| Cybersecurity |
| --- |
| Penetration Test Report |

Rekall Corporation

Penetration Test Report

**Student Note: Complete all sections highlighted in yellow.**

# 

## Confidentiality Statement

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

Table of Contents

[Confidentiality Statement 2](#_30j0zll)

[Contact Information 4](#_1fob9te)

[Document History 4](#_3znysh7)

[Introduction 5](#_2et92p0)

[Assessment Objective 5](#_3dy6vkm)

[Penetration Testing Methodology 6](#_2s8eyo1)

[Reconnaissance 6](#_17dp8vu)

[Identification of Vulnerabilities and Services 6](#_3rdcrjn)

[Vulnerability Exploitation 6](#_26in1rg)

[Reporting 6](#_lnxbz9)

[Scope 7](#_35nkun2)

[Executive Summary of Findings 8](#_44sinio)

[Grading Methodology 8](#_z337ya)

[Summary of Strengths 9](https://docs.google.com/document/d/1q3OdXheqkVOWvRg5ectfVmpn5Sos9XZWdIsqMif1PLw/edit#heading=h.3j2qqm3)

[Summary of Weaknesses 9](#_1y810tw)

[Executive Summary Narrative 10](https://docs.google.com/document/d/1q3OdXheqkVOWvRg5ectfVmpn5Sos9XZWdIsqMif1PLw/edit#heading=h.4i7ojhp)

[Summary Vulnerability Overview 1](https://docs.google.com/document/d/1q3OdXheqkVOWvRg5ectfVmpn5Sos9XZWdIsqMif1PLw/edit#heading=h.2xcytpi)3

Vulnerability Findings [1](https://docs.google.com/document/d/1q3OdXheqkVOWvRg5ectfVmpn5Sos9XZWdIsqMif1PLw/edit#heading=h.1ci93xb)4

# 

## Contact Information

| **Company Name** | F00LISH Cybersecurity Enterprises, LLC |
| --- | --- |
| **Contact Name** | Jordan Smith-St.Kitts |
| **Contact Title** | Penetration Tester |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 2025-02-28 | Jordan Smith-St.Kitts | Penetration Test Report |

# 

## Introduction

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

For the testing, we focused on the following:

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

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

### Assessment Objective

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

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

Rekall has outlined the following objectives:

Table 1: Defined Objectives

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

# 

## Penetration Testing Methodology

### Reconnaissance

### 

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

### Identification of Vulnerabilities and Services

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

### Vulnerability Exploitation

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

### Reporting

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

# 

## Scope

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

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

# 

# 

## Executive Summary of Findings

## 

### Grading Methodology

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

**Critical**: Immediate threat to key business processes.

**High**: Indirect threat to key business processes/threat to secondary business processes.

**Medium**: Indirect or partial threat to business processes.

**Low**: No direct threat exists; vulnerability may be leveraged with other vulnerabilities.

Informational: No threat; however, it is data that may be used in a future attack.

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

Chart

Description automatically generated with medium confidence

### 

### Summary of Strengths

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

**Web Application Security Strengths:**

* XSS mitigation strategies
* Awareness of sensitive data exposure risks
* Protections against local file inclusion
* Secure data handling practices
* Strong password security policies
* Effective injection handling
* Session management controls
* Resilience against PHP injection
* Directory traversal prevention

**Linux Security Strengths:**

* Prevention of remote code execution
* Awareness of privilege escalation risks
* Network scanning detection and response
* Protection against password guessing attacks
* Post-exploitation defenses
* Handling of ShellShock exploits
* SSL certificate security measures
* Proactive OSINT and network scanning strategies

#### Windows Security Strengths:

* Defense against SLMail service exploitation
* User enumeration awareness
* Post-exploitation task monitoring
* Protections against admin-level compromise
* Controls to prevent unauthorized privilege escalation
* Proactive OSINT and FTP enumeration defenses
* Awareness of file enumeration risks

### 

### 

### 

### 

### 

### Summary of Weaknesses

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

#### Web Application Vulnerabilities:

* SQL injection risks
* Command injection flaws
* XSS vulnerabilities
* Weak session management controls
* Local file inclusion risks
* Brute-force attack susceptibility
* Directory traversal weaknesses
* Exposure of sensitive data
* PHP injection risks

#### Linux Security Weaknesses:

* Remote code execution vulnerabilities
* Post-exploitation risks
* Gaps in network scanning defenses
* Privilege escalation concerns
* Unpatched software vulnerabilities (identified through Nessus scans)
* Aggressive scanning risks (especially for Drupal environments)
* OSINT and network scanning vulnerabilities
* SSL certificate security gaps

#### Windows Security Weaknesses:

* SLMail service exploitation risks
* Admin account compromise threats
* User enumeration vulnerabilities
* Privilege escalation concerns
* OSINT and FTP enumeration weaknesses
* File enumeration risks
* Post-exploitation risks on Windows 10 systems
* Internal HTTP enumeration vulnerabilities

### 

## 

## Executive Summary

**Web Application Breakdown**

#### 

**Flag 1:**

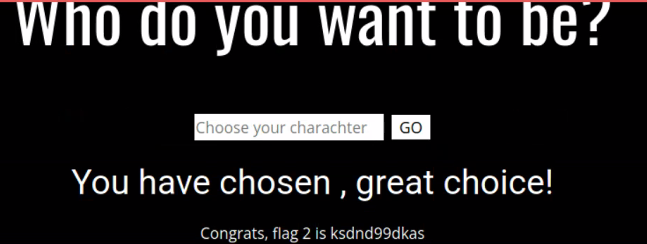
● To uncover Flag 1 I performed a XSS Payload exploit on the welcome page using the script: <script>alert</script>



#### 

**Flag 2:**

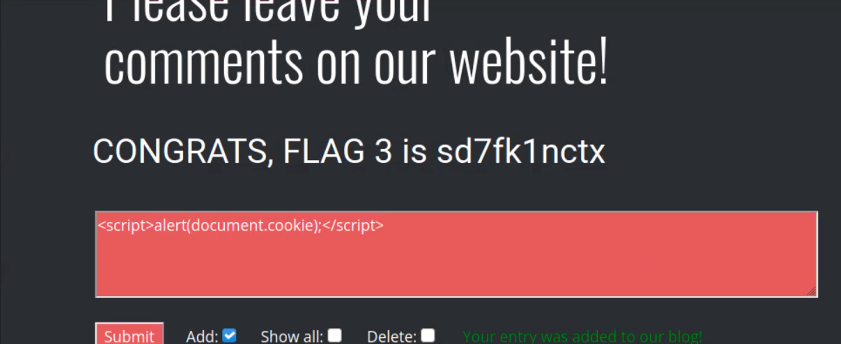
● For Flag 2, I figured out that this was a XSS Payload exploit and was used in the “Choose Your Character” field <SCRIPscriptT>alert("hi")</SCRIPscripTt>



#### 

**Flag 3:**

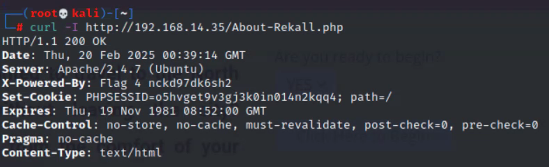
● For Flag 3, I had to navigate to the Rekall website 192.168.14.15/comments.php page to where in the comments box I used the following XSS script <script>alert(document.cookie);</script> to activate Flag 3 on screen.



#### 

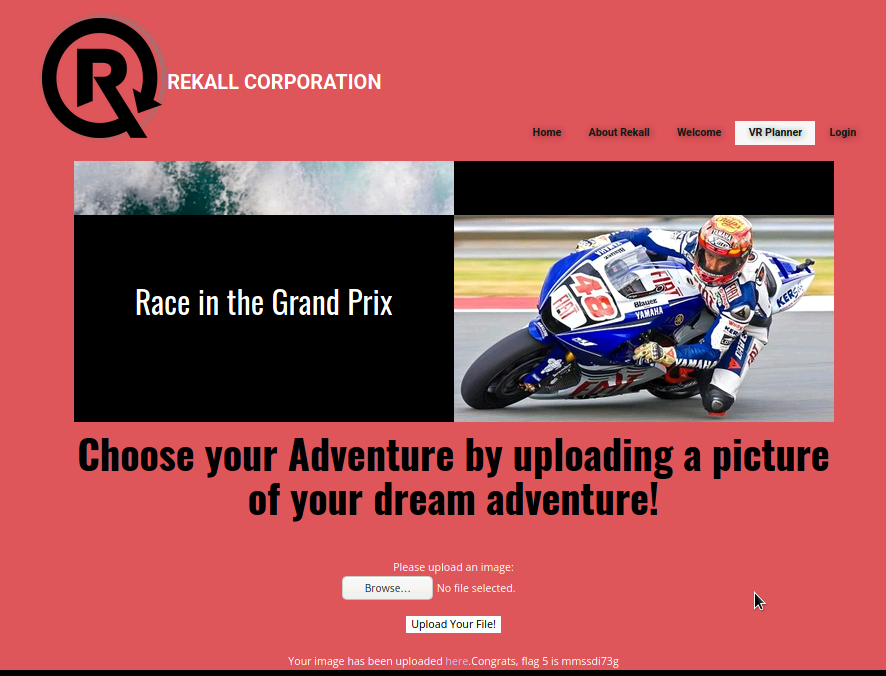
**Flag 4:**

● To capture Flag 4, within the linux terminal in Kali, l opened a new command prompt terminal and used the command: curl -v http://192.168.14.35/About-Rekall.php to grab website information and locate Flag 4.



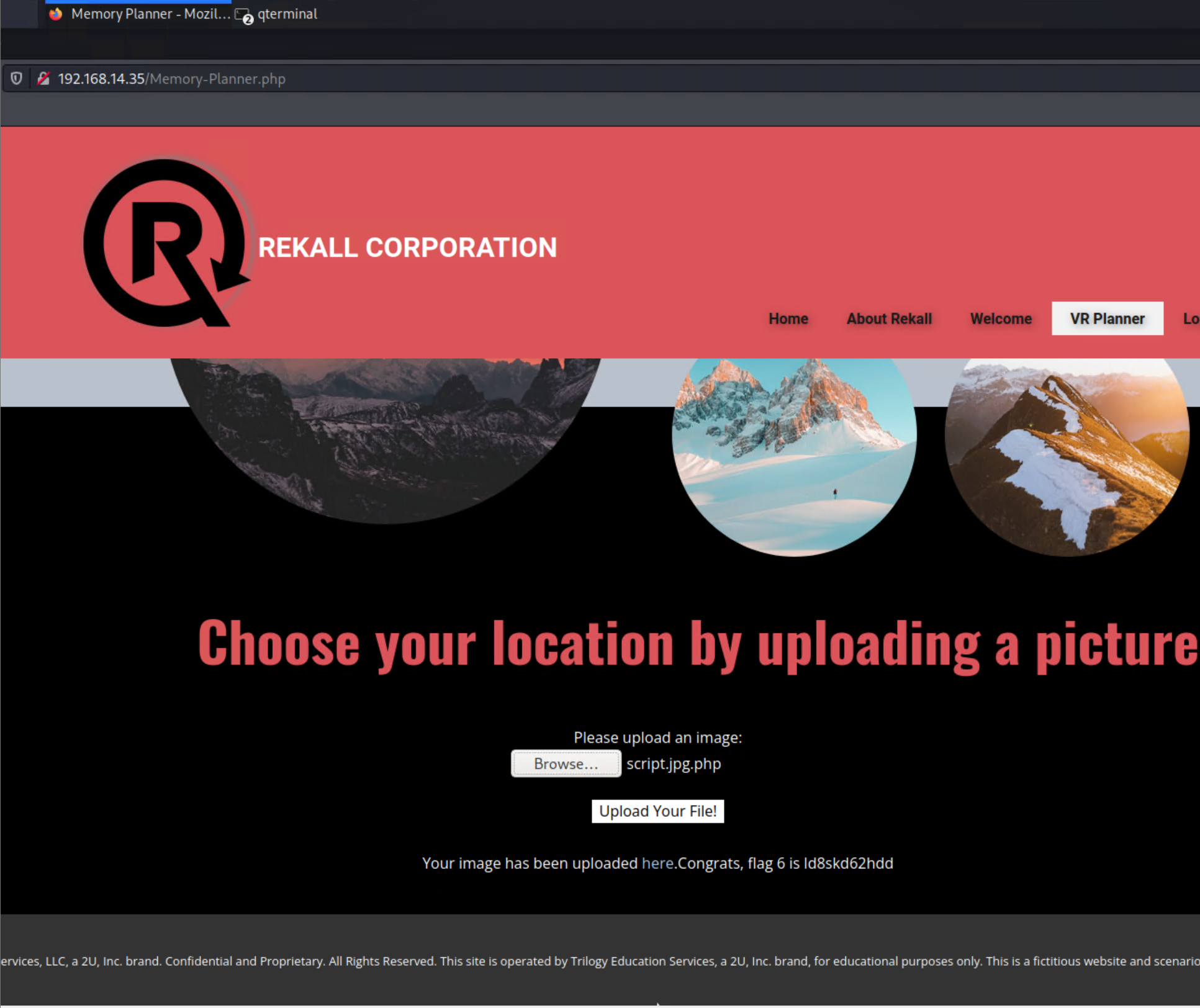
**Flag 5:**

● For Flag 5, in the linux terminal, I created a script.php file and uploaded it to the first upload field on the memory-planner page.



**Flag 6:**

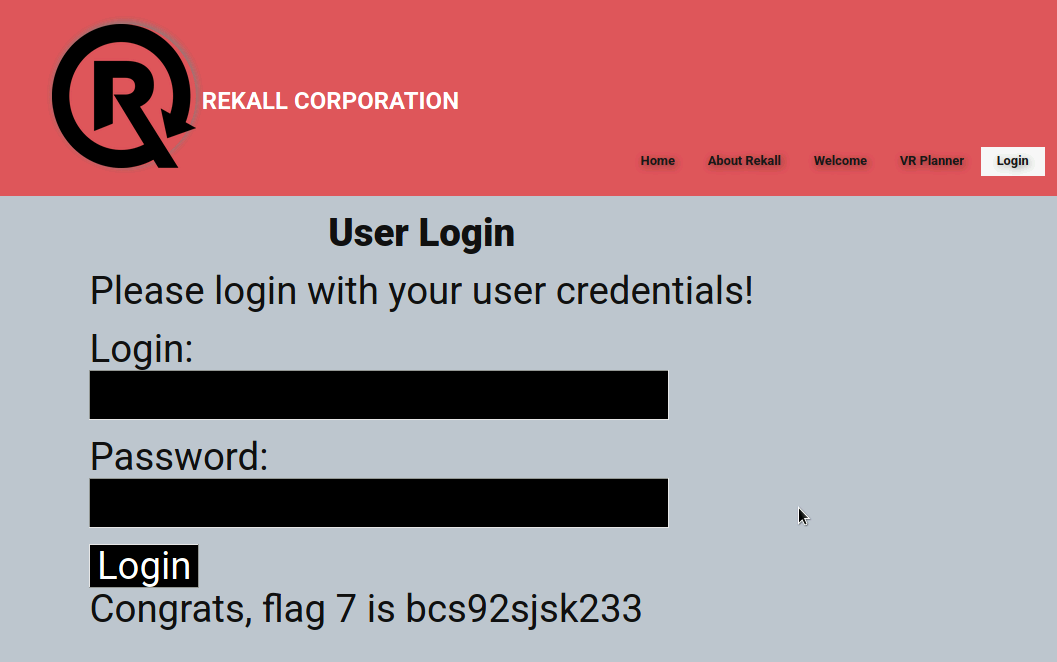
● To locate Flag 6 I uploaded a file ending in “jpg.php”. upload the file to the Memory-Planner.php page to reveal this flag.



#### 

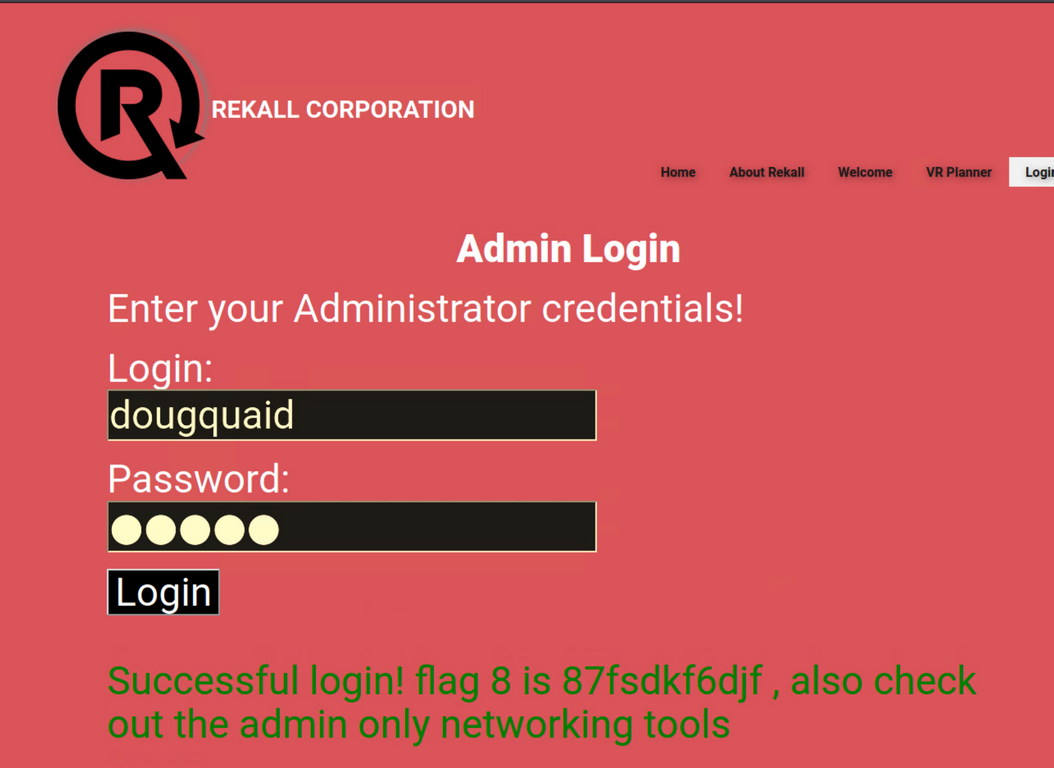
**Flag 7:**

● To locate Flag 7, I navigated to the 192.168.13.35/Login.php on the Rekall website and performed a SQL injection in the username and password section to find Flag 7.



**Flag 8:**

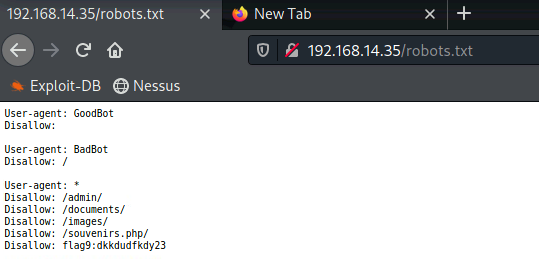
● I discovered Flag 8 within the HTML source code from the Login page. While inspecting the code I discovered the login credentials ‘dougquaid:kuato’ which then i used to log on the admin page and was able to get Flag 8.



#### 

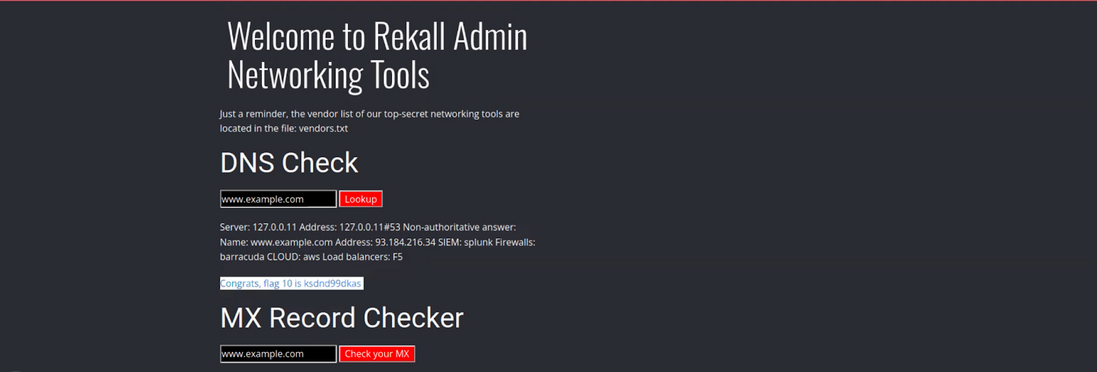
**Flag 9:**

● To locate Flag 9, I accessed 192.168.14.15/robots.txt which contained information on Flag 9.



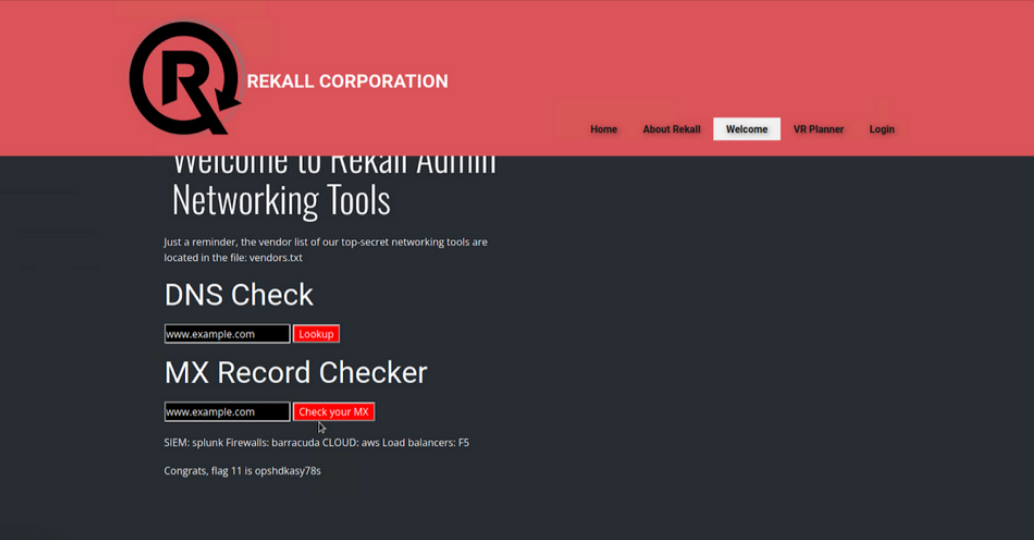
**Flag 10:**

● To capture Flag 10, I navigated to 192.168.14.15/network.php page and exploited a command injection vulnerability which allowed me to locate Flag 10.



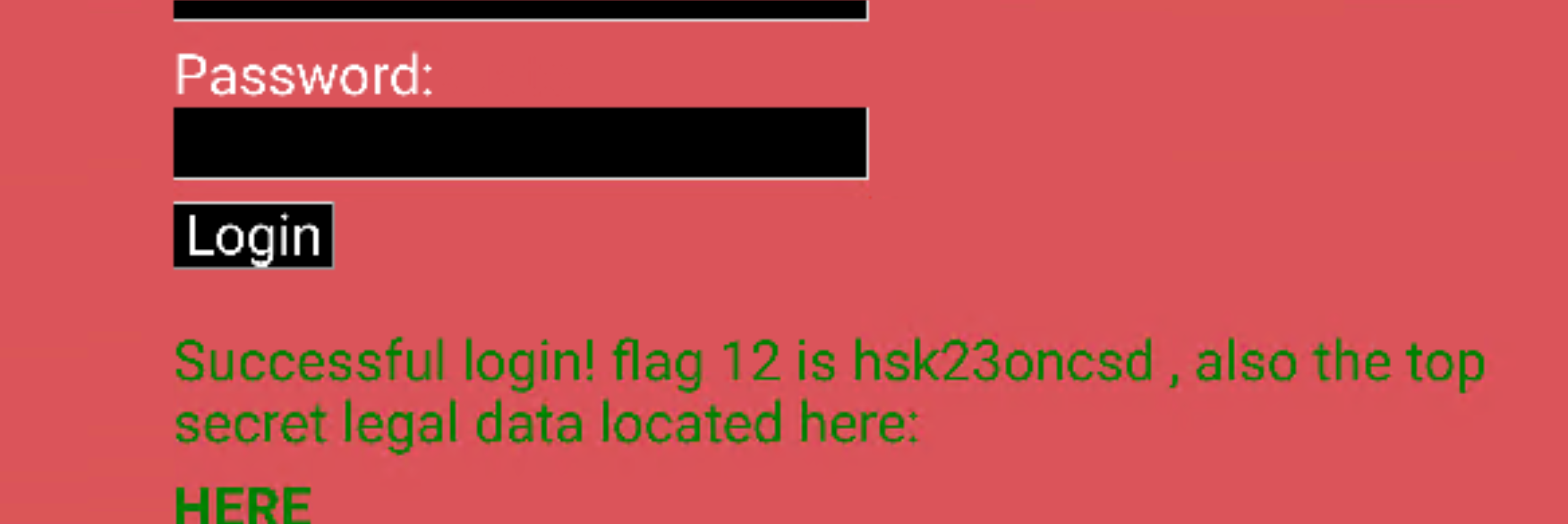
**Flag 11:**

● For Flag 11, I executed a command injection payload on the Networking page in the MX Record Checker section which allowed me to find Flag 11.



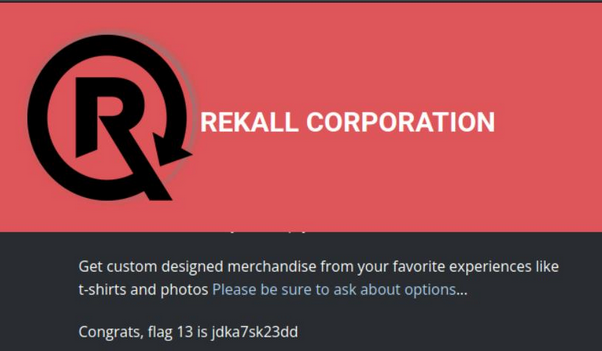
**Flag 12:**

● I accessed the 192.168.14.15/login.php page and attempted to log in using melina in both the user and password fields and gained entry. I discovered these credentials by using a simple password payload in burp intruder and found melina:melina. This led me to find Flag 12.



**Flag 13:**

● I navigated to 192.168.14.15/souvenirs.php page which I accessed after finding Flag 9. On this page, I exploited a PHP injection vulnerability using the payload ;system( ), which allowed me to reveal Flag 13.



**Flag 14:**

● I exploited a session management vulnerability on the admin\_legal\_data.php page by using Burp Intruder to brute-force session IDs. This allowed me to successfully retrieve Flag 14.



#### 

#### 

**Flag 15:**

● I discovered Flag 15 by exploiting a directory traversal vulnerability on the disclaimer.php page. By navigating to the old\_disclaimers directory, I found the flag in a file named disclaimer\_1.txt.

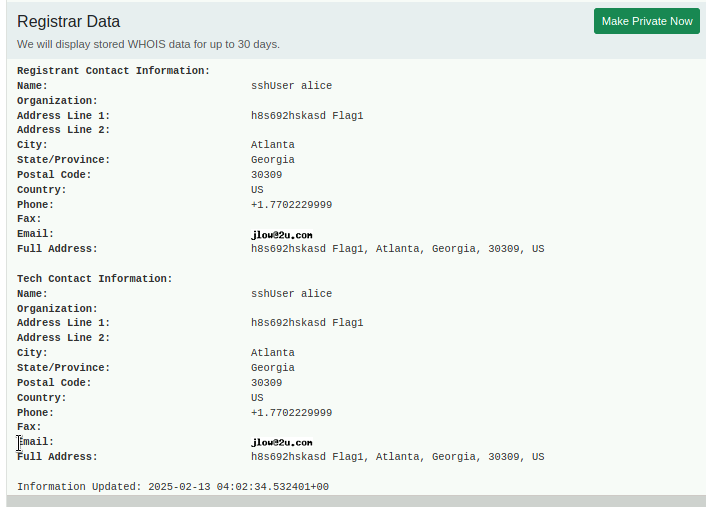


**Linux OS System**

#### 

**Flag 1:**

● To find Flag 1 on the Linux system, I visited https://centralops.net/co/DomainDossier.aspx and selected the "Domain WHOIS Record" option. This allowed me to view the WHOIS data for totalrekall.xyz, where I found Flag 1.

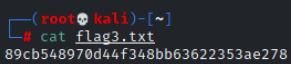


**Flag 2:**

● To find Flag 2, I navigated to https://centralops.net/co/DomainDossier.aspx and selected "DNS Records" on the Domain Dossier webpage. I then viewed the WHOIS data for totalrekall.xyz. The class was provided with the IP address for totalrekall.xyz which led me to Flag 2, but there was an issue with the IP address it returned.

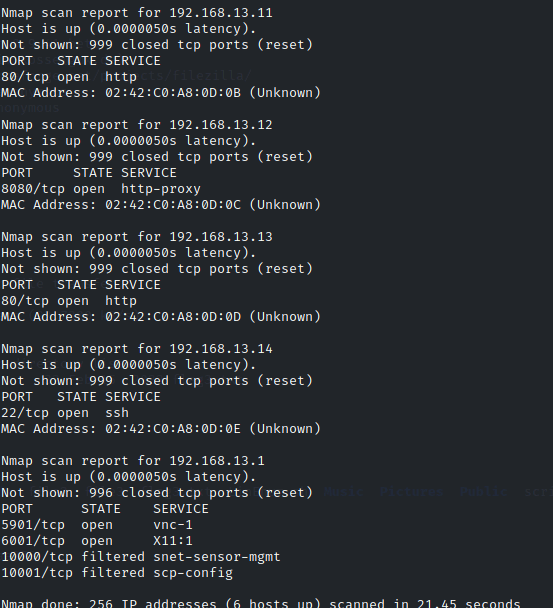
**Flag 3:**

● Flag 3 involved me performing an FTP connection to the IP address 172.22.117.20 to retrieve the data. After transferring the files to my local Kali machine, I was able to locate and extract Flag 3 and after performing the cat command on the flag3.txt I was able to gather the flag information.



**Flag 4:**

● To find Flag 4 I conducted an Nmap scan on the network using nmap 192.168.13.0/24, which revealed a total of five hosts. The flag corresponded to the number of detected hosts, excluding the one I was scanning from.

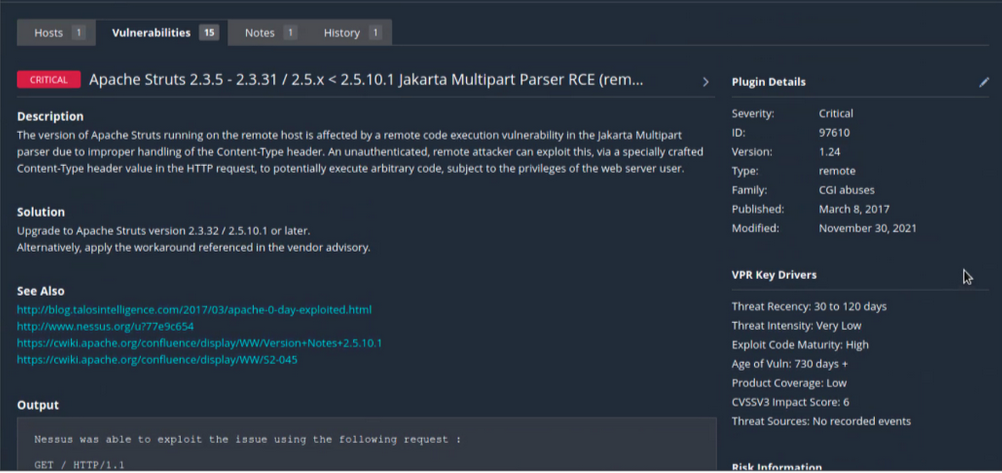


**Flag 5:**

● To uncover Flag 5 I ran an aggressive Nmap scan, which revealed that the host IP 192.168.13.1 was running Drupal.

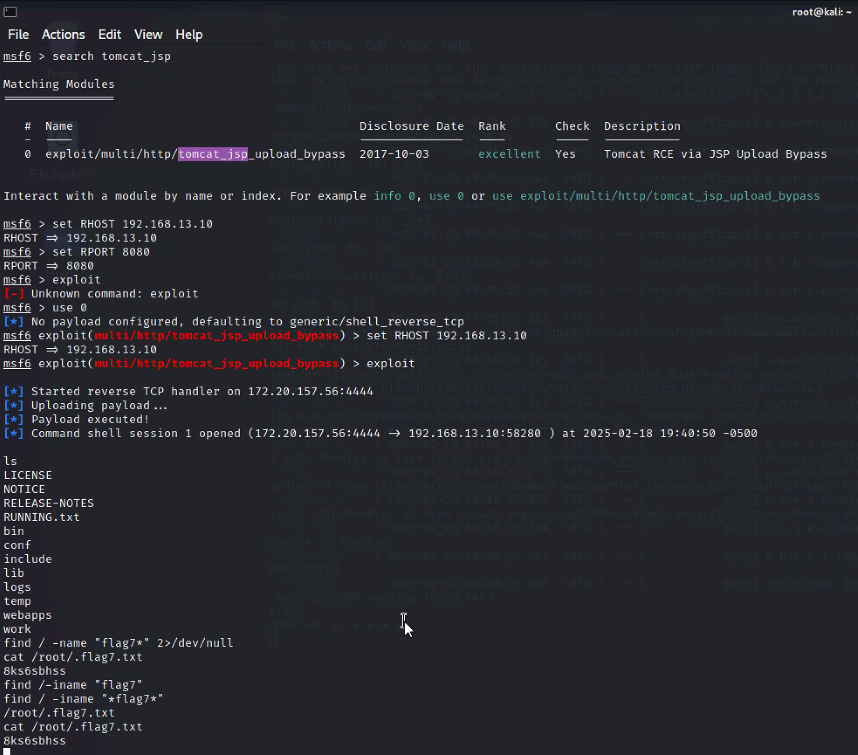
**Flag 6:**

● To find Flag 6, I ran a Nessus scan on the IP 192.168.13.12 and identified a critical vulnerability. The flag was displayed as ID 97610 in the top right corner of the scan results page.



**Flag 7:**

● Using MSFConsole, I searched for exploits targeting Tomcat and JSP. After configuring the RHOST to 192.168.13.10 and RPORT to 8080, I ran the exploit multiple times until I finally uncovered flag7.txt. A quick cat command later, and I secured.



**Flag 8:**

* To discover Flag 8 I used MSFConsole and searched for Shellshock exploits. I was able to use the exploit:

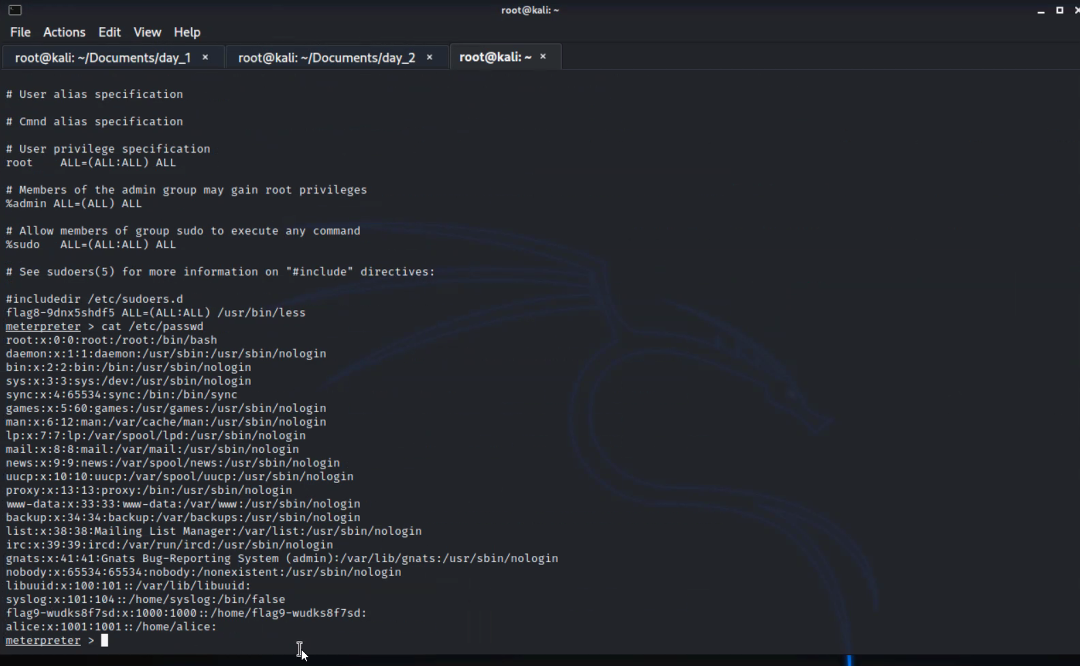
exploit/multi/http/apache\_mod\_cgi\_bash\_env\_exec

I set the necessary options, and ran the exploit. Once I got a shell, I ran cat /etc/sudoers and then I captured Flag 8.



**Flag 9:**

● Using the same machine where I previously discovered Flag 8, I conducted more investigation and was able to locate and capture Flag 9.



#### 

**Flag 10:**

* To find Flag 10 I went into MSFConsole, then I searched for Struts exploits and selected exploit/multi/http/struts2\_content\_type\_ognl. After setting RHOSTS to 192.168.13.12, I executed the exploit and gained access through Meterpreter. From there, I downloaded /root/flagisinThisfile.7z and extracted its contents. I ran cat on the extracted file to reveal Flag 10.



**Flag 11:**

● To get Flag 11 I launched MSFConsole and searched for Drupal exploits, selecting unix/webapp/drupal\_restws\_unserialize to gain a Meterpreter shell. After executing the exploit, I ran getuid to retrieve the server's username. Flag 11 was www-data.

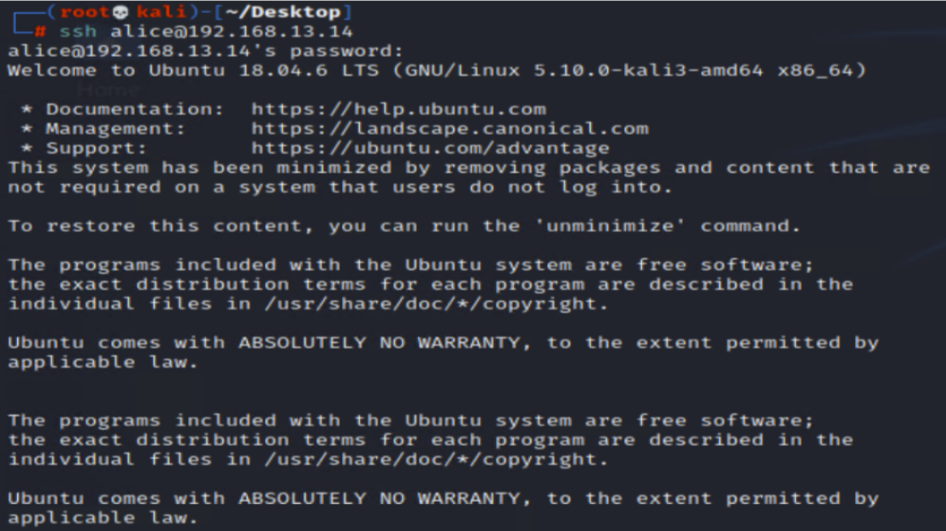
****

**Flag 12:**

● To retrieve Flag 12, I SSH'd into the server using ssh alice@192.168.13.14 and successfully guessed the password as "alice". Once inside, I performed privilege escalation by running

sudo -u#-1 cat /root/flag12.txt

After executing the command, I was able to access and retrieve the Flag 12.





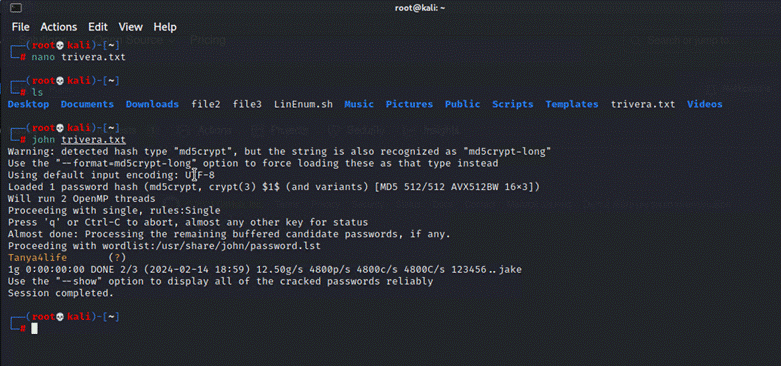
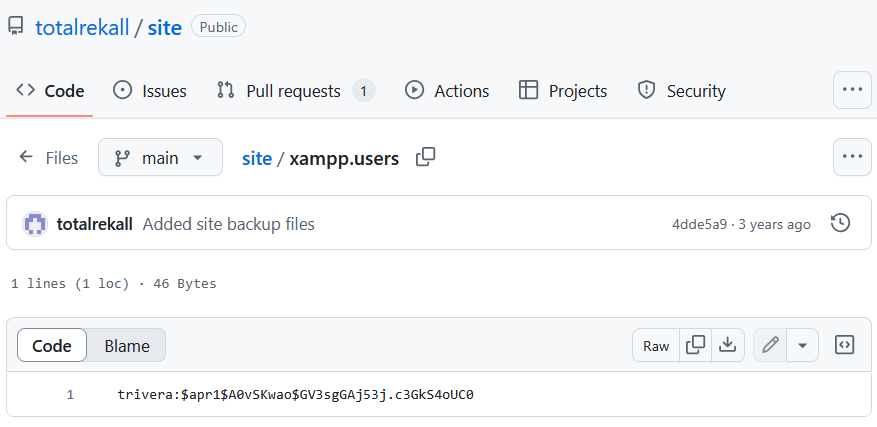
**Windows OS System**

**Flag 1:**

● I searched on GitHub repositories to find content related to totalrekall. While reviewing the repository, I found the xampp.users page, which contained the following credentials:

trivera:$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0

I saved the username and hash into a file named hash.txt using nano and then used John to crack the hash. The process revealed the password "Tanya4life", which unlocked Flag 1.

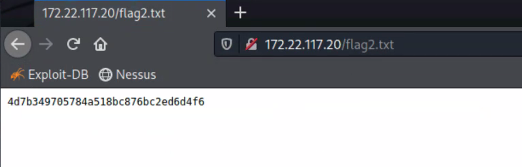


**Flag 2:**

● To find Flag 2, I conducted an Nmap scan to identify open ports. Knowing that the Windows network operates on the subnet 172.22.117.0/24 the scan revealed that 172.22.117.20 had an accessible service.

I navigated to 172.22.117.20 and used the credentials from Flag 1 (trivera:Tanya4life) to log in.

Once inside, I discovered flag2.txt, successfully retrieving Flag 2.



**Flag 3:**

● To locate Flag 3, I conducted an aggressive Nmap scan, which revealed an open FTP service on 172.22.117.20. I logged in as anonymous, I accessed the server using the command:

**ftp -p 172.22.117.20**

Once connected, I navigated through the directories and successfully retrieved the file containing Flag 3.



#### 

**Flag 4:**

● To locate Flag 4, I identified the SLMail service and used Metasploit via MSFConsole to exploit it. After selecting the appropriate exploit, I configured the following parameters:

LHOST = 172.22.117.100

RHOST = 172.22.117.20

RPORT = 110

Once the exploit was successfully executed, I navigated the system and used cat flag4.txt, revealing Flag 4.

#### 

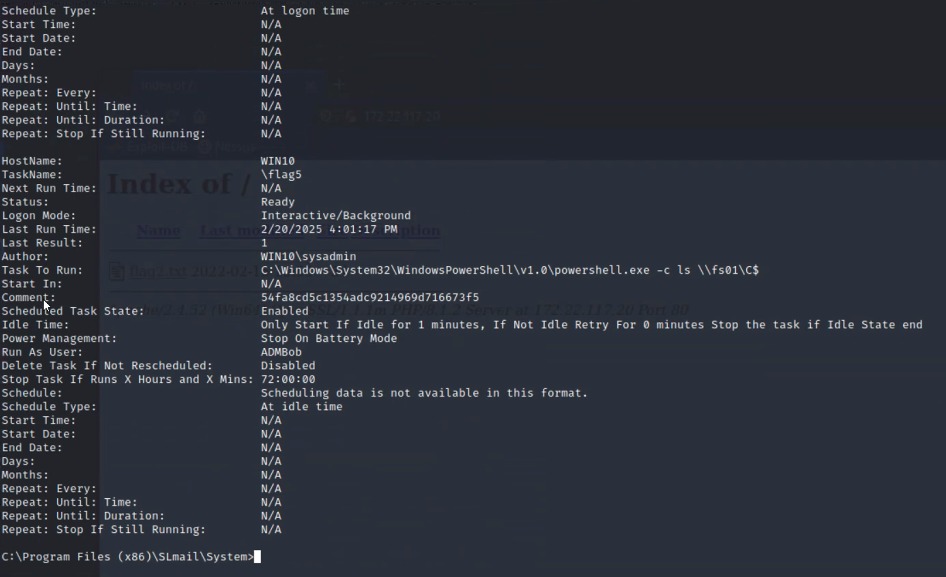
#### 

**Flag 5:**

● After gaining access to the Windows 10 machine, I evaluated scheduled tasks to identify potential issues. Within Meterpreter, I dropped into a command shell and executed the following command to query scheduled tasks:

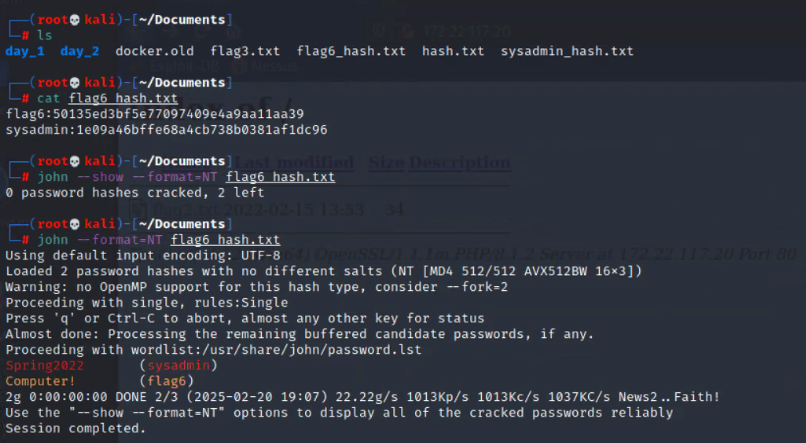
schtasks /query /tn "flag5" /v

This command displays detailed information about Flag 5 tasks, and successfully reveals Flag 5.



**Flag 6:**

● For Flag 6, continuing to exploit the same Windows 10 machine, I loaded Kiwi within Meterpreter to extract password hashes from system users. After retrieving the hashes, I saved them into a file and used John to crack the NTLM hash. This process successfully revealed the plaintext password, which was Flag 6.



**Flag 7:**

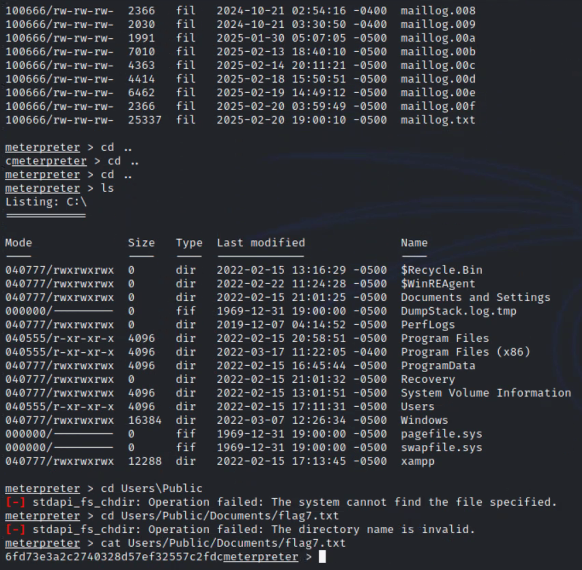
● To locate Flag 7 I continued on the same Windows 10 machine, I conducted a file search within Meterpreter to locate potential flag files. I used the following command:

search -f \*flag.txt\*

The search returned multiple results, but one stood out:

C:\Users\Public\Documents\flag7.txt

Upon navigating to this directory and inspecting the file, I successfully retrieved Flag 7.



#### 

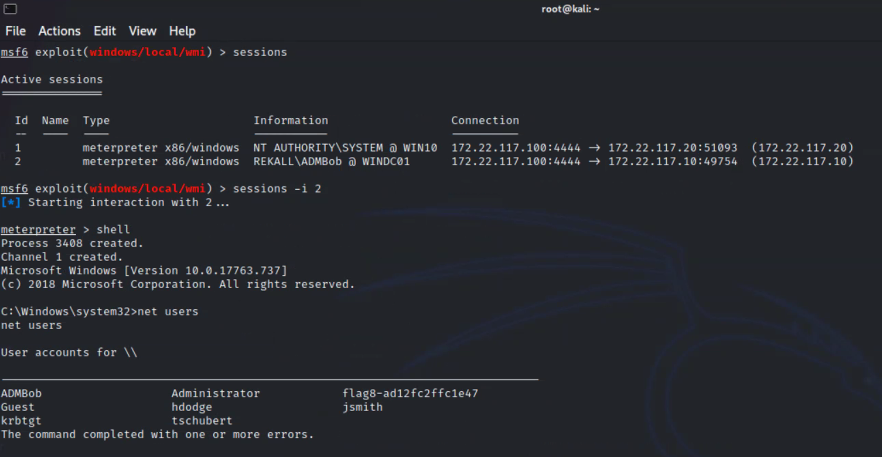
**Flag 8:**

● To find Flag 8 I was using Kiwi within Meterpreter. I dumped cached credentials from the Windows 10 machine.

The output revealed cached credentials for administrators, including ADMBob. I extracted the usernames and NTLM hashes to which I saved them into a .txt file and used John to crack them.

The cracked password was: Changeme!

Using these credentials, I moved to WinDC and enumerated user accounts discovering Flag 8.



#### 

**Flag 9:**

● Finding Flag 9 required me to continue to enumerate the WinDC machine. I conducted a file search in Meterpreter to locate flag files. I used the following command:

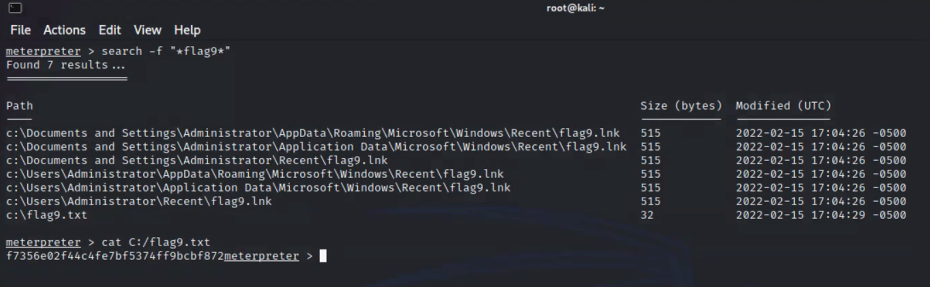
search -f \*flag9.txt\*

The search returned multiple results. I navigated through the filesystem and identified Flag 9 in a system directory.

To reveal its contents, I ran the command:

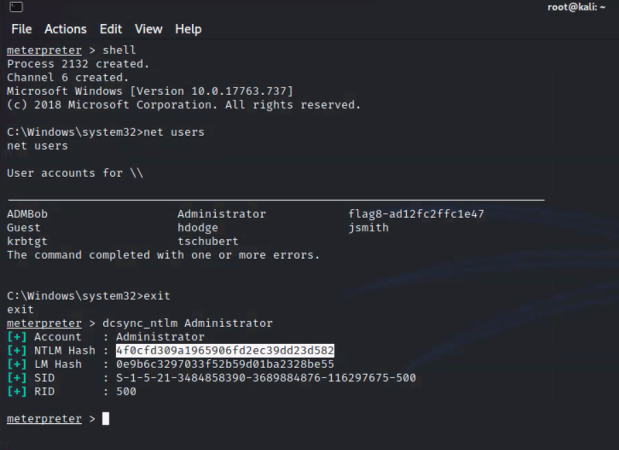
cat flag9.txt

And this successfully displayed the information for Flag 9.



**Flag 10:**

● For the final Flag, while in Meterpreter, I launched a shell and used Kiwi to perform a DCSync attack on the Administrator account. This method allowed me to gather the NTLM password hash directly from the Domain Controller. This successfully extracted the NTLM hash, which was Flag 10.



## **Summary Vulnerability Overview**

| **Vulnerability** | **Severity** |
| --- | --- |
| **Web Applications Vulnerabilities** |  |
|  |  |
| Flag 1: Reflected XSS on Welcome.php | **Critical** |
| Flag 2: XSS on Memory-Planner.php | **Critical** |
| Flag 3: XSS Stored Vulnerability on the Comments.php page | **Critical** |
| Flag 4: Sensitive Data Exposure | **Critical** |
| Flag 5: Local File Inclusion on the Memory-Planner.php | **Critical** |
| Flag 6: Local File Inclusion | **Critical** |
| Flag 7: SQL Injection on the Login.php | **Critical** |
| Flag 8: Sensitive Data Exposure | **High** |
| Flag: 9 Sensitive Data Exposure | **Medium** |
| Flag 10: Command Injection | **Medium** |
| Flag 11: Command Injection | **High** |
| Flag 12: Brute Force Attack | **Medium** |
| Flag 13: PhP Injection | **High** |
| Flag 14: Session Management | **Medium** |
| Flag 15: Directory Transversal on Disclaimer page | **Medium** |
|  |  |
| **Linux Operating System** |  |
|  |  |
| Flag 1: WHOIS Domain Recon | **Low** |
| Flag 2: IP Address Recon | **Low** |
| Flag 3: SSL Certificate Research | **Low** |
| Flag 4: Network Scanning | **Medium** |
| Flag 5: Aggressive Scan for Drupal | **Medium** |
| Flag 6: Nessus Scan and Vulnerability ID | **Medium** |
| Flag 7: Apache Tomcat Remote Code Execution Vulnerability | **Critical** |
| Flag 8: ShellShock Exploit | **High** |
| Flag 9: Escalating Access | **Medium** |
| Flag 10: Status Exploit | **Medium** |
| Flag 11: Drupal (CVE-2019-6340) | **High** |
| Flag 12: SSH Exploitation | **High** |
|  |  |
|  |  |
| **Windows Operating System** |  |
|  |  |
| Flag 1: Github Repository OSINT | **Critical** |
| Flag 2: HTTP Enumeration on Internal Network | **Critical** |
| Flag 3: FTP Enumeration | **High** |
| Flag 4: Metasploit the SLMail Service | **Medium** |
| Flag 5: Post Exploitation Task on Win10 | **High** |
| Flag 6: User enumeration on Windows 10 | **Critical** |
| Flag 7: File Enumeration | **Medium** |
| Flag 8: User enumeration pt. 2 | **High** |
| Flag 9: Escalating Access | **High** |
| Flag 10: Compromising Admin | **High** |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | 172.22.117.10  172.22.117.20  192.168.13.10  192.168.13.11  192.168.13.12  192.168.13.13  192.168.13.14  192.168.13.0/24  192.168.13.1  192.168.13.35  192.168.14.35  192.158.13.14  172.22.117.0/24 |
| Ports | 8009  80  8080  8081  21  22  79  106  110  135  139  443  445  5901  6001  10000  10001  SLMail POP3 |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 8 |
| **High** | 11 |
| **Medium** | 12 |
| **Low** | 3 |

## 

## Vulnerability Findings

| **Vulnerability 1** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 1 |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | To uncover Flag 1 I performed a XSS Payload exploit on the welcome page using the script: <script>alert</script> |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35/Welcome.php - Totalrekall.xyz |
| **Remediation** | Finding ways to implement input validation |

| **Vulnerability 2** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 2 |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | For Flag 2, I figured out that this was a XSS Payload exploit and was used in the “Choose Your Character” field <SCRIPscriptT>alert("hi")</SCRIPscripTt> |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35/Memory-planner.php - Totalrekall.xyz |
| **Remediation** | HTML Sanitization |

| **Vulnerability 3** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 3 |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | For Flag 3, I had to navigate to the Rekall website 192.168.14.15/comments.php page to where in the comments box I used the following XSS script <script>alert(document.cookie);</script> to activate Flag 3 on screen. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35/comments.php |
| **Remediation** | Comment box encoding |

| **Vulnerability 4** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 4 |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | To capture Flag 4, within the linux terminal in Kali, l opened a new command prompt terminal and used the command: curl -v http://192.168.14.35/About-Rekall.php to grab website information and locate Flag 4. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35/About-Rekall.php - TotalRekall.xyz |
| **Remediation** | Configure proper access channels |

| **Vulnerability 5** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 5 - Local File Inclusion (LFI) exploit |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | For Flag 5, in the linux terminal, I created a script.php file and uploaded it to the first upload field on the memory-planner page. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.45/memory-planner.php |
| **Remediation** | File Upload Validation - Make sure that only .jpg files or image files are the only required upload files to be uploaded |

| **Vulnerability 6** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 6 - LFI (advanced) |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | To locate Flag 6 I uploaded a file ending in “jpg.php”. upload the file to the Memory-Planner.php page to reveal this flag. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.45/Memory-planner.php |
| **Remediation** | File validation; Make sure that only the required files are the only files that can be uploaded. |

| **Vulnerability 7** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 7 - SQL injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | To locate Flag 7, I navigated to the 192.168.13.35/Login.php on the Rekall website and performed a SQL injection in the username and password section to find Flag 7. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.35/Login.php |
| **Remediation** | Don't build SQL queries by combining strings. They should use prepared statements to keep user inputs separate from what the SQL code is. |

| **Vulnerability 8** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 8 |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | I discovered Flag 8 within the HTML source code from the Login page. While inspecting the code I discovered the login credentials ‘dougquaid:kuato’ which then i used to log on the admin page and was able to get Flag 8. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.35, Login.php |
| **Remediation** | Conduct regular security audits and respond to any suspicious activities of login attempts. |

| **Vulnerability 9** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 9 - Sensitive data exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | To locate Flag 9, I accessed 192.168.14.15/robots.txt which contained information on Flag 9. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Ensure that only authorized users have access to certain information |

| **Vulnerability 10** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 10 - Command injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | To capture Flag 10, I navigated to 192.168.14.15/network.php page and exploited a command injection vulnerability which allowed me to locate Flag 10. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.15/network.php |
| **Remediation** | Identity compromised systems |

| **Vulnerability 11** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 11 - Command injection (advanced) |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | For Flag 11, I executed a command injection payload on the Networking page in the MX Record Checker section which allowed me to find Flag 11. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.15/network.php |
| **Remediation** | Identity compromised systems |

| **Vulnerability 12** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 12 - Brute force attacks |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | I accessed the 192.168.14.15/login.php page and attempted to log in using melina in both the user and password fields and gained entry. I discovered these credentials by using a simple password payload in burp intruder and found melina:melina. This led me to find Flag 12. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.35, Login.php |
| **Remediation** | Enforce strong password policies and educate users about strong and unique passwords. |

| **Vulnerability 13** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 13 - PHP injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | I navigated to 192.168.14.15/souvenirs.php page which I accessed after finding Flag 9. On this page, I exploited a PHP injection vulnerability using the payload ;system( ), which allowed me to reveal Flag 13. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.15/souvenirs.php |
| **Remediation** | implement security policies to prevent payloads being entered. |

| **Vulnerability 14** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 14 - Session management |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | I exploited a session management vulnerability on the admin\_legal\_data.php page by using Burp Intruder to brute-force session IDs. This allowed me to successfully retrieve Flag 14. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35/admin\_legal\_data.php |
| **Remediation** | Policies need to be in place to prevent payloads. |

| **Vulnerability 15** | **Web App Findings** |
| --- | --- |
| **Title** | Flag 15 - Directory traversal |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | I discovered Flag 15 by exploiting a directory traversal vulnerability on the disclaimer.php page. By navigating to the old\_disclaimers directory, I found the flag in a file named disclaimer\_1.txt. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35/disclaimer.php |
| **Remediation** | Security policies need to be in place to prevent payloads. |

| **Vulnerability 1** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 1 - Reconnaissance |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Low |
| **Description** | To find Flag 1 on the Linux system, I visited https://centralops.net/co/DomainDossier.aspx and selected the "Domain WHOIS Record" option. This allowed me to view the WHOIS data for totalrekall.xyz, where I found Flag 1. |
| **Images** |  |
| **Affected Hosts** | TotalRekall.xyz |
| **Remediation** | Conduct web application vulnerability scans regularly. |

| **Vulnerability 2** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 2 - Reconnaissance |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Low |
| **Description** | To find Flag 2, I navigated to https://centralops.net/co/DomainDossier.aspx and selected "DNS Records" on the Domain Dossier webpage. I then viewed the WHOIS data for totalrekall.xyz. The class was provided with the IP address for totalrekall.xyz which led me to Flag 2, but there was an issue with the IP address it returned. |
| **Images** |  |
| **Affected Hosts** | 34.102.136.180 |
| **Remediation** | Check IP Addresses for any suspicious activity. |

| **Vulnerability 3** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 3 - Reconnaissance |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Low |
| **Description** | Flag 3 involved me performing an FTP connection to the IP address 172.22.117.20 to retrieve the data. After transferring the files to my local Kali machine, I was able to locate and extract Flag 3 and after performing the cat command on the flag3.txt I was able to gather the flag information. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Monitor FTP Data |

| **Vulnerability 4** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 4 - Scanning |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Medium |
| **Description** | To find Flag 4 I conducted an Nmap scan on the network using nmap 192.168.13.0/24, which revealed a total of five hosts. The flag corresponded to the number of detected hosts, excluding the one I was scanning from. |
| **Images** |  |
| **Affected Hosts** | Nessus |
| **Remediation** | Identify and close any open ports that are not actively used. |

| **Vulnerability 5** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 5 |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Medium |
| **Description** | To uncover Flag 5 I ran an aggressive Nmap scan, which revealed that the host IP 192.168.13.1 was running Drupal. |
| **Images** |  |
| **Affected Hosts** | 192.198.13.1 |
| **Remediation** | Monitor scans to vulnerable threats. |

| **Vulnerability 6** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 6 - Scanning |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Medium |
| **Description** | To find Flag 6, I ran a Nessus scan on the IP 192.168.13.12 and identified a critical vulnerability. The flag was displayed as ID 97610 in the top right corner of the scan results page. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Update and patch software on a regular basis and always monitor for vulnerabilities. |

| **Vulnerability 7** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 7 - Exploit |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | Using MSFConsole, I searched for exploits targeting Tomcat and JSP. After configuring the RHOST to 192.168.13.10 and RPORT to 8080, I ran the exploit multiple times until I finally uncovered flag7.txt. A quick cat command later, and I secured. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.10 |
| **Remediation** | Configure access controls, limit user access and disable any unnecessary services. |

| **Vulnerability 8** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 8 - Scanning |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | To discover Flag 8 I used MSFConsole and searched for Shellshock exploits. I was able to use the exploit:  exploit/multi/http/apache\_mod\_cgi\_bash\_env\_exec  I set the necessary options, and ran the exploit. Once I got a shell, I ran cat /etc/sudoers and then I captured Flag 8. |
| **Images** |  |
| **Affected Hosts** | 192.198.13.1 |
| **Remediation** | Identify unnecessary services running on Linux systems |

| **Vulnerability 9** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 9 - Scanning |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Medium |
| **Description** | Using the same machine where I previously discovered Flag 8, I conducted more investigation and was able to locate and capture Flag 9. |
| **Images** |  |
| **Affected Hosts** | 192.198.13.1 |
| **Remediation** | Network scanning and access. |

| **Vulnerability 10** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 10 - Scanning |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Medium |
| **Description** | To find Flag 10 I went into MSFConsole, then I searched for Struts exploits and selected exploit/multi/http/struts2\_content\_type\_ognl. After setting RHOSTS to 192.168.13.12, I executed the exploit and gained access through Meterpreter. From there, I downloaded /root/flagisinThisfile.7z and extracted its contents. I ran cat on the extracted file to reveal Flag 10. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Monitor system configurations, scheduled tasks, and startup processes. |

| **Vulnerability 11** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 11 - Scanning |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | To get Flag 11 I launched MSFConsole and searched for Drupal exploits, selecting unix/webapp/drupal\_restws\_unserialize to gain a Meterpreter shell. After executing the exploit, I ran getuid to retrieve the server's username. Flag 11 was www-data. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.13 |
| **Remediation** | Perform security audits and vulnerability scans for sensitive data. |

| **Vulnerability 12** | **Linux Findings** |
| --- | --- |
| **Title** | Linux Flag 12 |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | To retrieve Flag 12, I SSH'd into the server using ssh alice@192.168.13.14 and successfully guessed the password as "alice". Once inside, I performed privilege escalation by running  sudo -u#-1 cat /root/flag12.txt  After executing the command, I was able to access and retrieve the Flag 12. |
| **Images** |  |
| **Affected Hosts** | 192.158.13.14 |
| **Remediation** | Enforce strong password policies, enforce multi-factor authentication and educate users on the importance of password strength. |

| **Vulnerability 1** | **Windows Findings** |
| --- | --- |
| **Title** | Windows Flag 1 - OSINT |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Critical |
| **Description** | I searched on GitHub repositories to find content related to totalrekall. While reviewing the repository, I found the xampp.users page, which contained the following credentials:  **trivera:$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0**  I saved the username and hash into a file named hash.txt using nano and then used John to crack the hash. The process revealed the password "Tanya4life",  which unlocked Flag 1. |
| **Images** |  |
| **Affected Hosts** | TotalRekall Web-Server |
| **Remediation** | Remove user Credentials from Github |

| **Vulnerability 2** | **Windows Findings** |
| --- | --- |
| **Title** | Windows Flag 2 - HTTP Enumeration |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Critical |
| **Description** | To find Flag 2, I conducted an Nmap scan to identify open ports. Knowing that the Windows network operates on the subnet 172.22.117.0/24 the scan revealed that 172.22.117.20 had an accessible service.  I navigated to 172.22.117.20 and used the credentials from Flag 1 (trivera:Tanya4life) to log in.  Once inside, I discovered flag2.txt, successfully retrieving Flag 2. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Remove credentials from the public, and have a better authentication process. |

| **Vulnerability 3** | **Windows Findings** |
| --- | --- |
| **Title** | Windows Flag 3 - FTP Enumeration |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | To locate Flag 3, I conducted an aggressive Nmap scan, which revealed an open FTP service on 172.22.117.20. I logged in as anonymous, I accessed the server using the command:  **ftp -p 172.22.117.20**  Once connected, I navigated through the directories and successfully retrieved the file containing Flag 3. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Monitor scans for vulnerabilities. |

| **Vulnerability 4** | **Windows Findings** |
| --- | --- |
| **Title** | Windows Flag 4 - Metasploit |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Medium |
| **Description** | I identified a machine running the SLMail service and used Metasploit via MSFConsole to exploit it. After selecting the appropriate exploit, I configured the following parameters:  LHOST = 172.22.117.100 (my local machine within the subnet)  RHOST = 172.22.117.20 (target machine)  RPORT = 110 (SLMail POP3 service port)  Once the exploit was successfully executed, I navigated the system and used cat flag4.txt, revealing Flag 4. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Install antivirus and anti-malware software on the Windows server hosting SLmail and monitor port 110. |

| **Vulnerability 5** | **Windows Findings** |
| --- | --- |
| **Title** | Windows Flag 5 - Common Tasks |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | After gaining access to the Windows 10 machine, I evaluated scheduled tasks to identify potential issues. Within Meterpreter, I dropped into a command shell and executed the following command to query scheduled tasks:  schtasks /query /tn "flag5" /v  This command displays detailed information about Flag 5 tasks, and successfully reveals Flag 5. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Review and update startup processes and scheduled tasks. |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Windows Flag 6 - User Enumeration |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Critical |
| **Description** | For Flag 6, continuing to exploit the same Windows 10 machine, I loaded Kiwi within Meterpreter to extract password hashes from system users. After retrieving the hashes, I saved them into a file and used John to crack the NTLM hash. This process successfully revealed the plaintext password, which was Flag 6. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Enforce strong password policies and store password hashes in a secure location. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Windows Flag 7 - File Enumeration |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Medium |
| **Description** | To locate Flag 7 I continued on the same Windows 10 machine, I conducted a file search within Meterpreter to locate potential flag files. I used the following command:  **search -f \*flag.txt\***  The search returned multiple results, but one stood out:  **C:\Users\Public\Documents\flag7.txt**  Upon navigating to this directory and inspecting the file, I successfully retrieved Flag 7. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Regularly monitor permissions and ensure users and groups have the correct privileges. |

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** | Windows Flag 8 - User Enumeration pt.2 |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | To find Flag 8 I was using Kiwi within Meterpreter. I dumped cached credentials from the Windows 10 machine.  The output revealed cached credentials for administrators, including ADMBob. I extracted the usernames and NTLM hashes to which I saved them into a .txt file and used John to crack them.  The cracked password was: Changeme!  Using these credentials, I moved to WinDC and enumerated user accounts discovering Flag 8. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Regularly audit account privileges and limit network segmentation to limit lateral movement. |

| **Vulnerability 9** | **Windows Findings** |
| --- | --- |
| **Title** | Windows Flag 9 - Escalating Access |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | Finding Flag 9 required me to continue to enumerate the WinDC machine. I conducted a file search in Meterpreter to locate flag files. I used the following command:  search -f \*flag9.txt\*  The search returned multiple results. I navigated through the filesystem and identified Flag 9 in a system directory.  To reveal its contents, I ran the command:  cat flag9.txt  And this successfully displayed the information for Flag 9. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Implement steps to detect privilege escalation attempts. |

| **Vulnerability 10** | **Findings** |
| --- | --- |
| **Title** | Windows Flag 10 - Compromising Admin |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | For the final Flag, while in Meterpreter, I launched a shell and used Kiwi to perform a DCSync attack on the Administrator account. This method allowed me to gather the NTLM password hash directly from the Domain Controller. This successfully extracted the NTLM hash, which was Flag 10. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Immediately isolate compromised accounts and systems to prevent further damage and data loss. |