Cybersecurity

Penetration Test Report

[redacted] Corporation
Penetration Test Report

****some names and ips have been changed due to NDA and privacy issues**

Confidentiality Statement

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

Table of Contents

Contact Information

Company Name	Hackatron
Contact Name	s asghar
Contact Title	Pentester

Document History

Version	Date	Author(s)	Comments
001	Nov 21, 2024	s asghar	

Introduction

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

For the testing, we focused on the following:

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

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

Assessment Objective

The primary goal of this assessment was to provide an analysis of security flaws present in [redacted] 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.

[redacted] has outlined the following objectives:

Table 1: Defined Objectives

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

Penetration Testing Methodology

Reconnaissance

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

Identification of Vulnerabilities and Services

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

Vulnerability Exploitation

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

Reporting

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

Scope

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

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

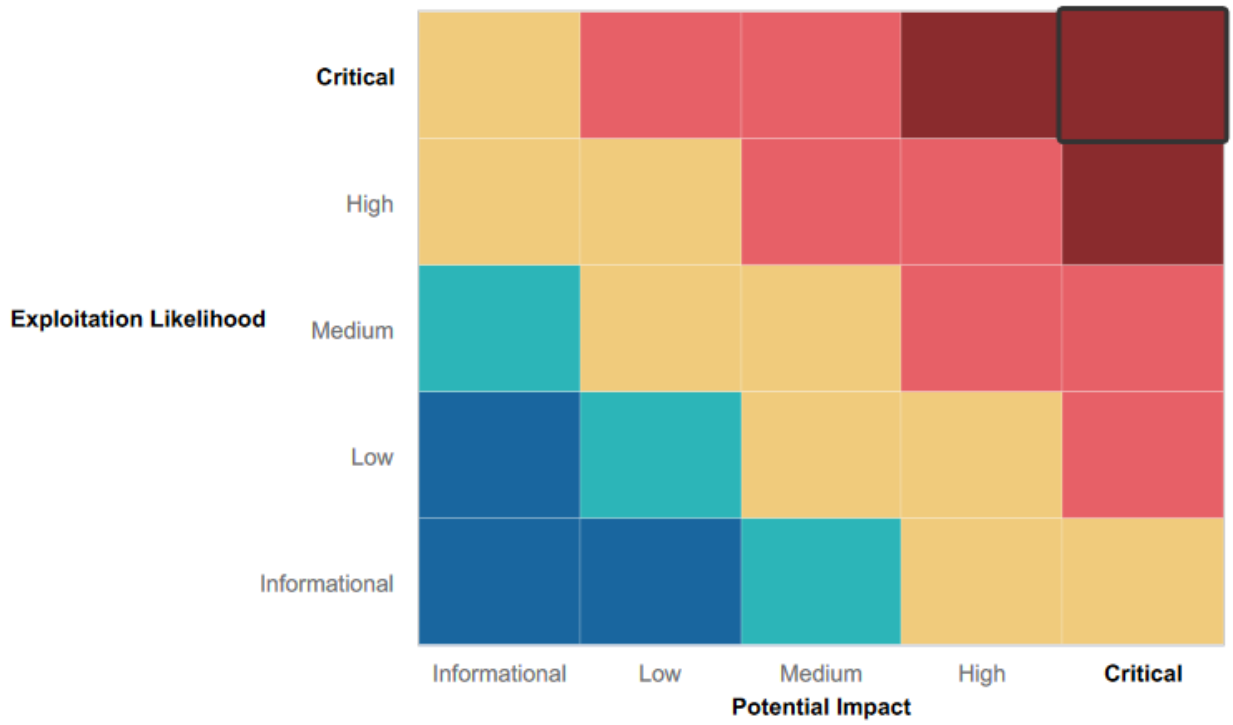
Executive Summary of Findings

Grading Methodology

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

Critical: Immediate threat to key business processes.
High: Indirect threat to key business processes/threat to secondary business processes.
Medium: Indirect or partial threat to business processes.
Low: No direct threat exists; vulnerability may be leveraged with other vulnerabilities.
Informational: No threat; however, it is data that may be used in a future attack.

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



Summary of Strengths

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

High-Level Summary of Strengths:

1. **Comprehensive Network Penetration Testing**:
   - The ethical hacking of Rekall Corporation's network was highly successful across both Linux and Windows servers.
   - Both external (web server) and internal (Linux and Windows) vulnerabilities were identified and exploited.
2. **Effective Use of Tools and Exploits**:
   - A wide range of tools such as Burp Suite, Metasploit, Nmap, and Zenmap were used effectively for scanning, exploiting, and post-exploitation activities.
   - Multiple exploit frameworks (e.g., Metasploit) were utilized to target specific vulnerabilities like **CVE-2017-5638 (Apache Struts)**, **CVE-2019-6340 (Drupal)**, and **CVE-2017-12617 (Apache Tomcat)**.
   - OSINT and domain reconnaissance tools were used to gather WHOIS data, aiding in credential guessing (e.g., Alice username).
3. **Diverse Exploitation Techniques**:
   - Exploits ranged from **SQL Injection** and **XSS** on the web server to **shellshock**, **local file inclusion**, and **remote code execution** on Linux servers.
   - On Windows, Metasploit's **SLMail** exploit, **FTP**, **POP3**, and web services were used to gain access to both the Windows 10 machine and Domain Controller (WinDC01).
4. **Privilege Escalation and Lateral Movement**:
   - Strong post-exploitation techniques, including **hash cracking** and **credential dumping** (e.g., **Kiwi**), were used to escalate privileges and gain system-level access on both Windows and Linux hosts.
   - Lateral movement from **Windows 10** to **WinDC01** was achieved using dumped credentials for **ADMBob**, highlighting effective lateral penetration and access to domain-wide data.
5. **Clear Documentation of Findings**:
   - Flags were clearly numbered and documented throughout the engagement, making it easy to track the exploitation process and identify which vulnerabilities were exploited.

Summary of Weaknesses

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

Summary of Weaknesses:

1. **Over-reliance on Exploitable Public Vulnerabilities**:
   - Many of the successful attacks relied on known public vulnerabilities (e.g., **CVE-2017-5638** and **CVE-2019-6340**) that were not patched by Rekall Corporation.
   - Lack of timely patching or configuration management is a significant weakness, which left systems exposed to easily avoidable attacks.
2. **Limited Security Measures on Web and Server Configurations**:

- Multiple web server vulnerabilities were found, including **XSS** (both reflected and stored), **SQL injection**, and **local file inclusion**, pointing to weak web application security practices.
- Sensitive data exposure, such as leaked credentials in PHP files, indicated inadequate data sanitization and security controls on the web server.

3. **Inadequate Session Management**:
   - The ability to exploit **session IDs** via **Burp Suite** to gain **admin access** to the web application suggests weaknesses in session management and user authentication protocols.
4. **Unsecured Internal Network**:
   - Several internal Linux and Windows servers had critical vulnerabilities (e.g., **CVE-2017-5638**, **CVE-2017-12617**), highlighting a lack of internal segmentation, monitoring, and proper patching.
   - **Shellshock** on one of the Linux servers (192.168.13.11) was a significant issue that allowed access to critical files and flags, demonstrating insufficient internal network defense.
5. **Weak Credential Policies**:
   - The ability to guess the SSH password for user **alice** on server 192.168.13.14 indicates weak password policies.
   - Additionally, **privilege escalation** and **hash cracking** were successful, suggesting weak passwords or poor credential management within the environment.
6. **Poor Configuration of Services**:
   - Services like **FTP**, **SMTP**, **POP3**, and web servers on **Windows 10** were found exposed, providing multiple vectors for exploitation.
   - **SLMail** and other service vulnerabilities were exploited due to poor configuration management and lack of hardening.
7. **Failure to Detect Exploits and Intrusions**:
   - No mention of detection tools or monitoring systems being in place to detect or mitigate these attacks.
   - Given the wide array of vulnerabilities exploited (including some that are well-known), Rekall Corporation's security monitoring appears to be inadequate, allowing the attacks to go undetected.

Executive Summary

Executive Summary:
The objective of this ethical penetration test was to assess the security posture of Rekall Corporation's network, which comprised a mix of Linux and Windows servers. The engagement aimed to identify vulnerabilities in both the internal and external network environments, exploiting weaknesses to gain unauthorized access, elevate privileges, and exfiltrate sensitive information. This assessment provided a comprehensive view of the vulnerabilities present in Rekall's infrastructure and identified critical gaps in their security posture.
Scope of the Assessment: The target network included a range of devices and services: five Linux-based servers (with IPs 192.168.13.10 through 192.168.13.14), a web server (192.168.14.35), and Windows-based systems (WinDC01 and Windows 10 machine). The approach involved scanning the network, identifying active hosts, and exploiting vulnerabilities through a combination of web, network, and system-based attacks.
Step-by-Step Assessment:

1. **Initial Network Discovery**:
    - A comprehensive network scan using **Nmap** revealed the presence of multiple active hosts, including Linux servers, a web server, and Windows machines. The scan also identified open ports and services running on these hosts, providing a clear map of the attack surface.
2. **Web Server Vulnerabilities**:
    - The external-facing web server (192.168.14.35) was found to be highly vulnerable. **XSS**, **SQL Injection**, **Local File Inclusion (LFI)**, and other critical web vulnerabilities were identified.
    - These vulnerabilities were exploited to gain unauthorized access to the web server and retrieve sensitive information, such as credentials stored in PHP files. The use of **Burp Suite** facilitated session ID manipulation, which led to administrative access on the web server.
3. **Exploitation of Linux Servers**:
    - On the internal network, the Linux servers were systematically scanned for vulnerabilities using **Metasploit** and other tools. Notable vulnerabilities such as **Shellshock** and **Struts2** were found and exploited, providing a foothold on several servers.
    - On one of the Linux servers (192.168.13.12), a critical **Apache Struts CVE** was exploited, allowing access to the system and subsequent **privilege escalation**. The discovery of user credentials and password hashes on compromised servers allowed for further access to sensitive data.
4. **Accessing Windows Systems**:
    - The Windows environment, which included a **Domain Controller** (WinDC01) and a **Windows 10 machine**, was found to have multiple open services like **FTP**, **SMTP**, and **HTTP**. These services were targeted through **Metasploit** exploits, and we successfully gained access to both machines.
    - The **SLMail** vulnerability was exploited to gain an initial foothold. Once inside, **privilege escalation** techniques were used to elevate from user to system-level access on the Windows 10 machine.
    - Laterally moving to **WinDC01**, we successfully dumped **Active Directory credentials**, cracked them, and gained full administrative access.
5. **Exfiltration of Sensitive Data**:
    - During post-exploitation activities, various sensitive files, such as password hashes and configuration files, were exfiltrated from both Linux and Windows systems. These files contained valuable information that could be used to further compromise the network.
6. **Use of OSINT**:
    - External reconnaissance techniques such as **OSINT** (Open Source Intelligence) and **WHOIS** lookups for the domain **totalrekall.xyz** provided critical information about the network and allowed for further exploitation. We also identified vulnerabilities based on public domain certificates and leveraged them for gaining additional access.

Key Findings:

1. **Web Server**:
    - The web application was highly vulnerable to common attack vectors like **XSS**, **SQL injection**, **LFI**, and **Command Injection**. These flaws allowed for unauthorized access and data exfiltration.
2. **Internal Network**:
    - Several internal servers, including both Linux and Windows hosts, were found to be misconfigured and vulnerable to known exploits such as **CVE-2017-5638 (Struts)** and **CVE-2017-12617 (Apache Tomcat)**.
    - Privilege escalation opportunities were abundant, and weak password policies were identified, allowing for easy lateral movement across the network.
3. **Windows Systems**:
    - Services on the Windows machines were improperly configured, allowing for easy exploitation via publicly known vulnerabilities. Once inside, privilege escalation was straightforward, allowing access to highly sensitive data on the **Domain Controller**.
4. **Credential Management**:
    - Weak password policies were a significant issue. Several accounts with easily guessable passwords (e.g., **alice** and **trivera**) were compromised, leading to full system access.

Conclusion: This penetration test highlighted significant security weaknesses within Rekall Corporation's infrastructure. The external-facing web server was highly vulnerable to a range of attacks, and internal systems lacked sufficient patching and security controls, making them susceptible to both remote and local exploits. Additionally, weak password policies and poor configuration practices further facilitated unauthorized access.
Immediate actions should include patching all identified vulnerabilities, improving password management, and implementing stronger internal network segmentation. Enhanced monitoring and detection capabilities are also recommended to detect and mitigate potential intrusions in the future.
The following screenshots and evidence are provided in the detailed report to support the findings, including successful exploitation steps, privilege escalation techniques, and the retrieval of sensitive data from the compromised systems.

Summary Vulnerability Overview

Vulnerability	Severity
The ethical penetration test of Rekall Corporation's network revealed multiple critical vulnerabilities across both web and internal network environments, affecting both Linux and Windows-based systems. These vulnerabilities were exploited to gain unauthorized access, escalate privileges, and exfiltrate sensitive information. Below is a summary of the specific vulnerabilities discovered, along with some additional vulnerabilities that were identified during the assessment. **Web Server Vulnerabilities:** 1. **Cross-Site Scripting (XSS) - Reflected**: o A **reflected XSS** vulnerability was found in the web server, allowing an attacker to inject malicious scripts into the web application, which would be executed on the victim's browser. This vulnerability could lead to session hijacking or malicious actions being performed on behalf of the user. 2. **Cross-Site Scripting (XSS) - Stored**: o A **stored XSS** vulnerability was identified, where malicious scripts were injected into the web server's stored data, causing them to be executed when other users accessed the infected data. This can lead to unauthorized access to sensitive information and session hijacking. 3. **Sensitive Data Exposure**: o Sensitive data, such as user credentials, were found exposed within PHP files on the web server. This indicates poor data sanitization practices and a lack of secure storage mechanisms for sensitive information. 4. **Local File Inclusion (LFI)**: o A **Local File Inclusion (LFI)** vulnerability was identified, which allowed an attacker to include files from the server's local file system. This could potentially lead to the exposure of sensitive files like /etc/passwd or configuration files containing credentials. 5. **SQL Injection**: o **SQL Injection** vulnerabilities were found, allowing an attacker to manipulate SQL queries executed by the application. This could lead to unauthorized access to the database, data exfiltration, and potential privilege escalation. 6. **Command Injection**: o **Command Injection** vulnerabilities were discovered, enabling an attacker to execute arbitrary commands on the server, potentially allowing for remote code execution or other malicious activities. 7. **Brute Force Attack**: o Weak or poorly protected authentication mechanisms on the web server allowed for **brute force attacks** to be successful. This could allow attackers to gain unauthorized access to user accounts or administrative functionality. 8. **PHP Injection**: o **PHP Injection** vulnerabilities were identified, which could allow an attacker to inject PHP code into the web server, enabling them to execute arbitrary code on the server and potentially take full control. 9. **Session Management**: o Poor session management, particularly weak session IDs, allowed for the **session hijacking** of administrative sessions.	

This vulnerability was exploited using **Burp Suite** to gain unauthorized admin access to the web application.

Linux Server Vulnerabilities:

1. **Shellshock (CVE-2014-6271)**:
   - The **Shellshock** vulnerability was present on server 192.168.13.11, allowing an attacker to execute arbitrary commands via specially crafted environment variables. This vulnerability was used to gain unauthorized access and exfiltrate sensitive data from the system.
2. **Struts2 Remote Code Execution (CVE-2017-5638)**:
   - The **Apache Struts2** vulnerability (CVE-2017-5638) was discovered on server 192.168.13.12, which allowed remote code execution via a specially crafted request. This critical vulnerability was exploited using **Metasploit**, providing a shell on the server and enabling further exploitation.
3. **Drupal Remote Code Execution (CVE-2019-6340)**:
   - The **Drupal vulnerability** (CVE-2019-6340) on server 192.168.13.13 allowed for remote code execution via the **drupal_restws_unserialize** exploit, which was successfully exploited using **Metasploit** to gain a reverse shell and obtain sensitive data.
4. **CVE-2019-14287**:
   - A privilege escalation vulnerability (CVE-2019-14287) was identified on server 192.168.13.14, allowing an attacker to gain **root access** by bypassing certain security restrictions. This vulnerability was exploited to gain full administrative control of the server.
5. **Unpatched Software**:
   - Critical unpatched vulnerabilities, as identified by **Nessus**, were present on various Linux servers, leaving them susceptible to exploits such as **remote code execution** and **privilege escalation**.

Windows System Vulnerabilities:

1. **SLMail Remote Code Execution (CVE-2017-16372)**:
   - A **SLMail** vulnerability on the Windows 10 machine was exploited, leading to remote code execution. The vulnerability allowed an attacker to take control of the system and escalate privileges.
2. **FTP and POP3 Service Exploits**:
   - Open FTP and POP3 services on the Windows 10 machine were exploited to gain access. Weak configuration and improper authentication mechanisms contributed to the success of these exploits.
3. **Privilege Escalation on Windows 10**:
   - **Privilege escalation** techniques were used to escalate from a user account to **system-level** access on the Windows 10 machine. This involved exploiting weak user credentials and leveraging system misconfigurations.
4. **Active Directory Credential Dumping**:
   - The **Kiwi** tool, used within **Metasploit**, was leveraged to dump **Active Directory** credentials, which were later cracked and used to gain administrative access to **WinDC01**.
5. **Weak Passwords**:
   - Several user accounts with weak passwords were discovered during the assessment (e.g., **alice** and **trivera**), which facilitated easy brute-forcing and subsequent unauthorized access to both the Linux and Windows systems.

Additional Vulnerabilities Identified:

1. **Weak Password Policies**:
   - A common weakness across both Linux and Windows systems was the use of weak passwords or poorly protected authentication mechanisms, which allowed for **brute-force attacks** and easy **credential guessing**.
2. **Lack of Network Segmentation**:
   - The internal network was not segmented sufficiently, allowing an attacker to move laterally across different systems after an initial compromise. This enabled the exploitation of multiple systems once access was gained on one host.
3. **Unrestricted Service Access**:
   - Several services, such as **FTP**, **SMTP**, **HTTP**, and **POP3**, were exposed on both Windows and Linux systems without proper firewalls or access controls. This increased the attack surface and made it easier for attackers to gain initial access to the network.
4. **Poor Web Application Security**:
   - The web application's failure to implement proper **input validation** and **output encoding** left it vulnerable to injection-based attacks such as **SQL Injection** and **XSS**, which could lead to data theft, unauthorized access, and potential system compromise.

Conclusion:

The penetration test revealed numerous vulnerabilities in Rekall Corporation's web and internal network systems. From critical web application flaws like **XSS** and **SQL Injection** to server-side vulnerabilities such as **Shellshock** and **CVE-2017-5638**, the security of both external and internal systems was significantly compromised. Additionally, weak password policies, misconfigurations, and the lack of proper network segmentation and monitoring compounded the risks.

Addressing these vulnerabilities through prompt patching, proper configuration management, and improved security controls, including **strong password policies** and **network segmentation**, will be crucial in enhancing Rekall Corporation's overall security posture.

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

Scan Type	Total
Hosts	**Linux Servers:** 1. **192.168.13.10** 2. **192.168.13.11** 3. **192.168.13.12** 4. **192.168.13.13** 5. **192.168.13.14** **Web Server:** 1. **192.168.14.35** (Web server IP) **Windows Systems:** 1. **172.22.117.10** (WinDC01 - Domain Controller) 2. **172.22.117.20** (Windows 10 machine) **Local Host (LHOST):** 1. **192.168.13.1** (Linux LHOST) 2. **172.22.117.100** (Windows LHOST) ****pls note that for linux server exploits for LHOST in certain cases Metasploit picked host was used instead.**
Ports	

Exploitation Risk	Total
Critical	**Port 80 (HTTP)** **Port 443 (HTTPS)** **Port 3306 (MySQL)** **Port 135 (MS RPC)** **Port 1433 (SQL Server)** **Port 445 (SMB)**
High	**Port 22 (SSH)** **Port 21 (FTP)** **Port 3389 (RDP)** **Port 25 (SMTP)** **Port 445 (SMB)**
Medium	**Port 8080 (HTTP Alternative)** **Port 110 (POP3)** **Port 21 (FTP)** (Windows)

	Port 25 (SMTP) (Windows) **Port 3306 (MySQL)** (Windows) **Port 3389 (RDP)** (Windows)
Low	**None identified** with a low risk rating during the test.

Here is a breakdown of the ports identified during the penetration test, categorized by their respective **exploit risk levels** (Critical, High, Medium, and Low). The categorization is based on the services running on those ports and their associated vulnerabilities:

Linux Servers (IP Range: 192.168.13.0/24)

1. **Port 22 (SSH)**
   - **Risk**: High
   - **Reason**: Commonly targeted by brute force attacks. Credentials or weak passwords may be exploited.
2. **Port 80 (HTTP)**
   - **Risk**: Critical
   - **Reason**: The web server was found vulnerable to **XSS**, **SQL Injection**, **LFI**, **Command Injection**, and other web-based vulnerabilities. Exploiting these can lead to remote code execution or unauthorized access.
3. **Port 443 (HTTPS)**
   - **Risk**: Critical
   - **Reason**: If misconfigured or vulnerable (e.g., improper certificate handling, SSL vulnerabilities), it can lead to security issues like **sensitive data exposure** or man-in-the-middle attacks.
4. **Port 21 (FTP)**
   - **Risk**: Medium
   - **Reason**: FTP was found open on some machines (e.g., Windows 10), and weak credentials or improper configuration could allow unauthorized access.
5. **Port 25 (SMTP)**
   - **Risk**: Medium
   - **Reason**: The SMTP service could be vulnerable to **command injection** or **brute force** attacks if not properly secured.
6. **Port 110 (POP3)**
   - **Risk**: Medium
   - **Reason**: Open POP3 services on Windows machines could be targeted for **credential guessing** or **man-in-the-middle** attacks.
7. **Port 3306 (MySQL)**
   - **Risk**: Critical
   - **Reason**: If exposed to the internet, **SQL Injection** and weak authentication mechanisms could be exploited to gain unauthorized access to the database.
8. **Port 445 (SMB)**
   - **Risk**: High

- **Reason**: SMB service was found open, making it susceptible to **EternalBlue** and other SMB exploits if not patched. This can allow an attacker to move laterally within the network.

9. **Port 3389 (RDP)**
   - **Risk**: **High**
   - **Reason**: RDP is often targeted by brute-force attacks, and if poorly configured or vulnerable, can lead to remote code execution or unauthorized access.
10. **Port 8080 (HTTP Alternative)**
    - **Risk**: **Medium**
    - **Reason**: Often used for web servers or application management interfaces. It can be targeted if exposed or vulnerable.

Windows Systems (IP Range: 172.22.117.0/24)

1. **Port 21 (FTP)**
   - **Risk**: **Medium**
   - **Reason**: FTP was exposed, and weak or default credentials could allow an attacker to gain access to the system.
2. **Port 25 (SMTP)**
   - **Risk**: **Medium**
   - **Reason**: Similar to Linux, **SMTP** can be vulnerable to **command injection** or **brute-force** attacks.
3. **Port 80 (HTTP)**
   - **Risk**: **High**
   - **Reason**: Exposed **HTTP** service on Windows 10 allows for exploitation of web-based vulnerabilities like **command injection** or **SQL injection**, leading to unauthorized access or data exfiltration.
4. **Port 443 (HTTPS)**
   - **Risk**: **High**
   - **Reason**: Similar to Port 80, HTTPS on Windows can be vulnerable to improper configuration, **SSL/TLS** vulnerabilities, or exposed sensitive data.
5. **Port 445 (SMB)**
   - **Risk**: **High**
   - **Reason**: SMB vulnerabilities (e.g., **EternalBlue**) were exploited to gain unauthorized access to Windows machines.
6. **Port 139 (NetBIOS)**

- Risk: High
- Reason: NetBIOS can be used to find vulnerabilities in file-sharing services and exposed SMB services on Windows machines.

7. Port 3389 (RDP)
   - Risk: High
   - Reason: Exposed RDP services can be attacked with brute-force techniques or targeted for remote code execution.
8. Port 135 (MS RPC)
   - Risk: Critical
   - Reason: MS RPC is vulnerable to exploits like MS08-067, which can allow attackers to gain remote control over Windows systems. This can also facilitate lateral movement within the network.
9. Port 1433 (SQL Server)
   - Risk: Critical
   - Reason: Exposed SQL Server services are vulnerable to SQL Injection or authentication bypass attacks, allowing an attacker to gain full access to sensitive databases.
10. Port 3306 (MySQL)
    - Risk: Medium
    - Reason: MySQL service found on some hosts could be vulnerable to SQL Injection and weak authentication, allowing attackers to access or manipulate databases.

Critical Ports Summary:

- Port 80 (HTTP)
- Port 443 (HTTPS)
- Port 3306 (MySQL)
- Port 135 (MS RPC)
- Port 1433 (SQL Server)
- Port 445 (SMB)

High Ports Summary:

- Port 22 (SSH)
- Port 21 (FTP)
- Port 3389 (RDP)
- Port 25 (SMTP)

- Port 445 (SMB)

Medium Ports Summary:

- Port 8080 (HTTP Alternative)
- Port 110 (POP3)
- Port 21 (FTP) (Windows)
- Port 25 (SMTP) (Windows)
- Port 3306 (MySQL) (Windows)
- Port 3389 (RDP) (Windows)

Low Ports Summary:

- None identified with a low risk rating during the test.

Conclusion:

The ports listed above, especially those marked as Critical and High, pose significant risks to Rekall Corporation's security posture. Immediate action should be taken to address these vulnerabilities, including patching and securing services exposed on critical ports like HTTP, SMB, RDP, and MS RPC. Additionally, all services should be carefully monitored and access restricted where possible to mitigate the risk of exploitation.

Vulnerability Findings

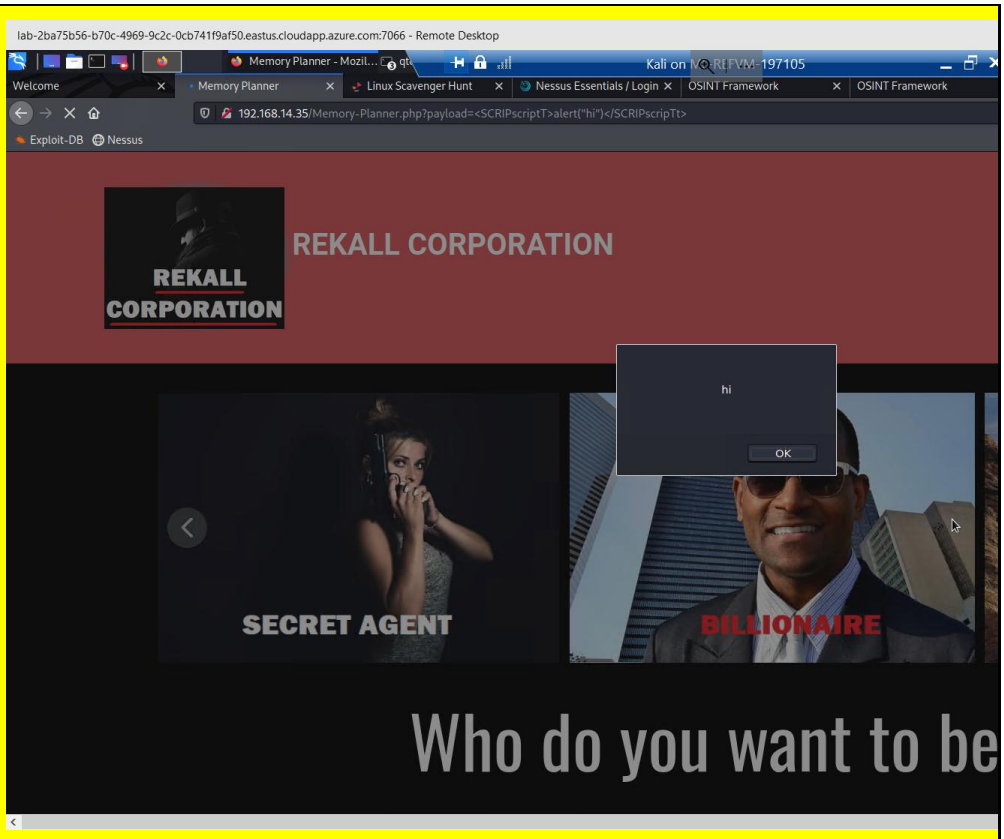
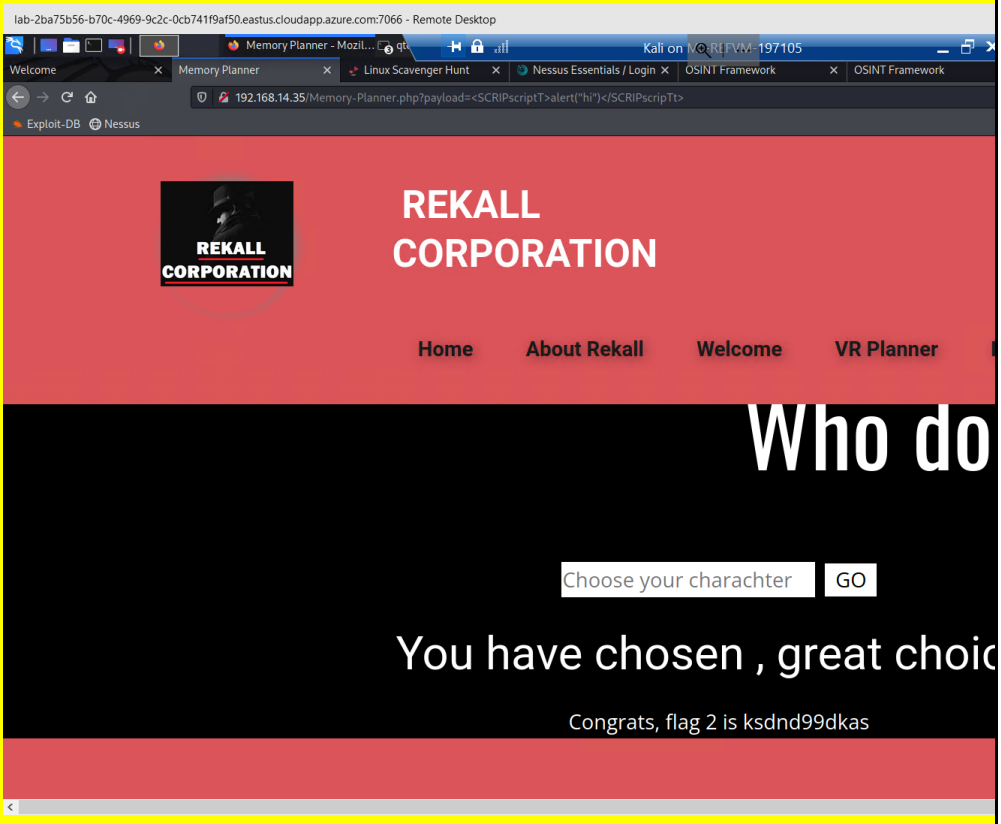
All 37 Vulnerabilities and Flags below

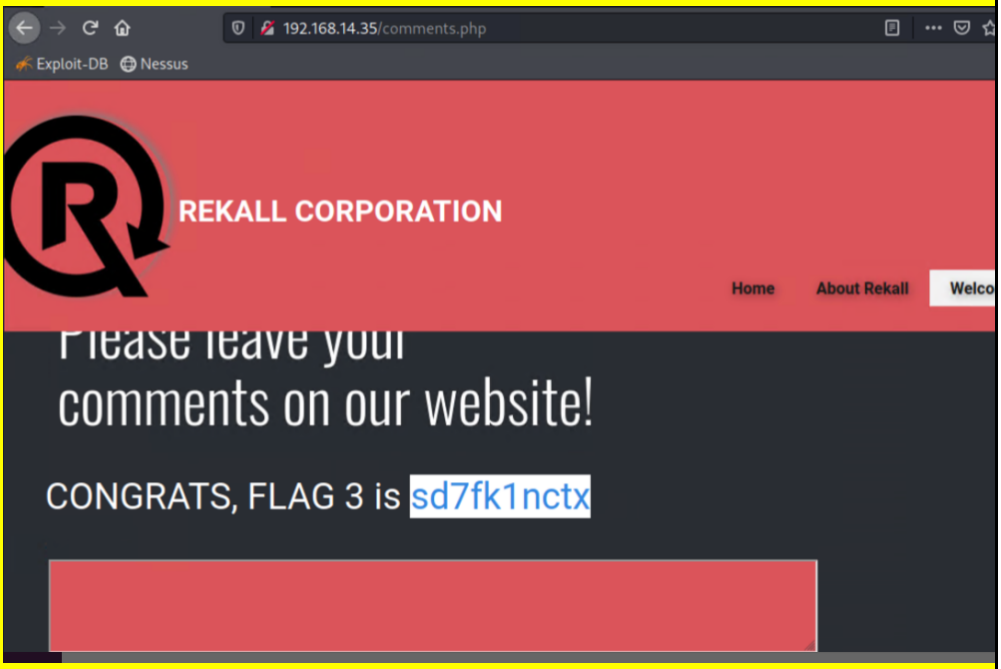
Web App Based Vulnerabilities Below

Vulnerability 1	Findings
Title	Cross-Site Scripting (XSS) - Reflected
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	Reflected XSS allows an attacker to inject malicious scripts into the web application. These scripts are executed in the victim's browser when they interact with the web application, leading to potential session hijacking, data theft, or other malicious actions.

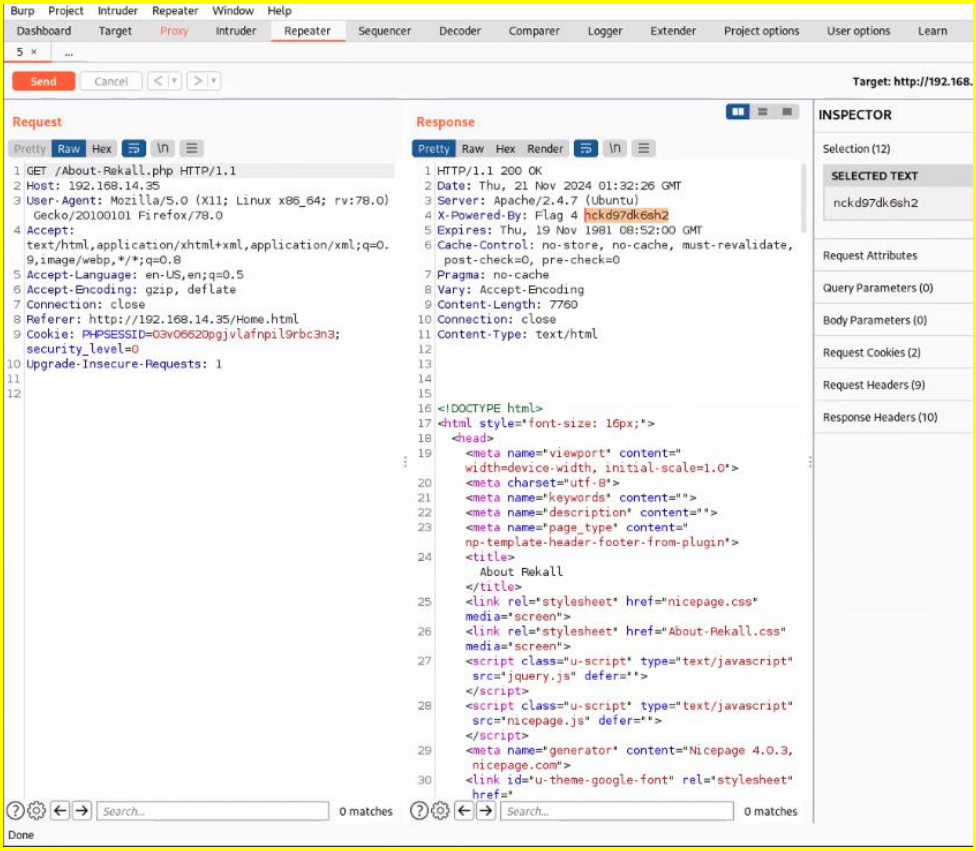
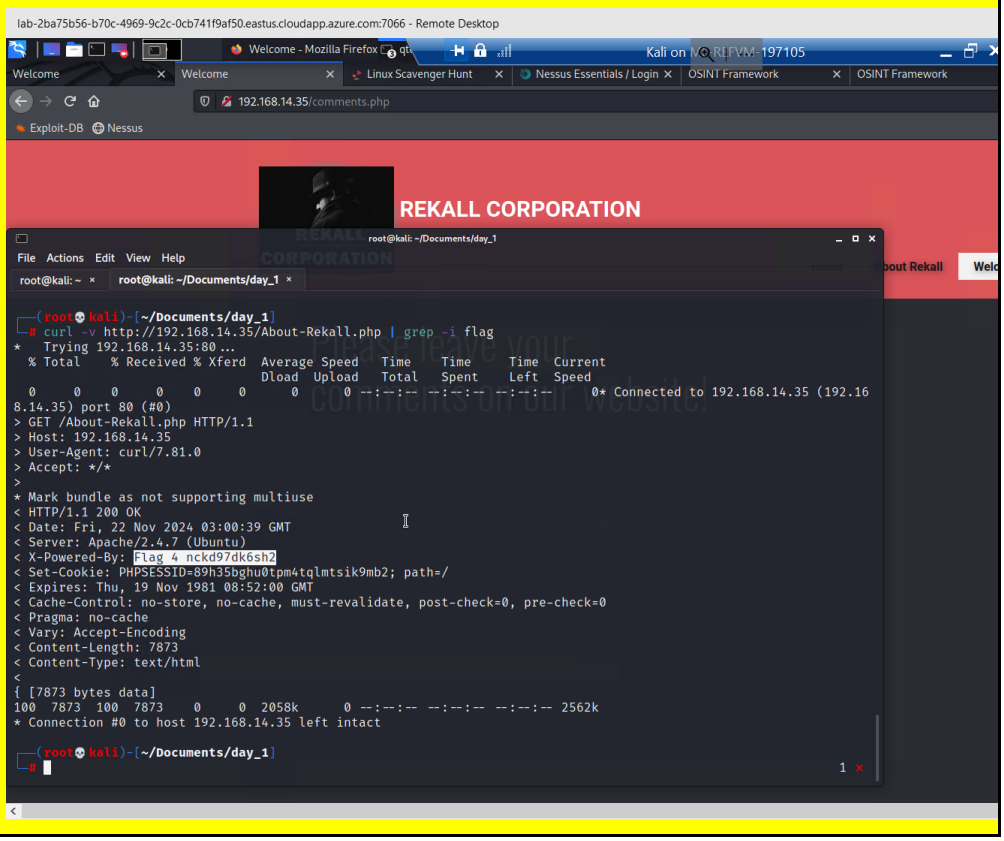
Images	
Affected Hosts	• 192.168.14.35 (Web server)
Remediation	Implement input validation, use output encoding techniques, and ensure proper sanitization of user inputs to prevent script execution.

Vulnerability 2	Findings
Title	Cross-Site Scripting (XSS) - Reflected (Advanced)
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	Reflected XSS allows an attacker to inject malicious scripts into the web application. These scripts are executed in the victim's browser when they interact with the web application, leading to potential session hijacking, data theft, or other malicious actions.

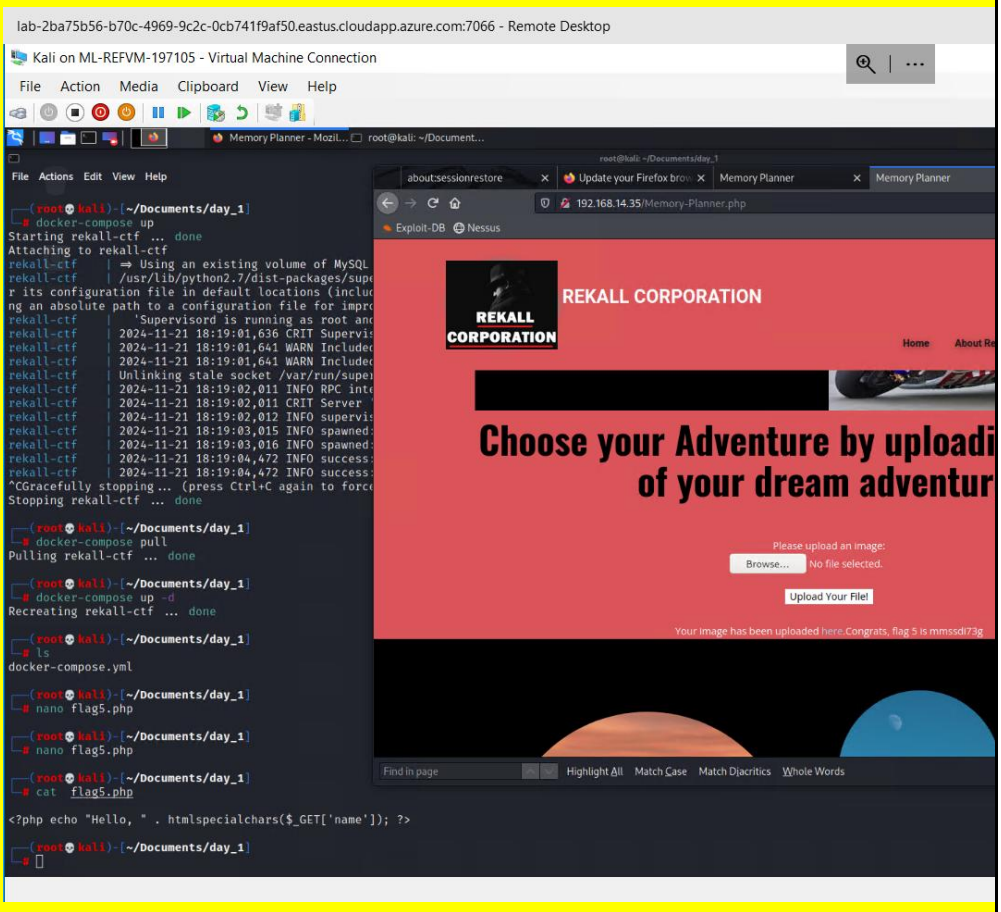
Images	 
Affected Hosts	192.168.14.35 (Web server)
Remediation	Implement input validation, use output encoding techniques, and ensure proper sanitization of user inputs to prevent script execution.

Vulnerability 3	Findings
Title	**Cross-Site Scripting (XSS) - Stored**
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	• Critical
Description	• Stored XSS occurs when an attacker injects a script that is stored on the server and executed when other users access the stored data. This could lead to unauthorized actions performed on behalf of the victim or data exposure.
Images	
Affected Hosts	• 192.168.14.35 (Web server)
Remediation	Implement secure coding practices, sanitize all user inputs, and use Content Security Policies (CSP) to reduce the risk.

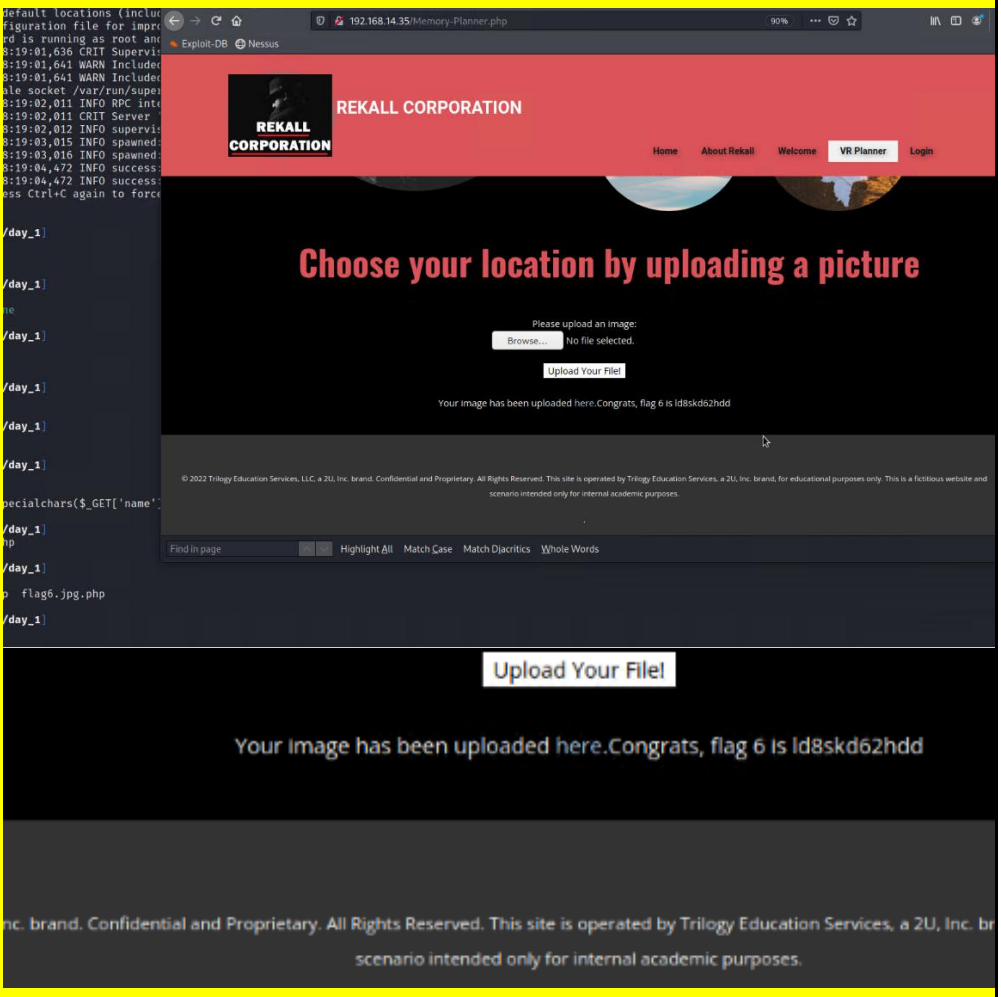
Vulnerability 4	Findings
Title	**Sensitive Data Exposure**
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical

Description	• Sensitive data, such as credentials and personal information, was found exposed in PHP files on the web server. This leads to the unauthorized access of confidential data, putting users at risk.
Images	 

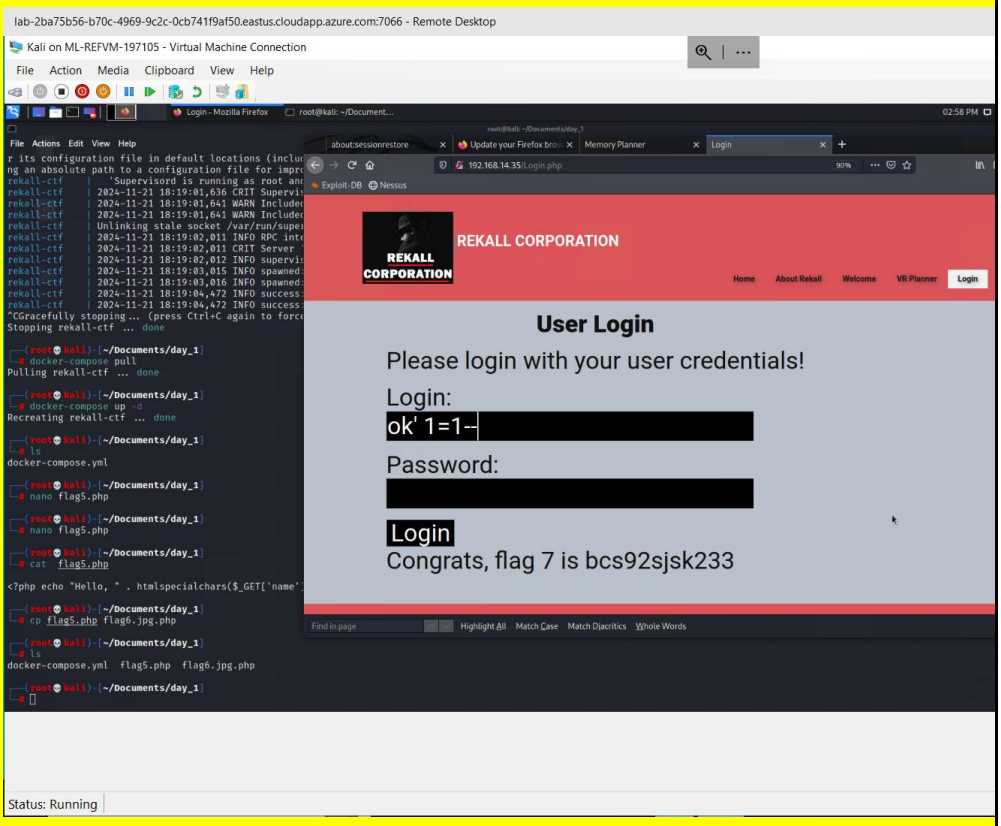
Affected Hosts	192.168.14.35 (Web server)
Remediation	Encrypt sensitive data both at rest and in transit, apply strong access controls, and ensure that sensitive data is not stored in accessible locations.

Vulnerability 5	**Findings**
Title	**Local File Inclusion (LFI)**
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	• Local File Inclusion vulnerabilities allow an attacker to access sensitive files on the server by exploiting improper file handling. This can lead to the disclosure of sensitive files like /etc/passwd or even remote code execution.
Images	
Affected Hosts	192.168.14.35 (Web server)

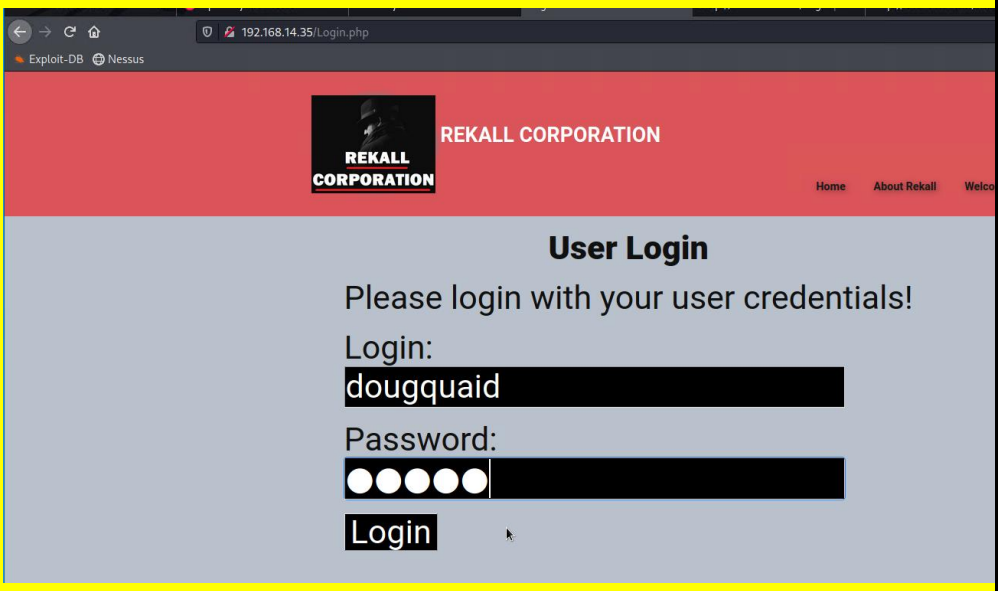
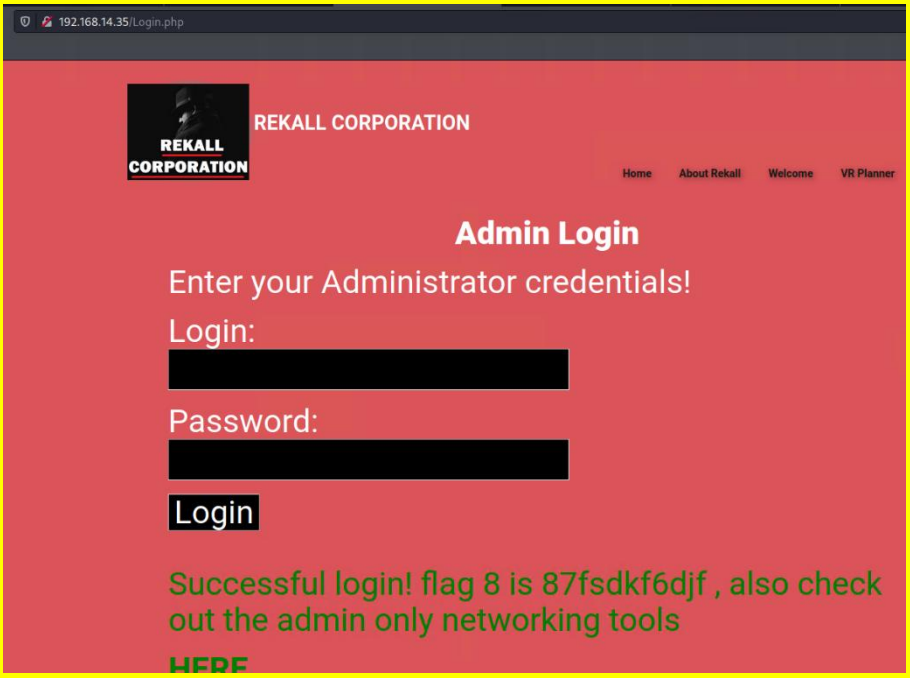
Remediation	Validate and sanitize file input from users, use allowlists for file paths, and implement least privilege access to sensitive files.

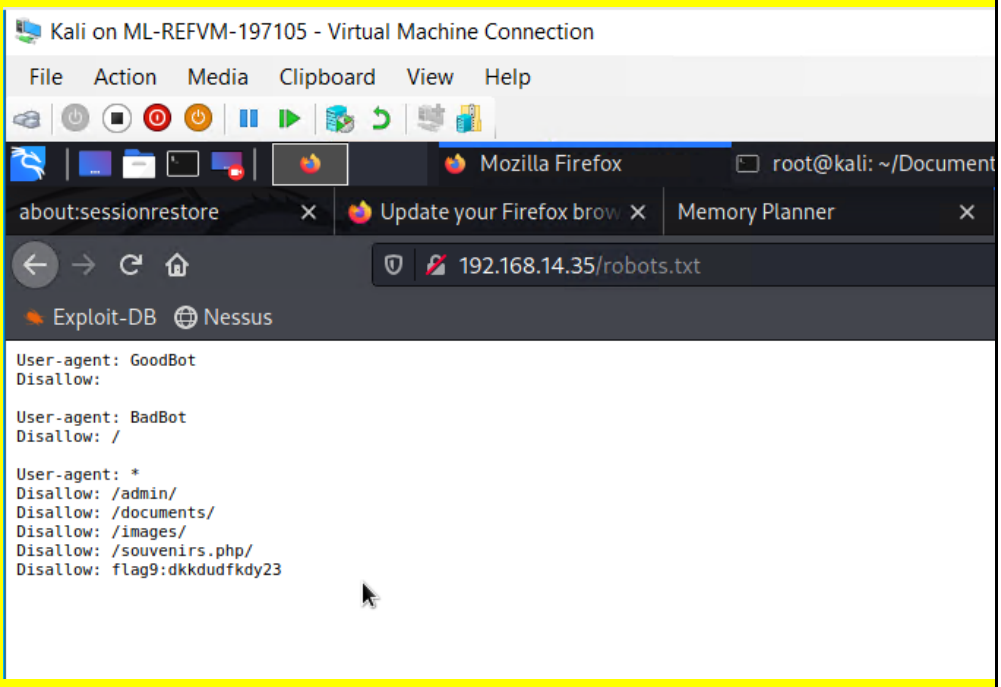
Vulnerability 6	**Findings**
Title	**Local File Inclusion (LFI) (Advanced)**
Type (Web app / Linux OS / WIndows OS)	• Web app
Risk Rating	Critical
Description	• Local File Inclusion vulnerabilities allow an attacker to access sensitive files on the server by exploiting improper file handling. This can lead to the disclosure of sensitive files like /etc/passwd or even remote code execution.
Images	
Affected Hosts	• 192.168.14.35 (Web server)

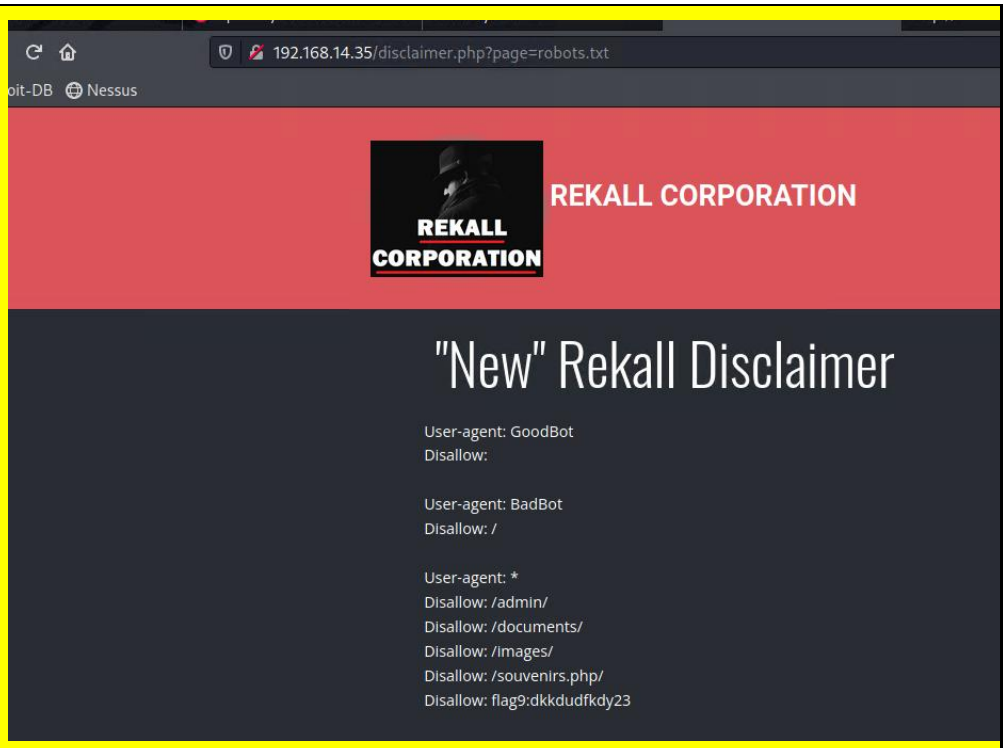
Remediation	Validate and sanitize file input from users, use allowlists for file paths, and implement least privilege access to sensitive files.

Vulnerability 7	**Findings**
Title	**SQL Injection**
Type (Web app / Linux OS / WIndows OS)	• Web app
Risk Rating	Critical
Description	• SQL Injection allows an attacker to manipulate SQL queries and gain unauthorized access to the database. This could lead to data exfiltration, privilege escalation, and remote code execution.
Images	
Affected Hosts	• 192.168.14.35 (Web server)
Remediation	Use prepared statements, parameterized queries, and input validation to protect against SQL injection attacks.

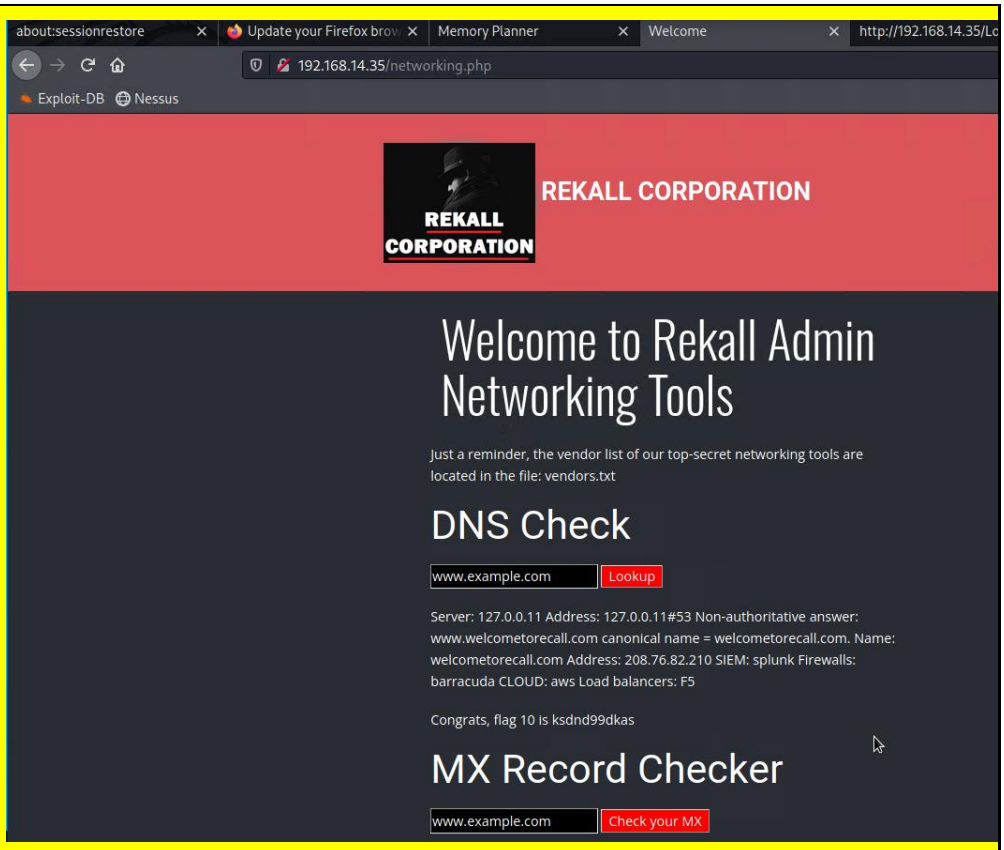
Vulnerability 8	**Findings**
Title	**Sensitive Data Exposure**

Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	• Sensitive data, such as credentials and personal information, was found exposed in PHP files on the web server. This leads to the unauthorized access of confidential data, putting users at risk.
Images	 
Affected Hosts	• 192.168.14.35 (Web server)
Remediation	Encrypt sensitive data both at rest and in transit, apply strong access controls, and ensure that sensitive data is not stored in accessible locations.

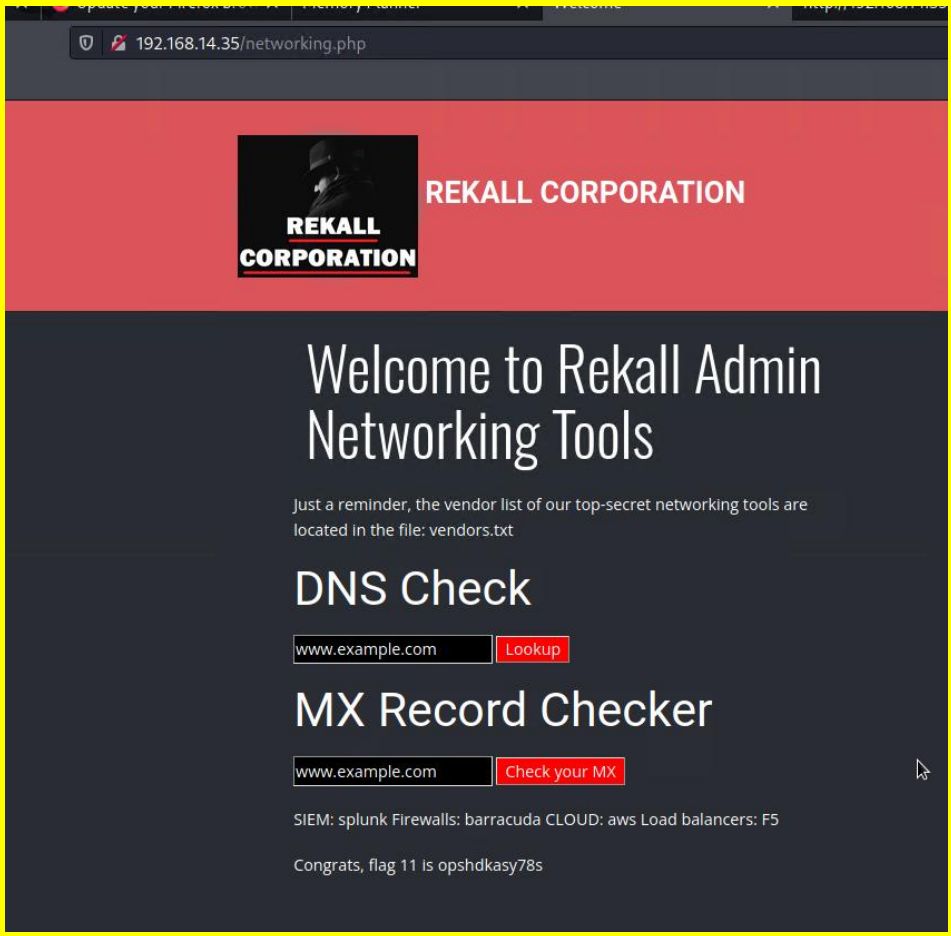
Vulnerability 9	Findings
Title	**Sensitive Data Exposure**
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	• Sensitive data, such as credentials and personal information, was found exposed in PHP files on the web server. This leads to the unauthorized access of confidential data, putting users at risk.
Images	

	
Affected Hosts	• 192.168.14.35 (Web server)
Remediation	Encrypt sensitive data both at rest and in transit, apply strong access controls, and ensure that sensitive data is not stored in accessible locations.

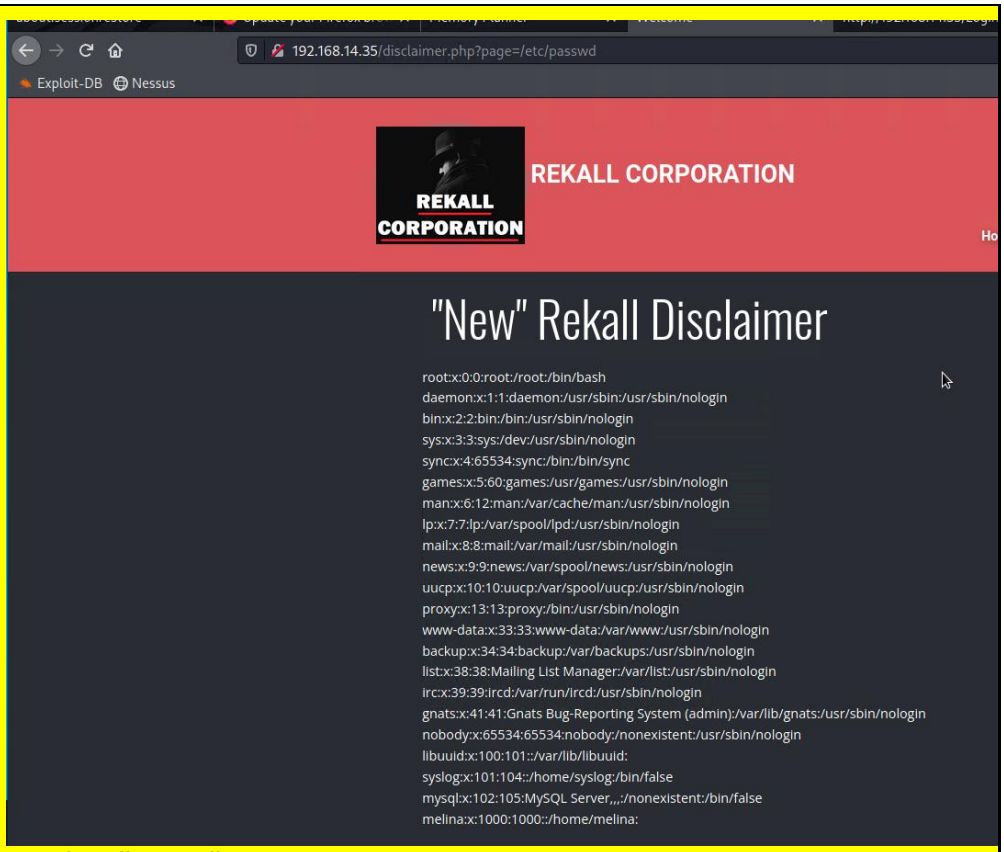
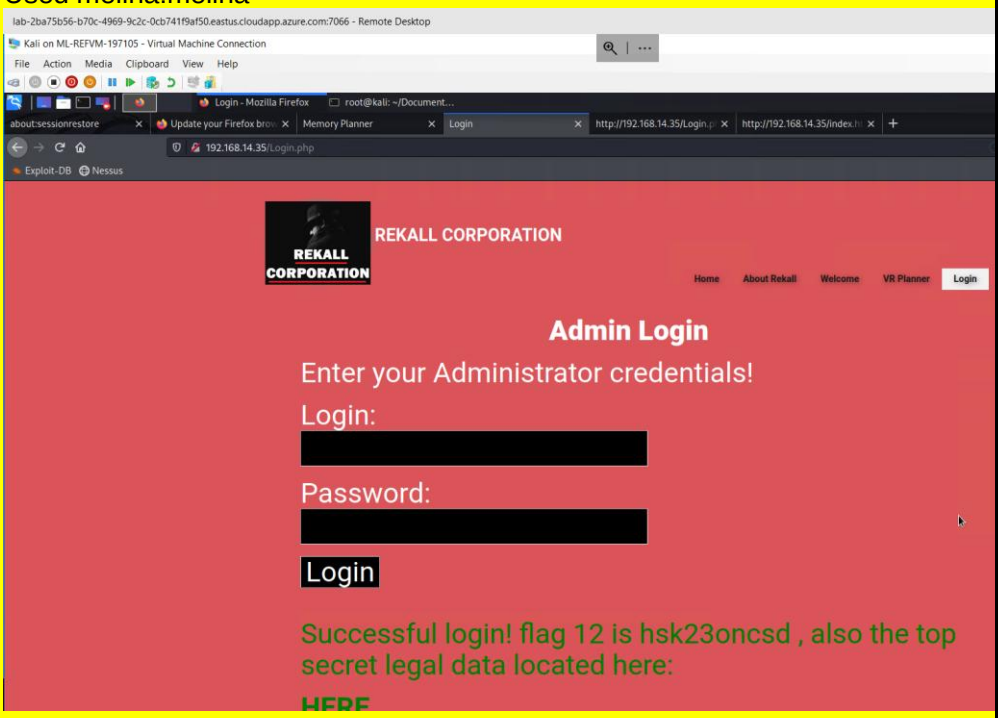
Vulnerability 10	Findings
Title	**Command Injection**
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	• Command injection vulnerabilities allow attackers to execute arbitrary system commands on the server. This can result in remote code execution and the full compromise of the server.

Images	
Affected Hosts	192.168.14.35 (Web server)
Remediation	Sanitize all user inputs, avoid system calls, and validate the commands executed by the application.

Vulnerability 11	Findings
Title	**Command Injection (Advanced)**
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	Command injection occurs when an attacker is able to inject and execute arbitrary system commands on the server, typically through unsanitized user input, leading to unauthorized access or control over the system.

Images	
Affected Hosts	• 192.168.14.35 (Web server)
Remediation	To prevent command injection, validate and sanitize user inputs rigorously, avoid passing user input directly to system commands, and use secure API methods or libraries that prevent command execution.

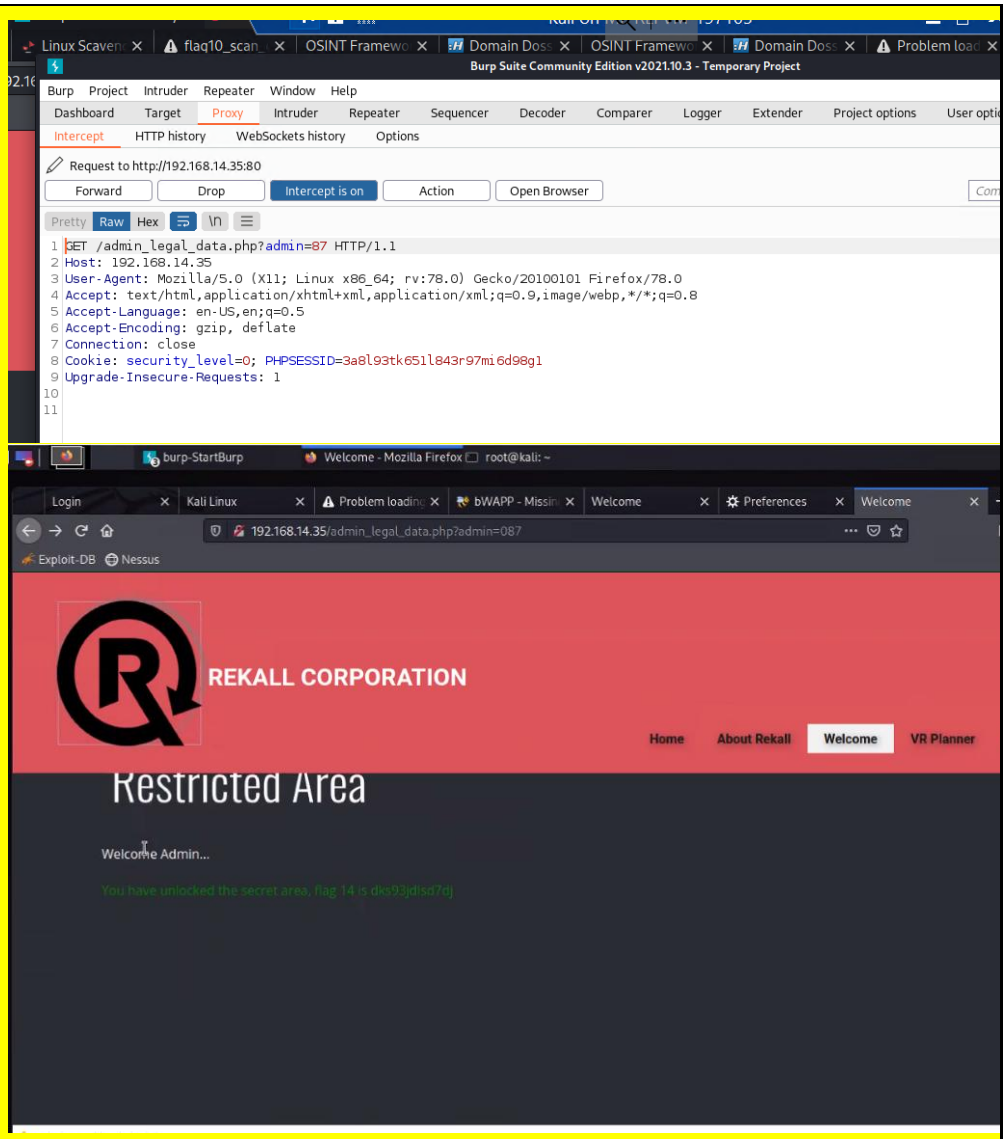
Vulnerability 12	Findings
Title	**Brute Force Attack (Weak Authentication)**
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	• The web application was vulnerable to brute force attacks on login forms, where weak passwords or poor authentication mechanisms could be exploited.

Images	 Used melina:melina 
Affected Hosts	• 192.168.14.35 (Web server)
Remediation	• Implement account lockout mechanisms, enforce strong password policies, and use multi-factor authentication.

Vulnerability 13	Findings

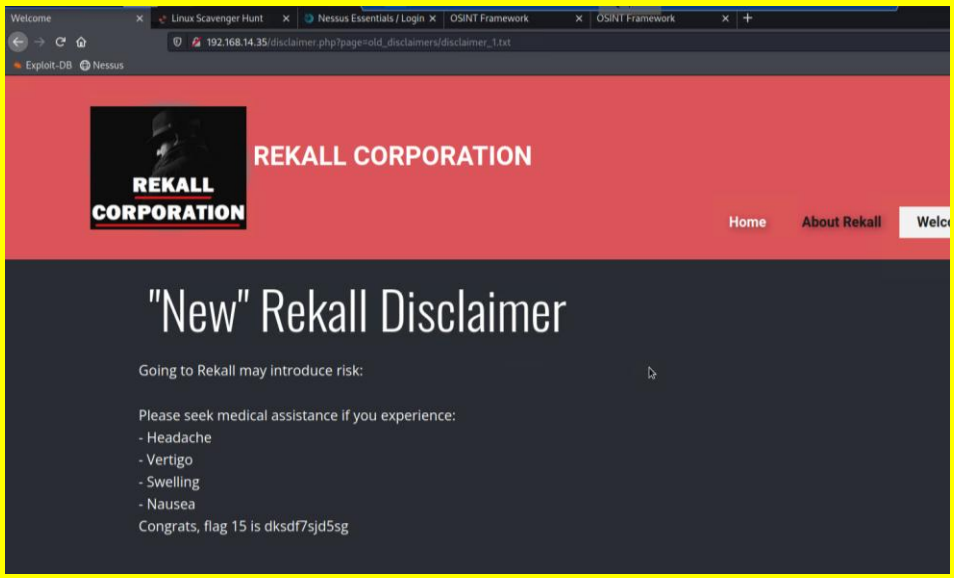
Title	**PHP Injection**
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	PHP injection vulnerabilities allow attackers to inject PHP code into the server, enabling remote code execution and potential full server compromise.
Images	
Affected Hosts	• 192.168.14.35 (Web server)
Remediation	Validate and sanitize user inputs, disable dangerous PHP functions, and implement proper access control to server-side scripts.

Vulnerability 14	Findings
Title	**Session Management (Session Hijacking)**
Type (Web app / Linux OS / WIndows OS)	• Web app
Risk Rating	Critical
Description	• Poor session management, including weak session IDs, allows attackers to hijack user sessions, gain unauthorized access to user accounts, and perform actions as the victim.
Images	



Affected Hosts	• 192.168.14.35 (Web server)
Remediation	Implement secure cookie attributes (e.g., HttpOnly, Secure), use unique and long session IDs, and employ session expiration mechanisms

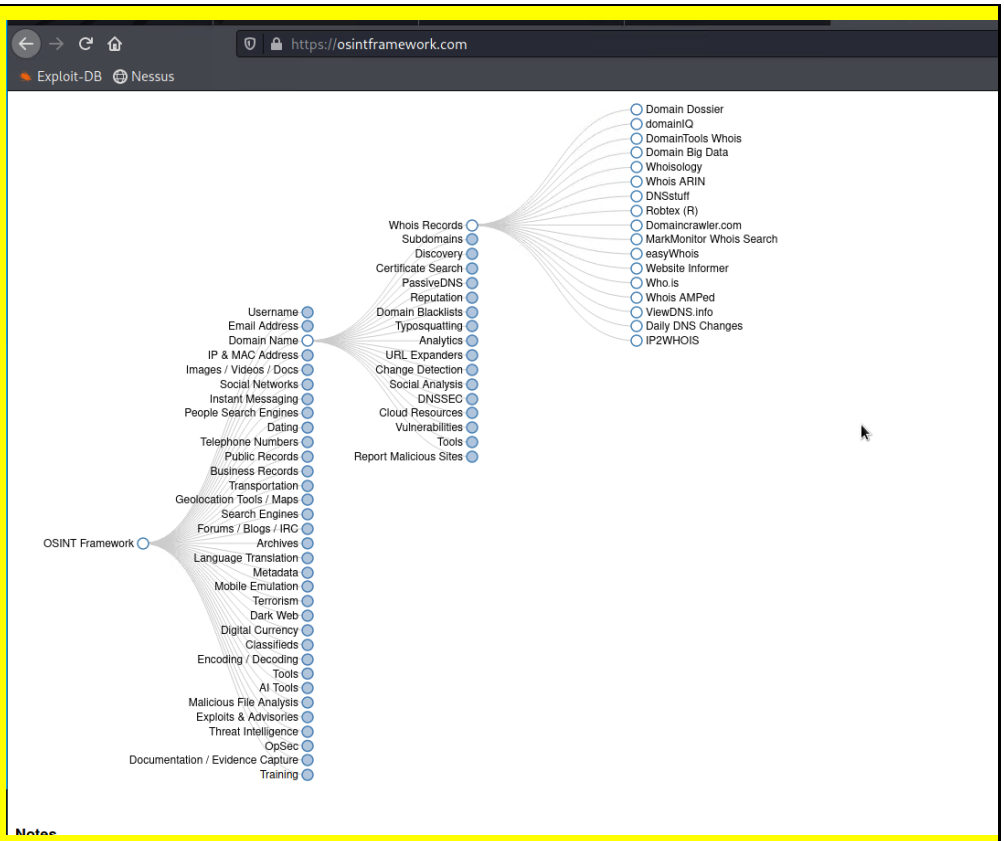
Vulnerability 15	Findings
Title	**Directory Traversal**
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	Directory traversal occurs when an attacker manipulates file paths to access directories and files outside of the intended web application directory, potentially exposing sensitive information.

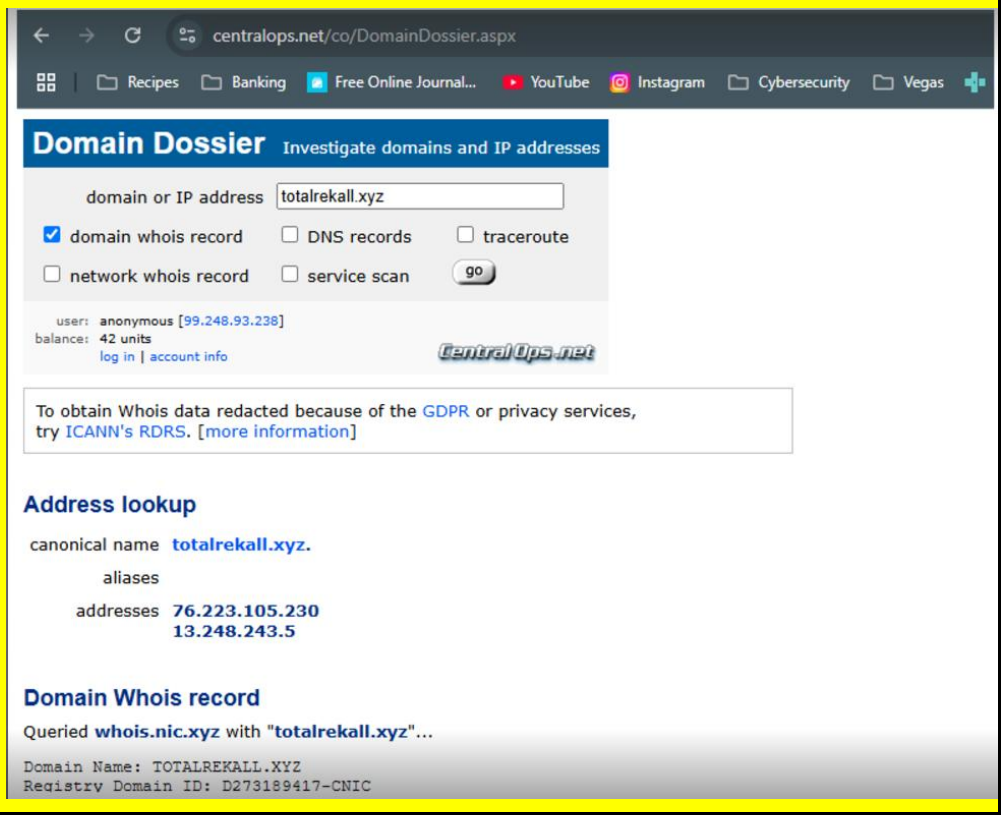
Images	
Affected Hosts	• 192.168.14.35 (Web server)
Remediation	To prevent directory traversal, validate and sanitize user input, use secure file path handling functions, and restrict file access to specific directories using proper access controls.

Linux Server Based Vulnerabilities Below

Vulnerability 1	Findings
Title	**Open Source Exposed Data**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	Exposing open-source data via a domain dossier on Linux servers occurs when sensitive information, such as configuration files, source code, or database credentials, is inadvertently accessible through the server's public domain or web interface. This can lead to attackers gaining insight into the server’s internal structure or sensitive data.

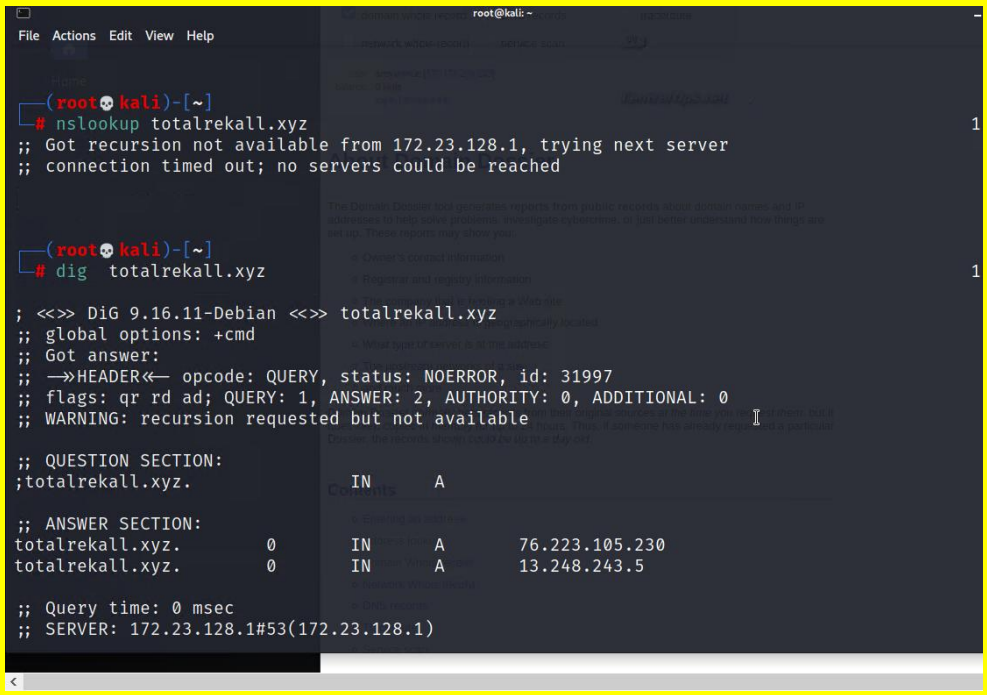
Images



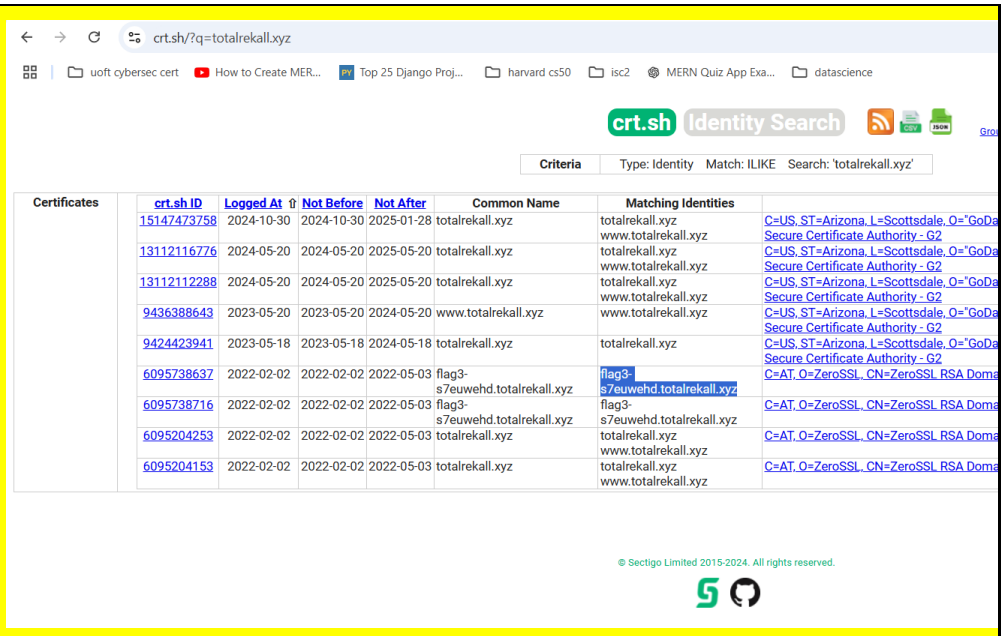


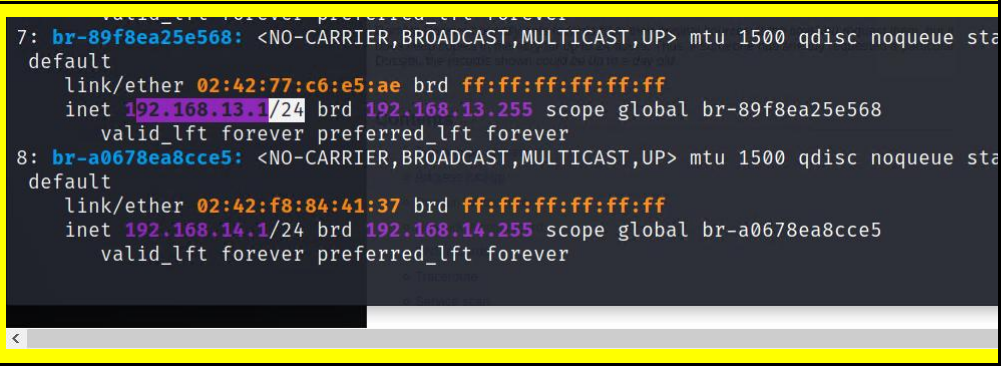
	Queried whois.godaddy.com with "totalrekall.xyz"... Domain Name: totalrekall.xyz Registry Domain ID: D273189417-CNIC Registrar WHOIS Server: whois.godaddy.com Registrar URL: https://www.godaddy.com Updated Date: 2024-02-03T15:15:56Z Creation Date: 2022-02-02T19:16:16Z Registrar Registration Expiration Date: 2025-02-02T23:59:5 Registrar: GoDaddy.com, LLC Registrar IANA ID: 146 Registrar Abuse Contact Email: abuse@godaddy.com Registrar Abuse Contact Phone: +1.4806242505 Domain Status: clientTransferProhibited https://icann.org/ Domain Status: clientUpdateProhibited https://icann.org/ep Domain Status: clientRenewProhibited https://icann.org/epp Domain Status: clientDeleteProhibited https://icann.org/ep Registry Registrant ID: CR534509109 Registrant Name: sshUser alice Registrant Organization: Registrant Street: h8s692hskasd Flag1 Registrant City: Atlanta
Affected Hosts	• Applicable to All hosts and Network
Remediation	To mitigate this risk, ensure that sensitive files are not accessible via the web, configure proper file permissions, use .htaccess or equivalent server configuration to restrict access to sensitive files, and avoid storing sensitive information in publicly accessible directories.

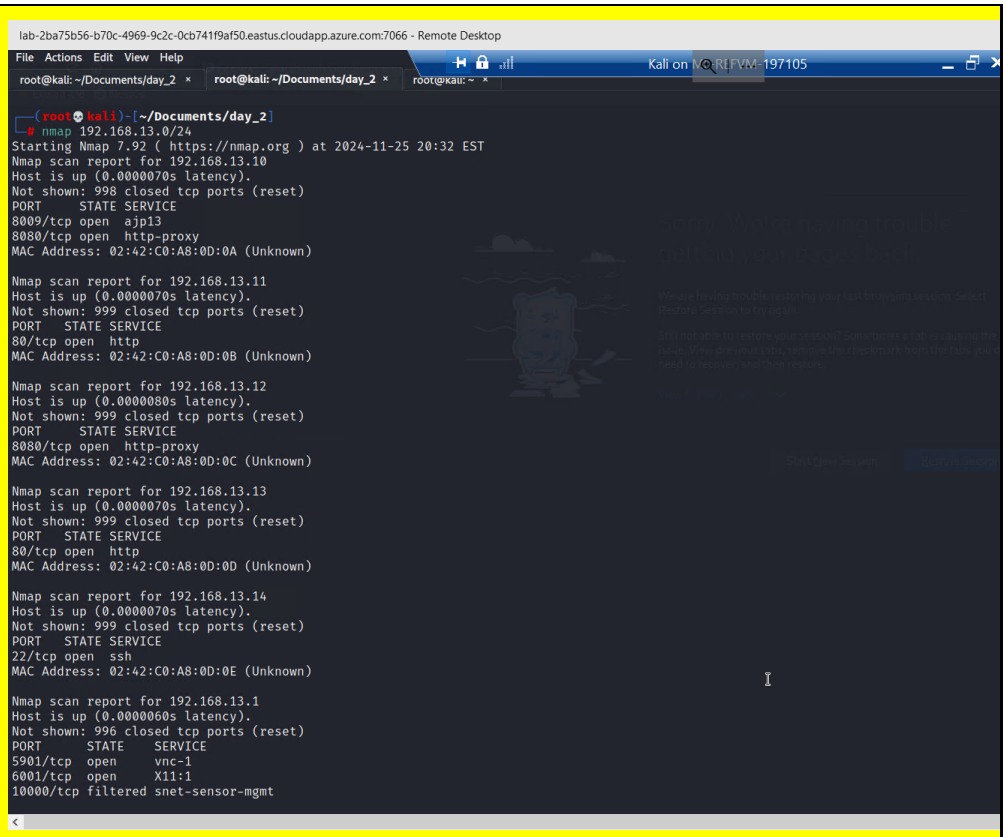
Vulnerability 2	Findings
Title	**PING**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	Exploited payload via ping occurs when an attacker crafts malicious payloads within ICMP (ping) packets, exploiting vulnerabilities in the server's or network's ability to process or respond to ping requests, potentially leading to denial of service (DoS) or unauthorized access.
Images	cybersec@cybersec: $ ping totalrekall.xyz PING totalrekall.xyz (76.223.105.230) 56(84) bytes of data. 64 bytes from a16e665f42988324c.awsglobalaccelerator.com (76.223.105.230): icmp_ 64 bytes from a16e665f42988324c.awsglobalaccelerator.com (76.223.105.230): icmp_ 64 bytes from a16e665f42988324c.awsglobalaccelerator.com (76.223.105.230): icmp_ 64 bytes from a16e665f42988324c.awsglobalaccelerator.com (76.223.105.230): icmp_ --- totalrekall.xyz ping statistics --- 4 packets transmitted, 4 received, 0% packet loss, time 3003ms rtt min/avg/max/mdev = 3.466/5.155/8.263/1.841 ms

	
Affected Hosts	• totalrekall.xyz; flag was on 172.23.128.1 originally but it was unavailable later, nslookup and dig results shown above besides ping.
Remediation	To prevent exploited payloads via ping, implement strict input validation on ICMP packets, limit the size of ping requests, configure firewalls to block unnecessary ping traffic, and ensure the network infrastructure can handle malicious traffic without vulnerabilities.

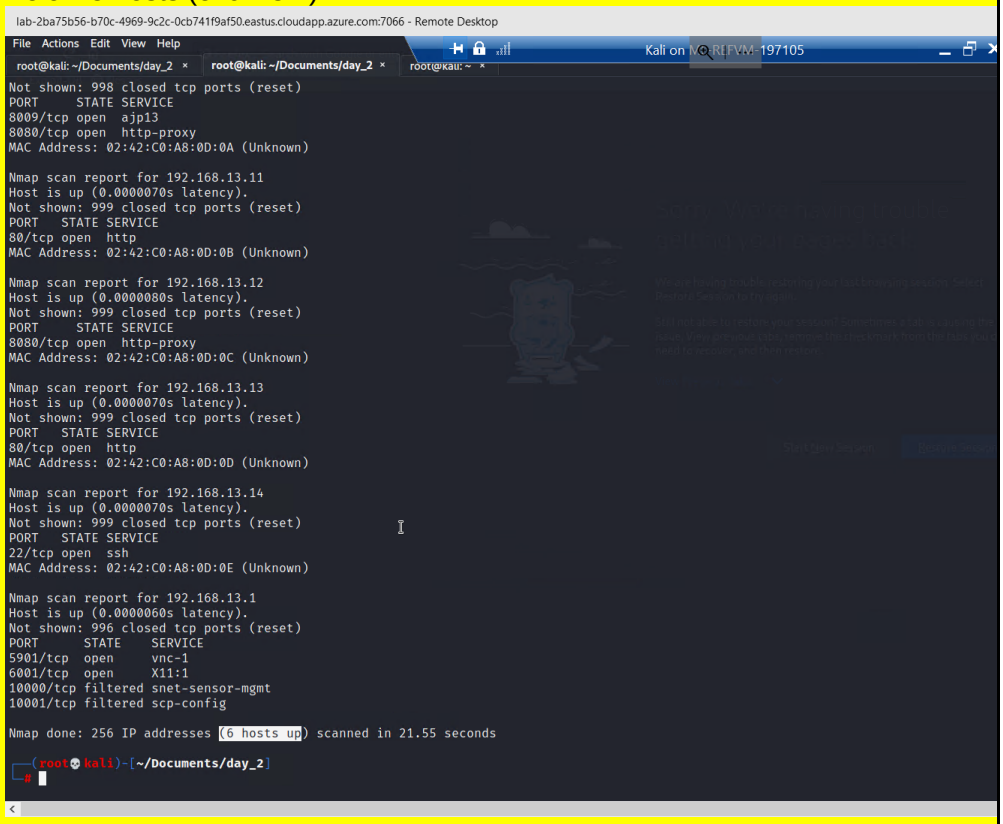
Vulnerability 3	Findings
Title	**Open Source Exposed Data**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	Exposing open-source data via a domain dossier on Linux servers occurs when sensitive information, such as configuration files, source code, or database credentials, is inadvertently accessible through the server's public domain or web interface. This can lead to attackers gaining insight into the server's internal structure or sensitive data.

Images	
Affected Hosts	• All hosts, domains and network.
Remediation	To mitigate this risk, ensure that sensitive files are not accessible via the web, configure proper file permissions, use .htaccess or equivalent server configuration to restrict access to sensitive files, and avoid storing sensitive information in publicly accessible directories.

Vulnerability 4	Findings
Title	**NMAP Scan**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Medium
Description	An Nmap scan is a network scanning technique used by attackers to identify hosts, services, and open ports on a network. While it's a legitimate tool for network administration, it can be exploited by attackers to gather information for further exploitation or attacks, such as identifying vulnerable services or misconfigured servers.
Images	

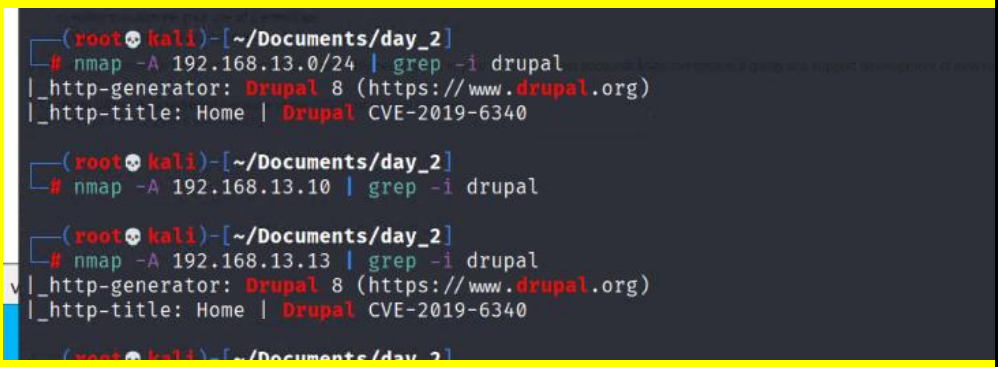


Below 5 hosts (excl 13.1)



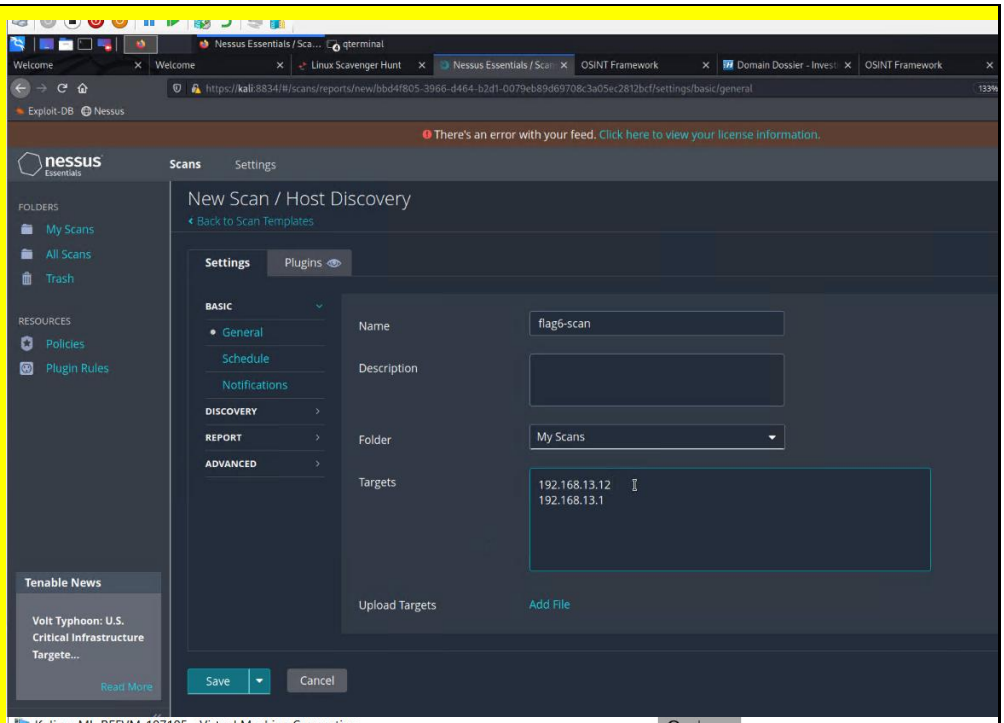
Affected Hosts	• All, network scan
Remediation	To prevent Nmap scans from identifying hosts, use network segmentation,

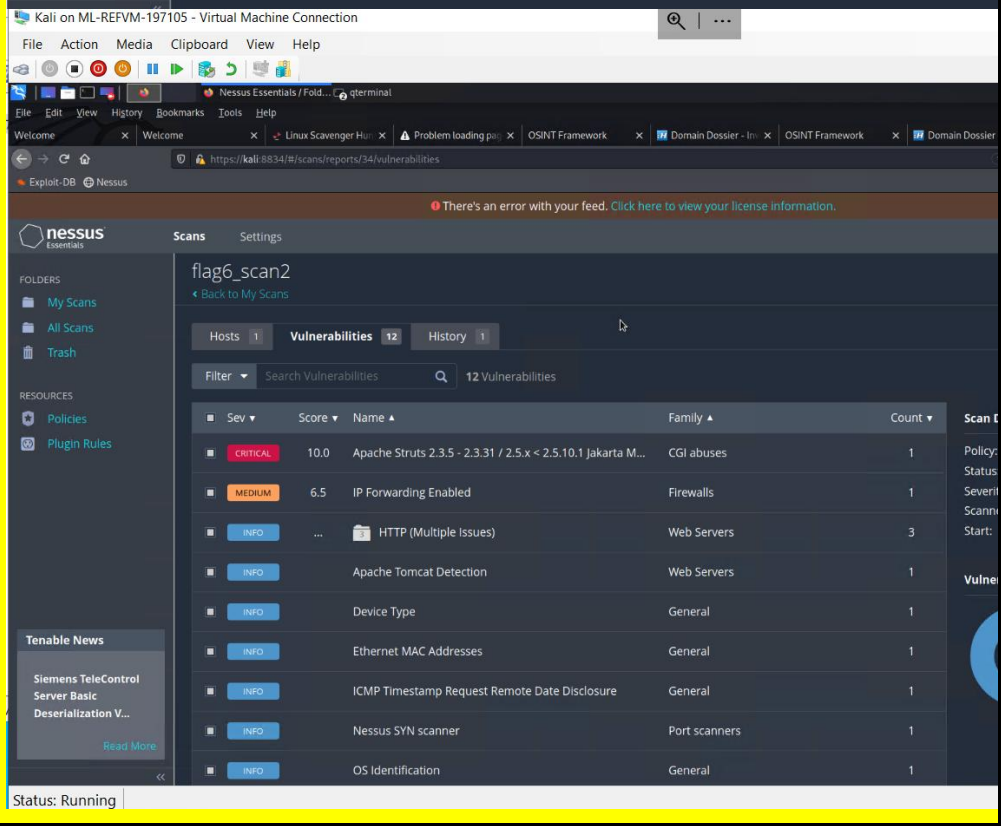
	implement firewalls to block unauthorized scanning attempts, enable intrusion detection systems (IDS) to detect scanning activity, and obfuscate services or change default ports to reduce the visibility of critical systems.

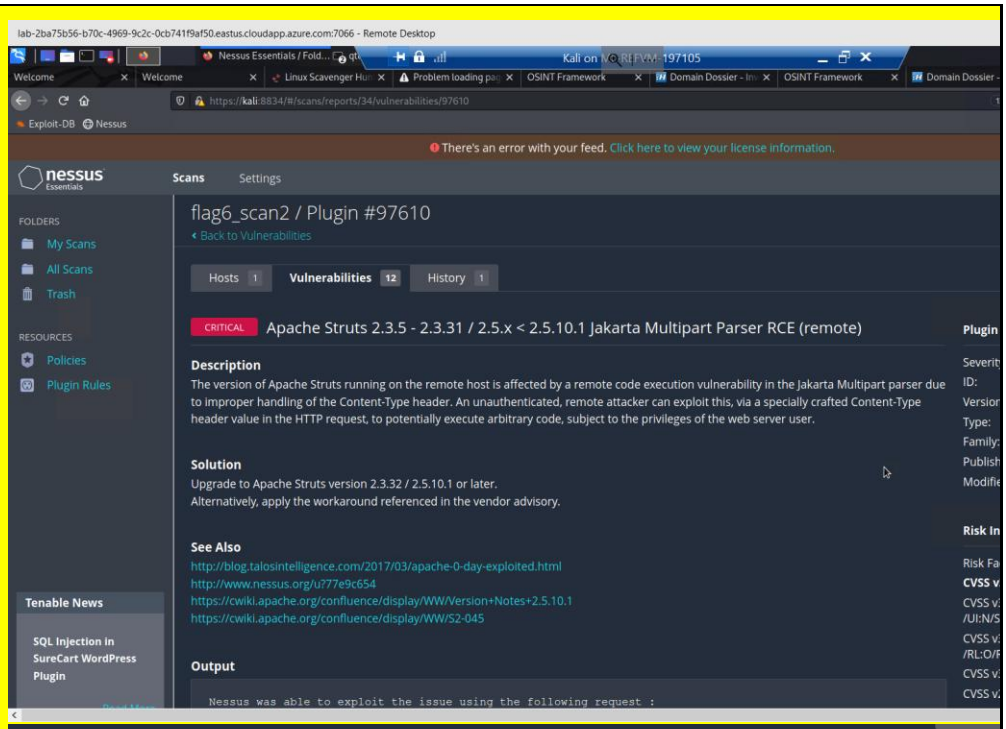
Vulnerability 5	Findings
Title	**Drupal Remote Code Execution (CVE-2019-6340)** Finding Drupal Host via Nmap Aggressive Scan
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	• The vulnerability in Drupal (CVE-2019-6340) allows remote code execution via the drupal_restws_unserialize exploit. Attackers can exploit this flaw to execute arbitrary code on the server. Aggressive Nmap scan identified this host running Drupal.
Images	
Affected Hosts	• 192.168.13.13
Remediation	Update Drupal to the latest version and apply security patches. Disable unused modules and limit access to the application.

Vulnerability 6	Findings
Title	**Struts2 Remote Code Execution (CVE-2017-5638)** **Struts Vulnerability Identification via Nessus Scan**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	• The Apache Struts2 vulnerability (CVE-2017-5638) allows for remote code execution through a specially crafted request, allowing an attacker to gain shell access to the server.

Images

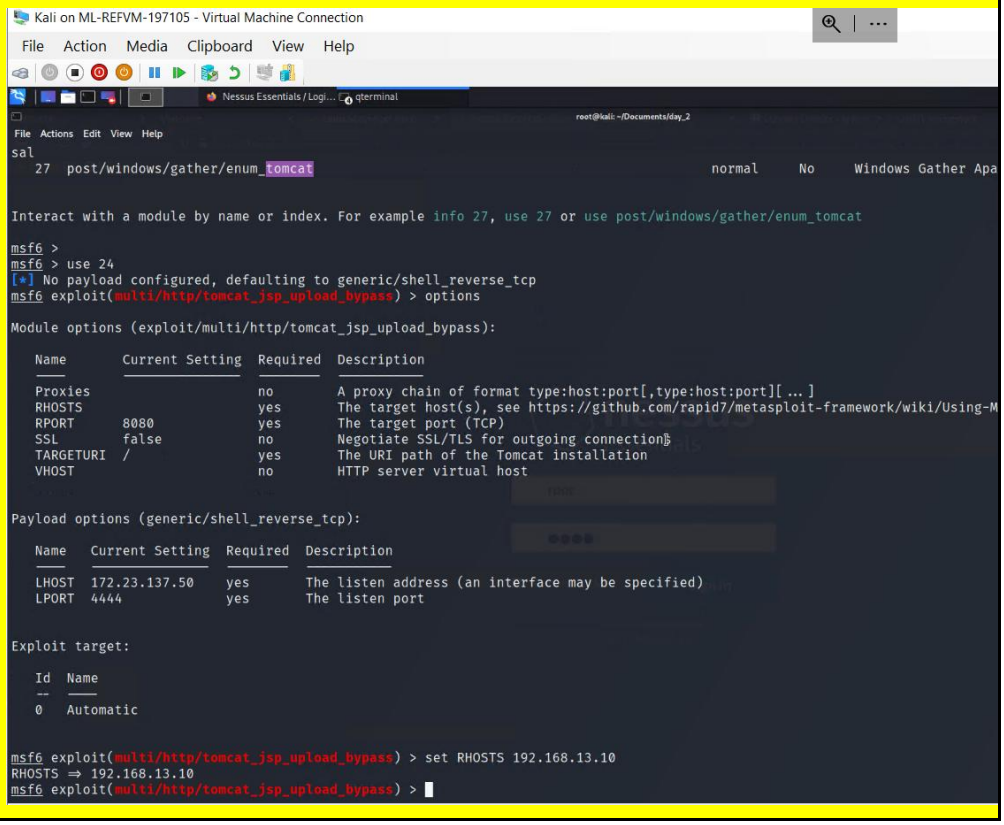


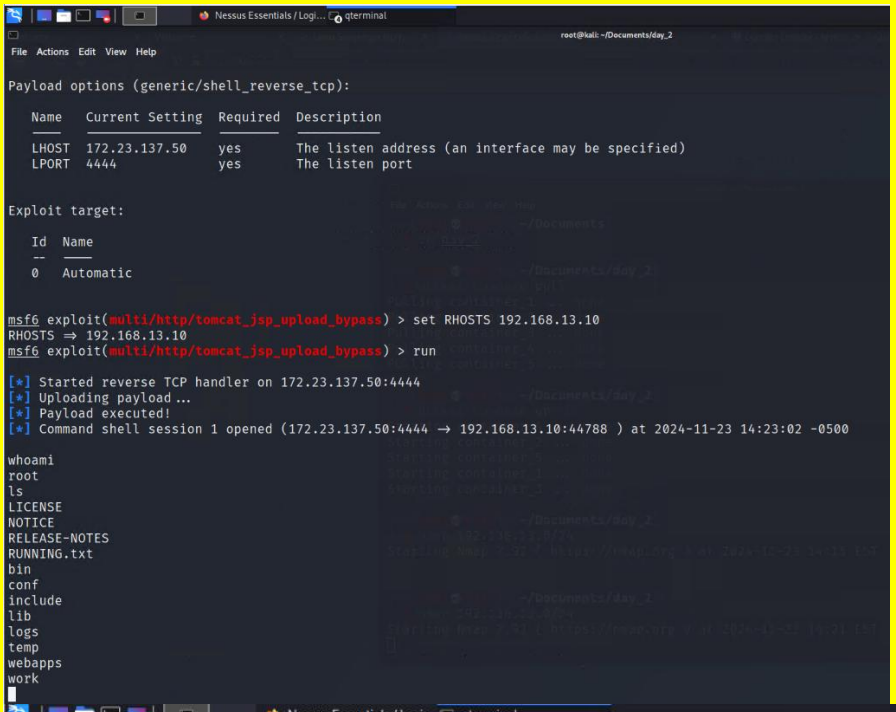


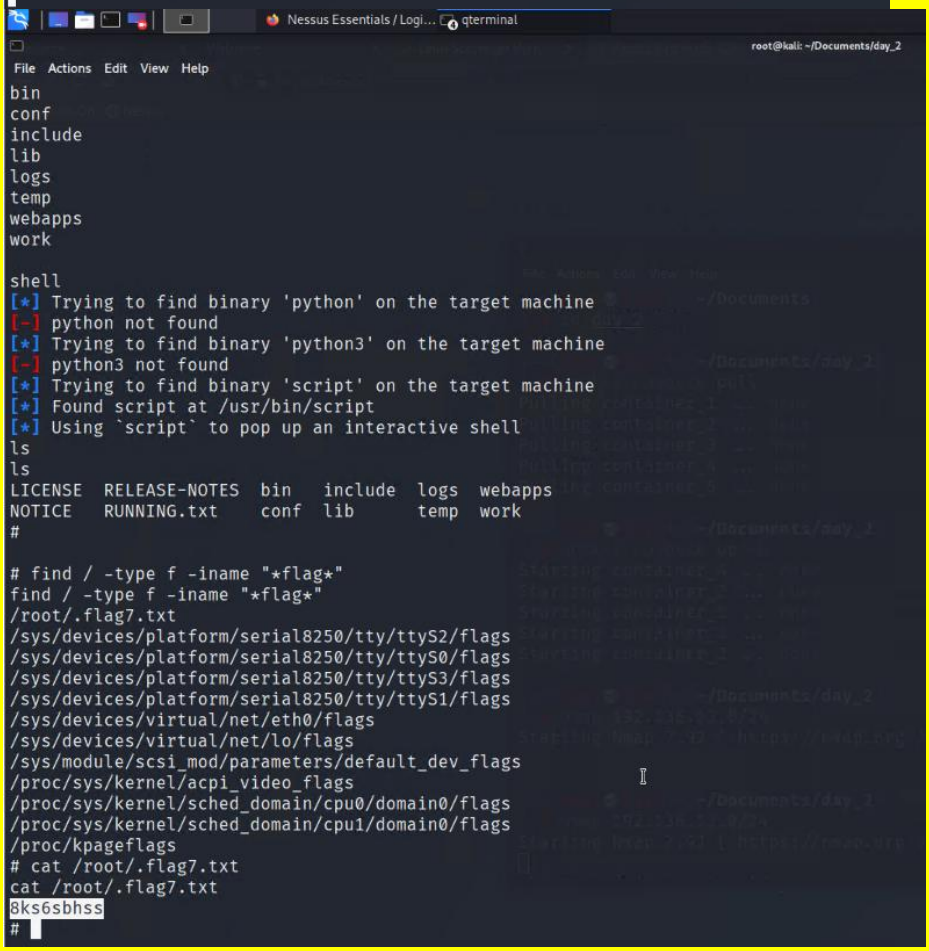


Picture was clipped above in the template so added a sliced one above to show ID.

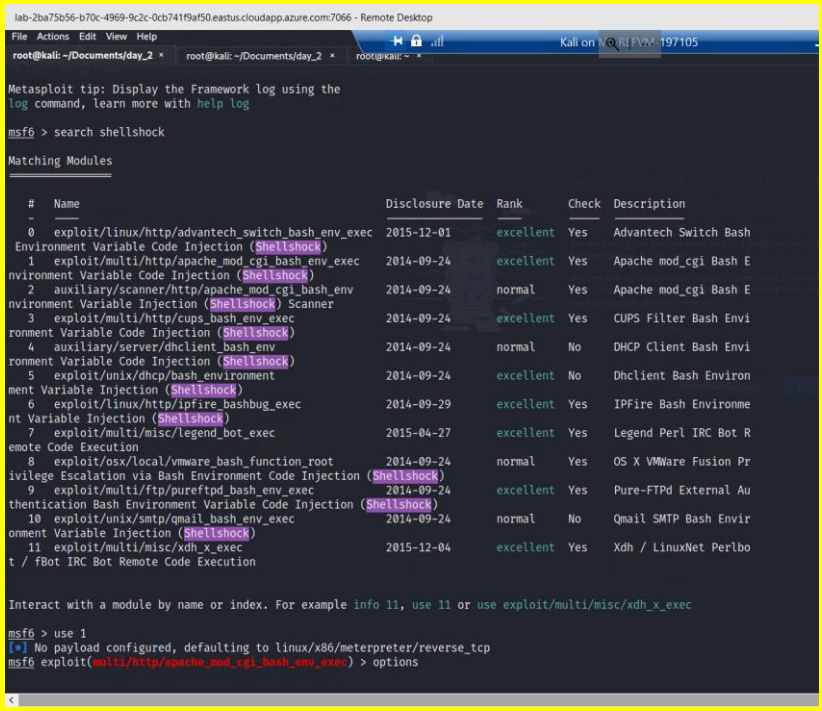
Affected Hosts	• 192.168.13.12
Remediation	• Apply the latest patches for Apache Struts2, restrict access to critical services, and monitor for abnormal behavior.

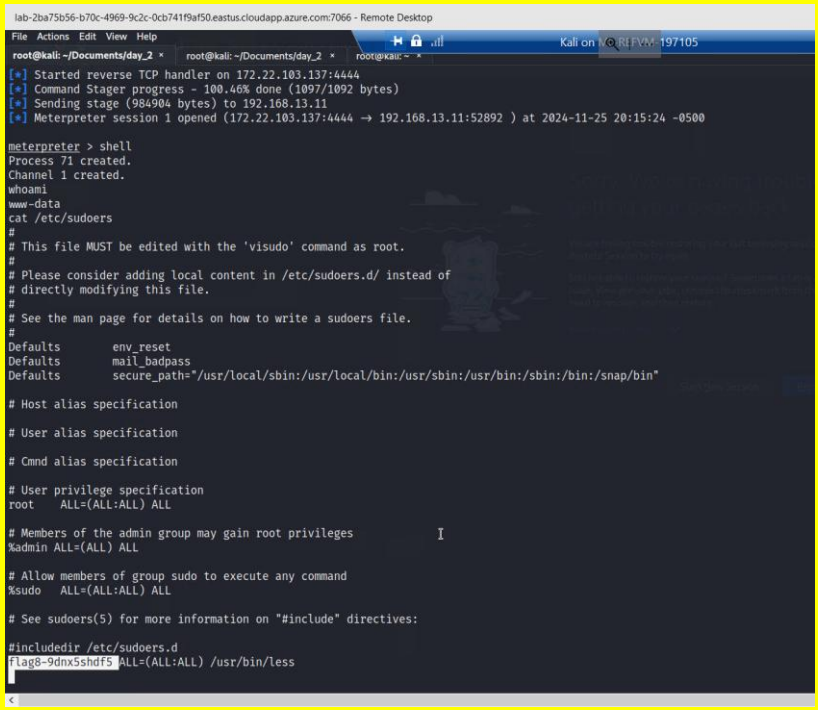
Vulnerability 7	Findings
Title	**Apache Tomcat Remote Code Execution (CVE-2017-12617)**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	The vulnerability in Apache Tomcat allows remote code execution via unauthenticated access to the service. This can be exploited to gain control over the system.
Images	



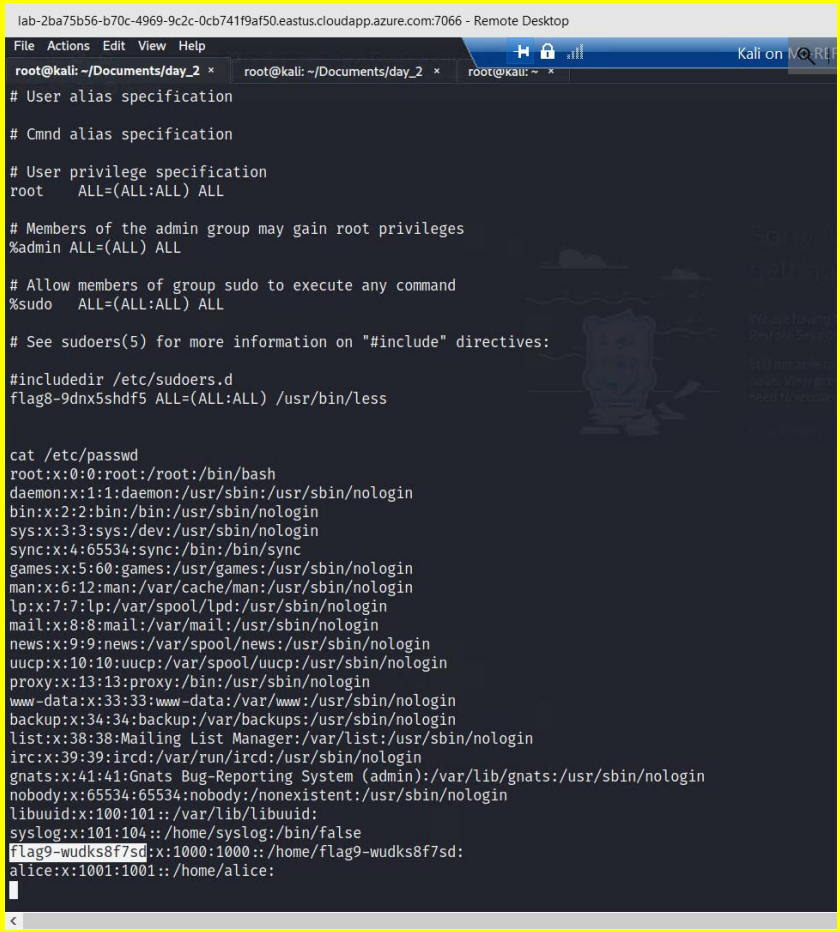


Affected Hosts	• 192.168.13.10
Remediation	Apply patches for Apache Tomcat, restrict access to Tomcat instances, and ensure proper configuration of the server.

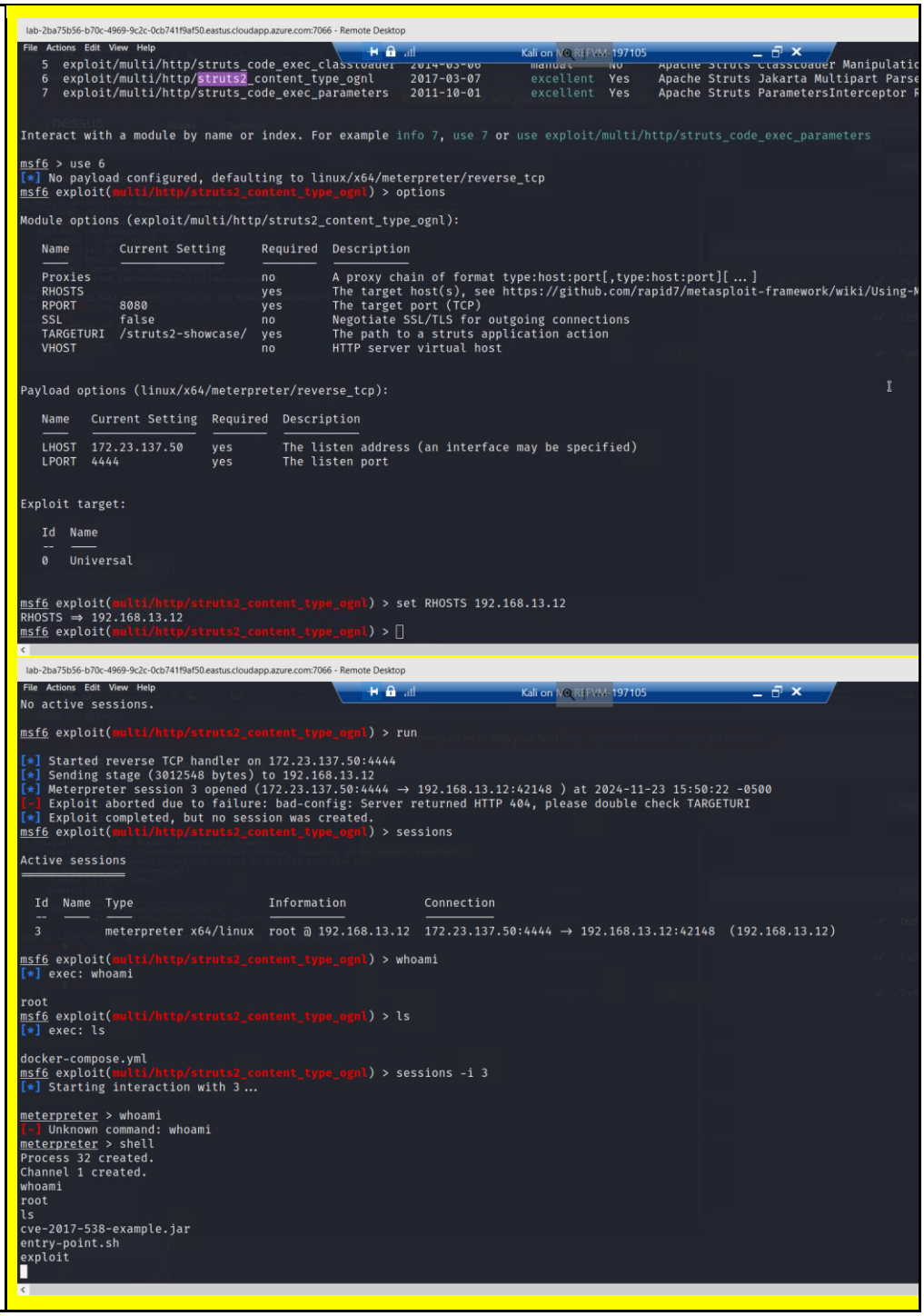
Vulnerability 8	Findings
Title	**Shellshock (CVE-2014-6271)**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	• The Shellshock vulnerability allows attackers to execute arbitrary commands via specially crafted environment variables, leading to remote code execution.
Images	

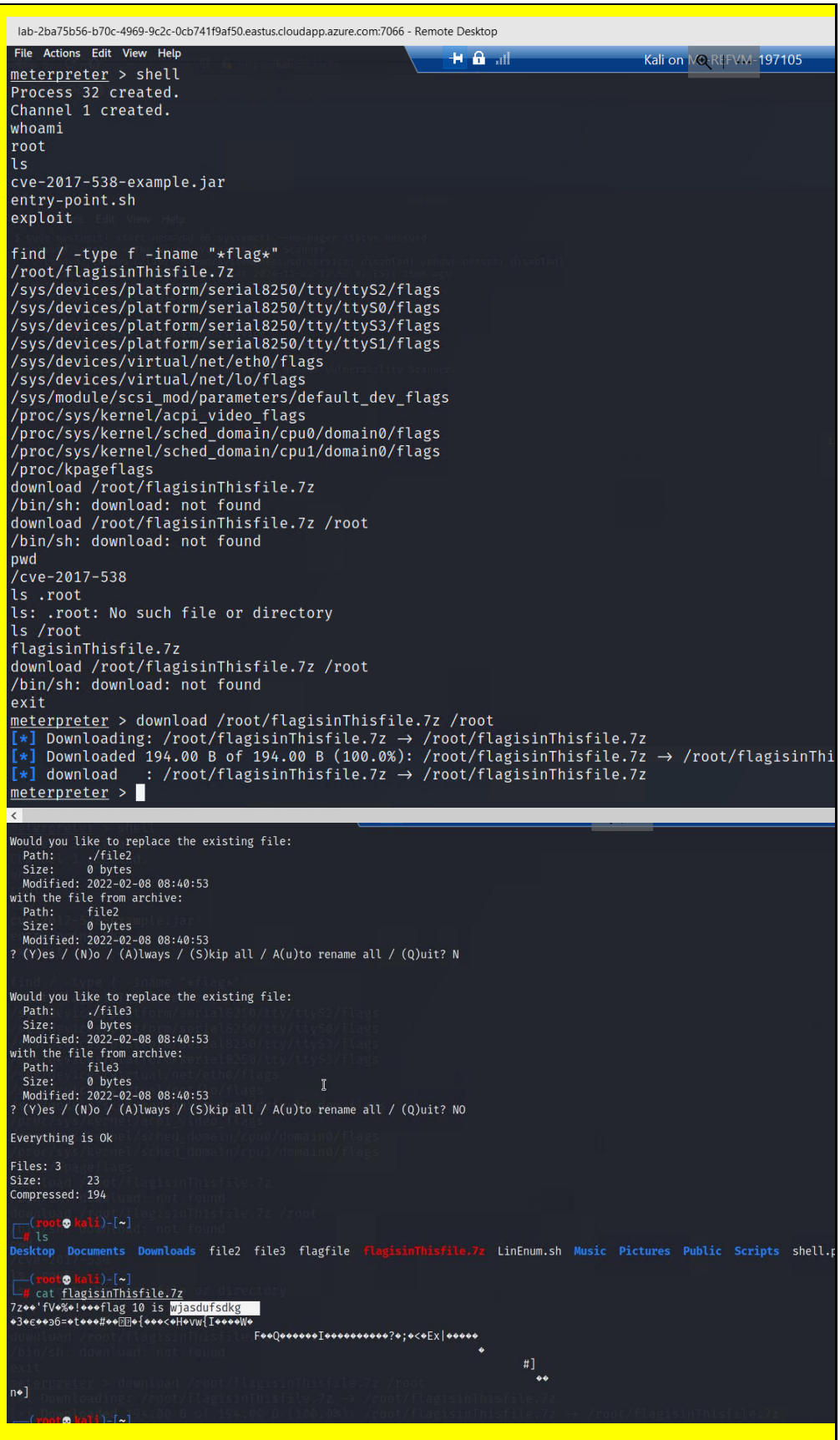
	
Affected Hosts	• 192.168.13.11
Remediation	Apply patches provided by the vendor, update Bash to the latest version, and limit the exposure of vulnerable services.

Vulnerability 9	Findings
Title	**Shellshock (CVE-2014-6271)**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	• The Shellshock vulnerability allows attackers to execute arbitrary commands via specially crafted environment variables, leading to remote code execution.

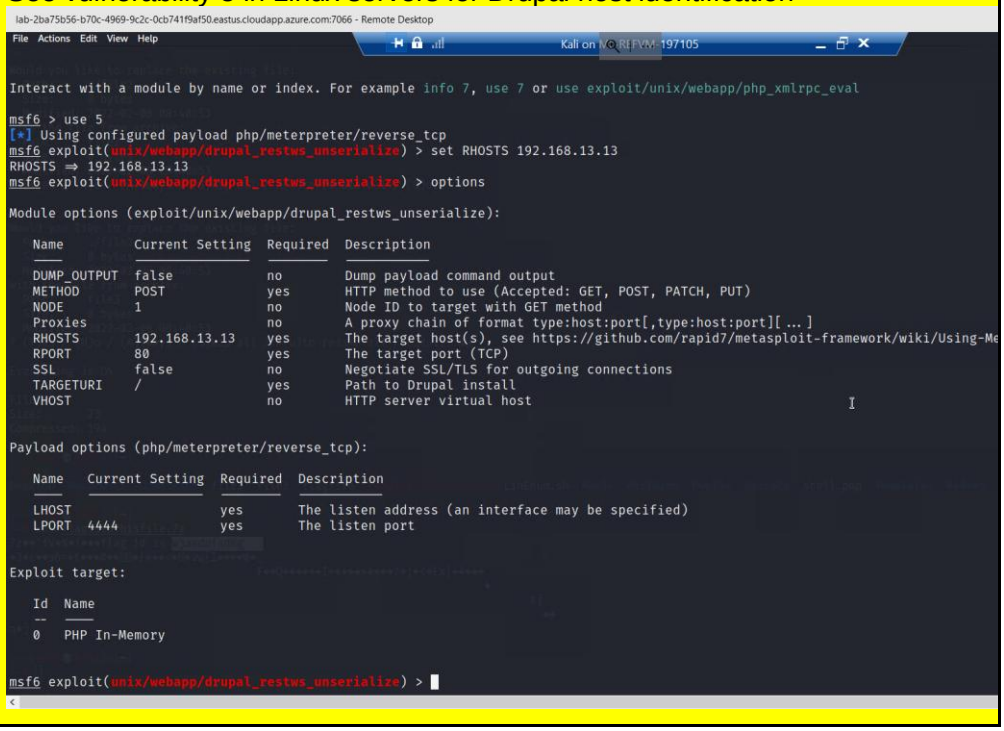
Images	 Obtained in the same shell of vulnerability 8 above.
Affected Hosts	• 192.168.13.11
Remediation	Apply patches provided by the vendor, update Bash to the latest version, and limit the exposure of vulnerable services.

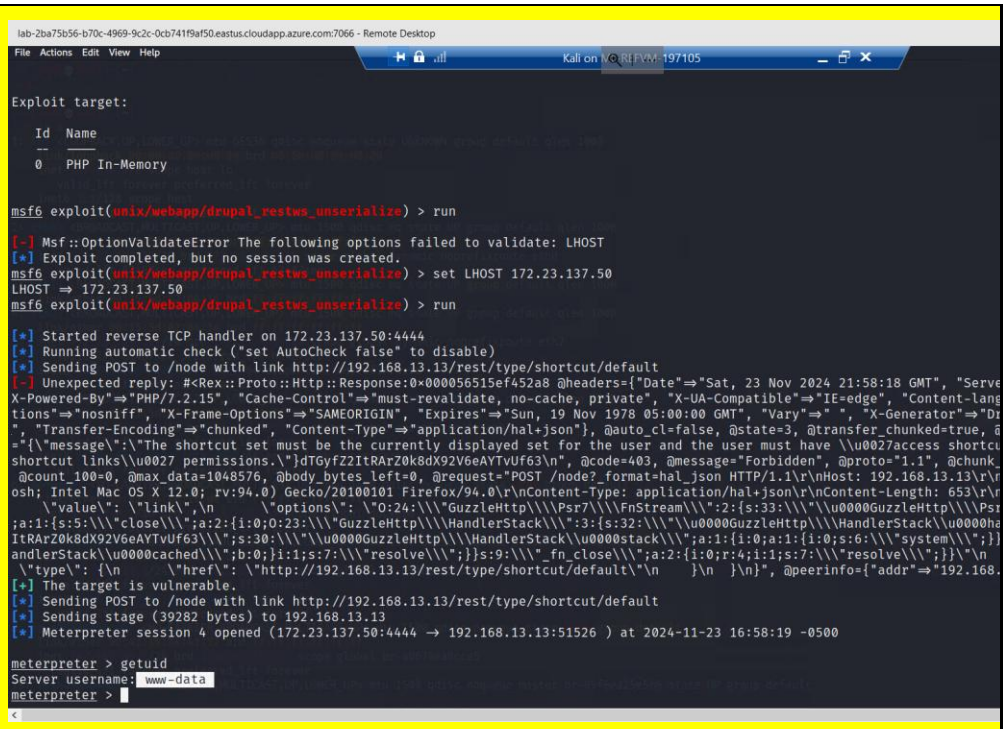
Vulnerability 10	**Findings**
Title	**Struts2 Remote Code Execution (CVE-2017-5638)**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	• The Apache Struts2 vulnerability (CVE-2017-5638) allows for remote code execution through a specially crafted request, allowing an attacker to gain shell access to the server.
Images	For Nessus scan see under Vulnerability 6 for Linux servers, few pictures are being shown here to give an idea.



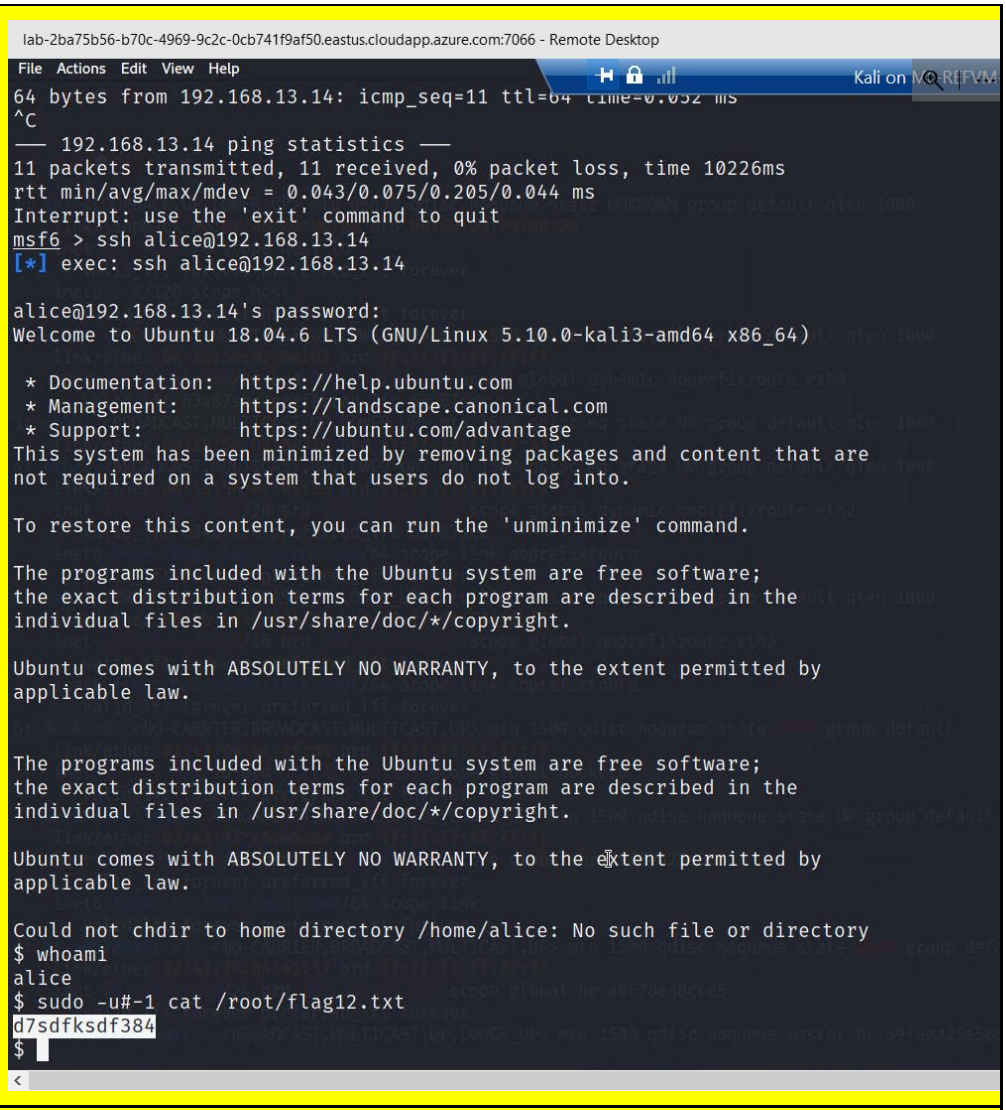


Affected Hosts	192.168.13.12
Remediation	• Apply the latest patches for Apache Struts2, restrict access to critical services, and monitor for abnormal behavior.

Vulnerability 11	Findings
Title	**Drupal Remote Code Execution (CVE-2019-6340)**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	• The vulnerability in Drupal (CVE-2019-6340) allows remote code execution via the drupal_restws_unserialize exploit. Attackers can exploit this flaw to execute arbitrary code on the server.
Images	See vulnerability 5 in Linux servers for Drupal host identification 

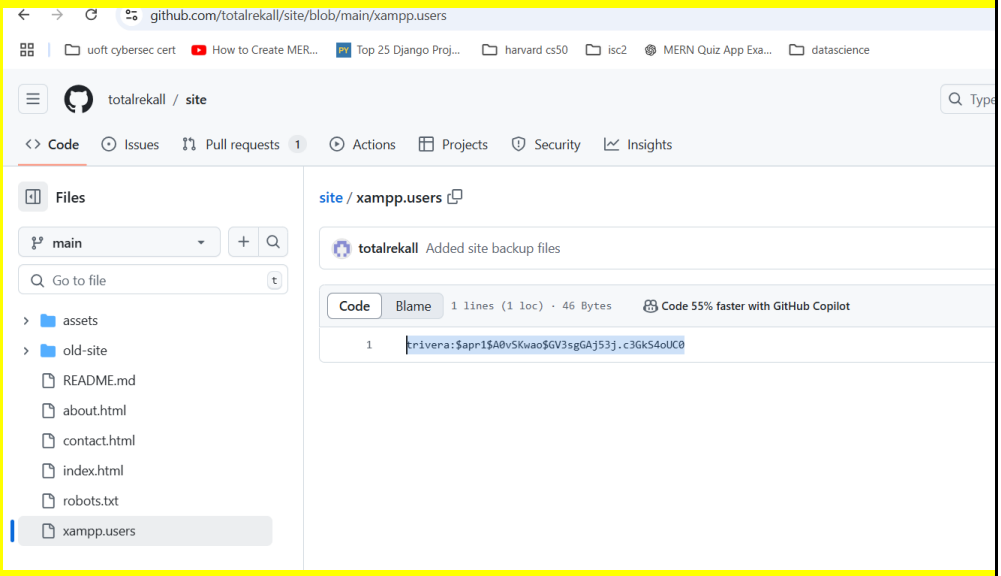
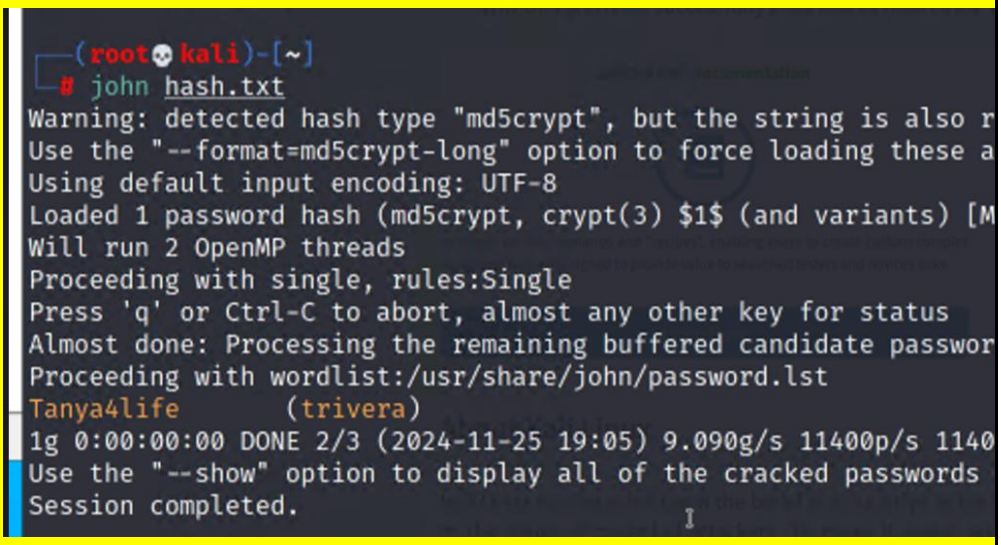
	
Affected Hosts	192.168.13.13
Remediation	Update Drupal to the latest version and apply security patches. Disable unused modules and limit access to the application.

Vulnerability 12	Findings
Title	**Privilege Escalation (CVE-2019-14287)**
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	This vulnerability allows an attacker to escalate privileges to root access by bypassing certain security restrictions.

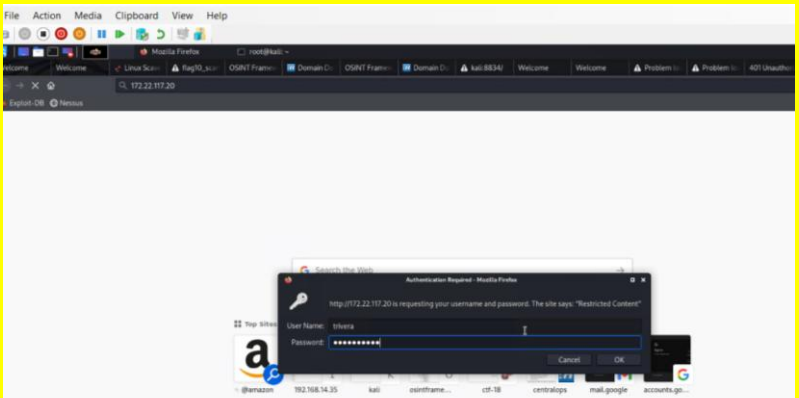
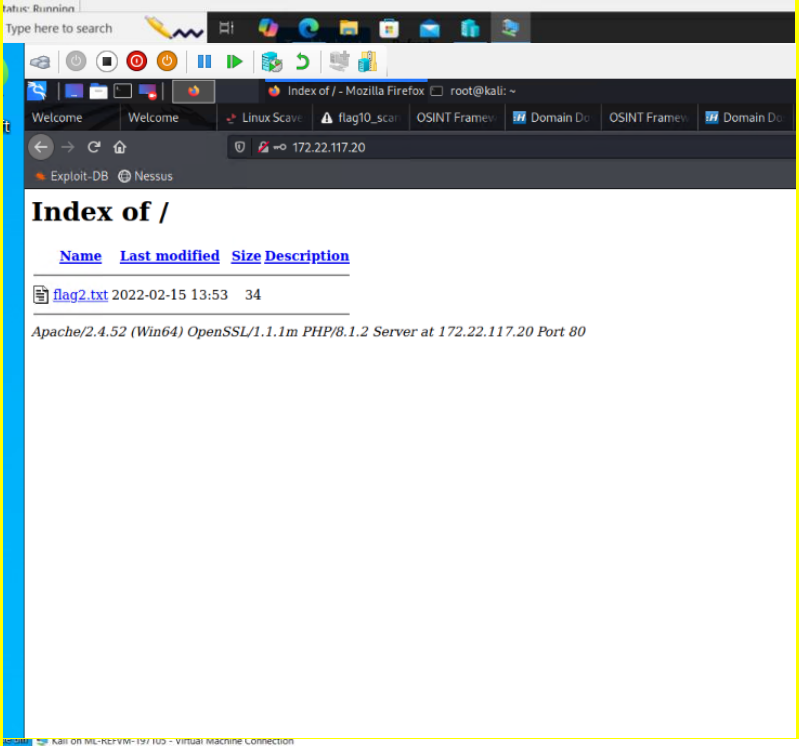
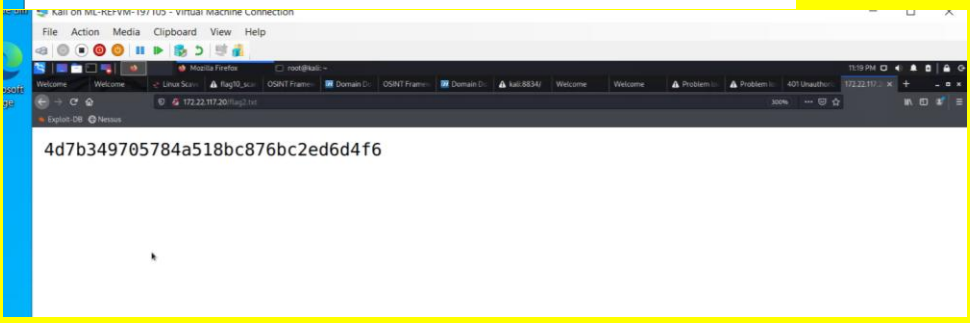
Images	
Affected Hosts	• 192.168.13.14
Remediation	Update the system to fix the vulnerability, apply least privilege policies, and restrict access to sensitive system files.

Windows Server Based Vulnerabilities Below

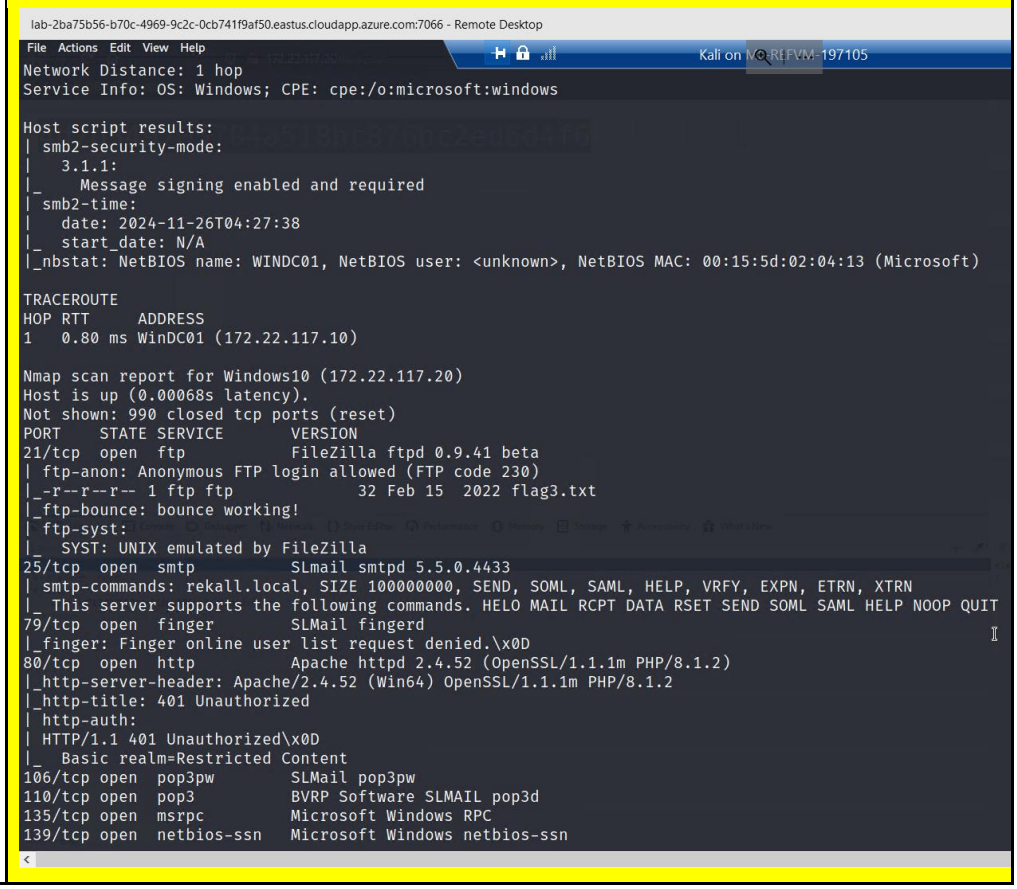
Vulnerability 1	Findings
Title	**OSINT: Open Source Exposed Vulnerabilities (Credentials)**
Type (Web app / Linux OS / WIndows OS)	Windows OS (Win10 Machine)
Risk Rating	High
Description	Open-source-based vulnerabilities on Windows 10 servers, particularly involving exposed credentials (e.g., hard-coded passwords, API keys, or

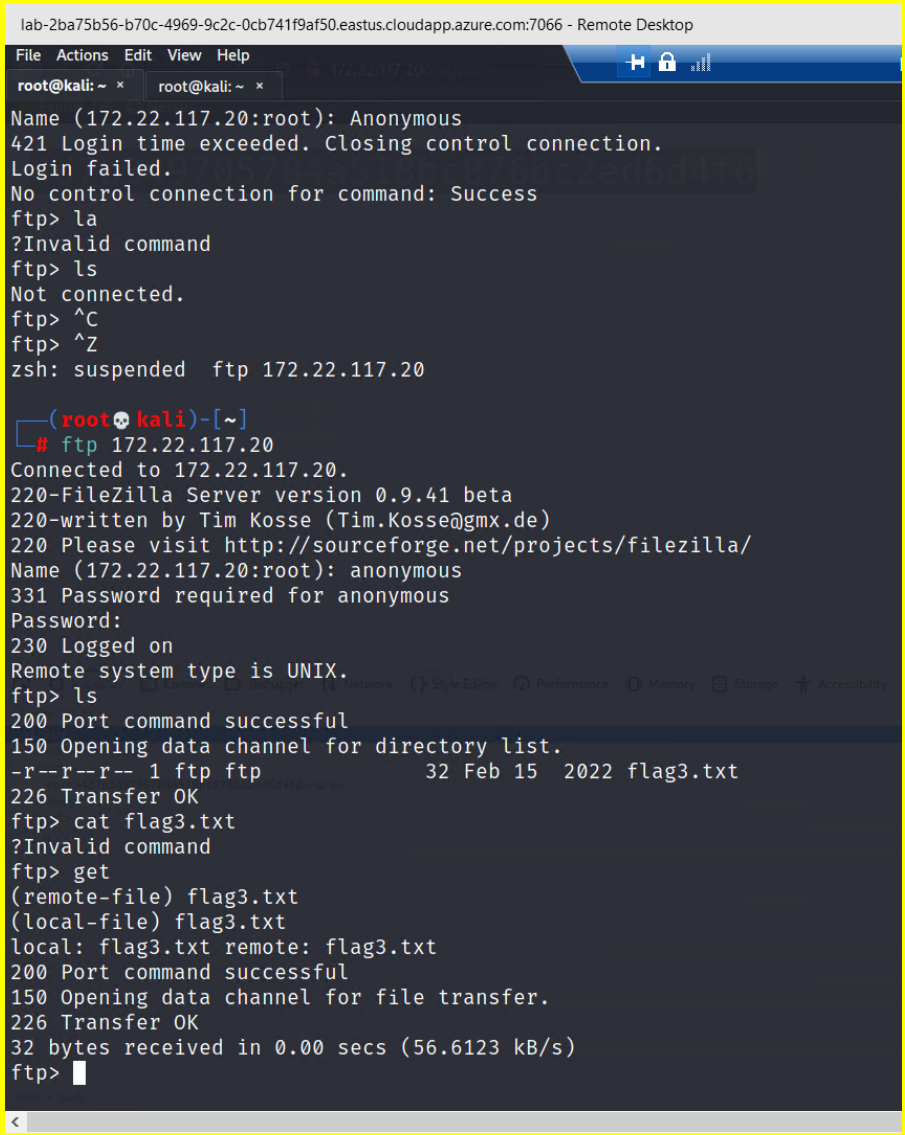
	database credentials), occur when sensitive information is stored insecurely within open-source software or its configurations. Attackers can exploit these vulnerabilities to gain unauthorized access or escalate privileges on the server, especially if credentials are publicly accessible or poorly protected.
Images	 
Affected Hosts	• 172.22.117.20
Remediation	To prevent this, ensure that credentials are not hard-coded into open-source software or configurations, use environment variables or secret management systems to store sensitive data securely, regularly audit code for exposed credentials, and apply access control policies. Additionally, ensure all open-source software is patched and configured securely to limit exposure.

Vulnerability 2	Findings
Title	**HTTP Enumeration**
Type (Web app / Linux OS / WIndows OS)	Windows OS (Win10 Machine)

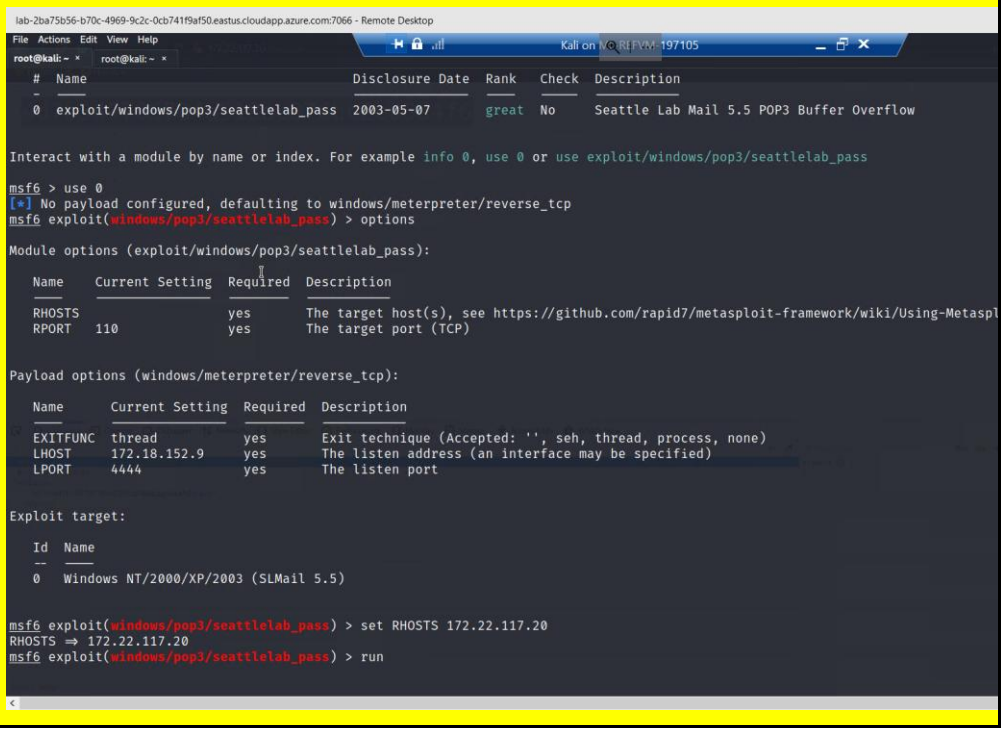
Risk Rating	High
Description	HTTP enumeration refers to the process of discovering information about a web application or server by analyzing HTTP responses, headers, or error messages. Attackers may use this technique to gain insight into the structure, software, or potential vulnerabilities of a web application, such as identifying server types, technology stacks, or unprotected resources.
Images	  

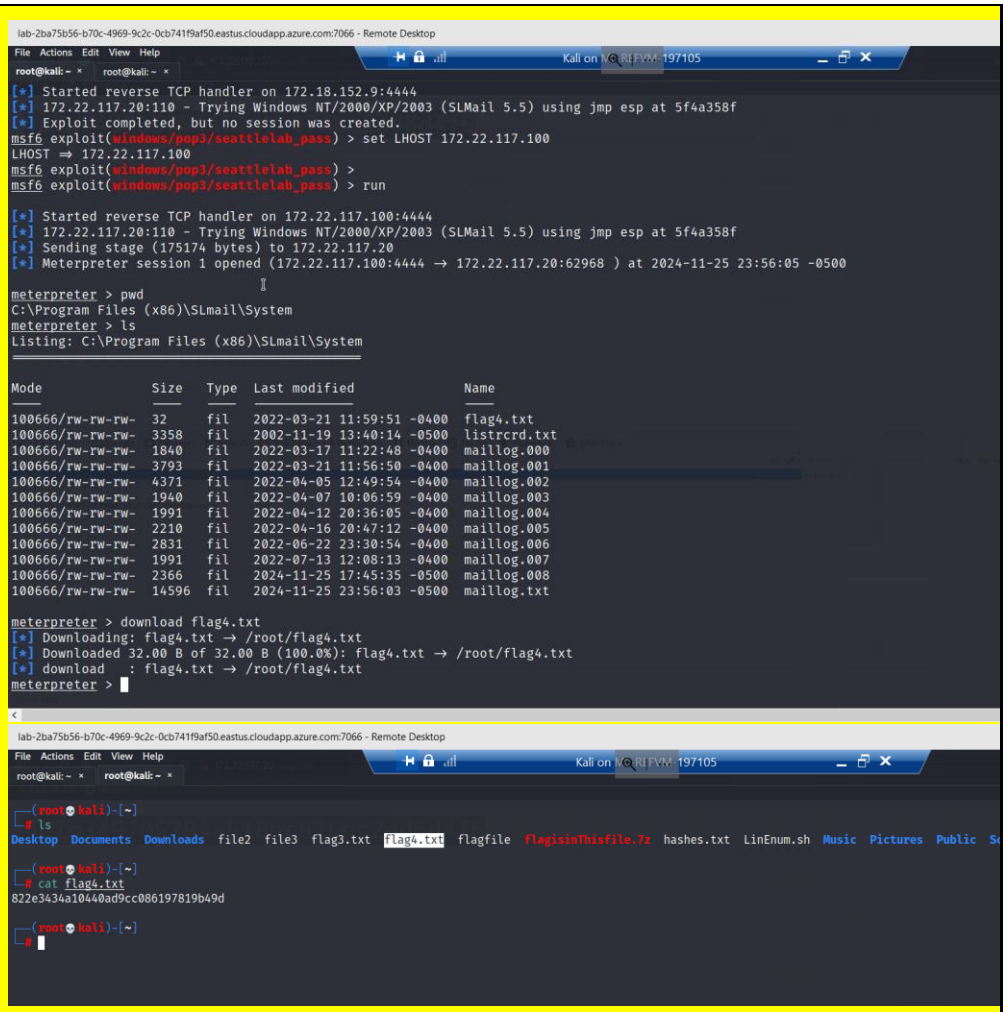
Affected Hosts	• 172.22.117.20
Remediation	To mitigate HTTP enumeration, configure web servers and applications to minimize the amount of information exposed in HTTP headers and error messages. Use custom error pages, disable unnecessary HTTP methods (e.g., TRACE, OPTIONS), and implement proper access control and security mechanisms to prevent unauthorized access to sensitive endpoints.

Vulnerability 3	Findings
Title	**FTP Enumeration**
Type (Web app / Linux OS / WIndows OS)	Windows OS (Win10 Machine)
Risk Rating	Medium
Description	FTP enumeration is a technique where attackers probe an FTP server to gather information about available directories, files, and users by analyzing responses to specific FTP commands. This can lead to the discovery of sensitive files, misconfigurations, or user credentials, potentially opening the door for unauthorized access.
Images	



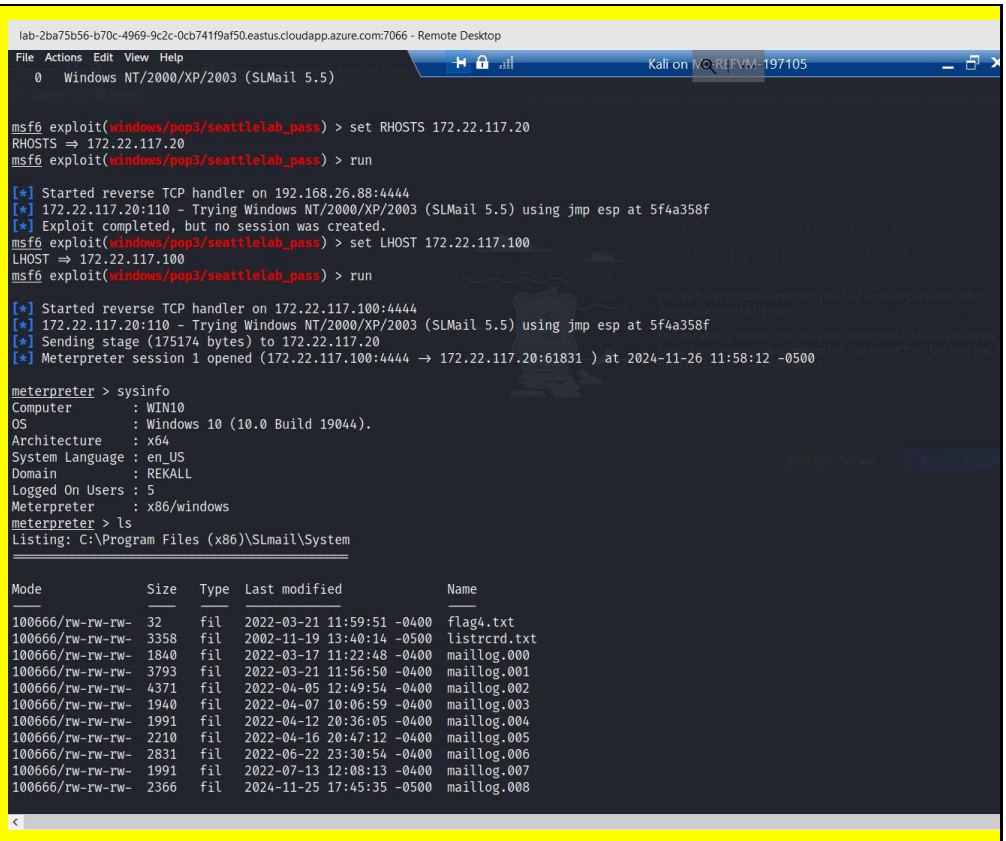
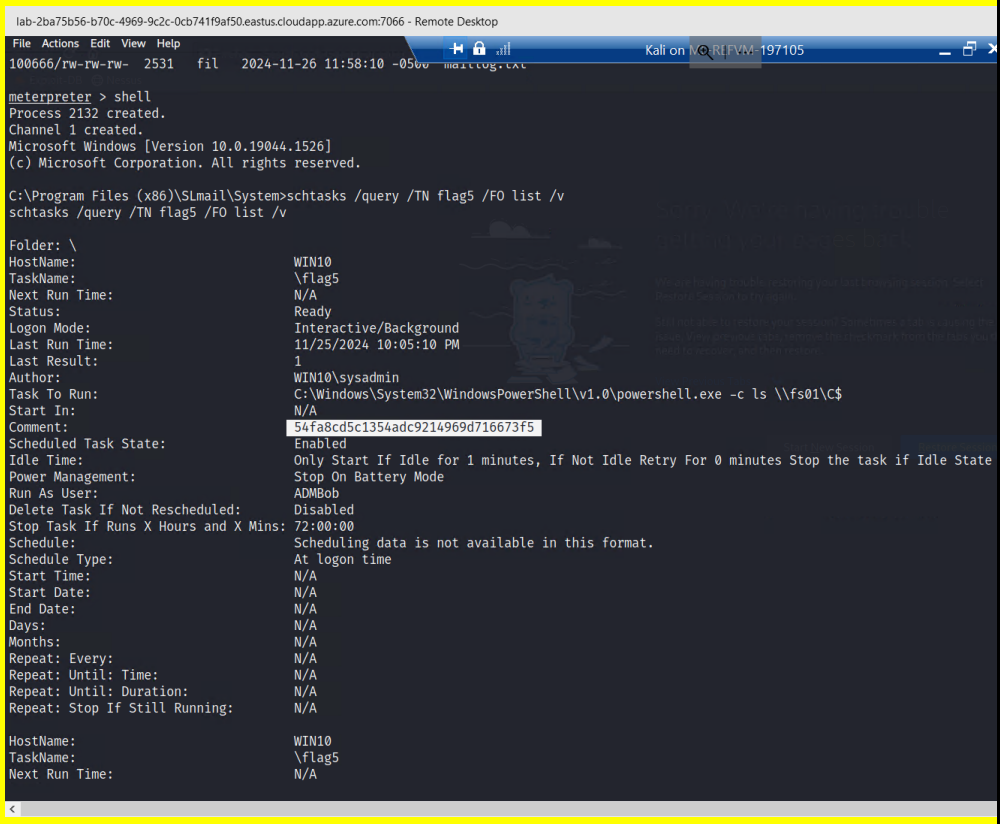
Affected Hosts	• 172.22.117.20
Remediation	To prevent FTP enumeration, disable anonymous FTP access, configure proper file and directory permissions, use strong authentication methods, and consider using secure alternatives like SFTP or FTPS. Additionally, implement logging and monitoring to detect and block suspicious FTP activity.

Vulnerability 4	Findings
Title	**SLMail Remote Code Execution (CVE-2017-16372)**
Type (Web app / Linux OS / WIndows OS)	Windows OS (Win10 Machine)
Risk Rating	High
Description	• The vulnerability in SLMail allows attackers to execute arbitrary code remotely. Exploiting this can provide an attacker with system-level access.
Images	

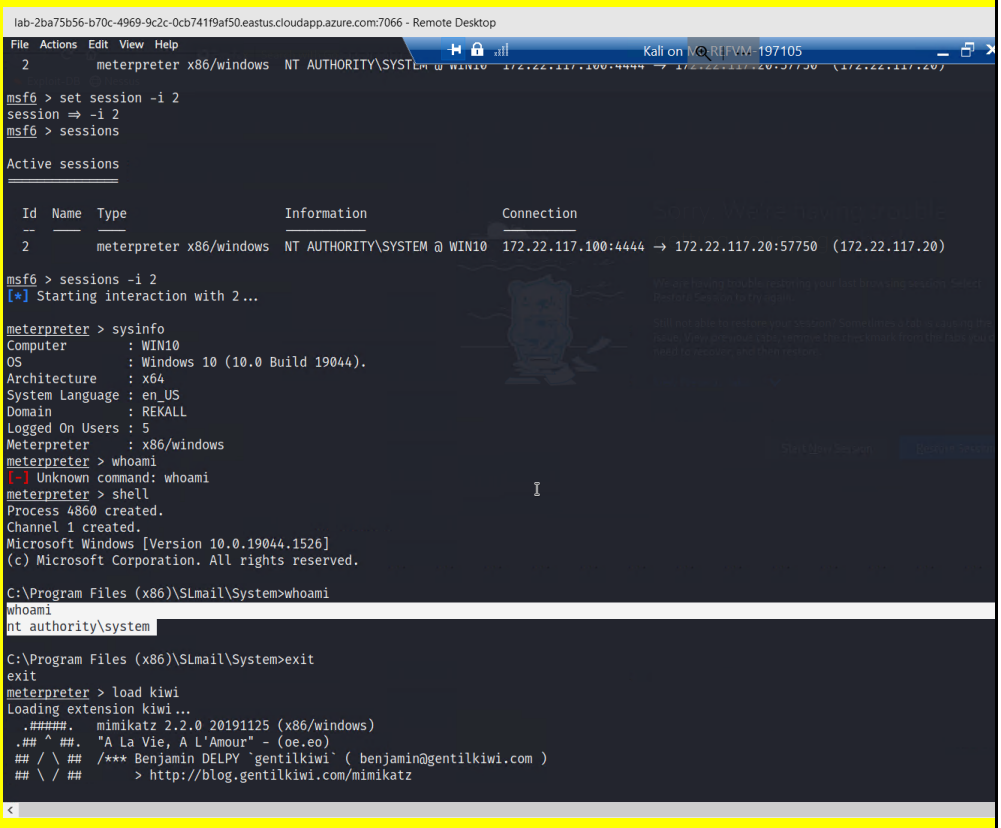


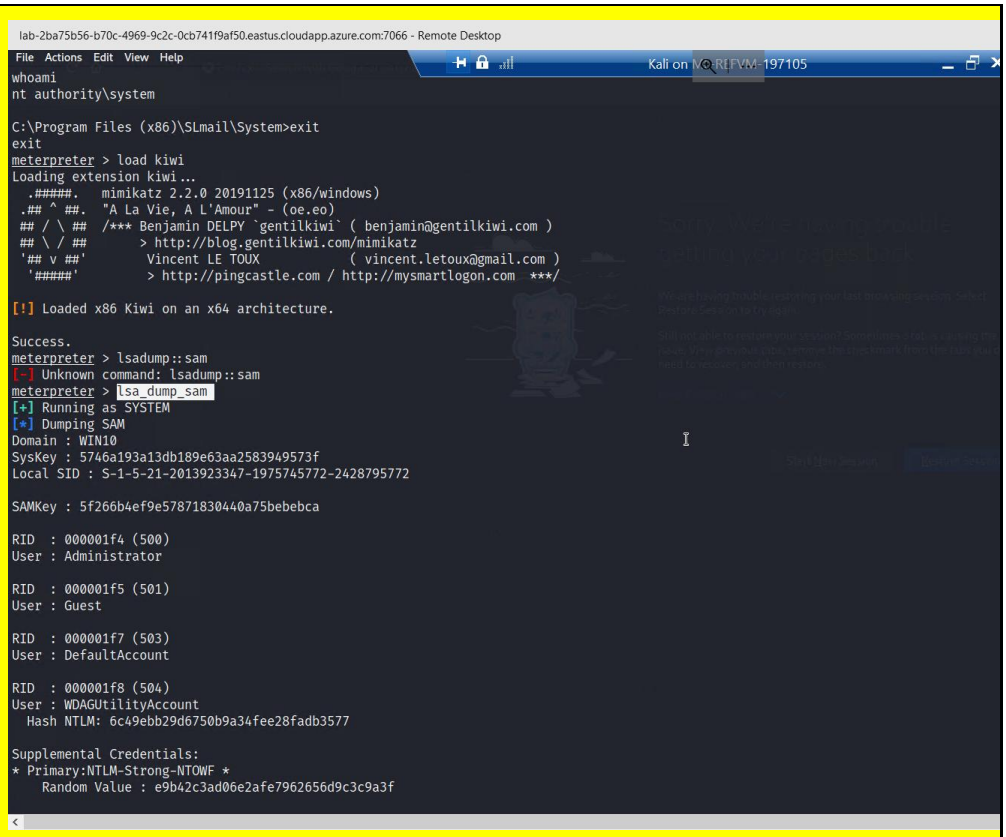
Affected Hosts	• 172.22.117.20
Remediation	Apply the vendor's patch for SLMail, disable unnecessary services, and ensure only authorized users can access the system.

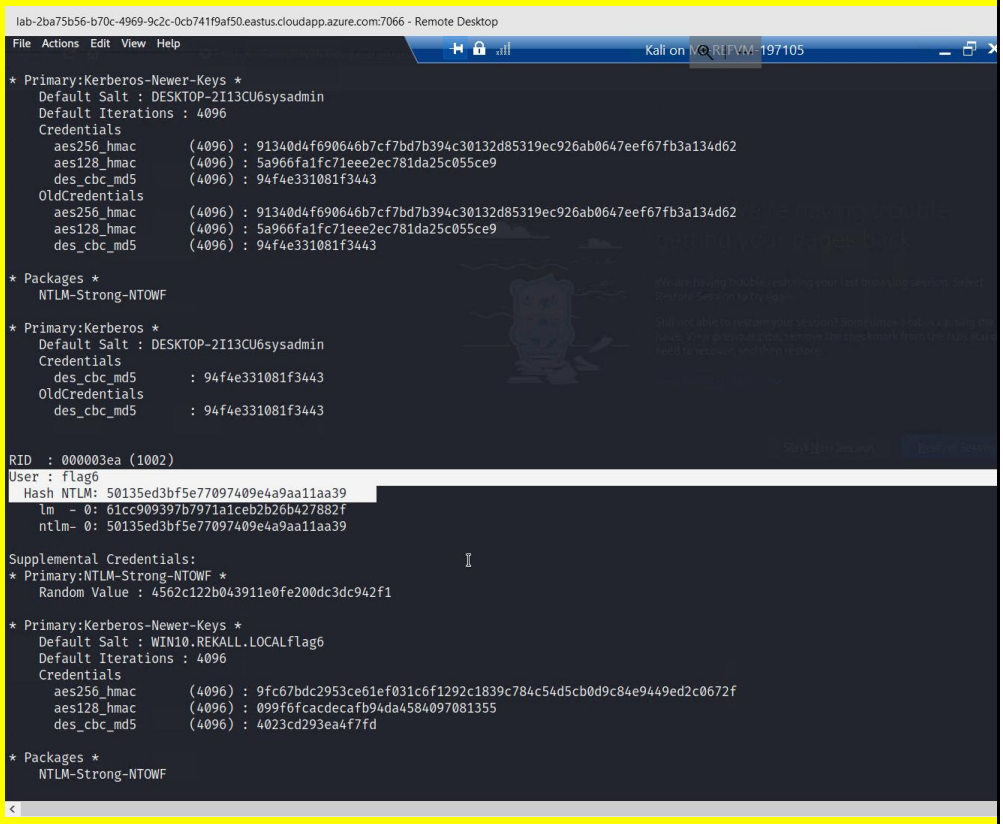
Vulnerability 5	Findings
Title	**Common Tasks/ Scheduled Tasks**
Type (Web app / Linux OS / WIndows OS)	Windows OS (Win10 Machine)
Risk Rating	High
Description	After exploitation, attackers may leverage **Windows 10 scheduled tasks** to maintain persistence or automate malicious actions. Scheduled tasks allow the execution of commands or scripts at specified times or under certain conditions, which can be hijacked to run malware, escalate privileges, or retrieve sensitive data without the user's knowledge.

Images	 
Affected Hosts	• 172.22.117.20
Remediation	To mitigate the risks associated with scheduled tasks, regularly audit and monitor scheduled tasks for any unauthorized or suspicious entries. Ensure

	that tasks are created with proper security settings, restrict user permissions to create tasks, and use endpoint protection software to detect malicious activities. Additionally, configure systems to notify administrators of task modifications. (e.g., TRACE, OPTIONS), and implement proper access control and security mechanisms to prevent unauthorized access to sensitive endpoints.

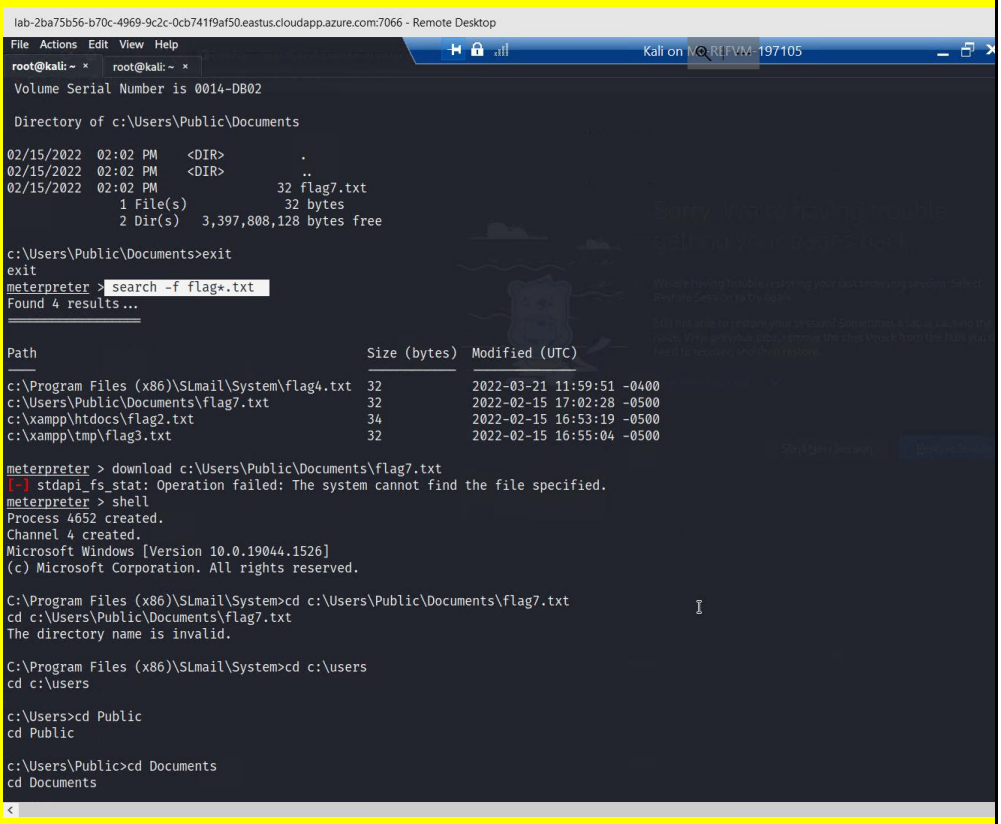
Vulnerability 6	Findings
Title	**User Enumeration**
Type (Web app / Linux OS / WIndows OS)	Windows OS (Win10 Machine)
Risk Rating	Medium to High
Description	User enumeration is a technique used by attackers to identify valid usernames within a system, application, or service. This can be done by observing system responses to authentication attempts (e.g., login failure messages, timing differences, or user-specific error codes). Successful enumeration can help attackers target specific accounts for brute force, social engineering, or credential stuffing attacks.
Images	

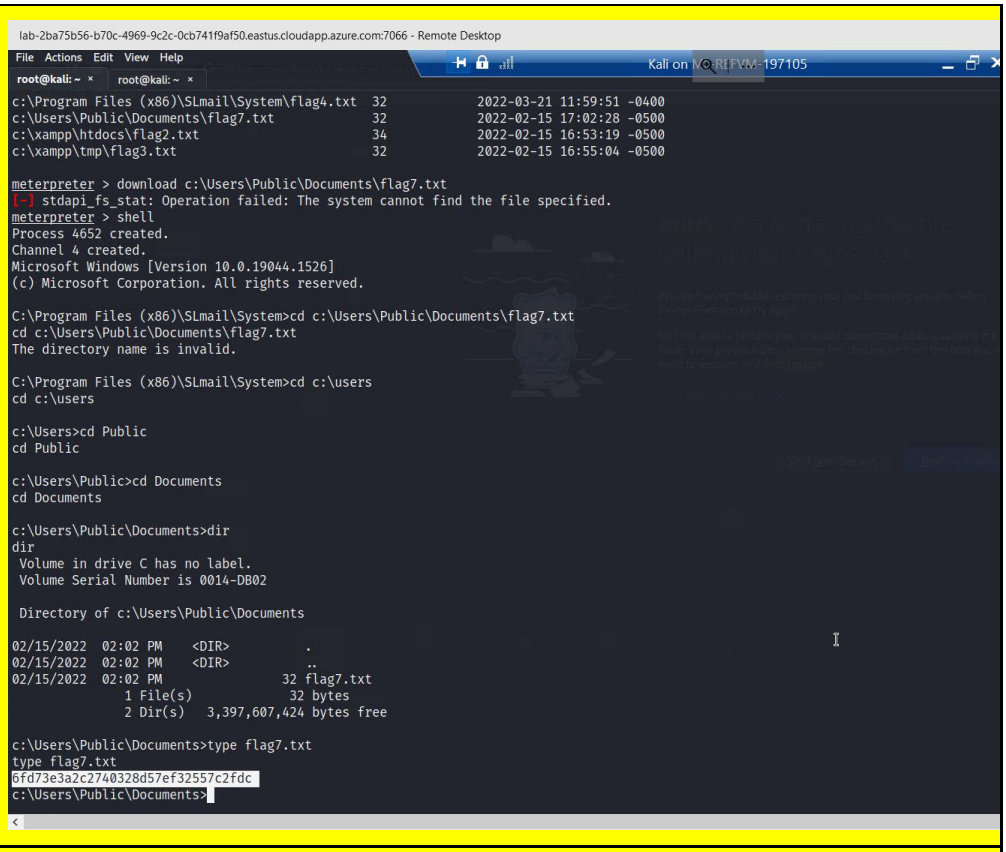




Affected Hosts	• 172.22.117.20
Remediation	To prevent user enumeration, standardize error messages to ensure they do not reveal whether a username or password is incorrect, implement account

	lockout or rate-limiting mechanisms to prevent brute force attacks, and monitor failed login attempts for suspicious patterns. Additionally, consider using multi-factor authentication (MFA) to further secure accounts.

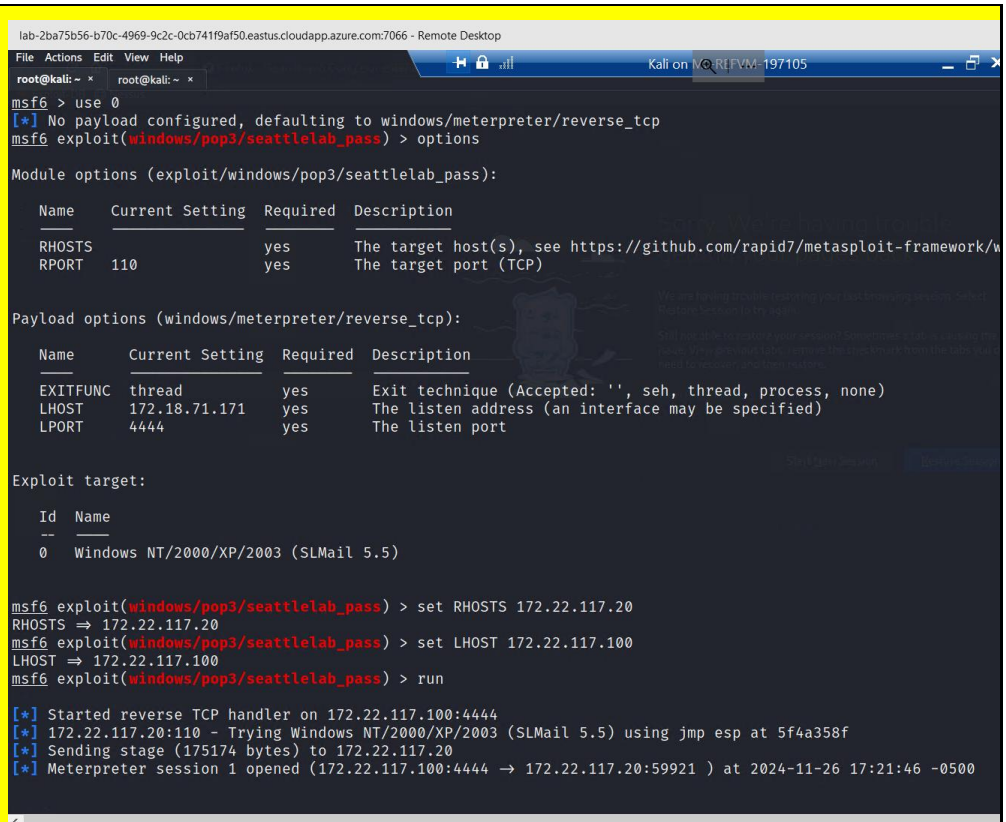
Vulnerability 7	Findings
Title	**File Enumeration**
Type (Web app / Linux OS / WIndows OS)	Windows OS (Win10 Machine)
Risk Rating	Medium to High
Description	File enumeration is a technique where an attacker identifies and lists accessible files or directories within a system by exploiting weak configurations or file handling mechanisms. This can involve probing a web server or system for file names, extensions, or paths, revealing sensitive or unauthorized files (e.g., configuration files, backups, or logs) that may lead to further exploitation.
Images	

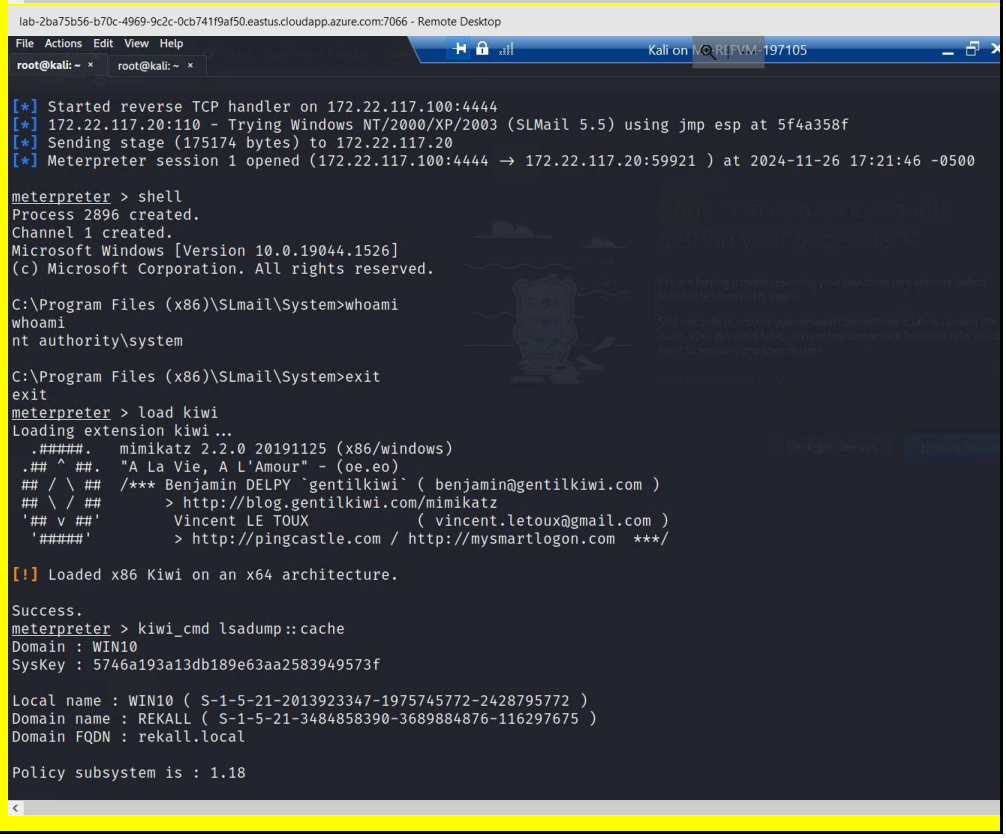


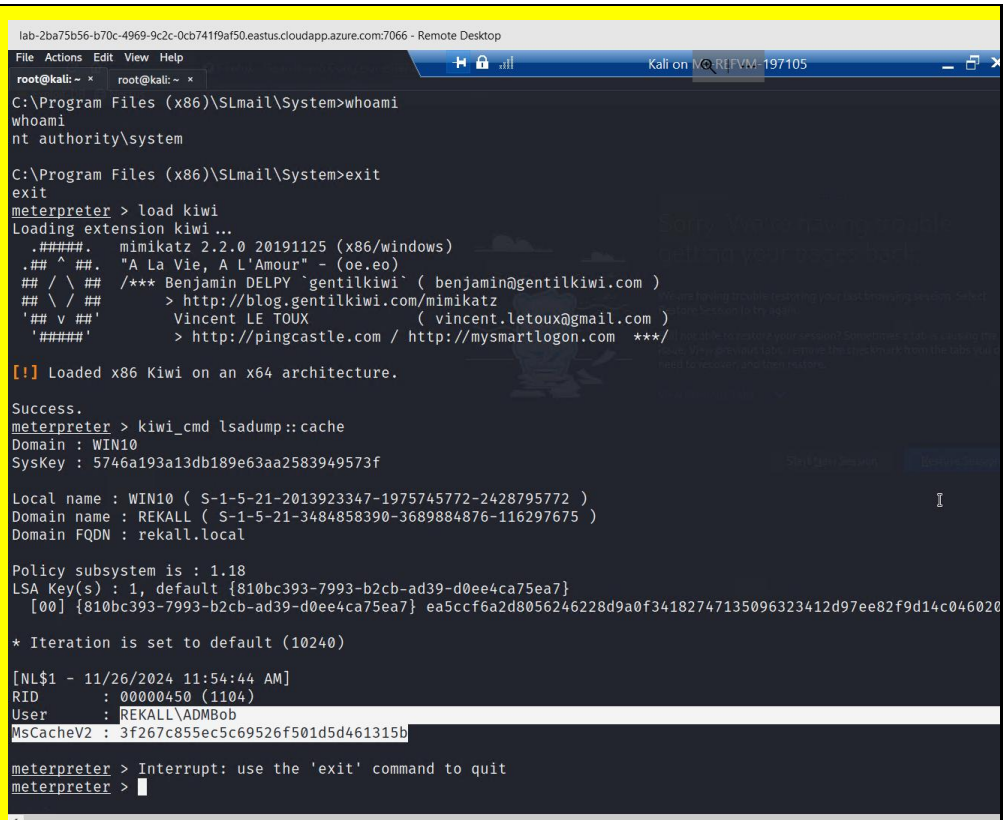
Affected Hosts	172.22.117.20 (Windows 10 machine)
Remediation	To prevent file enumeration, ensure proper file access controls and permissions, implement input validation to prevent directory traversal attacks, and restrict access to sensitive files using access control lists (ACLs). Additionally, configure web servers to prevent directory listing and disable access to unnecessary or sensitive files.

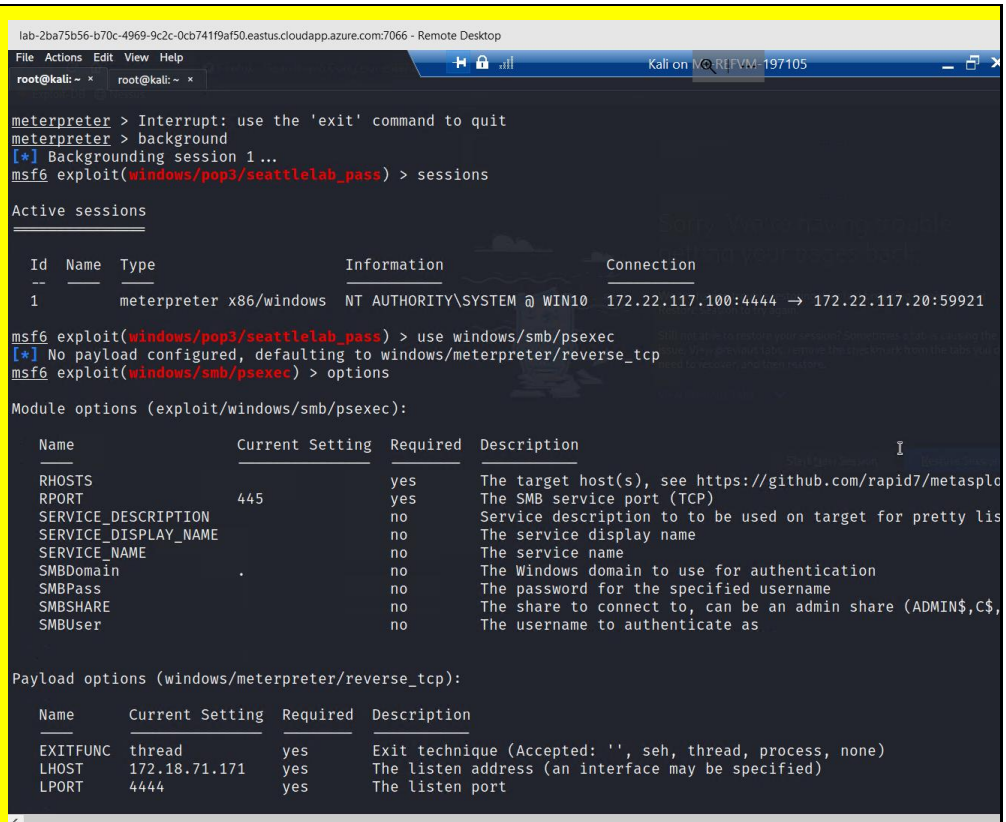
Vulnerability 8	Findings
Title	**User Enumeration 2**
Type (Web app / Linux OS / WIndows OS)	Windows OS (Windows 10 machine) & WinDC01
Risk Rating	• High to Critical; This risk is considered **critical** if it leads to full system compromise or major business impact.
Description	User enumeration leading to lateral movement occurs when attackers successfully identify valid usernames within a network or system through techniques like login probes or error message analysis. Once valid user accounts are identified, attackers can target those accounts for further attacks, such as password guessing, credential stuffing, or phishing. This can lead to lateral movement across the network, eventually gaining access to more critical systems or admin credentials, enabling them to escalate privileges and take full control.

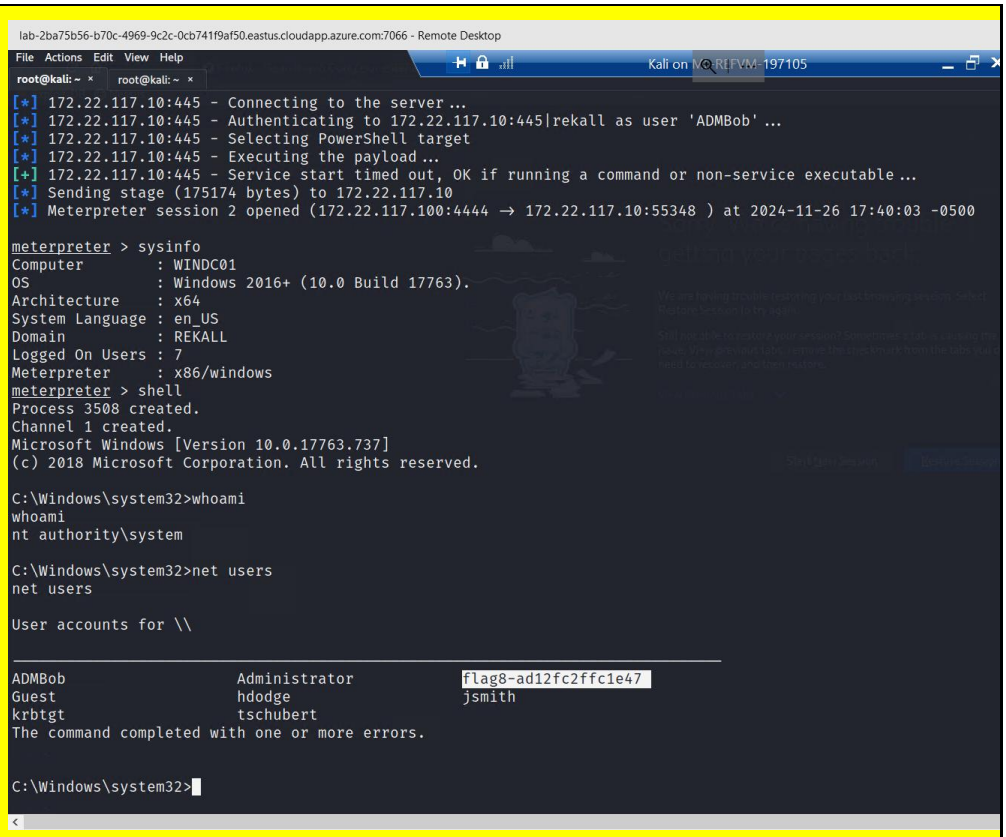
Images





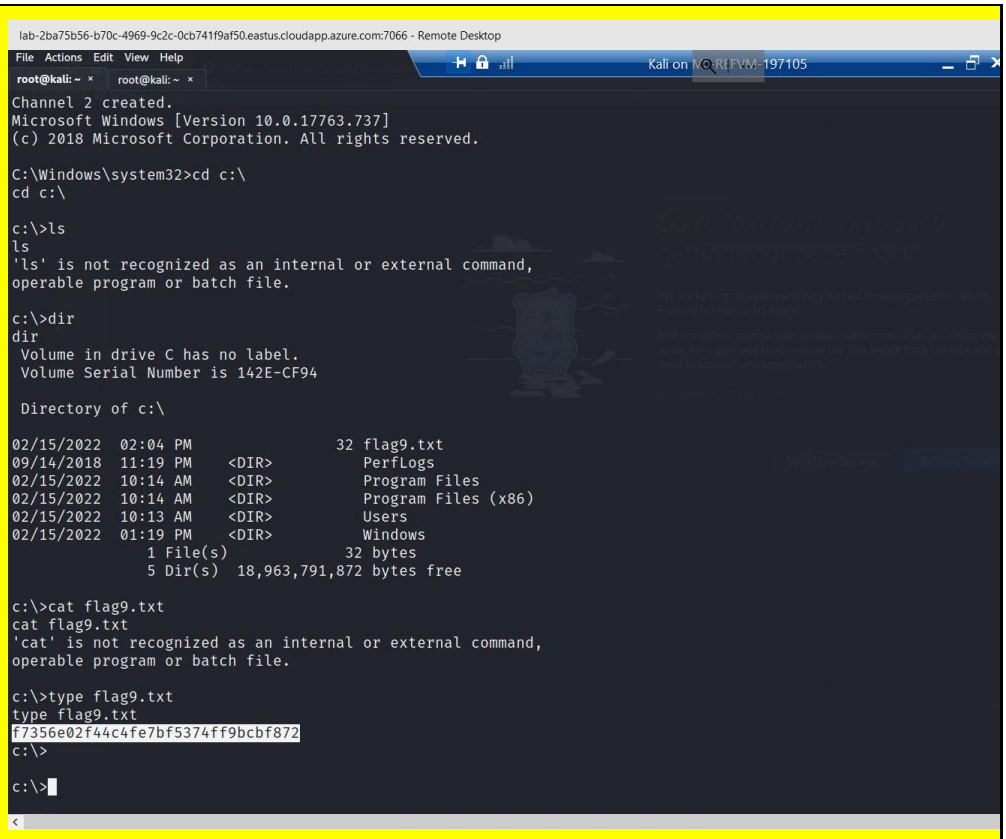




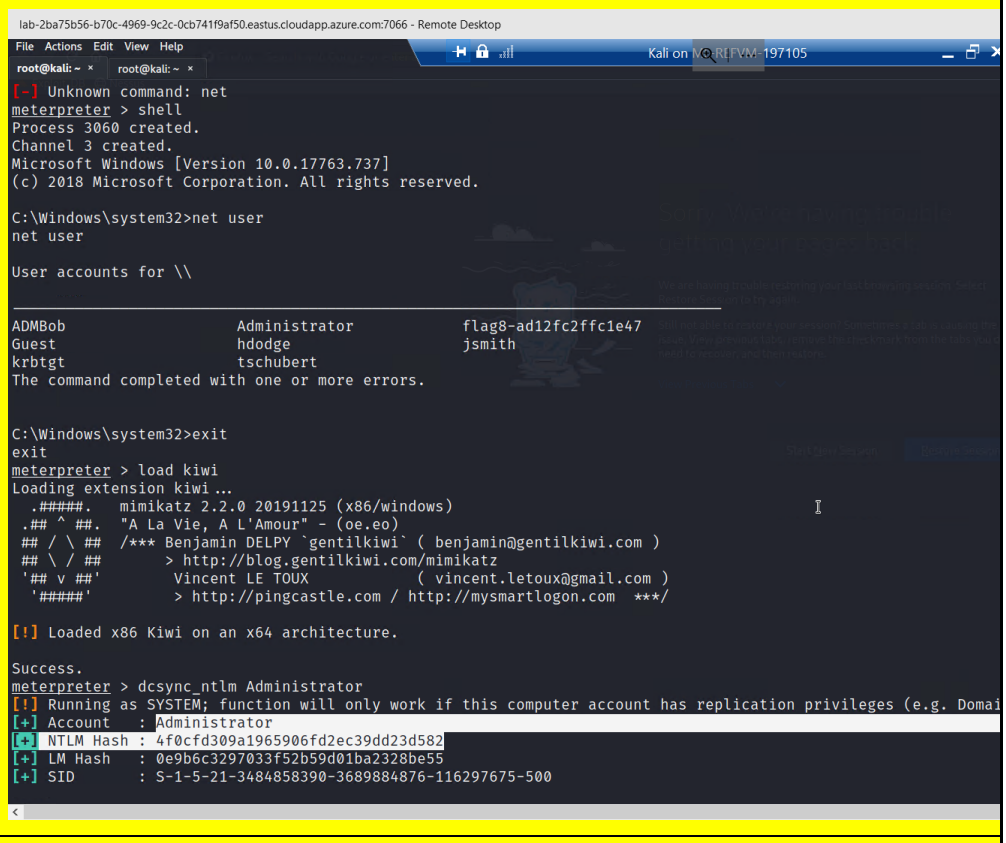


Affected Hosts	• 172.22.117.10 (WinDC01) & 172.22.117.20 (Windows 10 machine) • LHOST 172.22.117.100
Remediation	To prevent user enumeration and mitigate lateral movement, standardize error messages for login failures, implement account lockout mechanisms or rate limiting to slow down brute force attempts, and use multi-factor authentication (MFA). Additionally, employ network segmentation, ensure least privilege access policies, monitor for suspicious login patterns, and regularly audit user accounts and permissions.

Vulnerability 9	Findings
Title	**Escalating Access**
Type (Web app / Linux OS / WIndows OS)	172.22.117.10 (WinDC01) Windows OS
Risk Rating	Critical
Description	Escalating access on a Domain Controller (DC) via a Windows 10 server as an entry point occurs when an attacker compromises a less-secure system (e.g., a Windows 10 server) and uses it as a foothold to gain access to more critical systems in the network, such as the Domain Controller. Once on the DC, attackers can escalate their privileges, obtain domain administrator credentials, and explore the entire domain for sensitive information, including user accounts, credentials, and group policies, potentially leading to full network control.

Images	
Affected Hosts	• 172.22.117.10 (WinDC01) & 172.22.117.20 (Windows 10 machine) • LHOST 172.22.117.100
Remediation	To prevent this, apply the principle of least privilege, restrict access to Domain Controllers, and implement strong network segmentation to isolate critical systems. Regularly patch and harden all systems, including Windows 10 servers, and use multi-factor authentication (MFA) for critical system access. Monitor network traffic and user activities for signs of lateral movement, and employ endpoint detection and response (EDR) tools to detect and block unauthorized escalation attempts.

Vulnerability 10	Findings
Title	**Compromising Admin** dump Active Directory credentials
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	The **Kiwi** tool was used to dump Active Directory credentials, providing attackers with the ability to escalate privileges or move laterally within the network. Compromising an admin account on a Domain Controller (DC) is a critical security breach where an attacker gains access to an account with Domain Administrator privileges. This level of access allows the attacker to fully control the domain, including all user accounts, group policies, authentication systems, and sensitive data. The attacker can also modify security settings, disable auditing, and create backdoors for future access,

	effectively compromising the entire network.
Images	
Affected Hosts	172.22.117.10 (WinDC01)
Remediation	Implement monitoring of credential dumping tools, apply least privilege access, and use **Windows Defender** for credential theft detection. To prevent the compromise of admin accounts on a DC, implement strict access controls with the least privilege principle, use multi-factor authentication (MFA) for all administrator accounts, and regularly audit privileged account usage. Harden the Domain Controller by limiting RDP and direct access, using secure administrative practices like "Privileged Access Management," and regularly updating and patching the system to prevent exploitation of known vulnerabilities.