**12\_security03Linux基本防护+用户切换与提权**

**+SSH访问控制+selinux安全防护**

**一 Linux基本防护-用户账号安全**

**1.1 设置账号有效期**

**使用change工具**

chage -d yyyy-mm-dd 用户名 #设置密码最近一次修改的时间

-E yyyy-mm-dd,指定失效日期(-1取消)

chage命令的语法格式：

chage –l 账户名称 #查看账户信息

chage –E 时间 账户名称 #修改账户有效期

**案例: 失效的用户将无法登录**

使用chage命令将用户zhangsan的账户设为当前已失效（比如已经过去的某个时间）：

proxy ~]# useradd zhangsan

proxy ~]# chage -E 2019-12-31 zhangsan

尝试以用户zhangsan重新登录，输入正确的用户名、密码后直接闪退，返回登录页，说明此帐号已失效。

重设用户zhangsan的属性，将失效时间设为2019-12-31

proxy ~]# chage -E 2019-12-31 zhangsan #修改失效日期

proxy ~]# chage -l zhangsan #查看账户年龄信息

Last password change : May 15, 2017

Password expires : never

Password inactive : never

Account expires : Dec 31, 2019

Minimum number of days between password change : 0

Maximum number of days between password change : 99999

Number of days of warning before password expires : 7

**1.2 定义用户默认有效期（扩展知识）**

**/etc/login.defs**这个配置文件，决定了账户密码的默认有效期。

[root@proxy ~]# cat /etc/login.defs

PASS\_MAX\_DAYS 99999 #密码最长有效期

PASS\_MIN\_DAYS 0 #密码最短有效期

PASS\_MIN\_LEN 5 #密码最短长度

PASS\_WARN\_AGE 7 #密码过期前几天提示警告信息

UID\_MIN 1000 #UID最小值

UID\_MAX 60000 #UID最大值

**1.3 账户的锁定/解锁**

**passwd -S 用户名 #查看用户的密码状态**

**passwd -l 用户名 #锁定用户,使用户不能登录**

**passwd -u 用户名 #解锁用户,使用户能登录**

使用passwd或usermod命令将用户zhangsan的账户锁定。

proxy ~]# passwd -l zhangsan #锁定用户账号lock

锁定用户 zhangsan 的密码。

passwd: 操作成功

proxy ~]# passwd -S zhangsan #查看状态status

zhangsan LK 2018-02-22 0 99999 7 -1 (密码已被锁定。)

解除对用户zhangsan的锁定

proxy ~]# passwd -u zhangsan #解锁用户账号

解锁用户 zhangsan 的密码 。

passwd: 操作成功

proxy ~]# passwd -S zhangsan #查看状态

zhangsan PS 2018-08-14 0 99999 7 -1 (密码已设置，使用 SHA512 加密。)

**1.4 强制定期修改密码**

添加用户时的默认配置文件**/etc/login.defs**,一般不修改此文件

主要控制属性 PASS\_MAX\_DAYS PASS\_MIN\_DAYS PASS\_WARN\_AGE

修改PASS\_MAX\_DAYS的值,即可调整密码的有效期,达到定期修改密码的目的

**1.5 伪装登录提示信息**

修改配置文件:

/etc/issue: 配置本地登录后的系统提示信息

/etc/issue.net: 配置远程登录后的系统提示信息

**二 Linux基本防护-文件系统安全**

**2.1 程序和服务控制**

禁用非必要的系统服务

使用systemctl(红帽7)\chkconfig(红帽6)工具

systemctl start httpd chkconfig httpd start

systemctl enable httpd chkconfig httpd on

systemctl disable httpd chkconfig httpd off

**2.2 锁定/解锁保护文件**

**2.2.1 ext3/ext4的文件属性控制**

**命令: chattr 控制方式 属性 文件名**

**lsattr 文件名**

**2.2.2 + - =控制方式**

**2.2.3 属性i: 不可变(immutable[ɪˈmjuːtəbl])**

**属性a: 仅可追加(append only)**

zbserver ~]# lsattr /etc/passwd

---------------- /etc/passwd

zbserver ~]# chattr +a /etc/passwd

zbserver ~]# lsattr /etc/passwd

-----a---------- /etc/passwd

zbserver ~]# chattr -a /etc/passwd

zbserver ~]# lsattr /etc/passwd

---------------- /etc/passwd

**三 用户切换与提权**

用户切换与提权的应用场景

切换用户身份,when?

**SSH远程管理,;运维测试**

提升执行权限,when?

**管理权限细分**

**3.1 su切换的基本用法 [Substitute User,换人]**

快速切换为指定的其他用户,

普通用户执行时,需要验证目标用户的口令,

root执行时,无需验证口令.

**3.1.1 命令格式:**

**su [-] [目标用户]**

**su [-] [目标用户] -c “命令”**

**不指定目标用户时,默认视为root用户**

**切换用户时,带-表示连同系统环境一起切换,不带-表示仍在当前用户的环境**

**3.1.2 su切换的使用情况**

**安全日志/var/log/secure**:记录su验证,shell开启与关闭

**3.2 sudo提升执行权限**

**提权:让普通用户登录系统后,有执行root用户命令的权限**

**主配置文件 /etc/sudoers**

**3.2.1 sudo**

**Super or another Do,超级执行**

**管理员预先为用户设置执行许可**

**被授权用户有执行授权的命令,验证自己的口令**

**执行提权命令:sudo 提权命令**

**查看提权命令:sudo -l**

**3.2.2 配置sudo提权**

**3.2.2.1 修改方法**

**推荐:visudo**

**其他: vim /etc/sudoers**

**3.2.2.2 授权记录格式**

**用户 主机列表=命令列表**

**92 root ALL=(ALL) ALL**

**%开头表示组名 ALL=(ALL),目标身份,省略时表示root**

**wheel组的用户无需验证可执行所有命令**

**zbserver ~]# visudo**

**104 #%wheel ALL=(ALL) ALL #默认未开启**

**107 # %wheel ALL=(ALL) NOPASSWD: ALL #默认未开启**

**3.2.2.3 修改全局配置,启用日志**

**zbserver ~]# vim /etc/sudoers**

**118 Defaults logfile="/var/log/sudo"**

**案例 修改/etc/sudoers配置**

修改/etc/sudoers可以直接使用vim编辑该文件，或使用visudo命令修改该文件。

为softadm授予相关脚本的执行权限，允许通过systemctl工具来管理系统服务。

如果没有softadm账户可以先创建该账户。

proxy ~]# useradd softadm

proxy ~]# vim /etc/sudoers #修改文件后，需要使用wq!强制保存

.. ..

softadm ALL=(ALL) /usr/bin/systemctl

#授权softadm以root身份执行systemctl命令（ALL包括root）

切换为softadm用户，并验证sudo执行权限

proxy ~]# su - softadm

proxy ~]$ sudo -l

… …

[sudo] password for softadm: #输入softadm的口令

.. ..

用户 softadm 可以在该主机上运行以下命令：

(ALL) /usr/bin/systemctl

[softadm@proxy ~]$ systemctl start httpd #不用sudo时启动服务失败

Authentication is required

.. ..

[softadm@proxy ~]$ sudo systemctl restart httpd

#通过sudo启动服务成功

**案例 允许用户useradm通过sudo方式添加/删除/修改除root以外的用户账号**

**1）修改/etc/sudoers配置**

为useradm授予用户管理相关命令的执行权限，例外程序以!符号取反，放在后面。在执行相关程序时，可以利用通配符\*。

proxy ~]# useradd useradm

@proxy ~]# vim /etc/sudoers

.. ..

useradm ALL=(ALL) /usr/bin/passwd,!/usr/bin/passwd

root,/usr/sbin/user\*,!/usr/sbin/user\* \* root

**2）切换为useradm用户，验证sudo权限**

可以通过sudo方式来添加/删除/修改普通用户：

[useradm@proxy ~]$ sudo -l

(root) /usr/bin/passwd, !/usr/bin/passwd root, /usr/sbin/user\*,

!/usr/sbin/user\* \* root

[useradm@proxy ~]$ sudo useradd newuser01 #可以添加用户

[useradm@proxy ~]$ sudo passwd newuser01 #可以修改普通用户的口令

但是不能修改root用户的密码：

[useradm@proxy ~]$ sudo passwd root

对不起，用户 useradm 无权以 root 的身份在 localhost

上执行 /usr/bin/passwd root。

**3)允许wheel组成员以特权执行所有命令**

此案例用来展示sudo的便利性及设置不当带来的危险性，生产环境下慎用。

实现时参考下列操作(如果没有普通用户则先创建该账户)：

proxy ~]# vim /etc/sudoers

.. ..

104 %wheel ALL=(ALL) ALL #解除此行注释

proxy ~]# usermod -a -G wheel zengye

[zengye@proxy ~]$ sudo -l

用户 zengye 可以在该主机上运行以下命令：

(root) /bin/\*

**3.3 sudo别名设置**

**别名分类:用户别名,主机别名,命令别名**

**主要用途:提高可用性\易读性,简化配置,使记录更有条例**

**别名名称必须使用大写字母**

User\_Alias OPERATORS=

Host\_Alias MAILSERVERS=

Cmnd\_Alisas SOFTMGR=

zbserver ~]# **vim /etc/sudoers**

93 User\_Alias MYUSER=yaya,jing,plj,mac **#添加此4行**

94 Host\_Alias MYSER=localhost,zbserver

95 Cmnd\_Alias MYCMD=/bin /systemctl \* httpd,/bin/vim/e

tc/httpd/conf/httpd.conf

96 Cmnd\_Alias MYSOFT=/bin/rpm,/bin/yum,/sbin/fdisk /dev/vda

**97 MYUSER MYSER=MYCMD,MYSOFT #进行提权**

**98 bob MYSER=MYSOFT #进行提权**

root ALL=(ALL) ALL

第一个 ALL 指允许从任何终端、机器访问 sudo

第二个 (ALL) 指sudo 命令被允许以任何用户身份执行

第三个 ALL 表示所有命令都可以作为 root 执行

**四 SSH访问控制-SSH基本防护**

**4.1 SSH防护概述**

**4.1.1 存在安全隐患**

密码嗅探,键盘记录 暴力枚举账号,猜解密码

**4.1.2 常见的防护措施**

用户限制,黑白名单

更改验证方式(密码-->密匙对)

防火墙

**4.2 sshd基本安全配置**

**配置文件 /etc/ssh/sshd\_config**

Protocol 2 #SSH协议,添加此行

17 Port 3389 #修改ssh默认端口,默认为22

38 PermitRootLogin no #禁止root用户登录

64 PermitEmptyPasswords no #禁止密码为空的用户登录

115 UseDNS no #不解析客户机地址

37 LoginGraceTime 1m #登录限时

40 MaxAuthTries 3 #每连接最多认证次数

zbserver ~]# systemctl restart sshd #重启sshd服务

**测试**

room9pc01 ~]$ ssh root@192.168.2.5 -p 3389

root@192.168.2.5's password:

Permission denied, please try again. #禁止了root用户登录

zbserver ~]# useradd lisi #添加用户lisi不设置密码

room9pc01 ~]$ ssh lisi@192.168.2.5 -p 3389

lisi@192.168.2.5's password:

Permission denied, please try again. #不允许无密码用户登录

zbserver ~]# echo 123456 | passwd --stdin lisi

room9pc01 ~]$ ssh lisi@192.168.2.5 -p 3389

lisi@192.168.2.5's password: #输入密码后能成功登录

**4.3 sshd黑/白名单配置(同时都有配置时,优先执行黑名单)**

黑名单(deny):不允许使用名单里的用户连接ssh服务

白名单(allow):仅允许使用名单里的用户连接ssh服务

AllowUsers USER1 USER2 ...

DenyUsers USER1 USER2 ...

AllowGroups GROUP1 GROUP2 ...

DenyGroups GROUP1 GROUP2 ...

**案例:针对SSH访问采用仅允许的策略，未明确列出的用户一概拒绝登录**

**1）调整sshd服务配置，添加AllowUsers策略，仅允许用户lisi，其中lisi只能从网段192.168.2.5登录。**

注意：如果没有这些用户，需要提前创建用户并设置密码。

proxy ~]# vim /etc/ssh/sshd\_config

.. ..文件末尾添加以下4行

AllowUsers [lisi@192.168.2.5](mailto:lisi@192.168.2.5) #定义账户白名单

##DenyUsers USER1 USER2 #定义账户黑名单

##DenyGroups GROUP1 GROUP2 #定义组黑名单

##AllowGroups GROUP1 GROUP2 #定义组白名单

proxy ~]# systemctl restart sshd

**2）验证SSH访问控制，未授权的用户将拒绝登录。**

proxy ~]# ssh [lisi@192.168.2.5](mailto:lisi@192.168.2.5) -p 3389

lisi@192.168.2.5's password:

proxy ~]$ exit

proxy ~]# ssh [wangwu@192.168.2.5](mailto:wangwu@192.168.2.5) #未授权的用户被拒绝登录

root@192.168.4.5's password:

Permission denied, please try again.

**五 SSH访问控制-SSH密匙对验证**

**5.1 sshd验证方式控制**

**5.1.1 口令验证**:检查登录用户的口令(密码)是否一致

**5.1.2 密钥验证**:检查客户端私钥与服务器上的公钥是否匹配

**5.2 配置SSH密匙对验证**

**5.2.1 SSH服务器修改sshd配置文件,开启需要密码认证**

zbserver ~]# vim /etc/ssh/sshd\_config

66 PasswordAuthentication yes

**5.2.2 客户机生成ssh密钥对**

web1 ~]# ssh-keygen -f /root/.ssh/id\_rsa -N '' #客户机生成密钥

web1 ~]# ls -lh /root/.ssh/id\_rsa\*

-rw------- 1 root root 1.7K Sep 21 18:15 /root/.ssh/id\_rsa

-rw-r--r-- 1 root root 395 Sep 21 18:15 /root/.ssh/id\_rsa.pub

#其中.pub是公钥文件

**5.2.3 客户机传输公钥给ssh服务器,ssh服务器上查看文件确认收到公钥**

web1 ~]# ssh-copy-id [root@192.168.2.5](mailto:root@192.168.2.100) #传输公钥

zbserver ~]# vim /root/.ssh/authorized\_keys #2.5上确认收到公钥

web1 ~]# ssh [root@192.168.2.5](mailto:root@192.168.2.5) #客户机100ssh方式免密码登录5

**5.2.4** **SSH服务器修改sshd配置文件,关闭需要密码认证**

zbserver ~]# vim /etc/ssh/sshd\_config

66 PasswordAuthentication no

**六 SELinux安全防护-SElinux概述**

**6.1 什么是SELinux**

Security-Ehanced Linux

一套强化linux安全的扩展模块,美国国家安全局主导开发

SELinux的运作机制

集成到Linux内核(2.6及以上),操作系统提供可定制的策略和管理工具

**6.2 红帽的SELinux策略集**

**配置文件/etc/selinux/config**

SELINUX=enforcing/permissive/disabled

SELINUXTYPE=targeted/minimun/mls

其中,

SELINUXTYPE=targeted为推荐设置,仅保护最常见的/关键的网络服务,其他不限制

**设置selinux=enforcing后,需要在根下创建文件.autorelabel**

**6.3 SELinux控制**

6.3.1 开关控制,命令行使用setenforce命令设置

6.3.2 模式控制,修改配置文件/etc/selinux/config内SELINUX的值

**七 SELinux安全防护-SElinux策略设置**

**7.1 查看安全上下文**

Security Context: 为文件/目录/设备标记访问控制属性

属性构成:

用户:角色:访问类型:选项...

查看命令: ls -Z[d] /路径/文件[目录]

**7.2 修改安全上下文**

使用chcon工具: -t 指定访问类型;-R 递归修改

一般操作规律:移动的文件,原有的上下文属性不变;复制的文件,自动继承目标位置的上下文

**7.3 重置安全上下文**

7.3.1 使用restorecon工具:恢复为所在位置的默认上下文属性,-R 递归修改

7.3.2 /.autorelabel文件:下次重启后全部重置

**7.4 调整SELinux布尔值**

sebool值,服务功能的开关

7.4.1 使用getsebool查看

getsebool -a 列出所有布尔值

7.4.2 使用setsebool设置,-P永久修改

setsebool -P 选项 值 值为on或off

**案例:SELinux安全防护**

4.1 问题

本案例要求熟悉SELinux防护机制的开关及策略配置，完成以下任务：

将Linux服务器的SELinux设为enforcing强制模式

从/root目录下移动一个包文件到FTP下载目录，调整策略使其能够被下载

4.2 步骤

**步骤一：将Linux服务器的SELinux设为enforcing强制模式**

**1）固定配置：修改/etc/selinux/config文件**

确认或修改SELINUX为enforcing模式：

[root@proxy ~]# vim /etc/selinux/config

SELINUX=enforcing #设置SELinux为强制模式

SELINUXTYPE=targeted #保护策略为保护主要的网络服务安全

**[root@proxy ~]# touch /.autorelabel**

**2）临时配置：使用setenforce命令**

查看当前SELinux状态，如果是disabled则需要根据第1）步的配置重启系统；如果是permissive则使用setenforce命令修改为enforcing即可：

[root@proxy ~]# getenforce #查看当前状态为警告模式

Permissive

[root@proxy ~]# setenforce 1 #设置SELinux为强制模式

[root@proxy ~]# getenforce #查看当前模式为强制模式

Enforcing

[root@proxy ~]# setenforce 0 #设置SELinux为强制模式

[root@proxy ~]# getenforce #查看当前模式为警告模式

Permissive

**步骤二：在SELinux启用状态下，调整策略打开vsftpd服务的匿名上传访问**

**1）配置一个允许匿名上传的vsftpd服务作为测试环境**

[root@proxy ~]# setenforce 1

[root@proxy ~]# yum -y install vsftpd

.. ..

[root@proxy ~]# vim /etc/vsftpd/vsftpd.conf

anonymous\_enable=YES #开启匿名访问

anon\_upload\_enable=YES #允许上传文件

anon\_mkdir\_write\_enable=YES #允许上传目录

[root@proxy ~]# systemctl start vsftpd #启动服务

#默认Vsftpd共享目录为/var/ftp/

**步骤三：从/root目录下移动2个包文件到FTP下载目录，调整文件的安全上下文**

**1）建立两个FTP下载用的测试文件**

由root用户创建两个测试压缩包，一个直接建立到/var/ftp/目录下，另一个先在/root/下建立，然后移动至/var/ftp/目录。

#测试文件1，直接在ftp目录下创建文件

[root@proxy ~]# tar -czf /var/ftp/log1.tar /var/log

[root@proxy ~]# ls -lh /var/ftp/

-rw-r--r--. 1 root root 8M 8月 16 10:16 log1.tar

[root@proxy ~]# ls -Z /var/ftp/

-rw-r--r--. root root unconfined\_u:object\_r:public\_content\_t:s0 log1.tar

#测试文件2，在/root下建立，然后移动至/var/ftp目录

[root@proxy ~]# tar -czf log2.tar /var/log

[root@proxy ~]# mv log2.tar /var/ftp/

[root@proxy ~]# ls -lh /var/ftp/

-rw-r--r--. 1 root root 8M 8月 16 10:16 log2.tar

[root@proxy ~]# ls -Z /var/ftp/

-rw-r--r--. 1 root root unconfined\_u:object\_r:admin\_home\_t:s0 log2.tar

**3）通过FTP方式测试下载**

**使用wget命令分别下载这两个包文件，第二个包将会下载失败（看不到文件）。**

[root@proxy ~]# wget <ftp://192.168.4.5/log1.tar> #下载第一个文件，成功

[root@proxy ~]# wget <ftp://192.168.4.5/log2.tar> #下载第二个文件，失败

**4）检查该测试包的安全上下文，正确调整后再次下载第二个包成功。**

**文件已经存放到共享目录下，但客户端无法访问下载，是因为被SELinux拦截了！**

[root@proxy ~]# **ls -Z /var/ftp/**

-rw-r--r--. root root unconfined\_u:object\_r:public\_content\_t:s0 log1.tar

-rw-r--r--. 1 root root unconfined\_u:object\_r:admin\_home\_t:s0 log2.tar

[root@proxy ~]# **chcon -t public\_content\_t** /var/ftp/log2.tar.gz

[root@proxy ~]# ls -Z /var/ftp/log2.tar

-rw-r--r--. root root unconfined\_u:object\_r:public\_content\_t:s0 log2.tar

[root@proxy ~]# wget <ftp://192.168.4.5/log2.tar> #再次下载，成功

注意：上例中的chcon操作可替换为（效果相同）：

# **restorecon** /var/ftp/log2.tar.gz 或者

# **chcon --reference**=/var/ftp/log1.tar.gz /var/ftp/log2.tar.gz