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

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

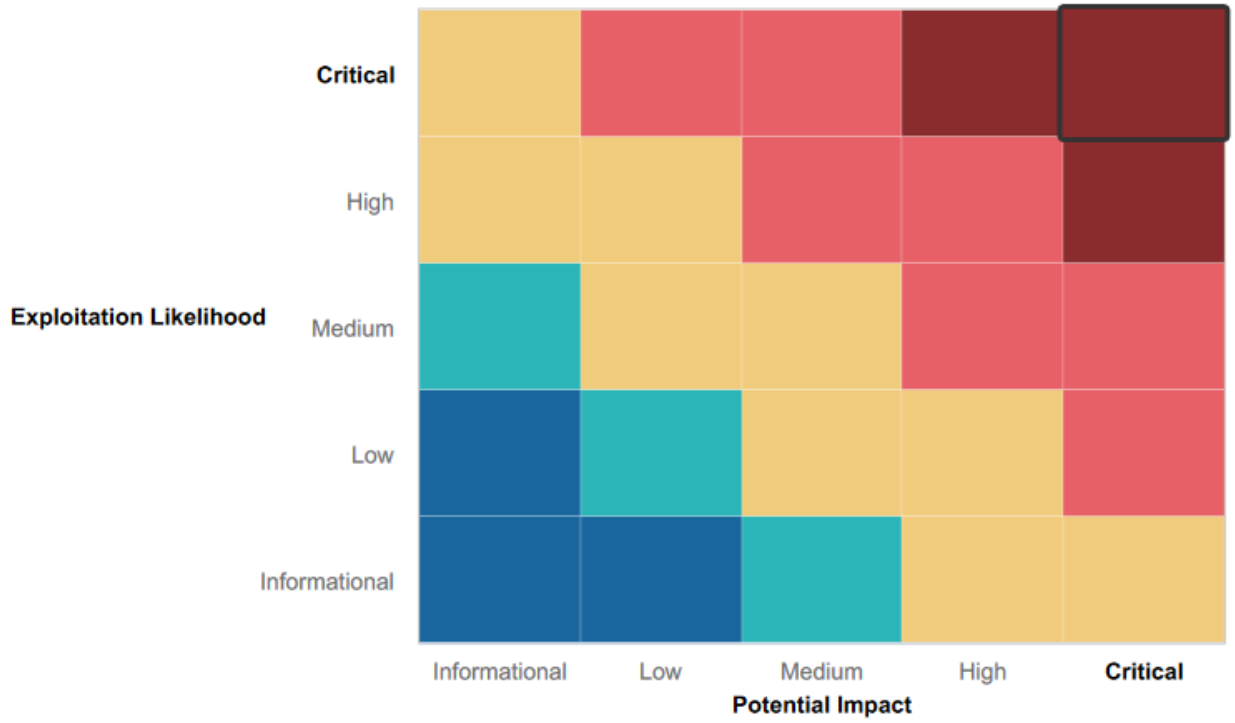
Executive Summary of Findings

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 RJG 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:

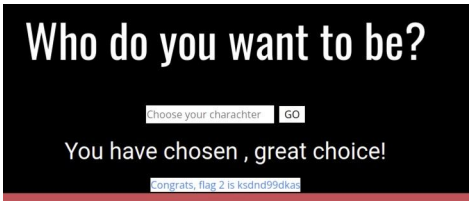
Scan Type	Total
Hosts	34.102.136.180 192.168.14.35 192.168.13.14 192.168.13.13 192.168.13.12 192.168.13.11 192.168.13.10 172.22.117.20 172.22.117.10
Ports	0-1000

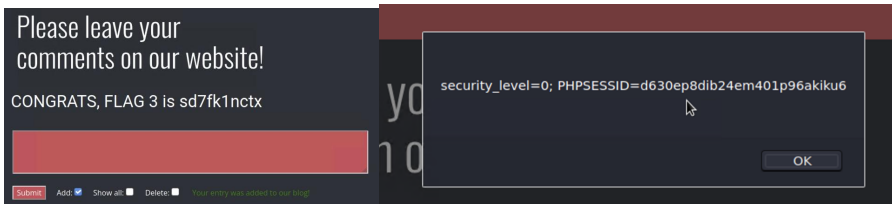
Exploitation Risk	Total
Critical	19
High	10
Medium	3
Low	5

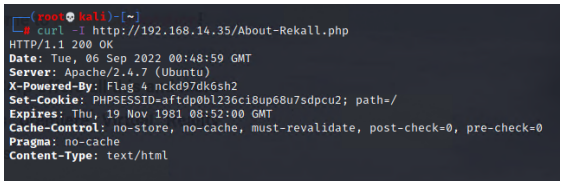
Vulnerability Findings

Day 1: Attacking the Web Application (15 Flags)

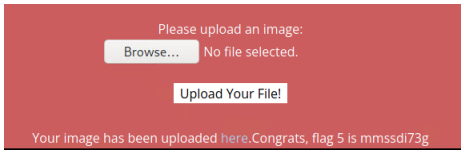
Vulnerability 1	Findings
Title	Flag 1: **f76sdfkg6sjf**
Type	Web Application
Vulnerability	XSS Reflected
Risk Rating	**Critical**
Method/Payload to Exploit	On the Welcome page, RJG was able to enter a reflected XSS payload **<script>alert("flag")</script>** in the "Put Your Name Here" field, making a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

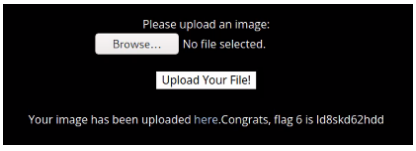
Vulnerability 2	Findings
Title	Flag 2: **ksdnd99dkas**
Type	Web Application
Vulnerability	XSS Reflected
Risk Rating	**Critical**
Method/Payload to Exploit	On the Memory-Planner page, basic input validation removes the word "script" from the payload. However, the validation was avoided by embedding the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

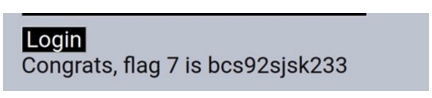
Vulnerability 3	Findings
Title	Flag 3: **sd7fk1nctx**
Type	Web Application
Vulnerability	XSS Stored
Risk Rating	**Critical**
Method/Payload to Exploit	RJG entered a stored XSS payload **<script>alert("flag")</script>** in the comment field and clicked the submit button, revealing sensitive information (security level and cookie information) as well as Flag 3.
Images	
Affected Hosts	http://192.168.14.35/comments.php
Remediation	Input validation, context-sensitive encoding, and a Web Application Firewall (WAF).

Vulnerability 4	Findings
Title	Flag 4: **nckd97dk6sh2**
Type	Web Application
Vulnerability	Sensitive Data Exposure
Risk Rating	**Low**
Method/Payload to Exploit	RJG used **curl -I http://192.168.14.35/About-Rekall.php** on Linux command line which returned an HTTP response header with Flag 4 embedded within it.
Images	
Affected Hosts	http://192.168.14.35/About-Rekall.php
Remediation	Access control rule to prevent or at least alert to usage of the **curl** command from a suspicious IP address.

Vulnerability 5	Findings
Title	Flag 5: **mmssdi73g**
Type	Web Application
Vulnerability	Local File Inclusion

Risk Rating	**Critical**
Method/Payload to Exploit	Conducted a local file inclusion (LFI) exploit by loading a file with **.php** extension to reveal Flag 5 (note that the contents of that file don't matter, only the extension).
Images	
Affected Hosts	http://192.168.14.35/Memory-Planner.php (second field)
Remediation	Restrict file extension to .jpg (or other image file type) and scan uploaded file.

Vulnerability 6	Findings
Title	Flag 6: **ld8skd62hdd**
Type	Web Application
Vulnerability	Local File Inclusion
Risk Rating	**Critical**
Method/Payload to Exploit	Input validation requires **.jpg** in the file name. To bypass, named a malicious script **script.jpg.php** and uploaded it to the "Choose your location" field revealing Flag 6.
Images	
Affected Hosts	http://192.168.14.35/Memory-Planner.php (third field)
Remediation	Restrict file extension to .jpg (or other image file type) and scan uploaded file.

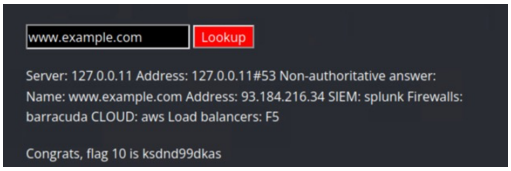
Vulnerability 7	Findings
Title	Flag 7: **bcs92sjsk233**
Type	Web Application
Vulnerability	SQL Injection
Risk Rating	**Critical**
Method/Payload to Exploit	Entered "**test**" as user name and an always-true expression into the password field: **1'or'1'='1** which revealed Flag 7.
Images	
Affected Hosts	http://192.168.14.35/Login.php (first field)

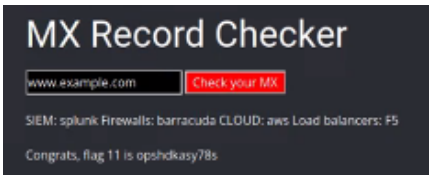
Remediation	Input validation and parameterized queries.

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

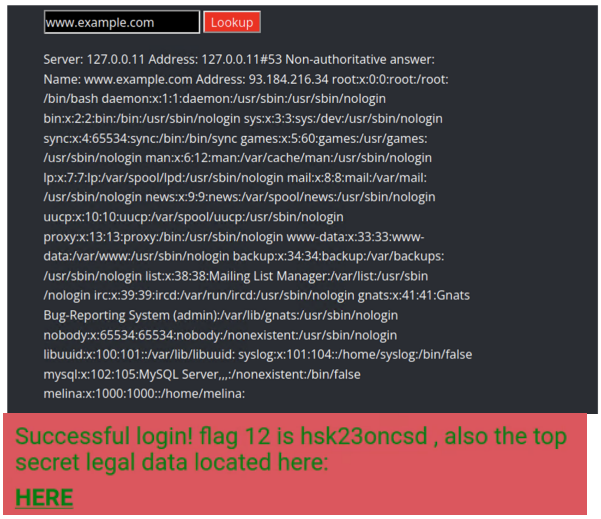
Vulnerability 9	Findings
Title	Flag 9: **dkkdudfkdy23**
Type	Web Application
Vulnerability	Sensitive Data Exposure
Risk Rating	**High**
Method/Payload to Exploit	Simply navigating to **http://192.168.14.35/robots.txt** revealed Flag 9 as well as the existence of a **/souvenirs.php** page which was used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

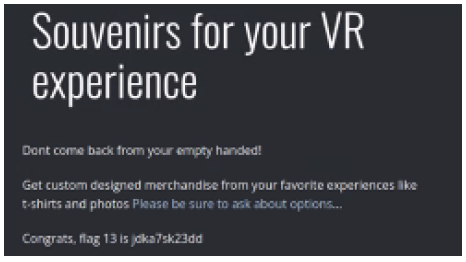
Vulnerability 10	Findings
Title	Flag 10: **ksdnd99dkas**

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

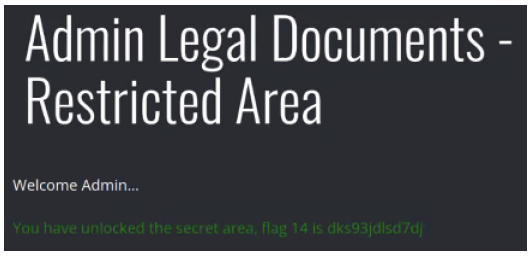
Vulnerability 11	**Findings**
Title	Flag 11: **opshdkasy78s**
Type	Web Application
Vulnerability	Command Injection
Risk Rating	**High**
Method/Payload to Exploit	Input validation strips "**&**" and "**;**" symbols, so the payload was modified to use a pipe: **www.example.com | cat vendors.txt** which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

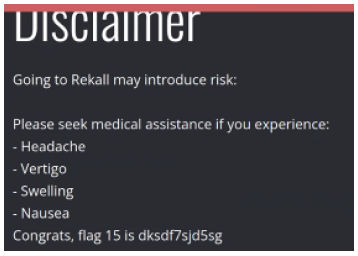
Vulnerability 12	**Findings**
Title	Flag 12: **hsk23oncsd**
Type	Web Application
Vulnerability	Password Guessing
Risk Rating	**Critical**
Method/Payload to Exploit	Entered command injection **www.example.com | cat /etc/passwd** into the second field of **http://192.168.14.35/networking.php** which revealed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

	sensitive data such as the existence of the user **melina**. On **http://192.168.14.35/Login.php**, correctly guessed that the password matched username: **melina** to reveal Flag 12 and the message: **also the top secret legal data located here** with a link to **http://192.168.14.35/admin_legal_data.php?admin=001** used to find Flag 14.
Images	
Affected Hosts	http://192.168.14.35/Login.php (second field)
Remediation	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
Title	Flag 13: **jdka7sk23dd**
Type	Web Application
Vulnerability	PHP Injection
Risk Rating	**Critical**
Method/Payload to Exploit	This hidden webpage was identified in the **robots.txt** file found in Flag 9. The exploit was adding a PHP injection to the URL to find the **/etc/passwd** file: **http://192.168.14.35/souvenirs.php?message=""; system('cat /etc/passwd')**.
Images	
Affected Hosts	http://192.168.14.35/souvenirs.php
Remediation	Input validation to prevent PHP injection. Access control to prevent access of sensitive files by other than system admins.

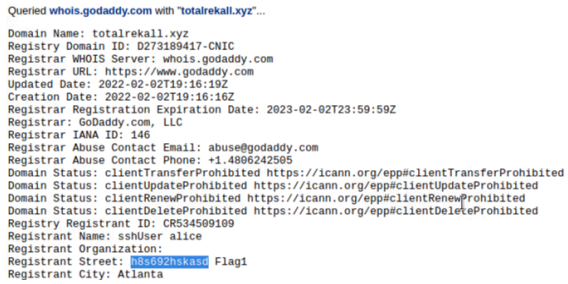
Vulnerability 14	Findings

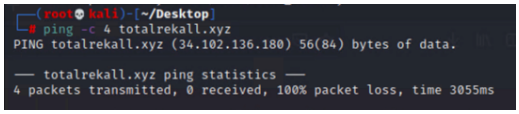
Title	Flag 14: **dks93jdlsd7dj**
Type	Web Application
Vulnerability	Broken Access Control (Session Management)
Risk Rating	**High**
Method/Payload to Exploit	The link to this page was provided when Flag 12 was acquired. Burp Intruder was used to iterate through different admin IDs in the URL. **87**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

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

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

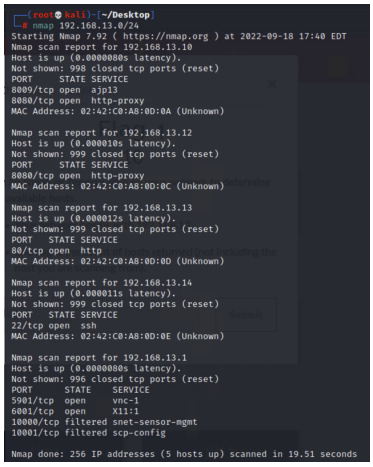
Reconnaissance Phase

Vulnerability 1	Findings
Title	Flag 1: **h8s692hskasd**
Type	Linux OS
Vulnerability	Open Source Exposed Data
Risk Rating	**Low**
Method/Payload to Exploit	Openly available WHOIS data on company's domain **totalrekall.xyz** found using **https://centralops.net/co/DomainDossier.aspx** which revealed Flag 1.
Images	
Affected Hosts	N/A
Remediation	Better management of domain registration information.

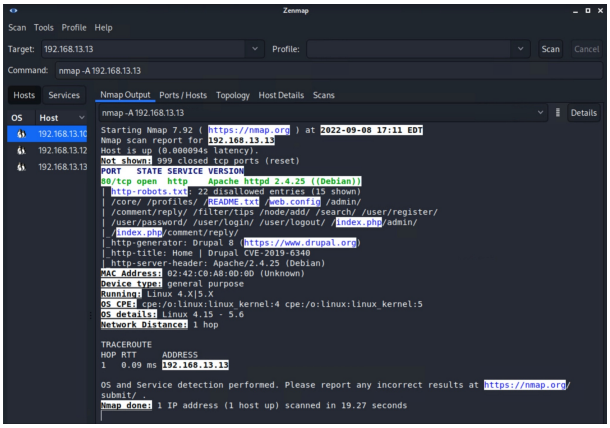
Vulnerability 2	Findings
Title	Flag 2: **34.102.136.180**
Type	Linux OS
Vulnerability	Open Source Exposed Data
Risk Rating	**Low**
Method/Payload to Exploit	Ran **ping -c 4 totalrekall.xyz** to reveal Flag 2 (the IP address of the host machine).
Images	
Affected Hosts	34.102.136.180
Remediation	Disable **ping** response on server (though that has disadvantages).

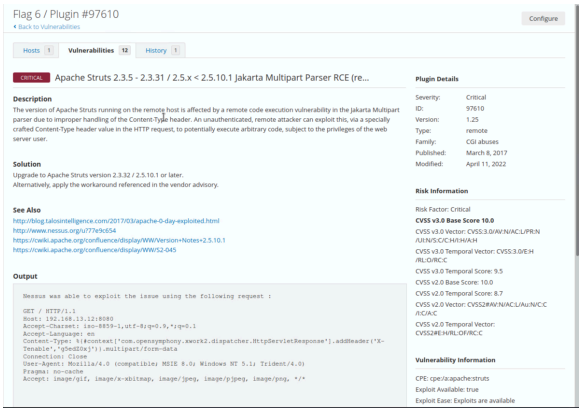
Vulnerability 3	Findings
Title	Flag 3: **s7euwehd**
Type	Linux OS

Vulnerability	Open Source Exposed Data
Risk Rating	**Low**
Method/Payload to Exploit	Searched **crt.sh** for information on **totalrekall.xyz** which revealed Flag 3.
Images	
Affected Hosts	N/A
Remediation	Better management of HTTPS certificate to only have intended information.

Vulnerability 4	Findings
Title	Flag 4: **5**
Type	Linux OS
Vulnerability	Open Source Exposed Data
Risk Rating	**Low**
Method/Payload to Exploit	Ran **nmap 192.168.13.0/24** to determine that there are 5 hosts excluding the host scanning from. Flag 4 was that number of hosts.
Images	
Affected Hosts	192.168.13.0/24
Remediation	Trigger alert if **nmap** is run on company's servers

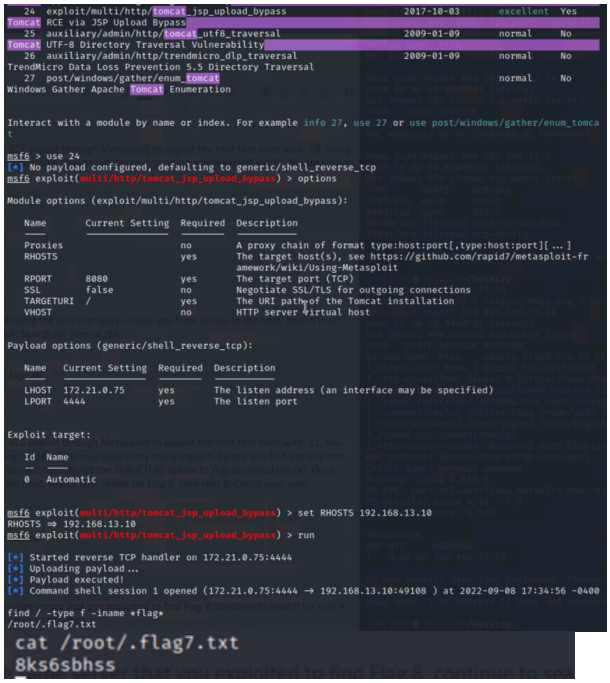
Vulnerability 5	Findings
Title	Flag 5: **192.168.13.13**
Type	Linux OS
Vulnerability	Open Source Exposed Data

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

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

Exploitation Phase

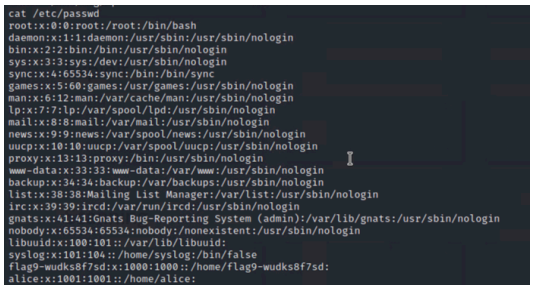
Vulnerability 7	**Findings**

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

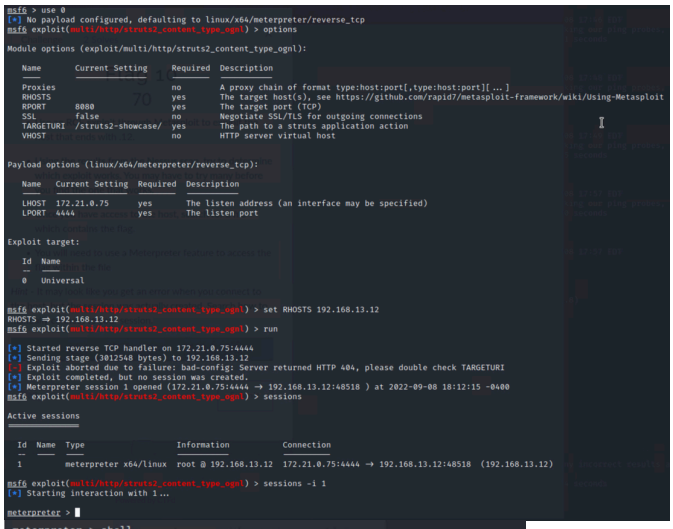
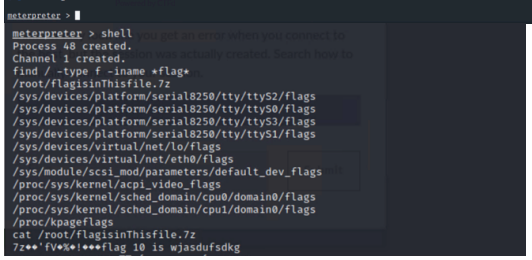
Vulnerability 8	Findings
Title	Flag 8: **9dnx5shdf5**
Type	Linux OS
Vulnerability	Shellshock (Remote Code Execution)
Risk Rating	**Critical**
Method/Payload to Exploit	Told to look for a “shocking” RCE exploit, ran MSFconsole and searched for exploits that have Shellshock. Ran **exploit/multi/http/apache_mod_cgi_bash_env_exec**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

Images	
Affected Hosts	192.168.13.11
Remediation	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
Title	Flag 9: **wudks8f7sd**
Type	Linux OS
Vulnerability	Shellshock (Remote Code Execution)
Risk Rating	**Critical**
Method/Payload to Exploit	Still in shell achieved through exploiting Shellshock vulnerability, ran **cat /etc/passwd** to reveal a suspicious user named **flag9-wudks8f7sd** which is Flag 9.

Images	
Affected Hosts	192.168.13.11
Remediation	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.

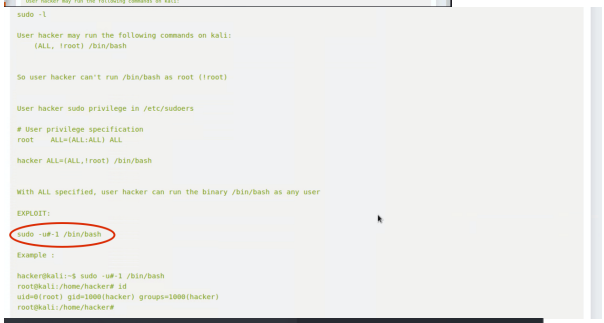
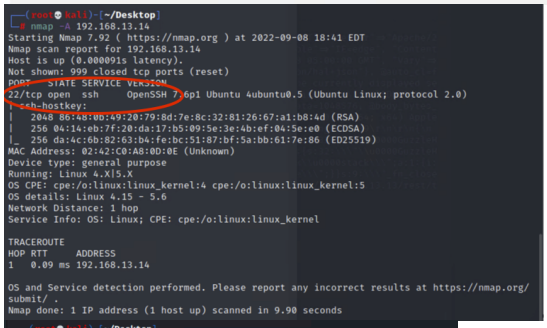
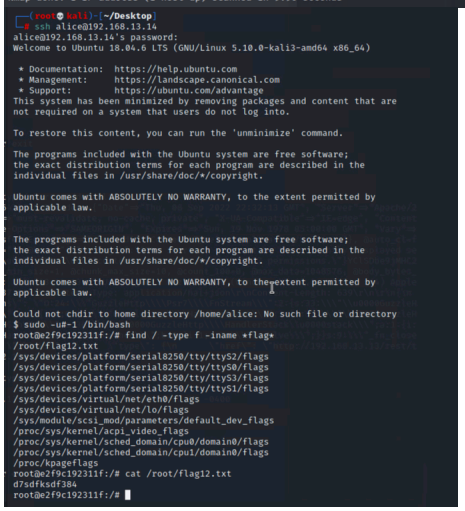
Post-Exploitation Phase

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

Remediation	Update to latest Apache Struts. Access control on important directories and files.

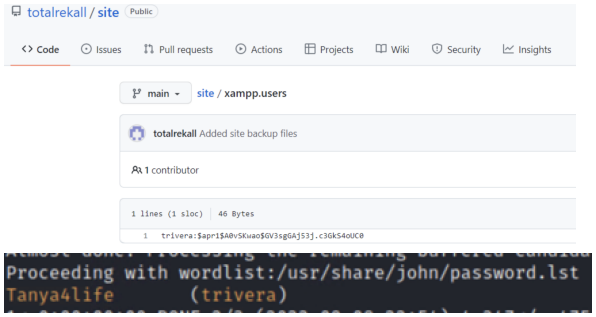
Vulnerability 11	Findings
Title	Flag 11: **www-data**
Type	Linux OS
Vulnerability	CVE-2019-6340 (Drupal)
Risk Rating	**High**
Method/Payload to Exploit	Nmap scan revealed Drupal vulnerability. In MSFconsole, used the exploit **unix/webapp/drupal_restws_unserialize**. Ran **getuid** in Meterpreter shell to reveal Flag 11.
Images	
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**

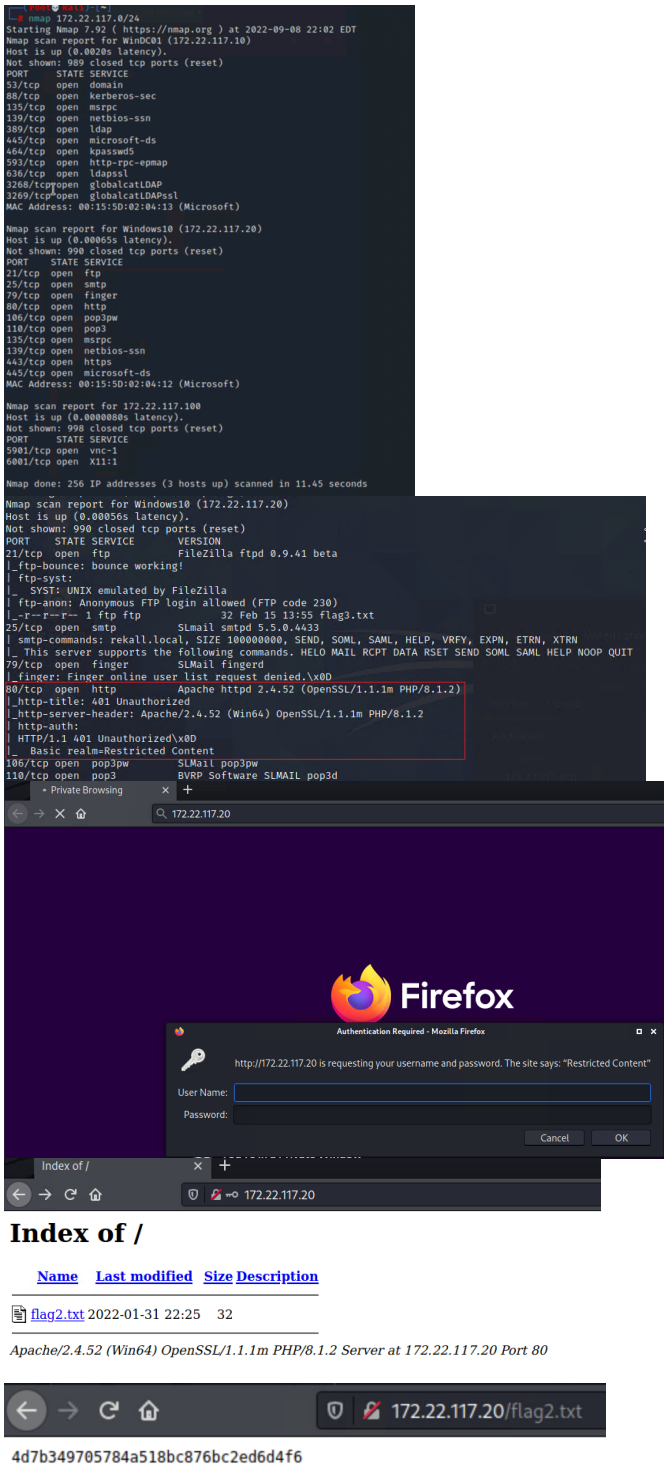
Method/Payload to Exploit	Using info Flag 1 revealed "**Registrant name: sshUser alice**" and nmap scan that showed open **port 22**, ran **ssh alice@192.168.13.14** and guessed the password was **alice**. Ran **sudo -u#-1 bin/bash** to bypass sudo restrictions and escalate privileges, then ran **find / -type f -iname *flag*** to find and **cat /root/flag12.txt**.
Images	   
Affected Hosts	192.168.13.14
Remediation	Stronger password for user **alice**. Disable SSH privilege for all but authorized users. Enable public-private key authentication.

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

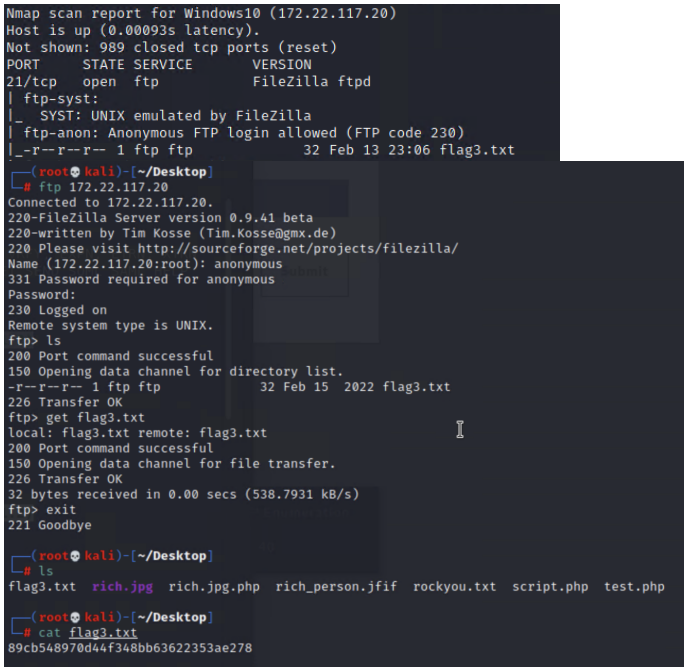
Reconnaissance Phase

Vulnerability 1	Findings
Title	Flag 1 - OSINT: **Tanya4life**
Type	Windows OS
Vulnerability	Sensitive Data Exposure
Risk Rating	**Medium**
Method/Payload to Exploit	Searching GitHub for "**totalrekall**" lead to the **xampp.users** page, which contained the credentials **trivera:$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0**. Ran **echo '$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0' > hash.txt john hash.txt** to crack the hash, revealing password is "**Tanya4life"** which is Flag 1.
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: **4d7b349705784a518bc876bc2ed6d4f6**
Type	Windows OS
Vulnerability	Sensitive Data Exposure
Risk Rating	**Medium**
Method/Payload to Exploit	Told that the Windows network has a subnet of 172.22.117.0/24, ran **nmap 172.22.117.0/24** which revealed two machines (**Win10: 172.22.117.20** and **WinDC01 @ 172.22.117.10**). A port scan of the Win10 machine revealed several open ports, including **port 80** which is **HTTP**. Navigating to 172.22.117.20 in a browser prompted for login credentials. Used **trivera** and **Tanya4life** to gain access and then found the file **flag2.txt** which contained Flag 2.

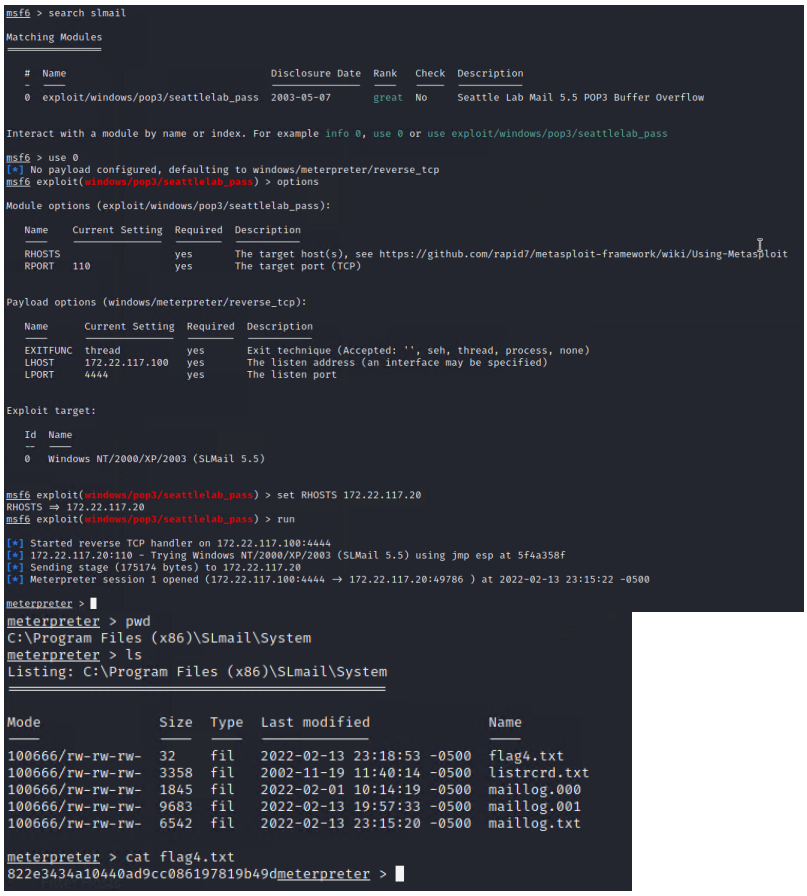
Images	
Affected Hosts	172.22.117.20
Remediation	Access control on sensitive directories and files.

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

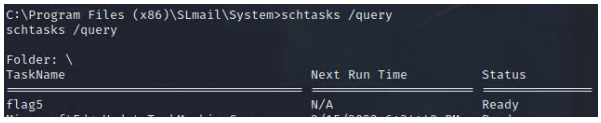
Vulnerability	Anonymous FTP Login allowed
Risk Rating	**High**
Method/Payload to Exploit	An aggressive port scan of the Win10 machine **nmap -A 172.22.117.20** also revealed open **port 21** and the vulnerability to anonymous FTP login. Ran **ftp 172.22.117.20** with user **anonymous** and password **anonymous**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

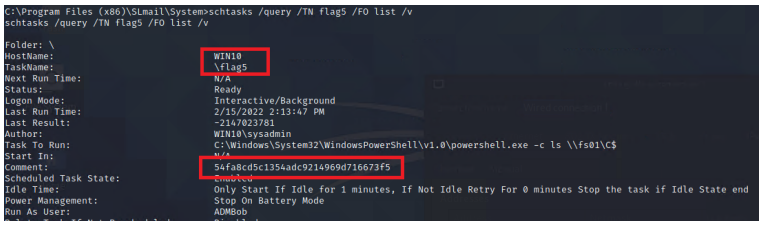
Exploitation Phase

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

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

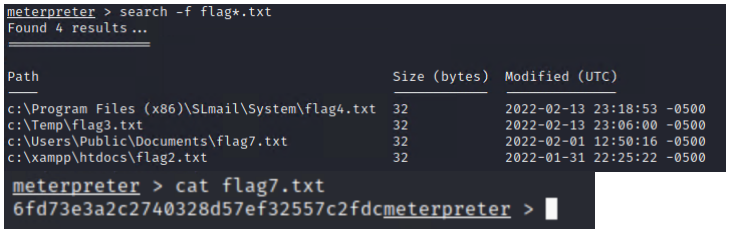
Post-Exploitation Phase

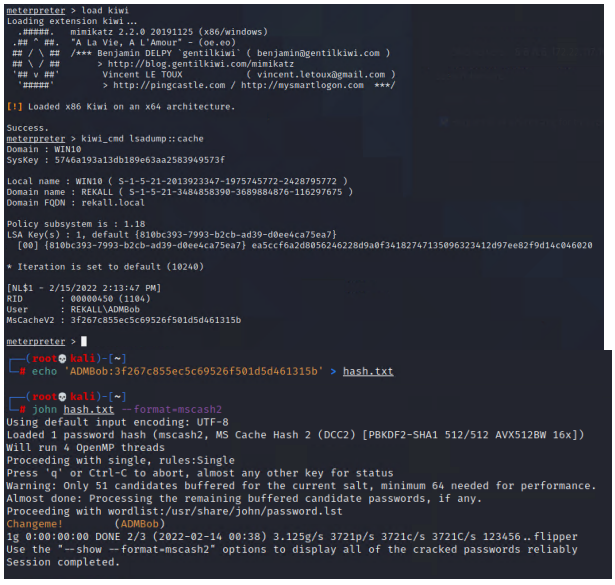
Vulnerability 5	Findings
Title	Flag 5 - Common Tasks: **54fa8cd5c1354adc9214969d716673f5**
Type	Windows OS
Vulnerability	Scheduling Task Abuse
Risk Rating	**Critical**
Method/Payload to Exploit	Within the same Meterpreter session from Flag 4, ran **shell** to get shell on Win10 machine. Hint given about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

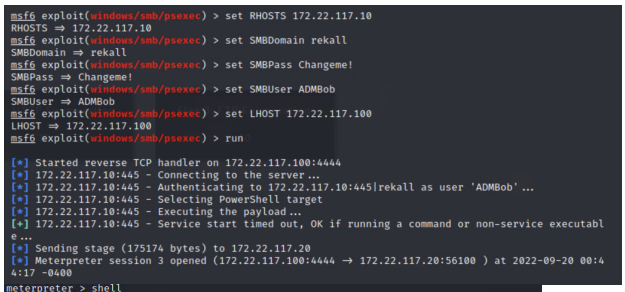
	
Affected Hosts	172.22.117.20
Remediation	Limit privileges of user accounts and remediate Privilege Escalation vectors so only authorized administrators can create scheduled tasks on remote systems.

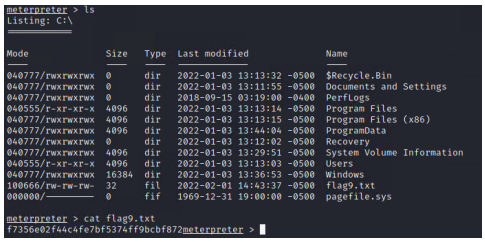
Vulnerability 6	**Findings**
Title	Flag 6 - User Enumeration: **Computer!**
Type	Windows OS
Vulnerability	Credential Dumping
Risk Rating	**High**
Method/Payload to Exploit	While still in the Meterpreter shell as the SYSTEM user, ran **load kiwi** then **lsa_dump_sam** which revealed a user named flag6 with NTLM password that was cracked with **john** 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

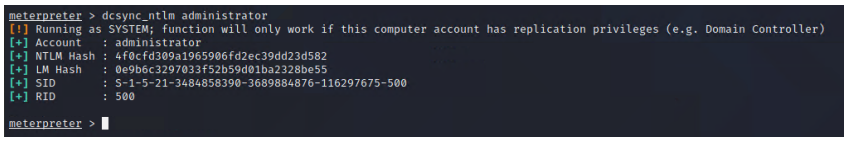
Risk Rating	High
Method/Payload to Exploit	While still in Meterpreter shell, ran **search -f flag*.txt** which revealed the file **flag7.txt** in the **C:\Users\Public\Documents** folder. Navigated to file and ran **cat**.
Images	
Affected Hosts	117.22.117.20
Remediation	Access controls on sensitive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

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

	 caca
Affected Hosts	172.22.117.20 and 172.22.117.10
Remediation	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 **SYSTEM shell**, ran **cd ** to move to **root**, ran **ls**, found and **cat flag9.txt** to reveal Flag 9.
Images	
Affected Hosts	117.22.117.10
Remediation	Proper privilege account management

Vulnerability 10	Findings
Title	Flag 10: **4f0cfd309a1965906fd2ec39dd23d582**
Type	Windows OS
Vulnerability	Broken Access Control
Risk Rating	**Critical**

Method/Payload to Exploit	Still in Meterpreter, ran **load kiwi**, then ran **dcsync_ntlm administrator** to reveal the NTLM password for the user **Administrator** which is Flag 10.
Images	
Affected Hosts	172.22.117.10
Remediation	Patching endpoints, implementing access controls, and continuous monitoring of network traffic.