3-10 OpenSSH远程控制及TCP Wrappers安全防护 YUM.note20203120005

OpenSSH远程控制及TCP Wrappers安全防护

一、SSH简介

二、OpenSSH的配置

1.服务监听选项

2.用户登录控制

3.登录验证方式

三、使用SSH客户端程序（Linux）

1.命令程序

1.ssh命令（远程登录命令）

2.scp命令（远程安全复制）

3.sftp命令（ 远程文件传输命令）

2.常见远程访问工具（windows）

四、构建密钥对验证的SSH体系

1.在SSH客户端以root用户身份创建密钥对

2.客户端将创建的公钥文件上传至SSH服务端

3.服务端将公钥信息导入用户root的公钥数据库文件

4.客户端以root用户身份连接服务器端root用户测试

五、TCP Wrappers安全防护

1.TCP Wrappers保护原理

2.保护机制的实现方式

3.TCP Wrappers保护的条件

4.访问控制策略的配置文件

5.配置项及格式

1.格式:

2.通配符

OpenSSH远程控制及TCP Wrappers安全防护

一、SSH简介

SSH（Secure Shell）是一种安全通道协议，主要用来实现字符界面的远程登录、远程复制等功能。SSH协其中

议对通信双方的数据传输进行了加密处理，其中包括用户登录时输入的用户口令。

与早期的TELNET（远程登录）、RSH（Remote Shell,远程执行命令）、RCP（Remote File Copy,远程文件复制）等应用相比。SSH协议提供了更好的安全性。

OpenSSH是实现SSH协议的开源软件项目，适用于各种UNIX、Linux操作系统。

SSH协议默认监听端口 ： TCP协议 22

SSH协议版本 ： V1、V2

二、OpenSSH的配置

OpenSSH服务器由openssh(客户端软件)、openssh-server(服务端软件)等软件包提供（默认已安装），并已将sshd添加为标准的系统服务。执行systemctl start sshd命令即可启动sshd服务，包括root在内的大部分用户（只要拥有合法的登录shell）都可以登录系统。

软件包名：openssh、openshh-server

服务名称：sshd

服务端主程序：/usr/sbin/sshd

服务端配置文件：/etc/ssh/sshd_config （守护进程daemon）

客户端配置文件：/etc/ssh/ssh_config

1.服务监听选项

```
[root@abc ~]# vim /etc/ssh/sshd_config
17 Port 22          #监听端口，建议修改为其他端口以提高在网络中的隐蔽性
19 ListenAddress 0.0.0.0        #监听地址，默认监听到0.0.0.0任意地址，修改时必须是本机I地址
20 protocol 2                   #  ssh协议的版本选用V2比V1的安全性更好
116  UseDNS  no            #禁用DNS反向解析，提高服务的响应速度  ----优化
[root@ssh-server ~]# systemctl restart sshd
    [root@ssh-server ~]# netstat -lnpt
    tcp        0      0 192.168.200.103:22      0.0.0.0:*               LISTEN
    2940/sshd
```

2.用户登录控制

sshd服务默认允许root用户登录，但在Internet中使用时是非常不安全的。

普遍做法：先以普通用户远程登入，进入安全 shell环境后，根据实际需要使用su命令切换为root用户。

```
[root@ssh-server ~]# vim /etc/ssh/sshd_config
 38 #LoginGraceTime 2m                    #登录验证时间为2分钟
 39 #PermitRootLogin no                     #禁止root用户登录
41 #MaxAuthTries 6                          #最大重试次数为6次
65 #PermitEmptyPasswords no          #禁止空密码用户登录
```

实验现象：

- 超过10s未输入密码时连接会话会直接断开
- 连续输入3次错误密码时连接会话会直接断开
- 通过root用户登录时反复提示输入密码

- test用户无法登录系统，反复提示输入密码

案例：

```
[root@ssh-server ~]# useradd test
[root@ssh-server ~]# passwd test
更改用户 test 的密码 。
新的 密码：
无效的密码： 密码少于 8 个字符
重新输入新的 密码：
passwd：所有的身份验证令牌已经成功更新。
[root@ssh-server ~]# vim /etc/ssh/sshd_config
PermitRootLogin no
[root@ssh-server ~]# systemctl restart sshd
新建会话创建test用户，登录
[test@ssh-server ~]$ su - root
密码：
[root@ssh-server ~]#
```

OpenSSH服务访问控制

AllowUsers	仅允许用户登录
DenyUsers	仅禁止用户登录

注意：

- AllowUsers不要与DenyUsers 同时使用
- 当服务器在Internet时，控制包含的IP地址时应是公司**公网地址**

```
141 Allowusers test root@192.168.200.128                    # 仅允许test root用户登录
```

其中root用户只能来源于192.168.200.128（VMnet8），即公司的公网地址

```
[root@ssh-server ~]# systemctl restart sshd
```

3.登录验证方式

SSH服务支持两种验证方式：可以只使用其中一种，也可以两种都有

- **密码验证 ---对称加密**

  对服务器中本地系统用户的登录名称、密码进行验证,这种方式使用最为简便

  但从客户端角度来看，正在连接的服务器有可能被假冒；

  从服务器角度来看，当遭遇穷举（暴力破解）攻击时防御能力比较弱

- **密钥对验证 ---非对称加密 #客户端生成公钥私钥，只需配比验证一次即可**

  要求提供相匹配的密钥信息才能通过验证。

  通常先在客户端中创建一对密钥文件（公钥、私钥）

然后将公钥文件放到服务器中的指定位置

远程登录时，系统将使用公钥私钥进行加密/解密关联验证

大大增强了远程管理的安全性。

该方式不易被假冒，且可以免交互登录(不询问密码)，在Shell中被广泛使用

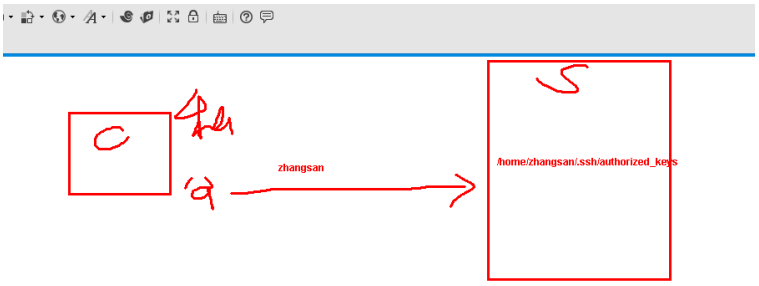
ps:

当密码验证密钥对验证都启用时，服务器优先使用密钥对验证。

对于安全性要求的服务器，建议将密码验证方式禁用，只允许密钥对验证。

没特殊要求，两种验证方式都可。

```
[root@ssh-server ~]# vim /etc/ssh/sshd_config
 66 PasswordAuthentication yes              #启用密码验证
 44 PubkeyAuthentication yes                  #启用密钥对验证
 48 AuthorizedKeysFile      .ssh/authorized_keys      #指定公钥库文件（可以保存多个客户端上传的公钥）
[root@ssh-server ~]# systemctl restart sshd
```



三、使用SSH客户端程序（Linux）

openssh客户端由openssh-clients软件包提供，默认已安装

包括：

ssh ： 远程登录命令

scp ： 远程复制命令

sftp ： 远程文件传输命令

1.命令程序

1.ssh命令（远程登录命令）

格式1：ssh user@host　　# 客户机登录用户与主机用户名相同时，可以省去user@

```
[root@ssh-client ~]# ssh root@192.168.200.103
```

格式2：ssh user@host command

```
[root@ssh-server ~]# ssh 192.168.200.103 touch /tmp/daociyiyou.txt
```

端口选项：-p 22

2.scp命令（远程安全复制）

格式1：scp [-r] file1 user@host:file2

```
[root@ssh-client ~]# scp client.txt 192.168.200.103:/tmp/
```

格式2：scp [-r] user@host:file1 file2

```
[root@ssh-client ~]# scp 192.168.200.103:/root/server.txt /boot/
```

3.sftp命令（远程文件传输命令）

操作1：[root@ssh-client ~]# sftp 192.168.200.103

sftp> get 33 # get ： 下载 put ： 上传

操作2：[root@ssh-client ~]# cd /tmp/

[root@ssh-client tmp]# sftp 192.168.200.103

root@192.168.200.103's password:

Connected to 192.168.200.103.

sftp> get 11

Fetching /root/11 to 11

sftp> exit

[root@ssh-client tmp]# ls

11

操作3：[root@ssh-client ~]# pwd

/root

[root@ssh-client ~]# sftp 192.168.200.103

sftp> ls

sftp> pwd

Remote working directory: /root

sftp> cd /tmp #切换内部环境

sftp> pwd

Remote working directory: /tmp

sftp> cd /root

sftp> lcd /tmp #切换外部环境

sftp> get 22

Fetching /root/22 to 22

sftp> quit

[root@ssh-client ~]# ls /tmp

2.常见远程访问工具（windows）

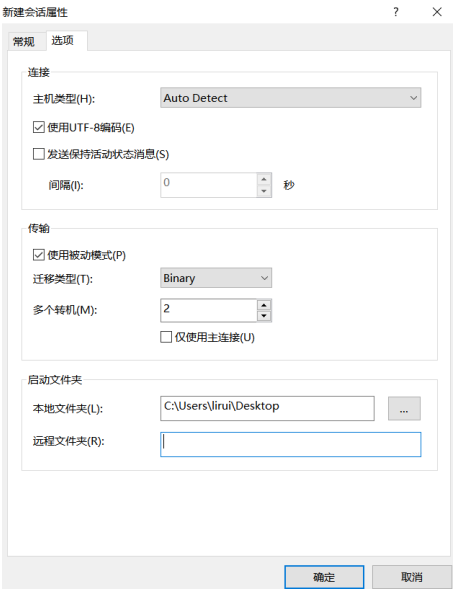
Xshell、SecureCRT # 最常用的，但是都是商业版软件都需要花钱

Putty （最好用英文原版 中文版不安全）

Xmanager （远程图形化界面） 远程连接工具

更换使用远程访问工具，侵权，防止公司被起诉

使用xftp来传输大文件（使用rz fz 传输下载小文件 ）



四、构建密钥对验证的SSH体系

密钥对验证步骤：

1.在SSH客户端以root用户身份创建密钥对

#不一定是root，用户名可以不一样，比如：用张三连李四

命令：ssh-keygen -t ecdsa

加密算法：ECDSA（新出来的，CentOS7才有）　RSA　　DSA

```
[root@ssh-client ~]# ssh-keygen -t ecdsa
    Generating public/private ecdsa key pair.
    Enter file in which to save the key (/root/.ssh/id_ecdsa):
                #保存文件位置，回车默认，需要更换自己敲路径就OK
    Enter passphrase (empty for no passphrase):
                # 暗号，直接回车，不需要，避免交互
    Enter same passphrase again:      # 回车
    Your identification has been saved in /root/.ssh/id_ecdsa.       #私钥
    Your public key has been saved in /root/.ssh/id_ecdsa.pub.       #公钥
    The key fingerprint is:
    SHA256:bqvBl6WoGyFUMM4LHo7jxpbaIRslgRymRtc+AWqurBc root@ssh-client
    The key's randomart image is:
    +---[ECDSA 256]---+
    | +o++            |
    |*o+o o           |
    |=*+ . .          |
    |B+.. o           |
```

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

```
[root@ssh-client ~]# ls -l .ssh/id_ecdsa*
-rw-------. 1 root root 227 3月  12 01:05 .ssh/id_ecdsa
-rw-r--r--. 1 root root 177 3月  12 01:05 .ssh/id_ecdsa.pub
```

新生成的密钥对文件中，id_ecdsa是私钥文件，默认600，对于私钥文件必须妥善保管，不能泄露给他人。id_ecdsa.pub是公钥文件，用来提供给SSH服务器。

2.客户端将创建的公钥文件上传至SSH服务端

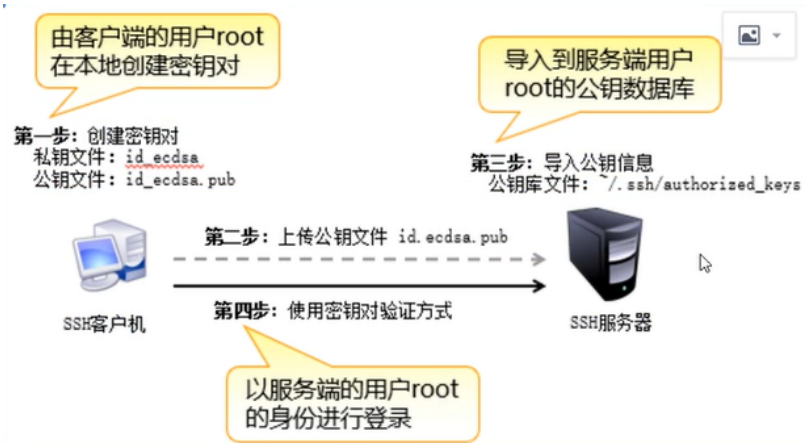
```
[root@ssh-client ~]# scp .ssh/id_ecdsa.pub root@192.168.200.103:/tmp/
root@192.168.200.103's password:
id_ecdsa.pub                                  100%  177     95.9KB/s   00:00
```

3.服务端将公钥信息导入用户root的公钥数据库文件

```
[root@ssh-server ~]# cat /tmp/id_ecdsa.pub
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLiSybYKh8m5mk78Rm31dTxdBEOPLdbMe76tNuXGqIzDsaDgULZjbCuqDCCDBViR8eH9IFBfpCaK3bpflgZzJTw= root@ssh-client
[root@ssh-server ~]# cat /tmp/id_ecdsa.pub >> .ssh/authorized_keys
```

4.客户端以root用户身份连接服务器端root用户测试

```
[root@ssh-client ~]# ssh root@192.168.200.103
Last login: Thu Mar 12 00:59:26 2020 from 192.168.200.104
[root@ssh-server ~]#
```



简便步骤：

```
[root@ssh-client ~]# ssh-keygen -t ecdsa                #客户端生成密钥对
[root@ssh-client ~]# ssh-copy-id root@192.168.200.103        #自动生成并放入
    /usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
    "/root/.ssh/id_ecdsa.pub"
    /usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
    out any
    that are already installed
    /usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
    prompted now it is to install the new keys
    root@192.168.200.103's password:
    Number of key(s) added: 1
    Now try logging into the machine, with:   "ssh 'root@192.168.200.103'"
    and check to make sure that only the key(s) you wanted were added.
[root@ssh-client ~]# ssh 192.168.200.103
    Last login: Thu Mar 12 01:40:03 2020 from 192.168.200.104
```

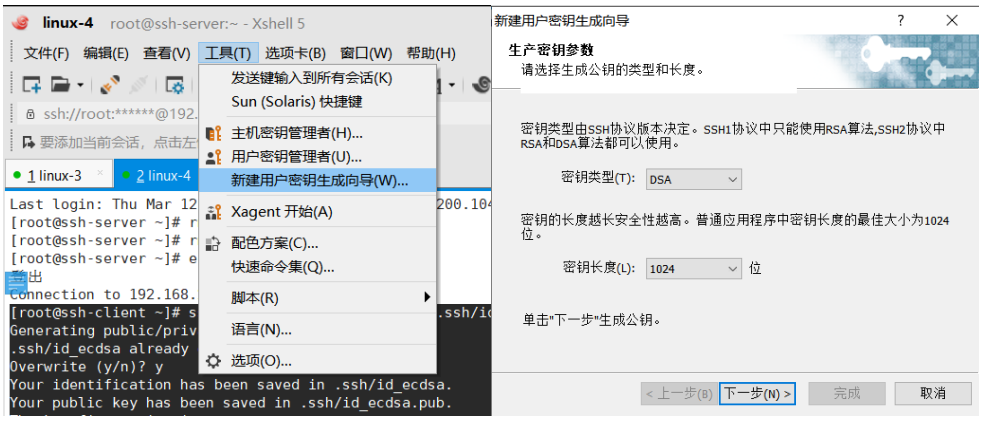
生成密钥免交互的命令：

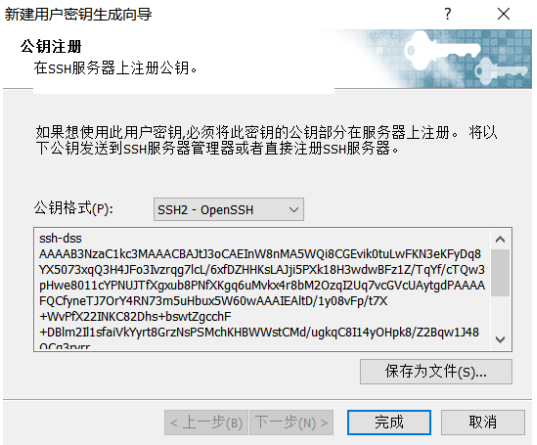
```
[root@ssh-client ~]# ssh-keygen -t ecdsa -P "" -f .ssh/id_ecdsa
    Generating public/private ecdsa key pair.
    .ssh/id_ecdsa already exists.
    Overwrite (y/n)? y
    Your identification has been saved in .ssh/id_ecdsa.
    Your public key has been saved in .ssh/id_ecdsa.pub.
    The key fingerprint is:
    SHA256:5WVD72OoOGd9ZmKSh1GwtUafJSBTi9LYT2DaCohTWRM root@ssh-client
    The key's randomart image is:
    +---[ECDSA 256]---+
    |    .oE.   =.=o= .|
    |   o.. .  B * B +o|
    |  o . .  + * O +..|
    |   .    . * = X   |
    |        S o O O +|
    |         . + = * |
    |          .      |
    |                 |
    |                 |
    +----[SHA256]-----+
[root@ssh-client ~]# ssh 192.168.200.103
```

root@192.168.200.103's password:

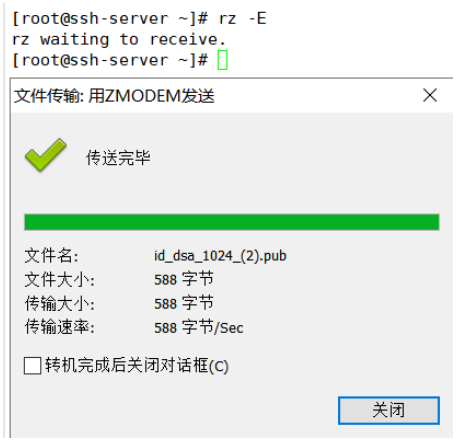
Last login: Thu Mar 12 01:48:59 2020 from 192.168.200.104

使用Windows如何验证密钥对：



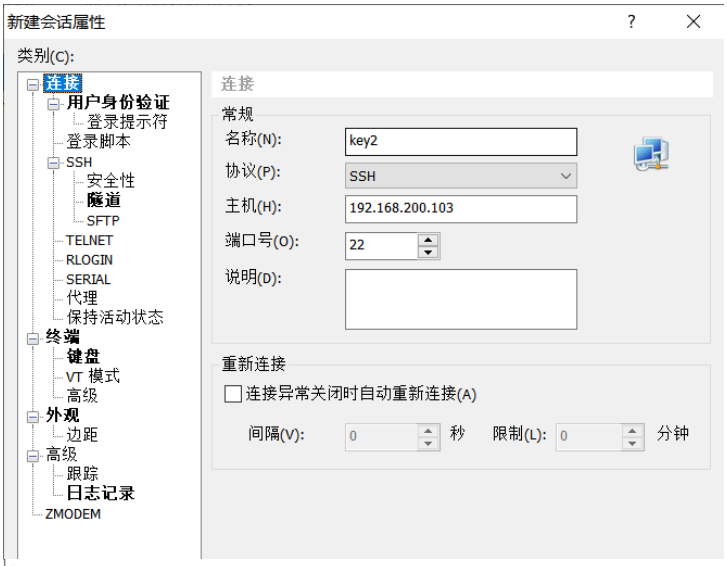


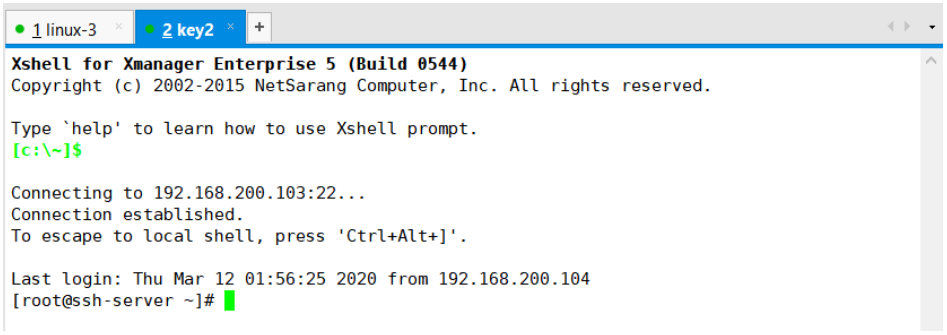
先把公钥保存为一个文件

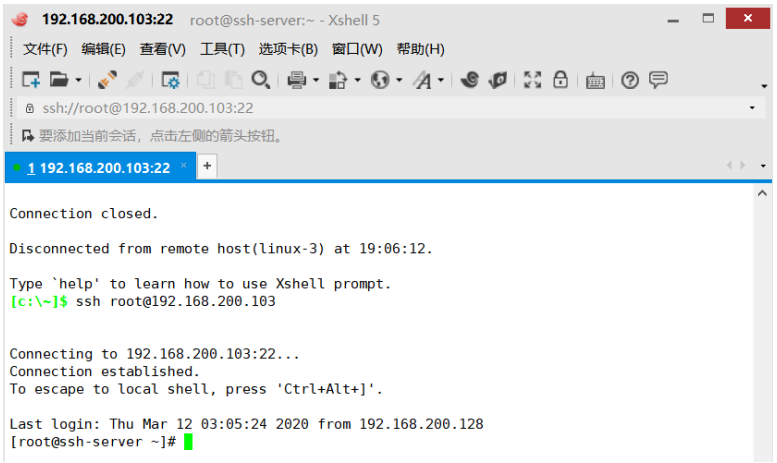


转义删除：

```
[root@ssh-server ~]# rm -rf \id_dsa_1024_(2).pub
bash: 未预期的符号 `(' 附近有语法错误
[root@ssh-server ~]# rm -rf id_dsa_1024_\(\2\)\.pub
[root@ssh-server ~]# cat id_dsa_1024_.pub >> .ssh/authorized_keys
```







五、TCP Wrappers安全防护

TCP外加包装，防止恶意请求破坏协议，安全保障，保护一部分协议，并不是全部

在Linux系统中，许多网络服务针对客户端提供了访问控制机制，如Samba、BIND、HTTPD、OpenSSH等。

TCP Wrappers(TCP封套)防护机制，是作为应用服务于网络之间的一道特殊防线，提供额外的安全保障。

1.TCP Wrappers保护原理

TCP Wrappers将TCP服务程序“包裹”起来

代为监听TCP服务程序的端口，增加了一个安全监测的过程

外来的连接请求必须先通过这层安全检测，获得许可后才能访问真正的服务程序。

2.保护机制的实现方式

1. 通过tcpd主程序对其他服务程序进行包装 ----**用的不多**
2. 由其他服务程序调用**libwrap.so.***链接库 -----**谁调用这个库文件谁就可以获得保护**

3.TCP Wrappers保护的条件

a. 必须是采用TCP协议的服务

b. 函数库中必须包含libwrap.so.0（共享链接库）、大多数服务通过这种方式

ldd命令：

```
[root@ssh-server ~]# ll /usr/sbin/sshd
-rwxr-xr-x. 1 root root 853040 4月  11 2018 /usr/sbin/sshd
[root@ssh-server ~]# ldd /usr/sbin/sshd
        # 查看sshd调用了哪些库文件
    libwrap.so.0 => /lib64/libwrap.so.0 (0x00007f7703eea000)
```

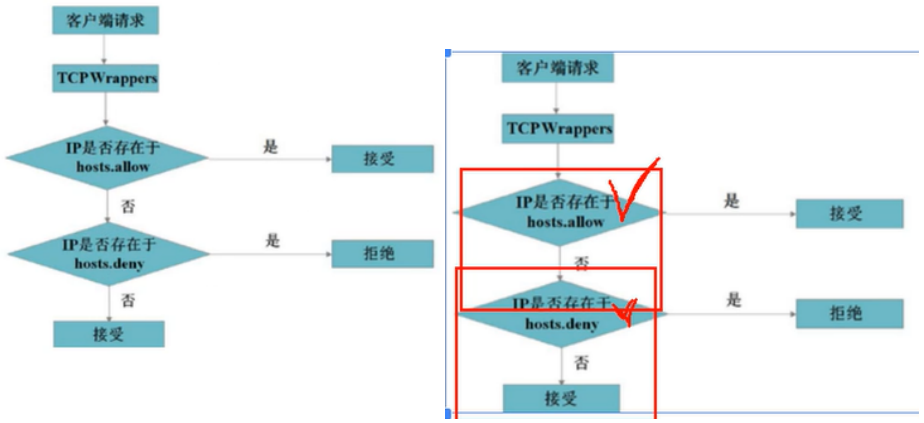
案例：

```
[root@ssh-server ~]# yum -y install vsftpd httpd
[root@ssh-server ~]# ldd /usr/sbin/vsftpd
    libwrap.so.0 => /lib64/libwrap.so.0 (0x00007f36a5877000)
[root@ssh-server ~]# ldd /usr/sbin/httpd
          #没找到wrap库文件，那么httpd 不受TCP Wrappers保护
```

4.访问控制策略的配置文件

```
/etc/hosts.sllow          #允许
/etc/hosts.deny           #拒绝
```

访问控制策略处理流程图



两种选择方案：

允许的多，就在deny里面写几个就可以，用下面的方案；

拒绝的多，就在allow里面写几个，在deny里面写拒绝所有。

ps:

由此可见，/etc/hosts.allow文件的优先级更高，若同一IP地址出现在hosts.allow中，也存在于hosts.deny中，则该IP地址的访问请求将被接收。

5.配置项及格式

1.格式:

服务列表：客户机地址列表

服务列表		客户机地址列表	
多个服务	例：vsftpd,sshd	**多个地址**	例：192.168.1.1,192.168.1.10
所有服务	ALL	**所有地址**	ALL
		通配符?	例：192.168.1.？，192.168.2.？？
		通配符*	例：192.168.1.1*
		网段地址	例：192.168.1.或 192.168.1.1/255.255.255.0
		域名	.crushlinux.com

2.通配符

？表示一位

* 表示任意长度　任意字符串，也可为空

192.168.1*　　1　10 - 19　100-199

192.168.1.2*　　2　20-29　　200-254

3.配置案例

实验要求：

仅允许IP地址为192.168.200.100~192.168.200.199的主机访问sshd服务，禁止其他所有地址的访问。

操作：

```
[root@ssh-server ~]# vim /etc/hosts.allow
sshd:192.168.200.1??
[root@ssh-server ~]# vim /etc/hosts.deny
sshd:ALL
```

多个服务逗号分开，后面的多个主机IP地址也用逗号分开

```
vsftp,sshd:192.168.200.1?,192.168.200.128
```

4.外网服务器配置案例

查看一下密码输入错误的次数，只要超过10次，就把这个地址拦截

```
[root@ssh-server ~]# vim /opt/hosts.deny.sh
#!/bin/bash
#Host.deny Shell Script
#Crushlinux


cat /var/log/secure | awk '/Failed/{print $(NF-3)}'| sort | uniq -c | awk '{print $2"="
```

```
    $1;}'> /tmp/black_ip.txtDEFINE=10
    for i in `cat /tmp/black_ip.txt`

    do

        IP=`echo $i | awk -F = '{print $1}'`
        NUM=`echo $i | awk -F = '{print $2}'`
        if [ $NUM -gt $DEFINE ]

        then

            grep $IP /etc/hosts.deny > /dev/null
            if [ $? -gt 0 ]

            then

                echo "sshd:$IP" >> /etc/hosts.deny
            fi
        fi

    done
[root@ssh-server ~]# bash /opt/hosts.deny.sh
[root@ssh-server ~]# cat /tmp/black_ip.txt | awk -F=' '{print $2,$1}' | sort -nr |
head -20
    58 192.168.200.128
    13 192.168.200.104
[root@ssh-server ~]# cat /etc/hosts.deny
    # hosts.deny    This file contains access rules which are used to
    #               deny connections to network services that either use
    #               the tcp_wrappers library or that have been
    #               started through a tcp_wrappers-enabled xinetd.
    #
    #               The rules in this file can also be set up in
    #               /etc/hosts.allow with a 'deny' option instead.
    #
    #               See 'man 5 hosts_options' and 'man 5 hosts_access'
    #               for information on rule syntax.
    #               See 'man tcpd' for information on tcp_wrappers
    #
```

```
    sshd:ALL
    sshd:192.168.200.104
    sshd:192.168.200.128
[root@ssh-server ~]# cat /tmp/black_ip.txt | more
    192.168.200.104=13
    192.168.200.128=58
```

怎么拦截的呢？主要查看了：

```
[root@ssh-server ~]# ll /var/log/secure
-rw-------. 1 root root 80049 3月  12 04:19 /var/log/secure
[root@ssh-server ~]# du -sh /var/log/secure
80K     /var/log/secure
[root@ssh-server ~]# ll /etc/hosts.deny
-rw-r--r--. 1 root root 511 3月  12 04:49 /etc/hosts.deny
```

一、YUM简介

- YUM是软件管理仓库
- 可以完成安装，卸载（最好别用），自动升级rpm软件包等任务
- 能够自动查找并解决rpm包之间的依赖关系
- 并一次安装所有依赖的相关软件包
- 无需管理员逐个，手工的去安装每个rpm包，使管理员在维护大量Linux服务器时更加轻松自如
- 在拥有大量Linux主机的本地网络中，在自己公司架设一台YUM服务器还可以缓解软件安装、升级等对Internet的依赖

YUM的前世今生~

a. YUM前身YUP（Yellow dog Updater,Yellow dog linux的软件更新器），

b. 最初由TSS公司（Terra soft Solutions , INC.）使用Python语言开发，

c. 后来杜克大学（Duke University）的Linux开发团队进行改进，

d. 命名为YUM（Yellow dog Updater,Modified）.

CentOS8里面DNF

源服务器：

- 源服务器是包含存放各种rpm安装包的软件仓库（Repository）和仓库数据（Repodata）
- 仓库数据是用来收集仓库目录下rpm软件包的头部信息的
- YUM软件仓库借助于HTTP或FTP协议进行发布
- 可以面向网络中的所有服务器，使其他Linux服务器（客户机）直接调用，而无须自己准备软件包。

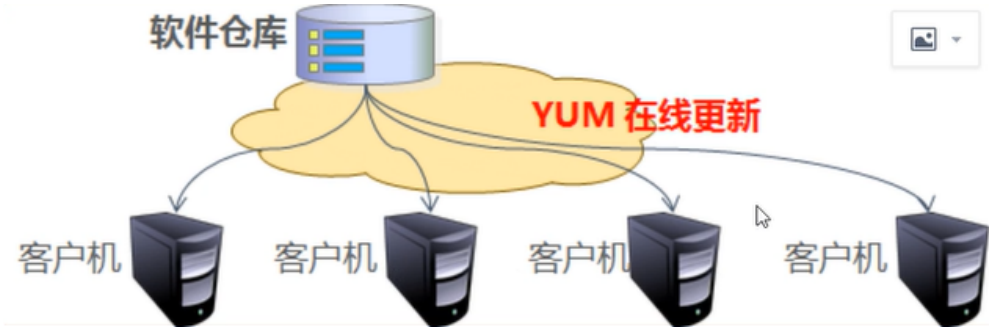
RPM软件包的来源

CentOS发布的RPM包集合

第三方组织发布的RPM包集合

用户自定义的RPM包集合

在CentOS7系统的安装光盘中，已针对目录Repository建立好Repodata数据，
因此只要简单地将整个光盘中的内容通过HTTP或FTP进行发布，就可以软件仓库了。



1.配置基于HTTP协议的YUM源服务器

```
[root@yum-server ~]# rpm -q httpd
    httpd-2.4.6-80.el7.centos.x86_64
        # 如果没安装httpd,先安装在进行下面的步骤
[root@yum-server ~]# umount /dev/cdrom
[root@yum-server ~]# mkdir /var/www/html/centos
[root@yum-server ~]# mount /dev/cdrom /var/www/html/centos
    mount: /dev/sr0 写保护，将以只读方式挂载
[root@yum-server ~]# ls /var/www/html/centos
    CentOS_BuildTag  GPL       LiveOS    RPM-GPG-KEY-CentOS-7
    EFI              images    Packages  RPM-GPG-KEY-CentOS-Testing-7
    EULA             isolinux  repodata  TRANS.TBL
[root@yum-server ~]# systemctl start httpd
[root@yum-server ~]# systemctl enable httpd          # 开机自启动
    Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service
    to /usr/l
    ib/systemd/system/httpd.service.
```

==========源服务器搭完了

```
[root@web yum.repos.d]# cd /etc/yum.repos.d/
[root@web yum.repos.d]#  ls
    backup  ComsenzDiscuz-DiscuzX-master.zip  DiscuzX  httpd-2.4.37.tar.gz
    local.repo
[root@web yum.repos.d]# vim local.repo
    #baseurl=file:///media/cdrom
    baseurl=http://192.168.200.103/centos
[root@web yum.repos.d]#  iptables -F
```

访问测试：http://192.168.200.103/centos/

安装测试：

```
[root@web yum.repos.d]#  yum clean all
[root@web yum.repos.d]#  yum -y install samba bind
```

2.配置基于本地的YUM源服务器

1.挂载光盘

2.配置仓库文件

```
[root@yum-server ~]#vim /etc/yum.repos.d/local.repo
    [amy]
    name=amy
    #baseurl=file:///media/cdrom
    baseurl=http://192.168.200.103/centos
    gpgcheck=1
    enable=1
    gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

3.配置基于FTP协议的YUM源服务器

```
[root@yum-server ~]# umount /dev/cdrom
[root@yum-server ~]# rpm -q vsftpd
    vsftpd-3.0.2-22.el7.x86_64
[root@yum-server ~]# mkdir /var/ftp/centos
[root@yum-server ~]# mount /dev/cdrom /var/ftp/centos
    mount: /dev/sr0 写保护，将以只读方式挂载
[root@yum-server ~]# ls /var/ftp/centos
    CentOS_BuildTag  GPL       LiveOS    RPM-GPG-KEY-CentOS-7
    EFI              images    Packages  RPM-GPG-KEY-CentOS-Testing-7
    EULA             isolinux  repodata  TRANS.TBL
```

```
[root@yum-server ~]# systemctl start vsftpd
[root@yum-server ~]# systemctl enable vsftpd
    Created symlink from /etc/systemd/system/multi-user.target.wants/vsftpd.service
    to /usr/
    lib/systemd/system/vsftpd.service.
```

4.修改客户端的yum文件---客户机指定YUM服务

```
[root@web ~]# cd /etc/yum.repos.d/
[root@web yum.repos.d]#  mkdir backup
[root@web yum.repos.d]#  mv *.repo backup
```

*.repo指定的软件源有效，要使用首先要联网，要找互联网上别人做好的服务器，而这些服务器多数在国外，用起来比较慢且效率低；放到backup文件夹里面，隐藏起来，让它找不到否则只要在/etc/yum.repos.d/下面，就会生效。

```
[root@web yum.repos.d]# vim local.repo
    [gabe]                                                          #仓库类别
    name=gabe                                                       #仓库名称（说明，
    描述）
    #baseurl=file:///media/cdrom
    baseurl=ftp://192.168.200.111/centos          #软件包URL访问路径
    gpgcheck=1                                                      #验证软件包的公
    钥
    enable=1                                                        #启用此仓库
    （默认启用状态）
    gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7       #软件包GPG公钥文件位置
```

5.手动创建软件包仓库

yum源：192.168.200.105

```
[root@yum-server other]# rz -E
[root@yum-server other]# ls
[root@yum-server other]# createrepo -g /var/ftp/centos/repodata/repomd.xml ./
#以现有的repodata目录为样板
[root@yum-server other]# ls
                        repodata
[root@yum-server other]# systemctl restart vsftpd
```

WEB:192.168.200.106

```
[root@web yum.repos.d]# vim local.repo
```

```
[sofia]
name=sofia
#baseurl=file:///media/cdrom
#baseurl=http://192.168.200.105/centos
baseurl=ftp://192.168.200.105/centos
gpgcheck=0
enable=1

[other]
name=other
baseurl=ftp://192.168.200.105/other
gpgcheck=0
enabled=1
```

```
[root@web yum.repos.d]# yum makecache
已加载插件：fastestmirror, langpacks
Loading mirror speeds from cached hostfile
other                                                    | 3.6 kB
00:00:00
sofia                                                    | 3.6 kB
00:00:00
(1/6): other/filelists_db                                |  16 kB
00:00:00
(2/6): other/primary_db                                  |  19 kB
00:00:00
(3/6): other/group_gz                                    | 1.1 kB
00:00:00
(4/6): other/other_db                                    | 6.6 kB
00:00:00
(5/6): sofia/other_db                                    | 1.3 MB
00:00:00
(6/6): sofia/filelists_db                                | 3.1 MB
00:00:00
元数据缓存已建立
==========================================================
==========================================================
```

6.CentOS7 更换互联网yum源

搜狐的源：

http://mirrors.sohu.com/help/CentOS-Base-sohu.repo

网易的源：

http://mirrors.163.com/.help/CentOS6-Base-163.repo

阿里云的源：

http://mirrors.aliyun.com/repo/Centos-7.repo

```
[root@localhost ~]# yum -y install wget
[root@localhost ~]# cd /etc/yum.repos.d/ ; mkdir backup
[root@localhost ~]# mv ./*.repo backup
[root@localhost ~]# wget -O /etc/yum.repos.d/CentOS-Base.repo
http://mirrors.aliyun.com/repo/Centos-7.repo
[root@localhost ~]# yum clean all && yum makecache
```

7.小结：各种yum源的配置方法

1、本地yum源（只能自己可以用）

2、基于HTTP或FTP协议发布yum源

3、指定互联网的镜像站（http://mirrors.aliyun.com/）

方法一：将连接地址粘到自己写的local.repo

方法二：将.repo下载下来，直接在网站复制

```
[root@web yum.repos.d]# wget https://mirrors.aliyun.com/repo/Centos-7.repo
[root@web yum.repos.d]# ls
```

PS：

这里指定的互联网上的yum源基本与光盘里的内容是差不多的，但是现在很多开发人员开发的很多新型的软件，是没有被集成到光盘镜像里面的，它们很多都被放到了EPEL源里面。

4、配置epel源

```
[root@web yum.repos.d]#rpm -ivh https://mirrors.aliyun.com/epel/epel-release-latest-7.noarch.rpm
[root@web yum.repos.d]# ls
backup  Centos-7.repo  epel.repo  epel-testing.repo                    #epel源
```

主机可以安装额外的软件包

8.中国大陆开源镜像站汇总

企业贡献	
搜狐开源镜像站	http://mirrors.sohu.com/
网易开源镜像站	http://mirrors.163.com/
阿里开源镜像站	https://mirrors.aliyun.com/centos/

大学教学	
北京理工大学	http://mirror.bit.edu.cn (IPv4 only) http://mirror.bit6.edu.cn (IPv6 only)
北京交通大学	http://mirror.bjtu.edu.cn (IPv4 only) http://mirror6.bjtu.edu.cn (IPv6 only) http://debian.bjtu.edu.cn (IPv4+IPv6)
兰州大学	http://mirror.lzu.edu.cn/
厦门大学	http://mirrors.xmu.edu.cn/
上海交通大学	http://ftp.sjtu.edu.cn/ (IPv4 only) http://ftp6.sjtu.edu.cn (IPv6 only)
清华大学	http://mirrors.tuna.tsinghua.edu.cn/ (IPv4+IPv6) http://mirrors.6.tuna.tsinghua.edu.cn/ (IPv6 only) http://mirrors.4.tuna.tsinghua.edu.cn/ (IPv4 only)
清华大学学生网管会	http://mirrors.tuna.tsinghua.edu.cn/
天津大学	http://mirror.tju.edu.cn/
中国科学技术大学	http://mirrors.ustc.edu.cn/ (IPv4+IPv6) http://mirrors4.ustc.edu.cn/ http://mirrors6.ustc.edu.cn/
西南大学	http://linux.swu.edu.cn/swudownload/Distributions/
东北大学	http://mirror.neu.edu.cn/ (IPv4 only) http://mirror.neu6.edu.cn/ (IPv6 only)
电子科技大学	http://ubuntu.uestc.edu.cn/
青岛大学	http://mirror.qdu.edu.cn/
重庆大学	http://mirrors.cqu.edu.cn/
开源中国社区	http://mirrors.oss.org.cn/
大连东软信息学院	http://mirrors.neusoft.edu.cn/
华中科技大学	http://mirror.hust.edu.cn/
中山大学	http://mirror.sysu.edu.cn/
浙江大学	http://mirrors.zju.edu.cn/
台湾淡江大学	http://ftp.tku.edu.tw/Linux/
Linux 运维派	https://mirrors.skyshe.cn/
CentOS官方镜像列表	http://mirror-status.centos.org/
CentOS全系列下载地址	http://www.centoscn.com/CentosSoft/iso/

Linux各个系列下载地址	http://www.centoscn.com/CentosSoft/linux/

二、yum命令手册

1.选项

-h = --help

-y = 回答所有的问题为yes

-q = 不显示安装信息，写脚本的时候会用到

2.命令

yum clean all

yum makecache

yum list -查看软件包列表

yum install

yum grouplist install--按照包组来安装，一系列软件包

yun info --查看软件信息

卸载命令 不要用下面的命令！！！！

earse=remove

yum remove package

yum erase package

yum groupremove group

误操作！！！不要信百度！！！信亮哥！！

yum update 更新 最好别用，针对性的升级，yunm update httpd ，一个一个去升级

yum groupupdate group 升级软件包组group

yum clean

yum makecache fast 快速重建缓存

yum localinstall 安装本地软件包

yum -y reinstall 重新安装

面试题

yum deplist packages　　#查看软件包的依赖关系

yum provides +绝对路径

[root@web yum.repos.d]# cd

[root@web ~]# which mkdir

/usr/bin/mkdir

[root@web ~]# rpm -qf /usr/bin/mkdir　　#rpm查，需要系统已经安装了软件包

coreutils-8.22-21.el7.x86_64

[root@web ~]# yum provides /usr/bin/mkdir　　#yum 查，没安装软件包只要在仓库里也能查

3.yum的客户端主配置文件

[root@web ~]# vim /etc/yum.conf

[main]

cachedir=/var/cache/yum/$basearch/$releasever　　#缓存目录

keepcache=1　　# 0表示不保留缓存，1表示软件包下载完不会自动清理

[root@web ~]# ls /var/cache/yum/x86_64/7/epel/packages/

[root@web ~]# ls /var/cache/yum/x86_64/7/base/packages/

[root@web ~]# ls /var/cache/yum/x86_64/7/extras/packages/

保留安装后的软件包

方法一：

keepcache=1，当我们安装软件包的时候，把rpm包都留下来，可以防止以后安装软件时没网络也能安装它，或者也可以加速，现场下载比较慢。

方法二：

[root@web ~]# mkdir salt-master

[root@web ~]# yum -y --downloadonly --downloaddir=salt-master/ install salt-master

下载存在依赖关系的软件包

这个方法更好，方法一的包都分散在不同的目录里面，这个都指定在一个目录下

例如：

[root@web ~]# yum -y --downloadonly --downloaddir=php/ install php php-devel php-mysql