操作系统安全配置

1.账号安全控制

1.基本安全措施

1.系统账号的清理、锁定

1. 程序用户的shell必须是/sbin/nologin,不能登录系统
2. 删除系统冗余用不到的账号
3. 长时间不用的账号，不确定是否使用，锁定账号usermod -L　passwd -l
4. i 锁 服务器账号固定，锁定账号配置文件chattr+i　解锁chatttr -i

```
[root@localhost ~]# useradd zhangsan
[root@localhost ~]# lsattr /etc/passwd /etc/shadow
[root@localhost ~]# chattr +i  /etc/passwd /etc/shadow
```

- 无法改用户属性，无法创建新用户，无法改密码
- i 锁，可以用于系统中不常用的文件

5. a 锁 ：不能删除现有文件，不能修改以前的内容　　----日志文件

chattr+a　解锁chatttr -a

2.密码安全控制

降低密码被猜出或暴力破解（穷举）风险，字典破解

```
[root@localhost ~]# vim /etc/login.defs              #只对新用户有效，对现有用户无效
PASS_MAX_DAYS    30             #密码最多使用30天，默认99999
```

```
PASS_MIN_DAYS   0               #密码最少使用0天
PASS_MIN_LEN    5                #可接受密码长度
PASS_WARN_AGE   7              #密码到期前的警告时间
[root@localhost ~]# useradd lisi
[root@localhost ~]# tail -2 /etc/shadow
zhangsan:!!:18324:0:99999:7:::
lisi:!!:18324:0:30:7:::
```

对已有用户的修改：

```
[root@localhost ~]# chage -M 30 zhangsan
[root@localhost ~]# tail -2 /etc/shadow
zhangsan:!!:18324:0:30:7:::
lisi:!!:18324:0:30:7:::
```

用户下次登录时必须修改密码

系统批量创建用户的时候使用

例如：政府发的卡，社保卡，。。。的初始密码

```
[root@localhost ~]# chage -d 0 zhangsan
```

小脚本~

```
[root@localhost ~]# vim user20.sh
#! /bin/bash
#批量创建用户的脚本
for i  in {1..20}
do
        useradd user$i
        echo "123456" | passwd --stdin user$i
        chage -d 0 user$i
done
[root@localhost ~]# chmod +x user20.sh
[root@localhost ~]# ./user20.sh
```

3.历史命令　history

风险，默认保存1000条，包含敏感信息，密码

修改历史命令数量

```
[root@localhost ~]# vim /etc/profile
HISTSIZE=100      #修改
[root@localhost ~]# source /etc/profile     #使之生效
```

历史命令的特点：11~111　>　12~112　后面进一个，前面退一个，维持100个

清空历史命令

```
[root@localhost ~]# vim ~/.bash_logout
# ~/.bash_logout
history -c
[root@localhost ~]# exit
[root@localhost ~]# ls .bash_history    #开机参考文件，有就加载出来
[root@localhost ~]# cat -n .bash_history
[root@localhost ~]# > .bash_history
[root@localhost ~]# cat -n .bash_history
[root@localhost ~]# cat -n .bash_logout     #退出系统时执行
     1  # ~/.bash_logout
```

```
    2  history -c
[root@localhost ~]# exit
[root@localhost ~]# history
```

使用技巧

```
[root@localhost ~]#mysql -uroot -p 123456;lhistory -c
```

- 分号回车的意思，连接两条命令，执行完命令直接删除历史记录
- 分号与&&（逻辑与），没逻辑与好，；单纯回车，逻辑与会对前面结果进行判断，成功了才执行，否则不执行

4.自动注销

锁屏习惯

设置自动注销

```
[root@localhost ~]# vim /etc/profile
```

末尾添加

```
export TMOUT=600
[root@localhost ~]# . /etc/profile  （注意点后面有空格）  或者  [root@localhost ~]# source /etc/profile
```

取消自动注销

暂时取消

```
[root@localhost ~]# unset TMOUT
```

永久注销

```
[root@localhost ~]# vim /etc/profile
```

删除末尾添加的export TMOUT=600

2.用户切换与提权

直接使用root的风险

- 操作失误造成破坏
- 降低特权密码被泄露的风险

1.su命令——切换用户

```
[root@localhost ~]# su - zhangsan
[root@localhost ~]# passwd zhangsan
[root@localhost ~]# passwd lisi
[root@localhost ~]# su - zhangsan
[zhangsan@localhost ~]$ su - lisi
[lisi@localhost ~]$ su - root
```

“ - ” = “ --login ” 表示进入环境

加 “-” ：表示进入用户的环境

不加 “-” ：表示只拥有了身份，还是原来的环境

```
[root@localhost ~]# su - zhangsan
[zhangsan@localhost ~]$ pwd

[root@localhost ~]# su zhangsan
[zhangsan@localhost root]$
```

默认所有用户允许使用su命令的风险：

1.其他用户可以尝试破解

2.账号登录混乱

3.普通用户权限不够，申请root权限

措施

PAM（Pluggable Authentication Modules） 可插拔认证模块

PAM_wheel认证模块 ： 只允许极个别用户使用su命令进行切换

PAM认证首先要确定哪一项服务，然后加载相应的PAM的配置文件（位于/etc/pam.d下），最后调用认证库文件（32位系统位于/lib/security下，64位系统位于/lib/security下）进行安全认证。

```
[root@localhost ~]# ls /etc/pam.d/
atd                  gdm-launch-environment  password-auth-ac  smartcard-auth     system-auth
chfn                 gdm-password            pluto             smartcard-auth-ac  system-auth-ac
chsh                 gdm-pin                 polkit-1          smtp               systemd-user
config-util          gdm-smartcard           postlogin         smtp.postfix       vlock
crond                ksu                     postlogin-ac      sshd               vmtoolsd
cups                 liveinst                ppp               sssd-shadowutils   xserver
fingerprint-auth     login                   remote            su
fingerprint-auth-ac  other                   runuser           sudo
gdm-autologin        passwd                  runuser-l         sudo-i
gdm-fingerprint      password-auth           setup             su-l
```

查看某个程序是否支持PAM认证

```
[root@localhost ~]# ls /etc/pam.d | grep sshd
```

```
[root@localhost ~]# cat /etc/pam.d/su
#%PAM-1.0
auth            sufficient      pam_rootok.so
# 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
auth            substack        system-auth
auth            include         postlogin
account         sufficient      pam_succeed_if.so uid = 0 use_uid quiet
account         include         system-auth
password        include         system-auth
session         include         system-auth
session         include         postlogin
session         optional        pam_xauth.so
```

认证类型

auth 认证管理（authentication management）：认证用户名和密码

account 账户管理（account management）：账号是否已过期/时间限制/被允许登录系统

password 密码管理（password management）：修改密码

session 会话管理（session management）：类似窗口

控制类型

required requisite sufficient optional include

实现过程：将授权使用su命令的用户添加到wheel组，修改/etc/pam.d/su认证配置以启用pam_wheel认证

```
[root@localhost ~]# vim /etc/pam.d/su
这行的#删掉>保存退出
auth            required        pam_wheel.so use_uid
[root@localhost ~]# gpasswd -a zhangsan wheel
正在将用户“zhangsan”加入到“wheel”组中
[root@localhost ~]# su - zhangsan
上一次登录：二 3月  3 16:37:07 CST 2020pts/0 上
[zhangsan@localhost ~]$ su - lisi
密码：
```

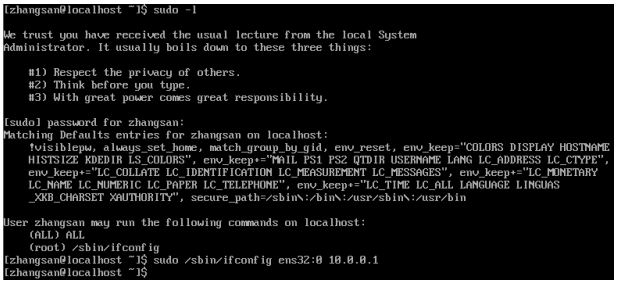
上一次登录：二 3月 3 15:05:24 CST 2020pts/1 上
[lisi@localhost ~]$ su - zhangsan
密码：
su: 拒绝权限
[root@localhost ~]# gpasswd -a lisi wheel
正在将用户“lisi”加入到“wheel”组中
[root@localhost ~]# su - lisi
上一次登录：二 3月 3 17:08:19 CST 2020pts/0 上
[lisi@localhost ~]$ su - zhangsan
密码：
上一次登录：二 3月 3 17:08:06 CST 2020pts/0 上
最后一次失败的登录：二 3月 3 17:08:33 CST 2020pts/0 上
最有一次成功登录后有 1 次失败的登录尝试。
[zhangsan@localhost ~]$

使用su命令切换用户的操作将会记录到安全日志/var/log/secure文件中
[root@localhost ~]# cat /var/log/secure

2.sudo命令——提升用户的执行权限
管理员预先授权，既可以让普通用户拥有一部分管理权限，又不需要将root用户密码告诉他

用户（user） 主机（MACHINE） 命令（COMMANDS）

[root@localhost ~]# visudo
或者 [root@localhost ~]# vim /etc/sudoers
zhangsan localhost=/sbin/ifconfig #在末尾处加入这行



sudo后面最好加绝对路径
[root@localhost ~]# sudo -l #查看授权
五分钟密码保留时间

fang localhosst=/sbin/*,!/sbin/ifconfig #fang可以执行在/sbin目录下的所有命令程序，除了ifconfig
Defaults logfile="/var/log/sudo" #启用sudo日志记录

[root@localhost ~]# cat /var/log/sudo
Mar 3 18:47:05 : zhangsan : TTY=tty2 ; PWD=/home/zhangsan ; USER=root ;
COMMAND=list
Mar 3 18:47:34 : zhangsan : TTY=tty2 ; PWD=/home/zhangsan ; USER=root ;
COMMAND=/bin/ping 127.0.0.1

2.系统引导和登录控制

1.开关机安全控制

物理安全---托管--严格控制人员进出
IDC机房

1.调整BIOS引导设置

- 将第一优先引导设备设为当前系统所在磁盘
- 禁止从其他设备引导系统（U盘，光驱，网络），对应的项为disabled。防止通过其他方式引导进入系统窃取数据。
- 将BIOS安全级别改为setup，并设好管理密码，以防止未授权修改
- 密码忘了，BIOS放电，恢复出厂设置

2.禁止Ctrl+alt+del快捷键重启

在真实服务器中，这个组合键代表重启系统

禁止Ctrl+alt+del快捷键重启的操作：

```
[root@localhost ~]# systemctl mask ctrl-alt-del.target
Created symlink from /etc/systemd/system/ctrl-alt-del.target to /dev/null.
[root@localhost ~]# systemctl daemon-reload            #生效设置--永久
```

禁止Ctrl+alt+del快捷键重启的操作：systemctl umask

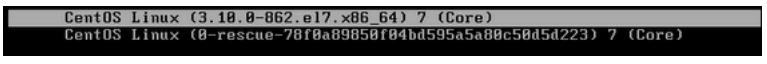
3.限制更改GRUB引导参数----做一遍永久

```
[root@localhost ~]# grub2-mkpasswd-pbkdf2
输入口令：
Reenter password:
PBKDF2 hash of your password is
grub.pbkdf2.sha512.10000.FD77D16614C502C4D891D47801CC9D1CE0963CFDDBB3EE9C6EEE975
58BB0B4A08A3C1090204A11E383E88F60519899280B13AE9C993DEAC822CECC74B07BE59B.816FF2298351D036C710DE525AAF050DD8394D
3E4704BBE089B6FFA39444B046DB4202FA21F69A23575C5C15D3C60EDBB3E87565C395A1478BC77220FA3CC931
[root@localhost ~]# cp /boot/grub2/grub.cfg /boot/grub2/grub.cfg.bak              #备份
[root@localhost ~]# cp /etc/grub.d/00_header /etc/grub.d/00_header.bak          #备份
[root@localhost ~]# vim /etc/grub.d/00_header
#在末尾处添加下面的
cat << EOF
set superusers="root"
password_pbkdf2 root  grub.pbkdf2.sha512.10000.FD77D16614C502C4D891D47801CC9D1CE0963CFDDBB3EE9C6EEE97558BB0B4A
08A3C1090204A11E383E88F60519899280B13AE9C993DEAC822CECC74B07BE59B.816FF2298351D036C710DE525AAF050DD8394D3E4704BBE089B6FFA3
EOF
```

使用grub2-mkconfig命令重新生成grub.cfg文件

```
[root@localhost ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-862.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-862.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-9dc322fbdb044a26b48372e3f270b62a
Found initrd image: /boot/initramfs-0-rescue-9dc322fbdb044a26b48372e3f270b62a.img
done
[root@localhost ~]# reboot
```

输入e



密码123 进入GRUB菜单--单用户

2.登录控制

默认6个tty终端，允许任何用户进行本地登录

限制本地登录的措施：

1.禁止root用户登录

```
[root@localhost ~]# vim /etc/securetty
```

在这两个前面加#，则root在tty1，tty2不能登录

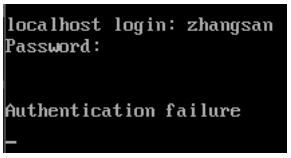
#tty1

#tty2

2.禁止普通用户登录

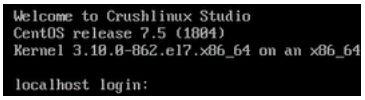
对系统做维护类操作，不希望被打扰，不让普通用户登录

[root@localhost ~]# touch /etc/nologin



[root@localhost ~]# rm -rf /etc/nologin

欢迎信息暴露的操作系统的版本和内核的版本



[root@localhost ~]# vim /etc/issue

[root@localhost ~]# vim /etc/issue

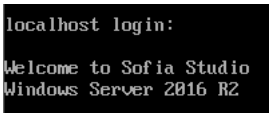
[root@localhost ~]# cp /etc/issue /etc/issue.net

cp：是否覆盖"/etc/issue.net"？ y

[root@localhost ~]# cat /etc/issue

Welcome to Sofia Studio

Windows Server 2016 R2



3.弱口令检测、端口扫描

1.弱口令检测——John the Ripper--简称JR

- **开源的密码破解软件，已知密文分析密码，先cp**
- **检测系统中的简单密码，防止出现安全风险**
- **http://www.openwall.com/john**

[root@localhost ~]# rz

[root@localhost ~]# tar xf john-1.8.0.tar.gz -C /usr/src/

[root@localhost ~]# cd /usr/src/john-1.8.0/

[root@localhost john-1.8.0]# ls -ld *

drwxr-xr-x 2 root root 208 3月   6 23:13 doc   #手册文档

lrwxrwxrwx 1 root root   10 5月 30 2013 README -> doc/README   #链接文件

drwxr-xr-x 2 root root 143 3月   6 23:13 run   #运行程序

drwxr-xr-x 2 root root 4096 3月   6 23:13 src   #源码文件

[root@localhost ~]# cd /usr/src/john-1.8.0/src

[root@localhost src]# make clean linux-x86-64

不要用John the Ripper直接对系统/etc/shadow文件进行分析，先cp出一份

[root@localhost src]# pwd

/usr/src/john-1.8.0/src

[root@localhost src]# cd ../run/

[root@localhost run]# ls

ascii.chr   john   lm_ascii.chr makechr   relbench unique

digits.chr john.conf mailer   password.lst unafs   unshadow

[root@localhost run]# cp /etc/shadow ./   #cp /etc/shadow到当前下

[root@localhost run]# ls

```
ascii.chr   john       lm_ascii.chr  makechr       relbench  unafs   unshadow
digits.chr  john.conf  mailer        password.lst  shadow    unique
[root@localhost run]# ./john shadow                    #暴力破解
[root@localhost run]# cat john.pot              #查看破解密码

[root@localhost run]# ./john --show /etc/shadow.txt              #查看已破解的账户列表
[root@localhost run]# > john.pot          #清空已破解的账户列表，以免影响下次破解
[root@localhost run]# ./john --wordlist=password.lst shadow          #用这个来对系统进行定期扫描，看看系统中有没有安全风险（过于简单的密码）
```

或者用[root@localhost run]# ./john --show shadow

或者用[root@localhost run]# cat john.pot

2.网络扫描——NMAP

端口扫描类安全评测工具

```
[root@localhost run]# yum clean all && yum makecache fast
[root@localhost run]# yum -y install nmap
```

扫描目标：主机名、IP地址、网络地址　　多个目标用空格来分

选项：

-p　：指定扫描端口

-n：禁用反向解析（加快扫描）

扫描类型：

-sV ：扫描服务软件版本信息

```
[root@localhost run]# nmap 127.0.0.1
[root@localhost run]# nmap -sT 127.0.0.1
[root@localhost run]# nmap -sU 127.0.0.1
```

扫描其他主机

```
[root@localhost run]# nmap -n -p 22 192.168.200.104
[root@localhost run]# nmap -sV -n -p 22 192.168.200.104
Starting Nmap 6.40 ( http://nmap.org ) at 2020-03-07 00:08 CST
Nmap scan report for 192.168.200.104
Host is up (0.00040s latency).
PORT      STATE                SERVICE VERSION
22/tcp       open  ssh            OpenSSH 7.4 (protocol 2.0)
MAC Address: 00:0C:29:3E:78:87 (VMware)
Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

扫描200网段中哪些主机是活的

```
[root@localhost run]# nmap -n -sP 192.168.200.0
```

检测192.168.200.100-110的主机是否开启文件共享服务

```
[root@localhost run]# nmap -p 139,445 192.168.200.100-110
```

入侵前了解情况的方法，系统运维人员检查的工具