Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Confidentiality Statement

This document contains confidential and privileged information from Rekall Inc. (henceforth known as Rekall). The information contained in this document is confidential and may constitute inside or non-public information under international, federal, or state laws. Unauthorized forwarding, printing, copying, distribution, or use of such information is strictly prohibited and may be unlawful. If you are not the intended recipient, be aware that any disclosure, copying, or distribution of this document or its parts is prohibited.

Table of Contents

Read me: This project was done in a group alongside Tracy Skelton, and Jackson Long. We all contributed input on screenshots, remediation, and tactics.

Contact Information

Company Name	Shadow Entity
Contact Name	John Wallace
Contact Title	Jr. Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	3/12/2023	John Wallace	N/A

Introduction

In accordance with Rekall policies, our organization conducts external and internal penetration tests of its networks and systems throughout the year. The purpose of this engagement was to assess the networks' and systems' security and identify potential security flaws by utilizing industry-accepted testing methodology and best practices.

For the testing, we focused on the following:

- Attempting to determine what system-level vulnerabilities could be discovered and exploited with no prior knowledge of the environment or notification to administrators.
- Attempting to exploit vulnerabilities found and access confidential information that may be stored on systems.
- Documenting and reporting on all findings.

All tests took into consideration the actual business processes implemented by the systems and their potential threats; therefore, the results of this assessment reflect a realistic picture of the actual exposure levels to online hackers. This document contains the results of that assessment.

Assessment Objective

The primary goal of this assessment was to provide an analysis of security flaws present in Rekall's web applications, networks, and systems. This assessment was conducted to identify exploitable vulnerabilities and provide actionable recommendations on how to remediate the vulnerabilities to provide a greater level of security for the environment.

We used our proven vulnerability testing methodology to assess all relevant web applications, networks, and systems in scope.

Rekall has outlined the following objectives:

Table 1: Defined Objectives

Objective
Find and exfiltrate any sensitive information within the domain.
Escalate privileges.
Compromise several machines.

Penetration Testing Methodology

Reconnaissance

We begin assessments by checking for any passive (open source) data that may assist the assessors with their tasks. If internal, the assessment team will perform active recon using tools such as Nmap and Bloodhound.

Identification of Vulnerabilities and Services

We use custom, private, and public tools such as Metasploit, hashcat, and Nmap to gain perspective of the network security from a hacker's point of view. These methods provide Rekall with an understanding of the risks that threaten its information, and also the strengths and weaknesses of the current controls protecting those systems. The results were achieved by mapping the network architecture, identifying hosts and services, enumerating network and system-level vulnerabilities, attempting to discover unexpected hosts within the environment, and eliminating false positives that might have arisen from scanning.

Vulnerability Exploitation

Our normal process is to both manually test each identified vulnerability and use automated tools to exploit these issues. Exploitation of a vulnerability is defined as any action we perform that gives us unauthorized access to the system or the sensitive data.

Reporting

Once exploitation is completed and the assessors have completed their objectives, or have done everything possible within the allotted time, the assessment team writes the report, which is the final deliverable to the customer.

Scope

Prior to any assessment activities, Rekall and the assessment team will identify targeted systems with a defined range or list of network IP addresses. The assessment team will work directly with the Rekall POC to determine which network ranges are in-scope for the scheduled assessment.

It is Rekall's responsibility to ensure that IP addresses identified as in-scope are actually controlled by Rekall and are hosted in Rekall-owned facilities (i.e., are not hosted by an external organization). In-scope and excluded IP addresses and ranges are listed below.

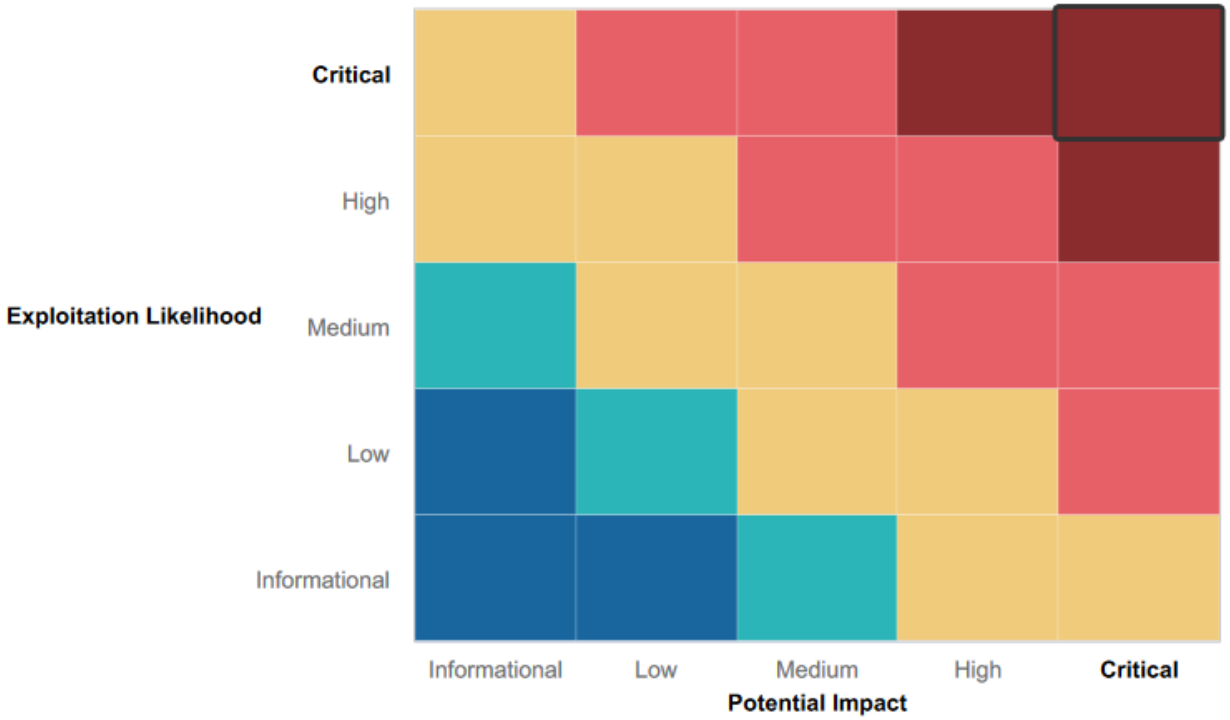
Executive Summary of Findings

Grading Methodology

Each finding was classified according to its severity, reflecting the risk each such vulnerability may pose to the business processes implemented by the application, based on the following criteria:

Critical: Immediate threat to key business processes.
High: Indirect threat to key business processes/threat to secondary business processes.
Medium: Indirect or partial threat to business processes.
Low: No direct threat exists; vulnerability may be leveraged with other vulnerabilities.
Informational: No threat; however, it is data that may be used in a future attack.

As the following grid shows, each threat is assessed in terms of both its potential impact on the business and the likelihood of exploitation:



Summary of Strengths

While the assessment team was successful in finding several vulnerabilities, the team also recognized several strengths within Rekall's environment. These positives highlight the effective countermeasures and defenses that successfully prevented, detected, or denied an attack technique or tactic from occurring.

- There were efforts towards Web Application Sanitization. Which slowed the process of command injections.

Summary of Weaknesses

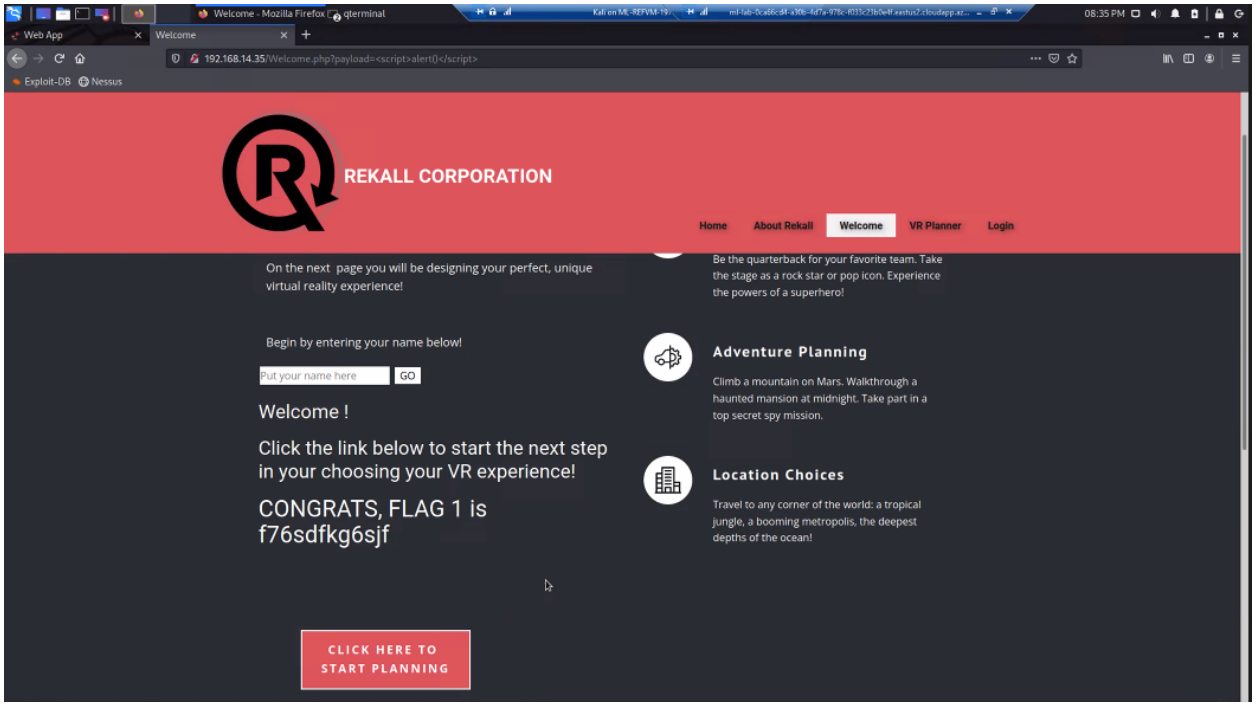
We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

- Weak passwords/password policy.
- Little effort in sanitizing user input validation.
- Weak firewall settings to block certain traffic requests and responses.
- Multiple open ports allowing for shells to be open.
- Found login credentials from Github, and used those to gain access to a shell.

Executive Summary

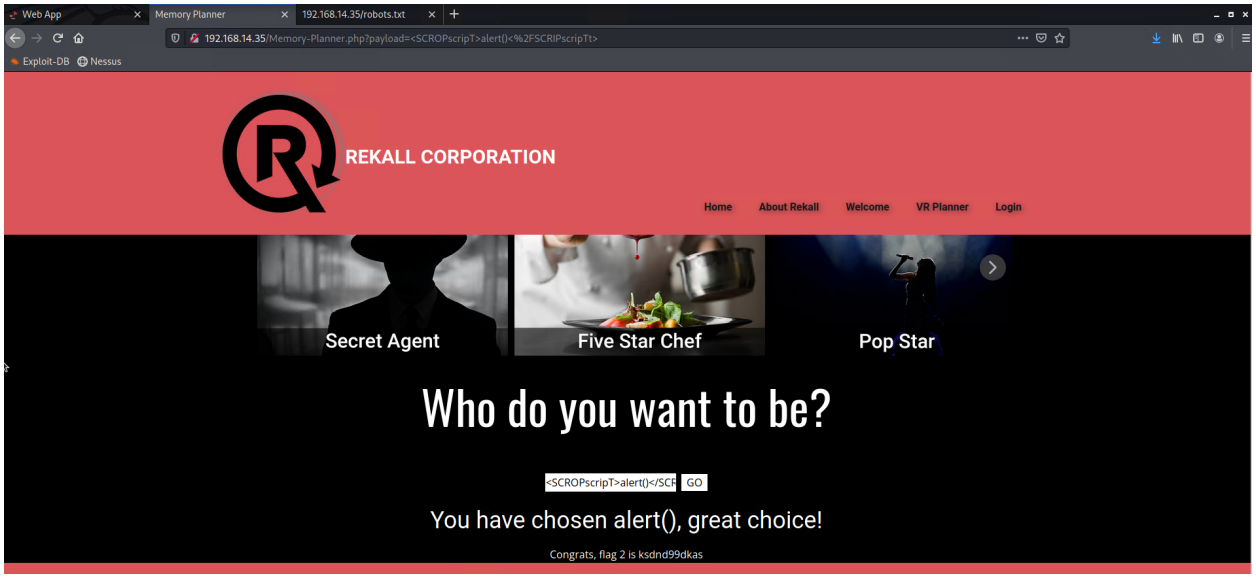
Beginning stages of our penetration test were targeting/exploiting Rekalls Web-Application. Our first attack (S.E) began by exploiting the “Name” field inside Rekalls “Welcome” page, by typing in a Javascript to give us an Alert Response. The same exploit was used on the “Memory-Planner” character entry field. Using the same exploit in the comments field on the comments page but modifying it; Shadow Entity was able to store scripts and allow users to pull data.

Using javascript Shadow Entity was able to successfully create an alert on Rekall Corps ‘Welcome Page’
<script>alert(1)</script> was ran to achieve this.



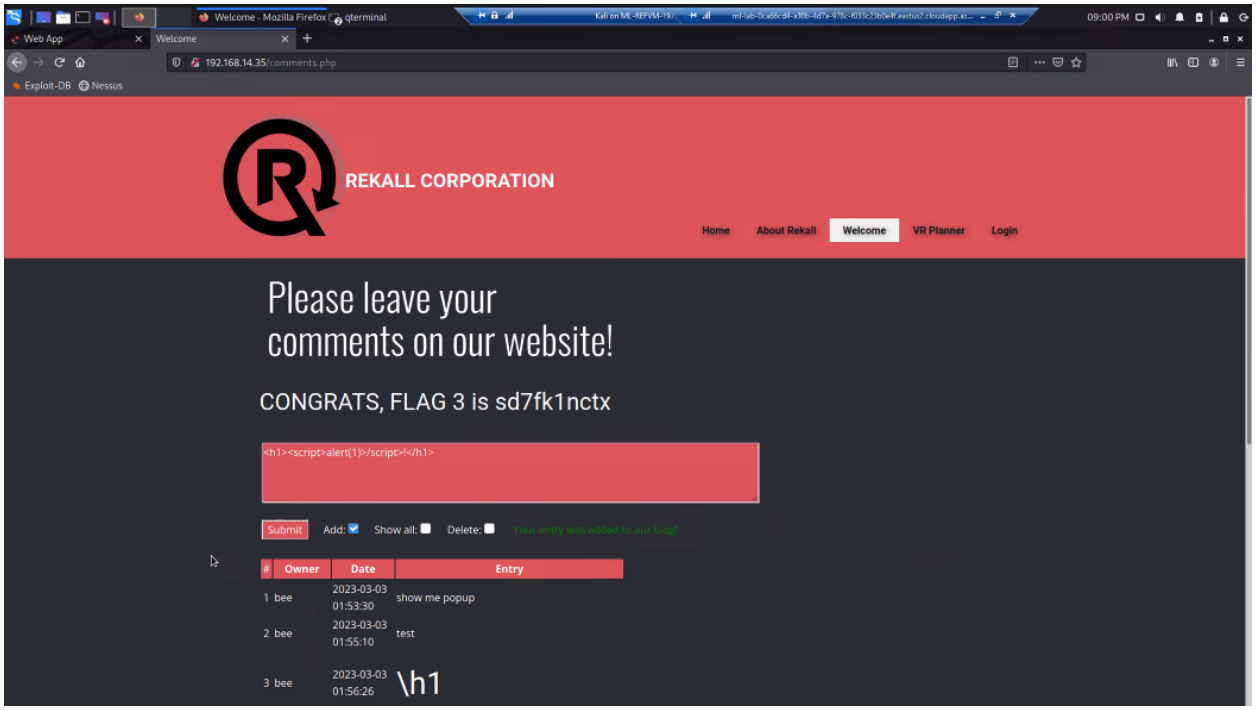
1.

Reflected XSS



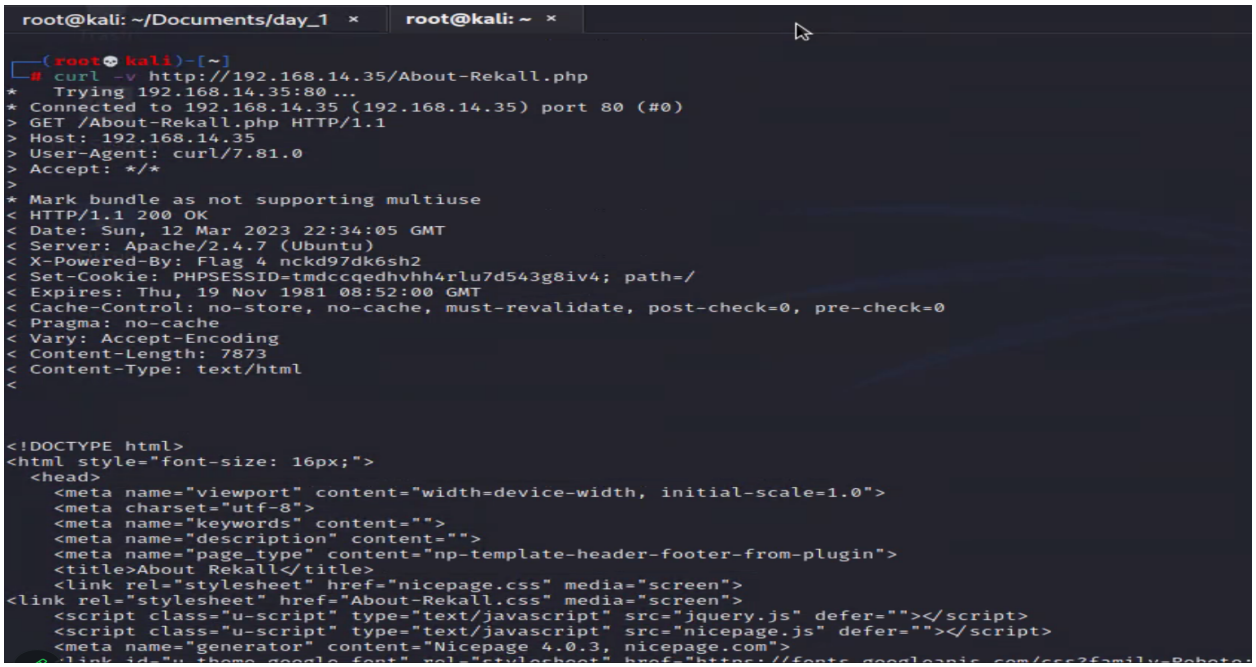
2.

Stored XSS Input



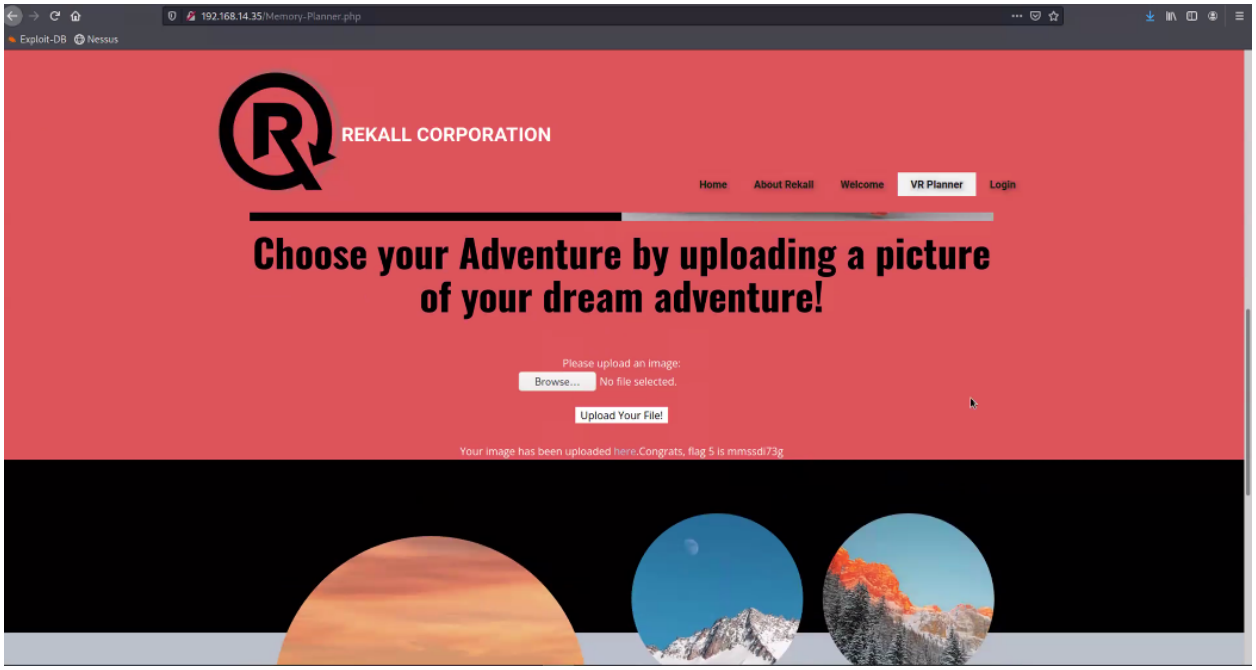
3.

Output from CURLing

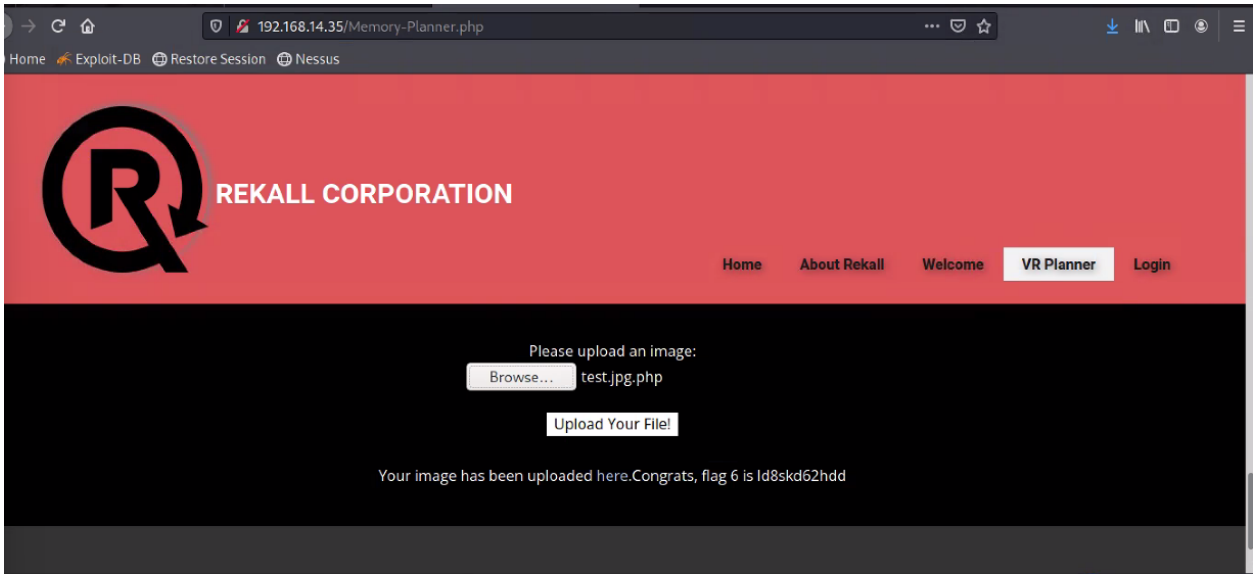


4.

Using test.jpg.php to both upload fields Shadow Entity was able to bypass user input Whitelist.

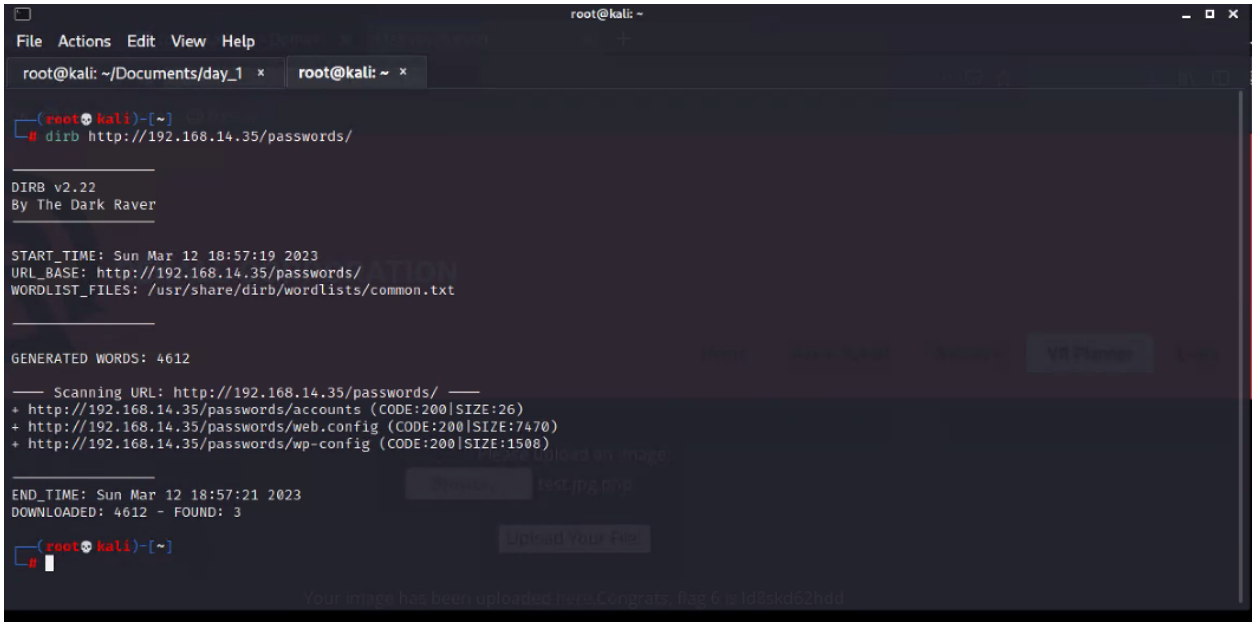


5.



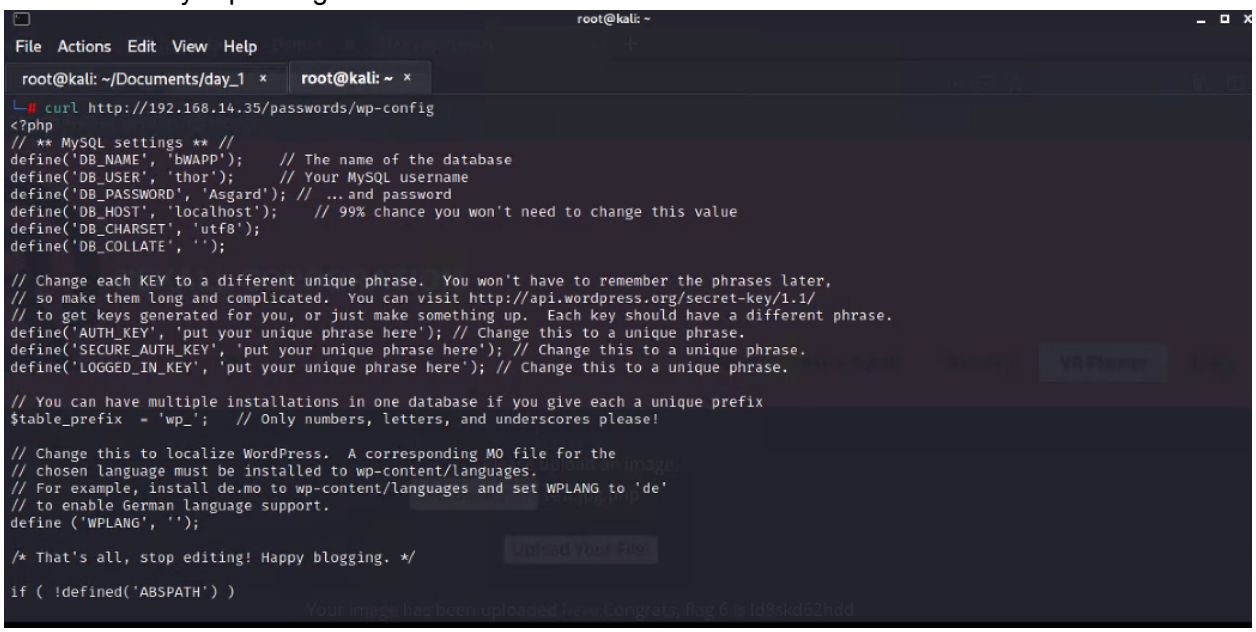
6.

Using dirb in the Command Line Shadow Entity was able to find a file that contains a user and password.



7.

Inside Directory ‘wp-config’

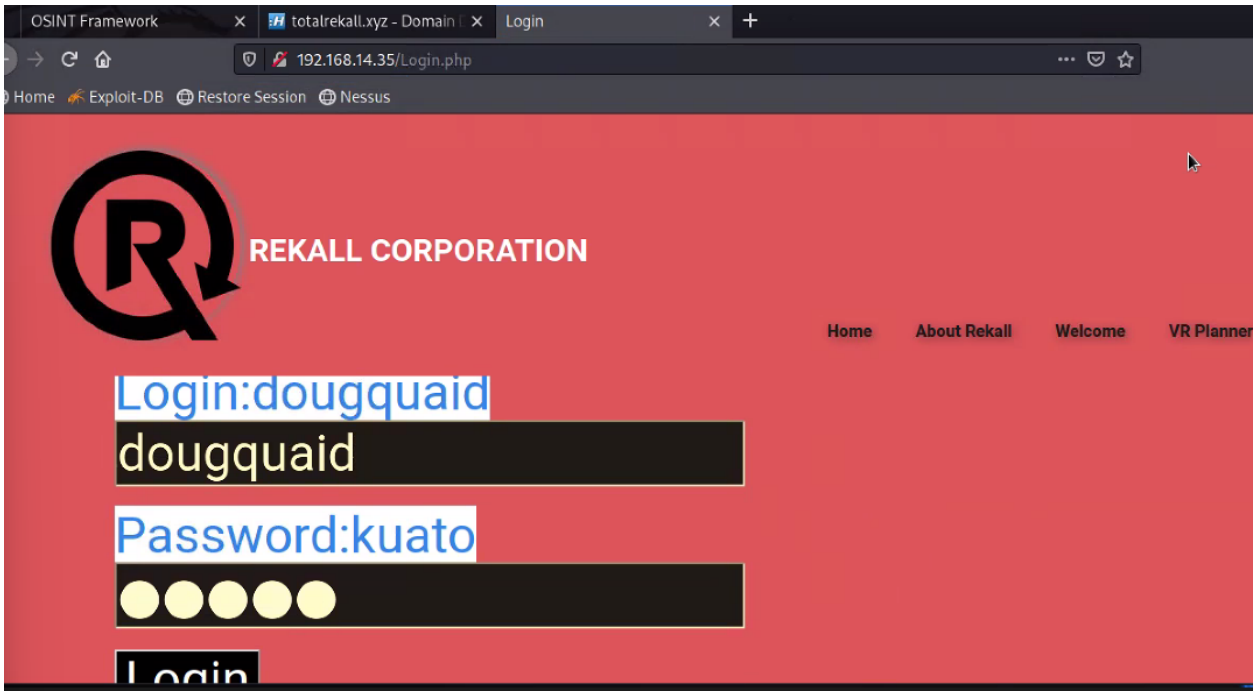


8.

Using a SQL injection in the login page Shadow Entity was able to login.

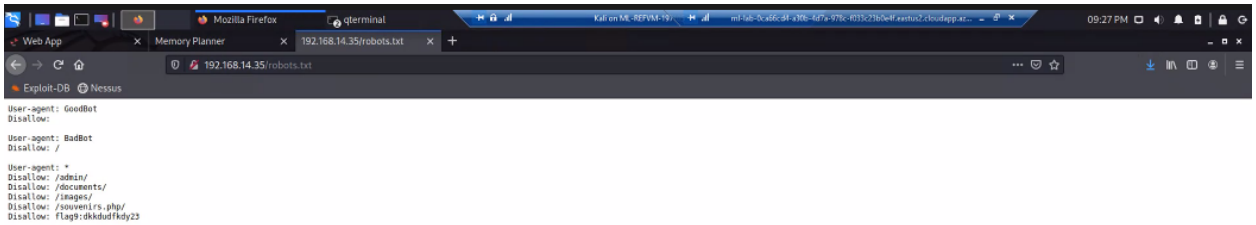
9.

By highlighting the System Administrator section of the “Login” page. S.E was able to find credentials.



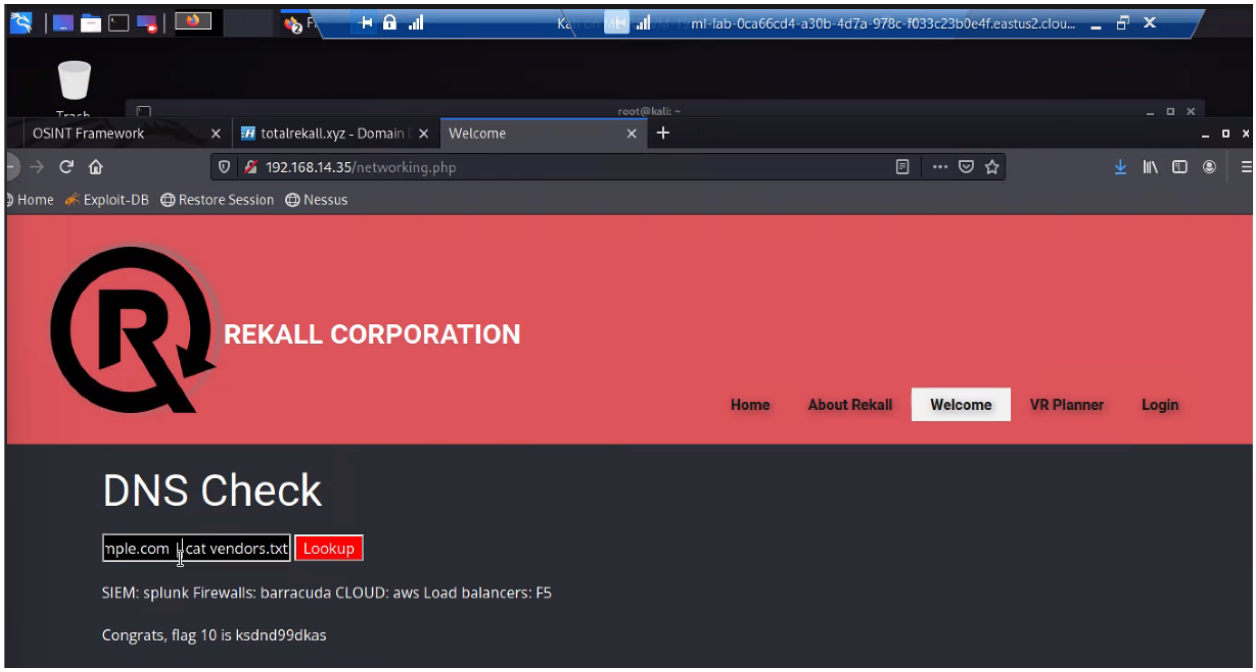
10.

Found the robots.txt by simply by adding /robots.txt to the end of Rekall Corp's URL



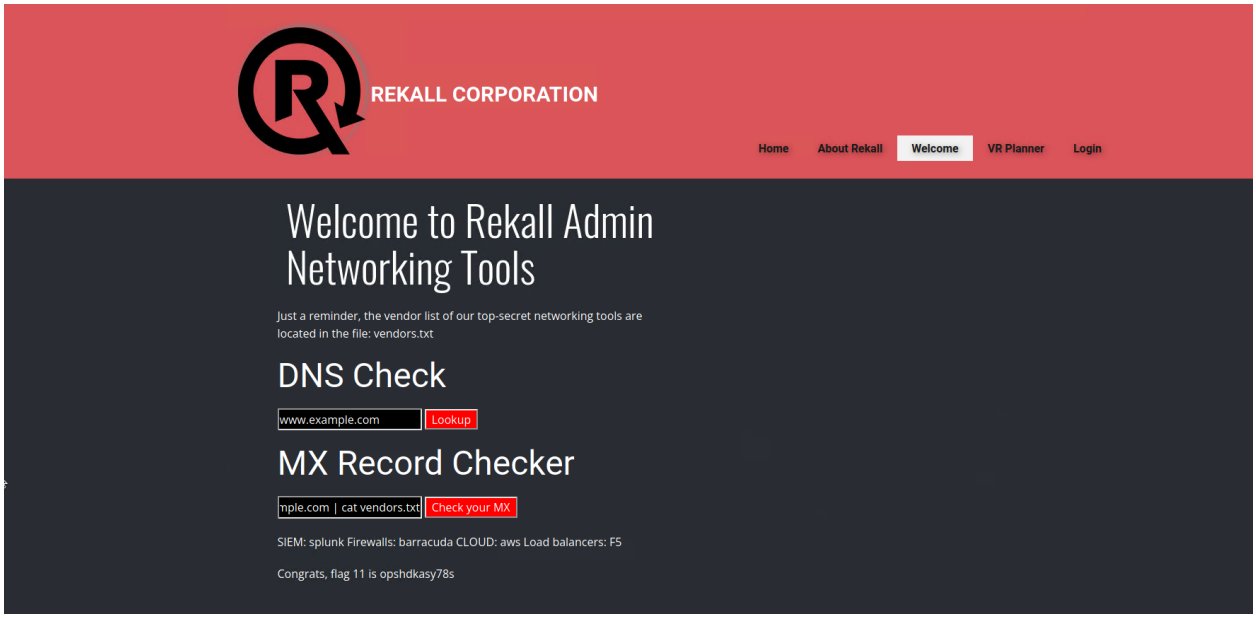
11.

DNS check field and result



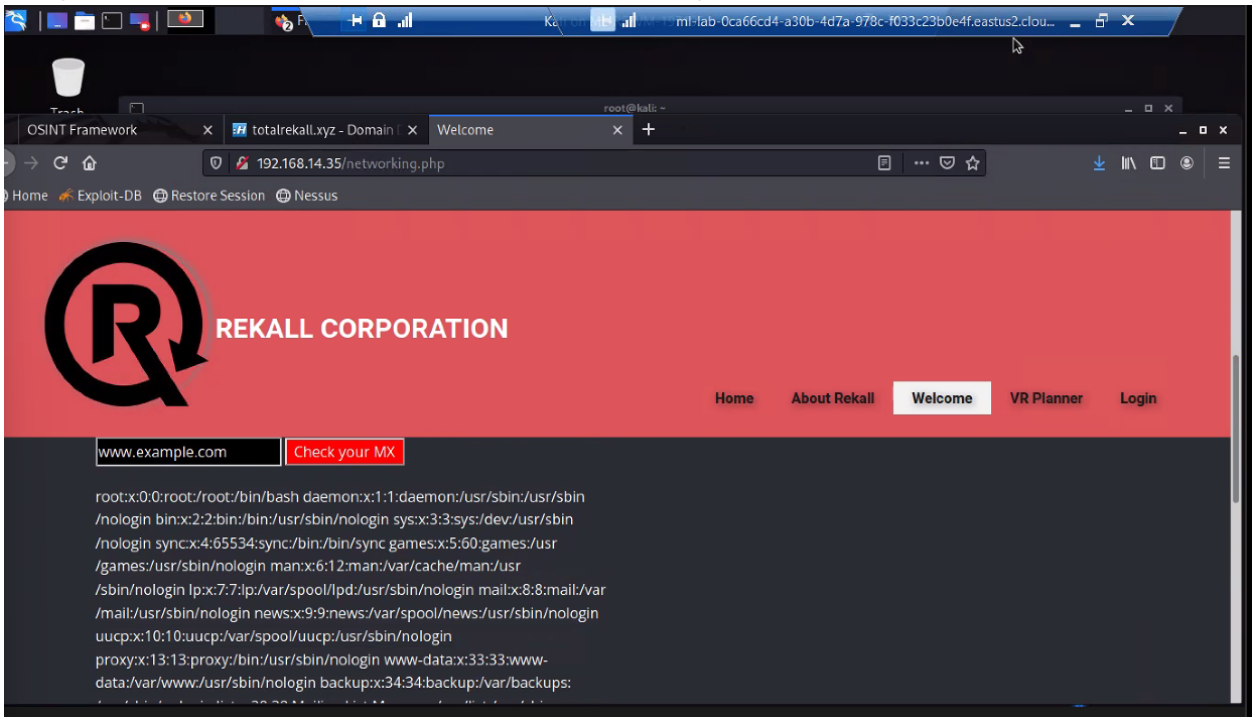
12.

MX-record Checker field input



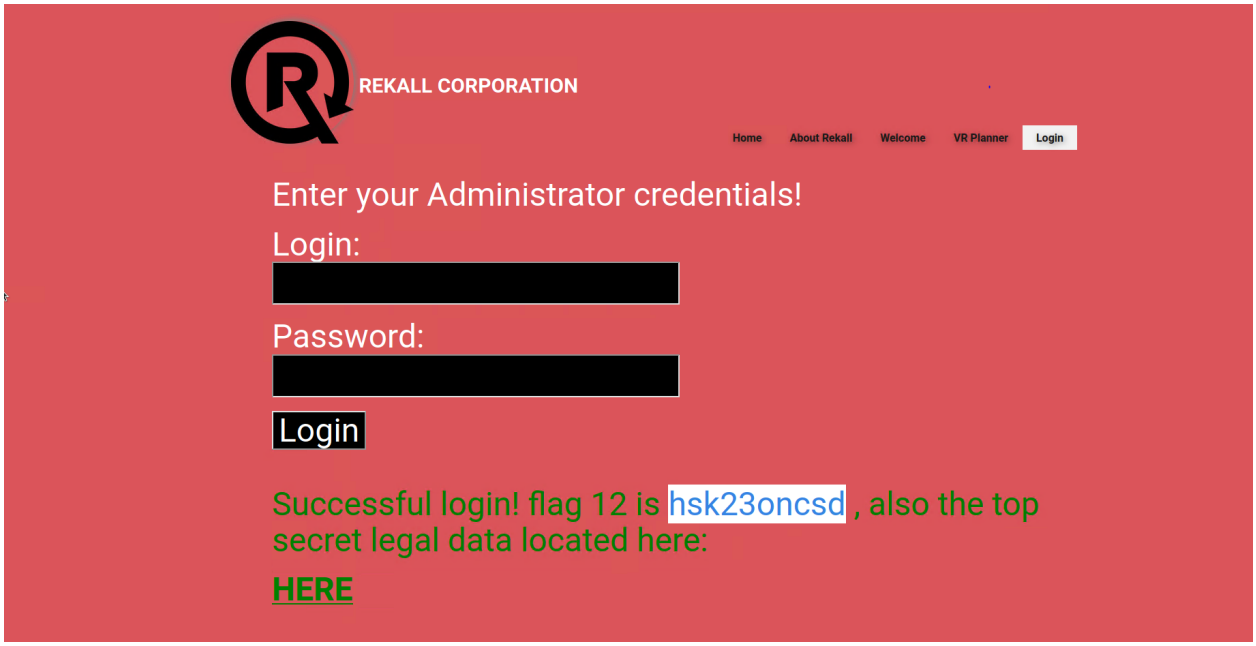
13.

Using the same input I pulled info from /etc/passwd using the ‘cat’ command



14.

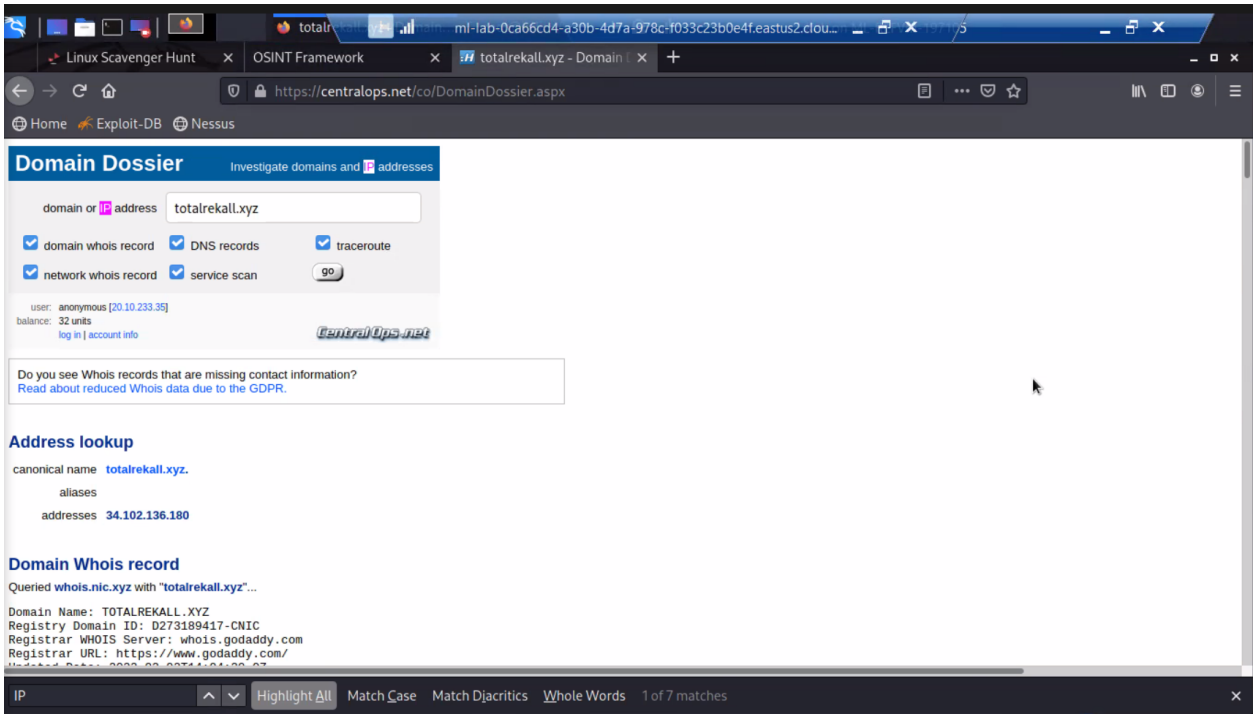
By searching the /etc/passwd directory S.E was able to find the login credentials for ‘melina’ and brute force her password.



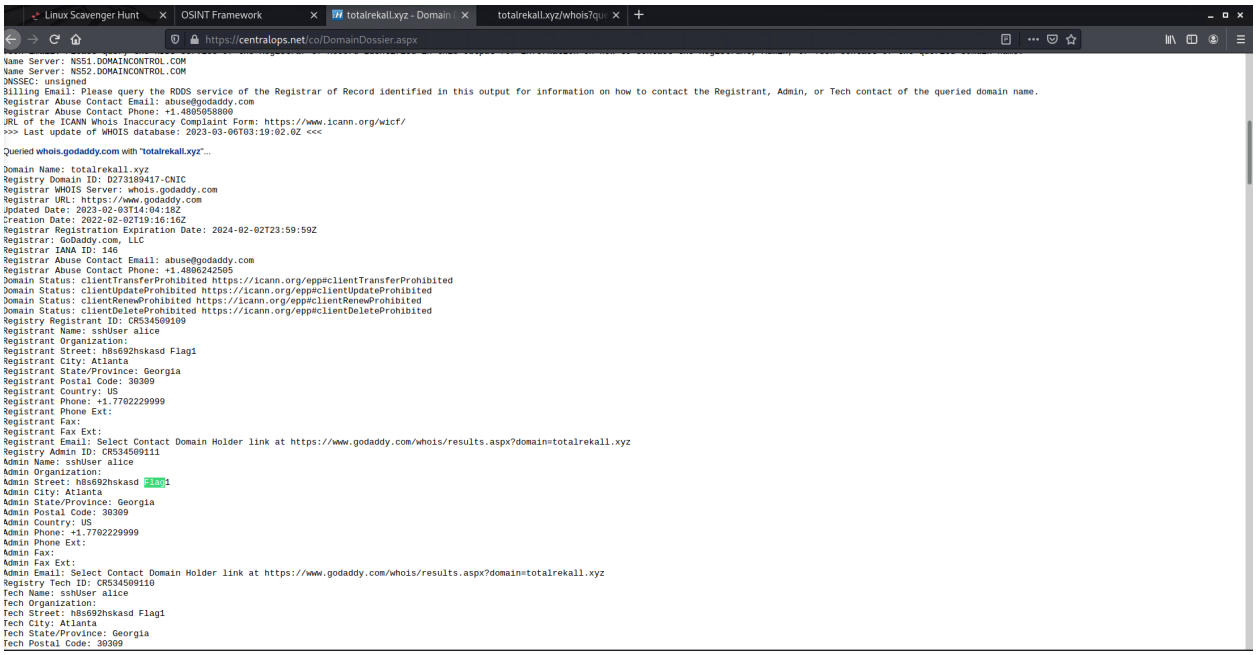
15.

Executive Summary Day Two

Using the OSINT Framework tool Domain Dossier, S.E was able to pull sensitive information from the database.

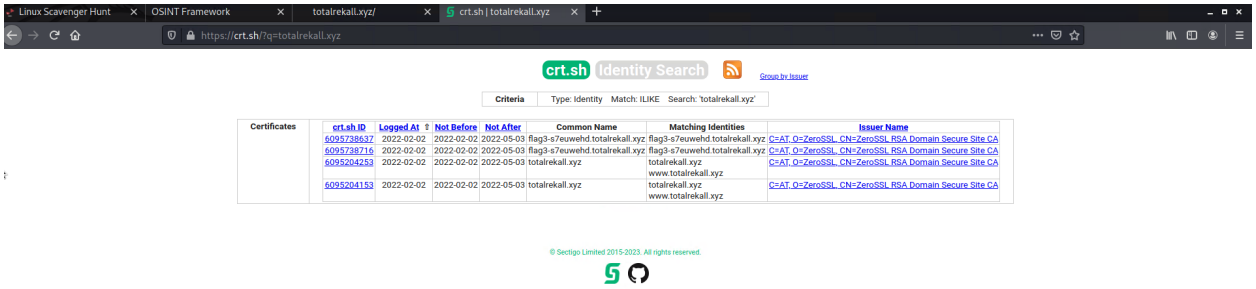


16.



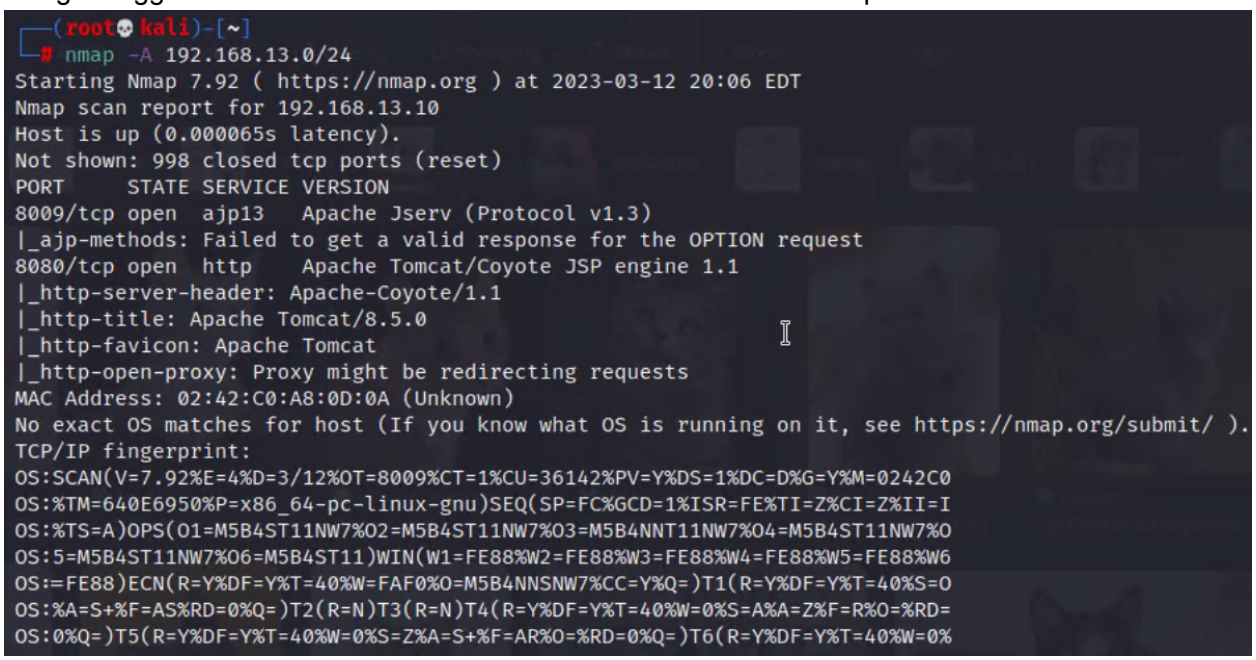
17.

By using cert.sh I was able to pull the certificate from Rekall Crop's Web Application.



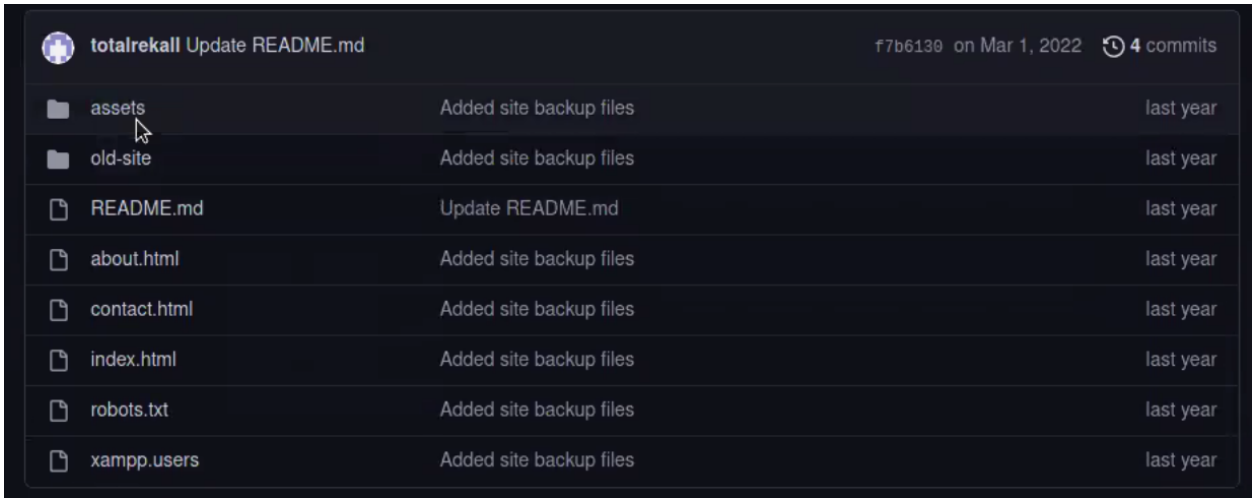
18.

Using an aggressive scan S.E was able to see all five hosts that are up.

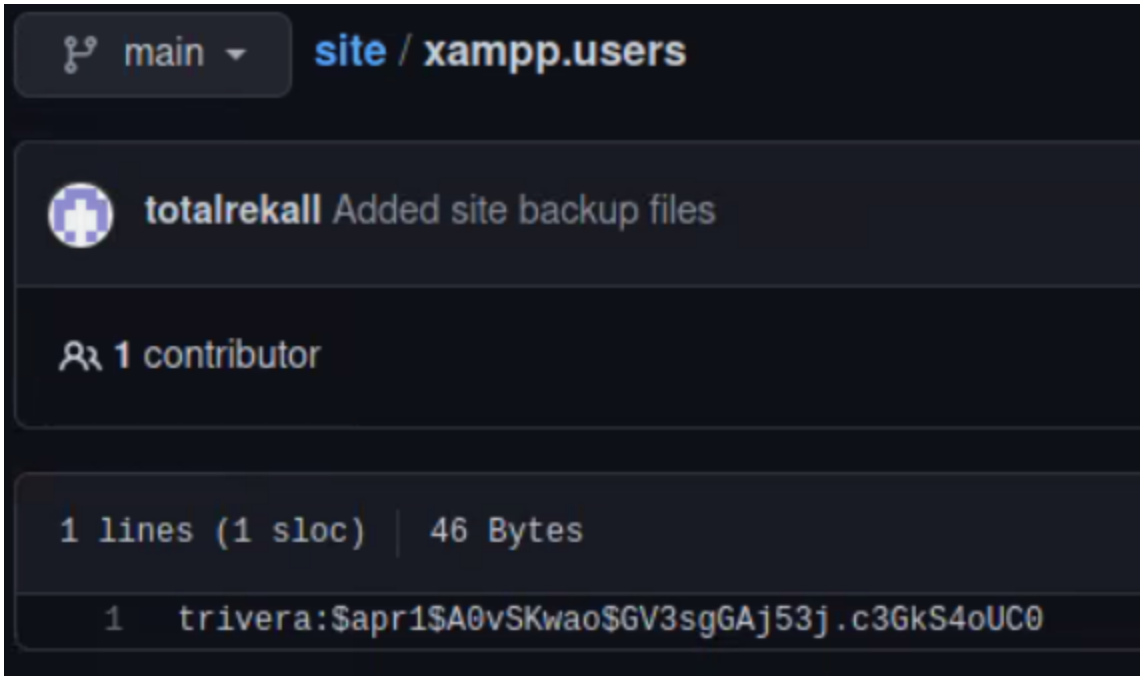


19.

S.E was able to find a GitHub page that has sensitive information and HTML code.



20.



21.

Summary Vulnerability Overview

Vulnerability	Severity
Flag 1: (Web App)Reflective XSS	medium
Flag 2: (Web App)Reflective XSS	medium
Flag 3: (Web App)Stored XSS	critical
Flag 4: (Web App)Exposed Sensitive Data	critical
Flag 5: (Web App)Local File Inclusion	critical
Flag 6 (Web App)Local File Inclusion	critical
Flag 7: (Web App)Exposed Sensitive Data	critical
Flag 8: (Web App)Exposed Sensitive Data	critical
Flag 9: (Web App)Exposed Sensitive Data	critical
Flag 10: (Web App)Command Injection	critical
Flag 11: (Web App)Command Injection	critical
Flag 12: (Web App)Brute-Force Attack	critical
Flag 13: (Linux)Open-Source Vulnerability	critical
Flag 14: (Linux)Domain Ping	low
Flag 15: (Linux)Open-Source Vulnerability	low
Flag 16: (Linux)Network Mapping Scan	medium
Flag 17: (Windows)Exposed Sensitive Data	critical
Flag 18: (Windows)Exposed Sensitive Data	critical
Flag 19: (Windows)Vulnerability FTP port 21	critical
Flag 20: (Windows)Vulnerability port 110	critical

The following summary tables represent an overview of the assessment findings for this penetration test:

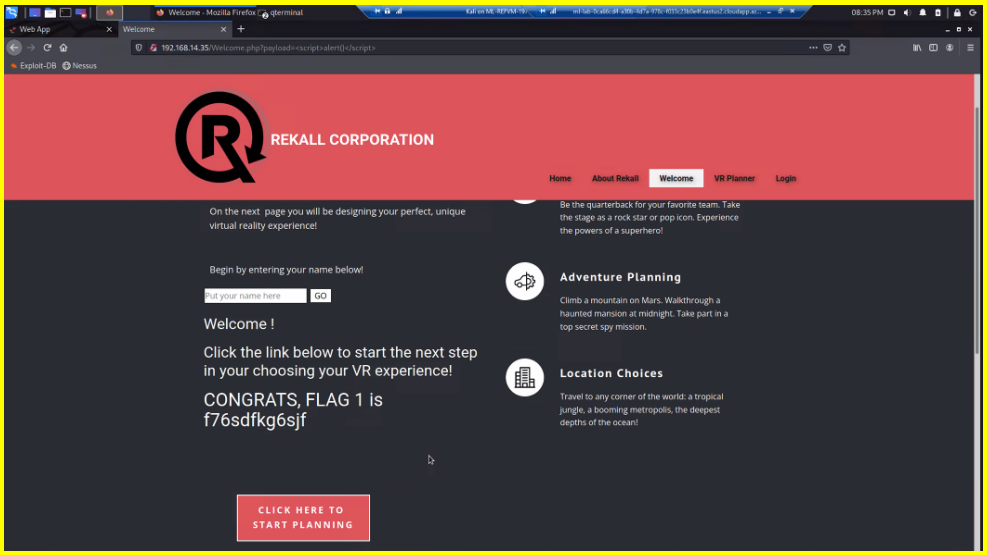
Scan Type	Total

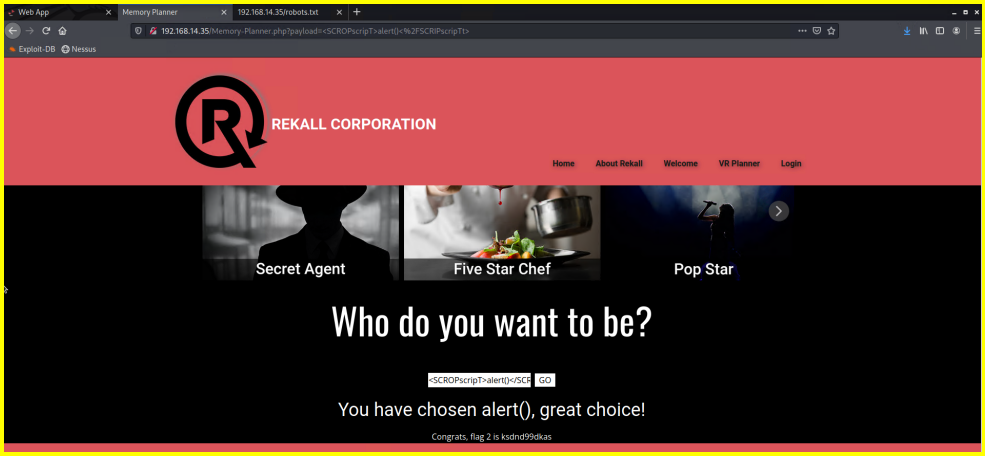
Hosts	• totalrekall.xyz
	• 192.168.13.0/24
	  ○ 192.168.13.10
	  ○ 192.168.13.11
	  ○ 192.168.13.12
	  ○ 192.168.13.13
	  ○ 192.168.13.14
	  ○ 192.169.13.1
	• 172.22.117.0/24
	  ○ 172.22.117.10
	  ○ 172.22.117.20
	• https://github.com/totalrekall/site
	• 192.168.14.35

Ports	80, 8080, 22, 5901, 6001, 10000, 10001, 3306, 53, 88, 135, 139, 389, 445, 446, 593, 636, 3268, 3269, 21, 25, 106, 110, 443, 79

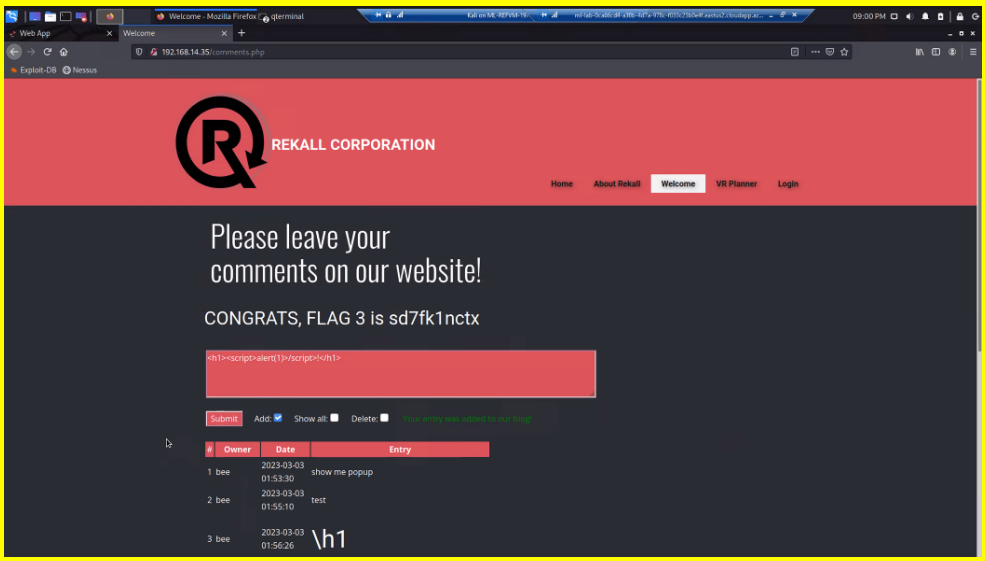
Exploitation Risk	Total
Critical	15
High	0
Medium	3
Low	2

Vulnerability Findings

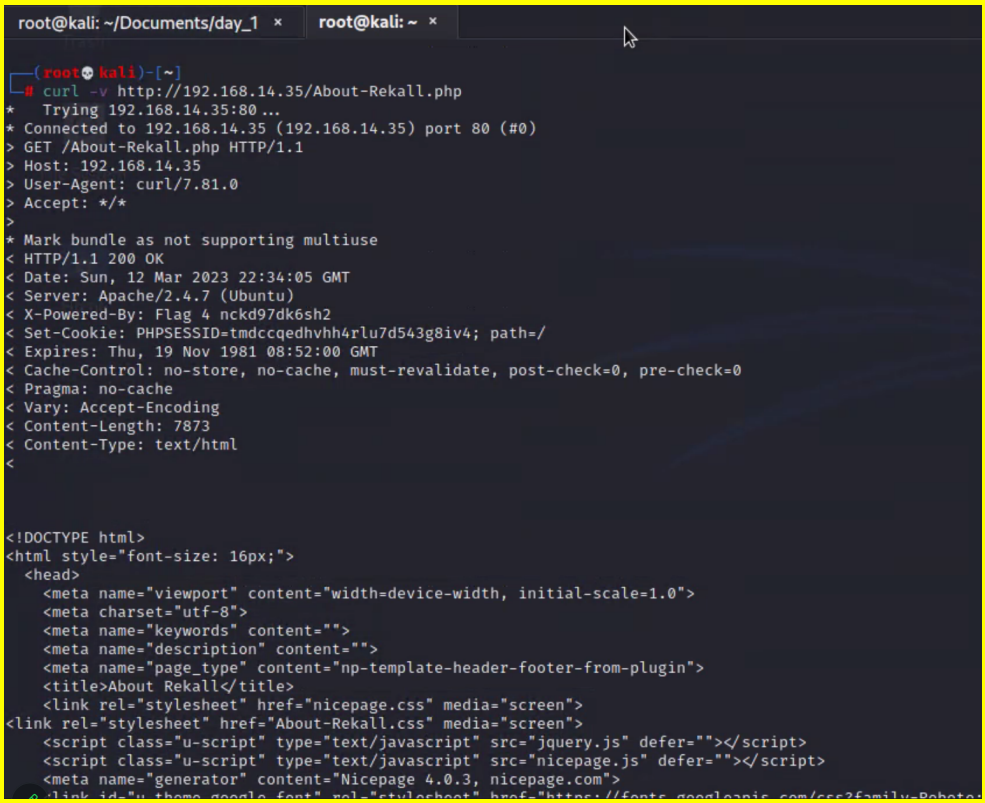
Vulnerability 1	Findings
Title	Reflective XSS
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Medium
Description	cross-site scripting (XSS) attack where an attacker injects malicious code into a website that is then executed by a victim's browser.
Images	
Affected Hosts	192.168.14.35
Remediation	User Input Validation/Sanitizing

Vulnerability 2	Findings
Title	Reflective XSS
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Medium
Description	Stored XSS is similar to reflective, but stored is permanently stored in a target's server.
Images	
Affected Hosts	192.168.14.35
Remediation	Similar to vulnerability number 1, reflective cross site scripting is nearly impossible to fully protect against, but it's important to protect against the most impactful inputs.

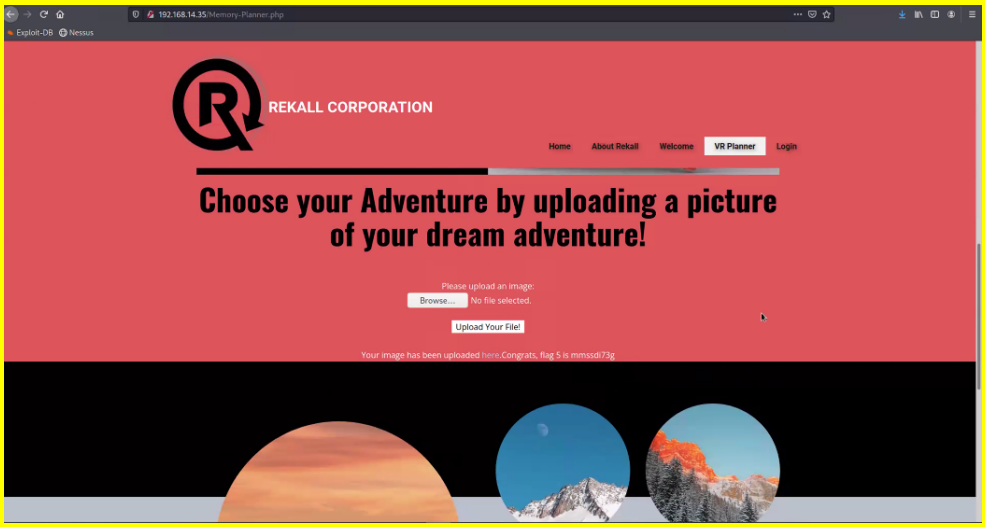
Vulnerability 3	Findings
Title	Stored XSS
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	critical
Description	Stored cross-site scripting is potentially more dangerous than Reflected XSS because user inputs are stored within the target servers. These stored scripts and inputs can be later executed by the user causing significant damages.

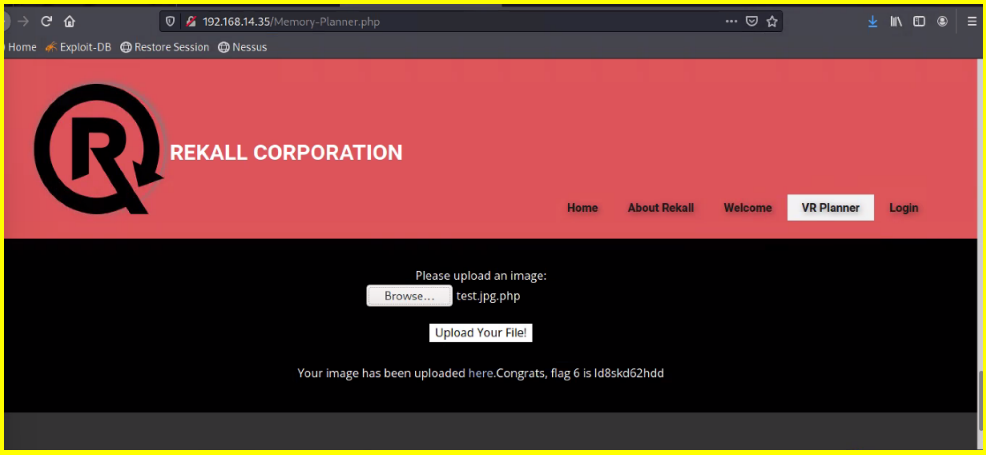
Images	
Affected Hosts	192.168.14.35
Remediation	WAF's can be useful against this kind of attack, because they can filter and monitor traffic within the web application.

Vulnerability 4	Findings
Title	Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	S.E was able to use the cURL command to access flag 4 which was in plaintext format.

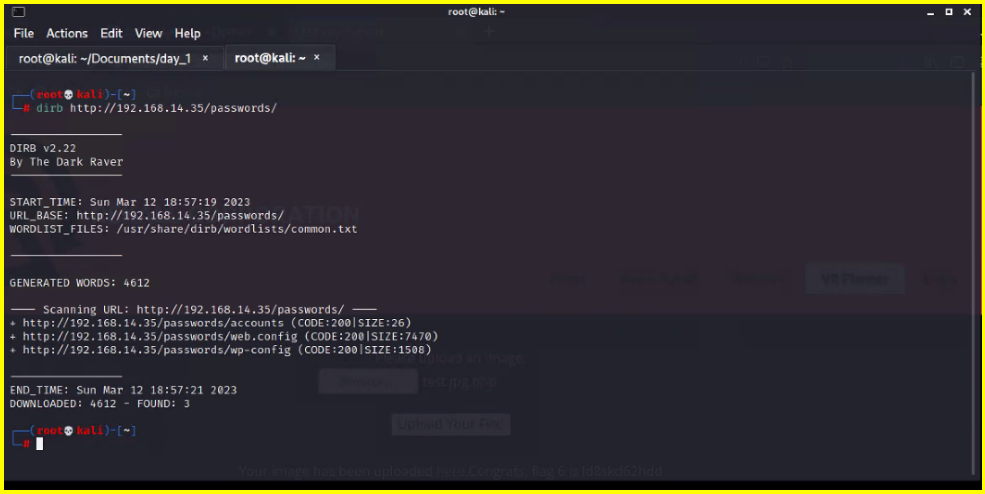
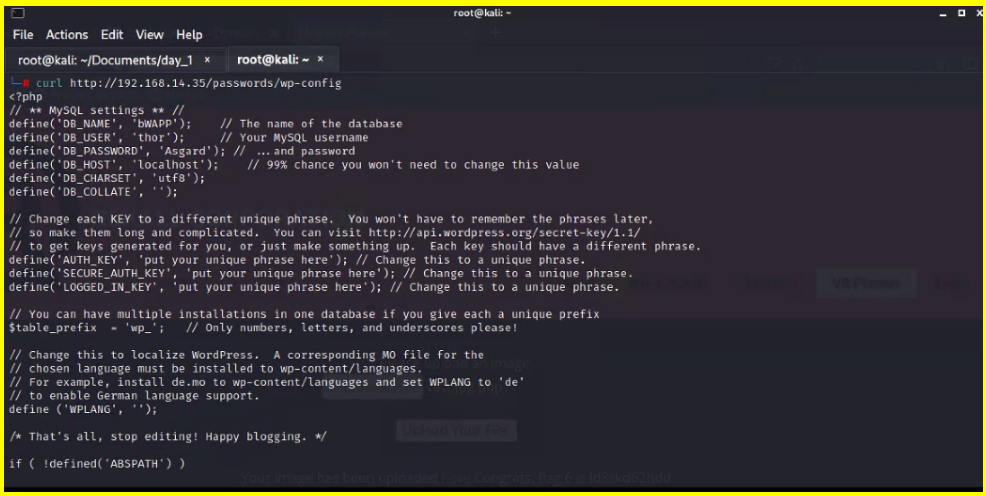
Images	
Affected Hosts	192.168.14.35
Remediation	Using an encryption method, hashes, keys can help hide sensitive data and make it much harder to access freely.

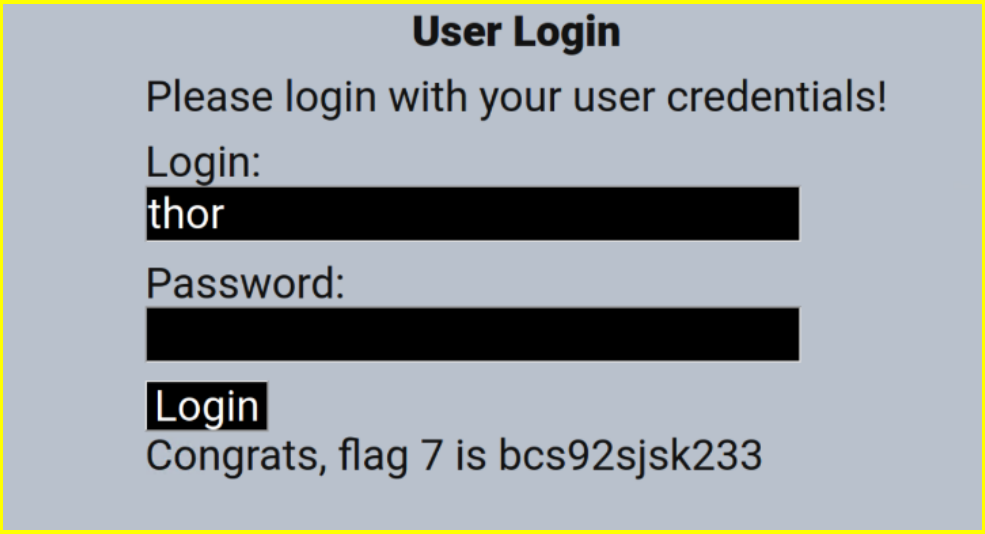
Vulnerability 5	Findings
Title	Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	When an attacker includes a path or hidden value in the input field. In this case S.E was able to upload a file that was not a jpg.

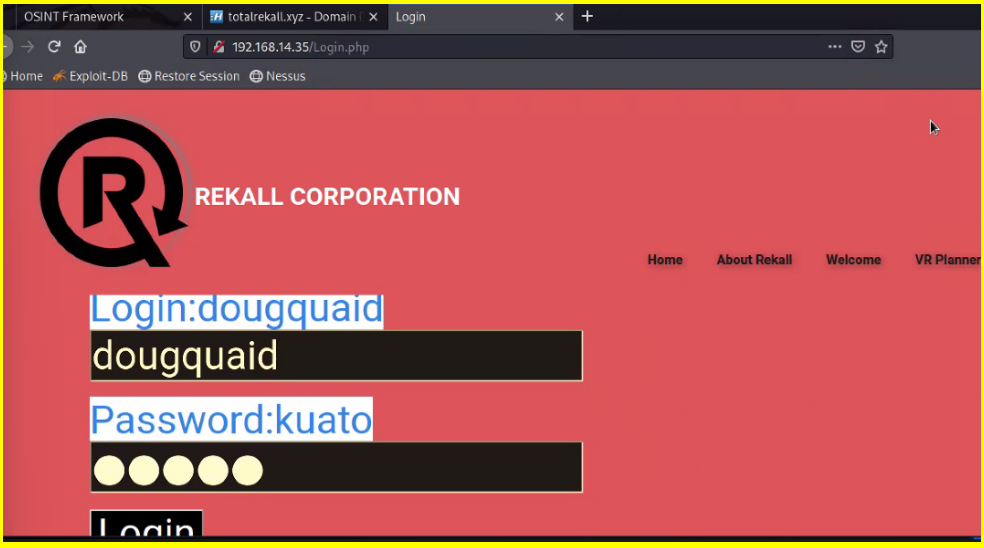
Images	
Affected Hosts	192.168.14.35
Remediation	Only allow jpg images to be submitted in this field.

Vulnerability 6	Findings
Title	Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	critical
Description	The exact same as vulnerability number 6.
Images	
Affected Hosts	192.168.14.35
Remediation	Only allow jpg images to be submitted in this field.

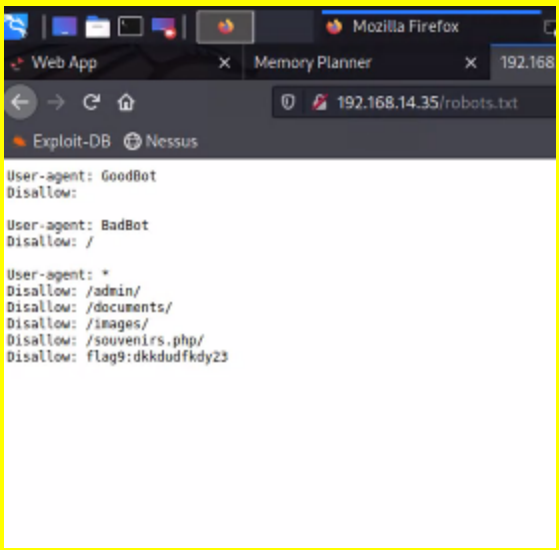
Vulnerability 7	Findings

Title	Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	critical
Description	Using Dirb S.E was able to scan every directory and locate a “passwords” directory. Then using cURL S.E was able to find log-in credentials, and successfully log in using them.
Images	 

	
Affected Hosts	192.168.14.35
Remediation	Using a WAF (web application firewall) to monitor HTTP requests and blocking constant HTTP requests from a single IP.

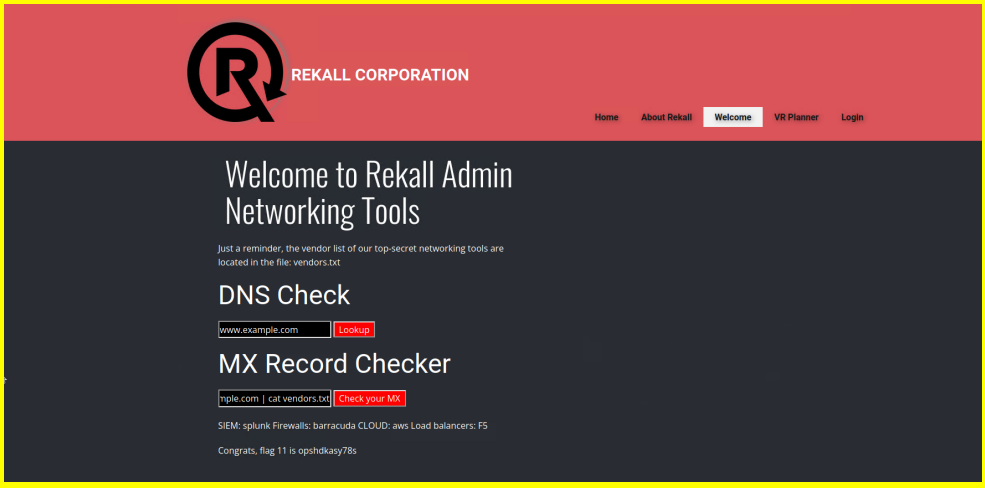
Vulnerability 8	**Findings**
Title	Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	critical
Description	Password/Username was added into the Web Applications HTML code.
Images	
Affected Hosts	192.168.14.35

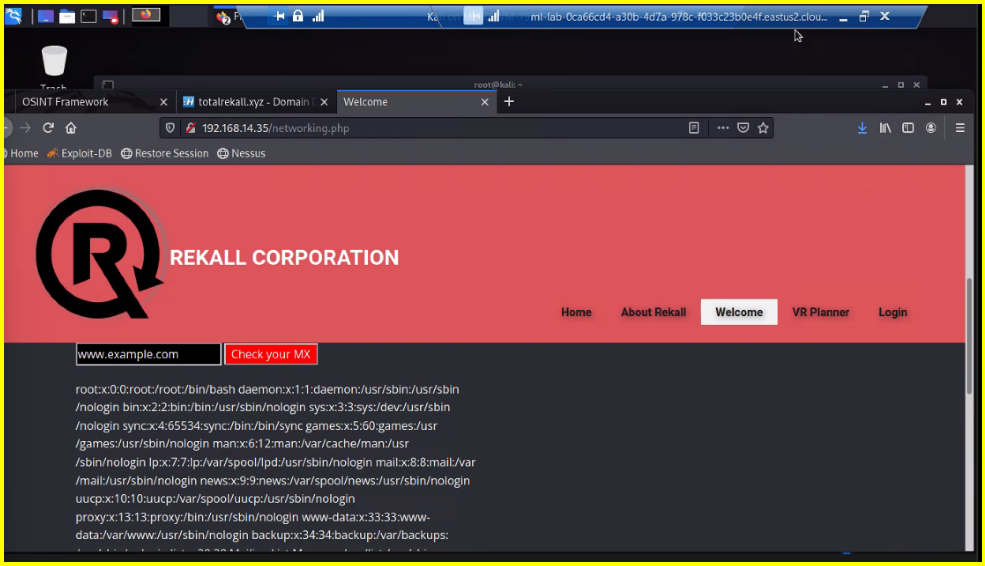
Remediation	Remove credentials from HTML code. Let the user know to change his password.

Vulnerability 9	Findings
Title	Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	critical
Description	S.E was able to find the robots.txt directory through manual input (typing /robots.txt)
Images	
Affected Hosts	192.168.14.35
Remediation	Do not use the robots.txt to hide directories and other information from search results.

Vulnerability 10	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	critical
Description	Using a ‘true’ statement and injecting it into the password field, S.E was successfully able to log in.

Images	
Affected Hosts	192.168.14.35
Remediation	Input validation/sanitization.

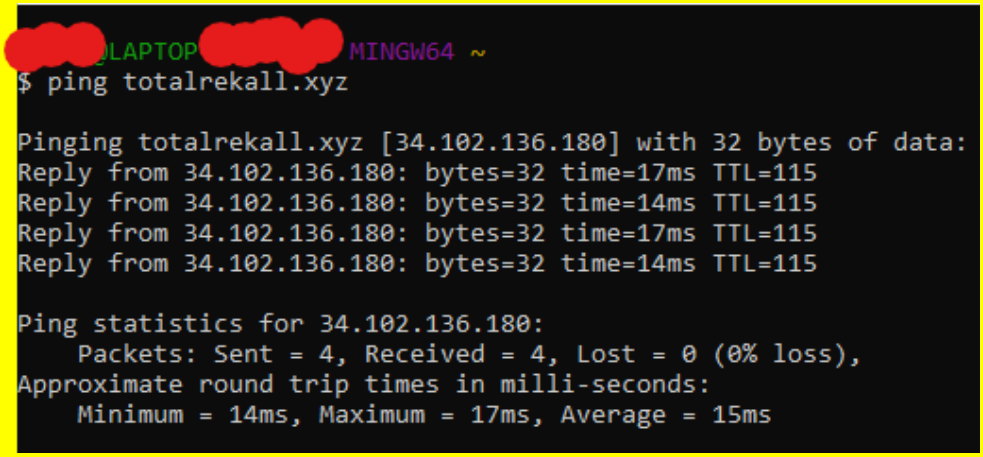
Vulnerability 11	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	critical
Description	Inside the DNS field we were able to use commands to pull information from the Vendors.txt file.
Images	
Affected Hosts	192.168.14.35
Remediation	Some ways to remediate this exploit would be to sanitize input, parameterized queries, whitelisting, WAF

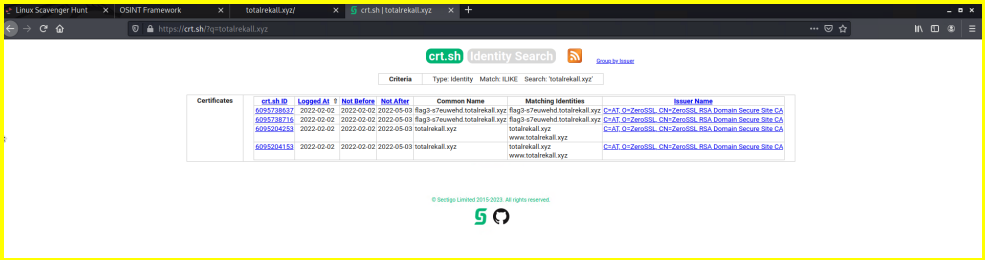
Vulnerability 12	Findings
Title	Brute Force
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	critical
Description	S.E was able to use command injections to get access to the etc/passwd file and then was abe to brute force the hashes using John The Ripper.
Images	
Affected Hosts	192.168.14.35
Remediation	You could remediate this the same way suggested for vulnerability 11.

Linux Server Vulnerabilities

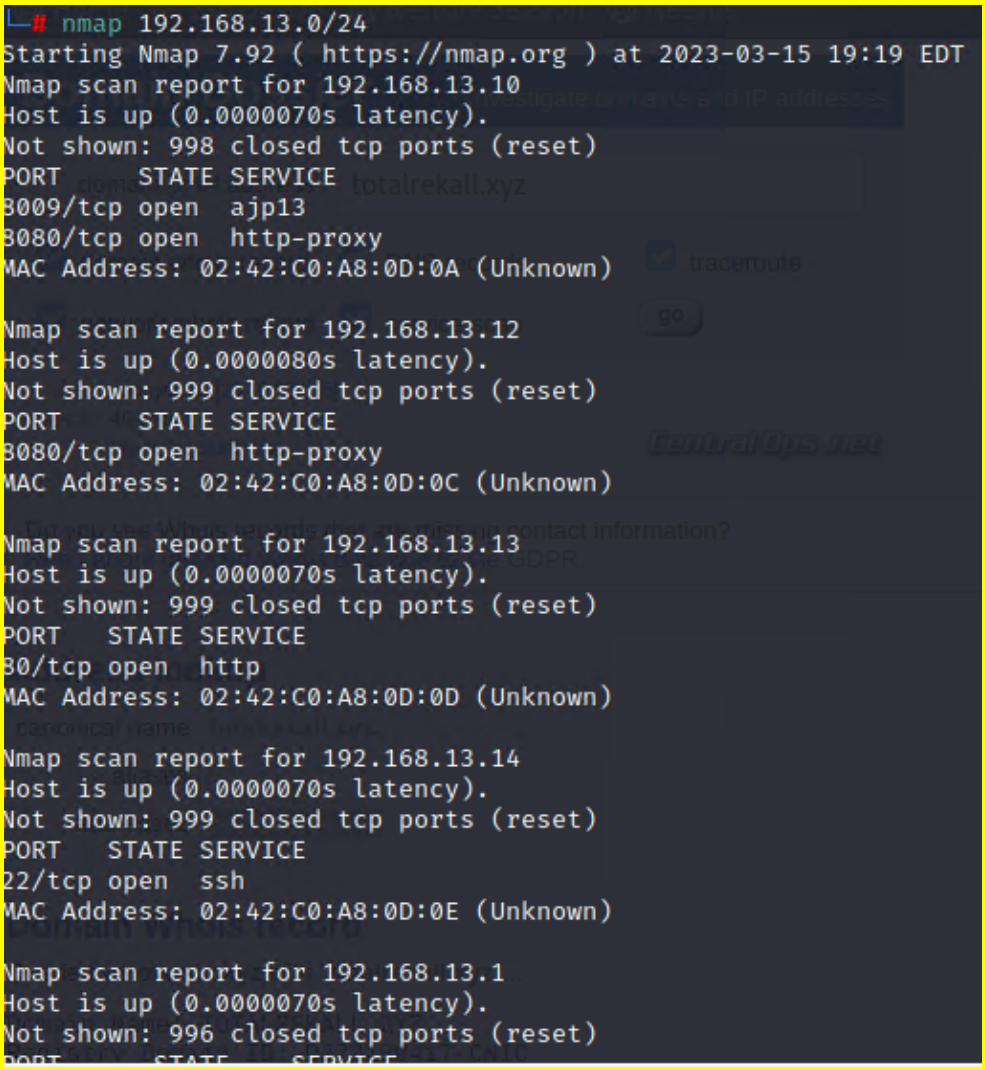
Vulnerability 13	Findings
Title	Open-Source Vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	critical
Description	Using the OSINTframework Domain Dossier to pull “WHOIS” data.

Images	
Affected Hosts	totalrekall.xyz
Remediation	Sensitive information should never be in plaintext format. Try using key encryption, hashing.

Vulnerability 14	Findings
Title	Domain Ping
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low-critical
Description	Pinging isn't inherently bad, and can actually be useful to see if a server is up and running. But it can be exploited to DDOS a server.
Images	
Affected Hosts	totalrekall.xyz
Remediation	As stated before Pinging can be exploited to flood a server. Use a firewall implement rate limiting: only allowing a certain amount of pings to be sent from a specific IP/network.

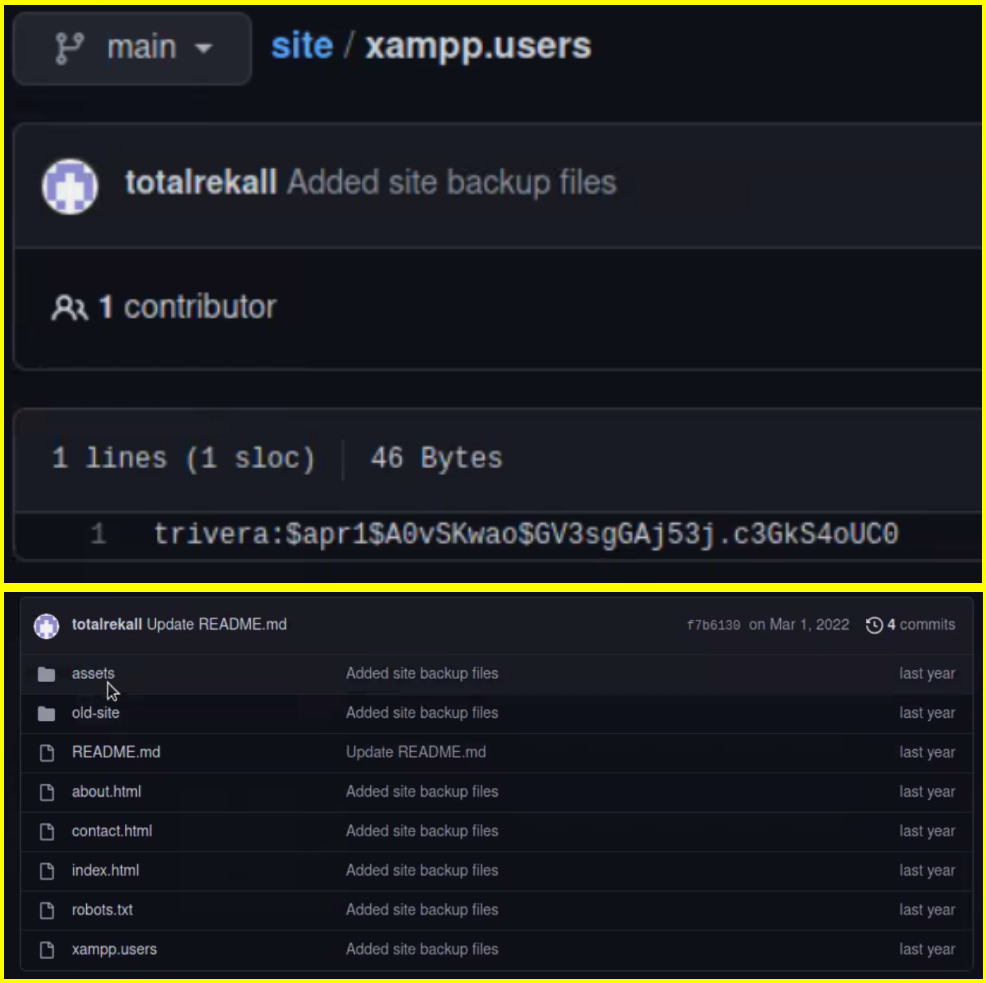
Vulnerability 15	Findings
Title	Open Source Vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low-critical
Description	Used cert.sh to pull certificates of totalrekall.xyz
Images	
Affected Hosts	totalrekall.xyz
Remediation	Similar to vulnerability 13, using various encryption methods can easily remediate this vulnerability.

Vulnerability 16	Findings
Title	Network Mapping
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low-critical
Description	Similar to vulnerability 14, network mapping/scanning isn't inherently a bad thing, but attackers can find devices on a network that aren't properly secured.

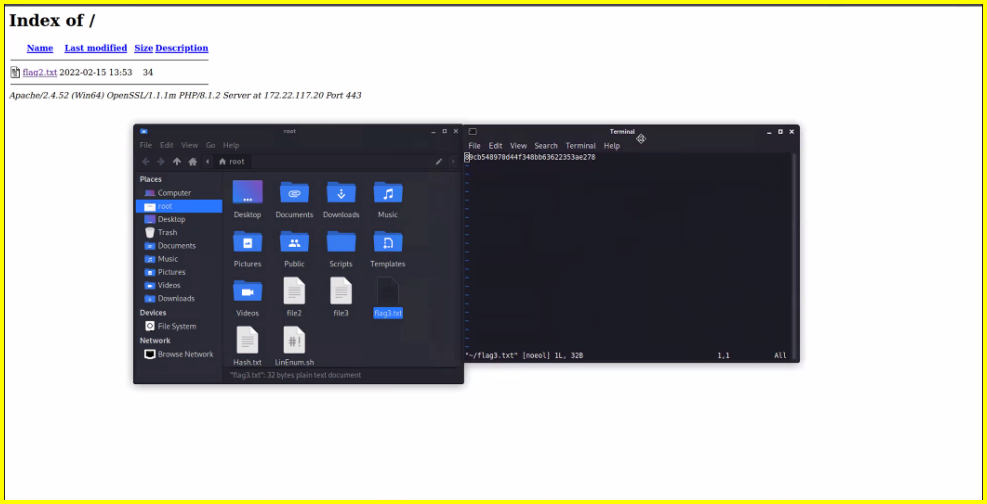
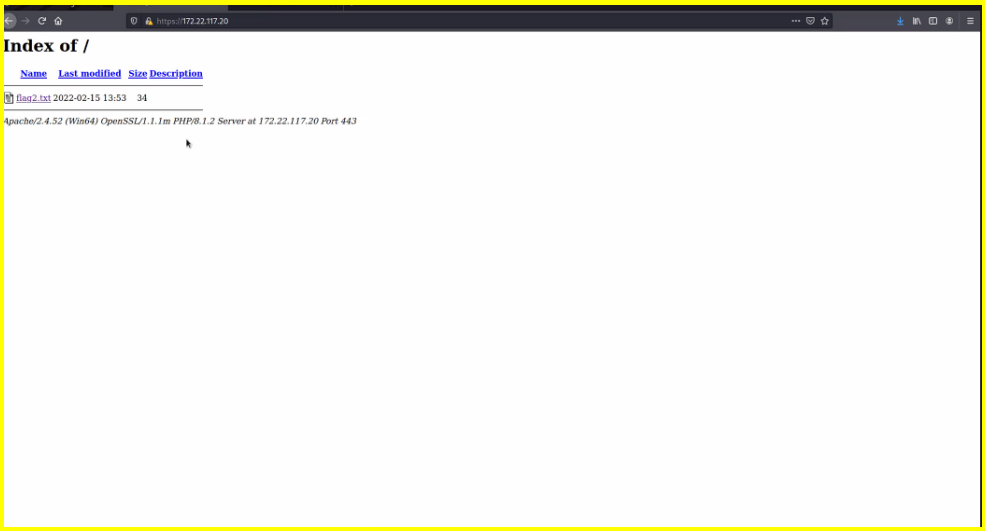
Images	
Affected Hosts	192.168.13.0/24 192.168.13.10 192.168.13.11 192.168.13.12 192.168.13.13 192.168.13.14 192.169.13.1
Remediation	To mitigate risk, implementing security measures such as access controls, firewalls, and intrusion detection and prevention systems.

Windows Server Vulnerabilities

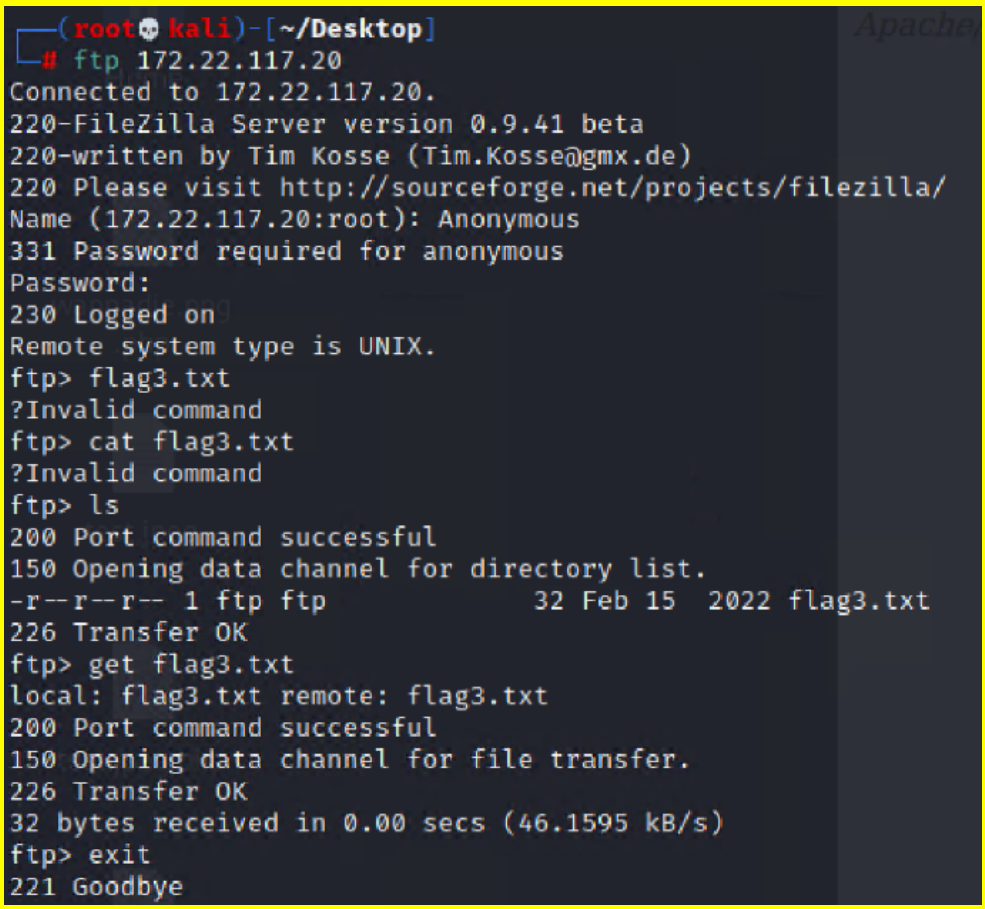
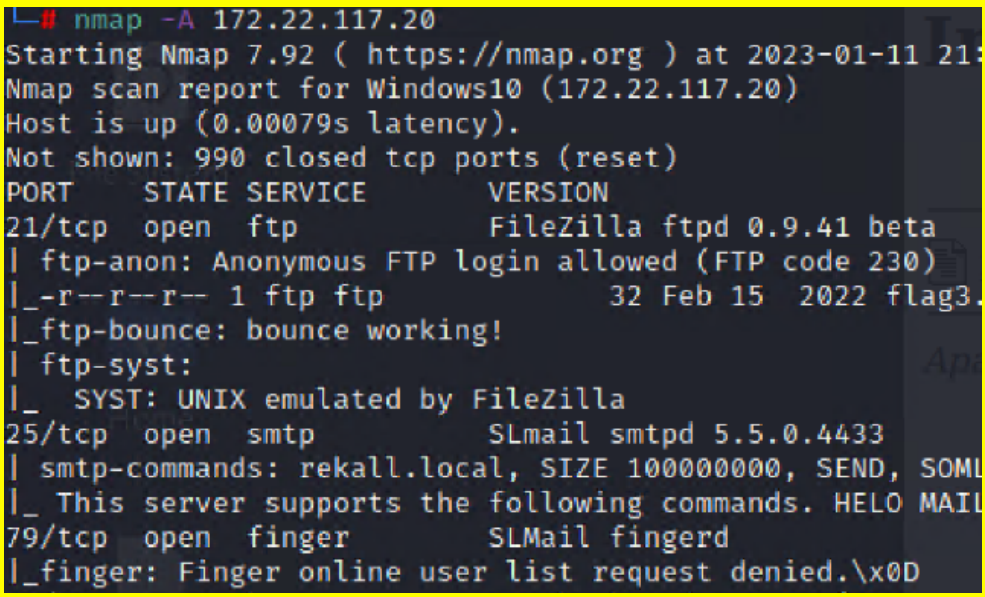
Vulnerability 17	Findings
Title	Exposed Sensitive Data

Type (Web app / Linux OS / WIndows OS)	Windows
Risk Rating	critical
Description	Found a github repository containing HTML code and login credentials.
Images	
Affected Hosts	Github totalrekall.xyz
Remediation	As stated before Pinging can be exploited to flood a server. Use a firewall implement rate limiting: only allowing a certain amount of pings to be sent from a specific IP/network.

Vulnerability 18	Findings
Title	Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Windows

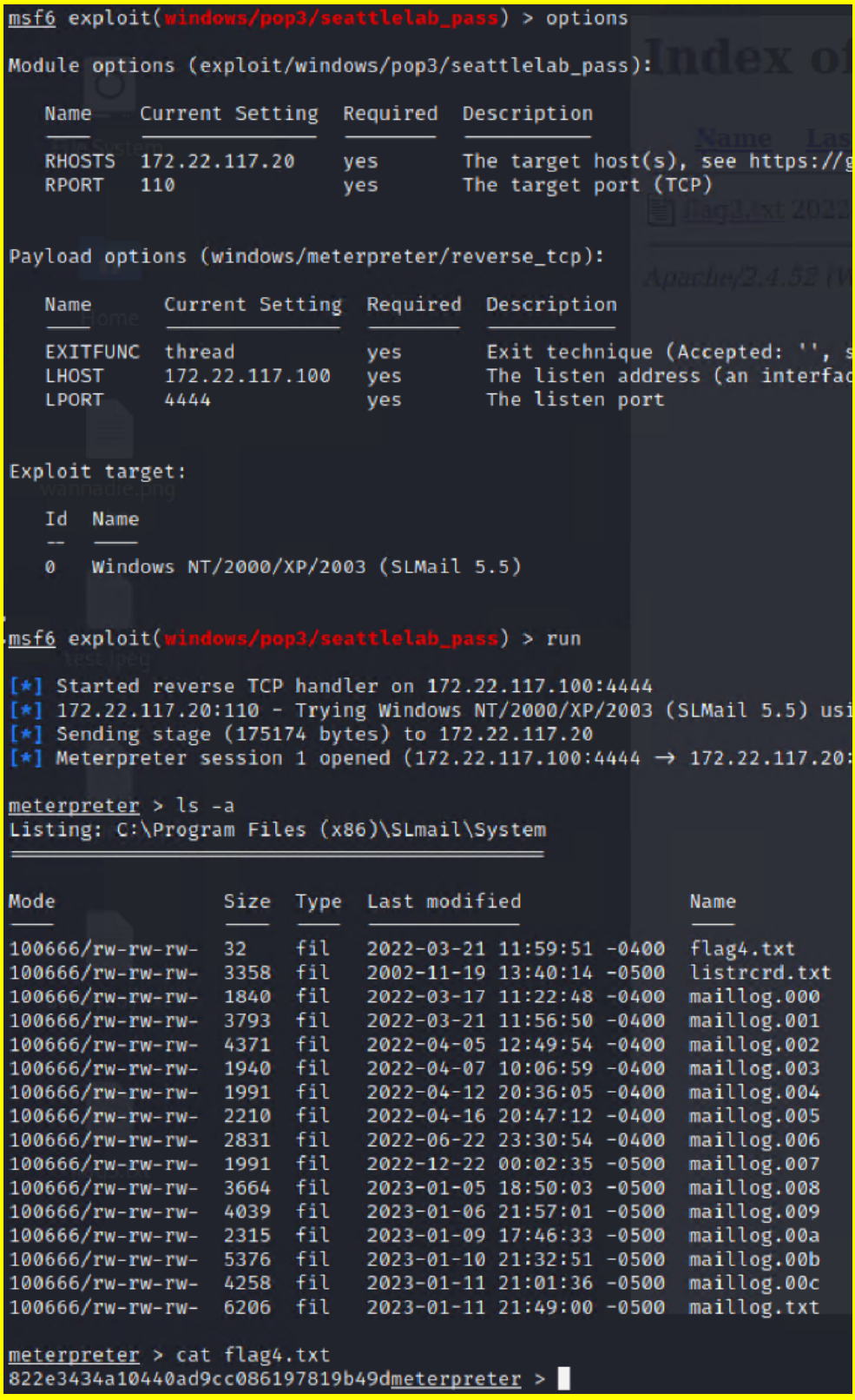
Risk Rating	Critical
Description	Using credentials found in the Github repository to log in and get information.
Images	 
Affected Hosts	172.22.117.0/24 172.22.117.10 172.22.117.20
Remediation	Do not post sensitive information on websites that are open source.

Vulnerability 19	Findings
Title	FTP port 21
Type (Web app / Linux OS / WIndows OS)	Windows
Risk Rating	Critical
Description	FTP port 21 anonymous login enabled

Images	 
Affected Hosts	172.22.117.20
Remediation	Port 21 being open allows easy access. Close 21. Use port 22. Only use FTP when urgently necessary. Disable anonymous login

Vulnerability 20	Findings

Title	Vulnerable port 110
Type (Web app / Linux OS / WIndows OS)	Windows
Risk Rating	Critical
Description	Vulnerable port 110 pop3

Images	
Affected Hosts	172.22.117.20
Remediation	Although POP3 is a protocol that uses clear text, it has the capability to be converted into an encrypted connection by using TLS/SSL. When compared to IMAP, POP3 is considered to be superior as IMAP stores messages, whereas POP3 does not.