| 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

[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** | White Fedora Penetration Testing |
| --- | --- |
| **Contact Name** | Tom Marcinonis |
| **Contact Title** | Owner/Operator |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 2/6/2024 | Tom Marcinonis | Pentesting Report |

# 

## Introduction

In accordance with Rekall policies, our organization conducts external and internal penetration tests of its networks and systems throughout the year. The purpose of this engagement was to assess the networks’ and systems’ security and identify potential security flaws by utilizing industry-accepted testing methodology and best practices.

For the testing, we focused on the following:

* Attempting to determine what system-level vulnerabilities could be discovered and exploited with no prior knowledge of the environment or notification to administrators.
* Attempting to exploit vulnerabilities found and access confidential information that may be stored on systems.
* Documenting and reporting on all findings.

All tests took into consideration the actual business processes implemented by the systems and their potential threats; therefore, the results of this assessment reflect a realistic picture of the actual exposure levels to online hackers. This document contains the results of that assessment.

### Assessment Objective

The primary goal of this assessment was to provide an analysis of security flaws present in Rekall’s web applications, networks, and systems. This assessment was conducted to identify exploitable vulnerabilities and provide actionable recommendations on how to remediate the vulnerabilities to provide a greater level of security for the environment.

We used our proven vulnerability testing methodology to assess all relevant web applications, networks, and systems in scope.

Rekall has outlined the following objectives:

Table 1: Defined Objectives

| **Objective** |
| --- |
| Find and exfiltrate any sensitive information within the domain. |
| Escalate privileges. |
| Compromise several machines. |

# 

## Penetration Testing Methodology

### Reconnaissance

### 

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

### Identification of Vulnerabilities and Services

We use custom, private, and public tools such as Metasploit, hashcat, and Nmap to gain perspective of the network security from a hacker’s point of view. These methods provide Rekall with an understanding of the risks that threaten its information, and also the strengths and weaknesses of the current controls protecting those systems. The results were achieved by mapping the network architecture, identifying hosts and services, enumerating network and system-level vulnerabilities, attempting to discover unexpected hosts within the environment, and eliminating false positives that might have arisen from scanning.

### Vulnerability Exploitation

Our normal process is to both manually test each identified vulnerability and use automated tools to exploit these issues. Exploitation of a vulnerability is defined as any action we perform that gives us unauthorized access to the system or the sensitive data.

### Reporting

Once exploitation is completed and the assessors have completed their objectives, or have done everything possible within the allotted time, the assessment team writes the report, which is the final deliverable to the customer.

# 

## Scope

Prior to any assessment activities, Rekall and the assessment team will identify targeted systems with a defined range or list of network IP addresses. The assessment team will work directly with the Rekall POC to determine which network ranges are in-scope for the scheduled assessment.

It is Rekall’s responsibility to ensure that IP addresses identified as in-scope are actually controlled by Rekall and are hosted in Rekall-owned facilities (i.e., are not hosted by an external organization). In-scope and excluded IP addresses and ranges are listed below.

# 

# 

## Executive Summary of Findings

## 

### Grading Methodology

Each finding was classified according to its severity, reflecting the risk each such vulnerability may pose to the business processes implemented by the application, based on the following criteria:

**Critical**: Immediate threat to key business processes.

**High**: Indirect threat to key business processes/threat to secondary business processes.

**Medium**: Indirect or partial threat to business processes.

**Low**: No direct threat exists; vulnerability may be leveraged with other vulnerabilities.

Informational: No threat; however, it is data that may be used in a future attack.

As the following grid shows, each threat is assessed in terms of both its potential impact on the business and the likelihood of exploitation:

Chart

Description automatically generated with medium confidence

### 

### Summary of Strengths

While the assessment team was successful in finding several vulnerabilities, the team also recognized several strengths within Rekall’s environment. These positives highlight the effective countermeasures and defenses that successfully prevented, detected, or denied an attack technique or tactic from occurring.

* Rekall is a going concern making money and able to mitigate the below vulnerabiilities

### Summary of Weaknesses

WFPT 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. Furthermore, many of the vulnerabilities start with the public facing web application.

The web application allows users several vectors to gain initial access to the network and then pivot to other machines and escalate privileges. For example, several fields on the web application allow users to compromise and twist the function of the fields to execute arbitrary commands or allow access to specific files or folders. This is causing the web application to perform in unintended ways and allows potential actors a foothold.

Once the foothold is gained, adversaries have several ways they can exploit the network to gain further access and escalate privileges. For example, the Windows 10 machine allows anonymous ftp login which results in an adversary being able to download specific files for analysis and exploitation offline. Further the linux machine has outdated software allowing an adversary to gain access using known exploits.

There are many vulnerabilities identified in the report but working logically through them, Rekall should focus on securing the web application first. Stopping an initial attack vector is critical. Once the web application is secure work on updating and securing the linux and windows machines.

## Executive Summary

For several days at the end of January WFPT conducted a penetration test of your web application and your servers. The findings in detail are contained within this report. Several major weaknesses were noted during the test and these are summarized here.

First, the web application is extremely vulnerable; it allows users to perform tasks and cause the web application to perform in unintended ways. For example, users can upload files of any type, they can enter script commands directly into text fields designed for other uses, they can issue commands from text fields to reveal the contents of directories that the web application has access to, they can execute sql commands (queries) and they can manipulate the address line of the browser to access web pages not intended for users to access. There are multiple vulnerable pages on the site including the welcome page, the VR Planner page and the about page. These are all very serious weaknesses and skilled attackers can easily exploit your web application and interrupt your services or steal sensitive company information.

In addition, your installation of Apache on your Linux server needs to be updated. There are several patchable vulnerabilities that need to be addressed. Utilizing the latest version of software will ensure that any known vulnerabilities can be stopped. You should set up a routine schedule where all software on your system is updated and patched by the manufacturer on a regular basis. If the manufacturer does not support this, consider alternative solutions to having outdated and potentially vulnerable software on your network.

Finally, the Windows OS installed on two machines needs to be updated. There are several serious weaknesses on the Windows machines. First, the machine ending in .20 allows anonymous ftp logins. This is a problem once a bad actor gains access to the network. Allowing bad actors to easily navigate to other computers on the system increases the attack vectors that can be employed.

## 

## 

## 

## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| **Day 1 Flag 1 Crossite Scripting Reflected**  Use more robust input validation. The validation set up on the comments page was easily bypassed | **High** |
|  |  |
| **Day 1 Flag 2 Crossite Scripting Reflected (Advanced)**  Use more robust input validation. The validation set up on the comments page was easily bypassed | **High** |
|  |  |
| **Day 1 Flag 3 Crossite Scripting (Stored)**  Use more robust input validation. The validation set up on the comments page was easily bypassed | **High** |
|  |  |
| **Day 1 Flag 4 Sensitive Data Exposure**  Remove the sensitive information from the headers of your website. If the information must be in the header, encrypt the information before putting it in the header. | **High** |
|  |  |
| **Day 1 Flag 5 Local File Inclusion**  Restrict file types that can be uploaded to those appropriate based on the feature. The application should perform filtering and content checking on all files uploaded to the server. | **High** |
|  |  |
| **Day 1 Flag 6 Local File Inclusion (Advanced)**  Restrict file types that can be uploaded to those appropriate based on the feature. The application should perform filtering and content checking on all files uploaded to the server. | **High** |
|  |  |
| **Day 1 Flag 7 SQL Injection**  Input validation and limiting the characters that can be used in the password field. Encoding the output so that code does not execute. | **High** |
|  |  |
| **Day 1 Flag 8 Sensitive Data Exposure**  Sensitive information such as usernames and passwords should never be stored in plain text. Usernames and passwords should not be stored in html where any observer can see them. Visit NIST for password guidelines. Implement a stronger password policy for better security | **High** |
|  |  |
| **Day 1 Flag 9 Sensitive Data Exposure**  Do not store sensitive information on the same server as the web application. Network segmentation would help with this. Separate the server hosting the web application from the server hosting credentials, configurations and so on. | **High** |
|  |  |
| **Day 1 Flag 10 Command Injection**  WFPT successfully used commands on the networking page to list out contents of files and show directory contents. | **High** |
|  |  |
| **Day 1 Flag 11 Command Injection (Advanced)**  WFPT successfully used commands on the networking page to list out contents of files and show directory contents. | **High** |
|  |  |
| **Day 1 Flag 12 Brute Force Attack**  There are many ways to help mitigate this type of attack. It is difficult to prevent entirely but you can use some methods to deter attackers such as:   1. Add additional information to the login to discourage automated attacks. 2. Instead of account lockouts due to a number of failed login attempts, employees could sign on but have fewer privileges. 3. You can add randomized text reporting the failed login so the string length is more difficult to determine for a successful log in. 4. Update your password policy to be more inline with NIST guidelines. | **Critical** |
|  |  |
| **Day 1 Flag 13 PHP Injection**  WFPT successfully input commands on the address line of the browser page to access files in other directories. | **High** |
|  |  |
| **Day 1 Flag 14 Session Management**  When accessing the restricted area it was noticed that the session id was =”001”. Using open source tools (Burp Suite) WFPT ran this through an intruder module and was able to determine that a session # of 87 was still active and allowed us access to the restricted part of the website. | **High** |
|  |  |
| **Day 1 Flag 15 Directory Traversal**  WFPT was able to successfully use commands on a browser input line to move through various directories on the server. This type of attack is designed to access files that are not on the web server and attempts to exploit weaknesses in security to access files in other directories. The screenshot below shows an old web page that is no longer in use. | **High** |
|  |  |
| **Day 2 Flag 1 Open Source Exposed Data**  Conducting OSINT WFPT used a lookup feature. While examining the information returned from the DNS search WFPT uncovered sensitive information in plain text. In addition to the street information WFPT found a username for the remote SSH service | **High** |
|  |  |
| **Day 2 Flag 2 Open Source Exposed Data**  Conducting OSINT WFPT used a lookup feature. While examining the information returned from the DNS search WFPT uncovered sensitive information in plain text. In addition WFPT was able to duplicate this success using the command line with a simple nslookup. (‘>nslookup -type=txt totalrekall.xyz’) | **High** |
|  |  |
| **Day 2 Flag 3 Open Source Exposed Data**  Performing a normal open source search for “totalrekall.xyz” yielded sensitive information in two certificate listings. This unintentional exposure of sensitive information can definitely be used to compromise your website | **High** |
|  |  |
| **Day 2 Flag 4 Scan Results**  Scanning using nmap revealed 5 hosts up and running with the following ip addresses: 192.168.13.10/11/12/13/14 | **Low** |
|  |  |
| **Day 2 Flag 5 Aggressive nmap scan**  While performing an aggressive scan of the network 192.168.13.0/24 it was discovered that the host at .13 is running Drupal. A new vulnerability was published regarding Drupal. Verify that your version of Drupal is up to date. | **Medium** |
|  |  |
| **Day 2 Flag 6**  WFPT performed a Nessus vulnerability scan on host 192.168.13.12 and found that there was a vulnerable version of Apache Struts running. This software needs to be updated immediately. | **Critical** |
|  |  |
| **Day 2 Flag 7 Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617)**  Utilizing a known vulnerability based on the scan performed earlier, WFPT successfully compromised your installation of Apache. Continuing to update your installation will be critical to eliminating and/or at least minimizing this potential weakness. | **High** |
|  |  |
| **Day 2 Flag 8 Shellshock**  Utilizing a known exploit and taking advantage of unpatched software, WFPT was able to obtain the contents of the “sudoers” file. This file grants permission to perform administrative tasks on the server. Gaining access to this file is a serious vulnerability as attackers can modify this file to obtain persistence. | **High** |
|  |  |
| **Day 2 Flag 9 Lateral Movement**  While using the shell created by the exploit from Flag 8 WFPT was able to show the contents of the “passwd” file to obtain a list of usernames that can be used to password spray or brute force later. In addition, there was sensitive information included in the file. | **High** |
|  |  |
| **Day 2 Flag 10 Struts - CVE-2017-5638**  Utilizing an out of date software WFPT successfully hacked into the .12 machine and exploit it. | **Critical** |
|  |  |
| **Day 2 Flag 11 Drupal CVE-2019-6340**  WFPT successfully gained access to the server by exploiting a known vulnerability. Because Rekall has not updated the version of the software this exploit is available for potential bad actors to exploit. | **High** |
|  |  |
| **Day 2 Flag 12 SSH login vulnerability**  As WFPT was conducting research we found a plain text username for SSH services “alice”. WFPT conducted simple password guesses to actually login remotely. | **High** |
|  |  |
| **Day 3 Flag 1 Password Cracking**  While conducting open source research WFPT successfully found a hash file on a Git repository. Using open source tools WFPT was able to crack the password and have login in credentials. | **High** |
|  |  |
| **Day 3 Flag 2**  Performing a scan on the Windows 10 machine revealed that there were two machines on the network. When attempting to access the machine ending in .20 it asked for credentials. The credentials gained earlier worked and allowed access. | **High** |
|  |  |
| **Day 3 Flag 3**  While performing an aggressive nmap scan on the Windows machine 172.22.117.20 it was discovered that anonymous ftp login was allowed. | **High** |
|  |  |
| **Day 3 Flag 4**  After performing an aggressive nmap scan it was discovered that Rekall is running a vulnerable version of SLMail. | **High** |
|  |  |
| **Day 3 Flag 5**  While examining the scheduled tasks on the Windows 10 machine it was revealed that sensitive information was contained in one of the scheduled tasks. | **High** |
|  |  |
| **Day 3 Flag 6**  Once WFPT successfully compromised the Windows machine we were able to dump the SAM file using a kiwi command. | **High** |
|  |  |
| **Day 3 Flag 7**  Utilizing offline tools WFPT successfully cracked the NTLM hash and was able to find that the username is ‘?’ and the password is ‘Computer!’ | **Medium** |
|  |  |
| **Day 3 Flag 8**  While exploring the file system using Kiwi, WFPT successfully found the username “ADMBob” and a hash to use to attempt to | **High** |
|  |  |
| **Day 3 Flag 9**  Once the credentials for the administrator were compromised, WFPT was able to navigate through directories to access various files. | **High** |
|  |  |
| **Day 3 Flag 10**  WFPT successfully obtained administrator credentials from a search using kiwi once we had gained access to the system. | **High** |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | [www.totalrekall.xyz](http://www.totalrekall.xyz)  3.33.130.190  15.197.148.33  192.168.13.0/24 |
| Ports | ftp (21)  SMB (445)  HTTP (80) |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 3 |
| **High** | 31 |
| **Medium** | 2 |
| **Low** | 1 |

## Vulnerability Findings

### DAY 1

| **Day 1 Flag 1** | **Findings** |
| --- | --- |
| **Title** | Cross Site Scripting (XSS reflected) |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | WFPT successfully added a script to the input box to manipulate the web application. |
| **Images** |  |
| **Affected Hosts** | www.www.totalrekall.xyz |
| **Remediation** | Mitre techniques T1190, T1587, and T1588  <https://portswigger.net/web-security/cross-site-scripting/preventing>  Use more robust input validation. The validation set up on the comments page was easily bypassed.  Add input validation and encoding to any user supplied field on your website.  Encode output data from the user before it is executed on the website.  <https://owasp.org/www-community/attacks/xss/> |

| **Day 1 Flag 2** | **Findings** |
| --- | --- |
| **Title** | Crosssite Scripting Reflected Advanced |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | WFPT successfully input a multi-complex script in the input box. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Mitre techniques T1190, T1587, and T1588  <https://portswigger.net/web-security/cross-site-scripting/preventing>  Use more robust input validation. The validation set up on the comments page was easily bypassed.  Add input validation and encoding to any user supplied field on your website.  Encode output data from the user before it is executed on the website.  <https://owasp.org/www-community/attacks/xss/> |

| **Day 1 Flag 3** | **Findings** |
| --- | --- |
| **Title** | Crossite Scripting (stored) |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | WFPT successfully added a comment that executes code each time the page is loaded. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Mitre techniques T1190, T1587, and T1588  <https://portswigger.net/web-security/cross-site-scripting/preventing>  Use more robust input validation. The validation set up on the comments page was easily bypassed.  Add input validation and encoding to any user supplied field on your website.  Encode output data from the user before it is executed on the website.  <https://owasp.org/www-community/attacks/xss/> |

| **Day 1 Flag 4** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | Utilizing commands from the command line WFPT was able to show the information in the headers of your webpages. One of these pages is the about page where looking at the headers revealed sensitive information. This weakness can be exposed with a simple curl command from the CLI or via Burp Suite. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Mitre techniques T1190, T1587 and T1588.  Remove the sensitive information from the headers of your website.  If the information must be in the header, encrypt the information before putting it in the header.  <https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure>  <https://portswigger.net/support/using-burp-to-test-for-sensitive-data-exposure-issues> |

| **Day 1 Flag 5** | **Findings** |
| --- | --- |
| **Title** | Local File Inclusion |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | Utilizing the VR planner page WFPT was able to upload a ‘.php’ file to the server. There was no validation of the file extension upon uploading. This is dangerous because attackers could potentially upload a malicious script to the server that could be executed at some later date. Mitigation is necessary to eliminate the ability of users to upload certain types of files. Since the files are pictures you should limit the file types that can be uploaded using this feature. Only image files should be allowed. Further, content checking is necessary since WFPT was able to change the file name (i.e. hello.php to hello.jpg.php) to fool the validation |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Mitre techniques T1204, T1587 and T1588.  Restrict file types that can be uploaded to those appropriate based on the feature.  <https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload>  The application should perform filtering and content checking on all files uploaded to the server. |

| **Day 1 Flag 6** | **Findings** |
| --- | --- |
| **Title** | LFI Advanced |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | Even though there was some validation on another field, WFPT successfully bypassed the validation by changing the file extension to still be able to upload a potentially malicious file. Utilizing a file name such as “hello.jpg.php” successfully bypasses input validation. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Mitre techniques T1204, T1587 and T1588.  Restrict file types that can be uploaded to those appropriate based on the feature.  <https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload>  The application should perform filtering and content checking on all files uploaded to the server. |

| **Day 1 Flag 7** | **Findings** |
| --- | --- |
| **Title** | SQL injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | By inputting the following in the password field of the login page WFPT successfully injected sql code into the website. “ok' or ‘1=1–. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | MITRE techniques T1204, T1587 and T1588  <https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html>  Input validation and limiting the characters that can be used in the password field.  Encoding the output so that code does not execute. |

| **Day 1 Flag 8** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | By examining the source code for the login page on the website WFPT found a password and username in plain text. By going back to the page WFPT used these credentials to successfully login as an administrator to the website. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Mitre techniques T1078, T1190, T1587 and T1588.  Sensitive information such as usernames and passwords should never be stored in plain text.  Usernames and passwords should not be stored in html where any observer can see them.  Visit NIST for password guidelines.  <https://pages.nist.gov/800-63-3/sp800-63b.html>  Implement a stronger password policy for better security. |

| **Day 1 Flag 9** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | There is a text file located in the same directory as the web application. This is a common file that tells search engine crawlers which pages they may access on your site. This file was accessible simply by typing the name into the address bar of the web browser. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | <https://developers.google.com/search/docs/crawling-indexing/robots/intro>  <https://owasp.org/www-project-top-ten/>  Do not store sensitive information on the same server as the web application.  Network segmentation would help with this. Separate the server hosting the web application from the server hosting credentials, configurations and so on. |

| **Day 1 Flag 10** | **Findings** |
| --- | --- |
| **Title** | Command Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | On the networking page accessed after inputting administrator credentials WFPT was able to execute several commands from the DNS lookup field. WFPT was able to display sensitive information from the server. See the Mitre page below. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Mitre Techniques: T1203, T1587, and T1588  Sensitive information such as usernames and passwords should never be stored in plain text.  Network segmentation can help prevent this type of attack by separating sensitive files from being accessed by the web application.  Control access to critical information by changing the permission levels for who can see sensitive files.  Input validation and output encoding will help to resolve this issue.  <https://owasp.org/www-community/attacks/Command_Injection>  <https://attack.mitre.org/techniques/T1203/> |

| **Day 1 Flag 11** | **Findings** |
| --- | --- |
| **Title** | Command injection (advanced) |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | Even with the small amount of input validation on the “mx record check” field, it was easily bypassed on the input field. This is a well-known easily exploitable feature. See the Mitre page below. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Mitre Techniques: T1203, T1587, and T1588  Sensitive information such as usernames and passwords should never be stored in plain text.  Network segmentation can help prevent this type of attack by separating sensitive files from being accessed by the web application.  Control access to critical information by changing the permission levels for who can see sensitive files.  Input validation and output encoding will help to resolve this issue.  <https://owasp.org/www-community/attacks/Command_Injection>  <https://attack.mitre.org/techniques/T1203/> |

| **Day 1 Flag 12** | **Findings** |
| --- | --- |
| **Title** | Brute Force Attack |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | Critical |
| **Description** | After utilizing command injection WFPT successfully displayed a list of users from the ‘etc/passwd’ file. Using this, WFPT utilized simple password guesses for a user “melina” and were able to crack this users password and gain access. At this point WFPT has gained credentials for a user on the network (malena) and an administrator (doug quaid). In both cases this is due to weak password policy and bad security policy.  Brute force attacks are difficult to stop but easy to detect because each failed login records an HTTP 401 error code in the web server logs. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Add additional information to the login (“CAPTCHA”) to discourage automated attacks.  Instead of account lockouts due to a number of failed login attempts, employees could sign on but have fewer privileges.  You can add randomized text reporting the failed login so the string length is more difficult to determine for a successful log in.  <https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks>    <https://owasp.org/www-community/attacks/Brute_force_attack>    Detect your web servers being scanned by brute force tools such as WFuzz, OWASP DirBuster and vulnerability scanners such as Nessus, Nikto, Acunetix, etc. This helps you quickly identify probable probing by bad actors who want to dig possible security holes.  Update your password policy to be more in line with NIST guidelines.  <https://www.auditboard.com/blog/nist-password-guidelines/> |

| **Day 1 Flag 13** | **Findings** |
| --- | --- |
| **Title** | PHP Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | WFPT was able to successfully inject commands into the command line of the browser. By utilizing the correct syntax WFPT was able to inject commands and get sensitive information from the server such as user id and directory listing. As you can see in the screenshot below, adding a command on the browser allowed WFPT to obtain the user id of the current user - “www-data”. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Mitre tactics: T1587, T1588, and T1659  For a comprehensive discussion about what php injection is see the following link: <https://www.stackhawk.com/blog/php-command-injection/>  For mitigation strategies see the following: <https://snyk.io/blog/prevent-php-code-injection/> |

| **Day 1 Flag 14** | **Findings** |
| --- | --- |
| **Title** | Session Management |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | When accessing the restricted area it was noticed that the session id was =”001”. Using open source tools (Burp Suite) WFPT ran this through an intruder module and was able to determine that a session # of 87 was still active and allowed us access to the restricted part of the website. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | <https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html>  <https://owasp.org/www-project-mobile-top-10/2014-risks/m9-improper-session-handling>  Sessions and cookies should not be easy to guess. You should use secure cookies so that cookies are encrypted. See below:  <https://owasp.org/www-community/controls/SecureCookieAttribute>  Do not display session information directly in the address bar of the browser. |

| **Day 1 Flag 15** | **Findings** |
| --- | --- |
| **Title** | Directory Traversal |
| **Type (Web app / Linux OS / WIndows OS)** | Web Application |
| **Risk Rating** | High |
| **Description** | WFPT was able to successfully use commands on a browser input line to move through various directories on the server. This type of attack is designed to access files that are not on the web server and attempts to exploit weaknesses in security to access files in other directories. The screenshot below shows an old web page that is no longer in use. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Mitre tactics: T1059  Use input validation to control what inputs will be accepted through any user accessible portal.  Remove old web pages from the web application directory and keep them offline.  Don’t allow the input field to be the entire string of code. Add code to the beginning and end of what the user types into the field.  <https://owasp.org/www-community/attacks/Path_Traversal> |

### DAY 2

| **Day 2 Flag 1** | **Findings** |
| --- | --- |
| **Title** | Open source exposed data |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | Conducting OSINT WFPT used a lookup feature. While examining the information returned from the DNS search WFPT uncovered sensitive information in plain text. In addition to the street information WFPT found a username for the remote SSH service. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Remove the sensitive information from the DNS records. |

| **Day 2 Flag 2** | **Findings** |
| --- | --- |
| **Title** | Open source exposed data |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | Conducting OSINT WFPT used a lookup feature. While examining the information returned from the DNS search WFPT uncovered sensitive information in plain text. In addition WFPT was able to duplicate this success using the command line with a simple nslookup. (‘>nslookup -type=txt totalrekall.xyz’) |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Remove the sensitive information from the data section of the DNS listing.  If the file must remain on the server, encrypt the file with a password and rename it. |

| **Day 2 Flag 3** | **Findings** |
| --- | --- |
| **Title** | Open source exposed data |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | Performing a normal open source search for “totalrekall.xyz” yielded sensitive information in two certificate listings. This unintentional exposure of sensitive information can definitely be used to compromise your website. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Remove the sensitive information from the CA listing  Do not include sensitive information in any CA request or listing in the future. |

| **Day 2 Flag 4** | **Findings** |
| --- | --- |
| **Title** | Scan results |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Low |
| **Description** | Scanning using nmap revealed 5 hosts up and running with the following ip addresses: 192.168.13.10/11/12/13/14 |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | The important thing is to keep all of your software up to date. If someone is scanning using a bot, they are looking for servers that have known vulnerabilities. Keeping the servers up to date keeps this as a piece of information rather than a vulnerability. |

| **Day 2 Flag 5** | **Findings** |
| --- | --- |
| **Title** | Aggressive nmap scan |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Medium |
| **Description** | While performing an aggressive scan of the network 192.168.13.0/24 it was discovered that the host at .13 is running Drupal. A new vulnerability was published regarding Drupal. Verify that your version of Drupal is up to date. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Update to a current version of Drupal  Visit the following website for specific actions to take to mitigate this risk. <https://nvd.nist.gov/vuln/detail/CVE-2024-22362> |

| **Day 2 Flag 6** | **Findings** |
| --- | --- |
| **Title** | Nessus scan results |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | WFPT performed a Nessus vulnerability scan on host 192.168.13.12 and found that there was a vulnerable version of Apache Struts running. This software needs to be updated immediately. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Patch the version of Apache Struts immediately.  Ensure that all software is on a routine schedule to download and install updates and patches to keep all vulnerabilities minimized.  <https://nvd.nist.gov/vuln/detail/CVE-2017-12617> |

| **Day 2 Flag 7** | **Findings** |
| --- | --- |
| **Title** | Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617) |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | Utilizing a known vulnerability based on the scan performed earlier, WFPT successfully compromised your installation of Apache. Continuing to update your installation will be critical to eliminating and/or at least minimizing this potential weakness. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | <https://nvd.nist.gov/vuln/detail/CVE-2017-12617>  Patch the Apache software to remove this vulnerability  Ensure that all software remains up to date and patched by implementing a routine update schedule for all software on your system. |

| **Day 2 Flag 8** | **Findings** |
| --- | --- |
| **Title** | Shellshock |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | Utilizing a known exploit and taking advantage of unpatched software, WFPT was able to obtain the contents of the “sudoers” file. This file grants permission to perform administrative tasks on the server. Gaining access to this file is a serious vulnerability as attackers can modify this file to obtain persistence. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Patch the software vulnerability immediately  Put all software on a routine update schedule so that vulnerabilities are patched in a timely manner. |

| **Day 2 Flag 9** | **Findings** |
| --- | --- |
| **Title** | Lateral Movement |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | While using the shell created by the exploit from Flag 8 WFPT was able to show the contents of the “passwd” file to obtain a list of usernames that can be used to password spray or brute force later. In addition, there was sensitive information included in the file. |
| **Images** |  |
| **Affected Hosts** | Scrub the etc/passwd file to ensure no sensitive information is included.  Patch the software to remove the vulnerability from the “shellshock” exploit. |
| **Remediation** |  |

| **Day 2 Flag 10** | **Findings** |
| --- | --- |
| **Title** | Struts - CVE-2017-5638 |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | Utilizing an out of date software WFPT successfully hacked into the .12 machine and exploit it. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Patch the software to remove the “struts”vulnerability  <https://nvd.nist.gov/vuln/detail/cve-2017-5638> |

| **Day 2 Flag 11** | **Findings** |
| --- | --- |
| **Title** | Drupal - CVE-2019-6340 |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | WFPT successfully gained access to the server by exploiting a known vulnerability. Because Rekall has not updated the version of the software this exploit is available for potential bad actors to exploit. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Patch the vulnerability in Drupal.  Set all software to be on a routine update schedule so that any known vulnerabilities get patched as soon as they are identified.  Drupal - CVE-2019-6340  <https://nvd.nist.gov/vuln/detail/CVE-2019-6340> |

| **Day 2 Flag 12** | **Findings** |
| --- | --- |
| **Title** | SSH login Vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | High |
| **Description** | As WFPT was conducting research we found a plain text username for SSH services “alice”. WFPT conducted simple password guesses to actually login remotely. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | <https://nvd.nist.gov/vuln/detail/CVE-2019-14287>  Update password policies to be more in line with NIST guidelines.  <https://pages.nist.gov/800-63-3/sp800-63b.html>  Implement a stronger password policy to deter simple guessing from gaining access to the system. |

### DAY 3

| **Day 3 Flag 1** | **Findings** |
| --- | --- |
| **Title** | Password Cracking |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | While conducting open source research WFPT successfully found a hash file on a Git repository. Using open source tools WFPT was able to crack the password and have login in credentials. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Do not store password hashes in easily accessible locations. While this was a Git repository other locations can be a desktop or other easily accessible locations. |

| **Day 3 Flag 2** | **Findings** |
| --- | --- |
| **Title** | Port Scan Vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | Performing a scan on the Windows 10 machine revealed that there were two machines on the network. When attempting to access the machine ending in .20 it asked for credentials. The credentials gained earlier worked and allowed access. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Remove all credentials from publicly accessible locations: Git, directories, files etc.  Establish better password controls. Either require users to update their passwords regularly or require them to be sufficiently difficult to guess. |

| **Day 3 Flag 3** | **Findings** |
| --- | --- |
| **Title** | Anonymous ftp login |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | While performing an aggressive nmap scan on the Windows machine 172.22.117.20 it was discovered that anonymous ftp login was allowed. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Do not allow anonymous ftp login  Establish a robust password policy that requires difficult to guess passwords and/or ones that change frequently. |

| **Day 3 Flag 4** | **Findings** |
| --- | --- |
| **Title** | SLMail Vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | After performing an aggressive nmap scan it was discovered that Rekall is running a vulnerable version of SLMail. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Update the mail handling software to eliminate this weakness. |

| **Day 3 Flag 5** | **Findings** |
| --- | --- |
| **Title** | Sensitive Information on Scheduled Tasks |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | While examining the scheduled tasks on the Windows 10 machine it was revealed that sensitive information was contained in one of the scheduled tasks. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Remove the sensitive information from the comments section of the scheduled task. |

| **Day 3 Flag 6** | **Findings** |
| --- | --- |
| **Title** | SAM file compromise |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | Once WFPT successfully compromised the Windows machine we were able to dump the SAM file using a kiwi command. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Remove the sensitive information from the sam file.  Update the SLMail software to remove this vulnerability. |

| **Day 3 Flag 7** | **Findings** |
| --- | --- |
| **Title** | Sensitive Information in accessible file location |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Medium |
| **Description** | Utilizing offline tools WFPT successfully cracked the NTLM hash and was able to find that the username is ‘?’ and the password is ‘Computer!’ |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Move the sensitive information to a more secure location.  password protect the sensitive information  encrypt the sensitive information. |

| **Day 3 Flag 8** | **Findings** |
| --- | --- |
| **Title** | Credentials listed in accessible file |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | While exploring the file system using Kiwi, WFPT successfully found the username “ADMBob” and a hash to use to attempt to |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Remove this listing.  Use a more secure method to encrypt the password of the administrators. |

| **Day 3 Flag 9** | **Findings** |
| --- | --- |
| **Title** | Directory Traversal |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | Once the credentials for the administrator were compromised, WFPT was able to navigate through directories to access various files. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Limit the locations each user can access  Control how and when each user can access resources. |

| **Day 3 Flag 10** | **Findings** |
| --- | --- |
| **Title** | Privilege Escalation |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | High |
| **Description** | WFPT successfully obtained administrator credentials from a search using kiwi once we had gained access to the system. |
| **Images** |  |
| **Affected Hosts** | www.totalrekall.xyz |
| **Remediation** | Find a more secure method to store administrator passwords. |