**Bash Shellshock 漏洞分析与复现**

一、漏洞简单介绍

Shellshock，又称Bashdoor，是在Unix中广泛使用的Bash shell中的一个安全漏洞，首次于2014年9月24日公开。许多网页服务器，使用bash来处理某些命令，从而允许攻击者在易受攻击的Bash版本上执行任意代码。这可使攻击者在未授权的情况下访问计算机系统。此漏洞的影响对象为Bash 1.14~Bash 4.3的Linux/Unix系统

1. 漏洞复现

1.检查当前Linux的bash版本

由于“破壳”漏洞只对Linux系统bash版本1.14~4.3有影响，所以在复现漏洞开始，先对当前Linux的bash版本号进行检验

命令行：bash -version

由此可见：此Linux的bash版本为4.1.2可以实现

2.检测此漏洞是否存在



命令行：env x=’() { :; };echo vulnerable’ bash -c “echo this is a test”

如果上一步输出vulnerable，并且打印this is a test 则说明漏洞存在

3.漏洞利用

Bash读取了环境变量，在定义foo之后就直接调用了后面的函数，一旦调用bash，自定义语句就直接触发

利用shellshock漏洞来获取权限

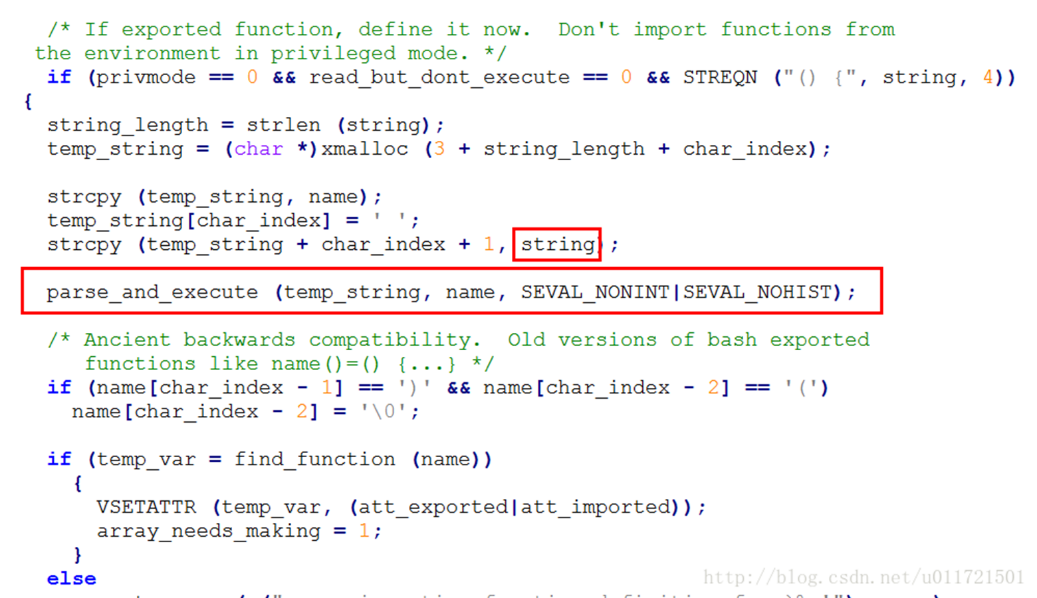
输入sudo ln -sf /bin/bash /bin/sh：使/bin/sh指向/bin/bash

利用漏洞

无需密码直接获取root

三、漏洞原因分析

这段代码是解析匿名函数的时候使用的，使用匿名函数我们才能达到执行的目的：parse\_and\_execute



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

简单来说，此漏洞的原理就是代码和数据部分没有正确的区分，如同SQL注入，通过设计特别的参数使得解析器错误的执行了参数中的命令。

可以看出，这个语句原本的意图是使用env命令创建一个临时环境，然后在里面执行一个bash命令。从解析上看，bash解析并没有问题，语法是正常的。所以应该是env命令处理变量名时的漏洞。bash可以将shell变量导出为环境变量，还可以将shell函数导出为环境变量！当前版本的bash通过以函数名作为环境变量名，以“（）{”开头的字串作为环境变量的值来将函数定义导出为环境变量。此漏洞在于bash处理这样的“函数环境变量”的时候，并没有以函数结尾“}”为结束，而是一直执行其后的shell命令。所以，在某种环境，bash会在给导出的函数定义处理环境时执行用户代码。

 strcpy (temp\_string + char\_index + 1, string);

parse\_and\_execute (temp\_string, name, SEVAL\_NONINT|SEVAL\_NOHIST); 

BASH除了可以将shell变量导出为环境变量，还可以将shell函数导出为环境变量。当前版本的bash通过以函数名作为环境变量名，以“（）{”开头的字串作为环境变量的值来将函数定义导出为环境变量。

此次爆出的漏洞在于BASH处理这样的“函数环境变量”的时候，并没有以函数结尾“}”为结束，而是一直执行其后的shell命令。