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	White Hills Information Security (WHIS)
Contact Name	Andrew Arnold
Contact Title	Junior Pentester

Document History

Version	Date	Author(s)	Comments
001	2/9/2023	Andrew Arnold	

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.

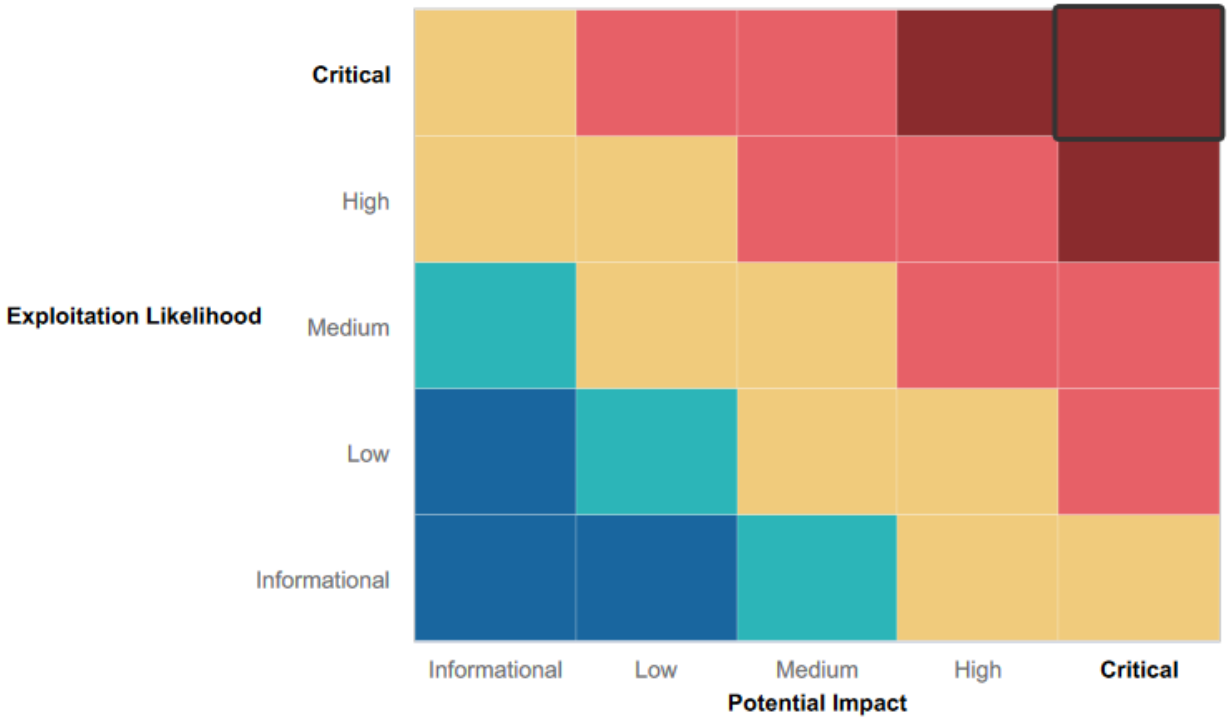
Executive Summary of Findings

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.

- Ensure network availability through a mitigation strategy that was put in place against denial of service (DDoS) attacks
- Highered penetration testing to identify vulnerabilities to mitigate
- Mapping network architecture

Summary of Weaknesses

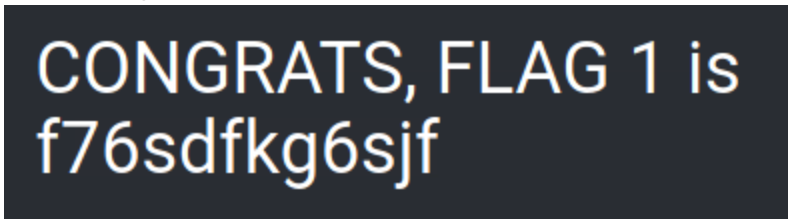
We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

- Web application is vulnerable to cross-site scripting attacks
- Web application is vulnerable to SQL injection attacks
- Log-in credentials are being stored in HTML source code
- Apache web server is outdated and vulnerable
- SLMail server is exploitable which allows shell access
- User access to password hashes allows for password cracking and privilege escalation
- The physical address of Rekall's server is publicly available which could lead to a physical penetration
- Sensitive data exposed to the public

Executive Summary

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

Welcome.php: On the welcome.php page a cross-site scripting (XSS) vulnerability was found. By typing an XSS script into the open text field, the WHIS team was able to inject client-side scripts into the web page.



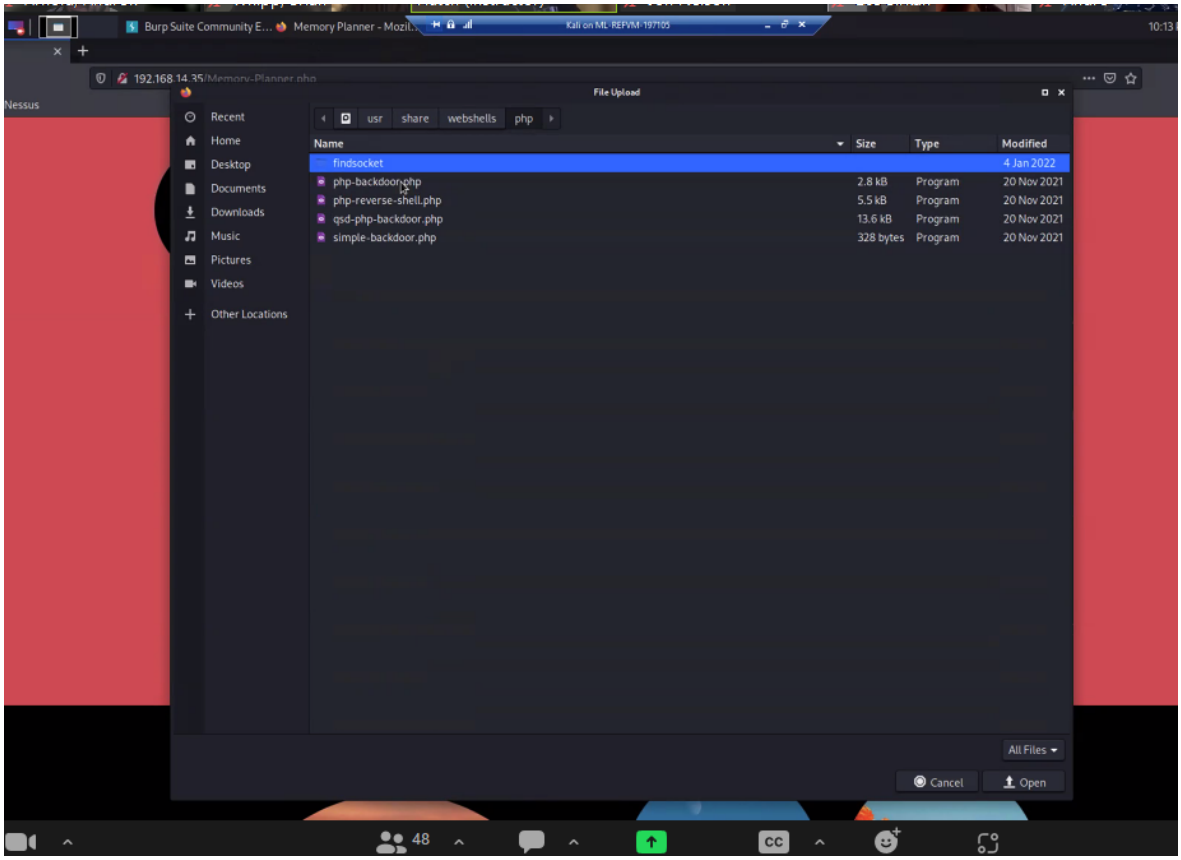
Memory-Planner.php: On the memory-planner.php page a cross-site scripting (XSS) vulnerability was found. By typing an advanced XSS script into the open text field, the WHIS team able to inject client-side scripts into the web page.

comments.php: On the comments.php page a cross-site scripting (XSS) stored vulnerability was found. By typing an XSS script into the open text field the WHIS team was able to store a possibly malicious script on the application server.

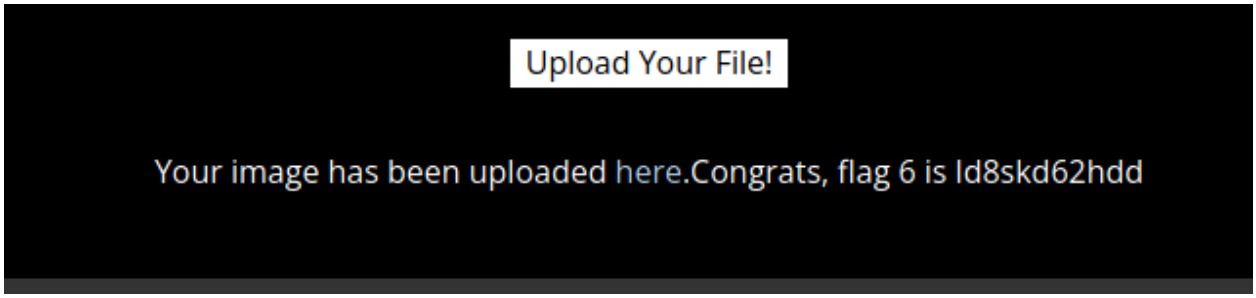
About-Rekall.php: This can be found by looking at the HTTP response headers in either BURP or by using a cURL request.

```
┌──(root💀kali)-[~]
└─# curl -v 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: Wed, 08 Feb 2023 02:59:14 GMT
< Server: Apache/2.4.7 (Ubuntu)
< X-Powered-By: Flag 4 nckd97dk6sh2
< Set-Cookie: PHPSESSID=le2hpvbbeggjp50pchuojtj6g4; 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
<
```

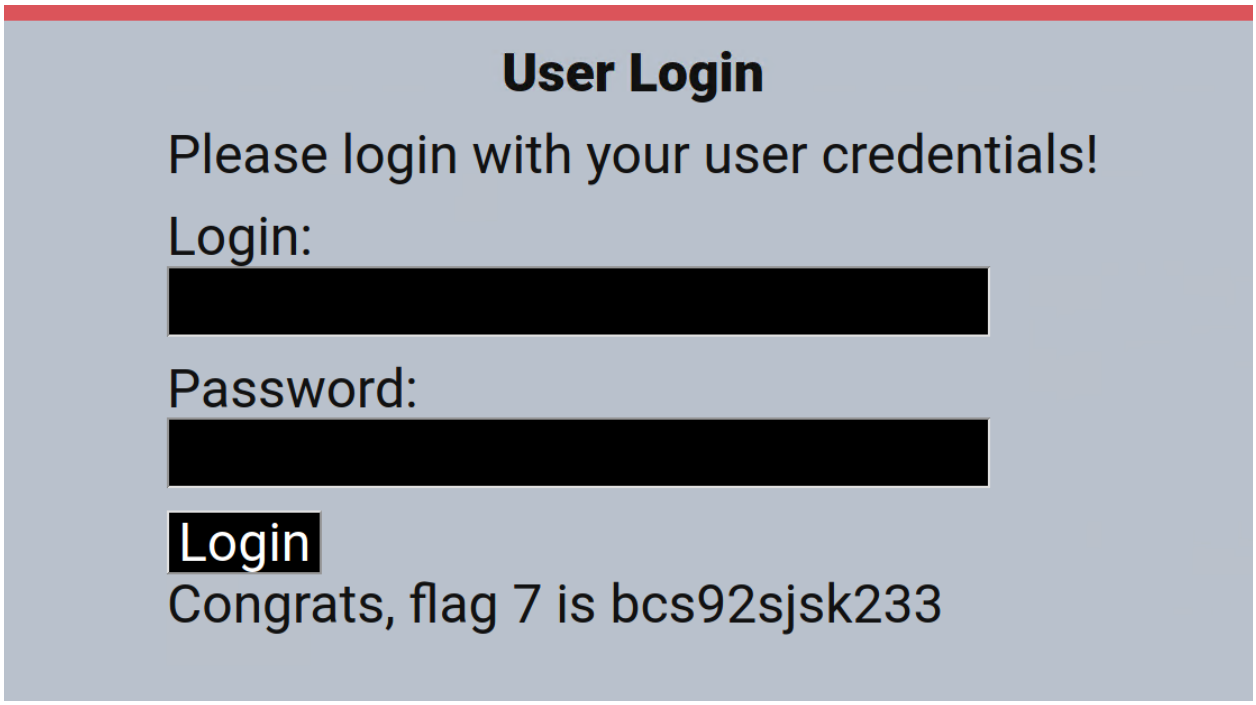
Memory-Planner.php (2): A local file inclusion vulnerability was found where a user can upload a possibly malicious script within a PHP file.



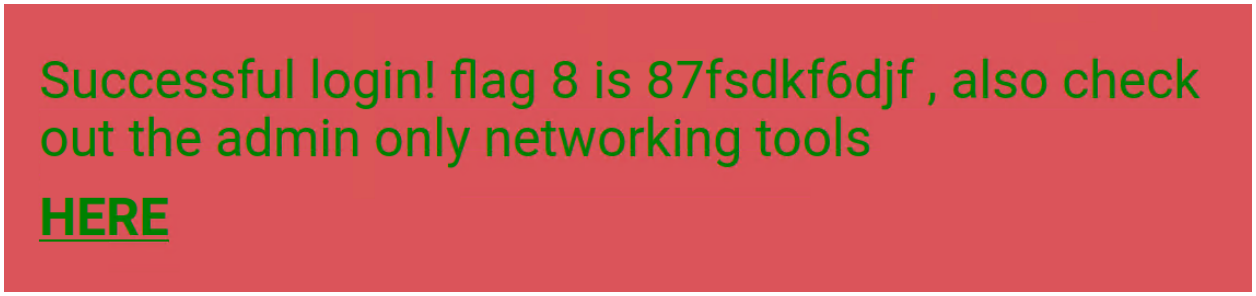
Memory-Planner.pgp (3): A local file inclusion vulnerability was found where a user can upload a possibly malicious script within a PHP file. Despite the input checking for the presence of .jpg, a user can name their malicious script script.jpg.php to bypass this validation check.



Login.php: An SQL injection vulnerability was found on the login.php web page. A user can bypass the login page by entering an SQL injection script to trick the computer into thinking they are a valid user.



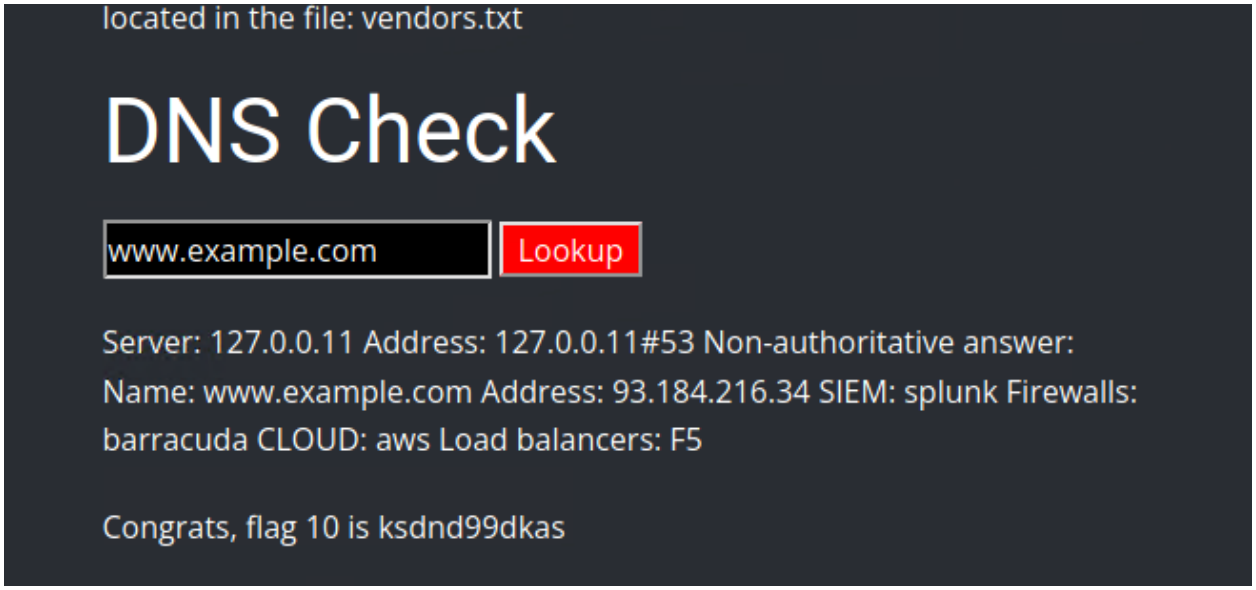
Login.php (2): The login.php web page has a sensitive data exposure vulnerability. The username, dougquaid, and password, kuato, can be enumerated by browsing the HTML text.



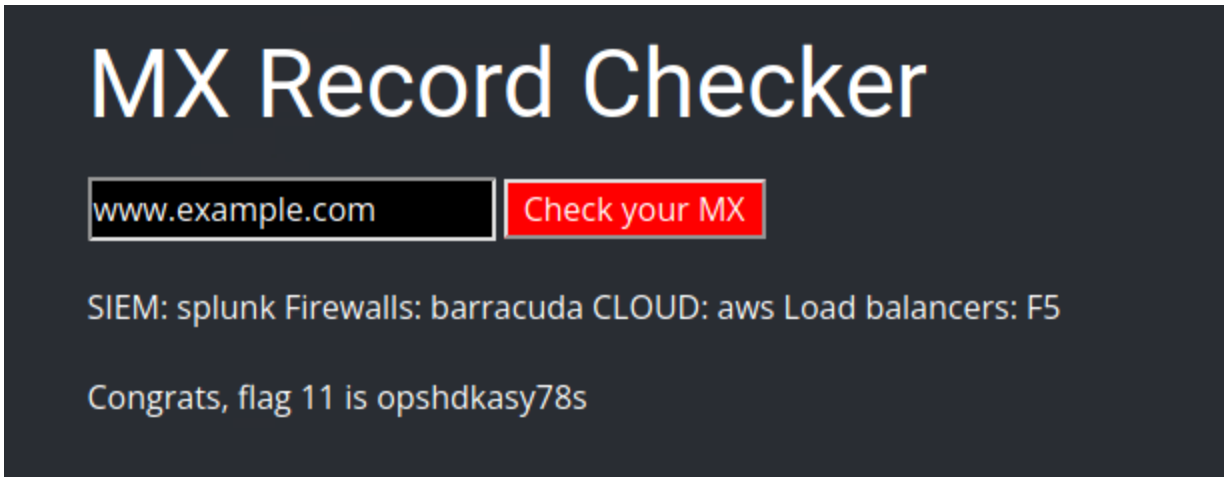
Robots.txt: A sensitive data exposure vulnerability was found on the robots.txt web page. This is a very common web page that is accessible by default.

```
User-agent: GoodBot
Disallow:

User-agent: BadBot
Disallow: /

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

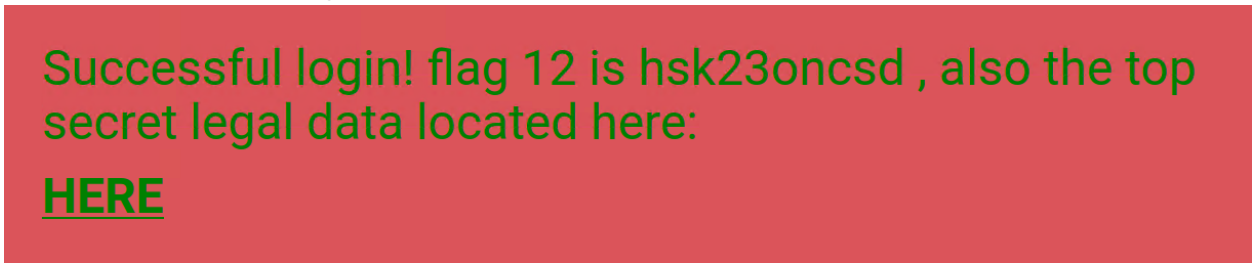
Networking.php: Within the first field on the networking.php page, a command injection vulnerability was found. The WHIS team was able to access and read files on the server.



Networking.php (2): Within the second field on the networking.php page, a command injection vulnerability was found. While this field did not allow the simple command injection that the first field did, it did allow a more advanced form of command injection. This allowed the WHIS team to navigate and read files on the server.

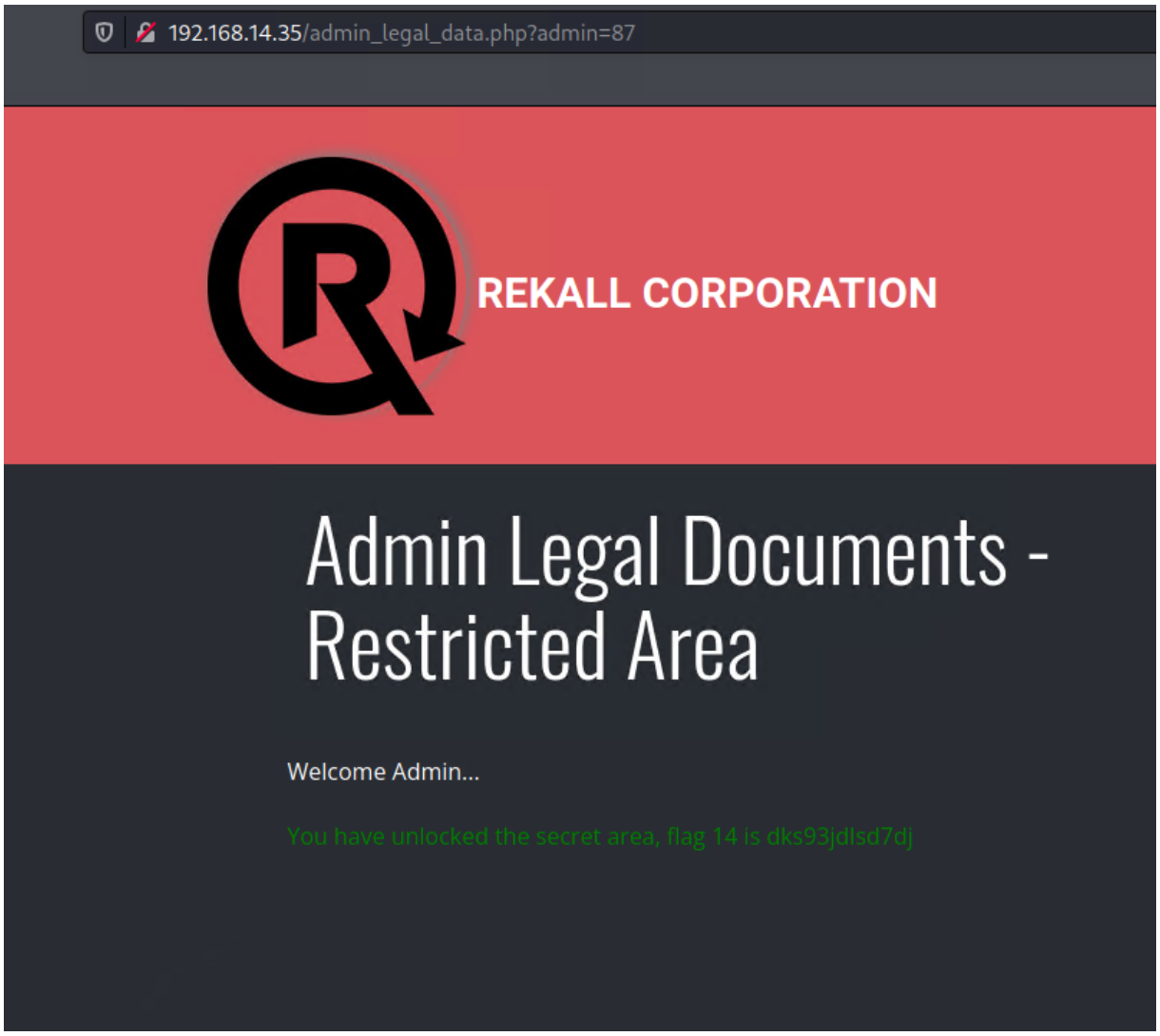


Login.php: By using the command injection vulnerability mentioned above, our team was able to enumerate a user, melina, from the servers /etc/passwd file. The WHIS team was then able to brute force her password and log in as username:melina password:melina.

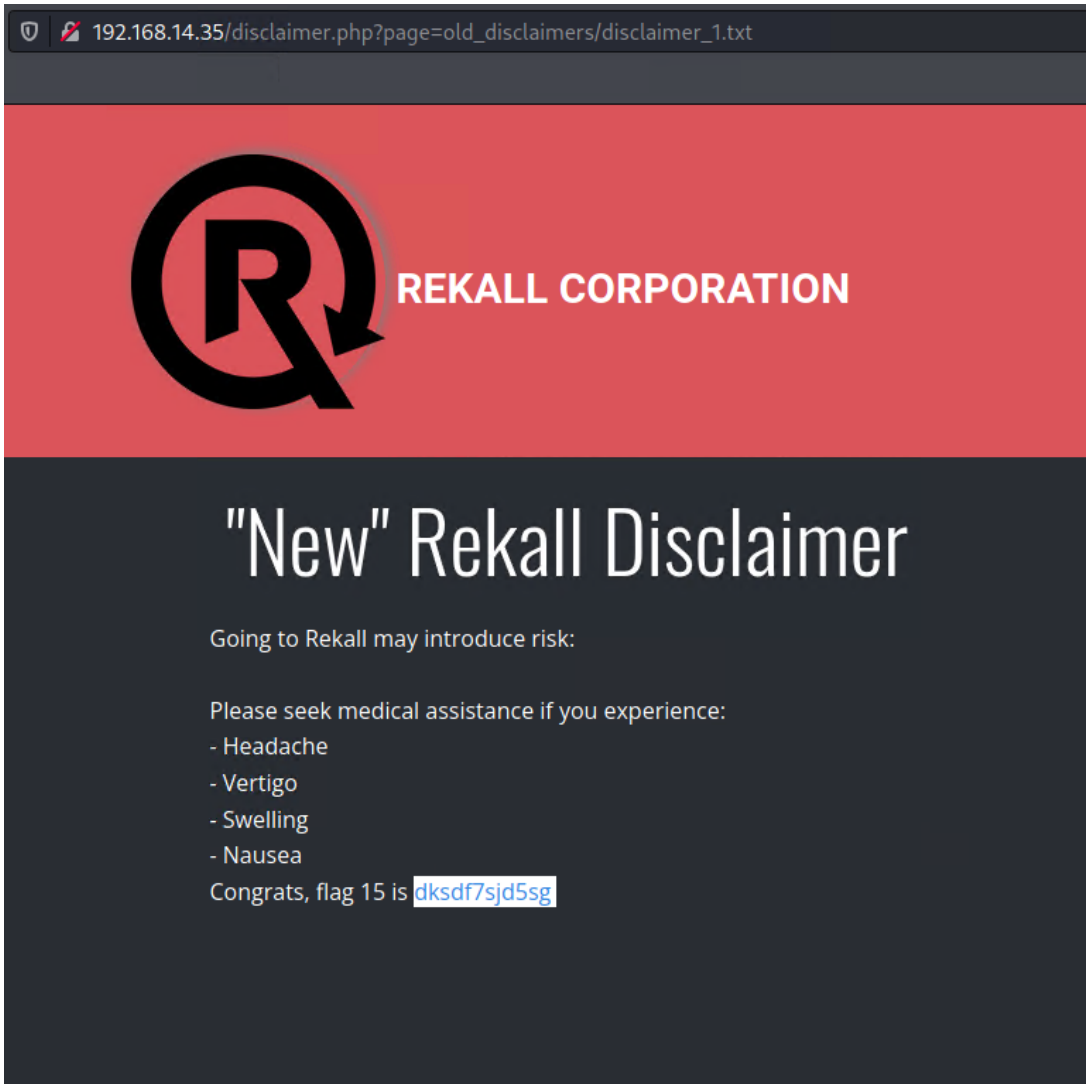


Souvenirs.php: Using the robots.txt web page, the WHIS team was able to enumerate the hidden webpage souvenirs.php. In the webpage address bar, a user is able to change the message value to a PHP injection exploit to access the web server files.

Admin_Legal_Data.php: On the admin_legal_data.php page a session management exploit was found. By brute forcing a login in to the user melina, the WHIS team was given access to the admin_legal_data.php webpage. Upon logging into this webpage, our team was able to use BURP to iterate through different session IDs until the secret value of 87 was identified.



Disclaimer.php: A directory traversal vulnerability was found on the disclaimer.php webpage. By using command injection on the networking.php webpage, WHIS was able to enumerate the old_disclaimers directory which allowed us to change the URL on disclaimer.php to include old_disclaimers/disclaimer_1.txt.



Open Source Exposed Data: By using https://centralops.net/co/DomainDossier.aspx our team was able to obtain access to exposed data about the totalrekall.xyz web page.

```
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
```

Address Lookup/Ping: By using an address lookup website, our team was able to determine the IP address the totalrekall.xyz

Address lookup

canonical name totalrekall.xyz.

aliases

addresses 34.102.136.180

Crt.sh: Our team was able to use crt.sh to identify all certificate authorities for totalrekall.xyz. As well, totalrekall.xyz had an open source exposed data vulnerability which allowed our team to identify other common names from the totalrekall web page.

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

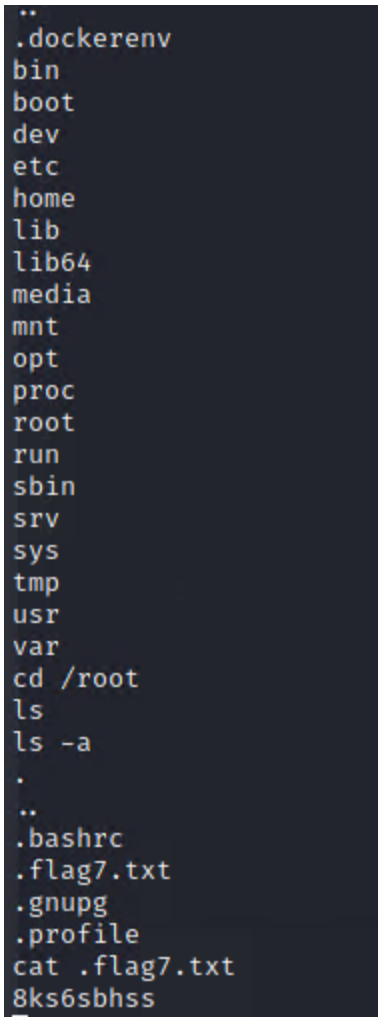
Nmap Scan: Our team ran an nmap scan of the network ip range found in the Address Lookup and found 5 hosts.

```
┌──(root💀kali)-[~/Documents]
└─# nmap -A 192.168.13.13
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-09 19:56 EST
Nmap scan report for 192.168.13.13
Host is up (0.000068s 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

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.06 seconds
```

Aggressive Nmap Scan: After running an aggressive Nmap scan, our team found that the server 192.168.13.13 was running Drupal.

```
┌──(root💀kali)-[~/Documents]
└─# nmap -A 192.168.13.13
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-09 19:56 EST
Nmap scan report for 192.168.13.13
Host is up (0.000068s 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

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.06 seconds
```

Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617): After running a Zenmap scan of the 192.168.13.10 server, our team found that the server was running an Apache Tomcat service. Our team was then able to use Metasploit to exploit the Tomcat service. After successfully connecting to the Meterpreter session, our team was able to enter a shell on the server and access files.



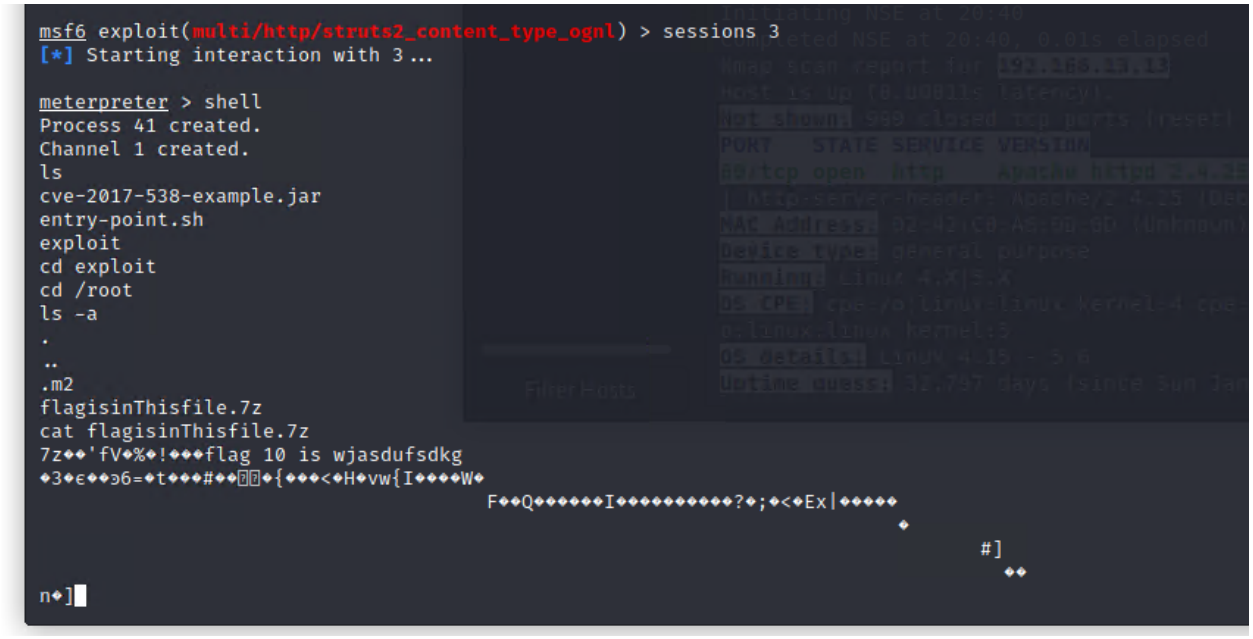
Shellshock: After running a Zenmap scan of the 192.168.13.11 server, our team found that the server was running an exploitable bug known as Shellshock. Our team was then able to use Metasploit to exploit the Shellshock bug. After successfully connecting to the Meterpreter session, our team was able to enter a shell, as root, on the server and access the sudoers file.

```
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
```

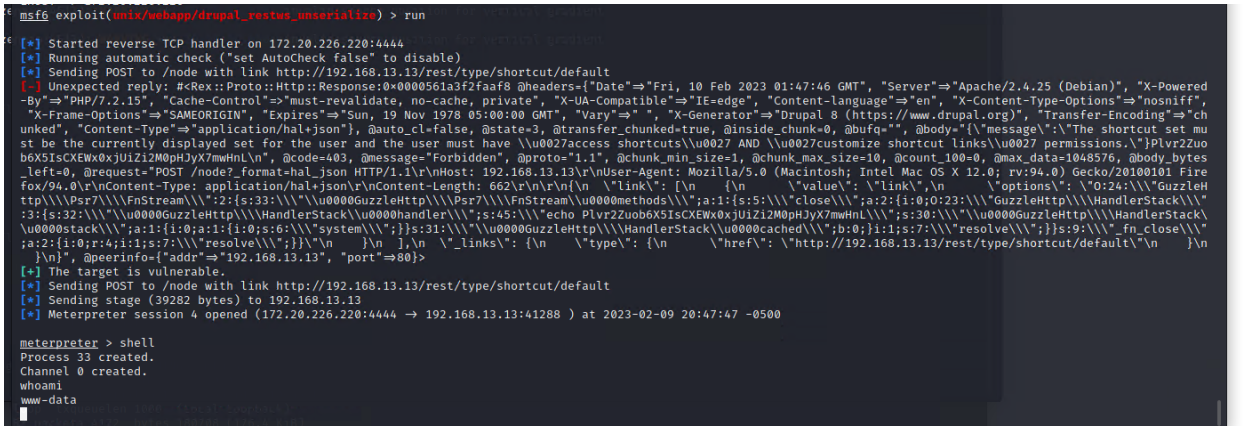
Shellshock vulnerability 2: After running a Zenmap scan of the 192.168.13.11 server, our team found that the server was running an exploitable bug known as Shellshock. Our team was then able to use Metasploit to exploit the Shellshock bug. After successfully connecting to the Meterpreter session, our team was able to enter a shell, as root, on the server and access the /etc/passwd file.

```
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:
```

Struts - CVE-2017-5638: After running a Nessus scan of the 192.168.13.12 server, our team found that the server was running an exploitable Struts vulnerability. Our team was then able to use Metasploit to exploit the Struts vulnerability. After successfully connecting to the Meterpreter session, our team was able to use Meterpreter to download a zipped file from the server. Once the zipped file was successfully downloaded, we were able to unzip the file on our host machine.

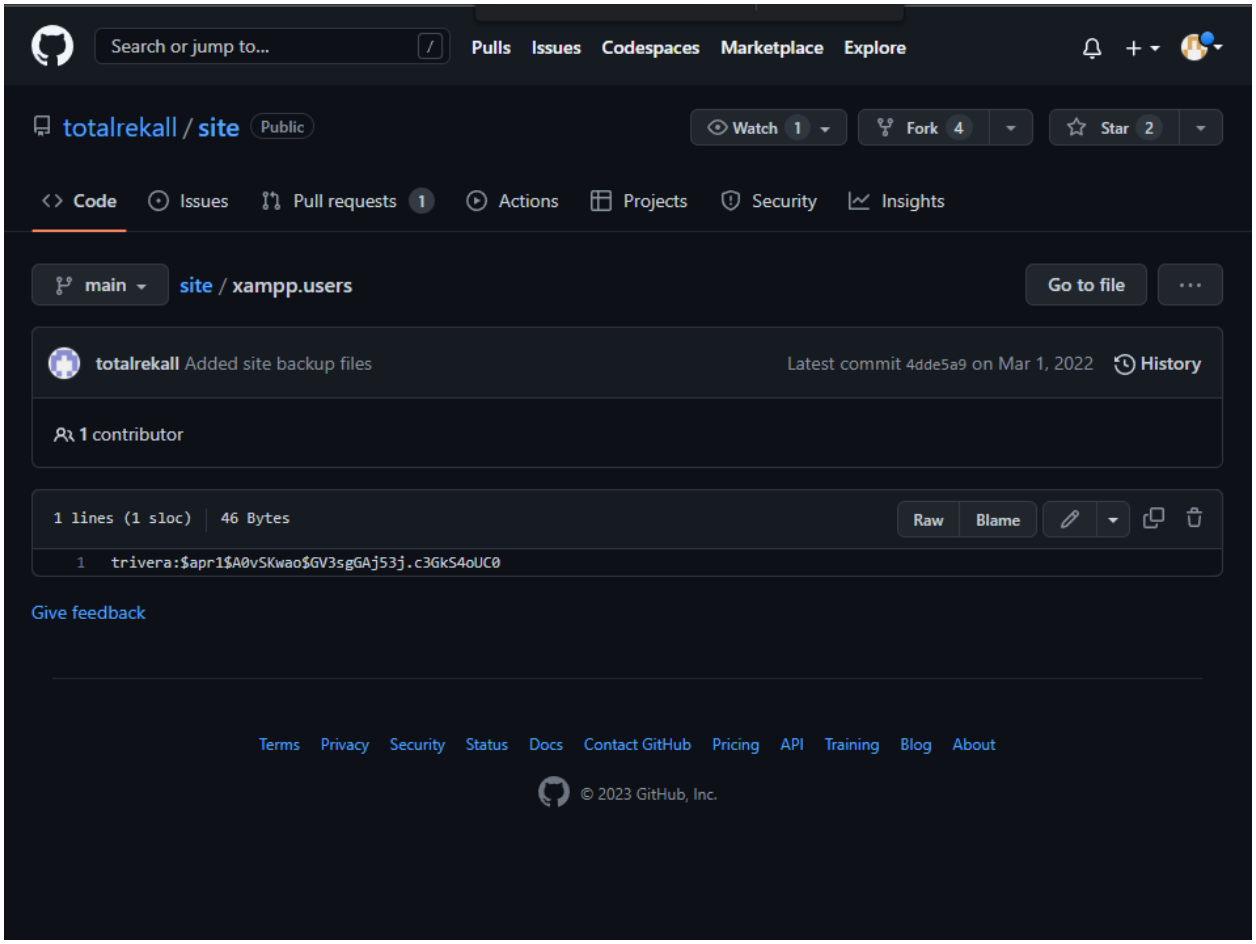


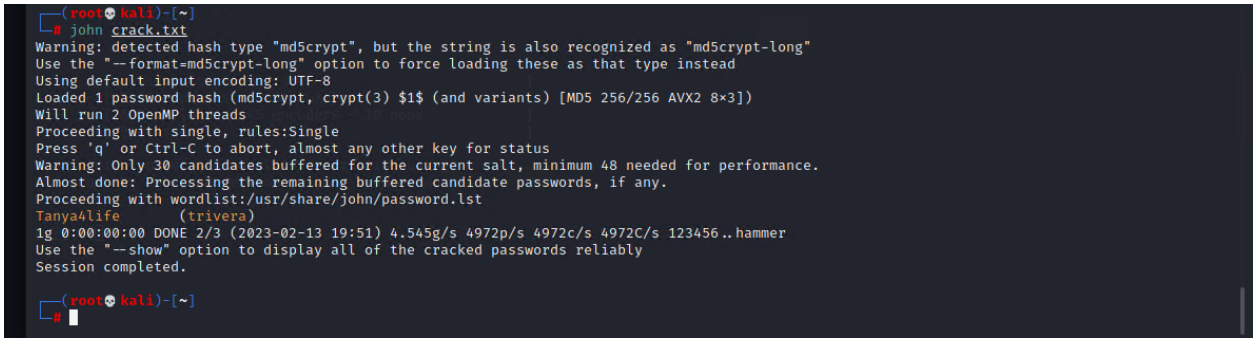
Drupal - CVE-2019-6340: After running a Nessus scan of the 192.168.13.13 server, our team found that the server was running an exploitable Drupal vulnerability. Our team was then able to use Metasploit to exploit the Drupal vulnerability. After successfully connecting to the Meterpreter session, our team used getuid to identify which user we were logged in as.



CVE-2019-14287: Returning back to the whois data from earlier, our team identified the user Alice. Our team was also able to identify that the user Alice has SSH access to the network. Our team was able to brute force the user Alice's password, Alice, and successfully SSH into the 192.168.13.14 server. After connecting to the server, our team was able to exploit the known CVE-2019-14287 vulnerability to escalate privilege to root. Once in root privilege, our team was able to access all files on the server.

Tanya4life: Our team was able to search GitHub and find the totalrekall GitHub page. After searching the repository, our team found the xampp.users page, which contained the credentials for a user trivera and their password hash. Our team was able to use John the Ripper to crack the user's password hash.





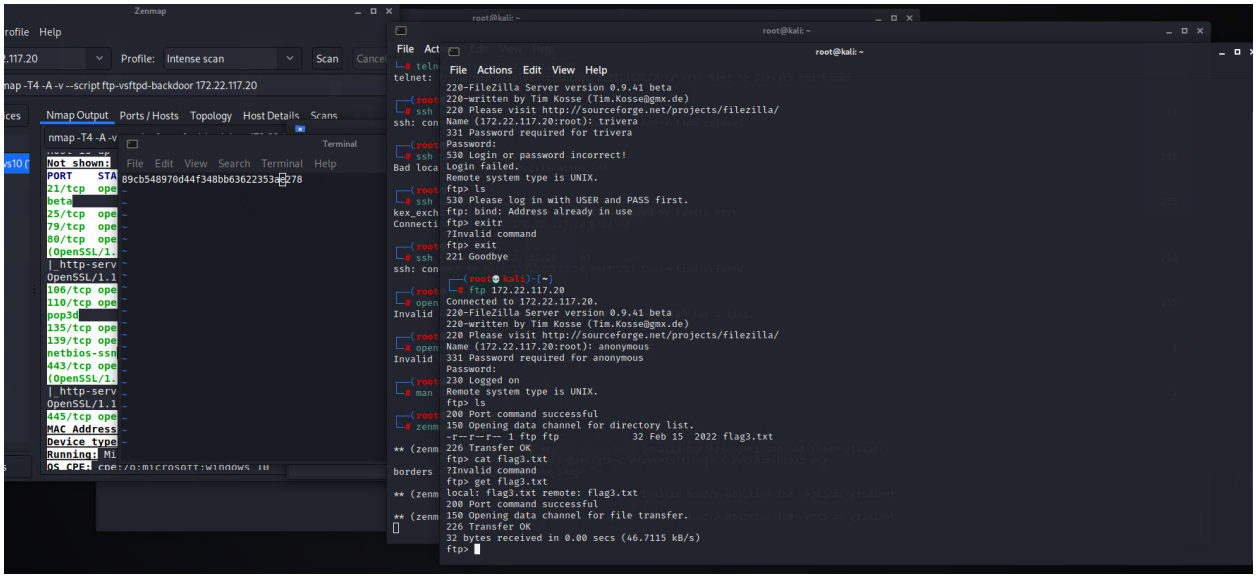
Port 80 http open: After running a port scan on the Win10 server, our team identified an open http port. Upon navigating the internet to the 172.22.117.20 web address, we were able to enter the trivera username with the tanya4life password to gain access to the web server.

Index of /

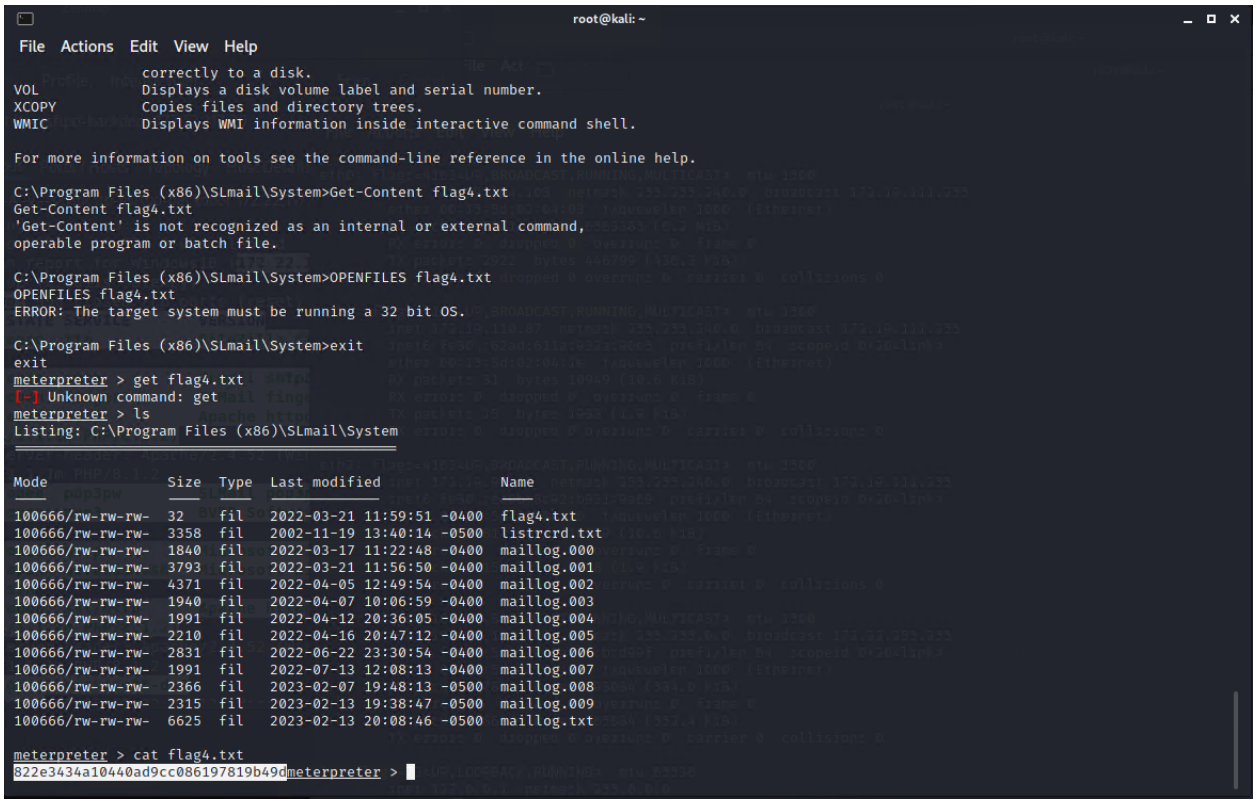
Name	Last modified	Size	Description
flag2.txt	2022-02-15 13:53	34	

Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2 Server at 172.22.117.20 Port 80

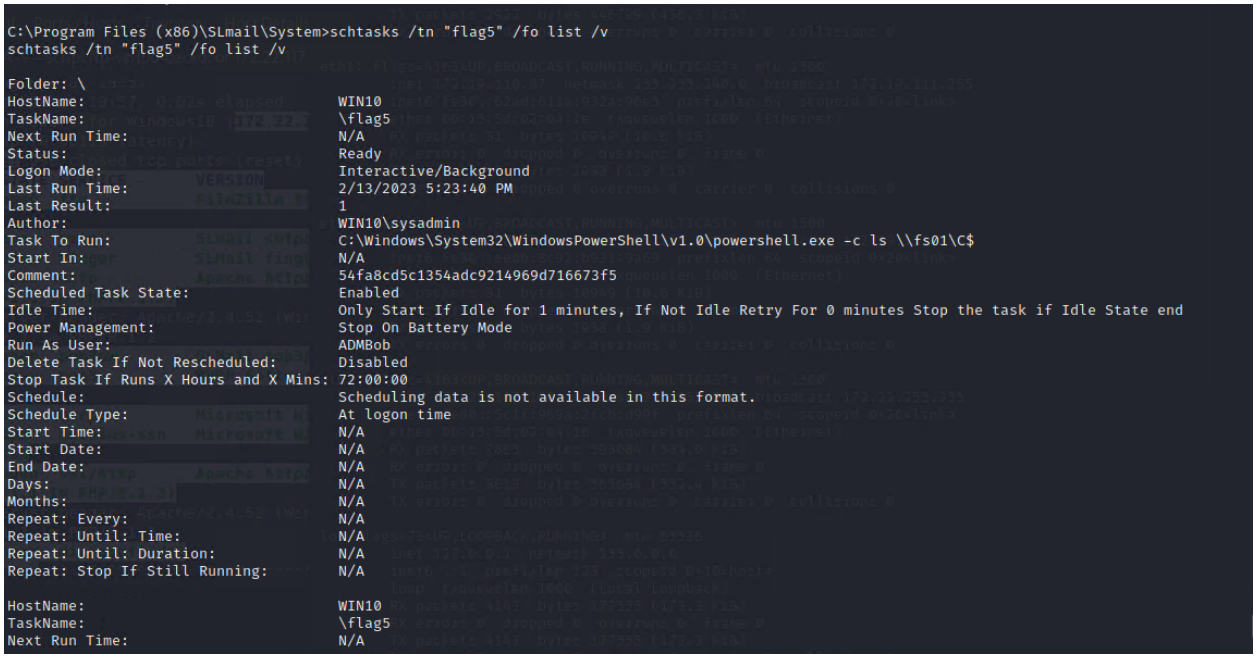
FTP anonymous: Going back to the Nmap scan of the 172.22.117.20 server, our team noted that the ftp anonymous service was turned on. Once logged into ftp anonymous, our team was able to download data from the server.



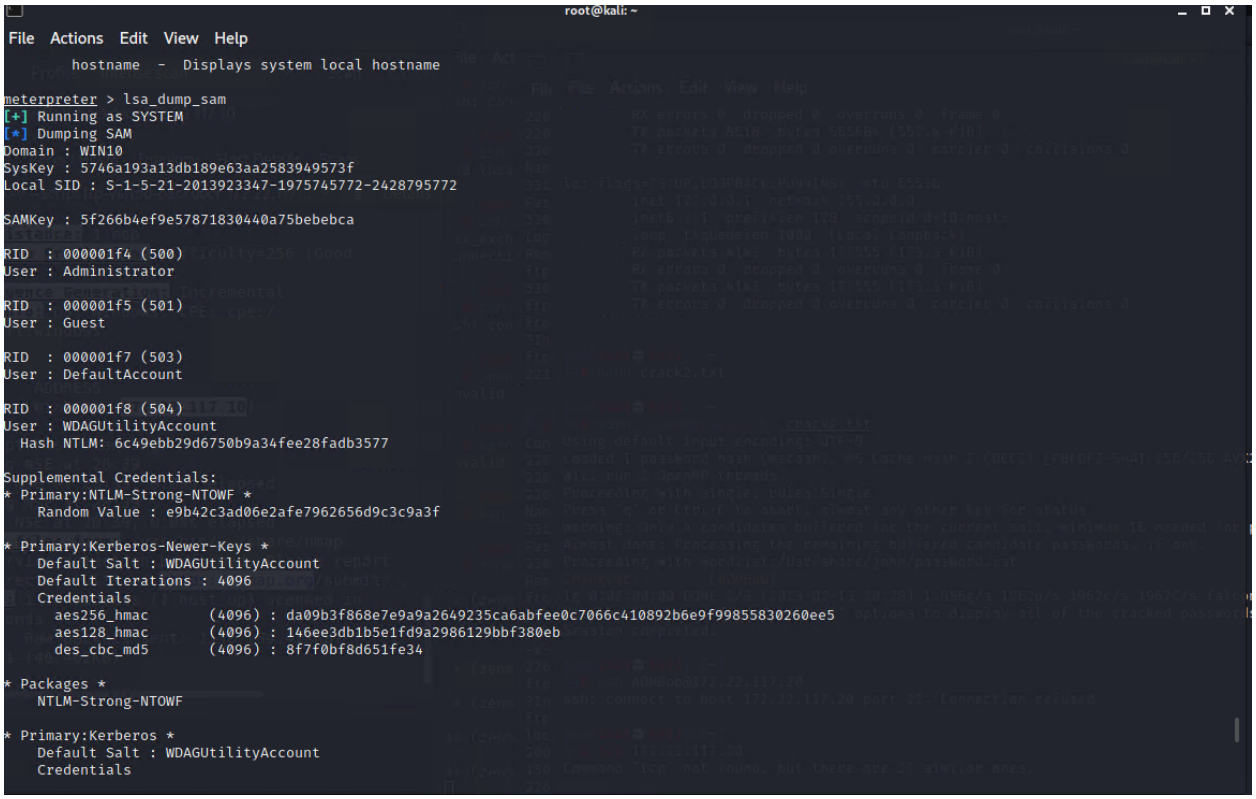
SLMail: Returning back to the Nmap scan of the 172.22.117.20 server, our team was able to identify that the SLMail service is running on SMTP port 25 AND on POP3 port 110. Using Metasploit, our team was able to exploit this service and gain access to a Meterpreter session. Once our team had access, we were able to view files on the server.



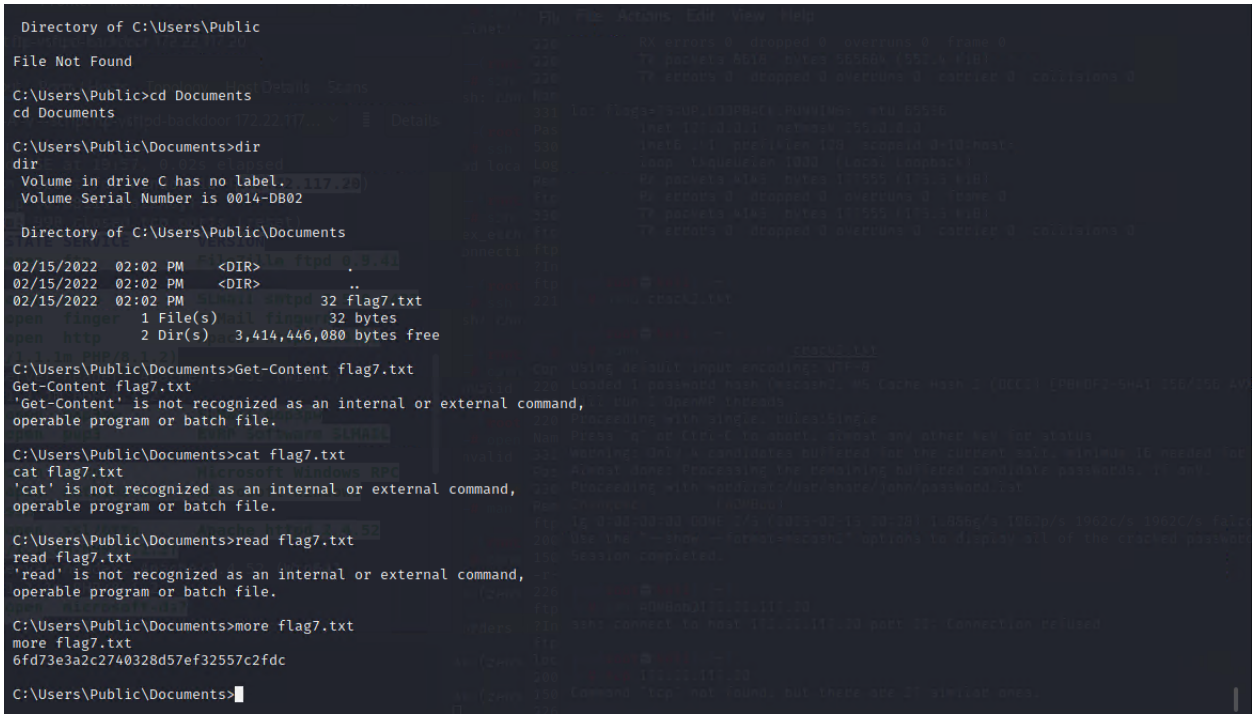
Schtasks: With our current access to the server 172.22.117.20, our team was able to query the scheduled tasks on that server to identify what happens on the server and when it happens. After finding a task of interest, our team was able to read that task and it's details to understand what the task does.



LSA and Kiwi: After compromising the 172.22.117.20 server using the SLMail vulnerability using metasploit, our team was given a meterpreter shell under system access. Using meterpreter, our team could then load the kiwi. Using the lsa_dump_sam command within kiwi, our team was able to access a user and their password hash. Once our team had the user's password hash, we used John the Ripper to crack the password.

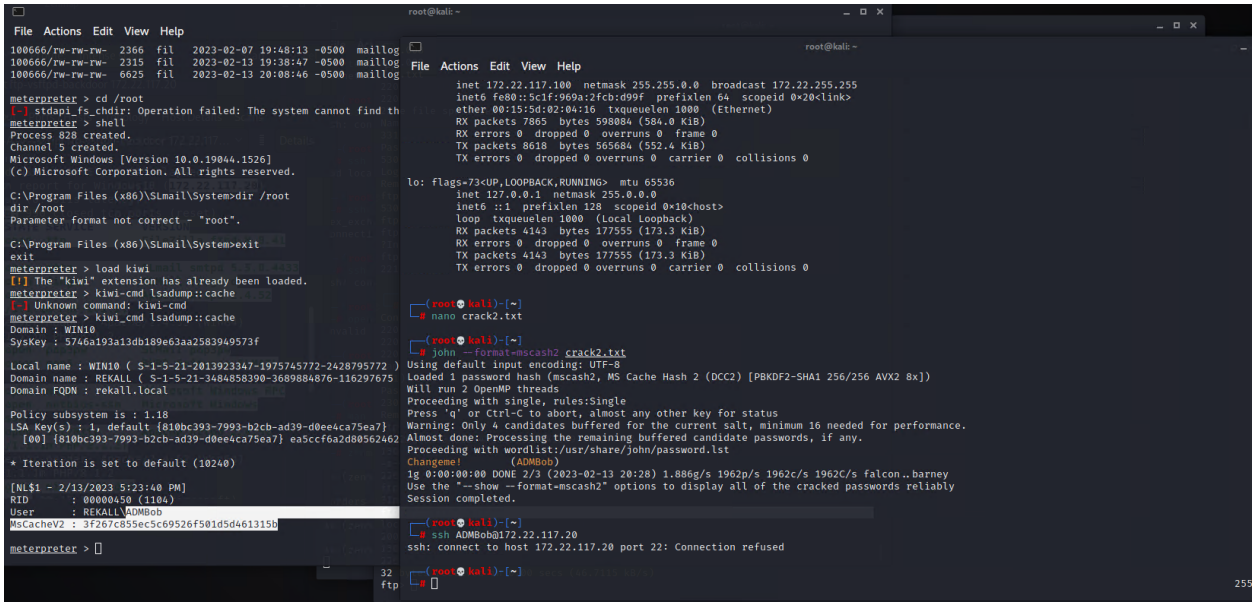


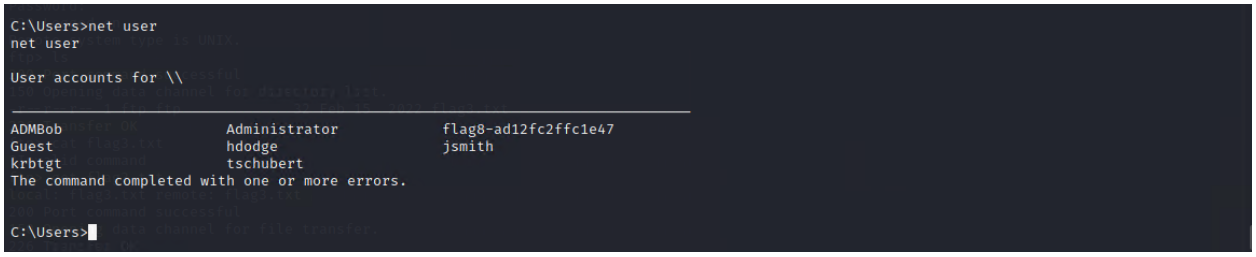
Search: After compromising the 172.22.117.20 server using the SLMail vulnerability using metasploit, our team was given a meterpreter shell under system access. Using search, our team was able to locate sensitive data. Once located, our team could use the more command in meterpreter to read those files.



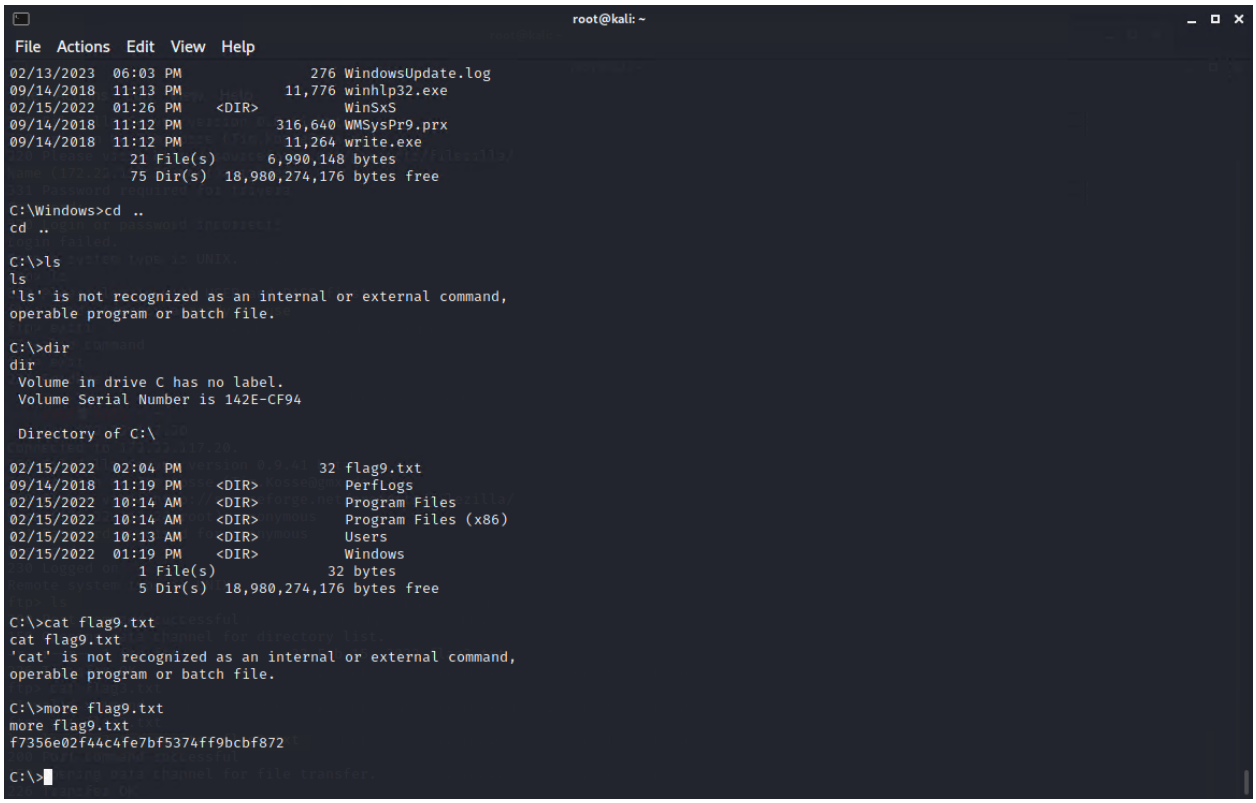
ADMBob: After compromising the 172.22.117.20 server using the SLMail vulnerability using metasploit, our team was given a meterpreter shell under system access. Using kiwi, our team was able to run an lsa cache dump to gain access to the user ADMBob and his password hash. Using John the Ripper, our team was able to crack ADMBob's password. Using these credentials, and the Psexec Metasploit module, our team was able to gain access to the Server 2019 machine. Once our

team had access, we were able to enter a shell within meterpreter and use the net user command to list all of the server users.





File Data: Once our team had access to the Server 2019 machine using the Psexec Metasploit module, we were able to access the files on the server. We were then able to read files using the cat/more command within meterpreter.



Kiwi DCSync: After identifying the Server2019 users, our team noticed an Administrator account. Using the kiwi DCSync command, our team was able to access the Administrator password hash.

```
meterpreter > dcsync_ntlm Administrator
[+] Account  : Administrator
[+] NTLM Hash : 4f0cfd309a1965906fd2ec39dd23d582
[+] LM Hash  : 0e9b6c3297033f52b59d01ba2328be55
[+] SID      : S-1-5-21-3484858390-3689884876-116297675-500
[+] RID      : 500

meterpreter >
```

```
┌──(root💀kali)-[~]
└─# impacket-secretsdump -just-dc-ntlm rekall.local/ADMBob:'Changeme!'@172.22.117.10

^[[AImpacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:4f0cfd309a1965906fd2ec39dd23d582:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:fa5875a009bc010f4a210826e8dabfaa:::
rekall.local\ADMBob:1104:aad3b435b51404eeaad3b435b51404ee:07783b44a8b3d69e8e7d55f9272df3f5:::
rekall.local\jsmith:1105:aad3b435b51404eeaad3b435b51404ee:7978dc8a66d8e480d9a86041f8409560:::
rekall.local\tschubert:1106:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
rekall.local\hdodge:1108:aad3b435b51404eeaad3b435b51404ee:fc9d7c3a3a1e86f1bcc35cd887cb74d5:::
rekall.local\flag8-ad12fc2ffc1e47:1109:aad3b435b51404eeaad3b435b51404ee:10e6f496b8ba9704de223de855ec6849:::
WINDC01$:1000:aad3b435b51404eeaad3b435b51404ee:d25cd9b8aa8e417aa4b8e9487c765aa4:::
^[[BWIN10$:1103:aad3b435b51404eeaad3b435b51404ee:ae7b5f635513e80e9e5420a0df103bbe:::
[*] Cleaning up...
```

Summary Vulnerability Overview

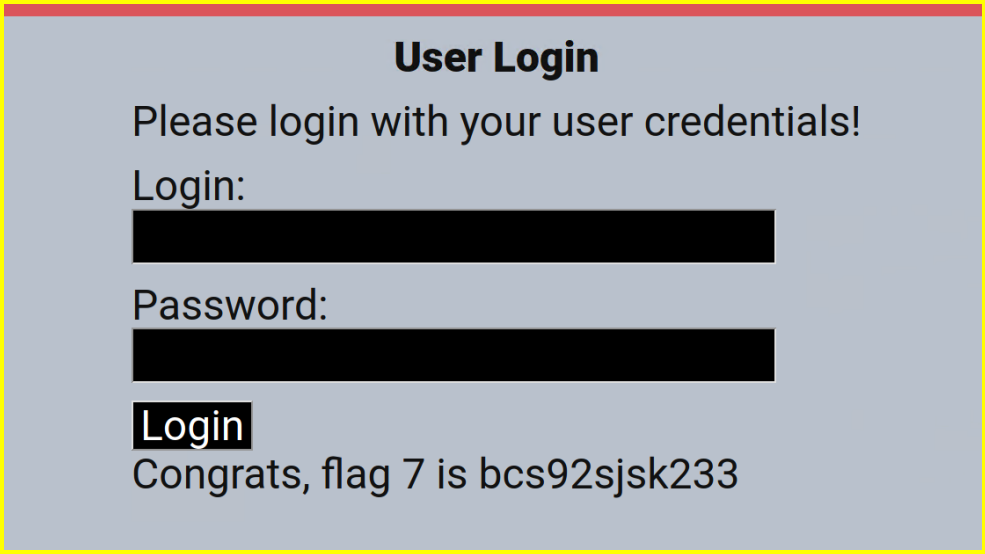
Vulnerability	Severity
SQL Injection	Critical
Command Injection	Critical
Shellshock	Critical
Struts - CVE-2017-5638	Critical
Drupal - CVE-2019-6340	Critical
Local File Inclusion	Critical
Sensitive Data Exposure	Critical
User Credential Exposure	Critical
Linux Privilege Escalation	Critical
SLMail	Critical
PHP Injection	Critical
Directory Traversal	Critical
GitHub Repo	Critical
Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617)	High
CVE-2019-14287	High
XSS Reflected	Medium
XSS Stored	Medium
Brute Force	Medium

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

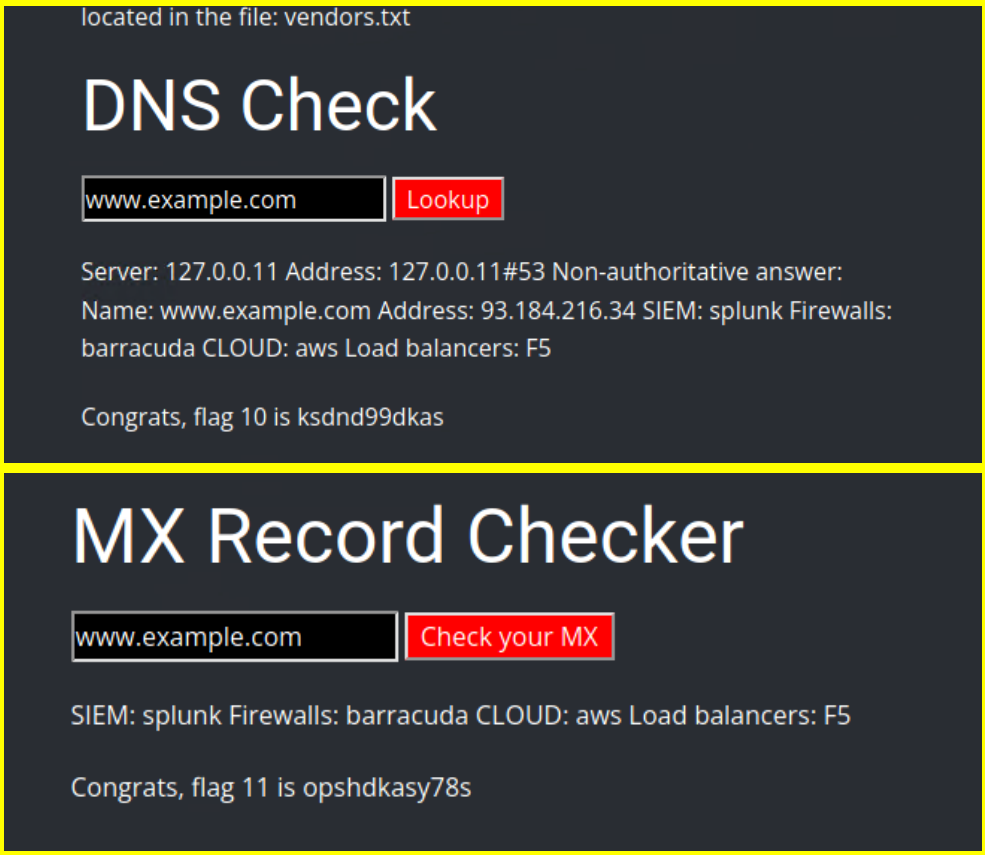
Scan Type	Total
Hosts	8
Ports	5

Exploitation Risk	Total
Critical	12
High	2
Medium	2
Low	0

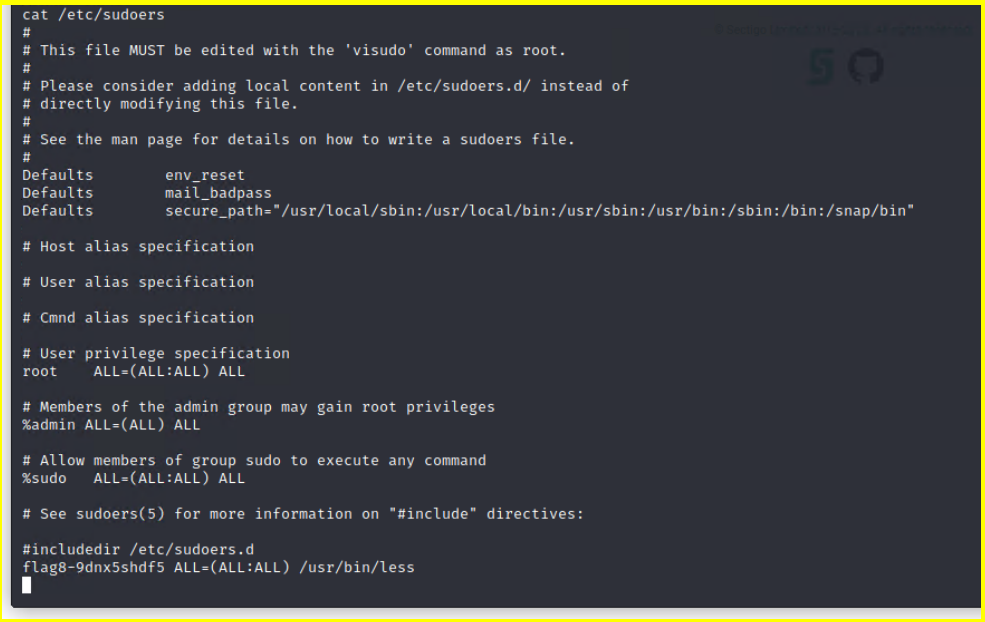
Vulnerability Findings

Vulnerability 1	Findings
Title	SQL Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	While accessing the Login.php page, a payload (admin or “1”=”1”) was entered into the toolbar intended for username and password which successfully resulted in an exploit.
Images	
Affected Hosts	192.168.14.35
Remediation	Do not allow the web app to accept direct input and/or implement character escaping.

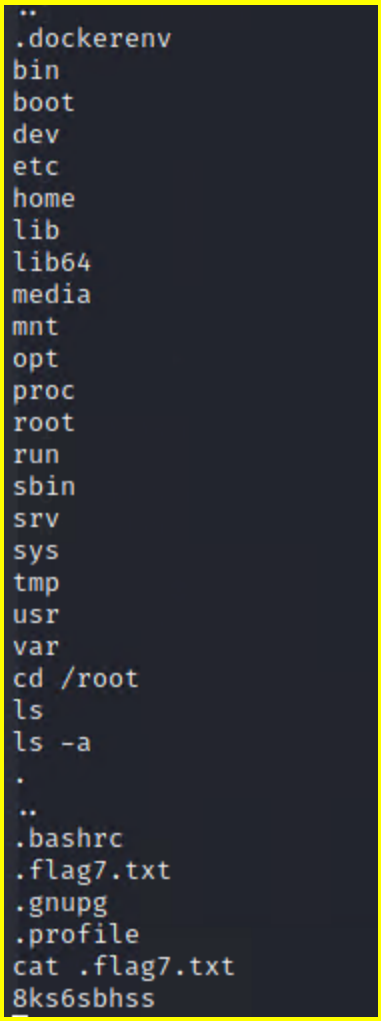
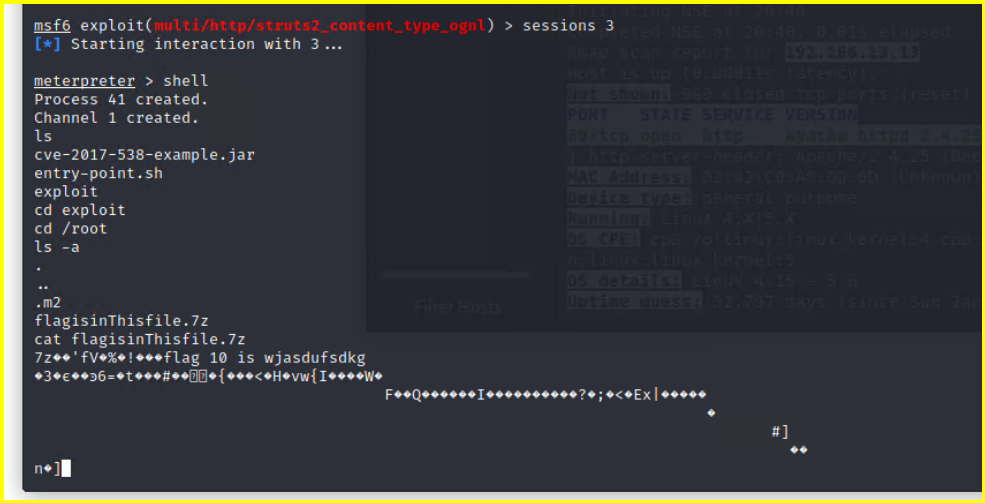
Vulnerability 2	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Server navigation allows from /networking.php using the www.welcometorecall.com && cat vendors.txt, www.welcometorecall.com ; cat vendors.txt, or www.welcometorecall.com | cat vendors.txt exploit.

Images	
Affected Hosts	192.168.14.35
Remediation	Implement input validation for unintended access

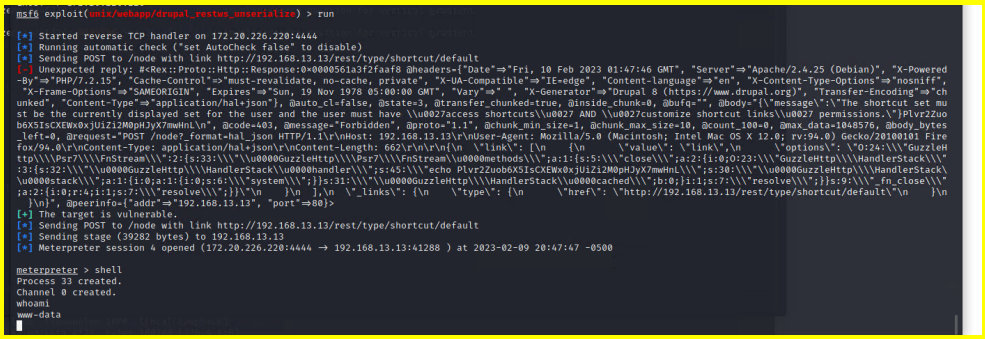
Vulnerability 3	Findings
Title	Shellshock
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Used MSFconsole and searched for exploits that have Shellshock. Ran exploit/multi/http/apache_mod_cgi_bash_env_exec and set the following options: target URI(The vulnerable webpage): /cgi-bin/shockme.cgi and RHOST: 192.168.13.11. Once the meterpreter session connects, go into a shell and run cat /etc/sudoers.

Images	
Affected Hosts	192.168.13.11
Remediation	Edit sudoers file to limit access for all sudo accounts.

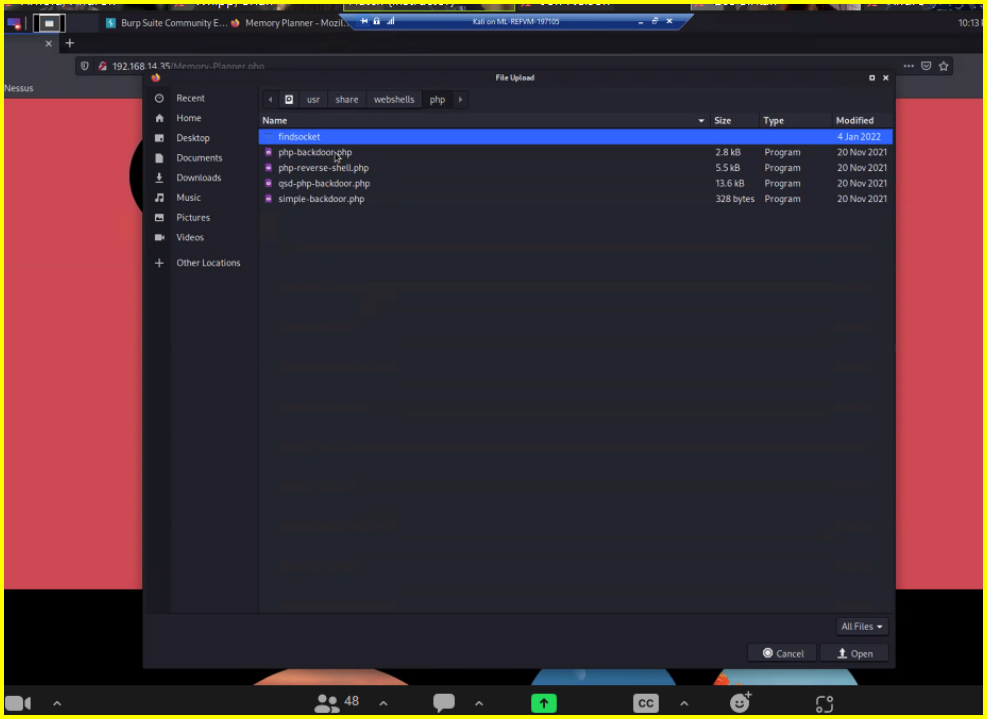
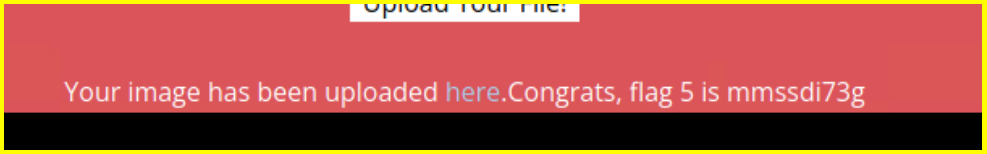
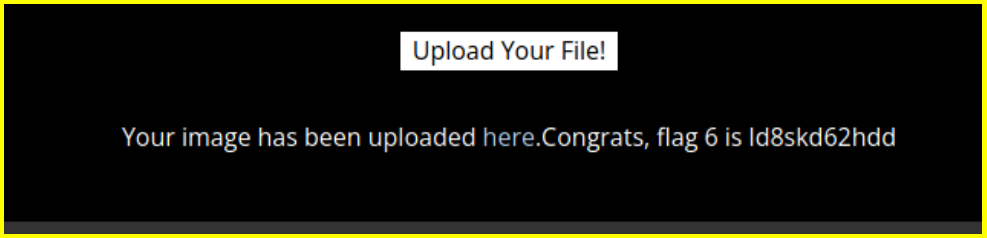
Vulnerability 4	Findings
Title	Struts - CVE-2017-5638
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Using the Nessus scan, WHIS determined that the host server is vulnerable to Struts. Running MSFconsole, WHIS used the Struts exploit multi/http/struts2_content_type_ognl. Set the RHOSTS to 192.168.13.12. Using Meterpreter, WHIS was able to download files from the server.

Images	 
Affected Hosts	192.168.13.12
Remediation	Patch and update

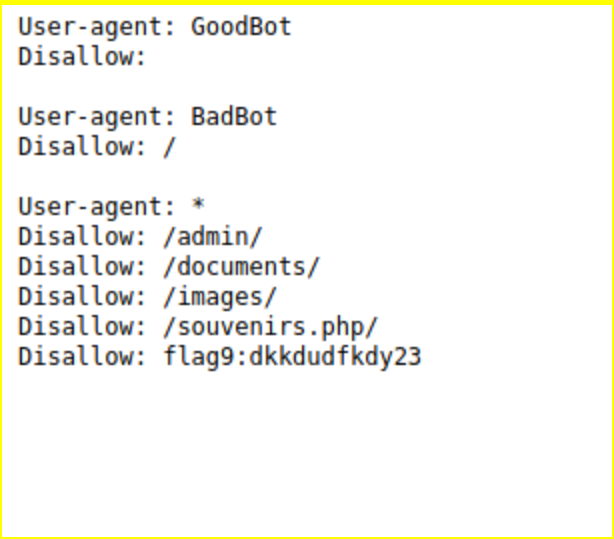
Vulnerability 5	Findings
Title	Drupal - CVE-2019-6340

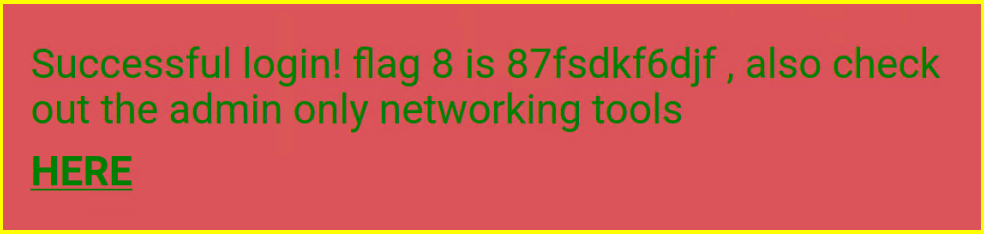
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Within MSFconsole, WHIS searched for Drupal exploits. WHIS was able to use Meterpreter exploit unix/webapp/drupal_restws_unserialize and set RHOSTS to 192.168.13.13. After gaining a Meterpreter shell, WHIS ran getuid on the host server.
Images	
Affected Hosts	192.168.13.13
Remediation	Patch and Update. Disable RESTful Web Services module if not in use.

Vulnerability 6	Findings
Title	Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	WHIS was able to upload, and execute, a .php and a .jpg file using the tool bar located on the memory-planner.php page.

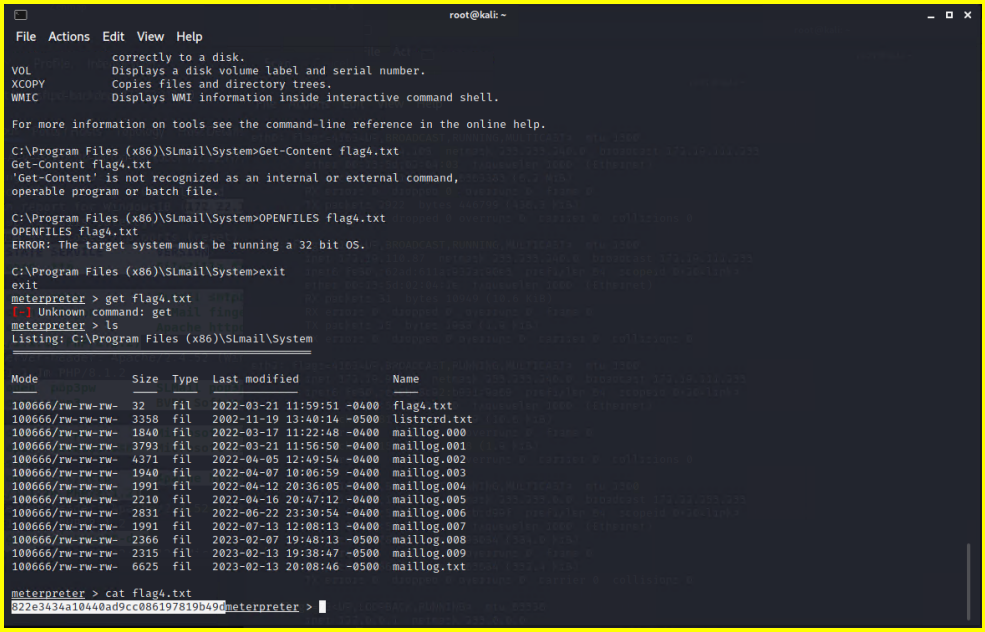
Images	  
Affected Hosts	192.168.14.35
Remediation	Disallow file paths from being able to be changed by the user directly.

Vulnerability 7	Findings
Title	Sensitive Data Exposure
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Robots.txt is available by default. WHIS was able to access /robots.txt by navigating to the page.

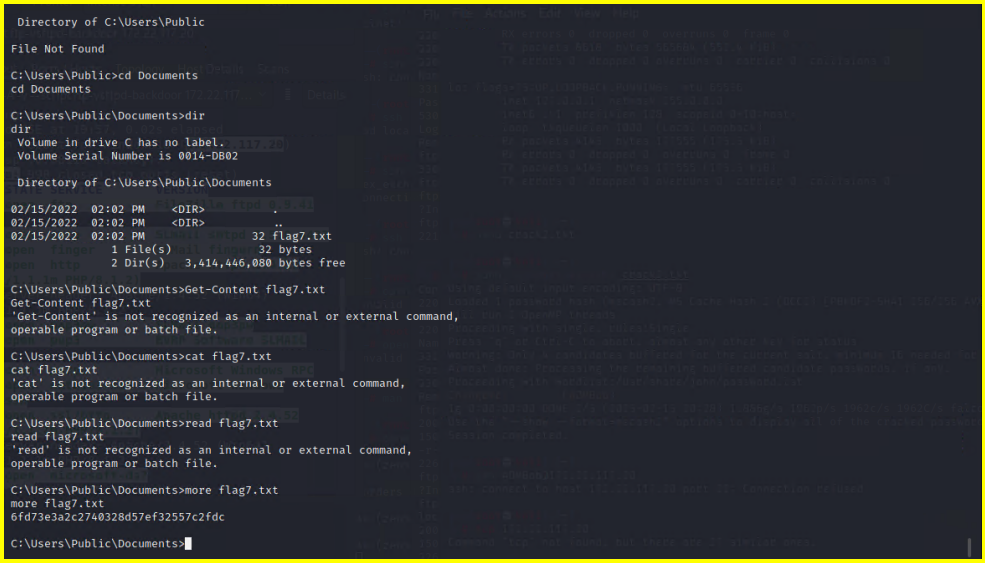
Images	
Affected Hosts	192.168.14.35
Remediation	Restrict access to sensitive data. Turn off default settings.

Vulnerability 8	Findings
Title	User Credential Exposure
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	The username dougquaid and the password (hash) kuato can be found in the HTML.
Images	
Affected Hosts	192.168.14.35
Remediation	Delete user credentials from the HTML code. Add two-factor authentication for all logins.

Vulnerability 9	Findings
Title	SLMail

Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Using searchsploit, WHIS was able to find a module that uses the port 110 SLMail exploit, windows/pop3/seattlelab_pass. After setting RHOSTS to 172.22.117.20, WHIS was granted a Meterpreter shell.
Images	
Affected Hosts	172.22.117.20
Remediation	Restrict access to port 110. Replace SLMail service.

Vulnerability 10	Findings
Title	Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617)
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Within MSFconsole, WHIS was able to search for exploits that use Tomcat and JSP. Using the multi/http/tomcat_jsp_upload_bypass exploit, and setting the RHOSTS to 138.168.13.10, WHIS was able to enter a Meterpreter shell.

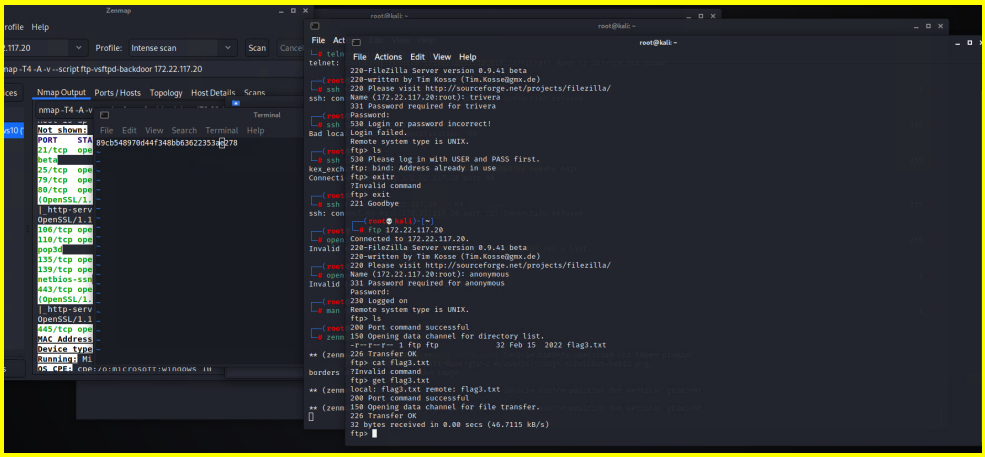
Images	
Affected Hosts	192.168.13.10
Remediation	Patch and update Apache Tomcat

Vulnerability 11	Findings
Title	CVE-2019-14287
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	WHIS was able to enumerate an SSH user, Alice. Using brute force, our team was able to guess the user password was also alice. WHIS was then able to ssh into the server using the ssh alice@192.168.13.14 command. Our team was then able to escalate user privilege with the sudo -u#-1 exploit.
Images	
Affected Hosts	192.168.13.14
Remediation	Close port 22. Require stronger login credentials. Implement two-factor authentication.

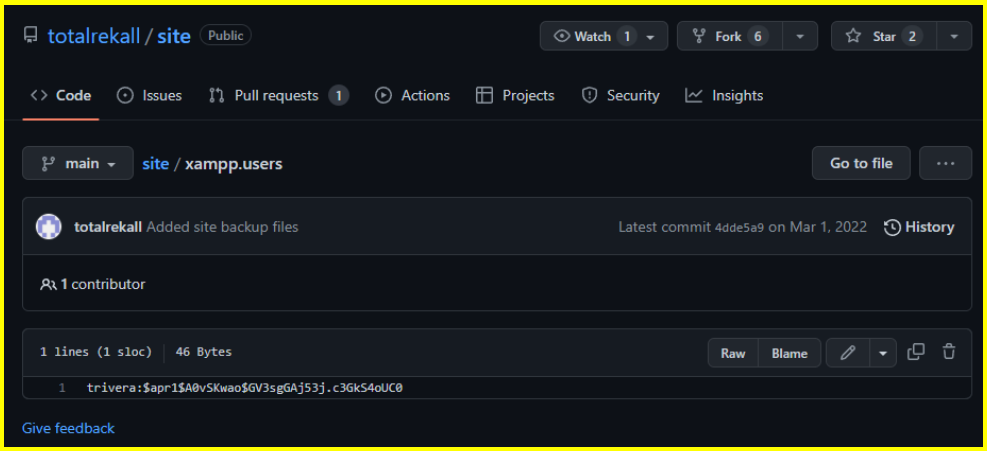
Vulnerability 12	Findings

Title	XSS Reflected
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	WHIS was able to reflect a potentially malicious script on the host home welcome.php page and the memory-planner.php page using the script <script>alert("hi")</script> and <SCRIPscriptT>alert("hi")</SCRIPscripTt> respectively.
Images	
Affected Hosts	192.168.14.35
Remediation	Input Validation

Vulnerability 13	Findings
Title	XSS Stored
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	WHIS was able to store a potentially malicious script on the host home comments.php page using the script <script>alert("hi")</script>
Images	
Affected Hosts	192.168.14.35
Remediation	Use XSS protection that does not allow injection of script code.

Vulnerability 14	Findings
Title	FTP Enumeration
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	By using an Nmap -A scan of the host server, WHIS enumerated that FTP was open on port 21 with anonymous access. Using ftp 172.22.117.20 anonymous grants access via the ftp service. Once logged in, our team was able to download and read server information.
Images	
Affected Hosts	172.22.117.20
Remediation	Restrict access to port 21.

Vulnerability 15	Findings
Title	GitHub Repo
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Using user credentials found in the totalrekall GitHub repo, WHIS was able to crack the password hash using John and gain login access.

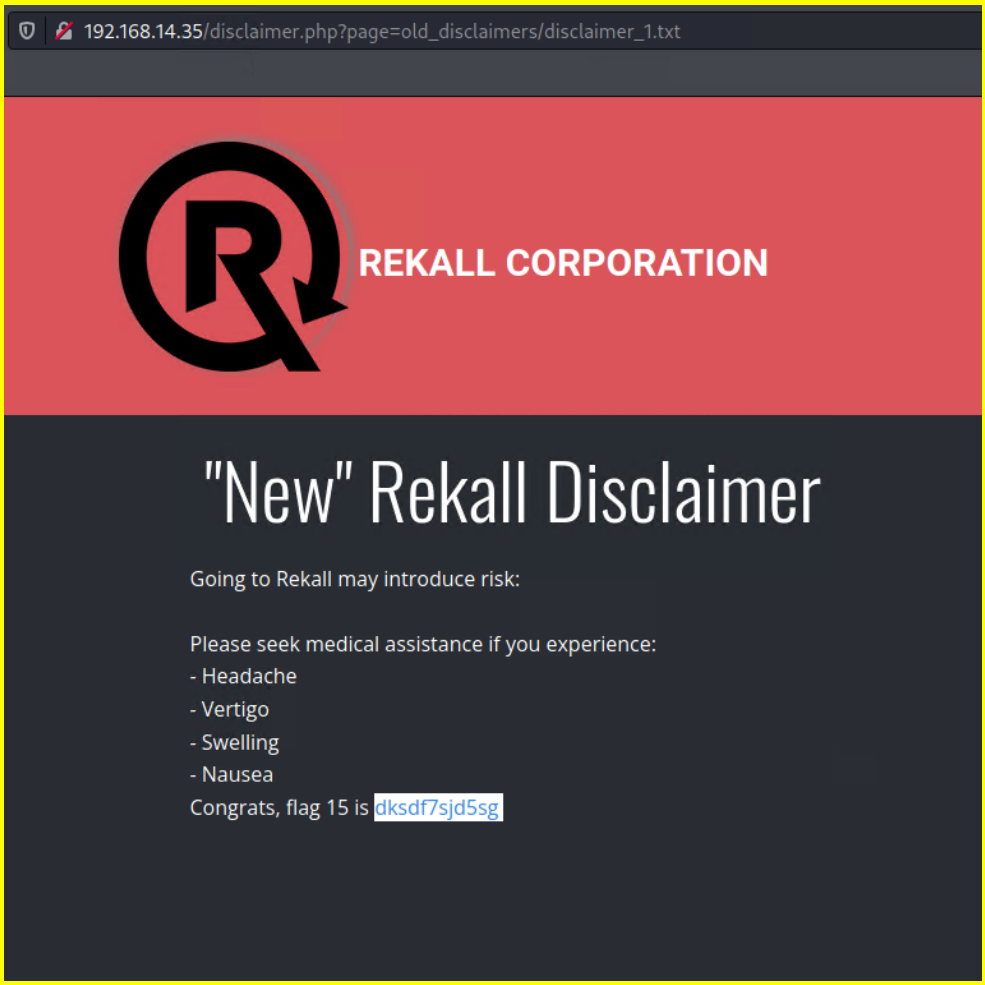
Images	
Affected Hosts	192.168.14.35
Remediation	Restrict GitHub access. Remove credentials from totalrekall repo.

Vulnerability 16	Findings
Title	Brute Force Attack
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	Using the command injection vulnerability, WHIS was able to view the /etc/passwd file on the 192.168.14.35 host server. In that file our team was able to enumerate the user melina. Using brute force, our team discovered the user's password was also melina.
Images	Successful login! flag 12 is hsk23oncsd , also the top secret legal data located here: **HERE**
Affected Hosts	192.168.14.35
Remediation	Require stronger passwords. Require two-factor authentication.

Vulnerability 17	Findings
Title	PHP Injection
Type (Web app / Linux OS / WIndows OS)	Web App

Risk Rating	Critical
Description	Souvenirs.php was enumerated from the robots.txt file. Using a php injection payloadhttp://192.168.13.35/souvenirs.php?message=""; system('cat /etc/passwd') , WHIS was able to access the web page.
Images	
Affected Hosts	192.168.14.35
Remediation	Use a php security linter. Avoid weak sanitization methods. Avoid using commands that call the operating environment directly from php, if possible.

Vulnerability 18	Findings
Title	Directory Traversal
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Using enumerated information, WHIS was able to locate the old_disclaimers directory and the contents within/ Our team was then able to change the URL to http://192.168.13.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt

Images	
Affected Hosts	192.168.14.35
Remediation	Fix command injection vulnerabilities by implementing input validation for unintended access.

Add any additional vulnerabilities below.