**Name: Samarth Jain**

**4SU20CS081**

**SDMIT Ujire**

**Professor: Bharath Kumar**

**Date: 16/09/2023**

**Assignment Details**

Assigned Date: 15/09/2023

Due Date: 16/09/2023

Topic: Remote File Inclusion

**Introduction**

File Inclusion, in the context of web security, refers to the ability of a web application to include or reference files, often external files, in its operation. These files can be either local files (local file inclusion, LFI) or remote files (remote file inclusion, RFI). File inclusion vulnerabilities can be exploited by attackers to execute malicious code, access sensitive data, or compromise the security of a web application.

There are two main types of file inclusion vulnerabilities:

1. Local File Inclusion (LFI)
2. Remote File Inclusion (RFI)

Remote File Inclusion (RFI) is a type of security vulnerability that occurs in web applications when an attacker is able to include or execute remote files on a server. This vulnerability typically arises from improper input validation or inadequate security measures in a web application. RFI can have serious consequences, as it allows an attacker to inject malicious code or scripts from a remote server into the web application, potentially compromising the server's integrity and data confidentiality.

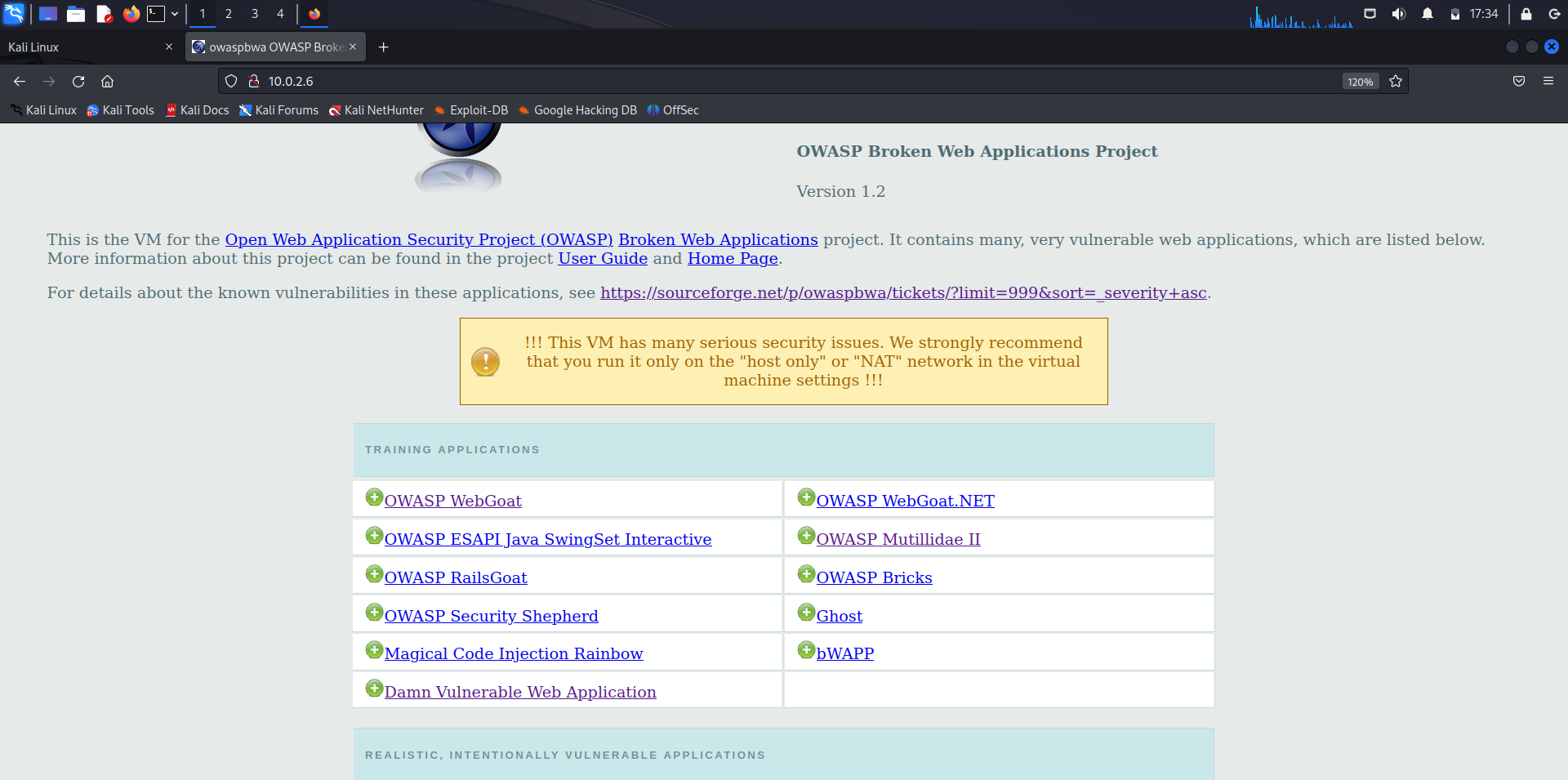
In a typical RFI attack, an attacker provides a URL or file path as input to a vulnerable web application. This input is then processed by the application without proper validation, and the application includes the remote file specified by the attacker. This can lead to the execution of malicious scripts, unauthorized access to sensitive files, or even full control of the web server, depending on the attacker's intentions and the security measures in place.

To prevent RFI attacks, web developers should implement robust input validation and sanitization techniques to ensure that user-supplied data is not used to include or execute remote files. Additionally, web servers should be configured with tight security settings to restrict access to sensitive directories and files. Regular security audits and monitoring can also help detect and mitigate RFI vulnerabilities before they can be exploited by attackers.

**Content**

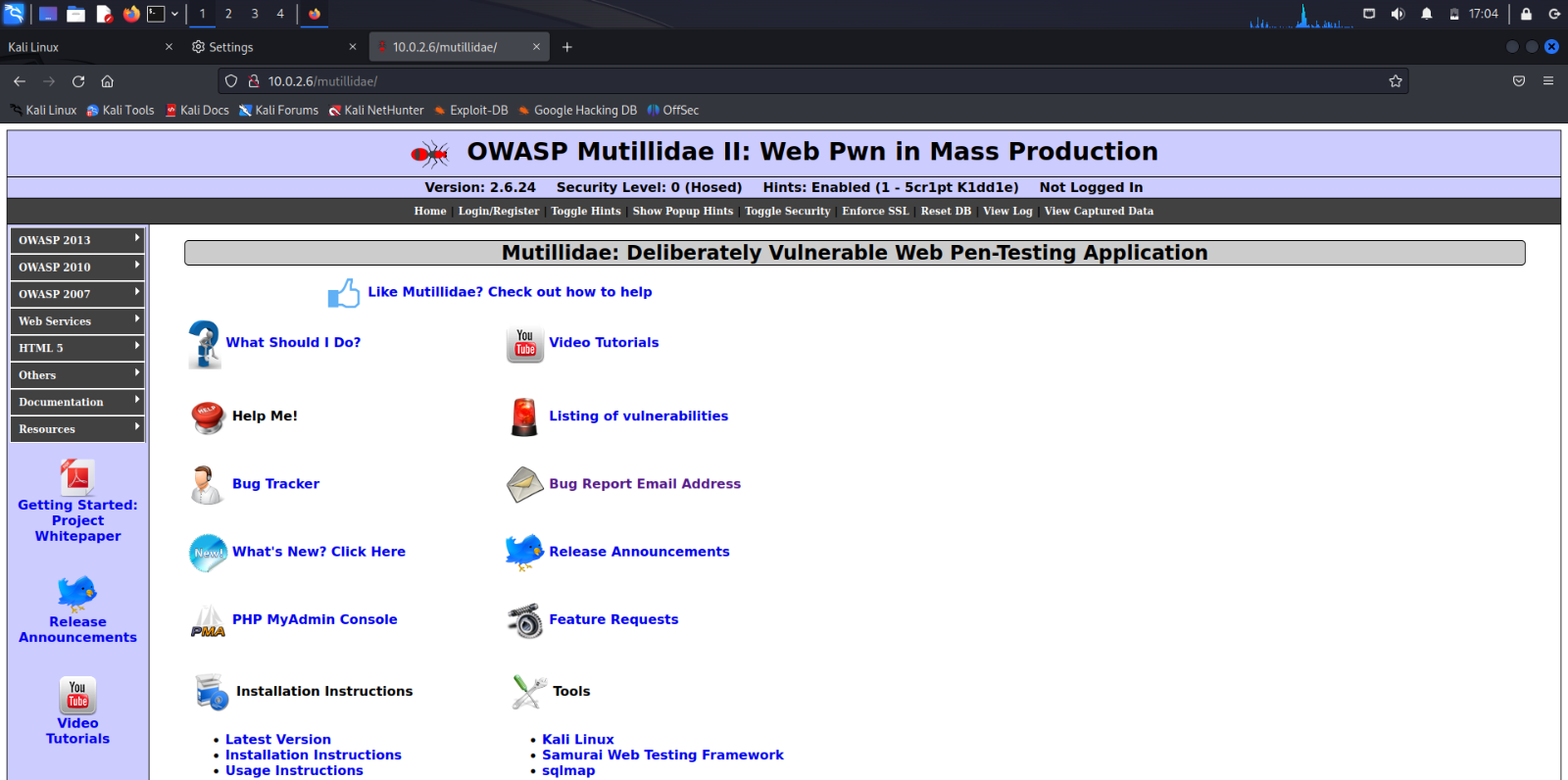
Open the OWASP Broken Web Application Project

Home page of Open Web Application Security Project (OWASP) Broken Web Application project



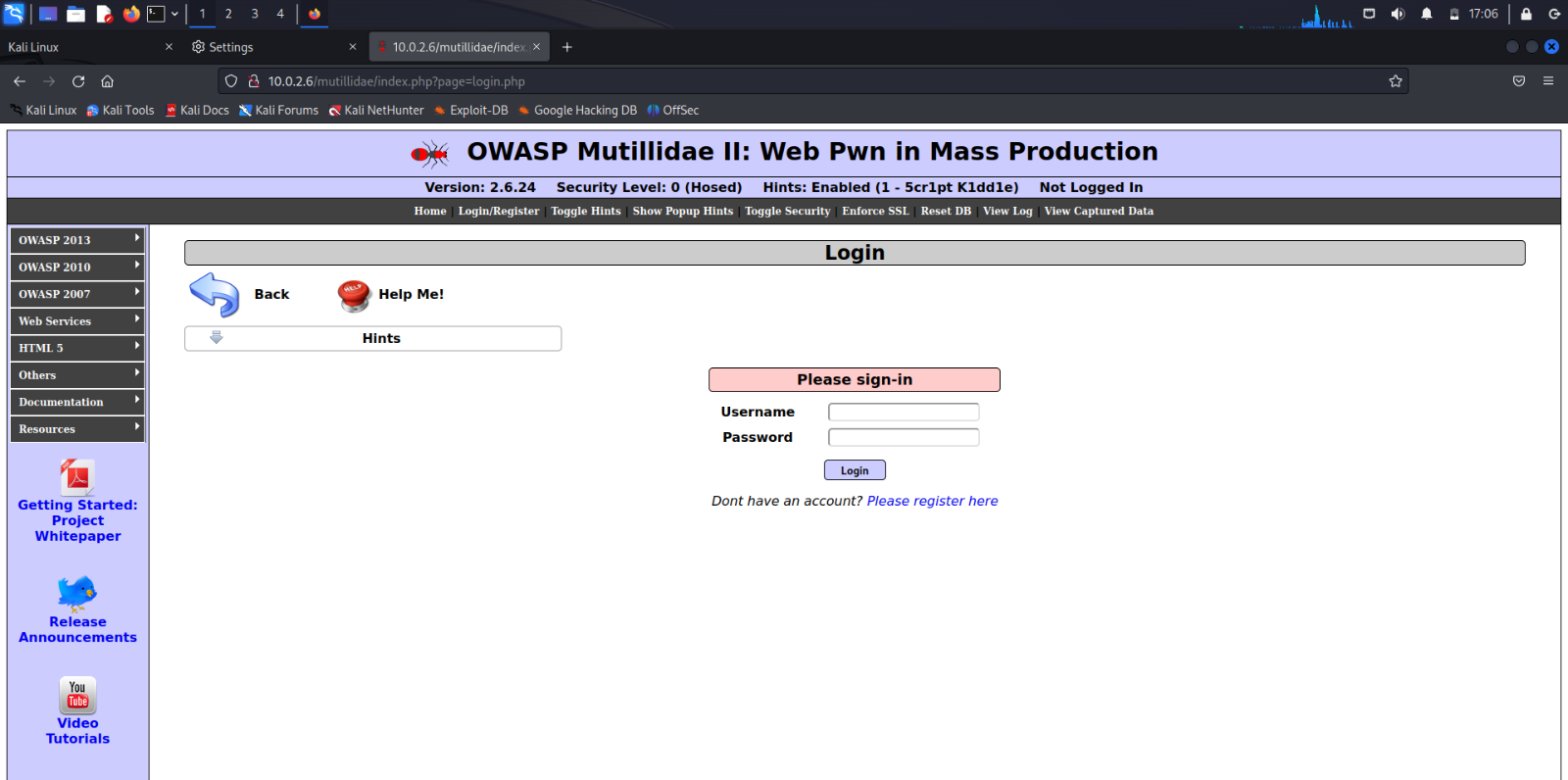
Open OWASP Mutillidae II

Home Page of OWASP Mutillidae II Web Application



Login page of Mutillidae II web application

As we can see in the URL of the webpage, the Login page is executing a php file called login.php



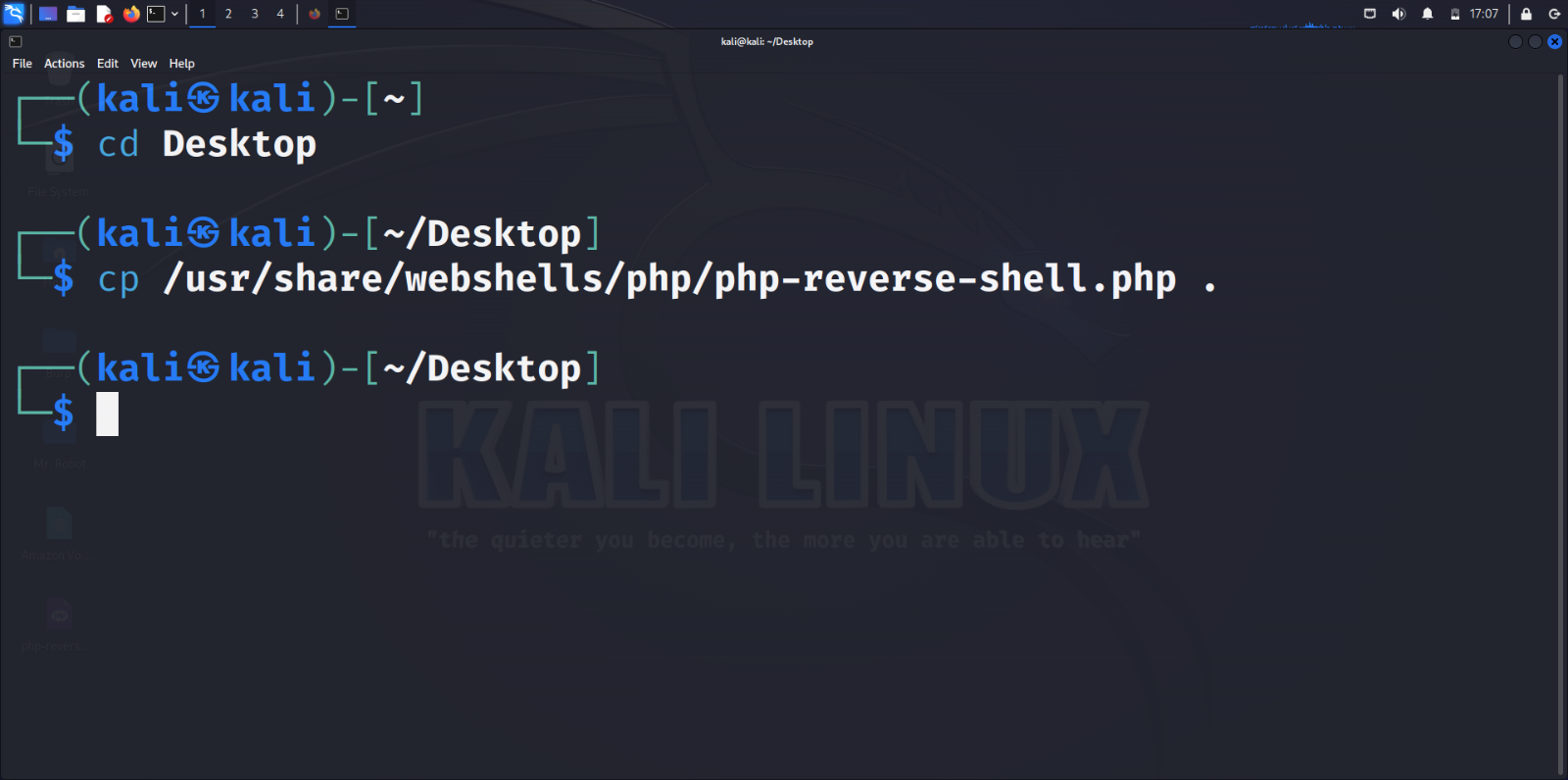
Create a file to be remotely included in the victim’s web application

Kali Linux contains a file called php-reverse-shell.php

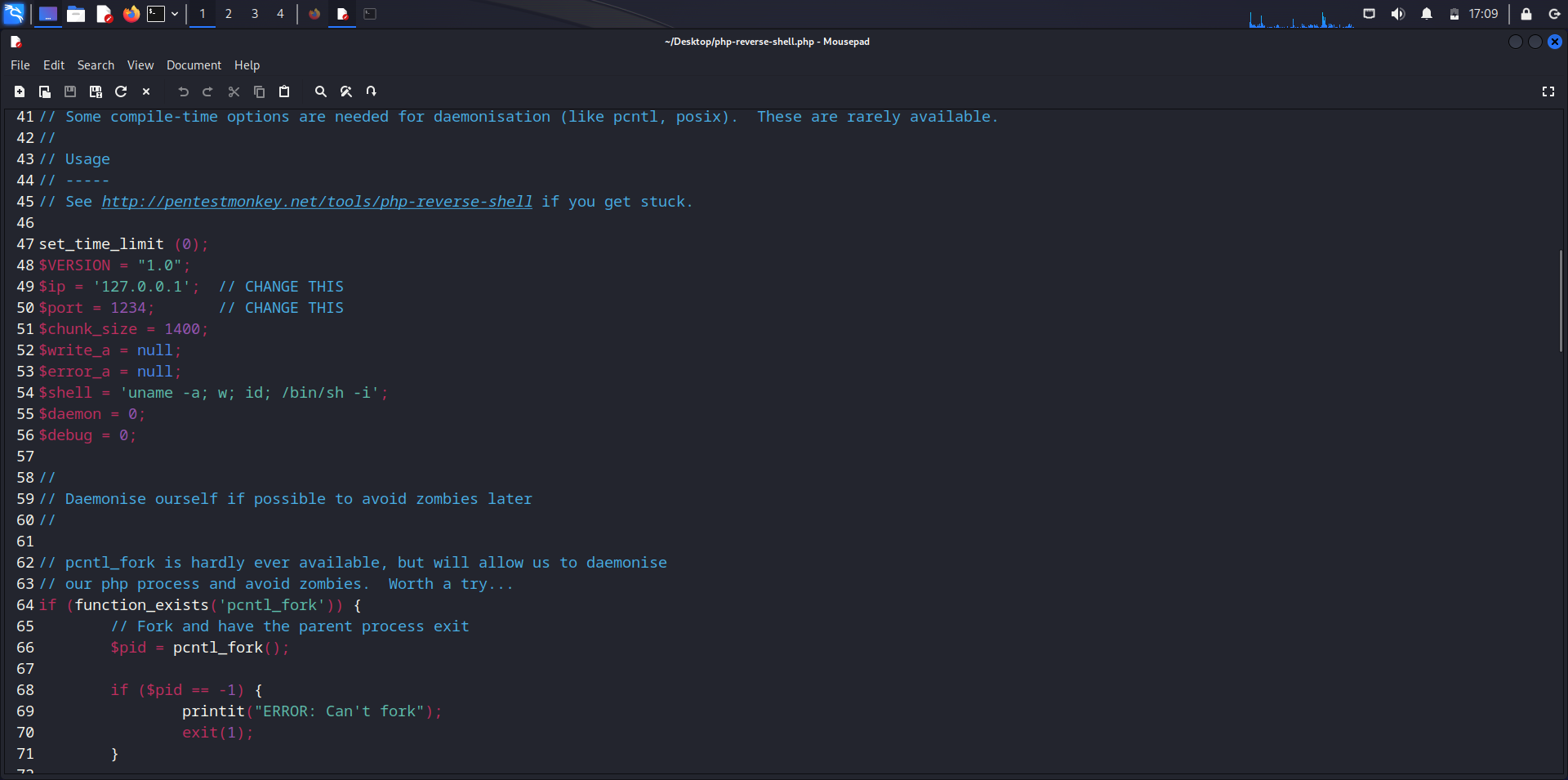
Copy the file onto the present working directory.

Syntax: cp source destination

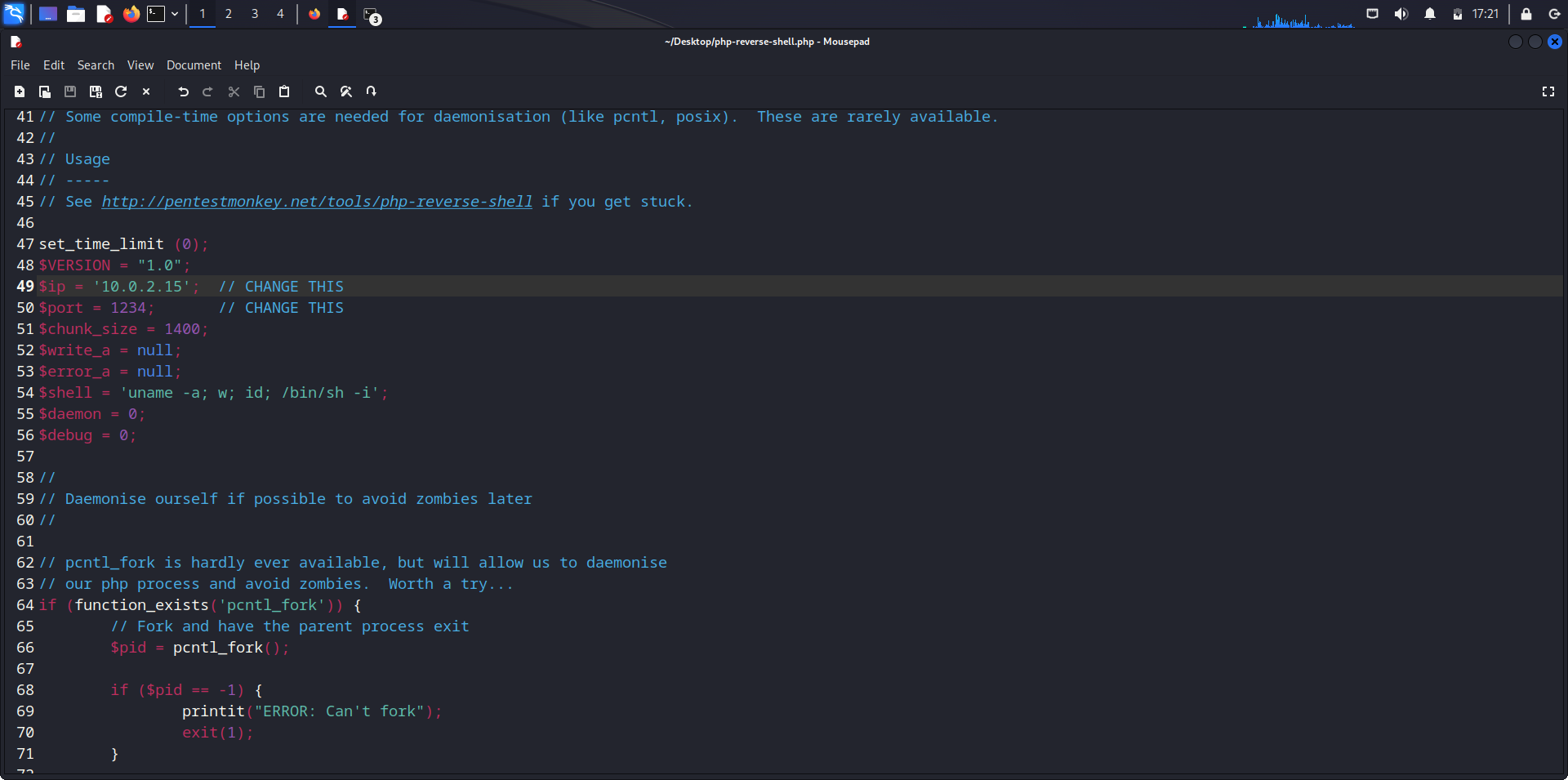
Command: cp /usr/share/webshells/php/php-reverse-shell.php .



The php-reverse-shell.php file is typically a script or PHP program that is used in the context of penetration testing or ethical hacking. It is designed to create a reverse shell connection between a target machine (the one running the PHP script) and an attacker-controlled machine. This type of script can be used to gain unauthorized access to a remote server or system for security testing purposes.



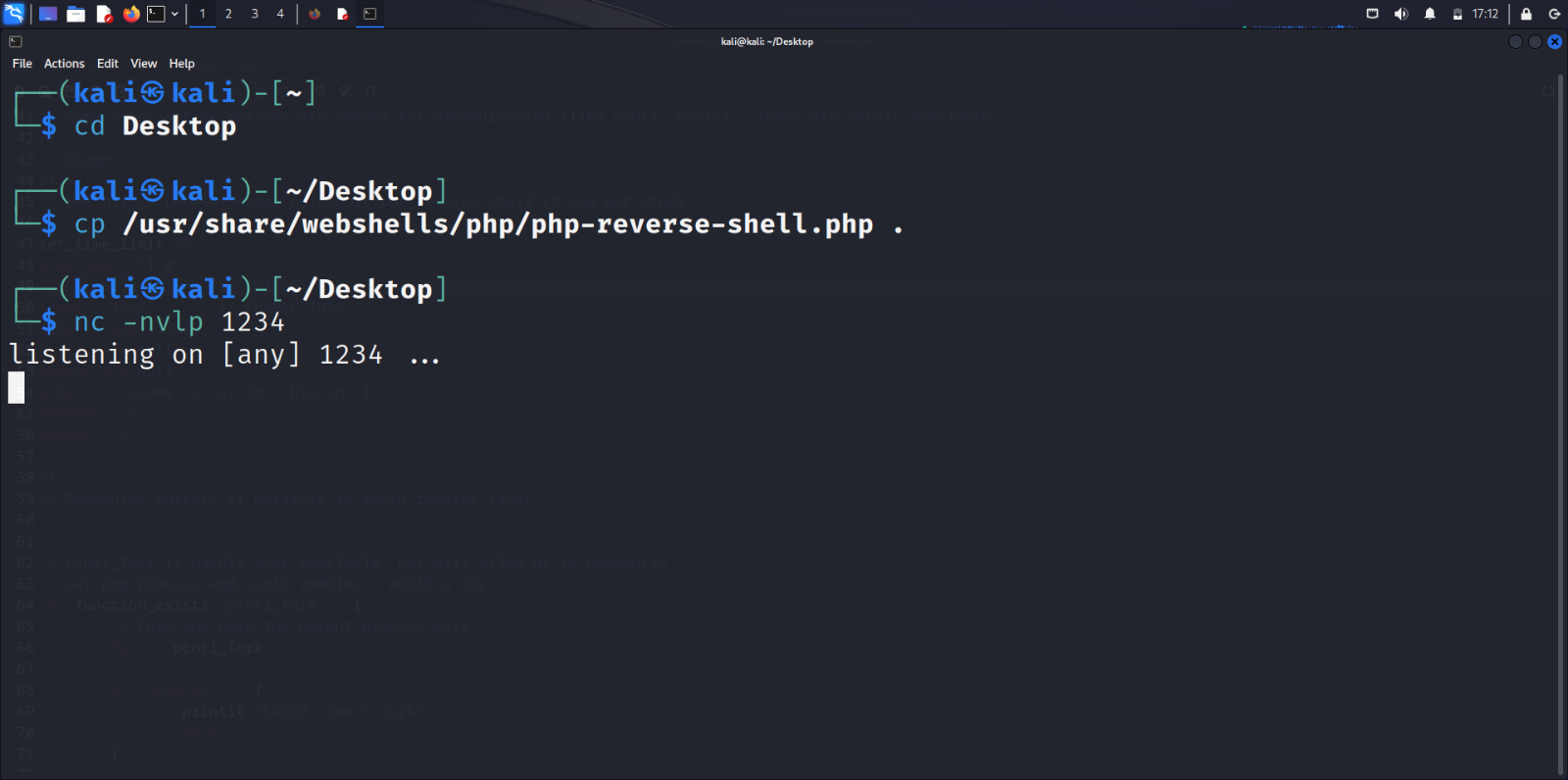
Change the $ip parameter of the file to the IP address of the remote system which is to gain shell access to the web application. Save the file after modification.



Setup a listener on port number specified in the php file for incoming network connections.

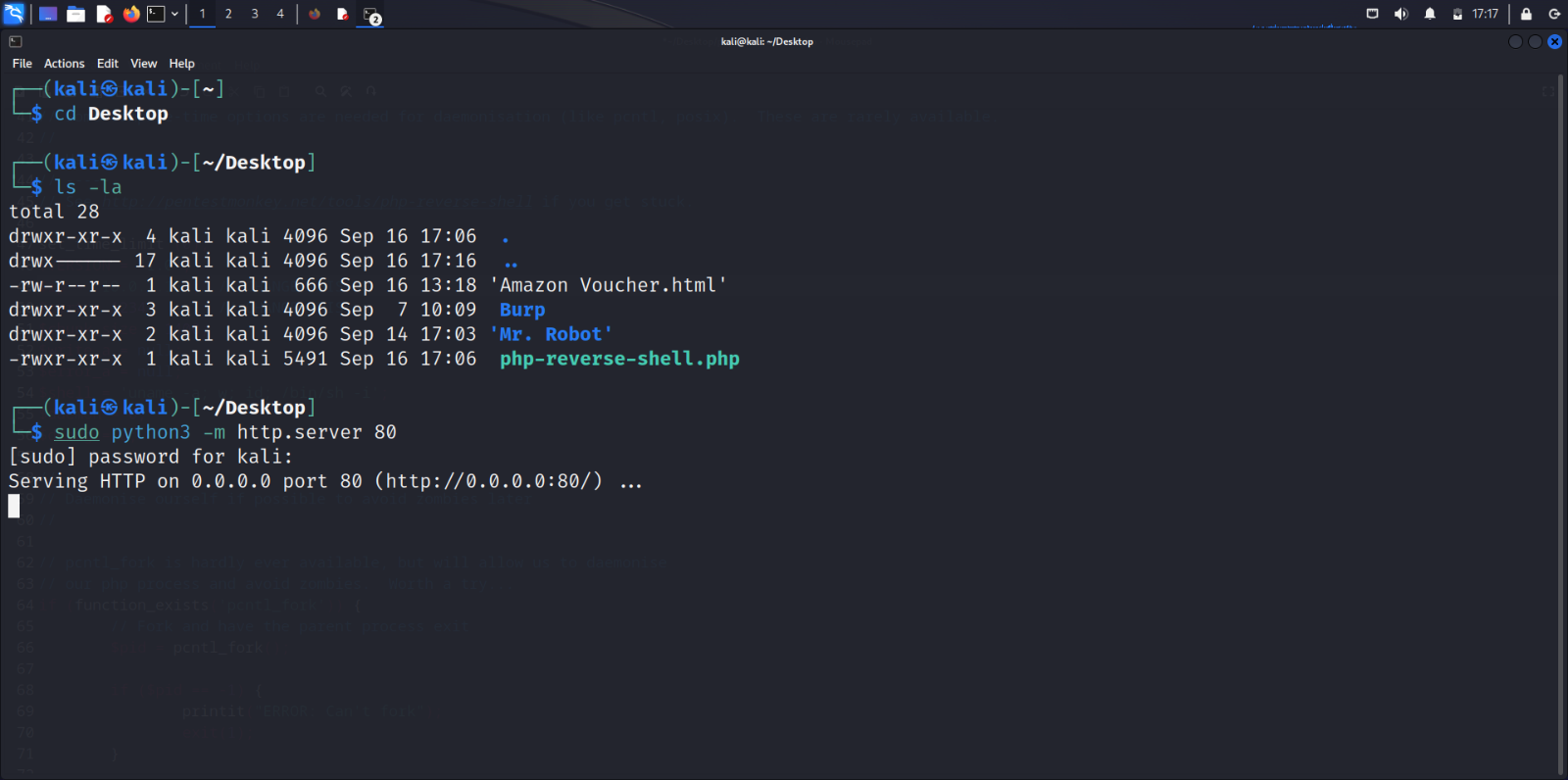
Syntax: nc -nvlp <port number>

Command: nc -nvlp 1234



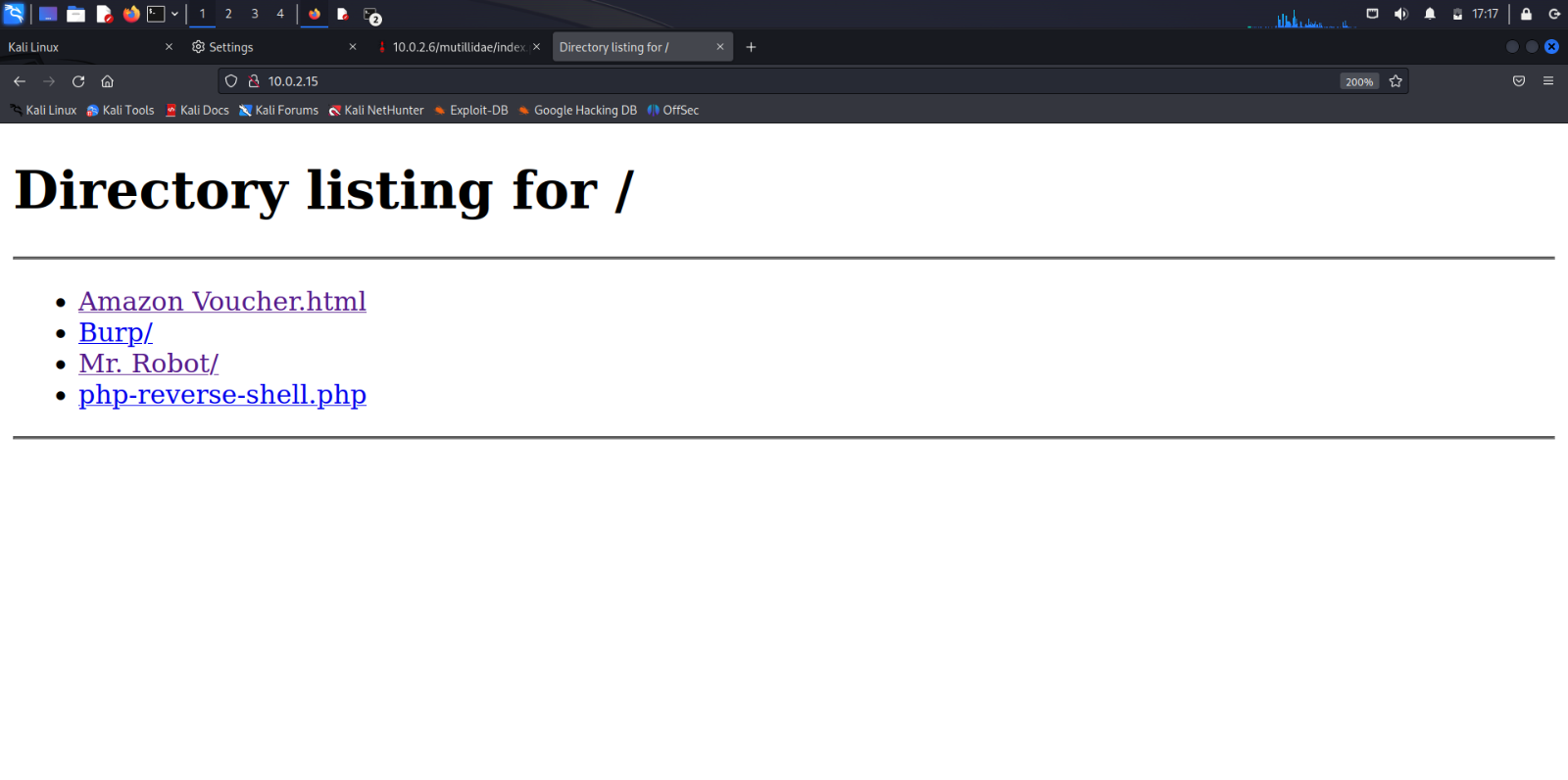
Create a local server using python to house the php file.

Command: sudo python3 -m http.server 80

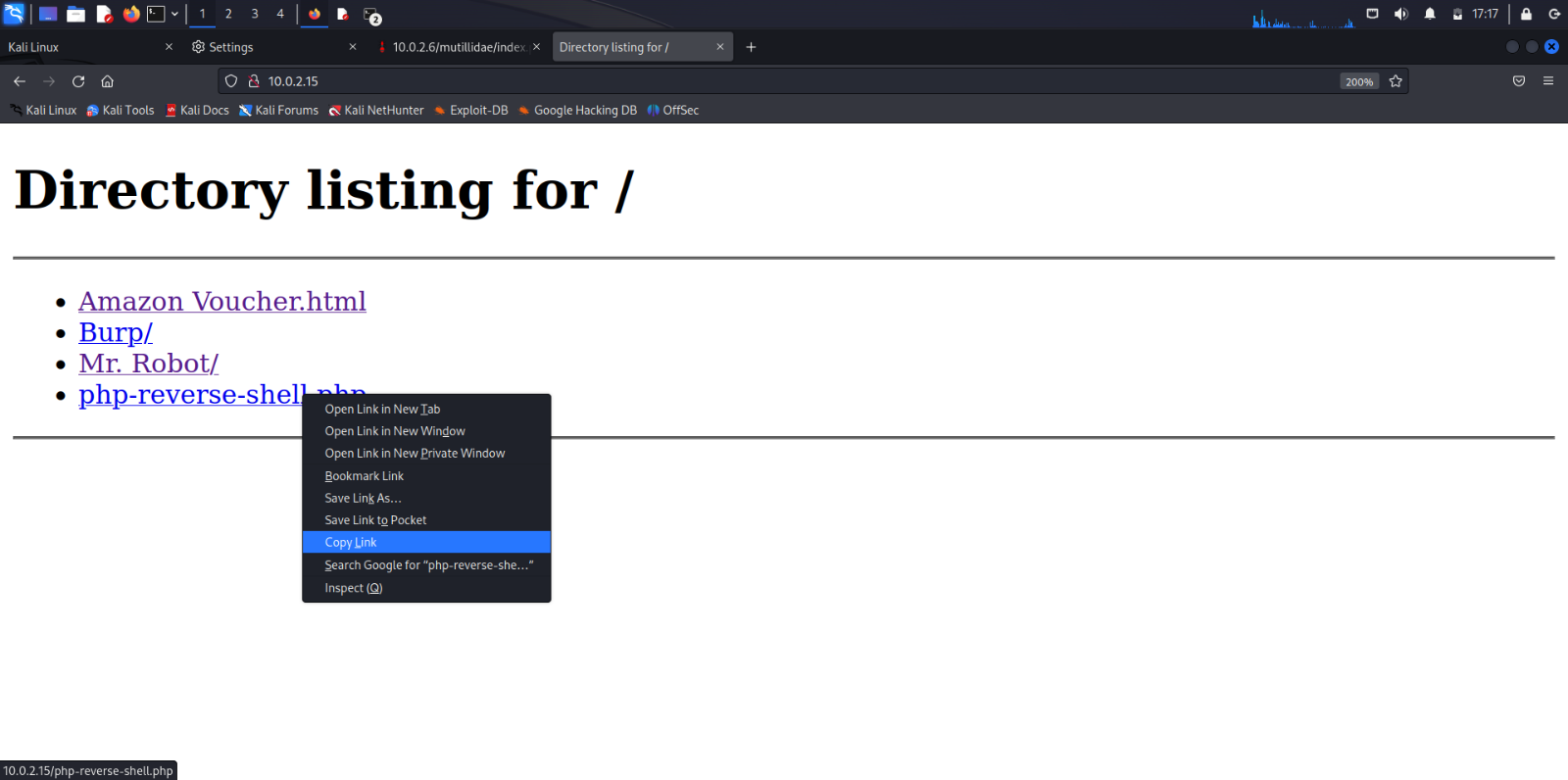


Open the server on the browser by giving the IP address of the server in the URL.

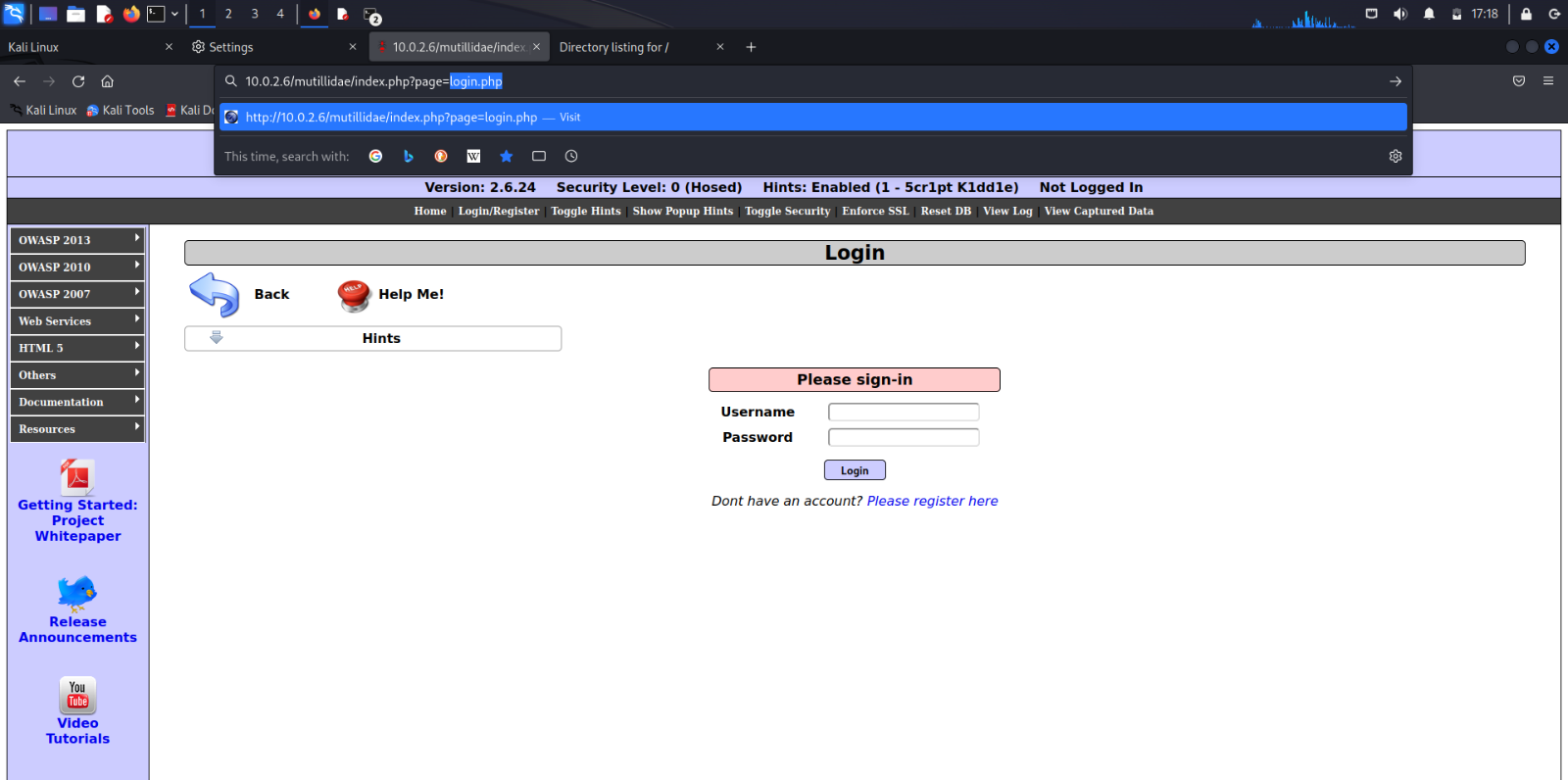
We can see the php-reverse-shell.php file.



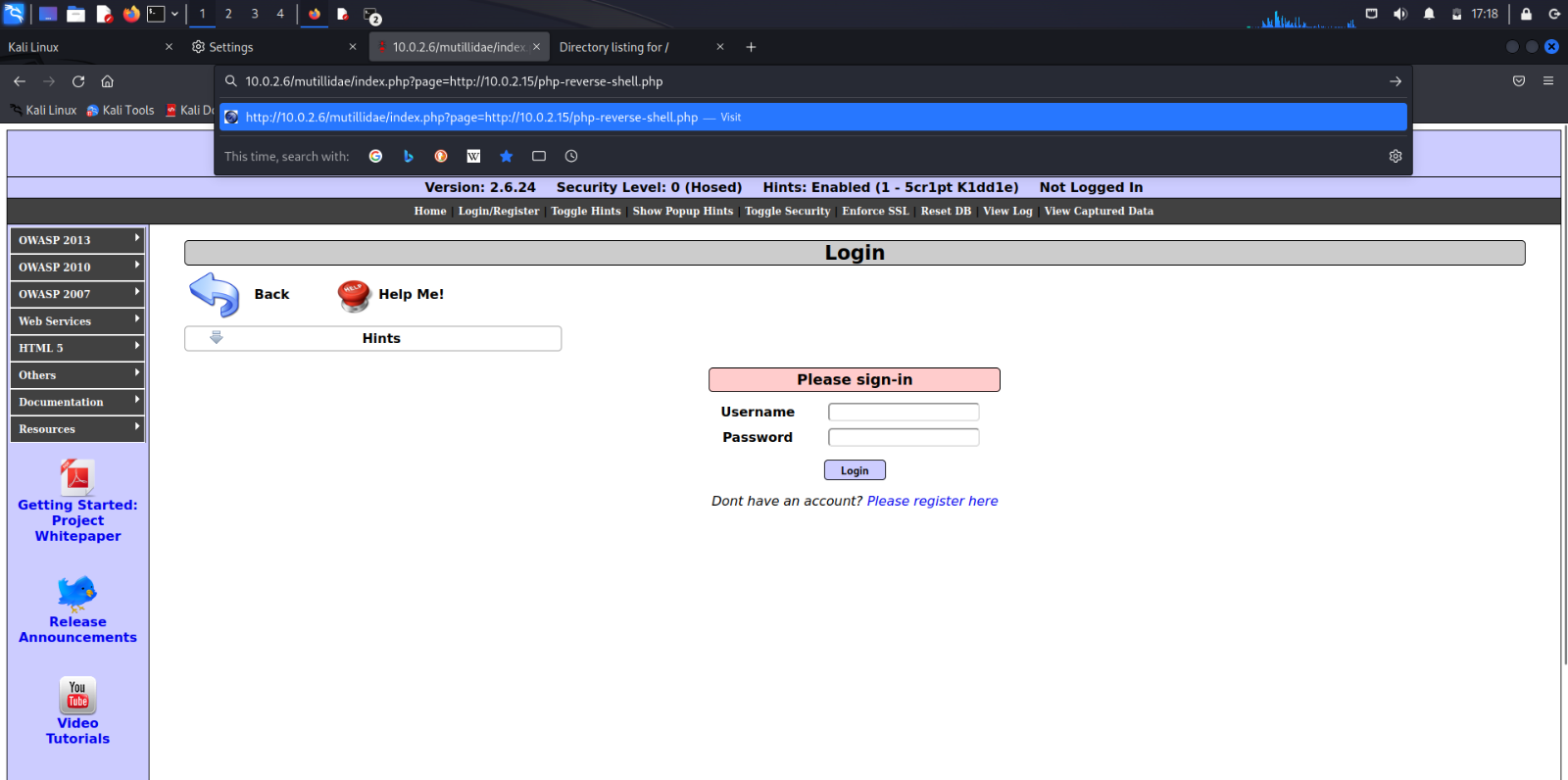
Copy the link address of the php-reverse-shell.php file.



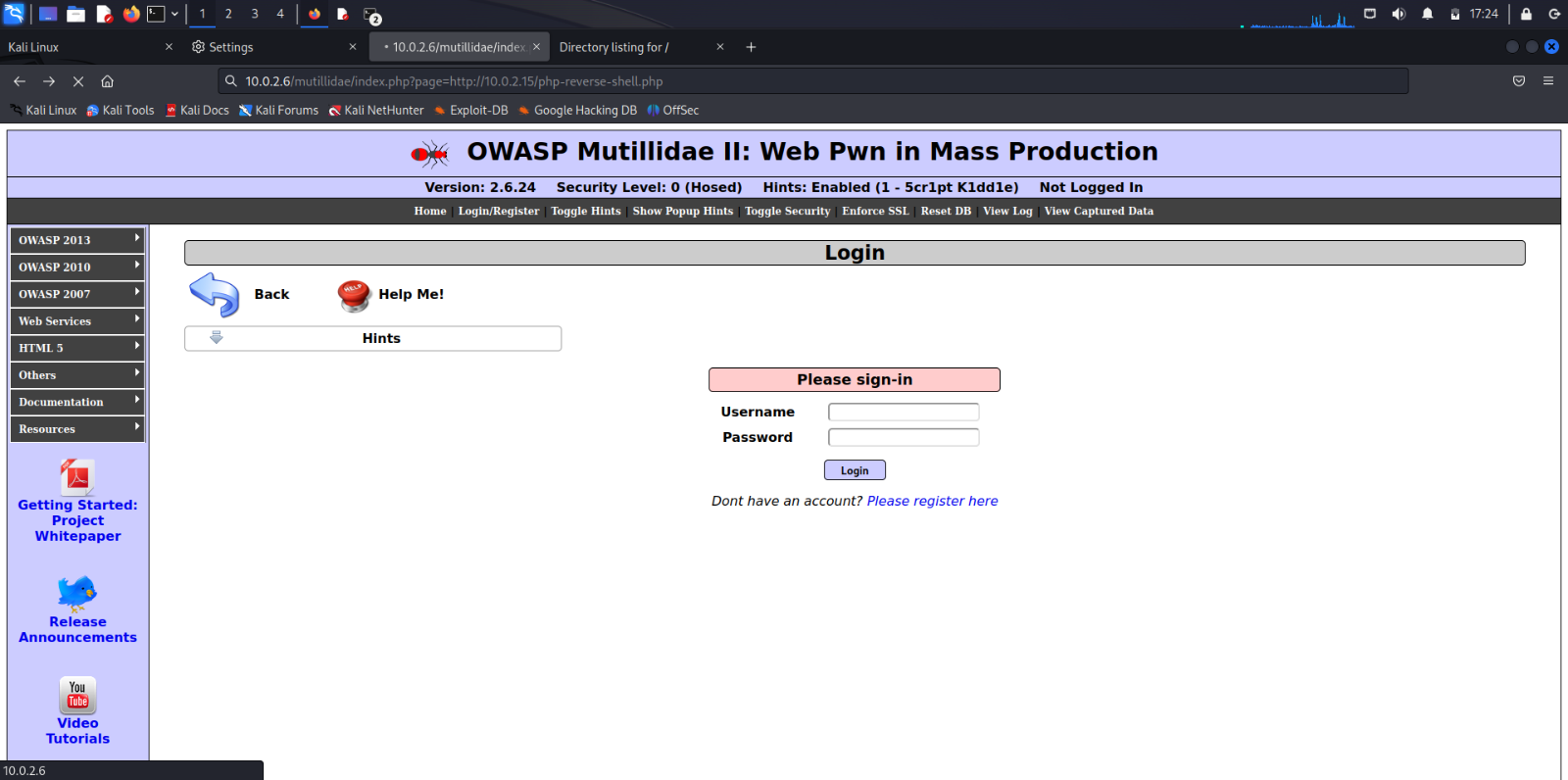
Select the login.php file which was being executing.



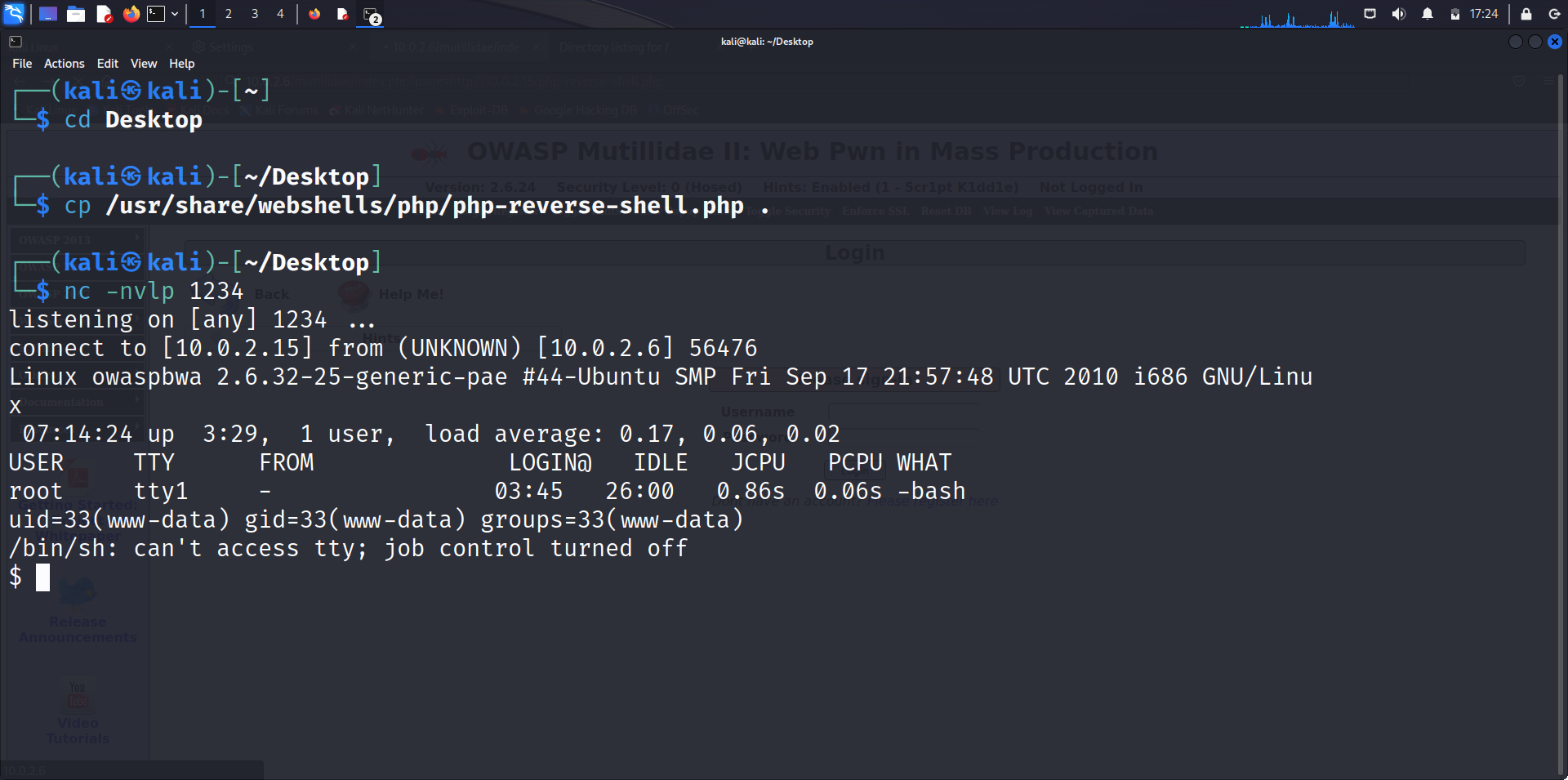
Replace the login.php file with the copied link address.



The php-reverse-shell.php has been remotely included in the login webpage of the Mutillidae II web application.



The malicious file php-reverse-shell.php has been executed and listener on port number 1234 has received a network connection. The attacker has gained unauthorized shell access to the Mutillidae II Web Application.



**Analysis**

The Remote File Inclusion (RFI) test report indicates that several vulnerabilities were identified within the web application. The RFI vulnerabilities allowed for the inclusion and execution of remote files on the target server, potentially leading to unauthorized access and data compromise. These findings highlight critical security weaknesses that could be exploited by malicious actors to gain control over the system. To mitigate these risks, immediate remediation steps should be taken, including implementing input validation and updating the application's security configurations to prevent RFI attacks.

**Conclusion**

In conclusion, the Remote File Inclusion (RFI) test report underscores the significant security risks associated with the tested web application. Multiple RFI vulnerabilities were discovered, exposing the application to the potential for remote code execution and unauthorized data access. These findings emphasize the urgent need for comprehensive security measures to be implemented. To ensure the application's safety and protect against potential attacks, immediate remediation actions, such as thorough input validation and secure server configurations, should be taken. Regular security assessments and ongoing vigilance are paramount to maintaining the web application's integrity and safeguarding sensitive information.