**X 管理服务器部署**

包括ssh-keygen

**ssh-keygen实现无密登录与批量分发**

上次总结时说到在ubuntu下使用teminal, ssh远程主机，但是每次都都需要输密码，今天就介绍下如何使用ssh而不需要每次都输密码，并且还可以实现批量分发（结合rsync或者ssh本身自带的scp）费话少说，let’s go。

ssh (sercure shell protocol) 是什么？   
ssh 是专为远程登录会话和其他网络服务提供的安全协议，利用ssh协议可以有效的防止远程中信息的泄露。

远程连接有两种方式

1. 基于口令的安全验证， 只要知道远程主机的贴与口令，就可以登录远程主机，
2. 基于密匙的安全验证，使用密匙生成口令后每次登录目标主机都不需要再输密码，SSH密钥总是成双出现，一把公钥，一把私钥。公钥可以自由放在所需要连接的服务器上，而私钥必须稳妥保管好。在登录的时候，远程主机会向用户发送一段随机字符串（1024beytes) 用户用自己的私钥加密后，再发回来，远程主机用事先储存的公钥进行解密，如果成功，就证明用户是可信的。允许登录。这样做的好处就是不受中间人攻击。

**ssh keygen 密钥生成**

我们使用ssh-keygen生成密匙。下面简单介绍下命令的语法

命令描述：ssh-keygen主要用来生成rsa keys （ssh 1） 或者生成ecdsa /rsa keys（ssh2).第一次输入命令会在家目录里创建一个隐藏目录**.ssh**并且在下面生成密钥文件，一个公钥一个密钥。 私钥名为**~/.ssh/identity, ~/.ssh/ecdsa, ~/.ssh/id\_dsa, ~/.ssh/id\_rsa** 其中一种这取决于ssh的版本，也可以使用-f来指定生成的文件名。公钥的文件名会在密钥文件名后加**.pub**

参数：

* -b bits, 指定生成密钥的（768-2048）bite，默认是2048,
* -C comment 增加备注
* -f filename, 指定密钥文件名
* -t type, 指定密钥的类型 rsa1 (ssh1); dsa ecdsa rsa (ssh2)

玩github的朋友可能已经注意到上传到github时可以选择ssh方式，而且官网还有一个专门的教程[Generating SSH keys](https://help.github.com/articles/generating-ssh-keys/" \t "_blank)其讲解基本与下面内容相似。

首先在客户端，生成密钥

[sandow@NFS01 ~]$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/sandow/.ssh/id\_rsa):   
Created directory '/home/sandow/.ssh'.  
Enter passphrase (empty for no passphrase):   
Enter same passphrase again:   
Your identification has been saved in /home/sandow/.ssh/id\_rsa.  
Your public key has been saved in /home/sandow/.ssh/id\_rsa.pub.  
The key fingerprint is:  
08:48:69:77:45:14:00:29:57:a8:62:3e:39:bd:57:78 sandow@NFS01  
The key's randomart image is:  
+--[ RSA 2048]----+  
| ...=o==. |  
| .+.= . |  
| ..=.. |  
|... . . |  
|o.o o S |  
| = . . E |  
| o . o |  
| . . |  
| . |  
+-----------------+

查看生成的文件,请记住这里每个文件的权限位，以后出问题很可能是这里的问题

[sandow@NFS01 .ssh]$ ll  
total 8  
-rw-------. 1 sandow sandow 1675 Nov 11 11:04 id\_rsa  
-rw-r--r--. 1 sandow sandow 394 Nov 11 11:04 id\_rsa.pub

这两个文件里面都是密钥，没什么好看的，反正也看不懂。那么接下来就需要把公钥，也就是**id\_rsa.pub** 分发到每台服务机上

[sandow@NFS01 .ssh]$ ssh-copy-id -i " -p 52113 sandow@172.16.1.41"  
The authenticity of host '[172.16.1.41]:52113 ([172.16.1.41]:52113)' can't be established.  
RSA key fingerprint is a9:5e:34:5c:4c:33:91:39:45:cd:6e:e9:7b:de:7c:c6.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '[172.16.1.41]:52113' (RSA) to the list of known hosts.  
sandow@172.16.1.41's password:   
Now try logging into the machine, with "ssh ' -p 52113 sandow@172.16.1.41'", and check in:  
  
 .ssh/authorized\_keys  
  
to make sure we haven't added extra keys that you weren't expecting.

因为之前已经分发过一次所以这次就会提示已经存在。没分发过的不会出现这个提示的。接下来我们再使用ssh -p 52113 172.16.1.41时便不会再需要密码。注意这里那个用户生成的密钥便只能用那个用户来登录，登录对应的主机用户与当前客户端的用户一致，除非指定用户

有时候我们不想连接到服务器仅想执行一个命令就退出，可以这样写

[sandow@NFS01 .ssh]$ ssh -p52113 172.16.1.41 /sbin/ifconfig eth1  
eth1 Link encap:Ethernet HWaddr 00:0C:29:69:77:D7   
 inet addr:172.16.1.41 Bcast:172.16.1.255 Mask:255.255.255.0  
 inet6 addr: fe80::20c:29ff:fe69:77d7/64 Scope:Link  
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
 RX packets:6289 errors:0 dropped:0 overruns:0 frame:0  
 TX packets:5443 errors:0 dropped:0 overruns:0 carrier:0  
 collisions:0 txqueuelen:1000   
 RX bytes:627464 (612.7 KiB) TX bytes:548720 (535.8 KiB)

因为远程连接执行命令风险很高，所以需要对远程连接加以一定的限制达到风险最低。下面有几种方案：

1. sudo方案   
   在服务机上对远程用户授权（使用visudo -e) 然后再进行ssh 远程命令   
   ssh -t -p 52113 sandow@172.16.1.41 sudo /bin/cp hosts /etc/
2. suid   
   对需要执行的命令设置suid让执行命令时拥有属主的权限。风险很高，

我们返回**~/.ssh**发现多了一个文件**~/.ssh/known\_hosts**密钥文件这里面便记录着连接的主机，如果把里面的内容删除后，再次连接服务器便会提示输入密码。

这里补充一点小知识，我们现在已经知道了怎么去连接了，但是如果默认端口被改了后怎么能知道sshd服务的端口号是多少呢可以用下面命令来查看

[root@backup ~]# netstat -lntup|grep sshd  
tcp 0 0 0.0.0.0:52113 0.0.0.0:\* LISTEN 29649/sshd   
tcp 0 0 :::52113 :::\* LISTEN 29649/sshd

可以看到这里的端口号是52113。如果我知道端口号怎么来查看对应的服务呢？

$ lsof -i :52113  
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME  
sshd 29649 root 3u IPv4 51691 0t0 TCP \*:52113 (LISTEN)  
sshd 29649 root 4u IPv6 51693 0t0 TCP \*:52113 (LISTEN)

**错误与优化**

报错字符串对应的可能问题：

1. no route to host可能为防火墙
2. Connection refused可能为防火墙：Connection refused还可能是连接的对端服务没开   
   ssh -p 22 sandow@10.0.0.19 /sbin/ifconfig

优化

1. 用密钥登录，不用密码登陆。
2. 防火墙封闭ssh指定源限制
3. 开启ssh只监听本地内网IP

**scp**

scp是一个远程的加密复制命令，用ssh加密传输，这个命令是全量copy，也有两种模式push， pull。

参数：

* -1 , ssh1,
* -2, ssh2
* -4, ipv4
* -6, ipv6
* -l limit , 限定宽带大小，基本单位为kbit/s
* -C, 使用压缩
* -P， 指定端口
* -p, 保留文件的最后修改时间，最后访问时间，权限模式
* -r, recurisively copy

远程推送：`scp -P52113 -rp /etc/hosts 172.16.1.41:~

[sandow@NFS01 .ssh]$ scp -P52113 -rp /etc/hosts 172.16.1.41:~  
hosts 100% 211 0.2KB/s 00:00

远程拉取： 把两个文件换个位置这里就不讲了

**批量分发**

OK，接下来就可以写个shell来批量分发指定文件到指定目录啦。这里我在家目录下创建一个send.sh. 内容如下

#!/bin/sh  
. /etc/init.d/functions  
  
if [ $# -lt 1 ];then  
 echo "USAGE:/BIN/SH $0 FILENAME"  
 exit  
fi  
  
for n in 41 8 9 10  
do  
 scp -P52113 -rp $1 172.16.1.${n}:$2 &>/dev/null  
if [ $? -eq 0 ];then  
 action "dis $1 to 172.16.1.${n}:$2" /bin/true  
else  
 action "dis $1 to 172.16.1.${n}:$2" /bin/false  
fi  
done

然后咱们来执行一下看看效果

[sandow@NFS01 ~]$ sh send.sh /etc/hosts ~  
dis /etc/hosts to 172.16.1.41:/home/sandow [ OK ]  
dis /etc/hosts to 172.16.1.8:/home/sandow [ OK ]  
dis /etc/hosts to 172.16.1.9:/home/sandow [FAILED]  
dis /etc/hosts to 172.16.1.10:/home/sandow [FAILED]

OK，如果是windows下的话可以安装**lrzsz**来实现与linux下的文件互传。下面是一些总结   
windows客户端和linux服务器之间传输数据工具：   
1）rz、sz（lrzsz）。<==讲过的。   
2）winscp WinSCP-v4.0.5 <==基于SSH,sftp   
3）SFX(xshell)   
4）SFTP <==基于SSH,加密传输   
5)samba,http,ftp,nfs   
FTP工具：vsftp, proftpd,SFTP

还有一些批量分发工具，改天我会介绍**saltstack** 与**ansible**. 其中**stltstack**是比较好用的高效的分发工具，有兴趣的同学可以先百度研究