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	Hyucking Hackers, LLC
Contact Name	Saum Sepehr, Andrew Yeh, Rupinderjit (Harry) Chauhan, Chris West, Trevor Arashiro
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	10/25/2023	Saum Sepehr	
001	10/25/2023	Rupinderjit (Harry) Chauhan	
001	10/25/2023	Andrew Yeh	
001	10/25/2023	Chris West	
001	10/25/2023	Trevor Arashiro	

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.

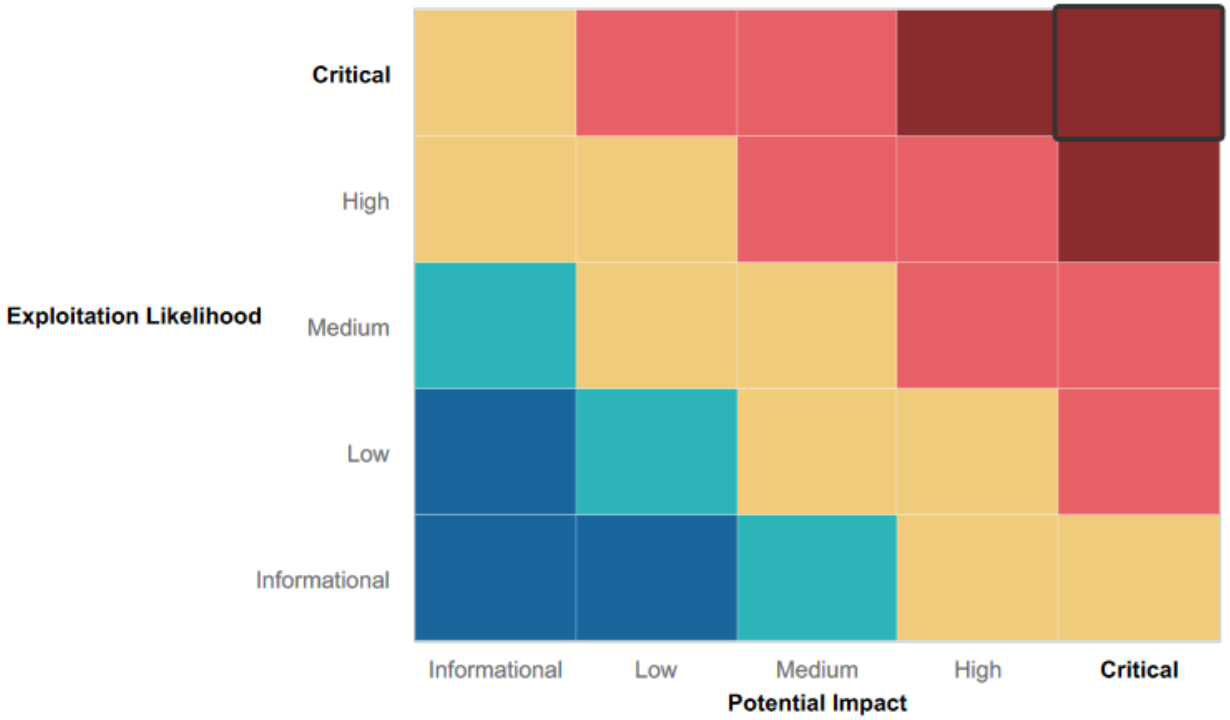
Executive Summary of Findings

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.

- Use of a Domain Controller with Admin control.
- Input Validation on the Web App

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 susceptible to SQL injections and XSS
- Publicly exposed sensitive information. Eg. HTML data of the web page as well as on a Github repository.
- Inefficient password policies
- Outdated Servers and Services. Eg. Apache, SLmail.
- Various open and insecure ports
- Insecure sensitive data in public directories

Executive Summary

We began the test by exploring the company's website and any publicly accessible data. We were able to obtain information during this reconnaissance phase, including open ports, admin credentials on the company github, and vulnerabilities found by Nessus scans. Additionally, we successfully collected information from the website through SQL injection and cross-site scripting which we later used to exploit the company’s systems.

Next, we found open ports that we used to exploit the company's systems using the data we collected from the Zenmap scan on the host. We discovered ports like FTP, SSH, Apache Tomcat, and Apache Struts in our scan. We used a tool called Metasploit to successfully find exploits for these particular ports and services. Following the systems' execution of these attacks, we obtained sensitive data, including user credentials and system information. We were able to escalate our privileges and permissions in order to collect more sensitive data after discovering these extra user credentials.

Summary Vulnerability Overview

Vulnerability	Severity
Web Flag 1: XSS found on website welcome page	Critical
Web Flag 2: XSS found on memory planner page bypassing input sanitation	Critical
Web Flag 3: Persistent XSS found on comments page	Critical
Web Flag 4: X-Powered-By header found in HTTP response	Medium
Web Flag 5: PHP script injected through image upload	Critical
Web Flag 6: PHP script injected through image upload bypassing sanitation	Critical
Web Flag 7: SQL injection on user login	Critical
Web Flag 8: Admin credentials found through page source	Critical
Web Flag 9: Sensitive data found on robots.txt	High
Web Flag 10: Linux command injection found on DNS check input	Critical
Web Flag 11: Linux command injection found on MX Record Checker input	Critical
Web Flag 12: Linux traversal through URL on disclaimer.php page	Critical
Web Flag 13: PHP code injection on souvenirs.php page	Critical
Web Flag 14: Restricted page found using admin id in URL	Critical
Web Flag 15: Old disclaimer page found through navigating directory	High
Linux Flag 1: Admin address found through domain lookup	High
Linux Flag 2: Certificate fingerprint of website found	Informational
Linux Flag 3: Open Source Exposed Data	Informational
Linux Flag 4: Nmap Scan of the network	Low
Linux Flag 5: Intense network scan	Low
Linux Flag 6: Nessus Scan	Low
Linux Flag 7: Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617)	Critical
Linux Flag 8: Shellshock	Critical
Linux Flag 9: Access Control	Critical
Linux Flag 10: Struts - CVE-2017-5638	Critical
Linux Flag 11: Drupal - CVE-2019-6340	Critical
Linux Flag 12: SSH password guessing CVE-2019-14287	Critical
Windows Flag 1: Open source exposed data	Critical
Windows Flag 2: Insecure password	Critical
Windows Flag 3: FTP port Open and vulnerable	Critical
Windows Flag 4: SLmail Remote buffer overflow CVE-2003-0264	Critical
Windows Flag 5: Access control for Scheduled tasks	Critical
Windows Flag 6: Insecure password	Critical
Windows Flag 7: Insecure file security measures	Critical

Windows Flag 8: Microsoft Windows Authenticated User Code Execution, PsExec	**Critical**
Windows Flag 9: Remote code execution	**Critical**
Windows Flag 10: Remote code execution	**Critical**

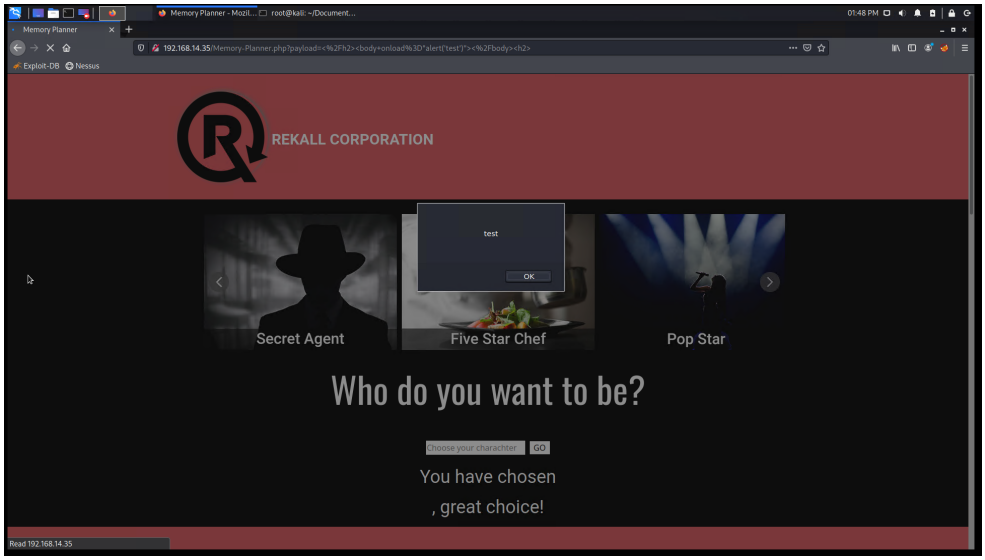
The following summary tables represent an overview of the assessment findings for this penetration test:

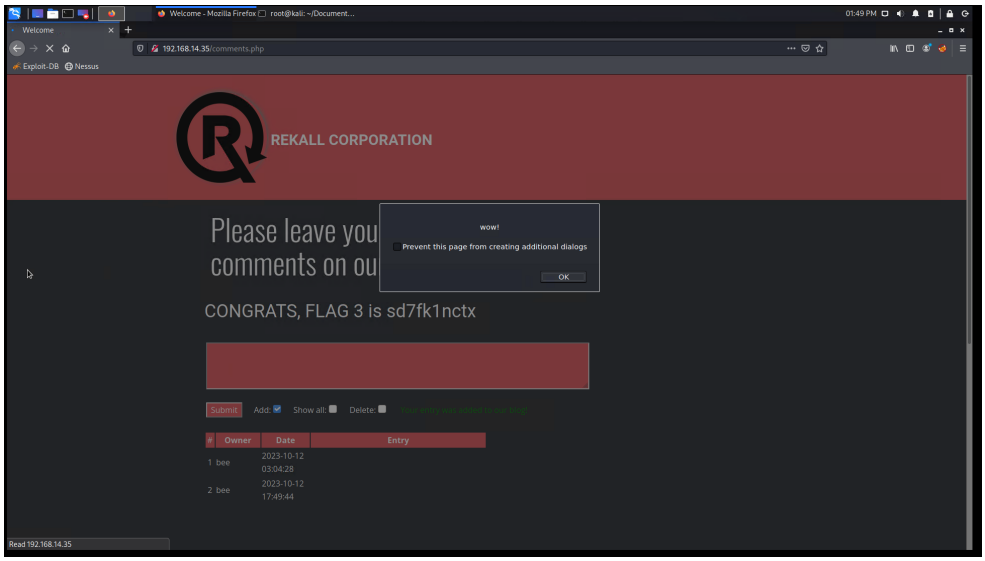
Scan Type	Total
Hosts	192.168.14.35 34.102.136.180 192.168.13.10 192.168.13.11 192.168.13.12 192.168.13.13 192.168.13.14 172.22.117.10 172.22.117.20
Ports	80,8080,21,22,110,445

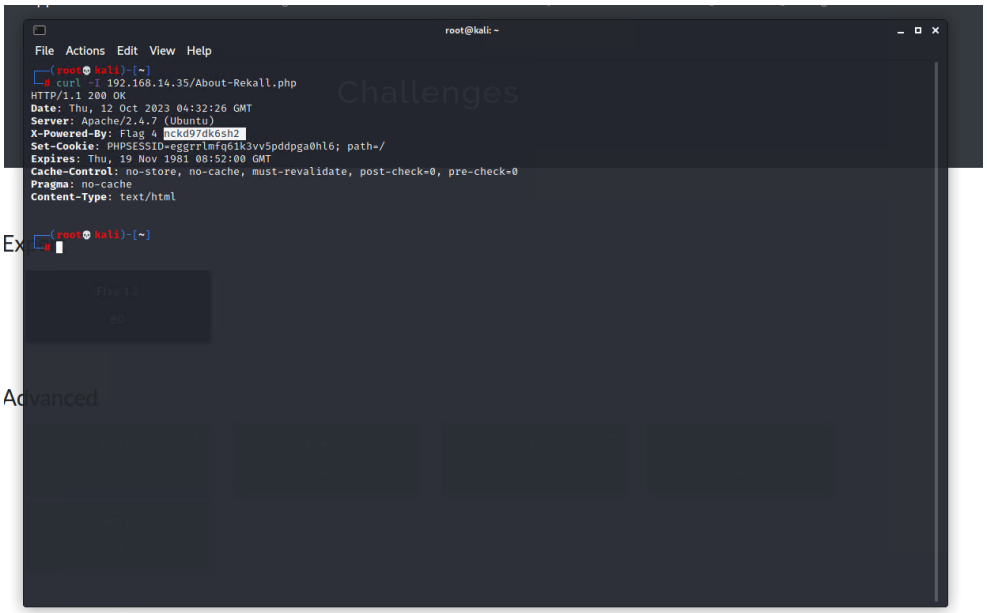
Exploitation Risk	Total
Critical	26
High	5
Medium	1
Low	3
Informational	2

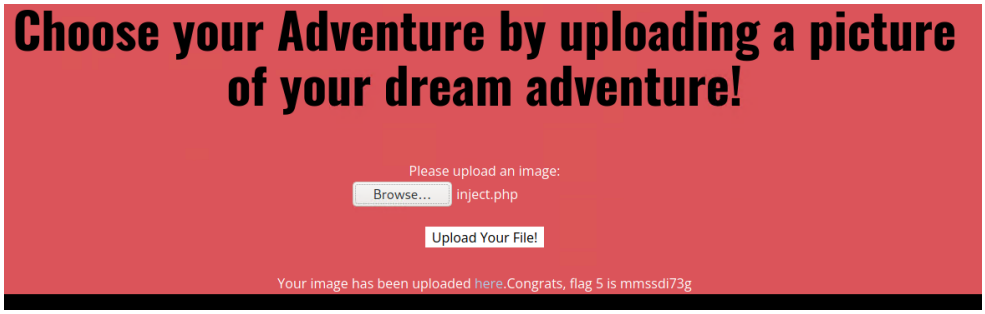
Vulnerability Findings

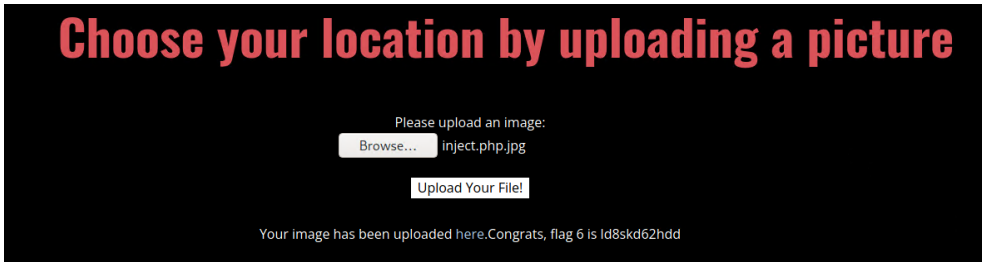
Vulnerability 1	Findings
Title	Web Flag 1: XSS found on website welcome page
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	On the welcome page, on the field to enter your name we were able to insert a script on the webpage.
Images	
Affected Hosts	192.168.13.35
Remediation	Input validation and sanitization, along with encoding user generated input and possibly implement security headers.

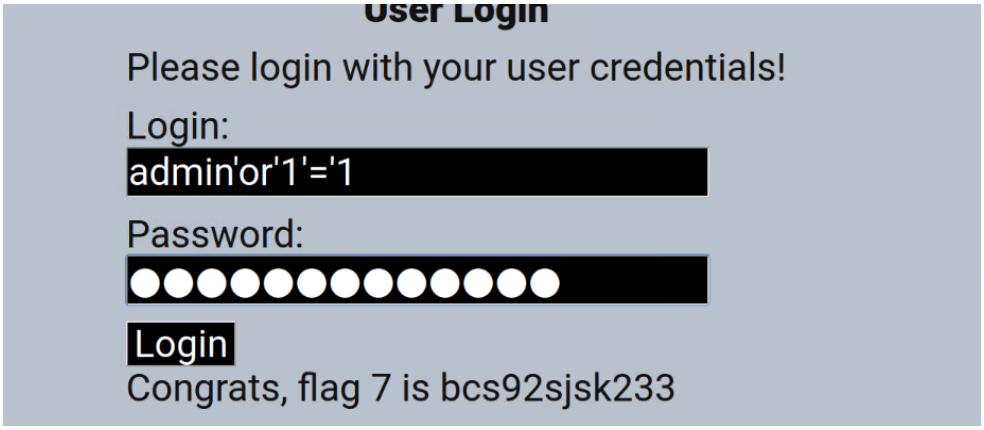
Vulnerability 2	Findings
Title	Web Flag 2: XSS found on memory planner page bypassing input sanitation
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	When attempting to insert a script we found that there was input sanitation which did prevent certain methods of inserting a script, but we were able to insert a script through other cases that the validator did not account for such as using other html tags and inserting a script through an onload attribute.
Images	 
Affected Hosts	192.168.13.35
Remediation	More input validation and sanitization, along with encoding user generated input and possibly implementing security headers.

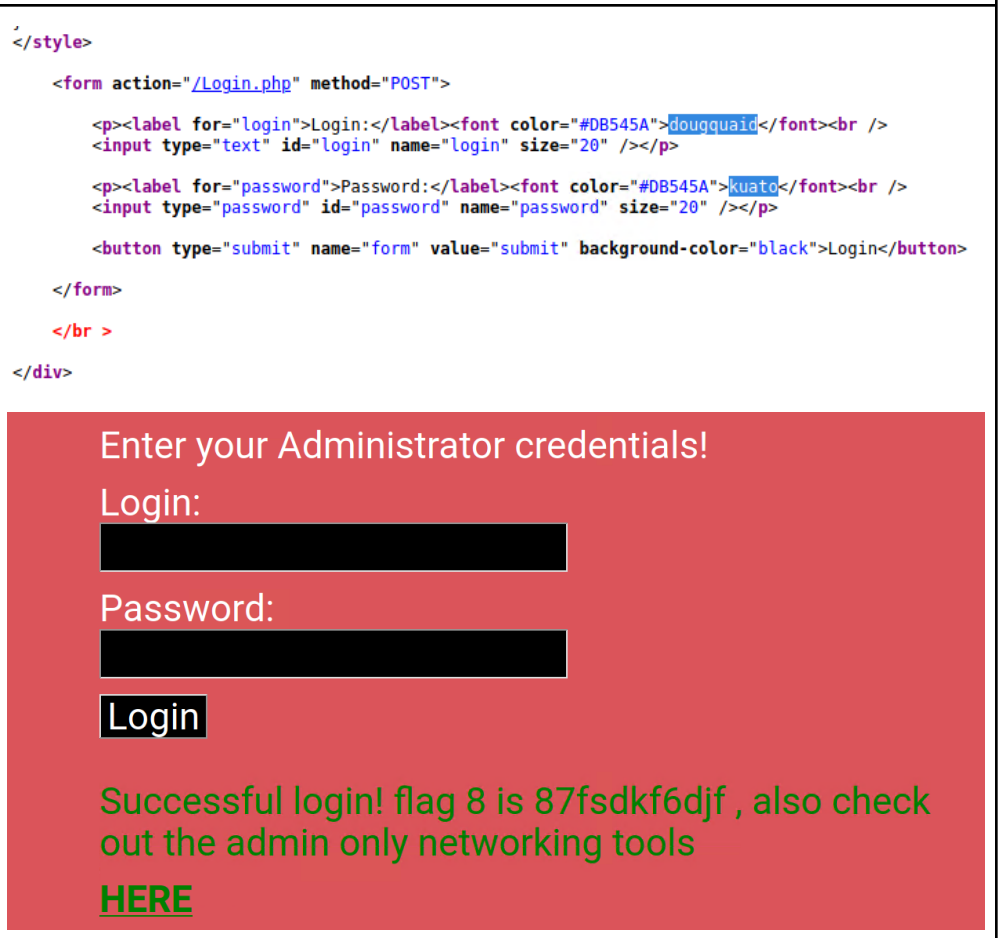
Vulnerability 3	Findings
Title	Web Flag 3: Stored XSS exploit found on comments page
Type (Web app / Linux OS / WIndows OS)	web app
Risk Rating	**Critical**
Description	We found another point for cross site scripting on the comments page, we even found it to be persistent since the injected script would be embedded in the comment element.
Images	
Affected Hosts	192.168.13.35
Remediation	Input validation and sanitization, along with encoding user generated input and possibly implement security headers.

Vulnerability 4	Findings
Title	Web Flag 4: X-Powered-By header found in HTTP response
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Medium**
Description	When using curl on the site, the X-Powered-By header was found, this header puts the site at risk as it reveals what the site is running with and can lead malicious actors to which exploits they can use.
Images	
Affected Hosts	192.168.13.35
Remediation	Avoid using this header in HTTP responses

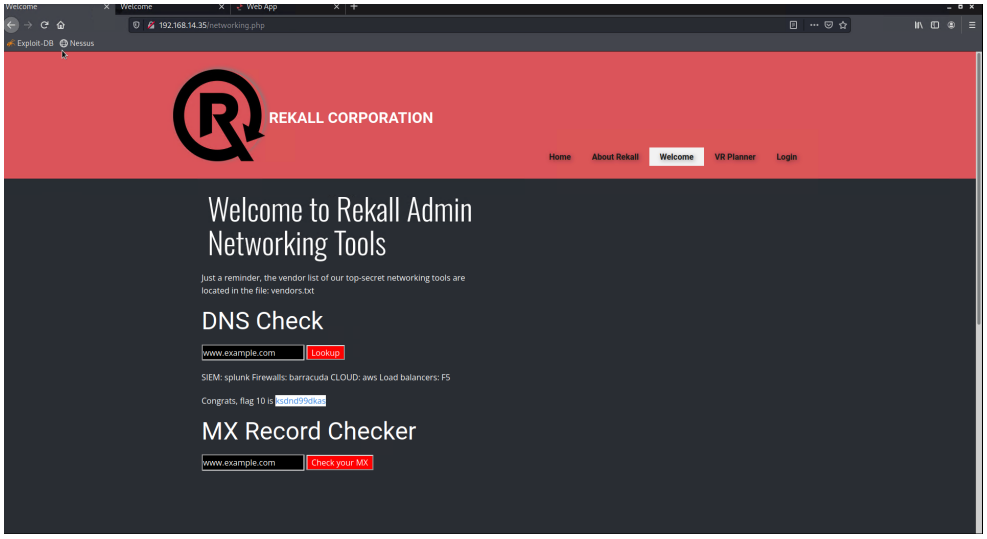
Vulnerability 5	Findings
Title	Web Flag 5: PHP script injected through image upload
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	We found that on the memory planner page, one image input field was able to upload a php script.
Images	
Affected Hosts	192.168.13.35
Remediation	Input validation and sanitation, file type validation and size limitation, but also scanning the files and comparing them to libraries of malware and malicious scripts.

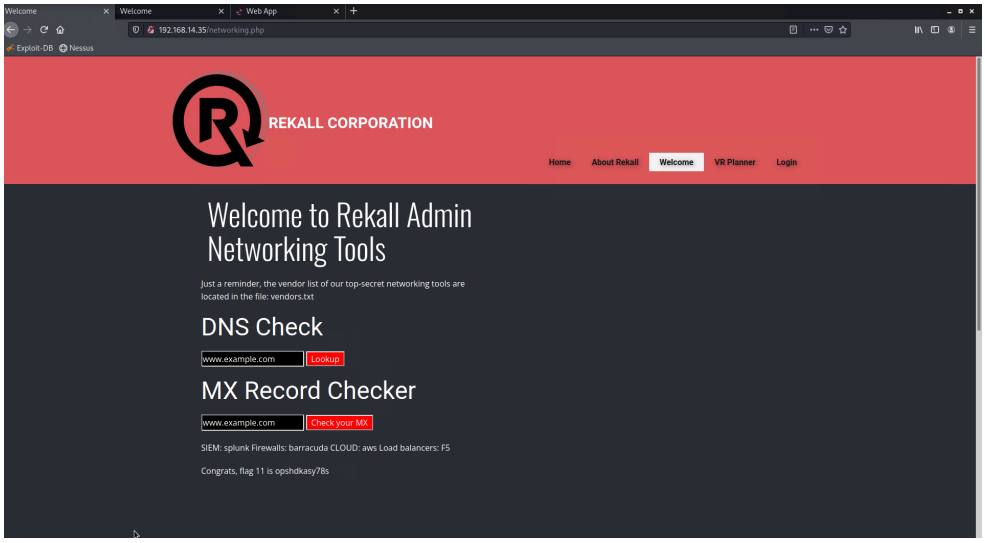
Vulnerability 6	Findings
Title	Web Flag 6: PHP script injected through image upload bypassing sanitation
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	On the memory planner page for the second image upload handler, it is found that a php script can be uploaded using a .jpg ending bypassing the
Images	
Affected Hosts	192.168.13.35
Remediation	Input validation and sanitation, and especially scanning files for malware.

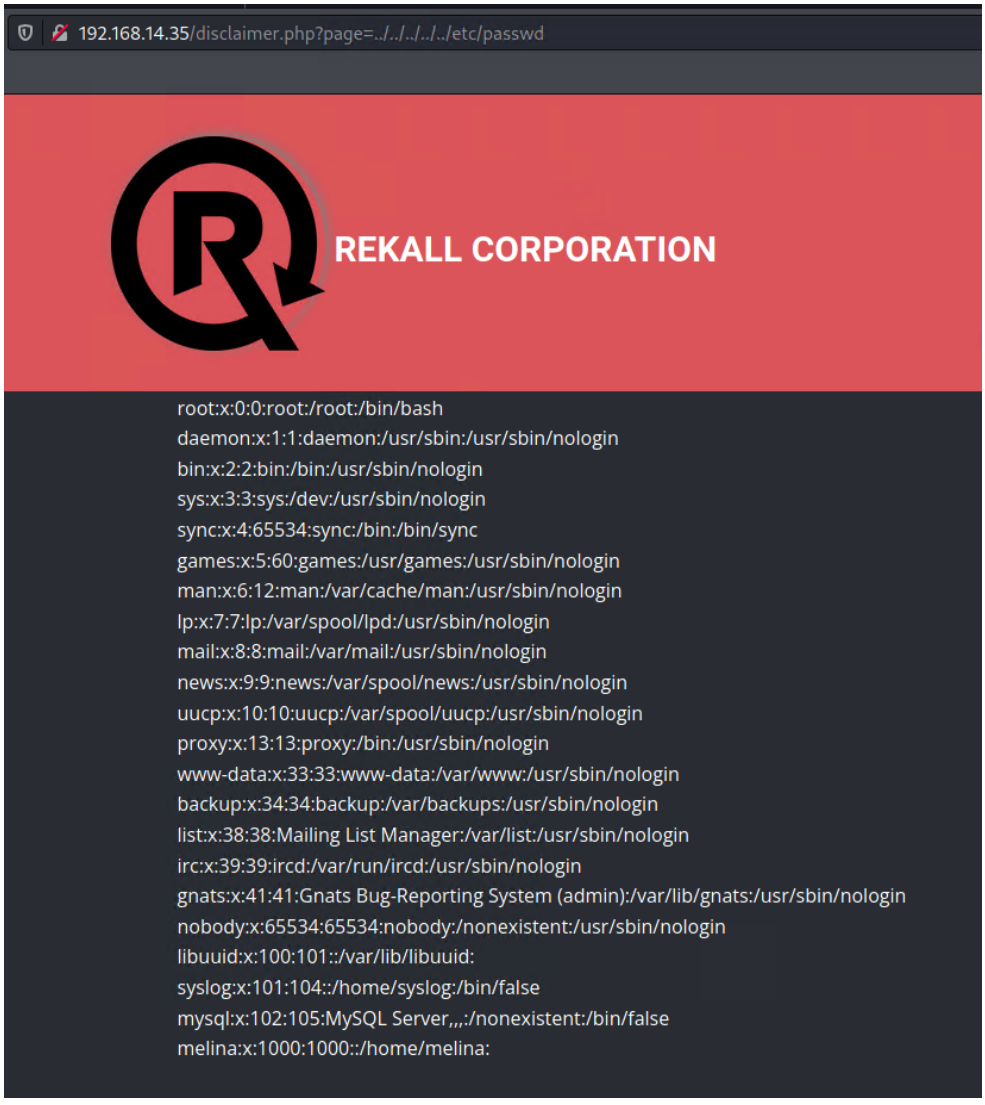
Vulnerability 7	Findings
Title	Web Flag 7: SQL injection on user login
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	On the user login, we were able to use SQL injection to gain access
Images	
Affected Hosts	192.168.13.35
Remediation	Input validation and sanitation, using prepared statements in SQL, but also hashing passwords before comparing and validating.

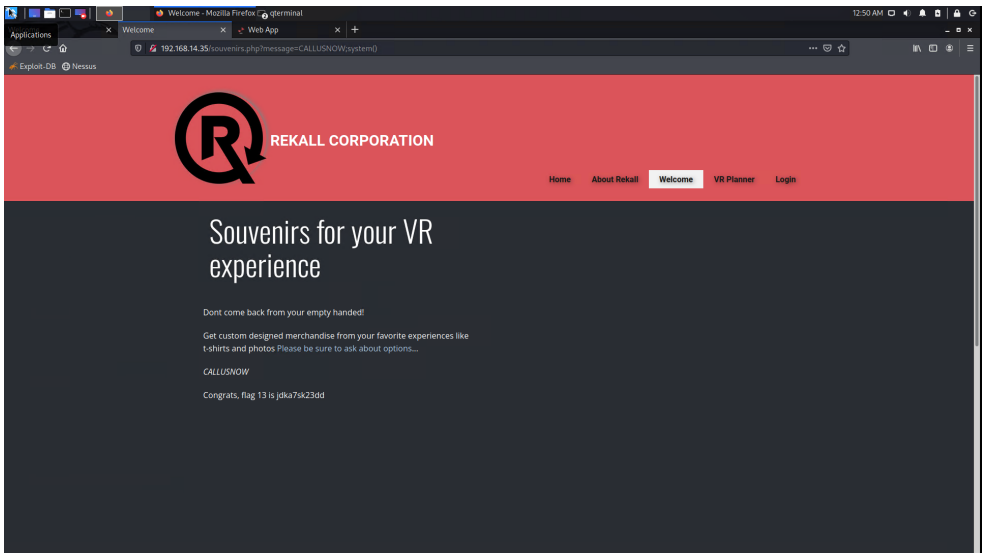
Vulnerability 8	Findings
Title	Web Flag 8: Admin credentials found through page source
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	Looking at the login page's page source, we found credentials embedded in the HTML. Using these credentials we were able to log in with administrator privileges.
Images	
Affected Hosts	192.168.13.35
Remediation	Do not store credentials in plain text anywhere, especially embedded in HTML.

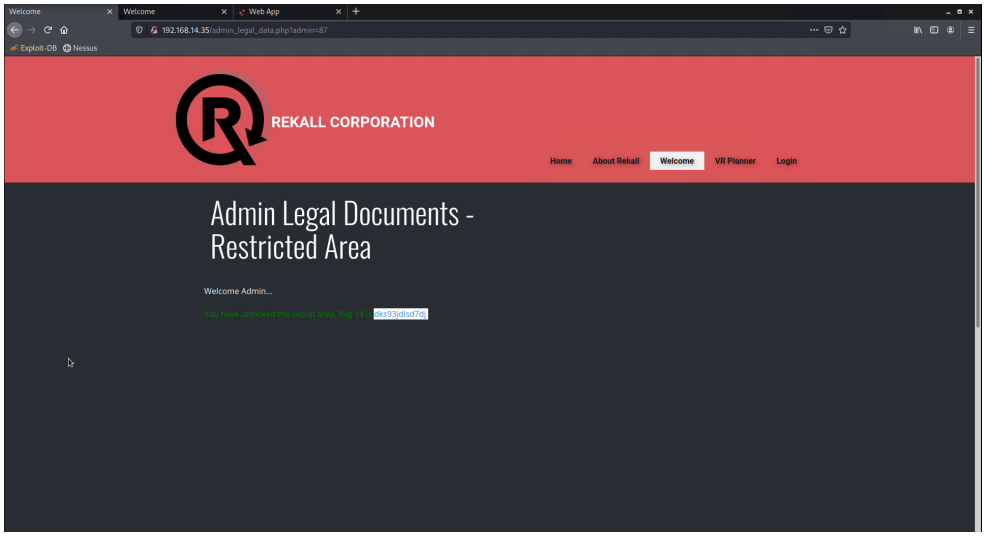
Vulnerability 9	Findings
Title	Web Flag 9: Sensitive data found on robots.txt
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	Looking at robots.txt we were able to find sensitive data in the text file.
Images	
Affected Hosts	192.168.13.35
Remediation	Edit the file to not disclose sensitive data.

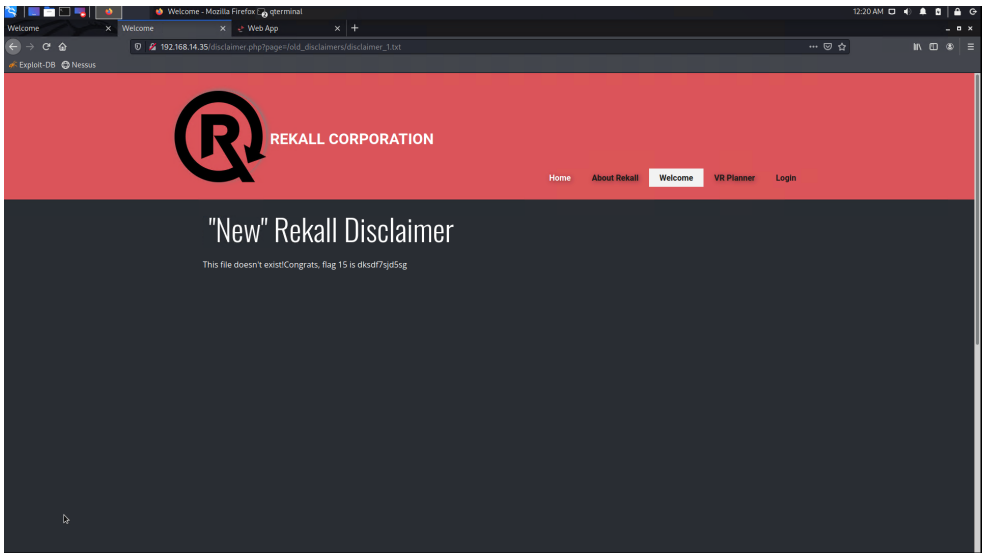
Vulnerability 10	Findings
Title	Web Flag 10: Linux command injection found on DNS check input
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	On the admin tools page, we found that the DNS check field can be used to inject other linux commands onto the web server.
Images	
Affected Hosts	192.168.13.35
Remediation	Input sanitation and verification, since the input is a URL, possibly implement a regex expression to validate for just URL inputs.

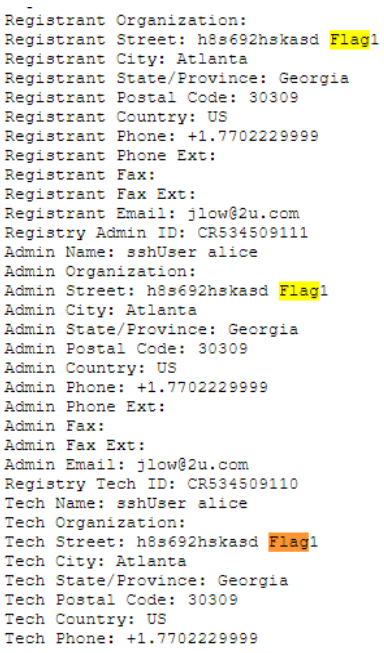
Vulnerability 11	Findings
Title	Web Flag 11: Linux command injection found on MX Record Checker input
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	On the networking tools page we found that the MX Record Checker was also vulnerable to command injection of linux commands.
Images	
Affected Hosts	192.168.13.35
Remediation	Input sanitation and verification, possibly implement a regex expression to only accept URL formatted inputs

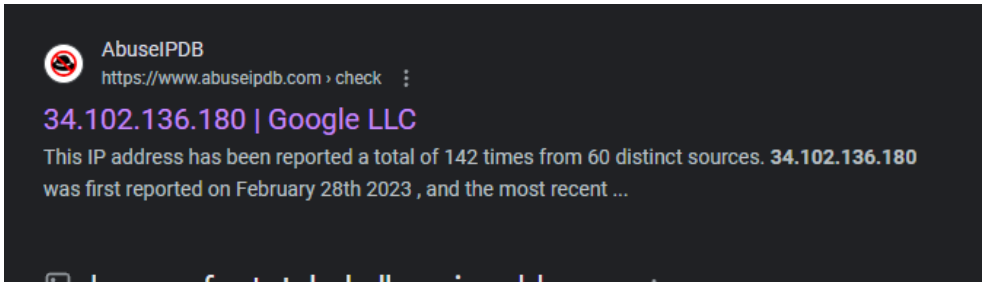
Vulnerability 12	Findings
Title	Web Flag 12: Linux traversal through URL on disclaimer.php page
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	Through using linux commands in the URL, we found that we can traverse the web server system and reveal other files such as etc/passwd
Images	
Affected Hosts	192.168.13.35
Remediation	Input validation to restrict and allow certain characters, implementing chroot jail to restrict the application.

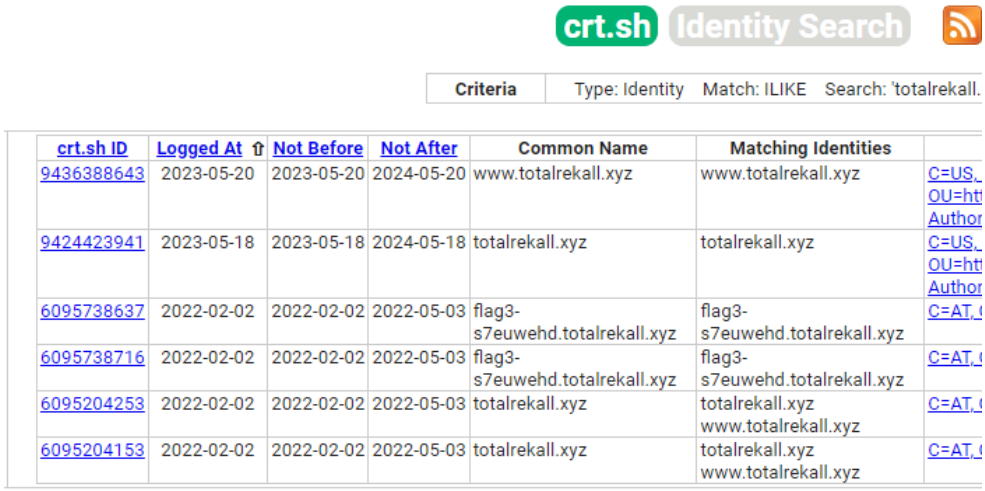
Vulnerability 13	Findings
Title	Web Flag 13: PHP code injection on souvenirs.php page
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	On the souvenirs page, we were able to insert a php command in the URL and execute that command, this can lead to malicious php script injection with just the URL.
Images	
Affected Hosts	192.168.13.35
Remediation	Input sanitation, avoid using ‘include’ and ‘require’ dynamically, using absolute paths, and disable URL includes.

Vulnerability 14	Findings
Title	Web Flag 14: Restricted page found using admin id in URL
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**Critical**
Description	Through brute force, we were able to find an admin session ID which allowed us to access a restricted page.
Images	
Affected Hosts	192.168.13.35
Remediation	Implement session cookies to restrict pages to users that have that cookie in their browser and not have the sessions be accessible through URL.

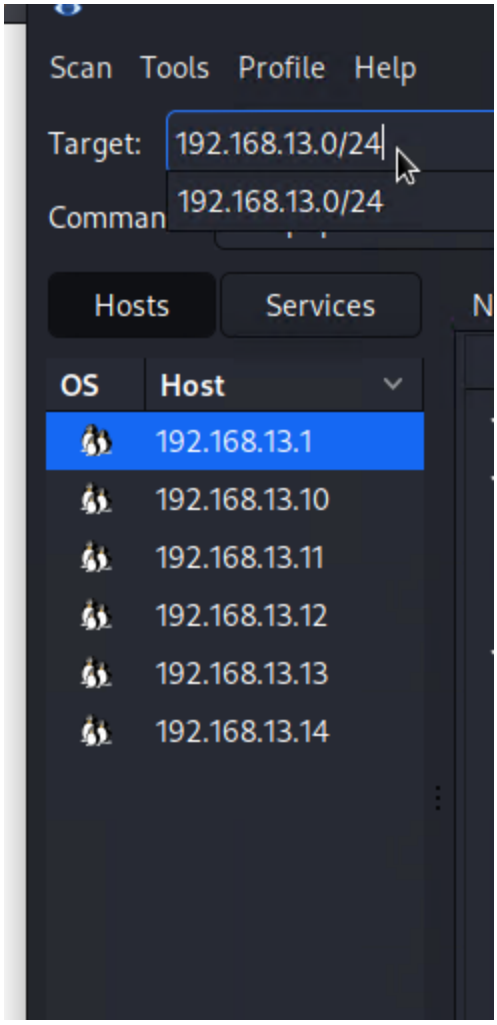
Vulnerability 15	Findings
Title	Web Flag 15: Old disclaimer page found through navigating directory
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	**High**
Description	We were able to find an old disclaimer page through linux traversal from vulnerability 10 and 11 using the ls command. We then found a directory called old_disclaimers and a file with an old disclaimers page that is accessable.
Images	
Affected Hosts	192.168.13.35
Remediation	Remove old pages that are not to be used from the web server

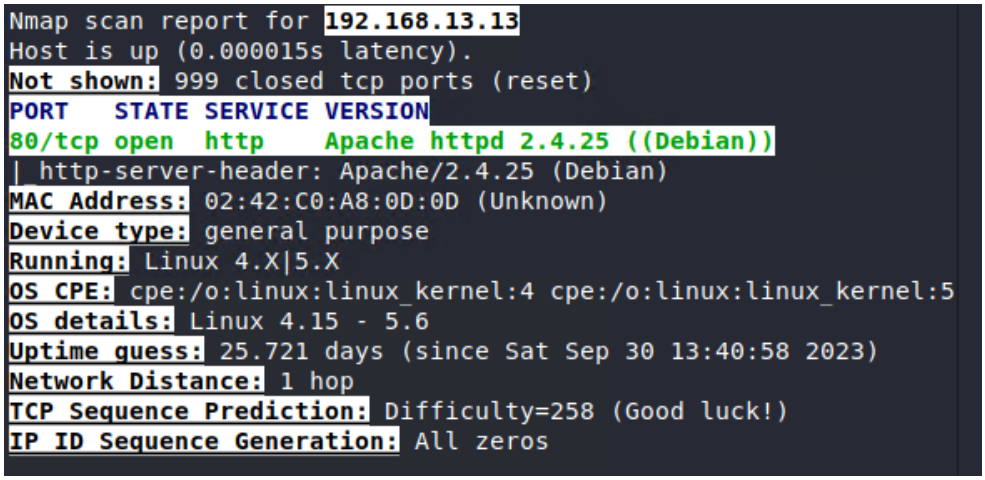
Vulnerability 16	Findings
Title	Flag 1: Admin address found through domain lookup
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**High**
Description	Following the OSI framework we found the domain dossier lookup and used it to search the domain, Resulting in the exposure of personal information belonging to the administrator.
Images	
Affected Hosts	34.102.136.180
Remediation	Remove user information from the WHOIS database and use a WHOIS privacy service.

Vulnerability 17	Findings
Title	Flag 2: Certificate fingerprint of website found
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Informational
Description	Google search for “totalrekall.xyz” we kept searching down the results and entered every single result until one worked.
Images	
Affected Hosts	34.102.136.180
Remediation	No remediation required

Vulnerability 18	Findings
Title	Flag 3: Open Source Exposed Data
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Informational
Description	A crt.sh search of totalrekall.xyz
Images	
Affected Hosts	34.102.136.180
Remediation	No remediation required

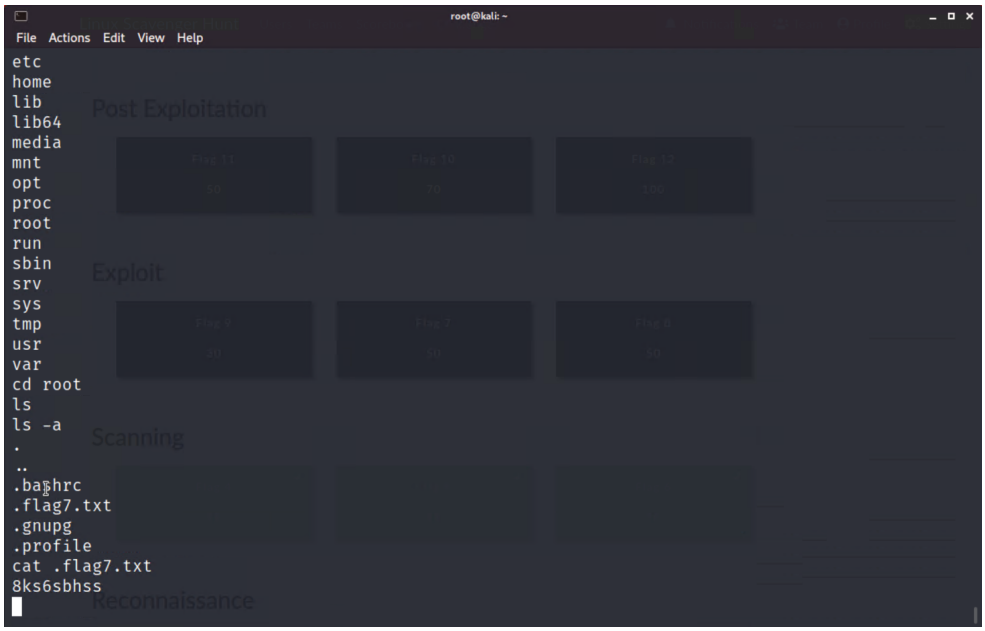
crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	
9436388643	2023-05-20	2023-05-20	2024-05-20	www.totalrekall.xyz	www.totalrekall.xyz	C=US, OU=htt Author
9424423941	2023-05-18	2023-05-18	2024-05-18	totalrekall.xyz	totalrekall.xyz	C=US, OU=htt Author
6095738637	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	C=AT,
6095738716	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	C=AT,
6095204253	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=AT,
6095204153	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=AT,

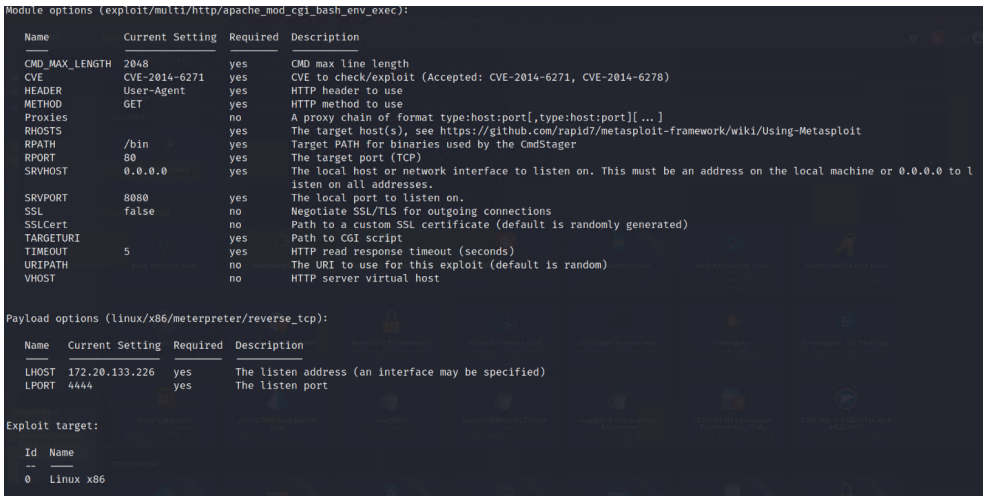
Vulnerability 19	Findings
Title	Flag 4: Nmap Scan of the network
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Low**
Description	We ran an Nmap scan for the network (nmap 192.168.13.0/24) to determine that there are 5 hosts excluding the host scanning from.
Images	
Affected Hosts	192.168.13.0/24
Remediation	Set firewall rules to prevent outside traffic from being able to read open ports

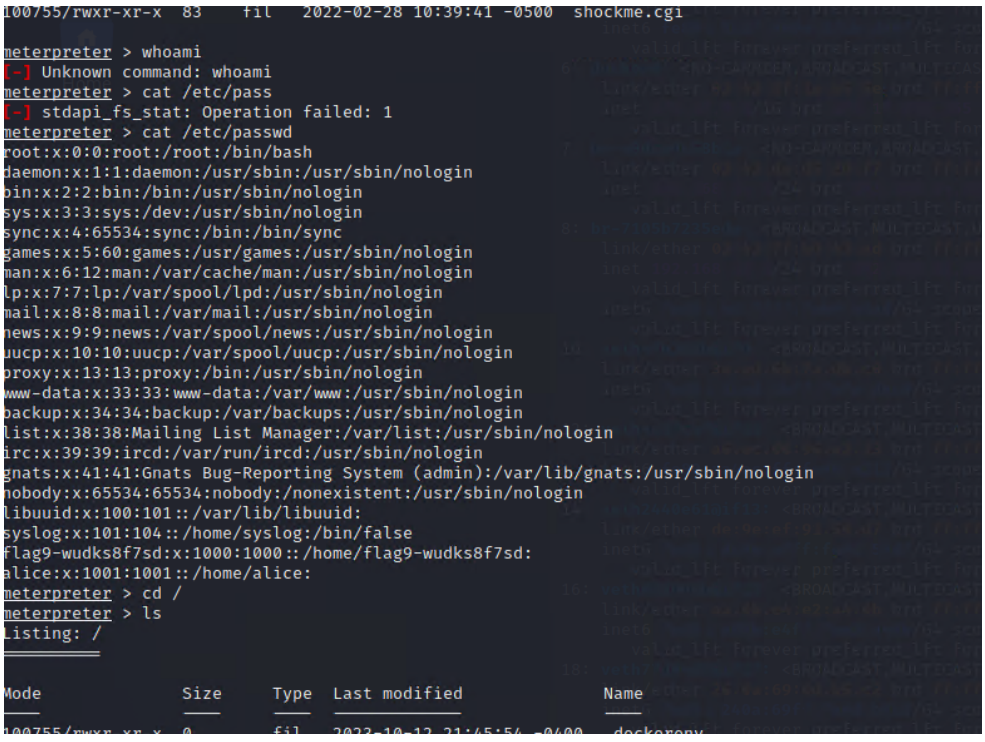
Vulnerability 20	Findings
Title	Flag 5: Intense network scan
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Low**
Description	We ran an intensive Nmap scan to discover the host runs Drupal.
Images	
Affected Hosts	192.168.13.13
Remediation	Set firewall rules to prevent outside traffic from being able to read open ports.

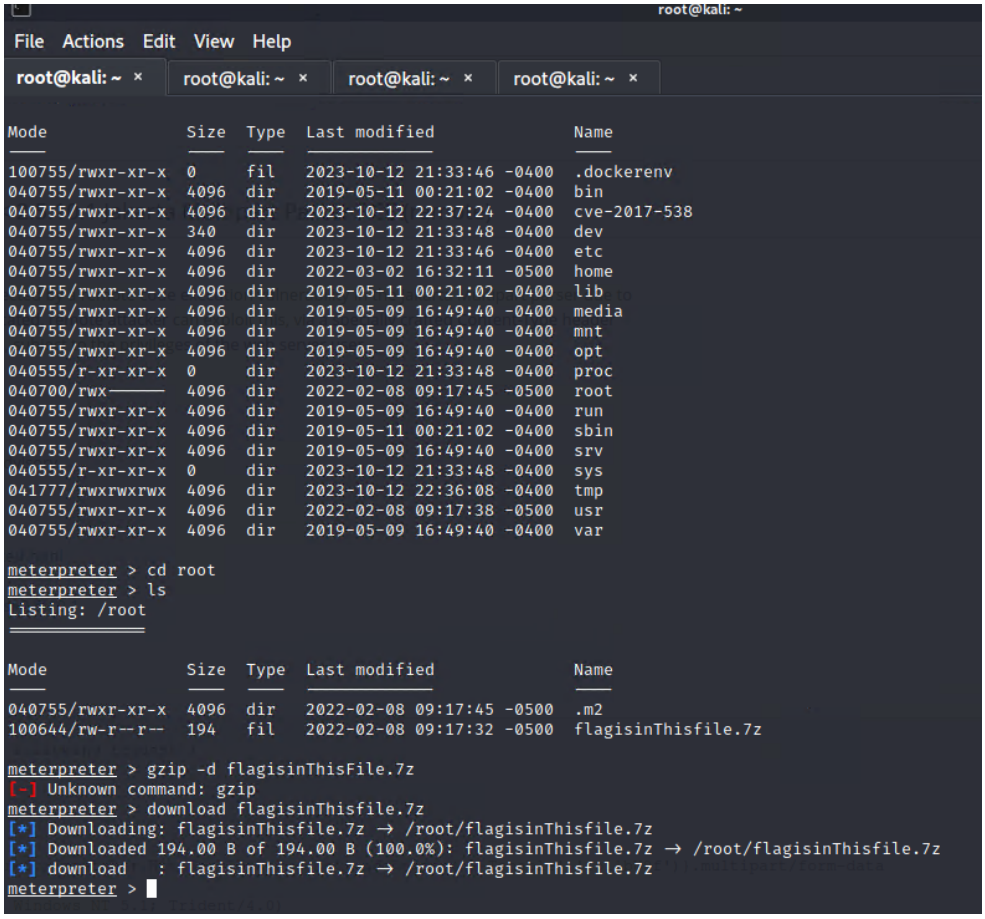
Vulnerability 21	Findings
Title	Flag 6: Nessus Scan
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Low**
Description	Running a Nessus scan revealed and Apache Struts vulnerability
Images	[bruh i dont have this]
Affected Hosts	192.168.13.12

Remediation	Update Apache regularly.

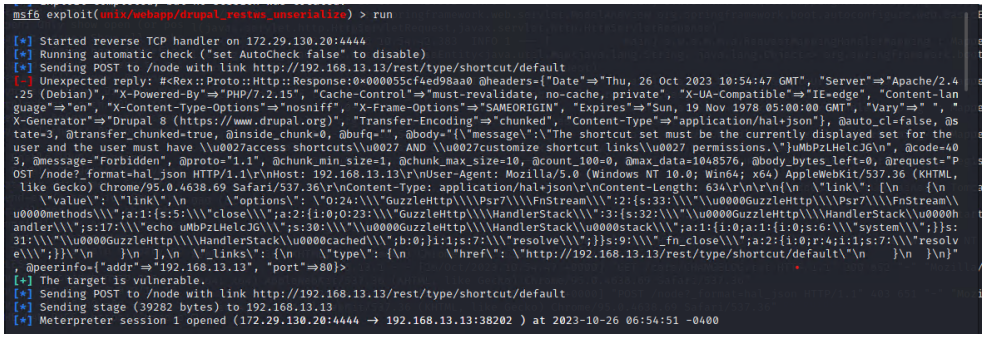
Vulnerability 22	Findings
Title	Flag 7: Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617)
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	We exploited and Apache remote code execution vulnerability
Images	
Affected Hosts	192.168.13.10
Remediation	Update Apache regularly. Closing port 8080 to unauthorized sources.

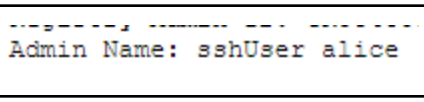
Vulnerability 23	Findings
Title	Flag 8: Shellshock
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	We ran an aggressive nmap scan on the target IP, then used msfconsole to find attacks using the keyword “shock”.
Images	
Affected Hosts	192.168.13.11
Remediation	Regularly update your Windows operating system with the latest security patches and updates.

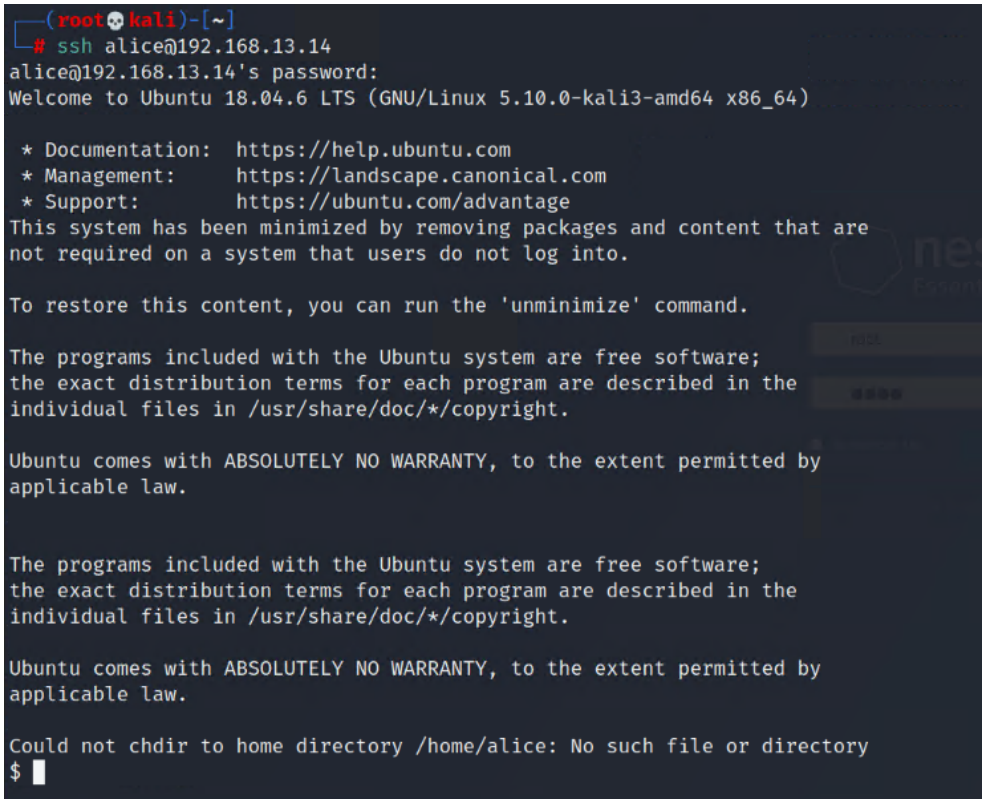
Vulnerability 24	Findings
Title	Flag 9: Access Control
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	After gaining a reverse shell, we were easily able to traverse the directory and find key information.
Images	
Affected Hosts	192.168.13.11
Remediation	Require sudo passwords for all users.

Vulnerability 25	Findings
Title	Flag 10: Apache Struts - CVE-2017-5638
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	Used exploit/multi/http/struts2_content_type_ognl to access 192.168.13.12, found flag archived in /root, downloaded the archive and unzipped it
Images	
Affected Hosts	192.168.13.12

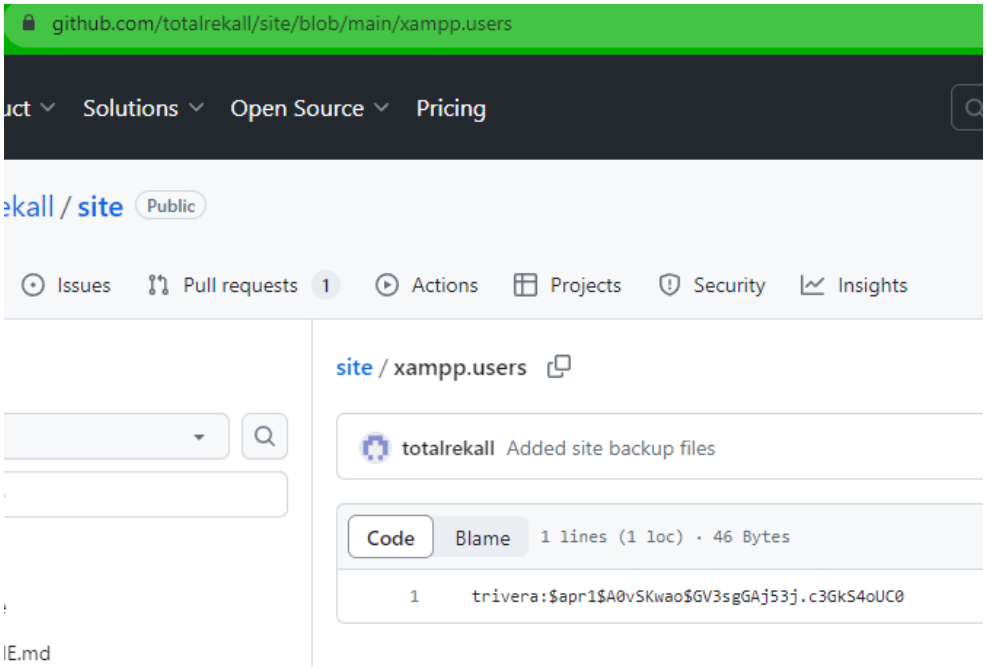
Remediation	Update Apache regularly.

Vulnerability 26	Findings
Title	Flag 11: Drupal - CVE-2019-6340
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	Searching for Drupal exploits.We used the following exploit to get a Meterpreter shell MSF: unix/webapp/drupal_restws_unserialize
Images	
Affected Hosts	192.168.13.13
Remediation	Update the Drupal system to the latest version. Additionally, implement security measure by closing Port 80 to external access

Vulnerability 27	Findings
Title	Flag 12: SSH password guessing CVE-2019-14287
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	Using the lookup I was able to find the user “alice”. Seeing she is an SSH user we tried to ssh using “alice” for both user and password.
Images	

	
Affected Hosts	192.168.13.14
Remediation	Stronger password policies and prevent outside sources from accessing port 22.

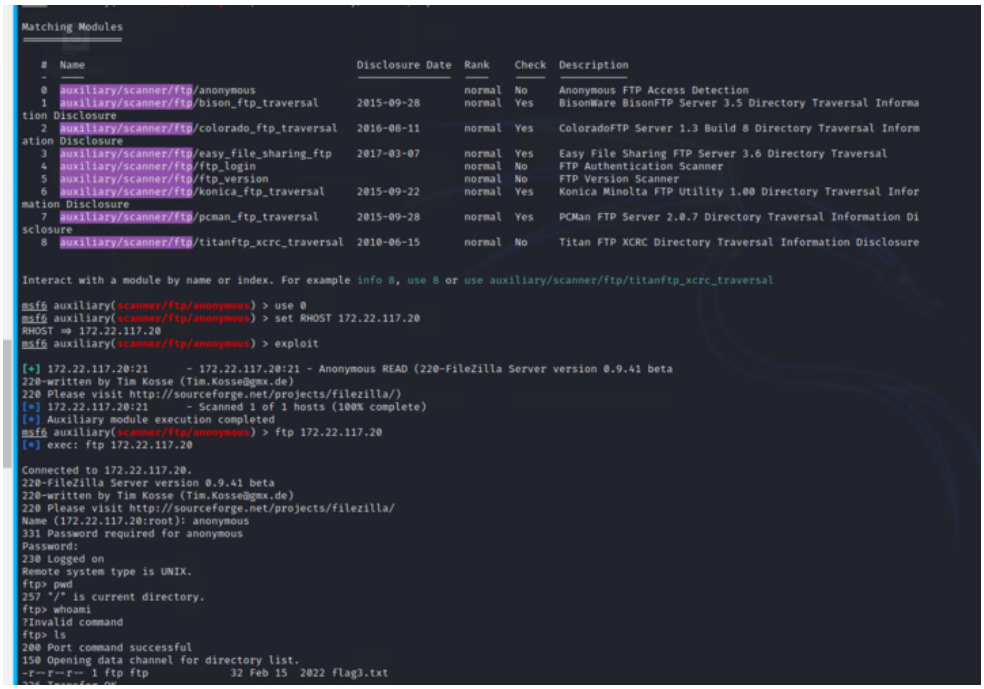
Vulnerability 28	Findings
Title	Flag 1: Open source exposed data.
Type (Web app / Linux OS / WIndows OS)	Windows OS

Risk Rating	**Critical**
Description	While searching through the website, we found a link that led the team to the company's github where we found user credentials for trivera.
Images	
Affected Hosts	https://github.com/totalrekall/site/blob/main/xampp.users
Remediation	Our team would recommend that the credentials either be removed from the github repository or the repository be made private with authorized access only.

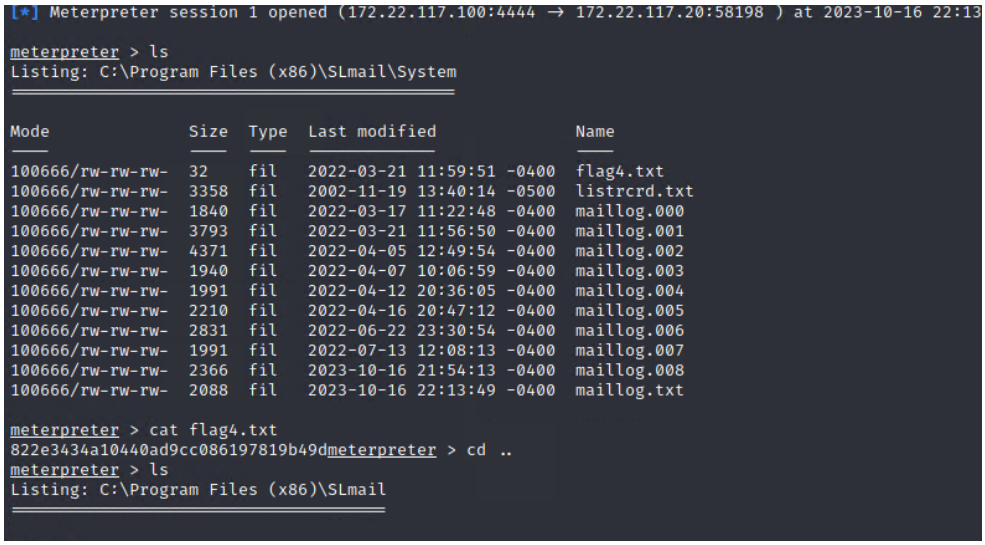
Vulnerability 29	**Findings**
Title	Flag 2: Apache HTTP - Web page Exploit
Type (Web app / Linux OS / WIndows OS)	Windows OS

Risk Rating	**Critical**
Description	After running a scan on the host 172.22.117.0/24 we discovered HTTP Apache server was active. We used the same host and conducted a web browser search. The web server requested a set of credentials. After using the credentials we discovered earlier for trivera we were able to access the necessary information.
Images	
Affected Hosts	172.22.117.20
Remediation	Our team recommends securing the Apache server access and preventing threat actors from using leaked credentials to gain information.

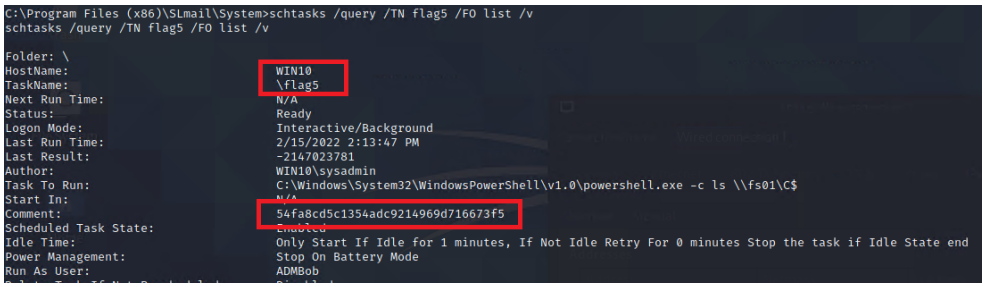
Vulnerability 30	Findings
Title	Flag 3: FTP port Open and vulnerable
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Critical**
Description	After conducting a scan on the host 172.22.117.0/20 our team discovered that port 21 (FTP) which also revealed that anonymous access to that port was possible. We were able to connect to that port and collect the necessary

	information.
Images	
Affected Hosts	172.22.117.20
Remediation	Our team would recommend that access through port 21 (FTP) should be restricted

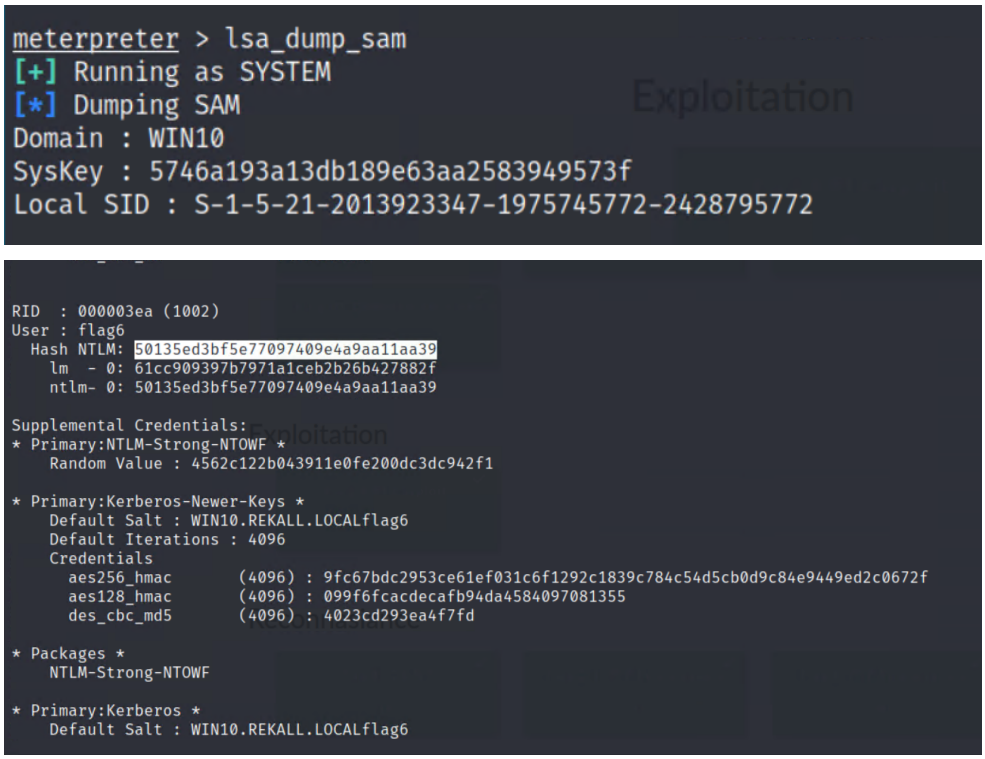
Vulnerability 31	Findings
Title	Flag 4: SLmail Remote buffer overflow CVE-2003-0264
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Critical**
Description	After running a scan on the host 172.22.117.20 we discovered the POP3 port was open which usually serves as a mail server for windows. Using a SLmail

	exploit, we were able to gain access to the machine and collect necessary information.
Images	
Affected Hosts	172.22.117.20
Remediation	Our team recommends that access to the mail servers should be restricted and secured.

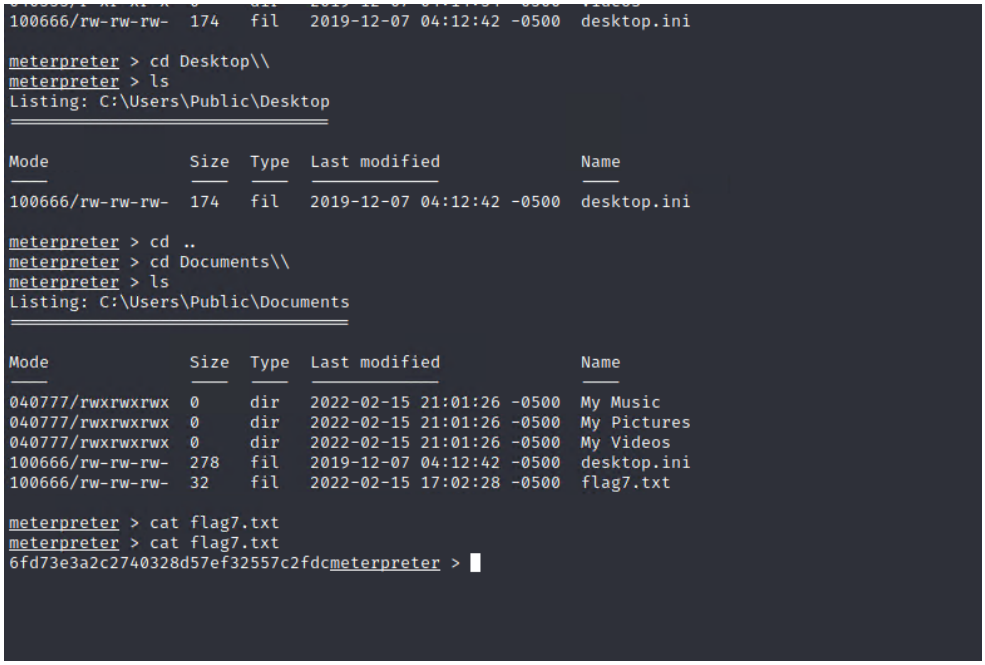
Vulnerability 32	Findings
Title	Flag 5: Access control for Scheduled tasks
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Critical**
Description	After gaining a reverse shell through the SLmail exploit, we were able to access details of the scheduled tasks within the Windows 10 Workstation.

Images	
Affected Hosts	172.22.117.20
Remediation	Limit access to who can view and change scheduled tasks within Windows.

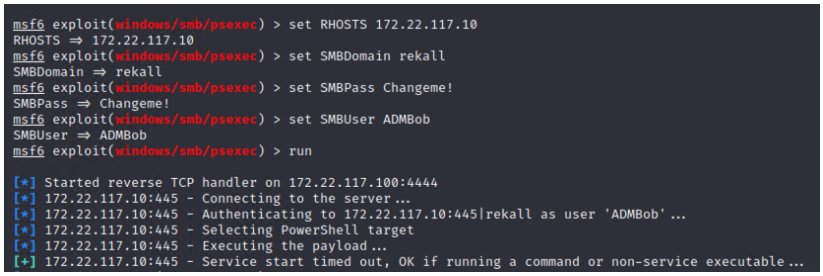
Vulnerability 33	Findings
Title	Windows Flag 6: Insecure password
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Critical**
Description	After compromising SLMail using Metasploit, the Meterpreter shell was the SYSTEM user. Kiwi was then loaded to run the lsa_dump_sam, which dumped

	hashed credentials that were later cracked by John the Ripper, giving us the plaintext passwords.
Images	
Affected Hosts	172.22.117.20
Remediation	Change all existing user passwords and implement a universal, secure password policy.

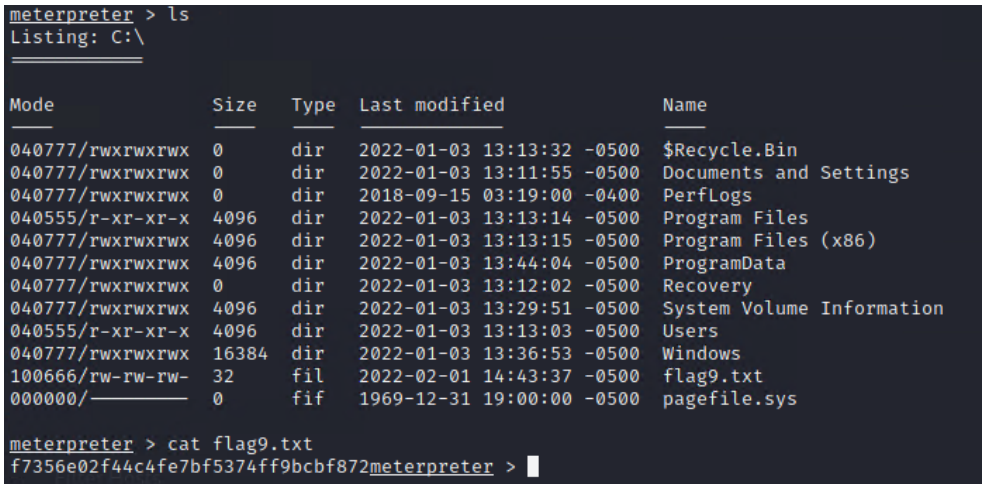
Vulnerability 34	Findings
Title	Windows Flag 7: Insecure file security measures
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	Within the same Meterpreter session from the SLmail exploit, we began simply

	exploring different directories and files to see what we can find. In the public documents folder was sensitive information.
Images	
Affected Hosts	172.22.117.20
Remediation	Keep sensitive information to more obscure and secure areas and restrict access to unauthorized users.

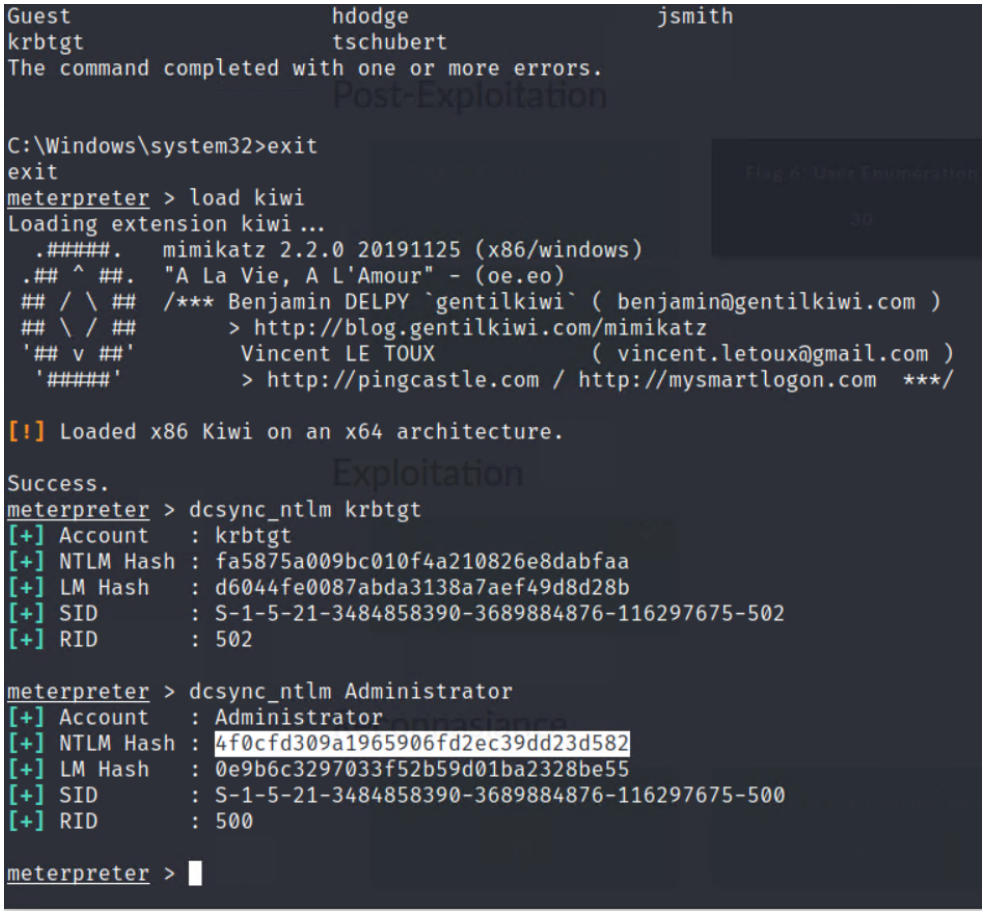
Vulnerability 35	Findings
Title	Flag 8: Microsoft Windows Authenticated User Code Execution, PsExec
Type (Web app / Linux OS / WIndows OS)	Windows OS

Risk Rating	Critical
Description	Using kiwi to dump the cached credentials on Win10 revealed that an administrator, ADMBob, had their credentials cached. We then cracked it with *John* to reveal the password: Changeme! This new credential has access to the Server2019 machine. By using the PsExec module in Metasploit with this credential, a SYSTEM shell was obtained on Server2019. By entering a command shell within Meterpreter, we can list the users with *net user.*
Images	  
Affected Hosts	172.22.117.20 and 172.22.117.10
Remediation	Implement stronger password policies. Restrict approval for PsExec Operations, and close unauthorized access to port 445.

Vulnerability 36	Findings
Title	Windows Flag 9: Remote code execution
Type (Web app / Linux OS / WIndows OS)	Windows OS

Risk Rating	High
Description	With recent access to the domain controller, we began to once again investigate the directories and files to see what sensitive information we can find. By moving to the root, C:\, and listing the files, we found insecure data.
Images	
Affected Hosts	172.22.117.10
Remediation	Keep sensitive information to more obscure and secure areas and restrict access to unauthorized users.

Vulnerability 37	Findings
Title	Windows Flag 10: Remote code execution
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Using kiwi to DCSync the Administrator user on the domain controller, we

	revealed their NTLM password hash.
Images	
Affected Hosts	172.22.117.10
Remediation	Implementing Microsoft Credential Guard, which helps protect domain credentials by utilizing virtualization-based security to isolate secrets so that only privileged system software can access them.