play with CVE-2014-6271

2020 年 10 月 28 日

source : read from secutiry course
machine used : Linux 3.13.0-35-generic 2014 i686 GNU/Linux
target : enviroment variables in bash
input command

```
\$ env x=’() { :;}; echo vulnerable’ bash −c ”echo test”
executed echo;
vulnerable
test
```

the result satisfy the condition of vulnrability. now let’s explain the detail of what this command do.

First, what is command *env* do. We can find info easily using *man env*. After that, we obtain,

```
env [OPTION]... [−] [NAME=VALUE]... [COMMAND [ARG]...]
Set each NAME to VALUE in the environment and run COMMAND.
```

which means we have a environment variable x. while executing *bash -c ”echo test”*, this command first read all environment[1] variable. When it comes to read x we set before, command *() :; ; echo vulnerable* will execute[2]. then the stdout will print ’vulnerable’.[3].

[1] why read and how to read?

[2] why reading environment means execute environment, what about the other environment value?

[3] what’s : mean in shell?

In the previous illustrate, attack command is *echo vulnerable*, which actually does nothing to attack though. But we can change this command to other real malicious command.

Here is a imaging example to use this vulnerability. Imaging there is a web server using CGI, the HTTP request , saying HTTP_USER_AGENT, is often included as environment variables. We can spoof user agent to be something like *’() :; ; echo foo’*, in which *echo foo* can be malicious command, saying create a vulnerable shell.