| Cybersecurity |
| --- |
| Penetration Test Report |

Rekall Corporation

Penetration Test Report

# 

## Confidentiality Statement

This document contains confidential and privileged information from Rekall Inc. (henceforth known as Rekall). The information contained in this document is confidential and may constitute inside or non-public information under international, federal, or state laws. Unauthorized forwarding, printing, copying, distribution, or use of such information is strictly prohibited and may be unlawful. If you are not the intended recipient, be aware that any disclosure, copying, or distribution of this document or its parts is prohibited.

Table of Contents

[Confidentiality Statement 2](#_30j0zll)

[Contact Information 4](#_1fob9te)

[Document History 4](#_3znysh7)

[Introduction 5](#_2et92p0)

[Assessment Objective 5](#_3dy6vkm)

[Penetration Testing Methodology 6](#_2s8eyo1)

[Reconnaissance 6](#_17dp8vu)

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

[Vulnerability Exploitation 6](#_26in1rg)

[Reporting 6](#_lnxbz9)

[Scope 7](#_35nkun2)

[Executive Summary of Findings 8](#_44sinio)

[Grading Methodology 8](#_z337ya)

[Summary of Strengths 9](#_3j2qqm3)

[Summary of Weaknesses 9](#_1y810tw)

[Executive Summary Narrative](#_4i7ojhp) 11

[Summary Vulnerability Overview 1](#_2xcytpi)8

Vulnerability Findings22

# 

## Contact Information

| **Company Name** | Safety Zone LLC (SZ) |
| --- | --- |
| **Contact Name** | Kevin Arnold |
| **Contact Title** | Penetration Tester |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | Feb. 15, 2025 | Kevin Arnold | First Draft - Day 1 |
| 002 | Feb. 18, 2025 | Kevin Arnold | Day 2 |
| 003 | Feb 20, 2025 | Kevin Arnold | Day 3 |
| 004 | Feb 21, 2025 | Kevin Arnold | Completed all remaining tests |
| 005 | Feb 23, 2025 | Kevin Arnold | Final Draft |

# 

## Introduction

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

For the testing, we focused on the following:

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

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

### Assessment Objective

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

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

Rekall has outlined the following objectives:

Table 1: Defined Objectives

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

# 

## Penetration Testing Methodology

### Reconnaissance

### 

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

### Identification of Vulnerabilities and Services

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

### Vulnerability Exploitation

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

### Reporting

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

# 

## Scope

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

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

# 

# 

## Executive Summary of Findings

## 

### Grading Methodology

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

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

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

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

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

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

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

Chart

Description automatically generated with medium confidence

### 

### Summary of Strengths

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

* Rekall has demonstrated a proactive concern towards cybersecurity by allowing the penetration test. This is a great achievement in itself.
* Network (domain) accounts and passwords are locked down from intruders by using standard access controls.
* The network structure itself is a sound design.
* Users are not granted permissions to sensitive files, demonstrating principles of least privilege.
* No account was working in our exploit to connect to the WinDC01 at 172.22.117.10 using kiwi\_cmd lsadump::sam
* Most sensitive information was not freely available on the website. No financial database or product environments were detected with our scans.
* Not all accounts contained weak passwords. The account for ADBob was a reasonable password since it was 9 characters long and contained an uppercase and a special character.
* The DC operated in a suitable and secure manner until it was exposed by an exploit.

### Summary of Weaknesses

SZ successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

* Some attention is required on the forward-facing websites to shore up access to files, server information and IP addresses made public.
* Some user training and a focus on security awareness is needed for the staff to not save passwords on the network.
* The IT admin staff should check open ports and needed security patches and OS updates on all servers and workstations.
* The IT admin staff should specifically examine the exploits detailed below - that were possible during the test - so that they cannot happen by an actual attacker.
* Specifically, we were able to exploit the SLMail service. This was done by using the Meterpreter exploit: windows/pop3/seattlelab\_pass on host 172.22.117.20. This is of importance because it allows remote access to that host. Be sure to upgrade your mail server to a modern and robust service such as Microsoft 365 that would thwart that attack.
* We later moved laterally from the Windows 10 machine to the WINDC01 using the windows/smb/psexec exploit and credentials found with kiwi.
* Moreover, special attention should be paid to your Apache Struts and RESTful Web Services to bring these up to vendor specifications. We were able to use an RCE exploit on a host 192.168.13.13 using the results of the nmap scan done earlier. This indicated a Drupal RESTful Web Services unserialize() RCE vulnerability (CVE-2019-6340). We recommend disabling RESTful Web Services if not required and to configure Web Application Firewall to block exploit attempts.
* We were able to exploit a host 192.168.13.14 with some information found during our OSINT reconnaissance. Then use privilege escalation to gain more access using SSH via the alice account found. We recommend checking the sudoers file for misconfigurations, restrict SSH access to specific users and monitor the sudo logs for suspicious activity.
* We found an exploit working using msfconsole on host 192.168.13.11 and created a remote shell. With remote access secured, we were able to search on the host while investigating the sudoers file. Apply all vendor patches to prevent remote code execution. Monitor for unusual activity on the network and in logs.
* We ran an additional exploit for lateral movement. Available accounts were dumped with kiwi using the command kiwi\_cmd lsadump::cache We successfully used these credentials in the exploit: windows/smb/psexec. We examined the users on the DC and found even more accounts of interest. We recommend disabling SMB v1 and restricting SMB shares using a firewall. Also, block unnecessary SMB access between workstations and only allow authenticated access.
* We were able to use a PHP code in the form of a script to exploit a web page and run a command on souvenirs.php by changing the message in the URL to: message=””; system(“ls”) This command, ls lists some names of secret files. We recommend to disable dangerous PHP functions which are set in php.ini

## Executive Summary

Great care has been taken in this penetration testing to identify critical vulnerabilities in the Rekall network. It should be noted that many companies are facing the exact issues identified herein and that by conducting this test you are well on your way to securing these and preventing an actual cyber incident. This will safeguard your data and your company’s reputation and reliability perceived by your business partners.

The reconnaissance phase using OSINT tactics revealed clues allowing further exploits. Special care should be taken for stricter controls on the user input fields on all the web pages described in the report. The ability to exploit the expected form input allowed for even further exploits to occur.

The same OSINT tactics allowed discovery of key company information about domain information, IP addresses and SSL certificates. Likewise to the web application findings, by hardening these exploitable findings, you can limit your attack surface. Then further exploits will be much less likely when attackers do not have this information.

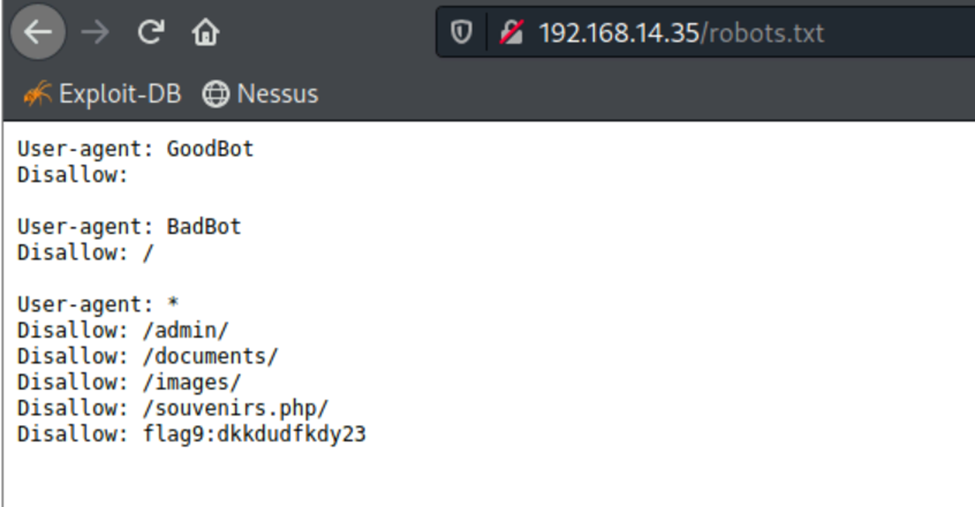
We recommend a robust staff training exercise that reinforces the importance of using complex passwords, not saving their usernames and passwords on the web source code or in files on the network.

So with this in mind, please examine the details in their specifics so that mitigation can proceed.

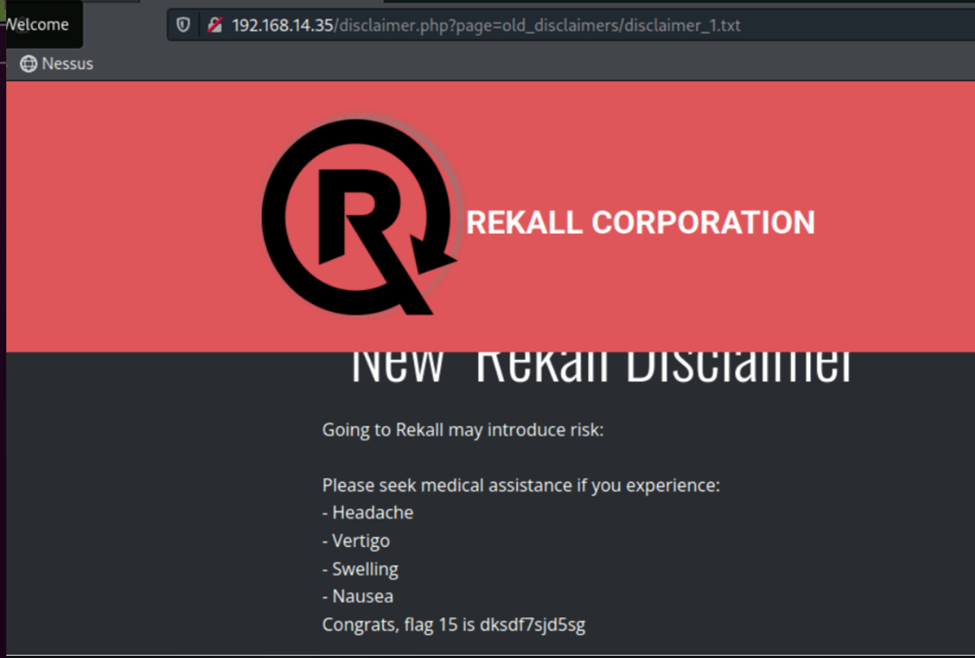
**Day 1:** We began our penetration test in earnest, according to the scope that is outlined. We used Cross-Site Scripting Payloads (XSS) on a number of fields within the Rekall webpages. Then the curl tool to reveal sensitive data within the header of a webpage.



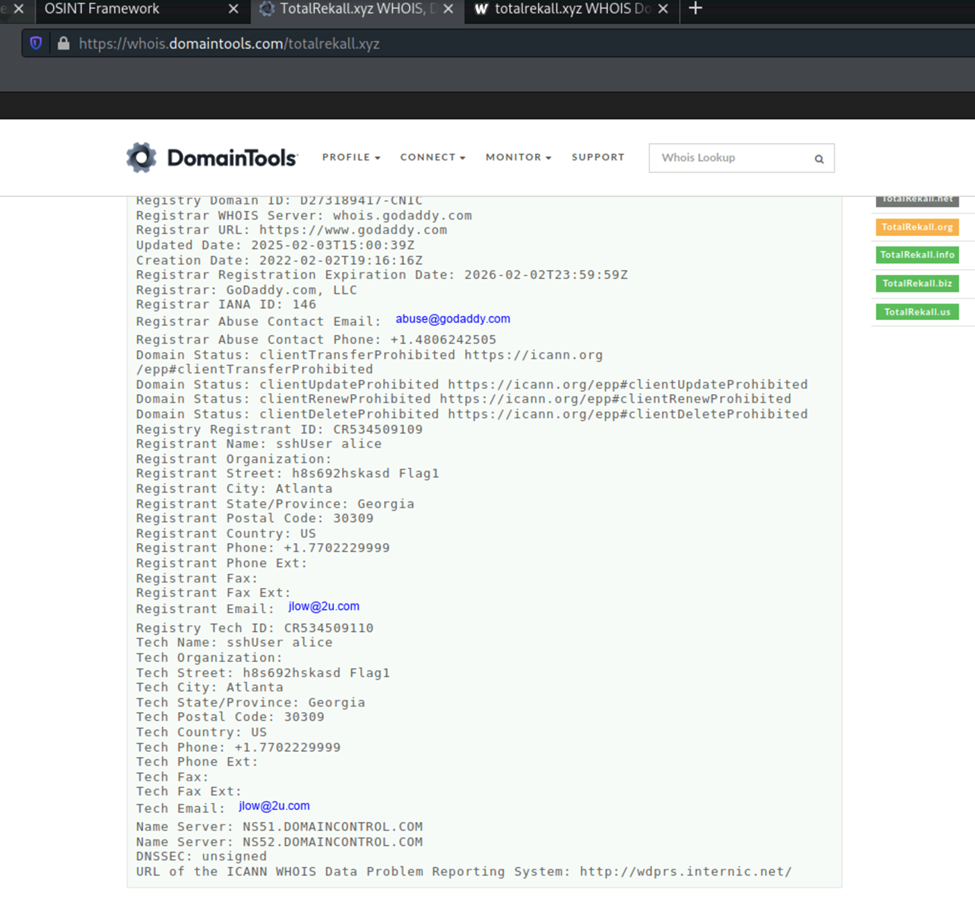
We also conducted a Local File Inclusion (LFI) exploit on a webpage to upload an unintended file. Further testing the web pages, we were able to conduct an SQL Injection attack. A login account was found in the HTML source code and we found a list of vendors for various services in a secret vendors.txt file. Sensitive data was also found in the robots.txt file.



Next, some command injections were performed on the Rekall Admin Networking Tools website. This gave even more information needed for successive tests, using the melina account to log into the secret legal area of the website. To round out Day 1, we successfully performed a PHP injection attack, a session management attack and then directory traversal to an older version of the Rekall Disclaimer page.



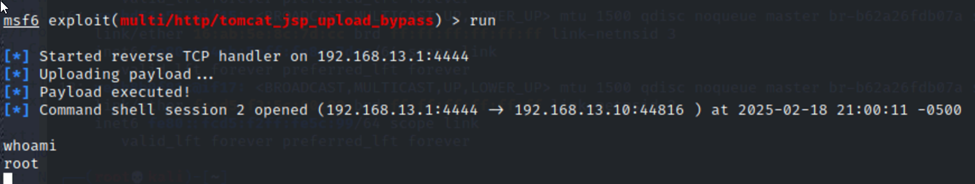
**Day 2**: The next day, SZ began checking for vulnerabilities on totalrekall.xyz website. Freely available tools were used throughout the tests. First, DomainTools Whois was used to determine information about the Rekall domain. Further OSINT tools revealed the IP address and SSL certificate information for Rekall.



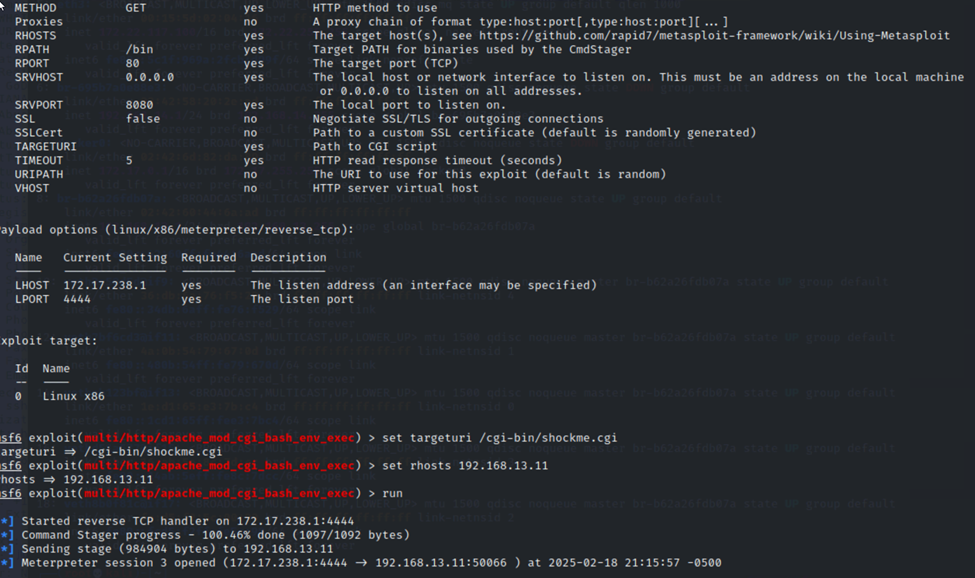
We next used zenmap and nmap tools to learn about the hosts on 192.168.13.0/24 and ran aggressive scans on the hosts to discover any vulnerabilities. One such was regarding Drupal 8 and has a known CVE for having a version lower than recommended by the vendor.



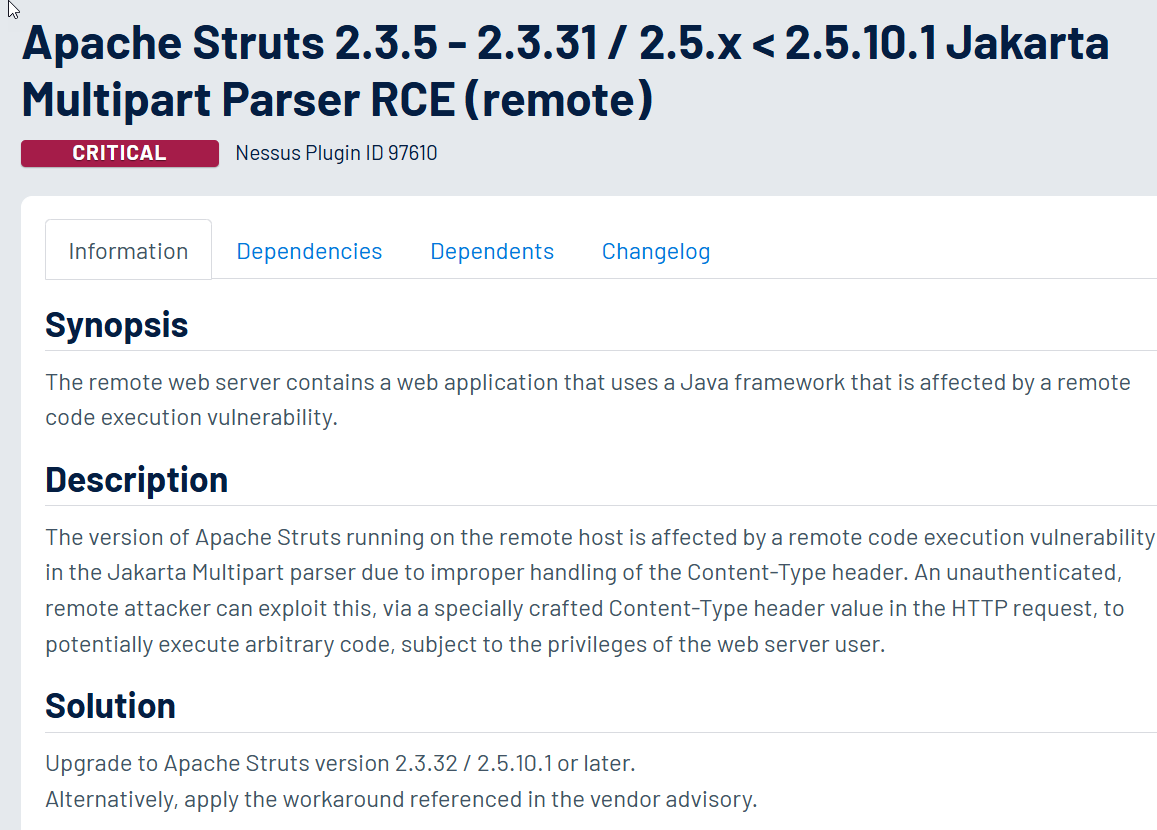
Next, a Nessus scan was used and this tool allowed us to confirm that on the host 192.168.13.12 there was a critical vulnerability (Nessus ID 97610). Next, Metasploit was used on another host 192.168.13.10 in which a Tomcat exploit could be used to gain root access.



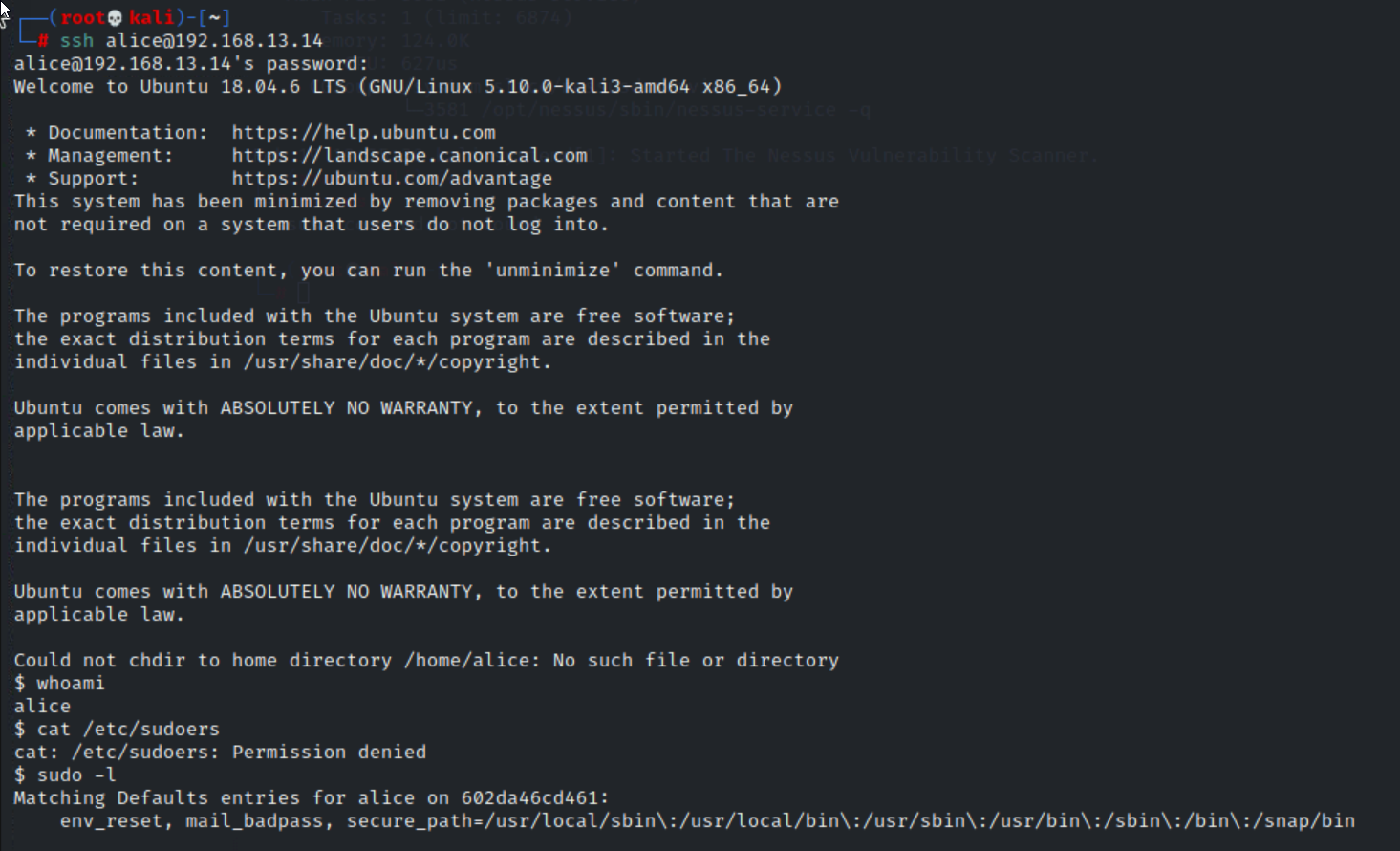
Again with Metasploit, the host 192.168.13.11 was found vulnerable to a ShockMe exploit and allowed us to view the sudoers file and also examine user accounts in /etc/passwd



Again with Metasploit, the host 192.168.13.12 was no less vulnerable since we were able to conduct an exploit related to Apache Struts. Meterpreter allowed us to create a command shell and enabled file navigation.



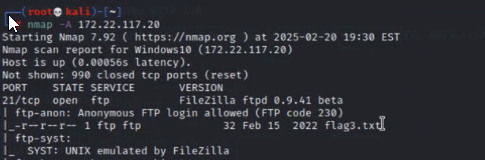
We were able to use an RCE exploit on a host 192.168.13.13 using the results of the nmap scan done earlier. This indicated a Drupal RESTful Web Services unserialize() RCE attack (CVE-2019-6340).

Finally on Day 2 of Linux pen testing, we were able to use an SSH connection with the user account alice as found, then navigate on the host 192.168.13.14 for files. 

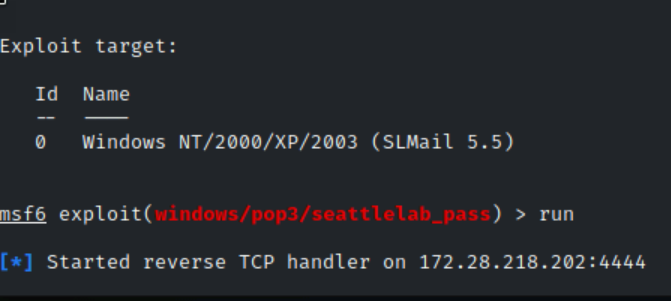
**Day 3**: Windows testing concludes the report with several more free tools at our disposal. On Github.com we were able to find user trivera credentials that once cracked with the tool John the Ripper, allowed us to access deeper on successive testing. An nmap scan showed that we could test the website <http://172.22.117.20> with those credentials and was successful at login. We could access files on the site:



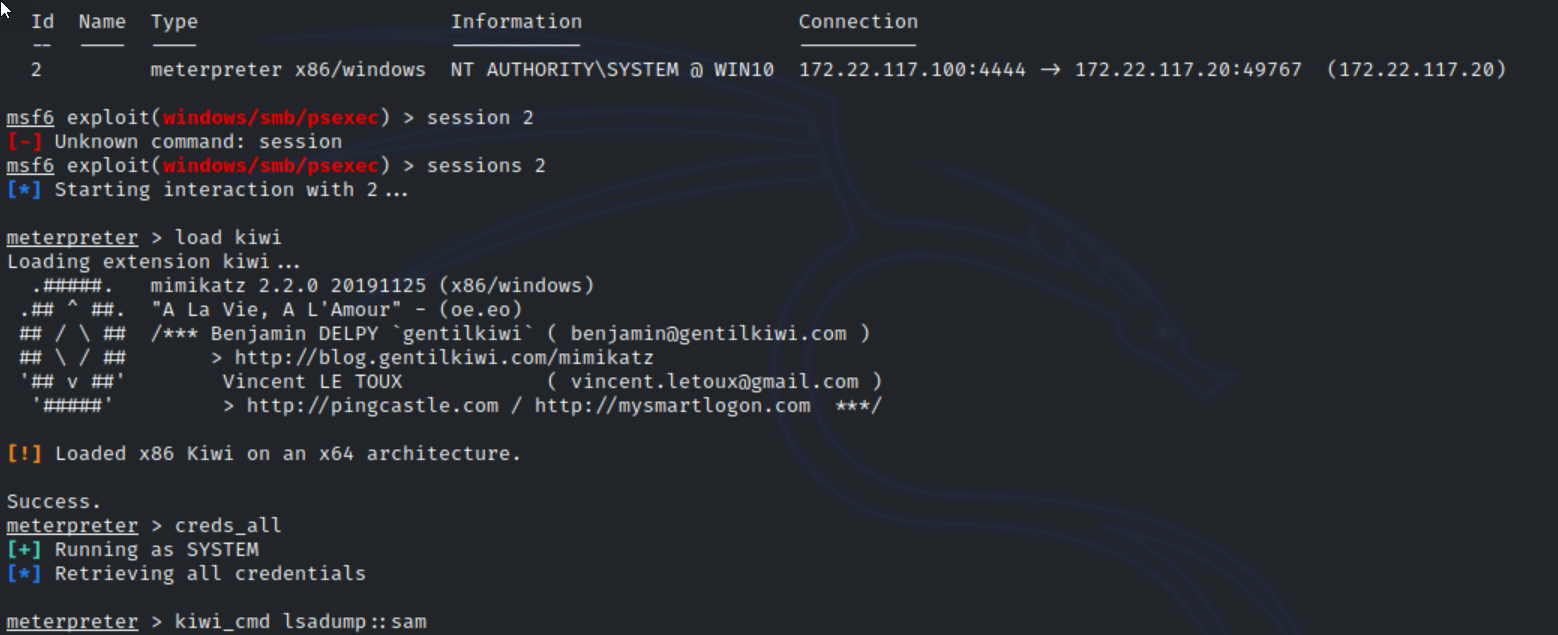
We then used ftp to access further files by using nmap to do aggressive scans of the network. We used default accounts set on the ftp server (username: anonymous).

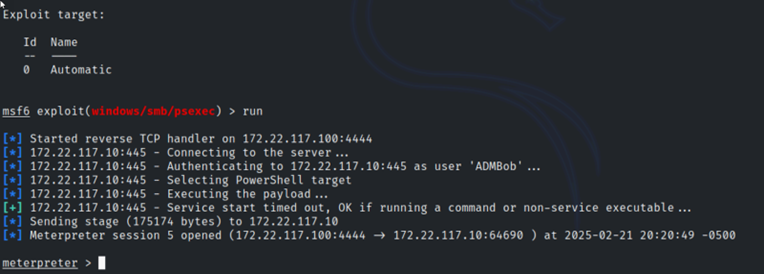


Next, the SLMail service was exploited. This was done by using the Meterpreter exploit: windows/pop3/seattlelab\_pass on host 172.22.117.20. It allowed for a remote connection to the Windows 10 machine.

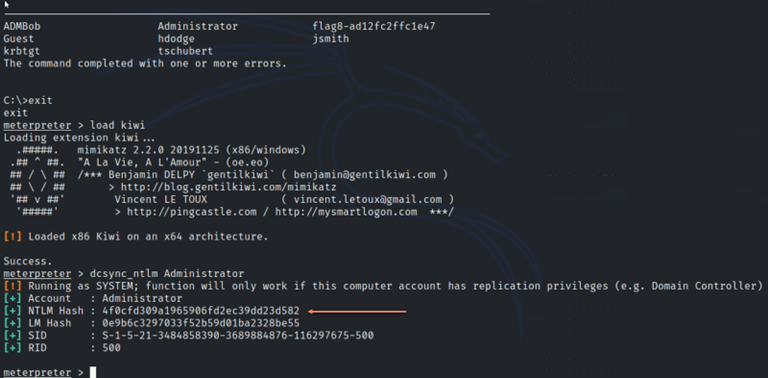


We then were able to examine the scheduled tasks running. Next we used the windows/smb/psexec exploit and the open source tool kiwi to do a credential dump of the user accounts on the Windows 10 machine. These were decrypted using the md5decrypt.net website. We were able to navigate to C:\Users\Public\Documents freely under the exploit.



We could then move laterally to the WinDC01 using more Meterpreter exploits and a found user account from kiwi. Here we could navigate on the C: drive and examine files.

To conclude, we used kiwi again with the command dcsync\_ntlm to find the Administrator account for full SYSTEM access.



## 

Please see Day 1, 2 and 3 complete details below for an expanded explanation of the test results.

## 

## 

## 

## Summary Vulnerability Overview

**Web App DAY 1**

| **Vulnerability** | **Severity** | **Reasoning** |
| --- | --- | --- |
| Reflected Cross-site Scripting (XSS) Payload on Welcome.php | **Medium** | XSS can be used to steal session cookies or deface content, but it depends on user interaction. |
| Cross-site Scripting Vulnerability on Memory-Planner.php | **Medium** | XSS can be used to steal session cookies or deface content, but it depends on user interaction. |
| Cross-site Scripting Vulnerability on Comments.php | **Medium** | XSS can be used to steal session cookies or deface content, but it depends on user interaction. |
| Sensitive Data Exposure via curl | **High** | If sensitive data is exposed, it poses a risk to business processes. |
| Local File Inclusion (LFI exploit) (.php file) | **Critical** | Allows remote code execution (RCE), enabling full system compromise. |
| Local File Inclusion (LFI exploit) (unanticipated file) | **High** | Could be used in combination with other exploits but may not lead to direct execution. |
| SQL Injection Attack on Login.php | **Critical** | Direct access to database = potential full compromise of user data and authentication bypass. |
| Sensitive Information in HTML Source Code | **Critical** | Direct exposure of credentials makes it easy for attackers to log in. |
| Sensitive Data Exposure in robots.txt | **Informational** | Does not pose a direct threat but can help attackers plan further exploits. |
| Command Injection Attack in DNS Check | **Critical** | Allows arbitrary command execution on the server, leading to full compromise. |
| Command Injection Attack n in MX Record Checker | **Critical** | Allows arbitrary command execution on the server, leading to full compromise. |
| Brute Force Attack | **High** | If account lockout is not enforced, it can allow attackers to gain access through trial and error. |
| PHP Injection vulnerability | **Critical** | Remote code execution is a major security risk, potentially leading to full system compromise. |
| Session Management to Restricted Area | **Medium** | Unauthorized access is concerning, but impact depends on what is exposed. |
| Directory Traversal Attack | **High** | Exposure of confidential information can lead to legal issues and reputation damage. |

**Linux OS DAY 2**

| **Vulnerability** | **Severity** | **Reasoning** |
| --- | --- | --- |

| Reconnaissance: OSINT information gathering | **Informational** | WHOIS data is public by design, but it can aid attackers in reconnaissance. |
| --- | --- | --- |
| Reconnaissance: IP address discovery | **Medium** | Internal IPs shouldn’t be publicly accessible; could be used in network attacks. |
| Reconnaissance: SSL certificate information discovery | **Informational** | SSL certificates are public, but attackers can use them to identify expiration dates or weak configurations. |
| Scanning: Available hosts discovery | **Medium** | Network enumeration can aid attackers in mapping the infrastructure for future attacks. |
| Scanning: Aggressive scan on discovered hosts | **Medium** | Aggressive scanning may expose running services, increasing attack surface knowledge. |
| Scanning: Nessus scan on a host for a critical vulnerability on 192.168.13.12 | **High** | Vulnerability scanning can expose exploitable weaknesses but doesn’t directly compromise the system. |
| Exploit: An RCE exploit through Metasploit on 192.168.13.10 | **Critical** | Remote code execution (RCE) provides full control over the target system. |
| Exploit: An RCE exploit through Metasploit on 192.168.13.11 | **Critical** | Remote code execution (RCE) provides full control over the target system. |
| Exploit: Navigating on the server for sensitive information | **High** | Access to /etc/passwd may allow for further privilege escalation or password cracking. |
| Post-Exploitation: RCE exploit on 192.168.13.12 | **Critical** | Apache Struts OGNL Injection (CVE-2017-5638) is a known high-impact RCE vulnerability. |
| Post-Exploitation: RCE exploit on 192.168.13.13 | **Critical** | RCE vulnerability granting full control over the host. |
| Post-Exploitation: Exploiting a host with privilege escalation on 192.168.13.14 | **Critical** | Combination of OSINT and privilege escalation can lead to full system compromise. |

**Windows OS DAY 3**

| **Vulnerability** | **Severity** | **Reasoning** |
| --- | --- | --- |

| Reconnaissance: OSINT search found user credentials | **High** | Stolen credentials pose an indirect threat but require further exploitation to access business systems. If credentials belong to critical accounts, the rating could be elevated. |
| --- | --- | --- |
| Reconnaissance: Used internal credentials to access the web server | **Medium** | Host and file access indicate partial compromise but do not directly threaten business processes unless the accessed file contains sensitive data. |
| Reconnaissance: Scanned network to access a file | **Medium** | Unauthorized access to FTP could expose sensitive data, but its impact depends on the file content and the role of the FTP server. |
| Exploitation: Use Metasploit to exploit a host using SLMail service | **High** | This provides remote control over the machine, potentially leading to further attacks such as privilege escalation or lateral movement. |
| Post-Exploitation: Access to scheduled tasks during the exploit | **Low** | This does not pose a direct threat, but task enumeration could be leveraged for persistence or privilege escalation. |
| Post-Exploitation: Access to local user accounts during the exploit | **High** | Enumeration of user accounts increases the risk of privilege escalation or credential abuse, which can compromise business operations. |
| Post-Exploitation: Examining folder structure during the exploit | **Low** | Unauthorized file access is a concern, but without knowing the file’s contents, it’s not a direct threat. Could be leveraged with other vulnerabilities. |
| Lateral Movement: Escalating from a Windows 10 machine to DC | **Critical** | Compromising the Domain Controller (DC) is a severe threat, as it can result in full domain takeover and disrupt key business functions. |
| Post-Exploitation: Enumeration of the DC folder structure | **Critical** | This suggests full access to the DC's file system, which could lead to exfiltration, service disruption, or further attacks. |
| Post-Exploitation: Administrator NTLM Hash Extraction via DCSync | **Critical** | The highest level of compromise, allowing complete domain control. An attacker can create persistence, access any system, and compromise key business processes. |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | http://192.168.14.35/Welcome.php  http://192.168.14.35/Memory-Planner.php  http://192.168.14.35/Comments.php  http://192.168.14.35/About-Rekall.php  http://192.168.14.35/Login.php  http://192.168.14.35/robots.txt  http://192.168.14.35/networking.php  http://168.192.14.35/disclaimer.php  http://192.168.14.35/souvenirs.php  http://192.168.14.35/admin\_legal\_data.php  totalrekall.xyz  192.168.13.10  192.168.13.11  192.168.13.12  192.168.13.13  192.168.13.14  Github.com  totalrekall domain  http://172.22.117.20  172.22.117.0/24  Windows 10 machine 172.22.117.20  WinDC01 172.22.117.10 |
| Ports | HTTP Port 80  LPORT Port 4444  SSH Port 22  POP3 Port 110  SMB Port 445  RPC Port 135  NetBIOS Ports 137-139 |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 14 |
| **High** | 9 |
| **Medium** | 9 |
| **Low** | 2 |
| **Informational** | 3 |

## 

## 

## 

## Vulnerability Findings

DAY 1: WEB APP PEN TESTING

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | Reflected Cross-site Scripting (XSS) Payload |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Medium** |
| **Description** | On the Welcome.php page, we were able to enter a payload making a pop-up appear. |
| **Images** | Script entered: <script>alert(“I love hacking”)</script>    The pop-up appears by entering the unexpected user input and clicking “GO”:    The fag is revealed: |
| **Affected Hosts** | http://192.168.14.35/Welcome.php |
| **Remediation** | * Stricter controls on the user input for the form. * Regular testing and QA in web development for vulnerabilities. |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | Cross-site Scripting Vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Medium** |
| **Description** | On the Memory-Planner.php page we were able to enter an XSS payload in the “Choose your Character” field to make a pop-up. |
| **Images** | Entered the script: <SCRscriptIPT>alert(“Kevin”)</SCRscriptIPT> into the field:    The pop-up occurs when clicking “GO”:    The flag is revealed: |
| **Affected Hosts** | http://192.168.14.35/Memory-Planner.php |
| **Remediation** | * Stricter controls on the user input for the form. * Regular testing and QA in web development for vulnerabilities. |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | Cross-site Scripting Vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Medium** |
| **Description** | By accessing the Comments.php page we were able to make a pop-up appear with unintended user input. |
| **Images** | Script entered: <script>alert(“Kevin was here”)</script>    The pop-up occurs after clicking “Submit” and the flag is revealed:    Other possible unintended information is revealed: |
| **Affected Hosts** | http://192.168.14.35/Comments.php |
| **Remediation** | * Stricter controls on the user input for the form. * Regular testing and QA in web development for vulnerabilities. |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **High** |
| **Description** | We were able to display sensitive information with the curl command which is freely available. |
| **Images** | Used curl -v<http://192.168.14.35/About-Rekall.php> to reveal the flag within the header information (in the X-Powered-By entry). |
| **Affected Hosts** | http://192.168.14.35/About-Rekall.php |
| **Remediation** | * Disable the verbose logging in Production. * Use secure protocols (https). * Review the server responses for sensitive information. |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Local File Inclusion (LFI exploit) |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Critical** |
| **Description** | We were able to create a test .php file and upload this unanticipated file type into the Memory Planner “Upload your File!” user interaction item. |
| **Images** | Created a file: Kevin.php      Clicked “Browse…” as instructed on the webpage and chose the Kevin.php file. We were able to upload it to the server and reveal the flag. |
| **Affected Hosts** | http://192.168.14.35/Memory-Planner.php |
| **Remediation** | * Secure the form by only allowing the file type that is anticipated. |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Local File Inclusion (LFI exploit) |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **High** |
| **Description** | We tested and were able to load an additional unanticipated file in the Memory Planner instead of the expected user submission of a .jpg file. |
| **Images** | Created a file Kevin.jpg.php and loaded it into the picture upload.    By clicking “Browse…” and selecting the test file: Kevin.jpg.php we were able to upload that potentially dangerous file to the server by clicking “Upload Your File!”. This revealed the flag. |
| **Affected Hosts** | http://192.168.14.35/Memory-Planner.php |
| **Remediation** | * Secure the form by only allowing the file type that is anticipated. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | SQL Injection Attack |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Critical** |
| **Description** | We were able to conduct an SQL Injection attack onto the Login.php page |
| **Images** | me' OR '1' = '1 |
| **Affected Hosts** | http://192.168.14.35/Login.php |
| **Remediation** | * Use parameterized queries (placeholders) instead of the direct user inputs within the query. |

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** | Sensitive Information in HTML Source Code |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Critical** |
| **Description** | We checked deeper into the Login.php page and found valid user credentials stored in the HTML Source Code. |
| **Images** | Opening the page’s Source Code and we were able to find the user account name and password (highlighted):    We tested that this is a valid username and password on the Login.php page and revealed a flag : |
| **Affected Hosts** | http://192.168.14.35/Login.php |
| **Remediation** | * Remove the credentials from the source code. * Secure the HTML code to prevent users from saving their credentials. |

| **Vulnerability 9** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Informational** |
| **Description** | The publicly available file: robots.txt contained more information than necessary. |
| **Images** | We navigated to the robots.txt page and found the following information, including the flag:    Attackers can see a high value target such as web pages not intended for the public. |
| **Affected Hosts** | http://192.168.14.35/robots.txt |
| **Remediation** | * Review the file and remove any sensitive information not intended for public viewing. |

| **Vulnerability 10** | **Findings** |
| --- | --- |
| **Title** | Command Injection Attack |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Critical** |
| **Description** | We were able to enter a command into the DNS Check and click “Lookup”. This was an unanticipated use of the field since commands should not work. |
| **Images** | Entered [www.example.com](http://www.example.com) && cat vendors.txt  Revealed a flag. |
| **Affected Hosts** | http://192.168.14.35/networking.php |
| **Remediation** | * Reject dangerous characters such as: && * Conduct regular security audits. |

| **Vulnerability 11** | **Findings** |
| --- | --- |
| **Title** | Command Injection Attack |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Critical** |
| **Description** | We were able to enter a command into the MX Record Checker and click “Check your MX”. This was an unanticipated use of the field since commands should not work. |
| **Images** | Entered [www.example.com](http://www.example.com/) | cat vendors.txt and clicked to Check your MX |
| **Affected Hosts** | http://192.168.14.35/networking.php |
| **Remediation** | * Reject dangerous characters such as: | * Conduct regular security audits. |

| **Vulnerability 12** | **Findings** |
| --- | --- |
| **Title** | Brute Force Attack |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **High** |
| **Description** | A Brute Force attack was proven successful. By altering the URL on the Disclaimer page, finding some usernames and testing some common logins. |
| **Images** | Altering the URL to include page=/etc/passwd    Revealed some prospective usernames and passwords:    Through trial and error (brute force) it was found melina/melina worked in the main login area: |
| **Affected Hosts** | http://168.192.14.35/disclaimer.php |
| **Remediation** | * Have all users use complex passwords and MFA. * Restrict the altering of URL’s. * Restrict the access of etc/passwd to all but administrators and scrutinize the web pages for this vulnerability. |

| **Vulnerability 13** | **Findings** |
| --- | --- |
| **Title** | PHP Injection vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Critical** |
| **Description** | We were able to use a PHP code in the form of a script to exploit a web page and run a command. |
| **Images** | Tried souvenirs.php as found in robots.txt  A screenshot of a computer  AI-generated content may be incorrect.  Clicking the “Please be sure to ask about options...” link changed the URL to a message:  A screen shot of a computer  AI-generated content may be incorrect.    Changed the message in the URL to: message=””; system(“ls”)  This revealed the flag and also, ls lists some names of secret files.    Php code expanded:  <?php  $message = “”;  System(“ls”);  ?> |
| **Affected Hosts** | http://192.168.14.35/souvenirs.php |
| **Remediation** | * Remove the page from the server if it is not intended for public use. * Only allow safe values in the URL as user input. * Disable dangerous PHP functions which are set in php.ini |

| **Vulnerability 14** | **Findings** |
| --- | --- |
| **Title** | Session Management to Restricted Area |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **Medium** |
| **Description** | We were able to use Burp Suite and use Intruder to open the restricted page |
| **Images** |  |
| **Affected Hosts** | http://192.168.14.35/admin\_legal\_data.php |
| **Remediation** | * Remove old pages from the server when no longer needed. * Monitor for unauthorized folder access. |

| **Vulnerability 15** | **Findings** |
| --- | --- |
| **Title** | Directory Traversal Attack |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | **High** |
| **Description** | Used directory traversal procedure to display a previous version of the “New” Disclaimer page. |
| **Images** | As found in the ls command from Flag 13, the “old\_disclaimers” page relates as being previous to the current version. Was able to traverse the directory to that file instead. |
| **Affected Hosts** | http://192.168.14.35/disclaimer.php |
| **Remediation** | * Remove old pages from the server when no longer needed. |

DAY 2: LINUX PEN TESTING

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | Reconnaissance: OSINT information gathering |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Informational** |
| **Description** | Using a freely available tool within <https://osintframework.com/> we were able to find the WHOIS domain information for the website totalrekall.xyz |
| **Images** | Navigating with Domain Name…Whois Records…DomainTools Whois…    Domain information and the flag were found: |
| **Affected Hosts** | totalrekall.xyz |
| **Remediation** | * Request the domain registrar to mask the whois details. * Regularly monitor for external exposure using OSINT tools. |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | Reconnaissance: IP address discovery |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Medium** |
| **Description** | Using a freely available web tool: nslookup.io we were able to discover that the internal IP address is open to discovery. Flag 2 is revealed on this page too. |
| **Images** | A further nslookup gave an IP address of: **34.102.136.180** |
| **Affected Hosts** | totalrekall.xyz |
| **Remediation** | * Request the domain registrar to mask the internal IP addresses. * Regularly monitor for external exposure using OSINT tools. |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | Reconnaissance: SSL certificate information discovery |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Informational** |
| **Description** | Again using a freely available tool within <https://osintframework.com/> we were able to find the SSL certificate information for the website totalrekall.xyz |
| **Images** | Navigating on <https://osintframework.com/>  OSINT Framework…Domain Name…Certificate Search…crt.sh - Certificate Search…  We used the tool to search the certificates for totalrell.xyz and discovered the certificate information, issuers and the flag. |
| **Affected Hosts** | totalrekall.xyz |
| **Remediation** | * Request to the Certificate Authority to opt out of Certificate Transparency logs. * Consider short-lived certificates to minimize exposure. |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Scanning: Available hosts discovery |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Medium** |
| **Description** | We were able to use zenmap to discover that there were 5 hosts on the network. |
| **Images** | Scan results for a zenmap search on 192.168.13.0/24 showing DNS resolution of 5 hosts (we did not include the host we were scanning from). |
| **Affected Hosts** | 192.168.13.10  192.168.13.11  192.168.13.12  192.168.13.13  192.168.13.14 |
| **Remediation** | * Restrict zone transfers to only trusted DNS servers. * Configure the firewall to limit the scanning. * Use Intrusion Detection and Prevention to reduce enumeration and scanning exposure. |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Scanning: Aggressive scan on discovered hosts |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Medium** |
| **Description** | We were able to run an Aggressive nmap scan on the 5 discover hosts. |
| **Images** | The host 192.168.13.13 is running Drupal 8. |
| **Affected Hosts** | 192.168.13.13 |
| **Remediation** | * Conduct the remediation listed for CVE-2019-6340 since it can lead to PHP code execution. Apply updates as per vendor’s instruction. |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Scanning: Nessus scan on a host for a critical vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | Running a Nessus scan on host: 192.168.13.12 found a critical vulnerability |
| **Images** | Scanning in Nessus:    ID 97610 is the flag |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | * CGI abuses should be remedied by following the Apache vendor’s instructions. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Exploit: An RCE exploit through Metasploit on a host |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | We were able to find an RCE exploit in msfconsole on host 192.168.13.10 to create a remote shell. |
| **Images** | Exploit options:    Created a remote shell:    The flag was found when searching on that host. |
| **Affected Hosts** | 192.168.13.10 |
| **Remediation** | * Apply all vendor patches to prevent remote code execution. * Monitor for unusual activity on the network and in logs. |

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** | Exploit: An RCE exploit through Metasploit on a host |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | We were able to find an RCE exploit in msfconsole on host 192.168.13.11 and create a remote shell. |
| **Images** | The RCE options and connection. Setting the TRAGETURI to the value: /cgi-bin/shockme.cgi    With remote access secured, we were able to search on the host for the flag while investigating the sudoers file: |
| **Affected Hosts** | 192.168.13.11 |
| **Remediation** | * Apply all vendor patches to prevent remote code execution. * Monitor for unusual activity on the network and in logs. |

| **Vulnerability 9** | **Findings** |
| --- | --- |
| **Title** | Exploit: Navigating on the server for sensitive information |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **High** |
| **Description** | We were able to search on the Linux server for further sensitive information, including into /etc/passwd and found flag 9. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.11 |
| **Remediation** | * Apply all vendor patches to prevent remote code execution. * Monitor for unusual activity on the network and in logs. |

| **Vulnerability 10** | **Findings** |
| --- | --- |
| **Title** | Post-Exploitation: RCE exploit on a host |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | We were able to exploit the host 192.168.13.12 using the results of the earlier Nessus scan. That identified Apache Struts Jakarta Multiport Parser OGNL Injection as the likely exploit. |
| **Images** | CVE Information:    The exploit multi/http/struts2\_content\_type\_ongl is chosen:    The options are set to the target and a session is made:    A file is found and opened to reveal the flag. |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | * Configure Web Application Firewall to block exploit attempts. * Use a safer multipart parser. * Limit public access to Apache Struts. * Update to Apache Struts higher version as shown. |

| **Vulnerability 11** | **Findings** |
| --- | --- |
| **Title** | Post-Exploitation: RCE exploit on a host |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | We were able to use an RCE exploit on a host 192.168.13.13 using the results of the nmap scan done earlier. This indicated a Drupal RESTful Web Services unserialize() RCE attack. (CVE-2019-6340) |
| **Images** | CVE Information:    Choosing the exploit:    Setting the options and making a connection:  The server username is www-data for the flag. |
| **Affected Hosts** | 192.168.13.13 |
| **Remediation** | * Disable RESTful Web Services if not required. * Configure Web Application Firewall to block exploit attempts. |

| **Vulnerability 12** | **Findings** |
| --- | --- |
| **Title** | Post-Exploitation: Exploiting a host with privilege escalation |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | We were able to exploit a host 192.168.13.14 with some information found when viewing flag 1 information. Then use privilege escalation to gain more access using SSH. |
| **Images** | Flag 1 information:    SSH into the host with username alice and password: alice.    The CVE of interest:    Utilizing the format given by the CVE, a bash shell is opened and an ls command reveals the final flag. |
| **Affected Hosts** | 192.168.13.14 |
| **Remediation** | * The remediation listed for Vulnerability1. * User training and complex passwords required. * Check sudoers file for misconfigurations. * Restrict SSH access to specific users. * Monitor the sudo logs for suspicious activity. |

DAY 3: WINDOWS PEN TESTING

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | Reconnaissance: OSINT search found user credentials |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **High** |
| **Description** | We were able to discover user credentials in the totalrekall Github repository. |
| **Images** | Searching in Github.com for the totalrekall site:    Within the xampp.users file, a hash credential was found:    The hash was cracked with john the ripper tool: username: trivera and password: Tanya4life |
| **Affected Hosts** | Github.com and the totalrekall domain |
| **Remediation** | * Remove the credentials from Github.com and monitor for any sensitive information in the future. * User awareness training |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | Reconnaissance: Used internal credentials to access the web server |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Medium** |
| **Description** | We were able to scan the network with nmap and discover the active hosts. Using the host 172.22.117.20 we were able to access the web server and use the previously found credentials to access a file. |
| **Images** | Scanning the network range 172.22.117.0/24 using nmap.  Hosts were found:  172.22.117.10 WinDC01  172.22.117.20 Windows 10  172.22.117.100 (local host)      Using <http://172.22.117.20> in a web browser we were able to get a login prompt.    Accessing the website with the found credentials leads to opening an Index of files, including flag2.txt: |
| **Affected Hosts** | <http://172.22.117.20> and 172.22.117.0/24 |
| **Remediation** | * Configure the firewall to limit the scanning. * Use Intrusion Detection and Prevention to reduce enumeration and scanning exposure. * Remove sensitive files from the website. * Use monitoring tools and logs to detect unusual login activity. |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | Reconnaissance: Scanned network to access a file |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Medium** |
| **Description** | We were able to scan the network for a suitable target for instigating an ftp connection. We could gain that access using some default credentials and to access a file. |
| **Images** | Scanning the network with nmap to ensure the target host:    Using an ftp connection to 172.22.117.20 using the default credentials: username: anonymous and password: anonymous.    Then searching successfully for the intended file: flag3.txt and retrieving it.    Opening the flag3.txt file for the flag: |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | * Configure the firewall to limit the scanning. * Use Intrusion Detection and Prevention to reduce enumeration and scanning exposure. * Set the ftp password with a complex username and password. |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Exploitation: Use Metasploit to exploit a host using SLMail service |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **High** |
| **Description** | Using Metasploit we were able to use the exploit: windows/pop3/seattlelab\_pass on host 172.22.117.20 to gain a Meterpreter session and search that host for the file. This exploited the SLmail service. |
| **Images** | Set the exploit using Metasploit:    Set the options and ran the exploit to get a connection session:    Searched the remote host for file access and reveal of the file and its contents: flag4.txt |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | * Replace the outdated SLMail with a modern mail server such as Microsoft 365. * Ensure Windows Defender and antivirus solutions are running and updated. * Enforce MFA for mail access. |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Post-Exploitation: Access to scheduled tasks during the exploit |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Low** |
| **Description** | We were able to view all scheduled tasks running on the Windows 10 machine 172.22.117.20 and identified an unnecessary one, \flag5. This was by virtue of having the SLmail exploit running. |
| **Images** | Found a scheduled task called Taskname: \flag5 which appears to run on computer startup. The unnecessary task is to run a powershell command to list the contents of the C: drive on computer startup.    The Comment section was a string of characters for the flag. |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | * Check for additional persistence mechanisms in the scheduled tasks (schtasks.exe). * Audit and remove unknown and unnecessary tasks that may lead to persistence by an attacker. * Restrict task creation to IT admins. * Deploy an EDR to scan for unusual task creations. |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Post-Exploitation: Access to local user accounts during the exploit |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **High** |
| **Description** | Still while connected to the host 172.22.117.20, we were able to examine the list of users on the local host, having done a further SMB exploit: windows/smb/psexec |
| **Images** | Examining the local users with the command net user. One of the user accounts is: flag6    Using a further exploit: windows/smb/psexec we were able to load kiwi which can be used to list the Hash NTLM for the found accounts:    The NTLM hash for user: flag6 is found:    We could then decrypt that hash using the free tool md5decrypt.net The password for user account flag6 is: Computer! |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | * Enforce complex passwords for all users with user training and password minimum requirements. * Block psexec over SMB in Windows Defender. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Post-Exploitation: Examining folder structure during the exploit |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Low** |
| **Description** | We were able to navigate to C:\Users\Public\Documents freely under the exploit and were able to identify the file flag7.txt and open it with a “type” command. |
| **Images** | Opening C:\Users\Public\Documents\flag7.txt: |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | * Restrict access to C:\Users\Public\Documents using NTFS permissions. * Move sensitive files out of this open access area. |

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** | Lateral Movement: Escalating from a Windows 10 machine to DC |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | We were able to move laterally from the Windows 10 machine to the WinDC01 using Meterpreter exploits. |
| **Images** | With the existing connection to the Windows 10 machine at 172.22.117.20 we put that session into the background and ran an additional exploit for lateral movement.    Since no account was working in our exploit to connect to the WinDC01 at 172.22.117.10 we returned to checking available accounts with kiwi and this time using the command kiwi\_cmd lsadump::cache  This time we had an additional account of interest. Username: ADMBob and password: Changeme!    We successfully used these credentials in the exploit: windows/smb/psexec    We examined the users on the DC and found even more accounts of interest, including the flag8. |
| **Affected Hosts** | Windows 10 machine 172.22.117.20  WinDC01 172.22.117.10 |
| **Remediation** | * Disable SMB v1 and restrict SMB shares using a firewall. * Block unnecessary SMB access between workstations and only allow authenticated access. |

| **Vulnerability 9** | **Findings** |
| --- | --- |
| **Title** | Post-Exploitation: Enumeration of the DC folder structure |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | Continued enumeration on WinDC10 allowed full navigation on the C: drive. |
| **Images** | The heart of the server is the C: drive (root) and when checking the contents with the command dir flag\*.\* we find flag9.txt:    Opening flag9.txt reveals the flag: |
| **Affected Hosts** | 172.22.117.10 |
| **Remediation** | * Restrict SYSTEM network account privileges. * Detect and block the SYSTEM account from unauthorized actions like dumping credentials or executing commands remotely. |

| **Vulnerability 10** | **Findings** |
| --- | --- |
| **Title** | Post-Exploitation: Administrator NTLM Hash Extraction via DCSync |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | We were able to display the hash of the Administrator password using the kiwi tool dcsync. |
| **Images** | Returning to Meterpreter, we loaded kiwi and ran the command: dcsync\_ntlm Administrator to display the NTLM Hash which was the final flag. |
| **Affected Hosts** | 172.22.117.10 |
| **Remediation** | * Ensure only DC’s can perform replication. * Monitor for dcsync activity in Windows Event Logs. |