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

Contact Information

Company Name	Next Level Cyber Group Pty LTD
Contact Name	04550999
Contact Title	

Document History

Version	Date	Author(s)	Comments
001	27/01/2023	Chukwuebuka Umenwa	

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.

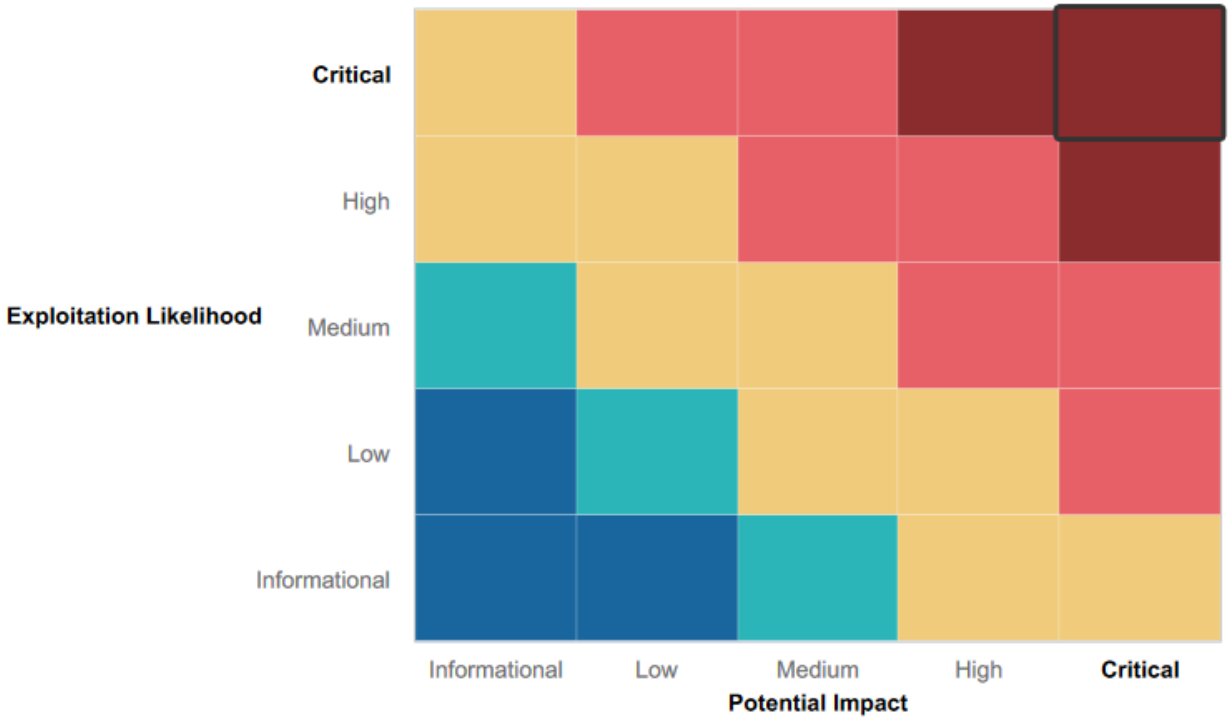
Executive Summary of Findings

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.

- During this engagement, I discovered that the company has some security controls in place such as physical security for instance. The company has a perimeter fence and takes record of every movement in and out of the organizations.
- There is a swipe card system in place where every employee has to use to gain access into the property.
- The server room is isolated and well protected.
- There are firewalls and antiviruses configured on every computer in the company.
- At Rekall, staff members are not allowed to take their work devices home and they do not come to work with their own devices.
- Email threat protection system is in place

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.

- I was able to conduct cross site scripting on the company's web app.
- I was able to brute force and crack some user passwords
- I was able establish some metasploit shell sessions such as shell shock, drupal and SLMail
- I was able to carry out command injection on the company's web application
- I was able to do SQL injection
- I was able to do some regular command injection on the linux server
- I was able to carry out some privilege escalation, granting me access to root
- I was able to find exposed GitHub credentials
- Was able to exploit anonymous FTP
- Apache web server is outdated and hackers can exploit this vulnreability
- When we carried out an Nmap scan, we discovered that there are some open ports that ordinarily should not be open.
- The servers physical address is publicly available
- Credentials are stored in HTML

Executive Summary

During the course of this penetration testing exercise we discovered so many vulnerabilities that may expose Rekall to some serious cyber attacks if not swiftly addressed. While most of them are critical some are low risk however, ever vulnerability should be considered a potential threat
This exercise was carried out following industry's best practices. We worked within the scope and objectives of this exercise. During this period, we did not expose any of Rekall's data to any risk. The whole test environment was contained. All the files needed for this exercise were backed up and were securely saved in a different location. We worked within the timeframe required and made sure this report was delivered ahead of schedule. In line with our privacy policy, this report and every attachment related to this report has not been seen by unauthorized persons, companies or groups.
We started this exercise by gaining access into Rekall infrastructure, maintained our presence undetected within the system. We enumerated within the system, gathered vital information and escalated our privilege to have full control over the server.
On the comment page of Rekall's web application we were able to conduct a cross site scripting where we injected some malicious scripts and were able to execute them successfully. Local file inclusion vulnerability was also noticed as we were able to upload files from the VR planner web page. On the login.php page we were able to carry out SQL injection. On the networking.php page we were able to conduct a command injection attack.
While continuing with the testing, we used OSINT and crt.sh to see Rekall's stored certificate. We also discovered that some login credentials were stored in the HTML source code of the login.php page. This is not a good practice as bad actors can easily have access to those credentials and use them to hack the organization's infrastructure. Going further we uncovered the apache struts vulnerability which is a result of the apache server not being up to date. It is encouraged that servers should be up to date at all times.
While testing the windows environment, we found some other vulnerabilities. For instance FTP port 21 was open and port 110 used for SLMail. These vulnerabilities can easily be exploited by bad actors to hack Rekall. Furthermore, using the metasploit tool, we were able to access password hashes which we cracked using John the ripper tool, having cracked these passwords we were able to establish a reverse shell connection to the system. Going forward, we recommend the implementation of a strong password policy and enabling two factor authentication.
On the linux environment we discovered that one of the hosts was running a drupal. Some IP addresses were exposed to the public which is vulnerable to attacks at any time. We equally used a stolen credential to gain access into one of the hosts. When we gained access, we were able to elevate our privileges to a root level. Shellshock exploit in metasploit was used to access the sudoer files.
In conclusion, these findings of vulnerabilities should be addressed as soon as possible. First the company should look at the most critical vulnerabilities and address them. While measures are to be put in place to address these vulnerabilities, however a very key important aspect of cybersecurity is the human asset. There should be continuous education of the Rektall workforce on cybersecurity. Having every vulnerability addressed and staff not being security conscious will amount to waste of time and resources.

Summary Vulnerability Overview

Vulnerability	Severity
Shell shock exploit	critical
Brute force	critical
SQL injection	High
Privilege escalation	Critical
Cross site scripting	High
Apache strut	Critical
Command Injection	Medium

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

Scan Type	Total
Hosts	172.22.117.20 172.22.117.10 192.168.13.10 192.168.13.11 192.168.13.12 192.168.13.13 192.168.13.14 192.168.14.35
Ports	21 22 80 106 110

Exploitation Risk	Total
Critical	4
High	2
Medium	1

Low	0

Vulnerability Findings

Vulnerability 1	Findings
Title	Brute Force
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	Critical
Description	This is a type of cyber attack that uses trial-and-error to guess or crack passwords, login credentials and encryption keys. I was able to conduct a successful brute force attack on the company's web app using an open source tool known as John the ripper.
Images	https://drive.google.com/file/d/11ZoR4EnqAcbKoB47dOjOCSbWIETQ_9SD/view?usp=share_link
Affected Hosts	192.168.14.35
Remediation	1. Developers should design web applications to require complex usernames and passwords. 2. Developers should design applications to lock out accounts after a number of failed attempts. 3. Rekall should implement multi-factor authentication. 4. Input validation should be implemented by developers

Vulnerability 2	Findings
Title	Cross site scripting
Type (Web app / Linux OS / WIndows OS)	Linux/web app
Risk Rating	High
Description	Attackers use cross-site scripting to inject malicious code into a web page. These scripts when executed can compromise the confidentiality and integrity of data transfers between the website and the client. I was able to carry out this attack while conducting the penetration testing exercise. This vulnerability is a high risk
Images	https://drive.google.com/file/d/1OqYMFqsr2Y5zTXh5_CnI8GEamQojFi9w/view?usp=share_link

Affected Hosts	192.168.13.35
Remediation	1. Input validation is the most common and effective way to mitigate against cross-site scripting. Data input should be filtered strictly based on what is expected or valid input. 2. Another effective way Rekall can mitigate against this is by encoding data on output and using appropriate response headers. 3. Web application firewall; Rekall Should create WAF rules to specifically address cross-site scripting by blocking abnormal requests. 4. Developers should disable java script in Rekall forms 5. Form code should always be kept up to date

Vulnerability 3	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	Medium
Description	Command injection also known as shell injection is a type of web application vulnerability which allows a cybercriminal to execute arbitrary commands by exploiting an application vulnerability such as insufficient input validation. A cybercriminal can use insecure transmissions of user data such as cookies and forms to inject a malicious command into the system shell on a web server in a bid to compromise the server.
Images	https://drive.google.com/file/d/1DjENewDGYiTT8UzjAP5UUmwSv-pBG5bd/view?usp=share_link
Affected Hosts	192.168.14.35
Remediation	1. Yet again I recommend input validation as one of the ways to mitigate against command injection. 2. use softwares that are containerised like SElinux or Apparmor 3. Rekall to avoid calls and user input in order to prevent threat actors from inserting characters into the OS command. 4. When executing system command such as execFile make sure to use only secure APIs 5. Developers should create a sort of white list of pre-approved input the system should accept. Anything outside this list should not be accepted by the system.

Vulnerability 4	Findings
Title	Privilege escalation
Type (Web app / Linux OS / WIndows OS)	Windows
Risk Rating	Critical
Description	This is when a bad actor successfully gains access to Rekall infrastructure. We were able to bypass the proper authorization channel and were able to escalate our privilege to a root. We then were able to access some vital company data.
Images	https://drive.google.com/file/d/1RIc7-emblLADVHREOoKjJl2sn4JipD-B/view?usp=share_link
Affected Hosts	192.168.14.35
Remediation	1. Using software that will follow least privilege such as SElinux and Apparmor 2. Always keep account up to date with comprehensive privilege account management 3. Always patch and update software

Vulnerability 5	Findings
Title	SQL injection
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	High
Description	This is when hackers modify an existing command to suit their needs. Because the system is vulnerable we were able to carry out this attack.
Images	https://drive.google.com/file/d/1kOeKjdU8CN1UkFbALI0mBnFLb3Jvs6yD/view?usp=share_link
Affected Hosts	192.168.14.35
Remediation	1. Input Validation 2. Keeping the database up to date 3. Update and patch system always 4. Use appropriate privileges

Vulnerability 6	Findings
Title	Shell Shock vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	High
Description	This is the type of vulnerability that allows systems containing a vulnerable version of bash to be exploited to run commands with higher privileges. We were able to carry out this attack using metasploit
Images	https://drive.google.com/file/d/1JHDg4EqFo96ZtAHijMdw66NvYWsUfQ87/view?usp=share_link
Affected Hosts	192.168.13.14
Remediation	1. Bash should be updated to latest version, 2. disable scripting 3. Use next generation firewall

Vulnerability 7	Findings
Title	Apache struts vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	High
Description	This vulnerability allows for remote code execution. Depending on the privileges, a bad actor could install programs , view, change or delete data or create a new user account with full privileges.
Images	https://drive.google.com/file/d/1jkhle-yOzbHn6UbQUnXkGa4TILDh1vvo/view?usp=share_link
Affected Hosts	192.168.14.35
Remediation	1. Run all software as a non privileged user to reduce the effect of a successful attack. 2. ALways upgrade to the apache struts.

Add any additional vulnerabilities below.

1. SLMail vulnerability
2. Sensity data credential dump
3. Open source exposed data