MALWARE ANALYSIS (Final Project report)

WIRESHARK PACKET SNIFFING

Naveen Kumar Jakkula | Soundarya Kasarla | Nikitha Chintala

INTRODUCTION

This study aims to investigate how web phishing might be identified and avoided using Wireshark, a well-known network protocol analyzer tool. We seek to discover trends and behaviors related to web phishing assaults through the capture and analysis of network traffic. Then, using data from the real world, we will assess how well these strategies worked.

A potent network protocol analyzer called Wireshark enables you to record and examine network data in real-time. The capacity to do packet sniffing, which is the act of intercepting and studying the network packets that are carried over a network, is one of the core aspects of Wireshark.

Sniffing packets can be employed for a variety of tasks, such as protocol creation, security research, and network troubleshooting. You may learn more about how your network behaves, spot possible difficulties and bottlenecks, and pinpoint the underlying cause of network issues by collecting and analyzing network packets.

In this article, we'll examine Wireshark packet sniffing in more detail and demonstrate how to utilize this potent tool to record and examine network data. Wireshark packet sniffing is a crucial ability that may help you better understand your network and enhance its efficiency and security, regardless of whether you work as a network administrator, security expert, or software developer. A report comparing HTTP, FTP, Telnet, and SSH will be delivered. Finding the protocol that is secure to use for both professional and personal data without any malicious effort to acquire the data would be the result.

TERMINOLOGY

Wireshark Tool: Wireshark is a well-known network protocol analyzer tool that is employed for real-time network traffic capture and analysis. Users can decode different communication protocols and observe and capture network packets. Network troubleshooting, security analysis, and protocol development may all be done with Wireshark.

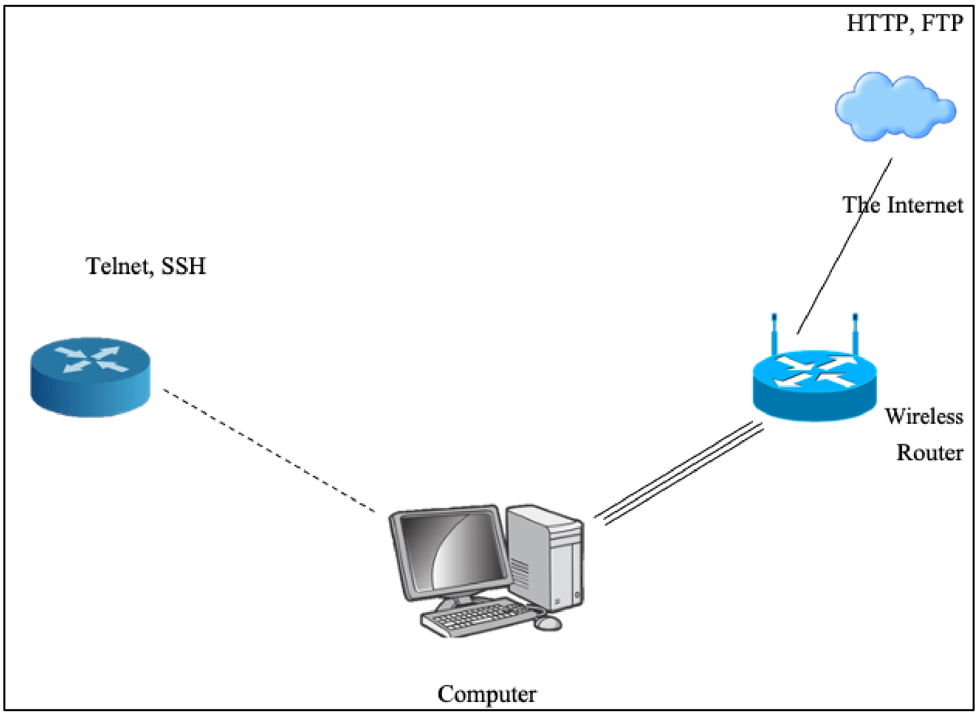
Web Phishing: Web phishing is a sort of cyberattack that uses phony websites or emails to persuade people to provide personal data, including usernames, passwords, and credit card numbers. In order to get the user to respond fast and give the desired information, phishing attempts frequently utilize social engineering tactics to instill a sense of urgency or fear.

Packet Sniffing: The act of capturing and examining data packets as they go through a network is known as packet sniffing. The source and destination addresses, the protocols being used, and the data payload may all be examined by using packet sniffers to capture and analyse data packets.

Ethical Hacking: Ethical hacking is a sort of security testing that involves attempting to get into a computer or network system in order to find flaws and vulnerabilities that an attacker may use against you. The majority of the time, ethical hacking is carried out by qualified experts who are permitted to carry out such testing.

Putty: PuTTY is a network file transfer programme, serial console, and terminal emulator that is free and open source. Such network protocols as SSH, Telnet, rlogin, and raw socket connections are supported. System administrators and developers that need to remotely access servers, network devices, and other systems utilizing secure protocols like SSH frequently choose PuTTY. With its support for several protocols, it is a flexible tool for controlling network infrastructure. It offers a straightforward and lightweight interface for connecting to distant computers and executing instructions.

METHODOLOGY:



Discussing in this report are Telnet, FTP, SSH and HTTP.

1. **Telnet:**

   Telnet is a network protocol used for remote access to network devices. Unlike SSH, Telnet does not encrypt data, making it less secure than SSH. Telnet is commonly used for remote management of network devices such as routers and switches.

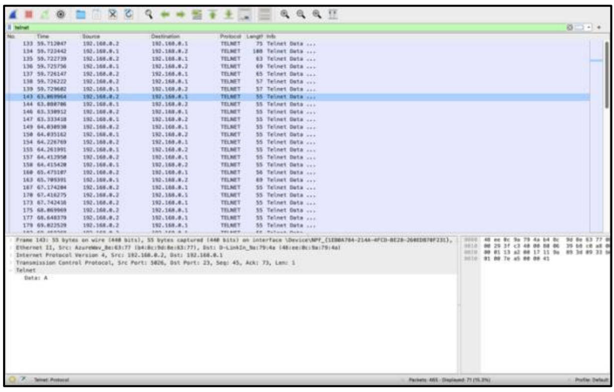
   This is the information we used in putty to use telnet. And using Wireshark we sniffed the information. Below are the attached screenshots.

User Access Verification

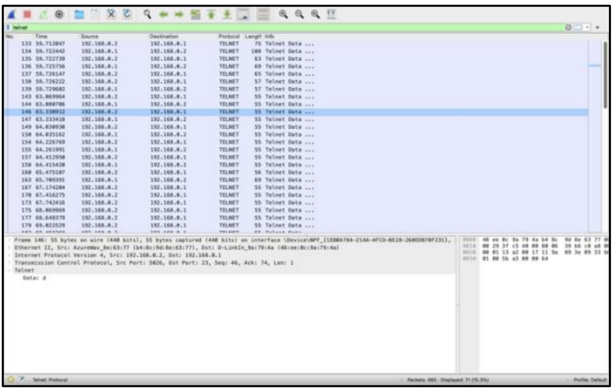
Username: ....P...... ....'.......XTERM....$..$AAddmmiinn

Password: 12345678

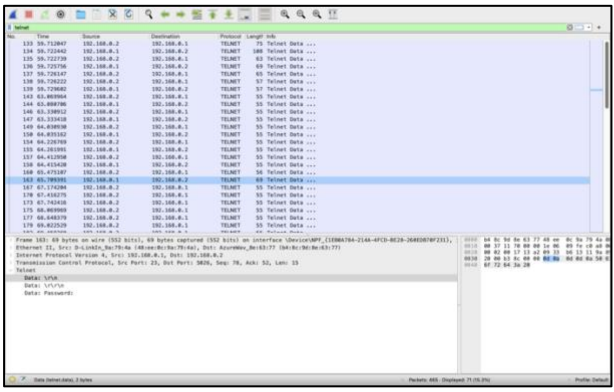
Screenshots from Wireshark:



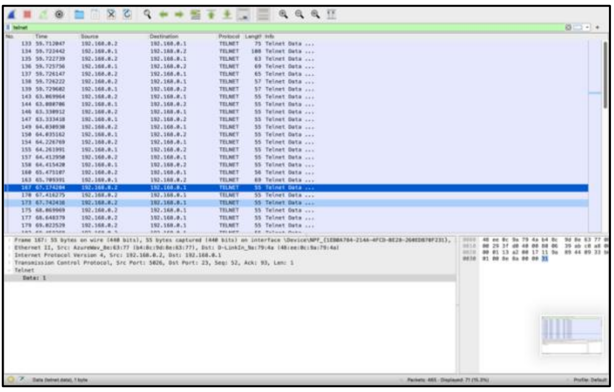
```
telnet
No.   Time       Source        Destination   Protocol Length Info
  133 59.712047  192.168.0.2   192.168.0.1   TELNET    75 Telnet Data ...
  134 59.723442  192.168.0.1   192.168.0.2   TELNET   100 Telnet Data ...
  135 59.722739  192.168.0.2   192.168.0.1   TELNET    63 Telnet Data ...
  136 59.725756  192.168.0.1   192.168.0.2   TELNET    69 Telnet Data ...
  137 59.726147  192.168.0.2   192.168.0.1   TELNET    65 Telnet Data ...
  138 59.726222  192.168.0.2   192.168.0.1   TELNET    57 Telnet Data ...
  139 59.729602  192.168.0.1   192.168.0.2   TELNET    57 Telnet Data ...
  143 63.069964  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  144 63.090706  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  146 63.330912  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  147 63.333418  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  149 64.030938  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  150 64.035162  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  154 64.226769  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  155 64.261991  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  157 64.412950  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  158 64.415420  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  160 65.475107  192.168.0.2   192.168.0.1   TELNET    56 Telnet Data ...
  163 65.709391  192.168.0.1   192.168.0.2   TELNET    69 Telnet Data ...
  167 67.174204  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  170 67.416275  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  173 67.742438  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  175 68.069969  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  177 68.648379  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  179 69.022529  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...

> Frame 143: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{1E0BA784-214A-4FCD-8E28-260ED878F231},
> Ethernet II, Src: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77), Dst: D-LinkIn_9a:79:4a (48:ee:0c:9a:79:4a)
> Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1
> Transmission Control Protocol, Src Port: 5026, Dst Port: 23, Seq: 45, Ack: 73, Len: 1
v Telnet
    Data: A

0000  48 ee 0c 9a 79 4a b4 8c  9d 8e 63 77 08
0010  00 29 3f c3 40 00 80 06  39 b0 c0 a8 0
0020  00 01 13 a2 00 17 11 9a  09 3d 09 33 b
0030  01 00 7e a5 00 00 41

Telnet Protocol          Packets: 465 · Displayed: 71 (15.3%)          Profile: Default
```



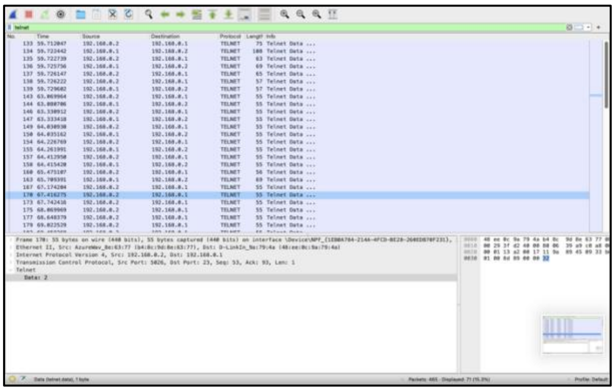
```
telnet
No.   Time       Source        Destination   Protocol Length Info
  133 59.712047  192.168.0.2   192.168.0.1   TELNET    75 Telnet Data ...
  134 59.723442  192.168.0.1   192.168.0.2   TELNET   100 Telnet Data ...
  135 59.722739  192.168.0.2   192.168.0.1   TELNET    63 Telnet Data ...
  136 59.725756  192.168.0.1   192.168.0.2   TELNET    69 Telnet Data ...
  137 59.726147  192.168.0.2   192.168.0.1   TELNET    65 Telnet Data ...
  138 59.726222  192.168.0.2   192.168.0.1   TELNET    57 Telnet Data ...
  139 59.729602  192.168.0.1   192.168.0.2   TELNET    57 Telnet Data ...
  143 63.069964  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  144 63.090706  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  146 63.330912  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  147 63.333418  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  149 64.030938  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  150 64.035162  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  154 64.226769  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  155 64.261991  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  157 64.412950  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  158 64.415420  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  160 65.475107  192.168.0.2   192.168.0.1   TELNET    56 Telnet Data ...
  163 65.709391  192.168.0.1   192.168.0.2   TELNET    69 Telnet Data ...
  167 67.174204  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  170 67.416275  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  173 67.742438  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  175 68.069969  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  177 68.648379  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  179 69.022529  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...

> Frame 146: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{1E0BA784-214A-4FCD-8E28-260ED878F231},
> Ethernet II, Src: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77), Dst: D-LinkIn_9a:79:4a (48:ee:0c:9a:79:4a)
> Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1
> Transmission Control Protocol, Src Port: 5026, Dst Port: 23, Seq: 46, Ack: 74, Len: 1
v Telnet
    Data: d

0000  48 ee 0c 9a 79 4a b4 8c  9d 8e 63 77 08
0010  00 29 3f c5 40 00 80 06  39 b6 c0 a8 0
0020  00 01 13 a2 00 17 11 9a  09 3e 09 33 b
0030  01 00 5b a3 00 00 64

Telnet Protocol          Packets: 465 · Displayed: 71 (15.3%)          Profile: Default
```



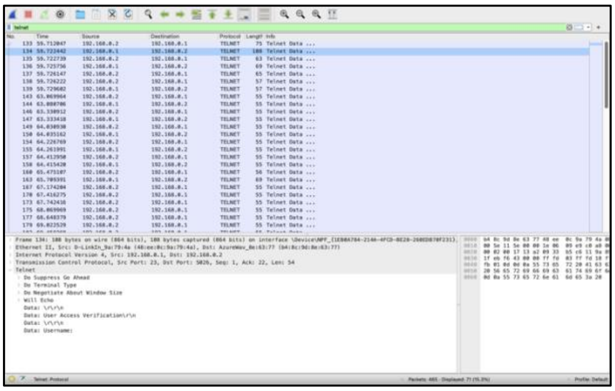
```
telnet
No.   Time       Source        Destination   Protocol Length Info
  133 59.712047  192.168.0.2   192.168.0.1   TELNET    75 Telnet Data ...
  134 59.723442  192.168.0.1   192.168.0.2   TELNET   100 Telnet Data ...
  135 59.722739  192.168.0.2   192.168.0.1   TELNET    63 Telnet Data ...
  136 59.725756  192.168.0.1   192.168.0.2   TELNET    69 Telnet Data ...
  137 59.726147  192.168.0.2   192.168.0.1   TELNET    65 Telnet Data ...
  138 59.726222  192.168.0.2   192.168.0.1   TELNET    57 Telnet Data ...
  139 59.729602  192.168.0.1   192.168.0.2   TELNET    57 Telnet Data ...
  143 63.069964  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  144 63.090706  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  146 63.330912  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  147 63.333418  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  149 64.030938  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  150 64.035162  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  154 64.226769  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  155 64.261991  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  157 64.412950  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  158 64.415420  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  160 65.475107  192.168.0.2   192.168.0.1   TELNET    56 Telnet Data ...
  163 65.709391  192.168.0.1   192.168.0.2   TELNET    69 Telnet Data ...
  167 67.174204  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  170 67.416275  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  173 67.742438  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  175 68.069969  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  177 68.648379  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  179 69.022529  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...

> Frame 163: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface \Device\NPF_{1E0BA784-214A-4FCD-8E28-260ED878F231},
> Ethernet II, Src: D-LinkIn_9a:79:4a (48:ee:0c:9a:79:4a), Dst: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77)
> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2
> Transmission Control Protocol, Src Port: 23, Dst Port: 5026, Seq: 78, Ack: 52, Len: 15
v Telnet
    Data: \r\n
    Data: \r\r\n
    Data: Password:

0000  b4 8c 9d 8e 63 77 48 ee  0c 9a 79 4a 08
0010  00 37 11 70 00 00 1e 06  09 fe c0 a8 00
0020  00 02 00 17 13 a2 09 33  b6 13 11 9a 0
0030  20 00 b3 8c 00 00 0d 0a  0d 0d 0a 50 6
0040  6f 72 64 3a 20

Data (telnet.data), 2 bytes          Packets: 465 · Displayed: 71 (15.3%)          Profile: Default
```



```
telnet
No.   Time       Source        Destination   Protocol Length Info
  133 59.712047  192.168.0.2   192.168.0.1   TELNET    75 Telnet Data ...
  134 59.723442  192.168.0.1   192.168.0.2   TELNET   100 Telnet Data ...
  135 59.722739  192.168.0.2   192.168.0.1   TELNET    63 Telnet Data ...
  136 59.725756  192.168.0.1   192.168.0.2   TELNET    69 Telnet Data ...
  137 59.726147  192.168.0.2   192.168.0.1   TELNET    65 Telnet Data ...
  138 59.726222  192.168.0.2   192.168.0.1   TELNET    57 Telnet Data ...
  139 59.729602  192.168.0.1   192.168.0.2   TELNET    57 Telnet Data ...
  143 63.069964  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  144 63.090706  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  146 63.330912  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  147 63.333418  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  149 64.030938  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  150 64.035162  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  154 64.226769  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  155 64.261991  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  157 64.412950  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  158 64.415420  192.168.0.1   192.168.0.2   TELNET    55 Telnet Data ...
  160 65.475107  192.168.0.2   192.168.0.1   TELNET    56 Telnet Data ...
  163 65.709391  192.168.0.1   192.168.0.2   TELNET    69 Telnet Data ...
  167 67.174204  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  170 67.416275  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  173 67.742438  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  175 68.069969  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  177 68.648379  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...
  179 69.022529  192.168.0.2   192.168.0.1   TELNET    55 Telnet Data ...

> Frame 167: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{1E0BA784-214A-4FCD-8E28-260ED878F231},
> Ethernet II, Src: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77), Dst: D-LinkIn_9a:79:4a (48:ee:0c:9a:79:4a)
> Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1
> Transmission Control Protocol, Src Port: 5026, Dst Port: 23, Seq: 52, Ack: 93, Len: 1
v Telnet
    Data: 1

0000  48 ee 0c 9a 79 4a b4 8c  9d 8e 63 77 08
0010  00 29 3f d0 40 00 80 06  39 ab c0 a8 0
0020  00 01 13 a2 00 17 11 9a  09 44 09 33 b
0030  01 00 0e 8a 00 00 31

Data (telnet.data), 1 byte          Packets: 465 · Displayed: 71 (15.3%)          Profile: Default
```



```
No.  Time       Source       Destination  Protocol Length Info
 133 59.712047  192.168.0.2  192.168.0.1  TELNET    75 Telnet Data ...
 134 59.722442  192.168.0.1  192.168.0.2  TELNET   108 Telnet Data ...
 135 59.722739  192.168.0.2  192.168.0.1  TELNET    63 Telnet Data ...
 136 59.725756  192.168.0.1  192.168.0.2  TELNET    69 Telnet Data ...
 137 59.726147  192.168.0.2  192.168.0.1  TELNET    65 Telnet Data ...
 138 59.726222  192.168.0.2  192.168.0.1  TELNET    57 Telnet Data ...
 139 59.729682  192.168.0.1  192.168.0.2  TELNET    57 Telnet Data ...
 143 63.069964  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 144 63.080786  192.168.0.1  192.168.0.2  TELNET    55 Telnet Data ...
 146 63.330912  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 147 63.333418  192.168.0.1  192.168.0.2  TELNET    55 Telnet Data ...
 149 64.030930  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 150 64.035162  192.168.0.1  192.168.0.2  TELNET    55 Telnet Data ...
 154 64.226769  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 155 64.261991  192.168.0.1  192.168.0.2  TELNET    55 Telnet Data ...
 157 64.412950  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 158 64.415420  192.168.0.1  192.168.0.2  TELNET    55 Telnet Data ...
 160 65.471107  192.168.0.2  192.168.0.1  TELNET    56 Telnet Data ...
 163 65.705391  192.168.0.1  192.168.0.2  TELNET    69 Telnet Data ...
 167 67.174204  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 170 67.416275  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 173 67.742416  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 175 68.069969  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 177 68.648379  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 179 69.022529  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...

> Frame 170: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{1EB0A784-2146-4FCD-8E28-260ED870F231},
> Ethernet II, Src: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77), Dst: D-LinkIn_9a:79:4a (48:ee:0c:9a:79:4a)
> Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1
> Transmission Control Protocol, Src Port: 5026, Dst Port: 23, Seq: 53, Ack: 93, Len: 1
v Telnet
    Data: 2
```



```
No.  Time       Source       Destination  Protocol Length Info
 133 59.712047  192.168.0.2  192.168.0.1  TELNET    75 Telnet Data ...
 134 59.722442  192.168.0.1  192.168.0.2  TELNET   108 Telnet Data ...
 135 59.722739  192.168.0.2  192.168.0.1  TELNET    63 Telnet Data ...
 136 59.725756  192.168.0.1  192.168.0.2  TELNET    69 Telnet Data ...
 137 59.726147  192.168.0.2  192.168.0.1  TELNET    65 Telnet Data ...
 138 59.726222  192.168.0.2  192.168.0.1  TELNET    57 Telnet Data ...
 139 59.729682  192.168.0.1  192.168.0.2  TELNET    57 Telnet Data ...
 143 63.069964  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 144 63.080786  192.168.0.1  192.168.0.2  TELNET    55 Telnet Data ...
 146 63.330912  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 147 63.333418  192.168.0.1  192.168.0.2  TELNET    55 Telnet Data ...
 149 64.030930  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 150 64.035162  192.168.0.1  192.168.0.2  TELNET    55 Telnet Data ...
 154 64.226769  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 155 64.261991  192.168.0.1  192.168.0.2  TELNET    55 Telnet Data ...
 157 64.412950  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 158 64.415420  192.168.0.1  192.168.0.2  TELNET    55 Telnet Data ...
 160 65.471107  192.168.0.2  192.168.0.1  TELNET    56 Telnet Data ...
 163 65.705391  192.168.0.1  192.168.0.2  TELNET    69 Telnet Data ...
 167 67.174204  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 170 67.416275  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 173 67.742416  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 175 68.069969  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 177 68.648379  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...
 179 69.022529  192.168.0.2  192.168.0.1  TELNET    55 Telnet Data ...

> Frame 134: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface \Device\NPF_{1EB0A784-2146-4FCD-8E28-260ED870F231},
> Ethernet II, Src: D-LinkIn_9a:79:4a (48:ee:0c:9a:79:4a), Dst: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77)
> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2
> Transmission Control Protocol, Src Port: 23, Dst Port: 5026, Seq: 1, Ack: 22, Len: 54
v Telnet
  > Do Suppress Go Ahead
  > Do Terminal Type
  > Do Negotiate About Window Size
  > Will Echo
    Data: \r\r\n
    Data: User Access Verification\r\n
    Data: \r\r\n
    Data: Username:
```

If you can observe data is clearly visible in the Wireshark bit by bit. So, it's an unsafe environment to use.

<u>Drawbacks</u>:

- ⇒ <u>Security Issues:</u> Telnet does not encrypt data, making it less secure than other remote access protocols such as SSH. This means that data transmitted over Telnet can be intercepted and read by unauthorized users.
- ⇒ <u>Authentication Issues</u>: Telnet does not provide strong authentication mechanisms, making it vulnerable to password-based attacks such as brute-force attacks and password sniffing.
- ⇒ <u>Limited Functionality:</u> Telnet has limited functionality compared to other remote access protocols. For example, Telnet does not support file transfer, which makes it less useful for certain applications.
- ⇒ <u>Limited Flexibility:</u> Telnet is not well-suited for complex network configurations or environments where multiple users need to access the same network device simultaneously.

Telnet and SSH are the wired protocols, therefore as we learned about telnet now let's see SSH.

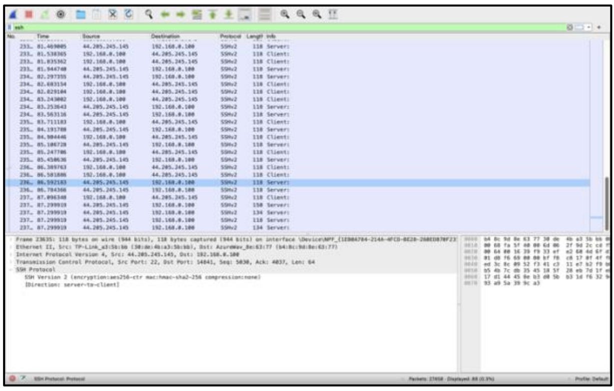
2. **<u>SSH:</u>**

SSH (Secure Shell) is a network protocol used for secure remote access to network devices. SSH encrypts all data transmitted between the client and server, providing a high level of security.

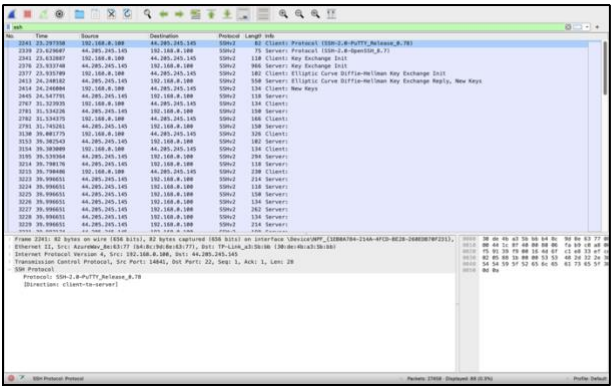
This is how the information is available on SSH. Let us try find the information using Wireshark.

```
..E..?.D"IT/.N.b......uh..K.<.1y..MA..k=...q
r..v.|..d,...]..O[;..s.U....#..<......6.|...}S..z.q...........~.K6.X../X]....
...Q...0?..Sm....v$\.Y..Jv.M-.._.#R2.x..Y............]..v...d...../...Q..2...VM=..+.%.X............>.
j...r.J..1-f.$.Q
..4P......;..........O)..'6k{I..rFH!...y.G4.".Ss8..l....
          .J'k...#...3D.....<..t%...p.K.c*.I8^...^|..
```

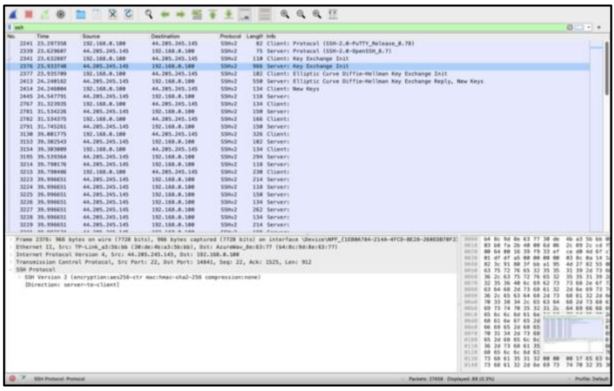
Screenshots from Wireshark:



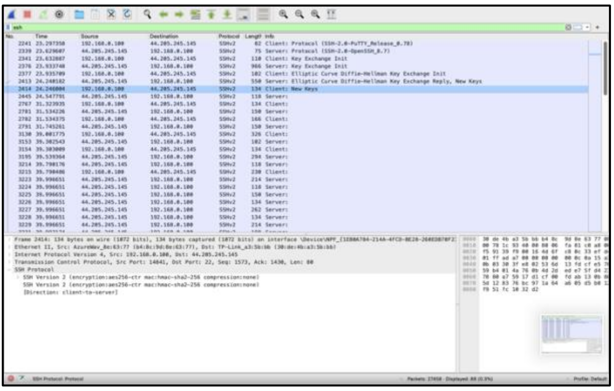
```
ssh
No.   Time       Source          Destination     Protocol Length Info
233…  81.469005  44.205.245.145  192.168.0.100   SSHv2    118 Server:
233…  81.538365  192.168.0.100   44.205.245.145  SSHv2    118 Client:
233…  81.835362  192.168.0.100   44.205.245.145  SSHv2    118 Client:
233…  81.944740  44.205.245.145  192.168.0.100   SSHv2    118 Server:
234…  82.297355  44.205.245.145  192.168.0.100   SSHv2    118 Server:
234…  82.683154  192.168.0.100   44.205.245.145  SSHv2    118 Client:
234…  82.829104  192.168.0.100   44.205.245.145  SSHv2    118 Client:
234…  83.243002  192.168.0.100   44.205.245.145  SSHv2    118 Client:
234…  83.253643  44.205.245.145  192.168.0.100   SSHv2    118 Server:
234…  83.563116  44.205.245.145  192.168.0.100   SSHv2    118 Server:
235…  83.711183  192.168.0.100   44.205.245.145  SSHv2    118 Client:
235…  84.191780  44.205.245.145  192.168.0.100   SSHv2    118 Server:
235…  84.984446  192.168.0.100   44.205.245.145  SSHv2    118 Client:
235…  85.106728  44.205.245.145  192.168.0.100   SSHv2    118 Server:
235…  85.247706  192.168.0.100   44.205.245.145  SSHv2    118 Client:
235…  85.458636  44.205.245.145  192.168.0.100   SSHv2    118 Server:
236…  86.309763  192.168.0.100   44.205.245.145  SSHv2    118 Client:
236…  86.581886  192.168.0.100   44.205.245.145  SSHv2    118 Client:
236…  86.592183  44.205.245.145  192.168.0.100   SSHv2    118 Server:
236…  86.784366  44.205.245.145  192.168.0.100   SSHv2    118 Server:
237…  87.096348  192.168.0.100   44.205.245.145  SSHv2    118 Client:
237…  87.299919  44.205.245.145  192.168.0.100   SSHv2    150 Server:
237…  87.299919  44.205.245.145  192.168.0.100   SSHv2    134 Server:
237…  87.299919  44.205.245.145  192.168.0.100   SSHv2    118 Server:
237…  87.299919  44.205.245.145  192.168.0.100   SSHv2    134 Server:

> Frame 23635: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface \Device\NPF_{1E00A784-214A-4FCD-8E28-260ED870F23
> Ethernet II, Src: TP-Link_a3:5b:bb (30:de:4b:a3:5b:bb), Dst: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77)
> Internet Protocol Version 4, Src: 44.205.245.145, Dst: 192.168.0.100
> Transmission Control Protocol, Src Port: 22, Dst Port: 14841, Seq: 5030, Ack: 4037, Len: 64
v SSH Protocol
    SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    [Direction: server-to-client]

SSH Protocol: Protocol                    Packets: 27458 · Displayed: 88 (0.3%)    Profile: Default
```



```
ssh
No.   Time       Source          Destination     Protocol Length Info
2241  23.297358  192.168.0.100   44.205.245.145  SSHv2    82 Client: Protocol (SSH-2.0-PuTTY_Release_0.78)
2339  23.629607  44.205.245.145  192.168.0.100   SSHv2    75 Server: Protocol (SSH-2.0-OpenSSH_8.7)
2341  23.632087  192.168.0.100   44.205.245.145  SSHv2    110 Client: Key Exchange Init
2376  23.933748  44.205.245.145  192.168.0.100   SSHv2    966 Server: Key Exchange Init
2377  23.935709  192.168.0.100   44.205.245.145  SSHv2    102 Client: Elliptic Curve Diffie-Hellman Key Exchange Init
2413  24.240182  44.205.245.145  192.168.0.100   SSHv2    550 Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys
2414  24.246004  192.168.0.100   44.205.245.145  SSHv2    134 Client: New Keys
2445  24.547791  44.205.245.145  192.168.0.100   SSHv2    118 Server:
2767  31.323935  192.168.0.100   44.205.245.145  SSHv2    134 Client:
2781  31.534226  44.205.245.145  192.168.0.100   SSHv2    150 Server:
2782  31.534375  192.168.0.100   44.205.245.145  SSHv2    166 Client:
2791  31.745261  44.205.245.145  192.168.0.100   SSHv2    150 Server:
3130  39.001775  192.168.0.100   44.205.245.145  SSHv2    326 Client:
3153  39.302543  44.205.245.145  192.168.0.100   SSHv2    102 Server:
3154  39.303009  192.168.0.100   44.205.245.145  SSHv2    134 Client:
3195  39.539364  44.205.245.145  192.168.0.100   SSHv2    294 Server:
3214  39.798176  44.205.245.145  192.168.0.100   SSHv2    118 Server:
3215  39.798486  192.168.0.100   44.205.245.145  SSHv2    230 Client:
3223  39.996651  44.205.245.145  192.168.0.100   SSHv2    214 Server:
3224  39.996651  44.205.245.145  192.168.0.100   SSHv2    118 Server:
3225  39.996651  44.205.245.145  192.168.0.100   SSHv2    150 Server:
3226  39.996651  44.205.245.145  192.168.0.100   SSHv2    134 Server:
3227  39.996651  44.205.245.145  192.168.0.100   SSHv2    262 Server:
3228  39.996651  44.205.245.145  192.168.0.100   SSHv2    134 Server:
3229  39.996651  44.205.245.145  192.168.0.100   SSHv2    214 Server:

> Frame 2241: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface \Device\NPF_{1E00A784-214A-4FCD-8E28-260ED870F231},
> Ethernet II, Src: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77), Dst: TP-Link_a3:5b:bb (30:de:4b:a3:5b:bb)
> Internet Protocol Version 4, Src: 192.168.0.100, Dst: 44.205.245.145
> Transmission Control Protocol, Src Port: 14841, Dst Port: 22, Seq: 1, Ack: 1, Len: 28
v SSH Protocol
    Protocol: SSH-2.0-PuTTY_Release_0.78
    [Direction: client-to-server]

SSH Protocol: Protocol                    Packets: 27458 · Displayed: 88 (0.3%)    Profile: Default
```



```
ssh
No.   Time       Source          Destination     Protocol Length Info
2241  23.297358  192.168.0.100   44.205.245.145  SSHv2    82 Client: Protocol (SSH-2.0-PuTTY_Release_0.78)
2339  23.629607  44.205.245.145  192.168.0.100   SSHv2    75 Server: Protocol (SSH-2.0-OpenSSH_8.7)
2341  23.632087  192.168.0.100   44.205.245.145  SSHv2    110 Client: Key Exchange Init
2376  23.933748  44.205.245.145  192.168.0.100   SSHv2    966 Server: Key Exchange Init
2377  23.935709  192.168.0.100   44.205.245.145  SSHv2    102 Client: Elliptic Curve Diffie-Hellman Key Exchange Init
2413  24.240182  44.205.245.145  192.168.0.100   SSHv2    550 Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys
2414  24.246004  192.168.0.100   44.205.245.145  SSHv2    134 Client: New Keys
2445  24.547791  44.205.245.145  192.168.0.100   SSHv2    118 Server:
2767  31.323935  192.168.0.100   44.205.245.145  SSHv2    134 Client:
2781  31.534226  44.205.245.145  192.168.0.100   SSHv2    150 Server:
2782  31.534375  192.168.0.100   44.205.245.145  SSHv2    166 Client:
2791  31.745261  44.205.245.145  192.168.0.100   SSHv2    150 Server:
3130  39.001775  192.168.0.100   44.205.245.145  SSHv2    326 Client:
3153  39.302543  44.205.245.145  192.168.0.100   SSHv2    102 Server:
3154  39.303009  192.168.0.100   44.205.245.145  SSHv2    134 Client:
3195  39.539364  44.205.245.145  192.168.0.100   SSHv2    294 Server:
3214  39.798176  44.205.245.145  192.168.0.100   SSHv2    118 Server:
3215  39.798486  192.168.0.100   44.205.245.145  SSHv2    230 Client:
3223  39.996651  44.205.245.145  192.168.0.100   SSHv2    214 Server:
3224  39.996651  44.205.245.145  192.168.0.100   SSHv2    118 Server:
3225  39.996651  44.205.245.145  192.168.0.100   SSHv2    150 Server:
3226  39.996651  44.205.245.145  192.168.0.100   SSHv2    134 Server:
3227  39.996651  44.205.245.145  192.168.0.100   SSHv2    262 Server:
3228  39.996651  44.205.245.145  192.168.0.100   SSHv2    134 Server:
3229  39.996651  44.205.245.145  192.168.0.100   SSHv2    214 Server:

> Frame 2376: 966 bytes on wire (7728 bits), 966 bytes captured (7728 bits) on interface \Device\NPF_{1E00A784-214A-4FCD-8E28-260ED870F2
> Ethernet II, Src: TP-Link_a3:5b:bb (30:de:4b:a3:5b:bb), Dst: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77)
> Internet Protocol Version 4, Src: 44.205.245.145, Dst: 192.168.0.100
> Transmission Control Protocol, Src Port: 22, Dst Port: 14841, Seq: 22, Ack: 1525, Len: 912
v SSH Protocol
  > SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    [Direction: server-to-client]

SSH Protocol: Protocol                    Packets: 27458 · Displayed: 88 (0.3%)    Profile: Default
```



```
ssh
No.   Time       Source          Destination     Protocol Length Info
2241  23.297358  192.168.0.100   44.205.245.145  SSHv2    82 Client: Protocol (SSH-2.0-PuTTY_Release_0.78)
2339  23.629607  44.205.245.145  192.168.0.100   SSHv2    75 Server: Protocol (SSH-2.0-OpenSSH_8.7)
2341  23.632087  192.168.0.100   44.205.245.145  SSHv2    110 Client: Key Exchange Init
2376  23.933748  44.205.245.145  192.168.0.100   SSHv2    966 Server: Key Exchange Init
2377  23.935709  192.168.0.100   44.205.245.145  SSHv2    102 Client: Elliptic Curve Diffie-Hellman Key Exchange Init
2413  24.240182  44.205.245.145  192.168.0.100   SSHv2    550 Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys
2414  24.246004  192.168.0.100   44.205.245.145  SSHv2    134 Client: New Keys
2445  24.547791  44.205.245.145  192.168.0.100   SSHv2    118 Server:
2767  31.323935  192.168.0.100   44.205.245.145  SSHv2    134 Client:
2781  31.534226  44.205.245.145  192.168.0.100   SSHv2    150 Server:
2782  31.534375  192.168.0.100   44.205.245.145  SSHv2    166 Client:
2791  31.745261  44.205.245.145  192.168.0.100   SSHv2    150 Server:
3130  39.001775  192.168.0.100   44.205.245.145  SSHv2    326 Client:
3153  39.302543  44.205.245.145  192.168.0.100   SSHv2    102 Server:
3154  39.303009  192.168.0.100   44.205.245.145  SSHv2    134 Client:
3195  39.539364  44.205.245.145  192.168.0.100   SSHv2    294 Server:
3214  39.798176  44.205.245.145  192.168.0.100   SSHv2    118 Server:
3215  39.798486  192.168.0.100   44.205.245.145  SSHv2    230 Client:
3223  39.996651  44.205.245.145  192.168.0.100   SSHv2    214 Server:
3224  39.996651  44.205.245.145  192.168.0.100   SSHv2    118 Server:
3225  39.996651  44.205.245.145  192.168.0.100   SSHv2    150 Server:
3226  39.996651  44.205.245.145  192.168.0.100   SSHv2    134 Server:
3227  39.996651  44.205.245.145  192.168.0.100   SSHv2    262 Server:
3228  39.996651  44.205.245.145  192.168.0.100   SSHv2    134 Server:
3229  39.996651  44.205.245.145  192.168.0.100   SSHv2    214 Server:

> Frame 2414: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface \Device\NPF_{1E00A784-214A-4FCD-8E28-260ED870F2
> Ethernet II, Src: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77), Dst: TP-Link_a3:5b:bb (30:de:4b:a3:5b:bb)
> Internet Protocol Version 4, Src: 192.168.0.100, Dst: 44.205.245.145
> Transmission Control Protocol, Src Port: 14841, Dst Port: 22, Seq: 1573, Ack: 1430, Len: 80
v SSH Protocol
  > SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    [Direction: client-to-server]

SSH Protocol: Protocol                    Packets: 27458 · Displayed: 88 (0.3%)    Profile: Default
```

If you can observe from the screenshots from the Wireshark, you can see no leak of information, i.e., no packets sniffed contains of usernames and passwords. Therefore, can be said it's a safe protocol.

Advantages:

- ⇒ Security: SSH encrypts all data transmitted between the client and server, providing a high level of security. This makes it a more secure choice compared to other remote access protocols such as Telnet, which do not encrypt data.
- ⇒ Authentication: SSH provides strong authentication mechanisms, including public-key authentication, which makes it more resistant to password-based attacks such as brute-force attacks and password sniffing.
- ⇒ Portability: SSH is widely supported and available on most operating systems, making it a portable choice for remote access to network devices.
- ⇒ User and Host Verification: SSH verifies both the user and host before allowing a connection, which ensures that only authorized users can access network devices.

Overall, SSH is a secure and flexible protocol that provides strong authentication mechanisms and a wide range of functionality, making it a popular choice for remote access to network devices and other applications.

We can now see that SSH is stronger than Telnet. Let's now check other protocols.

3. FTP:

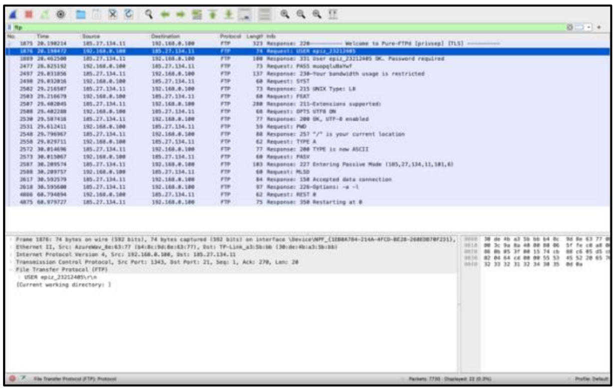
FTP (File Transfer Protocol) is a protocol used for transferring files over a network. FTP is commonly used for transferring large files and is often used in web development to upload and download files to a web server.

If you can observe in FTP, the username and password are clearly visible in a single line, whereas in Telnet when we sniffed the packets data was visible showing single letter at a time. These proves FTP can be more vulnerable than Telnet to use.

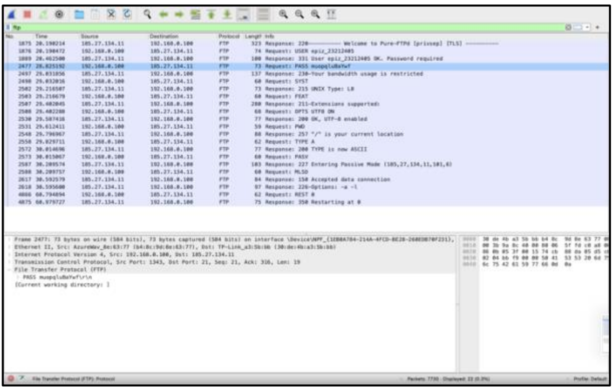
```
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 596 of 6900 allowed.
220-Local time is now 15:11. Server port: 21.
220-This is a private system - No anonymous login
220 You will be disconnected after 60 seconds of inactivity.
USER epiz_23212405
331 User epiz_23212405 OK. Password required
PASS muopqluBaYwf
230-Your bandwidth usage is restricted
230 OK. Current restricted directory is /
SYST
```

Let us now see the proof with the screenshots from the Wireshark tool.

Screenshots from Wireshark:

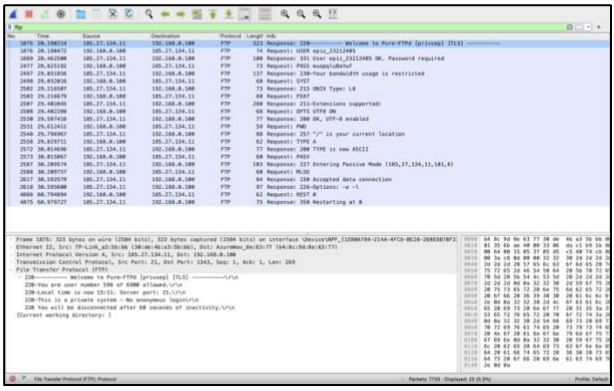


```
ftp
No.  Time       Source         Destination    Protocol Length Info
1875 20.190214  185.27.134.11  192.168.0.100  FTP      323 Response: 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
1876 20.190472  192.168.0.100  185.27.134.11  FTP       74 Request: USER epiz_23212405
1889 20.462500  185.27.134.11  192.168.0.100  FTP      100 Response: 331 User epiz_23212405 OK. Password required
2477 20.825192  192.168.0.100  185.27.134.11  FTP       73 Request: PASS muopqluBaYwf
2497 29.031056  185.27.134.11  192.168.0.100  FTP      137 Response: 230-Your bandwidth usage is restricted
2498 29.032016  192.168.0.100  185.27.134.11  FTP       60 Request: SYST
2502 29.216507  185.27.134.11  192.168.0.100  FTP       73 Response: 215 UNIX Type: L8
2503 29.216679  192.168.0.100  185.27.134.11  FTP       60 Request: FEAT
2507 29.402045  185.27.134.11  192.168.0.100  FTP      280 Response: 211-Extensions supported:
2508 29.402280  192.168.0.100  185.27.134.11  FTP       68 Request: OPTS UTF8 ON
2530 29.587418  185.27.134.11  192.168.0.100  FTP       77 Response: 200 OK, UTF-8 enabled
2531 29.612411  192.168.0.100  185.27.134.11  FTP       59 Request: PWD
2548 29.796967  185.27.134.11  192.168.0.100  FTP       88 Response: 257 "/" is your current location
2558 29.829711  192.168.0.100  185.27.134.11  FTP       62 Request: TYPE A
2572 30.014696  185.27.134.11  192.168.0.100  FTP       77 Response: 200 TYPE is now ASCII
2573 30.015067  192.168.0.100  185.27.134.11  FTP       60 Request: PASV
2587 30.209574  185.27.134.11  192.168.0.100  FTP      103 Response: 227 Entering Passive Mode (185,27,134,11,101,6)
2588 30.209757  192.168.0.100  185.27.134.11  FTP       60 Request: MLSD
2617 30.592579  185.27.134.11  192.168.0.100  FTP       84 Response: 150 Accepted data connection
2618 30.595600  185.27.134.11  192.168.0.100  FTP       97 Response: 226-Options: -a -l
4866 60.794894  192.168.0.100  185.27.134.11  FTP       62 Request: REST 0
4875 60.979727  185.27.134.11  192.168.0.100  FTP       75 Response: 350 Restarting at 0

Frame 1876: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{1EB8A784-214A-4FCD-8E28-260E0B70F231},
Ethernet II, Src: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77), Dst: TP-Link_a3:5b:bb (30:de:4b:a3:5b:bb)
Internet Protocol Version 4, Src: 192.168.0.100, Dst: 185.27.134.11
Transmission Control Protocol, Src Port: 1343, Dst Port: 21, Seq: 1, Ack: 270, Len: 20
File Transfer Protocol (FTP)
  USER epiz_23212405\r\n
[Current working directory: ]

File Transfer Protocol (FTP): Protocol    Packets: 7730 · Displayed: 22 (0.3%)    Profile: Default
```



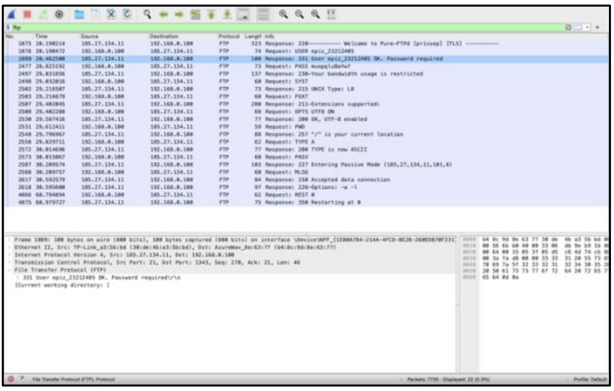
```
ftp
No.  Time       Source         Destination    Protocol Length Info
1875 20.190214  185.27.134.11  192.168.0.100  FTP      323 Response: 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
1876 20.190472  192.168.0.100  185.27.134.11  FTP       74 Request: USER epiz_23212405
1889 20.462500  185.27.134.11  192.168.0.100  FTP      100 Response: 331 User epiz_23212405 OK. Password required
2477 20.825192  192.168.0.100  185.27.134.11  FTP       73 Request: PASS muopqluBaYwf
2497 29.031056  185.27.134.11  192.168.0.100  FTP      137 Response: 230-Your bandwidth usage is restricted
2498 29.032016  192.168.0.100  185.27.134.11  FTP       60 Request: SYST
2502 29.216507  185.27.134.11  192.168.0.100  FTP       73 Response: 215 UNIX Type: L8
2503 29.216679  192.168.0.100  185.27.134.11  FTP       60 Request: FEAT
2507 29.402045  185.27.134.11  192.168.0.100  FTP      280 Response: 211-Extensions supported:
2508 29.402280  192.168.0.100  185.27.134.11  FTP       68 Request: OPTS UTF8 ON
2530 29.587418  185.27.134.11  192.168.0.100  FTP       77 Response: 200 OK, UTF-8 enabled
2531 29.612411  192.168.0.100  185.27.134.11  FTP       59 Request: PWD
2548 29.796967  185.27.134.11  192.168.0.100  FTP       88 Response: 257 "/" is your current location
2558 29.829711  192.168.0.100  185.27.134.11  FTP       62 Request: TYPE A
2572 30.014696  185.27.134.11  192.168.0.100  FTP       77 Response: 200 TYPE is now ASCII
2573 30.015067  192.168.0.100  185.27.134.11  FTP       60 Request: PASV
2587 30.209574  185.27.134.11  192.168.0.100  FTP      103 Response: 227 Entering Passive Mode (185,27,134,11,101,6)
2588 30.209757  192.168.0.100  185.27.134.11  FTP       60 Request: MLSD
2617 30.592579  185.27.134.11  192.168.0.100  FTP       84 Response: 150 Accepted data connection
2618 30.595600  185.27.134.11  192.168.0.100  FTP       97 Response: 226-Options: -a -l
4866 60.794894  192.168.0.100  185.27.134.11  FTP       62 Request: REST 0
4875 60.979727  185.27.134.11  192.168.0.100  FTP       75 Response: 350 Restarting at 0
```

```
Frame 2477: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface \Device\NPF_{1EB8A784-214A-4FCD-8E28-260E0B70F231},
Ethernet II, Src: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77), Dst: TP-Link_a3:5b:bb (30:de:4b:a3:5b:bb)
Internet Protocol Version 4, Src: 192.168.0.100, Dst: 185.27.134.11
Transmission Control Protocol, Src Port: 1343, Dst Port: 21, Seq: 21, Ack: 316, Len: 19
File Transfer Protocol (FTP)
  PASS muopqluBaYwf\r\n
[Current working directory: ]

File Transfer Protocol (FTP): Protocol    Packets: 7730 · Displayed: 22 (0.3%)    Profile: Default
```



```
ftp
No.  Time       Source         Destination    Protocol Length Info
1875 20.190214  185.27.134.11  192.168.0.100  FTP      323 Response: 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
1876 20.190472  192.168.0.100  185.27.134.11  FTP       74 Request: USER epiz_23212405
1889 20.462500  185.27.134.11  192.168.0.100  FTP      100 Response: 331 User epiz_23212405 OK. Password required
2477 20.825192  192.168.0.100  185.27.134.11  FTP       73 Request: PASS muopqluBaYwf
2497 29.031056  185.27.134.11  192.168.0.100  FTP      137 Response: 230-Your bandwidth usage is restricted
2498 29.032016  192.168.0.100  185.27.134.11  FTP       60 Request: SYST
2502 29.216507  185.27.134.11  192.168.0.100  FTP       73 Response: 215 UNIX Type: L8
2503 29.216679  192.168.0.100  185.27.134.11  FTP       60 Request: FEAT
2507 29.402045  185.27.134.11  192.168.0.100  FTP      280 Response: 211-Extensions supported:
2508 29.402280  192.168.0.100  185.27.134.11  FTP       68 Request: OPTS UTF8 ON
2530 29.587418  185.27.134.11  192.168.0.100  FTP       77 Response: 200 OK, UTF-8 enabled
2531 29.612411  192.168.0.100  185.27.134.11  FTP       59 Request: PWD
2548 29.796967  185.27.134.11  192.168.0.100  FTP       88 Response: 257 "/" is your current location
2558 29.829711  192.168.0.100  185.27.134.11  FTP       62 Request: TYPE A
2572 30.014696  185.27.134.11  192.168.0.100  FTP       77 Response: 200 TYPE is now ASCII
2573 30.015067  192.168.0.100  185.27.134.11  FTP       60 Request: PASV
2587 30.209574  185.27.134.11  192.168.0.100  FTP      103 Response: 227 Entering Passive Mode (185,27,134,11,101,6)
2588 30.209757  192.168.0.100  185.27.134.11  FTP       60 Request: MLSD
2617 30.592579  185.27.134.11  192.168.0.100  FTP       84 Response: 150 Accepted data connection
2618 30.595600  185.27.134.11  192.168.0.100  FTP       97 Response: 226-Options: -a -l
4866 60.794894  192.168.0.100  185.27.134.11  FTP       62 Request: REST 0
4875 60.979727  185.27.134.11  192.168.0.100  FTP       75 Response: 350 Restarting at 0
```

```
Frame 1875: 323 bytes on wire (2584 bits), 323 bytes captured (2584 bits) on interface \Device\NPF_{1EB8A784-214A-4FCD-8E28-260E0B70F2
Ethernet II, Src: TP-Link_a3:5b:bb (30:de:4b:a3:5b:bb), Dst: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77)
Internet Protocol Version 4, Src: 185.27.134.11, Dst: 192.168.0.100
Transmission Control Protocol, Src Port: 21, Dst Port: 1343, Seq: 1, Ack: 1, Len: 269
File Transfer Protocol (FTP)
  220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------\r\n
  220-You are user number 596 of 6900 allowed.\r\n
  220-Local time is now 15:11. Server port: 21.\r\n
  220-This is a private system - No anonymous login\r\n
  220 You will be disconnected after 60 seconds of inactivity.\r\n
[Current working directory: ]

File Transfer Protocol (FTP): Protocol    Packets: 7730 · Displayed: 22 (0.3%)    Profile: Default
```



```
ftp
No.  Time       Source         Destination    Protocol Length Info
1875 20.190214  185.27.134.11  192.168.0.100  FTP      323 Response: 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
1876 20.190472  192.168.0.100  185.27.134.11  FTP       74 Request: USER epiz_23212405
1889 20.462500  185.27.134.11  192.168.0.100  FTP      100 Response: 331 User epiz_23212405 OK. Password required
2477 20.825192  192.168.0.100  185.27.134.11  FTP       73 Request: PASS muopqluBaYwf
2497 29.031056  185.27.134.11  192.168.0.100  FTP      137 Response: 230-Your bandwidth usage is restricted
2498 29.032016  192.168.0.100  185.27.134.11  FTP       60 Request: SYST
2502 29.216507  185.27.134.11  192.168.0.100  FTP       73 Response: 215 UNIX Type: L8
2503 29.216679  192.168.0.100  185.27.134.11  FTP       60 Request: FEAT
2507 29.402045  185.27.134.11  192.168.0.100  FTP      280 Response: 211-Extensions supported:
2508 29.402280  192.168.0.100  185.27.134.11  FTP       68 Request: OPTS UTF8 ON
2530 29.587418  185.27.134.11  192.168.0.100  FTP       77 Response: 200 OK, UTF-8 enabled
2531 29.612411  192.168.0.100  185.27.134.11  FTP       59 Request: PWD
2548 29.796967  185.27.134.11  192.168.0.100  FTP       88 Response: 257 "/" is your current location
2558 29.829711  192.168.0.100  185.27.134.11  FTP       62 Request: TYPE A
2572 30.014696  185.27.134.11  192.168.0.100  FTP       77 Response: 200 TYPE is now ASCII
2573 30.015067  192.168.0.100  185.27.134.11  FTP       60 Request: PASV
2587 30.209574  185.27.134.11  192.168.0.100  FTP      103 Response: 227 Entering Passive Mode (185,27,134,11,101,6)
2588 30.209757  192.168.0.100  185.27.134.11  FTP       60 Request: MLSD
2617 30.592579  185.27.134.11  192.168.0.100  FTP       84 Response: 150 Accepted data connection
2618 30.595600  185.27.134.11  192.168.0.100  FTP       97 Response: 226-Options: -a -l
4866 60.794894  192.168.0.100  185.27.134.11  FTP       62 Request: REST 0
4875 60.979727  185.27.134.11  192.168.0.100  FTP       75 Response: 350 Restarting at 0
```

```
Frame 1889: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface \Device\NPF_{1EB8A784-214A-4FCD-8E28-260E0B70F231}
Ethernet II, Src: TP-Link_a3:5b:bd (30:de:4b:a3:5b:bd), Dst: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77)
Internet Protocol Version 4, Src: 185.27.134.11, Dst: 192.168.0.100
Transmission Control Protocol, Src Port: 21, Dst Port: 1343, Seq: 270, Ack: 21, Len: 46
File Transfer Protocol (FTP)
  331 User epiz_23212405 OK. Password required\r\n
[Current working directory: ]

File Transfer Protocol (FTP): Protocol    Packets: 7730 · Displayed: 22 (0.3%)    Profile: Default
```

Drawbacks:

- ⇒ Security Issues: FTP is not a secure protocol, as it transmits data in plain text, making it vulnerable to interception and unauthorized access. FTPS (FTP over SSL) and SFTP (Secure File Transfer Protocol) are more secure alternatives.
- ⇒ Firewall Configuration: FTP uses multiple ports, which can make it difficult to configure firewalls and network security.
- ⇒ Limited Error Recovery: FTP does not provide a reliable mechanism for recovering from transmission errors, which can result in data corruption or loss.
- ⇒ Limited Logging: FTP has limited logging and audit capabilities, which can make it difficult to track and monitor file transfer activity.

Overall, FTP is a simple and efficient protocol for transferring files between computers. However, its lack of security and limited error recovery and logging capabilities make it less attractive in certain situations. For more secure file transfer, alternatives such as FTPS and SFTP should be considered.

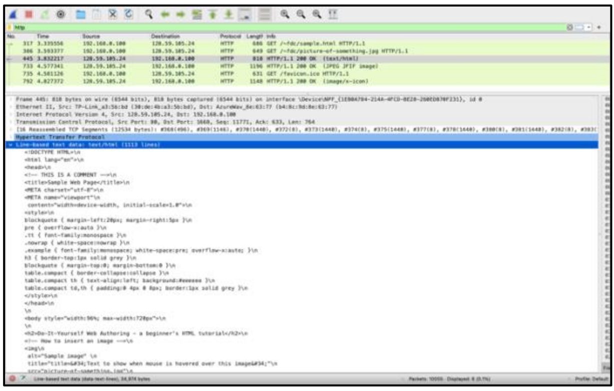
Finally, let us find out if HTTP is vulnerable or not.

4. **HTTP:**

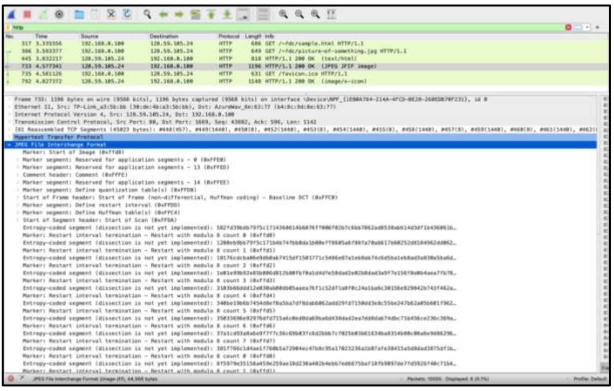
   HTTP (Hypertext Transfer Protocol) is the protocol used for transmitting data over the World Wide Web. HTTP is the foundation of the web and is used for transmitting HTML pages, images, and other resources between web servers and clients.

Let us first see the screenshots from the Wireshark tool and then discuss about the image above.

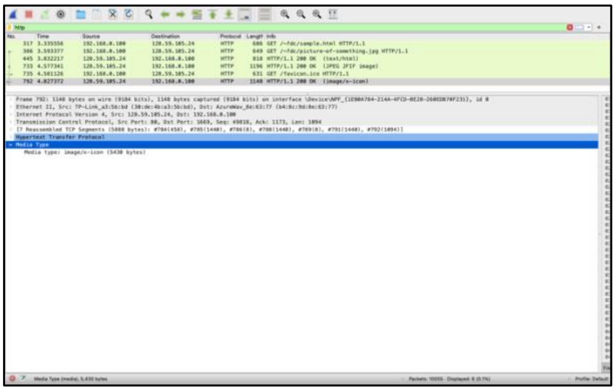
Screenshots from Wireshark:



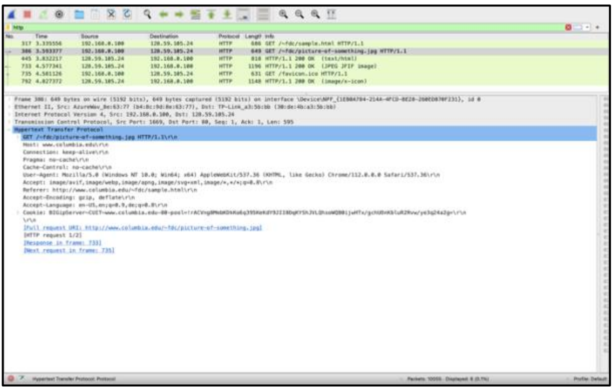
```
http
No.   Time       Source          Destination     Protocol Length Info
  317 3.335556   192.168.0.100   128.59.105.24   HTTP     686 GET /~fdc/sample.html HTTP/1.1
  386 3.503377   192.168.0.100   128.59.105.24   HTTP     649 GET /~fdc/picture-of-something.jpg HTTP/1.1
  445 3.832217   128.59.105.24   192.168.0.100   HTTP     818 HTTP/1.1 200 OK  (text/html)
  733 4.577341   128.59.105.24   192.168.0.100   HTTP    1196 HTTP/1.1 200 OK  (JPEG JFIF image)
  735 4.581126   192.168.0.100   128.59.105.24   HTTP     631 GET /favicon.ico HTTP/1.1
  792 4.827372   128.59.105.24   192.168.0.100   HTTP    1148 HTTP/1.1 200 OK  (image/x-icon)

Frame 445: 818 bytes on wire (6544 bits), 818 bytes captured (6544 bits) on interface \Device\NPF_{1E80A784-214A-4FCD-8E28-260ED870F231}, id 0
Ethernet II, Src: TP-Link_a3:5b:bd (30:de:4b:a3:5b:bd), Dst: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77)
Internet Protocol Version 4, Src: 128.59.105.24, Dst: 192.168.0.100
Transmission Control Protocol, Src Port: 80, Dst Port: 1668, Seq: 11771, Ack: 633, Len: 764
[16 Reassembled TCP Segments (12534 bytes): #368(496), #369(1146), #370(1440), #372(0), #373(1440), #374(0), #375(1440), #377(0), #378(1440), #380(0), #381(1440), #382(0), #383(
Hypertext Transfer Protocol
Line-based text data: text/html (1113 lines)
    <!DOCTYPE HTML>\n
    <html lang="en">\n
    <head>\n
    <!-- THIS IS A COMMENT -->\n
    <title>Sample Web Page</title>\n
    <META charset="utf-8">\n
    <META name="viewport"\n
     content="width=device-width, initial-scale=1.0">\n
    <style>\n
    blockquote { margin-left:20px; margin-right:5px }\n
    pre { overflow-x:auto }\n
    .tt { font-family:monospace }\n
    .nowrap { white-space:nowrap }\n
    .example { font-family:monospace; white-space:pre; overflow-x:auto; }\n
    h3 { border-top:1px solid grey }\n
    blockquote { margin-top:0; margin-bottom:0 }\n
    table.compact { border-collapse:collapse }\n
    table.compact th { text-align:left; background:#eeeeee }\n
    table.compact td,th { padding:0 4px 0 8px; border:1px solid grey }\n
    </style>\n
    </head>\n
    \n
    <body style="width:96%; max-width:720px">\n
    \n
    <h2>Do-It-Yourself Web Authoring - a beginner's HTML tutorial</h2>\n
    <!-- How to insert an image -->\n
    <img\n
     alt="Sample image" \n
     title="title=&#34;Text to show when mouse is hovered over this image&#34;"\n
     src="picture-of-something.jpg"\n

Line-based text data (data-text-lines), 34,974 bytes          Packets: 10055 · Displayed: 6 (0.1%)          Profile: Default
```



```
http
No.   Time       Source          Destination     Protocol Length Info
  317 3.335556   192.168.0.100   128.59.105.24   HTTP     686 GET /~fdc/sample.html HTTP/1.1
  386 3.503377   192.168.0.100   128.59.105.24   HTTP     649 GET /~fdc/picture-of-something.jpg HTTP/1.1
  445 3.832217   128.59.105.24   192.168.0.100   HTTP     818 HTTP/1.1 200 OK  (text/html)
  733 4.577341   128.59.105.24   192.168.0.100   HTTP    1196 HTTP/1.1 200 OK  (JPEG JFIF image)
  735 4.581126   192.168.0.100   128.59.105.24   HTTP     631 GET /favicon.ico HTTP/1.1
  792 4.827372   128.59.105.24   192.168.0.100   HTTP    1148 HTTP/1.1 200 OK  (image/x-icon)

Frame 733: 1196 bytes on wire (9568 bits), 1196 bytes captured (9568 bits) on interface \Device\NPF_{1E80A784-214A-4FCD-8E28-260ED870F231}, id 0
Ethernet II, Src: TP-Link_a3:5b:bb (30:de:4b:a3:5b:bb), Dst: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77)
Internet Protocol Version 4, Src: 128.59.105.24, Dst: 192.168.0.100
Transmission Control Protocol, Src Port: 80, Dst Port: 1669, Seq: 43882, Ack: 596, Len: 1142
[61 Reassembled TCP Segments (45023 bytes): #448(457), #449(1440), #450(0), #452(1440), #453(0), #454(1440), #455(0), #456(1440), #457(0), #459(1440), #460(0), #461(1440), #462(
Hypertext Transfer Protocol
JPEG File Interchange Format
    Marker: Start of Image (0xffd8)
    Marker segment: Reserved for application segments - 0 (0xFFE0)
    Marker segment: Reserved for application segments - 13 (0xFFED)
    Comment header: Comment (0xFFFE)
    Marker segment: Reserved for application segments - 14 (0xFFEE)
    Marker segment: Define quantization table(s) (0xFFDB)
    Start of Frame header: Start of Frame (non-differential, Huffman coding) - Baseline DCT (0xFFC0)
    Marker segment: Define restart interval (0xFFDD)
    Marker segment: Define Huffman table(s) (0xFFC4)
    Start of Segment header: Start of Scan (0xFFDA)
    Entropy-coded segment (dissection is not yet implemented): [illegible]…
    Marker: Restart interval termination - Restart with modulo 8 count 0 (0xffd0)
    Entropy-coded segment (dissection is not yet implemented): [illegible]…
    Marker: Restart interval termination - Restart with modulo 8 count 1 (0xffd1)
    Entropy-coded segment (dissection is not yet implemented): [illegible]…
    Marker: Restart interval termination - Restart with modulo 8 count 2 (0xffd2)
    Entropy-coded segment (dissection is not yet implemented): [illegible]…
    Marker: Restart interval termination - Restart with modulo 8 count 3 (0xffd3)
    Entropy-coded segment (dissection is not yet implemented): [illegible]…
    Marker: Restart interval termination - Restart with modulo 8 count 4 (0xffd4)
    Entropy-coded segment (dissection is not yet implemented): [illegible]…
    Marker: Restart interval termination - Restart with modulo 8 count 5 (0xffd5)
    Entropy-coded segment (dissection is not yet implemented): [illegible]…
    Marker: Restart interval termination - Restart with modulo 8 count 6 (0xffd6)
    Entropy-coded segment (dissection is not yet implemented): [illegible]…
    Marker: Restart interval termination - Restart with modulo 8 count 7 (0xffd7)
    Entropy-coded segment (dissection is not yet implemented): [illegible]…
    Marker: Restart interval termination - Restart with modulo 8 count 0 (0xffd0)
    Entropy-coded segment (dissection is not yet implemented): [illegible]…
    Marker: Restart interval termination - Restart with modulo 8 count 1 (0xffd1)

JPEG File Interchange Format (image/jfif), 44,569 bytes          Packets: 10055 · Displayed: 6 (0.1%)          Profile: Default
```



```
http
No.   Time       Source          Destination     Protocol Length Info
  317 3.335556   192.168.0.100   128.59.105.24   HTTP     686 GET /~fdc/sample.html HTTP/1.1
  386 3.503377   192.168.0.100   128.59.105.24   HTTP     649 GET /~fdc/picture-of-something.jpg HTTP/1.1
  445 3.832217   128.59.105.24   192.168.0.100   HTTP     818 HTTP/1.1 200 OK  (text/html)
  733 4.577341   128.59.105.24   192.168.0.100   HTTP    1196 HTTP/1.1 200 OK  (JPEG JFIF image)
  735 4.581126   192.168.0.100   128.59.105.24   HTTP     631 GET /favicon.ico HTTP/1.1
  792 4.827372   128.59.105.24   192.168.0.100   HTTP    1148 HTTP/1.1 200 OK  (image/x-icon)

Frame 792: 1148 bytes on wire (9184 bits), 1148 bytes captured (9184 bits) on interface \Device\NPF_{1E80A784-214A-4FCD-8E28-260ED870F231}, id 0
Ethernet II, Src: TP-Link_a3:5b:bd (30:de:4b:a3:5b:bd), Dst: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77)
Internet Protocol Version 4, Src: 128.59.105.24, Dst: 192.168.0.100
Transmission Control Protocol, Src Port: 80, Dst Port: 1669, Seq: 49018, Ack: 1173, Len: 1094
[7 Reassembled TCP Segments (5880 bytes): #784(458), #785(1440), #786(0), #788(1440), #789(0), #791(1440), #792(1094)]
Hypertext Transfer Protocol
Media Type
    Media type: image/x-icon (5430 bytes)

Media Type (media), 5,430 bytes          Packets: 10055 · Displayed: 6 (0.1%)          Profile: Default
```



```
http
No.   Time       Source          Destination     Protocol Length Info
  317 3.335556   192.168.0.100   128.59.105.24   HTTP     686 GET /~fdc/sample.html HTTP/1.1
  386 3.503377   192.168.0.100   128.59.105.24   HTTP     649 GET /~fdc/picture-of-something.jpg HTTP/1.1
  445 3.832217   128.59.105.24   192.168.0.100   HTTP     818 HTTP/1.1 200 OK  (text/html)
  733 4.577341   128.59.105.24   192.168.0.100   HTTP    1196 HTTP/1.1 200 OK  (JPEG JFIF image)
  735 4.581126   192.168.0.100   128.59.105.24   HTTP     631 GET /favicon.ico HTTP/1.1
  792 4.827372   128.59.105.24   192.168.0.100   HTTP    1148 HTTP/1.1 200 OK  (image/x-icon)

Frame 386: 649 bytes on wire (5192 bits), 649 bytes captured (5192 bits) on interface \Device\NPF_{1E80A784-214A-4FCD-8E28-260ED870F231}, id 0
Ethernet II, Src: AzureWav_8e:63:77 (b4:8c:9d:8e:63:77), Dst: TP-Link_a3:5b:bb (30:de:4b:a3:5b:bb)
Internet Protocol Version 4, Src: 192.168.0.100, Dst: 128.59.105.24
Transmission Control Protocol, Src Port: 1669, Dst Port: 80, Seq: 1, Ack: 1, Len: 595
Hypertext Transfer Protocol
    GET /~fdc/picture-of-something.jpg HTTP/1.1\r\n
    Host: www.columbia.edu\r\n
    Connection: keep-alive\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36\r\n
    Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8\r\n
    Referer: http://www.columbia.edu/~fdc/sample.html\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9,de;q=0.8\r\n
    Cookie: BIGipServer~CUIT~www.columbia.edu-80-pool=[illegible]\r\n
    \r\n
    [Full request URI: http://www.columbia.edu/~fdc/picture-of-something.jpg]
    [HTTP request 1/2]
    [Response in frame: 733]
    [Next request in frame: 735]

Hypertext Transfer Protocol: Protocol          Packets: 10055 · Displayed: 6 (0.1%)          Profile: Default
```

The Image mentioned in the Wireshark tool is the image shown above. When opened the website the Wireshark captured the website URL and then all the activities happened on the website. That is where we opened the image and that image got sniffed in the tool.

Drawbacks:

⇒ Security Issues: HTTP transmits data in plain text, making it vulnerable to interception and unauthorized access. HTTPS (HTTP over SSL/TLS) provides a more secure alternative.

⇒ Lack of State: HTTP is a stateless protocol, which means that each request is treated independently. This can make it difficult to maintain user sessions and other state information.

⇒ Limited Flexibility: HTTP is a rigid protocol that does not provide a lot of flexibility in terms of data formats and transmission methods.

⇒ Limited Error Recovery: HTTP does not provide a reliable mechanism for recovering from transmission errors, which can result in data corruption or loss.

Overall, HTTP is a lightweight and easy-to-use protocol that is well-suited for web applications and websites. However, its lack of security and state management, limited flexibility, and limited error recovery capabilities can make it less attractive in certain situations. For more secure and flexible web applications, HTTPS and other alternative protocols should be considered.

RESULTS:

The order of less vulnerable to hacking for SSH, Telnet, HTTP, and FTP is as follows:

SSH - SSH is the most secure protocol among these four, as it uses encryption to protect data in transit and provides strong authentication mechanisms.

HTTP - Although HTTP is not a secure protocol and transmits data in plain text, modern web applications and websites often use HTTPS, which provides encryption and authentication.

FTP - FTP is vulnerable to interception and unauthorized access as it transmits data in plain text and lacks security features such as encryption and strong authentication.

Telnet - Telnet is the least secure protocol among these four, as it transmits data in plain text and lacks encryption and strong authentication mechanisms, making it vulnerable to interception and unauthorized access.

Prevention of packet sniffing:

To avoid packet sniffing, consider implementing the following measures:

- ⇒ Use encryption: Use secure protocols that encrypt network traffic, such as HTTPS for web browsing, SSH for remote access, and SFTP for file transfers. Encryption makes it difficult for hackers to read the intercepted traffic.
- ⇒ Use a VPN: A virtual private network (VPN) creates an encrypted tunnel between the user's device and the VPN server, protecting all network traffic from interception and sniffing.
- ⇒ Disable unnecessary network services: Disable any unnecessary network services on your devices, such as file sharing or remote desktop, as these services can create security vulnerabilities that can be exploited by hackers.
- ⇒ Use a firewall: Use a firewall to block unauthorized network traffic and only allow trusted traffic to pass through.
- ⇒ Regularly update software and systems: Regularly update software and systems to ensure that they are patched against known security vulnerabilities that can be exploited by hackers.
- ⇒ Use strong passwords: Use strong and unique passwords for all accounts and consider using a password manager to generate and store complex passwords.
- ⇒ Avoid unsecured public Wi-Fi: Avoid connecting to unsecured public Wi-Fi networks, as these networks can be easily compromised by hackers.
- ⇒ By implementing these measures, users can protect their network traffic from packet sniffing and other forms of cyberattacks.

CONCLUSION:

In conclusion, the order of less vulnerable to hacking for SSH, Telnet, HTTP, and FTP is

SSH > HTTP > FTP > Telnet

While SSH is the most secure protocol among these four, Telnet is the least secure, as it transmits data in plain text and lacks encryption and strong authentication mechanisms. FTP is also vulnerable to interception and unauthorized access, while HTTP is often used with HTTPS to provide encryption and authentication.

It's important to note that for all these protocols, there are secure alternatives available, such as SFTP for FTP, HTTPS for HTTP, and SSH for Telnet. As such, it's recommended to use these secure alternatives whenever possible to minimize the risk of hacking and unauthorized access. By taking the necessary precautions and using secure protocols, organizations can protect their sensitive data and systems from potential cyber threats.

REFERENCES:

OpenSSH: https://www.openssh.com/features/security.html

Difference Between Telnet and SSH: https://www.geeksforgeeks.org/difference-between-telnet-and-ssh/

FileZilla: https://filezilla-project.org/ftp-security.php

Securing FTP with SSL/TLS: https://www.sans.org/reading-room/whitepapers/protocols/securing-ftp-ssl-tls-149

The Telnet Protocol: https://tools.ietf.org/html/rfc854

Telnet Vs SSH: https://www.educba.com/telnet-vs-ssh/