ShellShock 漏洞复现原理

朱凯

4042017046

1. ShellShock漏洞介绍

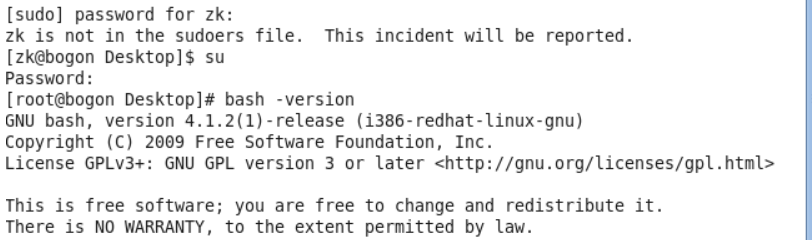
Shellshock是在Unix 中广泛使用的 Bashshell 中的一个安全漏洞，首次于 2014年 9 月 24 日公开。它的原理是利用了Bash在导入环境变量函数时候的漏洞，启动Bash的时候，它不但会导入这个函数，而且也会把函数定义后面的命令执行。

由于服务器的cgi脚本调用了bash命令，由于bash版本过低，攻击者把有害数据写入环境变量，传到服务器端，触发服务器运行Bash脚本，完成攻击。

影响对象为 Bash1.14~Bash4.3 的 Linux/Unix 系统

1.查看bash版本号：

$bash –version



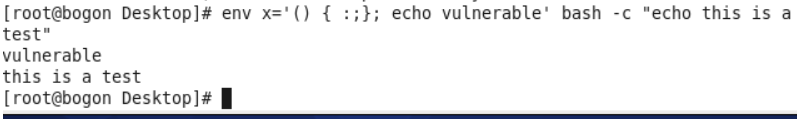
输入一条验证命令：

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

如果在一个含有版本号小于bash 4.3的linux或者unix系统，本地执行以上命令，可能会得到以下输出：

Vulnerable this is a test

本地执行结果如下：



输出vulnerable,说明bash有漏洞。

三、漏洞原理分析

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。

四、漏洞复现

检测存在漏洞并让/bin/sh 指向 /bin/bash

$ sudo ln -sf /bin/bash /bin/sh

然后写漏洞利用的代码

#include <stdio.h>

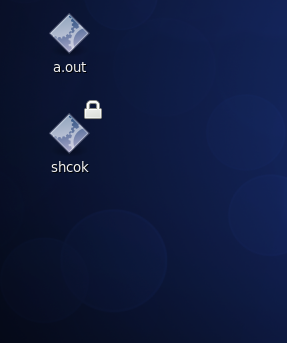
void main()

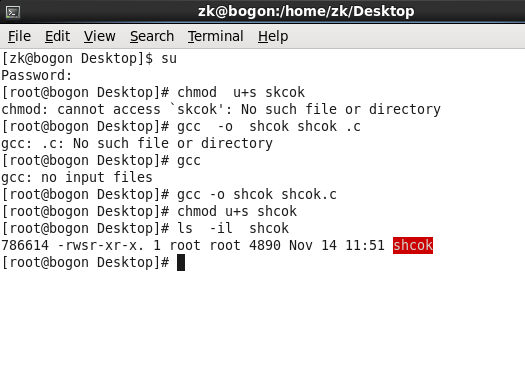
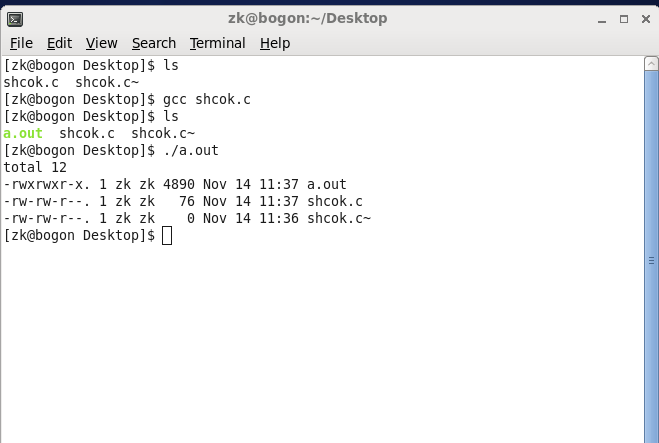
{

setuid(geteuid());

system("/bin/ls -l");

}

保存为shcok.c 

进行编译

进行攻击

