WebServe & Sniff

Name/ Mohamed Khaled Mahmoud
Date/ 11/08/2024
University/ Arab Open University (AOU)
Major/ Network & Security

WebServe & Sniff: Building and Analyzing an Apache-Hosted Hello World Site

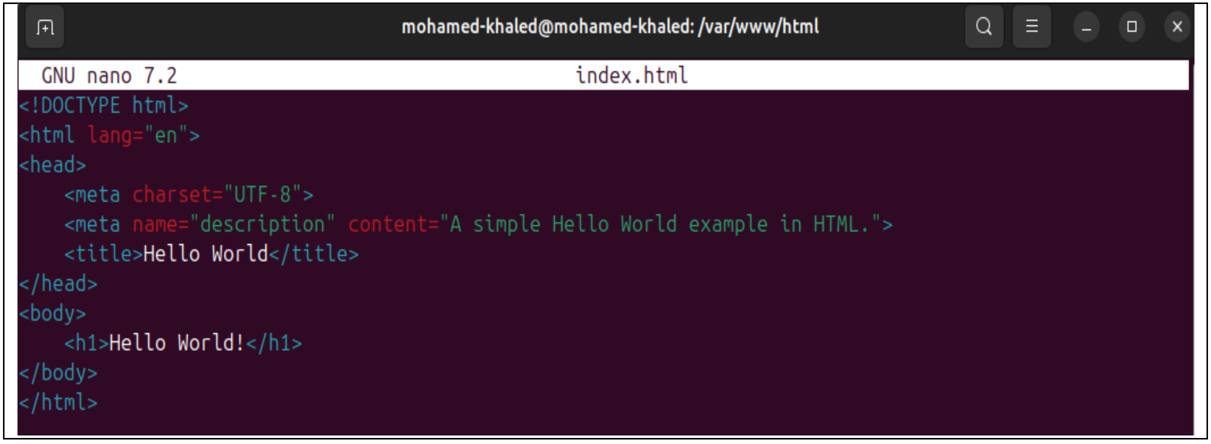
Objective

The goal of this project is to set up an Apache web server on Ubuntu, host a simple **'Hello World'** website, and analyze the *network traffic*, focusing on the HTTP protocol using **Wireshark** on a Windows machine.

1. Setting Up Apache Web Server on Ubuntu:

Steps and Resources

1. **Research and Installation**:
   - I started by updating the package list on my **Ubuntu** machine:
     - sudo apt-get update
     - sudo apt-get upgrade
   - Then, I installed Apache2 using the following command:
     - sudo apt-get install apache2
   - After the installation, I started the Apache service and enabled it to start on boot:
     - sudo systemctl start apache2
     - sudo systemctl enable apache2
2. **Creating the 'Hello World' Page**:
   - I navigated to the default web directory **/var/www/html/** and created an index.html file:
     - cd /var/www/html
     - sudo nano index.html
   - I added the following simple HTML code:



```
  GNU nano 7.2                              index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="description" content="A simple Hello World example in HTML.">
    <title>Hello World</title>
</head>
<body>
    <h1>Hello World!</h1>
</body>
</html>
```

3. **Verification**:
   - I found the IP address of my Ubuntu is using <*hostname -I*> command:
     - http://192.168.1.10
   - I then accessed the server from a ***web browser*** on my Windows machine by entering the Ubuntu server's IP address. The **'Hello World'** page was successfully displayed.

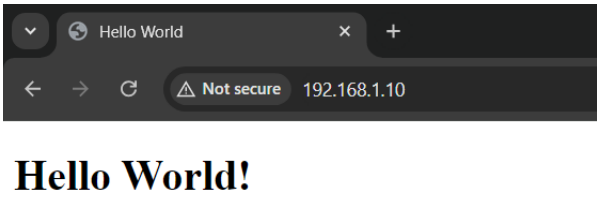
2. Accessing the Website from Windows Host Machine

Procedure

1. **Finding the Ubuntu Server’s IP**:
   - The IP address was found using the <*hostname -I*>command on Ubuntu.

2. **Connecting to the Server**:
   - On my Windows machine, I opened a web browser and entered the IP address http://192.168.10.1. The **'Hello World'** page was successfully loaded.

Screenshot:



Hello World!

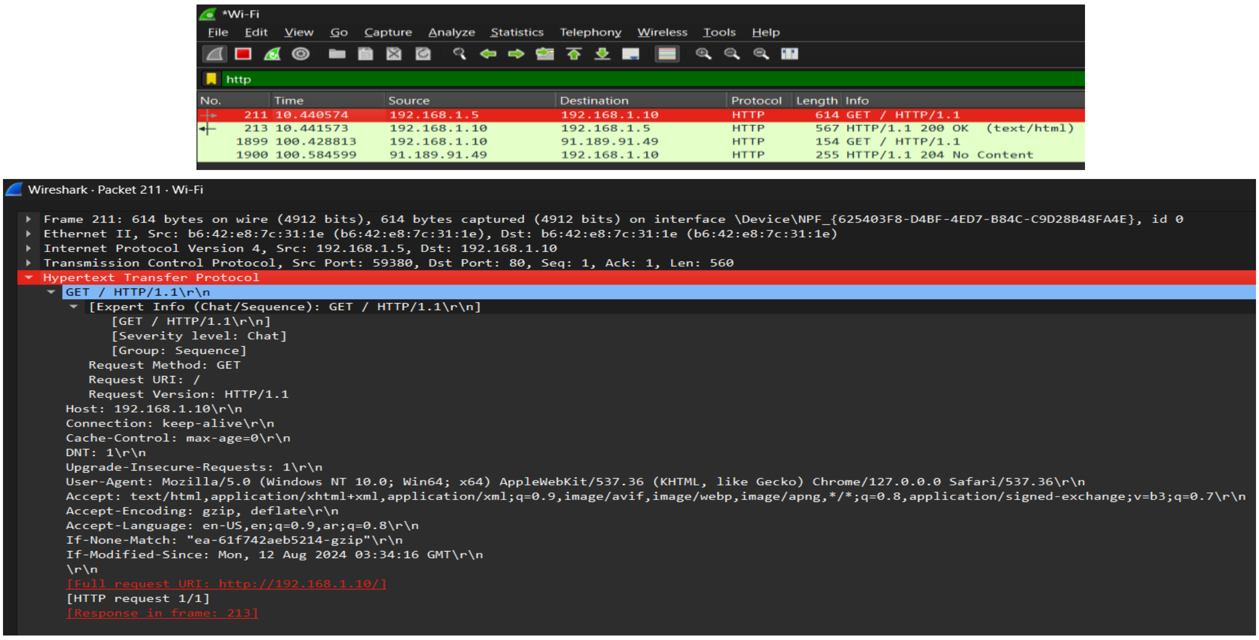
3. Analyzing HTTP Protocol using Wireshark on Windows Steps

1. **Installing Wireshark**:
   - I downloaded and installed **Wireshark** from the official website: https://www.wireshark.org/download.html

2. **Capturing Network Traffic**:
   - I opened **Wireshark** and started a new capture on the network interface that was connected to the same network as my **Ubuntu** machine.
   - While **Wireshark** was capturing traffic, I reloaded the **'Hello World'** webpage from the Windows browser.

3. **HTTP GET Request Analysis**:
   - I filtered the captured data using **http** to focus on HTTP traffic.
   - I identified the **HTTP GET** request for the ***'Hello World'*** page and examined the request and response headers.

Screenshot:


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Wireshark · Packet 211 · Wi-Fi

```
No.   Time        Source         Destination    Protocol Length Info
  211 10.440574   192.168.1.5    192.168.1.10   HTTP      614 GET / HTTP/1.1
  213 10.441573   192.168.1.10   192.168.1.5    HTTP      567 HTTP/1.1 200 OK  (text/html)
 1899 100.428813  192.168.1.10   91.189.91.49   HTTP      154 GET / HTTP/1.1
 1900 100.584599  91.189.91.49   192.168.1.10   HTTP      255 HTTP/1.1 204 No Content
```

```
Frame 211: 614 bytes on wire (4912 bits), 614 bytes captured (4912 bits) on interface \Device\NPF_{625403F8-D4BF-4ED7-B84C-C9D28B48FA4E}, id 0
Ethernet II, Src: b6:42:e8:7c:31:1e (b6:42:e8:7c:31:1e), Dst: b6:42:e8:7c:31:1e (b6:42:e8:7c:31:1e)
Internet Protocol Version 4, Src: 192.168.1.5, Dst: 192.168.1.10
Transmission Control Protocol, Src Port: 59380, Dst Port: 80, Seq: 1, Ack: 1, Len: 560
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      [GET / HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /
    Request Version: HTTP/1.1
  Host: 192.168.1.10\r\n
  Connection: keep-alive\r\n
  Cache-Control: max-age=0\r\n
  DNT: 1\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-US,en;q=0.9,ar;q=0.8\r\n
  If-None-Match: "ea-61f742aeb5214-gzip"\r\n
  If-Modified-Since: Mon, 12 Aug 2024 03:34:16 GMT\r\n
  \r\n
  [Full request URI: http://192.168.1.10/]
  [HTTP request 1/1]
  [Response in frame: 213]
```

Insights

- **Understanding HTTP Protocol**:
  - The GET request displayed the method, the requested URI (**'/index.html'**), and various headers such as **'Host'** and **'User-Agent'**.

  - The server's HTTP response included the status code (**'200 OK'**), content-type (**'text/html'**), and the body of the response, which was the HTML content of the 'Hello World' page.

Challenges Faced

- **Firewall Configuration**:
  - Initially, I was unable to access the server from my Windows machine due to firewall settings on the Ubuntu machine. I resolved this by allowing HTTP traffic through the firewall:
    - sudo ufw allow 'Apache'

- **Wireshark Interface Selection**:
  - Selecting the correct network interface in **Wireshark** was confusing at first. I overcame this by identifying the active interface using the Windows command:
    - ipconfig

Conclusion

This task provided hands-on experience in setting up a basic ***web server***, hosting a webpage, and analyzing **HTTP traffic**. The process enhanced my understanding of **web server configuration** and the intricacies of the HTTP protocol, as well as the importance of network traffic analysis in troubleshooting and security.