| 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** | InterroBang LLC |
| --- | --- |
| **Contact Name** | Nick Long |
| **Contact Title** | God-Emperor |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 11/06/2024 | Nick Long | Colorful |

# 

## 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.

* Certain fields of the Web application were protected against XSS attacks which slightly slowed our team down
* Some areas had proper basic protections which stopped our team from using SQL injections against the webpage
* Proper Input Validation was discovered on a few input fields

### 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.

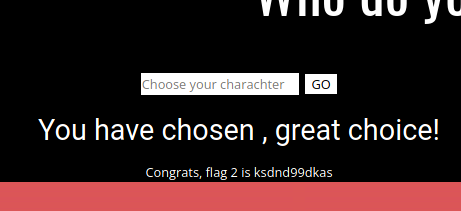
* While some areas were protected, most of the web app was vulnerable to at least one of the following: XSS attacks, Local File Inclusion, or command injections.
* Concerning the servers, several ports were found to have open vulnerabilities with a basic scan of the network
* Speaking of the Network, a simple Whois scan revealed an opening for threat actors.
* Vulnerabilities inside the system revealed openings through both Linux servers and Windows OS, including email(pop3), improperly stored sensitive data, and lateral access across machines.
* Using tools readily available to the community such as metasploit and kiwi, multiple breaches were able to be opened in alarming short amount of time.

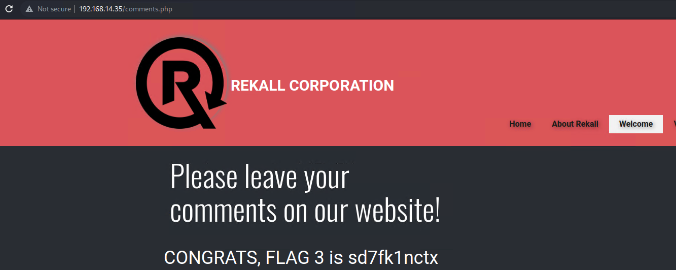
## Executive Summary

### Day One (Web Application)

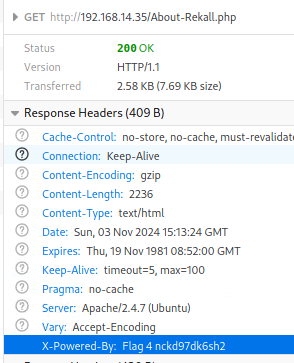
The first three flags were found using various XSS attacks.

## 

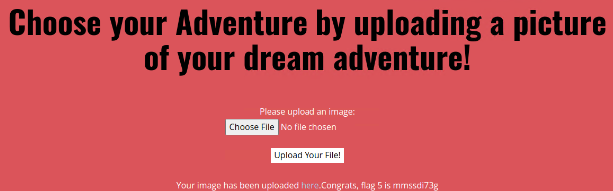


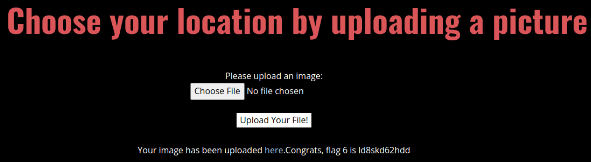


The Fourth flag was found due to Sensitive Data Exposure

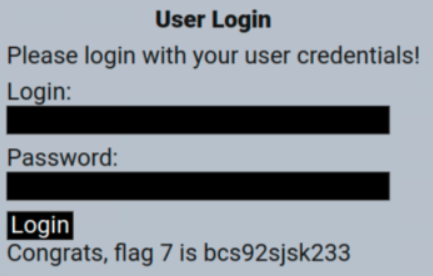


The Fifth and Sixth flags were discovered through Local File Inclusion attacks

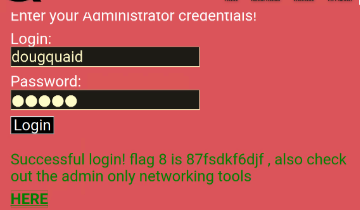




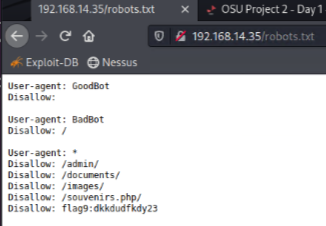
The Seventh Flag was found through directory traversal to gain authentic credentials and gain access to the system.



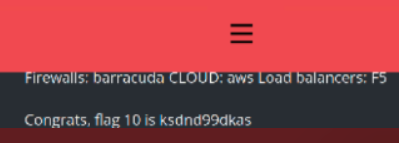
The Eighth Flag was discovered through dumb luck, highlighting on the webpage revealed Sensitive Data exposed for the whole world to see.

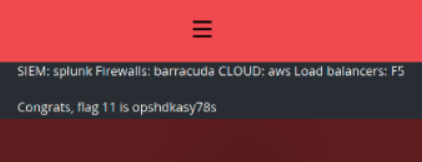


The Ninth Gate was stumbled on by writing the IP address of the site with /robots.txt at the end

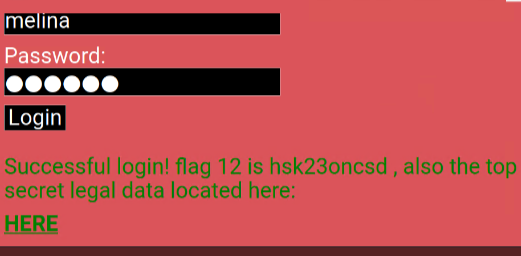


The tenth and eleventh Flags were revealed by using PHP injections

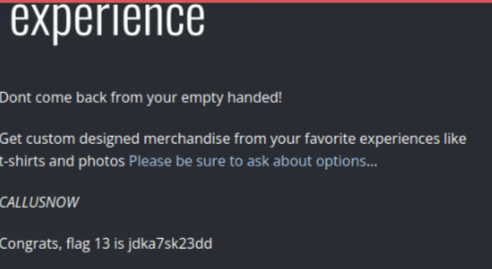




The Twelfth Flag was found by entering the same username & password in the login, earlier discovered during the traversal attack



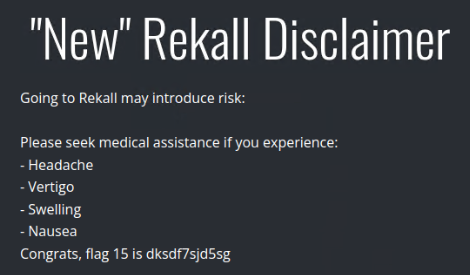
The Thirteenth Flag was found using a PHP injection.



The Fourteenth Flag was linked from the twelfth. That's just poor session management.



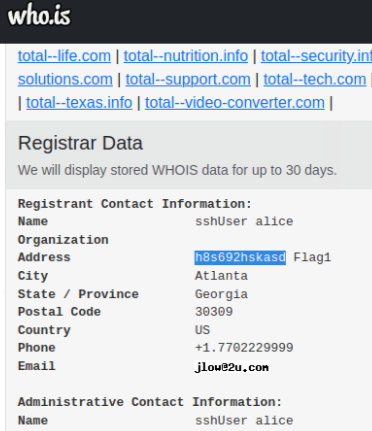
The fifteenth Flag was not discovered in time but a later session uncovered it through directory traversal



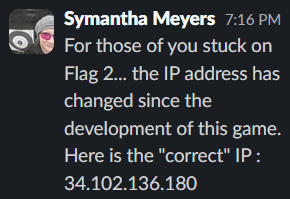
### 

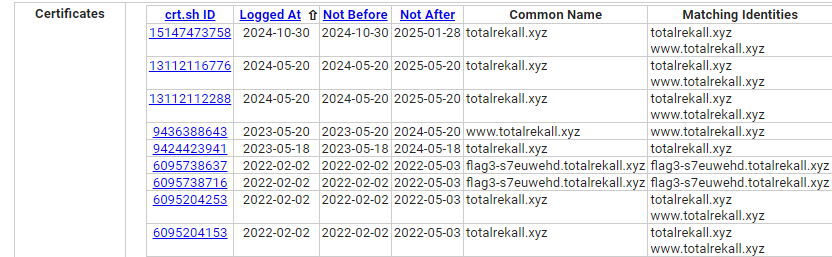
### Day Two (Linux Servers)

The First Flag was found with a simple WHOIS lookup

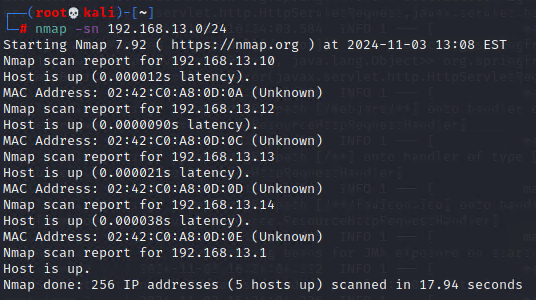


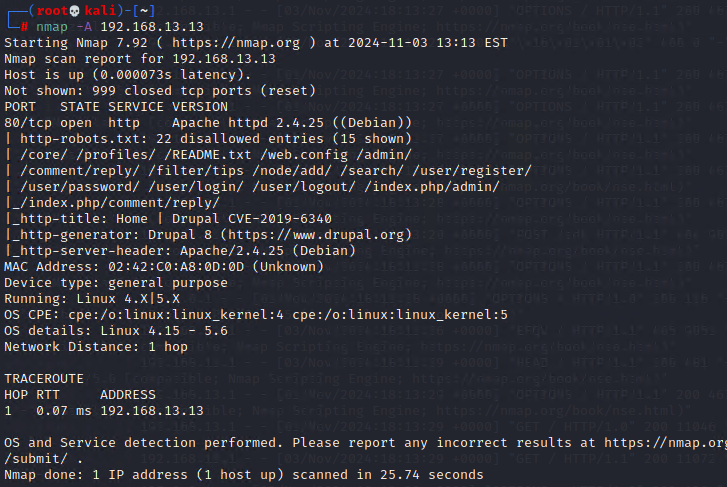
The Second Flag was thankfully given by a TA who discovered the solution didn’t match the answer key as one had been changed without updating the other



The Third Flag was found using a crt.sh search of ssl data

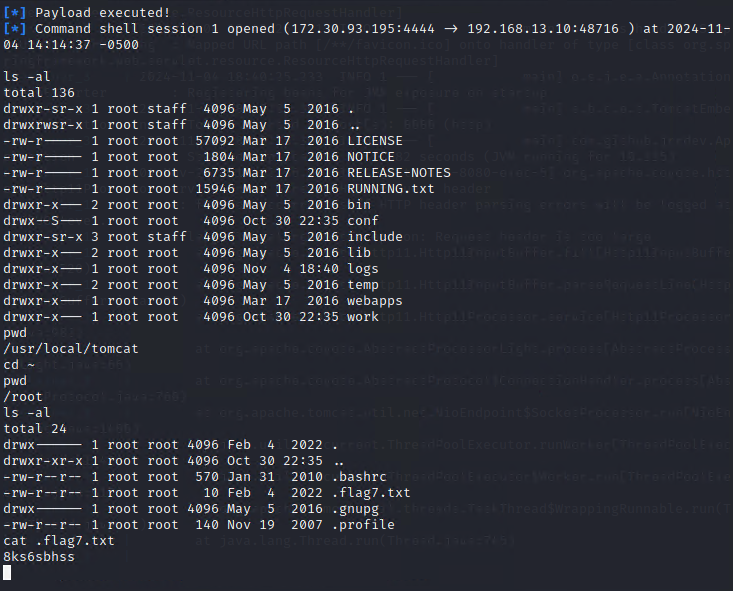
The Fourth Flag was simply counting open hosts from an NMAP scan



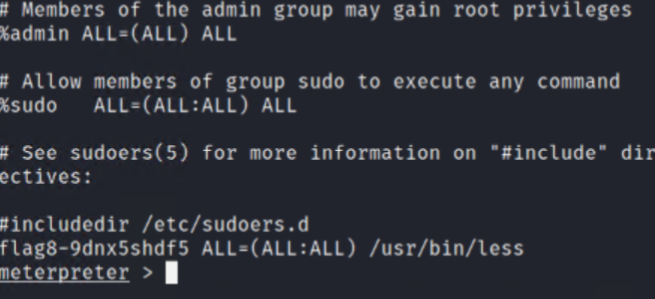
The Fifth Element was the IP address that was running Drupal, found through trial & erroris a 

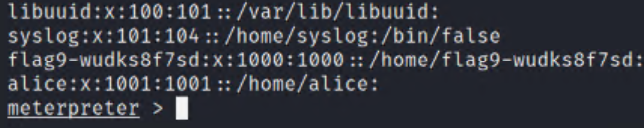
The Sixth Flag discovered using a Shodan.io search of weaknesses



The Seventh Flag was found using Metasploit to exploit a weakness through Apache Tomcat

The Eighth Flag was discovered by using Metasploit to use a shellshock exploit that granted system access

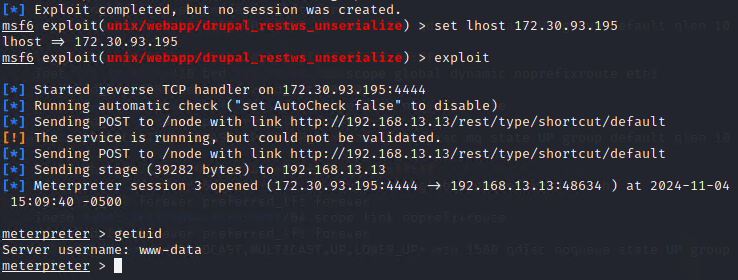


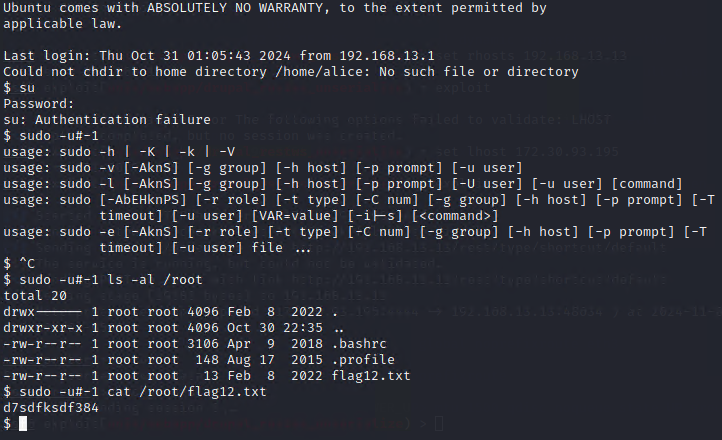
The Ninth Flag was discovered in the same system just a few files over

The Tenth Flag was found exploiting a struts weakness discovered from a nessus scan



The Eleventh Flag was found by exploiting the drupal discovered earlier



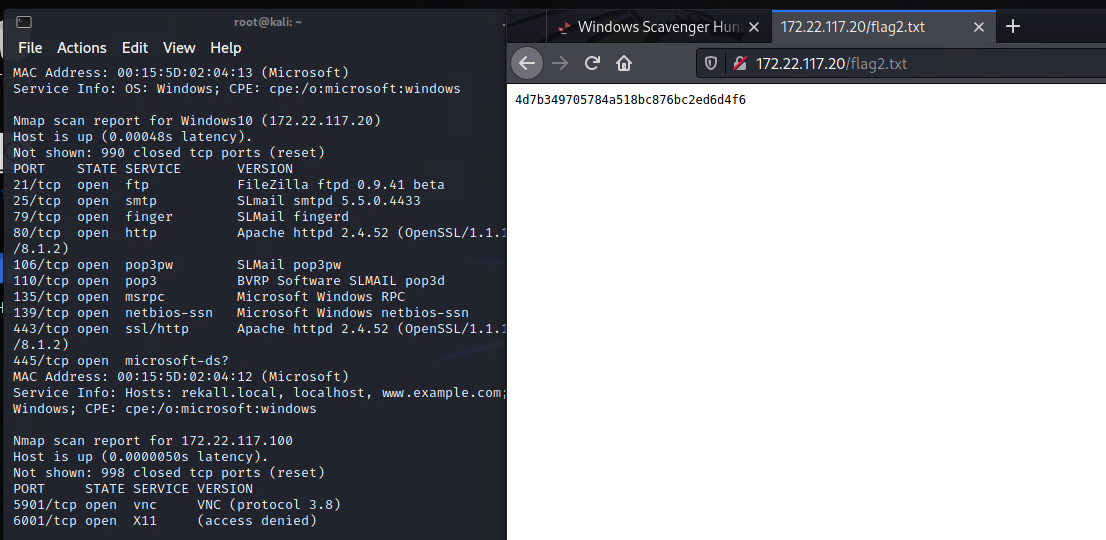
The twelfth Flag was discovered simply by sshing in the final open IP with guessed credentials; user: alice, password :alice

### Day Three (Windows Network)

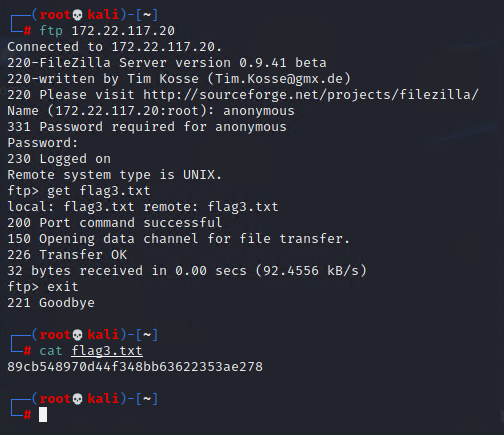
The First Flag was a hash on the Github of totalrekall found in the xampp.users page of the repository. john ripped it wide open

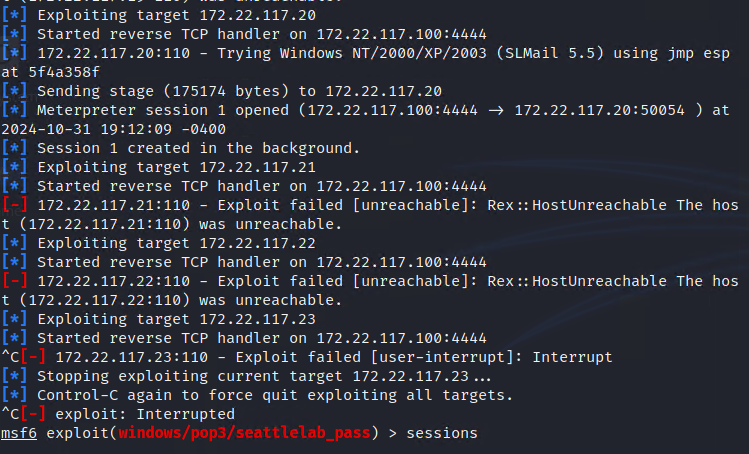


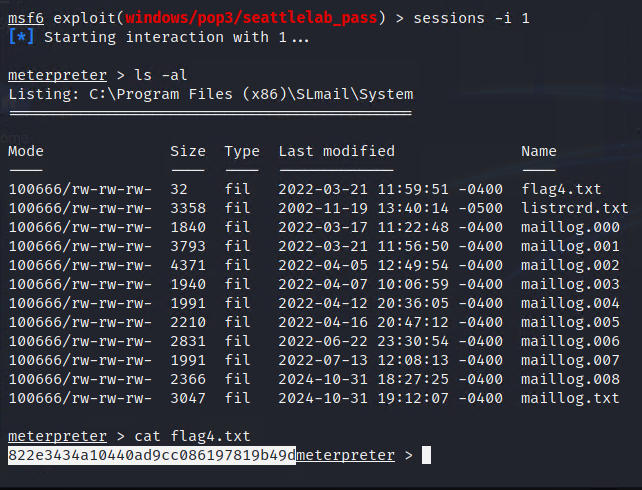
The Second Flag was on a webpage discovered through an NMAP scan

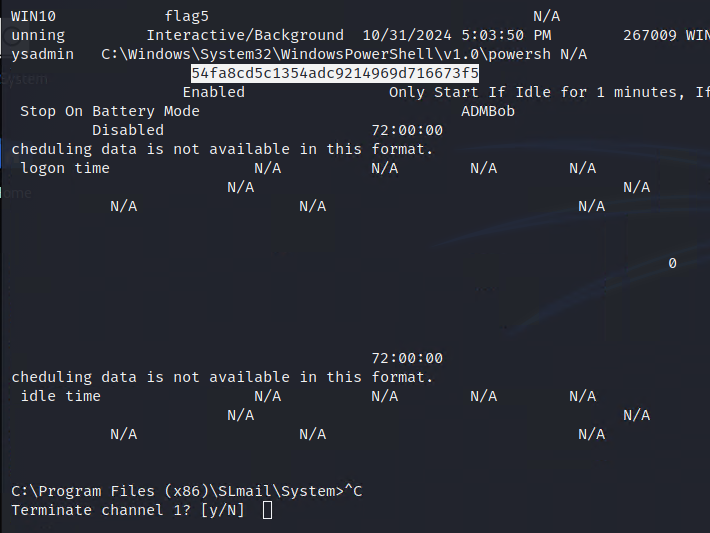


The Third Flag was found by setting an FTP up with the IP discovered with the second Flag

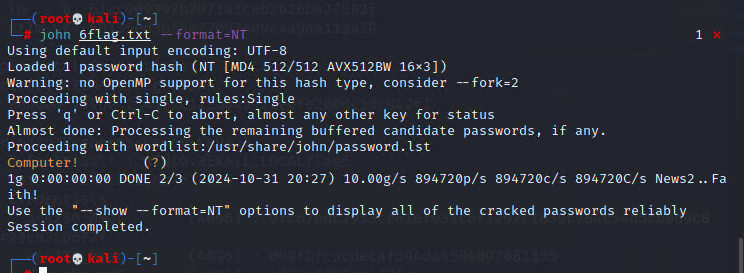


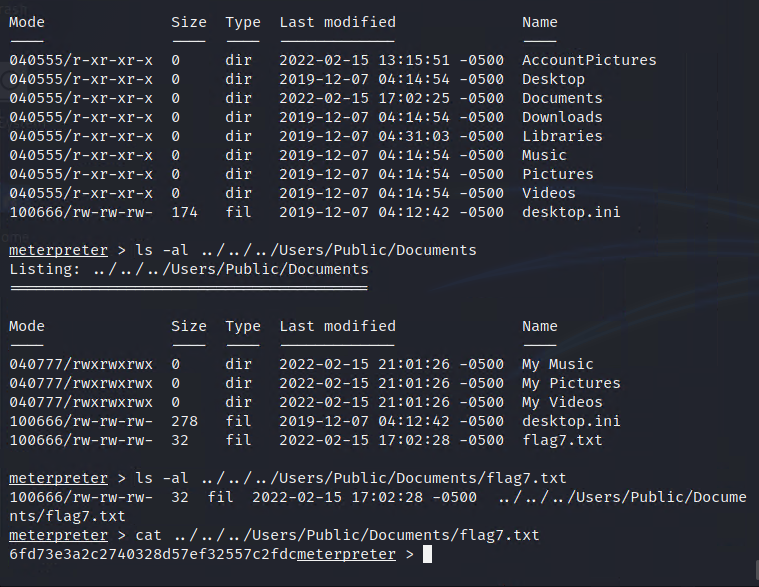
The Fourth Flag was discovered using a Metasploit exploit of the SLMail service

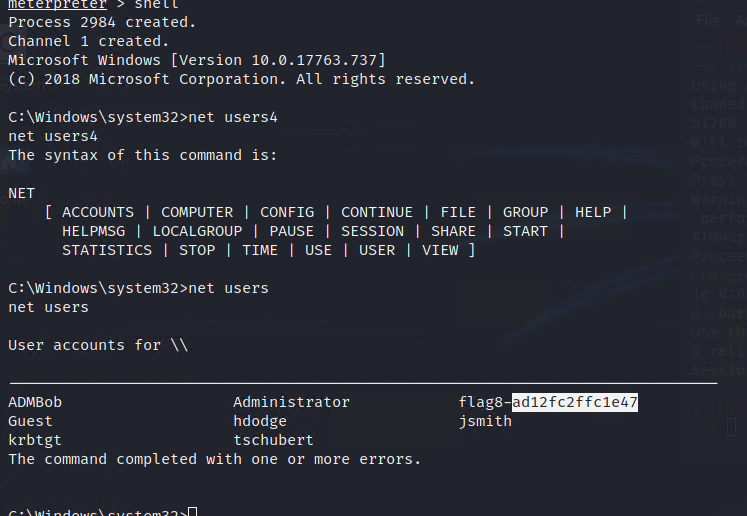
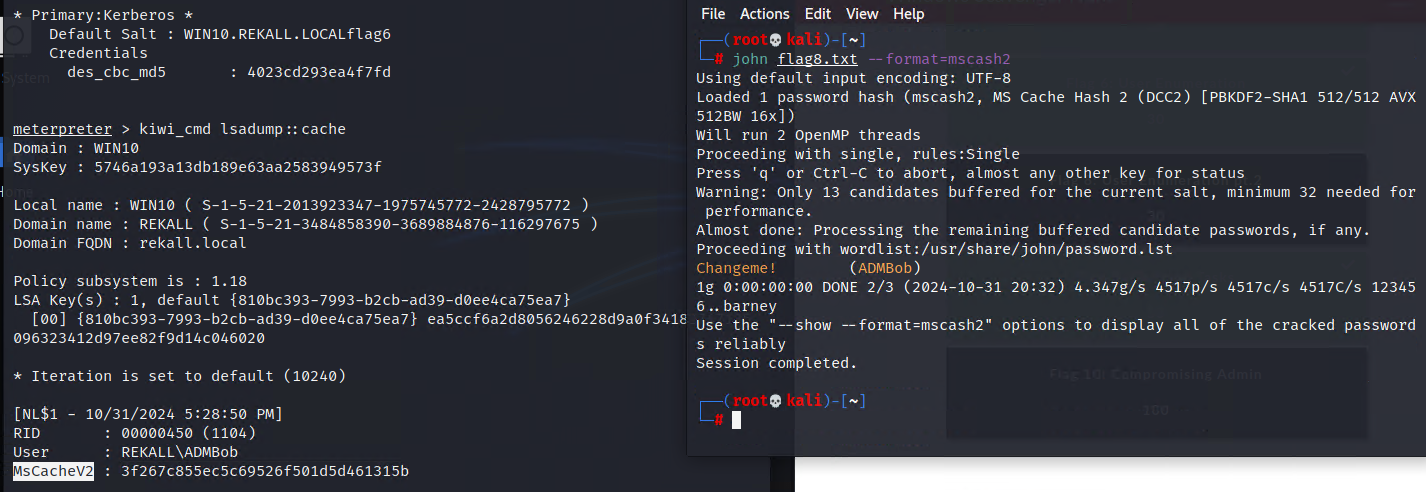


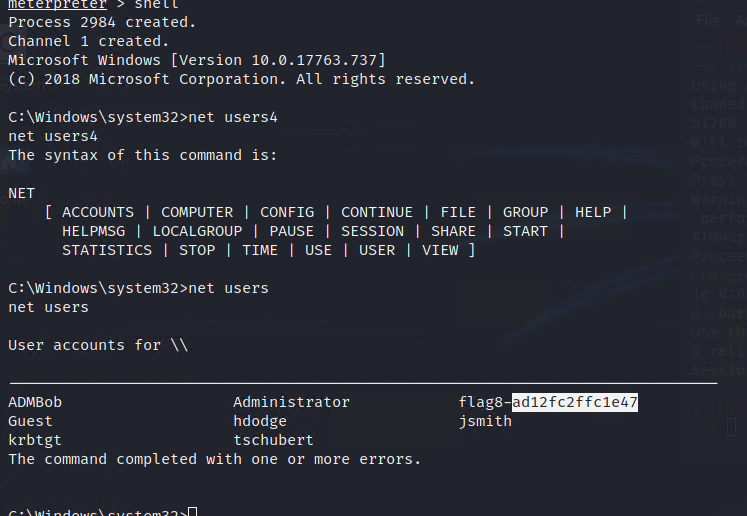
The Fifth Flag posed a personal struggle but was eventually discovered using a shell on the exposed mail system

The Sixth Sense was John Ripped again, this time through a special format

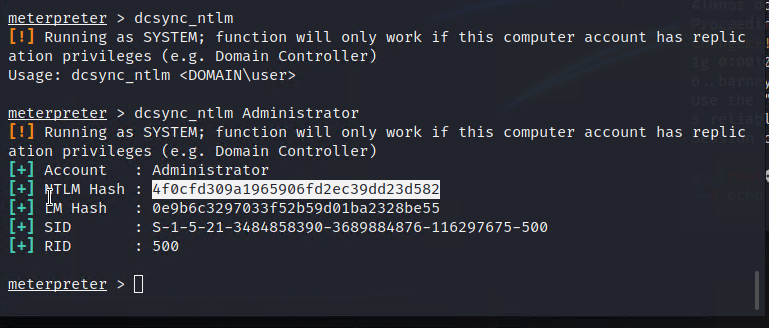


The Seventh Flag was found simply by searching the previously compromised machine

The Eighth Flag was found by employing a kiwi dump to get a cache’d hash which was then employed to crack a different machine

The Ninth Flag was found in just chillin’ in some system’s root directory

The Tenth Flag was exposed after using kiwi to dcsync the Administrator



Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| XSS (reflected or stored) Vulnerabilities on multiple web pages | **High** |
| Command Injection Vulnerabilities | **High** |
| Local File Injection Vulnerabilities | **Critical** |
| Sensitive Data Exposure | **Critical** |
| OpenSource Data Exposure | **Medium** |
| Apache Jakarta Vulnerabilities | **Critical** |
| Shellshock Vulnerability | **Critical** |
| FTP Vulnerability | **Critical** |
| Kiwi Credential Vulnerability | **High** |
| SLMail pop3 Vulnerability | **Critical** |
| Sensitive Data in shared directories | **Medium** |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | 192.168.14.35 192.168.13.10 192.168.13.11 192.168.13.12 192.162.13.13 192.168.13.14 172.22.117.20 |
| Ports | 21 22 80 106 110 |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 6 |
| **High** | 3 |
| **Medium** | 2 |
| **Low** | 0 |

## Vulnerability Findings

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | XSS (reflected or stored) Vulnerabilities on multiple web pages |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | Multiple fields on the web site were vulnerable to malicious script attacks |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | User Input Validation |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | Command Injection Vulnerabilities |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | multiple pieces of information were available to pull off the website by using command manipulation |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Use Strong Input Validation  Update and Patch often |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | Local File Injections Vulnerabilities |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | Critical |
| **Description** | Malicious php script file in both fields on the memory planner page. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Deny file paths from direct appendation  Introduce Dynamic path concatenation |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Windows Os |
| **Risk Rating** | Critical |
| **Description** | There are credentials out in the open on the github page for anyone to use to gain at least FTP access. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Scrub the Git Repository of any sensitive data  hardened access to port 21 |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | OpenSource Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Medium |
| **Description** | We were able to gain our first three footholds into the linux servers on day two by reading the publicly available info on a WHOIS search |
| **Images** |  |
| **Affected Hosts** | 192.168.13.11-14 |
| **Remediation** | Scrub any sensitive data from WHOIS records  Triple Check no sensitive information is publicly available |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Apache Jakarta Vulnerabilities |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | Critical |
| **Description** | Remote Control access was successfully obtained by exploiting the correct metasploit attack function |
| **Images** |  |
| **Affected Hosts** | 192.168.13.11 |
| **Remediation** | Update Apache often |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Shellshock |
| **Type (Web app / Linux OS / WIndows OS)** | Linux |
| **Risk Rating** | Critical |
| **Description** | Gained ability to shell at root level due to exposed shellshock vulnerability discovered through metasploit |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Update Bash often |

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** | FTP Vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Windows |
| **Risk Rating** | Critical |
| **Description** | Discovered FTP was exposed to anonymous connection, exploited that exposure to retrieve file from system |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Disable Anonymous authentication on FTP |

| **Vulnerability 9** | **Findings** |
| --- | --- |
| **Title** | Kiwi Credential Vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Windows Os |
| **Risk Rating** | High |
| **Description** | Through Metasploit, using the kiwi tool, we were able to dump many different user credentials allowing for another password hash cracking and access to yet another account |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Salt Password Hashes  Update Windows |

| **Vulnerability 10** | **Findings** |
| --- | --- |
| **Title** | SLMail pop3 Vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | Critical |
| **Description** | Deployed a metasploit module to exploit a discovered vulnerability in SLMail resulting in successful creation of remote session |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Restrict or close access to port 110  Disable & Remove SLMail, it’s outdated. |

| **Vulnerability 11** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data in shared directories |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | medium |
| **Description** | Discovered sensitive information inside public directories once inside compromised machine. |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Never store sensitive information in a publicly accessible directory! |