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

Contact Information

Company Name	Secure Matrix, LLC
Contact Name	Emil T Merdzhanov
Contact Title	Lead Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	6/17/2023	1	

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.

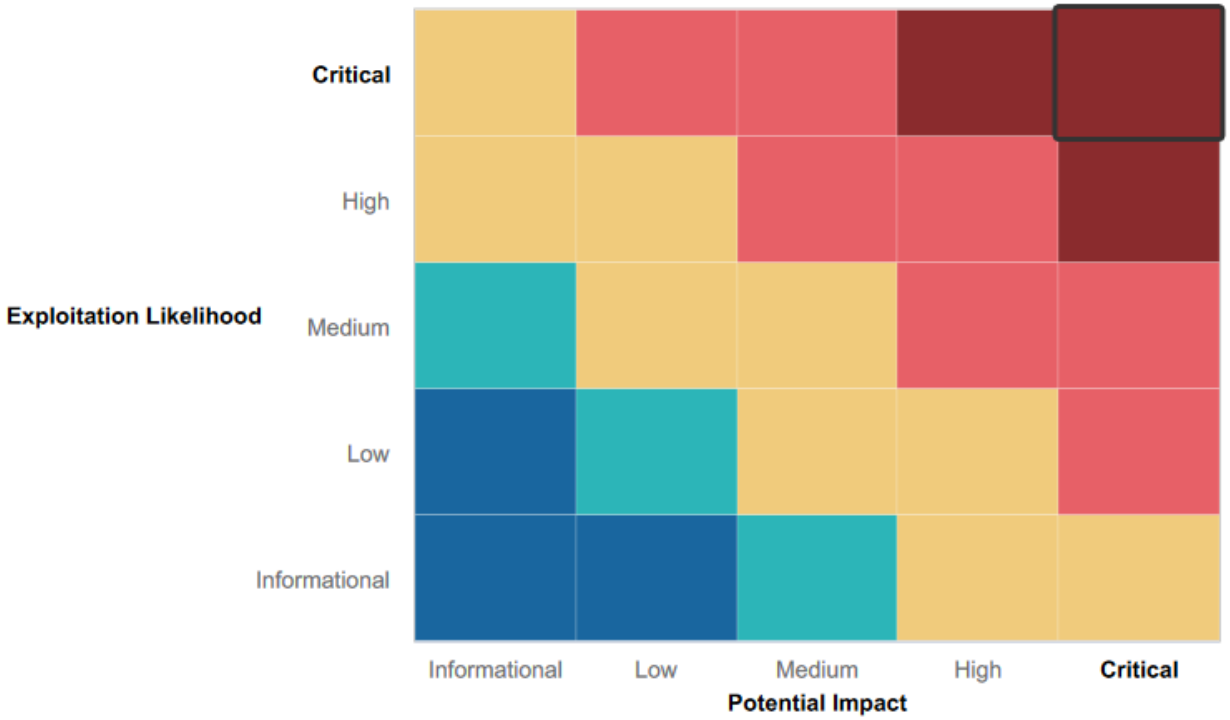
Executive Summary of Findings

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:



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.

- Limited Impact of Exploitation: The presence of low-risk vulnerabilities and successful mitigation of some high-risk vulnerabilities suggests that certain security measures are in place to mitigate risks and limit the impact of potential exploitation.
- Partial Protection of Passwords: While weak password storage practices were identified, the inability to crack password hashes indicates that some level of protection is in place for sensitive credentials. However, further improvements are recommended.

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.

1. **Cross-Site Scripting (XSS) Vulnerabilities:** Multiple instances of XSS vulnerabilities were discovered, posing a high risk of unauthorized code execution and potential data theft or manipulation.
2. **Command Injection Vulnerabilities:** The image upload functionality and other components were found to be vulnerable to command injection attacks, enabling arbitrary command execution and potential system compromise.
3. **SQL Injection Vulnerabilities:** The login page and potentially other sections of the website were susceptible to SQL injection, which could lead to unauthorized access, data breaches, or database manipulation.
4. **Unauthorized Access and Privilege Escalation:** Weak or easily guessable credentials, misconfigured permissions, and vulnerabilities in various services allowed unauthorized access and potential privilege escalation, exposing sensitive data and compromising system integrity.
5. **Information Leakage:** Information leakage was observed through the DomainDossier tool, potentially providing attackers with valuable insights for further exploitation or reconnaissance.
6. **Unprotected Docker Containers:** Vulnerabilities in the Docker container running the CTF website allowed unauthorized access to the system, emphasizing the need for secure exploitation container configurations and regular updates.
7. **Vulnerabilities in Network Services**: Vulnerabilities in SLMail services and other network services were identified, posing a high risk of unauthorized access, information disclosure, or compromise of the affected systems.
8. **Weak Password Storage and Hash Cracking:** Weak password storage practices were evident, leading to the extraction of password hashes. However, the cracking of these hashes was not successful, highlighting the importance of strong password storage mechanisms.

Addressing these weaknesses and vulnerabilities is crucial to fortify the system's security. Remediation efforts should focus on implementing secure coding practices, input validation and sanitization, strong access controls, regular patching and updates, secure container configurations, and password policies that enforce strong and unique passwords.

Executive Summary

The conducted penetration testing engagement revealed several significant vulnerabilities within the target system. These vulnerabilities include cross-site scripting (XSS) in various sections of the website, command injection through image uploads, SQL injection in the login page, unauthorized access to DNS records, and exploitation of Docker containers. Additionally, information leakage was identified in the DomainDossier tool, and vulnerabilities were discovered in SLMail services and the Task Scheduler.

The findings underscore the importance of implementing robust security measures to mitigate these risks. Remediation efforts should focus on implementing input validation and output encoding to prevent XSS attacks, securing file upload mechanisms, using parameterized queries to mitigate SQL injection, enforcing proper access controls, and regularly updating and patching Docker containers and system services. Furthermore, user awareness and training on secure password practices, network monitoring, and file integrity checks are crucial to enhancing overall system security.
By addressing these vulnerabilities and implementing the recommended remediations, organizations can fortify their systems against potential exploitation, unauthorized access, and data breaches.

Some of the techniques and tools we used during the test include:
Cross-Site Scripting (XSS) Exploitation:

- Technique: Injecting malicious scripts into a website to execute arbitrary code in users' browsers.

Command Injection:

- Technique: Exploiting vulnerabilities in systems that allow user-supplied input to be executed as commands.
-

SQL Injection:

- Technique: Exploiting poorly sanitized or insufficiently validated user inputs to manipulate SQL queries and gain unauthorized access to a database.

Unauthorized Access and Privilege Escalation:

- Technique: Leveraging weak credentials, vulnerabilities, or misconfigurations to gain unauthorized access to systems or escalate privileges.

Exploiting Vulnerable Services and Systems:

- Technique: Identifying and exploiting vulnerabilities in specific services or systems to gain unauthorized access or execute malicious actions.

It is important to note that these techniques and tools can be easily accessible to attackers with basic technical knowledge and can be implemented relatively quickly. Therefore, organizations must prioritize security measures such as secure coding practices, vulnerability assessments, regular patching and updates, strong access controls, and security awareness training to protect their systems and mitigate the risks associated with these techniques.

Summary Vulnerability Overview

Vulnerability	Severity
Cross-Site Scripting (XSS) Vulnerability	High
Cross-Site Scripting (XSS) Vulnerability in Comments Section	Medium
Arbitrary File Upload Vulnerability	High
SQL Injection Vulnerability in Login Page	High
Information Disclosure and Unauthorized Access Vulnerability	Medium
Information Disclosure through Robots.txt	Low
Command Injection Vulnerability in Admin Network Tools	High
Improper Configuration of Docker Container Security	Critical
Exposure in Domain Registration Details	Low
Active Hosts Discovery in Network Scan	Low
Drupal Host Identification from Aggressive Network Scan	Low
Critical Vulnerability Identification	Critical
Remote Code Execution (RCE) Exploit	High
Remote Code Execution (RCE) Exploit and Privilege Escalation	High
Flag Discovery in /etc/passwd File and File Integrity Protection	Medium
Successful Exploitation of Apache Struts RCE Vulnerability	High
Successful Exploitation, Privilege Escalation on 192.168.13.14	High
Successful Password Decryption and System Access in Windows OS	High
Successful System Access and Flag Retrieval through HTTP Server	Medium
Open Port FTP Access and Flag Retrieval	Critical
Successful Exploitation of SLMail Services and Flag Retrieval	High
Task Scheduler Investigation and Flag Discovery	Medium
Password Retrieval and Decryption from Compromised System	High
Flag Discovery in Exposed Public Folder	Low
Lateral Movement and Password Cracking for WinDC Access	High
System Exploration and Flag Discovery in "C:\system32\config"	Medium
Administrator Password Hash Retrieval and Unsuccessful	Low

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

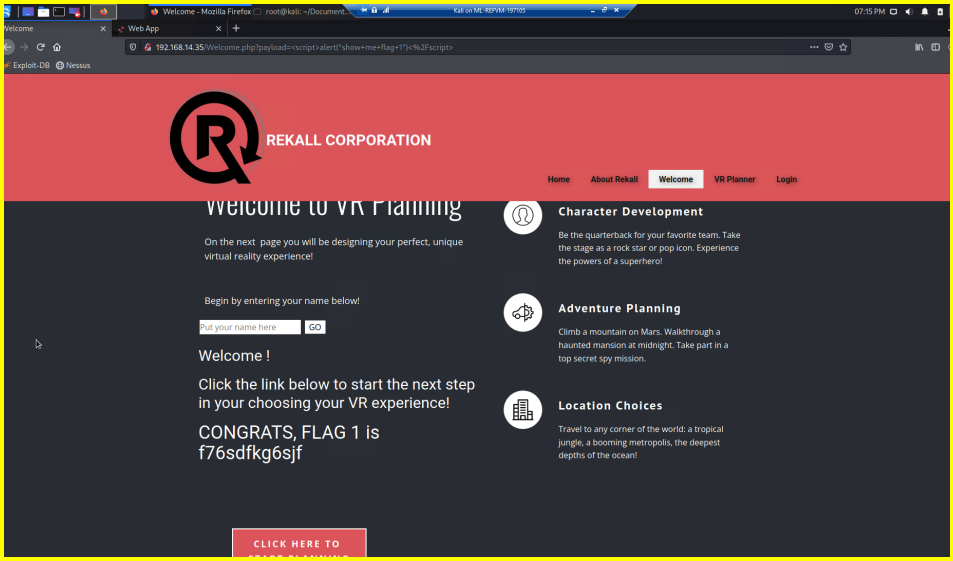
Scan Type	Total
Hosts	192.168.14.35; 34.102.136.180; 182.168.13.14; 182.168.13.10; 182.168.13.11;182.168.13.12; 182.168.13.13; 712.22.117.20

Ports	FTP 21, 22, 80, 443

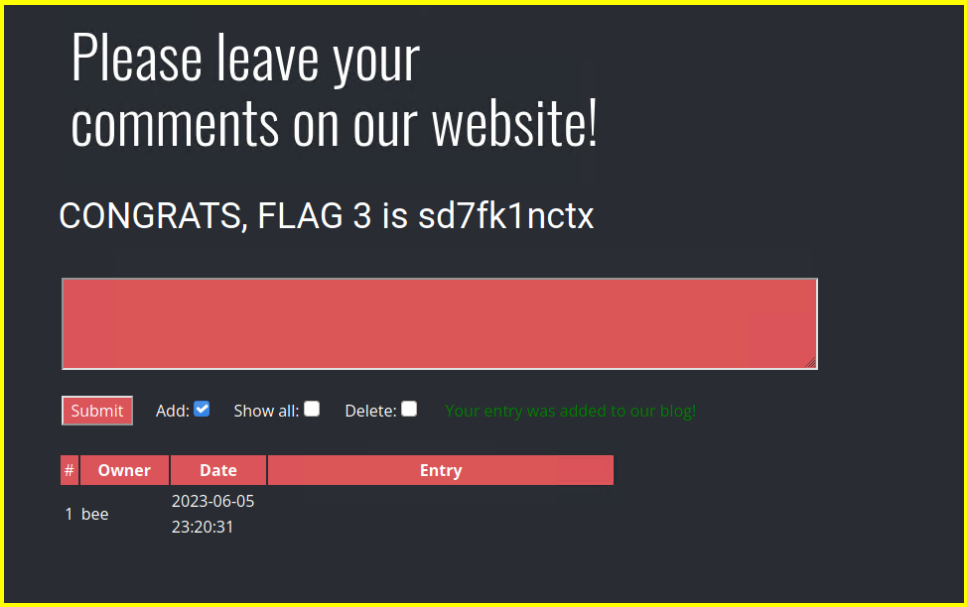
Exploitation Risk	Total
Critical	3
High	11
Medium	6
Low	6

Vulnerability Findings

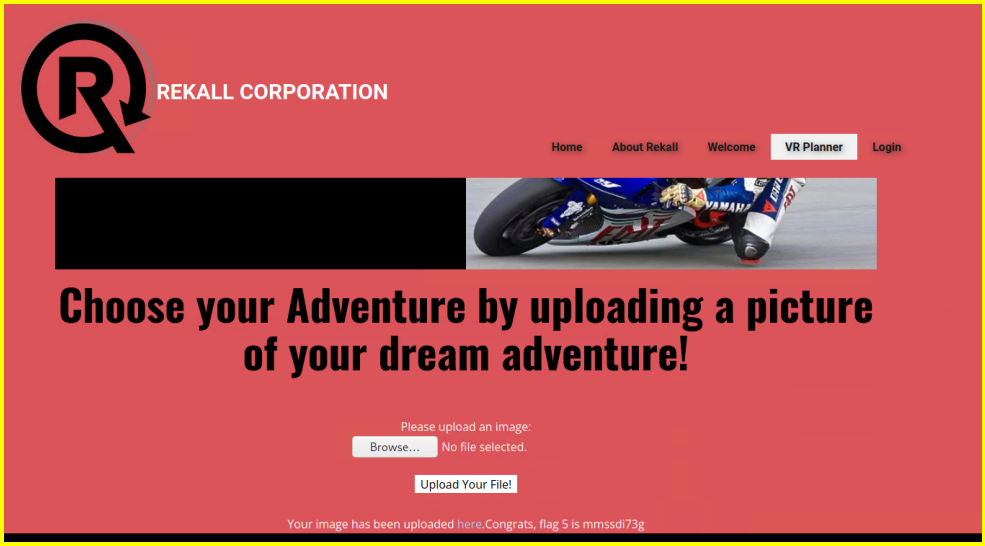
Vulnerability 1	Findings
Title	Cross-Site Scripting (XSS) Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	FLAG 1 is f76sdfkg6sjf The website at http://192.168.14.35/Welcome.php is found to be vulnerable to a cross-site scripting (XSS) attack. This vulnerability allows an attacker to inject malicious scripts into the website, which can then be executed on the client side, potentially compromising user data and exposing sensitive information. The exploitation of this vulnerability was demonstrated by injecting a script into the "put your name here" field, resulting in the display of Flag 1 (f76sdfkg6sjf) in an alert dialog box. This indicates that the website does not properly sanitize or validate user input, allowing arbitrary script execution.

Images	
Affected Hosts	http://192.168.14.35
Remediation	• **Input Validation and Sanitization:** Implement strict input validation and sanitization mechanisms to ensure that user-supplied data is properly filtered and does not contain malicious scripts. • **Output Encoding:** Encode any user-supplied data that is displayed on the website to prevent script execution. • **Content Security Policy (CSP):** Implement a Content Security Policy to restrict the execution of scripts from external sources and inline scripts. This helps mitigate the impact of XSS attacks by limiting the potential sources of executable code.

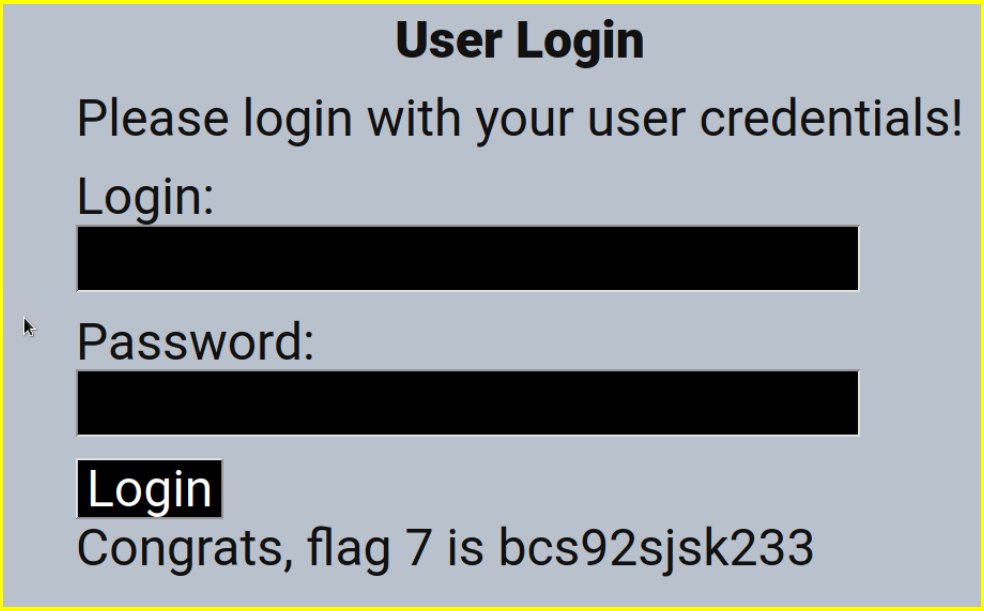
Vulnerability 2	Findings
Title	Cross-Site Scripting (XSS) Vulnerability in Comments Section
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	FLAG 3 is sd7fk1nctx The Comments section of the REKALL CORPORATION website, accessible at http://192.168.14.35/Comments.php, is found to be vulnerable to a cross-site scripting (XSS) attack. This vulnerability allows an attacker to inject malicious scripts into the website, which can then be executed on the client-side, potentially compromising user data and exposing sensitive information.

	The exploitation of this vulnerability was demonstrated by injecting a script in the comment payload field, resulting in the display of Flag 3. The injected script <script>alert("flag")alert</script> successfully triggered an alert dialog, indicating that the website does not properly sanitize or validate user input, allowing arbitrary script execution.
Images	
Affected Hosts	http://192.168.14.35
Remediation	• **Input Validation and Sanitization:** Implement strict input validation and sanitization mechanisms to ensure that user-supplied data is properly filtered and does not contain malicious scripts. • **Output Encoding:** Encode any user-supplied data that is displayed on the website to prevent script execution. Apply appropriate encoding methods depending on the context of

Vulnerability 3	Findings
Title	Arbitrary File Upload Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	FLAG 5 mmssdi73g The VR Planner website allows users to upload images, but it lacks proper validation and security measures, resulting in an arbitrary file upload vulnerability. This vulnerability enables an attacker to upload and execute arbitrary PHP scripts, which can lead to unauthorized access, data manipulation, or further exploitation of the system.

	Exploiting this vulnerability, you were able to upload the script.php file, which contains code that executes arbitrary system commands provided through the "cmd" parameter. The uploaded script allowed you to execute commands and obtain the flag.
Images	
Affected Hosts	http://192.168.14.35
Remediation	• **File Type Validation**: Implement strict file type validation on the server-side to ensure that only allowed file types (e.g., image files) can be uploaded. This validation should be based on both the file extension and the file's content. • **File Content Verification**: Implement mechanisms to verify the content of the uploaded files. This can include checking for file signatures or using antivirus scanners to detect potentially malicious files. • **Secure File Storage:** Store uploaded files outside the web root directory or implement access controls to prevent direct execution of uploaded PHP files. This ensures that even if a malicious file is uploaded, it cannot be directly executed.

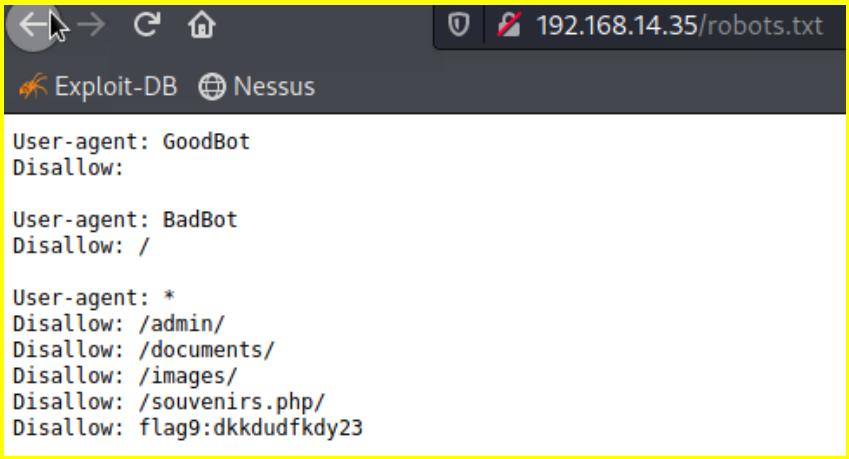
Vulnerability 4	Findings
Title	SQL Injection Vulnerability in Login Page
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	Flag 7 bcs92sjsk233 The login functionality of the website's login.php page is found to be vulnerable to SQL injection. This vulnerability allows an attacker to manipulate the SQL query used for authentication, bypassing the intended logic and potentially

	gaining unauthorized access to sensitive information. Exploiting this vulnerability, you entered the password value as "1' OR '1' = '1" in the password field, which resulted in the SQL query being altered to always evaluate to true. This allowed you to bypass the authentication mechanism and obtain the flag.
Images	
Affected Hosts	http://192.168.14.35
Remediation	• **Input Validation and Sanitization:** Perform strict input put. • **Principle of Least Privilege:** Ensure that the database user account used by the application has the least privileges necessary to perform its intended functions. This limits the potential impact of a successful SQL injection attack by restricting the attacker's access to sensitive data or operations. • **Error Handling and Reporting:** Implement robust error handling and reporting mechanisms that avoid exposing sensitive information or detailed error messages to users. Provide generic error messages to users while logging detailed error information securely for debugging purposes.

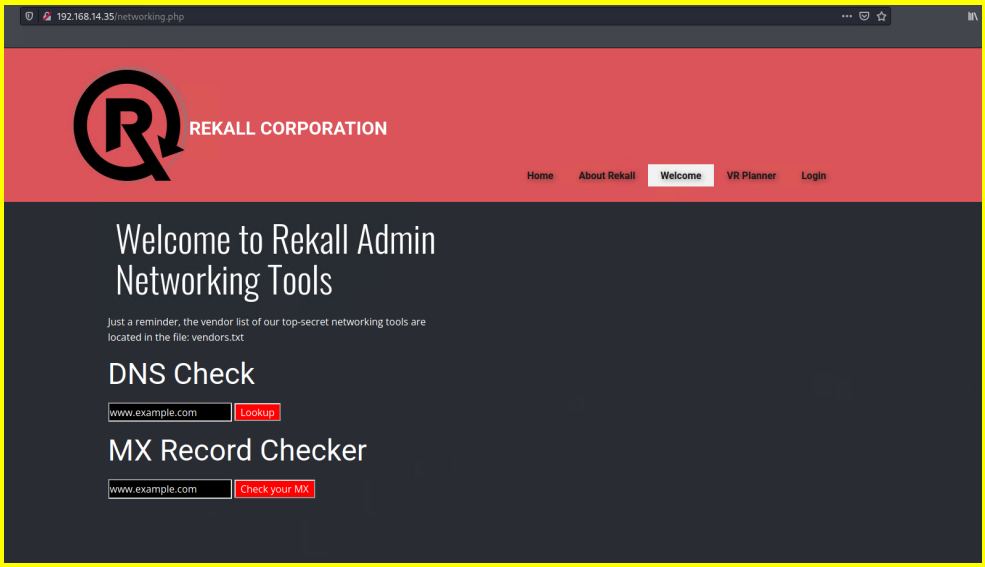
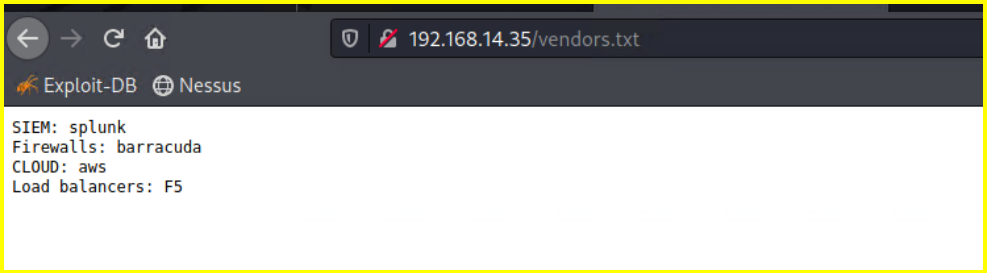
Vulnerability 5	Findings
Title	Information Disclosure and Unauthorized Access Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	FLAG 8: 87fsdkf6djf

	The login page source code of the website contains sensitive information, including the admin credentials (username: dougquaid, password: kuato), which should not be exposed to users. This information disclosure poses a security risk as it provides unauthorized individuals with access to administrative privileges and potentially sensitive areas of the website, such as DNS records. Exploiting this vulnerability, you were able to identify the admin credentials by examining the page source code. This reveals a lack of proper access controls and confidentiality measures, as sensitive information should not be accessible in this manner.
Images	Successful login! flag 8 is 87fsdkf6djf , also check out the admin only networking tools **HERE**
Affected Hosts	http://192.168.14.35
Remediation	• Secure Code Practices: Avoid including sensitive information, such as admin credentials, in the page source code or other client-side resources. Ensure that sensitive information is properly handled and stored securely on the server-side. • Access Controls: Implement strict access controls to restrict access to sensitive areas of the website, such as DNS records. Apply the principle of least privilege, providing users with only the privileges necessary for their specific roles or tasks.

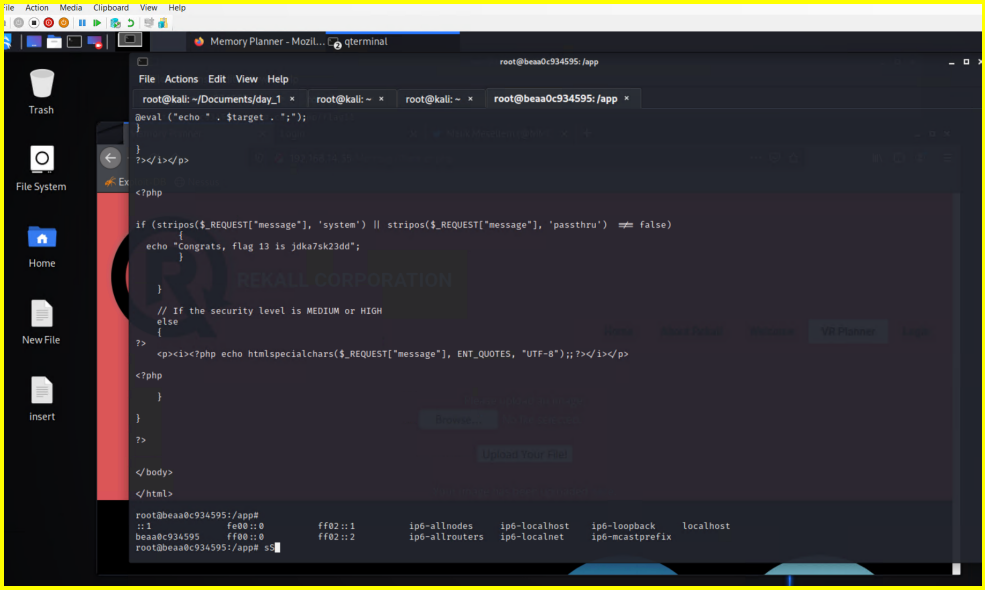
Vulnerability 6	Findings
Title	Information Disclosure through Robots.txt
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Low
Description	The robots.txt file on the website, accessed at http://192.168.14.35/robots.txt, contains instructions for web crawlers and other web robots on what areas of the website they should or should not crawl. While the robots.txt file is intended to be publicly accessible, it can inadvertently disclose information about the website's directory structure and potentially reveal sensitive areas. By examining the robots.txt file, you were able to gather information about the allowed and disallowed areas of the website for web crawlers. This

	information, although not necessarily harmful on its own, can aid in reconnaissance and potentially provide insights to attackers.
Images	
Affected Hosts	http://192.168.14.35
Remediation	• **Limited Exposure:** Review and assess the content of the robots.txt file to ensure that it does not expose sensitive directories, files, or areas of the website. Restrict access to sensitive areas through other means such as authentication, access controls, and secure configurations. • **Proper Configuration:** Ensure that the robots.txt file does not inadvertently disclose information that could be useful to potential attackers. Exclude any sensitive directories, such as administration or internal system areas, from being listed in the robots.txt file. •

Vulnerability 7	Findings
Title	Command Injection Vulnerability in Admin Network Tools
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	The Admin Network Tools feature on the website exhibits a command injection vulnerability, allowing unauthorized users to execute arbitrary commands on the underlying system. This vulnerability poses a significant risk as it enables an attacker to execute unauthorized commands, potentially leading to unauthorized access, data manipulation, or further system exploitation. Exploiting this vulnerability, you utilized the DNS check functionality to execute the command www.example.com;cat vendors.txt, which allowed you to retrieve the flag from the vendors.txt file. This demonstrates that the website does not properly validate or sanitize user-supplied input, allowing arbitrary command execution.

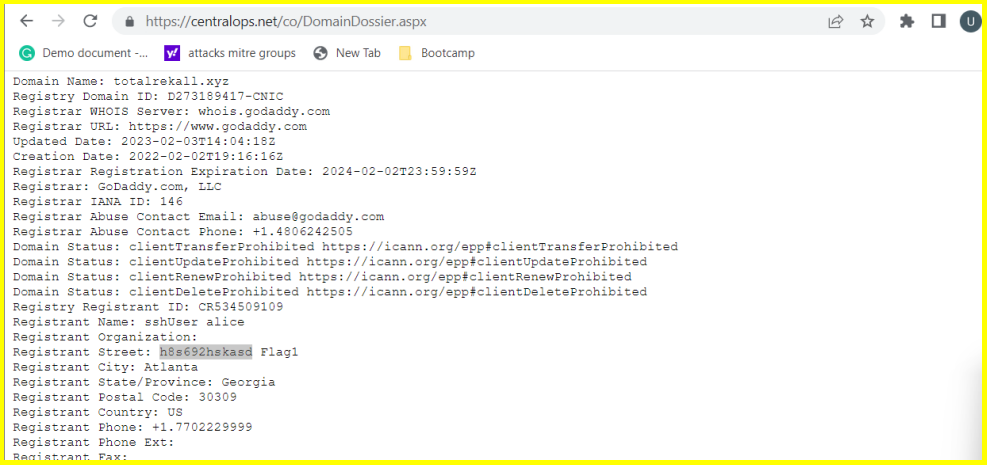
Images	  
Affected Hosts	http://192.168.14.35
Remediation	• **Input Validation and Sanitization:** Implement strict input validation and sanitization mechanisms to ensure that user-supplied data is properly filtered and does not contain malicious commands or characters. • **Parameterized Commands:** Utilize parameterized commands or safe APIs when executing system commands to prevent command injection

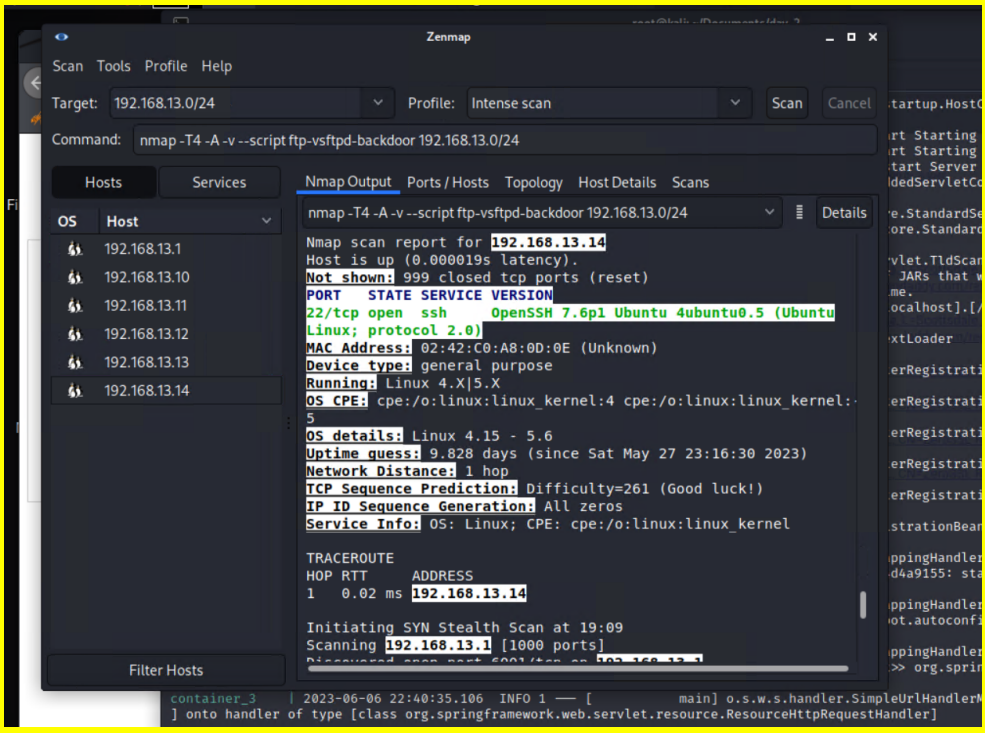
	attacks. This ensures that user-supplied input is treated as data rather than executable code. ● **Secure Configuration:** Disable or restrict access to any unnecessary system commands or functionalities that could be abused by potential attackers. Regularly review and update the system configuration to align with security best practices.

Vulnerability 8	Findings
Title	Improper Configuration of Docker Container Security
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	The Docker container hosting the CTF website has been found to have improper configuration, allowing direct access to sensitive files, including the flags. This misconfiguration poses a significant security risk, as it enables unauthorized users to obtain sensitive information without going through the intended challenge-solving process. By accessing the Docker container, you were able to directly find and retrieve all the flags, bypassing the intended security measures of the CTF website. This indicates a failure in properly securing the containerized environment.
Images	
Affected Hosts	http://192.168.14.35
Remediation	● **Secure Container Configuration:** Review and update the Docker container configuration to enforce proper security measures. Ensure unnecessary services, files, or directories are not exposed or

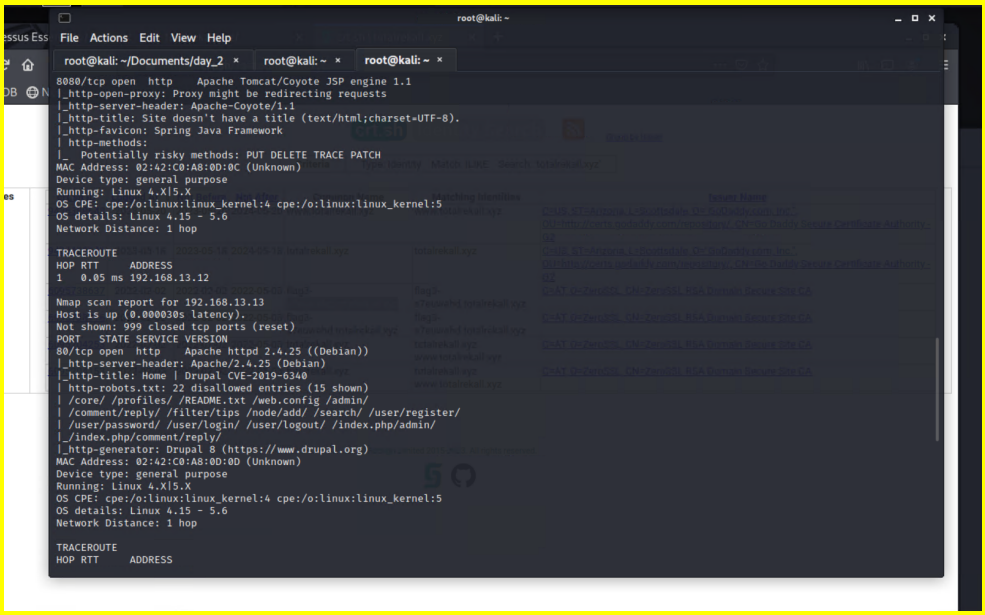
	accessible within the container. • **Restrict File Access:** Implement access controls and permissions within the Docker container to restrict access to sensitive files, including the flags. Only authorized processes or users should have the necessary privileges to access and retrieve this information.

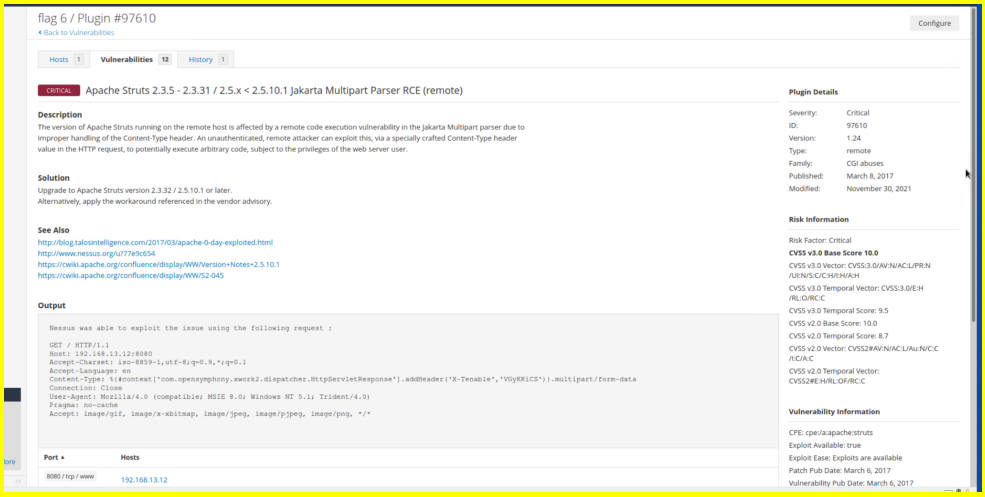
DAY 2

Vulnerability 9	Findings
Title	Exposure in Domain Registration Details
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Low
Description	Day 2 - Flag1: h8s692hskasd During the analysis of the domain "totalrecall.xyz" using DomainDossier, a discovery was made where the first flag, referred to as Flag1, was found within the register Street category. This finding indicates that sensitive information related to the CTF challenge, specifically Flag1, has been exposed publicly in the dom ain registration details. While the exposure of Flag1 itself does not pose an immediate security risk, it may provide attackers with hints or clues to progress through the CTF challenge without solving it legitimately. Additionally, it could impact the integrity and fairness of the CTF competition.
Images	
Affected Hosts	34.102.136.180
Remediation	**Privacy and Confidentiality Considerations:** Ensure that any sensitive information related to the CTF challenge, including flags, is properly protected and not publicly accessible. Avoid including such information in public registration details or any other publicly accessible sources.

Vulnerability 10	Findings
Title	Active Hosts Discovery in Network Scan
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low
Description	Flag 4 During a comprehensive network scan using Nmap or Zenmap, the network with an initial address of 192.168.13.0/24 was examined. The scan results indicate the presence of five active hosts within the network. This finding is significant as it aligns with flag number four, where the answer to the flag is "5." Discovering the exact number of active hosts in the network provides valuable information about the network's size and the number of devices connected to it. However, without additional context or vulnerability information, this finding does not pose an immediate security risk.
Images	
Affected Hosts	192.168.13.14
Remediation	**Network Segmentation**: Implement network segmentation to isolate different network segments or zones based on security requirements. This helps limit

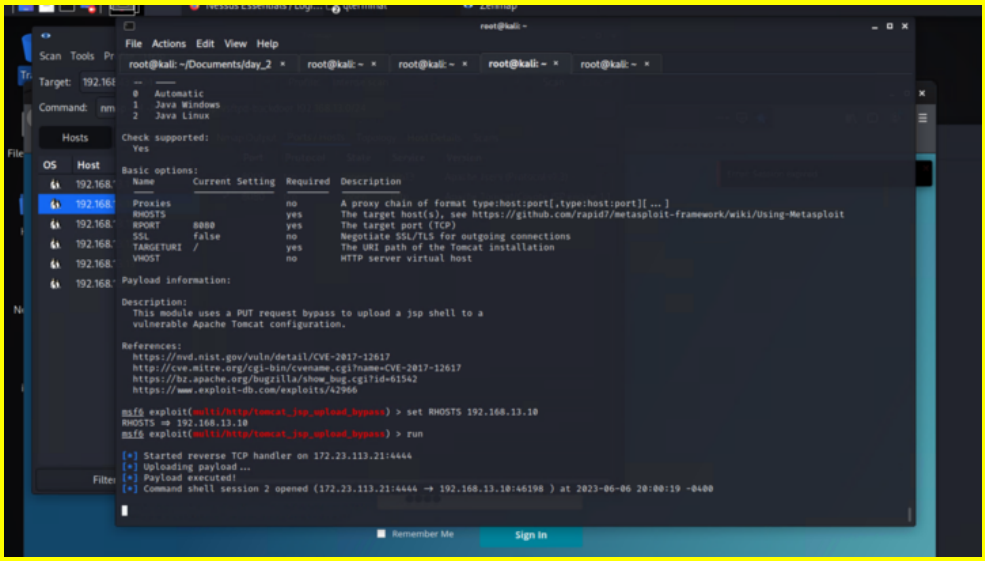
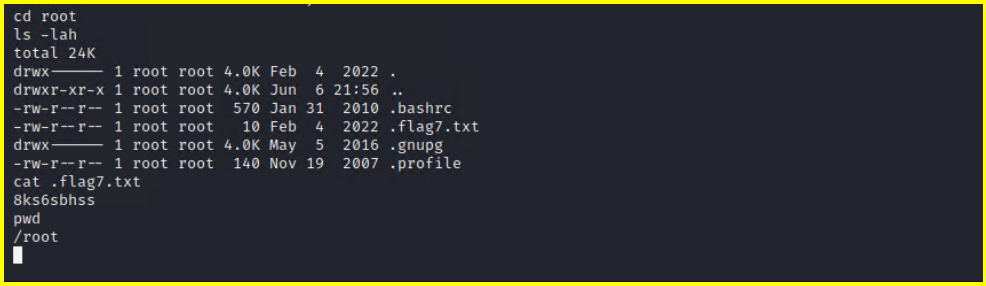
	the potential impact of a security breach or unauthorized access to critical resources.

Vulnerability 11	Findings
Title	Drupal Host Identification from Aggressive Network Scan
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low
Description	Flag 5 During an aggressive network scan, the host running Drupal with the IP address 192.168.13.13 was identified. This information was considered significant and marked as a flag. Identifying the presence of a Drupal instance can provide insights into the technologies used within the network and aid in further assessments or targeted attacks. However, without additional context or vulnerability information, the identification of a Drupal host itself does not pose an immediate security risk.
Images	
Affected Hosts	192.168.13.13
Remediation	**Monitoring and Intrusion Detection**: Implement robust monitoring and intrusion detection systems to identify and respond to any suspicious activities or potential attacks targeting the Drupal installation. Monitor for unusual traffic patterns, unauthorized access attempts, or changes in the Drupal environment.

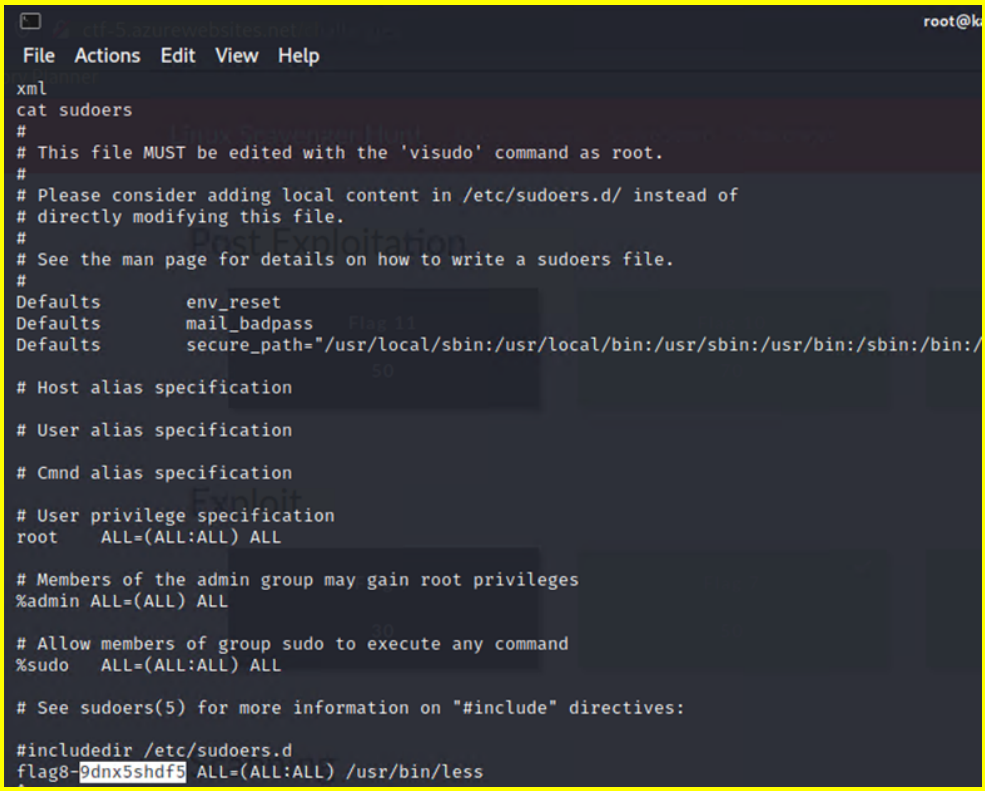
Vulnerability 12	Findings
Title	Critical Vulnerability Identification
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	FLAG 6 As instructed, a Nessus scan was conducted targeting the host ending with "192.168.13.12". The scan results revealed the presence of a critical vulnerability, and further examination of the vulnerability details was performed. During this analysis, it was observed that the flag corresponds to the ID number displayed at the top right corner of the page, specifically identified as 97610. This finding highlights the criticality of the identified vulnerability and the importance of addressing it promptly to mitigate potential security risks
Images	
Affected Hosts	192.168.12.13
Remediation	**Patch or Mitigate the Vulnerability:** Based on the specific details of the critical vulnerability, apply the recommended patches, security updates, or mitigation strategies provided by the vendor or security advisory. Prioritize the remediation of critical vulnerabilities to minimize the risk of exploitation.

Vulnerability 13	Findings
Title	Remote Code Execution (RCE) Exploit
Type (Web app / Linux OS / WIndows OS)	Linux OS

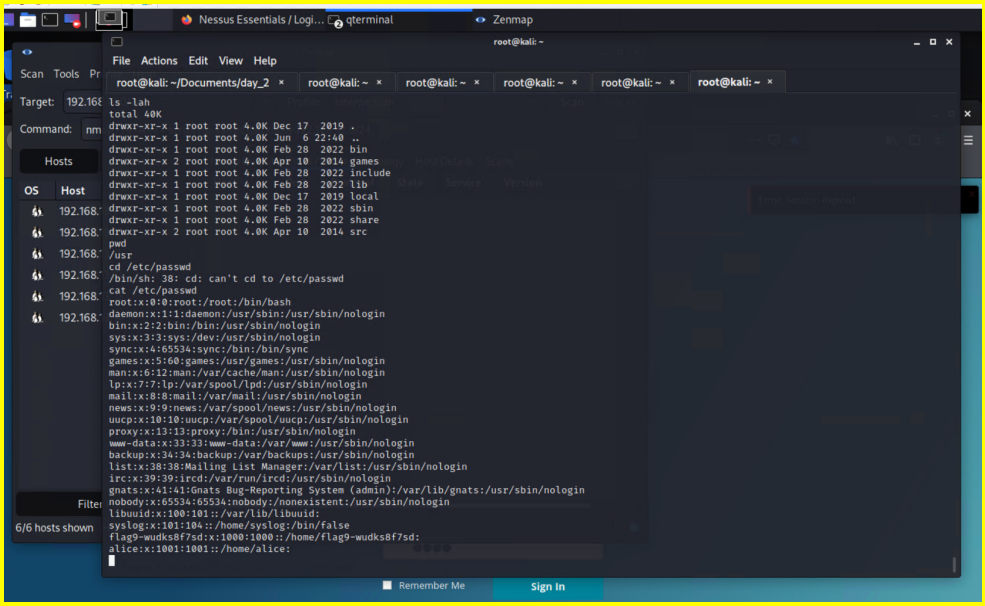
Risk Rating	High
Description	Flag 7 In line with the assigned task, an aggressive Nmap scan was conducted, and the host ending with 192.168.13.10 was identified as the target. Based on the Nessus vulnerability scan results, Apache Tomcat was found to be vulnerable, leading to a focused effort to exploit this specific weakness using Metasploit. After attempting various exploits, the payload "/multi/http/tomcat_jsp_upload_bypass" was successfully executed, establishing a connection between the LHOST (172.23.113.21) and the target RHOST (192.168.13.10). This allowed for unauthorized access to the target system. Once access was gained, a search was performed within the server to locate Flag 7. By executing the command "ls -lah" to list the contents of the root directory, a hidden text file named ".flag7.txt" was discovered, which contained the desired flag. This successful exploit and flag retrieval demonstrate the severity of the vulnerability and the potential impact it can have on the target system's security. **Vulnerabilities:** - Unpatched Apache Tomcat: The target host had an unpatched version of Apache Tomcat, which was susceptible to known vulnerabilities. - Remote Code Execution (RCE): Exploiting the Apache Tomcat vulnerability allowed the attacker to achieve remote code execution on the target system. - Insufficient Access Controls: The presence of the vulnerable Apache Tomcat server suggests that access controls might not have been properly configured, allowing unauthorized access to sensitive areas. - Hidden File: The flag was stored in a hidden text file (".flag7.txt") in the root directory, indicating a potential information disclosure issue.

Images	 
Affected Hosts	192.168.13.10
Remediation	• **Patch and Update Vulnerable Software**: Apply the latest security patches and updates to Apache Tomcat and other relevant software. • **Vulnerability Management:** Establish a robust vulnerability management program, including regular scanning, assessment, and remediation of vulnerabilities. This helps ensure that all identified vulnerabilities are addressed promptly. • **Intrusion Detection and Monitoring:** Deploy intrusion detection systems (IDS) and implement proper monitoring to detect and respond to unauthorized access or malicious activities. Regularly review logs and network traffic to identify suspicious behavior.

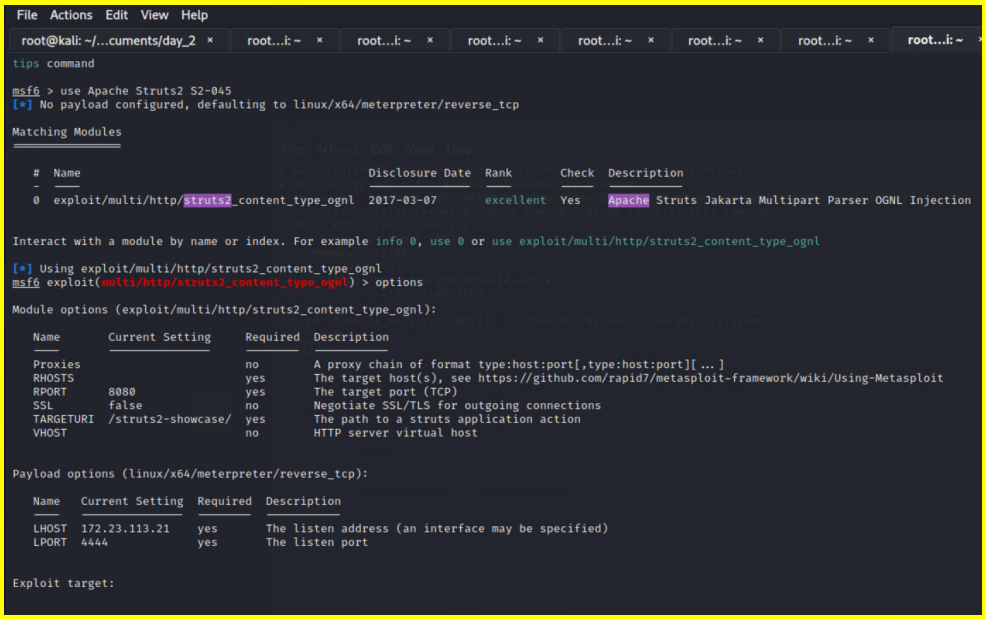
Vulnerability 14	Findings
Title	Remote Code Execution (RCE) Exploit and Privilege Escalation
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	Flag 8 The host ending with 192.168.13.11 was targeted using the Shocking exploit, an RCE (Remote Code Execution) technique, through Metasploit. The specific

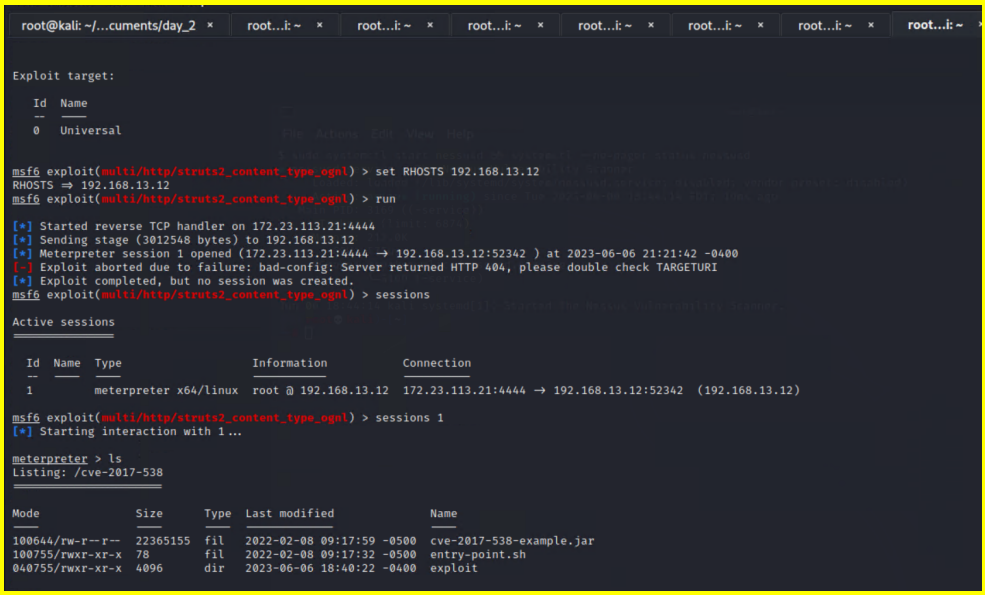
	exploit leveraged was the "Shocking" exploit, with the TARGETURI option set to "/cgi-bin/shockme.cgi". The successful execution of the exploit granted unauthorized access to the host, providing an entry point for further exploration. A search was performed on the server to locate Flag 8. Additionally, an examination of the sudo privileges was conducted. During this exploration, Flag 8 was discovered concealed within the sudoers file. The ease of escalating privileges in this scenario highlights the vulnerability present in the system, underscoring the importance of implementing appropriate remediation measures.
Images	
Affected Hosts	192.168.13.11
Remediation	• **Patch and Update Vulnerable Software:** Apply the latest security patches and updates to the affected software and components. Regularly monitor for security advisories and address known vulnerabilities promptly to prevent unauthorized access and exploitation. • **Privilege Escalation Monitoring:** Implement monitoring mechanisms to detect and alert on potential privilege escalation attempts. Regularly review system logs and use intrusion detection systems to identify suspicious activities or privilege abuse.

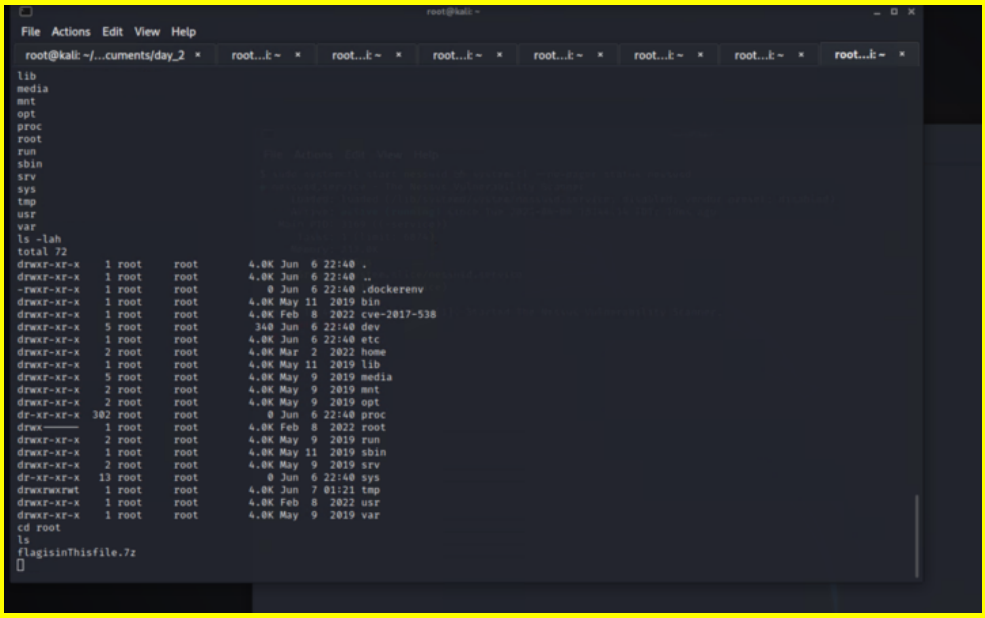
Vulnerability 15	Findings
Title	Flag Discovery in /etc/passwd File and File Integrity Protection

Type (Web app / Linux OS / WIndows OS)	Linux Os
Risk Rating	Medium
Description	Flag 9 Upon gaining access to the system and conducting a thorough exploration of folders and files, the flag 9 was discovered within the /etc/passwd file. The /etc/passwd file is a system database that stores important user account information, including usernames and encrypted password hashes. The discovery of the flag 9 within the /etc/passwd file emphasizes the importance of protecting sensitive system files and implementing appropriate access controls. Safeguarding the integrity of critical files, such as /etc/passwd, is crucial for maintaining the security and confidentiality of user account information.
Images	
Affected Hosts	192.168.13.11
Remediation	• **File Permissions and Access Controls:** Review and adjust the permissions of system files, including the /etc/passwd file, to ensure that only authorized users and processes have the necessary read and write access. • **Analyze log data** regularly to identify any suspicious activities or unauthorized access attempts. • **User Account Management:** Regularly review and update user accounts and their associated privileges. Remove unnecessary or inactive accounts to minimize the attack surface and potential risks associated with compromised accounts.

Vulnerability 16	**Findings**
Title	Successful Exploitation of Apache Struts RCE Vulnerability

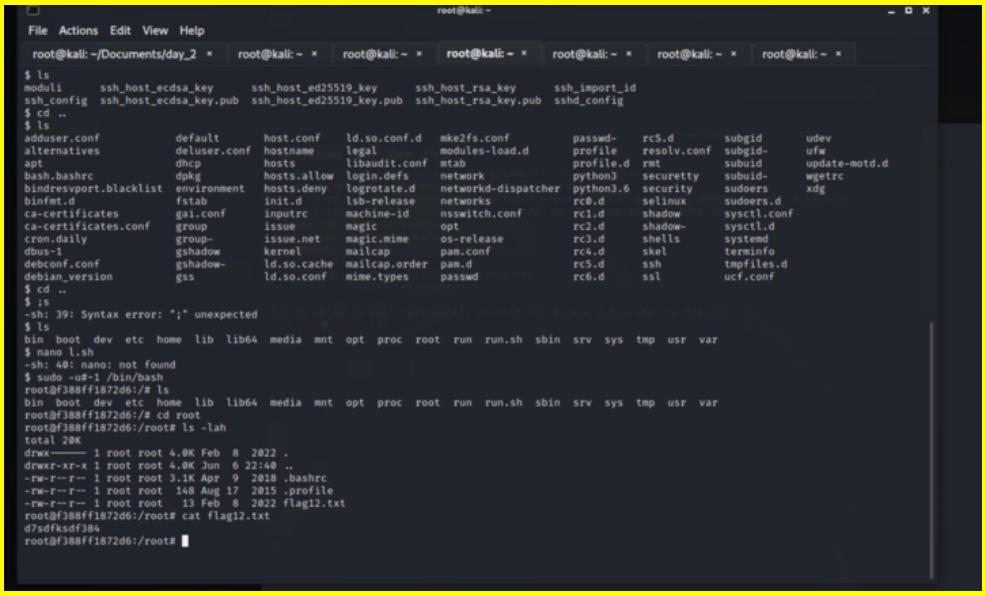
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	Flag 10 The host with the IP address 192.168.13.12 was targeted using an RCE (Remote Code Execution) exploit through Metasploit. Based on the previous research, it was determined that the target host is running Apache Struts 2.3.5-2.3.31 Jakarta Multipart, and the recommended exploit in Metasploit was multi/http/struts2_content_type_ognl. By configuring the RHOSTS parameter as 192.168.13.12 and establishing a session with the LHOST set as 172.22.117.21, successful exploitation was achieved, granting access to the system. To confirm access, the "ls" command was executed to list the contents of the current directory. Subsequently, the "cd /root" command was used to navigate to the root directory. Within this directory, the "cat" command was employed to display the contents of the file "flagisinThisfile.7z", which contained flag 10. This successful exploitation and flag retrieval highlight the severity of the Apache Struts RCE vulnerability and the potential impact it can have on the target system.
Images	





Affected Hosts	192.168.13.12
Remediation	• **Patch and Update Apache Struts:** Apply the latest security patches and updates to Apache Struts. Regularly monitor for security advisories and promptly address known vulnerabilities to prevent unauthorized access and exploitation. • **Intrusion Detection and Monitoring:** Implement robust intrusion detection and monitoring systems to detect and respond to potential attacks targeting Apache Struts or exploiting its vulnerabilities. Regularly review logs and network traffic to identify suspicious behavior.

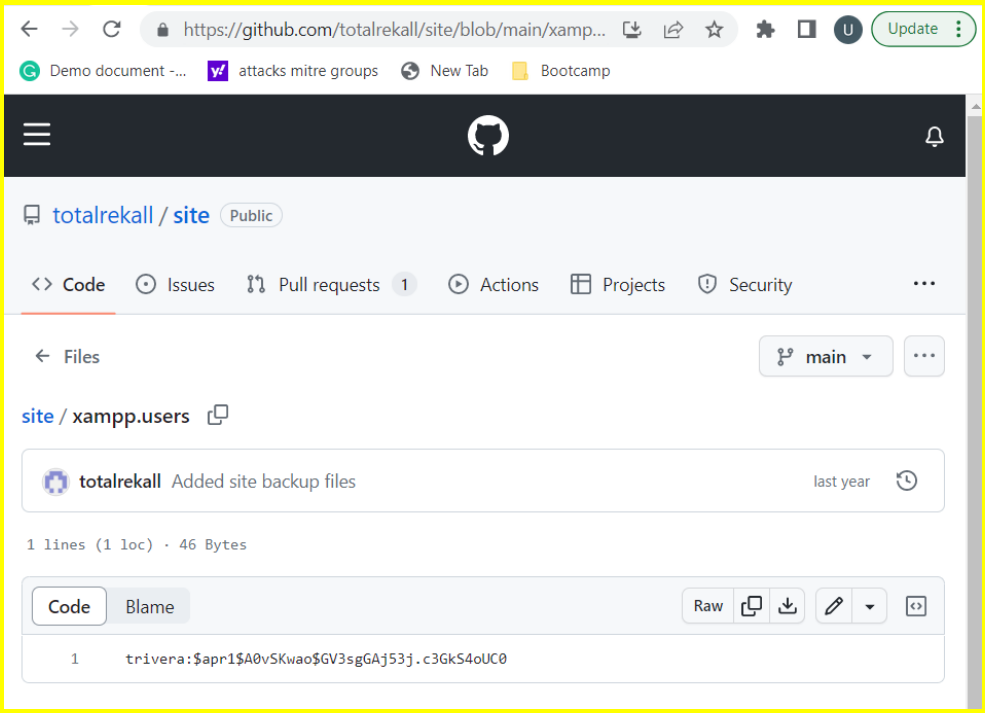
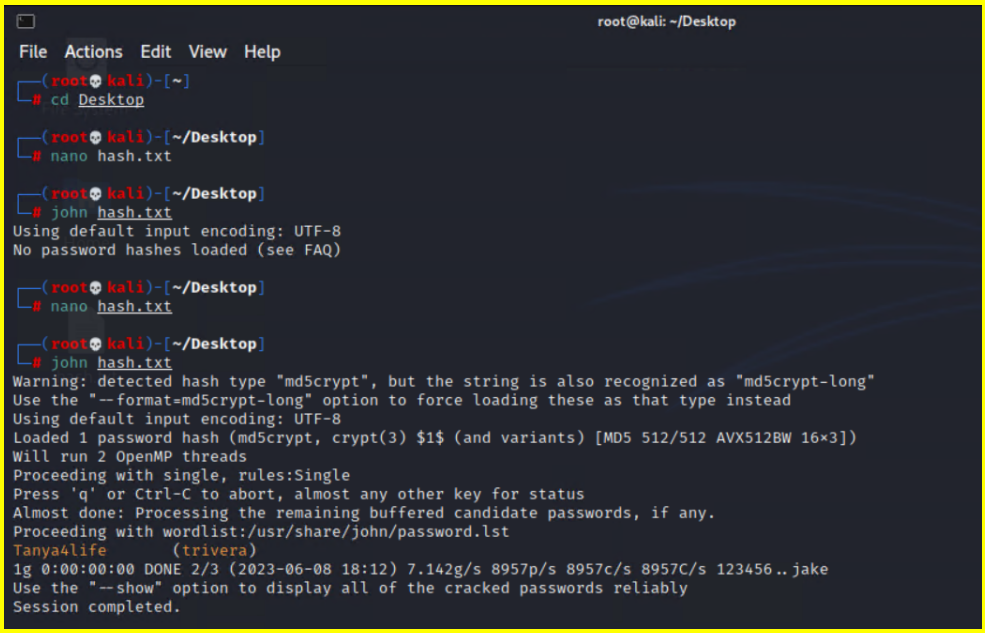
Vulnerability 17	Findings
Title	Successful Exploitation, Privilege Escalation on 192.168.13.14

Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	Flag 12 The host ending with 192.168.13.14 was successfully exploited using an exploit that did not rely on a specific CVE. The clue obtained from viewing Flag 1 provided the username "alice," which belonged to the register officer. Armed with this information, an attempt was made to guess the password to gain access to the host. Upon successfully accessing the host using the credentials "alice" for both the username and password via SSH at 192.168.13.14, a privilege escalation vulnerability was leveraged to escalate privileges and obtain the final flag. Surprisingly easy-to-guess credentials were used for authentication. By executing the command "sudo -u#-1 /bin/bash," the privileges were escalated to the root user. With root access, the system was explored, and Flag 12 was discovered. This was achieved by listing the contents of the directory using "ls," navigating to the "/root/" directory with "cd," and finally displaying the contents of the "flag12.txt" file using "cat." The successful exploitation, privilege escalation, and flag retrieval highlight the significance of strong and secure authentication practices, as well as the importance of addressing privilege escalation vulnerabilities.
Images	
Affected Hosts	192.168.13.14
Remediation	• **Strong Authentication:** Enforce the use of strong, unique passwords for all user accounts. Implement multi-factor authentication (MFA) to provide an additional layer of security. • **Password Management**: Implement a robust password management policy, including regular password rotation and avoiding the use of easily guessable passwords. Consider using password management tools or solutions to securely store and manage passwords.

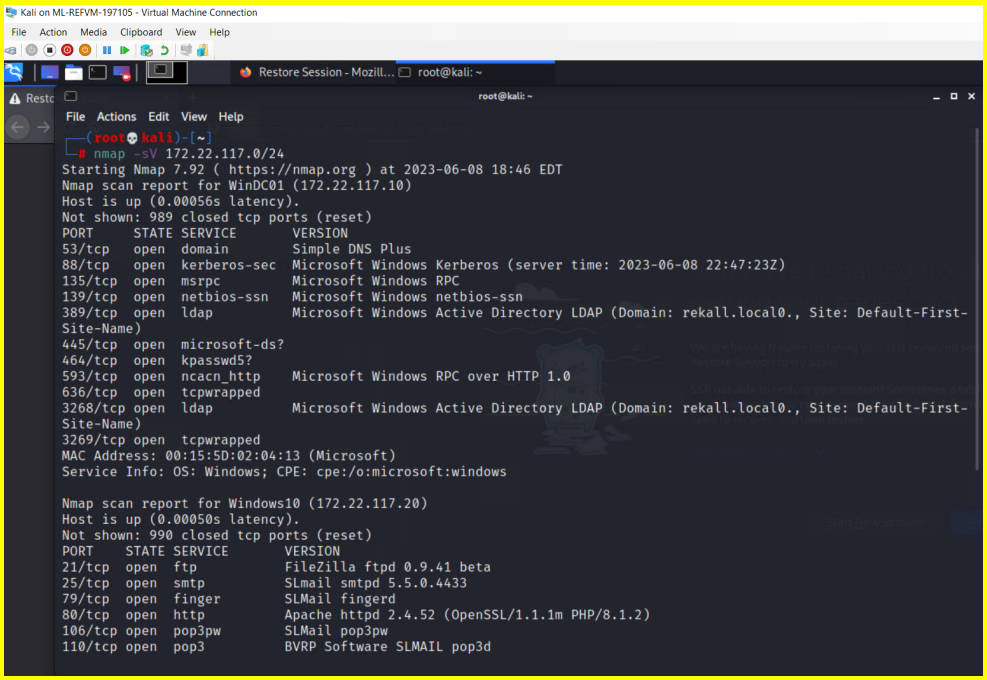
	• **Principle of Least Privilege:** Follow the principle of least privilege by granting users only the permissions necessary to perform their tasks. Avoid granting unnecessary administrative privileges to reduce the impact of potential privilege escalation.

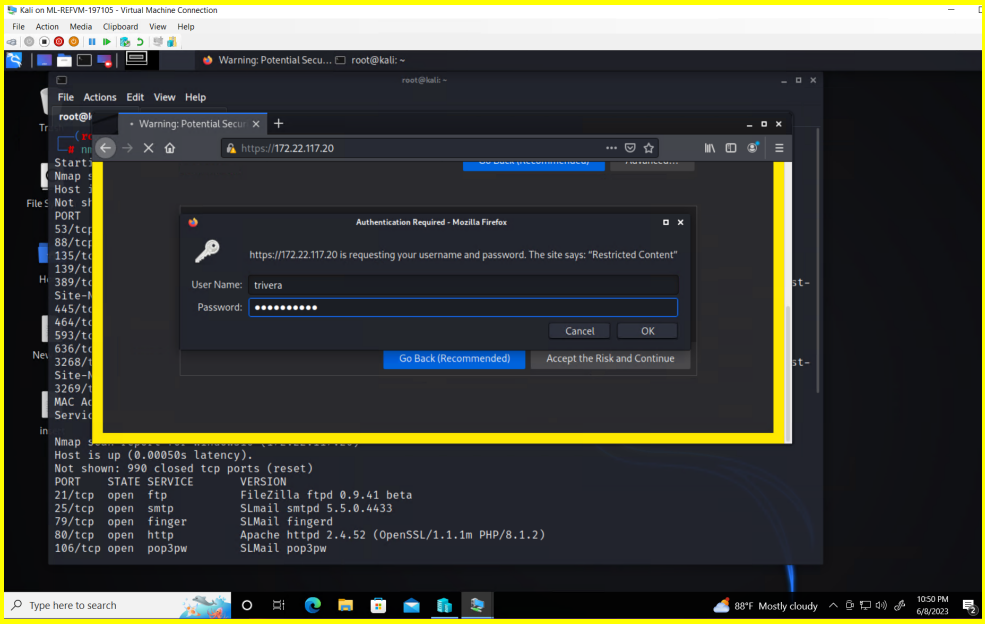
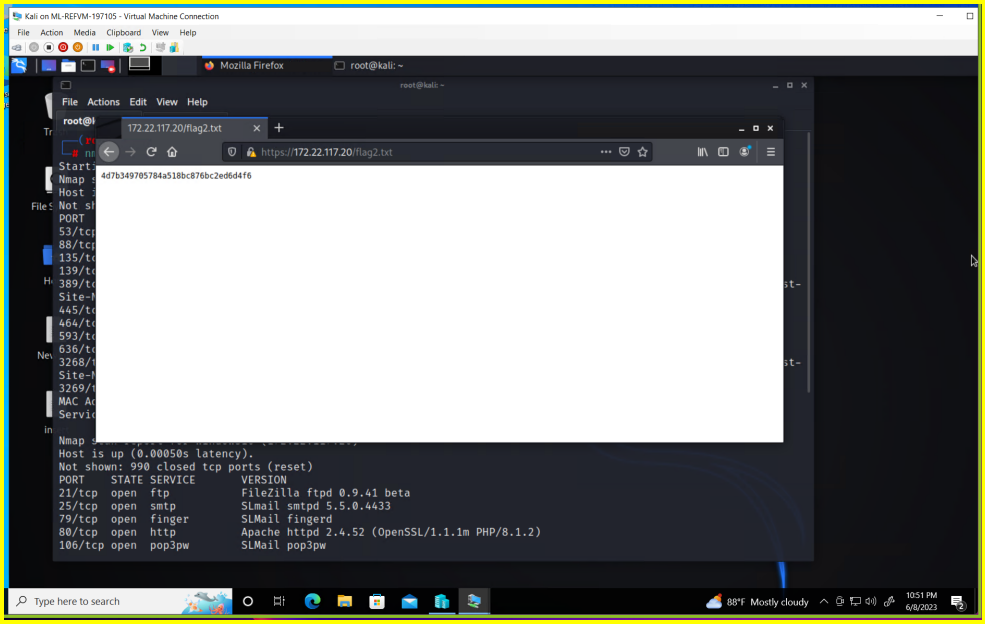
DAY 3

Vulnerability 18	Findings
Title	Successful Password Decryption and System Access in Windows OS
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	Flag 1 During the third day of the penetration testing engagement, the focus was shifted to Windows operating system machines. The initial step involved conducting OSINT activities, specifically exploring the GitHub repository for "totalrekall." Through this process, credentials related to a user named "triviera" and potential login hashes were successfully obtained. To decrypt the obtained hashes, the powerful password cracking tool "John the Ripper" was employed. Leveraging the capabilities of this tool, the password was successfully decrypted, granting unauthorized access to the system. This successful password decryption and subsequent system access highlight the vulnerabilities associated with weak or improperly stored passwords, emphasizing the importance of robust password management practices.

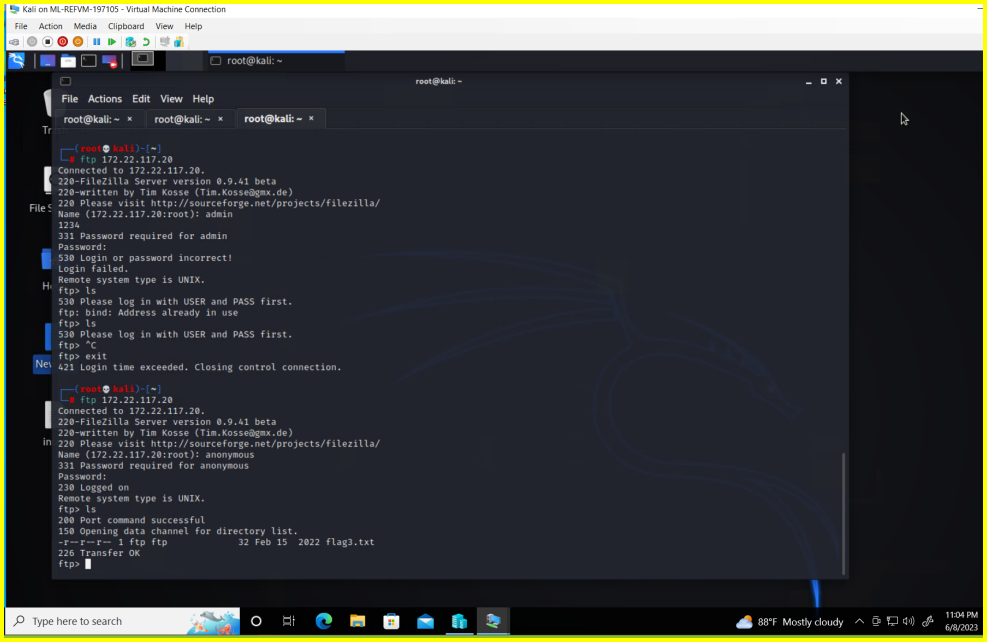
Images	 
Affected Hosts	172.22.117.20
Remediation	**Password Storage:** Store passwords securely using strong cryptographic hashing algorithms and salting techniques. Avoid storing passwords in plaintext or using weak encryption methods that can be easily cracked.

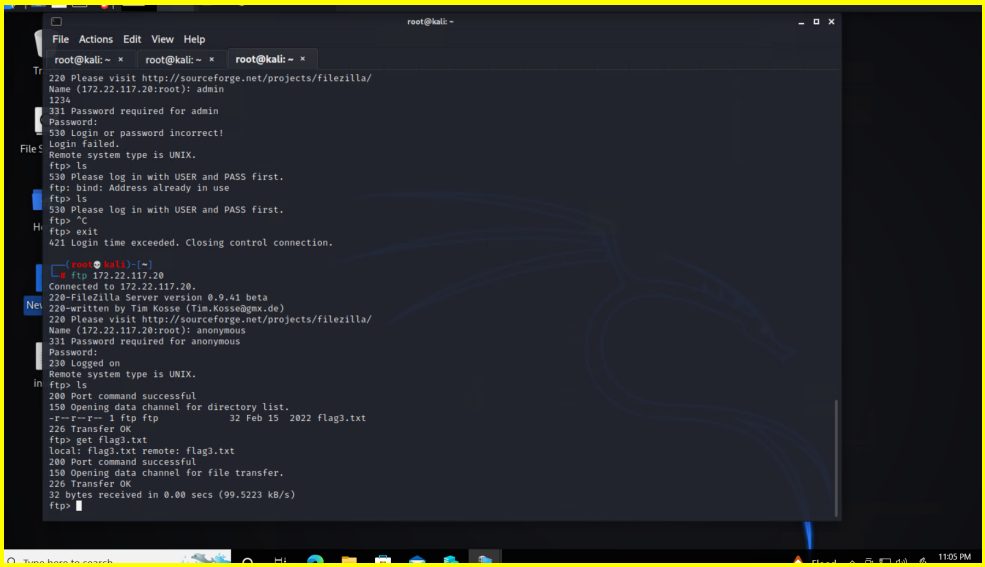
Vulnerability 18	Findings
Title	Successful System Access and Flag Retrieval through HTTP Server

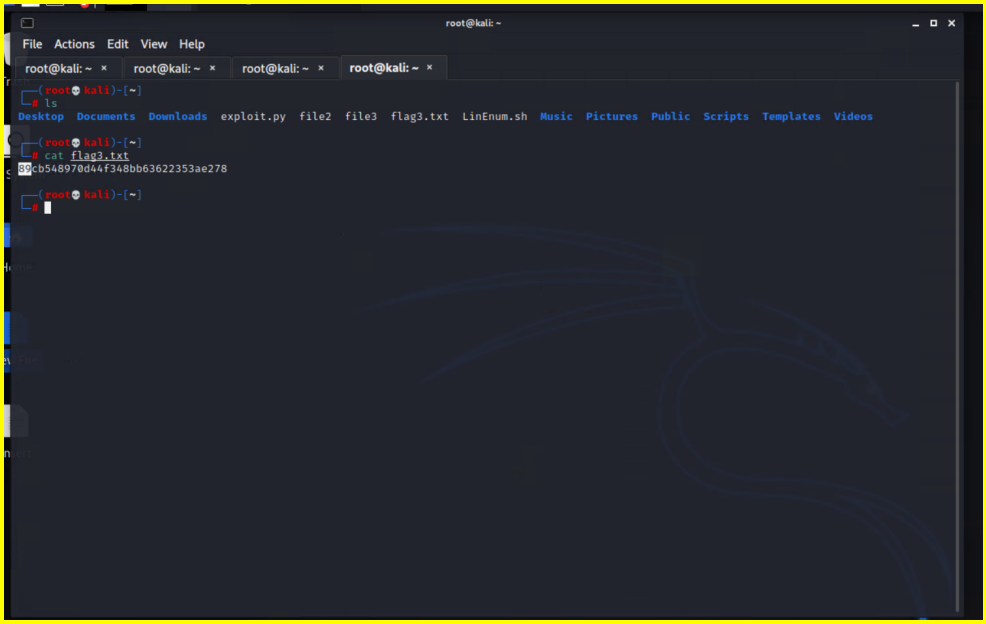
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	Flag 2 After completing the password decryption process, the next phase involved enumeration to identify potential avenues for gaining access to the system. With the acquired credentials, an Nmap scan was performed against the IP range 172.22.117.0/24, specifically targeting the Windows OS machines. Based on the results obtained from the Nmap scan, it was determined that accessing the HTTP server using the IP address 172.22.117.20 would be a promising path. Utilizing the obtained credentials, authentication was attempted on the HTTP server using the provided IP address. Consequently, successful entry into the system was achieved, allowing access to a file containing the second flag. This successful system access and flag retrieval emphasize the importance of thorough enumeration and the significance of properly securing web servers.
Images	

	 
Affected Hosts	172.22.117.20
Remediation	• **Vulnerability Management:** Regularly perform vulnerability scans and assessments to identify and address security weaknesses in the HTTP server and associated systems. • **Secure Authentication:** Implement strong authentication mechanisms on the HTTP server, such as multi-factor authentication (MFA) or strong password policies. Enforce the use of complex passwords and educate users about password security.

Vulnerability 19	Findings
Title	Open Port FTP Access and Flag Retrieval

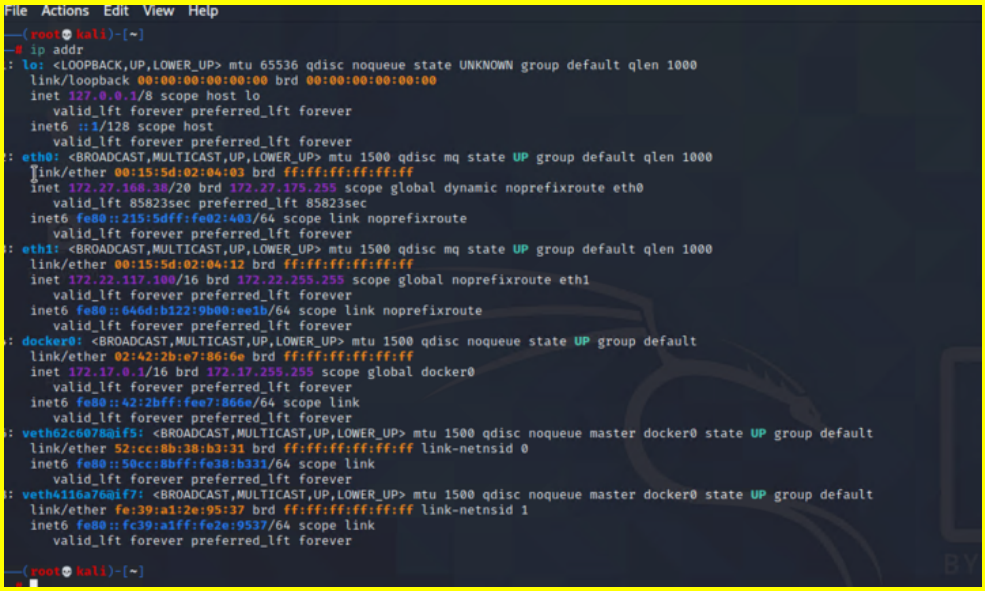
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Flag 3 Upon discovering that port 21 on the IP address 172.22.117.20 was open, indicating the presence of an FTP service, the decision was made to explore this avenue to gain access to the system. By executing the command "ftp 172.22.117.20," a login prompt was presented, but unfortunately, the previously obtained credentials did not work. To overcome this setback, common variations of usernames and passwords were attempted. After a few attempts, successful login was achieved using the username "anonymous" and the password "password." Once inside the system, a search was conducted within the home folder to locate the third flag, which was successfully found. This successful FTP access and flag retrieval highlight the importance of secure FTP configurations and the potential risks associated with weak or default credentials.
Images	

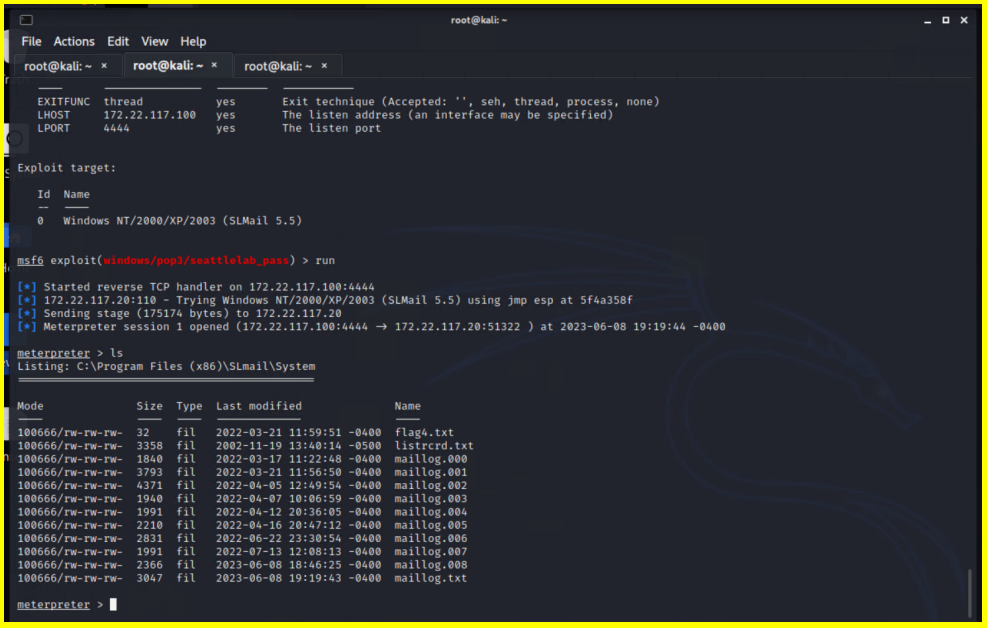


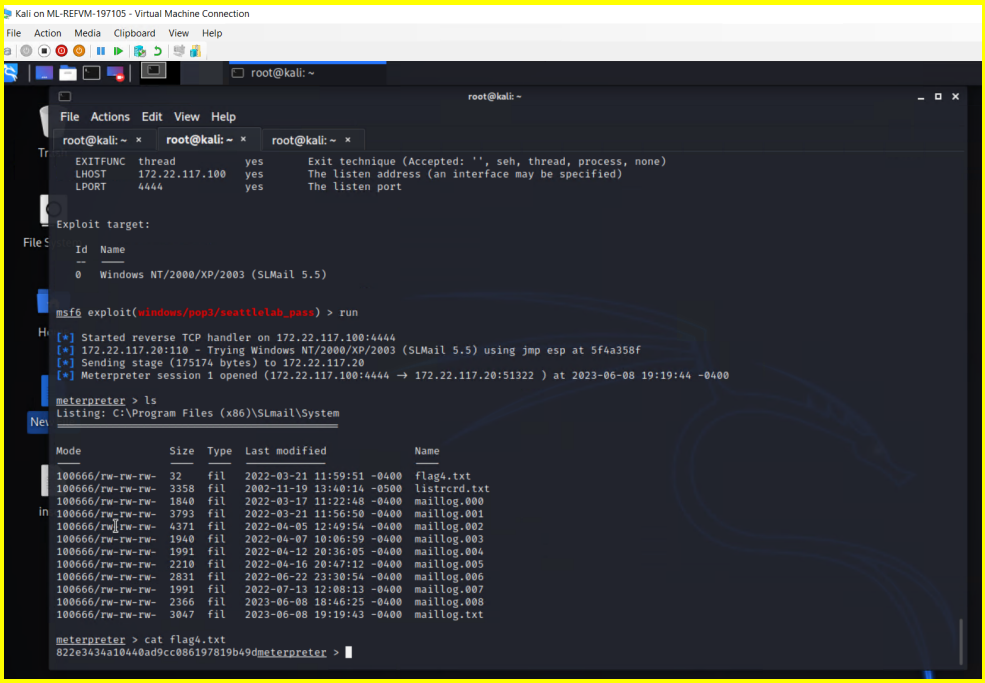


Affected Hosts	172.22.117.20
Remediation	1. **Secure FTP Configurations:** Consider implementing secure FTP protocols, such as FTPS (FTP over SSL/TLS) or SFTP (SSH File Transfer Protocol), which provide encryption for data transmission. Disable anonymous FTP access unless explicitly required. 2. **Regular Credential Audits:** Conduct periodic audits of FTP user accounts and passwords to identify weak or compromised credentials. Promptly deactivate or reset any accounts with suspicious or unauthorized access. 3. **FTP Security Best Practices:** Follow industry best practices for securing FTP servers, including the proper configuration of firewalls, intrusion detection systems, and intrusion prevention systems to monitor and protect against FTP-related attacks.

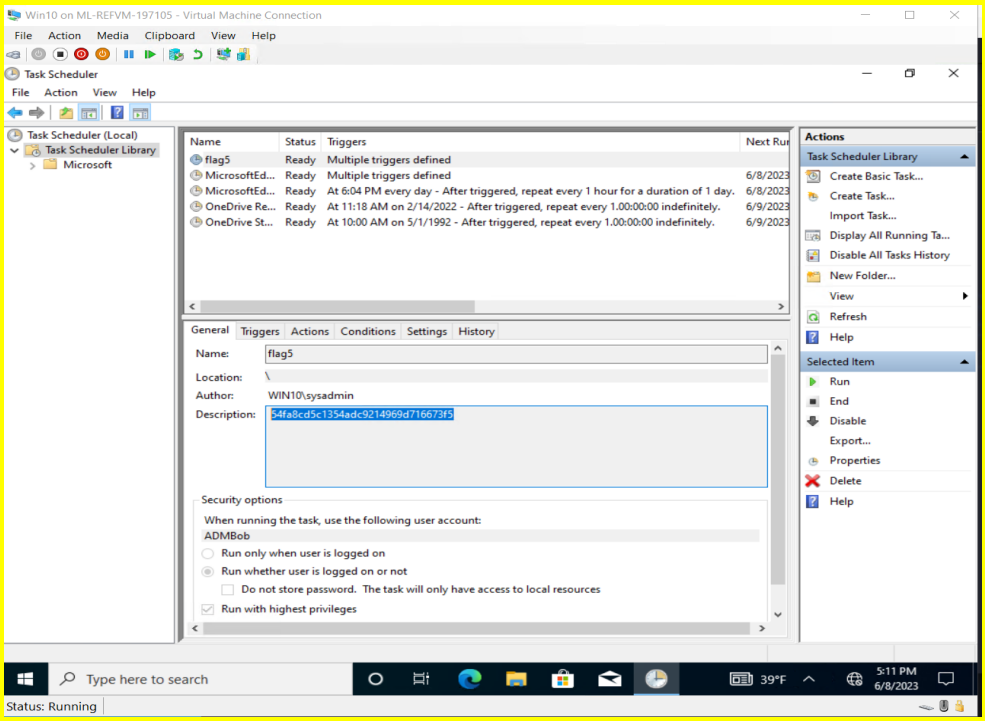
Vulnerability 20	Findings

Title	Successful Exploitation of SLMail Services and Flag Retrieval
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	Flag 4 822e3434a10440ad9cc08619b49d During the penetration testing process, a network scan using Nmap revealed that the system at IP address 172.22.117.20 was running SLMail services. Recognizing the potential vulnerability, the decision was made to exploit this service using the Metasploit framework. After conducting research and making multiple attempts, the specific payload "windows/pop3/seattlelab_pass" within Metasploit was successfully utilized. The exploit was configured by setting the local host (LHOST) to 172.22.117.100 and the target host (RHOST) to 172.22.117.20. This allowed for the exploitation of the system and unauthorized access. During the unauthorized access, the fourth flag was discovered in a text file located at C:\Program Files (x86)\SLmail\System on the compromised system. This successful exploitation and flag retrieval emphasize the importance of securing vulnerable services and the potential risks associated with unauthorized access.
Images	



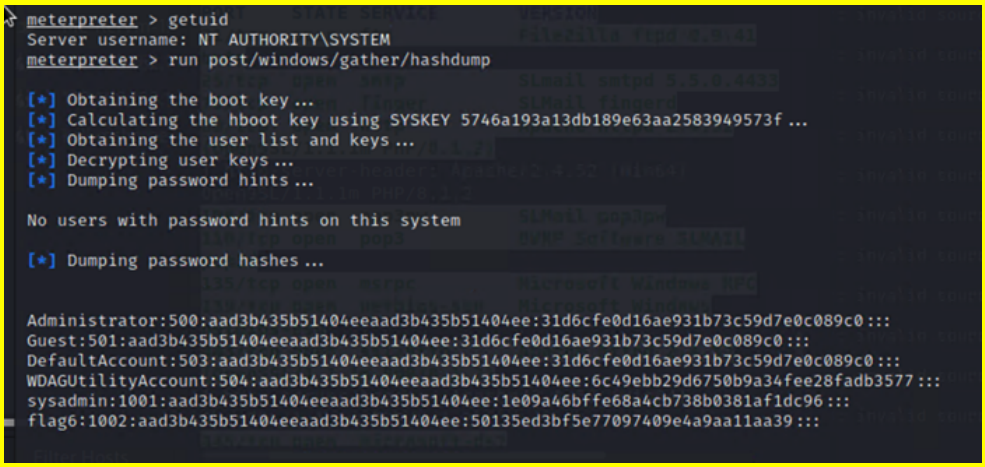
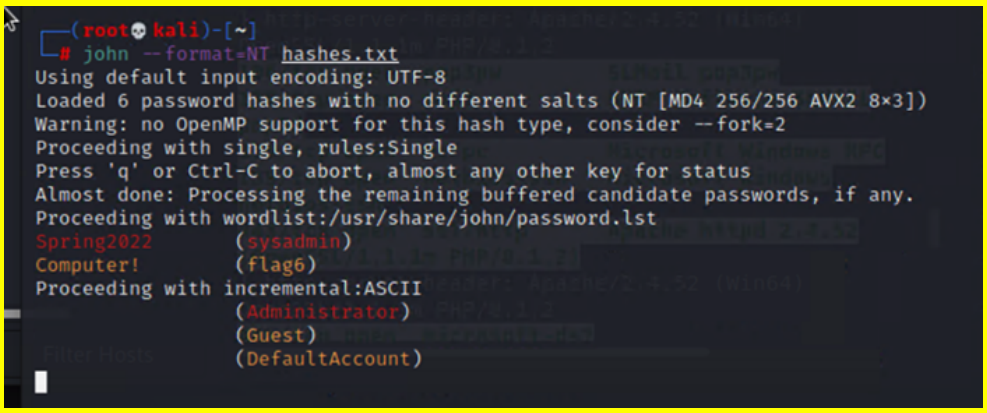
	
Affected Hosts	172.22.117.20
Remediation	• **Patch and Update Services:** Apply the latest security patches and updates to the SLMail services to address known vulnerabilities. Regularly monitor for security advisories related to the software and promptly apply necessary patches. • **Intrusion Detection and Monitoring:** Implement robust intrusion detection and monitoring systems to detect and respond to potential attacks targeting SLMail services or exploiting their vulnerabilities. Regularly review logs and establish alerts for potential security incidents.

Vulnerability 21	Findings
Title	Task Scheduler Investigation and Flag Discovery
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	Flag 5 After gaining unauthorized access to the Windows 10 machine, the investigation focused on the Task Scheduler. The Task Scheduler is a feature that enables the automation and scheduling of various tasks on the system. The objective was to understand its configuration and identify potential vulnerabilities. The Task Scheduler can be exploited by adversaries to schedule malicious activities or payloads, making it crucial to thoroughly examine its configuration.

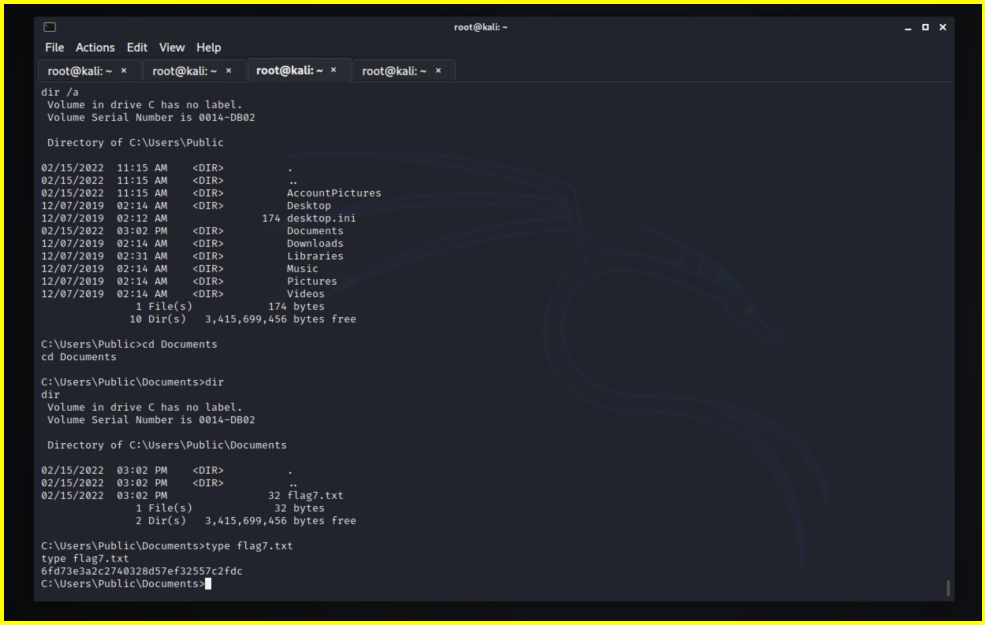
	This helps in identifying any unauthorized or suspicious tasks that may have been created. During the investigation, a hidden flag5 was discovered within the Task Scheduler. This flag serves as evidence of successful penetration testing activities. It is essential to address any vulnerabilities or suspicious tasks found in the Task Scheduler to mitigate the risk of unauthorized activities and maintain the security of the system.
Images	
Affected Hosts	172.22.117.20
Remediation	• **Regular Task Scheduler Audit:** Conduct regular audits of the Task Scheduler to identify any unauthorized or suspicious tasks. Review and analyze the configuration and history of tasks for any anomalies or potential security issues. • **Remove Unnecessary or Suspicious Tasks:** Disable or delete any tasks that are unnecessary, unfamiliar, or suspicious. Regularly review and validate the tasks to ensure that only authorized and legitimate tasks are scheduled.

Vulnerability 22	Findings
Title	Password Retrieval and Decryption from Compromised System
Type (Web app / Linux OS / WIndows OS)	Windows OS

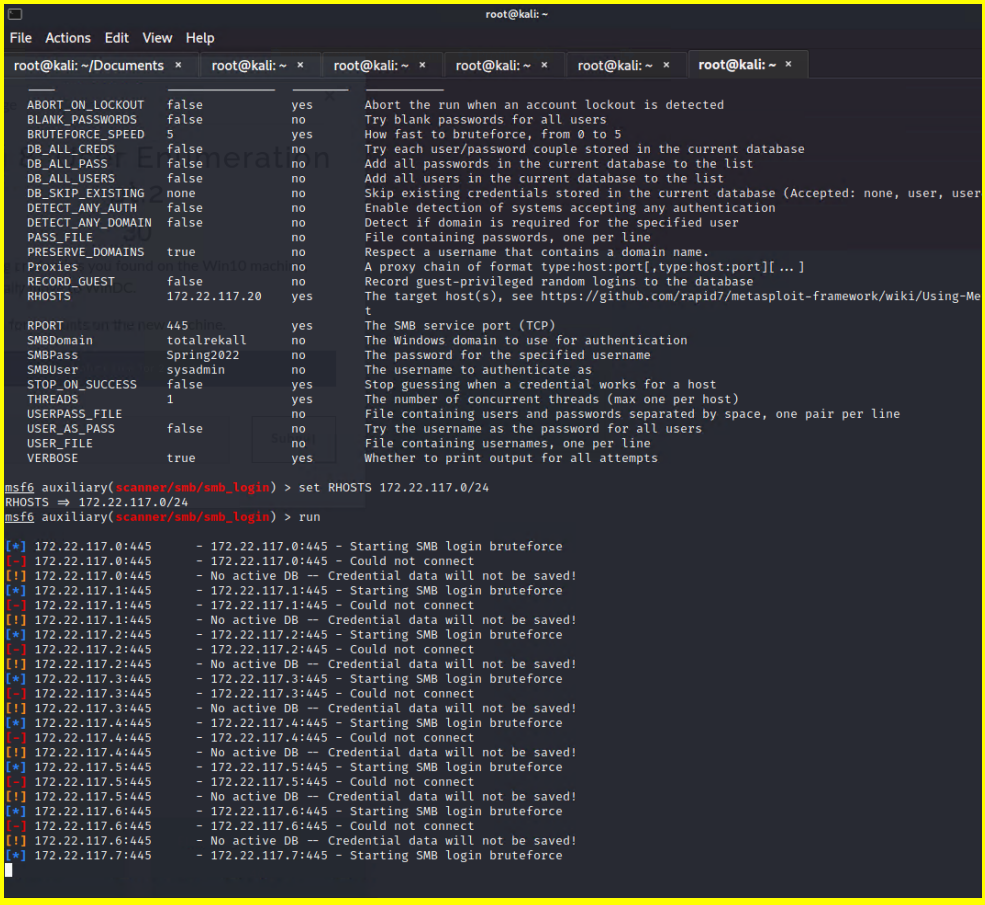
Risk Rating	High
Description	Flag 6 Following the successful exploit using Metasploit and establishing the Meterpreter session, the focus shifted towards obtaining further access and information from the compromised system. The objective was to retrieve passwords from a vulnerable password file stored on the compromised system. To achieve this, the post-exploitation module "post/windows/gather/hashdump" within the Meterpreter session was utilized. This module facilitated the dumping of password hashes stored on the system. Subsequently, the password cracking tool "John the Ripper" was employed to decrypt the password hashes, enabling the recovery of plain-text passwords for the user accounts "sysadmin" (password: Spring2022) and "flag6" (password: Computer!). The successful retrieval and decryption of passwords highlight the significance of implementing strong password policies and secure password storage mechanisms.
Images	

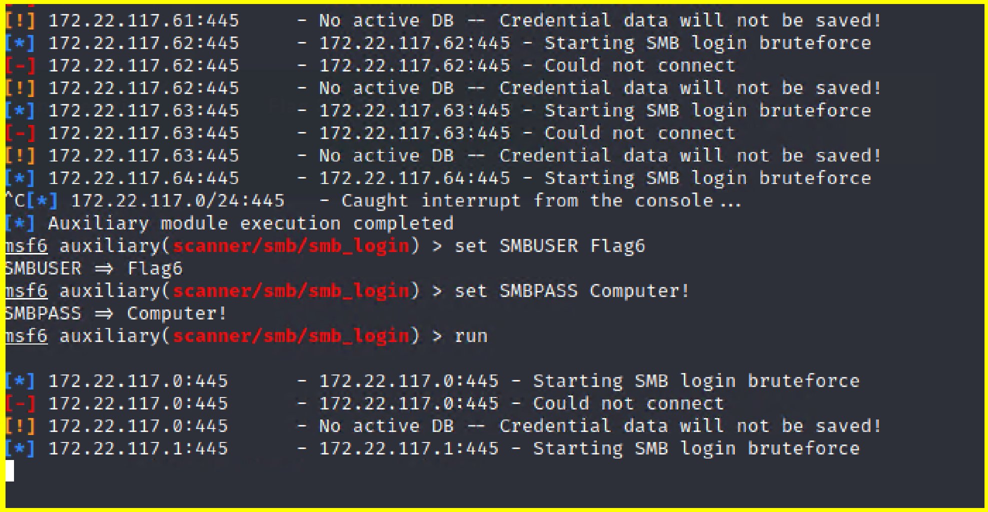
	 
Affected Hosts	172.22.117.20
Remediation	• **Strong Password Policies:** Enforce strong password policies that include a combination of uppercase and lowercase letters, numbers, and special characters. Encourage regular password updates and educate users about password security best practices. • **Secure Password Storage:** Implement secure password storage mechanisms that use strong cryptographic hashing algorithms with unique salts. Avoid storing passwords in plaintext or using weak encryption methods that can be easily cracked. • **Multi-Factor Authentication (MFA):** Implement MFA to add an additional layer of security, making it more challenging for attackers to gain unauthorized access even with compromised passwords.

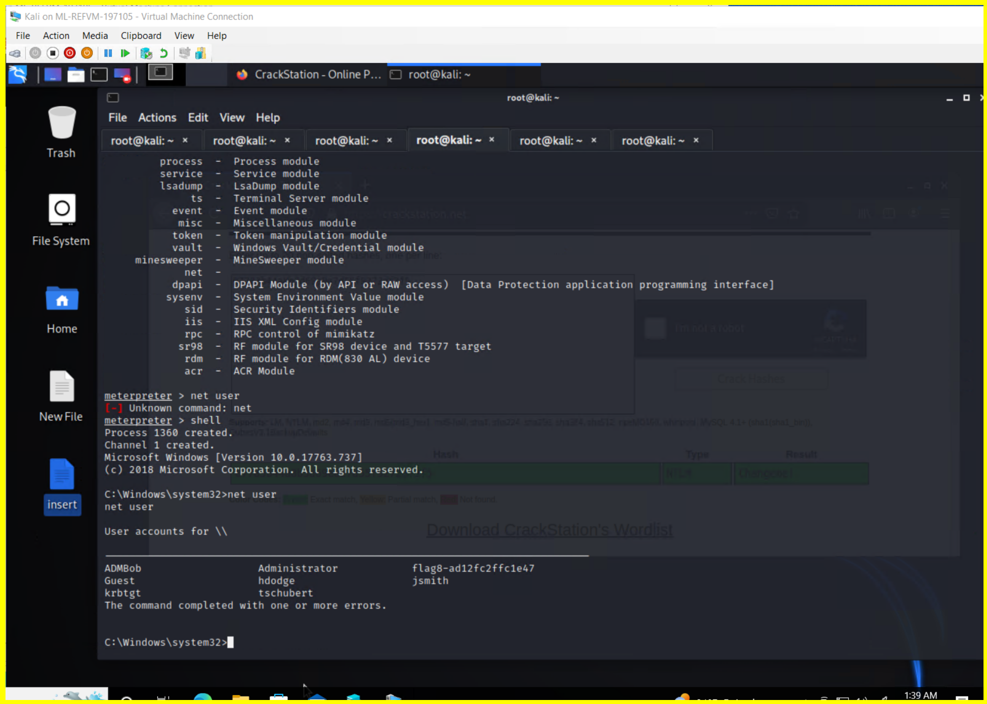
Vulnerability 23	Findings
Title	Flag Discovery in Exposed Public Folder
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Low
Description	Flag 7

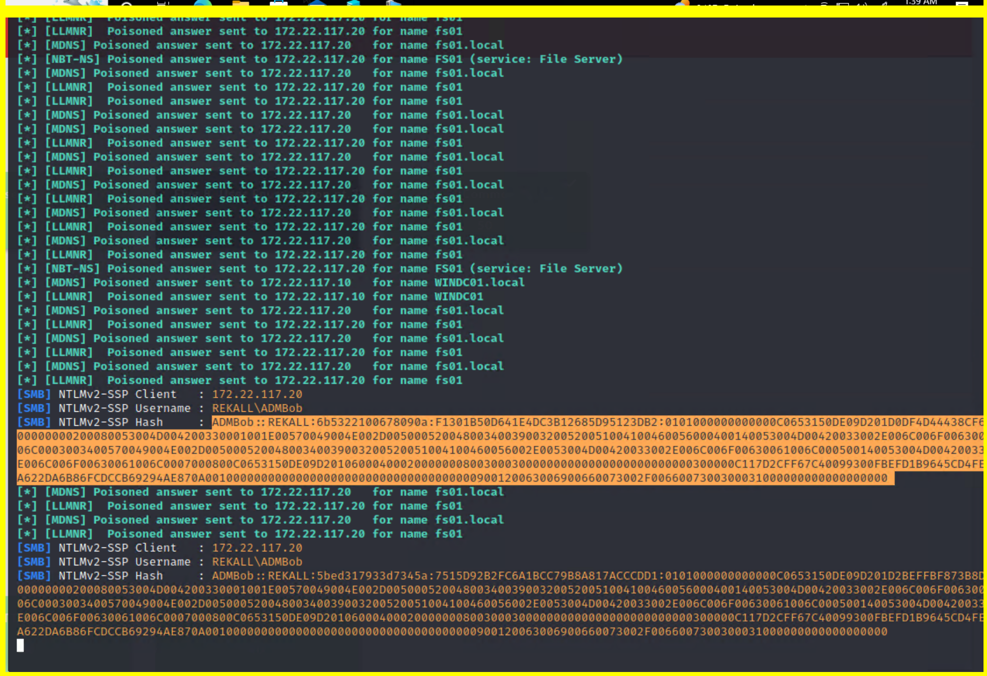
	During the exploitation of the system, further investigations were conducted to locate additional sensitive files. As a result, a hidden flag was discovered within the Public folder. The flag was stored in a text file named flag7.txt. The discovery of the flag emphasizes the importance of securing sensitive files and directories, such as the Public folder, to prevent unauthorized access and maintain the confidentiality of sensitive information.
Images	
Affected Hosts	172.22.117.20
Remediation	1. **User Awareness:** Educate users about the importance of data security and the risks associated with storing sensitive information in public or accessible directories. 2. **File Encryption:** Implement file-level encryption for sensitive files to protect their contents even if unauthorized access occurs. Encryption ensures that data remains secure, even if physical or logical access is compromised.

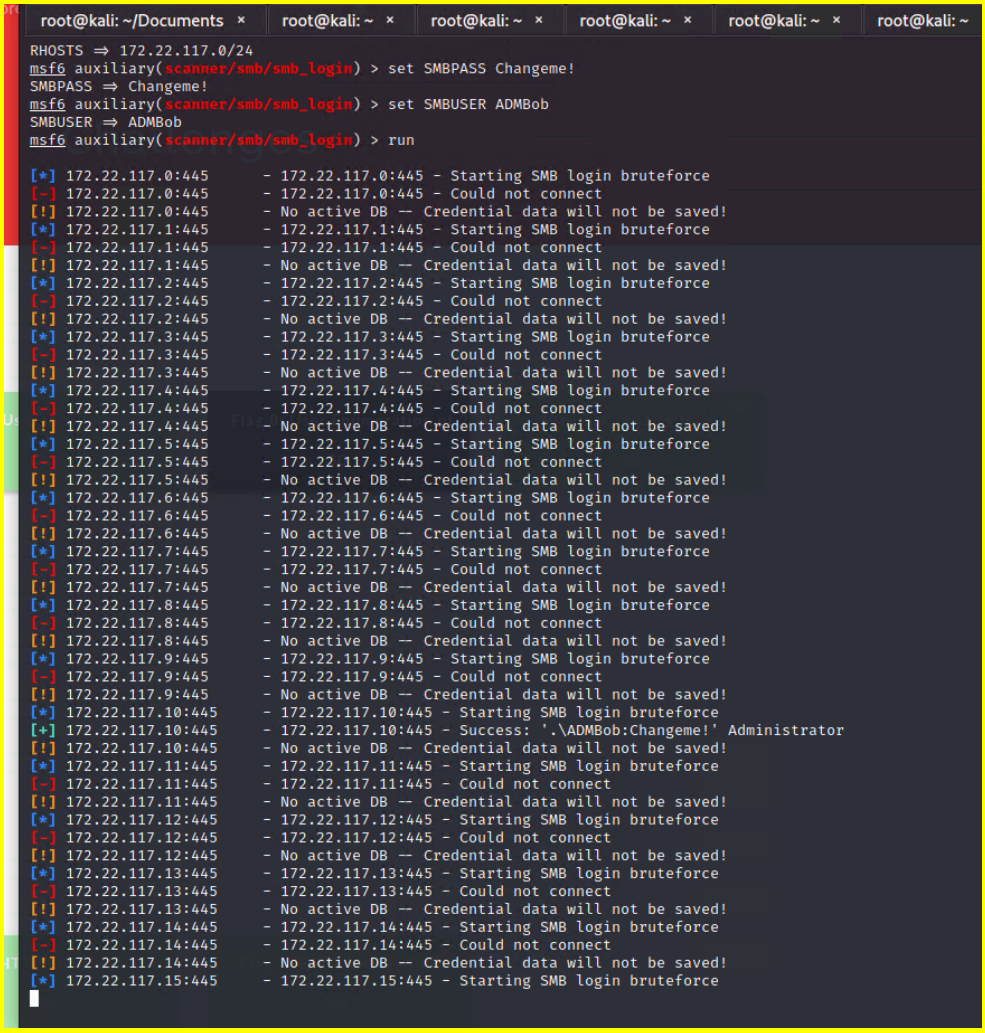
Vulnerability 24	Findings
Title	Lateral Movement and Password Cracking for WinDC Access
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	Flag 8 Using the obtained credentials from the Win10 machine, the focus shifted to performing lateral movement to the WinDC (Windows Domain Controller)

	system. The objective was to gain unauthorized access to this system using the available credentials. To accomplish this, the "auxiliary/scanner/smb/smb_login" payload within the framework was utilized. The credentials of the user "sysadmin" with the password "Spring2022" and the user "Flag6" with the password "Computer!" were provided. Through the scanning process, it was determined that the credentials of the user "Flag6" with the password "Computer!" allowed for successful access to the WinDC system. After gaining access, the Responder tool was set up to capture network traffic and listen for any exchanged credentials. During this process, Responder detected the credentials of a user named "ADMBob" along with their corresponding password hashes. Exploiting this information, the password-cracking tool "John the Ripper" was utilized to crack the password hashes associated with the "ADMBob" user. By successfully cracking the hashes, unauthorized access to the plaintext password of the "ADMBob" user was obtained. This lateral movement and password-cracking process highlight the significance of secure credential management, the potential risks of weak or compromised passwords, and the importance of implementing robust security controls to prevent unauthorized access.
Images	

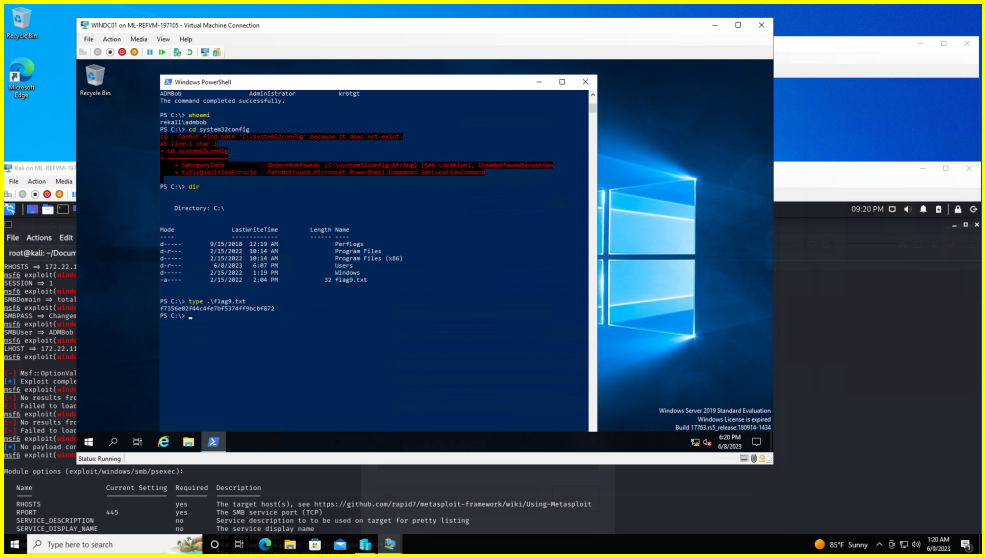






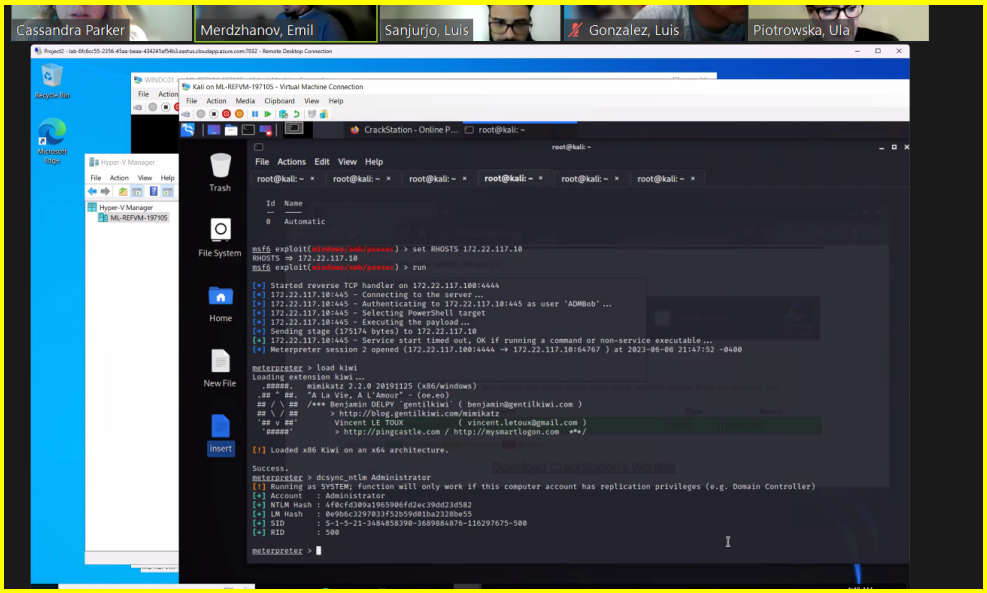


Affected Hosts	172.22.117.20
Remediation	• **Strong Password Policies:** Enforce strong password policies throughout the network, including the use of complex passwords, regular password updates, and avoiding the use of easily guessable passwords. • **Multi-Factor Authentication (MFA):** Implement MFA to add an extra layer of security, making it more challenging for attackers to gain unauthorized access even with compromised passwords.

Vulnerability 25	Findings
Title	System Exploration and Flag Discovery in "C:\system32\config"
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	Flag 9 After obtaining the credentials for the user "ADMBob" and successfully cracking his password, a scan of the system was conducted to determine the extent of his login privileges. Utilizing the "auxiliary/scanner/smb/smb_login" module, it was discovered that the user "ADMBob" possessed Administrator privileges on the machine with the IP address 172.22.117.10. Armed with this valuable information, login was initiated to the host at IP address 172.22.117.10 using the credentials of "ADMBob". Once successfully logged in, the system exploration process began to locate the elusive flag9, believed to be concealed in a critical location. The investigation led to the heart of the system, specifically the "C:\system32\config" directory. Within this directory, the presence of a text file named flag9.txt was identified, which contained the sought-after flag. This successful system exploration and flag discovery highlight the importance of securing critical system directories and the significance of privileged account management.
Images	
Affected Hosts	172.22.117.20
Remediation	• **System Directory Protection:** Implement access controls and file integrity monitoring on critical system directories, such as "C:\system32\config", to prevent unauthorized modifications or access to sensitive files.

	• **Regular System Audits:** Conduct regular audits and file system checks to detect any unauthorized changes or files within critical system directories. Promptly investigate and address any suspicious findings. • **Network Segmentation:** Implement network segmentation to limit lateral movement and contain potential security breaches. Isolate critical systems and directories from less secure areas of the network.

Vulnerability 26	Findings
Title	Administrator Password Hash Retrieval and Unsuccessful
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Low
Description	Flag 10 In the final step of the penetration testing process, the objective was to obtain the password hash for the Administrator account. To accomplish this, the Kiwi module within the Metasploit framework was utilized. Specifically, the "dsync_ntlm" module was loaded in Metasploit, targeting the Administrator account. The goal was to extract the hash associated with the Administrator account, as it provides valuable information for further analysis. While the hash was successfully obtained, unfortunately, cracking the hash to retrieve the plaintext password was unsuccessful. It is important to note that hash cracking can be a complex and time-consuming process, and in this case, the password could not be deciphered from the obtained hash. Cracking hashes requires computational power and the use of appropriate tools and techniques, such as leveraging wordlists or applying advanced password-cracking algorithms. The hash representing the Administrator account serves as the final flag, known as flag10. Although the plaintext password was not retrieved, the presence of the hash emphasizes the importance of securing password hashes and implementing strong password policies to prevent unauthorized access.

Images	
Affected Hosts	172.22.117.20
Remediation	**Password Storage Best Practices:** Implement secure password storage mechanisms that use strong cryptographic hashing algorithms and unique salts. Avoid storing passwords in plaintext or using weak encryption methods that can be easily cracked.

Conclusion

The conducted penetration testing engagement revealed several vulnerabilities, including cross-site scripting (XSS), command injection, SQL injection, unauthorized access, information leakage, and weaknesses in password storage. These findings emphasize the importance of implementing secure coding practices, input validation, access controls, and regular patching. It is crucial to prioritize vulnerability mitigation, strengthen password storage mechanisms, and enhance security awareness and training to fortify the system against potential exploitation. By addressing these vulnerabilities and implementing the recommended remediations, organizations can improve their overall security posture and safeguard against potential cyber threats.