| 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** | CyberSafe Corp |
| --- | --- |
| **Contact Name** | James Nguyen |
| **Contact Title** | CISO |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 08/27/2023 | James Nguyen |  |

# 

## 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.

* The website has some preventative measures that include restricting uploads or restricting certain commands/injections
* There have been some OS as well as application done

### Summary of Weaknesses

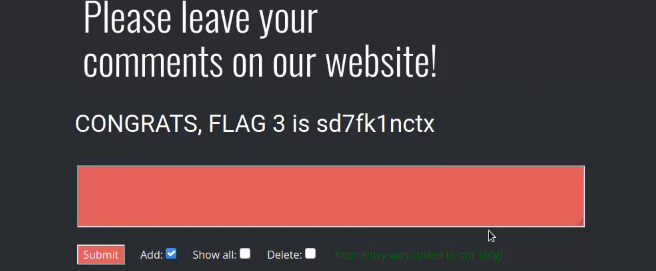
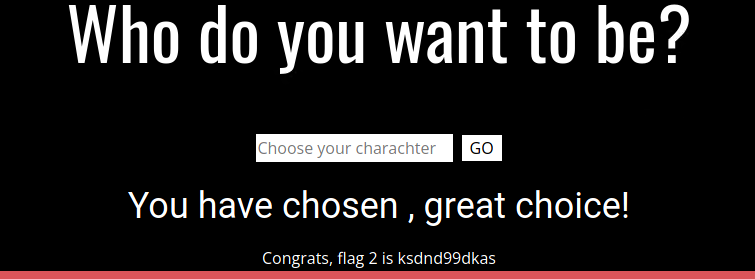
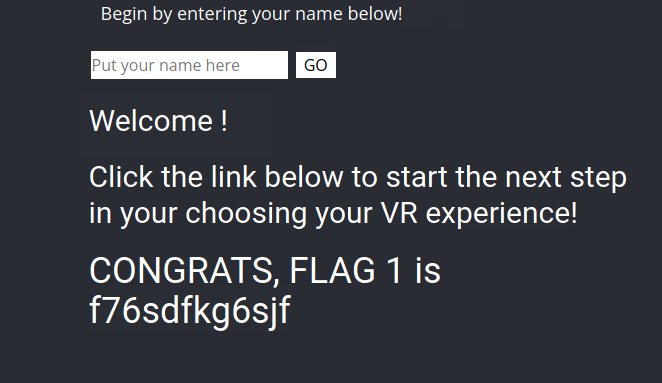
We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

* Passwords for everything on the site from credentials, ssh, server access is weak, making them vulnerable to being compromised
* Weak permissions allow for sensitive data to be exposed
* Old server versions allow remote access and compromise of server and data
* Website has inconsistencies that allow remote code input, injections, file uploads that could be malicious and cross site scripting
* Information can be publicly found and exposes sensitive information

## Executive Summary

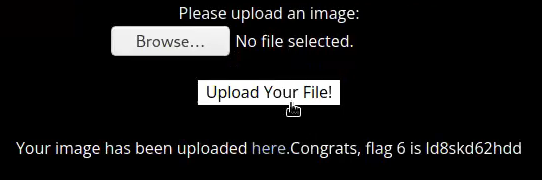
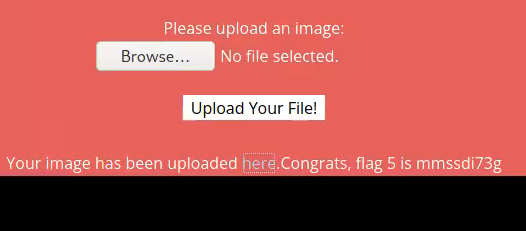
[Provide a narrative summary of your steps and findings, including screenshots. It’s fine to mention specifics (e.g., used Metasploit to exploit a vulnerable version of DistCC), but do not get too technical in these specifics. This should be an A–Z summary of your assessment.]

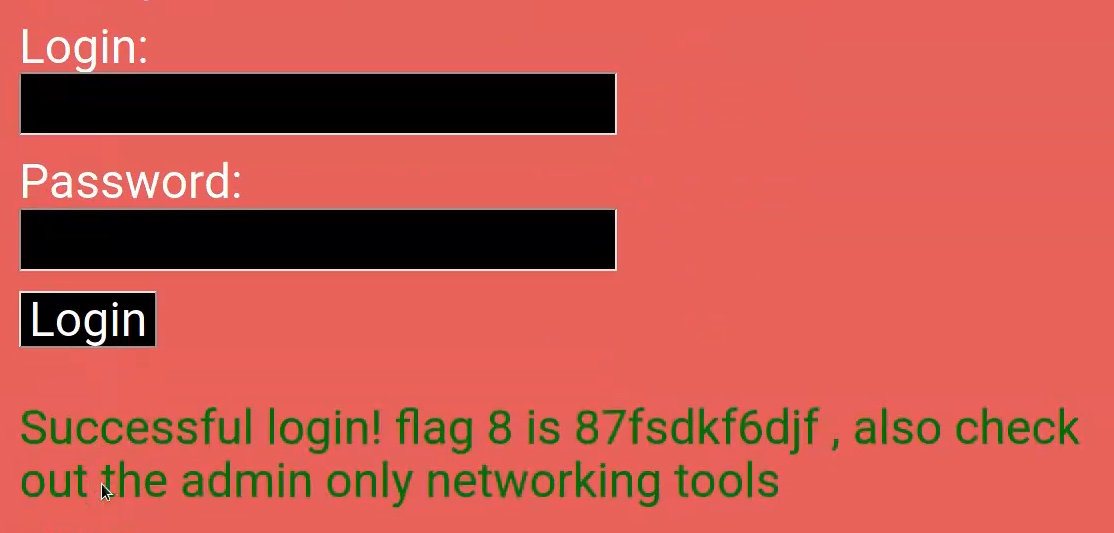
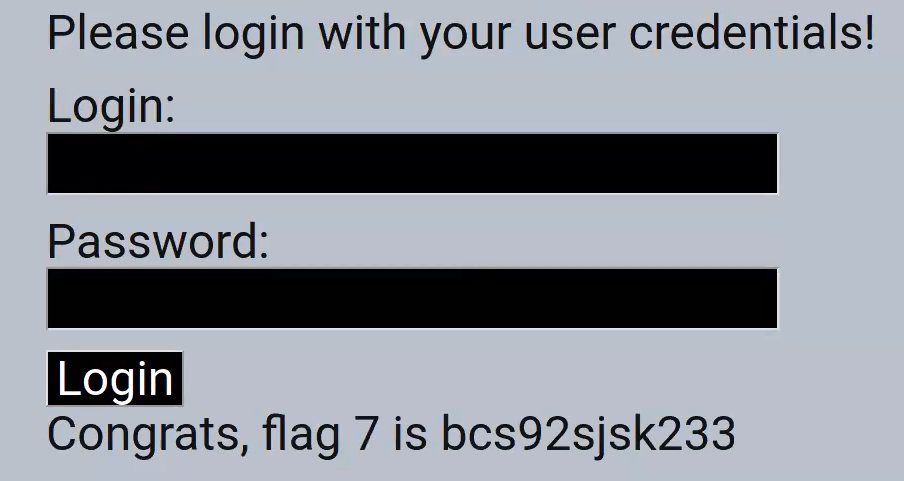
Rekall Corporation is highly vulnerable to attacks. There were compromises to be found at every corner, whether it was in the website, Linux, or Windows. These vulnerabilities were critical in showing sensitive data about employees as well as clients, it granted access to network resources, and had the potential to have malicious codes, files uploaded.

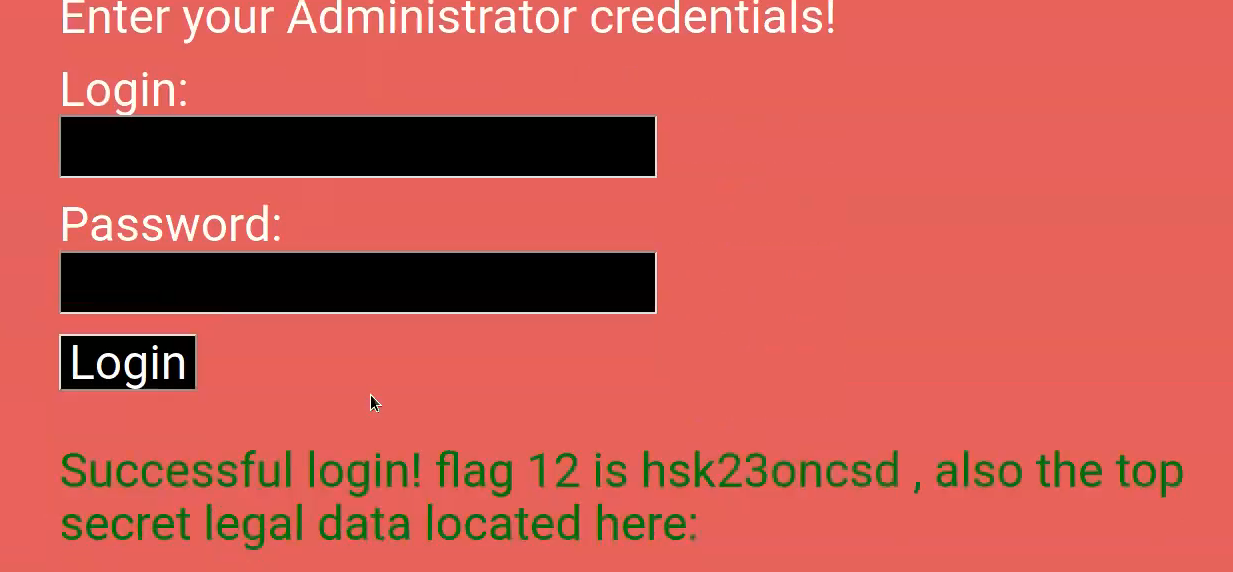
The website was vulnerable to several different types of attacks, starting with cross-site scripting. These XSS injection vulnerabilities were discovered on pages such as the welcome.php, comments.php, and memoryplanner.php

## 

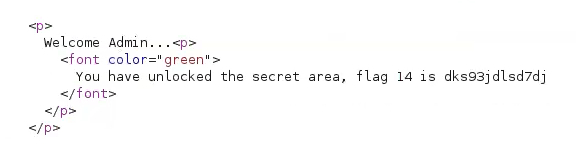
## The website allowed for uploads of ‘malicious’ files and the ability to run unauthorized scripts from different locations and view sensitive information such as a ‘secret legal data’ file. This allowed us to find credentials that allowed us further access.



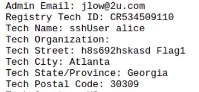


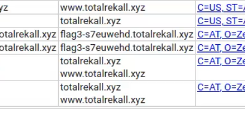
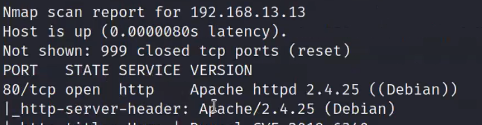


Credentials were also hidden in plainview and exposed within the HTML source code. This could be found within the login.php page.

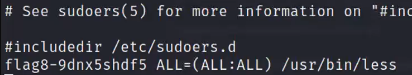
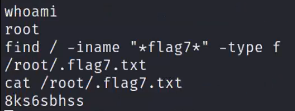


SSH credentials were easily found along with an insecure password that was the exact same as the login ID. The IP address and certificate were able to be looked up with port scans and a simple online search with http://crt.sh. With the use of a nessus scan and aggressive nmap scan, we were able to find vulnerabilities within the system and dig deeper with the tool Metasploit which in turn granted us access to information such as open ports, software versions and IP addresses.



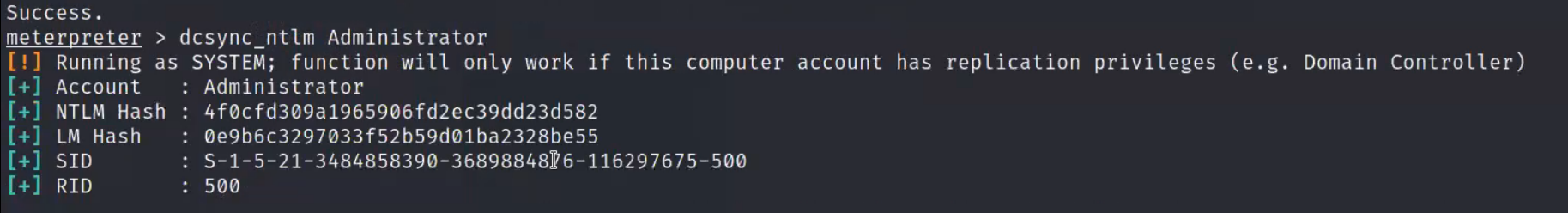
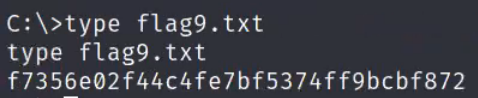
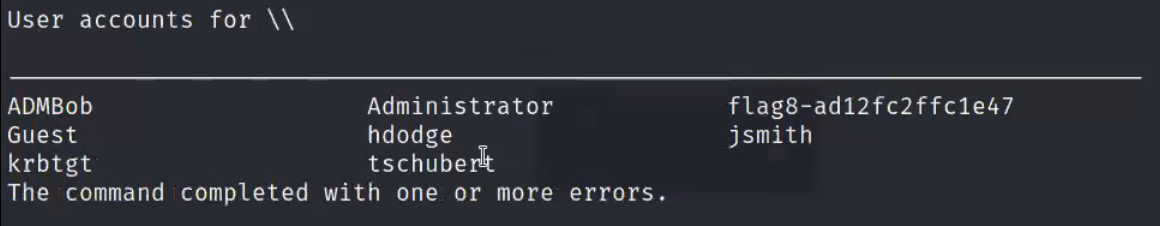
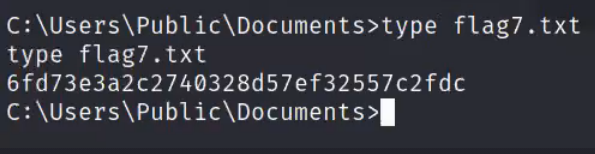
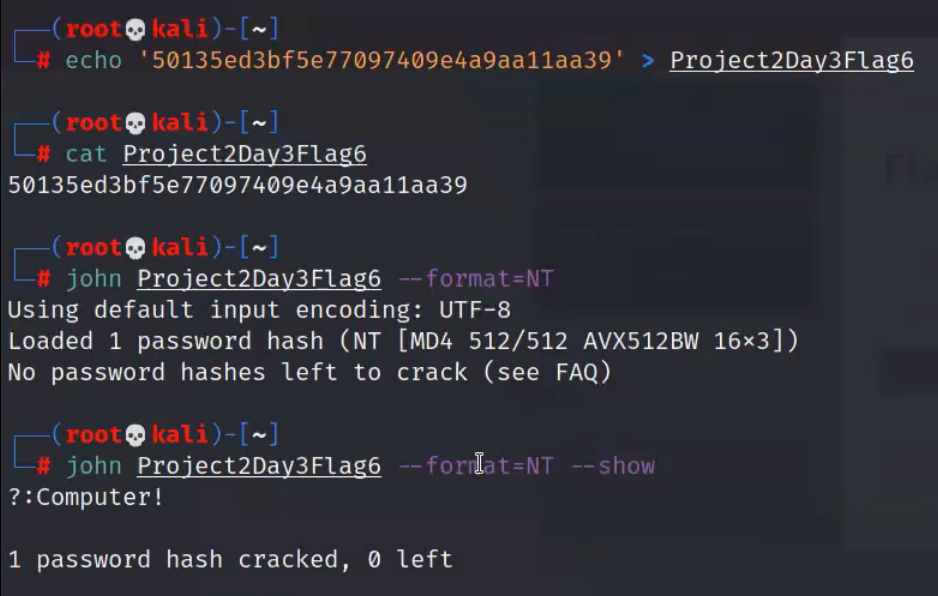
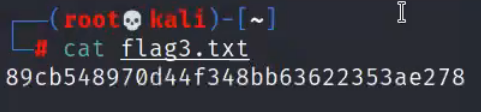
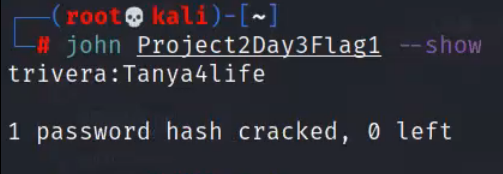


With these ssh credentials, we were able to log in and find vulnerabilities with sudo privileges. With a version vulnerability, you are able to change the user from a normal one into the root user with all root privileges available to them.

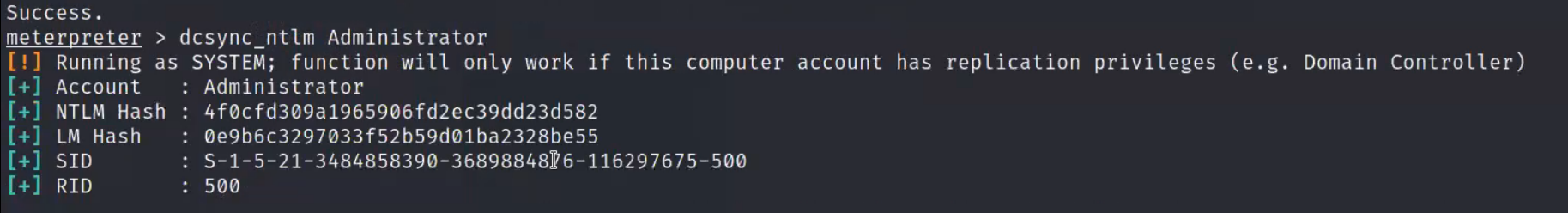
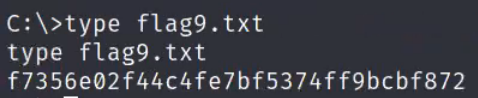
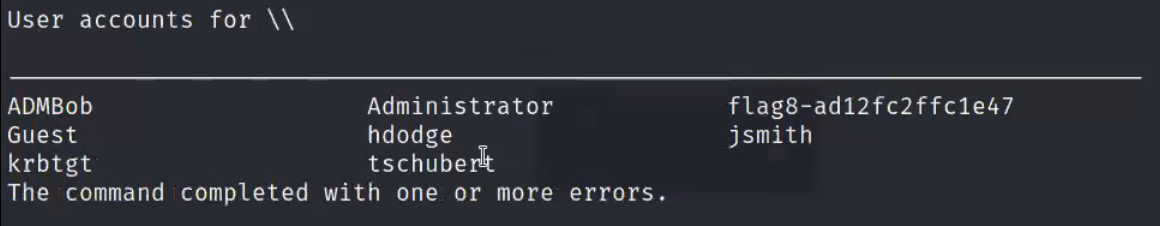
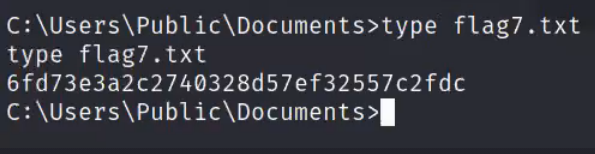
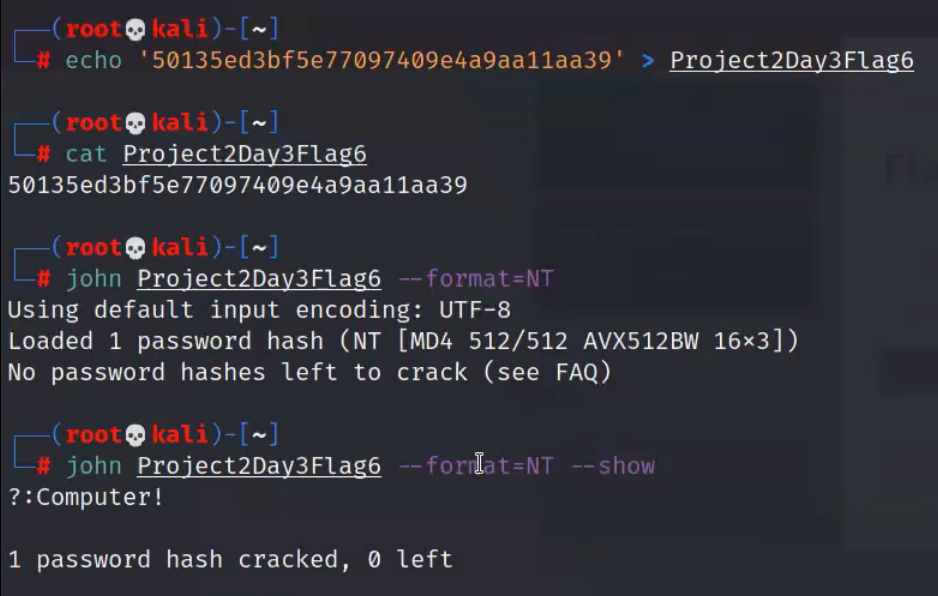
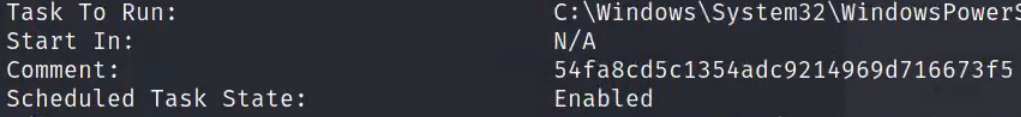
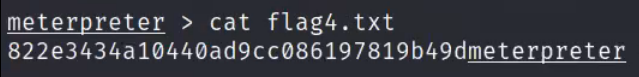
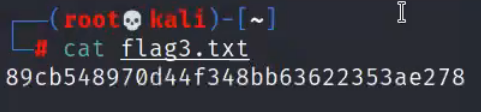
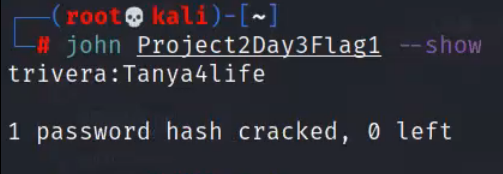




Rekall Corporation had user hashes and passwords out in the open within .txt files and were able to be cracked. And by dumping credentials using the local Security Accounts Manager (SAM) database, we were able to expose the password for the ADMBob account as well.



Overall, Rekall Corporation had unsecure protections spanning across their website as well as servers. Throughout our search, we have found negligence to protect themselves as well as their clients and information. These findings make it hard for customers to find safety for the things most important to them and easy for cybercriminals to virtually do anything that they want with the excess amount of information available to them with the weaknesses within their organization’s website, data, and servers.



## 

## 

## 

## 

## 

## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| SQL Injection | **Critical** |
| Sensitive Data Exposure | **Critical** |
| XSS Reflected | **Critical** |
| XSS Stored | **Critical** |
| Local File Inclusion | **Critical** |
| Command Injection | **Critical** |
| Brute Force Attack | **Critical** |
| PHP Injection | **Critical** |
| Session Management | **Critical** |
| Drupal | **Critical** |
| Open Source Exposed Data | **High** |
| Scan Results | **High** |
| Nessus Scan Results | **High** |
| Apache Tomcat | **High** |
| Shellshock | **High** |
| Struts | **High** |
| Directory Traversal | **Medium** |
| CVE-2019-14287 | **Medium** |
|  |  |
|  |  |
|  |  |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | 192.168.14.35  15.197.148.33  192.168.13.0/24  172.22.117.10  172.22.117.20 |
| Ports | Port 80  Port 8080  Port 8009  Port 22  Port 21  Port 110 |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 10 |
| **High** | 6 |
| **Medium** | 2 |
| **Low** | 0 |

## Vulnerability Findings

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | XSS Reflected |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Cross Site Scripting able to be used on multiple sections across the web app:   * Welcome.php * Comments.php * memoryplanner.php |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Sanitize your output data after coding with OWASP libraries |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | XSS Stored |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Used <script>alert(“hi”)</script> to reveal |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Reject and sanitize data that doesn’t meet criteria or expected format |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Critical information appears within the header of the website |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Notify affected parties, patch this section of HTML with corrected code |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Local File Inclusion |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | We were able to bypass security just by renaming the script with .jpg |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Implement a Web Application Firewall to block these requests |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | SQL Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | We used the payload 1=1-- |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Review code and prepare libraries of accepted queries to prevent anything other than appropriate parameterized queries |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Command Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | www.welcometorecall.com && cat vendors.txt |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Similar to an SQL injection, we must pay attention to code and use parameterized queries and libraries to prevent these attacks |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Brute Force Attack |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | username: melina password: melina within the /etc/passwd file |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Implement a lockout policy after a certain amount of attempts or 2FA/MFA |

Add any additional vulnerabilities below.

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** | PHP Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Changed the URL using http://192.168.13.35/souvenirs.php?message=""; system('cat /etc/passwd') |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Same as both command and SQL injection attacks |

| **Vulnerability 9** | **Findings** |
| --- | --- |
| **Title** | Session Management |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Used BurpSuite to test different session ID’s http://192.168.13.35/admin\_legal\_data.php?admin=87 |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Audit and monitor active sessions |

| **Vulnerability 10** | **Findings** |
| --- | --- |
| **Title** | Directory Traversal |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | Changed the URL to http://192.168.13.35/disclaimer.php?page=old\_disclaimers/disclaimer\_1.txt |
| **Images** |  |
| **Affected Hosts** | 192.168.35.14 |
| **Remediation** | Create stronger access controls to ensure that only authorized users are able to access sensitive data and functions |

| **Vulnerability 11** | **Findings** |
| --- | --- |
| **Title** | Open Source Exposed Data |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | The Domain Dossier website and crt.sh were able to pull up all information |
| **Images** |  |
| **Affected Hosts** | All |
| **Remediation** | Secure, take down and restrict access to the data until it is safe and under control |

| **Vulnerability 12** | **Findings** |
| --- | --- |
| **Title** | Scan Results |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | Nmap and an aggressive nmap scan were able to determine the hosts |
| **Images** |  |
| **Affected Hosts** | All |
| **Remediation** | Close and reconfigure open ports |

| **Vulnerability 13** | **Findings** |
| --- | --- |
| **Title** | Nessus Scan Results |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | The Nessus scan was able to find the critical vulnerability and found the ID |
| **Images** |  |
| **Affected Hosts** | All |
| **Remediation** | Same as scan results, close and reconfigure open ports |

| **Vulnerability 14** | **Findings** |
| --- | --- |
| **Title** | Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617) |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | Using the exploit multi/http/tomcat\_jsp\_upload\_bypass we are able to get a Meterpreter shell and command line |
| **Images** |  |
| **Affected Hosts** | All |
| **Remediation** | Update software to latest version to prevent old version vulnerabilities |

| **Vulnerability 15** | **Findings** |
| --- | --- |
| **Title** | Shellshock |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | MSFconsole was able to find an exploit that has Shellshock |
| **Images** |  |
| **Affected Hosts** | All |
| **Remediation** | Update your systems to the latest versions |

| **Vulnerability 16** | **Findings** |
| --- | --- |
| **Title** | Struts - CVE-2017-5638 |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | Nessus scan and MSFconsole were able to determine that there was a Struts vulnerability and exploit |
| **Images** |  |
| **Affected Hosts** | All |
| **Remediation** | Update your Apache Struts to a more secure, patched version |

| **Vulnerability 17** | **Findings** |
| --- | --- |
| **Title** | Drupal - CVE-2019-6340 |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Medium |
| **Description** | MSFconsole found Drupal exploits and using a Meterpreter shell, run getuid to find the username |
| **Images** |  |
| **Affected Hosts** | All |
| **Remediation** | Update the Drupal core to the latest version |

| **Vulnerability 18** | **Findings** |
| --- | --- |
| **Title** | CVE-2019-14287 |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Medium |
| **Description** | By using sshuser Alice, we were able to gain root privileges through an privilege escalation exploit |
| **Images** |  |
| **Affected Hosts** | All |
| **Remediation** | Identify systems running this vulnerable version of sudo and update to a patched and secure version |