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	Icebane Cybersecurity Experts
Contact Name	Michael Kramer
Contact Title	CEO

Document History

Version	Date	Author(s)	Comments
001	7/19/22-7/23/22	Michael Kramer	

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.

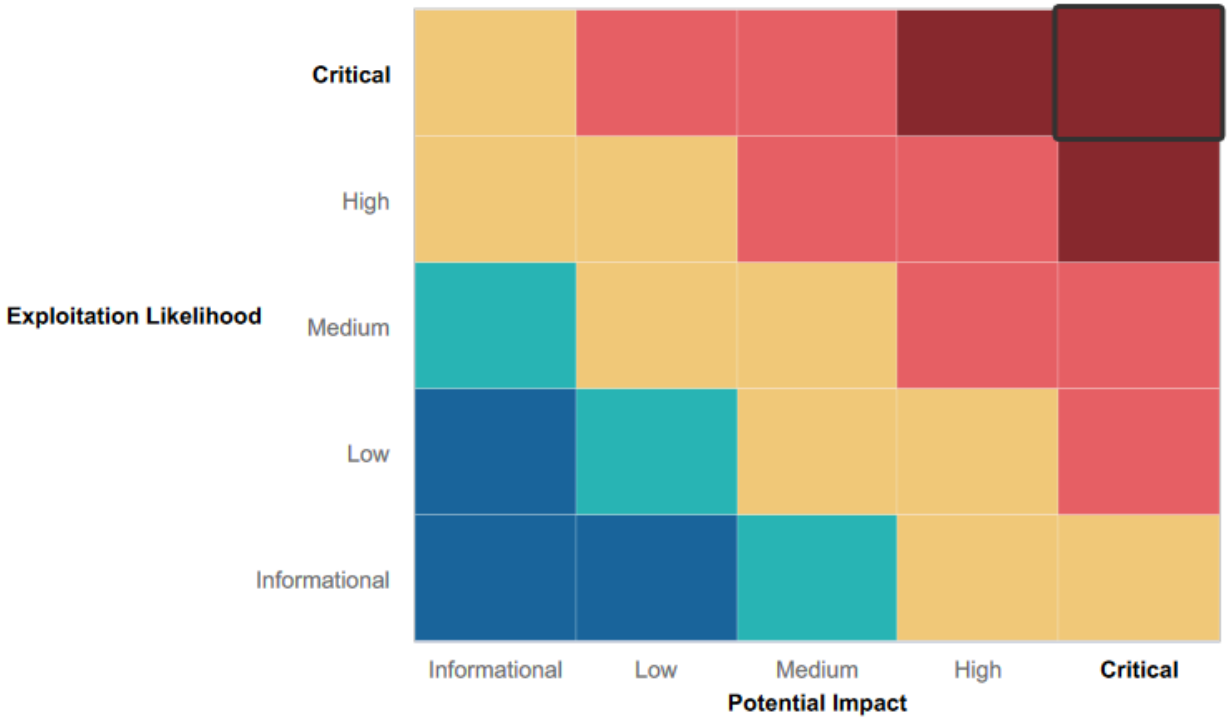
Executive Summary of Findings

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.

- Administrator on the Domain Controller has a strong password

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.

- Poor password policy
- Web application is vulnerable to command injection
- Web application is vulnerable to cross site scripting
- Web application is vulnerable to file inclusion
- Web application is vulnerable to SQL injection
- Sensitive data is left in multiple places easy to access or to find
- Multiple ports left open on Linux servers
- Multiple ports left open on Windows servers
- Software on Linux and Windows server not up to date
- Both Linux and Windows servers are vulnerable to out of the box modules in Metasploit
- Anonymous login allowed on Widows web server

Executive Summary

To start our tasks, we started with attacking the web application first to check for vulnerabilities. Right away we noticed cross site scripting was allowed on the site. While some parts had some protections against it, we were still able to easily bypass it (Day_1_flag_2.PNG). We then checked for Local File Inclusion since fields were available to upload files. Both locations were vulnerable to, while one had input validation, it wasn't properly set up and we were still able to get around it (Day_1_flag_6.PNG). While peeking around we noticed that some sensitive data was easily found in files that we shouldn't have access to. In addition, we also found data left in the open probably for easy access so they could easily remember passwords (Day_1_Flag_8.PNG Day_1_flag_9.PNG). The web application was also susceptible to Command Injection and we were able to access files in the system (Day_1_Flag_10.PNG Day_1_flag_11.PNG). Using this access we were able to access the file with a list of users and brute forced an easily guessed password (day_1_flag_12.PNG). On our second day of checking the network we decided to check for vulnerabilities through Linux. After doing some initial reconnaissance to determine IPs address and Domain names (Flag_1.PNG Flag_3.PNG), we did a few scans determine if any software was out of date or had known exploits that we could take advantage of (Flag_5.PNG Flag_6.PNG). Using Metasploit (a hacking tool used with preloaded modules for known vulnerabilities) we were then able to gain root access to a few local machines (Flag_7-1.PNG Flag_7-2.PNG Flag_8-1.PNG flag_8-2.PNG). We also noticed login credentials during our credential scan and were able to brute force into a machine with a guessed password (FLag_12-1.PNG Flag_12-2.PNG). On the third day we went after the Windows machines. We did a search on the net and found username and password hash on Github which we were able to crack using John (Flag1_1.PNG Flag1_2.PNG). After doing another scan of those machines we found that we could log into one machine anonymously (Flag3_1.PNG Flag3_2.PNG). From the same scan of the machine we were able to use an exploit from their version of SLMail via Metasploit allowing us shell access to the machine (Flag4_1.PNG Flag4_2.PNG). Using this shell access, we were able access files on the system (Flag7.PNG) as well as get a list of other users on this local machine and download credentials which we were able to crack using John (Flag8_1.PNG Flag8_2.PNG). From there we logged into the Domain Controller using Metasploit and a smb/psexec exploit using the same credentials of the we found on the local machine and gathered user info (Flag8_3.PNG Flag8_4.PNG). We then dumped the credentials password hash of the Administrator for the Domain Control but was unable to crack it (Key10.PNG).

Summary Vulnerability Overview

Vulnerability	Severity
Poor password policy	Critical
Reused credentials and passwords	Critical
Sensitive data exposure	Critical
Command Injection	High
Local file inclusion	High
SQL injection	High
Directory Traversal	High
Apache Tomcat Remote Code Execution	High
Struts Vulnerability	High
Drupal Vulnerability	High
Sudo Vulnerability	Critical
Anonymous FTP login	High
SLMail Vulnerability	High
SMB PSexec Vulnerability	High
Cross Site Scripting	High
Session Management	High
Open Ports	High
Apache Shellshock Vulnerability	High

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

Scan Type	Total
Hosts	totalrekall.xyz, 192.168.13.0/24, 172.22.117.0/24
Ports	21, 22, 25, 79, 80, 106, 110, 135, 139, 443, 445, 8009, 8080

Exploitation Risk	Total
Critical	4
High	14
Medium	0
Low	0

Vulnerability Findings

Vulnerability 1	Findings
Title	Poor password policy
Type (Web app / Linux OS / WIndows OS)	Web application / Linux OS / Windows OS
Risk Rating	Critical
Description	Passwords on multiple accounts were very easily guessed or cracked due to poor policy.
Images	day_1_flag_12.PNG Flag_12-2.PNG Flag6_2.PNG
Affected Hosts	192.68.13.0/24, 172.22.117.0/24
Remediation	Stronger password policy and/or Two factor authentication are remediations to fix this issue.

Vulnerability 2	Findings
Title	Reused credentials/passwords
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	We were able to gain access to the Domain Controller on the Windows System using the same username and password from a local machine's account.
Images	Flag8_1.PNG Flag8_2.PNG Flag8_3.PNG Flag8_4.PNG
Affected Hosts	172.22.117.0/24
Remediation	Set a policy that requires a different username and/or password between local machines and Domain Controller accounts.

Vulnerability 3	Findings
Title	Sensitive Data Exposure
Type (Web app / Linux OS / WIndows OS)	Web Application / Linux OS / Windows OS

Risk Rating	Critical
Description	We were able to find login credentials via multiple venues. One was left on an open source platform. Another was found in the certificate registry. A third was left in the http code for a web app page.
Images	Flag1_1.PNG FLag_12-1.PNG Day_1_Flag_8.PNG
Affected Hosts	192.168.13.14, 192.168.13.35, 172.22.117.20
Remediation	Force affected users to change to stronger passwords immediately and require mandatory training on why it's critical not to have such information left in the open.

Vulnerability 4	Findings
Title	Sudo Vulnerability CVE-2019-14287
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	We were able to use the sudo command to run and get access to folders and files on the system's root user.
Images	Flag_12-2.PNG
Affected Hosts	192.168.13.14
Remediation	Update the Linux OS or at the very least update the sudo command inside the Linux OS to the current version.

Vulnerability 5	Findings
Title	Anonymous FTP Login
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	After running an nMap scan of the system we found that Anonymous ftp login was allowed and gave us access to system files.
Images	Flag3_1.PNG Flag3_2.PNG
Affected Hosts	172.22.117.20
Remediation	Disable Anonymous ftp login or severely limit access to Anonymous users.

Vulnerability 6	Findings
Title	SMB PSExec Vulnerability
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	After gaining credentials on a local machine and lucky to realize they were reused on the Domain Controller, we were able to use Metasploit's SMB/PSexec module to log in.
Images	Flag8_3.PNG Flag8_4.PNG
Affected Hosts	172.22.117.10
Remediation	Disable SMB protocols and/or close the port that allows SMB access (445)

Vulnerability 7	Findings
Title	Drupal - CVE-2019-6340
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	We were able to get Remote Control Access using Metasploit since they were running an older version of Drupal.
Images	Flag_11.PNG
Affected Hosts	192.168.13.13
Remediation	Update Drupal with either the latest version or an updated security patch to prevent this exploit from happening further.

Vulnerability 8	Findings
Title	Struts - CVE-2017-5638
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	Using a struts vulnerability we were able to gain a command shell in the

	system using Metasploit.
Images	Flag_10-1.PNG
Affected Hosts	192.168.13.12
Remediation	Web application firewalls can mitigate this attack if the rules are set to approve valid content types or ban OGNL expressions.

Vulnerability 9	Findings
Title	Open Ports
Type (Web app / Linux OS / WIndows OS)	Linux OS / Windows OS
Risk Rating	High
Description	Various ports were left open.
Images	Flag_5.PNG Flag2_1.PNG
Affected Hosts	192.168.13.0/24 172.22.117.0/24
Remediation	Only having necessary ports open or blocking all ports except for permissive access should help mitigate a lot of exploits.

Vulnerability 10	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 an Apache Tomcat exploit, we were able to get a command shell via Metasploit to gain access to the local machine
Images	Flag_7-1.PNG
Affected Hosts	192.168.13.10
Remediation	Updating the software to the latest version of Tomcat will fix the vulnerability.

Vulnerability 11	Findings

Title	Cross Site Scripting
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	High
Description	The web application was vulnerable in a few areas to Cross Site Scripting. Even some moderate measures were by-passed
Images	Day_1_flag_2.PNG
Affected Hosts	192.168.13.35
Remediation	Two ways (using both preferred) to prevent this is to encode data on output and validate input data on arrival.

Vulnerability 12	**Findings**
Title	Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	High
Description	Using a simple script saved as an accepted file, we were able to run commands from the web url. We were also able to bypass some protection in a later field.
Images	Day_1_flag_6.PNG
Affected Hosts	192.168.13.35
Remediation	There are many possible ways to mitigate this. ID assignation, which saves the files in a secure database and gives ID numbers preventing from altering paths. Whitelisting is another way which will only use verified and secured whitelisted files and ignore all others. Also giving your servers better instructions to make sure they send download headers automatically instead of executing files in a directory.

Vulnerability 13	**Findings**
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web Application

Risk Rating	High
Description	After finding a .php page that we shouldn't have access to, we were able to determine it was susceptible to command injection. Using this we were able to access local files on the machine.
Images	Day_1_Flag_10.PNG Day_1_flag_11.PNG
Affected Hosts	192.168.13.35
Remediation	There are multiple factors to prevent this, all of them requiring Validation of some sort. The most robust one is to validate that the input contains only alphanumeric characters, no other syntax or whitespace.