# shellshock漏洞复现

4042017018

罗栋兰

1. 漏洞原理

Bash在运行的过程中会调用操作系统的环境变量，并且会执行一些设置命令。通过ShellShock漏洞，入侵者可以把某些”本来没有权限执行的语句或者命令“，注入到环境变量里。当bash设置环境变量的时候，就会执行这些”被注入“命令。这样子便绕过用户权限的限制，把一些”没有权限执行的命令“，变成”具有执行权限的命令“了。从而可以在系统内任意执行Bash命令语句，胡作非为（相当于获得root超级用户权限）。

上面其实已经简单的描述了ShellShock的原理，下面详细说一下ShellShock漏洞的一些细节。

（1）如何测试自己的系统有没有ShellShock漏洞。

假设是Linux 系統的管理员，在 shell 中输入下面这条测试命令：

env x=’() { :;}; echo XXXXXX’ bash -c “echo This is a test code"

可能会出现两种情况：

   1）「XXXXXX」就是表示系统有漏洞，需要马上更新。

   2）「bash: warning: x: ignoring function definition attempt」表示不存在漏洞。

（2）为什么说出现XXXXXX就是有漏洞呢？

咱们先看看ShellShock漏洞的过程（引自Bash ShellShock漏洞简析）。

   1）Linux WEB Server一般可以提供CGI接口，允许远程执行Bash命令；

   2）对于HTTP头部，CGI脚本解析器会将其当作环境变量，调用bash的env相关函数设置到临时环境变量中；

   3）HTTP协议允许发送任意客户端自定义的HTTP头部；

   4）这样就产生了一个完整的可供Bash命令注入的场景，客户端故意发送构造好的带攻击命令的HTTP头部到服务端，服务端调用设置环境变量的函数，直接执行了客户端指定的头部里面的命令。并且还会将结果一并返回给客户端。

再回到刚才那个漏洞测试语句看一下：

    1）env x=’() { :;}; echo XXXXXX’

     这一整句是一条设置环境系统的语句，按理来说，它的功能只是执行系统环境变量的功能。

    2）”echo XXXXXX“

     这是一条shell打印语句，理论上不应该出现在环境变量设置里面，就算出现了也不应该执行的。表示打印“XXXXXX”这个字符串的意思。理论上，这句命令是不应该被执行的，如果被执行了，就意味着有漏洞。入侵者可以把”echo XXXXXX“ 这句命令，换成其他的shell命令，从而做一些不被允许的事情。

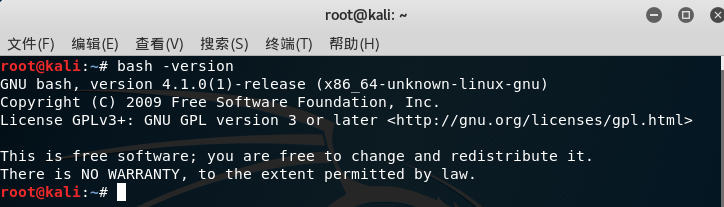
    3）通过这个漏洞就获取到超级用户root权限的说法不太准确。

1. 漏洞验证方法

第一步：由于自带bash版本高于4.1 重新安装bash-命令

wget <http://labfile.oss.aliyuncs.com/bash-4.1.tar.gz>

$ tar xf bash-4.1.tar.gz  
$ cd bash-4.1  
$ ./configure  
$ make & make install  
安装完成  
查看bash版本号  
$bash -version



第二步：漏洞信息验证

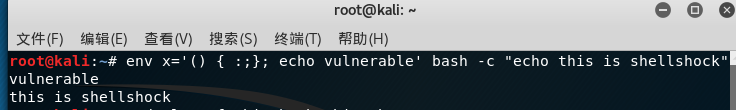
命令：env x='() { :;}; echo vulnerable' bash -c "echo this is shellshock "

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 shellshock

其中如果出现第一行vulnerable则说明该系统存在一个由bash程序缺陷导致的任意命令执行漏洞。

本地执行结果如下：



第三步：漏洞真身复现

如果上一步输出vulnerable，那么让/bin/sh指向/bin/bash

命令：sudo ln -sf /bin/bash /bin/sh

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。命令：

export ShellShock='() { :; }; echo hello shellshock'

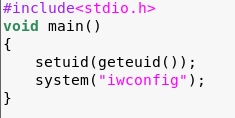
bash

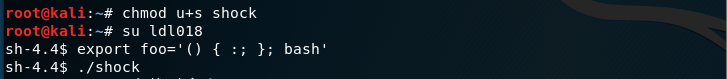


第四步：攻击获取权限

攻击Set-UID程序。我们知道system()函数将调用“/bin/sh -c”来运行指定的命令，这也意味着/bin/bash会被调用。这就能利用shellshock来获取权限。首先确保安装了带有漏洞版本的bash版本，并让/bin/sh指向/bin/bash 此操作在第三步已经完成。

编辑如下C代码：





赋予SUID权限，成功拿到root权限。

漏洞攻击原理分析我们这里使用该语句的原因是希望read uid = effective uid

这就说明如果real uid 和effective uid相同的话，定义在环境变量中的内容 在该程序中任然有效，但是两个uid如果不同，环境变量失效，就无法发动攻击了。