Experiment No. 9

Title: Network Sniffing - Wireshark

Batch: A3 **Roll No.: 16010421059** **Experiment No.: 09**

Aim: To perform network sniffing using wire shark tool

Resources needed: Wire shark tool

Theory

Wireshark is a network packet analyzer. Any network packet analyzer will try to capture network packets and will try to display that packet data as detailed as possible in human readable format. Wireshark is an open source software project, and is released under the GNU General Public License (GPL). We can freely use Wireshark on any number of computers, without worrying about license keys. In addition, all source code is freely available under the GPL. Because of that, it is very easy for people to add new protocols to Wireshark, either as plug-in, or built into the source code. In the past, such tools were either very expensive, proprietary. However, with the advent of Wire-shark, all that has changed. Wireshark is perhaps one of the best open source packet analyzers available today.

What Wireshark is not.....

Here are some things Wireshark does not provide:
1. Wireshark isn't an intrusion detection system. It will not warn us when someone does strange things on our network that he/she isn't allowed to do. However, if strange things happen, Wireshark might help you figure out what is really going on.
2. Wireshark will not manipulate things on the network, it will only "measure" things from it. Wireshark doesn't send packets on the network or do other active things.

Applications of Wireshark:
Here are some applications. Many people use Wireshark for doing following things,

- Network administrators use it to troubleshoot network problems.
- Network security engineers use it to examine security problems (Network Forensics.)
- Developers use it to debug protocol implementations.
- People use it to learn network protocol internals.

Beside these examples Wireshark can be helpful in many other situations too.

Features of Wireshark:
The following are some of the many features Wireshark has:

- Available for UNIX and Windows operating systems.
- Capture live packet data from a chosen network interface.
- Open files containing packet data captured with tcpdump/WinDump and a number of other packet capture programs.

- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.
  ……and a lot more!

Most important menus are : 1) Capture 2) Analyze 3) Statistics
Students are expected to explore all these menus and sub-menus in details.

Wireshark can capture traffic from many different network media types including wireless LAN as well. Which media types are supported, depends on many things like the operating system we are using and the hardware support.

Physical interfaces supported:

- ATM - capture ATM traffic
- Bluetooth- capture Bluetooth traffic .
- Cisco HDLC links - capture on synchronous links using Cisco HDLC encapsulation.
- Ethernet- capture on different topologies, including switched networks.
- Framerelay – captures framerelay traffic.
- IrDA capture IrDA traffic - currently limited to Linux.
- PPP links - capture on dial-up lines, ISDN connections and PPP-over-Ethernet (PPPoe, e.g. ADSL)
- Tokenring - capture on Tokenring adapters, promiscuous mode and switched networks
- USB- capture of raw USB traffic
- WLAN- capture on 802.11 (WLAN, Wi-Fi) interfaces, including "monitor mode" , raw 802.11 headers and radio information

Virtual interfaces :

- Loopbak - capture traffic from a machine to itself, including the IP address 127.0.0.1
- Pipes - use UNIX pipes to capture from other applications (even remote!)
- VLAN – capture VLAN traffic, including VLAN tags.

In addition to this, Wireshark can do following things.

- Import files from many other capture programs.
- Wireshark can open packets captured from a large number of other capture programs.
- Export files for many other capture programs.
- Wireshark can save packets captured in a large number of formats of other capture programs.
- Can be used as a protocol decoder.

Procedure / Approach /Algorithm / Activity Diagram:

1. Go to the official website of Wire shark (www.wireshark.org) and download the stable version of Wire shark for 64 bit windows operating system.
2. After successful installation you will get the blue icon of Wire shark on the desktop.
3. Click on the icon and start the software.
4. Choose an interface and start capturing the packets.
5. Study the packet details of all the protocols.
6. Understand colour code in details.
7. Perform the statistics for a particular protocol. (Every student should perform for different protocol).

Implementation:

Task1: Design your own registration and login pages (along with user database of registered users)

Task2: Run wire shark and capture the login page request data using wire shark and locate the captured password.

Questions:

1**. Difference between Burp Suite and Wireshark:**

Burp Suite and Wireshark are both cybersecurity tools, but they serve different purposes and have distinct features:

- Burp Suite:
  - Category: Web Application Security Testing Tool
  - Purpose: Burp Suite is primarily used for web application security testing and analysis.
  - Features:
    - Intercepting and modifying HTTP requests and responses.
    - Scanning web applications for vulnerabilities like cross-site scripting (XSS) and SQL injection.
    - Automated crawling and scanning of web applications.

- Session handling for various authentication mechanisms.
- Tools for analyzing, spidering, and brute-forcing web applications.

- Wireshark:
  - Category: Network Protocol Analyzer
  - Purpose: Wireshark is used for capturing and analyzing network traffic.
  - Features:
    - Capturing and inspecting packets on a network.
    - Protocol analysis and decoding to understand network communication.
    - Can capture a wide range of network data, not just web traffic.
    - Often used for troubleshooting network issues and understanding how data flows through a network.

2. **Methods and Security Mechanisms to Protect Passwords from Wireshark and Similar Tools:**

When it comes to protecting passwords from tools like Wireshark, the primary concern is securing the transmission of passwords over a network. Here are some methods and security mechanisms to help mitigate the risk of password leakage:

- Transport Layer Security (TLS/SSL): Use HTTPS for web applications and secure communication channels for other services. TLS encrypts data in transit, making it much harder for eavesdroppers to intercept passwords.

- Hash and Salt Passwords: Don't store plaintext passwords. Instead, store salted and hashed versions of passwords in databases. This way, even if the database is breached, attackers won't obtain the actual passwords.

- Multi-Factor Authentication (MFA): Implement MFA to add an extra layer of security. Even if an attacker obtains a password, they would still need another factor, such as a one-time code from a mobile app, to gain access.

- Network Segmentation: Isolate sensitive systems and data from the rest of the network. This limits the exposure of passwords in case of a breach.

- Intrusion Detection and Prevention Systems (IDS/IPS): Use IDS and IPS solutions to monitor network traffic for suspicious activity and automatically block or alert on potential threats.

- Strong Password Policies: Encourage users to create strong, unique passwords, and enforce password policies that include length, complexity, and regular changes.

- Educate Users: Train users about the importance of password security and the risks of using the same password across multiple accounts.

- Regular Security Audits: Conduct security audits to identify vulnerabilities, including those that might expose passwords, and promptly address any issues found.

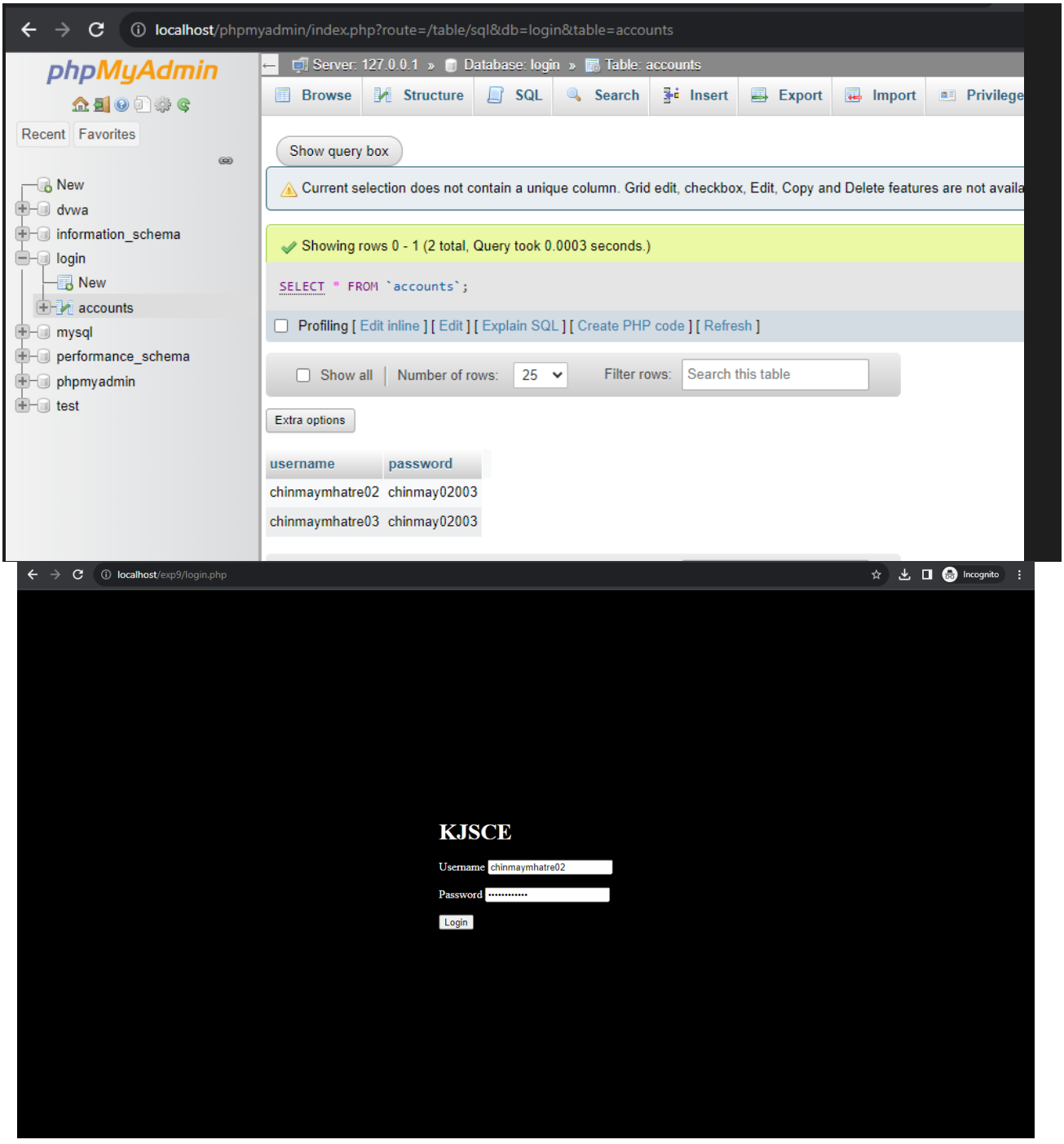
- Implement Rate Limiting: Limit the number of login attempts to prevent brute-force attacks on login credentials.

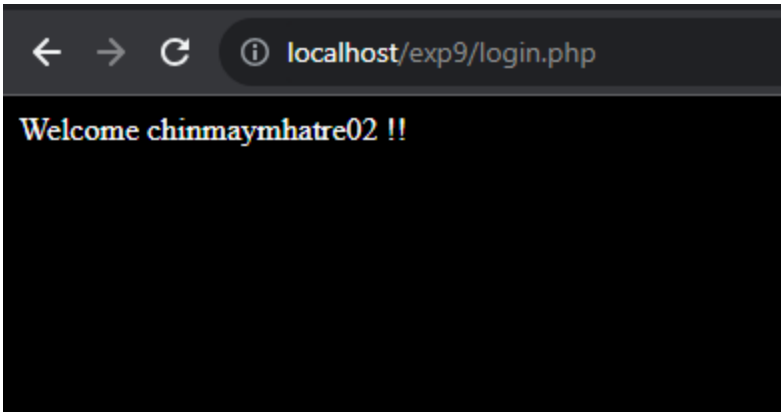
- Use Authentication Tokens: For web applications, consider using tokens like JSON Web Tokens (JWT) for authentication. Tokens can be more secure than sending plaintext passwords with each request.

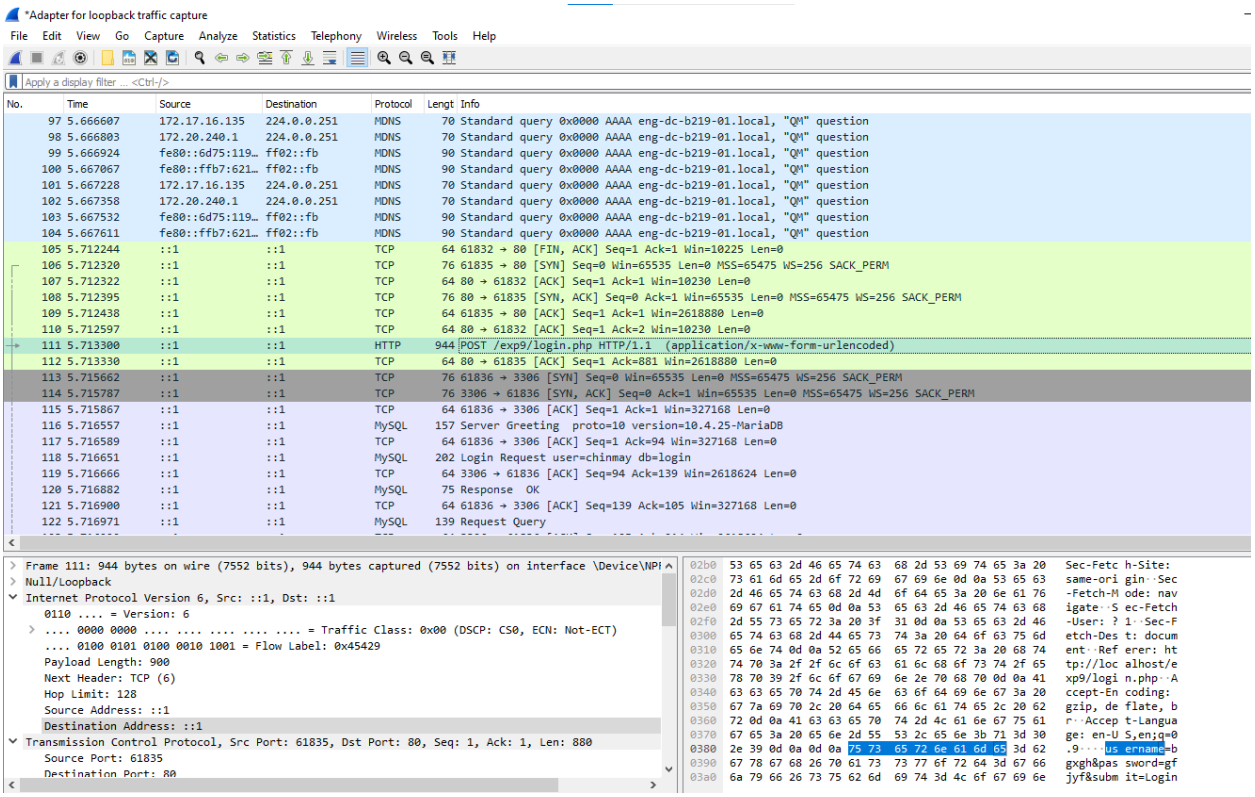
Result: Task 1 implementation code and Task 2 screenshots.

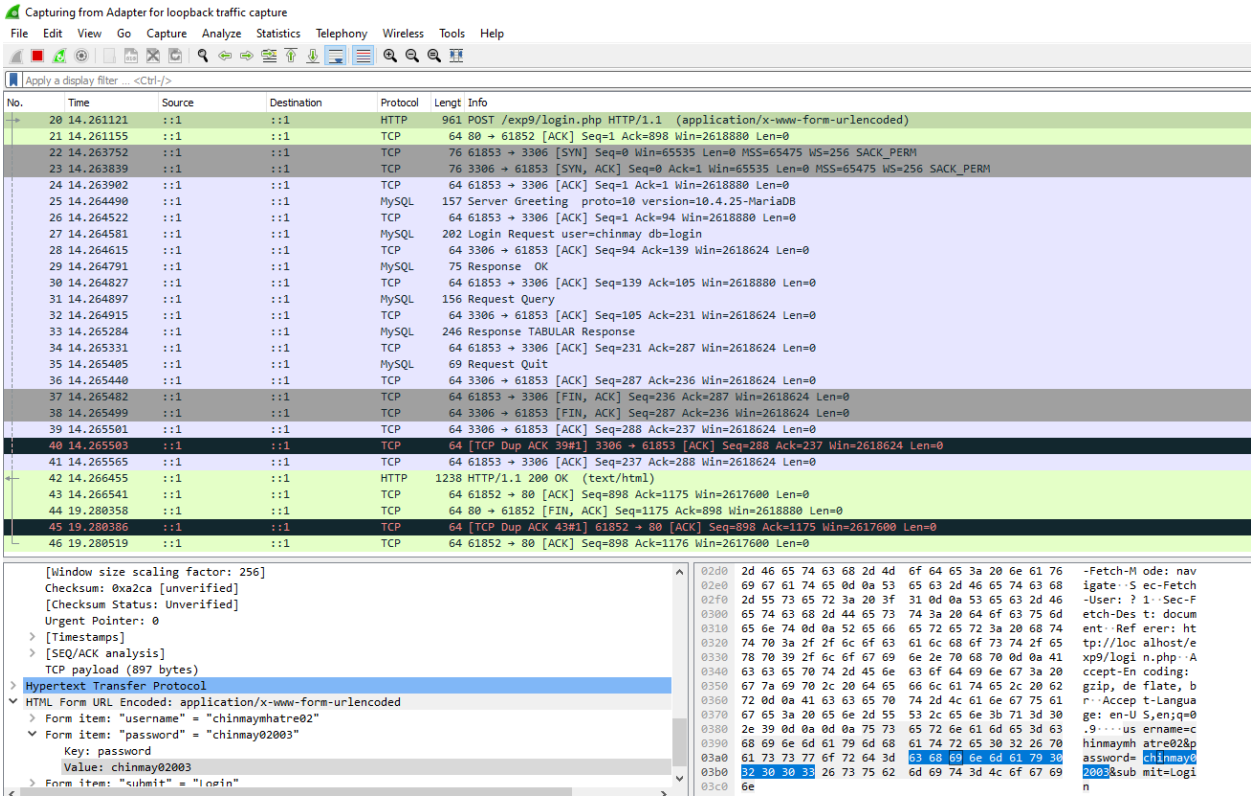
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
<style>
    body{
        background-color:black;
        color:white;
    }
    .container{
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}
</style>
</head>
<body>
    <?php
if (isset($_POST['submit'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];


    $dbHost = 'localhost';
    $dbUser = 'chinmay';
    $dbPassword = 'chinmay02003';
    $dbName = 'login';

    $conn = new mysqli($dbHost, $dbUser, $dbPassword, $dbName);


    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }


    $username = $conn->real_escape_string($username);
```

```
    $password = $conn->real_escape_string($password);


    $query = "SELECT * FROM accounts WHERE username = '$username' AND password =
'$password'";
    $result = $conn->query($query);

    if ($result->num_rows > 0) {

        echo "Welcome $username !! ";
    } else {

        echo "Invalid username or password";
    }

    $conn->close();
}
?>
  <div class="container">
        <h1>KJSCE</h1>
        <form method="post" action="login.php">
            <label for="username">Username</label>
            <input type="text" id="username" name="username">
            <br><br>
            <label for="password">Password</label>
            <input type="password" id="password" name="password">
            <br><br>
            <input type="submit" id="submit" name="submit" value="Login">
        </form>
    </div>
</body>
</html>
```


localhost/phpmyadmin/index.php?route=/table/sql&db=login&table=accounts
phpMyAdmin
Recent
Favorites
New
dvwa
information_schema
login
New
accounts
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: login » Table: accounts
Browse
Structure
SQL
Search
Insert
Export
Import
Privilege
Show query box
Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not availa
Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)
SELECT * FROM `accounts`;
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Extra options
username
password
chinmaymhatre02
chinmay02003
chinmaymhatre03
chinmay02003
localhost/exp9/login.php
Incognito
KJSCE
Username
chinmaymhatre02
Password
Login


localhost/exp9/login.php
Welcome chinmaymhatre02 !!


*Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Lengt Info
97 5.666607 172.17.16.135 224.0.0.251 MDNS 70 Standard query 0x0000 AAAA eng-dc-b219-01.local, "QM" question
98 5.666803 172.20.240.1 224.0.0.251 MDNS 70 Standard query 0x0000 AAAA eng-dc-b219-01.local, "QM" question
99 5.666924 fe80::6d75:119… ff02::fb MDNS 90 Standard query 0x0000 AAAA eng-dc-b219-01.local, "QM" question
100 5.667067 fe80::ffb7:621… ff02::fb MDNS 90 Standard query 0x0000 AAAA eng-dc-b219-01.local, "QM" question
101 5.667228 172.17.16.135 224.0.0.251 MDNS 70 Standard query 0x0000 AAAA eng-dc-b219-01.local, "QM" question
102 5.667358 172.20.240.1 224.0.0.251 MDNS 70 Standard query 0x0000 AAAA eng-dc-b219-01.local, "QM" question
103 5.667532 fe80::6d75:119… ff02::fb MDNS 90 Standard query 0x0000 AAAA eng-dc-b219-01.local, "QM" question
104 5.667611 fe80::ffb7:621… ff02::fb MDNS 90 Standard query 0x0000 AAAA eng-dc-b219-01.local, "QM" question
105 5.712244 ::1 ::1 TCP 64 61832 → 80 [FIN, ACK] Seq=1 Ack=1 Win=10225 Len=0
106 5.712320 ::1 ::1 TCP 76 61835 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
107 5.712322 ::1 ::1 TCP 64 80 → 61832 [ACK] Seq=1 Ack=1 Win=10230 Len=0
108 5.712395 ::1 ::1 TCP 76 80 → 61835 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
109 5.712438 ::1 ::1 TCP 64 61835 → 80 [ACK] Seq=1 Ack=1 Win=2618880 Len=0
110 5.712597 ::1 ::1 TCP 64 80 → 61832 [ACK] Seq=1 Ack=2 Win=10230 Len=0
111 5.713300 ::1 ::1 HTTP 944 POST /exp9/login.php HTTP/1.1 (application/x-www-form-urlencoded)
112 5.713330 ::1 ::1 TCP 64 80 → 61835 [ACK] Seq=1 Ack=881 Win=2618880 Len=0
113 5.715662 ::1 ::1 TCP 76 61836 → 3306 [SYN] Seq=0 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
114 5.715787 ::1 ::1 TCP 76 3306 → 61836 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
115 5.715867 ::1 ::1 TCP 64 61836 → 3306 [ACK] Seq=1 Ack=1 Win=327168 Len=0
116 5.716557 ::1 ::1 MySQL 157 Server Greeting proto=10 version=10.4.25-MariaDB
117 5.716589 ::1 ::1 TCP 64 61836 → 3306 [ACK] Seq=1 Ack=94 Win=327168 Len=0
118 5.716651 ::1 ::1 MySQL 202 Login Request user=chinmay db=login
119 5.716666 ::1 ::1 TCP 64 3306 → 61836 [ACK] Seq=94 Ack=139 Win=2618624 Len=0
120 5.716882 ::1 ::1 MySQL 75 Response OK
121 5.716900 ::1 ::1 TCP 64 61836 → 3306 [ACK] Seq=139 Ack=105 Win=327168 Len=0
122 5.716971 ::1 ::1 MySQL 139 Request Query
Frame 111: 944 bytes on wire (7552 bits), 944 bytes captured (7552 bits) on interface \Device\NPF
Null/Loopback
Internet Protocol Version 6, Src: ::1, Dst: ::1
0110 .... = Version: 6
.... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
.... 0100 0101 0100 0010 1001 = Flow Label: 0x45429
Payload Length: 900
Next Header: TCP (6)
Hop Limit: 128
Source Address: ::1
Destination Address: ::1
Transmission Control Protocol, Src Port: 61835, Dst Port: 80, Seq: 1, Ack: 1, Len: 880
Source Port: 61835
02b0 53 65 63 2d 46 65 74 63 68 2d 53 69 74 65 3a 20 Sec-Fetc h-Site:
02c0 73 61 6d 65 2d 6f 72 69 67 69 6e 0d 0a 53 65 63 same-ori gin··Sec
02d0 2d 46 65 74 63 68 2d 4d 6f 64 65 3a 20 6e 61 76 -Fetch-M ode: nav
02e0 69 67 61 74 65 0d 0a 53 65 63 2d 46 65 74 63 68 igate··S ec-Fetch
02f0 2d 55 73 65 72 3a 20 3f 31 0d 0a 53 65 63 2d 46 -User: ? 1··Sec-F
0300 65 74 63 68 2d 44 65 73 74 3a 20 64 6f 63 75 6d etch-Des t: docum
0310 65 6e 74 0d 0a 52 65 66 65 72 65 72 3a 20 68 74 ent··Ref erer: ht
0320 74 70 3a 2f 2f 6c 6f 63 61 6c 68 6f 73 74 2f 65 tp://loc alhost/e
0330 78 70 39 2f 6c 6f 67 69 6e 2e 70 68 70 0d 0a 41 xp9/logi n.php··A
0340 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 ccept-En coding:
0350 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 2c 20 62 gzip, de flate, b
0360 72 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 67 75 61 r··Accep t-Langua
0370 67 65 3a 20 65 6e 2d 55 53 2c 65 6e 3b 71 3d 30 ge: en-U S,en;q=0
0380 2e 39 0d 0a 0d 0a 75 73 65 72 6e 61 6d 65 3d 62 .9····us ername=b
0390 67 78 67 68 26 70 61 73 73 77 6f 72 64 3d 67 66 gxgh&pas sword=gf
03a0 6a 79 66 26 73 75 62 6d 69 74 3d 4c 6f 67 69 6e jyf&subm it=Login



Outcomes:

CO 4: Understand Security issues related to Software, Web and Networks.

Conclusion: (Conclusion to be based on the objectives and outcomes achieved**)**

Performed network sniffing using wire shark tool successfully.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:
Books/ Journals/ Websites:

1. https://www.wireshark.org/ (software)
2. https://en.wikipedia.org/wiki/Wireshark

3. https://www.wireshark.org/docs/
4. https://www.youtube.com/watch?v=UBfSgjUCEi0