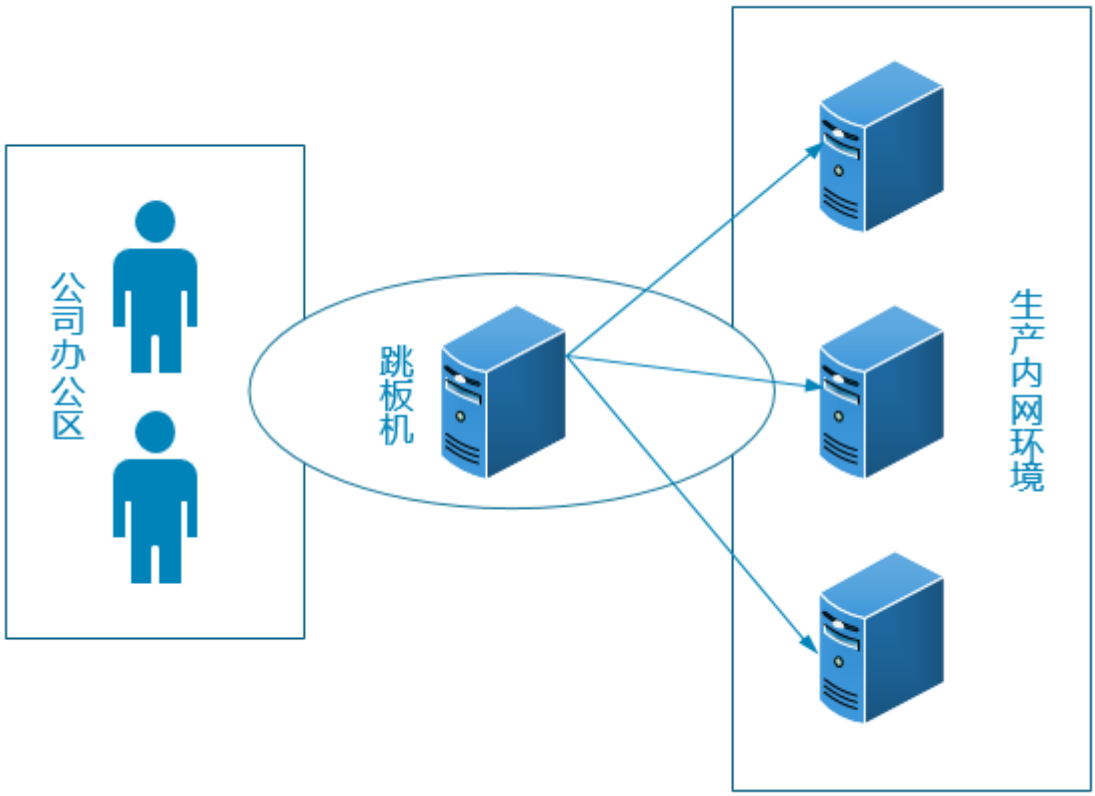
任务背景

为了最大程度的保护公司内网服务器的安全，公司内部有一台服务器做跳板机。运维人员在维护过程中首先要统一登录到这台服务器，然后再登录到目标设备进行维护和操作。由于开发人员有时候需要通过跳板机登录到线上生产环境查看一些业务日志，所以现在需要运维人员针对不同的人员和需求对账号密码进行统一管理，并且遵循**权限最小化**原则。



任务要求

1. 跳板机上为每个开发人员创建一个账号，并且只能在指定的目录里管理自己的文件，不能删除别人的文件。（跳板机完成）
2. 线上生产服务器，禁止使用root用户远程登录。（生产服务器完成）
3. 线上生产服务器sshd服务不允许使用默认端口，防止黑客进入端口扫描。（生产服务器完成）
4. 线上生产服务器的业务用户的密码使用工具随机生成。（生产服务器完成）

课程目标

- 了解ssh服务的认证方式
- 能够禁止root远程登录
- 能够更改ssh服务的默认端口
- 熟练使用相关客户端工具，如ssh远程登录，scp远程拷贝文件

涉及知识点

- 用户权限管理（旧知识点）

- ssh服务配置（新知识点）
- 生成随机密码工具（新知识点）

理论储备

一、什么是服务

- 运行在操作系统后台的一个或者多个程序，为系统或者用户提供特定的服务
- 可靠的，并发的，连续的不间断的运行，随时接受请求
- 通过交互式提供服务

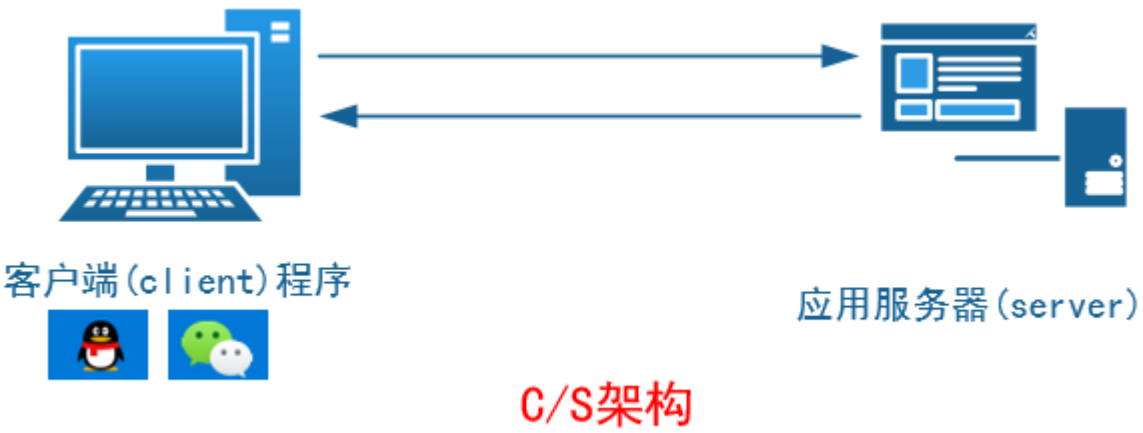
二、服务器架构模型

- B/S(browser/server) 浏览器/服务器

概念：这种结构用户界面是完全通过浏览器来实现，使用http协议 优势：节约开发成本

- C/S（client/server）客户端/服务器

概念：指的是客户端和服务端之间的通信方式，客户端提供用户请求接口，服务端响应请求进行对应的处理，并返回给客户端 优势：安全性较高，一般面向具体的应用



两者区别：

B/S： 1、广域网，只需要有浏览器即可 2、一般面向整个互联网用户，安全性低 3、维护升级简单

C/S： 1、专用网络、小型局域网，需要具体安装对应的软件 2、一般面向固定用户，安全性较高

思考1:

我们通过网络是如何找到我们想要访问的服务的?

IP(提供服务的服务器)+Port(找到相应的服务)

三、端口号设定

说明:端口号只有整数，范围是从0 到65535 • 1～255：**一般是知名端口号**，如:ftp 21号、web 80、ssh 22、telnet 23号等 • 256～1023：通常都是由Unix系统占用来提供特定的服务 • 1024~5000：客户端的临时端口，随机产生 • 大于5000：为互联网上的其他服务预留

思考2:

如何查看系统默认的注册端口?

```
/etc/serivces
```

四、常见的网络服务

- 文件共享服务：FTP、SMB、NFS、HTTP
- 域名管理服务：DNS
- 网站服务：Apache(httpd)、Nginx、Lighttpd、IIS
- 邮件服务: Mail
- 远程管理服务：SSH、telnet
- 动态地址管理服务:DHCP

五、SSH服务概述

1. SSH介绍

- SSH是Linux下远程管理的工具，相比Telnet安全，运维人员必备的神器!
- SSH的全称Secure Shell，安全的shell，是Client/Server架构，默认端口号为22，TCP/IP协议
- SSH其实用于商业，而OpenSSH即为开源的，在Linux中默认安装
- SSH有v1和v2版本
  - ssh v1：有漏洞，容易受到攻击
  - ssh v2：通过公钥加密（数字签名和密钥交换）的方式进行，确保服务器端的身份识别

2. SSH加密算法

- des 对称的公钥加密算法,安全低，数据传输速度快；使用同一个秘钥进行加密或解密
- rsa 非对称的公钥加密算法,安全,数据传输速度慢 ， SSH默认的加密算法

3. SSH认证方式

- 基于用户密码的认证
- 基于秘钥对的认证（免密码登录）

六、SSH服务配置

掌握搭建所有服务的思路：

1. 关闭防火墙和selinux（实验环境）
2. 配置yum源 为了安装软件
3. 软件三步曲
    - 安装软件
    - 确认成功安装
    - 查看软件列表　配置文件、启动脚本、二进制命令
4. 了解配置文件 man 5 xxx.conf
5. 根据需求通过修改配置文件完成服务的搭建
6. 启动服务（开机自启动） 端口处于监听状态
7. 测试验证

```
1. 关闭防火墙和selinux
略
2. 配置yum源
本地yum源（略）
3. 软件三部曲
1）安装软件
[root@testdb ~]# yum list|grep openssh
openssh.x86_64                          5.3p1-94.el6                    @anaconda-CentOS-
201311272149.x86_64/6.5
openssh-clients.x86_64                  5.3p1-94.el6                    @anaconda-CentOS-
201311272149.x86_64/6.5
openssh-server.x86_64                   5.3p1-94.el6                    @anaconda-CentOS-
201311272149.x86_64/6.5
openssh-askpass.x86_64                  5.3p1-94.el6                    local
openssh-ldap.x86_64                     5.3p1-94.el6                    local
通过查看，确定openssh-server 和openssh-clients软件包已经安装
2）确认成功安装
3）查看软件列表
openssh-server
openssh-clients
[root@testdb ~]# rpm -ql openssh-server
/etc/rc.d/init.d/sshd       服务启动脚本
/etc/ssh/sshd_config        服务的配置文件
/usr/sbin/sshd              二进制的命令(程序本身)
/usr/share/man/man5/sshd_config.5.gz        配置文件的man文档
/usr/share/man/man8/sftp-server.8.gz
/usr/share/man/man8/sshd.8.gz

[root@testdb ~]# rpm -ql openssh-clients
/etc/ssh/ssh_config         客户端的配置文件
/usr/bin/scp                /usr/bin/一些客户端工具，命令
/usr/bin/sftp
/usr/bin/slogin
/usr/bin/ssh
/usr/bin/ssh-add
```

```
/usr/bin/ssh-agent
/usr/bin/ssh-copy-id
/usr/bin/ssh-keyscan

4.了解配置文件
1）先查看配置文件（语法，选项）
2）再通过man文档查找不知道的选项
man 5 sshd_config

5. 根据需求通过修改配置文件来完成服务搭建
没有需求，不需要更改任何配置文件
6. 启动服务（开机自启动）
默认系统自动启动的
[root@testdb ~]# chkconfig --list|grep ssh
sshd              0:off 1:off 2:on  3:on  4:on  5:on  6:off
[root@testdb ~]# netstat -nltp|grep ssh
tcp        0      0 0.0.0.0:22                  0.0.0.0:*                   LISTEN
 1029/sshd
tcp        0      0 :::22                       :::*                        LISTEN
 1029/sshd
[root@testdb ~]# /etc/init.d/sshd status
openssh-daemon (pid  1029) is running...

7.测试验证

[root@node1 ~]# ssh 10.1.1.2
The authenticity of host '10.1.1.2 (10.1.1.2)' can't be established.
RSA key fingerprint is 9e:5e:e5:e1:c0:4a:3c:37:f2:f5:c8:a0:76:3c:36:3e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.1.1.2' (RSA) to the list of known hosts.
root@10.1.1.2's password:
Last login: Sun Nov  4 23:04:07 2018 from 10.1.1.254
[root@testdb ~]#
```

任务解决方案

1. 创建用户并授权

```
创建用户：code1~code3
创建code组和相应用户：
[root@jumper-server ~]# groupadd code
[root@jumper-server ~]# useradd code1 -G code
[root@jumper-server ~]# useradd code2 -G code
[root@jumper-server ~]# useradd code3 -G code

使用非交互式设置密码：
[root@jumper-server ~]# echo 123|passwd --stdin code1
Changing password for user code1.
[root@jumper-server ~]# echo 123|passwd --stdin code2
Changing password for user code2.
```

```
passwd: all authentication tokens updated successfully.
[root@jumper-server ~]# echo 123|passwd --stdin code3
Changing password for user code3.
passwd: all authentication tokens updated successfully.
```

为开发人员创建数据目录：

```
[root@jumper-server ~]# mkdir /data/code -p
[root@jumper-server ~]# ll -d /data/code/
drwxr-xr-x 2 root root 4096 Nov  6 15:16 /data/code/
```

依据权限最小化原则来更改权限：

```
[root@jumper-server ~]# setfacl -m g:code:rwx  /data/code/
[root@jumper-server ~]# chmod o+t /data/code/
```

测试验证：

略

2. 禁止root远程登录

```
# vim /etc/ssh/sshd_config
#PermitRootLogin yes
PermitRootLogin no
```

重启服务让配置生效

```
[root@app1 ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
```

测试验证：

```
[root@jumper-server ~]# ssh root@10.1.1.2
root@10.1.1.2's password:
Permission denied, please try again.
root@10.1.1.2's password:
Permission denied, please try again.
root@10.1.1.2's password:
Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

3. 更改默认端口

将默认的22号端口更改为10022

思路：

1. 查看在当前服务器中10022是否被使用

```
netstat -a|grep 10022
ss -a|grep 10022
lsof -i 10022
grep 10022 /etc/services
```

2. 修改配置文件

```
[root@app1-server ~]# vim /etc/ssh/sshd_config
#Port 22
Port 10022
```

```
[root@app1-server ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
```

3. 测试验证

1）在线上环境创建创建业务用户pos

```
[root@app1 ~]# useradd pos
[root@app1 ~]# echo 123|passwd --stdin pos
Changing password for user pos.
passwd: all authentication tokens updated successfully.
[root@app1 ~]# netstat -nltp|grep sshd
tcp        0      0 0.0.0.0:10022               0.0.0.0:*                   LISTEN
 2241/sshd
tcp        0      0 :::10022                    :::*                        LISTEN
 2241/sshd
```

2）跳板机测试远程访问

```
[root@jumper-server ~]# ssh -lpos 10.1.1.2
ssh: connect to host 10.1.1.2 port 22: Connection refused
[root@jumper-server ~]# ssh -lpos 10.1.1.2 -p10022
pos1@10.1.1.1's password:
Last login: Sun Aug 26 15:08:20 2018 from 10.1.1.1
```

说明：

如果更改了端口，那么远程连接时必须指定端口号。

补充：

更改客户端配置文件，不想验证指纹：

```
# vim /etc/ssh/ssh_config
#   StrictHostKeyChecking ask
   StrictHostKeyChecking no
```

4. 业务用户密码随机

线上业务用户密码随机生成

环境：

当前服务器不能上外网，需要安装pwgen软件本地光盘里没有。

解决：

jumper-server可以访问互联网，那么配置一个epel源，直接下载并安装pwgen工具

注意：

修改/etc/yum.conf文件文件

keepcache=1   修改0为1，表示缓存下载的软件包到本地。

epel源：

```
[root@jumper-server yum.repos.d]# cat epel.repo
[epel]
name=xxx
baseurl=http://mirrors.aliyun.com/epel/6/x86_64/
enabled=1
gpgcheck=0
```

```
# yum -y install pwgen
```
找到下载的软件包:
```
[root@jumper-server yum.repos.d]# ls /var/cache/yum/x86_64/6/epel/packages/
pwgen-2.08-1.el6.x86_64.rpm
```

拷贝该软件到生产服务器:
```
# scp /var/cache/yum/x86_64/6/epel/packages/pwgen-2.08-1.el6.x86_64.rpm 10.1.1.2:/tmp
```

生产服务器安装使用即可
```
# rpm -ivh /tmp/pwgen-2.08-1.el6.x86_64.rpm
```

命令的使用:
```
# pwgen --help
Usage: pwgen [ OPTIONS ] [ pw_length ] [ num_pw ]
```
参数:
```
-c or –capitalize
```
密码中至少包含一个大写字母
```
-A or –no-capitalize
```
密码中不包含大写字母
```
-n or –numerals
```
密码中至少包含一个数字
```
-0 or –no-numerals
```
密码中不包含数字
```
-y or –symbols
```
密码中至少包含一个特殊符号
```
-s or –secure
```
生成完全随机密码
```
-B or –ambiguous
```
密码中不包含模糊字符（如1,l,O,0）
```
-C
```
在列中打印生成的密码
```
-1
```
不要在列中打印生成的密码，即一行一个密码

```
#pwgen -cnBs1 10 3
```
生成长度为10，包含大写、数字、不包含模糊字符完全随机的3个密码

```
[root@app1 tmp]# pwgen -cnBs1 10 5
J9xREhKP9b
CJXoJffW4F
nTi4mAzjPx
UiJTiVcz9i
xTFK7KLbRc

[root@app1 tmp]# echo 'J9xREhKP9b' |passwd --stdin pos
Changing password for user pos.
passwd: all authentication tokens updated successfully.
```
注意:
在实际的工作中，以shell脚本形式定期修改用户的密码，并且用户名和密码对应会保存到一个密码文件里!

补充扩展

1. ssh客户端工具

```
ssh --help
man ssh

[user@]hostname [command]

[code1@jumper-server ~]$ ssh pos@10.1.1.2
pos@10.1.1.2's password:
Last login: Mon Nov  5 03:33:48 2018 from 10.1.1.1
[pos@app1 ~]$ exit
logout
Connection to 10.1.1.2 closed.
[code1@jumper-server ~]$ ssh -lpos -p10022 10.1.1.2
pos@10.1.1.2's password:
Last login: Mon Nov  5 03:51:16 2018 from 10.1.1.1
[pos@app1 ~]$
[pos@app1 ~]$ exit
logout
Connection to 10.1.1.2 closed.
[code1@jumper-server ~]$ ssh -lpos -p10022 10.1.1.2 date
pos@10.1.1.2's password:
Mon Nov  5 03:52:35 CST 2018

-l: 指定连接用户
-p: 指定端口

注意:
当前本机用户是root，如果不指定连接用户，那么会让你输入远程主机的root密码。
如果指定了连接用户pos1，那么不管你当前是什么用户就只让你输入远程主机的pos1用户密码。
```

2. scp客户端工具

```
cp 本地文件的拷贝
scp 远程文件拷贝
用法:
将本地文件拷贝到远程:
scp 需要拷贝的文件  远程服务器
scp file1 pos1@10.1.1.1:/tmp/
scp -r dir1 pos1@10.1.1.1:/tmp

将远程文件拷贝到本地:
scp 远程文件  本地路径
scp -r pos1@10.1.1.1:/tmp/dir1 /data/code

-r:递归拷贝目录
-P:指定远程服务器的端口
# scp -P10022 1.txt pos@10.1.1.2:/tmp
```