Experiment No. 9

Title: Network Sniffing - Wireshark

Batch: B2 Roll No.: 16010420117 Experiment No.: 9

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 Wireshark . (www.wireshark.org) and download the stable version of Wireshark for 64 bit windows operating system.
2. After successful installation you will get the blue icon of Wireshark 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 color 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:

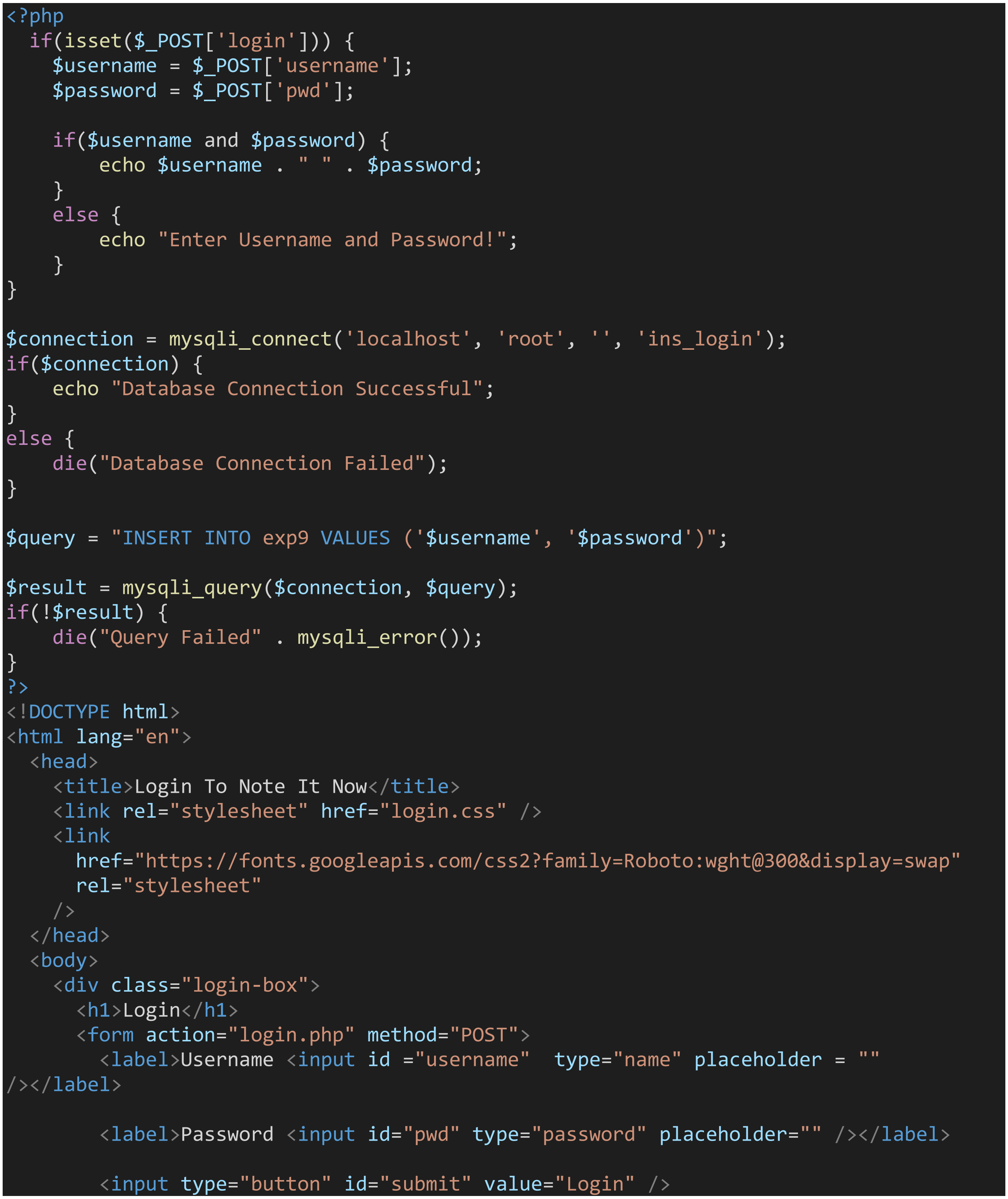
1) What is the difference between Burp suite and wire shark tools.

Ans: Burp Suite is an application penetration testing tool that functions as a web proxy server between the browser and target application. It acts on the application layer (OSI-7), finding exploits and vulnerabilities. It is an MITM tool that deals with the HTTP/HTTPS protocol, and is mainly used by application security professionals and developers.

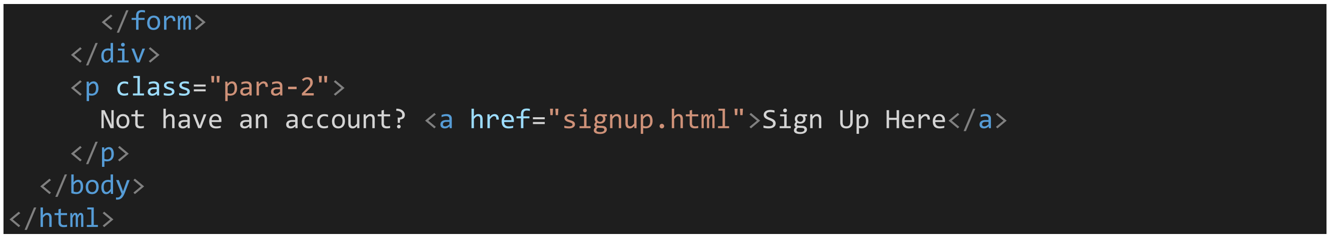
Wireshark (formerly Ethereal) is a network packet sniffer that mainly deals with raw data capture at the packet level. It can be used to analyse protocols other than HTTP/HTTPS/TCP, and acts at lower levels of OSI model (1 through 4) than Burp Suite. It is mainly used by network and security engineers.

Result: Task 1 implementation code and Task 2 screenshots.

PHP:

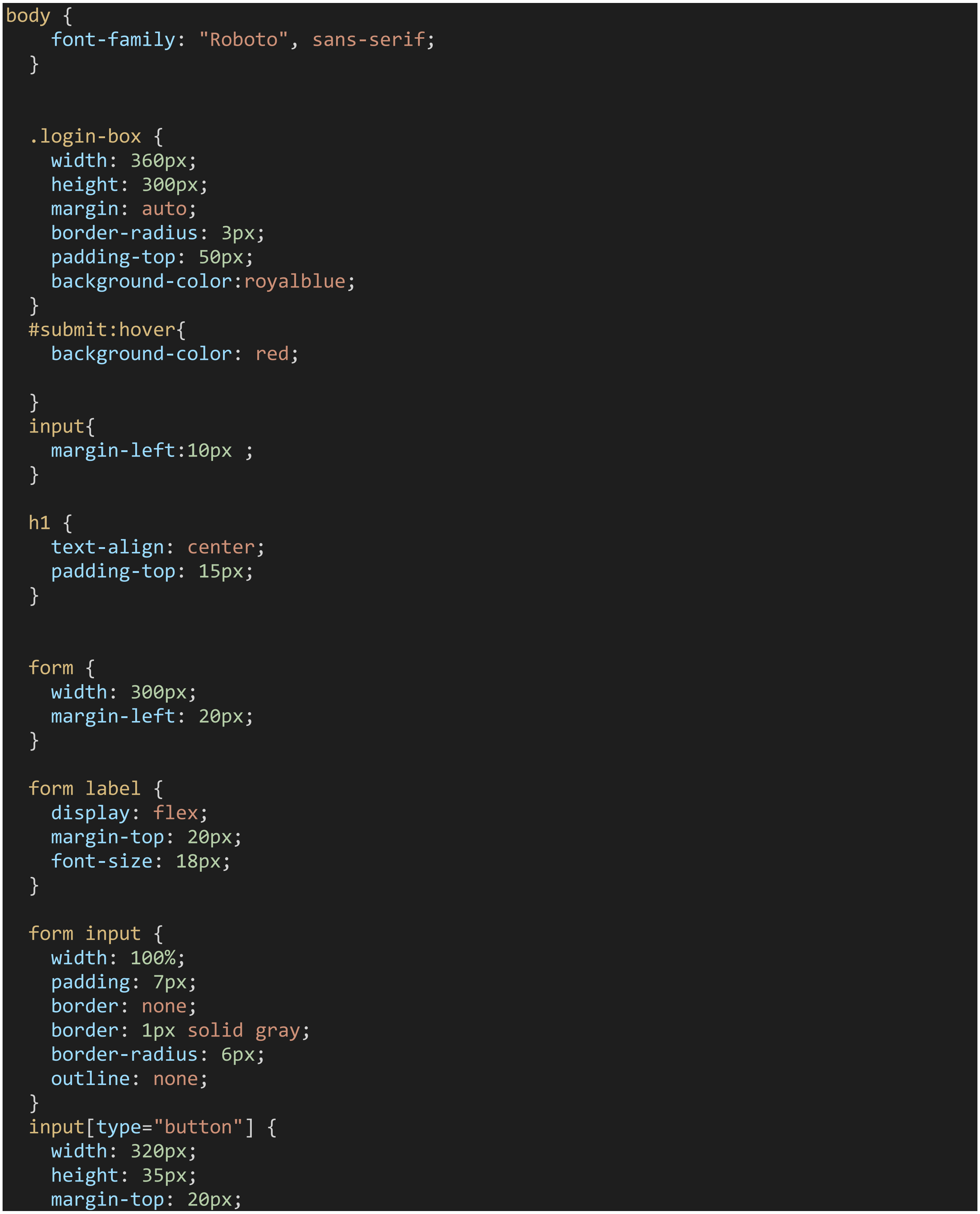


```
<?php
  if(isset($_POST['login'])) {
    $username = $_POST['username'];
    $password = $_POST['pwd'];

    if($username and $password) {
        echo $username . " " . $password;
    }
    else {
        echo "Enter Username and Password!";
    }
}

$connection = mysqli_connect('localhost', 'root', '', 'ins_login');
if($connection) {
    echo "Database Connection Successful";
}
else {
    die("Database Connection Failed");
}

$query = "INSERT INTO exp9 VALUES ('$username', '$password')";

$result = mysqli_query($connection, $query);
if(!$result) {
    die("Query Failed" . mysqli_error());
}
?>
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Login To Note It Now</title>
    <link rel="stylesheet" href="login.css" />
    <link
      href="https://fonts.googleapis.com/css2?family=Roboto:wght@300&display=swap"
      rel="stylesheet"
    />
  </head>
  <body>
    <div class="login-box">
      <h1>Login</h1>
      <form action="login.php" method="POST">
        <label>Username <input id ="username"  type="name" placeholder = ""
/></label>

        <label>Password <input id="pwd" type="password" placeholder="" /></label>

        <input type="button" id="submit" value="Login" />
```

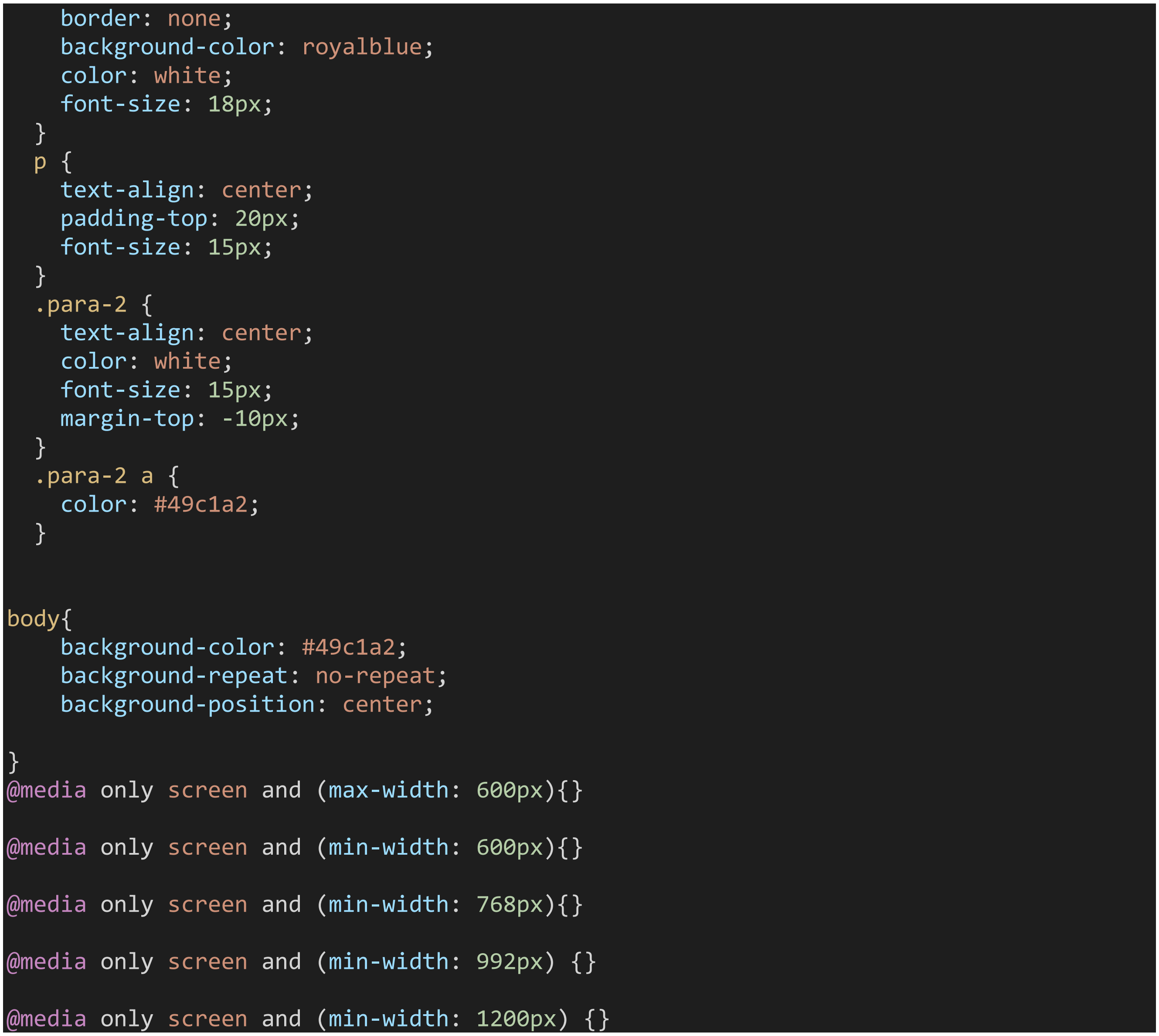


```
      </form>
    </div>
    <p class="para-2">
      Not have an account? <a href="signup.html">Sign Up Here</a>
    </p>
  </body>
</html>
```

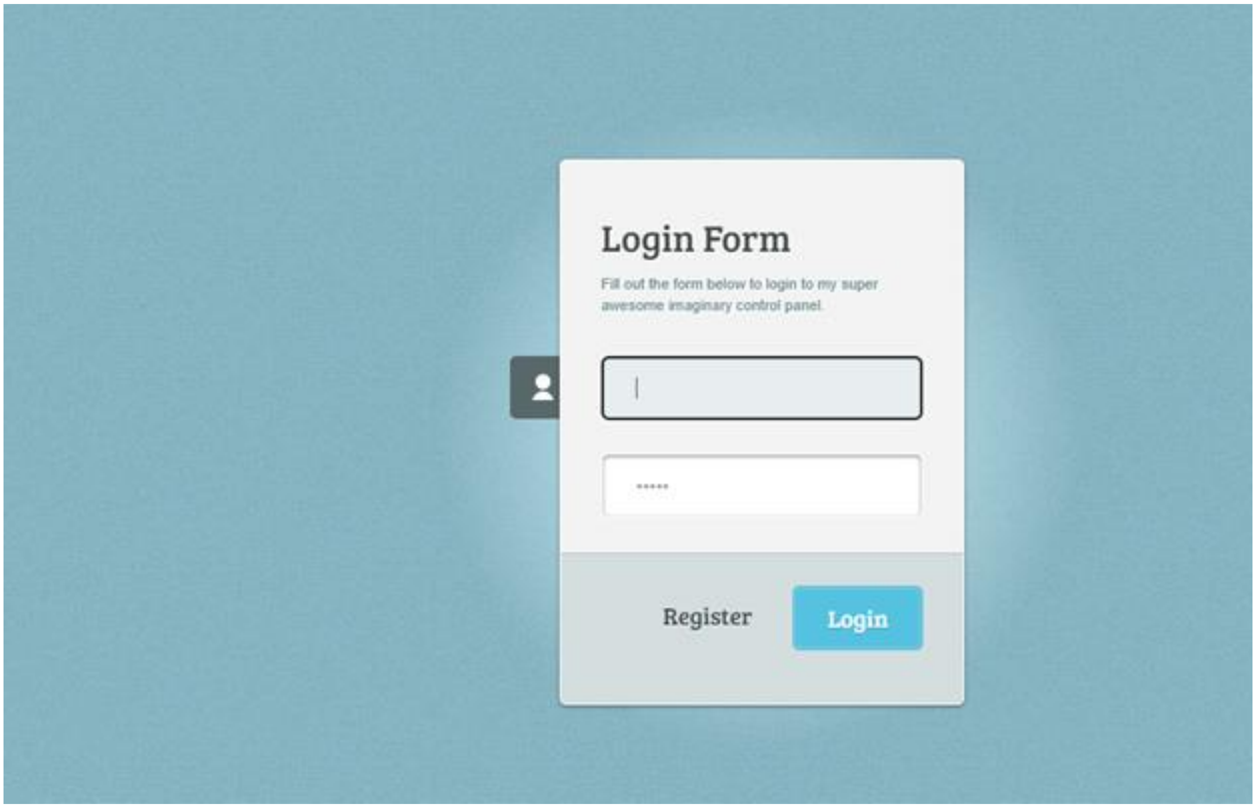
CSS:



```css
body {
    font-family: "Roboto", sans-serif;
  }


  .login-box {
    width: 360px;
    height: 300px;
    margin: auto;
    border-radius: 3px;
    padding-top: 50px;
    background-color:royalblue;
  }
  #submit:hover{
    background-color: red;

  }
  input{
    margin-left:10px ;
  }

  h1 {
    text-align: center;
    padding-top: 15px;
  }


  form {
    width: 300px;
    margin-left: 20px;
  }

  form label {
    display: flex;
    margin-top: 20px;
    font-size: 18px;
  }

  form input {
    width: 100%;
    padding: 7px;
    border: none;
    border: 1px solid gray;
    border-radius: 6px;
    outline: none;
  }
  input[type="button"] {
    width: 320px;
    height: 35px;
    margin-top: 20px;
```



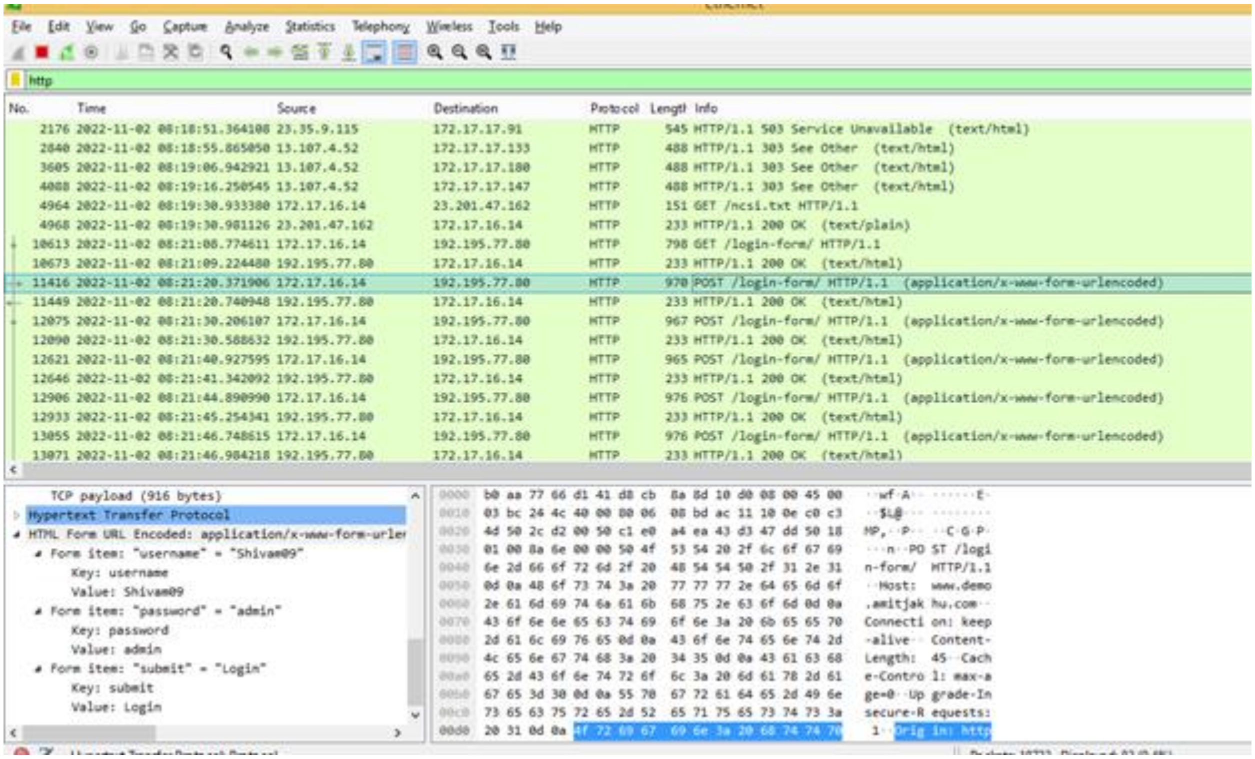
```
    border: none;
    background-color: royalblue;
    color: white;
    font-size: 18px;
  }
  p {
    text-align: center;
    padding-top: 20px;
    font-size: 15px;
  }
  .para-2 {
    text-align: center;
    color: white;
    font-size: 15px;
    margin-top: -10px;
  }
  .para-2 a {
    color: #49c1a2;
  }


body{
    background-color: #49c1a2;
    background-repeat: no-repeat;
    background-position: center;

}
@media only screen and (max-width: 600px){}

@media only screen and (min-width: 600px){}

@media only screen and (min-width: 768px){}

@media only screen and (min-width: 992px) {}

@media only screen and (min-width: 1200px) {}
```



Login Form

Fill out the form below to login to my super awesome imaginary control panel.

Output:



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.    Time                        Source          Destination     Protocol Length Info
 2176 2022-11-02 08:18:51.364108 23.35.9.115     172.17.17.91    HTTP      545 HTTP/1.1 503 Service Unavailable  (text/html)
 2840 2022-11-02 08:18:55.865050 13.107.4.52     172.17.17.133   HTTP      488 HTTP/1.1 303 See Other  (text/html)
 3605 2022-11-02 08:19:06.942921 13.107.4.52     172.17.17.180   HTTP      488 HTTP/1.1 303 See Other  (text/html)
 4088 2022-11-02 08:19:16.250545 13.107.4.52     172.17.17.147   HTTP      488 HTTP/1.1 303 See Other  (text/html)
 4964 2022-11-02 08:19:30.933380 172.17.16.14    23.201.47.162   HTTP      151 GET /ncsi.txt HTTP/1.1
 4968 2022-11-02 08:19:30.981126 23.201.47.162   172.17.16.14    HTTP      233 HTTP/1.1 200 OK  (text/plain)
10613 2022-11-02 08:21:08.774611 172.17.16.14    192.195.77.80   HTTP      798 GET /login-form/ HTTP/1.1
10673 2022-11-02 08:21:09.224480 192.195.77.80   172.17.16.14    HTTP      233 HTTP/1.1 200 OK  (text/html)
11416 2022-11-02 08:21:20.371906 172.17.16.14    192.195.77.80   HTTP      970 POST /login-form/ HTTP/1.1  (application/x-www-form-urlencoded)
11449 2022-11-02 08:21:20.740948 192.195.77.80   172.17.16.14    HTTP      233 HTTP/1.1 200 OK  (text/html)
12075 2022-11-02 08:21:30.206107 172.17.16.14    192.195.77.80   HTTP      967 POST /login-form/ HTTP/1.1  (application/x-www-form-urlencoded)
12090 2022-11-02 08:21:30.588632 192.195.77.80   172.17.16.14    HTTP      233 HTTP/1.1 200 OK  (text/html)
12621 2022-11-02 08:21:40.927595 172.17.16.14    192.195.77.80   HTTP      965 POST /login-form/ HTTP/1.1  (application/x-www-form-urlencoded)
12646 2022-11-02 08:21:41.342092 192.195.77.80   172.17.16.14    HTTP      233 HTTP/1.1 200 OK  (text/html)
12906 2022-11-02 08:21:44.890990 172.17.16.14    192.195.77.80   HTTP      976 POST /login-form/ HTTP/1.1  (application/x-www-form-urlencoded)
12933 2022-11-02 08:21:45.254341 192.195.77.80   172.17.16.14    HTTP      233 HTTP/1.1 200 OK  (text/html)
13055 2022-11-02 08:21:46.748615 172.17.16.14    192.195.77.80   HTTP      976 POST /login-form/ HTTP/1.1  (application/x-www-form-urlencoded)
13071 2022-11-02 08:21:46.984218 192.195.77.80   172.17.16.14    HTTP      233 HTTP/1.1 200 OK  (text/html)
```

```
    TCP payload (916 bytes)
▷ Hypertext Transfer Protocol
◢ HTML Form URL Encoded: application/x-www-form-urler
  ◢ Form item: "username" = "Shivam09"
      Key: username
      Value: Shivam09
  ◢ Form item: "password" = "admin"
      Key: password
      Value: admin
  ◢ Form item: "submit" = "Login"
      Key: submit
      Value: Login
```

```
0000  b0 aa 77 66 d1 41 d8 cb  8a 8d 10 d0 08 00 45 00   ··wf·A·· ······E·
0010  03 bc 24 4c 40 00 80 06  08 bd ac 11 10 0e c0 c3   ··$L@··· ········
0020  4d 50 2c d2 00 50 c1 e0  a4 ea 43 d3 47 dd 50 18   MP,··P·· ··C·G·P·
0030  01 00 8a 6e 00 00 50 4f  53 54 20 2f 6c 6f 67 69   ···n··PO ST /logi
0040  6e 2d 66 6f 72 6d 2f 20  48 54 54 50 2f 31 2e 31   n-form/  HTTP/1.1
0050  0d 0a 48 6f 73 74 3a 20  77 77 77 2e 64 65 6d 6f   ··Host:  www.demo
0060  2e 61 6d 69 74 6a 61 6b  68 75 2e 63 6f 6d 0d 0a   .amitjak hu.com··
0070  43 6f 6e 6e 65 63 74 69  6f 6e 3a 20 6b 65 65 70   Connecti on: keep
0080  2d 61 6c 69 76 65 0d 0a  43 6f 6e 74 65 6e 74 2d   -alive·· Content-
0090  4c 65 6e 67 74 68 3a 20  34 35 0d 0a 43 61 63 68   Length:  45··Cach
00a0  65 2d 43 6f 6e 74 72 6f  6c 3a 20 6d 61 78 2d 61   e-Contro l: max-a
00b0  67 65 3d 30 0d 0a 55 70  67 72 61 64 65 2d 49 6e   ge=0··Up grade-In
00c0  73 65 63 75 72 65 2d 52  65 71 75 65 73 74 73 3a   secure-R equests:
00d0  20 31 0d 0a 4f 72 69 67  69 6e 3a 20 68 74 74 70    1··Orig in: http
```

Outcomes: Understand Security issues related to Software, Web and Networks

Conclusion: **(**Conclusion to be based on the objectives and outcomes achieved**)**
Successfully performed network sniffing using wire shark tool

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