Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Student Note: Complete all sections highlighted in yellow.

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

Contact Information

Company Name	CR4CK'D
Contact Name	Jackson Long
Contact Title	Jr. Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	3/12/2023	Jackson Long	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.

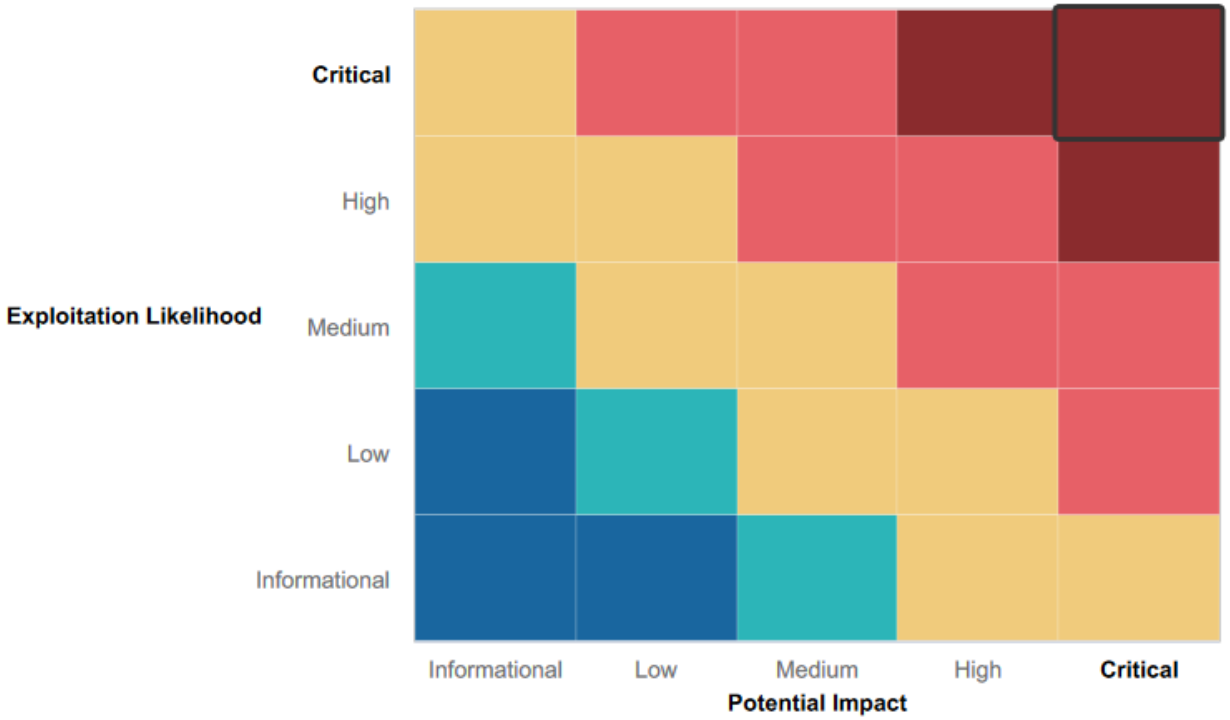
Executive Summary of Findings

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

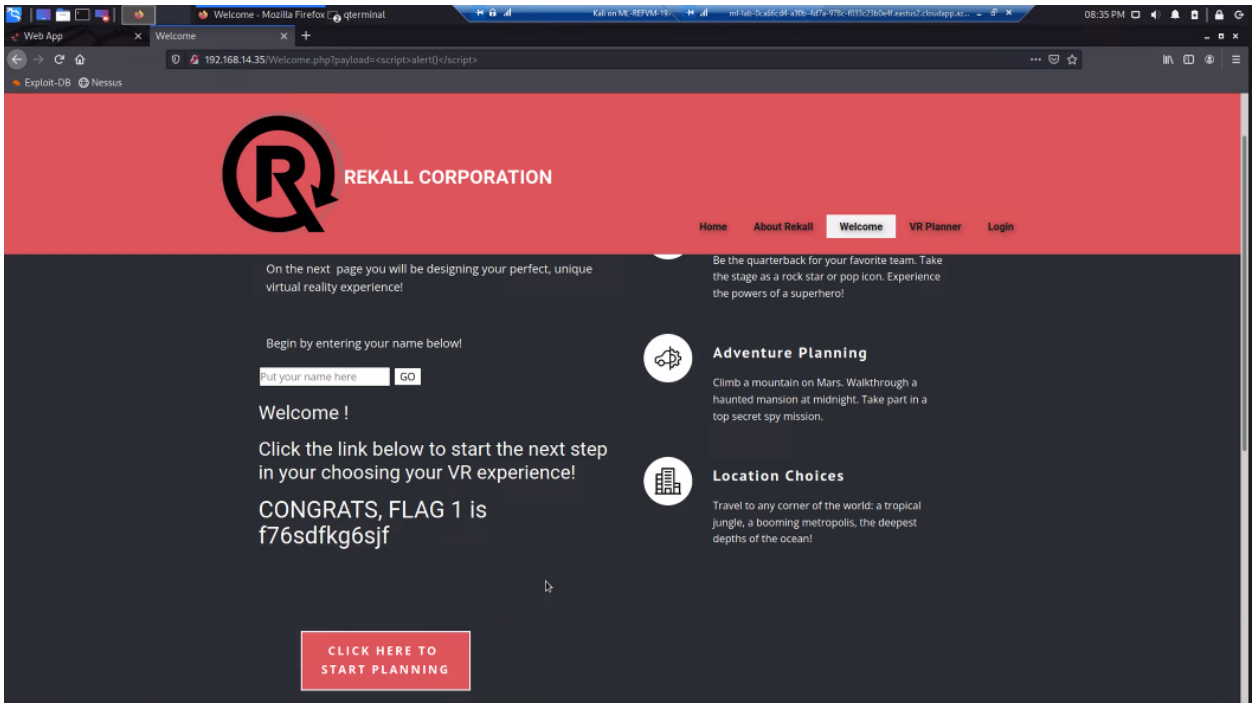
We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

- 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.

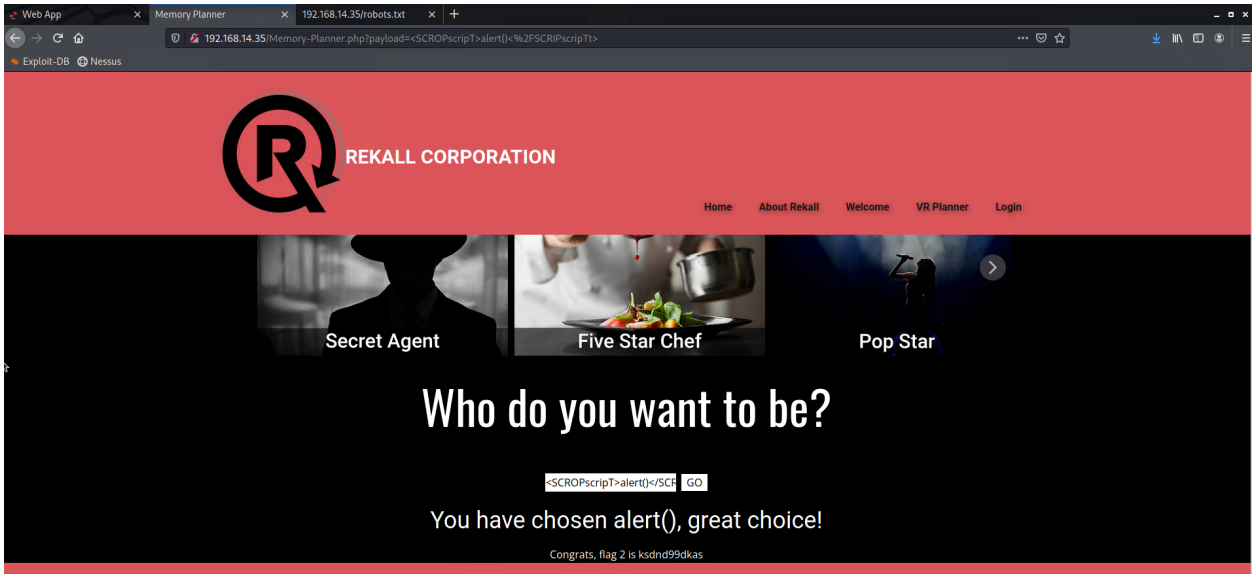
Executive Summary

Beginning stages of our penetration test were targeting/exploiting Rekall's Web-Application. Our first attack (C4) 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; CR4CK'D was able to store scripts and allow users to pull data.

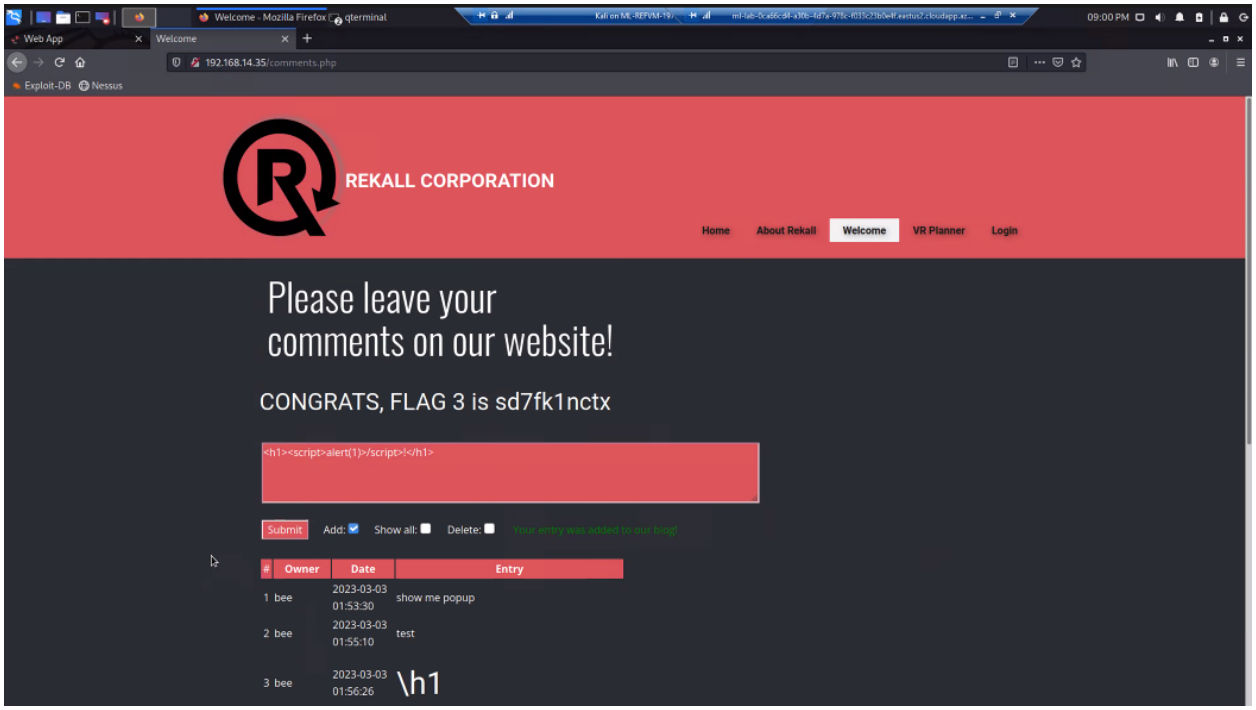
Using javascript CR4CK'D was able to successfully create an alert on Rekall Corps 'Welcome Page' <script>alert(1)</script> was run to achieve this.



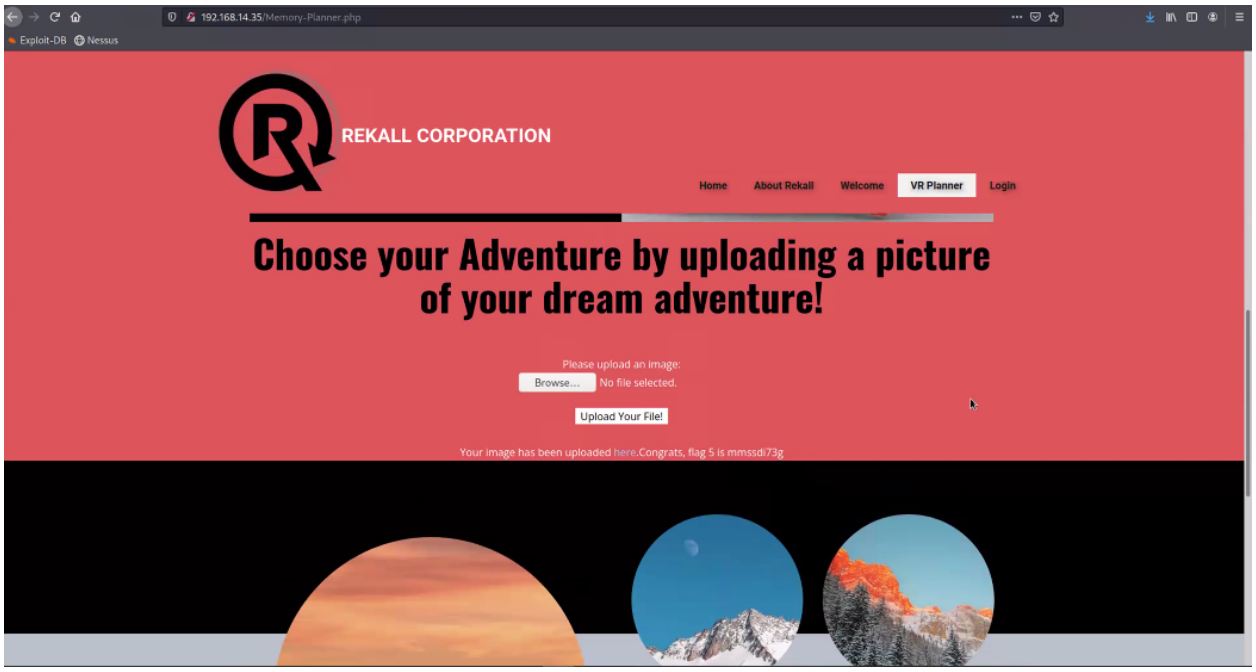
Reflected XSS

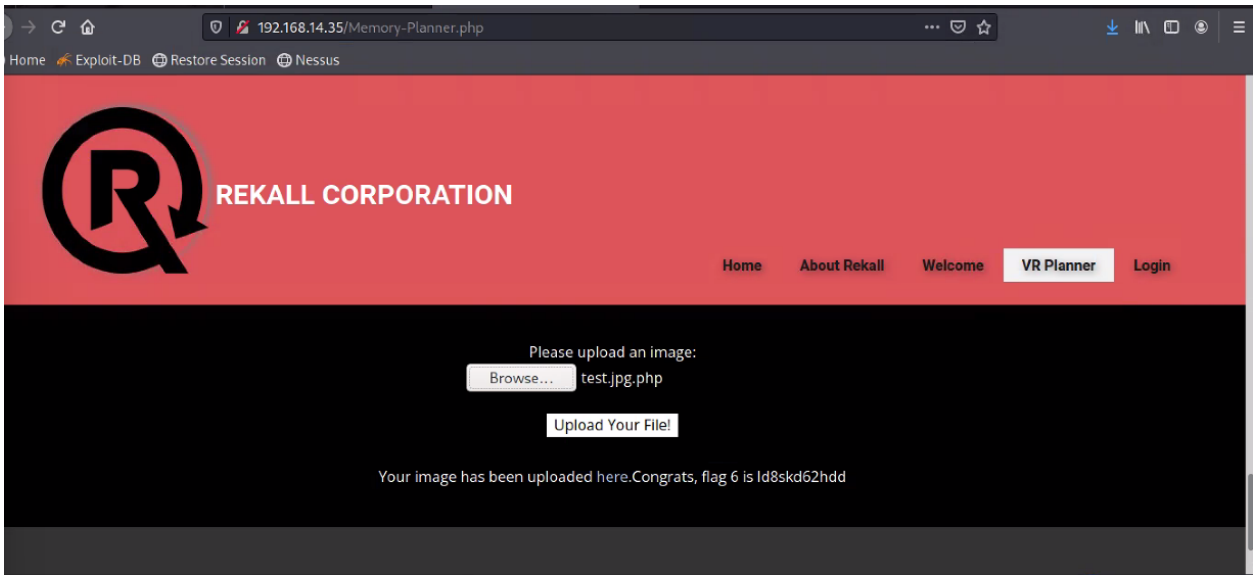


Stored XSS Input



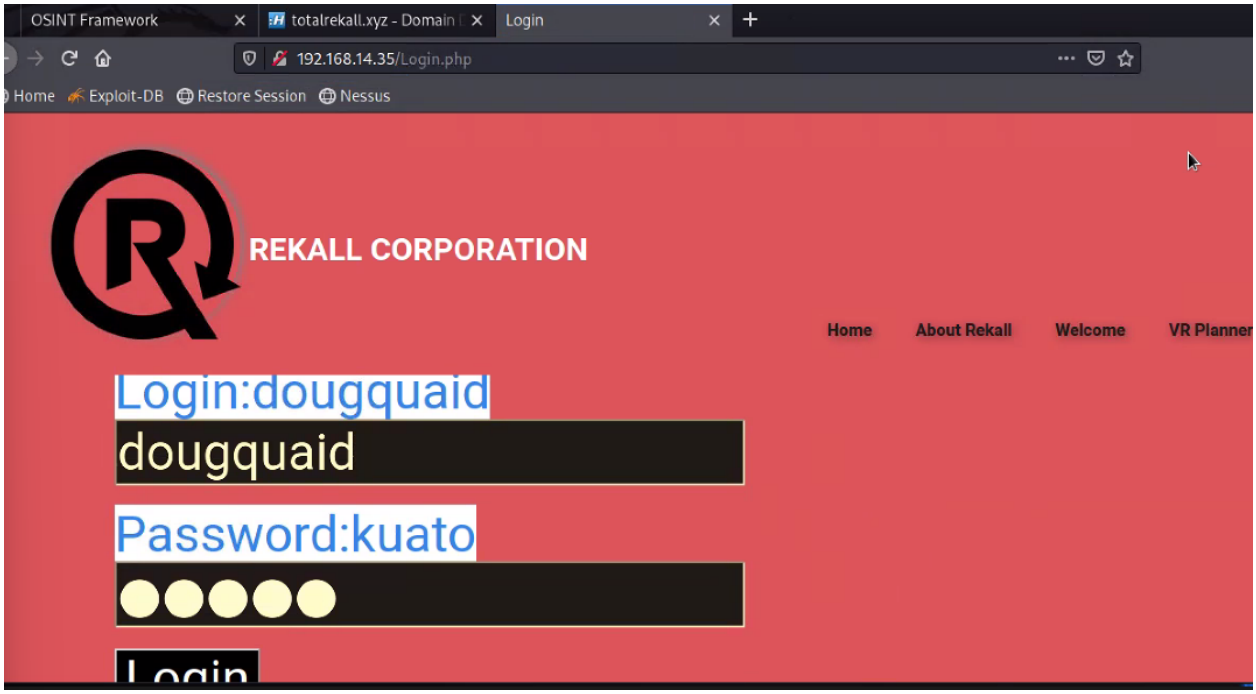
Using test.jpg.php in both upload fields CR4CK'D was able to bypass user input whitelist



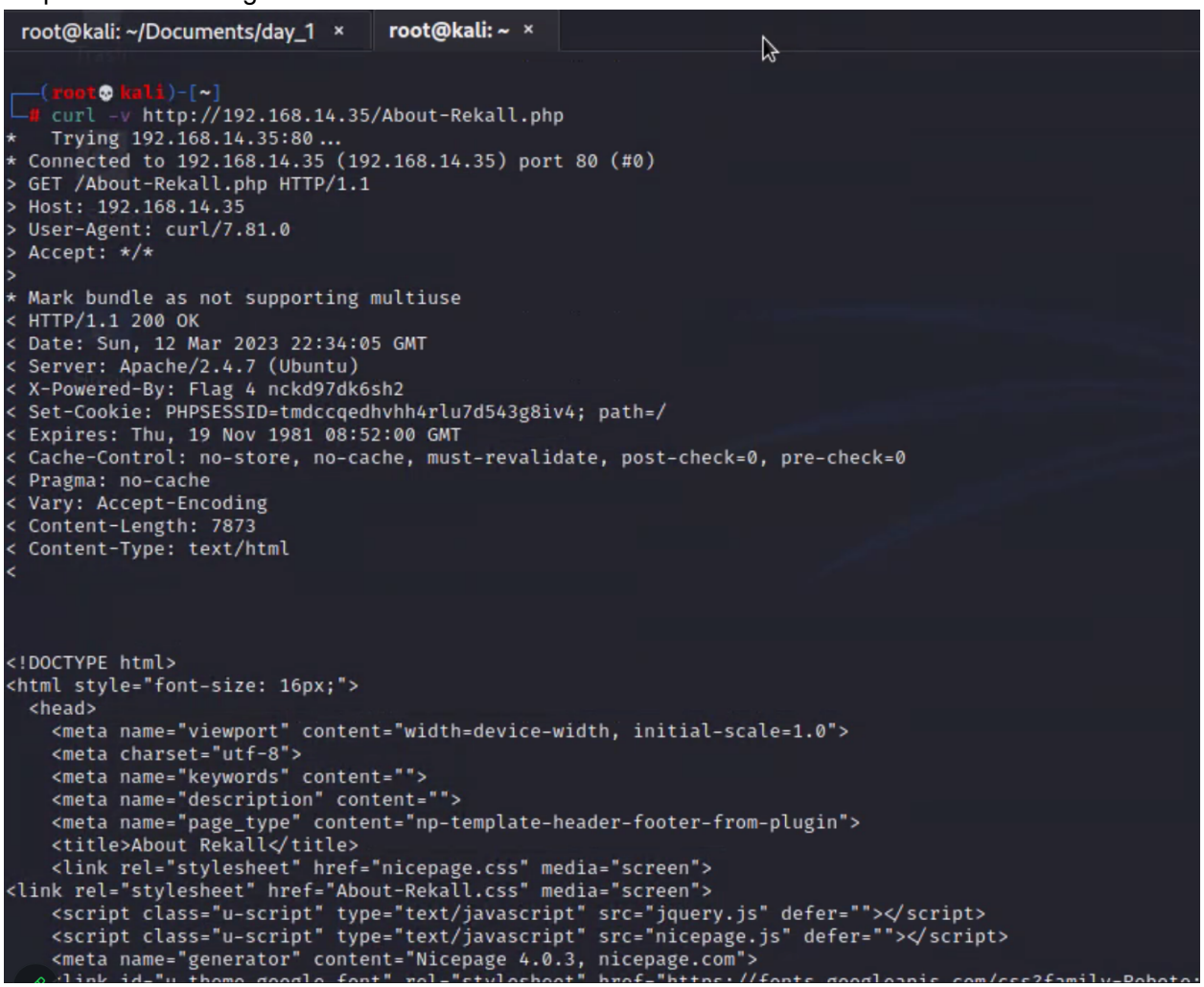


Using SQL Injection on the Login page, CR4CK'D was able to access a successful login by using (any password or '1'='1)

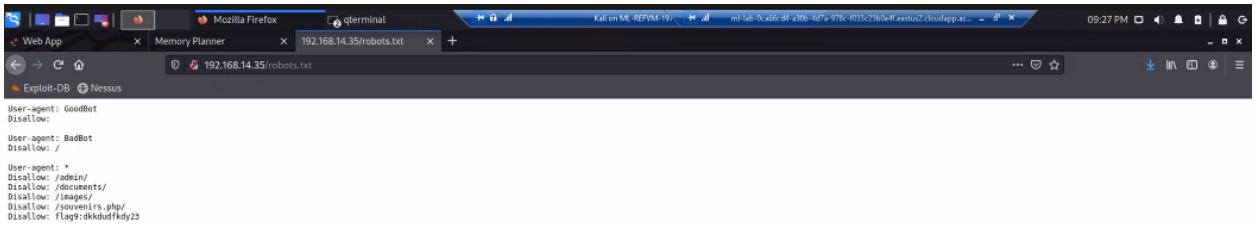
By searching the HTML, C4 was able to find credentials to log into an admin page



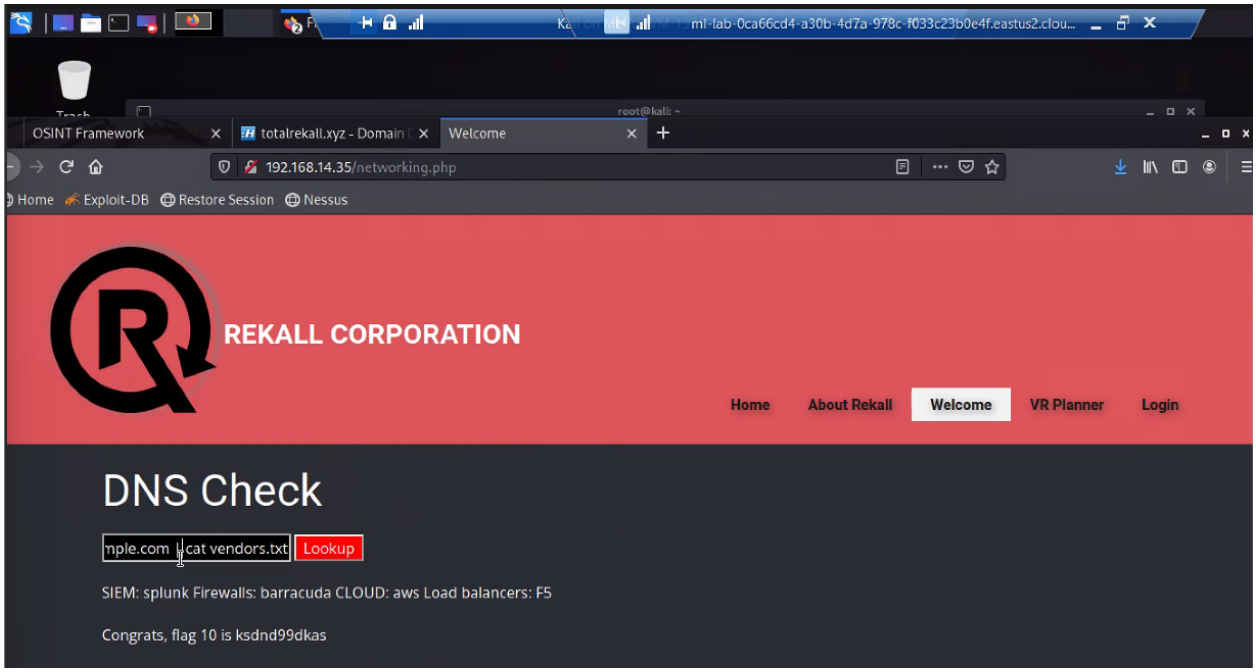
Output from CURLing



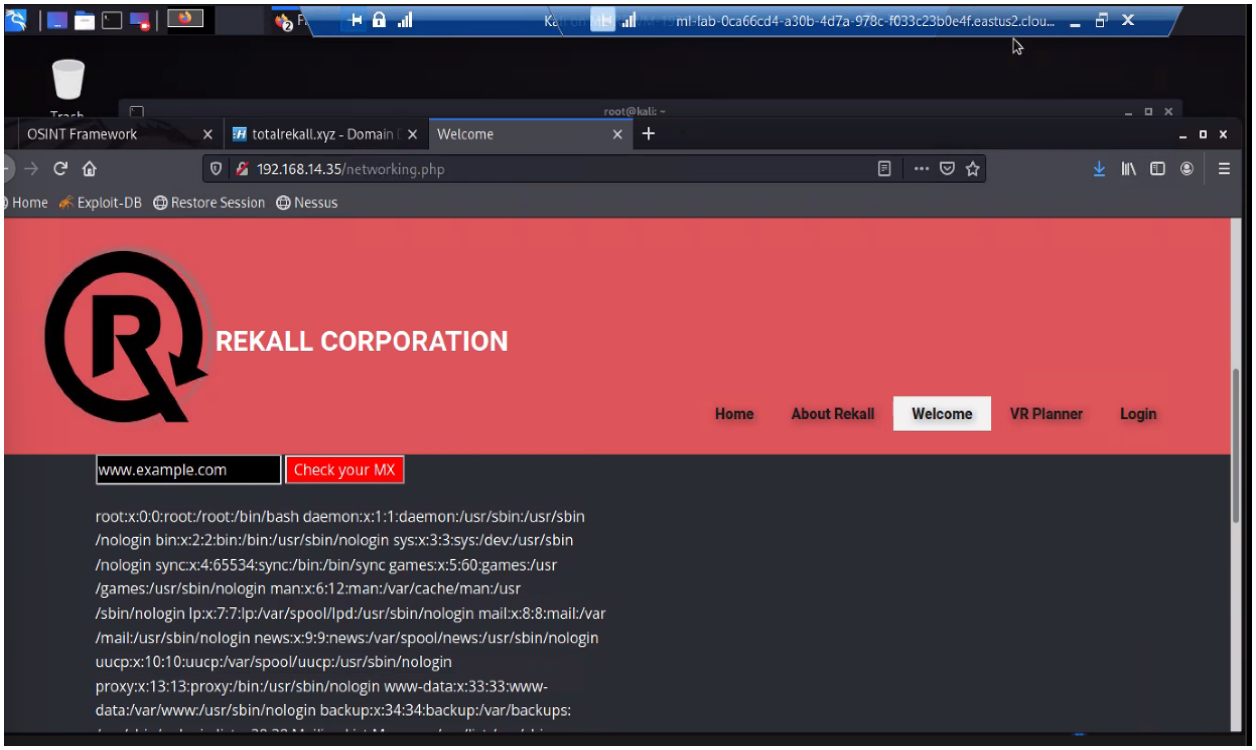
Found the robots.txt by adding /robots.txt at the end of the URL



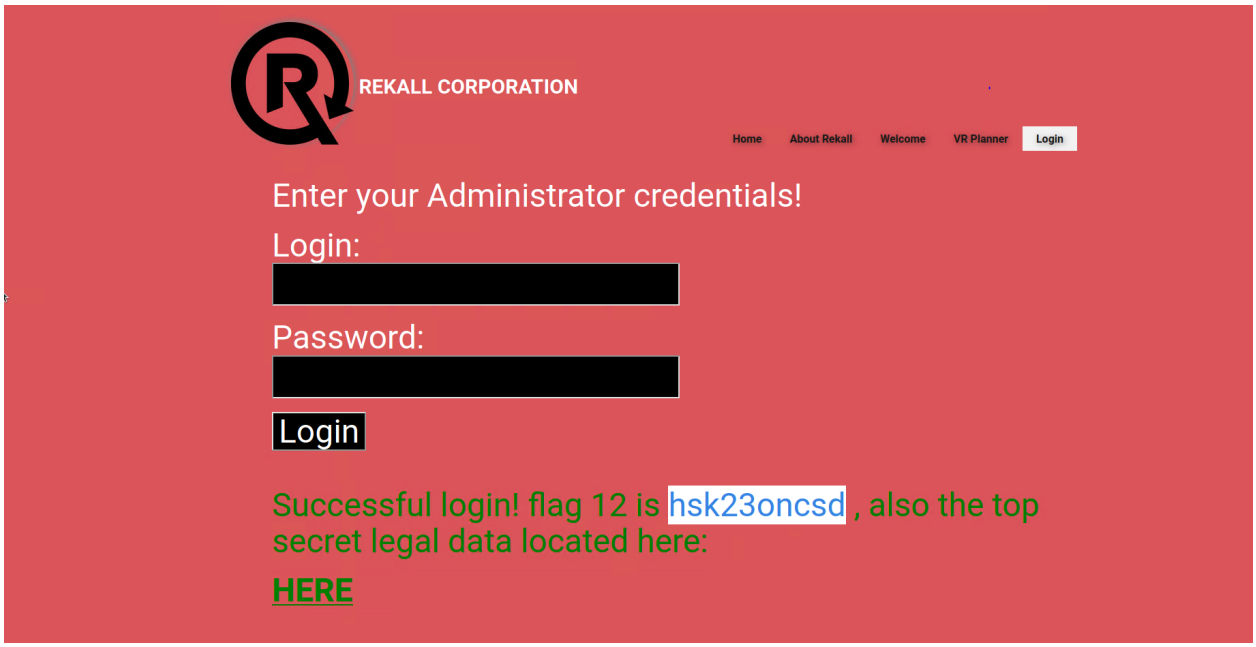
Stored XSS to gain access to company information



MX-Record Checker field input to search the /etc/passwd

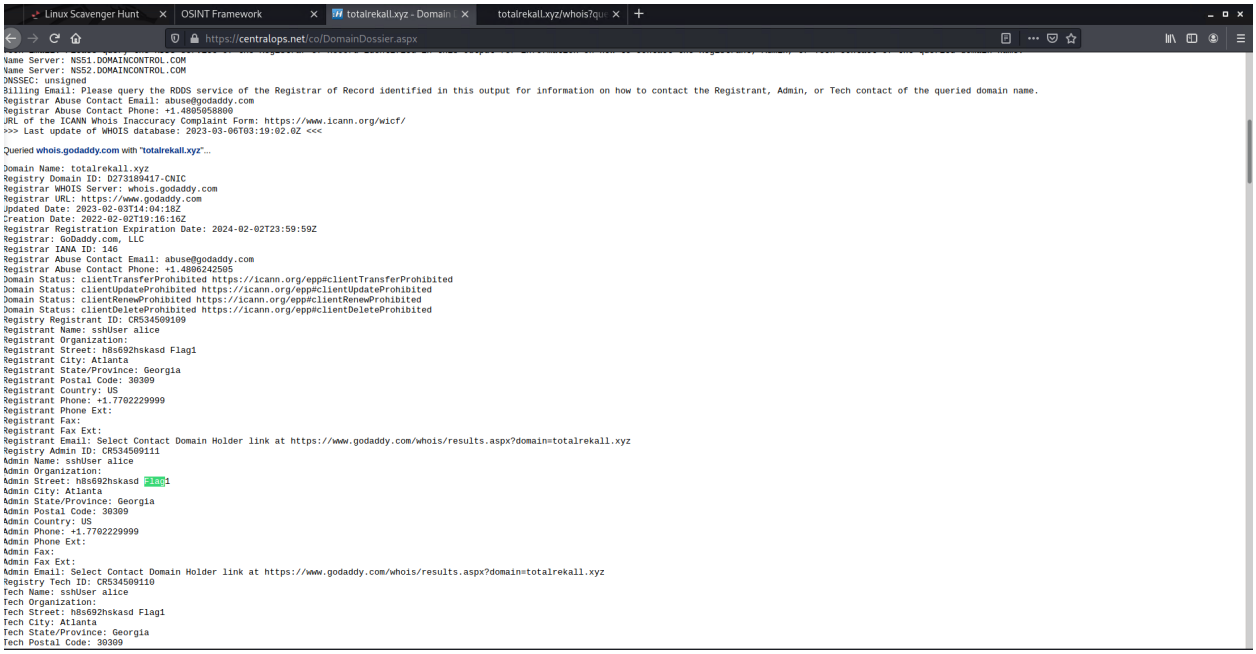


By searching the etc/passwd, we were able to find a user “melina” and bruteforce her password which was also “Melina”

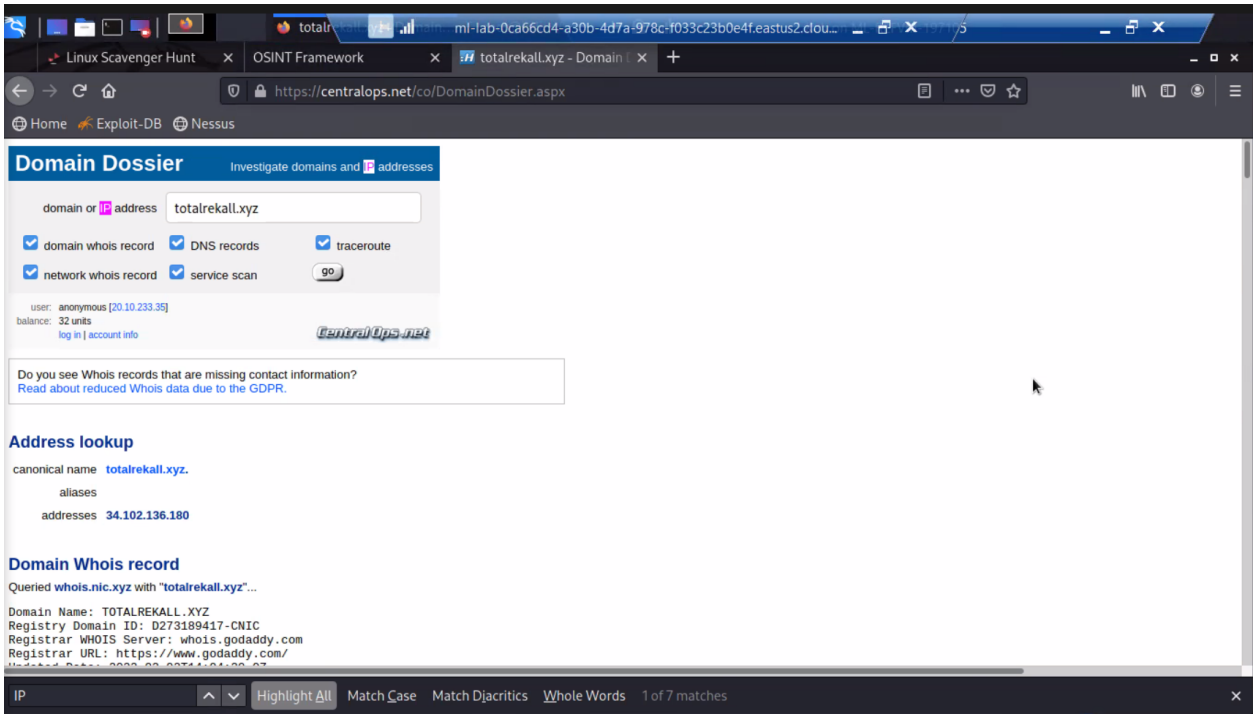


Executive Summary Day Two

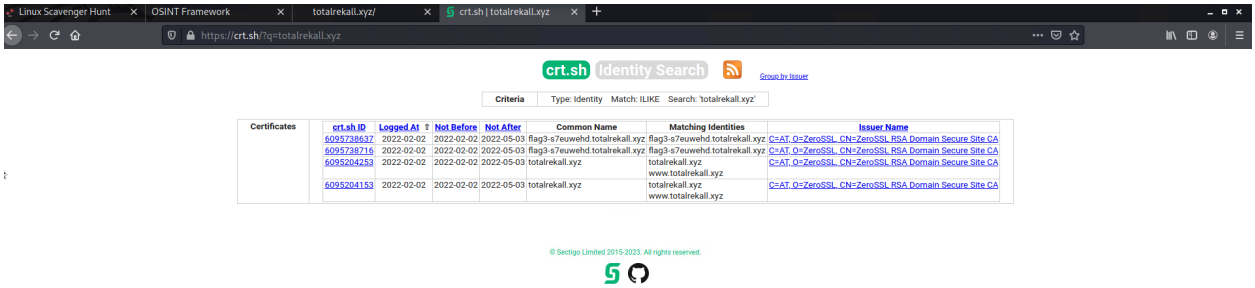
By accessing the OSNIT Framework, C4 was able to pull sensitive information from the database.



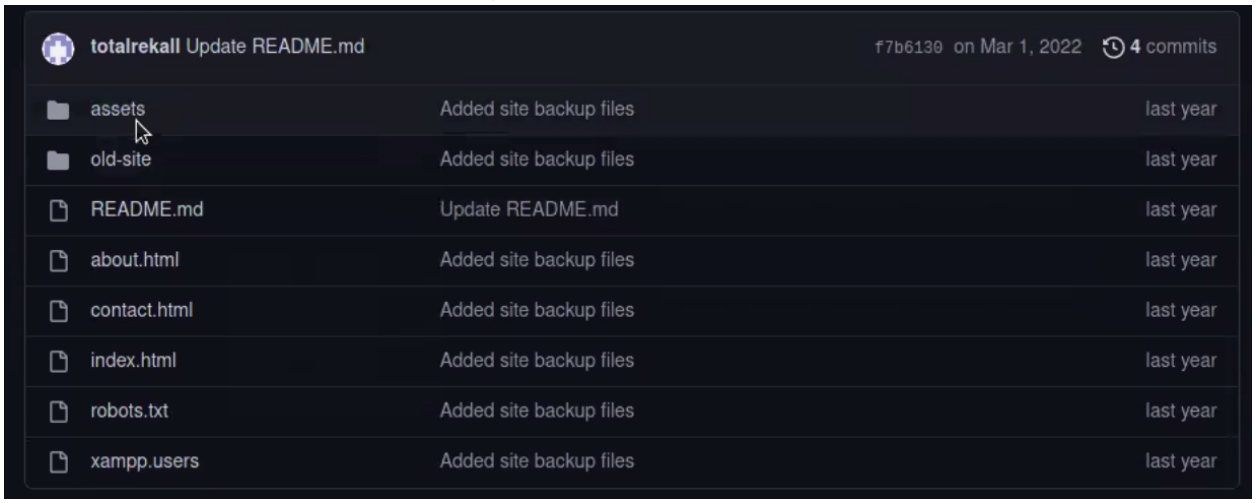
We were also able to gain an Address Lookup for IPs and more

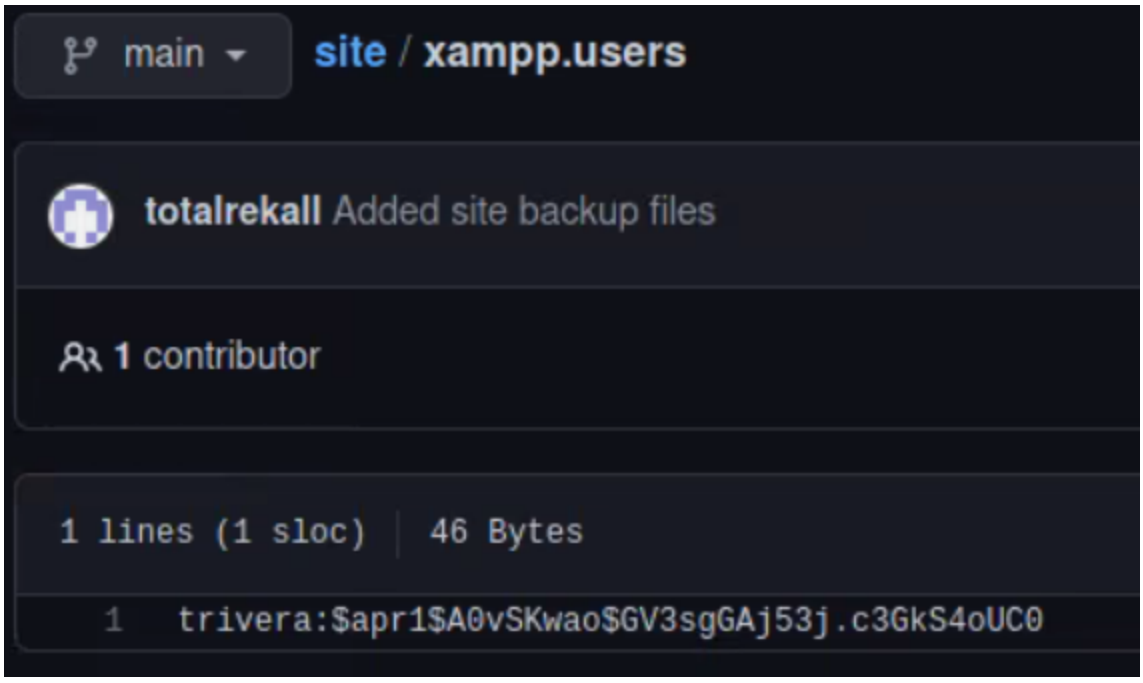


By searching on crt.sh, we were able to pull the certificate information for the website totalrekall.xyz



We were able to find a GitHub repository that contained sensitive information and HTML code





21.

Summary Vulnerability Overview

Vulnerability	Severity
Web App- Reflective XSS	Medium
Web App- Stored XSS	Medium
Web App- Stored XSS	Medium
Web App- Exposed Sensitive Data	Critical
Web App- Local File Inclusion	High
Web App- Local File Inclusion	High
Web App- Exposed Sensitive Data	Critical
Web App- Exposed Sensitive Data	Critical
Web App- Exposed Sensitive Data	Critical
Web App- Command Injection	Critical
Web App- Command Injection	Critical
Web App- Brute Force Attack	Critical
Linux- Open Source Vulnerability	Critical
Linux- Domain Ping	Low
Linux- Open Source Vulnerability	Low
Linux- Network Mapping Scan	Medium
Windows- Exposed Sensitive Data	Critical
Windows- Password Guess	Critical
Windows- Vulnerability FTP port 21	Critical
Windows- Vulnerability port 110	Critical

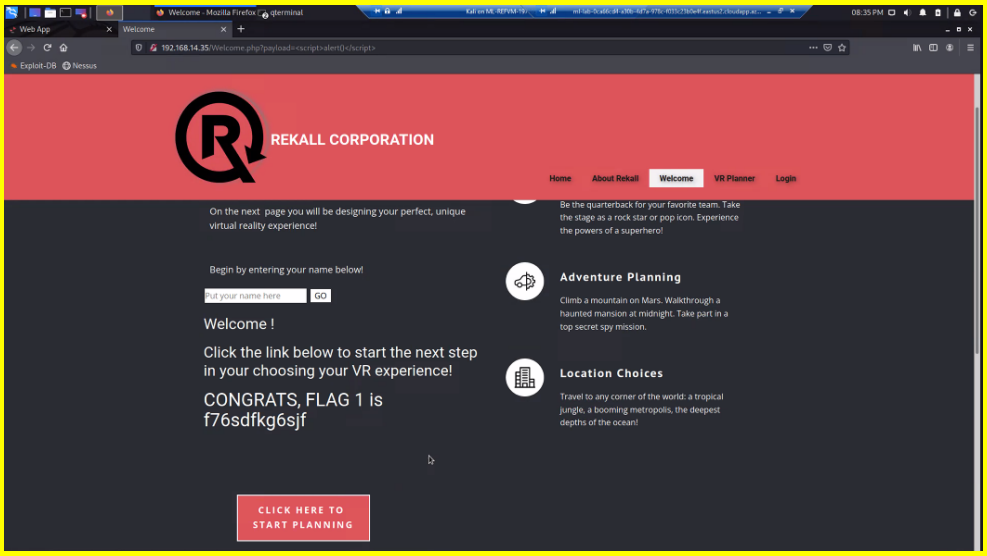
The following summary tables represent an overview of the assessment findings for this penetration test:

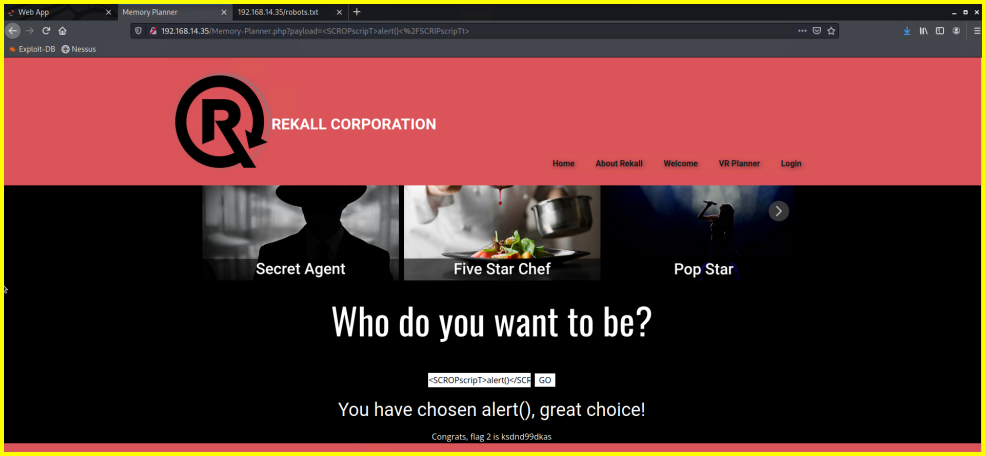
Scan Type	Total
Hosts	172.22.117.20 192.168.13.10 192.168.13.11 192.168.13.12 192.168.13.14 192.168.13.1 192.168.14.34

Ports	80, 8080, 22, 5901, 6001, 10000, 10001, 3306, 53, 88, 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 21, 25, 106, 110, 443, 79

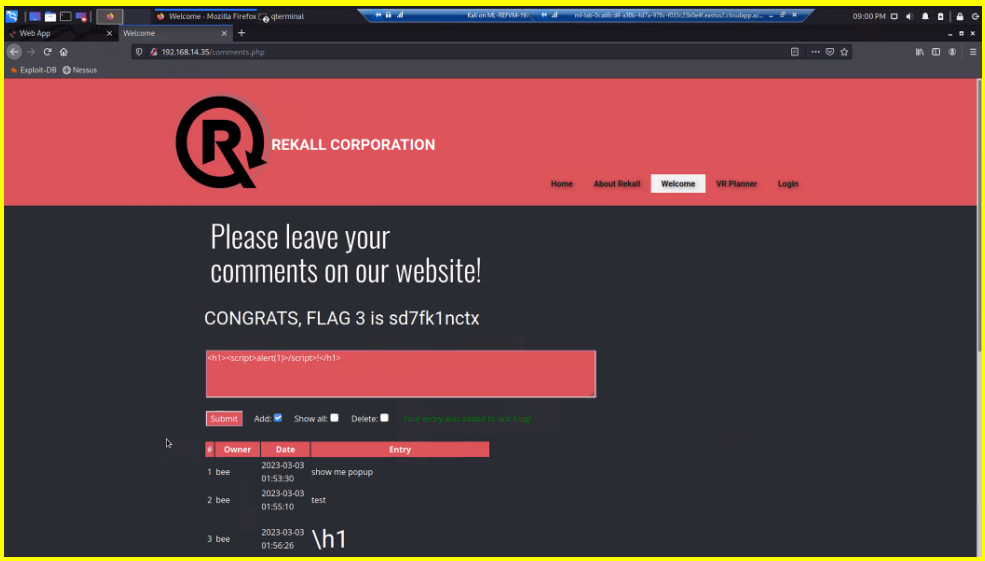
Exploitation Risk	Total
Critical	12
High	2
Medium	4
Low	2

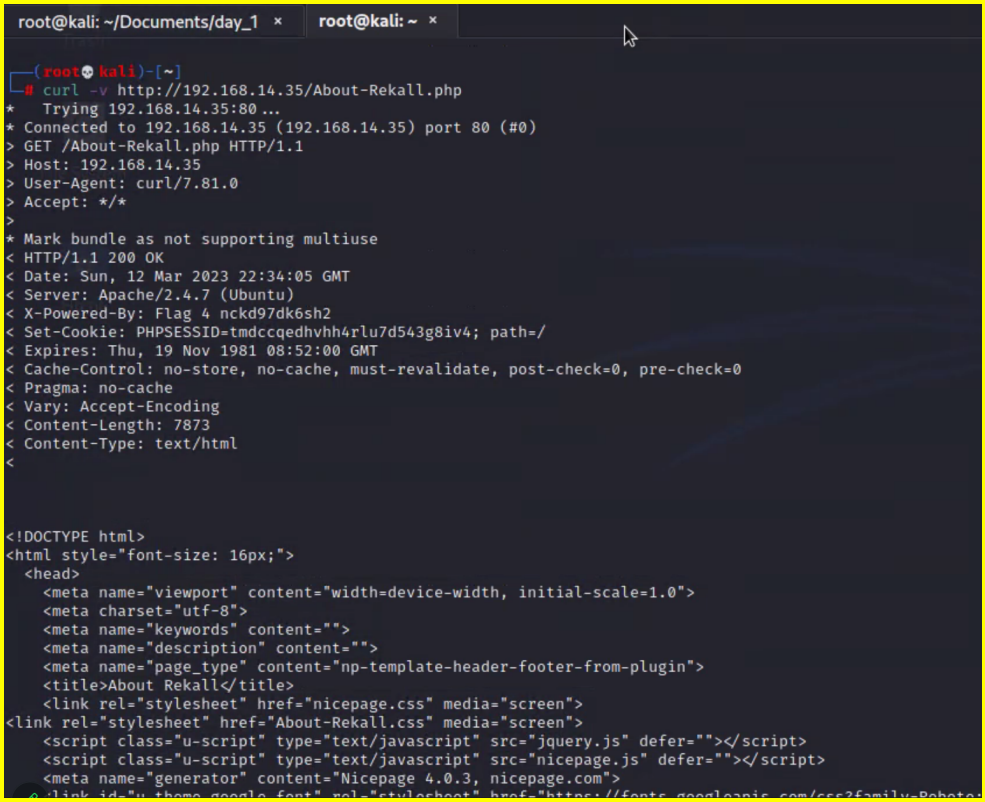
Vulnerability Findings

Vulnerability 1	Findings
Title	Reflected XSS
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	Cross-site scripting is when an attacker injects malicious code into a website that is then executed on a victim's browser. <script>alert(1)</script> was the script used.
Images	
Affected Hosts	192.168.14.35
Remediation	User input sanitizing

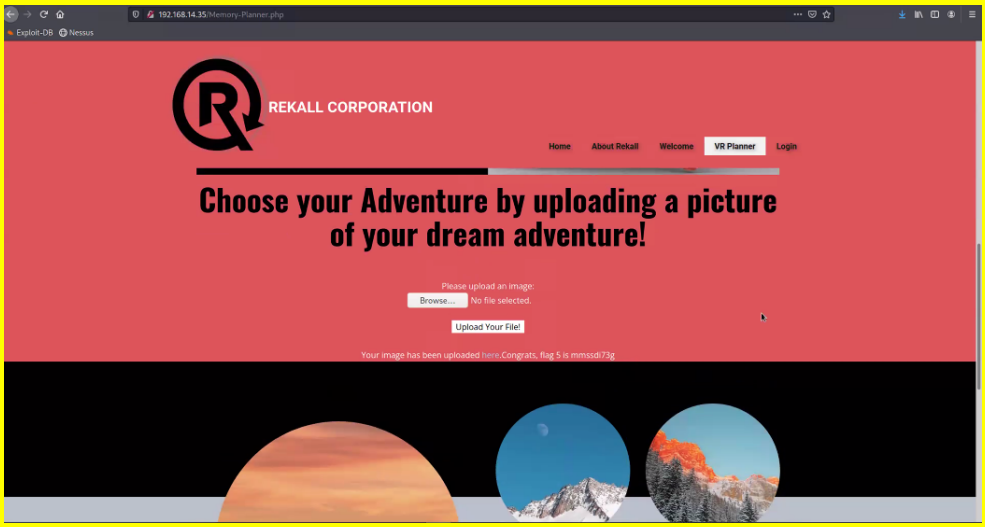
Vulnerability 2	Findings
Title	Reflective XSS
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	Most commonly found through links, this attack occurs when a script is ‘reflected’ off a web app to a victim's browser
Images	
Affected Hosts	192.168.14.35
Remediation	Reflective XSS is difficult to remedy fully, but you can protect yourself similarly to stored XSS by sanitizing against impactful inputs

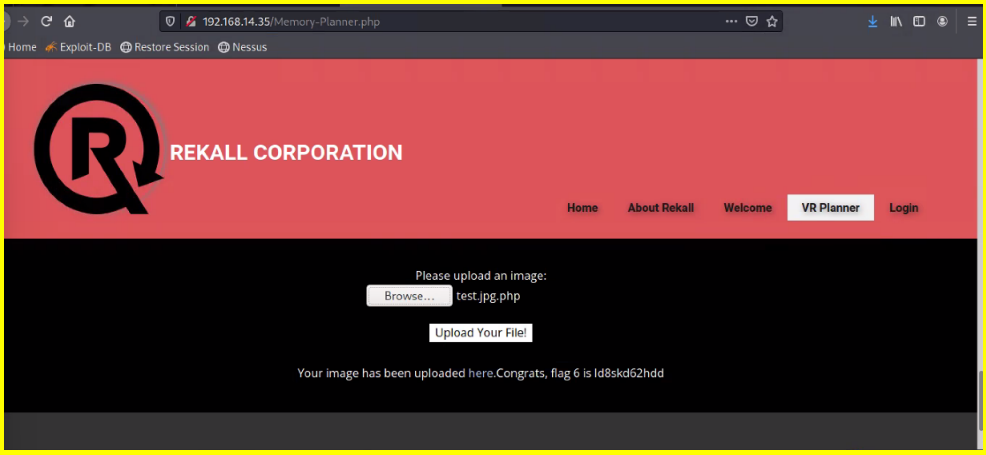
Vulnerability 3	Findings
Title	Stored XSS
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	Stored XSS is potentially more dangerous than Reflective because the inputs are stored in the targets server

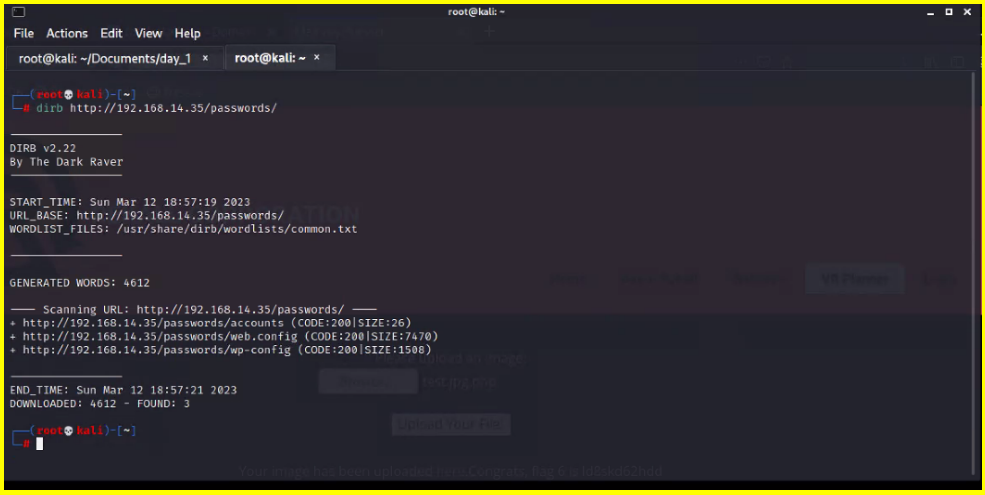
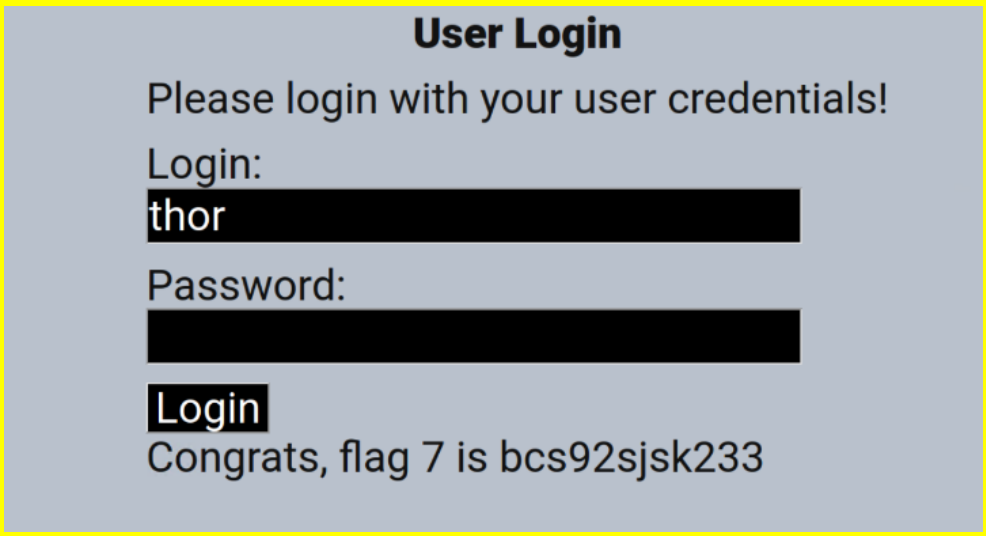
Images	
Affected Hosts	192.168.14.35
Remediation	WAF can be used in a Stored XSS attack because it can filter traffic in the app.

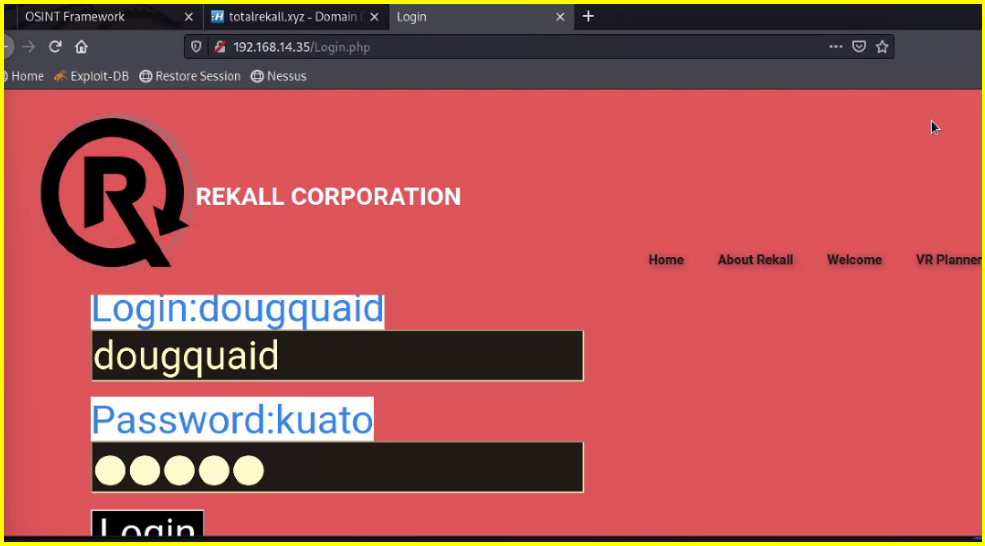
Vulnerability 4	Findings
Title	Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Using cURL, we were able to find sensitive data in plaintext format
Images	
Affected Hosts	192.168.14.35
Remediation	Encrypt sensitive data such as hashes, keys, and any other data that should not be viewed by those without credentials for viewing.

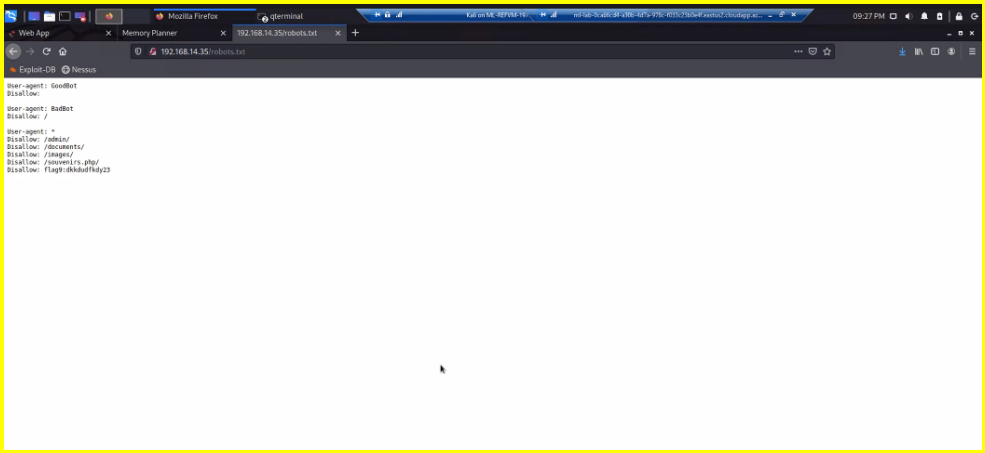
Vulnerability 5	Findings
Title	Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	An attack includes a path or hidden value in an input field. We were able to upload a malicious script by adding .jpg to make it appear as an image.

Images	
Affected Hosts	192.168.14.35
Remediation	Only allow images to be submitted to the field.

Vulnerability 6	Findings
Title	Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Similar to before, but because it was a .php page, we simply bypassed the parameter by including .php in the file upload.
Images	
Affected Hosts	192.168.14.35
Remediation	Strengthen the parameters to only allow certain files to be uploaded

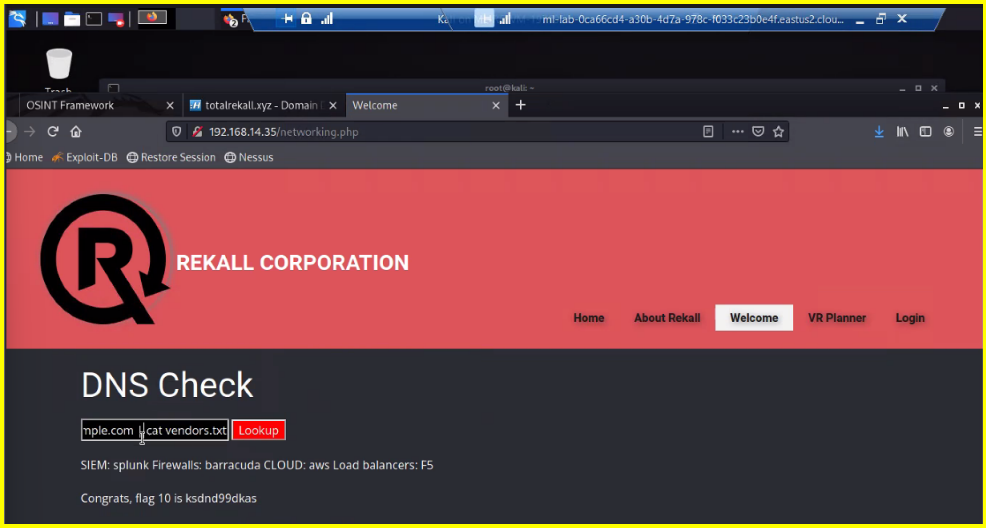
Vulnerability 7	Findings
Title	Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	We were able to use Dirb and locate a passwords directory. After cURLing, we found login credentials for Thor and were able to login.
Images	 
Affected Hosts	192.168.14.35
Remediation	A web application firewall can be used to keep the HTML secure

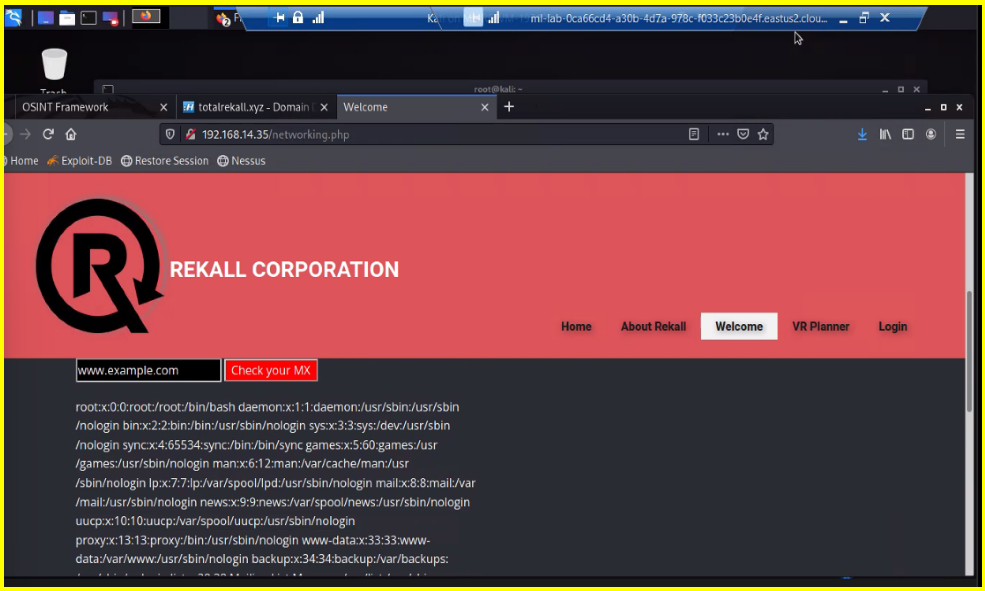
Vulnerability 8	Findings
Title	Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Password/Username hidden in the HTML lines
Images	
Affected Hosts	192.168.14.35
Remediation	Keep user and password credentials stored in a secure encrypted location.

Vulnerability 9	Findings
Title	Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Located the robots.txt directory on the web app by searching through the URL
Images	
Affected Hosts	192.168.14.35
Remediation	Refrain from using the robots.txt to hide information from search results

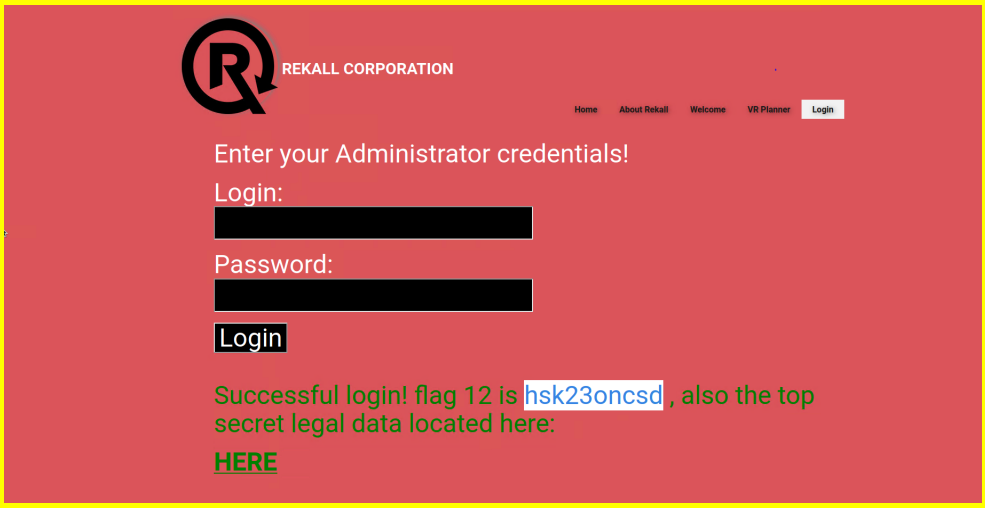
Vulnerability 10	Findings
Title	Command injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Using a SQL injection in the password field to make the password used ‘true’

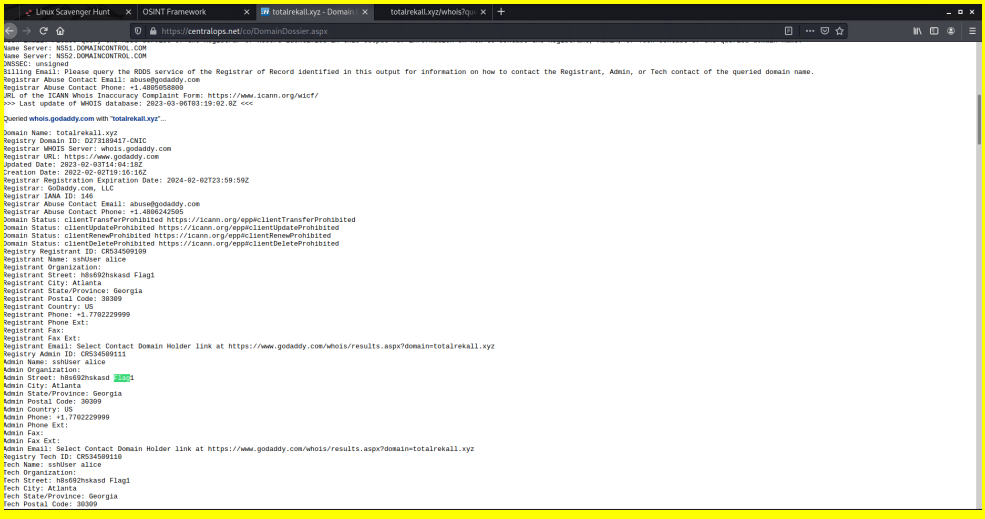
Images	
Affected Hosts	192.168.14.35
Remediation	Input validation/sanitization

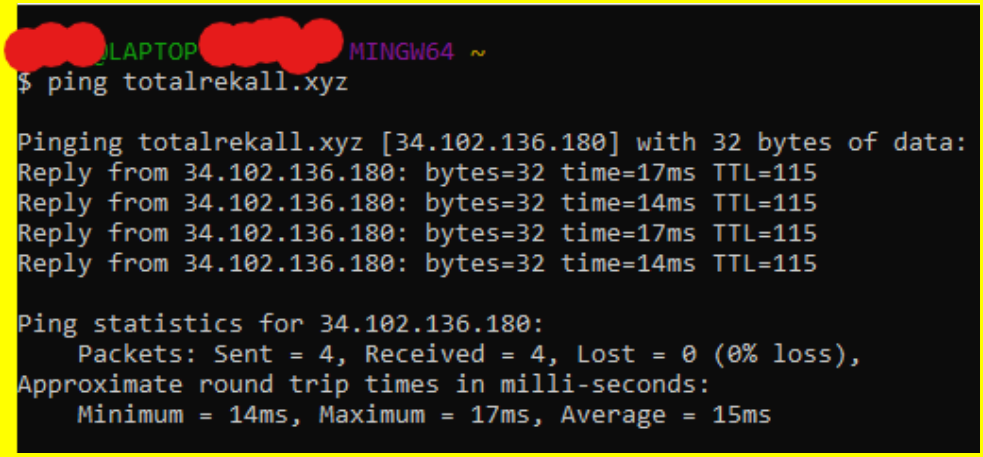
Vulnerability 11	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	In the DNS Check field, we were able to use simple Command injections to locate the secret vendors.txt file
Images	
Affected Hosts	192.168.14.35
Remediation	Sanitizing as always the input and parameterized queries as well as whitelisting and using a WAF.

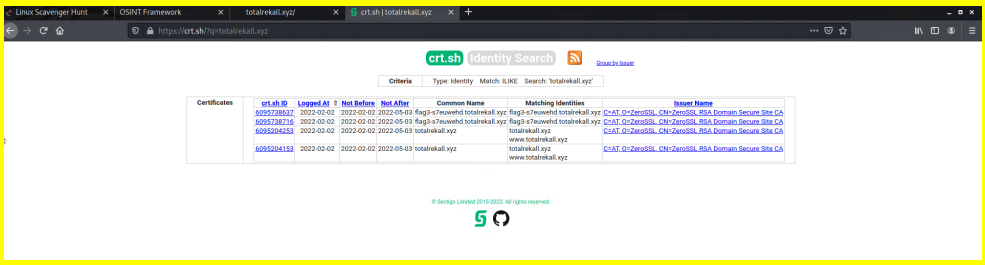
Vulnerability 12	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	In the MX field, we were able to do something similar by searching the /etc/passwd file by modifying the command injection
Images	
Affected Hosts	192.168.14.35
Remediation	Sanitizing as always the input and parameterized queries as well as whitelisting and using a WAF.

Vulnerability 13	Findings
Title	Brute Force Attack
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	By searching the /etc/passwd in the previous vulnerability, we were able to find a user ‘Melina’. By using brute force techniques, we found the password was also ‘Melina’.

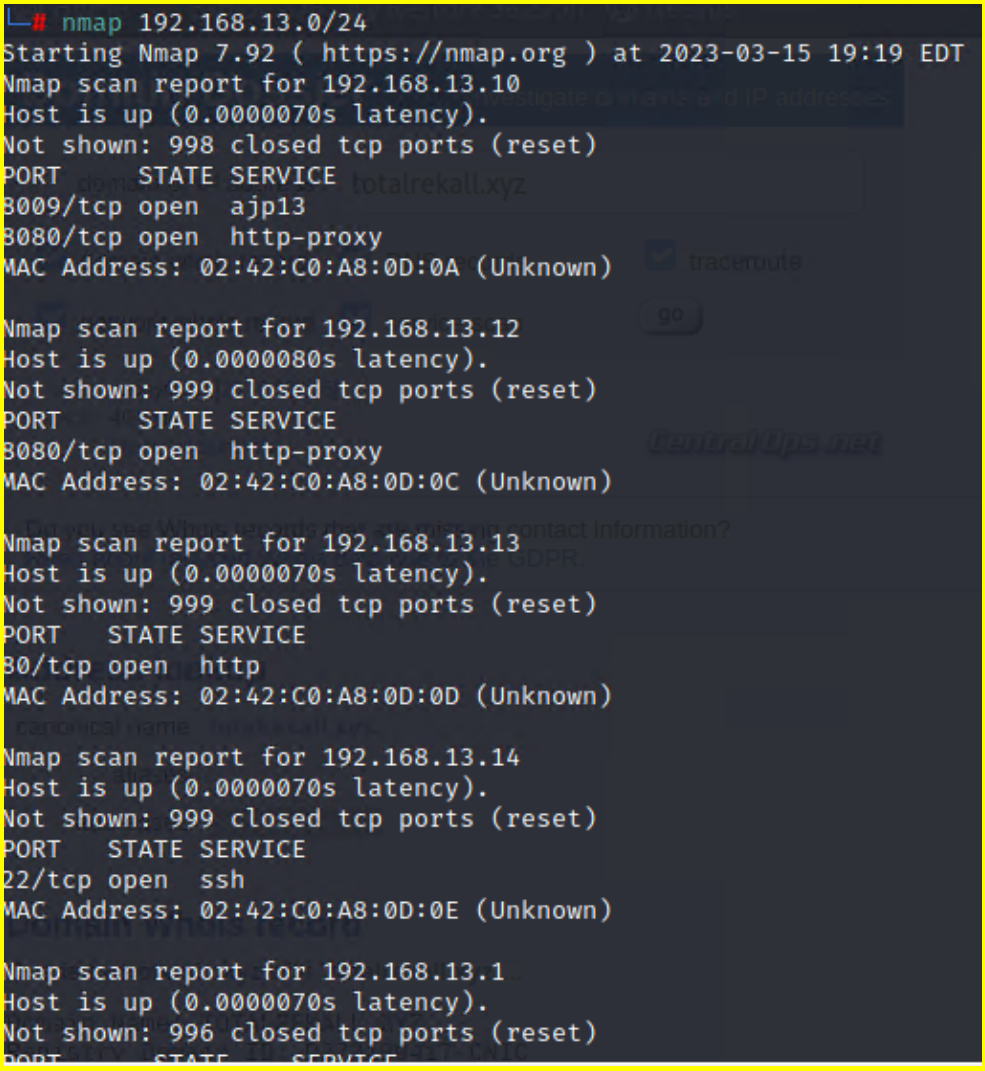
Images	
Affected Hosts	192.168.14.35
Remediation	Enforce stricter passwords in the company. Brute force attacks can take years at a time with stronger passwords with required special characters.

Vulnerability 14	Findings
Title	Open Source Vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	By searching OSNIT, we were able to pull sensitive ‘WHOIS’ company data
Images	
Affected Hosts	Totalrekall.xyz
Remediation	Important data or information should never be stored in plaintext, key encryptions can prevent this information from being leaked or obtained by attackers

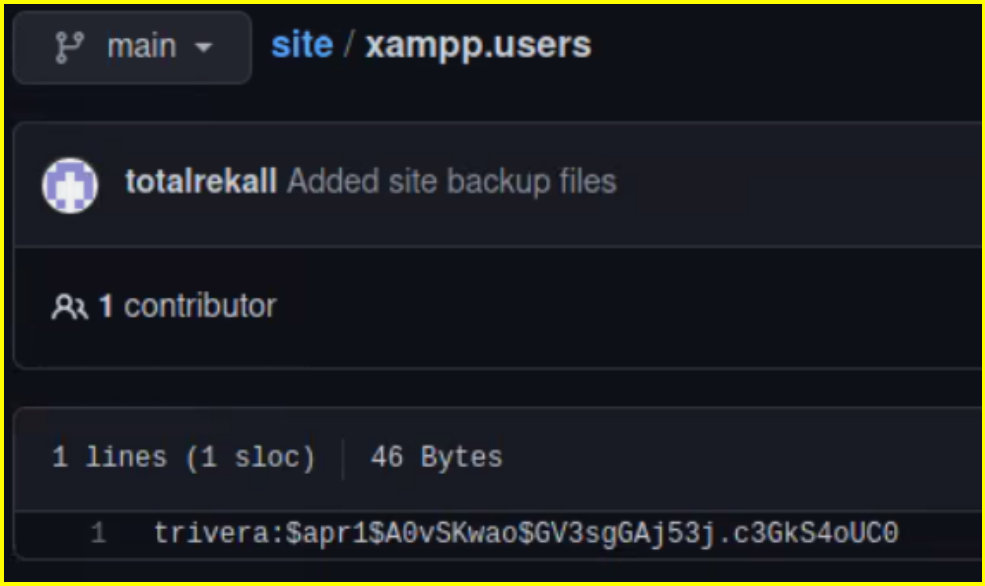
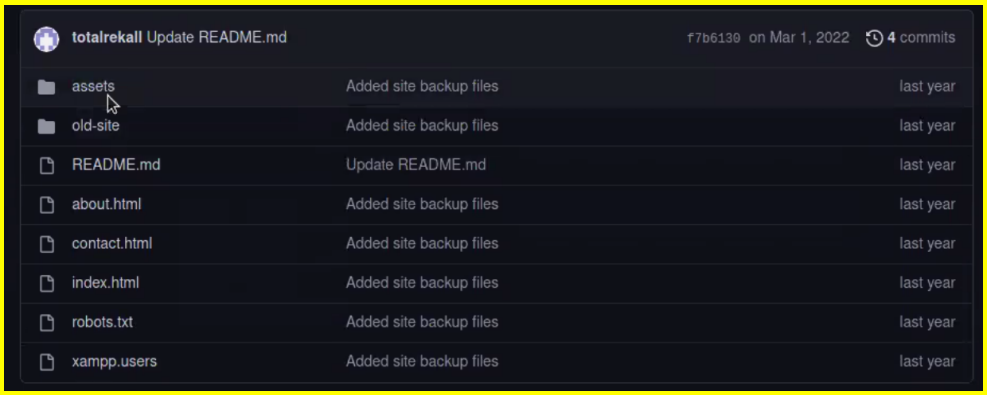
Vulnerability 15	Findings
Title	Domain Ping
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low
Description	Pinging isn't bad and is a useful tool to see if the server is responding, but can but used in a DDOS attack on the server as well
Images	
Affected Hosts	Totalrekall.xyz
Remediation	Having strong firewalls in place to help filter what data comes into the server can help prevent a DOS attack. Load balancers also help strengthen the server.

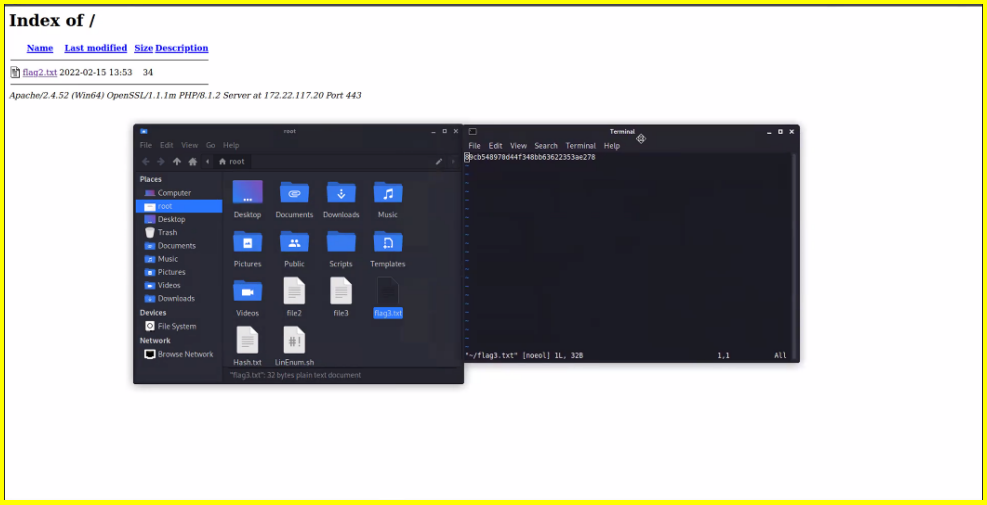
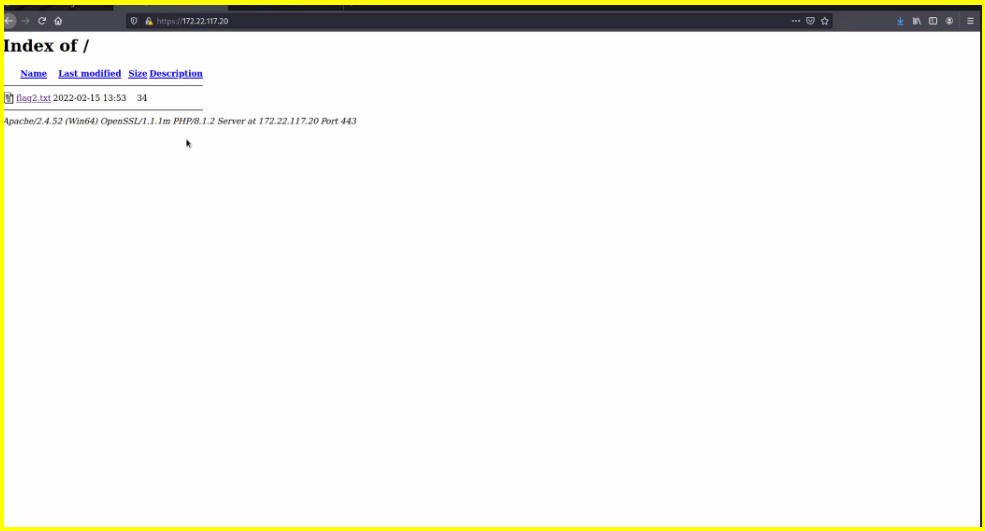
Vulnerability 16	Findings
Title	Open Source Vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low
Description	Searched on Cert.sh to pull the certificates for totalrekall.xyz
Images	

Affected Hosts	Totalrekall.xyz
Remediation	Stronger firewalls and multiple encryption methods can be used to keep this open source information from harming the company.

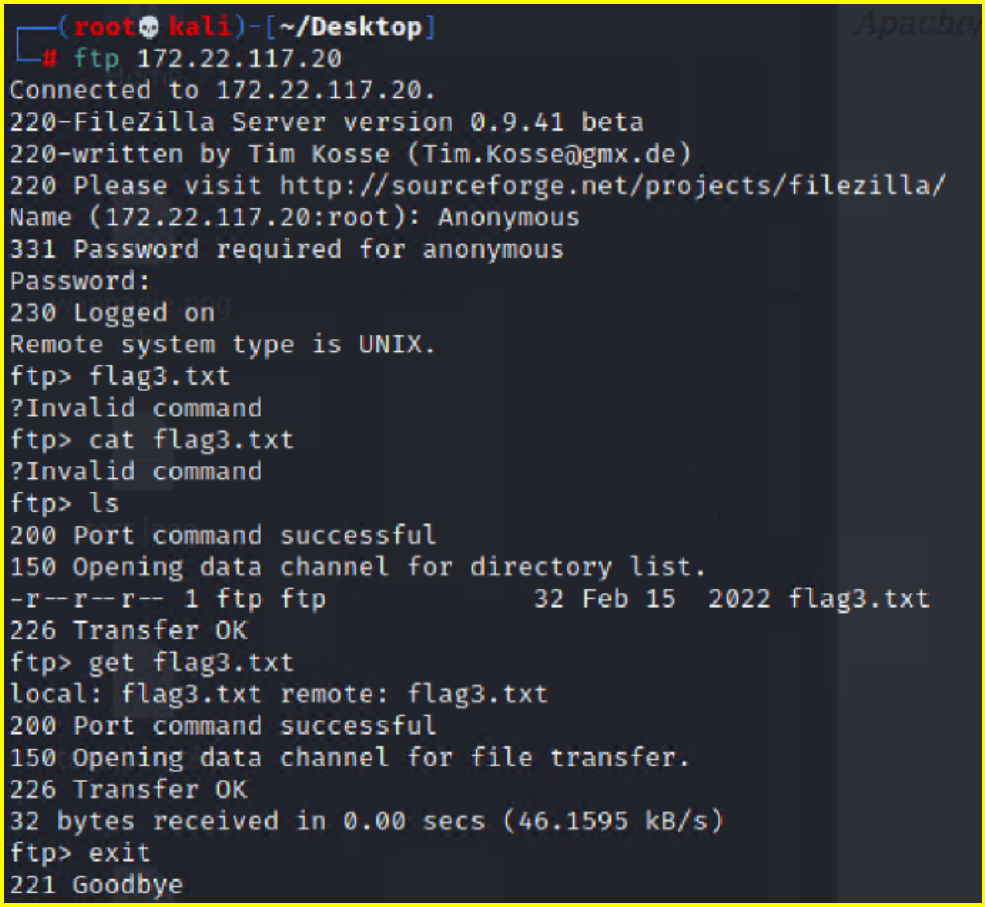
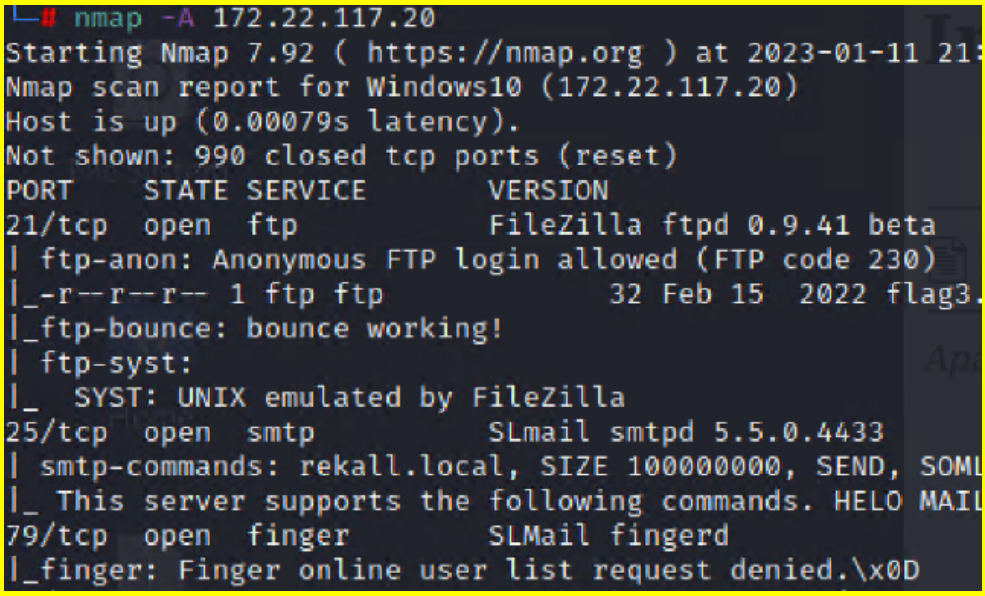
Vulnerability 17	**Findings**
Title	Network Maping
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Medium
Description	The scan itself isn't a vulnerability, however weaknesses can be found with these Nmap scans such as open hosts and ports and
Images	
Affected Hosts	192.168.13.0/24 192.168.13.10 192.168.13.11

	192.168.13.12 192.168.13.13 192.168.13.14 192.169.13.1
Remediation	Strong firewalls that can limit the rate at which a server can be pinged and making sure that exploitable ports are closed when not in use

Vulnerability 18	Findings
Title	Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	C4 found a Github repository containing HTML code and login credentials
Images	 
Affected Hosts	Totalrekall.xyz
Remediation	Having a stronger firewall implement rate limiting, only allowing X amount of pings to be sent.

Vulnerability 19	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 the server.
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 20	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	Close port 21. Only use FTP when urgently necessary. Disable anonymous login

Read Me:
This project was done in a group with John Wallace, and Tracy Skelton. We all contributed input on screenshots, remediation, and tactics.