上章回顾

- 如何对一个用户进行锁定与解锁定?
  - 锁定账户：passwd -l 用户名 、 usermod -L 用户名
  - 解锁账户：passwd -u 用户名 、 usermod -U 用户名（账户被锁定后，任何一种方式都可以解锁）
  - 查看用户状态：passwd -S 用户名
- 使用su切换用户时,“-”代表什么意思?
  - 连同Shell环境一起切换
- sudo 配置文件的格式中三个部分代表什么意思?
  - 用户名 来自那：权限
  - 组名 来自那：权限
- 如何限制更改grub引导参数?
  - 设置开机编辑模式中的密码
  - grub2-setpassword
- john和nmap作用是什么
  - john：检测弱密码
  - nmap：扫描计算机的端口状态

本章内容

- 学会构建SSH远程登录服务
- 学会使用SSH客户端工具
- 学会编写TCP Wrappers访问策略

一、SSH协议以及密码学概述

1、SSH，全称Secure Shell（安全外壳），是一种网络安全协议

2、SSH是一种用于安全访问远程服务器的协议，为客户机提供安全的Shell环境，用于远程管理的工具。

3、SSH账户验证分为密码认证和密钥对认证，加密算法主要分为两种：

1）对称加密算法（DES）

- **发送方使用密钥将明文数据加密成密文，然后发送出去**
- **接收方收到密文后，使用同一个密钥，将密文解密成明文进行读取**

2）非对称加密算法（RSA、DSA）

- **发送方使用接收方发送过来的公钥，将明文数据加密为密文，然后向外发出**
- **接收方收到密文后，使用自己的私钥将密文数据解密进行读取**

3）两者的区别

- 对称加密：使用同一个密钥进行加密解密，密钥容易泄漏造成数据丢失；但是加密速度、传输速度快，效率更高，安全性偏低
- 非对称加密：使用**公加私解**的方式完成加密解密，安全性高，但是加密速度、传输速度较慢

4、SSH服务端

1）Linux下默认安装这ssh（OpenSSH）服务，类似于windows中的远程桌面功能

2）使用TCP协议中的 22 端口号

3）使用 sshd 程序控制服务启动

4）主配置文件：/etc/ssh/sshd_config

二、SSH服务端配置

1、主配置文件的常用配置项

- **Port 22：指定端口号**
- **Protocol 2：定义版本号，默认是版本二，支持非对称加密等功能**
- **ListenAddress 192.168.1.1：监听IP地址**
- **UserDNS no：禁止反向解析，反向解析为了防止假冒的IP连接服务器，会把IP解析成域名，查看IP是否是伪造的；关闭后加快了ssh的连接**
- **PermitRootLogin no：禁止root用户登陆**
- **PermitEmptyPasswords no：禁止空密码登录**
- **LoginGraceTime 2m：登陆时间**
- **MaxAuthTries 6：重试次数**
- **AllowUsers jerry tom admin@192.168.1.1：用户控制，允许登入的用户列表，用户之间使用空格分隔，不可以和DenyUsers同时使用**
- **PasswordAuthentication yes：开启密码验证登陆**
- **PubKeyAuthentication yes：开启密钥对登陆**
- **AuthorizedKeysFile .ssh/authorized_keys：服务端口公钥文件存放位置**

2、登陆ssh的验证方式

1）密码验证

- 使用服务器的本地用户身份登陆验证
- 对于客户端而言使用简单
- 对服务器而言防御力较弱，有被暴力破解的风险

2）密钥对验证

- 要求提供匹配的密钥信息才能通过验证
- 客户机生产一对密钥，然后将公钥传到服务器的指定位置，远程登陆时，系统使用私钥、公钥进行加密解密
- 由于不需要密码，所以增强了远程管理的安全性

三、SSH远程登入

1、SSH客户端指令

1）语法：

- ssh host
- ssh user@host
- ssh -l user host：指定连接用户，不指定使用当前登录的用户身份登录对方
  - -p：指定端口

2、SSH密码登陆验证的步骤

1）客户端向服务器发送一个登陆请求

2）服务器把自己的公钥发给客户端

3）客户端使用服务器发送过来的公钥把自己的密码进行加密，并且发送给SSH服务端

4）服务端收到后使用私钥进行解密

5）将解密的密码和本地/etc/shadow文件中的密码对比认证

6）认证成功则返回登陆结果，并且发送一个随机回话口令给客户端，这个口令用于后期两台主机进行数据传输时临时加密的口令

3、案例

1）正常登录验证

> 当客户端输入yes确认对方的公钥指纹后，server端的公钥就会被存放到客户机的用户家目录里~/.ssh/known_hosts文件中，下次再访问就直接通过密码登录，不需要再确认公钥。

```
[root@hostA ~] netstat -anpt | grep sshd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      993/sshd

tcp        0     52 192.168.62.128:22       192.168.62.1:52119      ESTABLISHED
37663/sshd: root@pt
tcp6       0      0 :::22                   :::*                    LISTEN      993/sshd


[root@hostA ~] ssh 192.168.62.129
The authenticity of host '192.168.62.129 (192.168.62.129)' can't be established.
ECDSA key fingerprint is SHA256:UADrn8TIlvgz4rdfPa2P5rdZ8Gwz0tuAXL99eBlL7Xk.
ECDSA key fingerprint is MD5:c8:80:dc:a8:4d:42:a0:30:ef:e9:1d:54:25:55:49:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.62.129' (ECDSA) to the list of known hosts.

root@192.168.62.129's password:
Last login: Thu Jan 11 15:40:46 2024 from 192.168.62.1
[root@hostB ~] exit
登出
Connection to 192.168.62.129 closed

[root@hostA ~] ssh aaa@192.168.62.129

aaa@192.168.62.129's password:
Last failed login: Thu Jan 11 15:44:43 CST 2024 from 192.168.62.128 on ssh:notty
There was 1 failed login attempt since the last successful login.
[aaa@hostB ~]$

```

2）使用其他端口号登录

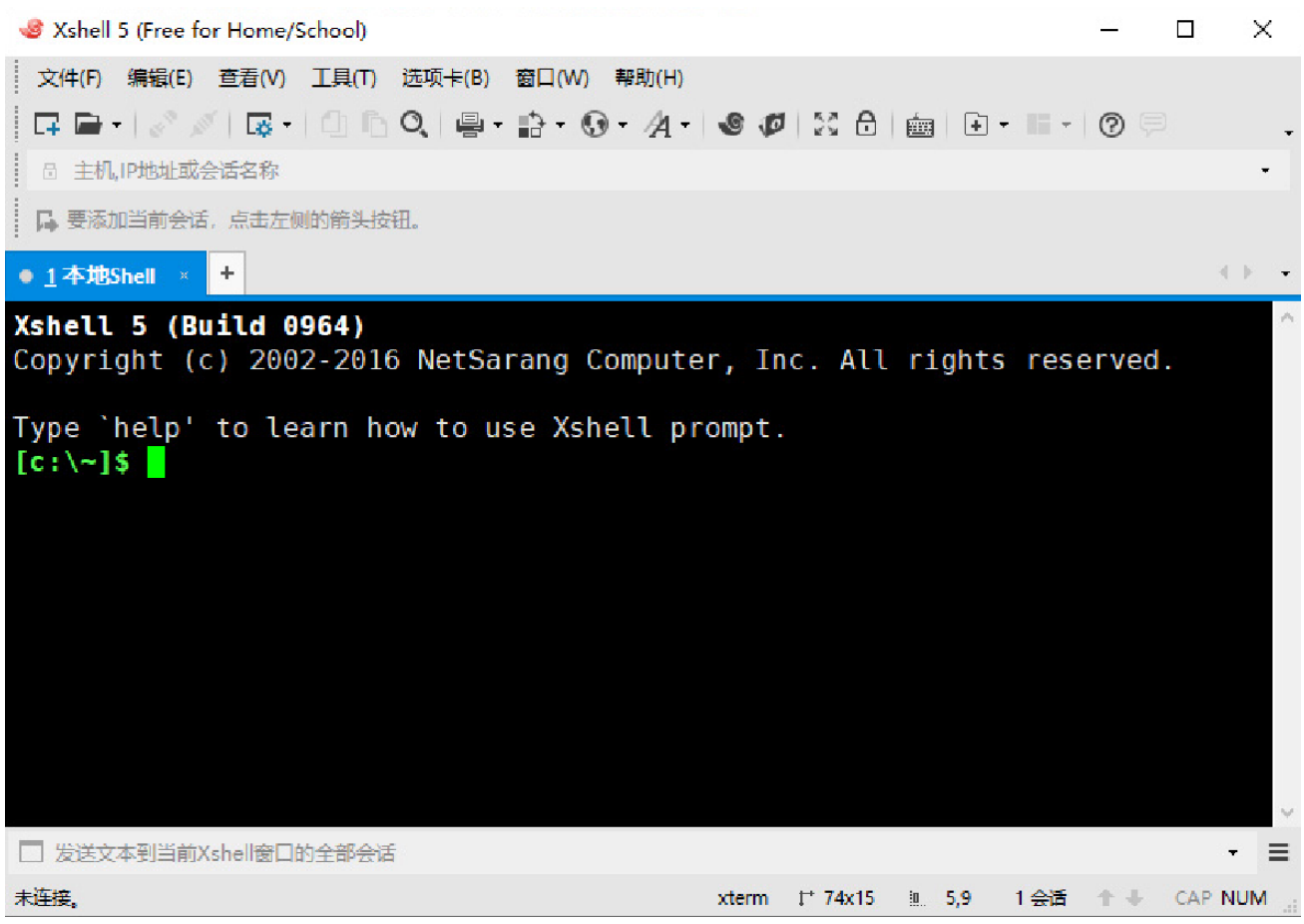
```
[root@hostB ~] netstat -anpt | grep ssh
```

```
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN
1819/sshd
tcp        0     52 192.168.62.129:22       192.168.62.1:56295      ESTABLISHED
1676/sshd: root@pts
tcp        0      0 192.168.62.129:2222     192.168.62.128:60668    ESTABLISHED
1820/sshd: aaa [pri
tcp6       0      0 :::2222                 :::*                    LISTEN
1819/sshd

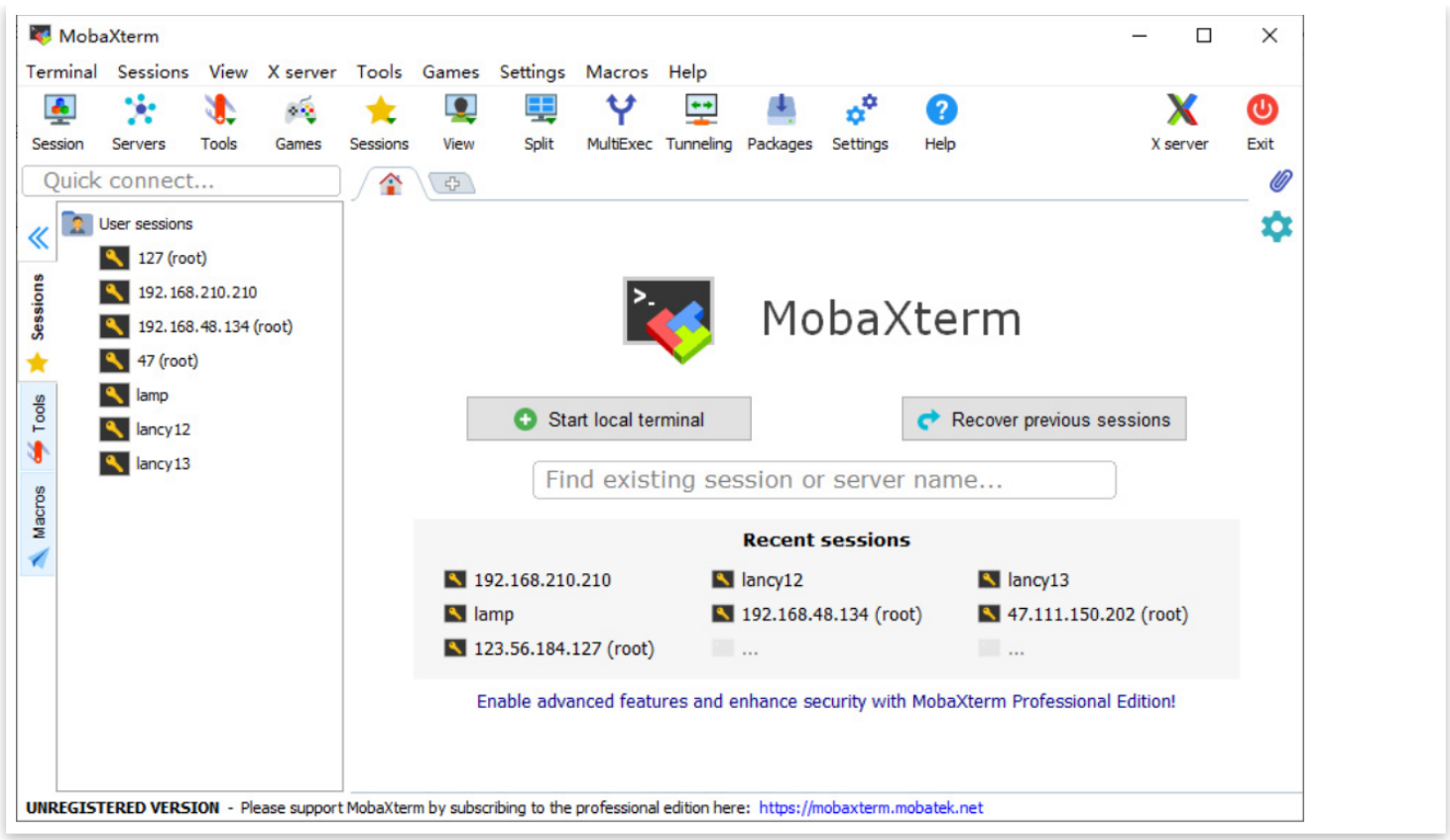
[root@hostA ~] ssh -p 2222 aaa@192.168.62.129
aaa@192.168.62.129's password:
Last login: Thu Jan 11 15:45:42 2024 from 192.168.62.128
```

4、windows下常用的SSH客户端软件

1）XShell：



2）MobaXterm：



四、使用scp命令

1、主要用于远程复制文件、数据

2、语法

- scp user@host:/aaa/file /root //将对方服务器中的/aaa/file文件复制到本地的/root中
- scp file1 user@host:file2 //将本地文件拷贝到远程
  - -r：递归拷贝目录
  - -P：指定服务器端口

3、案例（将当前文件复制到目标服务器）

```
[root@hostA ~] scp -P 2222 /root/hostAAA root@192.168.62.129:/root/

root@192.168.62.129's password:
hostAAA
[root@hostB ~] ls
anaconda-ks.cfg  hostAAA
                                                                    100%
    0     0.0KB/s   00:00
```

4、案例（从服务器下载文件到本机）

```
[root@hostA ~] scp -P 2222 root@192.168.62.129:/root/hostBBB /root/
```

```
root@192.168.62.129's password:
hostBBB                                       100%    0     0.0KB/s   00:00
```

5、使用sftp安全下载文件

1）语法：sftp user@host

- ls：用于列出远程目录的内容。
- cd directoryname：用于更改远程目录。
- get filename：用于从远程主机获取文件。
- put filename：用于将本地文件发送到远程主机。
- rm filename：用于删除远程主机上的文件。
- exit：用于终止与远程主机的连接。

```
[root@hostA ~] sftp -P 2222 root@192.168.62.129
root@192.168.62.129's password:
Connected to 192.168.62.129.
sftp>
```

五、SSH访问登陆案例

1、案例环境一：创建用户并且授权

1）环境：开发部门3个员工，zhangsan，lisi，wanwu

```
[root@localhost ~] useradd zhangsan
[root@localhost ~] useradd lisi
[root@localhost ~] useradd wanwu

[root@localhost ~] echo "qm123.com" | passwd --stdin zhangsan
[root@localhost ~] echo "qm123.com" | passwd --stdin lisi
[root@localhost ~] echo "qm123.com" | passwd --stdin wanwu
```

2、实验案例二：禁止root远程登录

1）注意：在生产服务器server端完成

```
[root@localhost ~] vim /etc/ssh/sshd_config
#PermitRootLogin yes
PermitRootLogin no

```

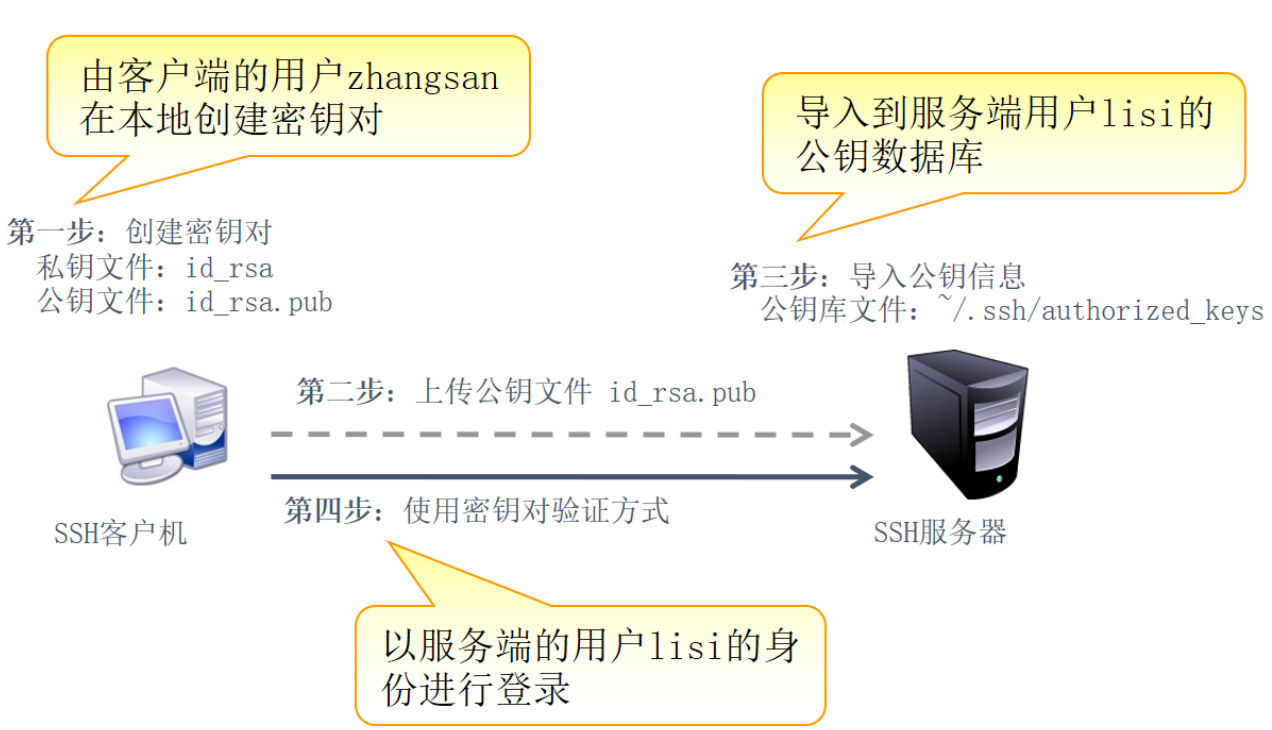
```
[root@localhost ~] systemctl restart sshd

[root@localhost ~] ssh root@192.168.1.1
[root@localhost ~] ssh lisi@192.168.1.1
```

3、实验案例三：仅允许zhangsan、lisi用户登录，仅允许王五通过某个IP地址登录

```
[root@localhost ~] vim /etc/ssh/sshd_config
AllowUsers zhangsan lisi wangwu@192.168.1.1
[root@localhost ~] systemctl restart sshd
```

六、使用密钥对登录



1、密钥对登陆具体执行步骤

1）客户端使用命令生成密钥对并且指定加密算法，默认存放在当前用户的宿主下

- ssh-keygen -t rsa

```
[root@hostA ~] ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):     //确认密钥匙文件
存放位置及名称
Enter passphrase (empty for no passphrase):                  // 输入密码短语（为空表示没有密码
短语）：
```

```
Enter same passphrase again:
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:
SHA256:6199xfNhAlPP1L/NUTmhmLW28tX1ozSuUSex9E8ABz0 root@hostA
The key's randomart image is:
+---[RSA 2048]----+
|            o=.o+|
|            =oEoo|
|           = *.+=|
|            = =+=|
|        S  . O XX|
|         .  *.OoX|
|        .  ..+. +|
|       .   .o  . |
|        ....     |
+----[SHA256]-----+
```

2）使用scp命令将公钥文件上传到服务端

```
[root@hostA ~] scp .ssh/id_rsa.pub  root@192.168.62.129:/tmp/
root@192.168.62.129's password:
id_rsa.pub                                    100%  392   980.0KB/s   00:00
```

3）服务端将公钥文件存储到需要使用密钥对登陆的用户的宿主目录的指定目录下，并且修改文件名称

> 注意服务器中的.ssh目录权限中的other不允许具有 x 权限，否则可能无法完成密钥对登陆

```
[root@hostB ~] mkdir -p /home/aaa/.ssh
[root@hostB ~] mv /tmp/id_rsa.pub /home/aaa/.ssh/authorized_keys
[root@hostB ~] tail -1 /home/aaa/.ssh/authorized_keys
ssh-rsaAAAB3NzaC1yc2EAAAADAQABAAABAQCWk2wf1PcEfi7kr/AX3y6VM6wRBDh2lqQZXJw0Dl2bUmM1E1uBYzXy2qydO8GN1XRQcRAy3kEW7snC9IyQ7ZYRGxJbqElAenNSDuqJyjdx8v9ryt5zCynNTKm4gm9p65t7cuC0Tkq6fo9NeZ7MCrlR2x6wIYNMaQMBDMCMOyQDF+UFcsl17/AVSvGorDGxoS2mBEdE9JWCcRuBfja3DRmiDy9gE77IoadzrYp4ceg3OoUagUmHBJBS9JN1UzK+2lBU5NI8S0BRaKcS7ui5Jobru+IFQyT+1WJx0oNMHLpU3JS9FTRros6AWDd927gXSwfZiJzQNYtvBfH8iKjFvfW5 root@hostA

```

4）客户端验证结果

```
[root@hostA ~] ssh aaa@192.168.62.129
Last login: Thu Jan 11 15:48:07 2024 from 192.168.62.128
[aaa@hostB ~]$
```

2、密钥对登陆优化步骤

1）可以将2、3步骤进行简化

2）ssh-copy-id -i 公钥文件 user@host

3）验证密码后，会将公钥自动添加到目标主机user宿主目录下的.ssh/authorized_keys文件结尾

```
[zhangsan@localhost~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub lisi@192.168.4.254
```

七、TCP Wrappers控制

1、Wrappers概述

1）是一个工作在第四层的安全工具，对现有状态连接的特定服务进行安全检测并且实现访问控制

2）凡是包含有libwrap.so库文件的的程序就可以受TCP_Wrappers的安全控制

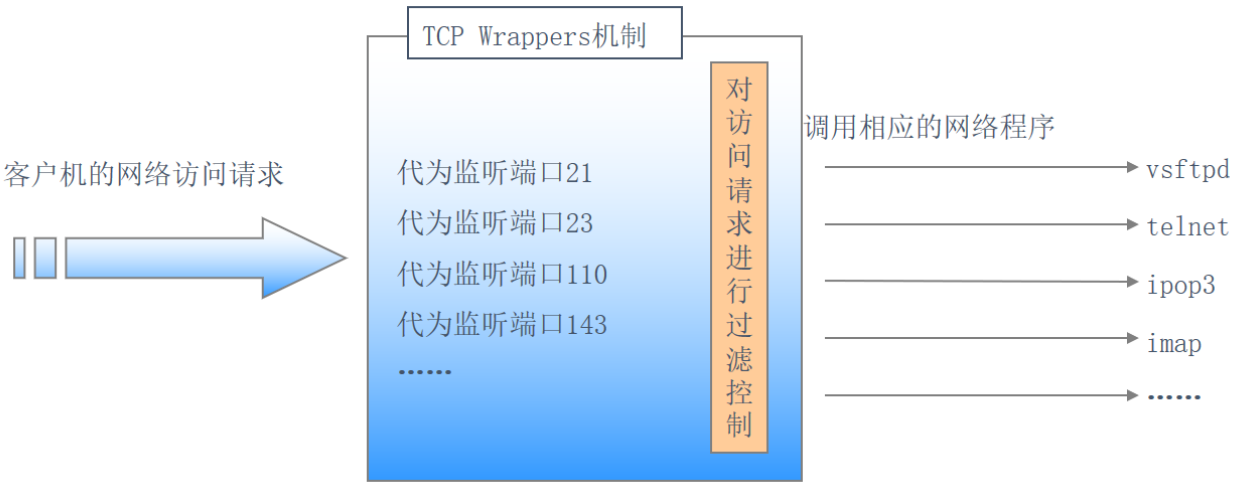
3）主要功能就是控制谁可以进行访问，常见的控制服务：Telnet、VSftp、sshd等

2、Wrappers保护原理

1）TCP_Wrappers有一个TCP的守护进程叫做tcpd。

2）以为ssh为例，每当有ssh的连接请求时，tcpd将会截获请求，然后读取管理员设置的访问控制文件，符合要求则会把连接原封不动的转给ssh进程，然后由ssh完成后续工作

3）如果这次连接发送的ip不符合访问控制文件中的设置，则会中断连接请求，拒绝提供ssh服务



3、保护机制的实现方式

1）方式一：通过tcpd主程序对其他服务程序进行包装

2）方式二：由其他服务调用libwrap.so.*链接库

4、访问控制列表

1）/etc/hosts.allow

2）/etc/hosts.deny

5、策略编写格式

1）服务列表：客户机地址列表

2）服务器列表：

- 多个服务器以逗号分隔，ALL表示所有服务

3）客户机地址列表：

- 多个服务器以逗号分隔，ALL表示所有服务
- 允许使用通配符？和*
- 网段地址：192.168.1. 或者 192.168.1.0/255.255.255
- 区域地址：.qm.com

6、应用顺序

1）先检查hosts.allow文件

2）再检查hosts.deny文件

3）如果都匹配不上默认允许

7、策略应用实例

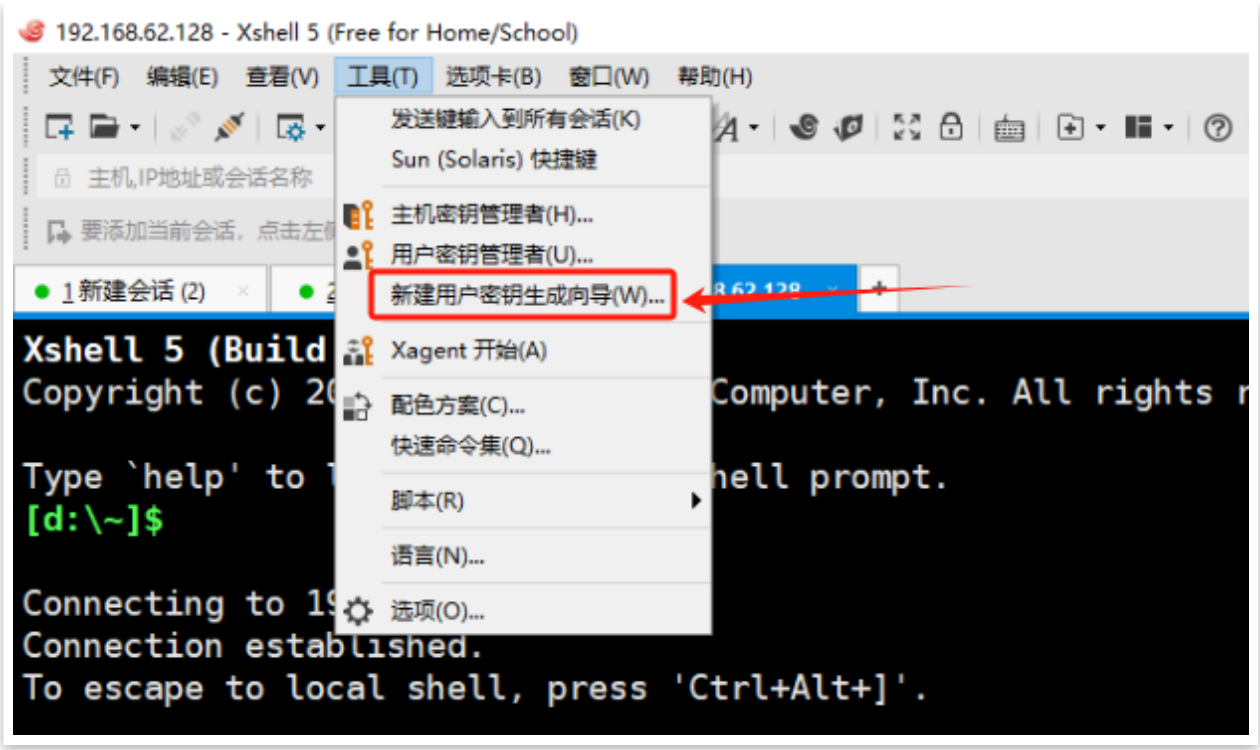
```
[root@hostB ~] vim /etc/hosts.deny
sshd: 192.168.62.128

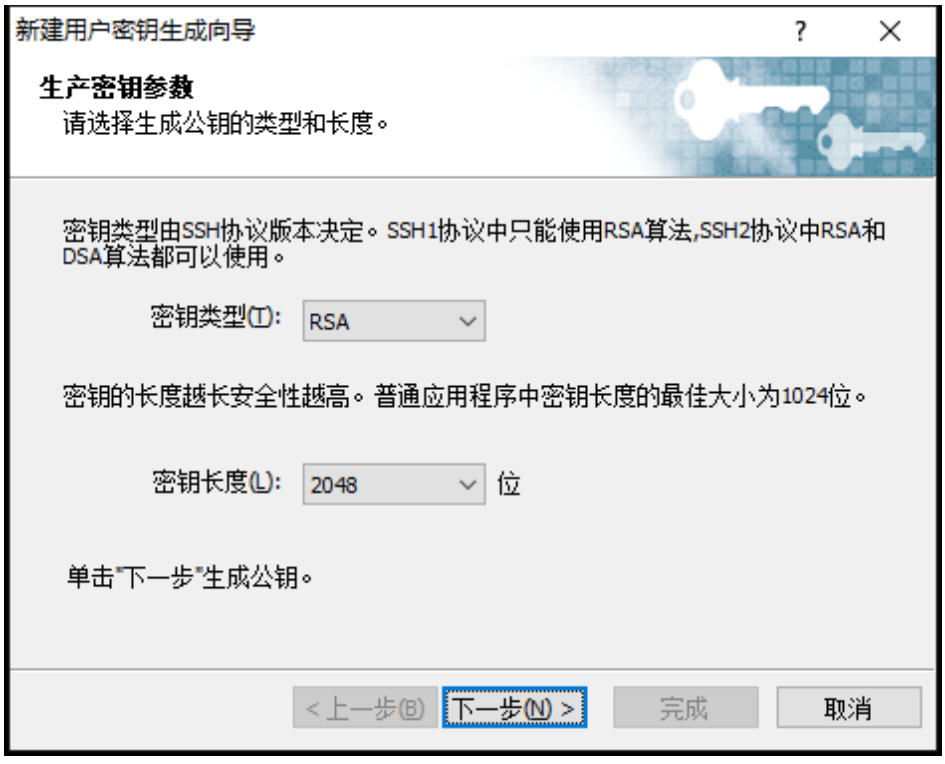
[root@hostA ~] ssh aaa@192.168.62.129
ssh_exchange_identification: read: Connection reset by peer
```

八、Xshell软件使用密钥对登陆（扩展）

1、本地生成密钥对

1）方法一：使用Xshell生成

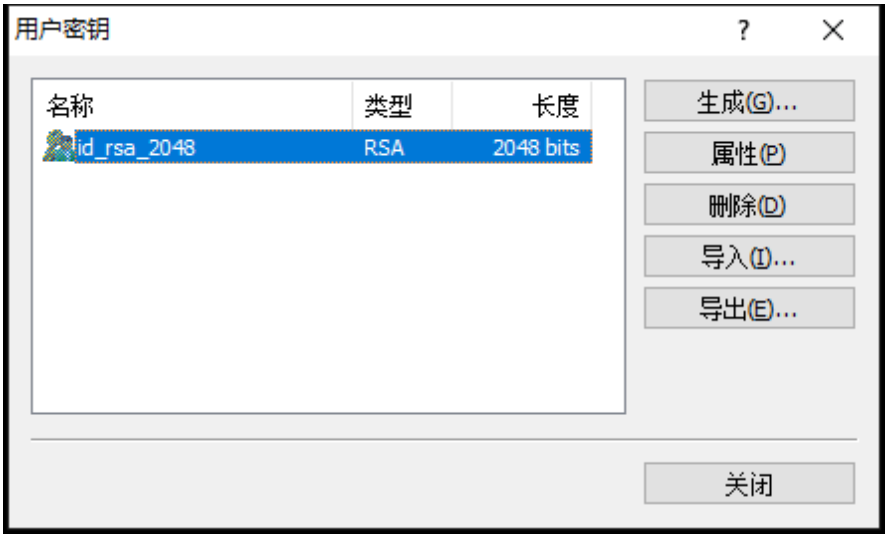

192.168.62.128 - Xshell 5 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
发送键输入到所有会话(K)
Sun (Solaris) 快捷键
主机密钥管理者(H)...
用户密钥管理者(U)...
新建用户密钥生成向导(W)...
Xagent 开始(A)
配色方案(C)...
快速命令集(Q)...
脚本(R)
语言(N)...
选项(O)...
主机,IP地址或会话名称
要添加当前会话，点击左
1 新建会话 (2)
Xshell 5 (Build
Copyright (c) 2
Computer, Inc. All rights r
Type `help' to
hell prompt.
[d:\~]$
Connecting to 19
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.


新建用户密钥生成向导
生产密钥参数
请选择生成公钥的类型和长度。
密钥类型由SSH协议版本决定。SSH1协议中只能使用RSA算法,SSH2协议中RSA和DSA算法都可以使用。
密钥类型(T): RSA
密钥的长度越长安全性越高。普通应用程序中密钥长度的最佳大小为1024位。
密钥长度(L): 2048 位
单击"下一步"生成公钥。
< 上一步(B)
下一步(N) >
完成
取消


新建用户密钥生成向导
?
生成公钥对
生成公钥和私钥对。
2048 位 RSA 用户密钥对正在生成,请稍等...
0 1 0 1 1 0 1 0 0 0 0 1 0 1 0
1 0 1 1 0 1 1 0 1 0 0 0 0 1 0
1 0 1 0 1 0 1 0 1 1 0 1 0 0 0
0 1 0 1 0 1 0 1 1 0 1 1 0 1 0
公钥对已成功生成。
请单击"下一步"输入用户密钥的名称和密码。
< 上一步(B)
下一步(N) >
完成
取消


新建用户密钥生成向导
?
用户密钥信息
请输入生成用户密钥的名称和密码。
请输入用户密钥的名称。
密钥名称(K):
id_rsa_2048 (2)
请输入给用户密钥加密的密码。
密码(P):
确认(C):
(重新键入密码)
请单击"下一步"在SSH服务器上注册公钥。
< 上一步(B)
下一步(N) >
完成
取消





2）使用cmd命令生成

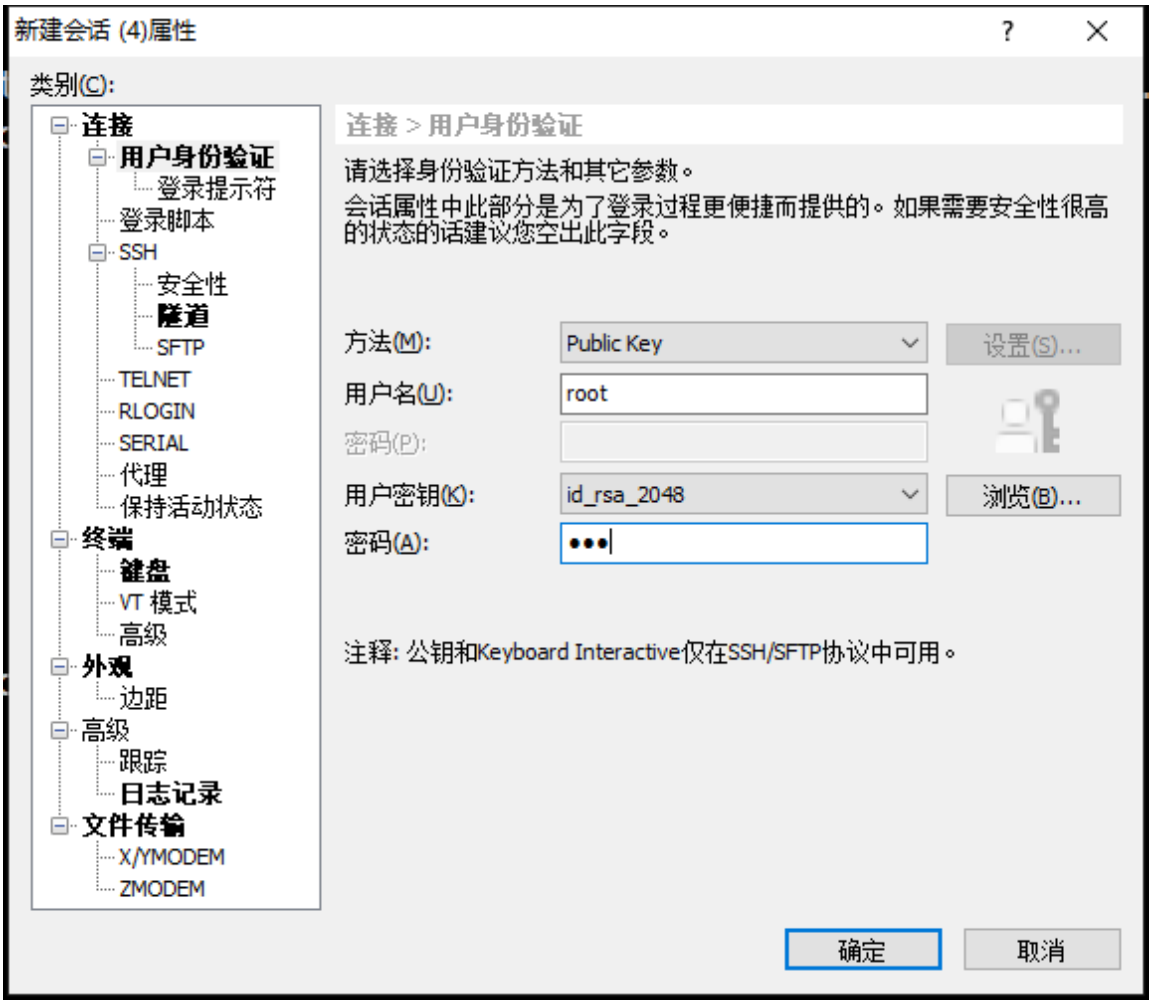
```
C:\Users\Administrator>  ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\Administrator/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\Administrator/.ssh/id_rsa.
Your public key has been saved in C:\Users\Administrator/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Ww61WXdw15WjIxfNLyzBalMya2d4ikQQID9ZAO9DkV8 administrator@Ai2023021011525
The key's randomart image is:
+---[RSA 3072]----+
|  o.+++o    .  + B|
|   +.+  E o +. Bo|
```

```
|    *. o  .B.o+ +|
|   o .. ..B===o..|
|    o  .S++*o... |
|     .  .=.      |
|        . .      |
|                 |
|                 |
+----[SHA256]-----+
```

2、导入密钥

```
[root@hostA ~] ls id*
id_rsa_2048.pub
[root@hostA ~] cat id_rsa_2048.pub >> /root/.ssh/authorized_keys
[root@hostA ~] chmod 600 /root/.ssh/authorized_keys
```

3、配置使用ssh密钥对登陆



作业：

实验要求描述

- 允许用户qm从任意地址登录，采用密钥对验证（公钥文件必须使用scp方式上传）
  - 使用 scp 上传公钥文件的截图
  - 服务端将公钥文件移动到 qm 目录下的截图
  - 客户端秘钥对登录得截图
- 允许用户"name"从主机192.168.x.x 登录
  - 查看自身的IP地址，然后登录成功的截图
  - 修改一个IP，再次查看，登录失败的截图
- 禁止其他所有用户远程登录
  - 其他随意用户登录失败的截图