

NSD SECURITY DAY01

1. [案例1：Linux基本防护措施](#)
2. [案例2：使用sudo分配管理权限](#)
3. [案例3：提高SSH服务安全](#)
4. [案例4：SELinux安全防护](#)

1 案例1：Linux基本防护措施

1.1 问题

本案例要求练习Linux系统的基本防护措施，完成以下任务：

1. 修改用户zhangsan的账号属性，设置为2019-12-31日失效（禁止登录）
2. 临时锁定用户lisi的账户，使其无法登录，验证效果后解除锁定
3. 修改tty终端提示，使得登录前看到的第一行文本为“Windows Server 2012 Enterprise R2”，第二行文本为“NT 6.2 Hybrid”
4. 锁定文件/etc/resolv.conf、/etc/hosts，以防止其内容被无意中修改

1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：修改用户zhangsan的账户属性，设置为2019-12-31日失效（禁止登录）

1) 正常情况下，未过期的账号可以正常登录，使用chage可以修改账户有效期。

01. chage命令的语法格式：
02. chage - l 账户名称 //查看账户信息
03. chage - E 时间 账户名称 //修改账户有效期

[Top](#)

2) 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）：

```
01. [root@proxy ~] # useradd zhangsan
02. [root@proxy ~] # chage -E 2015-05-15 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。

3) 重设用户zhangsan的属性，将失效时间设为2019-12-31

```
01. [root@proxy ~] # chage -E 2019-12-31 zhangsan           //修改失效日期
02. [root@proxy ~] # chage -l zhangsan                     //查看账户年龄信息
03. Last password change           : May 15, 2017
04. Password expires               : never
05. Password inactive              : never
06. Account expires                : Dec 31, 2019
07. Minimum number of days between password change        : 0
08. Maximum number of days between password change        : 99999
09. Number of days of warning before password expires     : 7
```

4) 定义默认有效期（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。

```
01. [root@proxy ~] # cat /etc/login.defs
02. PASS_MAX_DAYS 99999           //密码最长有效期
03. PASS_MIN_DAYS 0               //密码最短有效期
04. PASS_MIN_LEN 5                //密码最短长度
```

[Top](#)

05.	PASS_WARN_AGE	7	//密码过期前几天提示警告信息
06.	UID_MIN	1000	//UID最小值
07.	UID_MAX	60000	//UID最大值

步骤二：临时锁定用户zhangsan的账户，使其无法登录，验证效果后解除锁定

1) 锁定用户账号

使用passwd或usermod命令将用户zhangsan的账户锁定。

```
01. [root@proxy ~] # passwd -l zhangsan //锁定用户账号lock
02. 锁定用户 zhangsan 的密码。
03. passwd: 操作成功
04.
05. [root@proxy ~] # passwd -S zhangsan //查看状态status
06. zhangsan LK 2018-02-22 0 99999 7 -1 (密码已被锁定。)
```

2) 验证用户zhangsan已无法登录，说明锁定生效

输入正确的用户名、密码，始终提示“Login incorrect”，无法登录。

3) 解除对用户zhangsan的锁定

```
01. [root@proxy ~] # passwd -u zhangsan //解锁用户账号
02. 解锁用户 zhangsan 的密码。
03. passwd: 操作成功
04.
05. [root@proxy ~] # passwd -S zhangsan //查看状态
06. zhangsan PS 2018-08-14 0 99999 7 -1 (密码已设置，使用 SHA512 加密。)
```

[Top](#)

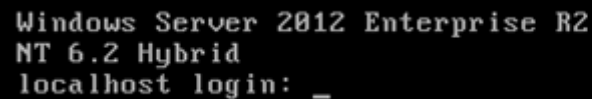
步骤三：修改tty登录的提示信息，隐藏系统版本

1) 账户在登录Linux系统时，默认会显示登陆信息（包括操作系统内核信息）
/etc/issue这个配置文件里保存的就是这些登陆信息，修改该文件防止内核信息泄露。

```
01. [ root@proxy ~] # cat /etc/issue //确认原始文件
02. Red Hat Enterprise Linux Server release 6.5 ( Santiago)
03. Kernel \r on an \m
04.
05. [ root@proxy ~] # cp /etc/issue /etc/issue.origin //备份文件
06.
07. [ root@proxy ~] # vim /etc/issue //修改文件内容
08. Windows Server 2012 Enterprise R2
09. NT 6.2 Hybrid
```

2) 测试版本伪装效果

退出已登录的tty终端，或者重启Linux系统，刷新后的终端提示信息会变成自定义的文本内容，如图-1所示。



```
Windows Server 2012 Enterprise R2
NT 6.2 Hybrid
localhost login: _
```

图-1

附加：对于操作系统来说，文件系统也可以通过添加额外属性来提高性能与安全性。

```
01. [ root@proxy ~] # cat /etc/fstab
02. /dev/vda1 /boot xfs defaults,noexec 0 0
```

[Top](#)

- 03. `/dev/vda3 /home xfs defaults,noatime 0 0`
- 04. 备注：
- 05. noexec属性可以让分区下的所有程序都不可执行，包括病毒与木马
- 06. noatime让分区下的所有文件都不再更新atime时间，atime时间为文件的访问时间

步骤四：锁定文件/etc/resolv.conf、/etc/hosts

1) 语法格式：

- 01. `# chattr +i 文件名` //锁定文件（无法修改、删除等）
- 02. `# chattr -i 文件名` //解锁文件
- 03. `# chattr +a 文件名` //锁定后文件仅可追加
- 04. `# chattr -a 文件名` //解锁文件
- 05. `# lsattr 文件名` //查看文件特殊属性

2) 使用+i锁定文件，使用lsattr查看属性

- 01. `[root@proxy ~] # chattr +i /etc/resolv.conf`
- 02. `[root@proxy ~] # lsattr /etc/resolv.conf`
- 03. `-----i----- /etc/resolv.conf`

3) 使用+a锁定文件(仅可追加)，使用lsattr查看属性

- 01. `[root@proxy ~] # chattr +a /etc/hosts`
- 02. `[root@proxy ~] # lsattr /etc/hosts`

[Top](#)

```
03.  ----- a ----- /etc/hosts
```

4) 测试文件锁定效果

```
01. [root@proxy ~]# rm -rf /etc/resolv.conf
02. rm: 无法删除"/etc/resolv.conf": 不允许的操作
03. [root@proxy ~]# echo xyz > /etc/resolv.conf
04. -bash: resolv.conf: 权限不够
05.
06.
07. [root@proxy ~]# rm -rf /etc/hosts //失败
08. [root@proxy ~]# echo "192.168.4.1 xyz" > /etc/hosts //失败
09. [root@proxy ~]# echo "192.168.4.1 xyz" >> /etc/hosts //成功
```

5) 恢复这两个文件原有的属性 (避免对后续实验造成影响)

```
01. [root@proxy ~]# chattr -i /etc/resolv.conf
02. [root@proxy ~]# chattr -i /etc/hosts
03. [root@proxy ~]# lsattr /etc/resolv.conf /etc/hosts
04. ----- /etc/resolv.conf
05. ----- /etc/hosts
```

2 案例2：使用sudo分配管理权限

[Top](#)

2.1 问题

本案例要求利用sudo机制分配管理操作权限，主要完成以下任务：

1. 使用su命令临时切换账户身份，并执行命令
2. 允许softadm管理系统服务的权限
3. 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
4. 允许wheel组成员以特权执行所有命令
5. 为sudo机制启用日志记录，以便跟踪sudo执行操作

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：使用su命令临时切换账户身份，并以root执行命令

su(Substitute User)命令可以快速切换账户身份，普通用户切换账户身份时需要输入密码，root使用su命令切换任何身份都不需要密码，如法格式如下：

```
01. # su - [账户名称]
02. # su - [账户名称] -c '命令'
```

1)从普通用户切换为root账户身份(如果没有普通账户则需要先创建)

```
01. [zhangsan@proxy ~]# whoami
02. zhangsan
03. [zhangsan@proxy ~]# su -           //切换账户，默认切换为root账户
04. 密码:                             //输入root的密码
05. [root@proxy ~]# whoami             //确认结果
06. root
```

[Top](#)

2)以普通身份创建文件(如果没有普通账户则需要先创建)，以root身份重启服务

```
01. [ root@proxy ~] # su - zhangsan - c "touch /tmp/test.txt" //管理员切换普通用户
02. [ root@proxy ~] # ll /tmp/test.txt
03.
04.
05. [ zhangsan@proxy ~] # su - - c "systemctl restart sshd" //以管理员重启服务
06. 密码 :
07. ● sshd.service - OpenSSH server daemon
08. Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
09. active: active (running) since 五 2018-01-19 08:59:40 CST; 1 months 4 days ago
```

步骤二：允许softadm管理系统服务的权限

1) 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。

```
01. [ root@proxy ~] # useradd softadm
02. [ root@proxy ~] # vim /etc/sudoers //修改文件后，需要使用wq强制保存
03. ...
04. softadm ALL=(ALL) /usr/bin/systemctl
05. //授权softadm以root身份执行systemctl命令 (ALL包括root)
```

2) 切换为softadm用户，并验证sudo执行权限

[Top](#)


```

01. [ root@proxy ~] # su - softadm
02. [ softadm@proxy ~] $ sudo -l
03. ... ..
04. [ sudo] password for softadm:           //输入softadm的口令
05. ... ..
06. 用户 softadm 可以在该主机上运行以下命令：
07.    ( ALL) /usr/bin/systemctl
08.
09. [ softadm@proxy ~] $ systemctl start httpd           //不用sudo时启动服务失败
10. Authentication is required
11. ... ..
12. [ softadm@proxy ~] $ sudo systemctl restart httpd    //通过sudo启动服务成功

```

步骤三：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*。

```

01. [ root@proxy ~] # useradd useradm
02. [ root@proxy ~] # vim /etc/sudoers
03. ... ..
04. useradm ALL=( ALL) /usr/bin/passwd,!/usr/bin/passwd root,/usr/sbin/user*,
05.    !/usr/sbin/user* * root

```

[Top](#)

2) 切换为用户useradm，验证sudo权限

可以通过sudo方式来添加/删除/修改普通用户：

01. [useradm@proxy ~] \$ sudo -l
02. ...
03. 用户useradm可以在该主机上运行以下命令：
04. (root) /usr/bin/passwd, !/usr/bin/passwd root, /usr/sbin/user*,
05. !/usr/sbin/user* * root
06. [useradm@proxy ~] \$ sudo useradd newuser01 //可以添加用户
07. [useradm@proxy ~] \$ sudo passwd newuser01 //可以修改普通用户的口令
08. 更改用户 newuser01 的密码。
09. 新的 密码：
10. 重新输入新的 密码：
11. passwd：所有的身份验证令牌已经成功更新。

但是不能修改root用户的密码：

01. [useradm@proxy ~] \$ sudo passwd root
02. 对不起，用户 useradm 无权以 root 的身份在 localhost 上
03. 执行 /usr/bin/passwd root。

步骤四：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)：

01. [root@proxy ~] # vim /etc/sudoers
02. ...

[Top](#)

```
03.  %wheel ALL=( ALL)  ALL
04.  [ root@proxy ~] # usermod - a - G wheel zengye
05.  [ zengye@proxy ~] $ sudo - l
06.  ...
07.  用户 zengye 可以在该主机上运行以下命令：
08.  ( root) /bin/*
```

步骤五：为sudo机制启用日志记录，以便跟踪sudo执行操作

1) 修改/etc/sudoers配置，添加日志设置

```
01.  [ root@proxy ~] # visudo
02.  Defaults logfile="/var/log/sudo"
03.  ...
```

2) 以root (默认有所有权限) 执行sudo操作

```
01.  [ root@proxy ~] # sudo - l //查看授权的sudo操作
02.  [ softadm@proxy ~] # sudo systemctl status httpd //查看授权的sudo操作
```

3) 确认日志记录已生效

```
01.  [ root@proxy ~] # tail /var/log/sudo
02.  ...
03.  May 16 22:14:49: root : TTY=pts/1; PWD=/root ; USER=root ; COMMAND=list
```

[Top](#)

```
04. Feb 22 22:35:43 : softadm : TTY=pts/11 ; PWD=/home/softadm ; USER=root ;
05.      COMMAND=/bin/systemctl status httpd
```

3 案例3：提高SSH服务安全

3.1 问题

本案例要求提高Linux主机上SSH服务端的安全性，完成以下任务：

1. 配置基本安全策略（禁止root、禁止空口令）
2. 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
3. 实现密钥验证登录（私钥口令）、免密码登入
4. 确认密钥验证使用正常后，禁用口令验证

3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置基本安全策略

1) 调整sshd服务配置，并重载服务

```
01. [ root@proxy ~] # vim /etc/ssh/sshd_config
02. ...
03. Protocol 2 //SSH协议
04. PermitRootLogin no //禁止root用户登录
05. PermitEmpty Passwords no //禁止密码为空的用户登录
06. UseDNS no //不解析客户机地址
07. LoginGraceTime 1m //登录限时
08. MaxAuthTries 3 //每连接最多认证次数
09. ...
10. [ root@proxy ~] # systemctl restart sshd
```

[Top](#)

2) 测试基本安全策略

尝试以root用户SSH登录，失败：

```
01. [root@proxy ~] # ssh root@192.168.4.5
02. root@192.168.4.5's password:
03. Permission denied, please try again.
```

将服务器上用户kate(如无该账户则先创建)的密码设为空，尝试SSH登录，也会失败：

```
01. [root@proxy ~] # passwd -d kate //清空用户口令
02. 清除用户的密码 kate。
03. passwd: 操作成功
04.
05. [root@proxy ~] # ssh kate@192.168.4.5
06. kate@192.168.4.5's password:
07. Permission denied, please try again.
```

步骤二：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zhangsan、tom、useradm，其中useradm只能从网段192.168.4.0/24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。

```
01. [root@proxy ~] # vim /etc/ssh/sshd_config
02. ...
```

[Top](#)

```
03. AllowUsers zhangsan tom useradm@192.168.4.0/24 //定义账户白名单
04. ##Deny Users USER1 USER2 //定义账户黑名单
05. ##Deny Groups GROUP1 GROUP2 //定义组黑名单
06. ##AllowGroups GROUP1 GROUP2 //定义组白名单
07. [ root@proxy ~] # systemctl restart sshd
```

2) 验证SSH访问控制，未授权的用户将拒绝登录。

```
01. [ root@proxy ~] # ssh useradm@192.168.4.5 //已授权的用户允许登录
02. useradm@192.168.4.5's password:
03. [ useradm@proxy ~] $ exit
04. [ root@proxy ~] # ssh root@192.168.4.5 //未授权的用户被拒绝登录
05. root@192.168.4.5's password:
06. Permission denied, please try again.
```

步骤三：实现密钥对验证登录（私钥口令）、免密码登入

1) 准备客户机测试环境

为客户机的用户root建立SSH密钥对

使用ssh-keygen创建密钥对，将私钥口令设为空（直接回车）：

```
01. [ root@client ~] $ ssh-keygen
02. Generating public/private rsa key pair.
03. Enter file in which to save the key ( /root/.ssh/id_rsa ):
04. Created directory '/root/.ssh'.
05. Enter passphrase ( empty for no passphrase ): //直接回车将口令设为空
```

[Top](#)

```

06. Enter same passphrase again: //再次回车确认
07. Your identification has been saved in /root/.ssh/id_rsa.
08. Your public key has been saved in /root/.ssh/id_rsa.pub.
09. The key fingerprint is:
10. 63: 6e: cf: 45: f0: 56: e2: 89: 6f: 62: 64: 5a: 5e: fd: 68: d2
11. The key's randomart image is:
12. +- [ RSA 2048]-----+
13. |          |
14. |          |
15. |      . . . |
16. |      ==  |
17. |    S = B . |
18. |    o B = . o |
19. |    ++ = E . |
20. |    . ++ o  |
21. |    o      |
22. +-----+
23. [ root@client ~] $ ls -lh ~/.ssh/id_rsa* //确认密钥对文件
24. -rw----- . 1 root root 1.8K 8月 15 10: 35 /root/.ssh/id_rsa
25. -rw- r-- r-- . 1 root root 403 8月 15 10: 35 /root/.ssh/id_rsa.pub

```

2) 将客户机上用户root的公钥部署到SSH服务器

以用户root登入客户机，使用ssh-copy-id命令将自己的公钥部署到服务器：

```

01. [ root@client ~] $ ssh-copy-id root@192.168.4.5
02. root@192.168.4.5's password:
03. Now try logging into the machine, with "ssh 'root@192.168.4.5'", and check in:

```

[Top](#)

04. `.ssh/authorized_keys`
05. `to make sure we haven't` added extra keys that you weren't expecting.

3) 在服务器上确认客户机用户root上传的公钥信息

默认部署位置为目标用户的家目录下 `~/.ssh/authorized_keys`文件：

01. `[root@proxy ~] # tail - 2 ~/.ssh/authorized_keys`
02. `ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAzz+5AiFmGQ7Lfuiv7eBnOcmRO9JRTcqRoy nG02y 5`
03. `RyFL+LxR1lpEbKnrUy IZDk5uaX1Y8rwsf +pa7UZ2Ny qmUEv NSUoOhQy DGsU9SPy A dzRCCv DgwpOFhaHi / OFnT +zqjAqXH2M9f FYEVUU4PIVL8HT 19zCQRVZ /`

4) 在客户机上测试SSH密钥对验证

在客户机用户root的环境中，以远程用户root登入192.168.4.5主机时，无需验证口令即可登入（因为私钥口令为空）：

01. `[root@client ~] $ ssh root@192.168.4.5` //免交互直接登入
02. `Last login: Thu Aug 15 10:48:09 2013 from 192.168.4.100`

步骤四：确认密钥验证使用正常后，禁用口令验证

1) 调整sshd服务配置，将PasswordAuthentication设为no

01. `[root@proxy ~] # vim /etc/ssh/sshd_config`
02. `...`
03. `PasswordAuthentication no` //将此行yes改成no
- 04.

[Top](#)


```
05. [root@proxy ~] # systemctl restart sshd
```

4 案例4：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：

1. 将Linux服务器的SELinux设为enforcing强制模式
2. 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

实现此案例需要按照如下步骤进行。

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：

```
01. [root@proxy ~] # vim /etc/selinux/config
02. SELINUX=enforcing           //设置SELinux为强制模式
03. SELINUXTYPE=targeted       //保护策略为保护主要的网络服务安全
```

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：

```
01. [root@proxy ~] # getenforce           //查看当前状态为警告模式
02. Permissive
03. [root@proxy ~] # setenforce 1         //设置SELinux为强制模式
```

[Top](#)

```
04. [ root@proxy ~] # getenforce           //查看当前模式为强制模式
05. Enforcing
06. [ root@proxy ~] # setenforce 0         //设置SELinux为强制模式
07. [ root@proxy ~] # getenforce           //查看当前模式为警告模式
08. 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

```
01. [ root@proxy ~] # setenforce 1
02. [ root@proxy ~] # yum -y install vsftpd
03. ...
04. [ root@proxy ~] # vim /etc/vsftpd/vsftpd.conf
05. anonym_ous_enable=YES           //开启匿名访问
06. anon_upload_enable=YES          //允许上传文件
07. anon_mkdir_write_enable=YES     //允许上传目录
08. [ root@proxy ~] # systemctl start vsftpd //启动服务
09. //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。

```
01. //测试文件1, 直接在ftp目录下创建文件
02. [ root@proxy ~] # tar -czf /var/ftp/log1.tar /var/log
```

[Top](#)

```
03. [root@proxy ~]# ls -lh /var/ftp/
04. -rw-r--r--. 1 root root 8M 8月 16 10:16 log1.tar
05. [root@proxy ~]# ls -Z /var/ftp/
06. -rw-r--r--. root root unconfined_u:object_r:public_content_t:s0 log1.tar
07.
08. //测试文件2,在/root下建立,然后移动至/var/ftp目录
09. [root@proxy ~]# tar -czf log2.tar /var/log
10. [root@proxy ~]# mv log2.tar /var/ftp/
11. [root@proxy ~]# ls -lh /var/ftp/
12. -rw-r--r--. 1 root root 8M 8月 16 10:16 log2.tar
13. [root@proxy ~]# ls -Z /var/ftp/
14. -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）。

```
01. [root@proxy ~]# wget ftp://192.168.4.5/log1.tar //下载第一个文件，成功
02.
03. [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！

```
01. [root@proxy ~]# ls -Z /var/ftp/
02. -rw-r--r--. root root unconfined_u:object_r:public_content_t:s0 log1.tar
```

[Top](#)

```
03. -rw-r--r--. 1 root root unconfined_u:object_r:admin_home_t:s0 log2.tar
04.
05. [ root@proxy ~] # chcon -t public_content_t /var/ftp/d2.tar.gz
06. [ root@proxy ~] # ls -Z /var/ftp/log2.tar
07. -rw-r--r--. root root unconfined_u:object_r:public_content_t:s0 log2.tar
08.
09. [ 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