TTS 10.0 COOKBOOK

(NSD SECURITY DAY03)

版本编号 10.0

2019-06

达内 IT 培训集团

NSD SECURITY DAY03

1. 案例 1：Linux 基本防护措施

- **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

- **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。

```
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
```

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)：

```
[root@proxy ~]# useradd zhangsan
[root@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。

3）重设用户 zhangsan 的属性，将失效时间设为 2019-12-31

```
[root@proxy ~]# chage -E 2019-12-31 zhangsan           //修改失效日期
[root@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
```

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。

```
[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
```

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。

```
[root@proxy ~]# passwd -l zhangsan                    //锁定用户账号 lock
锁定用户 zhangsan 的密码。
passwd: 操作成功

[root@proxy ~]# passwd -S zhangsan                    //查看状态 status
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

```
[root@proxy ~]# passwd -u zhangsan                    //解锁用户账号
解锁用户 zhangsan 的密码 。
passwd: 操作成功

[root@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。)
```

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。

```
[root@proxy ~]# cat /etc/issue                        //确认原始文件
Red Hat Enterprise Linux Server release 6.5 (Santiago)
Kernel \r on an \m

[root@proxy ~]# cp /etc/issue /etc/issue.origin       //备份文件

[root@proxy ~]# vim /etc/issue                        //修改文件内容
Windows Server 2012 Enterprise R2
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。

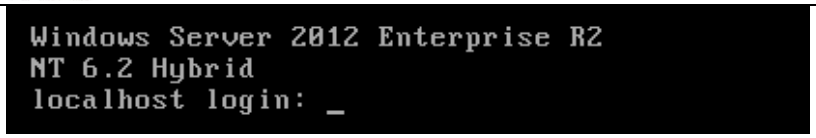

图-1

步骤四：锁定文件/etc/resolv.conf、/etc/hosts

1）语法格式：

```
# chattr +i  文件名                    //锁定文件（无法修改、删除等）
# chattr -i  文件名                    //解锁文件
# chattr +a  文件名                    //锁定后文件仅可追加
# chattr -a  文件名                    //解锁文件
# 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

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

3）使用+a 锁定文件(仅可追加)，使用 lsattr 查看属性

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

4）测试文件锁定效果

```
[root@proxy ~]# rm -rf /etc/resolv.conf
rm: 无法删除"/etc/resolv.conf": 不允许的操作
[root@proxy ~]# echo xyz > /etc/resolv.conf
-bash: resolv.conf: 权限不够


[root@proxy ~]# rm -rf  /etc/hosts                        //失败
[root@proxy ~]# echo "192.168.4.1  xyz" > /etc/hosts     //失败
[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）

```
[root@proxy ~]# chattr -i /etc/resolv.conf
[root@proxy ~]# chattr -i /etc/hosts
[root@proxy ~]# lsattr /etc/resolv.conf /etc/hosts
--------------- /etc/resolv.conf
--------------- /etc/hosts
```

2. 案例 2：使用 sudo 分配管理权限

- **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

- **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：

```
# su - [账户名称]
# 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)

```
[zhangsan@proxy ~]# whoami
zhangsan
[zhangsan@proxy ~]# su -                    //切换账户，默认切换为 root 账户
密码:                                       //输入 root 的密码
[root@proxy ~]# whoami                      //确认结果
root
```

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

```
[root@proxy ~]# su - zhangsan -c "touch /tmp/test.txt"     //管理员切换普通用户
[root@proxy ~]# ll /tmp/test.txt


[zhangsan@proxy ~]# su - -c "systemctl restart sshd"       //以管理员重启服务
密码:
● sshd.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset:
enabled)
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。

```
[root@proxy ~]# useradd softadm
[root@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）
```

2）切换为 softadm 用户，并验证 sudo 执行权限

```
[root@proxy ~]# su - softadm
[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
```

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*。

```
[root@proxy ~]# useradd useradm
[root@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
.. ..
用户 useradm 可以在该主机上运行以下命令：
    (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
更改用户 newuser01 的密码 。
新的 密码：
重新输入新的 密码：
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：

```
[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。
```

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)：

```
[root@proxy ~]# vim /etc/sudoers
.. ..
%wheel ALL=(ALL)  ALL
[root@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/*
```

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

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

2）以 root（默认有所有权限）执行 sudo 操作

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

3）确认日志记录已生效

```
[root@proxy ~]# tail /var/log/sudo
.. ..
May 16 22:14:49 : root : TTY=pts/1 ; PWD=/root ; USER=root ; COMMAND=list
Feb 22 22:35:43 : softadm : TTY=pts/11 ; PWD=/home/softadm ; USER=root ;
   COMMAND=/bin/systemctl status httpd
```

3. 案例 3：提高 SSH 服务安全

- **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

- **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

```
[root@proxy ~]# vim /etc/ssh/sshd_config
.. ..
Protocol 2                                          //SSH 协议
```

```
PermitRootLogin no                                  //禁止 root 用户登录
PermitEmptyPasswords no                             //禁止密码为空的用户登录
UseDNS  no                                          //不解析客户机地址
LoginGraceTime  1m                                  //登录限时
MaxAuthTries  3                                     //每连接最多认证次数
.. ..
[root@proxy ~]# systemctl restart sshd
```

2）测试基本安全策略

尝试以 root 用户 SSH 登录，失败：

```
[root@proxy ~]# ssh root@192.168.4.5
root@192.168.4.5's password:
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：

```
[root@proxy ~]# passwd -d kate                      //清空用户口令
清除用户的密码 kate。
passwd: 操作成功

[root@proxy ~]# ssh kate@192.168.4.5
kate@192.168.4.5's password:
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。

```
[root@proxy ~]# vim /etc/ssh/sshd_config
.. ..
AllowUsers zhangsan tom useradm@192.168.4.0/24      //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
[root@proxy ~]# systemctl restart sshd
```

2）验证 SSH 访问控制，未授权的用户将拒绝登录。

```
[root@proxy ~]# ssh useradm@192.168.4.5             //已授权的用户允许登录
useradm@192.168.4.5's password:
[useradm@proxy ~]$ exit
[root@proxy ~]# ssh root@192.168.4.5                //未授权的用户被拒绝登录
root@192.168.4.5's password:
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）：

```
[root@client ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):              //直接回车将口令设为空
Enter same passphrase again:                             //再次回车确认
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
63:6e:cf:45:f0:56:e2:89:6f:62:64:5a:5e:fd:68:d2
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|         . . .   |
|          = =    |
|       S = B .   |
|      o B = . o  |
|       + + = E .|
|      . + + o    |
|         o       |
+-----------------+
[root@client ~]$ ls -lh ~/.ssh/id_rsa*                   //确认密钥对文件
-rw-------. 1 root root 1.8K 8月  15 10:35 /root/.ssh/id_rsa
-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：

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

```
[root@proxy ~]# tail -2 ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAzz+5AiFMGQ7LfuiV7eBnOcmRO9JRTcqRoynGO2y5
RyFL+LxR1IpEbkNrUyIZDk5uaX1Y8rwsf+pa7UZ2NyqmUEvNSUo0hQyDGsU9SPyAdzRCCvDgwpOFhaHi
/OFnT+zqjAqXH2M9fFYEVUU4PIVL8HT19zCQRVZ/q3acQA34UsQUR0PpLJAobsf1BLe2EDM8BsSHckDGsNo
DT9vk+u3e83RaehBMuy1cVEN5sLAaIrIeyM8Q0WxQNlqknL908HRkTlTeKrRoHbMnOBFj8StwlnscKHlkrs
KkhUf8A9WWz/vL4GDwGND5jdca3I2hdITAySjMdfL1HMHnMYOgMjPM0Q== root@192.168.4.100
```

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）：

```
[root@client ~]$ ssh root@192.168.4.5                    //免交互直接登入
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

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

```
PasswordAuthentication no                                    //将此行 yes 改成 no

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

4. 案例 4：SELinux 安全防护

- **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

- **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：

```
[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
```

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/d2.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
```