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	GoodCorp, LLC
Contact Name	Valentina Jardan
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	7/13/2024	Valentina Jardan	Initial Document Setup
002	7/14/2024	Valentina Jardan	Vulnerabilities Web Application
003	7/15/2024	Valentina Jardan	Vulnerabilities Linux Servers
004	7/16/2024	Valentina Jardan	Vulnerabilities Windows Servers

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.

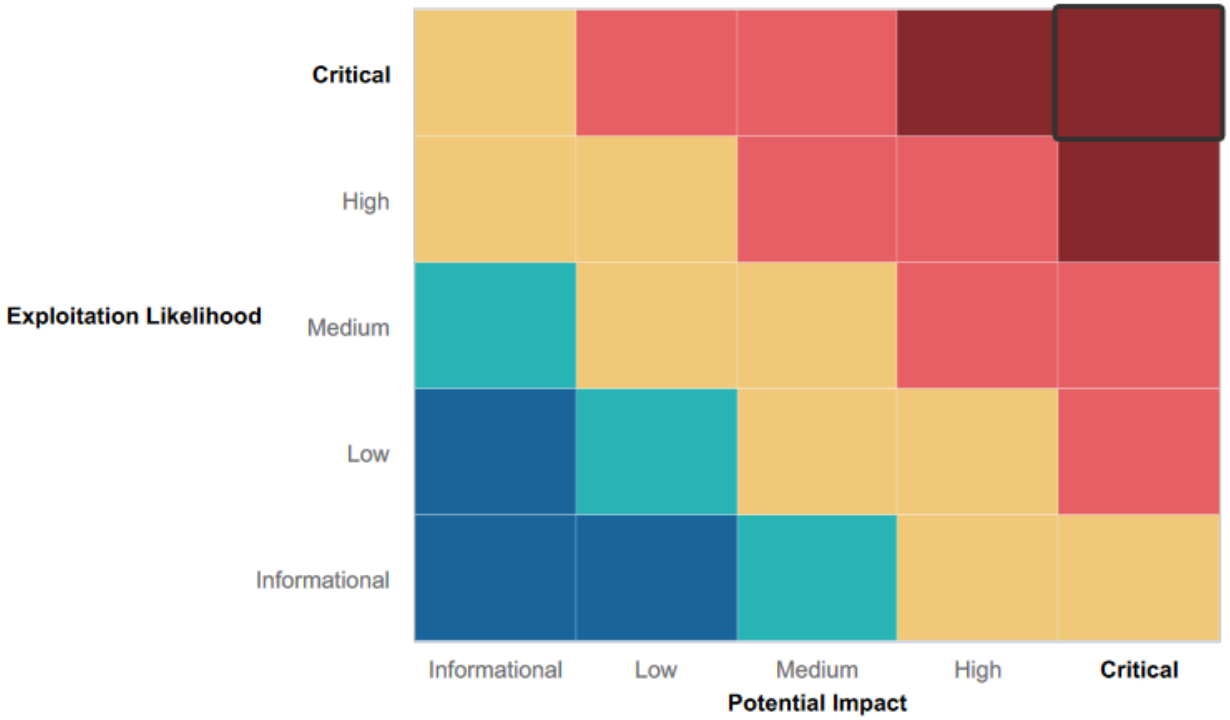
Executive Summary of Findings

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.

- Some input fields on the Rekall website are using input validation and are well protected against basic XSS, SQL, Command Injections and demanding extensive testing.
- Rekall Inc. demonstrates a commitment to proactive security by engaging GoodCorp LLC to conduct a penetration test. This approach helps identify and address vulnerabilities before they can be exploited by malicious actors.

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 is vulnerable
    - Cross Site Scripting & PHP Injection
    - SQL Injection
    - Command Injection
    - Local File Inclusion (LFI)
    - Brute force attacks
    - Session Hijacking
    - Directory traversal
- Sensitive Data Exposure
    - in the HTTP web page header
    - in public pages and files
    - in Totalrekall GitHub repositories
    - 'C:\Users\Public\Documents' directory on Windows 10 machine
- Credentials are being stored in HTML source code
- Several ports are open revealing potential vulnerabilities
- Apache web server is outdated and vulnerable
- SLMail server is outdated and vulnerable
- Apache Tomcat is outdated and vulnerable
- Drupal 8 core versions is vulnerable Remote Code Execution (RCE)
- Poor Access Control on /etc/passwd
- Unnecessary or potentially malicious scheduled tasks on the Windows 10 machine
- Windows Local Account Credential Dumping Vulnerability
- Privilege Escalation using sudo vulnerability
- Anonymous access FTP vulnerability

Executive Summary

Note to TA: While real penetration testing reports typically omit details about specific tools and techniques, usernames, and passwords to protect confidentiality, this report includes them due to the educational nature of this exercise, as required by the Module 18 Challenge Instruction.

This report details the penetration testing engagement conducted by GoodCorp LLC for Rekall Inc. The objective of this engagement was to assess the security posture of Rekall 's systems and identify potential vulnerabilities. This report outlines the identified vulnerabilities, along with recommendations for improvement.

GoodCorp performed a comprehensive security assessment of a simulated environment, including the **Web Application**, **Linux servers**, and **Windows servers**, to identify potential vulnerabilities.

Below is a **step-by-step explanation** and all the vulnerabilities discovered during the penetration test, along with a summary of the methodology used.

Security Assessment of Rekall's Web Application

GoodCorp's first day concentrated on identifying weaknesses in Rekall's web application. We began by testing for Cross-Site Scripting (XSS) vulnerabilities. Our efforts revealed a reflected XSS vulnerability on the Welcome.php and Memory-Planner.php pages. By injecting a harmless script, we were able to trigger an alert, confirming the vulnerability's presence.

Check the Vulnerability for technical details	Reflected Cross Site Scripting (XSS) Vulnerability on the Welcome.php page Reflected Cross Site Scripting (XSS) Vulnerability on the Memory-Planner.php page

We were able to identify an additional vulnerability on the Rekall web application's comments.php page. This vulnerability is classified as a Stored Cross-Site Scripting (XSS) issue. Unlike Reflected XSS where the malicious script disappears after the page loads (mentioned above), Stored XSS vulnerabilities allow attackers to embed malicious scripts that persist on the server and execute whenever the page loads, so any user visiting the comments section could be impacted.

Check the Vulnerability for technical details	Stored Cross Site Scripting (XSS) Vulnerability on the Comments.php page

GoodCorp discovered another critical security vulnerability on the Login.php page. The vulnerability exposed sensitive user credentials (usernames and passwords), directly within the page source code. This means anyone with access to the page source, could easily view these credentials and potentially gain unauthorized access to user accounts.

Check the Vulnerability for technical details	User Credentials Exposure on Login.php page

In addition to this critical vulnerability, the robots.txt file, which can be used to control search engine indexing, may have also exposed sensitive information. Another potential sensitive information leakage was found in the HTTP header.

Check the Vulnerability for technical details	Sensitive Data Exposure in robot.txt file Sensitive Data Exposure in the HTTP webpage header

Then GoodCorp identified a potential Local File Inclusion (LFI) vulnerability on the "memory-planner.php" page. This vulnerability allowed us to upload a malicious PHP script, demonstrating the potential for attackers to gain unauthorized access to the system.

We identified another instance of LFI on the same page, but with a partial mitigation – it only allowed uploading JPG files. However, this limited protection can be bypassed by simply renaming a malicious PHP script with a JPG extension (e.g., "mypicture.php.jpg").

It's crucial to address such vulnerabilities to prevent attackers from compromising the system. These strategies could include:

- Validating all user-uploaded files to ensure they are legitimate image files.
- Restricting file uploads to specific directories with limited access.

Check the Vulnerability for technical details	Local file inclusion (LFI) vulnerability on the Memory-Planner.php page Local file inclusion (LFI) vulnerability on the Memory-Planner.php page “Choose your location” section

GoodCorp also discovered critical vulnerabilities on the Rekall web application that can lead to data breaches, unauthorized access, and system compromise:

- **SQL Injection on Login.php page**: The Login.php page was susceptible to an SQL Injection attack. This means a malicious actor could potentially inject malicious code into the login form, which can then be executed by the database server.
- **Command Injection on Networking.php page**: The Networking.php page was vulnerable to a Command Injection attack. Using this vulnerability in the ‘DNS Check’ field we were able to discover the contents of the ‘vendors.txt’ file. Located directly beneath the "DNS Check" field, there is the "MX Record Checker, while this feature offered some safeguards against basic attacks, it was ultimately compromised through a more sophisticated approach.
- **PHP Injection (in URL) on souvenit.php page**: Souvenit.php page interacts with user input submitted through a URL parameter. The vulnerability lies in how the souvenit.php page processes this user input. The application doesn't properly validate and sanitize the URL parameter, so we were able to inject malicious code into the URL. This code was interpreted and executed by the PHP script on the server, leading to security risks.

These vulnerabilities underscore the importance of rigorously validating and sanitizing all user input before processing it. By implementing proper security measures, Rekall can significantly reduce the risk of these attacks and protect their application and user data.

	SQL injection vulnerability on the Login.php page

Check the Vulnerability for technical details	Command injection on Networking.php page DNS check field Command injection on Networking.php page MX record check field PHP injection vulnerability on souvenirs.php page

Command injection vulnerability in Networking.php page (described above) allowed us to gain unauthorized access to usernames on the system. Once we discovered a username ("melina"), we needed to determine the password to access the account. We can use a brute-force attack technique using a tool called Burp Intruder. This involves systematically trying different passwords from a large list against the login page until a successful login will be achieved.

Check the Vulnerability for technical details	Brute force attacks vulnerability on Login.php page

The Rekall website appears to be vulnerable to session hijacking due to predictable session IDs. This vulnerability lies in how these session IDs are generated. Because the format is predictable, we were able to exploit this issue using Burp Repeater. By predicting valid session IDs, we could impersonate a legitimate user and gain unauthorized access to their account and potentially their private information.

Check the Vulnerability for technical details	Session management vulnerability on Admin Legal Documents page

Our testing identified a directory traversal vulnerability within the Rekall web application. This vulnerability, also known as path traversal, is a back-end component issue that allowed us to access unauthorized files and directories on the server. By manipulating the URL structure, we were able to view unintended files that wouldn't normally be accessible through the intended functionality of the application.

Check the Vulnerability for technical details	Directory traversal vulnerability on disclaimer.php page

By addressing these web application vulnerabilities and implementing proper security measures, Rekall can significantly reduce the risk of unauthorized access and malicious attacks.

Security Assessment of Rekall's Linux Servers

On the second day of our assessment, we conducted penetration testing on the Linux servers.

Using Open-Source Intelligence (OSINT) tools, GoodCorp successfully retrieved the WHOIS information for "totalrekall.xyz." which revealed some potential “sensitive” information. This information can be valuable for malicious attackers. By exploiting publicly available data, attackers could use it for further network scanning and vulnerability identification.

Check the Vulnerability for technical details	Open source exposed data

The testing identified potential exposure of sensitive information in two locations:

- **Domain Name System (DNS) TXT Records**: A query using nslookup identified data within the TXT records for "totalrekall.xyz" that may be sensitive. Further analysis is required to determine the exact nature of this information and its potential impact.
- **Public SSL Certificate:** An examination of the publicly available SSL certificate for "totalrekall.xyz" on crt.sh revealed potential security concerns. A more in-depth analysis is recommended to understand the specific issue and recommend mitigation strategies.

Check the Vulnerability for technical details	Potential Sensitive Information exposed in Domain Name System (DNS) Text (TXT) records Public certificate potential sensitive data exposure

Our initial network reconnaissance involved a combined effort using Nmap and Zenmap:

- **Nmap**: We utilized Nmap to perform a comprehensive port scan of Rekall's internal network. This scan identified several open ports, which could indicate potential vulnerabilities.
- **Zenmap**: Zenmap served as a complementary tool, specifically searching for outdated or vulnerable services running on the identified open ports. This information helps us prioritize further investigation based on potential exploitability.

The combined use of Nmap and Zenmap revealed several open ports on Rekall's network. These open ports are potential entry points for attackers and warrant further investigation to assess their associated vulnerabilities.

Check the Vulnerability for technical details	Several ports open revealing potential vulnerabilities

Following up on the Zenmap scan, GoodCorp used Nessus to perform a vulnerability scan on a specific host identified earlier (192.168.13.12). This scan revealed a critical vulnerability within the Apache Struts framework (CVE-2017-5638: RCE in Jakarta Multipart Parser).

The identified vulnerability, referenced as CVE-2017-5638, is a Remote Code Execution (RCE) flaw within the Jakarta Multipart parser used by specific versions of Apache Struts (2.3.5-2.3.31 or 2.5.x before 2.5.10.1). This vulnerability could potentially allow attackers to execute arbitrary code on the affected system, granting them unauthorized access and potentially leading to data theft or compromise.

Check the Vulnerability for technical details Note to TA: There is only one vulnerability related to Struts, while the CTF scenario involved finding two flags I created two to prove that the flag has been found.	Apache Struts 2.3.5-2.3.31/2.5x<2.5.10.1 Jakarta Multipart parser RCE vulnerability Struts2 OGNL Injection Vulnerability (CVE-2017-5638)

Referring to the initial scanning with Zenmap, GoodCorp utilized Metasploit to identify potential vulnerabilities on the target machine (192.168.13.10). After evaluating various exploits, a critical

vulnerability in Apache Tomcat (CVE-2017-12617) was discovered. This vulnerability allows for Remote Code Execution (RCE), meaning an attacker could potentially gain full control of the server by executing arbitrary code on it.

We successfully exploited this vulnerability to establish a Meterpreter session on the target machine, solidifying the presence of the RCE flaw.

Check the Vulnerability for technical details	Apache Tomcat JSP Upload Bypass Vulnerability (CVE-2017-12617)

GoodCorp identified potential indicators of a Shellshock vulnerability (CVE-2014-6210) on the machine with the IP address 192.168.13.11. Shellshock is a critical vulnerability in Bash, a common Unix shell interpreter, that allows attackers to execute malicious code through specially crafted environment variables.

Following established testing methodologies, GoodCorp attempted to exploit this potential vulnerability. This attempt was successful, granting GoodCorp a command shell on the target machine. This demonstrates the presence of an exploitable vulnerability on the system.

Check the Vulnerability for technical details	Shellshock Vulnerability (CVE-2014-6210)

The security assessment of the Linux host (192.168.13.11) identified a critical configuration issue: unrestricted access to the /etc/passwd file. While this file doesn't store passwords directly, it contains sensitive information about user accounts on the system. This misconfiguration could be exploited by malicious actors to gather information for further attacks. It's crucial to restrict access to the /etc/passwd file and other sensitive system files. This can be achieved by implementing appropriate file permissions and access control mechanisms.

Check the Vulnerability for technical details	Poor Access Control on /etc/passwd

Our testing identified another critical vulnerability in Drupal 8 core – a Remote Code Execution (RCE) flaw. This vulnerability allows malicious actors to inject and execute arbitrary code on affected Drupal websites remotely. Successful exploitation of this vulnerability could lead to a complete compromise of the underlying server

Check the Vulnerability for technical details Note to TA: There is only one vulnerability related to Drupal, while the CTF scenario involved finding two flags. I created two to prove that the flag has been found.	Remote code execution (RCE) vulnerability in Drupal 8 core versions Drupal remote code execution (RCE) CVE-2019-6340

During the last step of our day 2 penetration test, we identified a potential privilege escalation vulnerability on a Linux host (192.168.13.14). This vulnerability is related to CVE-2019-14287, a flaw in sudo versions before 1.8.28 that could allow bypassing certain access restrictions. This vulnerability can allow an attacker to elevate their privileges to the root user on the system. This would grant them unrestricted access to the system and its data, posing a significant security risk for Rekall.

Check the Vulnerability for technical details	Privilege Escalation using sudo vulnerability CVE-2019-14287

Security Assessment of Rekall's Windows Servers

On the third day of our assessment, we conducted penetration testing on the Windows servers.

As part of the initial reconnaissance phase, GoodCorp investigated the public Totalrekall GitHub repository to identify any publicly available information relevant to the penetration testing. During this process, we discovered a potential security issue: a hashed user credential was inadvertently stored within the repository.

Check the Vulnerability for technical details	Sensitive data exposed on Total Rekall GitHub repositories

A vulnerability assessment was conducted using Nmap to identify open ports. This scan revealed an open HTTP port on a Windows machine with the IP address 172.22.117.20.
A web browser was used to successfully access the web interface on that machine. Sensitive data was able to be collected on Windows machine. This finding suggests a potential lack of authentication controls for accessing this interface. An attacker could potentially exploit this vulnerability to gain unauthorized access to sensitive information.

Check the Vulnerability for technical details	Sensitive Data Exposure on Windows machine (172.22.117.20)

During the initial network reconnaissance phase, a vulnerability scan identified that port 21 (FTP) was open on the Windows machine with the IP address 172.22.117.20. Further analysis revealed that anonymous access was allowed on the FTP server. Anonymous access on FTP servers poses a security risk because it allows any user to connect and potentially browse, download, or modify files stored on the server.

Check the Vulnerability for technical details	Anonymous access FTP vulnerability

A port scan identified port 110 (POP3) open on the Windows machine with the IP address 172.22.117.20. Further investigation revealed the presence of a POP3 server running a specific vulnerable version of Seattle Lab Mail (SLMail), vulnerable to buffer overflow attacks. These attacks can be exploited by sending a specially crafted username that exceeds a specific length. A successful exploit could potentially allow an attacker to gain unauthorized access to the system.

Check the Vulnerability for technical details	POP3 server of Seattle Lab Mail (SLMail) vulnerability

Using a previously established connection on the target machine (172.22.117.20), further exploration was conducted to assess potential persistence mechanisms. This process involved examining the system's scheduled tasks. Scheduled tasks can be used by legitimate applications to automate tasks. However, malicious actors can also leverage them to maintain unauthorized access to a system even after initial access is lost.

Check the Vulnerability for technical details	Unnecessary or potentially malicious scheduled tasks on the Windows 10 machine

Using the same previously established connection on the target machine (172.22.117.20) we used a credential dumping tool (kiwi) to extract user credentials. This access allowed us to crack another user's password hash and escalate privileges to a different account.

Check the Vulnerability for technical details	Windows Local Account Credential Dumping Vulnerability (Kiwi Attack)

During testing, a security risk was identified concerning the location of sensitive data on a Windows 10 machine. The 'Public' folder, by default, grants read access to any user on the system. Our investigation revealed the presence of sensitive data within this folder.

Check the Vulnerability for technical details	Sensitive Data in 'C:\Users\Public\Documents' directory

GoodCorp successfully performed credential dumping on a Windows 10 machine within the Rekall CTF environment. This allowed our team to extract credentials for a new user and crack the password.

Check the Vulnerability for technical details	Credential Dumping on Windows 10 machine

Since we had dumped domain credentials, cracked their hashes, and established persistence on one machine, we tried to expand our access on the network. Now that we had a new set of credentials, we could test its access by spraying the credentials across the network. Once the scan got to 172.22.117.10, we saw a successful logon. This is the IP address of WINDC01, meaning the credentials we cracked were highly privileged and we now have access to a DC. Now that we knew we had credentials that could access the DC, we performed lateral movement, which is the act of moving through the network from one machine to another.
We successfully performed lateral movement to WINDC01 by using WMI via Metasploit.

Check the Vulnerability for technical details	Lateral Movement from Windows10 to WINDC01

During our authorized penetration test, we successfully established a privileged connection to a Domain Controller (WINDC01) using a SYSTEM-level Meterpreter shell. This simulated a potential worst-case scenario where an attacker gains extensive access to the domain.
To demonstrate this potential compromise, we used DCSync, a commonly used legitimate administrative tool for domain management. DCSync can also be misused to extract password hashes, which we simulated in this test.
Our objective within the penetration test was to simulate a real-world attacker and assess the domain's security posture. By successfully retrieving a simulated Administrator password hash, we were able to demonstrate the potential consequences of a successful domain compromise.

Check the Vulnerability for technical details	Credential Access on the domain controller WINDC01

Note to TA: Some vulnerabilities mentioned above, example "Lateral Movement " or "Credential Access on DC " is not something we should add in the "Summary Vulnerability Overview" section below, those are more likely a consequence of a vulnerability and should be mentioned only in Executive Summary. But, because the CTF scenario involved finding all the flags, I created a vulnerability for each flag to prove that the flag had been found. This way the report is not messed up and much easier to read.

Conclusion

The penetration test was successful in achieving all predefined objectives outlined in the scope of work. This included to exfiltrate any sensitive information within the domain, escalate privileges and compromise several machines.

The penetration test revealed potential security risks across Rekall's web application, Linux and Windows servers. These risks, if exploited, could lead to service disruptions and data loss.

The 'Vulnerability Findings' section below presents a comprehensive analysis of discovered vulnerabilities, along with corresponding mitigation recommendations.

To strengthen Rekall's overall security posture, we recommend prioritizing the immediate remediation of vulnerabilities classified as "critical" and "high." This can be achieved through a combined approach, including revising corporate password policies, implementing employee security awareness training, secure coding practices to prevent common vulnerabilities like SQL injection, cross-site scripting (XSS), command injection, conduct routine security audits, conducting a comprehensive review of running services and open ports and much more.

Summary Vulnerability Overview

Vulnerability	Severity
Reflected Cross Site Scripting (XSS) Vulnerability on the Welcome.php page	Critical
Reflected Cross Site Scripting (XSS) Vulnerability on the Memory-Planner.php page.	Critical
Stored Cross Site Scripting (XSS) Vulnerability on the Comments.php page.	Critical
Sensitive Data Exposure in the HTTP webpage header	Medium
Local file inclusion (LFI) vulnerability on the Memory-Planner.php page	Critical
Local file inclusion (LFI) vulnerability on the Memory-Planner.php page “Choose your location” section	Critical
SQL injection vulnerability on the Login.php page	Critical
User Credentials Exposure on Login.php page	Critical
Sensitive Data Exposure in robot.txt file	Medium
Command injection in Networking.php page DNS check field	Critical
Command injection in Networking.php page MX record check field	Critical
Brute force attacks vulnerability on Login.php page	Critical
PHP injection vulnerability on souvenirs.php page	Critical
Session management vulnerability on Admin Legal Documents page	Critical
Directory traversal vulnerability on disclaimer.php page	Critical
Open source exposed data	Low
Potential Sensitive Information exposed in Domain Name System (DNS) Text (TXT) records.	Low
Public certificate potential sensitive data exposure (found on crt.sh)	Low
Several ports open revealing potential vulnerabilities	High
Remote code execution (RCE) vulnerability in Drupal 8 core versions	Critical
Apache Struts 2.3.5-2.3.31/2.5x<2.5.10.1 Jakarta Multipart parser RCE vulnerability	Critical
Apache Tomcat JSP Upload Bypass Vulnerability (CVE-2017-12617)	Critical
Shellshock Vulnerability (CVE-2014-6210)	Critical
Poor Access Control on /etc/passwd	High
Struts2 OGNL Injection Vulnerability (CVE-2017-5638)	Critical
Drupal remote code execution (RCE) CVE-2019-6340	Critical
Privilege Escalation using sudo vulnerability CVE-2019-14287	Critical
Sensitive data exposed on Totalrekall GitHub repositories	Critical
Sensitive Data Exposure on Windows machine (172.22.117.20)	Critical
Anonymous access FTP vulnerability	Critical
POP3 server of Seattle Lab Mail (SLMail) vulnerability	High
Unnecessary or potentially malicious scheduled tasks on the Windows 10 machine	High

Windows Local Account Credential Dumping Vulnerability (Kiwi Attack)	High
Sensitive Data in 'C:\Users\Public\Documents' directory	Medium
Credential Dumping on Windows 10 machine	Critical
Lateral Movement from Windows10 to WINDC01	Critical
Credential Access on the domain controller WINDC01	Critical

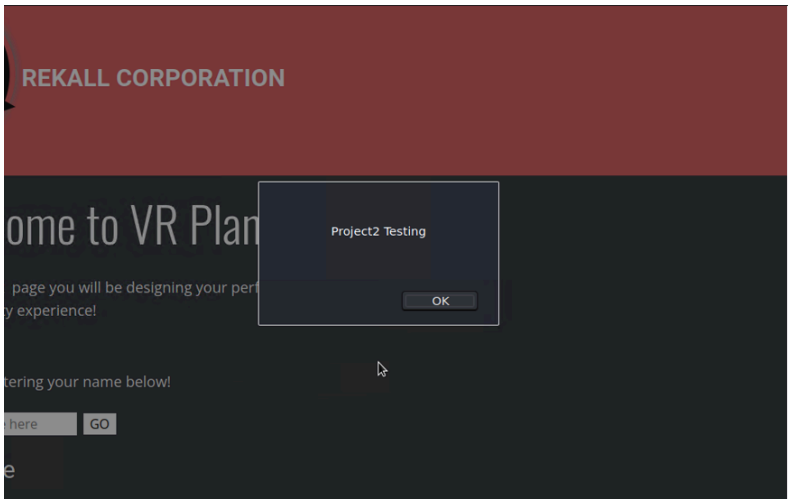
The following summary tables represent an overview of the assessment findings for this penetration test:

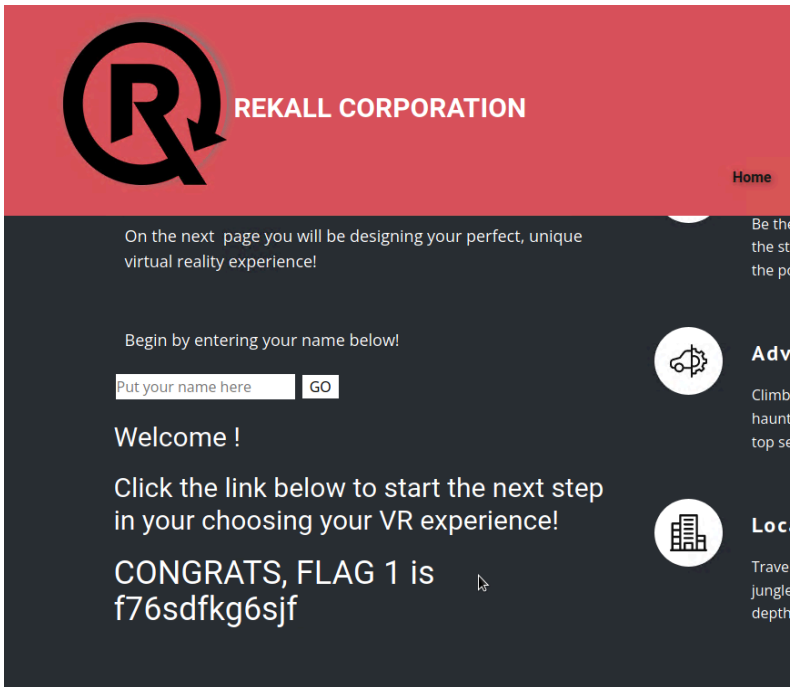
Scan Type	Total
Hosts	192.168.14.35 – totalrekall.xyz 192.168.13.10 - Linux 192.168.13.11 - Linux 192.168.13.12 - Linux 192.168.13.13 - Linux 192.168.13.14 - Linux 192.168.13.1 – Linux 172.22.117.20 – Windows10 172.22.117.10 – Windows Domain Controller 172.22.117.100 – Windows host
Ports	80 (HTTP) 21(FTP), 25(SMTP), 110 (POP3), 135(RPC), 8080 (Http Apache Tomcat), 22 (SSH), 88 (Kerberos).

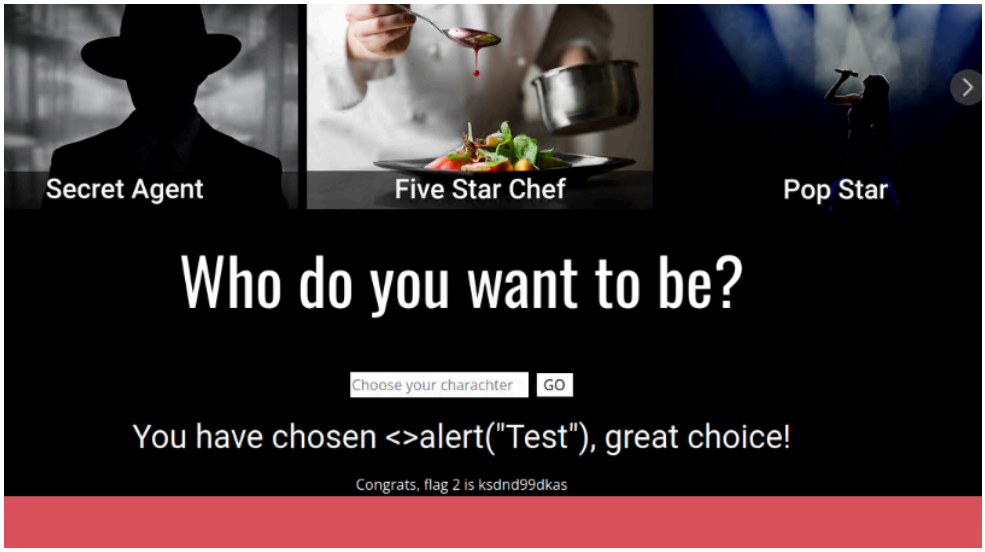
Exploitation Risk	Total
Critical	26
High	5
Medium	3
Low	3

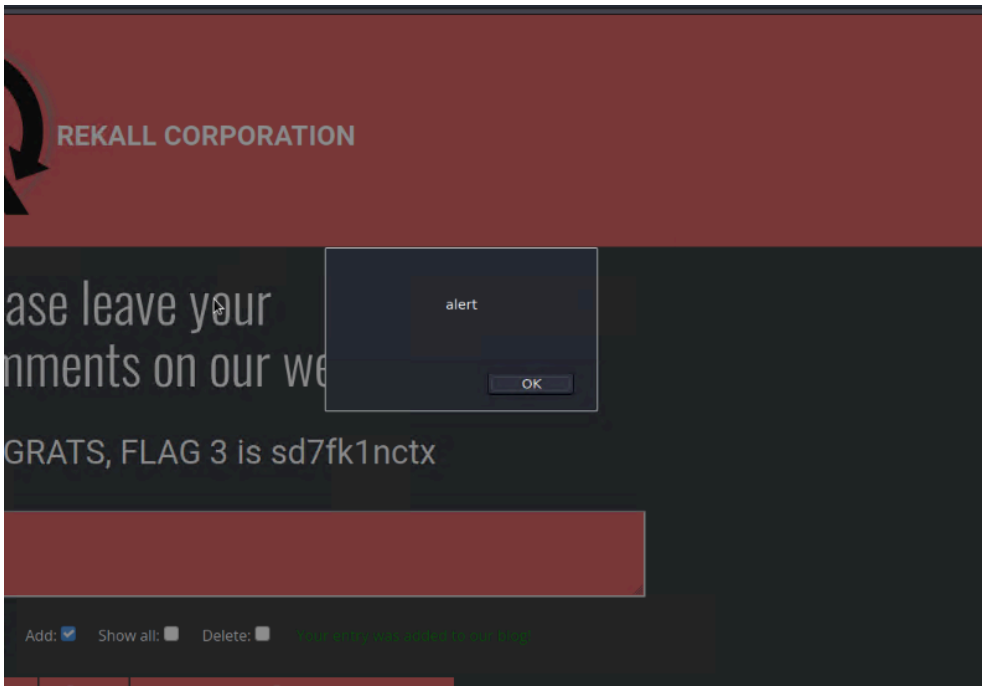
Vulnerability Findings

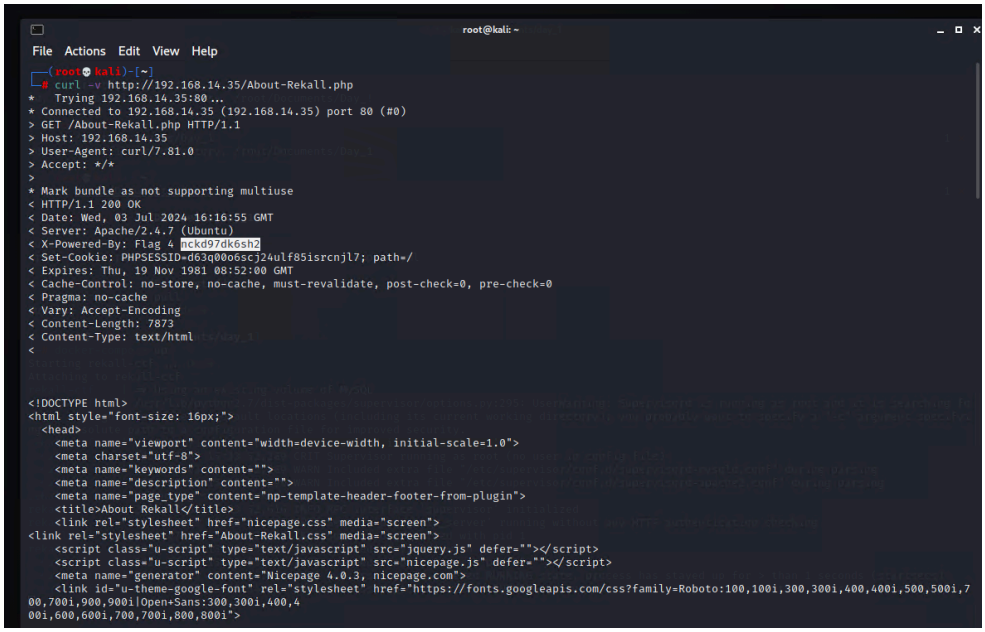
Vulnerabilities Web App

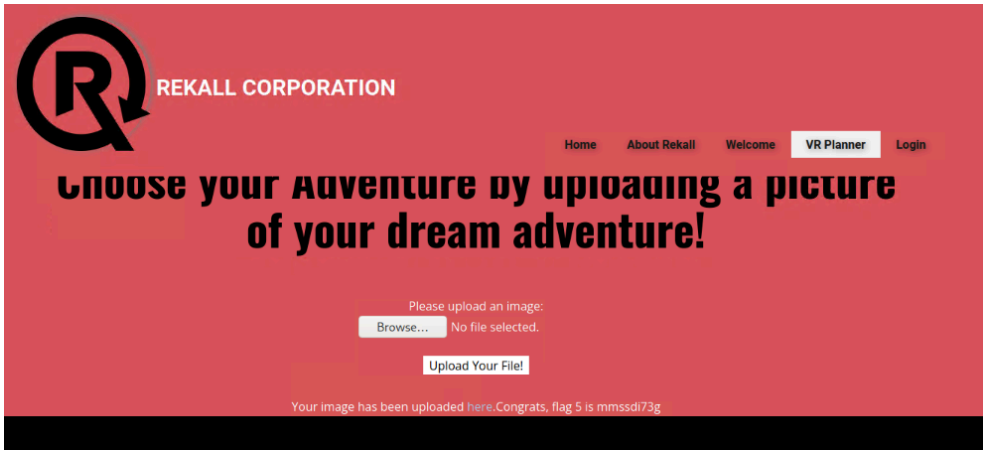
Vulnerability 1	Findings
Title	Reflected Cross Site Scripting (XSS) Vulnerability on the Welcome.php page
Type (Web app / Linux OS / WIndows OS)	**Web app**
Risk Rating	**Critical**
Description	On the Welcome.php page, we were able to successfully insert a reflected XSS payload where it says "Put Your Name Here." example of payload: <script> alert(“Project2 Testing”)</script> The successful payload made a pop up appear.
Images	

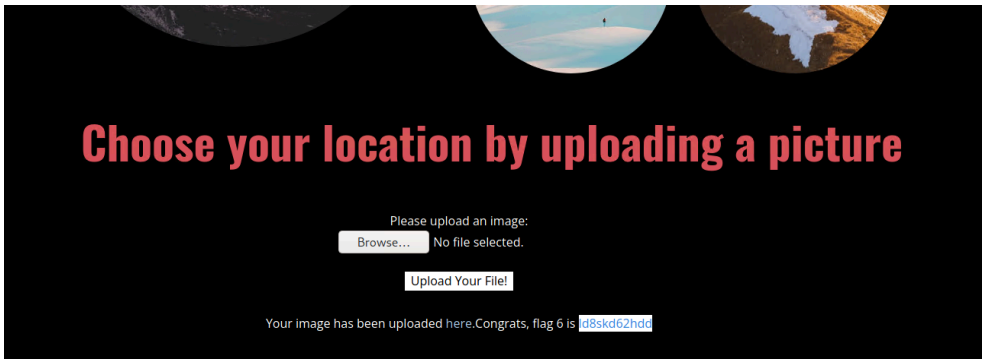
	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Thoroughly validating all user input before it's processed or displayed by the application. • Thoroughly sanitizing all user input removing any potentially malicious characters from the input that could be interpreted as code. • Output Encoding, example: converting any special characters in the output to their HTML entity equivalents. • Regularly scan the web application for vulnerabilities using automated tools or penetration testing services.

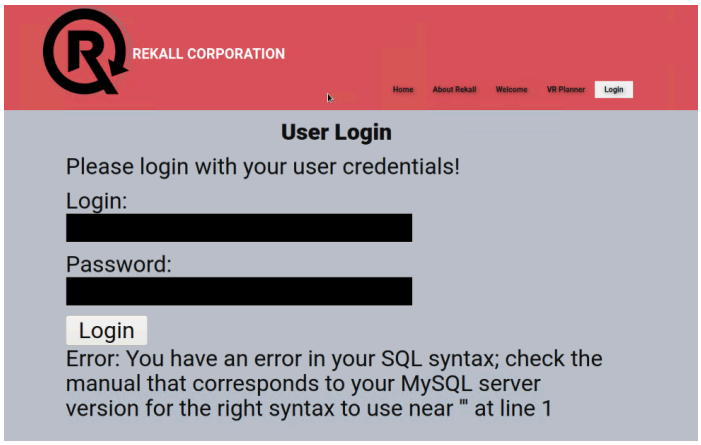
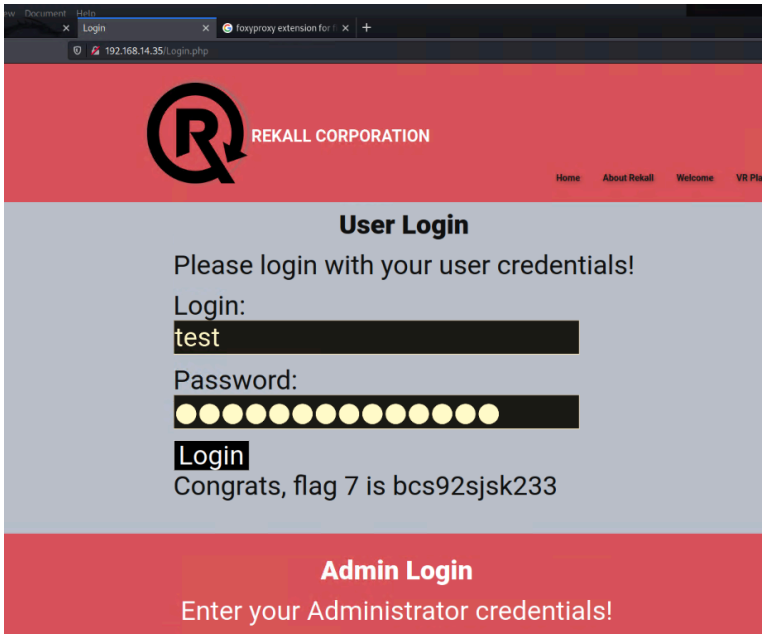
Vulnerability 2	Findings
Title	Reflected Cross Site Scripting (XSS) Vulnerability on the Memory-Planner.php page.
Type (Web app / Linux OS / WIndows OS)	**Web app**
Risk Rating	**Critical**
Description	On the Memory-Planner.php page, we were able to successfully insert a reflected XSS payload in the "Choose Your Character" field. The Memory page field implementing basic input validation code is removing the case insensitive word "script". However, our testing identified a bypass method involving minor script input modifications, such as replacing '<script>' with '<SCRIPscriptT>'. example of payload: <SCRIPTscript>alert(“Test”)</SCRIPscriptT>
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Thoroughly validating all user input before it's processed or displayed by the application. • Thoroughly sanitizing all user input removing any potentially malicious characters from the input that could be interpreted as code. • Output Encoding, example: converting any special characters in the output to their HTML entity equivalents. • Regularly scan the web application for vulnerabilities using automated tools or penetration testing services.

Vulnerability 3	Findings
Title	Stored Cross Site Scripting (XSS) Vulnerability on the Comments.php page.
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	On the Comments.php section (welcome page), we were able to successfully insert a XSS payload in the "Comment" section/field. example of payload: <script> alert(“alert”)</script> The payload is permanently stored on the web server and executed any time the stored data is displayed to a user's browser.
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Thoroughly validating all user input before it's processed or displayed by the application. • Thoroughly sanitizing all user input removing any potentially malicious characters from the input that could be interpreted as code. • Output Encoding, example: converting any special characters in the output to their HTML entity equivalents. • Regularly scan the web application for vulnerabilities using automated tools or penetration testing services.

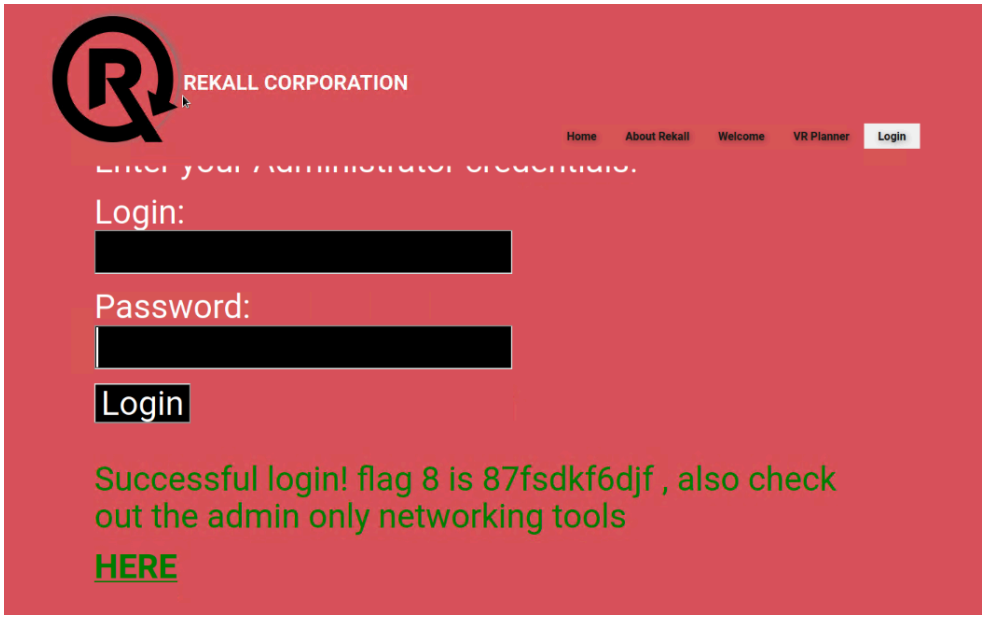
Vulnerability 4	Findings
Title	Sensitive Data Exposure in the HTTP webpage header
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Medium**
Description	Found a potential security issue in the About-Rekall.php webpage. The header information might be leaking sensitive data. Used the following command to retrieve the header information: curl -v http://192.168.14.35/About-Rekall.php Analyzing the output, we identified exposed sensitive data.
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Regularly analyze the HTTP header information retrieved using curl -v or browser developer tools. • Investigate the source of the data in the header. This may involve examining server configuration files, application code, or libraries responsible for generating the header information.

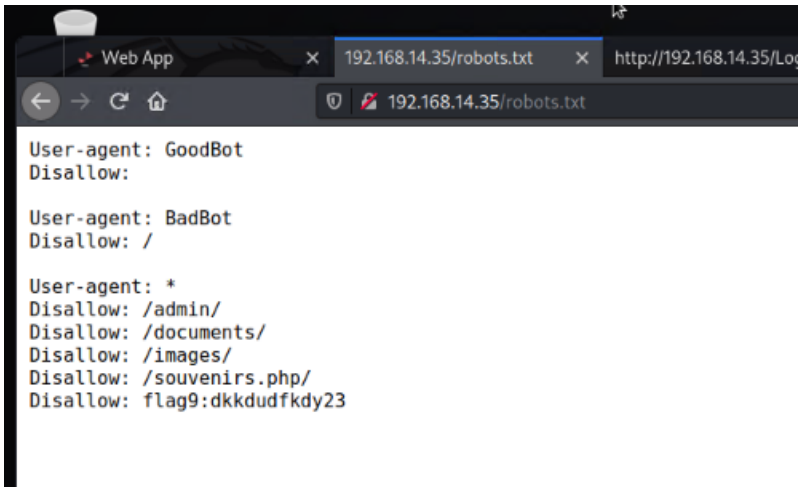
Vulnerability 5	Findings
Title	Local file inclusion (LFI) vulnerability on the Memory-Planner.php page
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	A Local File Inclusion (LFI) vulnerability was identified in the second field on the Memory-Planner.php page. This vulnerability allowed us to upload a PHP script and execute it on the server. An attacker could exploit this vulnerability to gain unauthorized access to sensitive information on the server, such as user data, configuration files, or application code.
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Implement proper input validation to prevent arbitrary file inclusion. • Validate all user-supplied input to ensure it only includes intended data and cannot be manipulated to reference unintended files. • Restrict file uploads to specific allowed file types and locations on the server.

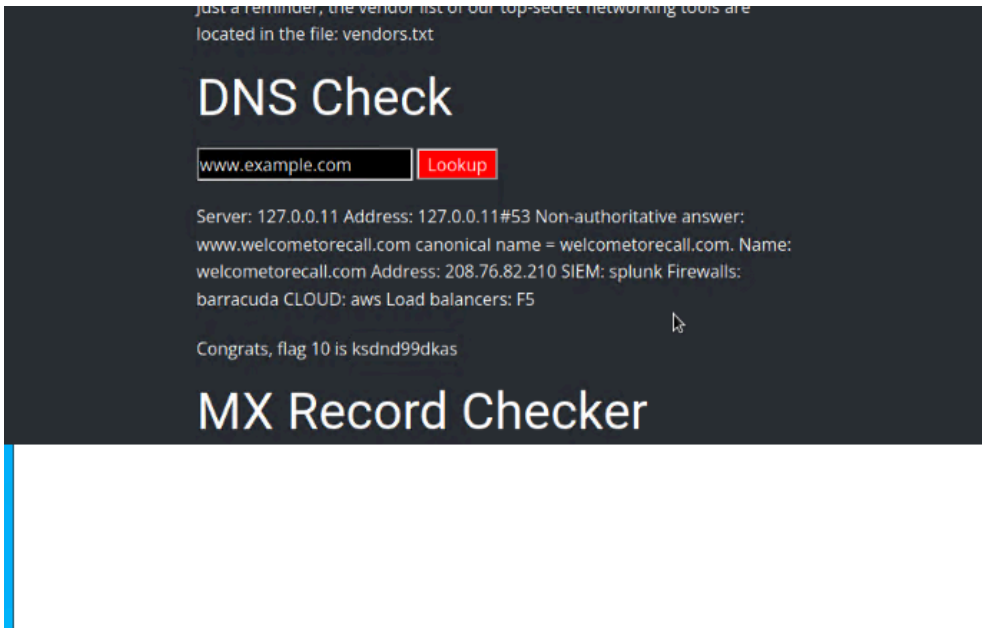
Vulnerability 6	Findings
Title	Local file inclusion (LFI) vulnerability on the Memory-Planner.php page “Choose your location” section
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	The analysis identified another potential Local File Inclusion (LFI) vulnerability on Memory-Planner.php page (“Choose your location” section). While file upload restrictions limited accepted files to JPGs, it was possible to bypass this by appending '.php' to the filename (e.g., location.jpg.php').
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Implement proper input validation to prevent arbitrary file inclusion. • Validate all user-supplied input to ensure it only includes intended data and cannot be manipulated to reference unintended files. • Restrict file uploads to specific allowed file types and locations on the server.

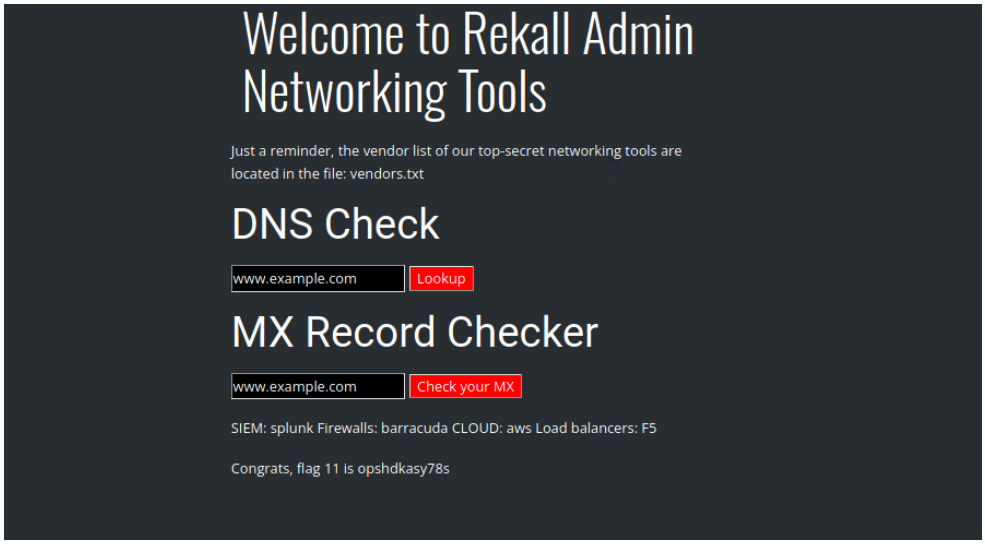
Vulnerability 7	Findings
Title	SQL injection vulnerability on the Login.php page
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	A critical SQL injection vulnerability was identified on the Login.php page. This vulnerability allows an attacker to bypass authentication and potentially gain unauthorized access to the system. The payload was entered in the second field on the user login page. Our initial test demonstrated potential SQL injection vulnerability. Subsequent testing efforts successfully identified an exploitable payload. Example of payload: 'Name or "1=1"'
Images	 

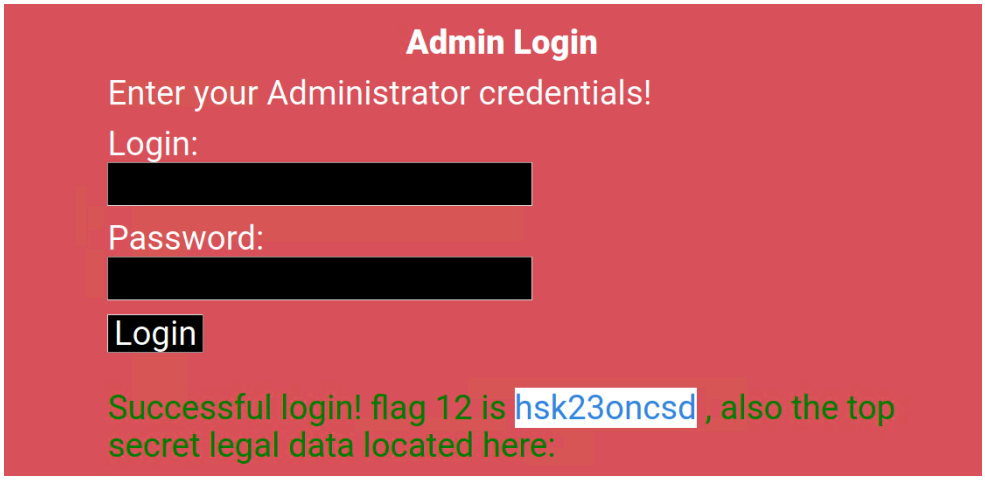
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	● Validate and sanitization all user input before it's processed or used in an SQL query. ● Grant database accounts only the minimum permissions required to perform their designated tasks. This minimizes the potential damage if an attacker gains unauthorized access through an SQLi exploit. ● Regularly scan your web application for vulnerabilities using automated tools or penetration testing services.

Vulnerability 8	Findings
Title	User Credentials Exposure in Login.php page
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	A critical security vulnerability was identified on the Login.php page. User credentials (username and password) were found to be stored in plain text within the page source code.
Images	 `<form action="/Login.php" method="POST">` `<p><label for="login">Login:</label><font color="#DB545A">dougquaid</font> ` `<input type="text" id="login" name="login" size="20" />` `<p><label for="password">Password:</label><font color="#DB545A">kuato</font> ` `<input type="password" id="password" name="password" size="20" /></p>` `<button type="submit" name="form" value="submit" background-color="black">Login</button>` `</form>`
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Implement secure coding practices to prevent similar incidents in the future. • Carefully examine the source code to pinpoint the specific information being exposed and ensure there is no sensitive data that can be publicly viewed

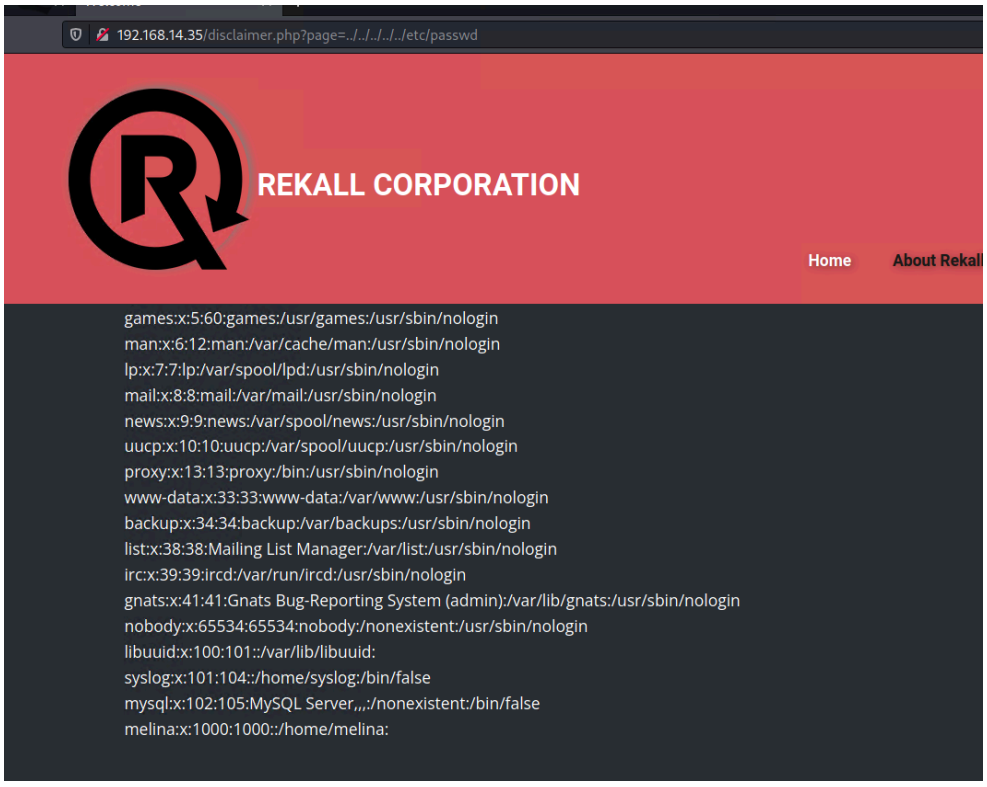
Vulnerability 9	Findings
Title	Sensitive data exposure in robot.txt file
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Medium**
Description	A potential security issue was identified in the robots.txt file. This file is intended to communicate with web crawlers and robots about which parts of a website should not be indexed or crawled. However, in this case, the robots.txt file might be leaking sensitive data.
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Examine the contents of your robots.txt file and remove any sensitive information and could be exploited by attackers. • If certain resources need to be restricted from crawling but can't be removed from robots.txt, consider alternative security measures (example: use access control methods to restrict access in a secure location, not within robots.txt.) • Educate developers and website administrators about the proper use of robots.txt and the importance of avoiding sensitive data exposure. • Schedule periodic reviews of your robots.txt file to ensure it remains up-to-date and doesn't inadvertently expose sensitive information.

Vulnerability 10	Findings
Title	Command injection in Networking.php page DNS check field
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	A critical vulnerability was discovered in the DNS Check tool on the networking.php page. This vulnerability appeared to be a command injection flaw. By exploiting the command injection vulnerability, it was possible to access the contents of the vendors.txt file. Payload: 127.0.0.1 && cat vendors.txt
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Thoroughly validate and sanitize all user input before it's used to construct system commands. • Use safe programming techniques, example: avoid string concatenation for constructing system commands • Grant applications and users only the minimum permissions required to perform their designated tasks. • Regularly scan your web application for vulnerabilities using automated tools or penetration testing services. • Educate developers and other personnel involved in web development about secure coding practices and the dangers of command injection attacks.

Vulnerability 11	Findings
Title	Command injection in Networking.php page MX record check field
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	The MX Record Checker functionality on the networking.php page also revealed a vulnerability. While this vulnerability offered more resistance to command injection basic attacks compared to the one identified in the DNS Check tool, it was still exploitable. The input validation didn't allowed ";" or "&" Payload: 127.0.0.1 | cat vendors.txt
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Thoroughly validate and sanitize all user input before it's used to construct system commands. • Use safe programming techniques, example: avoid string concatenation for constructing system commands • Grant applications and users only the minimum permissions required to perform their designated tasks. • Regularly scan your web application for vulnerabilities using automated tools or penetration testing services. • Educate developers and other personnel involved in web development about secure coding practices and the dangers of command injection attacks.

Vulnerability 12	Findings
Title	Brute force attacks vulnerability on Login.php page
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	Command injection vulnerability in Networking.php page using "DNS check field" and/or "MX Record Checker" allowed us to potentially gain unauthorized access to usernames on the system. Once we discovered a username ("melina"), we needed to determine the password to access the account. We used a brute-force attack technique using a tool called Burp Intruder. This involved systematically trying different passwords from a large list against the login page until a successful login was achieved. **Note**: In this case, brute-forcing wasn't necessary. The password's weakness allowed us to guess it quickly. **Note2**: There is another way to potentially gain unauthorized access to usernames on the system. We also were able to manipulate the URL with the dot-slash method to view the /etc/passwd file on the web application server. (See the screenshots below).
Images	

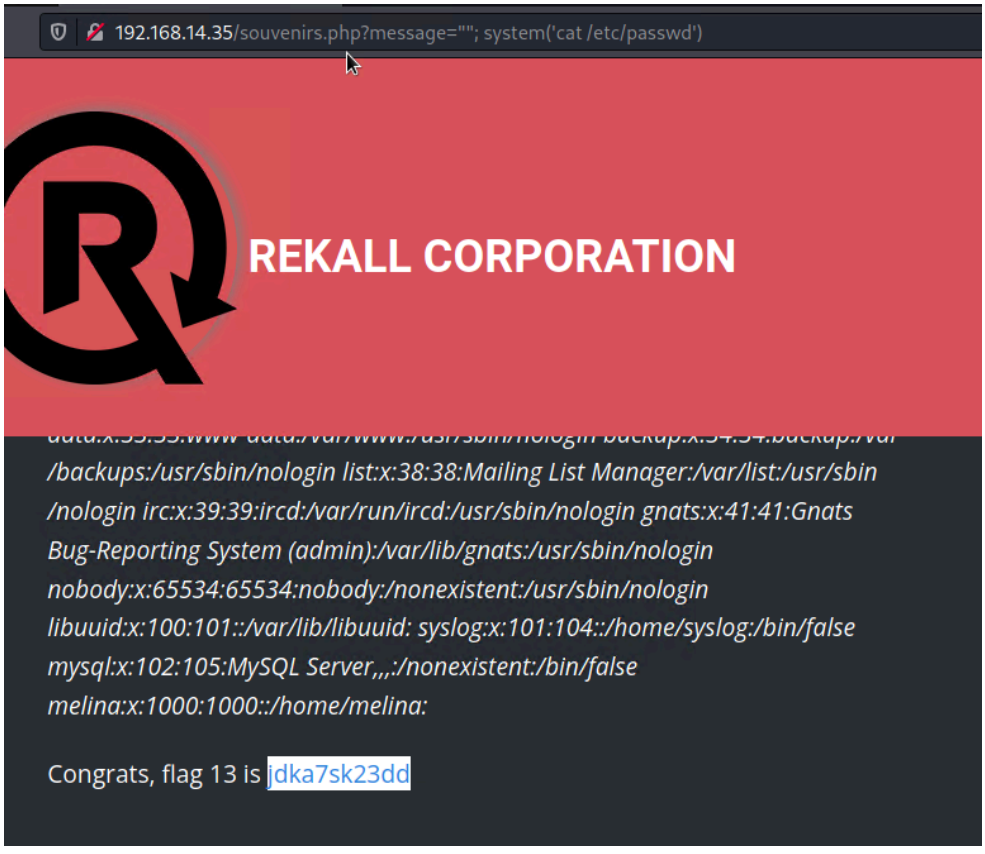
<table>
<tr><td></td><td> 


</td></tr>
<tr><td>Affected Hosts</td><td>192.168.14.35 – totalrekall.xyz</td></tr>
<tr><td>Remediation</td><td>

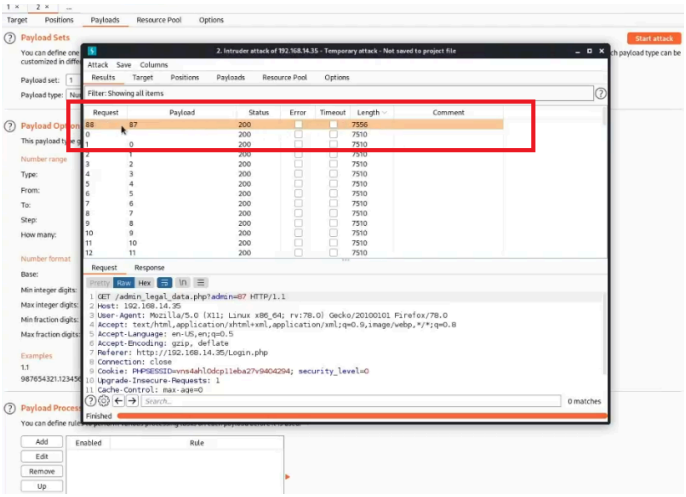
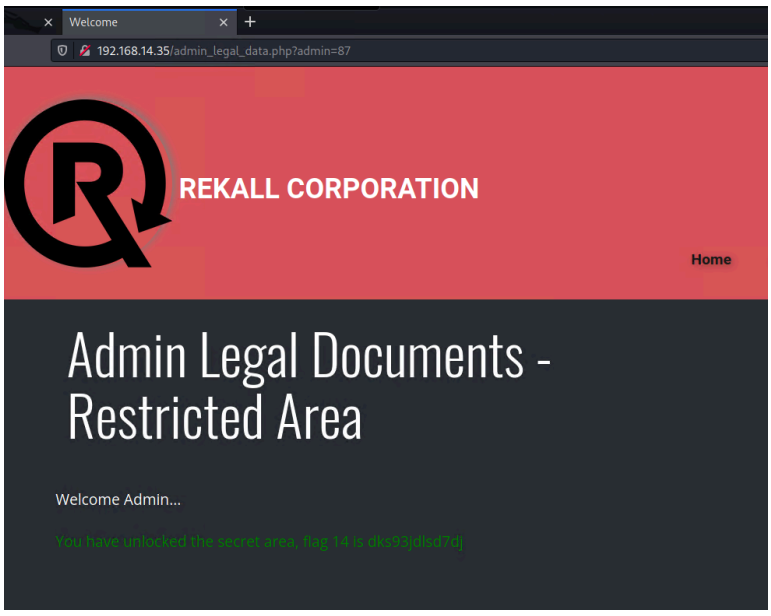
Implement a minimum password length requirement (e.g., 12 characters or more)
Enforce password complexity by requiring a combination of uppercase and lowercase letters, numbers, and symbols.
Prevent users from reusing their past passwords.
Consider enforcing periodic password changes (e.g., every 3-6 months).
Implement a lockout policy that automatically locks an account after a certain number of failed login attempts (e.g., 5-10 attempts)
Implement Multi-Factor Authentication (MFA) as an additional layer of security.

</td></tr>
</table>

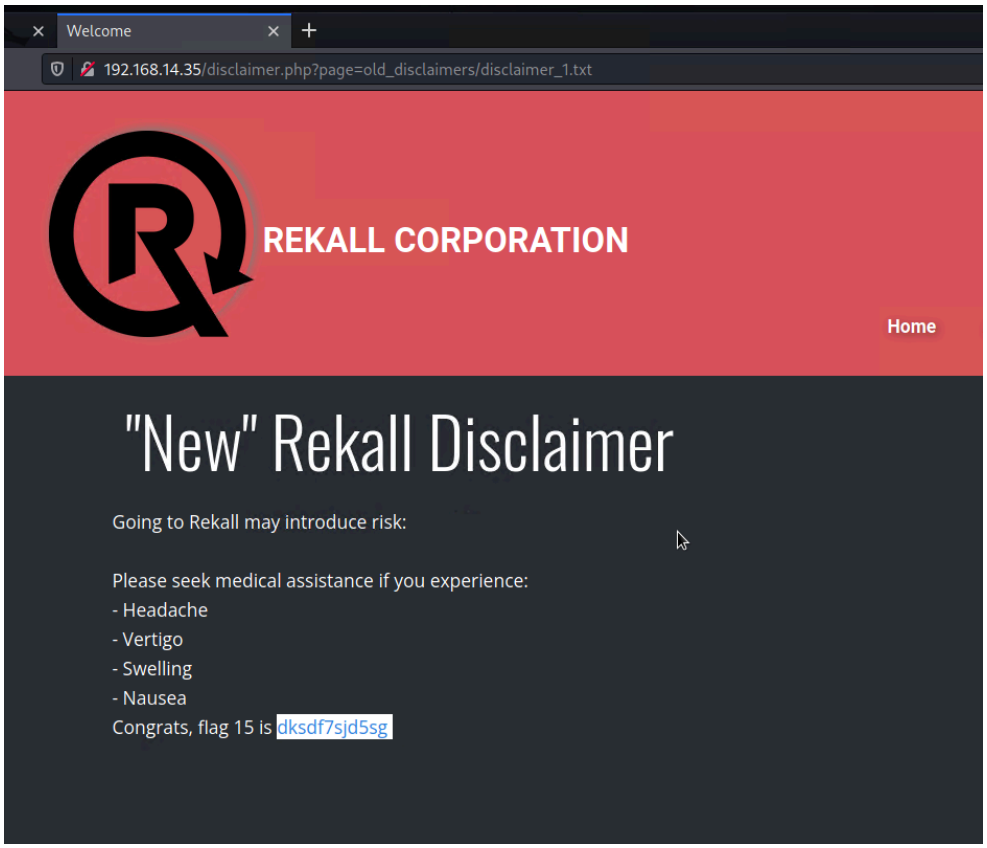
	● Educate employees about password security best practices. ● Consider implementing CAPTCHAs or challenge-response mechanisms during login attempts.

Vulnerability 13	Findings
Title	PHP injection vulnerability in souvenirs.php page
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	The Rekall website's souvenirs.php page appears to be vulnerable to PHP injection. To exploit the PHP injection vulnerability we injected a PHP code to retrieve the contents of /etc/passwd. PHP injection payload: "?message=""; system('cat /etc/passwd')" This payload executed the command and displayed the contents of the /etc/passwd file on the server.
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Validate and sanitize user-supplied input before incorporating it into PHP code. • Remove or encode any malicious characters from the user input before using it in your PHP code. • Consider disabling functions like eval and system in the PHP environment if they are not absolutely necessary for your application.

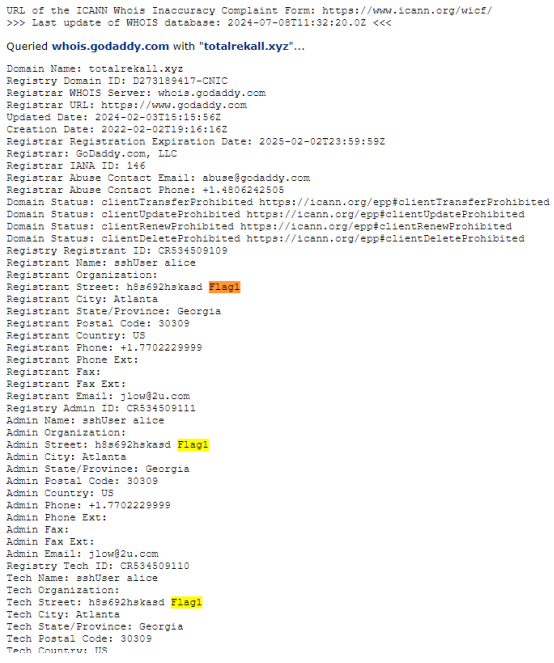
	● Regularly update your PHP version and any relevant frameworks you use to patch known vulnerabilities that attackers might exploit for PHP injection attacks. ● Conduct regular security assessments and penetration testing to identify and address potential PHP injection vulnerabilities in your web applications.

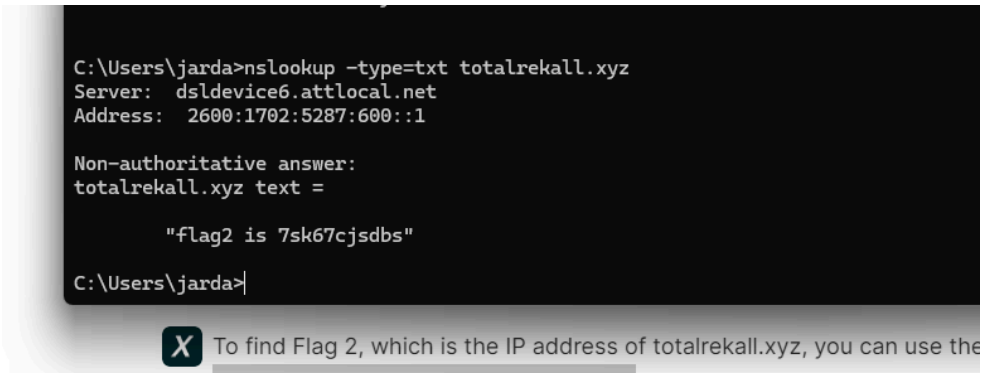
Vulnerability 14	Findings
Title	Session management vulnerability on Admin Legal Documents page
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	The Rekall website appears to be vulnerable to session hijacking due to predictable session IDs. This vulnerability lies in how these session IDs are generated. Because the format is predictable, we were able to exploit this issue using Burp Repeater. By predicting valid session IDs, we could impersonate a legitimate user and gain unauthorized access to their account and potentially their private information.
Images	 
Affected Hosts	192.168.14.35 – totalrekall.xyz

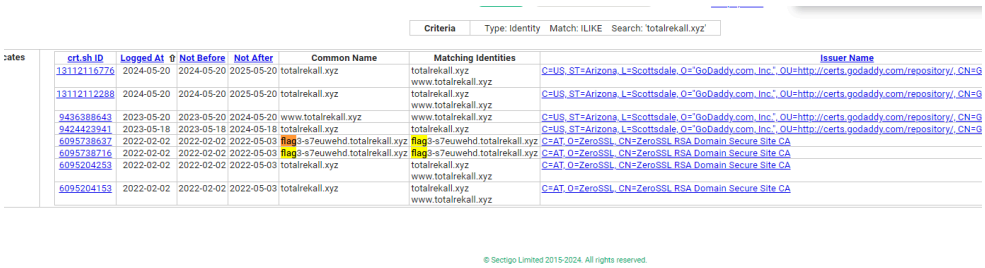
Remediation	● Implement a strong, random session ID generation algorithm that uses a combination of letters, numbers, and symbols. ● Set a reasonable session timeout period for inactive sessions. ● Enforce HTTPS (Hypertext Transfer Protocol Secure) for all communication between the website and users' browsers. ● If using HTTPS, consider setting the "Secure" flag on session cookies. This restricts the cookie's transmission only over secure HTTPS connections.

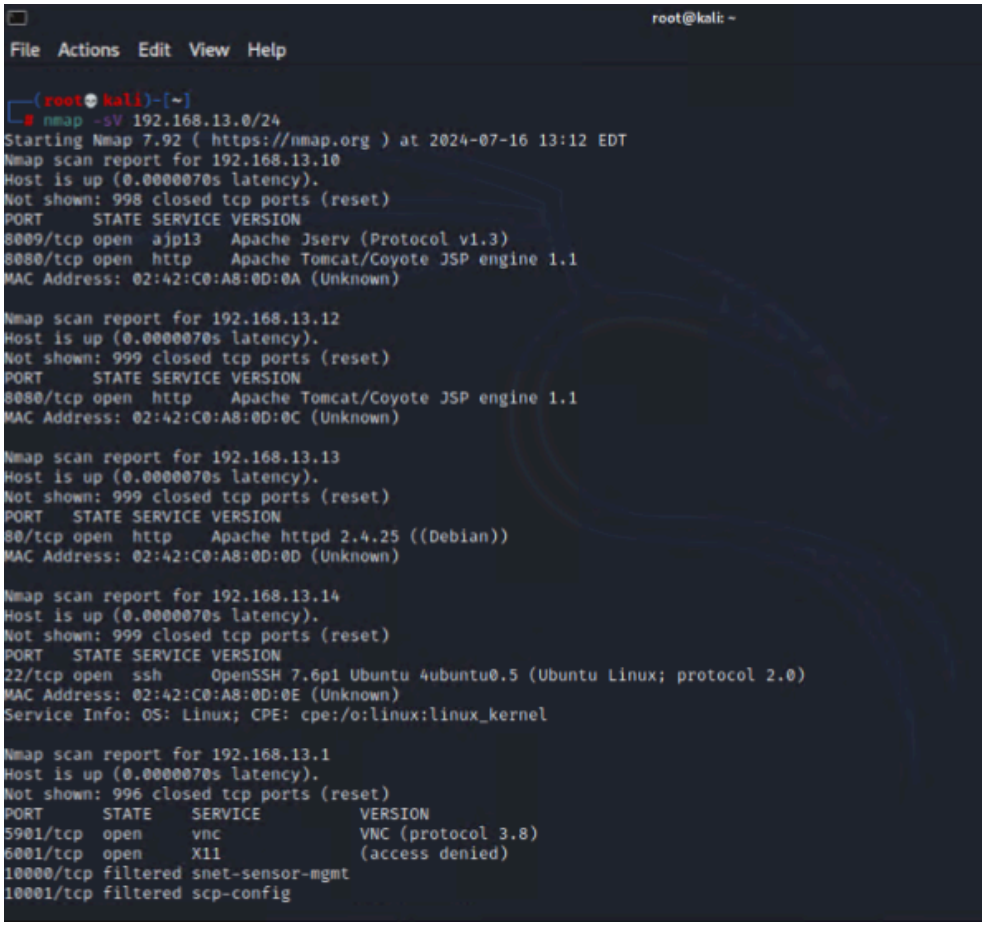
Vulnerability 15	Findings
Title	Directory traversal vulnerability in disclaimer.php page
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	Directory traversal or path traversal is a back-end component vulnerability in which we were able to access files and directories from rekall website. We are able to manipulate the URL to view unintended files on the web application server.
Images	
Affected Hosts	192.168.14.35 – totalrekall.xyz
Remediation	• Validate and sanitize user-supplied input before incorporating it into file system paths. • Remove or encode any malicious characters from the user input before using it in a file path. • Implement access control mechanisms that restrict which files and directories users can access based on their privileges. • Regularly conduct security assessments and penetration testing to identify and address potential directory traversal vulnerabilities in the web applications.

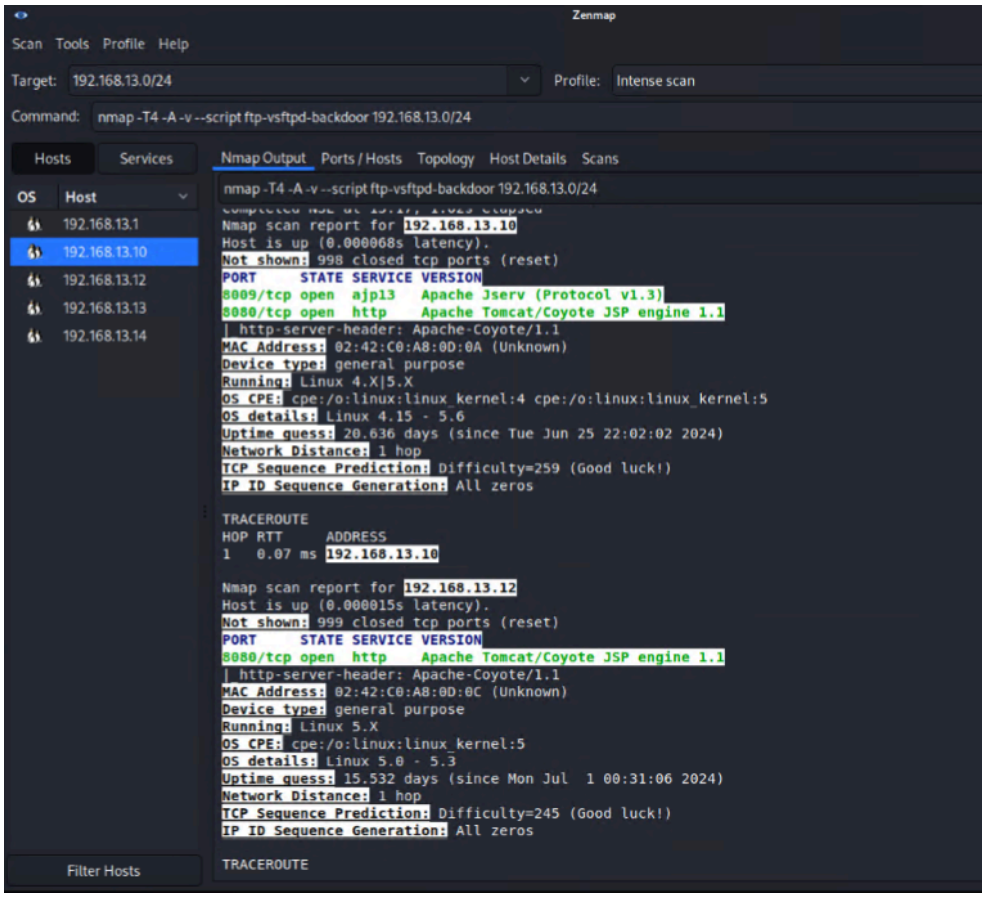
Vulnerabilities Linux Server

Vulnerability 1	Findings
Title	Open source exposed data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Low**
Description	During our investigation, we utilized Open Source Intelligence (OSINT) tools to gather information about the target, "totalrekall.xyz". This included retrieving the WHOIS data, which revealed some potential “sensitive” information. This information can be valuable for both ethical security testing and malicious reconnaissance by adversaries. By exploiting publicly available data, attackers could use it for further network scanning and vulnerability identification.
Images	
Affected Hosts	totalrekall.xyz
Remediation	Redact any sensitive data from publicly available information. Initiate a WHOIS record update process to remove or replace any sensitive details.

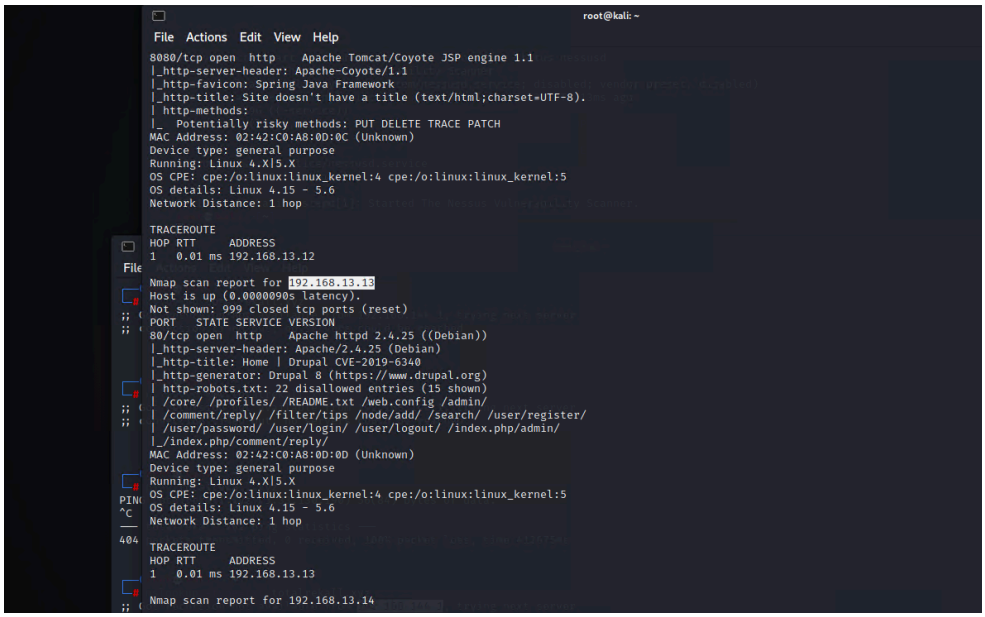
Vulnerability 2	Findings
Title	Potential Sensitive Information exposed in Domain Name System (DNS) Text (TXT) records.
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Low**
Description	A query using nslookup type=txt for totalrekall.xyz revealed data requiring further investigation to determine its sensitivity.
Images	
Affected Hosts	totalrekall.xyz
Remediation	Evaluate the process for creating and managing DNS records to ensure sensitive information isn't inadvertently included in TXT records.

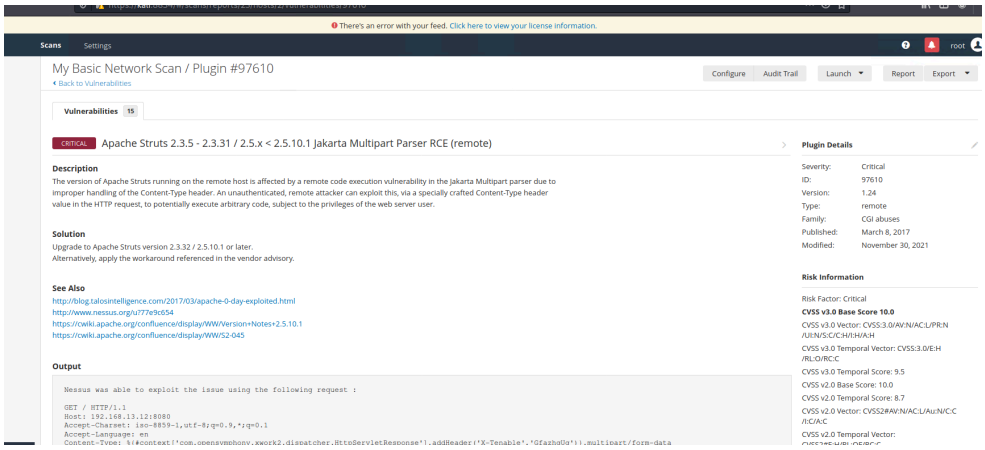
Vulnerability 3	Findings
Title	Public certificate potential sensitive data exposure (found on crt.sh)
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Low**
Description	Publicly available SSL certificate information for totalrekall.xyz indicates a potential security issue. Further analysis is recommended.
Images	
Affected Hosts	totalrekall.xyz
Remediation	• Regularly monitor certificate transparency logs (like crt.sh) to identify any certificates issued. • Ensure that only necessary information is included and potential leaks are minimized.

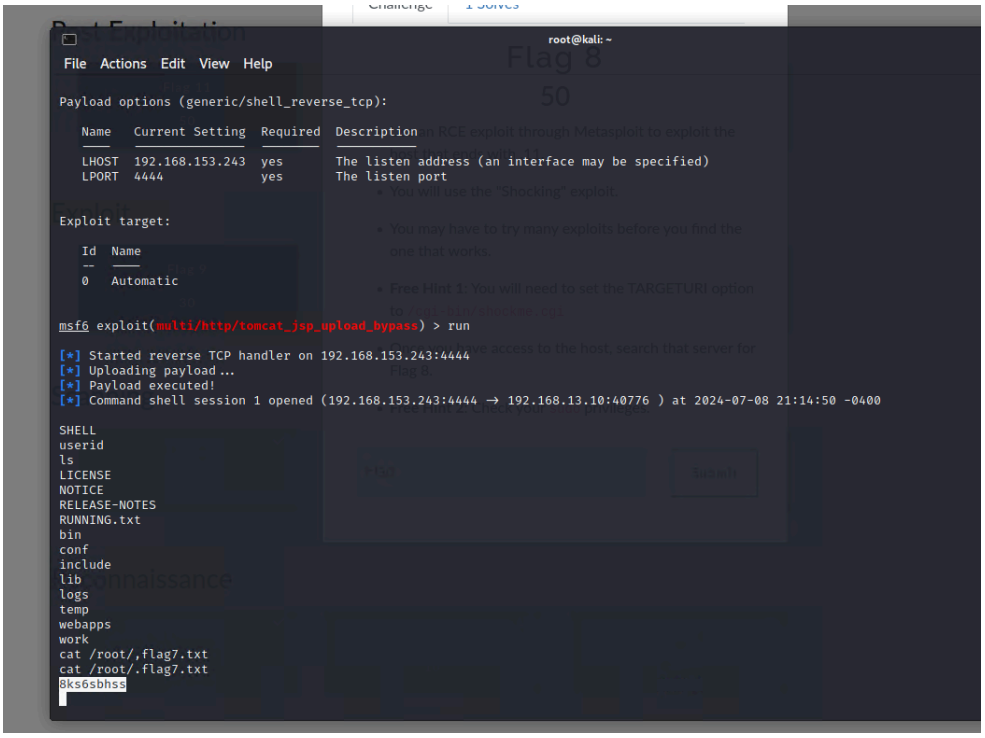
Vulnerability 4	Findings
Title	Several ports open revealing potential vulnerabilities
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**High**
Description	We used Nmap to perform a port scan on the internal network to determine open ports and running services, and Zenmap, to search the machines on the network for any potentially vulnerable services that are outdated or could potentially be abused. Several open ports were discovered with basic nmap scans, revealing potential vulnerabilities throughout Rekall's network.
Images	

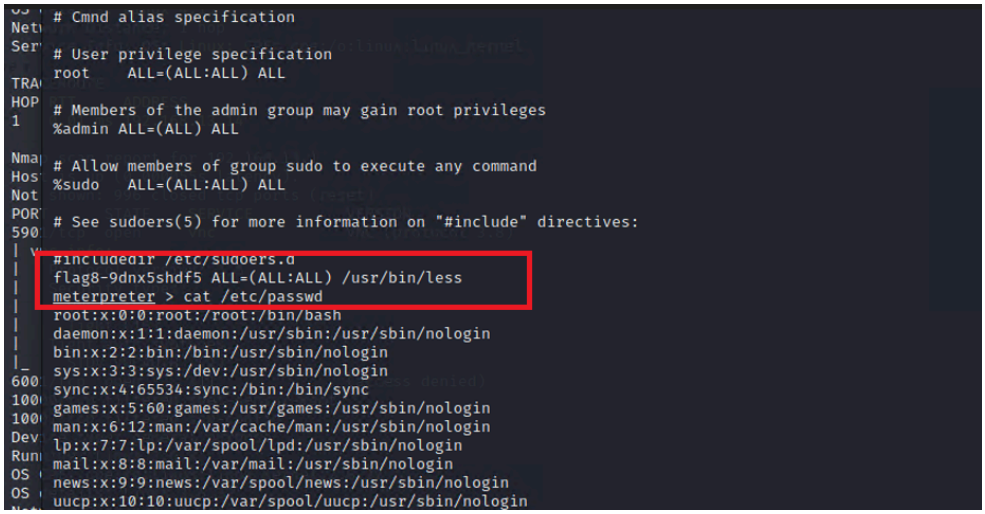


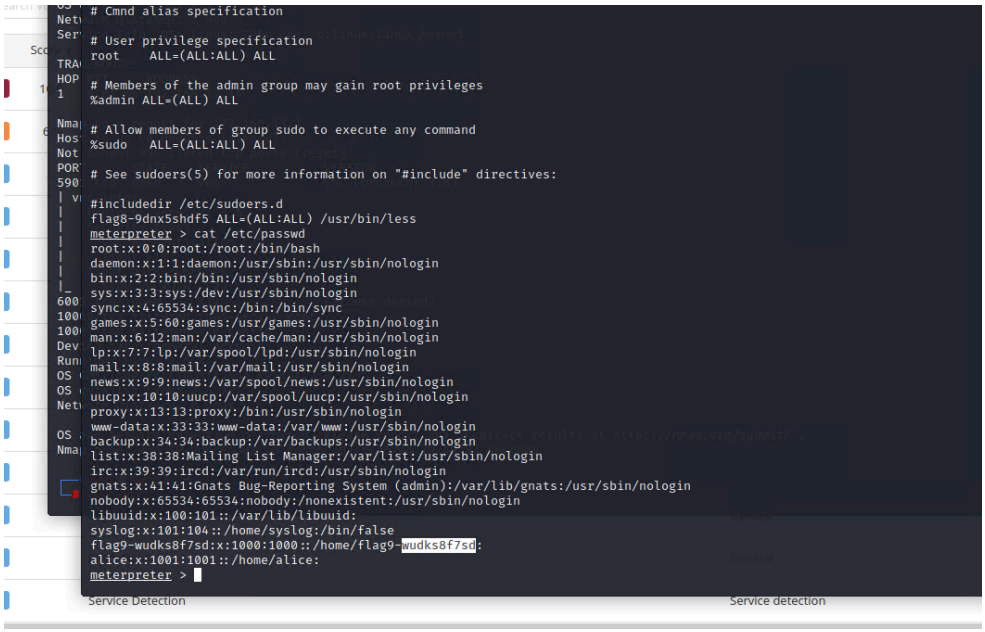
Affected Hosts	192.168.13.10 192.168.13.11 192.168.13.12 192.168.13.13 192.168.13.14
Remediation	• If a service is not actively being used, it's best to disable it and close the corresponding port. This reduces the attack surface and makes the system more secure. • If a service using an open port is deemed necessary but not essential, consider additional security measures, like implementing stricter access controls for the service, such as limiting access to specific IP addresses or users. • Update the service software to the latest version to address any known vulnerabilities. • If a firewall is not already present, consider deploying one to manage incoming and outgoing network traffic. • Regularly review open ports and associated services to ensure their continued necessity. • It's important to conduct periodic vulnerability scans to identify any potential security risks associated with open ports.

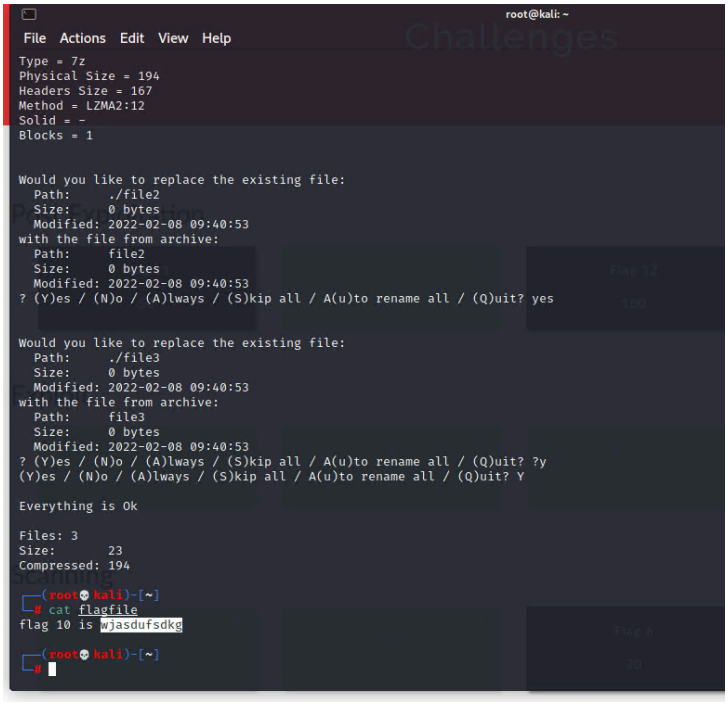
Vulnerability 5	Findings
Title	Remote code execution (RCE) vulnerability in Drupal 8 core versions
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	This vulnerability allows attackers to remotely inject and execute malicious code on Drupal websites. Successful exploitation could result in a complete compromise of the server, including data theft, website defacement, or launching further attacks against other systems.
Images	
Affected Hosts	192.168.13.13
Remediation	• Upgrading Drupal to versions 8.6.10 or 8.5.11 (or later) is the recommended mitigation strategy. These versions contain patches that address the vulnerability. • If immediate patching is not feasible, consider disabling the RESTful Web Services (rest) module or any other web service modules that might be vulnerable.

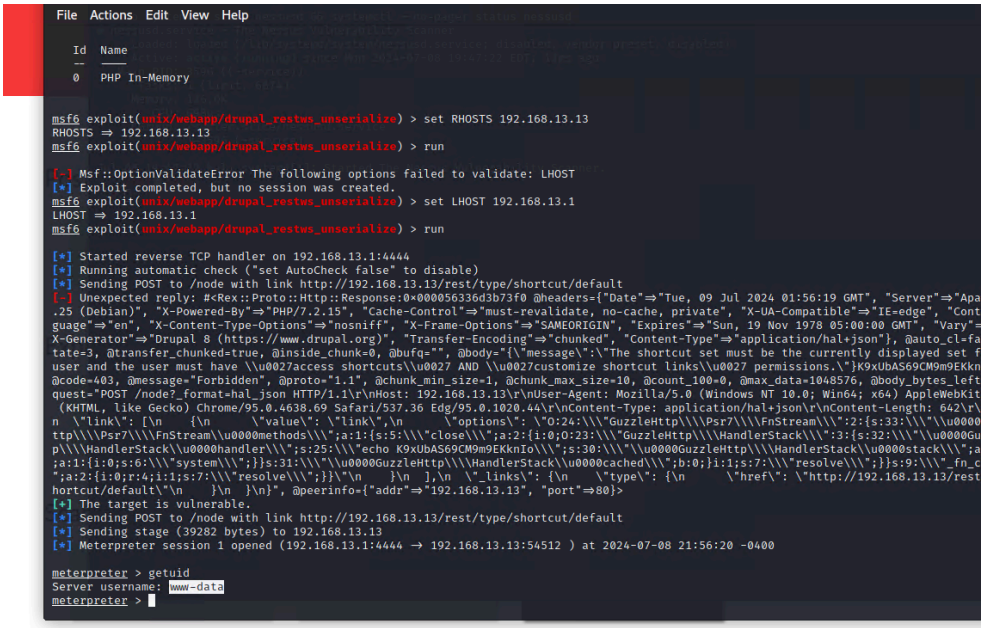
Vulnerability 6	Findings
Title	Apache Struts 2.3.5-2.3.31/2.5x<2.5.10.1 Jakarta Multipart parser RCE vulnerability (Nessus)
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	Nmap and Nessus scans detected a remote code execution (RCE) vulnerability (CVE-2017-5638) in the Jakarta Multipart parser used by Apache Struts 2.3.5-2.3.31 or 2.5.x versions before 2.5.10.1. This vulnerability allows attackers to execute programs, steal source code, or exfiltrate data.
Images	
Affected Hosts	192.168.13.12
Remediation	• Upgrade Apache Struts to a version not affected by the vulnerability • Implement a vulnerability scanning strategy to identify and address potential security issues proactively.

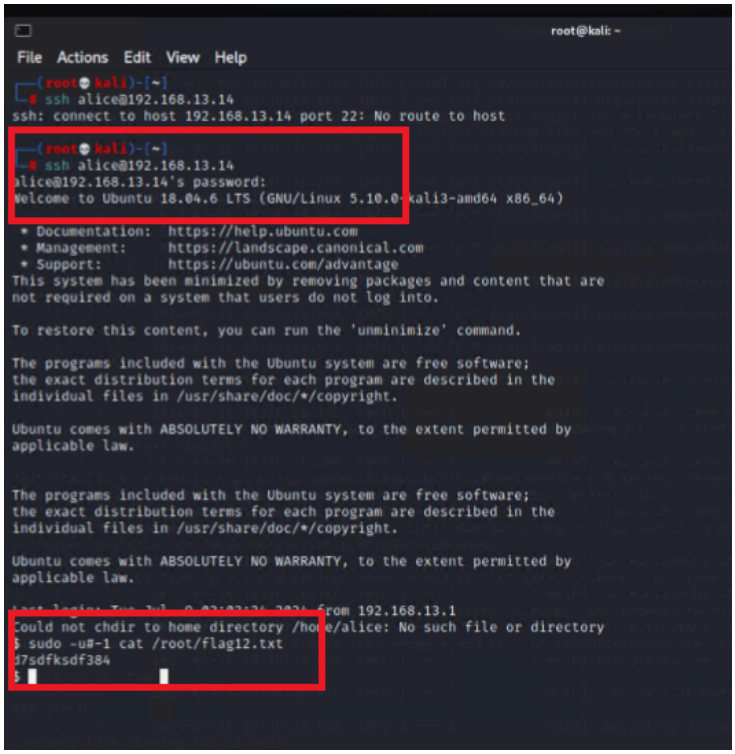
Vulnerability 7	Findings
Title	Apache Tomcat Remote Code Execution (RCE) (CVE-2017-12617)
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	Successfully exploited the vulnerability in Apache Tomcat that allows for remote code execution (RCE). This means an attacker could take control of the server remotely by executing arbitrary code on it.
Images	
Affected Hosts	192.168.13.10
Remediation	• Upgrading Tomcat to a version that is not affected by the vulnerability • Review your Tomcat server configuration and disable any unused functionalities like JSP uploads if not required. • Implement a vulnerability scanning strategy to identify and address potential security issues proactively.

Vulnerability 8	Findings
Title	Shellshock Vulnerability (CVE-2014-6210)
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	Shellshock is a critical vulnerability in Bash, a common Unix shell interpreter. It allows attackers to inject malicious code into seemingly harmless environment variables. When these variables are processed by Bash, the attacker's code can be executed, potentially granting them unauthorized access to the system. We were able to exploit this vulnerability and gained access to the system and access to several sensitive areas of the host, including the sudoers file.
Images	
Affected Hosts	192.168.13.11
Remediation	Patch Bash to a version that is not affected by the vulnerability.

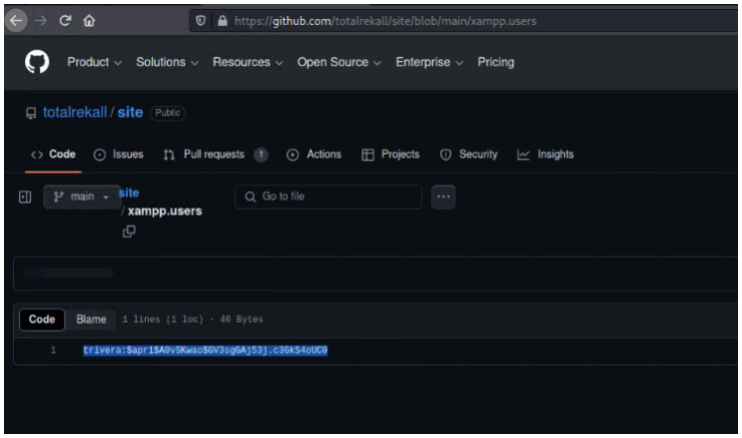
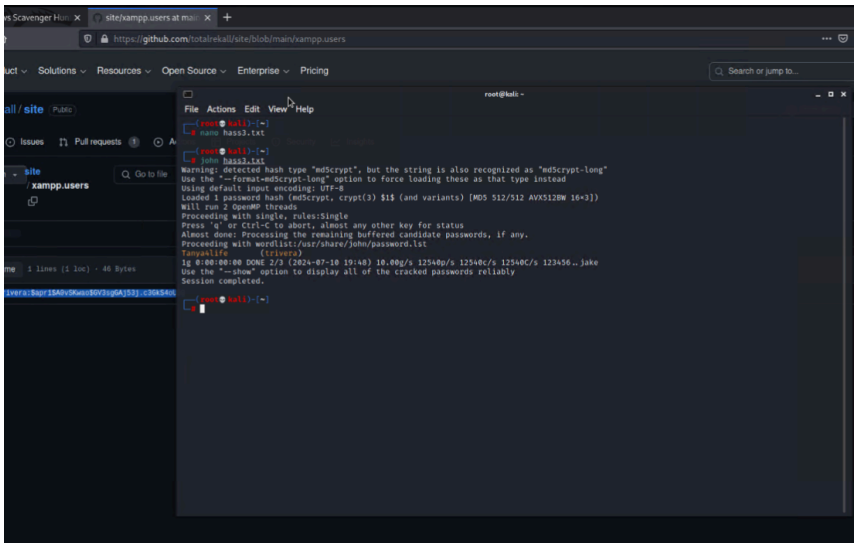
Vulnerability 9	Findings
Title	Poor Access Control on /etc/passwd
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**High**
Description	A security assessment of the Linux host at 192.168.13.11 identified an issue: unrestricted access to the /etc/passwd file. While it doesn't contain passwords, it reveals user accounts, enabling potential credential stuffing attacks.
Images	
Affected Hosts	192.168.13.11
Remediation	• Restrict access to the `/etc/passwd` file. • Regularly audit file permissions to identify and address any anomalies.

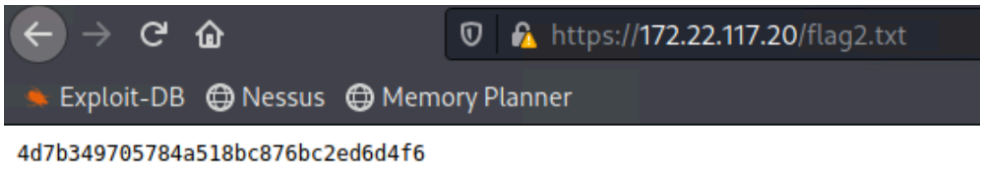
Vulnerability 10	Findings
Title	Struts2 OGNL Injection Vulnerability (CVE-2017-5638)
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	The vulnerability exploited by multi/http/struts2_content_type_ognl in Metasploit it's a remote code execution (RCE) vulnerability in Apache Struts versions 2.3.5-2.3.31 and 2.5.x versions before 2.5.10.1. This vulnerability allows attackers to inject malicious code into an HTTP request header. An attacker can exploit this vulnerability to gain remote code execution on the vulnerable Struts server.
Images	
Affected Hosts	192.168.13.12
Remediation	Upgrade your Apache Struts application to a version not affected by CVE-2017-5638.

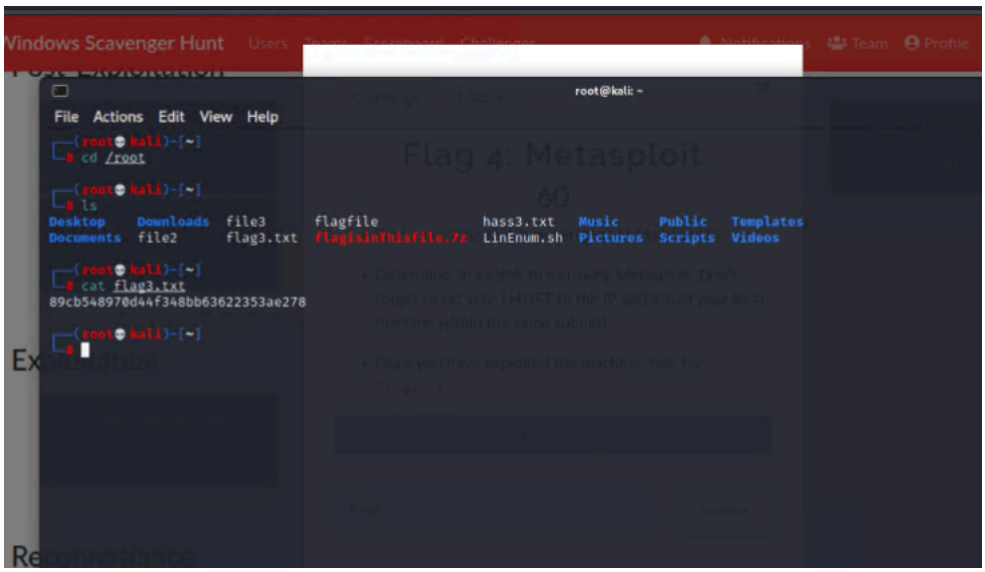
Vulnerability 11	Findings
Title	Drupal remote code execution (RCE) CVE-2019-6340
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	Drupal CVE-2019-6340 is a critical remote code execution (RCE) vulnerability affecting certain versions of Drupal 8 core. An attacker could exploit this vulnerability to inject malicious code into a Drupal website through crafted requests.
Images	
Affected Hosts	192.168.13.13
Remediation	Upgrading Drupal to versions 8.6.10 or 8.5.11 (or later) is the recommended mitigation strategy. These versions contain patches that address the vulnerability. If immediate patching is not feasible, consider disabling the RESTful Web Services (rest) module or any other web service modules that might be vulnerable.

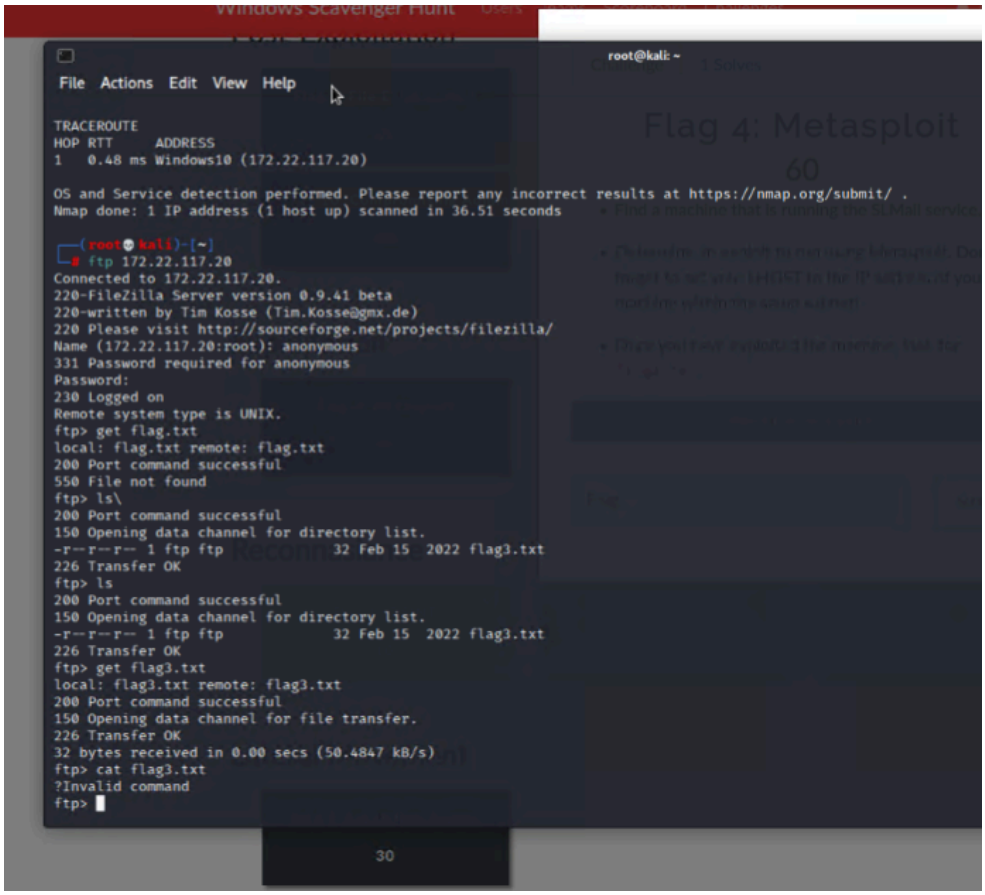
Vulnerability 12	Findings
Title	Privilege Escalation using sudo vulnerability CVE-2019-14287
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	An open-source WHOIS lookup revealed 'alice' as a possible username. We then tried to SSH to the server with that username. Once logged in, we run the command “sudo -u#-1 cat /root/flag12.txt“ to exploit the sudo vulnerability. The command above leverages the CVE-2019-14287 vulnerability to execute the command as the target user. This vulnerability allows an attacker to bypass access restrictions and execute commands as the target user by specifying the user ID -1 or 4294967295.
Images	
Affected Hosts	192.168.13.14
Remediation	• Upgrading sudo to version 1.8.28 or later • Review and tighten sudoers configuration files to limit which users can run sudo and what commands they can execute with sudo privileges. • Implement system monitoring tools to detect any unauthorized attempts to escalate privileges.

Vulnerabilities Windows Server

Vulnerability 1	Findings
Title	Sensitive data exposed on Totalrekall GitHub repositories
Type (Web app / Linux OS / WIndows OS)	Windows OS / GitHub
Risk Rating	**Critical**
Description	During our investigation of the public Totalrekall GitHub repository, we identified a potential security vulnerability involving a hashed user credential.
Images	 
Affected Hosts	Totalrekall github
Remediation	• Conduct a comprehensive review of all repositories to identify and remove any sensitive data that could be publicly accessed. • Implement mandatory password changes for all users and enforce a strong password policy that includes complexity requirements and regular rotation schedules.

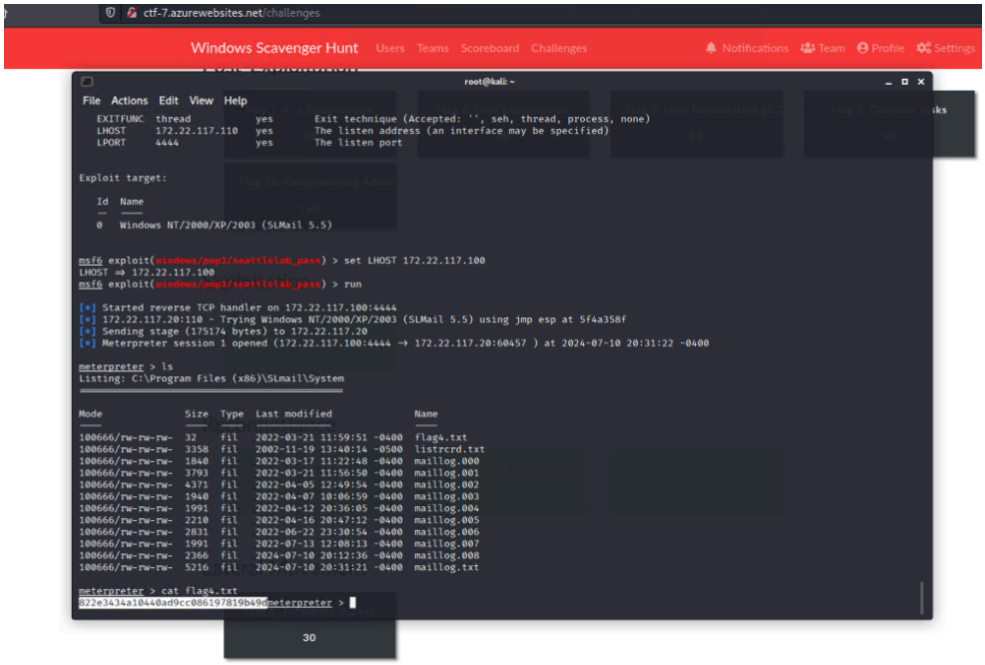
Vulnerability 2	Findings
Title	Sensitive Data Exposure on Windows machine (172.22.117.20)
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Critical**
Description	Sensitive data was able to be collected on Windows machine IP 172.22.117.20.
Images	  4d7b349705784a518bc876bc2ed6d4f6
Affected Hosts	172.22.117.20
Remediation	• Conduct a comprehensive review of all files on windows machine to identify and remove any sensitive data that could accessed. • Implement mandatory password changes for all users and enforce a strong password policy that includes complexity requirements and regular rotation schedules.

Vulnerability 3	Findings
Title	Anonymous access FTP vulnerability
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Critical**
Description	A security assessment of the Windows host at 172.22.117.20 identified an open port 21, allowing anonymous FTP connections. This vulnerability grants unrestricted access to anyone, enabling them to browse (enumerate) or even steal (exfiltrate) files from the server. During this pentesting step, we successfully exploited this vulnerability.
Images	

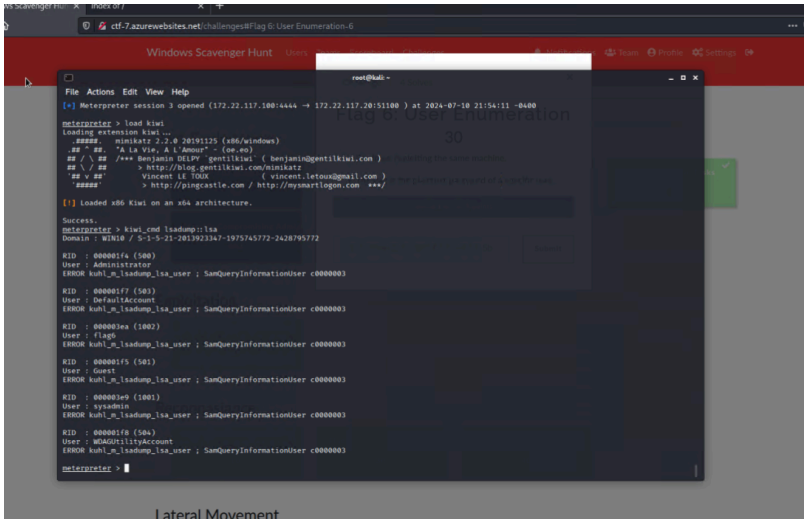
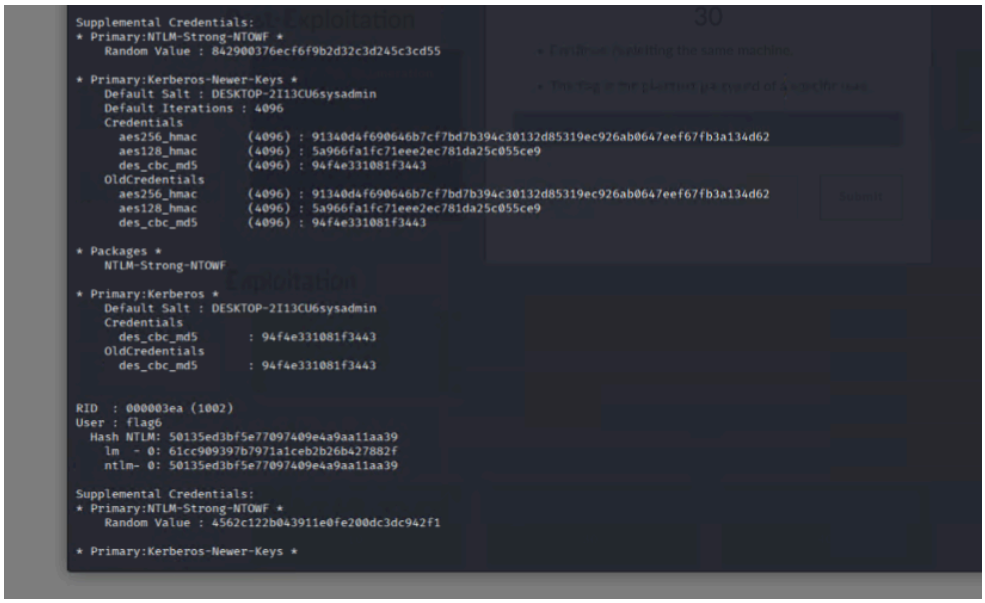
<table>
<tr><td></td><td> 
</td></tr>
<tr><td>Affected Hosts</td><td>172.22.117.20</td></tr>
<tr><td>Remediation</td><td>

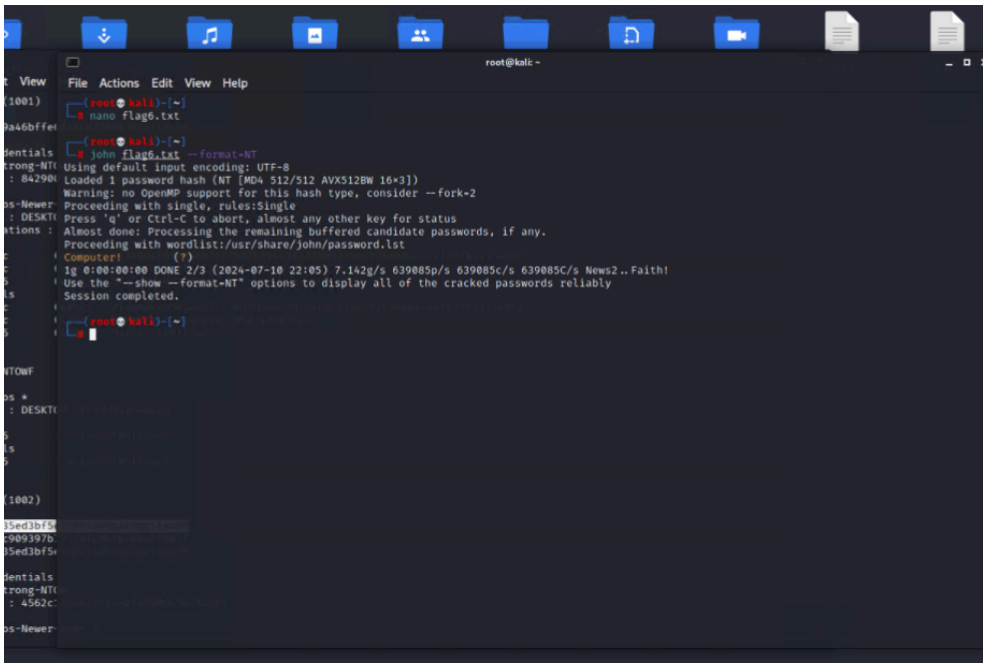
Disable anonymous login to eliminate the risk of unauthorized access through anonymous FTP clients.
Create individual user accounts for authorized users who require access to the FTP server.
Implement ACLs to restrict access for authorized users.
Consider Alternative Secure File Transfer Protocols SFTP

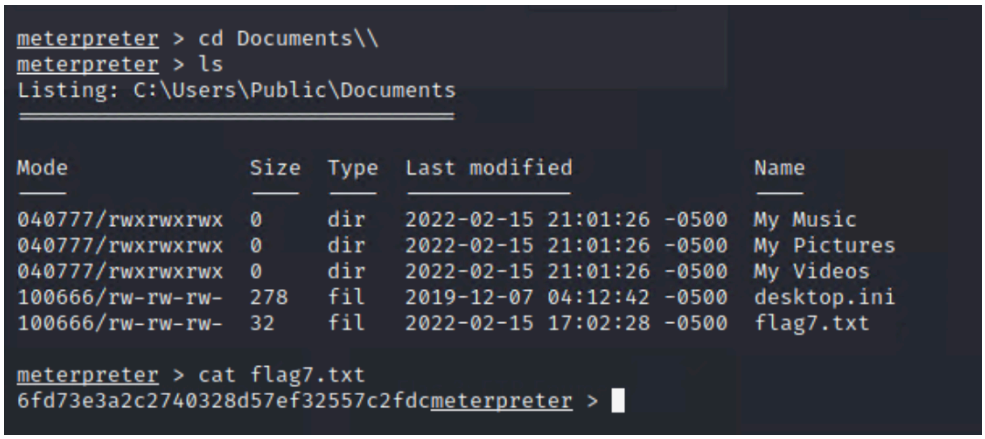
</td></tr>
</table>

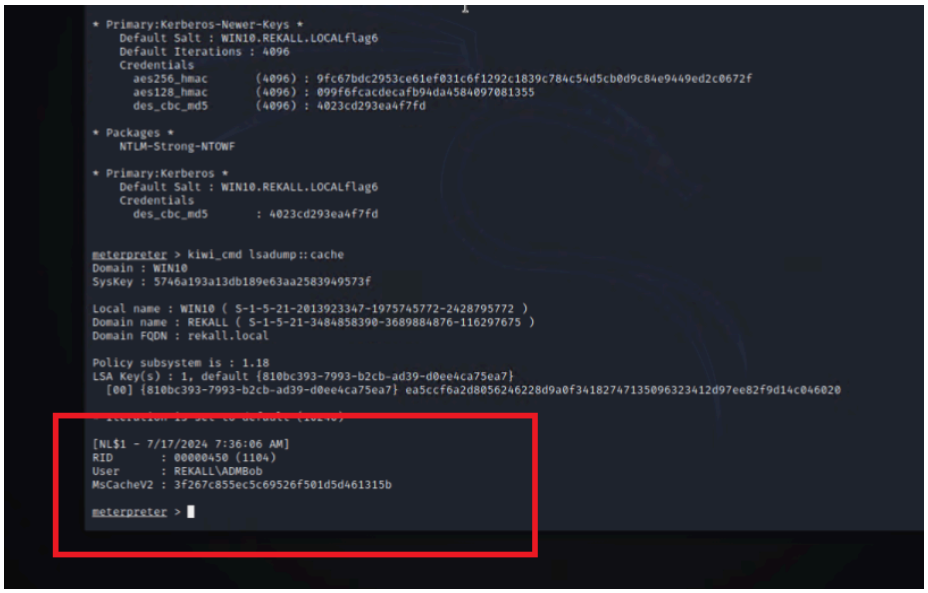
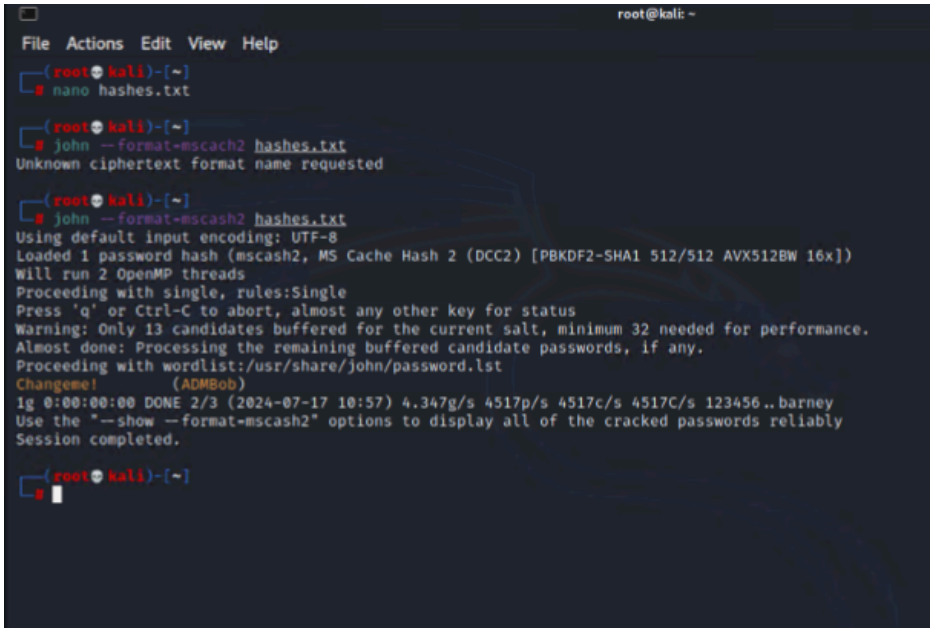
Vulnerability 4	Findings
Title	POP3 server of Seattle Lab Mail (SLMail) vulnerability
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	Seattle Lab Mail (SLMail) version 5.5 suffers from a critical buffer overflow vulnerability in its POP3 server. This flaw allows attackers to remotely execute malicious code on the vulnerable machine simply by sending a username (often mistaken for a password) exceeding a specific length. We were able to successfully exploit this vulnerability through use of windows/pop3/seattlelab_pass exploit within Metasploit which resulted in a successful Meterpreter session.
Images	
Affected Hosts	172.22.117.20
Remediation	Upgrade to a newer version of Seattle Lab Mail

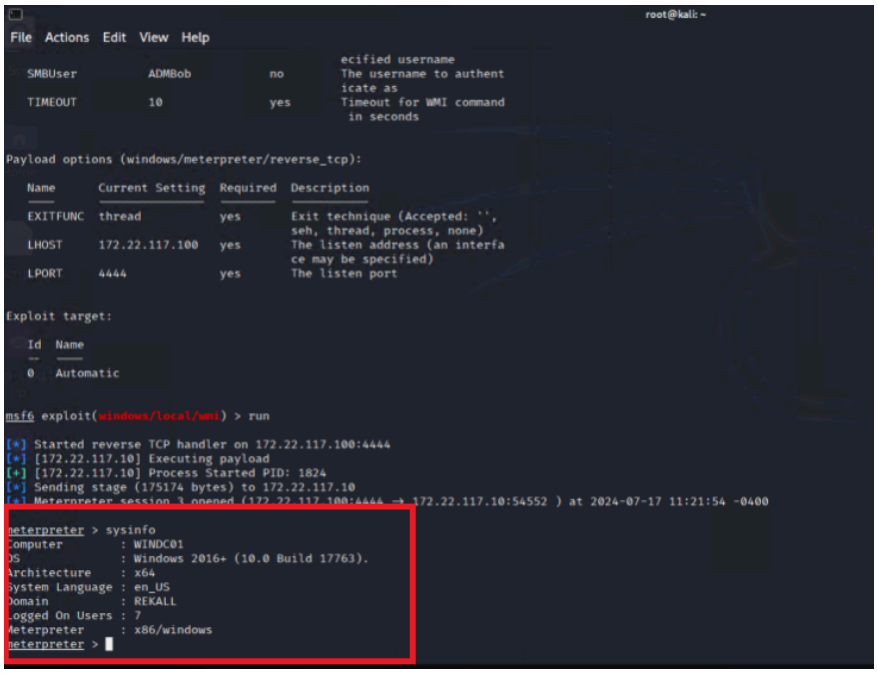
Vulnerability 5	Findings
Title	Unnecessary or potentially malicious scheduled tasks on the Windows 10 machine
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	After we gained access to the Win10 machine, we were able to view all scheduled tasks. Scheduled tasks on a Windows system can therefore be leveraged for persistence and maintaining access even after losing initial access.
Images	
Affected Hosts	172.22.117.20
Remediation	• Regularly evaluate and clean up scheduled tasks. • Review and Disable Suspicious Tasks • By reviewing and identifying unnecessary or suspicious scheduled tasks, participants can mitigate the risk of losing access to the machine if an unauthorized task interferes with their control. • Review and adjust permissions for creating and modifying scheduled tasks. Ideally, limit this privilege to authorized users (e.g., administrators).

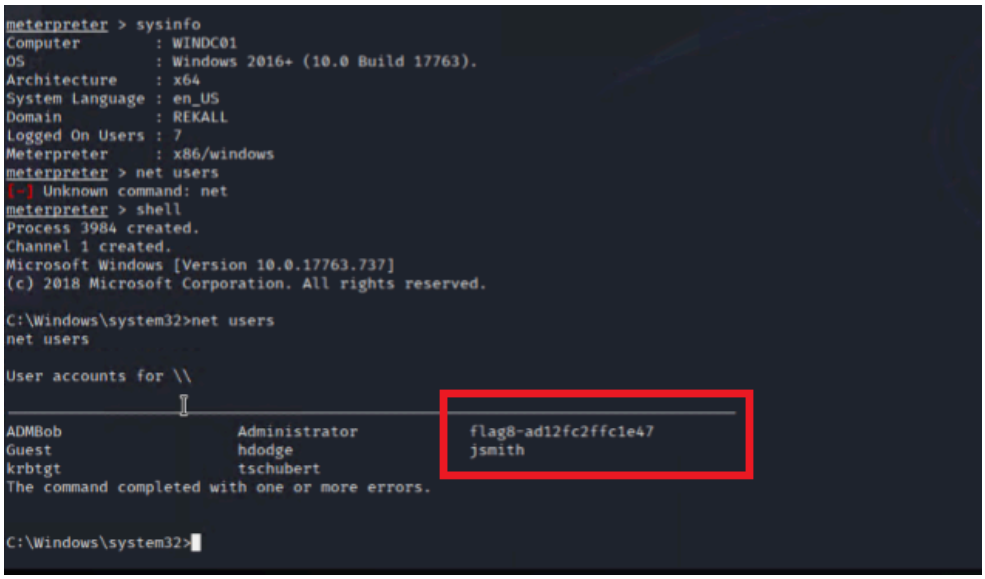
Vulnerability 6	Findings
Title	Windows Local Account Credential Dumping Vulnerability (Kiwi Attack)
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**High**
Description	Following a successful system compromise, we used a credential dumping tool (kiwi) to extract user credentials. This access allowed us to crack another user's password hash and escalate privileges to a different account.
Images	 

	
Affected Hosts	172.22.117.20
Remediation	• Implement the principle of least privilege for user accounts. • Consider implementing application control solutions that can restrict unauthorized processes from accessing or interacting with LSASS. • Enforce strong password policies for all user accounts. • Implement MFA for all user accounts, especially those with administrative privileges. • For Windows 10 Enterprise editions and later, enable Windows Defender Credential Guard. This feature isolates credential hashes in a virtualized environment, making them less accessible to attackers who might dump LSASS memory.

Vulnerability 7	Findings
Title	Sensitive Data in 'C:\Users\Public\Documents' directory
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Medium**
Description	During a recent security assessment of a Windows 10 machine, our investigation identified sensitive information stored within the user 'Public' folder. This folder, by default, grants read access to any user on the system.
Images	
Affected Hosts	172.22.117.20
Remediation	• Restrict folders and files permissions, this will prevent anyone other than authorized users from viewing the contents of the Public folder. • Move those files and folders to a more secure location on the system • Disable unnecessary accounts or guest accounts that might have access to the Public folder. • Educate employees about the importance of proper file storage practices and discourage them from storing sensitive data in the Public folder.

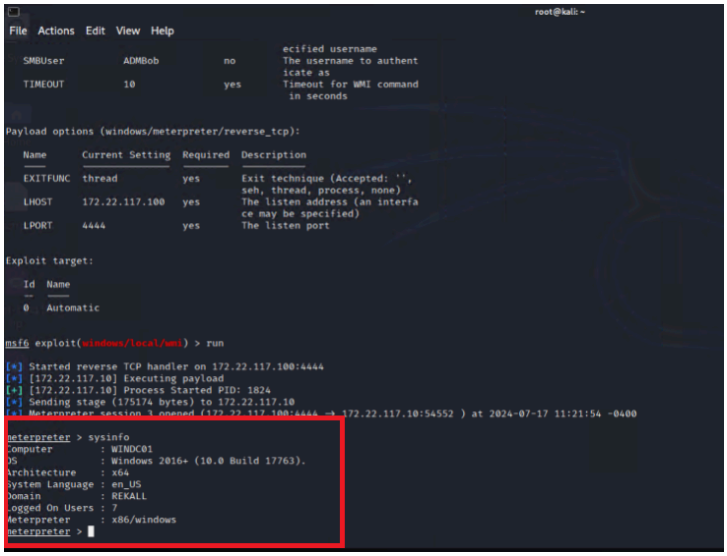
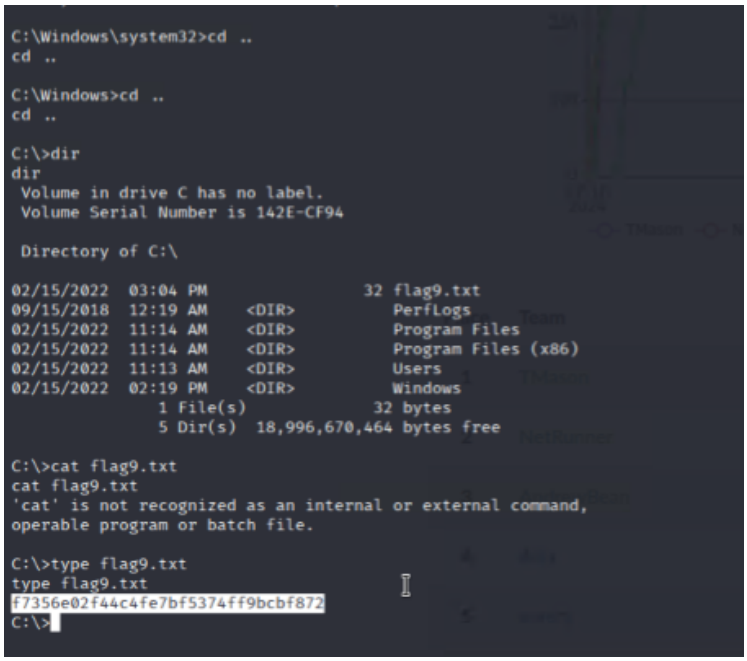
Vulnerability 8	Findings
Title	Credential Dumping on Windows 10 machine
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Critical**
Description	GoodCorp successfully performed credential dumping on a Windows 10 machine within the Rekall CTF environment. This allowed our team to extract credentials for a new user, crack the password, and leverage Windows/local/wmi to establish a connection to Domain Controller WINDC01.
Images	 



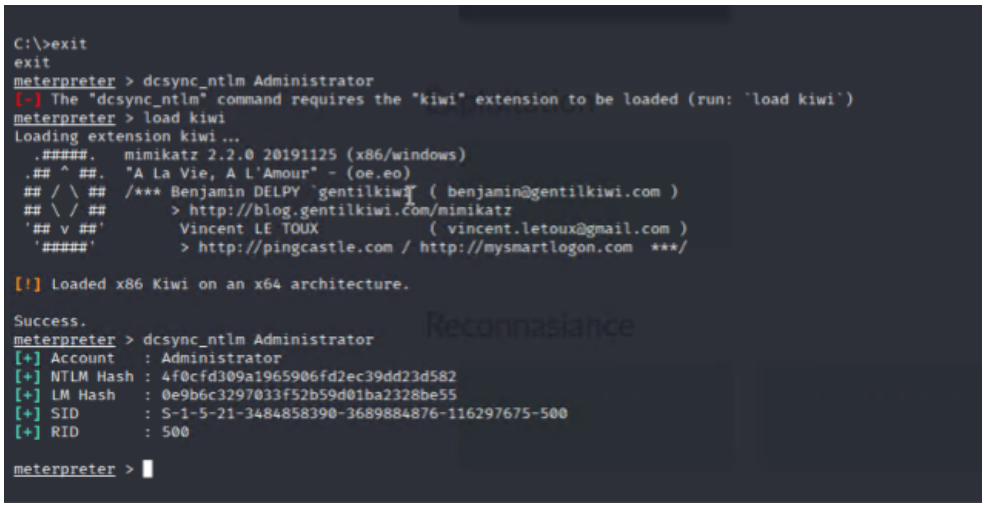


Affected Hosts	172.22.117.20 172.22.117.10
Remediation	• Enable Credential Guard in Windows 10, this feature isolates the Local Security Authority Subsystem Service (LSASS) in a virtualized environment, making it harder for attackers to dump credentials. • Enable Protected Process Light (PPL): PPL adds another layer of protection to LSASS, making it more difficult to access and dump credentials. • Implement the principle of least privilege, granting users only the necessary permissions. • Keep operating systems, applications, and security software up-to-date with the latest patches. • Require complex passwords with a mix of characters, including uppercase and lowercase letters, numbers, and special symbols. • Implement Multi-Factor Authentication (MFA) to add an extra layer of

	security beyond passwords. ● Educate employees about the risks of phishing, social engineering, and other cyber threats. ● Conduct regular security assessments to identify vulnerabilities and weaknesses.

Vulnerability 9	Findings
Title	Lateral Movement from Windows10 to WINDC01
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Critical**
Description	We successfully performed lateral movement to WINDC01 by using WMI via Metasploit
Images	 
Affected Hosts	172.22.117.10

Remediation	● Isolate critical systems: Place Domain Controllers in a highly isolated network segment. ● Restrict inbound/outbound traffic: Implement strict firewall rules to limit communication. ● Ensure operating systems, applications, and antivirus software are up-to-date.

Vulnerability 10	Findings
Title	Credential Access on the domain controller WINDC01
Type (Web app / Linux OS / WIndows OS)	WIndows OS
Risk Rating	**Critical**
Description	We used DCSync, a commonly used legitimate administrative tool for domain management to extract password hashes. We successfully retrieved the Administrator password hash.
Images	
Affected Hosts	172.22.117.10
Remediation	• Enforce Multi-Factor Authentication (MFA) for the Administrator account and other privileged accounts. • Require frequent password changes for the Administrator account. • Implement a strict account lockout policy for the Administrator account. • Configure strict firewall rules to limit inbound and outbound traffic to the DC. • Implement an Intrusion Detection System (IDS) to monitor network traffic for suspicious activity.