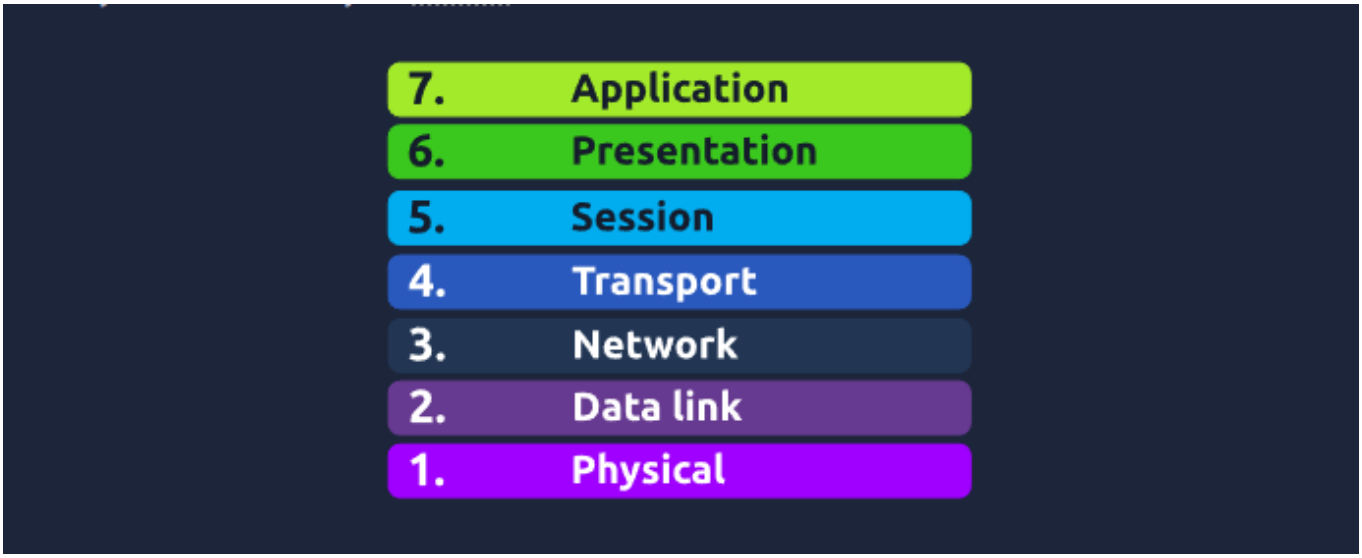
Networking Concepts

OSI Model

defines a framework for computer network communication



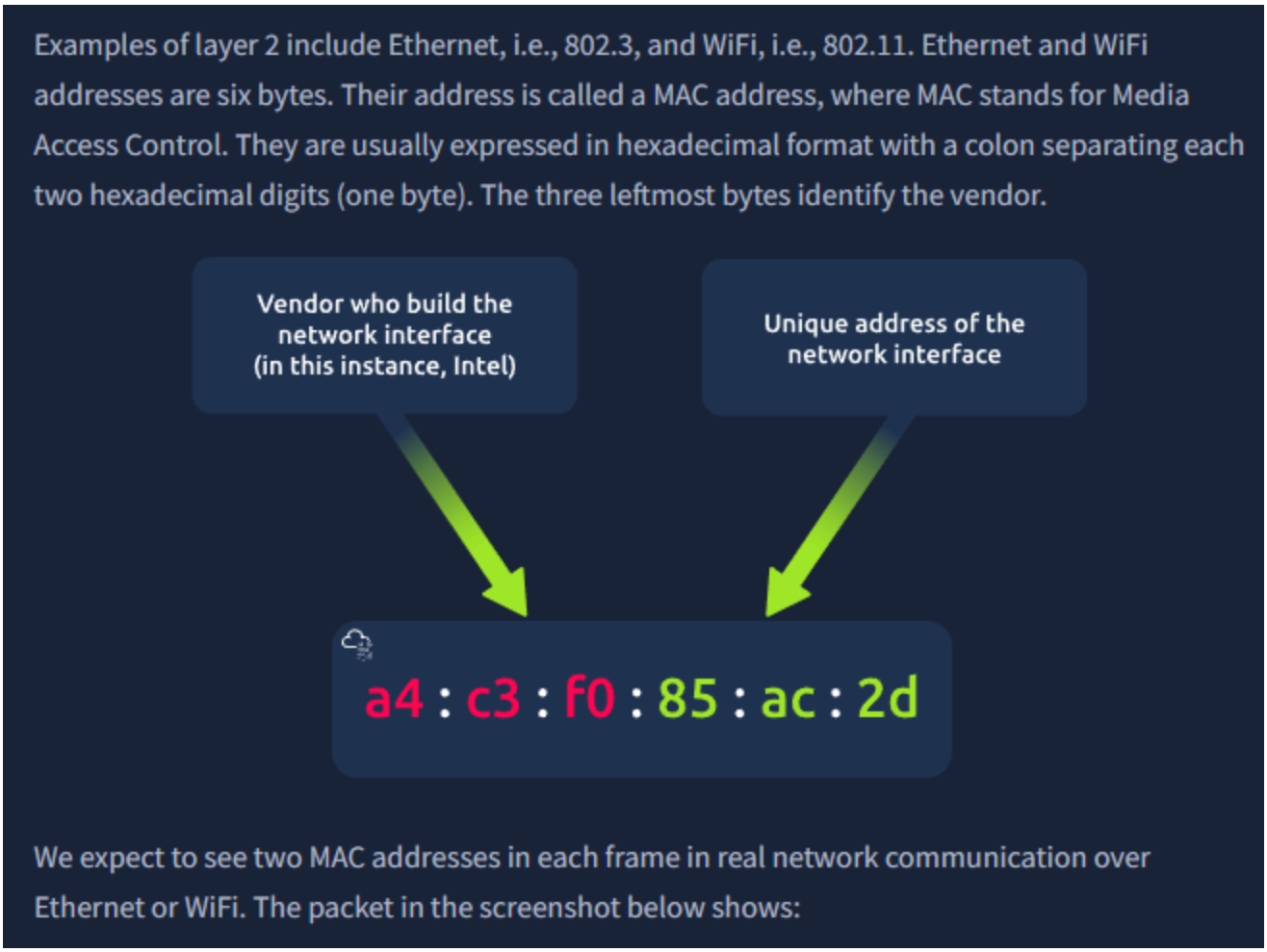
Layer 1: Physical layer

Deals with physical connection between devices this can range from wires, electrical and wireless signals to ethernet cables and wifi radio bands

Layer 2: Data Link Layer

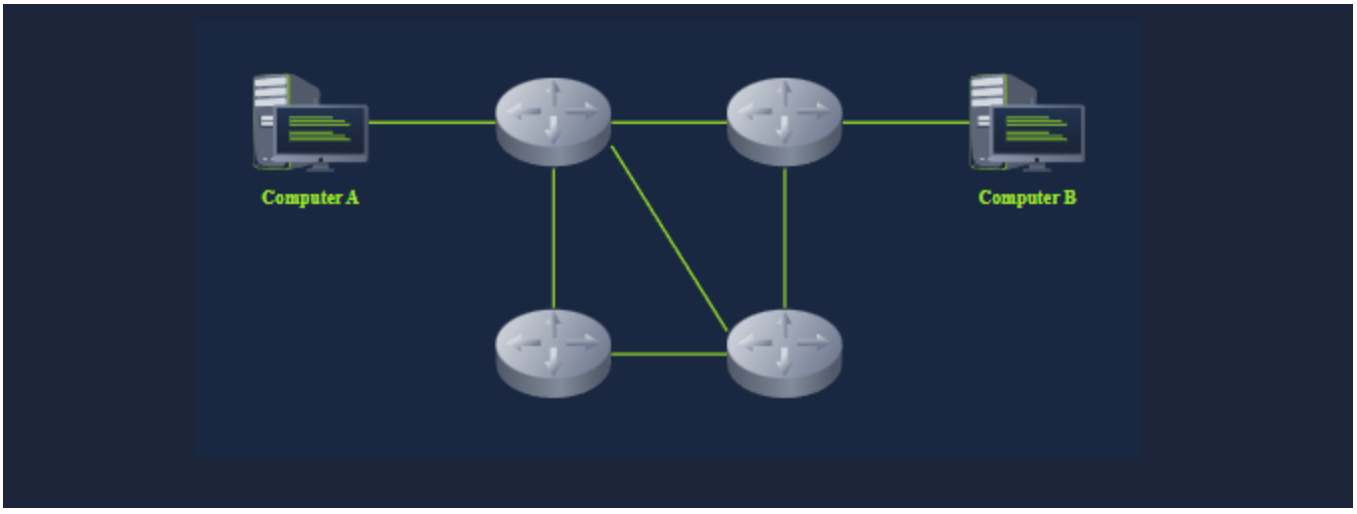
Represents the protocol that enables data transfer between nodes on the same network segment

network segment - group of networked devices using shared medium or channel for data transfer



Layer 3: Network Layer

concerned with sending data between different networks. Handles logical addressing and routing. Network layer will route the network packets through the path it deems better.



Examples of the network layer include Internet Protocol (IP), Internet Control Message Protocol (ICMP), and Virtual Private Network (VPN) protocols such as IPSec and SSL/TLS VPN.

Layer 4: Transport Layer

Enables end-to-end communication between running applications on different hosts.

Examples of layer 4 are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

Layer 5: Session Layer

Responsible for establishing, maintaining, and synchronising communication between applications running on different hosts. Data synchronisation ensures that data is transmitted in the correct order and provides mechanisms for recovery in case of transmission failures.

Examples of the session layer are Network File System (NFS) and Remote Procedure Call (RPC).

Layer 6: Presentation Layer

Ensures the data is delivered in a form the application layer can understand. Layer 6 handles data encoding, compression, and encryption. An example of encoding is character encoding, such as ASCII or Unicode.

Layer 7: Application Layer

Provides network services directly to end-user applications. Your web browser would use the HTTP protocol to request a file, submit a form, or upload a file.

protocols are HTTP, FTP, DNS, POP3, SMTP, and IMAP.

Layer Number	Layer Name	Main Function	Example Protocols and Standards
Layer 7	Application layer	Providing services and interfaces to applications	HTTP, FTP, DNS, POP3, SMTP, IMAP
Layer 6	Presentation layer	Data encoding, encryption, and compression	Unicode, MIME, JPEG, PNG, MPEG
Layer 5	Session layer	Establishing, maintaining, and synchronising sessions	NFS, RPC
Layer 4	Transport layer	End-to-end communication and data segmentation	UDP, TCP
Layer 3	Network layer	Logical addressing and routing between	IP, ICMP, IPSec

Layer Number	Layer Name	Main Function	Example Protocols and Standards
		networks	
Layer 2	Data link layer	Reliable data transfer between adjacent nodes	Ethernet (802.3), WiFi (802.11)
Layer 1	Physical layer	Physical data transmission media	Electrical, optical, and wireless signals

TCP/IP Model

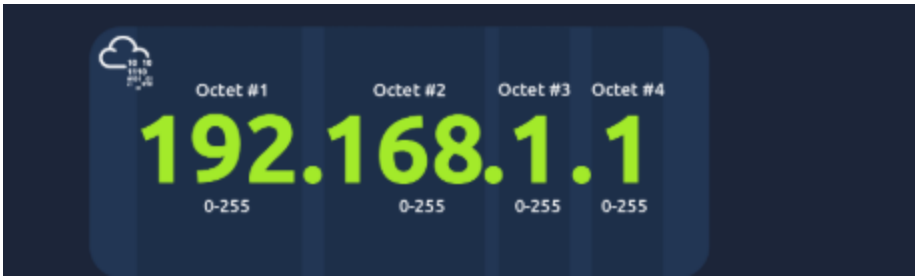
- **Application Layer**: The OSI model application, presentation and session layers, i.e., layers 5, 6, and 7, are grouped into the application layer in the TCP/IP model.
- **Transport Layer**: This is layer 4.
- **Internet Layer**: This is layer 3. The OSI model's network layer is called the Internet layer in the TCP/IP model.
- **Link Layer**: This is layer 2.

Layer Number	ISO OSI Model	TCP/IP Model (RFC 1122)	Protocols
7	Application Layer	Application Layer	HTTP, HTTPS, FTP, POP3, SMTP, IMAP, Telnet, SSH,
6	Presentation Layer		
5	Session Layer		
4	Transport Layer	Transport Layer	TCP, UDP
3	Network Layer	Internet Layer	IP, ICMP, IPSec
2	Data Link Layer	Link Layer	Ethernet 802.3, WiFi 802.11
1	Physical Layer		

- Application
- Transport
- Network
- Link
- Physical

IP Addresses and Subnets

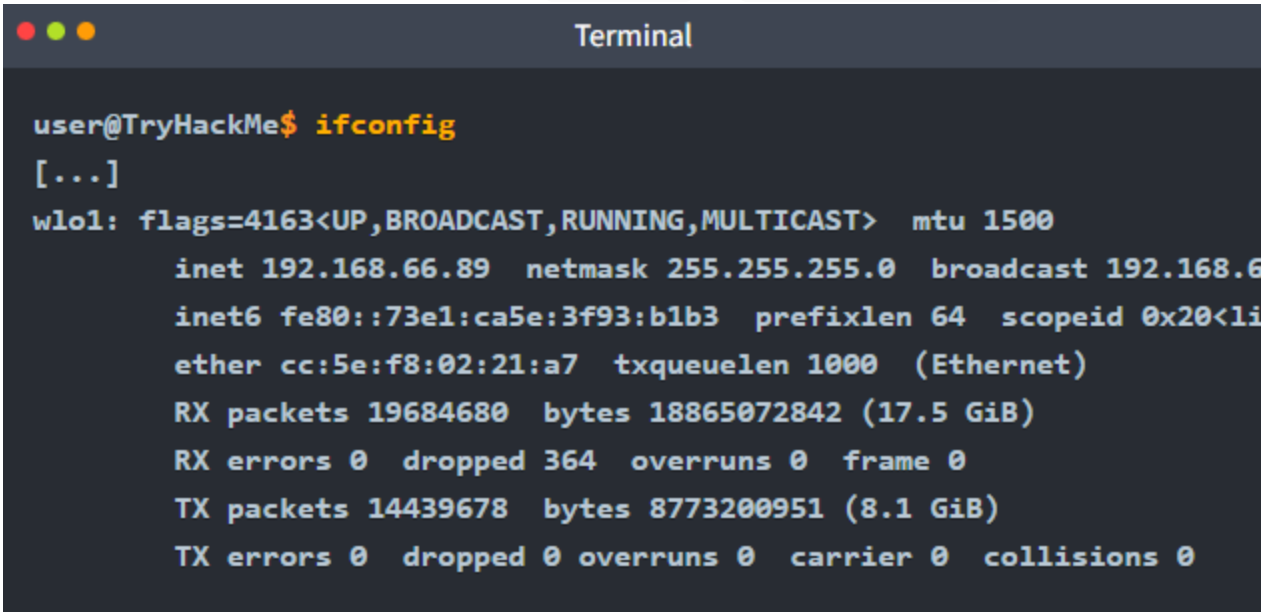
An IP address comprises four octets, i.e., 32 bits. Being 8 bits, an octet allows us to represent a decimal number between 0 and 255.



the 0 and 255 are reserved for the network and broadcast addresses, respectively. In other words, `192.168.1.0` is the network address, while `192.168.1.255` is the broadcast address. Sending to the broadcast address targets all the hosts on the network.

Looking Up Your Network Configuration

MS Windows command line using the command `ipconfig`. On Linux and UNIX-based systems, you can issue the command `ifconfig` or `ip address show`



- The host (laptop) IP address is `192.168.66.89`
- The subnet mask is `255.255.255.0`
- The broadcast address is `192.168.66.255`

a subnet mask of `255.255.255.0` can also be written as `/24`. The `/24` means that the leftmost 24 bits within the IP address do not change across the network, i.e., the subnet. In other words, the leftmost three octets are the same across the whole subnet; therefore, we can expect to find addresses that range from `192.168.66.1` to `192.168.66.254`

Private Addresses

- Public IP addresses
- Private IP addresses

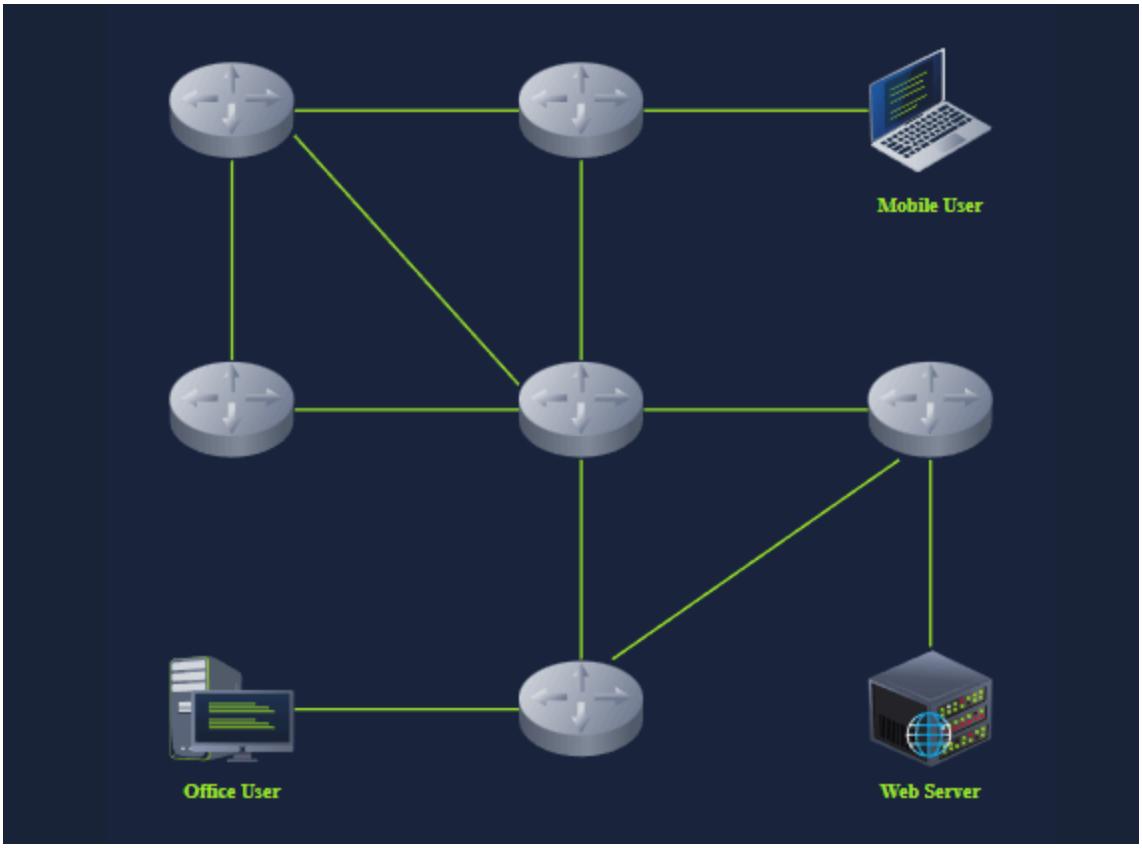
RFC 1918 defines the following three ranges of private IP addresses:

- `10.0.0.0` - `10.255.255.255` (`10/8`)
- `172.16.0.0` - `172.31.255.255` (`172.16/12`)
- `192.168.0.0` - `192.168.255.255` (`192.168/16`)

For a private IP address to access the Internet, the router must have a public IP address and must support Network Address Translation (NAT)

Routing

a router forwards data packets to the proper network. Usually, a data packet passes through multiple routers before it reaches its final destination. inspecting the IP address and forwarding the packet to the best network (router) so the packet gets closer to its destination.



UDP and TCP

UDP (User Datagram Protocol)

- UDP is a simple connectionless protocol that operates at the transport layer.
- does not even provide a mechanism to know that the packet has been delivered.

- there is no guarantee that the UDP packet has been received successfully.
- better speed than a transport protocol that provides “confirmation.”

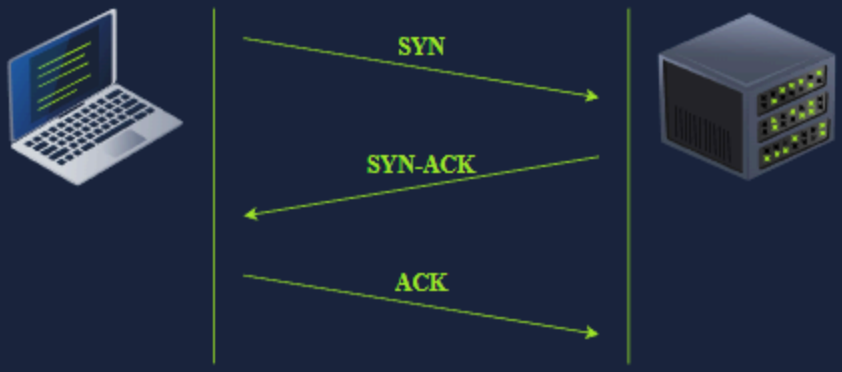
TCP (Transmission Control Protocol)

- connection-oriented transport protocol
- uses various mechanisms to ensure reliable data delivery
- requires the establishment of a TCP connection before any data can be sent

In TCP, each data octet has a sequence number; this makes it easy for the receiver to identify lost or duplicated packets. The receiver, on the other hand, acknowledges the reception of data with an acknowledgement number specifying the last received octet.

A TCP connection is established using what's called a three-way handshake. Two flags are used: SYN (Synchronise) and ACK (Acknowledgment). The packets are sent as follows:

1. SYN Packet: The client initiates the connection by sending a SYN packet to the server. This packet contains the client's randomly chosen initial sequence number.
2. SYN-ACK Packet: The server responds to the SYN packet with a SYN-ACK packet, which adds the initial sequence number randomly chosen by the server.
3. ACK Packet: The three-way handshake is completed as the client sends an ACK packet to acknowledge the reception of the SYN-ACK packet.



Encapsulation

the process of every layer adding a header (and sometimes a trailer) to the received unit of data and sending the “encapsulated” unit to the layer below.

Encapsulation is an essential concept as it allows each layer to focus on its intended function. In the image below, we have the following four steps:

- **Application data**: It all starts when the user inputs the data they want to send into the application. For example, you write an email or an instant message and hit the send button. The application formats this data and starts sending it according to the application protocol used, using the layer below it, the transport layer.
- **Transport protocol segment or datagram**: The transport layer, such as TCP or UDP, adds the proper header information and creates the TCP **segment** (or UDP **datagram**). This segment is sent to the layer below it, the network layer.
- **Network packet**: The network layer, i.e. the Internet layer, adds an IP header to the received TCP segment or UDP datagram. Then, this IP **packet** is sent to the layer below it, the data link layer.
- **Data link frame**: The Ethernet or WiFi receives the IP packet and adds the proper header and trailer, creating a **frame**.

The Life of a Packet

Based on what we have studied so far, we can explain a *simplified version* of the packet's life. Let's consider the scenario where you search for a room on TryHackMe.

1. On the TryHackMe search page, you enter your search query and hit enter.
2. Your web browser, using HTTPS, prepares an HTTP request and pushes it to the layer below it, the transport layer.
3. The TCP layer needs to establish a connection via a three-way handshake between your browser and the TryHackMe web server. After establishing the TCP connection, it can send the HTTP request containing the search query. Each TCP segment created is sent to the layer below it, the Internet layer.
4. The IP layer adds the source IP address, i.e., your computer, and the destination IP address, i.e., the IP address of the TryHackMe web server. For this packet to reach the router, your laptop delivers it to the layer below it, the link layer.
5. Depending on the protocol, The link layer adds the proper link layer header and trailer, and the packet is sent to the router.
6. The router removes the link layer header and trailer, inspects the IP destination, among other fields, and routes the packet to the proper link. Each router repeats this process until it reaches the router of the target server.

Telnet

The TELNET (Teletype Network) protocol is a network protocol for remote terminal connection.

we can use `telnet` to connect to any server listening on a TCP port number.

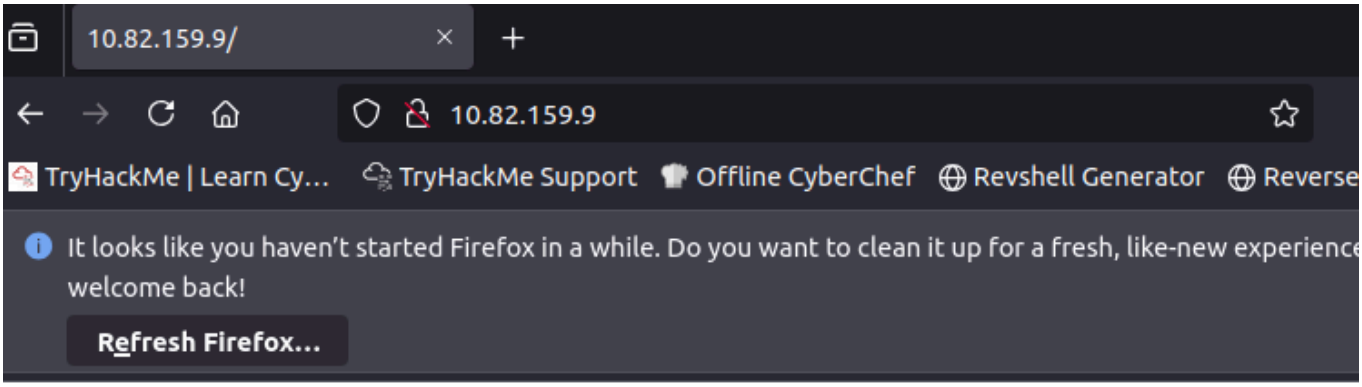
requesting a web page using telnet - After connecting to port 80, you need to issue the command `GET / HTTP/1.1` and identify the host where anything goes, such as `Host: telnet.thm`

```
root@ip-10-82-92-127:~# telnet 10.82.159.9 80
Trying 10.82.159.9...
Connected to 10.82.159.9.
Escape character is '^]'.

HTTP/1.0 400 Bad Request
Content-Type: text/html
Content-Length: 345
Connection: close
Date: Thu, 08 Jan 2026 17:30:05 GMT
Server: lighttpd/1.4.63

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <title>400 Bad Request</title>
 </head>
 <body>
  <h1>400 Bad Request</h1>
 </body>
</html>
```

I then went onto the browser and typed the address in the url and got the flag:



THM{TELNET_MASTER}