|  |
| --- |
| 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](#_Toc193300784)

[Contact Information 4](#_Toc193300785)

[Document History 4](#_Toc193300786)

[Introduction 5](#_Toc193300787)

[Assessment Objective 5](#_Toc193300788)

[Penetration Testing Methodology 6](#_Toc193300789)

[Reconnaissance 6](#_Toc193300790)

[Identification of Vulnerabilities and Services 6](#_Toc193300791)

[Vulnerability Exploitation 6](#_Toc193300792)

[Reporting 6](#_Toc193300793)

[Scope 7](#_Toc193300794)

[Executive Summary of Findings 8](#_Toc193300795)

[Grading Methodology 8](#_Toc193300796)

[Summary of Strengths 9](#_Toc193300797)

[Summary of Weaknesses 9](#_Toc193300798)

[Executive Summary 10](#_Toc193300799)

[Summary Vulnerability Overview 11](#_Toc193300800)

[Vulnerability Findings 13](#_Toc193300801)

## Contact Information

|  |  |
| --- | --- |
| **Company Name** | CyberG1 |
| **Contact Name** | William Strahlend |
| **Contact Title** | Penetration Tester |

## Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Comments** |
| 1.0.0 | 03/03/2025 | William Strahlend | Initial Version. Reported findings from Web app. |
| 1.1.0 | 03/04/2025 | William Strahlend | Reported findings from Linux servers |
| 1.2.0 | 03/06/2025 | William Strahlend | Reported findings from Windows servers |
| 1.3.0 | 03/16/2025 | William Strahlend | Rated vulnerabilities |
| 2.0.0 | 03/20/2025 | William Strahlend | Added report summaries and overviews for final report. |

## Introduction

In accordance with Rekall policies, CyberG1 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, CyberG1 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.

CyberG1 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

### 

CyberG1 begins assessments by checking for any passive (open source) data that may assist the assessors with their tasks. If internal, CyberG1 will perform active recon using tools such as Nmap and Bloodhound.

### Identification of Vulnerabilities and Services

CyberG1 uses 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 CyberG1 performs 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, CyberG1 writes the report, which is the final deliverable to the customer.

## Scope

Prior to any assessment activities, Rekall and CyberG1 will identify targeted systems with a defined range or list of network IP addresses. CyberG1 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 CyberG1 was successful in finding several vulnerabilities, CyberG1 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.

* While on the “What do you want to be?” page, the field did restrict the <script> tags to mitigate an attack using JavaScript code.
* In the “Choose your location” area of that page, the file upload field did restrict files to only be able to upload JPG file types.
* DNS Records did not reveal any highly sensitive data and appeared to all be hostmaster data.
* Host 172.22.117.20 had password protected access to a file location that contained sensitive data.

### Summary of Weaknesses

CyberG1 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.

* Many of the vulnerabilities were due to outdated system software. By simply updating them, the vulnerabilities are mitigated
* Other vulnerabilities were due to not configuring the web app with proper restrictions, like prohibiting the use of special characters in the data fields or not restricting file uploads to a specific file type and scanning that file for malware when it is uploaded.
* Other vulnerabilities was storing sensitive data “out in the wild”. These were things like usernames and password hashes.
* Simple passwords was another thing that could have been easily mitigated by enforcing a strict and complex password policy. The password hashes found were cracked in seconds due to the passwords simplicity.
* Properly configuring the website would have stopped the ability to perform Directory Traversal where usernames and passwords were found, along with other sensitive data.
* Open ports made for simple exploitation. Simply turning them off would have prevented exploitation.

## Executive Summary

### Accessing the totalrekall.xyz Web App

When first beginning to use the app at totalrekall.xyz, we used JavaScript, to enter code into the Name field on the Welcome page. <script>alert(“You have been hacked!”)</script>. Navigating to the comments page, we were also able to use JavaScript store code onto the site into the comment field <script>alert(“You have been hacked!”)</script>. After that we shifted our focus to using a PHP file with a basic command and we successfully uploaded the file to the site. This exposed a vulnerability in the web app where someone can upload and execute a malicious code file to the website. Then, using a Directory Traversal Attack, we discovered a robots.txt file. We accessed the file and found it exposed sensitive data.

When navigating to the Rekall Admin Networking Tools Page, we found instructions left on the page, informing us a vendors.txt file was available. By appending vendors.txt to the IP address in the address bar of the browser we were able to access your vendor data. We then used data from this file and entered it into the DNS check, using the word splunk and selected the Lookup button, and targeted data was revealed there as well.

Navigating to the “What do you want to be?” page, we used a modified version of the previous Reflected XSS injection, where we were able to deliver a payload using basic JavaScript but disguising it with a using the tag twice within the command. The following is what we used:<SCRIPscriptT>alert(“You have been hacked!”)</SCRIPscriptT>

Further down that page in the location area, similar to a previous exploit, we used the same PHP file, hello.php, but due to a file type restriction, we had to modify the PHP file and resave it as hello.jpg so we could continue to successfully exploit the site.

Navigating back to the Admin Networking Tools Page, we used content from the vendors.txt file again. We used the word splunk in the MX Record Checker field and selected the Lookup button, and more targeted data was revealed.

From there we accessed the Login page and by using credentials that we enumerated from a previous exploit we entered in the username with some SQL code and performed a SQL Injection attack. We used <username> ‘OR 1=1’ and <password> ‘OR 1=1’ and we successfully exploited the site and obtained additional data.

Finally, using credentials from a Directory Traversal exploit again, we started executing Brute Force Attack on the Login page. Due to a simple password, we were able to guess the password for the username and successfully logged into the site.

### Accessing the Linux OS Network

To access the Linux OS Network, we started by using the OSINT Framework. We navigated to who.is and entered in the domain name totalrekall.xyz. The results revealed sensitive data within the who.is page about the website, providing an ssh username and other sensitive data. Then, using Domain Dossier from the OSINT Framework tool, we attempted to see what the DNS records would reveal, but it appears PII is not available there. After not finding any data there, we looked up certificate data at crt.sh and found an expired certificate that had not been revoked yet. Along with that more sensitive data was revealed.

From there we decided to run an Nmap scan which showed how many hosts are up and what services are available on those hosts. This exposed the following issues:

* 192.168.13.10
  + Port 8009 open with service ajp13
* 192.168.13.14
  + Port 22 open with service ssh
* 192.168.13.1
  + Port 5901 open with service vnc-1
  + Port 6001 open with service X11:1

We then ran an aggressive Nmap scan and found 192.168.13.13 was running Drupal on it. While Drupal itself it not an issue, it was not up-to-date, therefore making the network vulnerable.

Then, using Nessus to scan the network, we found IP address, 192.168.13.12 to have an Apache Remote Code Execution (RCE) vulnerability.

After that, using Metasploit module exploit(mutli/http/tomcat\_jsp\_upload\_bypass), we accessed IP address 192.168.13.10 and ran the bash command which took me to the root directory. From there, we enumerated for our target information and was able to obtain the data. Continuing in Metasploit we used exploit module exploit(multi/http/apache\_mod\_cgi\_bash\_env\_exec) with the receiving host of 192.168.13.11 and the target uri of /cgi-bin/shockme.cgi was able to obtain initial access and accessed targeted data by accessing the sudoers file. Continuing on that host we used the Shellshock Exploit, and then used the command cat /etc/passwd to obtain access to the list of user accounts on the system.

Once again, using Nessus, it revealed that 192.168.13.12 is vulnerable to the Struts exploitation. We then used the Metasploit module exploit(multi/http/struts2\_content\_type\_ognl) and we established access to a meterpreter shell at root access on 192.168.13.12 host. From there we enumerated and found a zip file that contained sensitive data.

### Accessing the Windows OS Network

We started out using Google search engine to search for some information in public areas. We found a Github repository for Rekall Corporation and there is a file named xampp.users that contains sensitive data which included a username and password hash that we cracked using John the Ripper to extract the hashed password for the username.

Then we ran Nmap on 172.22.117.0/24 network and found that 172.22.117.20 had the OpenSSL service running on port 80. Using that information we accessed the IP Address 172.22.117.20 directly through a browser and was prompted to enter in user credentials with it stating “Restricted Content”. Using the credentials found from the previous enumeration we entered in those credentials and successfully logged in and had access to sensitive content.

After that we used Nmap to run a report and it revealed port 21 was open on IP 172.22.117.20. Using the command ftp -p 172.22.117.20 and entering the username, Anonymous, then just pressing the ENTER button for the password, access was granted. Then simply entering in the command  
get flag3.txt the file was retrieved and we had access to the sensitive data.

We continued with data from the Nmap report, host 172.22.117.20 revealed port 25 was open using the SLmail service. Using the Metasploit exploit module, exploit(windows/pop3/seattlelab\_pass) we obtained initial access with the use of a meterpreter shell. From there, we enumerated for targeted content and found a file with sensitive content. Continuing with the same exploit with SLmail on 172.22.117.20, we ran the shell command and gained access as NT authority\SYSTEM. From there we enumerated through the scheduled tasks for targeted data and was able obtain sensitive data. While continuing to enumerate within the exploit of SLmail, we were able to find sensitive data in the C:\Users\Public\Documents folder. From there, we loaded up Kiwi in Metasploit and ran the command lsa\_dump\_sam. This provided an NTLM hash. We then ran the command   
kiwi\_cmd lsadump::cache and that provided us a username. Then we took the username and hash, and used John the Ripper to crack the hash and obtained the password due to its simplicity.

Moving on to IP address 172.22.117.10, we continued using Metasploit and used the exploit module exploit(windows/smb/psexec). Using the credentials discovered from a previous exploit, we successfully obtained initial access and enumerated through the system and obtained sensitive data in the C: Drive.

In conclusion, while some areas did have some mitigations in place, most of them were not complete in their configurations and left those areas still vulnerable to attacks. Unfortunately, a majority of what we found revealed your systems to be in a very critical state of vulnerability. Our recommendation is to review our report and develop some remediation plans with actionable items and milestones. We will setup a meeting to discuss these results in two weeks to give you time to review and make some plans. There we will discuss next steps and discuss whether or not you believe a follow-up test is necessary.

Thank you for the opportunity to assist you in secureing your systems.

## 

## 

## 

## 

## Summary Vulnerability Overview

|  |  |
| --- | --- |
| **Vulnerability** | **Severity** |
| Reflected Cross-Site Scripting (XSS) | **Critical** |
| Stored/Persistent Cross-Site Scripting (XSS) | **Critical** |
| Local File Inclusion (LFI): Uploading Malicious Code Files to Web App | **Critical** |
| Admin Login Credential Data Exposure | **Critical** |
| Sensitive Data Exposure through Directory Traversal Attack | **Critical** |
| Command Injection | **Critical** |
| Advanced Reflected XSS | **Critical** |
| Advanced Local File Inclusion (ALFI) | **Critical** |
| Advanced Command Injection | **Critical** |
| SQL Injection | **Critical** |
| Brute Force Attack | **High** |
| Sensitive Data Exposure - WHOIS | **High** |
| DNS Record Exposure | **Informational** |
| Certificate Information Exposure | **High** |
| Network Scan Results | **Low** |
| Drupal Exploit | **Low** |
| Remote Code Execution (RCE) Vulnerability | **Medium** |
| Remote Code Execution (RCE) Vulnerability: Access to root | **Critical** |
| Shell Shock Exploit for User Access Data | **Critical** |
| Shell Shock Exploit for Sensitive Data | **Critical** |
| Struts Exploitation | **Critical** |
| Public Data Exposure | **Critical** |
| Open Port Exploit: OpenSSL | **Critical** |
| FTP Exploit | **Critical** |
| SLmail Exploit | **Critical** |
| SYSTEM Shell Access | **Critical** |
| Sensitive Data Exposure | **High** |
| SAM Credential Exposure | **High** |
| Improper Access Control | **Critical** |

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

|  |  |
| --- | --- |
| **Scan Type** | **Total** |
| Hosts | totalrekall.xyz (76.223.105.230) 192.168.13.1, 192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14, 192.168.13.0/24, 192.168.14.35,  172.22.117.10, 172.22.117.20, github.com/totalrekall/site/ |
| Ports | 22, 5901, 6001, 8009 |

## Exploitation Risk

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **20** | **5** | **1** | **2** | **1** |  |
|  | **Critical** | **High** | **Medium** | **Low** | **Informational** |  |

## Vulnerability Findings

|  |  |
| --- | --- |
| **Vulnerability 1** | **Findings** |
| **Title** | Reflected Cross-Site Scripting (XSS) |
| **Platform** | Web app |
| **Risk Rating** | Critical |
| **Description** | Using JavaScript, we were able to enter code into the name field on the Welcome page. <script>alert(“You have been hacked!”)</script> | |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Modify input validation rules to exclude special characters and keyword command prompts and tags to prevent malicious code to be execute on the site. |

|  |  |
| --- | --- |
| **Vulnerability 2** | **Findings** |
| **Title** | Stored/Persistent Cross-Site Scripting (XSS) |
| **Platform** | Web app |
| **Risk Rating** | Critical |
| **Description** | Using JavaScript, we were able to store code onto the site into the comment field on the Comments page.  <script>alert(“You have been hacked!”)</script> | |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Modify input validation rules to exclude special characters and keyword command prompts and tags to prevent malicious code to be stored and executed on the site. |

|  |  |
| --- | --- |
| **Vulnerability 3** | **Findings** |
| **Title** | Local File Inclusion (LFI): Uploading Malicious Code Files to Web App |
| **Platform** | Web app |
| **Risk Rating** | Critical |
| **Description** | Using a PHP file with a basic command, we successfully uploaded the file to the site. This exposed a vulnerability in the web app where someone can upload and execute a malicious code file to the website. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Implement File Upload Validation to restrict file upload types to only image files to prevent malicious files from being uploading to the site and executed. |

|  |  |
| --- | --- |
| **Vulnerability 4** | **Findings** |
| **Title** | Admin Login Credential Data Exposure |
| **Platform** | Web app |
| **Risk Rating** | Critical |
| **Description** | While on the Login page, pressing CTRL+A to select everything on that page revealed hidden administrative level login credentials right on the page. |
| **Images** | A screenshot of a computer  AI-generated content may be incorrect.  A red background with black lines and green text  AI-generated content may be incorrect. |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Remove all sensitive content from the source code of the webpage. |

|  |  |
| --- | --- |
| **Vulnerability 5** | **Findings** |
| **Title** | Sensitive Data Exposure through Directory Traversal Attack |
| **Platform** | Web app |
| **Risk Rating** | Critical |
| **Description** | While using the Directory Traversal Attack, robots.txt was discovered and accessed, exposing sensitive data. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Restrict access to sensitive files like robots.txt to authorized users only. |

|  |  |
| --- | --- |
| **Vulnerability 6** | **Findings** |
| **Title** | Command Injection |
| **Platform** | Web app |
| **Risk Rating** | Critical |
| **Description** | From the instructions left on the Rekall Admin Networking Tools page, accessed the vendors.txt file by appending it to the IP address in the address bar of the browser to reveal the vendor data. We used data from this file and entered it into the DNS check, using the word splunk and selected the Lookup button, and targeted data was revealed. |
| **Images** | A screenshot of a computer  AI-generated content may be incorrect. |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Remove text instructions on where to find sensitive data and also restrict access to sensitive files like vendors.txt to authorized users only. |

|  |  |
| --- | --- |
| **Vulnerability 7** | **Findings** |
| **Title** | Advanced Reflected XSS |
| **Platform** | Web app |
| **Risk Rating** | Critical |
| **Description** | Using a modified version of the previous Reflected XSS injection, we were able to delivery a payload using basic JavaScript but disguising it with a using the tag twice within the command. The following is what we used:  <SCRIPscriptT>alert(“You have been hacked!”)</SCRIPscriptT> | |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | While the script tag was prevented, it did not completely prevent an attack. Modify input validation rules to exclude special characters as well as keyword command prompts and tags to prevent malicious code from being executed on the site. |

|  |  |
| --- | --- |
| **Vulnerability 8** | **Findings** |
| **Title** | Advanced Local File Inclusion (ALFI) |
| **Platform** | Web app |
| **Risk Rating** | Critical |
| **Description** | Similar to a previous exploit, we used the same PHP file hello.php ,  but due to a file type restriction, we had to modify the PHP file and resave it as hello.jpg so we could continue to successfully exploit the site. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Modify file upload rules to verify file types to prevent a hacker from just changing their file type to suit the requirements. Also, every file uploaded should be scanned for any malware embedded within it. You can use an API plugin like Filestack Upload:  https://www.filestack.com/products/file-upload/ |

|  |  |
| --- | --- |
| **Vulnerability 9** | **Findings** |
| **Title** | Advanced Command Injection |
| **Platform** | Web app |
| **Risk Rating** | Critical |
| **Description** | Again, using content from the vendors.txt file by appending it to the IP address in the address bar of the browser to reveal the vendor data. We used the word splunk in the MX Record Checker field and selected the Lookup button, and targeted data was revealed. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Restrict access to sensitive files like vendors.txt to authorized users only. |

|  |  |
| --- | --- |
| **Vulnerability 10** | **Findings** |
| **Title** | SQL Injection |
| **Platform** | Web app |
| **Risk Rating** | Critical |
| **Description** | By using credentials that were enumerated from a previous exploit and entering into the field <username> ‘OR 1=1’ and <password> ‘OR 1=1’ we were able to successfully exploit the site and obtain additional data. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Modify user input validation to exclude the use of special characters in the username and password fields, specifically ones that can be used in the case of SQL Injection like the following: ‘ “ = < > ! and more. |

|  |  |
| --- | --- |
| **Vulnerability 11** | **Findings** |
| **Title** | Brute Force Attack |
| **Platform** | Web app |
| **Risk Rating** | High |
| **Description** | Using credentials from a Directory Traversal exploit, started executing Brute Force Attack on the Login page. Due to a simple password, was able to successfully login. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Enforce a complex password policy to ensure users are not creating simple passwords that make networks and applications vulnerable to attacks. |

|  |  |
| --- | --- |
| **Vulnerability 12** | **Findings** |
| **Title** | Sensitive Data Exposure - WHOIS |
| **Platform** | Linux OS Server |
| **Risk Rating** | High |
| **Description** | By using the OSINT Framework, we navigated to who.is and entered in the domain name totalrekall.xyz. The results revealed sensitive data within the who.is page about the website, providing an ssh username and other sensitive data. |
| **Images** |  |
| **Affected Hosts** | totalrekall.xyz |
| **Remediation** | Immediately remove sensitive data from the domain registrar, specifically the SSH credential data, and configure the domain to Private settings to ensure organization data is not revealed to the public, like the Organization Address, Email Address, Phone Number, etc. |

|  |  |
| --- | --- |
| **Vulnerability 13** | **Findings** |
| **Title** | DNS Record Exposure |
| **Platform** | Linux OS Server |
| **Risk Rating** | Informational |
| **Description** | Using Domain Dossier from the OSINT Framework tool, it appears PII is not available. NOTE: The IP Address requested for Flag 2 does not match the current records for totalrekall.xyz.  The answer for Flag 2 is 34.102.136.180 while the current records  show 76.223.105.230 as the IP Address. |
| **Images** |  |
| **Affected Hosts** | totalrekall.xyz (76.223.105.230) |
| **Remediation** | While no apparent PII is available currently, regularly perform audits to ensure PII is not available via DNS records. |

|  |  |
| --- | --- |
| **Vulnerability 14** | **Findings** |
| **Title** | Certificate Information Exposure |
| **Platform** | Linux OS Server |
| **Risk Rating** | High |
| **Description** | Using certificate search tools revealed an expired certificate that was not revoked. |
| **Images** | A screenshot of a computer  AI-generated content may be incorrect. |
| **Affected Hosts** | totalrekall.xyz |
| **Remediation** | Immediately revoke all expired certificates and regularly perform audits to ensure certificates are current and revoking any that are about to expire or already expired. |

|  |  |
| --- | --- |
| **Vulnerability 15** | **Findings** |
| **Title** | Network Scan Results |
| **Platform** | Linux OS Server |
| **Risk Rating** | Low |
| **Description** | Ran an Nmap scan showing how many hosts are up and what services are available on those hosts. This exposed the following issues:   * 192.168.13.10   + Port 8009 open with service ajp13 * 192.168.13.14   + Port 22 open with service ssh * 192.168.13.1   + Port 5901 open with service vnc-1   + Port 6001 open with service X11:1   NOTE: while my scan results showed 6 hosts up, the answer for the flag was 5 to satisfy the CTF. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.10, 192.168.13.14, 192.168.13.1 |
| **Remediation** | Immediately close Ports 8009, 22, 5901, 6001. Perform regular audits monitoring ports that may end up open. Close any that are security risks. If port 22 is required to be open, employ complex password policies and lockout after a predetermined set of times for a predetermined amount of time to reset. |

|  |  |
| --- | --- |
| **Vulnerability 16** | **Findings** |
| **Title** | Drupal Exploit |
| **Platform** | Linux OS Server |
| **Risk Rating** | Low |
| **Description** | Ran an aggressive Nmap scan to find out which host had Drupal on it. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.13 |
| **Remediation** | Ensure RESTful Web Services PATCH and POST requests are disabled and make sure no other web service module is enabled, (e.g., JSON:API or RESTful Web Services in Drupal 7.0). Make sure all other updates to services are employed. (source: https://www.tenable.com/cve/CVE-2019-6340) |

|  |  |
| --- | --- |
| **Vulnerability 17** | **Findings** |
| **Title** | Remote Code Execution (RCE) Vulnerability |
| **Platform** | Linux OS Server |
| **Risk Rating** | Medium |
| **Description** | Using Nessus to scan the network, found IP address, 192.168.13.12 to have an Apache Remote Code Execution (RCE) vulnerability. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Upgrade Apache to version 2.3.32/2.5.10.1 or later. |

|  |  |
| --- | --- |
| **Vulnerability 18** | **Findings** |
| **Title** | Remote Code Execution (RCE) Vulnerability: Access to root |
| **Platform** | Linux OS Server |
| **Risk Rating** | Critical |
| **Description** | Using Metasploit module exploit(mutli/http/tomcat\_jsp\_upload\_bypass) , accessed IP address 192.168.13.10 and ran the bash command which took me to the root directory. From there, I enumerated for my target information and was able to obtain the data. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.10 |
| **Remediation** | Upgrade to the most current version of Apache Tomcat. |

|  |  |
| --- | --- |
| **Vulnerability 19** | **Findings** |
| **Title** | Shell Shock Exploit for User Access Data |
| **Platform** | Linux OS Server |
| **Risk Rating** | Critical |
| **Description** | Using Metasploit exploit module exploit(multi/http/apache\_mod\_cgi\_bash\_env\_exec) with the receiving host of 192.168.13.11 and the target uri of /cgi-bin/shockme.cgi was able to obtain initial access and was able to access the targeted data by accessing the sudoers file. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.11 |
| **Remediation** | Update bash packages that address CVE-2014-6271 and CVE-2014-7169 and restrict access to files with Principle of Least Privilege in mind. |

|  |  |
| --- | --- |
| **Vulnerability 20** | **Findings** |
| **Title** | Shell Shock Exploit for Sensitive Data |
| **Platform** | Linux OS Server |
| **Risk Rating** | Critical |
| **Description** | Continuing with the Shellshock Exploit, used the command cat /etc/passwd to obtain access to the list of user accounts on the system. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.11 |
| **Remediation** | Update bash packages that address CVE-2014-6271 and CVE-2014-7169 and restrict access to files with Principle of Least Privilege in mind. |

|  |  |
| --- | --- |
| **Vulnerability 21** | **Findings** |
| **Title** | Struts Exploitation |
| **Platform** | Linux OS Server |
| **Risk Rating** | Critical |
| **Description** | Nessus revealed that 192.168.13.12 is vulnerable to the Struts exploitation. Using Metasploit module exploit(multi/http/struts2\_content\_type\_ognl) we established a meterpreter shell at root access on 192.168.13.12 host. From there we enumerated and found a zip file that contained sensitive data. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Update Apache Struts to include the patches related to this exploit. Schedule patch updates periodically. |

|  |  |
| --- | --- |
| **Vulnerability 22** | **Findings** |
| **Title** | Public Data Exposure |
| **Platform** | Windows OS Server |
| **Risk Rating** | Critical |
| **Description** | Used Google search engine to search for some information in public areas. Found a Github repository for Rekall Corporation and there is a file named xampp.users that contains sensitive data. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 and github.com/totalrekall/site/ |
| **Remediation** | Delete the xampp.users file or remove the Github repository altogether if it no longer serves a purpose. |

|  |  |
| --- | --- |
| **Vulnerability 23** | **Findings** |
| **Title** | Open Port Exploit: OpenSSL |
| **Platform** | Windows OS Server |
| **Risk Rating** | Critical |
| **Description** | Ran Nmap on 172.22.117.0/24 network and found that 172.22.117.20 had the OpenSSL service running on port 80 .  Accessed the IP Address 172.22.117.20 directly through a browser and was prompted to enter in user credentials with it stating “Restricted Content”. Using the credentials found from the previous enumeration, we used John the Ripper to crack the hashed password for the username. Entered in those credentials  and successfully logged in and had access to sensitive content. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | For all public facing content, remove all sensitive data and shut down services that create vulnerabilities for access to content that can jeopardize the company. For the Windows10 machine, if it has no public-facing purpose, close all ports that would allow access into the machine and bring it completely behind the protection of the firewall and repurpose that machine for internal access only. |

|  |  |
| --- | --- |
| **Vulnerability 24** | **Findings** |
| **Title** | FTP Exploit |
| **Platform** | Windows OS Server |
| **Risk Rating** | Critical |
| **Description** | From the Nmap report, it revealed port 21 was open on IP 172.22.117.20 . Using the command ftp -p 172.22.117.20 and entering the username, Anonymous , then just pressing the ENTER button for the password, access was granted.  Then simply entering in get flag3.txt the file was retrieved and we had access to the sensitive data. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Immediately disable Anonymous FTP Login and close port 21. Switch use to Port 22 for SFTP and configure SFTP to limit access to only predefined IP addresses that have been verified as trusted systems. |

|  |  |
| --- | --- |
| **Vulnerability 25** | **Findings** |
| **Title** | SLmail Exploit |
| **Platform** | Windows OS Server |
| **Risk Rating** | Critical |
| **Description** | Using the Nmap report on the network, host 20 revealed port 25 was open using the SLmail service. Using the Metasploit exploit module, exploit(windows/pop3/seattlelab\_pass) we obtained initial access with the use of a meterpreter shell. From there, we enumerated for targeted content and found the file with sensitive content. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Immediately, update SLmail services to the most up-to-date patch and schedule regular stable patch update checks to ensure everything remains up-to-date. Also, if the SLmail service is not necessary on that system, disable it. |

|  |  |
| --- | --- |
| **Vulnerability 26** | **Findings** |
| **Title** | SYSTEM Shell Access |
| **Platform** | Windows OS Server |
| **Risk Rating** | High |
| **Description** | Using the same exploit with SLmail on 172.22.117.20 IP, we ran the shell command and gained access as NT authority\SYSTEM . From there we enumerated through the scheduled tasks for targeted data and was able obtain sensitive data. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Immediately update to SLmail 05.1.0.4433 or newer. If access is not necessary through this system, close the port and service. |

|  |  |
| --- | --- |
| **Vulnerability 27** | **Findings** |
| **Title** | Sensitive Data Exposure |
| **Platform** | Windows OS Server |
| **Risk Rating** | High |
| **Description** | While continuing to enumerate within the exploit of SLmail, we were able to find sensitive data in the C:\Users\Public\Documents folder. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Remove all sensitive data from all public profiles on systems and periodically perform audits to ensure sensitive data is not stored in public profile directories. |

|  |  |
| --- | --- |
| **Vulnerability 28** | **Findings** |
| **Title** | SAM Credential Exposure |
| **Platform** | Windows OS Server |
| **Risk Rating** |  |
| **Description** | While still in the SLmail exploit, loaded up Kiwi in Metasploit and ran the command lsa\_dump\_sam . This provided an NTLM hash. We then ran the command kiwi\_cmd lsadump::cache and that provided us a username. We then took the username and hash, and used John the Ripper to crack the hash and obtained the password due to its simplicity. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Immediately disable SYSTEM-Level access by implementing the policy of “Principle of Least Privilege” and also enforce a password policy for strong, complex passwords to protect against malicious actors. |

|  |  |
| --- | --- |
| **Vulnerability 29** | **Findings** |
| **Title** | Improper Access Control |
| **Platform** | Windows OS Server |
| **Risk Rating** | Critical |
| **Description** | Using Metasploit, used the exploit module exploit(windows/smb/psexec) to exploit access to 172.22.117.10 . Using the credentials discovered from a previous exploit, we successfully obtained initial access and enumerated through the system and obtained sensitive data in the C: Drive. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.10 |
| **Remediation** | Remove all sensitive data from main directories and employ proper access controls for all files and directories using the policy of “Principle of Least Privilege”. Also, enforce a strong, complex password policy to mitigate against malicious actors. |