1.禁止其它机器ping自己的系统，因为ping包是计算机间交流数据的传输，没有经过任何的加密处理，为了不让不法分子利用这一个漏洞我们需要禁用ping

禁止ping的命令：echo 1 > /proc/sys/net/ipv4/icmp\_echo\_igore\_all

恢复ping的命令：echo 0 > /proc/sys/net/ipv4/icmp\_echo\_igore\_all

2.对/etc下的配置文件进行备份，这样就可以发现系统文件是否被非法修改过。如果发生系统文件已经被破坏的情况，也可以使用系统备份来恢复到正常的状态

3.改进服务器的登录系统，由于root是linux系统的超级管理员，如果密码设置太简单，就很容易被黑客暴力破解，为了服务器的安全起见，为了更好地保护公司数据安全，最好禁止root用户直接登录系统：

3.1新建一个用户，用来登录

user add xk

3.2设置密码

echo XiangKui | passwd --stdin xk

3.3修改SSHD配置，禁用root登录

vim /etc/ssh/sshd\_config

找到 “#PermitRootLogin yes”把注释去掉，把后面的yes改成no。

3.4重启SSHD服务

systemctl restart sshd

3.5下次登录系统的时候先用xk用户登录，然后在使用su - root命令切换root下即可

注：su的用法

su [OPTION选项参数] [用户]，参数说明：

-或者-l：登录并改变到所切换的用户环境；

-c：切换到对应的用户环境，执行一个命令，然后退出；

注意：

1.su在不加任何参数，默认为切换到root用户，但没有转到root用户家目录下，也就说这时虽然切换为root用户了，但并没有改变root登录环境；

2.su - root 和 su - 是一样的功能，都是切换到root用户，并且改变到root用户的环境；

4.用户提权：

4.1有时候我们只需要切换到某个用户，执行一个命令，然后就退出。上面我妈知道可以使用su来切换用户，但是需要知道用户的密码，特别是切换到root时需要知道root的密码才可以，这样做比较不安全。这个时候我们就可以使用sudo。通过sudo我们能把某些超级权限有针对性的下放，并且不需要普通用户知道root密码，所以sudo相对于权限无限制性的su来说，还是比较安全的。sudo执行命令的流程是当前用户切换到root（或其它指定切换到的用户），然后以root（或其它指定切换到的用户）身份执行命令，执行完成后直接退回到当前用户；而这些的前提是要通过sudo的配置文件/etc/sudoers来进行授权。

4.2sudo命令：

/usr/bin/sudo

4.2.1参数：

sudo -u username#uid User以指定用户的身份执行命令。后面的用户是除root以外的，可以是用户名，也可以是#uid

sudo -k kill清除“入场卷”上的时间，下次在使用sudo时要在输入密码。

sudo -K sure kill与-k类似，但是它还要撕毁“入场卷”，也就是删除时间戳文件。

sudo -b command在后台执行指定的命令。

4.2.2特点：

sudo能够限制指定用户在指定主机上运行某些命令。

sudo可以提供日志，忠实地记录每个用户使用sudo做了些什么，并且能将日志传到中心主机或者日志服务器。

sudo为系统管理员提供配置文件，允许系统管理员集中地管理用户的使用权限和使用的主机。它默认的存放路径是/etc/sudoers。

sudo使用时间戳文件来完成类似“检票”的系统。当用户执行sudo并且输入密码后，用户获得了一张默认存活为5分钟的入场卷。超时以后，用户必须重新输入密码

4.3sudoers文件

修改/etc/sudoers，不建议直接使用vim，而是使用visudo。因为修改/etc/sudoers文件需遵循一定的语法规则，使用visudo的好处在于，当修改完毕/etc/sudoers文件，离开修改页面时，系统会自动检验/etc/sudoers文件的语法。

因此修改/etc/sudoers文件命令如下：

#visudo

root ALL = (ALL) ALL <----大约76行的位置

#用户名 被管理主机的地址（ALL是所有）=（可使用的身份）授权命令（绝对路径）

#%wheel ALL=（ALL） ALL <----大约84行的位置

#%组名 被管理主机的地址（ALL是所有）=（可使用的身份）授权命令（绝对路径）

这两行是系统为我们提供的模板，我们参照它写自己就可以了

例：

lamp ALL=（ALL） /usr/bin/vim /etc/resolv.conf //这样lamp用户就拥有打开/etc/resolv.conf的权限了。

5.防火墙的策略

centos系统7以下的版本默认用的防火墙都是iptables，而7以上默认用的防火墙是firewalld，相比iptables的四表五链而言会更加方便管理，但是firewalld的底层用的也是调用了iptables，那么我们下面开始对这两种防火墙进行讲解：

5.1iptables：

我们先来了解一下iptables，iptables是一种“包的过滤防火墙”，那什么是防火墙呢？防火墙是一道保护性的安全屏障它主要起到对内网进行：1保护、2隔离。

iptables的过滤主要是依靠它的四表五链

5.1.1四表:

filter表：包的过滤表

nat表：地址转换表

mangle表：包的标记表

raw表：状态跟踪表

5.1.2五链：

INPUT链（入站规则）

OUTPUT链（出站规则）

FORWARD链（转发规则）

PREROUTING链（路由前规则）

POSTROUTING链（路由后规则）

5.1.3iptables的基本使用方法

1）iptables语法格式

iptables [-t 表名] 选项 [链名] [条件] [-j 目标操作]

例1：

iptables -t filter -I INPUT -p REJECT //filter表插入一条拒绝所有入站请求策略

例2：iptables -I INPUT -p icmp -j REJECT //不打表名默认是filter表，例2的意思是filter表插入一条拒绝icmp的入站请求

注：不指定表默认使用filter表

可以不指定链，默认为对应表的所有链

如果没有找到匹配条件，这执行防火墙默认规则

‘选项/链名/目标操作’ 用大写字母，其余的用小写

目标操作：

ACCEPT:允许通过/放行

DROP：直接丢弃，不给出任何回应

REJECT:拒绝通过，必要时会给出提示

LOG：记录日志，然后传给下一条规则 //防火墙默认都是匹配即停止，但是LOG除外，默认规则的优先级最低

iptables命令的常用选项：

|  |  |  |
| --- | --- | --- |
| 类别 | 选项 | 描述 |
| 添加规则 | -A | 追加一条防火墙规则至链的末尾位置 |
|  | -I | 插入一条防火墙规则至链的开头 |
| 查看规则 | -L | 查看iptables所有规则 |
|  | -n | 以数字形式显示地址，端口等信息 |
|  | --line-numbers | 查看规则时显示规则的行号 |
| 删除规则 | -D | 删除链内指定的序号或内容的一条规则 |
|  | -F | 清空所有规则 |
| 默认规则 | -P | 为指定的链设置默认规则 |

2）创建规则的案例：

例1

iptables -t filter -A INPUT -p tcp -j ACCEPT //追加规则至fiter表中的INPUT链的末尾，运行任何人使用tcp协议访问本机

例2

iptables -I INPUT -p udp -j ACCEPT //插入规则到filter表中的INPUT链的最前，运行任何人使用udp协议来访问本机

例3

iptables -I INPUT 2 -p icmp -j ACCEPT //插入规则到filter表中的INPUT链的第二行，运行任何人访问本机

iptables -nL INPUT //查看INPUT链的规则

iptables -nL //查看所有的防火墙规则

5.1.4删除规则，清空所有规则

例1

iptables -D INPUT 3 //删除filter表中INPUT链的第3条规则

例2

iptables -F //清空filter表中所有链的防火墙规则

例3

iptables -t nat -F //清空nat表中所有的链的防火墙规则

例4

iptables -t mangle -F //清空mangle表中所有链的防火墙规则

5.1.5设置防火墙默认规则

iptables -t filter -P INPUT DROP //将iptables的默认规则配置为拒绝所有

iptables -nL //就可以看到Chain INPUT（policy DROP）

5.2iptables的进阶

5.2.1filter过滤和转发控制

要求：

1.针对linux主机进行出战，入站控制

2.利用ip\_forward机制实现linux路由/网关功能

3.在linux网关上实现数据包转发访问控制

5.2.2根据防火墙保护对象不同，防火墙可以分为主机刑防火墙与网络型防火墙

1.主机型防火墙：主要保护的是服务器本机（过滤威胁本机的数据包）。

2.网络型防火墙：主要保护的是防火墙后面的其他服务器，如web服务器，FTP服务器等