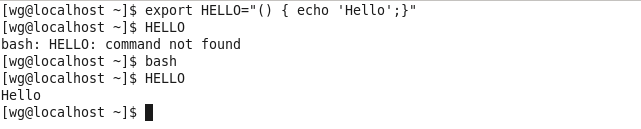
Shellshock漏洞复现

一、本地实现



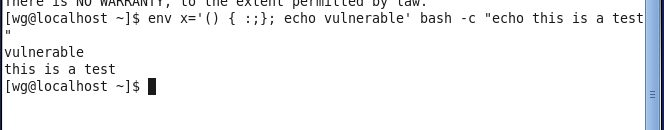
如果环境变量的值以字符“() {”开头，那么这个变量就会被当作是一个导入函数的定义（Export）

只有在shell启动的时候这个定义才生效

下面对Bash独特的定义方式进行测试，因为这种独特的方法只会在shell启动的时候生效，所以使用测试这个漏洞时常用的env命令，这个命令只需要一行可以实现刚才的定义和调用

env命令代表“先设置下面的环境变量，再运行下面的程序”，并且执行完成后当前的环境变量不受影响。

bash命令指定了-c选项是为了启动bash时就执行HELLO函数，这里必须新起一个bash，因为只有bash启动的时候才会去解析函数的定义。



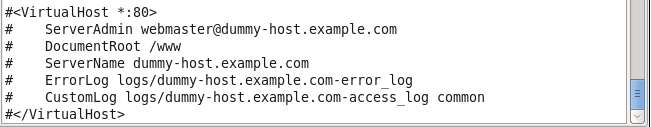
问题出现在bash解析完函数定义，执行函数的时候，它自动导入函数的解析器越过了函数定义的结尾，接着执行后面的代码，并且由于每一个新的bash启动时都会触发这个漏洞，相当于任意代码都能被执行了。

二、远程实现

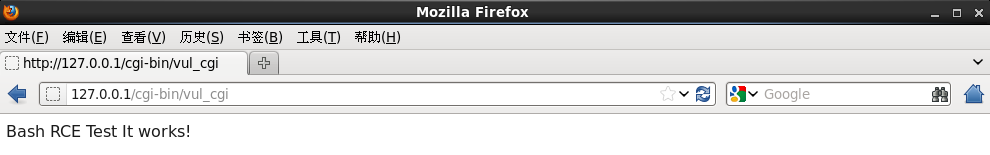
构造一个具有bash漏洞的cgi脚本



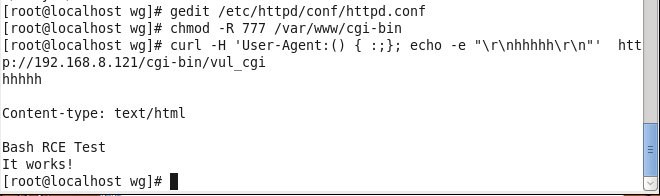
配置好apache服务器后，



访问一下自己写的这个脚本



为了成功访问，还需要提升一下脚本文件的权限



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。

Curl同样是一个这样形式的命令，在中间用刚才的形式插入一段想要加入的命令，例如echo -e “\r\nhhhhh\r\n”，访问之后发现同样会输出这一段异常，之后才会输出正常的网页。

总结：本地的漏洞形式很容易实现，但是远程实现的时候由于需要对apache服务器进行配置，花费了很多时间查资料才成功访问，不过也因此又学会了一些知识。由于这个漏洞是基于bash的，所以影响非常广泛，不只是像apache、DHCP这样的服务器，在和用户进行交互的时候会使用bash脚本，并且在很多linux的底层框架里也会应用到，所以就导致了这个漏洞始终没有被彻底修复。，