| 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](#_q44pijayqthm) 2

[Contact Information](#_oj1z2ita1tzd) 4

[Document History](#_otyf6rjzay1l) 4

[Introduction](#_nb9r73bykkqu) 5

[Assessment Objective](#_3dy6vkm) 5

[Penetration Testing Methodology](#_tgl1cd57dphe) 6

[Reconnaissance](#_17dp8vu) 6

[Identification of Vulnerabilities and Services](#_3rdcrjn) 6

[Vulnerability Exploitation](#_26in1rg) 6

[Reporting](#_lnxbz9) 6

[Scope](#_zf3a4holuf89) 7

[Executive Summary of Findings](#_x5kpi12umwq3) 8

[Grading Methodology](#_z337ya) 8

[Summary of Strengths](#_nntf0x9u7qng) 9

[Summary of Weaknesses](#_1y810tw) 9

[Executive Summary](#_53qin3q0b0zf) 10

[Rekall web app](#_tesz7376u80q) 10

[Rekall Linux servers](#_s7l7mtvih8h2) 12

[Rekall Windows servers](#_3nvd3iz168xk) 13

[Summary Vulnerability Overview](#_tr5kilpcz7z8) 15

[Vulnerability Findings](#_ft98cc3eh3ql) 16

# 

## Contact Information

| **Company Name** | John Mercer Security |
| --- | --- |
| **Contact Name** | John Mercer |
| **Contact Title** | Investigator |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 07/24/2022 | John Mercer | initial version |

# 

## 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.

Inscope: 172.22.117.0/24, 192.168.13.0/24, 192.168.14.0/24

Excluded: 34.102.136.180

# 

# 

## 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.

* Password complexity is OK, but not fully enforced.

### 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.

* The Rekall web app has a number of fields that do not have sufficient input validation and permit attack.
* The Rekall web application has a login page vulnerable to SQL injection.
* The Rekall web application has a number of pages with parameters in the URL that do not have sufficient input validation and permit attack.
* The Rekall web application has a number of files on the server that are accessible and contain sensitive information.
* Publicly available information about the domain, SSL certificate, and GitHub repository contain sensitive data.
* Apache applications on several Linux servers subject to multiple vulnerabilities and versions should be updated.
* Sudo vulnerability exists and should be updated.
* Anonymous FTP allows access to files.
* Seattle Lab Mail on Windows system has a vulnerability and should be updated.
* LSASS credential dumping on Windows servers exposes password hashes.
* Windows password weakness allows for lateral movement.
* Weak passwords allow for brute force attacks.

## Executive Summary

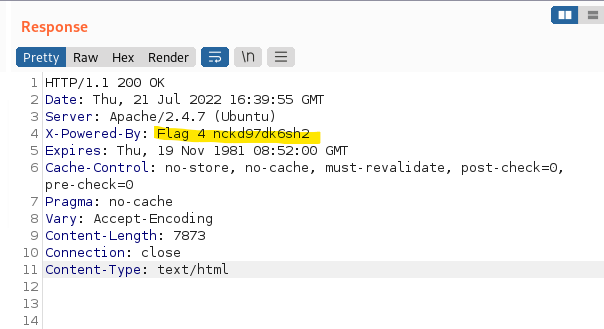
## Rekall web app

In the Rekall web application, on the welcome.php page if you include a script such as <script>alert(“hello”)</script> in the “Put your name here” field, the first flag will be displayed because there is no input validation.

On the Memory-Planner.php page, you can add a similar script to the “Choose your charachter” [sic] field and display flag 2. However, since this field has basic input validation that will remove the word “script” from the input, you need to enter it twice, such as <sscriptcript>alert(“hello”)</sscriptcript>.

On the Comments.php page, You can add a script like<script>alert(“hello”)</script> to the comments field to produce flag 3 due to similar lack of input validation.

Flag 4 is exposed in the header of the response HTML for the pages.



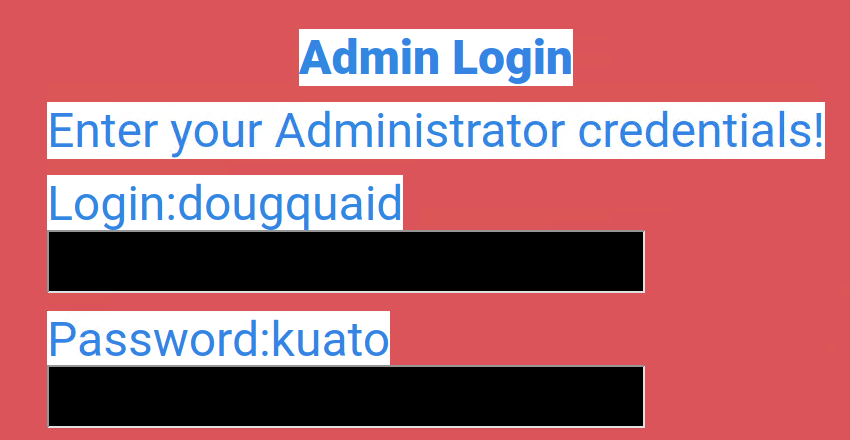
Flag 5 is on the Memory-Planner.php page in the “Choose you Adventure” file upload button. If you upload a php file instead of a picture, the flag is shown, so there is no verification that the file is actually a picture.

Flag 6 is on the same Memory-Planner.php page on the “Choose your location” file upload button. It checks to verify the filename includes “.jpg”, but still does not verify that the file is a picture.

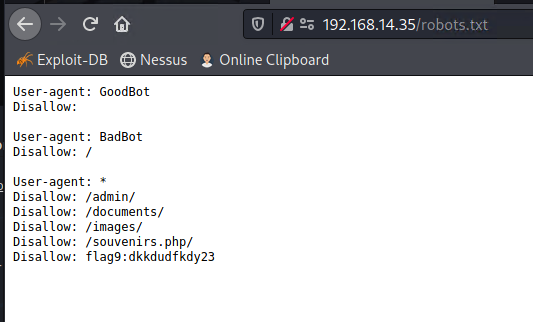
Flag 7 is accessible via a SQL injection on the login.php page. If the password field for the user section has a malicious entry (such as “ x’ or ‘1’=’1 “) it will show the flag.

Flag 7 is also accessible if you login using usernames and passwords listed in passwords/heroes.xml.

Flag 8 is included in the HTML for the logon.php page, and is only hidden by using text the same color as the page’s background. Simply selecting the page contents reveals it.



Flag 9 is in the robots.txt file at the root of the website. While this file is useful, it is best to not include sensitive information in it.



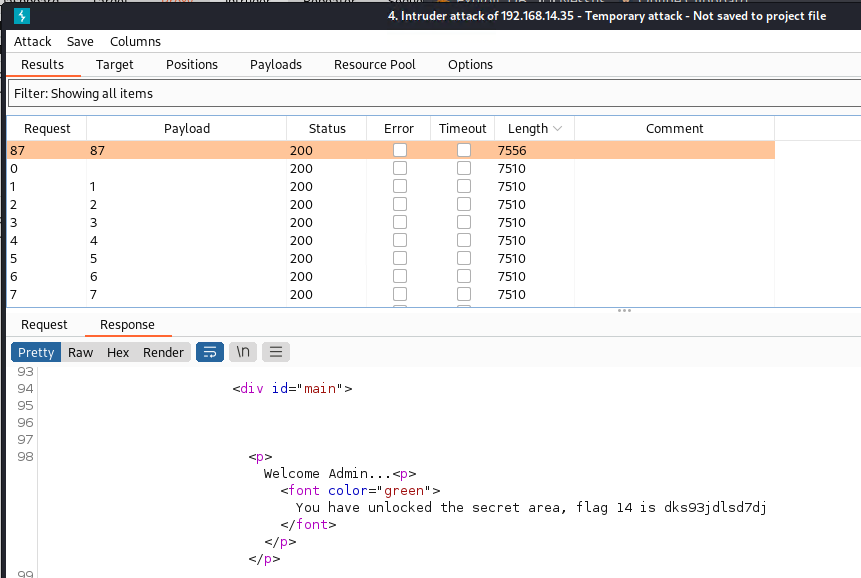
Flag 10 is discovered on the Networking.php page, which is linked to from the login page when you log in as the administrator from flag 8. The DNS Check field does not verify input and allows for command injection such as “www.example.com|cat vendors.txt”.

Flag 11 is also discovered on the Networking.php page and has the same vulnerability on the MX Record Checker field. The same command injection works here too.

For flag 12, we continue to use the command injection vulnerability to browse the server and read the /etc/passwd file to find the account melina. We can guess at the password and it turns out to be the same as the username (melina). Entering it on the login.php page shows us flag 12.

In the robots.txt file, we learn of the existence of the Souvenirs.php page, which takes a message parameter in the URL. If you modify the message to include a command (like ;system('whoami')) you will get flag 13.

When you log in as melina for flag 12, you get a link to the Admin\_legal\_data.php page. The URL for this page has an admin parameter that references an admin or session number. After trying a few with automated tools, we found one that shows flag 14.



From the Welcome page there is a link to the Disclaimers.php page that includes a page parameter which allows you to view files on the site at will. If you display the file old\_disclaimers/disclaimer\_1.txt, it will also show you flag 15. Simply viewing this file via command injection does not reveal the flag; it must be done via this page.

## Rekall Linux servers

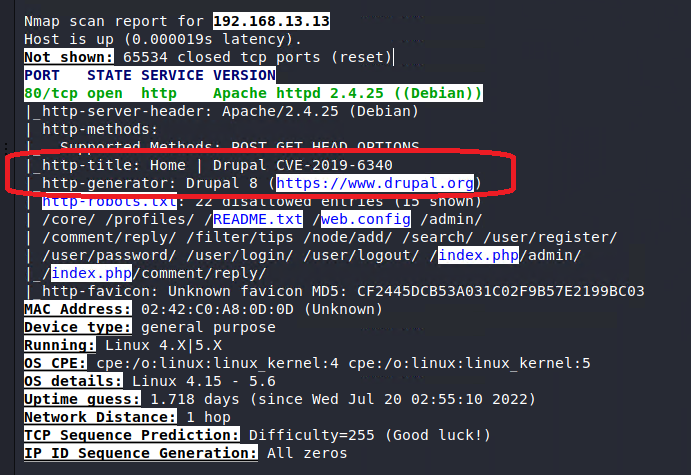
Registering for a website and filling in the information on who owns it is normal behavior, but you should verify that the information provided is not compromising. Flag 1 is included in the publicly available WHOIS data for the totalrekall.xyz domain.

Flag 2 is in the same location. Understand that public IP addresses need to be available in DNS for users to find your site.

Flag 3 is in the SSL certificate information for the totalrekall.xyz domain, also publicly available.

Flag 4 is simply counting the number of systems that respond to an nmap scan of the network.

Flag 5 is looking at those same nmap scan results for a system running the Drupal application.



If you run a Nessus vulnerability scan on the machine at 192.168.13.12, you will find it has one critical vulnerability with Nessus ID number 97610 (which is flag 6). We will use this vulnerability later for flag 10.



Flag 7 is found by exploiting a vulnerability in Apache Tomcat on server 192.168.13.10. On that server you can find the hidden file /root/.flag7.txt.

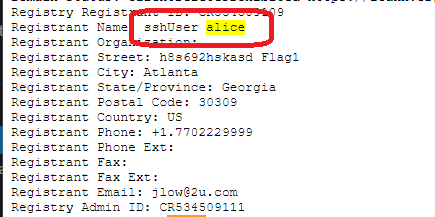
Flag 8 can be found by exploiting the ShellShock vulnerability in Apache on 192.168.13.11. On that server you can find the flag in /etc/sudoers.

Flag 9 is on the same server but in the /etc/passwd file as a username.

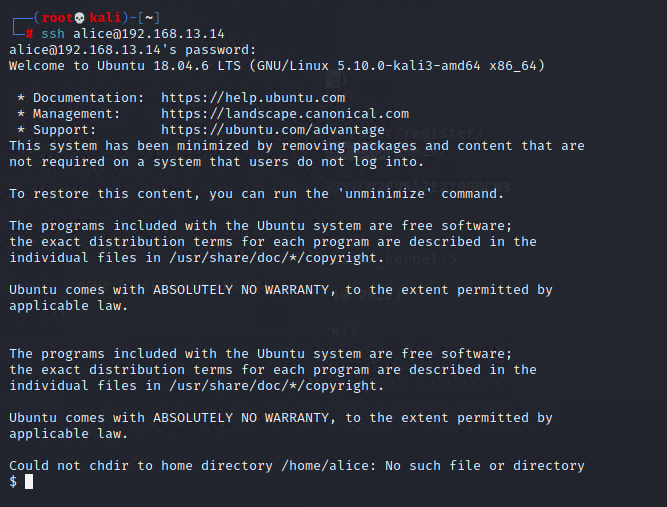
Flag 10 is found on server 192.168.13.12 using the vulnerability exposed in the Nessus scan. Once connected to the server, we can search for the flag file and locate /root/flagisinThisfile.7z. After using meterpreter to download the file locally and 7zip to unzip it, we get the flag.

Flag 11 is on the server identified as running Drupal (flag 5), which has a known vulnerability. The flag is the account name on the machine the exploit is using.

In the publicly available WHOIS data (flag 1) is other useful information, such as an SSH username.



Using that name and guessing the password, we can connect to the server at 192.168.13.14.



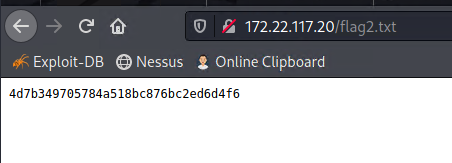
Using a privilege escalation vulnerability, we can get root access and find the file for flag 12.

## 

## Rekall Windows servers

Flag 1 is included in the publicly available GitHub data for the totalrekall. It shows a username and password hash that is easily cracked.

Flag 2 is available on the website of the Windows 10 system using the obtained username and password from flag 1.



Flag 3 is available on the FTP site of the Windows 10 system using anonymous login.

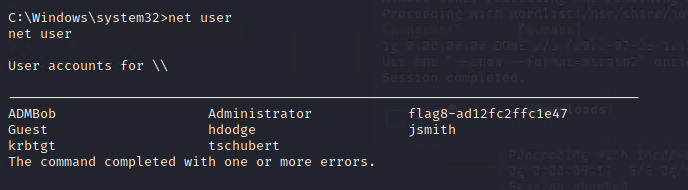
Flag 4 is found on the Windows 10 system using an exploit in SLMail.

Flag 5 is found on the Windows 10 system as the comment for a task named flag 5.

Flag 6 is found as a local user on the Windows10 system. The password was extracted using an LSA dump and cracked using John the Ripper.

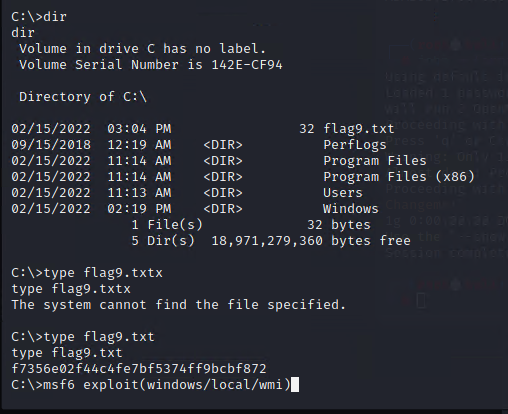
Flag 7 is found in C:\Users\Public\Documents\flag7.txt on the Windows 10 system.

For flag 8 we started with an LSA cache dump to look for domain accounts and found ADMBob and the password hash. Using John the Ripper, we cracked the password. We then used that account to connect to the WinDC domain controller and found the user named flag 8.



Alternatively, flag 8 can be found by connecting directly to the domain controller using a different vulnerability that results in system level permissions.

Flag 9 is found at C:\flag9.txt on the WinDC system, showing we have access to the files on the server.



Flag 10 is the hash of the administrator account on the domain controller machine.

## 

## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| XSS Reflected | **Medium** |
| XSS Stored | **Medium** |
| Sensitive data exposure in HTML header | **Low** |
| Local File Inclusion | **Medium** |
| SQL Injection | **Critical** |
| Sensitive data exposure in HTML body | **Critical** |
| Sensitive data exposure in file | **Low** |
| Command Injection | **Critical** |
| Brute Force Attack | **Critical** |
| PHP Injection | **High** |
| Session Management | **Low** |
| Directory Traversal | **Low** |
| Sensitive data exposure in file | **Critical** |
| Open source exposed data | Informational |
| Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617) | **Critical** |
| GNU Bash Environment Variable Command Injection Vulnerability (CVE-2014-6271) a.k.a. ShellShock | **Critical** |
| Sensitive data exposure in file | **Medium** |
| Apache Struts Jakarta Multipart parser vulnerability (CVE-2017-5638) a.k.a. Struts | **Critical** |
| Drupal core - Highly critical - Remote Code Execution (CVE-2019-6340) | **Critical** |
| Sudo policy bypass (CVE-2019-14287) | **Critical** |
| Open source exposed data | Informational |
| Sensitive data exposure, anonymous FTP | **Medium** |
| Seattle Lab Mail 5.5 POP3 Buffer Overflow (CVE-2003-0264) | **Critical** |
| Sensitive data exposure in service | **Critical** |
| LSASS credential dumping - local account | **Critical** |
| Sensitive data exposure in file | **Critical** |
| LSASS credential dumping - domain account | **High** |
| Microsoft Windows Authenticated User Code Execution (CVE-1999-0504) | **High** |
| Microsoft Windows Authenticated User Code Execution (CVE-1999-0504) | **Critical** |
| LSASS credential dumping - local account | **Critical** |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | 192.168.14.35  192.168.13.10-14  172.22.117.10  172.22.117.20 |
| Ports | 21, 22, 25, 79, 80, 106, 110, 135, 139, 389, 8009, 8080 |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 16 |
| **High** | 3 |
| **Medium** | 5 |
| **Low** | 4 |
| Informational | 2 |

## Vulnerability Findings

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | XSS Reflected |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Medium** |
| **Description** | Welcome.php Username field and Memory-Planner.php “Choose your character” field both vulnerable to Cross-Site-Scripting attack where a script can be entered into the field. <script>alert(“hello”)</script> for flag 1 and <sscriptcript>alert(“hello”)</sscriptcript> for flag 2. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | better input validation |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | XSS Stored |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Medium** |
| **Description** | Comments.php comments field vulnerable to Cross-Site-Scripting attack where a script can be entered into the field and saved on page. Such as <script>alert(“hello”)</script> for flag 3. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | better input validation |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | Sensitive data exposure in HTML header |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Low** |
| **Description** | Potentially compromising information provided in the HTML response header, visible when using Burp Suite to view the site. Flag 4 revealed. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Review Header information on pages and responses. |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Local File Inclusion |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Medium** |
| **Description** | Memory\_Planner.php page has two locations (“Choose your Adventure” and “Choose your Location”) that allow for uploading a file. Both are expecting an image file (e.g. .jpg) but neither verifies that an image is actually uploaded; the location field only checks that .jpg is in the filename. Uploading a .php file reveals flags 5 and 6. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | better input validation |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | SQL Injection |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Critical** |
| **Description** | Login.php has a field to enter a password that is vulnerable to SQL injection and could allow for a non-user to log in by entering something like x' or '1' = '1 as the password. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | better input validation |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Sensitive data exposure in HTML body |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Critical** |
| **Description** | The login.php page includes an administrative username and password in the HTML for the page. SImply selecting the page contents shows flag 8. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Review HTML for sensitive data. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Sensitive data exposure in file |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Low** |
| **Description** | The file robots.txt is normally used to control search engine robots that scan the web. But in this case flag 9 is included in the file. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Review robots.txt for sensitive data. |

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** | Command Injection |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Critical** |
| **Description** | Networking.php has two fields that are expecting a domain name for DNS lookup or MX lookup, but neither have input validation and are vulnerable to a command injection attack. Simply appending “|cat vendors.txt” to the domain name allows sending commands to the underlying OS and reveals flags 10 and 11. Later use of this vulnerability allowed for easy browsing the server’s files and folder structure. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | better input validation |

| **Vulnerability 9** | **Findings** |
| --- | --- |
| **Title** | Brute Force Attack |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Critical** |
| **Description** | Browsing the server using the Command Injection vulnerability exposed /etc/passwd which includes usernames. Entering those names on the login.php page and guessing at the passwords allowed us to log in and obtain flag 12. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Better password policies that prevent having the username as a password. |

| **Vulnerability 10** | **Findings** |
| --- | --- |
| **Title** | PHP Injection |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **High** |
| **Description** | Souviners.php has a message parameter in the URL that allows adding system commands (such as ;system('whoami')) to give an attacker the ability to run commands on the system. Doing so shows flag 13. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | better input validation |

| **Vulnerability 11** | **Findings** |
| --- | --- |
| **Title** | Session Management |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Low** |
| **Description** | Admin\_legal\_data.php has a parameter for admin or session with a numerical value. By scanning and entering the correct value, you can unlock the admin area and get flag 14. For the page as is, the attacker doesn’t have access to any specific additional admin access, but that depends on the data stored in this restricted area so the risk could be higher. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Don’t tie the admin access to a persistent ID number, rather use the login process. |

| **Vulnerability 12** | **Findings** |
| --- | --- |
| **Title** | Directory Traversal |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Low** |
| **Description** | Disclaimers.php includes a page parameter which allows you to view files on the server at will. If you display the file old\_disclaimers/disclaimer\_1.txt, it will also show you flag 15. Access is limited to the account www-data. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Better input validation or lock down this page to only show files from a specific folder. |

| **Vulnerability 13** | **Findings** |
| --- | --- |
| **Title** | Sensitive data exposure in file |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | **Critical** |
| **Description** | Using the Command Injection and Directory Traversal vulnerabilities, locate and read /passwords/heroes.xml. Use the information in that file to log in as a user on login.php to expose flag 7 via alternate route. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Password policies that prevent storing passwords in cleartext in files. |

| **Vulnerability 14** | **Findings** |
| --- | --- |
| **Title** | Open source exposed data |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS and Windows OS |
| **Risk Rating** | Informational |
| **Description** | Flags 1 and 2 are exposed in the information submitted when registering the domain totalrekall.xyz. Link: <https://centralops.net/co/DomainDossier.aspx> |
| **Images** |  |
| **Affected Hosts** | totalrekall.xyz domain |
| **Remediation** | review publicly available documents for sensitive data. |

| **Vulnerability 15** | **Findings** |
| --- | --- |
| **Title** | Open source exposed data |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS and Windows OS |
| **Risk Rating** | Informational |
| **Description** | Flag 3 is in the SSL certificate information for the totalrekall.xyz domain. While having an SSL certificate is normal, it is best to review the information provided when obtaining and registering it. |
| **Images** |  |
| **Affected Hosts** | totalrekall.xyz domain |
| **Remediation** | Review publicly available documents for sensitive data. |

| **Vulnerability 16** | **Findings** |
| --- | --- |
| **Title** | Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617) |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | Using Metasploit exploit(multi/http/tomcat\_jsp\_upload\_bypass), you can gain remote code execution on 192.168.13.10 as root. Browsing the files shows flag 7. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.10 |
| **Remediation** | Apply updates per vendor instructions. |

| **Vulnerability 17** | **Findings** |
| --- | --- |
| **Title** | GNU Bash Environment Variable Command Injection Vulnerability (CVE-2014-6271) a.k.a. ShellShock |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | Using the Metasploit exploit(multi/http/apache\_mod\_cgi\_bash\_env\_exec) you can gain remote code execution on 192.168.13.11 as www-data. Even with that access you can read /etc/sudoers and find flag 8. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.11 |
| **Remediation** | Apply updates per vendor instructions. |

| **Vulnerability 18** | **Findings** |
| --- | --- |
| **Title** | Sensitive data exposure in file |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Medium** |
| **Description** | On the same server as flag 8 (using the same exploit) you can browse the files and find flag 9 in the file /etc/passwd. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.11 |
| **Remediation** | Apply updates per vendor instructions. |

| **Vulnerability 19** | **Findings** |
| --- | --- |
| **Title** | Apache Struts Jakarta Multipart parser vulnerability (CVE-2017-5638) a.k.a. Struts |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | Using Metasploit exploit(multi/http/struts2\_content\_type\_ognl) as identified in the Nessus scan for flag 6 you can connect to 192.168.13.12. Browsing the server files you can find /root/flahisinThisfile.7z. Using meterpreter to download the file and 7-Zip to unzip it gets flag 10. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Apply updates per vendor instructions. |

| **Vulnerability 20** | **Findings** |
| --- | --- |
| **Title** | Drupal core - Highly critical - Remote Code Execution (CVE-2019-6340) |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | Using Metasploit exploit(unix/webapp/drupal\_restws\_unserialize) on 192.168.13.13 (the system with Drupal from flag 5), we are able to exploit a remote code execution vulnerability and run commands on the system as www-data. The username is flag 11. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.13 |
| **Remediation** | Apply updates per vendor instructions. |

| **Vulnerability 21** | **Findings** |
| --- | --- |
| **Title** | Sudo policy bypass (CVE-2019-14287) |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | We connected to system 192.168.13.14 using SSH and a username obtained in OSINT along with flag 1 and guessing at the password. Once connected as a user, we used a sudo exploit to elevate our privileges to root and obtained flag 12. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.14 |
| **Remediation** | Update sudo package. |

| **Vulnerability 22** | **Findings** |
| --- | --- |
| **Title** | Open source exposed data |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Informational |
| **Description** | On the totalrekall GitHub site is the file xampp.users which contains a username and password hash. Which is cracked to reveal flag 1. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Review publicly available documents for sensitive data. |

| **Vulnerability 23** | **Findings** |
| --- | --- |
| **Title** | Sensitive data exposure, anonymous FTP |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Medium** |
| **Description** | An nmap scan shows device 172.22.117.20 has an open port for FTP. Connecting to that system via the FTP protocol as an anonymous user reveals flag 3. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Restrict use of FTP to authorized users, or turn off completely if not needed. |

| **Vulnerability 24** | **Findings** |
| --- | --- |
| **Title** | Seattle Lab Mail 5.5 POP3 Buffer Overflow (CVE-2003-0264) |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | An nmap scan shows device 172.22.117.20 is running the SLMail application version 5.5 which has a vulnerability. Using Metasploit exploit(windows/pop3/seattlelab\_pass) we can gain remote code execution on this system as system and obtain flag 4. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Apply updates per vendor instructions. |

| **Vulnerability 25** | **Findings** |
| --- | --- |
| **Title** | Sensitive data exposure in service |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | Since we are on the device as system, we have full access to the services. Looking at the services you can find one named flag 5. Looking at the details of this service reveals the value needed in the comments field. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Review data listed in service descriptions. Review scheduled tasks for unnecessary tasks. |

| **Vulnerability 26** | **Findings** |
| --- | --- |
| **Title** | LSASS credential dumping - local account |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | Since we are on the device as system, we have access to the LSASS security database and can dump the password hashes for users. The passwords can be cracked by using John the Ripper. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Enable protected mode on LSASS, and limiting credential caching. |

| **Vulnerability 27** | **Findings** |
| --- | --- |
| **Title** | Sensitive data exposure in file |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | Since we are on the device as system, we have full access to the file system. For flag 7 we can find it in the file c:\users\public\documents\flag7.txt. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Limit locations of sensitive data. No realistic way to stop the system account from browsing. |

| **Vulnerability 28** | **Findings** |
| --- | --- |
| **Title** | LSASS credential dumping - domain account |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **High** |
| **Description** | Since we are on the device as system, we have access to the LSASS security database and can dump the password hashes for domain users as well. The passwords can be cracked by using John the Ripper. We were then able to connect to a domain controller using that account |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Enable protected mode on LSASS, and limiting credential caching. |

| **Vulnerability 29** | **Findings** |
| --- | --- |
| **Title** | Microsoft Windows Authenticated User Code Execution (CVE-1999-0504) |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **High** |
| **Description** | By using the domain username and password and Metasploit exploit(windows/local/wmi) we are able to obtain remote code execution on the domain controller from the Windows 10 machine as the identified user (ADMBob). Flag 8 is found as a user on the device. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.10 |
| **Remediation** | Make sure all accounts have a password or are disabled. |

| **Vulnerability 30** | **Findings** |
| --- | --- |
| **Title** | Microsoft Windows Authenticated User Code Execution (CVE-1999-0504) |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | By using the domain username and password and Metasploit exploit(windows/smb/psexec) we are able to obtain remote code execution on the domain controller from the Windows 10 machine as system. Flag 8 is found as a user on the device. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.10 |
| **Remediation** | Make sure all accounts have a password or are disabled. |

| **Vulnerability 31** | **Findings** |
| --- | --- |
| **Title** | LSASS credential dumping - local account |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | Once moved laterally to the domain controller, we use the kiwi module for metasploit to dump the LSASS database and obtain the hash of the administrator password. Since this is a domain controller, the administrator account is a domain administrator and we would have access to all systems on the domain. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.10 |
| **Remediation** | Enable protected mode on LSASS, and limiting credential caching. |