Linux 网络基础

1 知识回顾

网络地址：互联网协议地址（IP地址）为互联网上每一个网络或主机分配一个逻辑地址，IP地址工作在网络层。

IP的分类：IPV4 IPV6

物理地址：物理地址（MAC地址）为每一个设备设置一个固定的地址，MAC地址工作在链路层。

MAC地址：00-23-5A-15-99-42

协议分类：

应用层协议：	FTP、HTTP、SMTP、Telnet、DNS等
传输层协议：	TCP、UDP等
网络层协议：	IP、ICMP、ARP等
数据链路层协议：	PPP协议等
物理层协议：	不常用

常见端口：/etc/service #端口配置文件

20/21	ftp服务	文件共享
22	ssh服务	安全远程管理
23	telnet服务	不安全远程管理
25	smtp：简单邮件传输协议	发信
465	smtp(ssl)	发信
110	pop3：邮局协议	收信
143	imap4	收信
993	imap4(ssl)	收信
80	www服务（http://）	网页访问
443	www服务（https://）	加密网页访问
3306	mysql端口	数据库连接端口
53	DNS端口	域名解析端口

2 常见网络配置

临时配置： 使用命令调整网络参数简单、快速，可直接修改运行中的网络参数一般只适合在调试网络的过程中使用，系统重启以后所做的修改将会失效

固定配置： 通过配置文件修改网络参数修改各项网络参数的配置文件适合对服务器设置固定参数时使用需要重载网络服务或者重启以后才会生效

注意：CentOS 7.x 很多操作都可以通过命令实现永久生效了，减少了手动修改配置文件出现错误的几率。

2.1 iP 地址配置

ifconfig：网卡临时配置命令

示例：

```
ifconfig  eth0  192.168.12.250  netmask  255.255.255.0
ifconfig  eth0  192.168.12.250/24
```

网卡配置文件：

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0                 #设备名称
NAME=eth0                   #网卡名称
BOOTPROTO=static            #连接方式（dhcp/static）
ONBOOT=yes                  #是否开机加载
IPADDR=192.168.12.250       #IP地址
NETMASK=255.255.255.0       #子网掩码（PREFIX=24）
GATEWAY=192.168.12.1        #网关
DNS1=8.8.8.8                #DNS
```

重启网络服务： service network restart

启动关闭网卡： ifup 网卡名 ifdown 网卡名

2.2 主机名配置

临时生效：hostname 主机名

永久生效：/etc/sysconfig/network

注： 区别 /etc/rc.d/rc.sysinit 和 /etc/sysconfig/network

2.3 网关配置

route ：查看系统中的路由表信息

临时：

添加：route add default gw ip

删除：route del default gw ip

永久：

/etc/sysconfig/network-scripts/ifcfg-eth0

2.4 DNS 配置

配置文件：在相应的配置文件中填写DNS服务器地址

局部：/etc/sysconfig/network-scripts/ifcfg-eth0

DNS=ip

全局：/etc/resolv.conf

nameserver ip

测试命令：nslookup

nslookup www.atguigu.com

主机映射文件：/etc/hosts

#用于保存主机名和IP地址的映射记录，但这种映射只是本机的映射，也就是说其映射关系是仅自己可见

主机映射文件和 DNS 服务器的比较：

默认情况下，系统首先从 hosts 文件查找解析记录

hosts 文件只对当前的主机有效

hosts 文件可减少 DNS 查询过程，提高解析效率

3 网络常用命令

3.1 网络信息查看命令

netstat：查看系统的网络连接状态、路由信息、接口等

常用选项：

-a：显示所有活动连接

-n：以数字形式显示

-t：查看 TCP 协议相关信息

-u：查看 UDP 协议相关信息

-p：显示 PID 和进程名

-l：监听

3.2 网络节点测试命令

traceroute：测试从当前主机到目的主机之间经过的网络节点数，用于追踪数据包在网络上传输时的全部路径，它默认发送的数据包大小是40字节，默认使用ICMP协议

常用选项:

-p　使用UDP端口进行测试，默认端口为33434

-q 3 指定测试时发送的数据包个数（即测试次数）

-n　以IP的方式进行连接测试，避开DNS的解析

注意：该命令在使用NAT模式时无法实现效果，请切换桥接模式（官方回复）

3.3 网络连通性测试命令

```
ping
      -i    指定间隔时间
      -c    指定ping的次数
      -s    指定数据包的大小
```

3.4 地址解析命令

```
arp   地址解析协议，将ip地址解析成MAC地址
      -a    查看所有
      -d    ip地址，删除某条ARP记录
```

3.5 网络探测扫描命令

```
nmap
      -sP       探测某网段内有哪些主机是存活的
      -sT       探测某主机上开启了哪些TCP端口
```

4 远程管理工具

```
Windows —> Linux:      Xshell、SecureCRT等
 Linux —> Windwos:     rdesktop命令（图形界面）
 Linux —> Linux:       ssh命令
```

5 ssh 安全远程管理

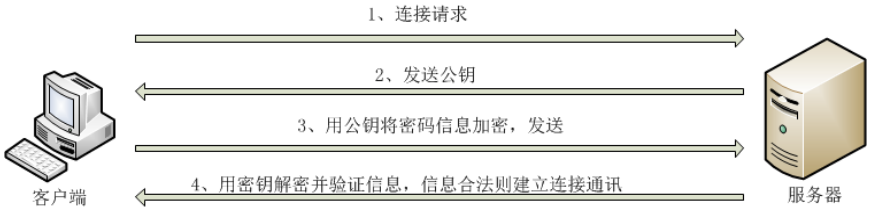
5.1 什么是 ssh

ssh 是 Secure Shell 的缩写，是一个建立在应用层上的安全远程管理协议。ssh 是目前较为可靠的传输协议，专为远程登录会话和其他网络服务提供安全性。利用 ssh 协议可以有效防止远程管理过程中的信息泄露问题。
ssh可用于大多数UNIX和类UNIX操作系统中，能够实现字符界面的远程登录管理，它默认使用22端口，采用密文的形式在网络中传输数据，相对于通过明文传输的Telnet协议，具有更高的安全性。

5.2 ssh 的登录验证模式

ssh 提供了基于账户密码（口令）和密钥对两种登录验证方式，这两者都是通过密文传输数据的。

账户密码验证：



账户密码登录认证过程中传输的是用户的账户名和密码，密码具有足够的复杂度才能具有更高的安全性。

Linux主机之间的远程管理工具是ssh命令，所以我们直接使用ssh进行远程登录

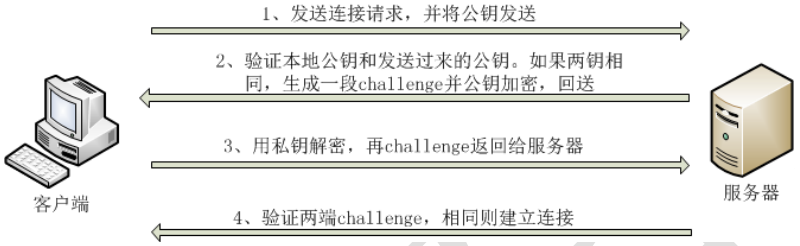
格式：

ssh 用户名@IP地址

ssh root@192.168.88.20

windows远程登录Linux主机一般使用第三方工具，比如Xshell等工具

密钥对验证：



1. 首先需要在 Client 上创建一对密钥，并且需要把公钥放在需要访问的 Server 上
2. 当 Client 需要连接 Server 时，Client 端的软件就会向 Server 端发出登录请求，请求使用密钥对中的的公钥进行安全验证
3. Server 收到请求之后，会在该用户的家目录下查询公钥文件，拿 Client 发送过来的公钥和自己家目录下的公钥进行比较
4. 如果两个公钥一致，Server 就用公钥加密“challenge（质疑）”，并把它发送给 Client 软件。Client 收到加密内容之后，使用本地的私钥进行解密，再把解密结果发送给 Server 端，Server 端验证成功后，允许登录

注意：若第3个步骤对比结果失败，则 Server 端会通知 Client 端此公钥未在本机注册，无法验证登录

5.3 配置 ssh 服务

5.3.1 环境准备

准备好两台Linux操作系统的主机，配置好相关网络参数，实现可以正常通信，并将主机名修改为不同的名字

临时关闭防护功能：

```
iptables -F    #清空防火墙规则
setenforce 0   #临时关闭SELinux
```

永久关闭防护功能：

```
chkconfig iptables off                                  #设置防火墙开机不自启动
sed -i ‘7s/enforcing/disabled/’ /etc/selinux/config     #永久关闭SELinux
```

注意：以上两条命令执行后，需要重启服务器才能生效，切记

5.3.2 用户密码验证

Linux主机之间的远程管理工具是ssh命令，所以我们直接使用ssh进行远程登录

格式：

```
ssh 用户名@IP地址
ssh root@192.168.88.20
```

windows远程登录Linux主机一般使用第三方工具，比如Xshell等工具

格式：

```
ssh root@192.168.88.20
```

5.3.3 密钥对验证

Linux 主机之间的密钥对登录验证

1. 客户端生成密钥对文件

   ssh-keygen -t rsa -b 2048

   -t 指定加密类型（rsa/dsa等）

   -b 指定密钥对加密长度

   询问1：执行过程中会询问保存位置，一般默认保存在当前用户家目录下的.ssh/目录下

   询问2：是否对密钥文件进行加密

   加密：若加密，则在调用密钥文件时需要先验证密钥的密码，密码正确才能使用密钥文件

   不加密：若不加密，则密钥文件可以直接被调用，整个登录验证过程无需输入任何密码，即为免密登录

2. 将公钥文件上传至服务器端

   ssh-copy-id 用户名@服务器IP地址

   #该用户名和要用来登录服务器的用户名一致

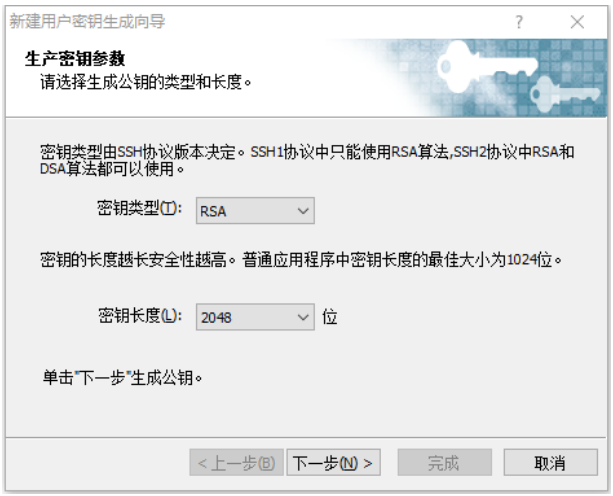
3. 客户端尝试登录服务器

   ssh 用户名@服务器IP地址

   #密钥对验证优先级大于账户密码验证

Windows使用密钥对登录Linux

1. 使用Xshell自带的密钥对生成向导生成密钥对



2. 将公钥导入Linux主机的指定用户下的指定公钥配置文件内
   后面用哪个用户登录就放在谁家里，这里我们先用root用户做实验
   在root家目录下，找到 .ssh 目录，然后在里面创建 authorized_keys 文件，并且将公钥写入进去
3. 使用windows尝试登录指定用户

5.3.4 禁止使用密码登录

当我们学会了使用密钥对进行验证后，建议生产环境下将账户密码登录功能关掉
配置文件：/etc/ssh/sshd_config
选项：

```
PasswordAuthentication no
```

注意：ssh的配置文件中，并不是注释掉的就是不生效的，有些是默认生效，需要修改时一定要取消注释再修改

5.3.5 禁止使用 root 远程登录

root 在系统中是一个可以为所欲为的角色，我们可以在平时的操作中用普通用户操作，在有需要修改一些系统设置的时候再从普通用户切换到 root 用户，这样可以最大限度的避免因为误操作而对系统造成破坏，同时也可以避免黑客在暴力破解后直接使用 root 用户登录系统，一般在远程登录管理上我们会禁止直接使用 root 用户登录
配置文件：/etc/ssh/sshd_config
选项：

```
PermitRootLogin no
```

5.3.6 修改默认端口、限制 ssh 监听 IP

修改默认端口：ssh 作为一个用来远程管理服务器的工具，需要特别的安全，默认情况下使用TCP的22端口，若不进行修改，很容易被利用遭到攻击，所以我们一般都会修改端口，尽量修改一个高位端口（范围1-65535）
配置文件：/etc/ssh/sshd_config

选项：

Port 59527

ssh -p 端口 用户名@服务器IP

限制ssh监听IP：有些服务器则安全级别更高一些，不允许使用外网直接登录，只有通过局域网才能登录，我们可以在机房里设置其中一台能够被外网远程连接，其他的主机都通过这个机器进行远程连接即可

配置文件：/etc/ssh/sshd_config

选项：

ListenAddress 192.168.88.100

5.4 ssh 服务相关命令

scp：安全的远程文件复制命令

scp是secure copy的简写，用于在Linux下进行远程拷贝文件的命令，类似于命令有cp，scp传输是加密的，所以可能会稍微影响一点速度。另外，scp还非常不占资源，不会提高多少系统负荷

格式：scp 本地文件 用户名@服务器IP:目录

scp /root/atguigu.txt root@192.168.88.20:/tmp

-P 端口 #若端口不是默认22，则需要使用此格式指定端口

sftp：安全的文件传输协议

sftp是Secure FileTransferProtocol的缩写，安全文件传送协议。sftp与ftp有着几乎一样的语法和功能。由于这种传输方式使用了加密/解密技术，所以sftp比ftp更安全一些，但传输效率比普通的FTP要低得多

格式：sftp 用户名@服务器IP

sftp

-oPort=端口 #若端口不是默认22，则需要使用此格式指定端口

交互命令：

help：查看在交互模式下支持哪些命令

pwd/lpwd：pwd是查看服务器所在路径；lpwd是查看客户端所在路径

ls/lls：ls是查看服务器当前目录下的文件列表；lls是查看客户机当前所在路径的所有文件列表

put：将客户机中的指定文件上传到服务器端

get：将服务器端的指定文件下载到客户机的当前所在目录

rm：删除掉服务器端的指定文件

quit：退出sftp的交互模式，断开和服务器之间的连接

6 TCP Wrappers（简单防火墙）

6.1 TCP Wrappers 简介

TCP_Wrappers是一个工作在第四层（传输层）的的安全工具，对有状态连接（TCP）的特定服务进行安全检测并实现访问控制，界定方式是凡是调用libwrap.so库文件的的程序就可以受TCP_Wrappers的安全控制。它的主要功能就是控制谁可以访问，常见的程序有rpcbind、vsftpd、sshd，telnet。

判断方式：

1. 查看对应服务命令所在位置

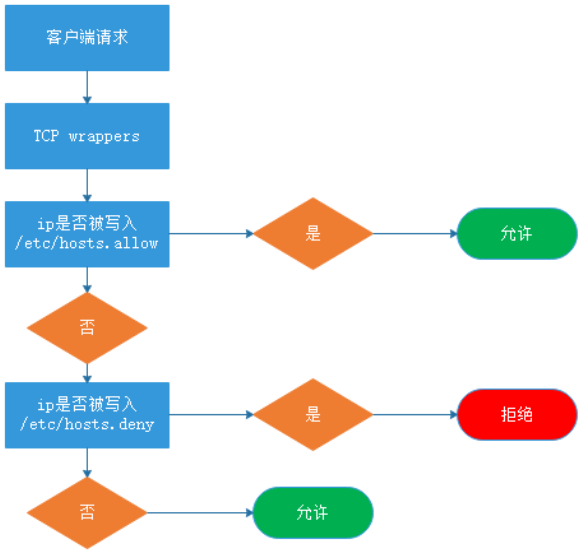
   which sshd

2. 查看指定命令执行时是否调用libwrap.so文件

   ldd /usr/sbin/sshd | grep libwrap.so

6.2 TCP Wrappers 工作原理

以ssh为例，每当有ssh的连接请求时，先读取系统管理员所设置的访问控制文件，符合要求，则会把这次连接原封不动的转给ssh进程，由ssh完成后续工作；如果这次连接发起的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。



1.优先查看 hosts.allow,匹配即停止

2.允许个别，拒绝所有：hosts.allow 文件添加允许的策略，hosts.deny 文件添加 all

3.拒绝个别，允许所有：hosts.allow 文件为空，hosts.deny 文件添加单个拒绝的策略

6.3 TCP Wrappers 的使用

TCP_Wrappers的使用主要是依靠两个配置文件/etc/hosts.allow, /etc/hosts.deny，以此实现访问控制，默认情况下，/etc/hosts.allow，/etc/hosts.deny什么都没有添加，此时没有限制

配置文件编写规则：

```
service_list@host: client_list
```

service_list： 是程序（服务）的列表，可以是多个，多个时，使用，隔开

@host：设置允许或禁止他人从自己的哪个网口进入。这一项不写，就代表全部

client_list：是访问者的地址，如果需要控制的用户较多，可以使用空格或，隔开

格式如下：

基于IP地址： 192.168.88.1 192.168.88.

基于主机名： www.atguigu.com .atguigu.com 较少用

基于网络/掩码： 192.168.0.0/255.255.255.0

内置ACL：ALL(所有主机)、LOCAL(本地主机)

实验案例：

拒绝单个 IP 使用 ssh 远程连接：

配置文件：
hosts.allow：空着
hosts.deny：sshd:192.168.88.20

拒绝某一网段使用 ssh 远程连接：

hosts.allow：空着
hosts.deny：sshd:192.168.88.

仅允许某一 IP 使用 ssh 远程连接：

hosts.allow：sshd:192.168.88.20
hosts.deny：sshd:ALL