4. File Upload Vulnerability

Objective: To understand the risks associated with file upload

Tools: DVWA, Burp Suit, a vulnerable web application

File upload vulnerability is a security risk where an application or website allows users to upload files, and the application does not properly validate or handle those files. This can lead to various types of attacks, such as:

1. **Malware Distribution**: Attackers might upload malicious files (e.g., viruses, worms) that can infect other users or systems.
2. **Remote Code Execution**: If the application allows certain types of files to be executed on the server (e.g., PHP scripts), an attacker could upload a script and execute it, leading to server compromise.
3. **Data Theft**: An attacker could upload a file that exploits vulnerabilities in the server to access sensitive information.
4. **Denial of Service (DoS)**: Uploading large or numerous files can exhaust server resources, causing it to become unresponsive.

Common Causes of File Upload Vulnerabilities:

- **Insufficient Validation**: Not checking file types, sizes, or contents properly.
- **Lack of Proper File Handling**: Not ensuring files are not executable or accessible on the server.
- **Insecure File Storage**: Storing files in a web-accessible directory or using predictable file paths.

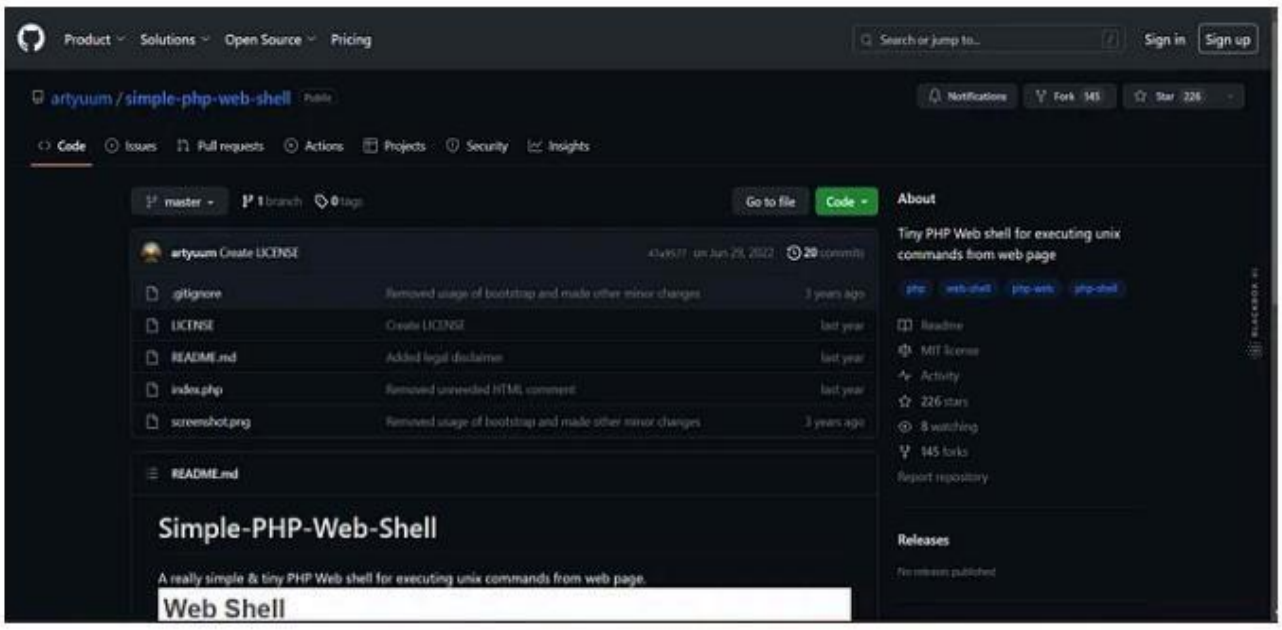
Mitigation Strategies:

1. **Validate and Sanitize**: Ensure file uploads are checked for allowed types, sizes, and contents. Use white lists for file types and reject anything that doesn't meet the criteria.
2. **Use Secure Storage**: Store files in a non-web-accessible directory and use random, unique names for files.
3. **Limit Executable Content**: Ensure uploaded files cannot be executed on the server. For example, use a file extension filter and deny execution of scripts.
4. **Implement Size and Rate Limits**: Restrict the size of files and the number of files a user can upload within a certain timeframe.
5. **Regular Security Reviews**: Periodically review and test your file upload functionality to identify and fix potential vulnerabilities.

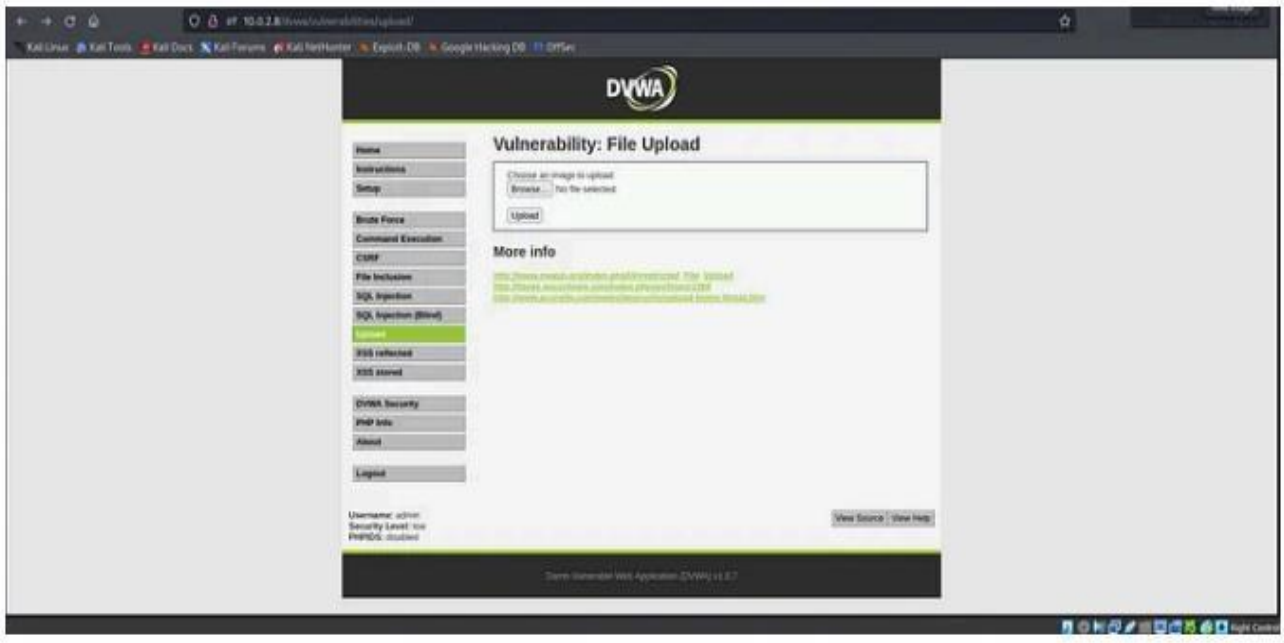
Properly addressing file upload vulnerabilities is crucial to maintaining the security and integrity of your systems and protecting user data.

DVWA → File Upload Vulnerabilities
(Low-Medium-High Security)

1. **Low Security**

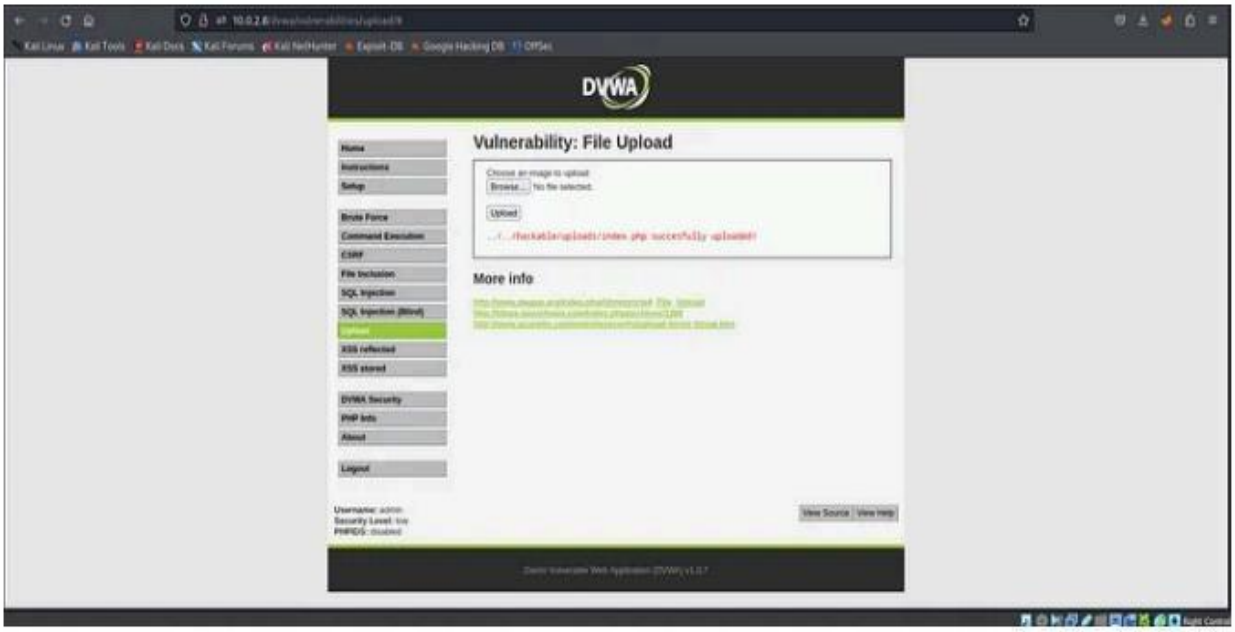


First, go to the GitHub website: https://github.com/artyuum/simple-php-webshell. From there, download the shell. Then, download the index.php file.

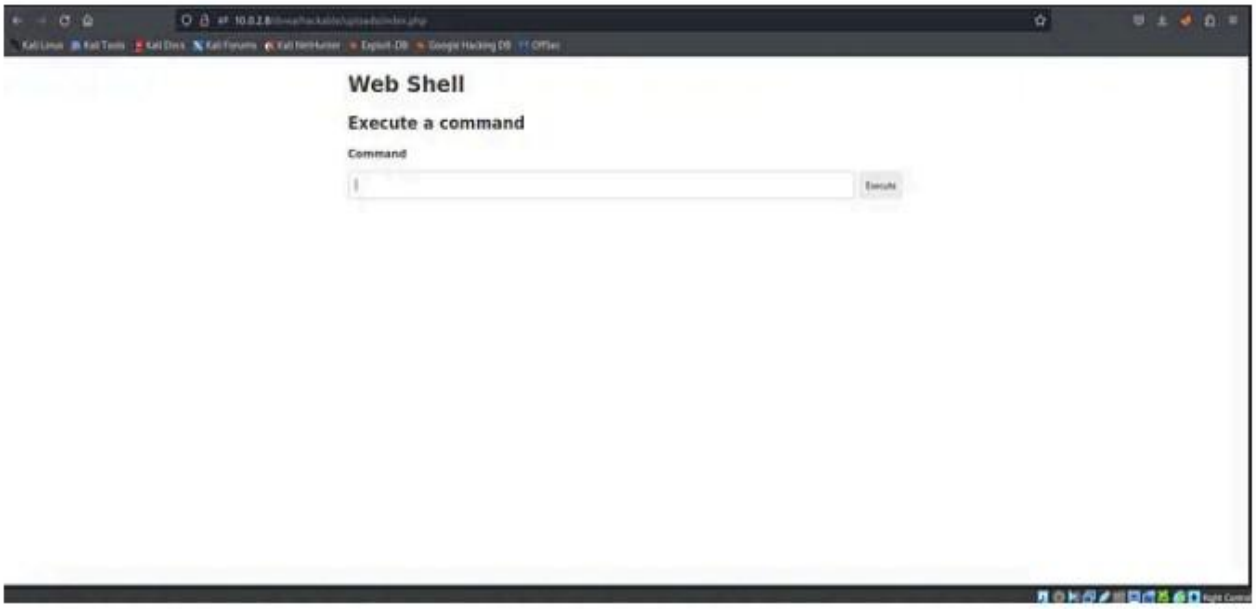


Make sure you first lower the security level on the DVWA site.

Then, go to the "Upload" section and press the "Upload" button to upload your file. Choose the downloaded shell file and proceed.

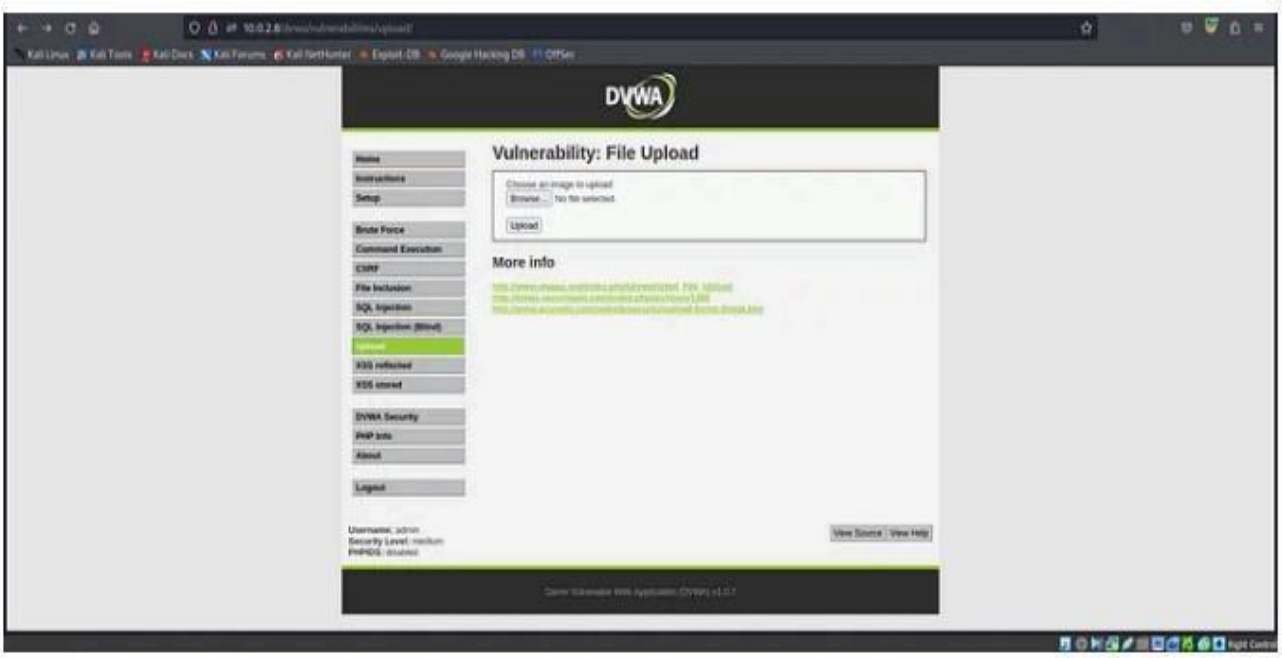


You will land with this screen. Afterward, copy the part written in red text. Delete the "#" from the URL above and paste the copied text.

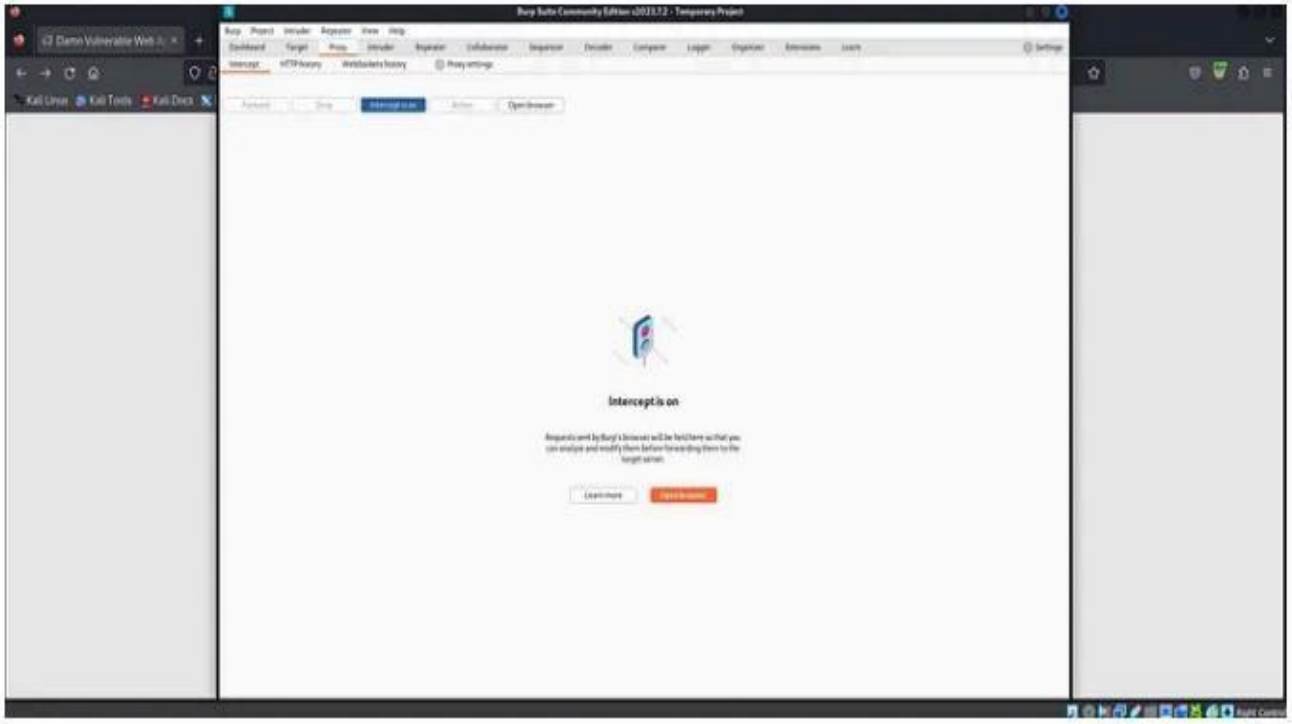


If you see something like this, congratulations, you have succeeded. You can execute desired commands from here.

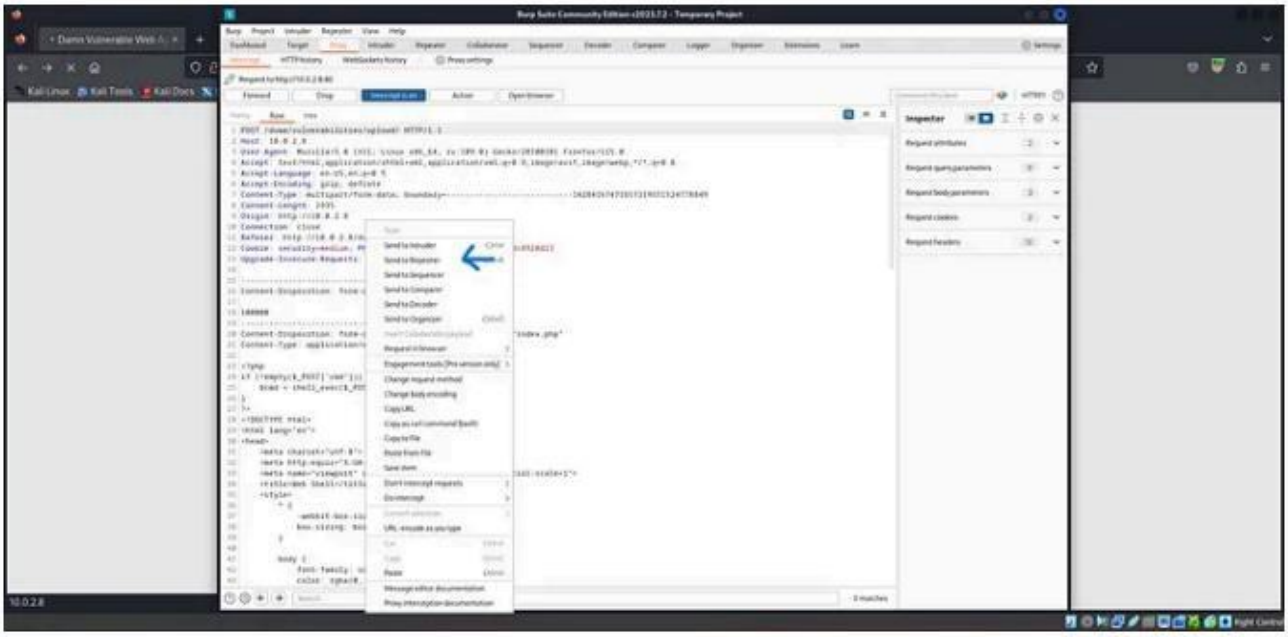
2. Medium Security



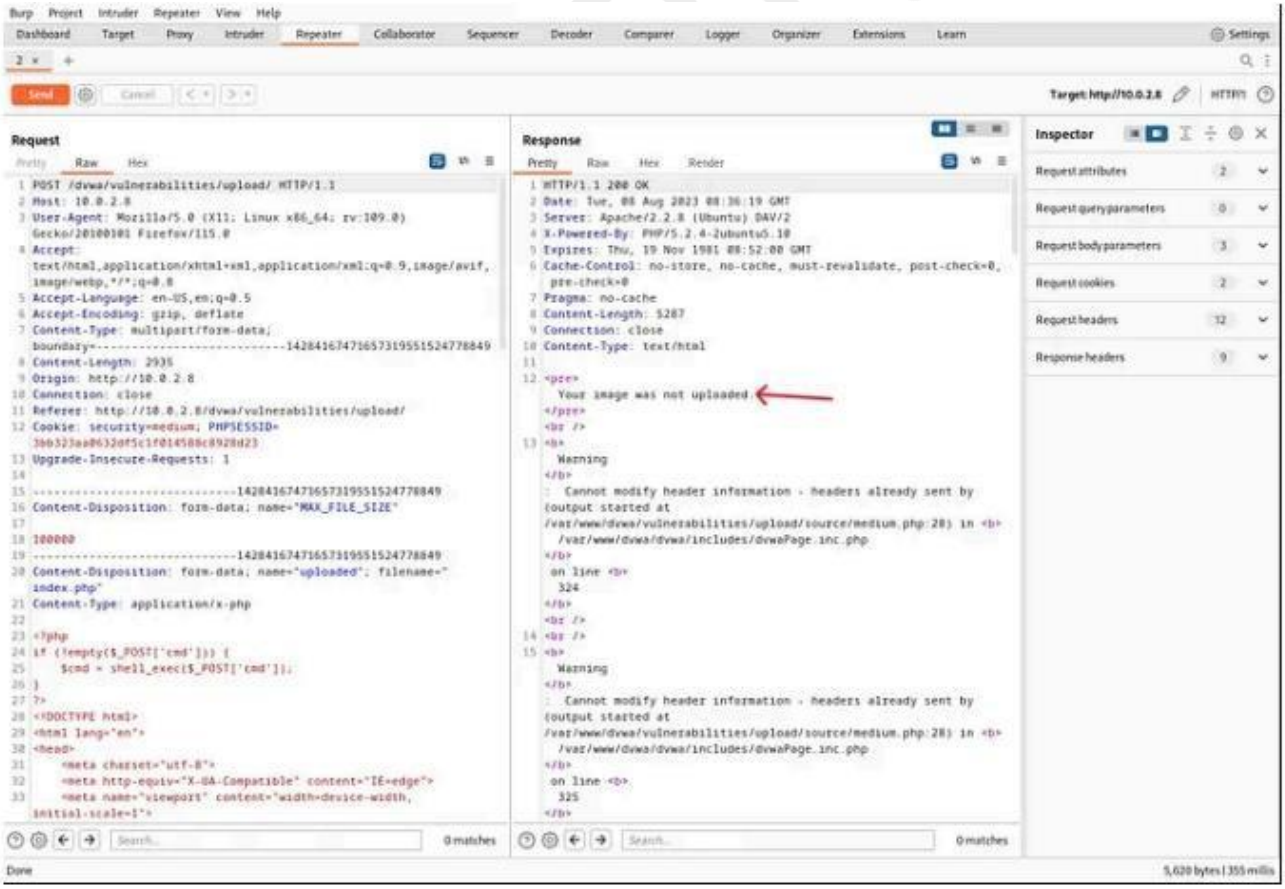
We have turned the security level back to normal and arrived here.



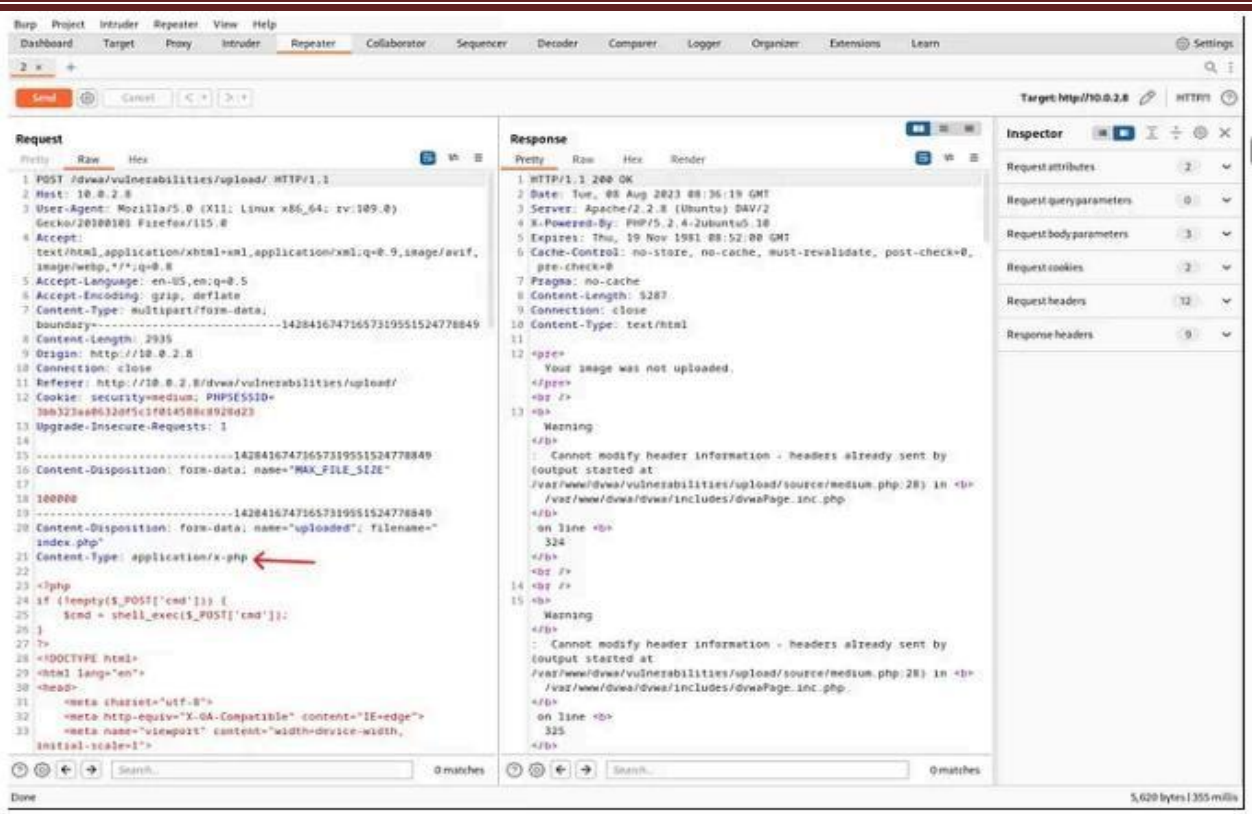
We open our Burp Suite and enable the intercept. Then, we upload our file.



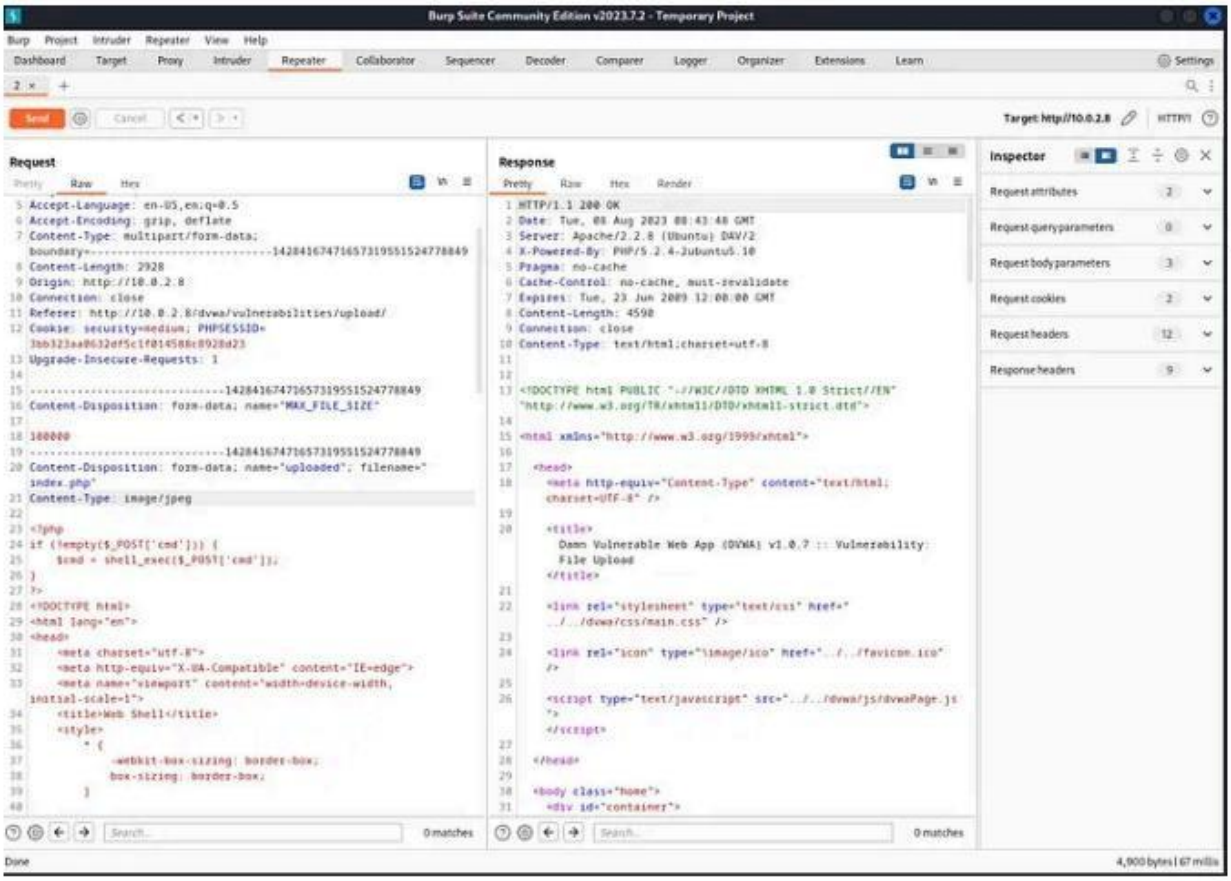
"When you click 'Upload,' you will encounter a screen like this. Right-click on that screen and send it to the repeater. Then, turn off the intercept.



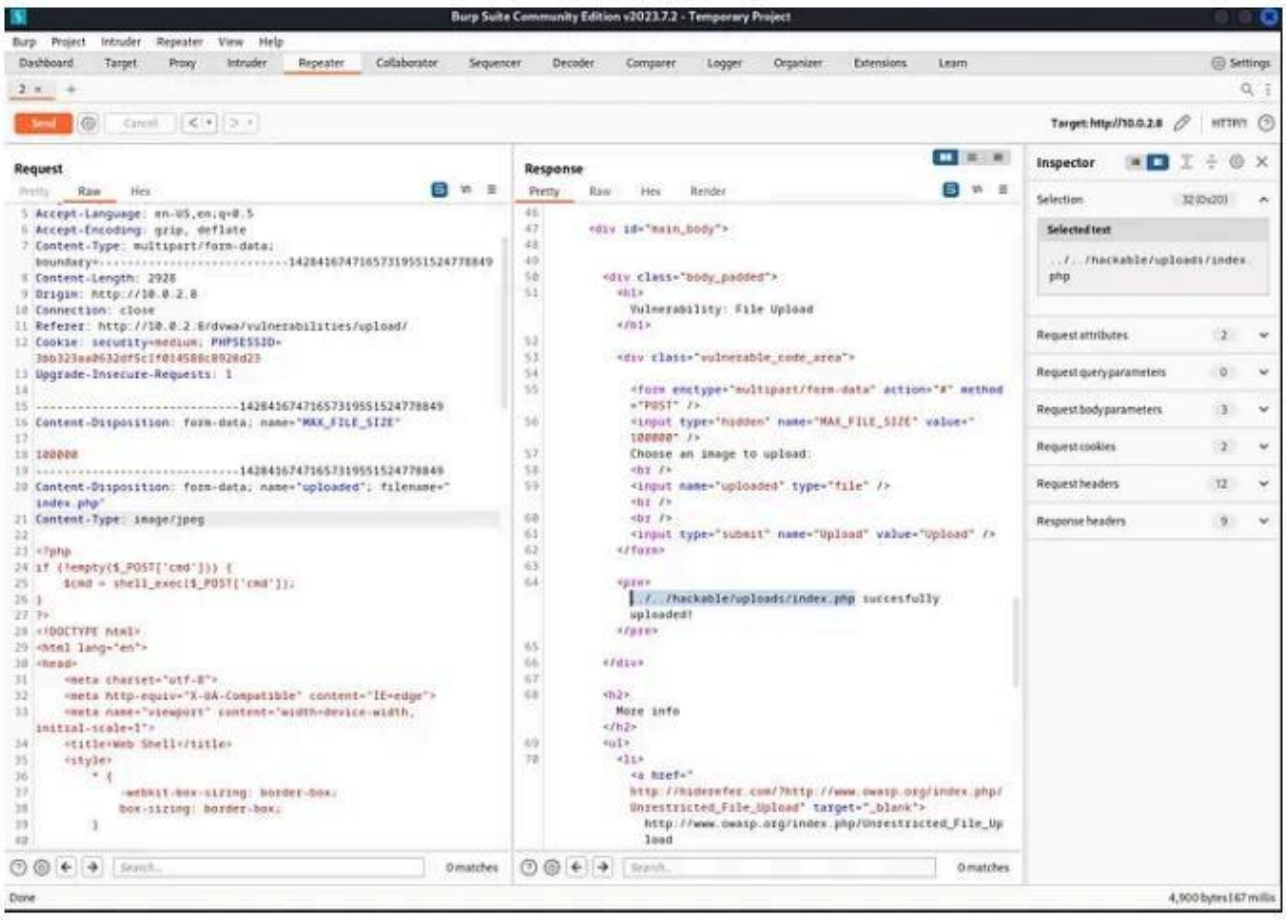
"As seen, when we press 'Send,' it gives us an error message saying 'your image was not uploaded.' From this, we understand that our file needs to be an image."



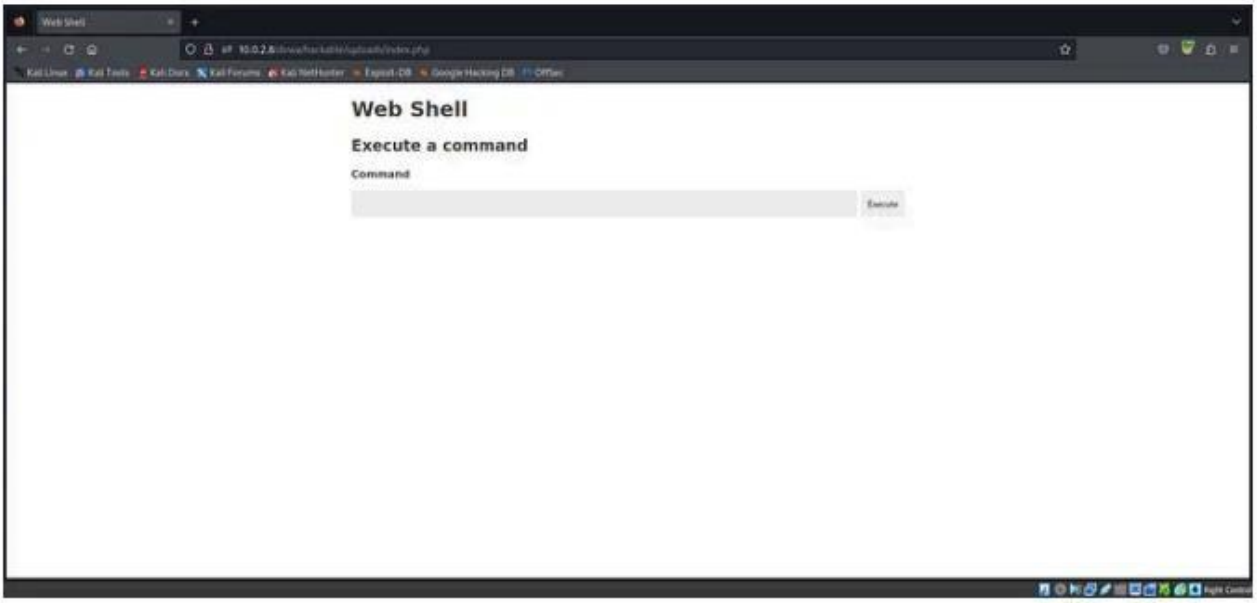
"To fix this, we write 'image/jpeg' in the indicated place on the photo. When we do this, the website will treat our PHP file as if it's an image. Then, we can access the content.



"As you can see, the file we uploaded successfully entered the system."



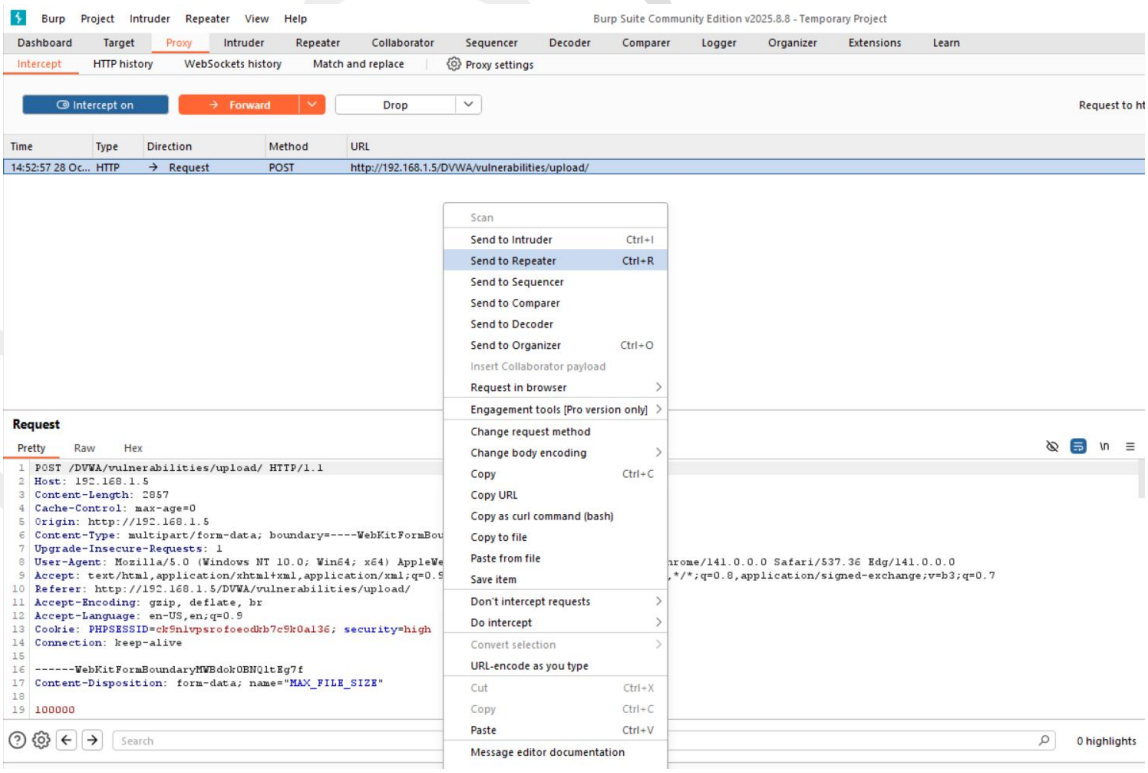
"We find out where our file is uploaded from the Response section. As shown here, then we copy it and paste the URL into our document."



Congratulations!

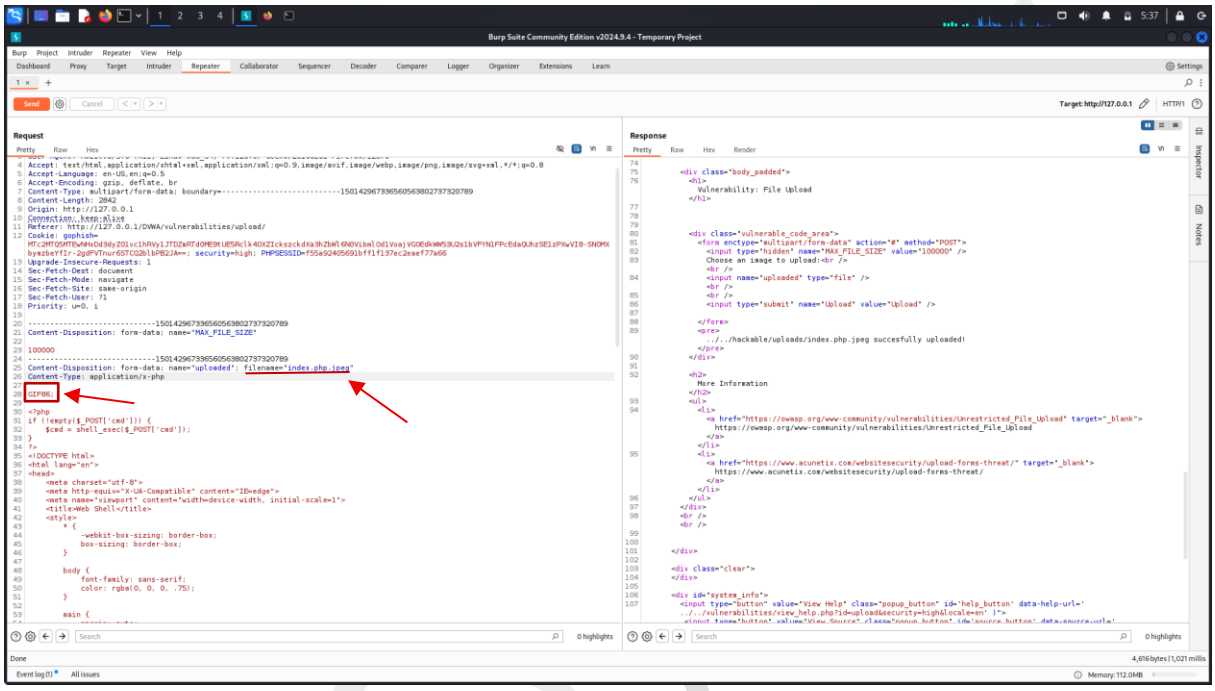
3. High Level Security

"Once again, we raise our security level to High and proceed. We uploaded our file again, but we encountered an error once more.Then, we reactivate the intercept in Burp Suite. We send our result to the repeater again."

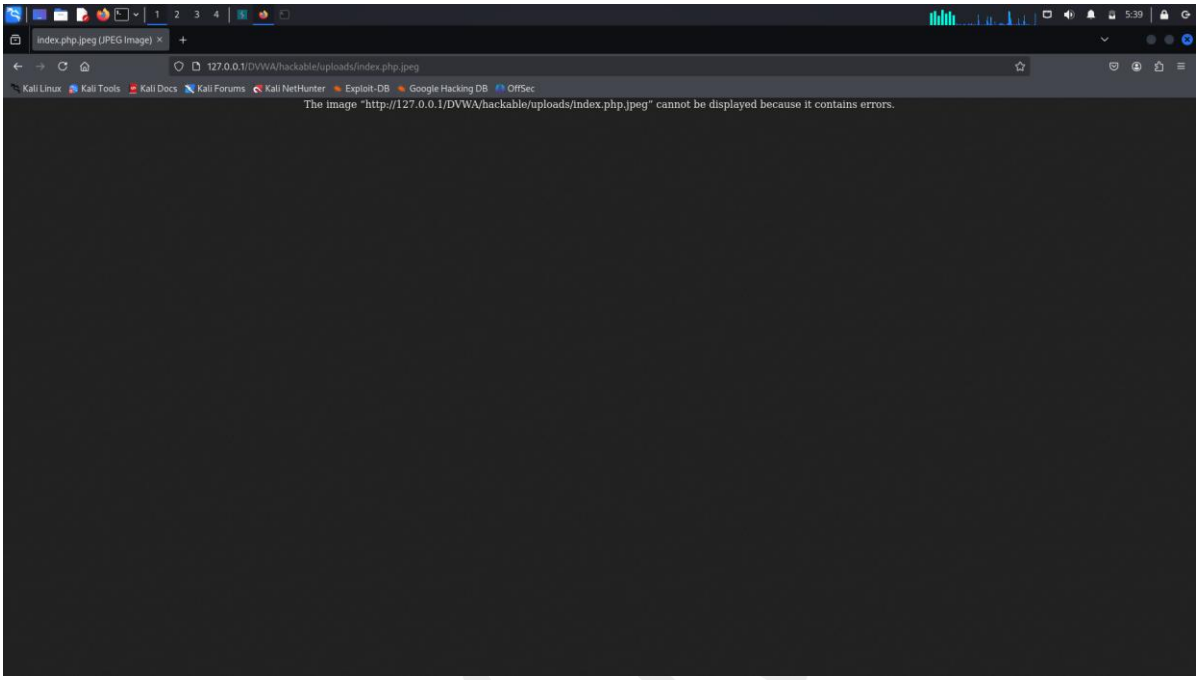


“This time, we don’t need to deal with the content type. The site doesn’t check it. After that, when we press ‘‘Change the filename index.php to index.php.jpeg and add some image content(GIF86;) at above the php code’’

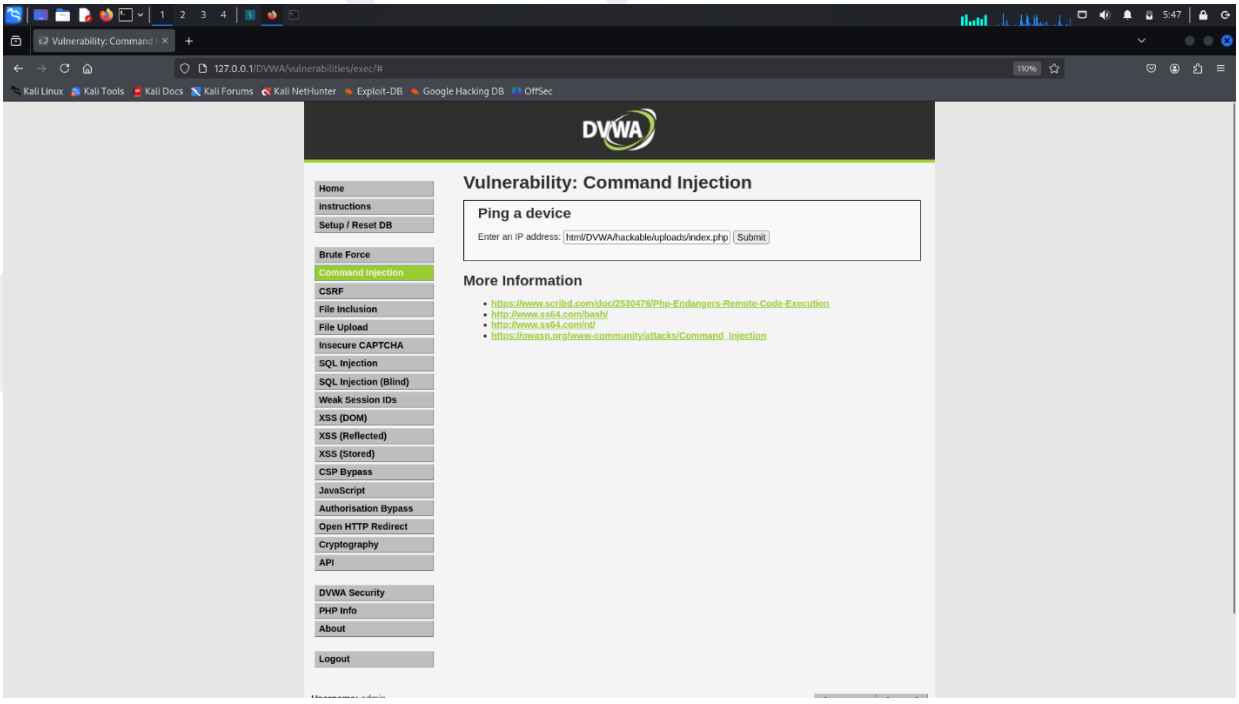
‘Send,’ it gets uploaded.”



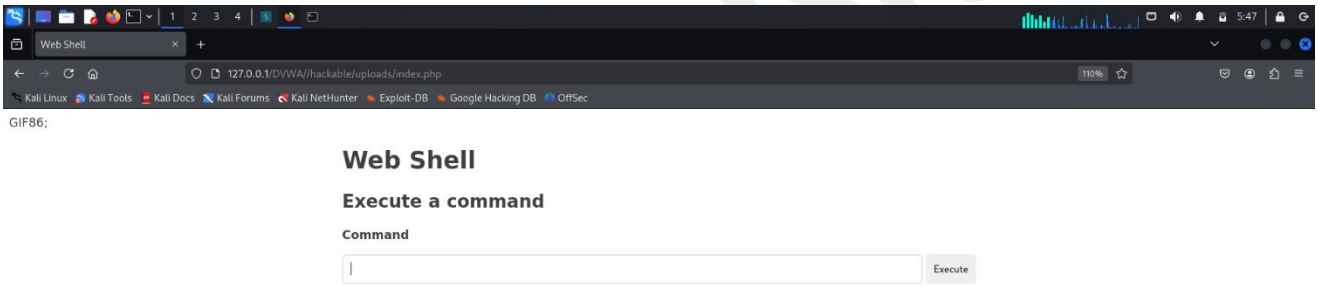
“We find out where our file is uploaded from the Response section. As shown here, then we copy it and paste the URL into our domain.”



Even though we can see the page saying cannot be display because it contains error. So go back to dvwa command injection and type : "127.0.0.1 |mv /var/www/html/DVWA/hackable/uploads/index.php.jpeg /var/www/html/DVWA/hackable/uploads/index.php", This command is used to rename index.php.jpeg to index.php .



Then Enter the domain : “127.0.0.1/DVWA/hackable/uploads/index.php” Which is copied from brupsuite response just remove .jpeg



Our result is that we can now execute the desired command.