| Cybersecurity |
| --- |
| Penetration Test Report |

Rekall Corporation

Penetration Test Report

# 

## Confidentiality Statement

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

Table of Contents

[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)1

Vulnerability Findings [1](#_1ci93xb)2

# 

## Contact Information

| **Company Name** | Badger Security, LLC |
| --- | --- |
| **Contact Name** | Charles Bigger |
| **Contact Title** | Pentester |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 05/08/2023 | Charles Bigger |  |

# 

## 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 a perspective of the network’s 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 manually test each identified vulnerability and use automated tools to exploit these issues. The exploitation of a vulnerability is defined as any action we perform that gives us unauthorized access to the system or 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 input fields on the majority of the Rekall website are using some form of input validation
* It took a significant amount of time and numerous attempts to find an input field that could be breached through command injection.
* The team attempted numerous Metasploit exploits before finding one that was successful.

### 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.

* Rekall’s web application is vulnerable to XSS and SQL payload injection.
* User credentials have been stored in the HTML source code.
* The Apache server in use has a vulnerability to remote code execution in the Jakarta Multipart Parser.
* The SLMail server is vulnerable to Metasploit exploits that allow bad actors to gain shell access.
* The team was able to gain access to password hashes allowing password cracking and privilege escalation.
* NMAP scanning revealed open ports.
* Credentials were revealed while conducting IP scanning.
* The discovered open ports allowed the team to conduct file enumeration and gain unauthorized access.

## Executive Summary

Badger Security, LLC conducted a thorough penetration test of Rekall’s network systems and web application finding several critical vulnerabilities that are recommended to be immediately addressed in order to prevent any bad actors from compromising the network. The Badger Security team was able to infiltrate Rekall’s assets, exfiltrate sensitive data, and escalate privileges within their systems, as detailed below.

Our team began the testing with Rekall’s Web Application first. It was determined to be vulnerable to XSS Reflected attacks with the team running malicious script on the home page. The Web App was also identified as having several vulnerabilities: Local File Inclusion vulnerability as files can be uploaded from the VR Planner web page, an XSS Stored vulnerability was identified on the Comments page allowing the team to run scripted code, SQL Injection vulnerabilities were found on the Login.php toolbar, as well as the Networking.php page is vulnerable to a Command Injection attack.

The team using open-source research was able to locate an exposed password on the public Rekall GitHub site. They were then able to use this data to infiltrate a Windows 10 machine where they discovered that FTP Port 21 and Port 110 were open and vulnerable. Port 110 is used for the SLMail server which Rekall is using for its email which has a vulnerability that should be corrected. The team was able to take advantage of this vulnerability to gain access to a password hash file which was subsequently cracked and enabled the creation of a reverse shell. Additionally, scheduled tasks were found within the Windows 10 Machine Task Scheduler allowing the team to use Metepreter to display directories through public Windows directories.

The vulnerabilities exposed by Badger Security’s team could be exploited maliciously to cause massive and irreparable damage to Rekall’s network assets and curtail the functionality of the business in general. The detailed vulnerability findings section provides descriptions of each of the discovered vulnerabilities found and suggestions for mitigation.

## 

## 

## 

## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| Cross Site Scripting (XSS) | **Critical** |
| Local File Inclusion (LFI) | **Critical** |
| Command Injection | **Critical** |
| Apache Struts 2.3.5-2.3.31/2.5x<2.5.10.1 Jakarta Multipart parser RCE  (remote) | **Critical** |
| Exposed data on totalrekall Public Github website | **High** |
| SLMail POP3 | **Medium** |
| Open Source exposed data | **Medium** |
|  |  |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | 34.10.2.136.180 - totalrekall.xyz  192.168.13.10 - Linux  192.168.13.11 - Linux  192.168.13.12 - Linux  192.168.13.13 - Linux  192.168.13.14 - Linux  172.22.117.20 – Windows10  172.22.117.10 – Windows Domain Controller |
| Ports | 80 (HTTP) 21(FTP), 25(SMTP), 110 (POP3), 135 (RPC), 8009 (TCP), 8080 |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 4 |
| **High** | 1 |
| **Medium** | 2 |
| **Low** | 0 |

## Vulnerability Findings

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | Cross Site Scripting (XSS) |
| **Type (Web app / Linux OS / WIndows OS)** | Web application |
| **Risk Rating** | **Critical** |
| **Description** | The team was able to successfully insert alerts into the input fields for, “Begin by entering your name below” and “Comments” on the Welcome page, as well as, in the “Choose Your Character” field of the Memory Planner page using cross-site scripting.  The Memory page field included some input validation settings,  but the team was able to eventually route around them with minor changes to the script input using <SCRIPscriptT> instead of <script>.  The ability to insert these scripts would allow bad actor to conduct multiple exploits including redirecting customers to a spoofed web page, installing a key logger, or capture a user’s cookies. This would facilitate the theft customer data for use to gain unauthorized entry into Rekall’s system for further attacks. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | client side and server side input validation |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | Local File Inclusion (LFI) |
| **Type (Web app / Linux OS / WIndows OS)** | Web application |
| **Risk Rating** | **Critical** |
| **Description** | The team was successful in uploading a custom .php script file using the file upload functionality of the Memory Planner page. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Prevent file paths from being able to be appended directly and, if possible, restrict API to allow inclusion only from a directory and the directories below it. |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | Command Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web application |
| **Risk Rating** | **Critical** |
| **Description** | The team successfully injected the command /whoami.txt into the user name input fields on the login page, obtaining the user name and password for an administrator. Using that login data the team discovered hidden Networking page that contained information about the location of sensitive data contained in the vendors.txt file. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Never call out to OS commands from application-layer code – limit the  use of command execution functions  Use an API to filter input  Stringent input validation against a whitelist of permitted values  Validating input is only alphanumeric characters or numbers with no other  syntax or white space |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Apache Struts 2.3.5-2.3.31/2.5x<2.5.10.1 Jakarta Multipart parser RCE  (remote) |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | Nmap and Nessus scans revealed vulnerability on Apache server – Apache  Struts 2.3.5-2.3.31/2.5x<2.5.10.1 Jakarta Multipart parser RCE (remote). RCE  exploits can result in allowing attackers to run programs, retrieve source code  or exfiltrate data.  The team successfully employed a Remote Code Execution (RCE) script through Metasploit to exploit this vulnerability. They used script -  multi/http/tomcat\_jsp\_upload\_bypass to gain access to Linux machine  192.168.13.10 and retrieved the data from the /etc/password and /etc/shadow  files which contain sensitive user and password data and permissions  information. |
| **Images** |  |
| **Affected Hosts** | 1920168.13.10 |
| **Remediation** | Upgrade to the latest version of Apache Struts |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Exposed data on totalrekall Public Github website |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | The team discovered the public GitHub site for totalrekall had an exposed hash for a user. After cracking the hash the team performed nmap scans of the IP range for the network and were able to identify the IPs for a Windows host, Windows 10 machine and the Domain Controller. Through this process, they also discovered that port 80 (http) was open on the Windows10 machine. The team then used the credentials found on the GitHub site to log in to the Windows 10 machine. Performing an aggressive nmap scan revealed that port 21 (FTP) was also open and allowed anonymous access. We were able to log in to the Windows 10 machine through port 21. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 – Windows10  172.22.117.10 – Windows Domain Controller  172.22.117.100 – Windows host |
| **Remediation** | Remove hashes from the Github site  Salt hashes to make them more difficult to crack  Require complex passwords that are regularly updated  Switch to FTPS or SFTP which are more secure than standard FTP  which is vulnerable to sniffing, spoofing and brute force attacks. |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | SLMail POP3 |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Medium |
| **Description** | Using Metasploit, the team used Metasploit exploit/windows/pop3/seattlelab\_pass to gain entry through port 110 on the Windows10 machine. Once access was gained, they were able to view files and permissions. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 – Windows10 |
| **Remediation** | Convert to IMAP using port 143 and close port 110. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Open Source Exposed Data |
| **Type (Web app / Linux OS / WIndows OS)** | Web application |
| **Risk Rating** | Medium |
| **Description** | The team discovered this vulnerability while viewing WHOIS data on the Domain Dossier webpage, which contained sensitive information for the domain totalrekall.xyz. |
| **Images** |  |
| **Affected Hosts** | 34.10.2.136.180 - totalrekall.xyz |
| **Remediation** | Ensure no sensitive data is being shared publicly and sanitize WHOIS records. |