

# 目录

01

## 系统加固

Windows操作系统加固

Linux操作系统加固

02

## 中间件加固

IIS加固

Apache加固

Nginx加固

03

## 数据库加固

Mysql加固

Mongodb加固

# Linux安全加固



# 目录

01

Linux系统基本操作

02

Linux系统加固

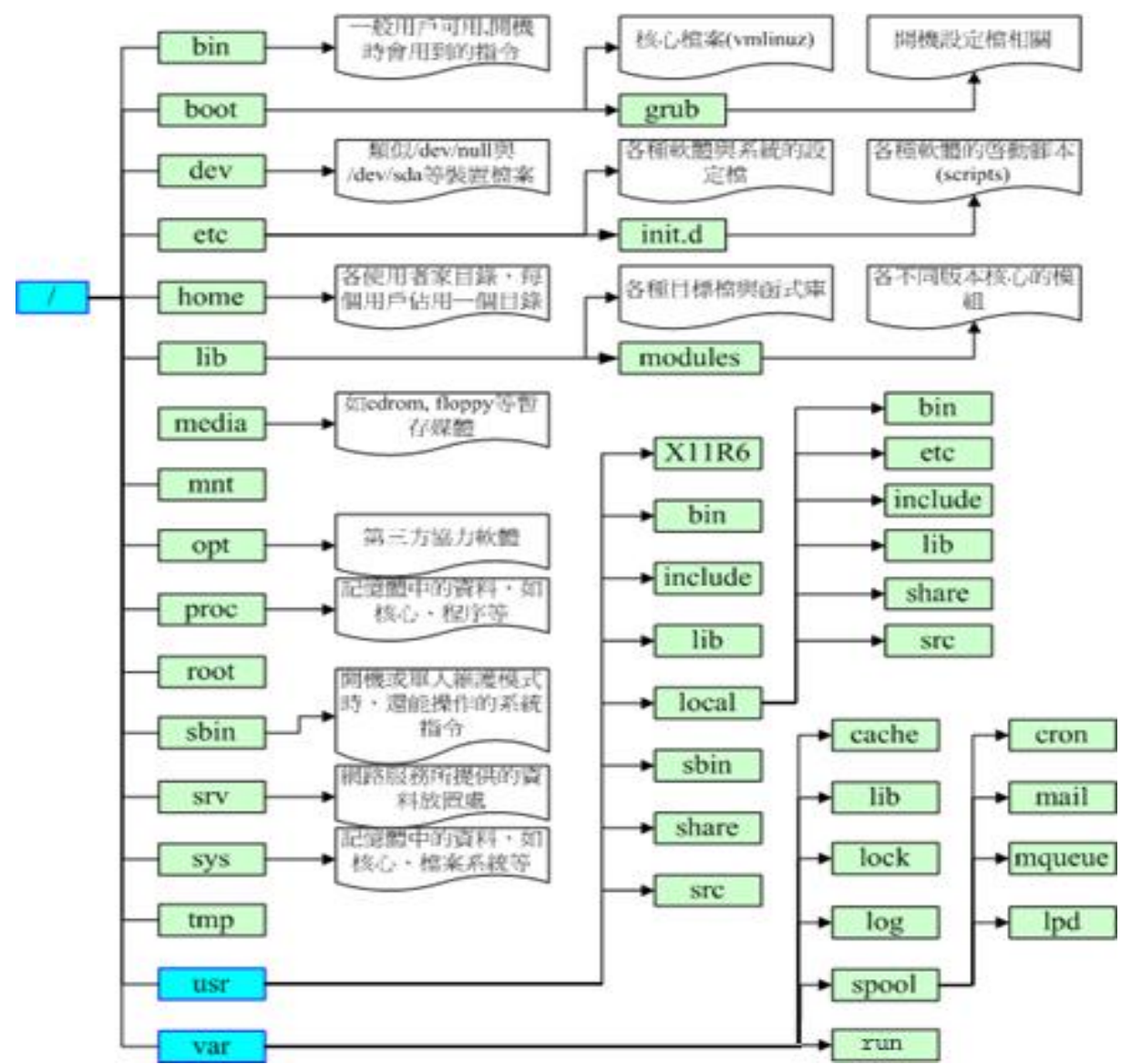
# 文件结构图及关键文件功能介绍

---

# Linux文件结构

```
root@localhost:/
[ root@localhost /]# ls
bin boot dev etc home lib lib64 lost+found media misc mnt net opt proc root sbin selinux srv sys tmp usr var
[ root@localhost /]# ll
总用量 98
dr-xr-xr-x.  2 root root  4096 11月 12 04:00 bin
dr-xr-xr-x.  5 root root 1024 11月 12 04:08 boot
drwxr-xr-x. 19 root root 3820 11月 19 23:52 dev
drwxr-xr-x. 118 root root 12288 11月 19 23:54 etc
drwxr-xr-x.  3 root root  4096 9月 23 2011 home
dr-xr-xr-x. 11 root root  4096 11月 12 03:56 lib
dr-xr-xr-x.  9 root root 12288 11月 12 03:56 lib64
drwx-----  2 root root 16384 11月 12 03:47 lost+found
drwxr-xr-x.  3 root root  4096 11月 19 23:54 media
drwxr-xr-x.  2 root root    0 11月 19 23:52 misc
drwxr-xr-x.  3 root root  4096 11月 12 04:20 mnt
drwxr-xr-x.  2 root root    0 11月 19 23:52 net
drwxr-xr-x.  3 root root  4096 11月 12 04:00 opt
dr-xr-xr-x. 167 root root    0 11月 19 23:52 proc
dr-xr-x---. 25 root root  4096 11月 19 23:54 root
dr-xr-xr-x.  2 root root 12288 11月 12 04:20 sbin
drwxr-xr-x.  7 root root    0 11月 19 23:52 selinux
drwxr-xr-x.  2 root root  4096 9月 23 2011 srv
drwxr-xr-x. 13 root root    0 11月 19 23:52 sys
drwxrwxrwt. 17 root root  4096 11月 20 00:14 tmp
drwxr-xr-x. 13 root root  4096 11月 12 03:48 usr
drwxr-xr-x. 22 root root  4096 11月 12 03:59 var
[ root@localhost /]#
```

# Linux文件结构图



# 二级目录

目录	功能
/bin	放置的是在单人维护模式下能够被操作的指令，在/bin底下的指令可以被root与一般账号所使用
/boot	这个目录只要在放置开机会使用到的文件，包括Linux核心文件以及开机选单与开机所需配置的文件等等
/dev	在Linux系统上，任何装置与接口设备都是以文件的形态存在于这个目录当中的
/etc	系统主要的配置文件几乎都放置在这个目录内，例如人员的账号密码文件，各种服务的启动档，系统变量配置等
/home	这是系统默认的用户家目录(home directory)
/lib	/lib放置的则是在开机时会用到的函式库，以及在/lib或/sbin底下的指令会呼叫的函式库

# 二级目录

目录	功能
/media	/media底下放置的是可移出的装置，包括软盘、光盘、DVD等等装置都杂事挂载与此
/opt	给第三方协议软件放置的目录
/root	系统管理员（root）的家目录
/sbin	放置/sbin底下的为开机过程中所需要的，里面包括了开机、修复、还原系统所需要的指令。
/srv	srv可视为[service]的缩写，是一些网络服务启动之后，这些服务所需要取用的数据目录
/tmp	这是让一般使用者或者是正在执行的程序暂时放置文件的地方



# 文件

目录	功能
/etc/passwd	记录系统本地用户的属性信息，如UId,Gid,家目录等信息
/etc/shadow	存放用户的口令等信息，只有系统管理员用户能查看
/etc/pam.d/system-auth	账户安全配置文件
/etc/login.defs	设置用户账户限制的配置文件，对root用户无效
/etc/securetty	网络配置文件
/etc/pam.d/su	su命令配置
/etc/hosts.allow /etc/hosts.deny	允许/拒接某网段远程连接到主机
/etc/profile	保存Linux全局环境变量信息，如umask,bash历史命令设置

# 账号和权限

---

# 系统用户


账号分类:

超级管理员	uid=0,
系统默认用户	系统程序使用, 从不登录
新建普通用户	uid大于500

```
root@localhost:/
[ root@localhost /]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
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
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/cache/rpcbind:/sbin/nologin
rtkit:x:499:497:RealtimeKit:/proc:/sbin/nologin
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
abrt:x:173:173:/:/etc/abrt:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
```

```
[root@svr5 ~]# head -1 /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
```



字段1：用户帐号的名称  
字段2：密码字符串或占位符 **x**  
字段3：用户帐号的UID号  
字段4：所属基本组的GID号  
字段5：用户全名  
字段6：宿主目录  
字段7：登录Shell程序的路径

```
[root@svr5 ~]# head -1 /etc/shadow
```

```
root:$1$SmIKPNho$qNq...OGBSmvil6b1:15908:0:99999:7:::
```

字段1：用户帐号的名称

字段2：加密后的密码字符串

字段3：上次修改密码的时间

字段4：密码的最短有效天数，默认0

字段5：密码的最长有效天数，默认99999

字段6：密码过期前的警告天数，默认7

字段7：密码过期后多少天禁用此用户账号

字段8：帐号失效时间，默认值为空

字段9：保留字段（未使用）

# 用户管理

添加用户：

`useradd <用户名>`

删除用户：

`userdel [-r][-f] <用户名>`

锁定/解锁用户：

`passwd -l <用户名>`

`passwd -u <用户名>`

用户属性：

`usermod -L <用户名>` 锁定用户

`usermod -U <用户名>` 解锁用户

查看当前用户

`id`

# 权限管理

---

# 解析文件权限

- 执行 `ls -l ...` 命令查看
  - 输出信息包括7个字段

```
[root@svr5 ~]# ls -ld /etc/passwd /boot
drwxr-xr-x 4 root root 1024 07-10 17:22 /boot
-rw-r--r-- 1 root root 1715 08-24 11:42 /etc/passwd
```

权限位 硬链接数 属主 属组 大小 最后修改时间 文件/目录名称

## 权限和归属

- 访问权限
  - 读取：允许查看内容-**read**
  - 写入：允许修改内容-**write**
  - 可执行：允许运行和切换-**execute**
- 归属关系
  - 所有者：拥有此文件/目录的用户-**user**
  - 所属组：拥有此文件/目录的组-**group**
  - 其他用户：除所有者、所属组以外的用户-**other**
  - 所有用户：以上三类归属合称-**all**

共同决定最终权限



查看权限:

```
[root@localhost ~]# ls -l
```

总用量 98

```
dr-xr-xr-x.  2 root root  4096 11月 12 04:00 bin
```

修改权限:

```
[root@localhost 1234]# chmod 777 test
```

总用量 0

```
-rwxrwxrwx. 1 root root 0 11月 20 02:09 test
```

chown

```
[root@localhost 1234]# chown 123:123 test
```

总用量 0

```
-rwxrwxrwx. 1 123 123 0 11月 20 02:09 test
```

chgrp

```
[root@localhost 1234]# chgrp root test
```

总用量 0

```
-rwxrwxrwx. 1 123 root 0 11月 20 02:09 test
```

# 设置合理的初始文件权限

很奇妙的UMASK:

```
[root@localhost 1234]# umask  
0022
```

umask值为0022所对应的默认文件和文件夹创建的缺省权限分别为644和755

文件夹其权限规则为:  $777-022=755$

文件其权限规则为:  $777-111-022=644$  (因为文件默认没有执行权限)

修改UMASK值:

- 1、直接在命令行下umask xxx (重启后消失)
- 2、修改/etc/profile中设定的umask值

```
root@ubuntu:~/test# umask 027  
root@ubuntu:~/test# mkdir mask  
root@ubuntu:~/test# ls -l  
total 4  
drwxr-x--- 2 root root 4096 Apr 10 17:12 mask  
root@ubuntu:~/test#
```

# 脏牛漏洞

```
ubuntu:~$ uname -a
ubuntu:~$ gcc --version
ubuntu:~$ gcc -pthread dirtycow.c -o dirtycow
ubuntu:~$ echo ABCDEFGHIJKLMN > target.txt
ubuntu:~$ chmod 644 target.txt
ubuntu:~$ sudo chown root:root target.txt
ubuntu:~$ ls -l target.txt
-rw-r--r-- 1 root root 15 10月 30 13:14 target.txt
```

```
ubuntu:~$ ./dirtycow target.txt 1234567890
mmap 7fa185de3000
Hack success!
procselfmem 52150
madvise 0
ubuntu:~$ ls -l target.txt
-rw-r--r-- 1 root root 15 10月 30 13:14 target.txt
```

# 系统加固

---

# 锁定系统中多余的自建帐号

执行命令

```
#cat /etc/passwd
```

```
#cat /etc/shadow
```

查看账户、口令文件，与系统管理员确认不必要的账号。对于一些保留的系统伪帐户如：bin, sys,

adm, uucp, lp, nuucp, hpdb, www, daemon等可根据需要锁定登陆。

加固方法:

使用命令passwd -l <用户名>锁定不必要的账号。

使用命令passwd -u <用户名>解锁需要恢复的账号。

# 检查shadow中空口令帐号

检查方法:

```
# awk -F ":" '($2=="!"){print $1}' /etc/shadow
```

```
root@localhost:~# awk -F ":" '($2=="!"){print $1}' /etc/shadow
mysql
stunnel4
Debian-exim
arpwatch
speech-dispatcher
redsocks
ss1h
as
```

加固方法:

使用命令 `passwd -l <用户名>` 锁定不必要的账户。

使用命令 `passwd -u <用户名>` 解锁需要恢复的账户。

使用命令 `passwd <用户名>` 为用户设置密码

# 设置系统密码策略

执行命令

```
#cat /etc/login.defs|grep PASS查看密码策略设置
```

加固方法：

```
#vi /etc/login.defs修改配置文件
```

PASS_MAX_DAYS	90	#用户的密码最长使用天数
PASS_MIN_DAYS	0	#两次修改密码的最小时间间隔
PASS_MIN_LEN	7	#密码的最小长度
PASS_WARN_AGE	9	#密码过期前多少天开始提示

# 禁用root之外的超级用户

检测方法:

`awk -F ":" '($3=="0"){print $1}' /etc/passwd` 检查用户ID为0的用户

加固方法:

`passwd -l <用户名>`

锁定用户

```
[root@localhost 1234]# awk -F ":" '($3=="0"){print $1}' /etc/passwd
root
[root@localhost 1234]# █
```



# 限制能够su为root的用户

#cat /etc/pam.d/su,查看是否有auth required /lib/security/pam\_wheel.so这样的配置条目

```
[root@localhost 1234]# cat /etc/pam.d/su | grep wheel
# Uncomment the following line to implicitly trust users in the "wheel" group.
#auth      sufficient      pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the "wheel" group.
#auth      required_       pam_wheel.so use_uid
```

加固方法

在头部添加：

auth required /lib/security/pam\_wheel.so group=wheel

这样，只有wheel组的用户可以su到root

#usermod -G10 test 将test用户加入到wheel组

# 重要文件加上不可改变属性

把重要文件加上不可改变属性

```
[root@ayazero /]# chattr +i /etc/passwd
```

```
[root@ayazero /]# chattr +i /etc/shadow
```

```
[root@ayazero /]# chattr +i /etc/gshadow
```

```
[root@ayazero /]# chattr +i /etc/group
```

```
[root@ayazero /]# chattr +i /etc/inetd.conf
```

```
[root@ayazero /]# chattr +i /etc/httpd.conf
```

# Umask安全

```
root@TUbuntu:~# umask  
0022  
root@TUbuntu:~#
```

第一个0代表suid 丢弃的权限；

第二个0代表本文件/目录拥有者什么权限都没丢弃

第三个2代表本文件/目录的用户组丢弃了w权限

第四个2代表本文件/目录的其他用户能使用的权限只有有r和x

**实际权限=7777-0022**

设置合理的umask权限

**umask 0022**

## 禁止root用户进行远程登录

检查方法：

```
# cat /etc/ssh/sshd_config | grep PermitRootLogin 是否为no
```

加固方法

```
#vi /etc/ssh/sshd_config
```

```
PermitRootLogin no
```

```
[[root@localhost ~]# cat /etc/ssh/sshd_config | grep PermitRootLogin  
PermitRootLogin no
```

## 更改服务端口

```
#vi /etc/ssh/sshd_config
```

```
Port 2222
```

更改ssh端口

# 屏蔽SSH登录banner信息

屏蔽SSH登录banner信息

检查方法：

```
# cat /etc/ssh/sshd_config
```

查看文件中是否存在banner字段，或banner字段为NONE

```
# cat /etc/motd
```

查看文件内容，该处内容作为banner信息显示给登录用户

加固方法：

```
#vim /etc/ssh/sshd_config
```

添加：

```
banner NONE
```

```
#vim /etc/motd
```

删除全部内容或更新成自己想要添加的内容

## 仅允许SSH协议版本2

有两个SSH协议版本，仅使用SSH协议版本2会更安全，SSH协议版本1有安全问题，包括中间人攻击（man-in-the-middle）和注入（insertion）攻击。

编辑/etc/ssh/sshd\_config文件并查找下面这样的行：

Protocol 2,1

修改为

Protocol 2

```
|root@localhost:~# cat /etc/ssh/sshd_config|grep Protocol  
|Protocol 2
```

# 防止误使用Ctrl+Alt+Del重启系统

检查方法：

#cat /etc/inittab |grep ctrlaltdel 查看输出行是否被注释

ca::ctrlaltdel:/sbin/shutdown -t3 -r now

加固方法：

#vim /etc/inittab

在行开否添加注释符号“#”

#ca::ctrlaltdel:/sbin/shutdown -t3 -r now

```
root@localhost:~# cat /etc/inittab |grep ctrlaltdel
#ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

# 设置账户锁定登录失败锁定次数、锁定时间

检查方法:

# cat /etc/pam.d/system-auth | grep auth 查看有无auth required pam\_tally.so 条目的设置

加固方法:

# vi /etc/pam.d/system-auth

auth required pam\_tally.so oner=filad deny=6 unlock\_time=300 设置为密码连续错误6次, 锁定时间300秒

解锁用户: faillog -u <用户名> -r

```
bash-4.2# cat /etc/pam.d/system-auth | grep auth
# User changes will be destroyed the next time authconfig is run.
auth      required      pam_env.so
auth      sufficient     pam_fprintd.so
auth      sufficient     pam_unix.so nullok try_first_pass
auth      requisite      pam_succeed_if.so uid >= 1000 quiet success
auth      required      pam_tally.so oner=filad deny=6 unlock_time=300
password  requisite      pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=
password  _sufficient     pam_unix.so sha512 shadow nullok try_first_pass use_authtok
```



# 修改账户TMOUT值，设置自动注销时间

检查方法：

`cat /etc/profile|grep TMOUT` 查看有无TMOUT的设置

加固方法：

`#vim /etc/profile`

增加

`TMOUT=600` 无操作600秒后自动退出

```
root@localhost:~# cat /etc/profile|grep TMOUT
TMOUT = 600
```

# 设置BASH保留历史命令的条目

检查方法:

```
cat /etc/profile | grep HISTSIZE
```

```
HISTSIZE=1000
```

加固方法:

```
#vim /etc/profile
```

修改HISTSIZE=5即保留最新执行的5条命令

```
root@localhost:~# cat /etc/profile | grep HISTSIZE
HISTSIZE=5
```

# 用户注销时删除命令记录

检查方法:

查看/etc/skel/.bash\_logout文件, 增加如下行

```
rm -f $HOME/.bash_history
```

这样, 系统中的所有用户注销时都会删除其命令记录, 如果只需要针对某个特定用户, 如root用户进行设置, 则可只在该用户的主目录下修改/\$HOME/.bash\_history文件增加相同的一行即可。

```
root@localhost:~# cat /etc/skel/.bash_logout|grep rm
rm -f $HOME/.bash_history
```

# 设置系统日志策略配置文件

日志的主要用途是系统审计、监测追踪和分析统计。

为了保证 Linux 系统正常运行、准确解决遇到的各种各样的系统问题，认真地读取日志文件是管理员的一项非常重要的任务。

UNIX/ Linux采用了syslog工具来实现此功能，如果配置正确的话，所有在主机上发生的事情都会被记录下来，不管是好的还是坏的。

centos6起/etc/syslog.conf不再有！而是/etc/rsyslog.conf代替！

检查方法：

# ps -aef | grep syslog 确定syslog服务是否启用

# cat /etc/rsyslog.conf 查看syslogd的配置，并确认日志文件是否存在

系统日志 (默认) /var/log/messages

cron日志 (默认) /var/log/cron

安全日志 (默认) /var/log/secure

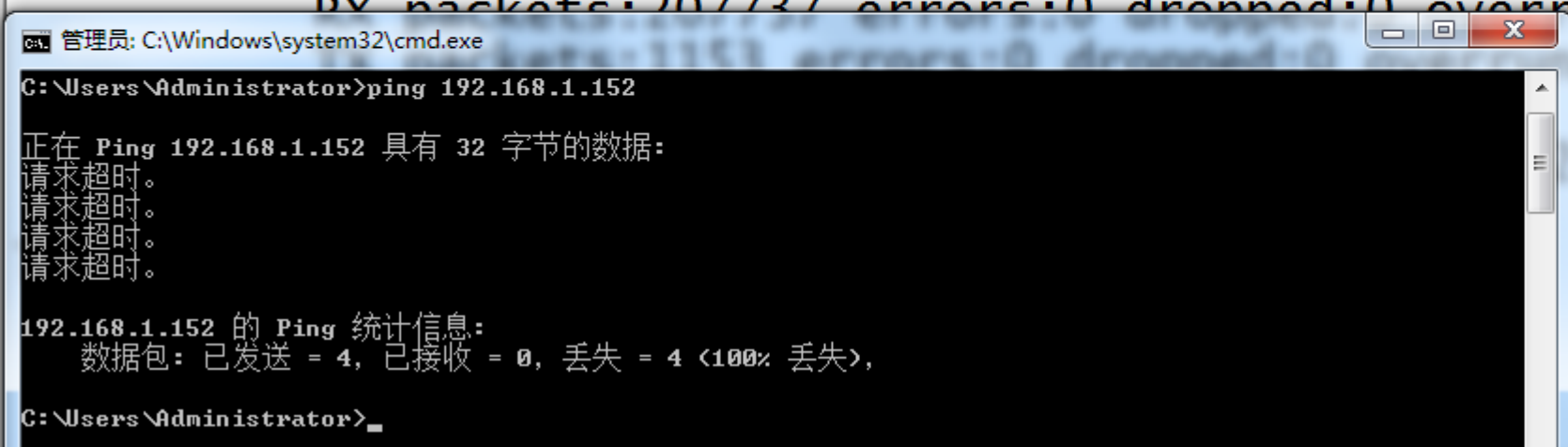
# 阻止系统响应任何从外部/内部来的ping请求

加固方法:

执行命令 `echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all`

其他客户端就不能ping通你的服务器了。

```
root@localhost:~# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
root@localhost:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:13:a8:e9
          inet addr:192.168.1.152  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe13:a8e9/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:207737 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:140 (129.0 KiB)
```



THANK YOU