| 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** | Ohio Visual Inc. |
| --- | --- |
| **Contact Name** | Orlando Guerra |
| **Contact Title** | Penetration Test Reporter |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 12/31 | O. Guerra |  |
| 002 | 1/1 | O. Guerra |  |
| 003 | 1/4 | O. Guerra |  |

# 

## 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.

* JavaScript Client-side Input Validation
* Security Testing and Monitoring for Apache servers
* Injection points are not readily available

### 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.

* Web App Servers can be infected by XSS, SQL, and PHP Injection
* Scripts can be uploaded where images can be browsed.
* GitHub Repository files
* Passwords lack encryption
* Remote Shell access through e-mail protocols
* Exposed HTML data
* Remote Code Execution
* NMap scanning of open local ports
* FTP Enumeration

## Executive Summary

Ohio Visual Inc. was contracted to conduct a Penetration test of TotalRekall’s web applications and operating systems consisting of Windows and Linux. During our testing we found various malicious practices that could be used against the company. Files and systems were able to be extracted, uploaded, and changed. Sensitive data that was exposed was used to conduct these tests.

On day one, Web Applications were tested. Cross Site Scripting (XSS) was one of the first vulnerabilities found. Reflected XSS was successful on the Welcome page. A Stored XSS attack was successful on the Comments page. SQL Injection using true statements were used on the password line of the Login page leading to infiltration. Local File Inclusion was recognized as a possible attack and within a file uploading section of the Memory Planner page, a PHP script was executed. By using sensitive data we are able to inject PHP script into the URL therefore changing what is displayed on the screen, in this case it was password files. Review of HTML source code shows usernames and passwords were found within the HTML source of the Login page.

On day two, Linux Servers were examined. Using Open Source Intelligence we uncovered IP’s on public record, this needs to be scrubbed. Both Linux and Windows machines were exposed to NMap scanning. This led us to all open ports within the system to conduct testing. Metasploit Meterpreter shells were able to be opened up on accessible ports across the server. HTTP ports were used to inject remote code into Linux servers.

On day three, Windows servers were also vulnerable. Having open ports for file transferring can be used against the server to extract important information. A look at account management and hardening should take place. E-mail servers were found to be unsecure. As there was no protection set up, enumerating remote hosts was all we needed for file extraction.

We hope you read through this report with confidence that the mitigation techniques suggested are stated to leading TotalRekall and Ohio Visual to create more secure operating systems and company web apps.

## 

## 

## 

## 

## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| 1. XSS Reflected | **Medium** |
| 2. XXS Stored | **Medium** |
| 3. PHP Injection | **Critical** |
| 4. SQL Injection | **Critical** |
| 5. Local File Inclusion | **Critical** |
| 6. Sensitive Data in Website HTML | **High** |
| 7.. Domain Dossier Exposure | **Medium** |
| 8. Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617) | **Critical** |
| 9. OSINT: GitHub Site Repository | **High** |
| 10. OSINT: Github Site Password Crack and Successful Login | **High** |
| 11. NMap Scan and Aggresive NMap Scans | **Critical** |
| 12. FTP Enumeration | **Medium** |
| 13. SLMail Port Exploit Using Metasploit | **Critical** |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | 3.33.130.190 totalrekall.xyz  15.197.148.33  172.22.117.20 Windows 10  172.22.117.10 W Domain Controller  172.22.117.100 W Host  192.168.13.1 Linux  192.168.13.10  192.168.13.12  192.168.13.13  192.168.13.14  192.168.14.35 Web App |
| Ports | 21 FTP  22 SSH  80 HTTP  25 SMTP  110 POP3  8009 AJP13 -Apache Jserv  8080 HTTP -Apache Tomcat /Coyote JSP |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 6 |
| **High** | 3 |
| **Medium** | 4 |
| **Low** | 0 |

## Vulnerability Findings

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | XSS Reflected |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | Using HTML or JavaScript we are able to create a non-persistent threat on Rekall’s website. The malicious script can steal cookie data or create pop-ups on the victim’s webpage. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | 1. Validate user input by creating whitelists of permissible input values. 2. Output data should be encoded so that the active server does not use special characters to post raw information on the website. |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | XSS Stored |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | The attacker comments <script>alert(“Ohio”)</script> which is then interpreted as JavaScript and displays the pop up window. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | 1. When redisplaying user-controlled data we must encode all non-alphanumeric characters as HTML entities making a Java standard tag library for use. |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | PHP Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | Critical |
| **Description** | By using sensitive data we are able to inject PHP script into the URL therefore changing what is displayed on the screen. In this case the system ‘/etc/passwd’ file was reproduced. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | 1. Dynamic code execution needs to be avoided by avoiding the use of core functionality in PHP such as the use of ‘system()’ |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | SQL Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | We conducted an SQL Injection on the Login page. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | 1. Create a regex pattern to make the input pattern harder to figure out such as brace or more forms of quotes. 2. Creating a SQLCommand to store procedures to run against a SQL server. 3. SQLCommand for code blocks. |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Local File Inclusion |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | .php files have been or can be uploaded using script files in PHP language. Ohio Visual used a simple one to test out the upload. It was successful in the toolbar “Upload Your File!” |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | 1. As this means malicious script can be uploaded onto the server. Server side validation must be used to only allow certain file types. 2. The use of databases and indexes for permissible file types. 3. Software API’s for communication. |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data in Website HTML |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | High |
| **Description** | Usernames and passwords were found within the HTML source of the Login page. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | 1. Deleting unnecessary information in code that can be used for harmful access. 2. Changing within the HTML code of how usernames and passwords are remembered, within the input type for the two fields we can turn off auto-complete or the use of libraries. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Domain Dossier Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Medium |
| **Description** | Using Centralops.net’s Domain Dossier we are able to pull a whois record on our website. There is local IP’s related to the website posted here. |
| **Images** |  |
| **Affected Hosts** | 3.33.130.190  15.197.148.33 |
| **Remediation** | 1. We clean up the totalrekall WhoIs record through the domain’s Registrar services. we could make certain information public while taking away other information altogether. |

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** | Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617) |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | A vulnerability was found in Tomcat’s Servlet context that abuses Java servers request and response methods to applications. Using Metasploit in Kali we are able to open a meterpreter shell into a session. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.10 |
| **Remediation** | 1. Block HTTP methods that allow remote code. 2. Update to newer versions of Tomcat that require more authentication. |

| **Vulnerability 9** | **Findings** |
| --- | --- |
| **Title** | OSINT: GitHub Site Repository |
| **Type (Web app / Linux OS / WIndows OS)** | WIndows OS |
| **Risk Rating** | High |
| **Description** | A hash for a possible password and account were found on GitHub. /totalrekall/sites/xampp.users. Putting this hash through the john gives us trivera: Tanya4life. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 W10  172.22.117.10 WDC  172.22.117.100 W Host |
| **Remediation** | 1. Remove these credentials from GitHub repositories 2. We can use the git filter-repo tool or use BFG Repo-Cleaner. This will rewrite the history of the repository. We can also reference salting passwords to make them harder to crack in the future. |

| **Vulnerability 10** | **Findings** |
| --- | --- |
| **Title** | OSINT: Github Site Password Crack and Successful Login |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | After the enumeration of the hosts using a port scan we see an opening in port 80 on host 172.22.117.20. logging in using the provided credentials from the repo we have access into the area that is flagged. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.100  172.22.117.10  172.22.117.20 |
| **Remediation** | 1. Intrusion Detection and Prevention Systems 2. SSH keys for root logins 3. Using HTTPS port 443 for login transmissions, this might mean blocking port 80 altogether. |

| **Vulnerability 11** | **Findings** |
| --- | --- |
| **Title** | NMap Scan and Aggresive NMap Scans |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS /Linux Servers |
| **Risk Rating** | Critical |
| **Description** | A port scan of the subnet that Kali Linux is on, 172.22.117.0/24, will show us that port 80: HTTP, 21:FTP connection is a vulnerability. If the attacker already found the GitHub page totalrekall/sites/xampp.users, they have credentials in place. 5 more Linux servers were found with ports open connected to the webpage. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20  172.22.117.10  192.168.13.1  192.168.13.10  192.168.13.12  192.168.13.13  192.168.13.14 |
| **Remediation** | 1. Obfuscating IP’s with SOScleaner tool 2. IP blocking for unauthorized users outside the network trying log -ins. 3. Blocking probes and slowing scans to disallow information retrieval. |

| **Vulnerability 12** | **Findings** |
| --- | --- |
| **Title** | FTP Enumeration |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Medium |
| **Description** | An open port such as Port 21 lets an attacking party to enumerate files. Subsequently, we moved files by downloading them using a get command. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | 1. No File Transfer Protocol Access for the port. 2. Using Strong Encryption. 3. This would mean suggesting the use of SFTP or FTPS protocols with hardening. 4. Hardening Account Management. 5. Restricting and changing permissions for important files. |

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | SLMail Port Exploit Using Metasploit |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Critical |
| **Description** | Seattle Lab Mail (SLMail) runs on port 110. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | 1. Firewalls can be set up to close port 110 as email delivery with POP3 is not secure. 2. Using port 995, a more secure POP3 port instead. 3. IMAP is a more modern email protocol which can be used instead. |

Add any additional vulnerabilities below.