

**Report CTF**

**CTF Double Trouble Penetration Testing Report**

Date: 09/09/2024

Version: 1.0

**1 Table Of Contents**

1. Document Revision History………………………………………………………….........………………………...........…2
2. Executive Summary....................................................................................................................................3
3. Scope..........................................................................................................................................................4
4. Risk Categories...........................................................................................................................................5
5. Pentest Methodology..................................................................................................................................6
6. Visual Summary..........................................................................................................................................7
7. Findings Summary......................................................................................................................................8

**1 Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Version** | **Contact Details** |
| Abhishek Joshi | 09/09/2024 | 1.0 | Email: [abhishekjoshi266@gmail.com](mailto:abhishekjoshi266@gmail.com)  Mob No: +91 -- |

# u e

**2 Executive Summary**

An analysis of a black box penetration test conducted on the Double Trouble “CTF Machine” is presented in this document. Based on a thorough security assessment performed by Internal Security Team in September of 2024.

This assessment was conducted On-Premises by the Security team. An assessment was conducted on the 09th of September to 09th of September 2024. As a comprehensive strategy for this assessment, Security Team concreted the black box penetration testing methodology and technique. To facilitate this, Company provided a walkthrough of the application and provided access to the test environment with valid different privilege accounts.

Testing was carried out by identifying vulnerabilities with the intent of accessing critical information. The objective of performing this activity was to assess the security risks associated with the developed applications and identify vulnerabilities that cybercriminals could leverage to compromise the application. The report summarizes the security findings related to the Company applications and network.

**This assessment aimed to:**

Analyze the application for technical vulnerabilities that an attacker may exploit to compromise the CTF Machine.

Provide recommendations for risk mitigation that may arise on successful exploitation of these vulnerabilities.

**3 Scope**

### Scope

The section defines the scope and boundaries of the project.

### Constraints and Limitations

The assessments, and the result(s) / finding(s) made are highly subjective to target system(s) and service(s) visibility and availability at that given point of time.

### Target Scope

Identify weaknesses that might be exploited by adversaries who have authorized or unauthorized access to Company Technical Skill Test and underlying infrastructure:

Test Perform On Double Trouble CTF Environment Without Credential as Black Box Testing.

Following Machine was in the scope of the penetration test.

### Machine and Environment Details

|  |  |  |
| --- | --- | --- |
| Sr . No CTF Name Url: | | |
| 1 | Double Trouble Thor | Machine Url: <https://www.vulnhub.com/entry/doubletrouble-1,743/>  Machine IP ( 192.168.56.134 ) |

### Contact Details

|  |  |
| --- | --- |
| Names Contact Details | |
| Abhishek Joshi (Penetration Tester) | Skype ID: live:.cid.afc2ad6f8029ae21  Mail: [a](mailto:Abhishek.joshi@fendahl.com)bhishekjoshi266@gmail.com |

### 

**4 Risk Categories**

### Risk Categories & Rationales

Pentest use a simple risk categorisation of each vulnerability to focus the triage process at the risks which truly matter. The Common Vulnerability Scoring System (CVSS) is an industry standard formula. It generates a risk score between 0.0 and 10.0.

The table below explains the risk categories and demonstrates rule-of-thumb equivalency with CVSS scores:

|  |  |  |
| --- | --- | --- |
| **Risk Category** | **CVSS Score** | **Rationales** |
| Critical | 8.1 – 10.0 | Poses a severe risk which is easy to exploit. Begin the process of remediating immediately after the issue has been presented. |
| High | 6.1 – 8.0 | Poses a significant risk and can be exploited. Address these as soon as possible after any critical risks have been remediated. |
| Medium | 4.1 – 6.0 | Poses an important risk but may be difficult to exploit. Pentest recommends remedial work within 3 months of discovery. |
| Low | 2.1 – 4.0 | Poses a minor risk or may be exceedingly difficult to exploit. Address these over the long-term during testing cycles |
| Informational | 0.0 – 2.0 | Loss of sensitive information, or a discussion point. These are not directly exploitable but may aid an attacker. Remediate these to create a true defence-in-depth security posture, |

CVSS is not applicable to all risks. For example, it is incapable of capturing the risk of a “flat network design”. Experience has told us that this is a “high” risk in most cases.

For this reason, the reader may find vulnerabilities which have no CVSS rating in our reports.

We endeavour to provide the reason for omitting the risk score when that is the case, and to provide CVSS by default in all applicable cases.

**5 Pentest Methodology**

### Methodology

The penetration testing methodology is typically based on the NIST security methodology. The focus shifts from traditional application security, where the primary threat is from multiple sources over the Internet. The key difference is in the client-side security, file system, hardware, and network security. Traditionally for Thick Client Applications, an end user is in control of the device. Security Team used the NIST & MITRE Attack Framework testing guide for conducting penetration test of the systems and applications. The testing was done to simulate as closely as possible the viewpoint of completely external attacker, the steps involved are

1] Setup

2] Discovery

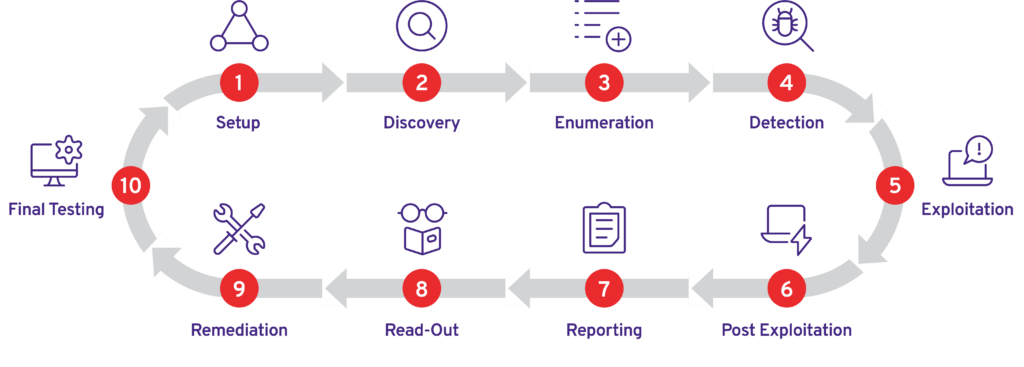
3] Enumeration

4] Detection

5] Exploitation

6] Post-Exploitation

7] Reporting



### Visual Summary

Graphical representation of Identified Vulnerabilities to Severity Risk rating

|  |  |  |
| --- | --- | --- |
| Sr. No. Severity Level Frequency | | |
| 1 | Critical | 2 |
| 2 | High | 1 |
| 3 | Medium | 5 |
| 4 | Low | 1 |

Table: Representing Severity Level

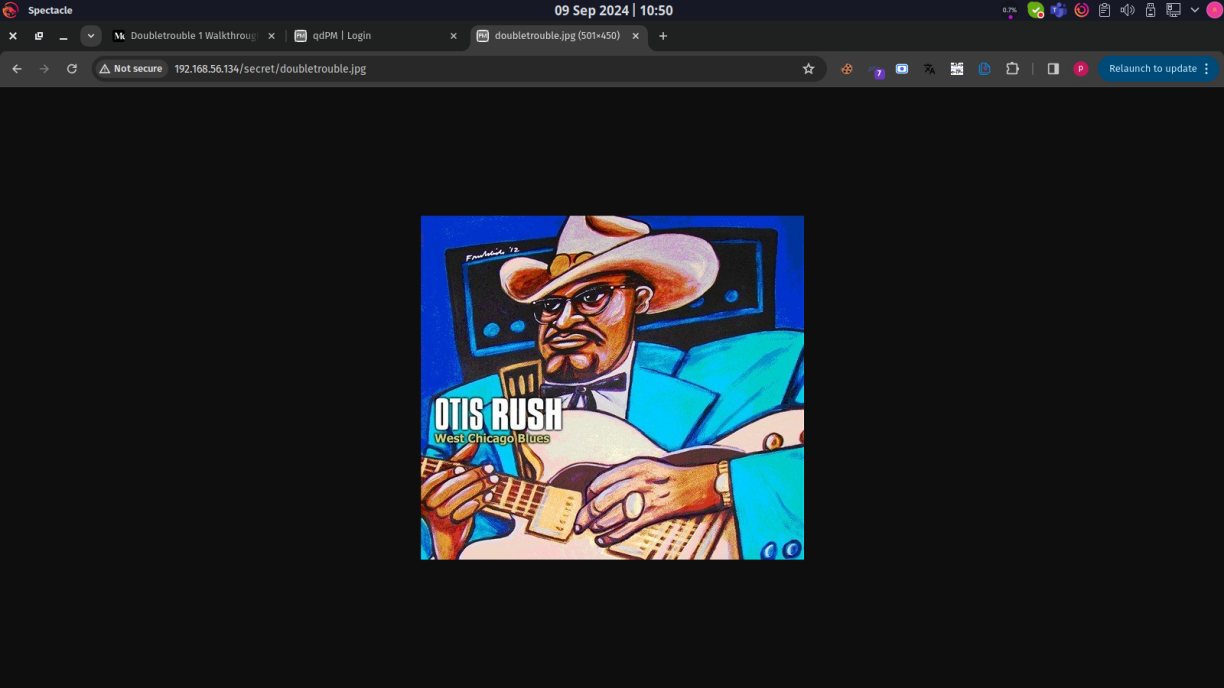
### Findings Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Observed Vulnerability** | **Risk Rating** | **Status** | **Comments** |
| 1. | Weak Steganography Password (Insufficient Secret Key) | Critical | Not Fixed | -- |
| 2. | Insecure Data Hiding(Steganography) leading to Credential Exposure | Critical | Not Fixed | -- |
| 2. | Insecure File Upload Vulnerability | Critical | Not Fixed | -- |
| 3. | Browsable Sensitive Directories | High | Not Fixed | -- |
|  | Brute Force Attack | Medium | Not Fixed | -- |
|  | Missing 2FA | Medium | Not Fixed | -- |
|  | SSH Brute Force | Medium | Not Fixed | -- |
|  | Broken Authentication On Change Pass | Medium | Not Fixed | -- |
|  | Broken Authentication on Change Email | Medium | Not Fixed | -- |
|  | Missing Captcha | Low | Not Fixed | -- |

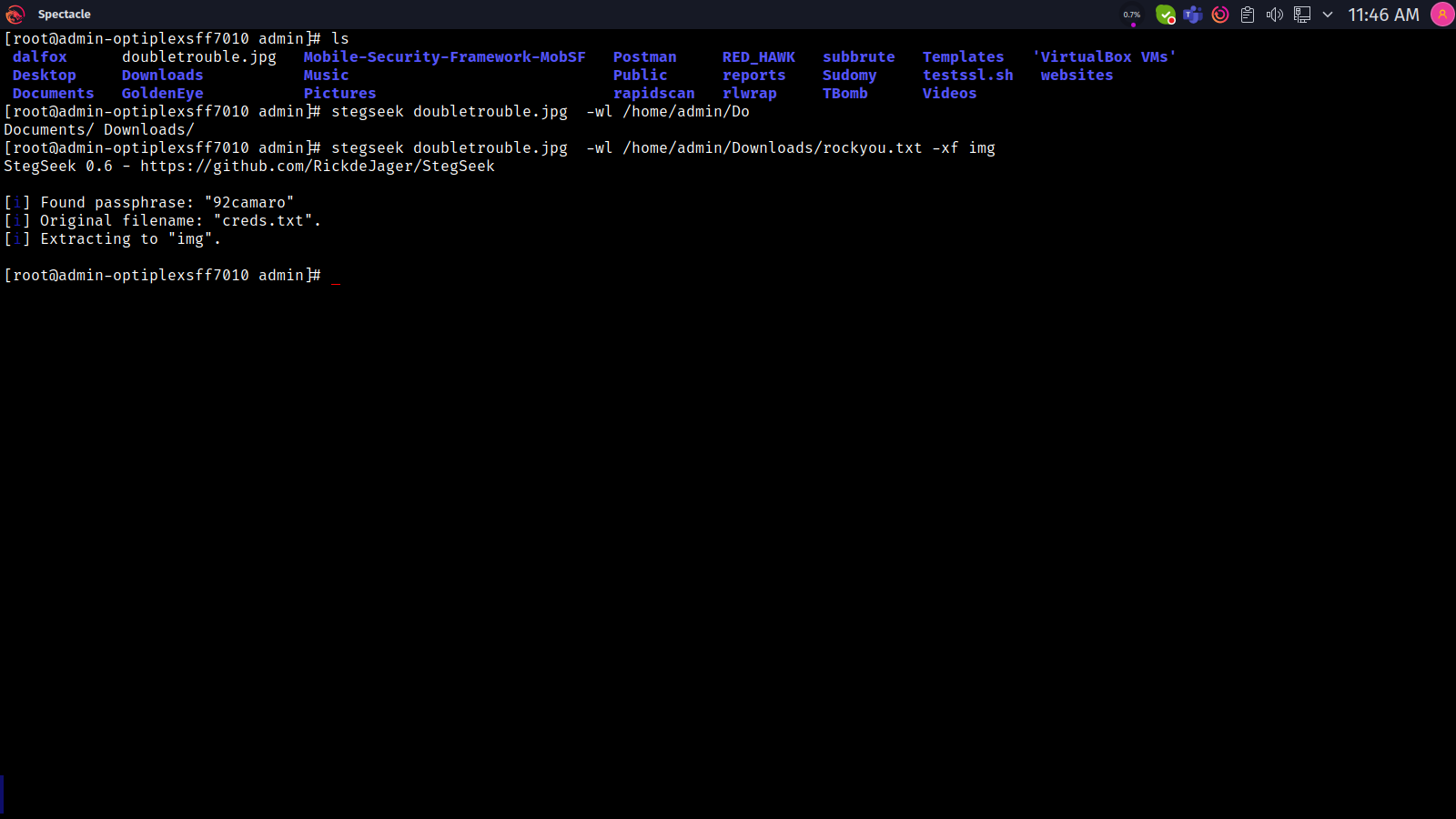
#### Weak Steganography Password (Insufficient Secret Key)

|  |  |
| --- | --- |
| **Vulnerability** | Weak Steganography Password (Insufficient Secret Key) |
| **Description** | The attacker was able to extract hidden data from an image using stegseek and brute force attacks. This suggests that the password or secret key used to hide the data was weak or insufficient, allowing the attacker to crack it and access the hidden information. |
| **Risk/Impact** | The impact of this vulnerability is that sensitive data, which was intended to be hidden, has been compromised. This can lead to unauthorized access to confidential information, intellectual property theft, or other malicious activities. The fact that the attacker was able to extract the data using brute force attacks also indicates that the password or secret key was not sufficiently complex or secure. |
| **CVSS Score** | 9.0 Critical |
| **Path:** | 192.168.56.134 |
| **Remidiation / Solution** | Remidiation:  - Implement strong encryprion with strong password while hiding secret data behind the image |
| **Refrence Url:** | <https://cwe.mitre.org/data/definitions/1391.html> |

Below Screenshots shows that attacker found one secret image on sensitive directory.



Below Screenshots shows that attacker found that image have some secret data and start pass cracking to extract secret info in image.



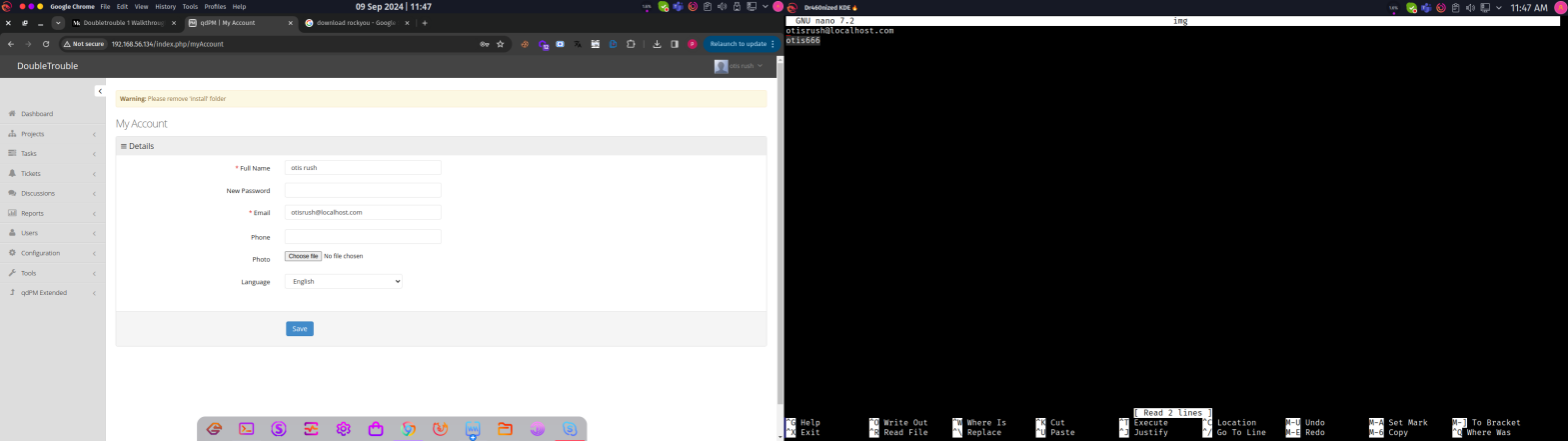


=

#### Insecure Data Hiding (Steganography) leading to Credential Exposure

|  |  |
| --- | --- |
| **Vulnerability** | Insecure Data Hiding (Steganography) leading to Credential Exposure |
| **Description** | An attacker was able to crack the steganography used to hide sensitive data, including a username and password, within an image. This allowed the attacker to extract the credentials and use them to take over the associated account. |
| **Risk/Impact** | An attacker able to crack password of stegnograph image and gain sensitive credentials. |
| **CVSS Score** | 9.5 High |
| **Path:** | 192.168.56.134 |
| **Remidiation / Solution** | Remidiation:  - Use secure steganography techniques: Implement robust steganography methods that use strong encryption algorithms and secure key management practices.  - Choose strong passwords: Ensure that passwords are complex, unique, and regularly updated.  - Implement multi-factor authentication: Add an extra layer of security to prevent unauthorized access, even if credentials are compromised. |
| **Refrence Url:** | <https://cwe.mitre.org/data/definitions/200.html> |

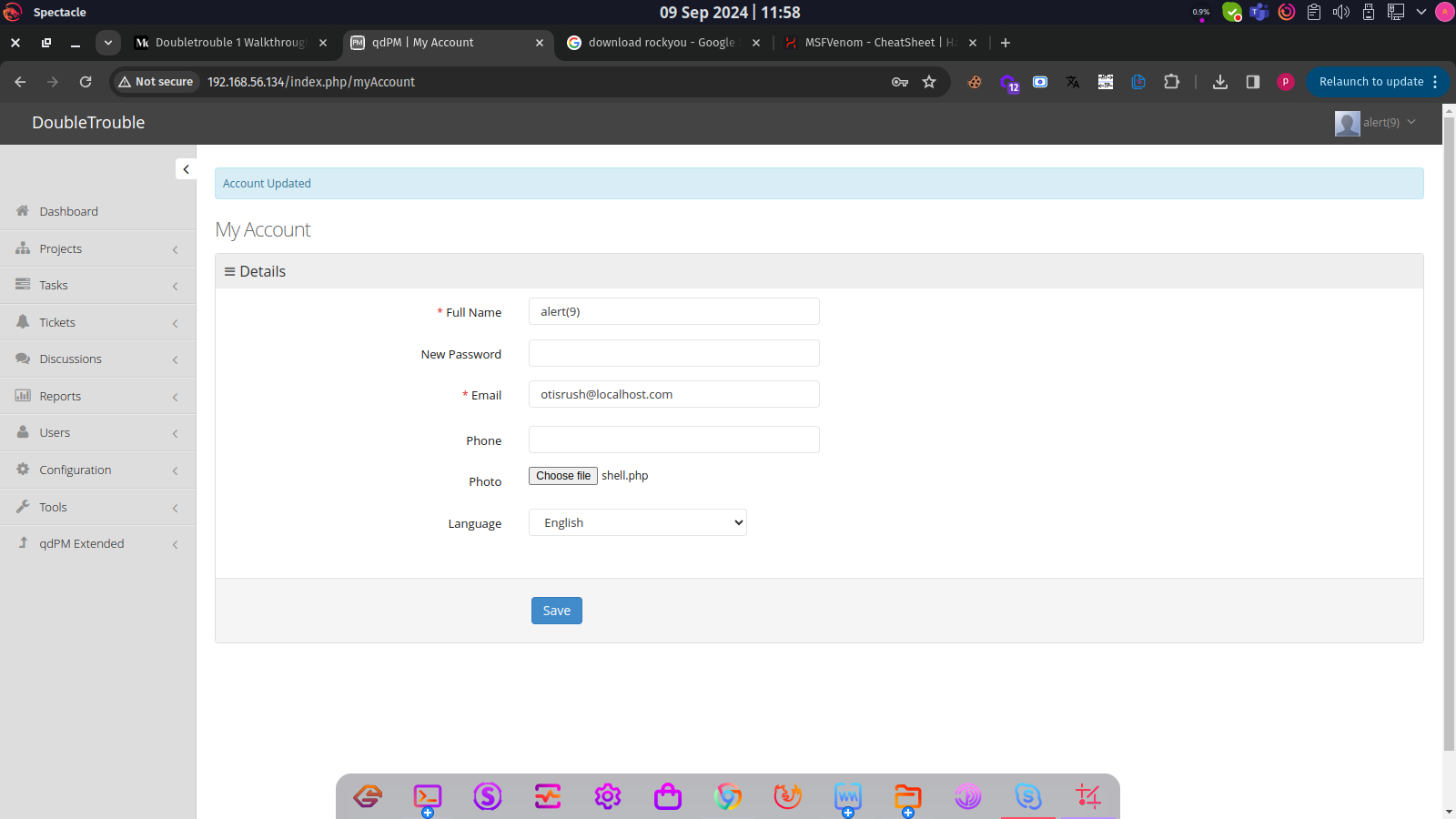
Below Screenshots shows that attacker found user credential while chrack stegnograph image and gain access.

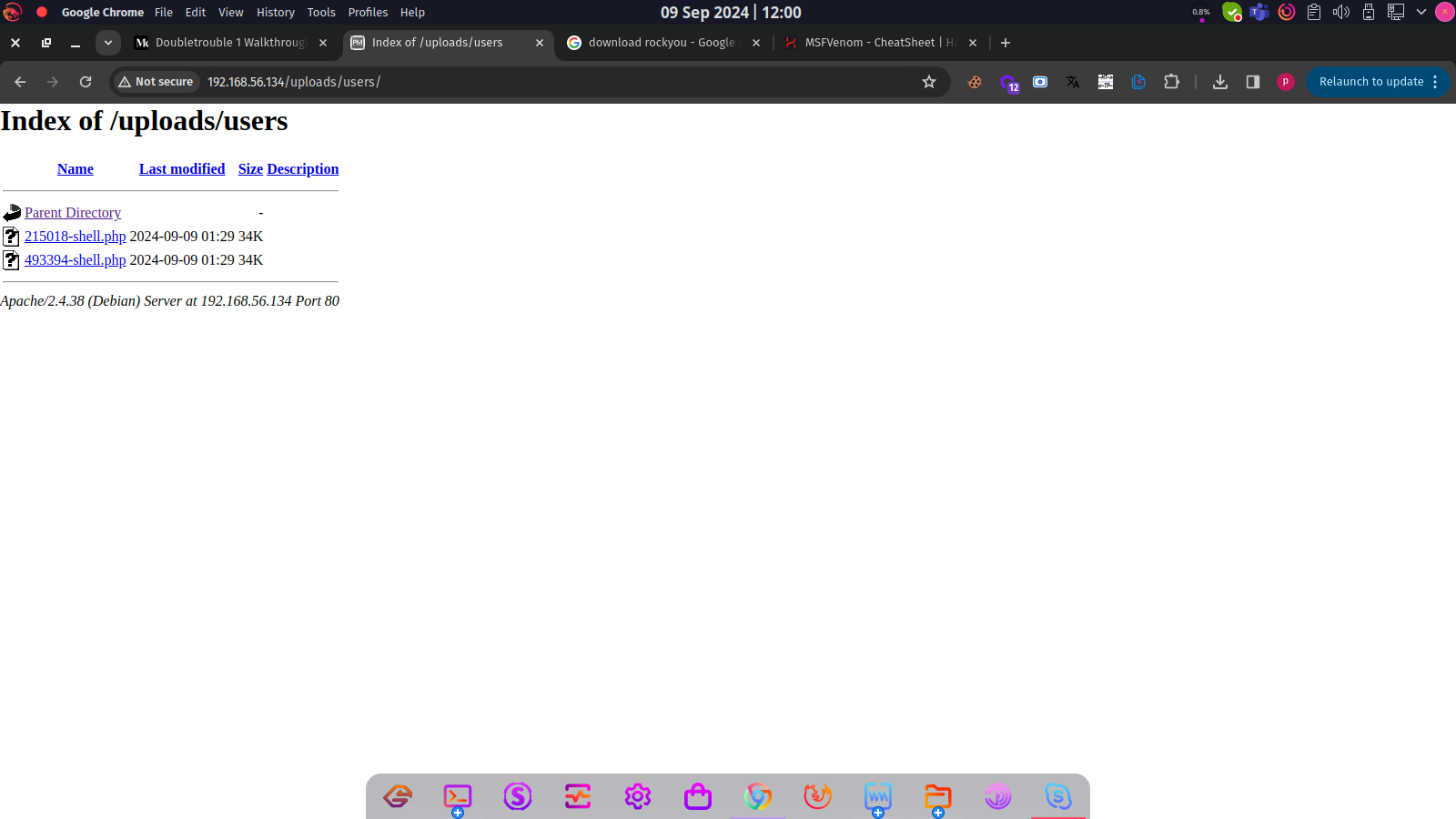


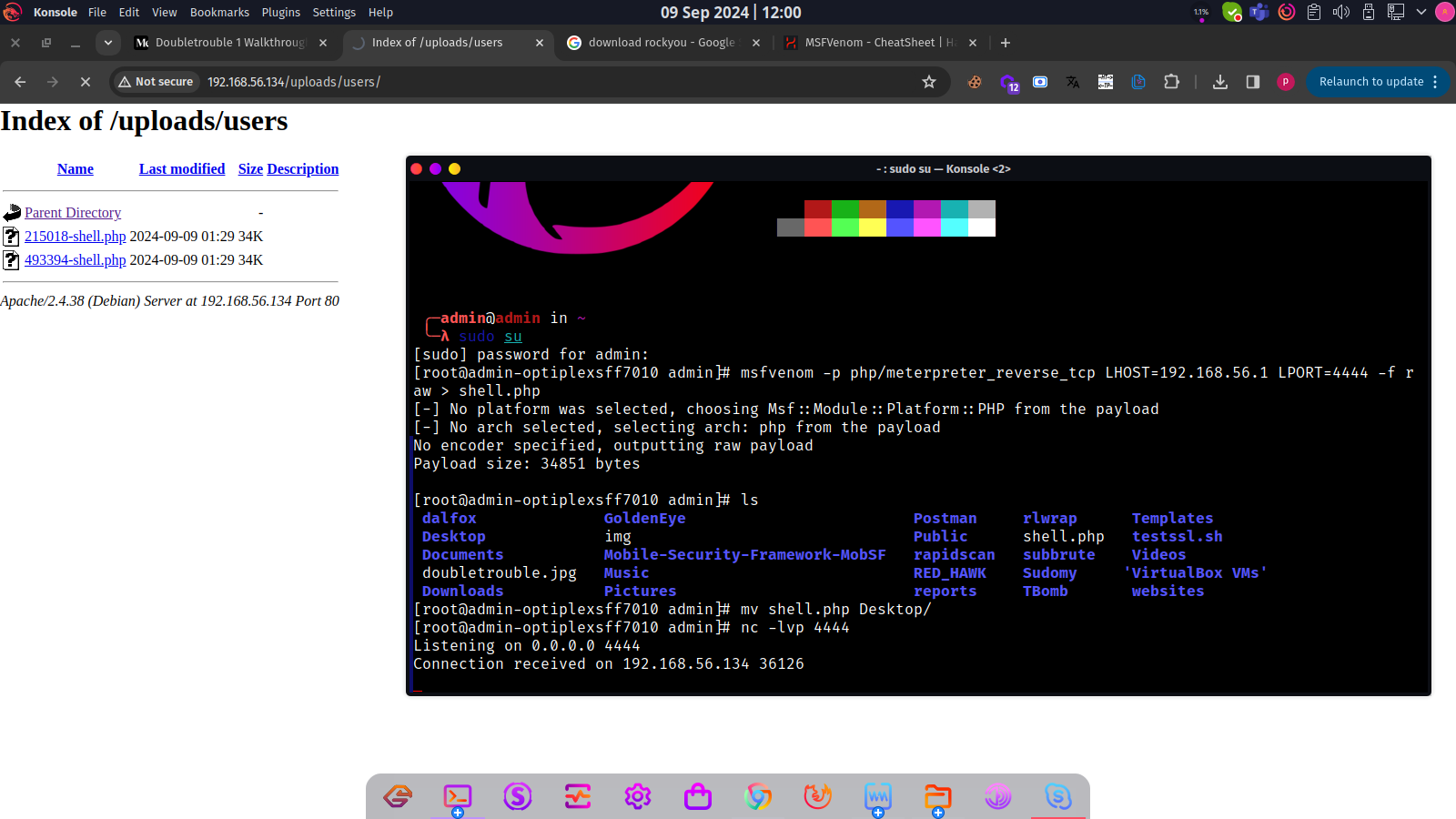
#### Insecure File Upload Vulnerability

|  |  |
| --- | --- |
| **Vulnerability** | Insecure File Upload Vulnerability |
| **Description** | The attacker exploited a vulnerability in the file upload functionality of the application, allowing them to upload a malicious PHP reverse shell script. This vulnerability typically arises from insufficient validation and sanitization of uploaded files, enabling the execution of arbitrary code on the server. |
| **Risk/Impact** | An attacker can execute arbitrary code on the server, potentially leading to complete control over the affected system.. |
| **CVSS Score** | 9.7 High |
| **Path:** | 192.168.56.134 |
| **Remidiation / Solution** | Remidiation:  - File Type Validation: Implement strict validation to ensure only allowed file types (e.g., images, documents) can be uploaded. Avoid allowing executable files or scripts.  - File Size Limits: Set limits on the size of uploaded files to prevent denial-of-service attacks.  - Rename Uploaded Files: Rename files upon upload to prevent execution based on the original filename and extension.  - Use Secure Directories: Store uploaded files outside the web root or in a directory with restricted access to prevent direct execution.  - Implement Content Security Policies: Use Content Security Policies (CSP) to help mitigate the risk of executing malicious scripts. |
| **Refrence Url:** | <https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload> |

Below Screenshots shows that attacker can able to upload php reverse shell and make a connection



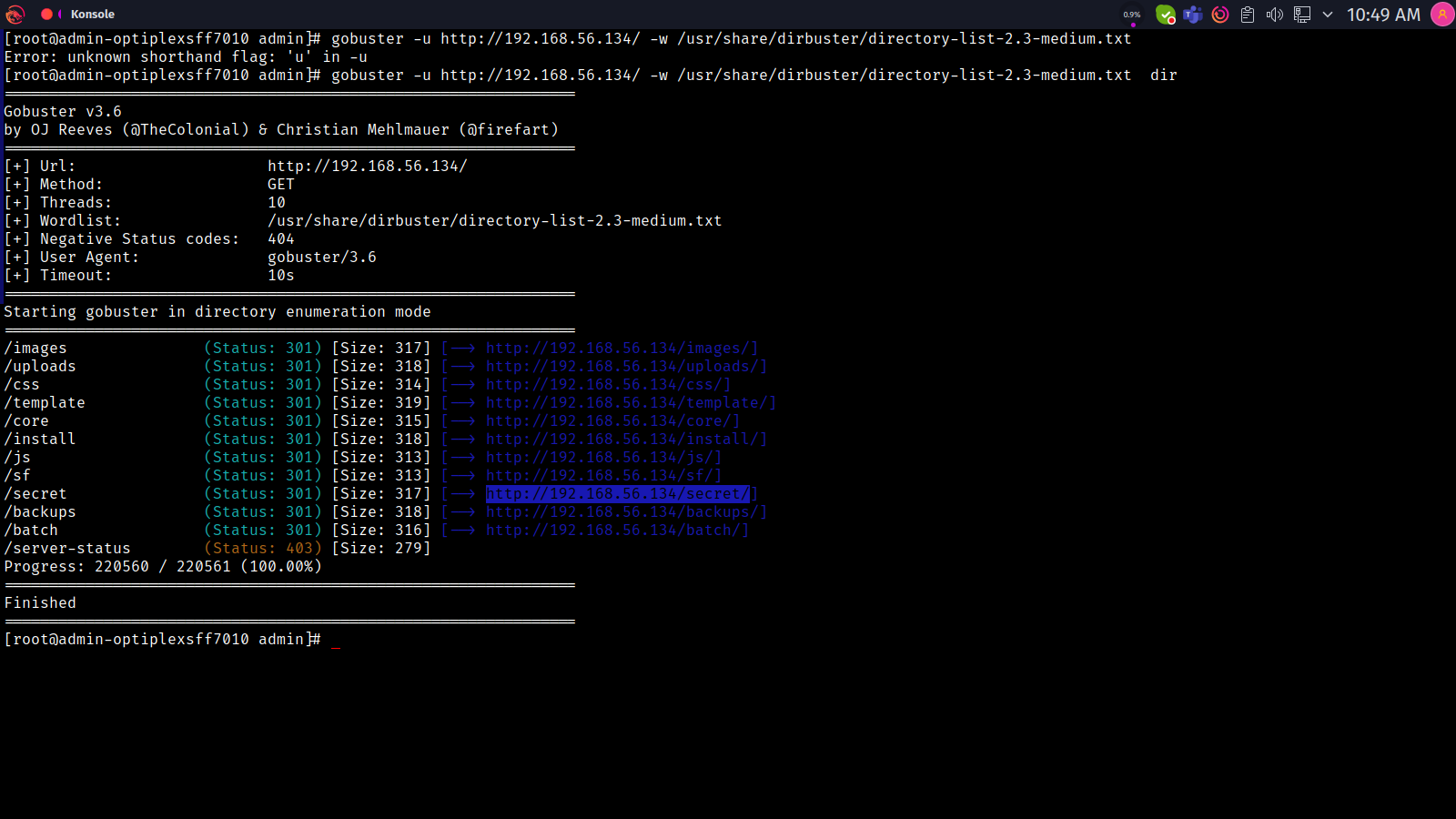
4 (0)161 233 0100 

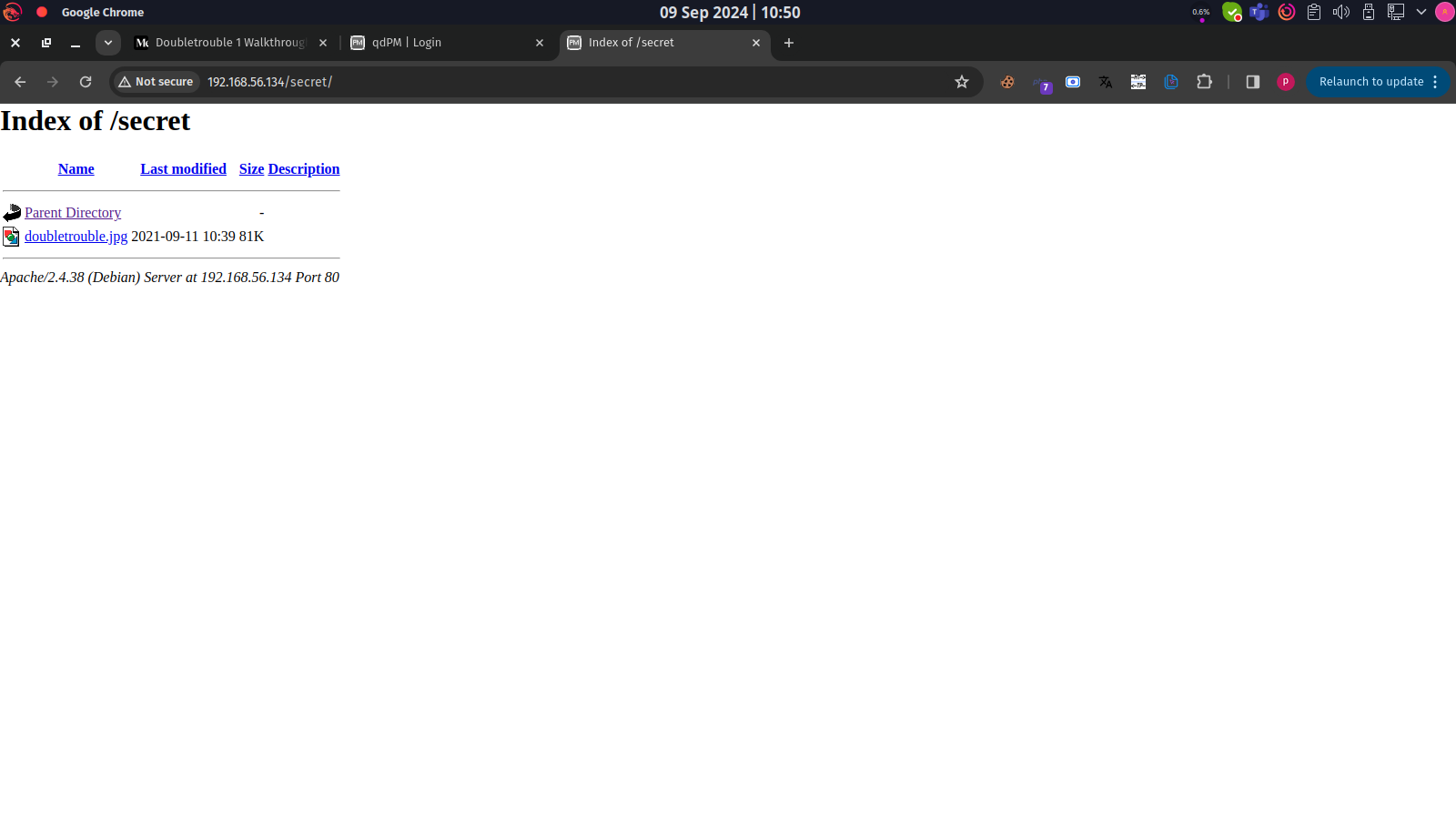


#### Browsable Sensitive Directories

|  |  |
| --- | --- |
| **Vulnerability** | Browsable Sensitive Directories |
| **Description** | The attacker was able to browse sensitive directories, which led to the discovery of a secret image containing steganographed data. This vulnerability occurs when directory indexing is not properly configured, allowing unauthorized access to sensitive files and directories. |
| **Risk/Impact** | An attacker can retrive Sensitive data, such as the steganographed image, can be accessed and potentially cracked, revealing confidential information. |
| **CVSS Score** | 8.0 High |
| **Path:** | 192.168.56.134 |
| **Remidiation / Solution** | Remidiation:  - Use secure steganography techniques: Implement robust steganography methods that use strong encryption algorithms and secure key management practices.  - Choose strong passwords: Ensure that passwords are complex, unique, and regularly updated.  - Implement multi-factor authentication: Add an extra layer of security to prevent unauthorized access, even if credentials are compromised. |
| **Refrence Url:** | <https://portswigger.net/kb/issues/00600100_directory-listing> |

Below Screenshots shows that attacker found sensitive directories.

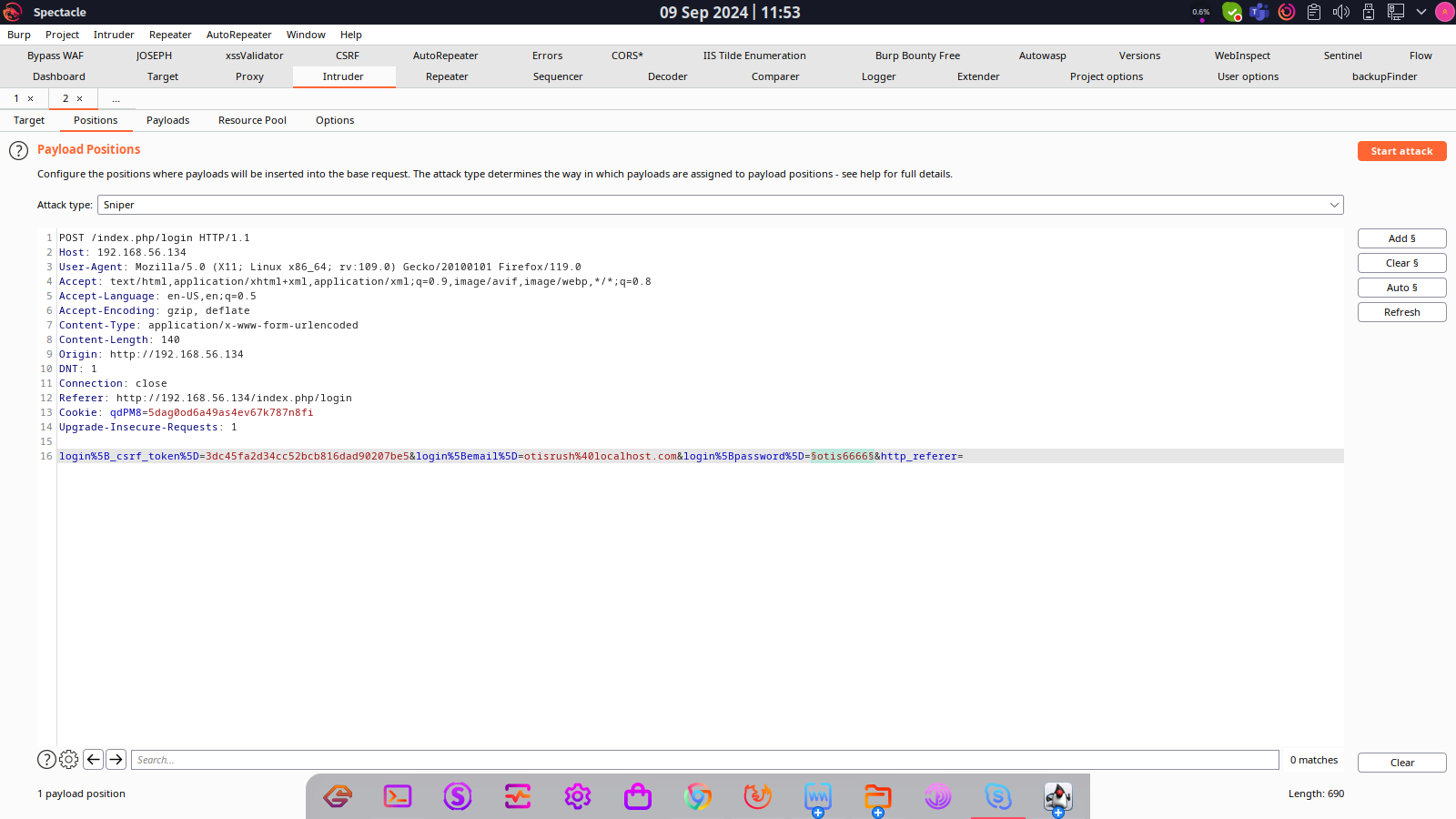


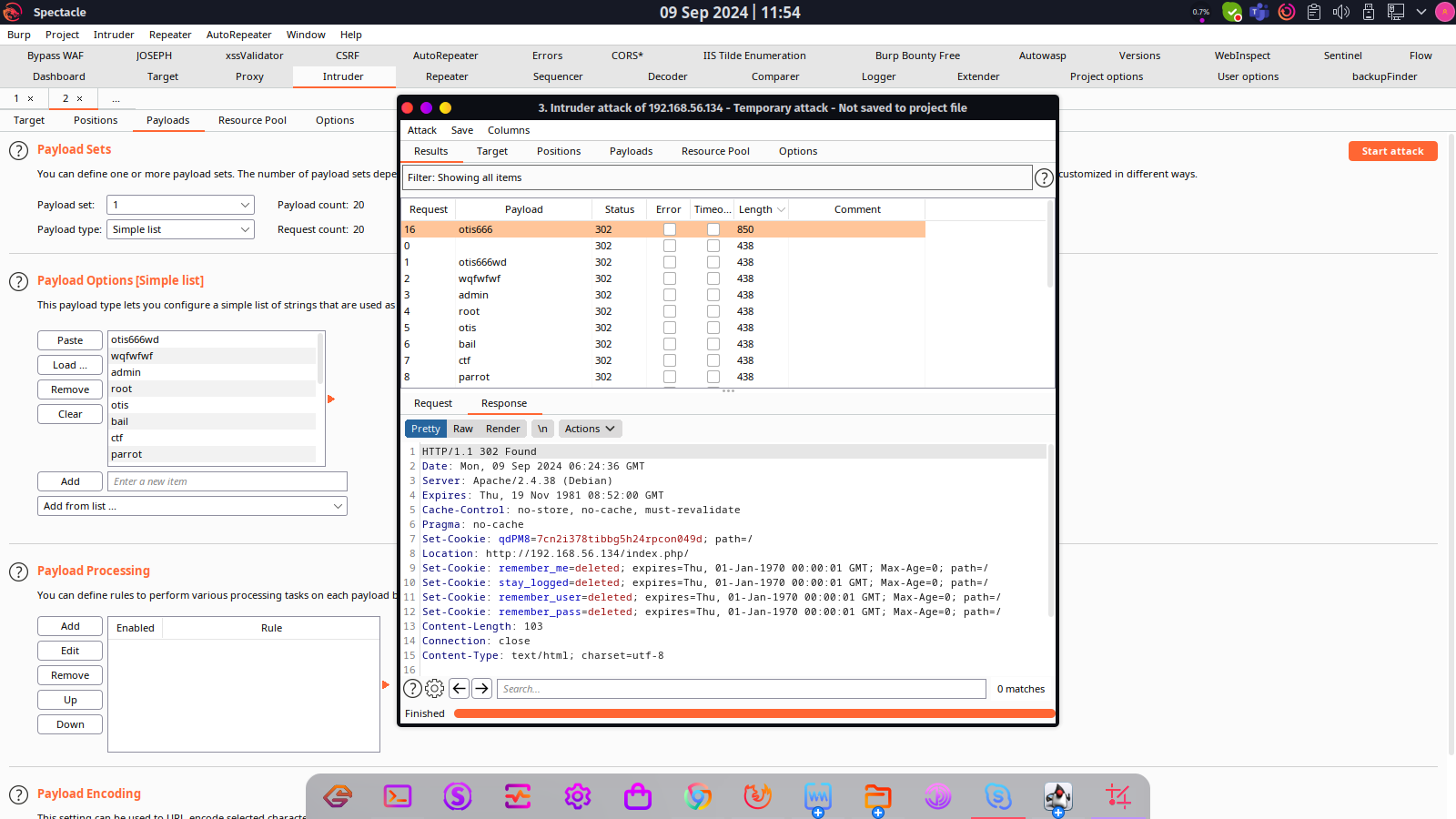


#### Brute Force Attack

|  |  |
| --- | --- |
| **Vulnerability** | Brute Force Attack |
| **Description** | A brute force attack is a type of attack where an attacker uses automated tools to try a large number of combinations of usernames and passwords to gain unauthorized access to a system, network, or application.. |
| **Risk/Impact** | An attacker attempts to guess the correct combination by trying all possible combinations until they find the correct one. |
| **CVSS Score** | 5.0 Medium |
| **Path:** | 192.168.56.134 |
| **Remidiation / Solution** | Remidiation:  - Implement Account Lockout Mechanism  - Implement Rate Limiting  - Implement Captcha |
| **Refrence Url:** | <https://owasp.org/www-community/attacks/Brute_force_attack> |

Below Screenshots shows that attacker can able to brute force user.



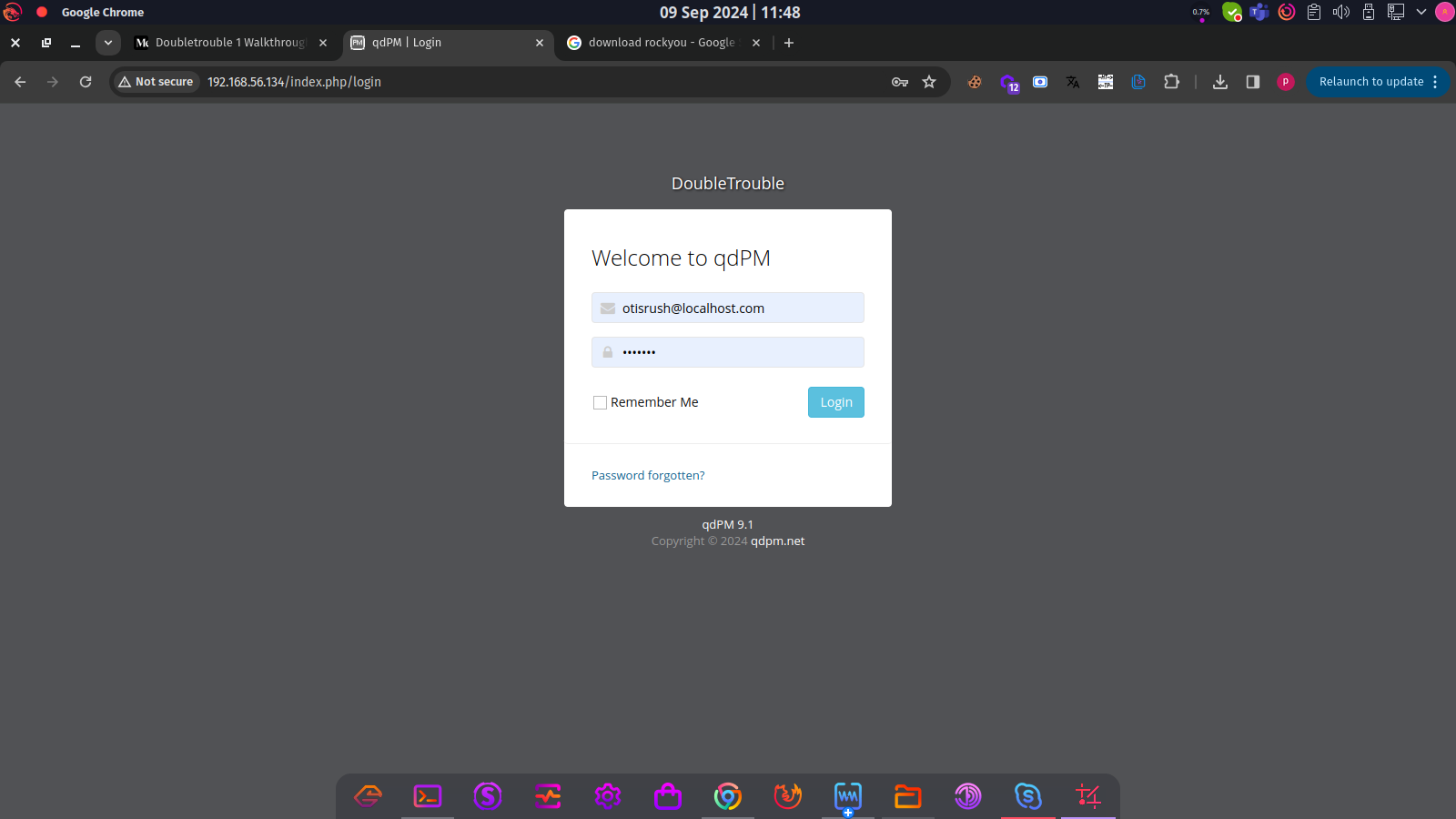


#### Missing 2FA

|  |  |
| --- | --- |
| **Vulnerability** | Missing 2FA |
| **Description** | Missing two-factor authentication (2FA) refers to the absence of an additional layer of security beyond just a username and password. 2FA typically requires users to provide two different forms of verification before they can access an account or system. This can include something they know (password), something they have (a mobile device, hardware token), or something they are (biometric verification). Without 2FA, accounts are more vulnerable to unauthorized access, especially if passwords are weak, reused, or compromised. |
| **Risk/Impact** | An Attackers can gain access to user accounts by simply obtaining or guessing passwords, leading to potential data breaches. |
| **CVSS Score** | 5.0 Medium |
| **Path:** | 192.168.56.134 |
| **Remidiation / Solution** | Remidiation:  - Implement 2FA |
| **Refrence Url:** | <https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html> |

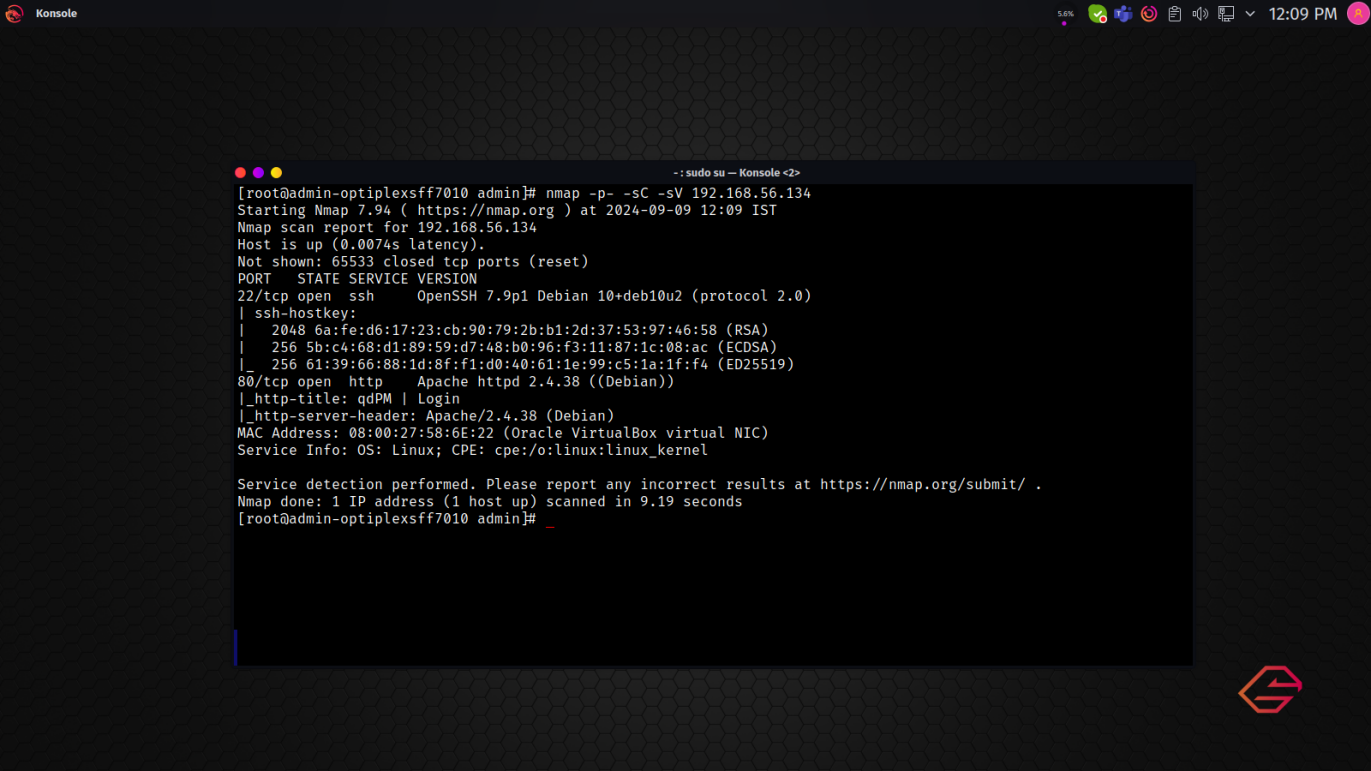
Below Screenshots shows that there is no 2fa is implemented

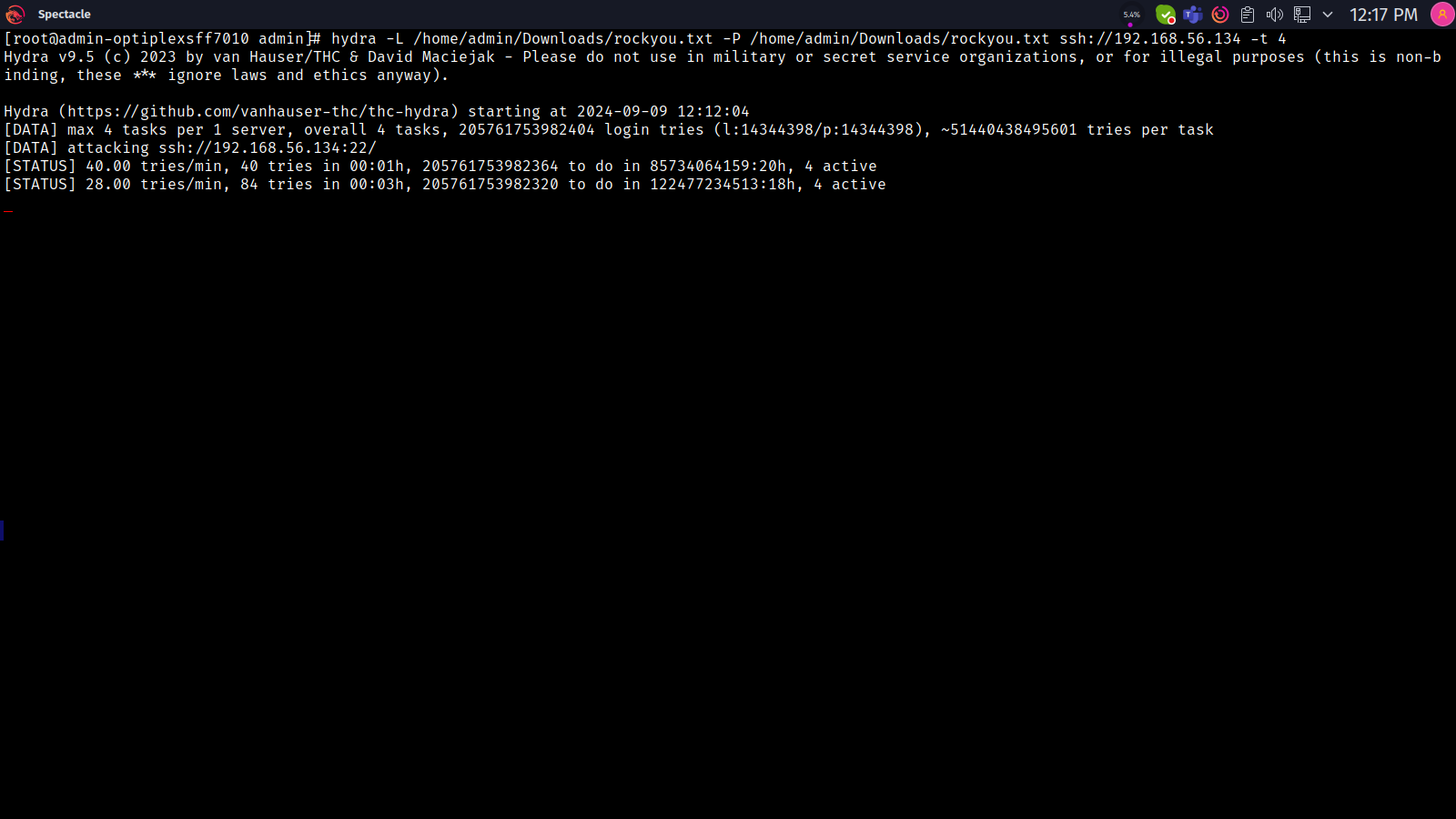
.



#### SSH Brute Force

|  |  |
| --- | --- |
| **Vulnerability** | SSH Brute Force |
| **Description** | SSH Brute Force Attack is a type of attack where an attacker attempts to guess the password of a valid SSH user account by trying a large number of possible passwords. This is typically done using automated tools that can try thousands of passwords per minute. The attacker may use a dictionary of common passwords, or may use a cracking tool to generate random passwords. |
| **Risk/Impact** | An Attackers can gain access to SSH accounts by brute force user and password. |
| **CVSS Score** | 5.0 Medium |
| **Path:** | 192.168.56.134 |
| **Remidiation / Solution** | Remidiation:  - Implement rate limiting  - block unauthoried IP  - filter SSH protocol by firewall |
| **Refrence Url:** | <https://nvd.nist.gov/vuln/detail/CVE-2020-1616> |

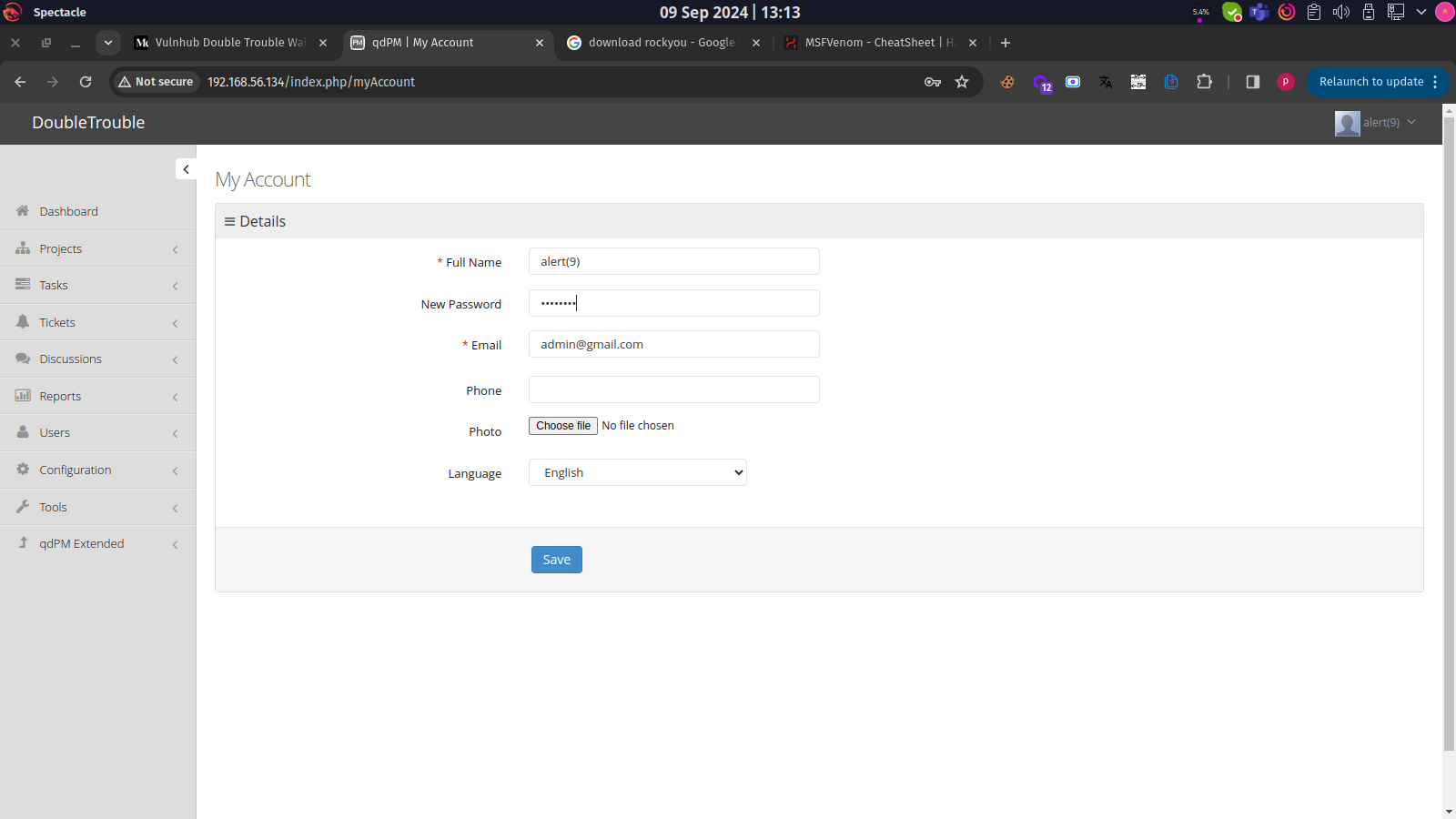
Below Screenshots shows that SSH is communicable and attacker able to brute force user account  
  




#### Broken Authentication On Change Pass

|  |  |
| --- | --- |
| **Vulnerability** | Broken Authentication On Change Pass |
| **Description** | Broken Authentication on Password Change occurs when a web application allows users to change their passwords without requiring proper authentication mechanisms, such as entering the old password or providing an OTP (One-Time Password) for verification. This vulnerability can arise from poor session management, inadequate access controls, or insufficient validation of user identity during sensitive operations. |
| **Risk/Impact** | An attacker could exploit this vulnerability to change a victim's password and gain unauthorized access to their account, leading to potential data theft or manipulation. |
| **CVSS Score** | 5.0 Medium |
| **Path:** | 192.168.56.134 |
| **Remidiation / Solution** | Remidiation:  - Require Old Password Verification  - Implement Multi-Factor Authentication (MFA) |
| **Refrence Url:** | <https://cwe.mitre.org/data/definitions/620.html> |

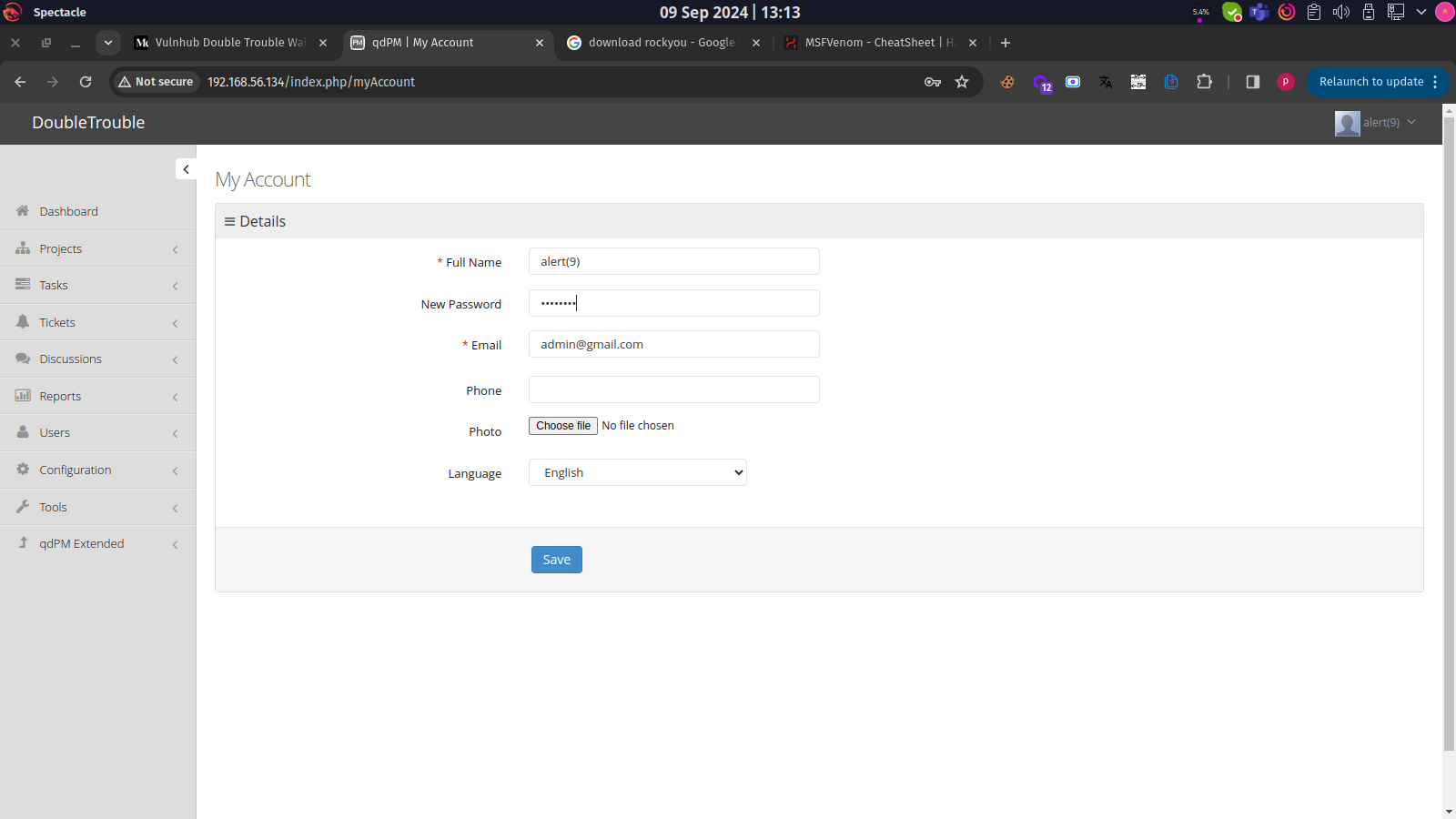
Below Screenshots shows that website is not asking for old pass or there is no MFA while change pass



#### Broken Authentication On Change Email

|  |  |
| --- | --- |
| **Vulnerability** | Broken Authentication On Change Email |
| **Description** | when a web application allows users to change their registered email addresses without requiring proper authentication mechanisms, such as entering the current password or providing an OTP (One-Time Password) for verification. This vulnerability can arise from inadequate access controls or insufficient validation of user identity during sensitive operations. |
| **Risk/Impact** | An attacker could exploit this vulnerability to change a victim's email and gain unauthorized access to their account, leading to potential data theft or manipulation. |
| **CVSS Score** | 5.0 Medium |
| **Path:** | 192.168.56.134 |
| **Remidiation / Solution** | Remidiation:  - Require Password Verification  - Implement Multi-Factor Authentication (MFA) |
| **Refrence Url:** | <https://cwe.mitre.org/data/definitions/620.html> |

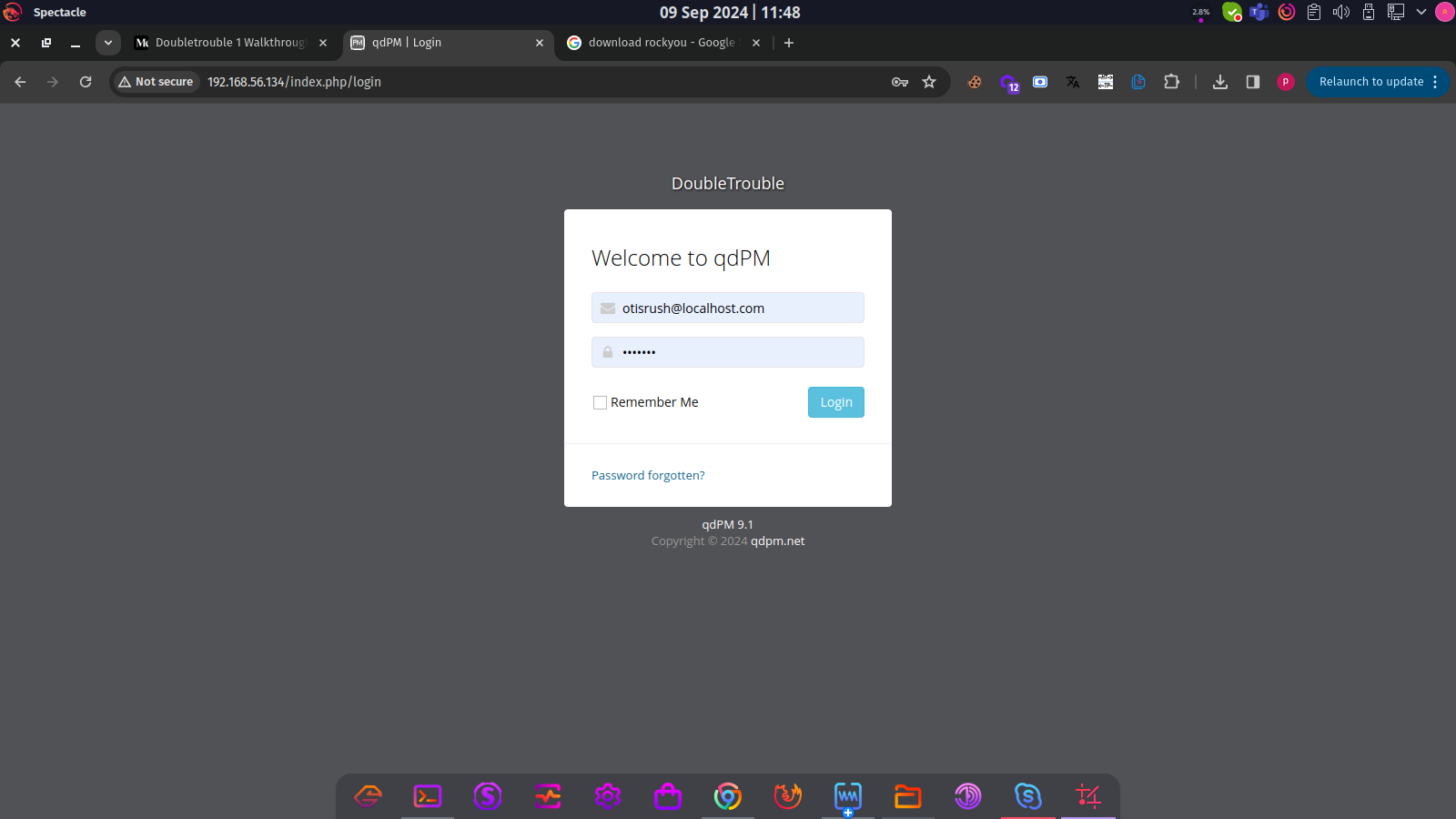
Below Screenshots shows that website is not asking for pass or there is no MFA while change email



#### Missing Captcha

|  |  |
| --- | --- |
| **Vulnerability** | Missing Captcha |
| **Description** | Missing CAPTCHA refers to the absence of a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) mechanism in web forms or applications that require user interaction, such as login forms, registration forms, password resets, and feedback forms. CAPTCHA is designed to distinguish between human users and automated bots, which can help prevent abuse and automated attacks |
| **Risk/Impact** | Without CAPTCHA, an attacker can easily automate requests to exploit vulnerabilities, such as brute-force attacks on login forms or account creation to flood the system with fake accounts.. |
| **CVSS Score** | 2.0 Low |
| **Path:** | 192.168.56.134 |
| **Remidiation / Solution** | Remidiation:  - Implement strong captcha mechanism |
| **Refrence Url:** | <https://nvd.nist.gov/vuln/detail/CVE-2022-2913> |

Below Screenshots shows that there is no captcha implemented



**END OF REPORT**