| 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)3

Vulnerability Findings [1](#_1ci93xb)4

# 

## Contact Information

| **Company Name** | Good Times LLC |
| --- | --- |
| **Contact Name** | Colin O’Donoghue |
| **Contact Title** | CISO of Good Times |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 08/02/23 | Colin O’Donoghue | Good Times! |
| 002 | 08/03/23 | Colin O’Donoghue | Good Times Again! |
| 003 | 08/07/23 | Colin O’Donoghue | Good Times Once More! |
| 004 | 08/14/23 | Colin O’Donoghue | Good Times 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.

* Our team was unsuccessful in the desired goal for Windows OS penetration, thus proving that there were positive security aspects in place.
* Having this exercise be performed speaks to the strength of Rekall’s wanting to invest in the safety of their sensitive 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.

* There are fundamental flaws in the security of the Web Application that can lead to a flood of unwanted traffic.
* Our team found user credentials for Rekall in ways that anyone can access the information publicly.
* There were IP Addresses that were vulnerable to being exploited by means of open ports, granting unwanted access from threat actors.
* The use of gihub.com to store information that led to a password being cracked was a simple way of gaining access to potentially escalate privileges.

## Executive Summary

Good Times LLC was founded on being able to assist the everyday company with security measures that may need to be implemented to ensure the longevity and success for all of those involved within the company umbrella. There are an abundance of cyber attacks that happen on a daily basis, and we pride ourselves in offering the best services for mitigating these attacks and ensuring our clients feel confident and secure regarding the wellbeing of their private information. It is our goal to exploit as many vulnerabilities as we can to show our clients the weak points in their networks, applications, and operating systems so that they can take the necessary steps in locking down their exposures to threat actors.

This report will reveal a number of different techniques used to gain access into a system, but mostly we will focus on the exposures of sensitive data that could potentially lead to further, more advanced exploitations by experienced hackers. Sometimes, a company or person is confident that their information is safe, but these tests will prove that even if some files are hidden, they can still be accessible. We were successful in most of our exploitations, but there were some potential vulnerabilities that we could not exploit. With the allocated time our team was given, we had more success in finding data through port scanning, SQL injections, and using public sources to find information that could be used against Rekall in the future. The main reason we started there was because if this data is available to the public, imagine what a seasoned hacker would do to find more confidential data.

We began with maneuvering through the Web Application and found admin credentials in HTML as well as using SQL injection to return unauthorized query information. Also, we used ‘robot.txt’ in the URL to find hidden files that can be used against Rekall. When this was completed, we went into the terminal and ran nmap and nessus scans to find 5 vulnerable hosts and open ports with nessus finding at least 12 vulnerabilities. We also spent time on 2 public domains (GitHub and Domain Dossier) to find more useful information when cracking a user’s password and another regisitrant’s contact information.

The seven vulnerabilities listed below are a good jumping off point for Rekall to evaluate the strength of their security. We have also provided detailed descriptions of our findings and listed a few recommended mitigation strategies to review for guidance to properly ensure that these attacks do not happen. Our team focussed hard on the Reconnaissance aspect of this penetration test and found that sensitive data was indeed there for the taking if the tester had fundamental knowledge of where to begin. We found everything from user credentials, password hashes, and open ports with vulnerable hosts during this test and a number of these findings could lead to a user escalating privileges and developing persistence within the company network. Once a hacker is in, they can most certainly stay in and go undetected if these mitigations that we provided are not utilized.

Overall, there were some security measures in place that are quite admirable, but also there are some things to clean up. Below is the list of vulnerabilities that Good Times LLC focused on with a detailed description of what happened.

## 

## 

## 

## 

## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| Sensitive Data Exposure | **Critical** |
| Cracked Password | **Critical** |
| Vulnerable Hosts - Open Ports (Network) | **Critical** |
| Nessus Scan - Results | **High** |
| SQL Injection | **High** |
| Open Source Exposed Data | **Medium** |
| Sensitive Data Exposure | **Medium** |
|  |  |
|  |  |
|  |  |
|  |  |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | (11):  34.102.136.180  172.22.117.20  172.22.117.10  172.22.117.100  192.168.13.1  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 | (5):  21 - FTP  22 - SSH  80 - HTTP  110 - POP3  8080 - TCP |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 3 |
| **High** | 2 |
| **Medium** | 2 |
| **Low** | 0 |

## Vulnerability Findings

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | During this exercise, our team was looking for any data that may seem useful for future penetration practices. Navigating the website was simple without having to sort through too many pages. The web pages showed to have no sensitive data exposed to the public on the front end, so we checked the back end by keying ‘CTRL + U’ that opened up the HTML page linked to each page on the web app. When we tried this while on the Login page, we were able to find valid credentials for a user inside the HTML page. The user ‘doug quaid’ and their password ‘kuato’ were displayed in the code, and when we entered in the values, we were able to gain access to have Administrator privileges. |
| **Images** |  |
| **Affected Hosts** | http://192.168.14.35/Login.php |
| **Remediation** | HTML Sanitization - inspect the HTML and remove unwanted content. |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | SQL Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | While still on the Login page, we were attempting to test the vulnerability of another login field for regular users. We were unable to find more credentials, so we began using a technique called SQL Injection to test the strength of the login fields. SQL Injection is used to find data by entering in malicious SQL code and having the field return data that was not meant to be displayed. In this case, our team was successful by entering “‘OR’ 1=1” into the password field and were able to determine that this would return information from the query that would not be intended for an unauthorized user. |
| **Images** |  |
| **Affected Hosts** | http://192.168.14.35/Login.php |
| **Remediation** | Input Validation - always validate data entered by the users to meet a certain criteria. |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | Moving into exposing more sensitive data, our team went forth to try to find files that would pertain to any more useful information to exploit the web app. During the ending phase of our reconnaissance, one member took it upon themselves to begin a process of entering in .txt files into the URL field. After a few attempts, we were able to publicly access the robots exclusion standard settings using the ‘robots.txt’ link. This provided us a list of other pages we could access by simply entering the file link at the end of the URL for a greater look inside of more potential vulnerable files. This is a fundamental technique that can be very harmful for a company if the only thing a threat actor needs to do is to try to enter in common file names at the end of a URL. Most advanced hackers will exploit this early and then it can be too late. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35. |
| **Remediation** | Securing File Privileges - set appropriate permissions for file access. |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Vulnerable Hosts - Open Ports (Network) |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | After our experience with the Web App, we shifted focus to the Linux Machine and saw what kind of hosts and ports may be vulnerable to an attack. Linux can be very useful for hackers since a lot of the tools used for scanning, exposing information, and exploiting that information are already installed and can show the results in a simple, easy-to-read format. To scan the network for hosts and ports, our team used the nmap command and ran it against a subnet of IP addresses. When we ran ‘nmap 192.168.13.0/24’, the results showed that 5 hosts with their open ports were exposed to attacks. This is critical because these IP addresses with corresponding open ports are the gateways for a hacker to gain access and establish persistence. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14, 192.168.13.1 |
| **Remediation** | IP Blocking - ensure unauthorized users cannot access the computer network. |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Nessus Scan - Results |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | Another tool we were able to successfully use was the Nessus scan. Nessus is another scanner that is used to identify any vulnerabilities in computer systems, as well as networks and applications. The process involves port scanning and shows the user running the scan a list of vulnerability detections that a person can exploit and use to their advantage. When a Nessus scan for 192.168.13.12 was run, we found 12 vulnerabilities. The highest threat was that the Apache server was vulnerable to Remote Code Execution (RCE) exploits that can be used for privilege escalation and denial of service (DoS) attacks. The other remaining 11 vulnerabilities were not further researched, however it would be best to go back and revisit them to ensure stronger security in the network. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Upgrade Apache Struts - this will increase security and patch vulnerabilities. |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Open Source Exposed Data |
| **Type (Web app / Linux OS / WIndows OS)** | Web App OS |
| **Risk Rating** | Medium |
| **Description** | While finding more information on the target, we came across a website entitled “Domain Dossier” to see what kind of information it could provide for our exercise. This website has a lot of useful information regarding IP addresses, DNS records, and in this case, WHOIS data. When searching for the information regarding ‘totalrekall.xyz’, our team was able to find certain contact information for one particular registrant that could be useful for future exploits. Although this is not necessarily a direct threat to any major information, it is personal information nonetheless and should not be as easy to access. |
| **Images** |  |
| **Affected Hosts** | https://centralops.net/co/DomainDossier.aspx |
| **Remediation** | Edit WHOIS records - ensure no sensitive data is being shared publicly. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Cracked Password |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | During our search into more public forums for useful information, our team decided to look into GitHub, an available resource for anyone that provides an environment for storing and sharing data for anyone to see and use. A lot of coding files and projects are stored here. Upon searching for totalrekall.xyz’s GitHub page, our team found a link to the ‘xampp.users’ page that allowed us to enter into a repository regarding user credentials. Inside of the file, we found a hash that is consistent with the ones that are used for passwords. Extracting that hash, opening a terminal in Linux, storing the hash and using John the Ripper, our team was able to obtain the password for trivera. The password found was ‘Tanya4life’ This is a critical discovery because using a public source to store user credentials is an extremely easy way for hackers to crack a hashed password and use the user’s information against them and/or confidential information. |
| **Images** |  |
| **Affected Hosts** | totalrekall.xyz |
| **Remediation** | Strong Password Policies - implement a more frequent password changing policy that requires the length of the password to be longer and have requirements for capital letters and special characters. |