



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

网络空间安全导论

第十一章实验报告

数字证书的使用

目录

1	课程实验原理及要求	2
1.1	实验原理	2
1.1.1	实验一	2
1.1.2	实验二	2
1.2	实验目的	2
1.3	实验思路	3
1.3.1	实验一：使用私钥访问SSH服务器	3
1.3.2	实验二：为网站添加HTTPS	3
2	实验环境配置	3
2.1	实验环境	3
2.2	云服务器配置	3
2.2.1	SSH配置	3
2.2.2	Nginx配置	7
3	实验步骤	9
3.1	实验一	9
3.2	实验二	13
4	总结	16
5	参考文献	16

1 课程实验原理及要求

1.1 实验原理

1.1.1 实验一

非对称加密算法生成一对密钥（公钥和私钥），其中，私钥由一方安全保管，而公钥则可对外公开，如果用其中一个密钥加密数据，只有对应密钥才可以解密，利用这一特性可以实现远程服务器对用户身份的认证。在使用私钥访问 SSH 服务器时，用户可以提前将公钥上传至服务器，当用户发起登陆请求时，用户方将利用私钥对服务器发来的随机字符串进行加密，并将密文发送回服务器；服务器收到密文后会根据用户方提供的公钥对密文进行解密，如果成功则用户身份得到验证

1.1.2 实验二

HTTP协议传输的数据都是明文的，且不校验通信的双方的身份，所以为了安全起见可以采用HTTPS协议进行通信，它是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议。数字证书是HTTPS实现安全传输的基础，它由权威的CA机构颁发。

HTTPS通信流程大致如下：

- 1) 服务器从可信CA机构申请证书，本实验可采用自签名生成证书
- 2) 客户端请求服务器建立连接
- 3) 服务器发送网站证书（证书中包含公钥）给客户端
- 4) 客户端验证服务器数字证书，验证通过则协商建立通信

1.2 实验目的

感受PKI在互联网中扮演什么角色，更清楚地认识到证书在网络通信过程中提高安全性保障的重要作用

- 1、会使用私钥对远程服务器进行访问，增强服务器安全意识。
- 2、观察没有PKI服务支持时的Web流量内容
- 3、利用证书实现HTTPS服务，然后观察结果

1.3 实验思路

1.3.1 实验一：使用私钥访问SSH服务器

实验思路：

- 1、生成私钥，通过OpenSSL工具生成公私钥对
- 2、上传公钥到远程服务器对应位置
- 3、开启SSH服务，通过私钥进行安全链接
- 4、关闭SSH密码登录功能，服务器只能通过私钥访问，提高安全性，并测试验证无法通过密码进行登录

1.3.2 实验二：为网站添加HTTPS

实验思路：

- 1、在虚拟机安装并配置Nginx
- 2、自己生成公私钥对为网站安装证书，添加HTTPS 协议
- 3、通过网络分析器分别对HTTP 协议会话和HTTPS 会话进行解析，观察通信内容的区别

2 实验环境配置

2.1 实验环境

- 1、一台云虚拟机和一台本地计算机（我使用了两台虚拟机）
- 2、云服务器需要安装SSH服务和Ngnix服务

2.2 云服务器配置

2.2.1 SSH配置

打开虚拟机，先ping一下检查网络

```
server@ubuntu: ~  
File Edit View Search Terminal Help  
server@ubuntu:~$ ping www.baidu.com  
PING www.a.shifen.com (220.181.38.150) 56(84) bytes of data.  
64 bytes from 220.181.38.150: icmp_seq=1 ttl=128 time=33.2 ms  
64 bytes from 220.181.38.150: icmp_seq=2 ttl=128 time=40.5 ms  
64 bytes from 220.181.38.150: icmp_seq=3 ttl=128 time=40.9 ms  
^C  
--- www.a.shifen.com ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2003ms  
rtt min/avg/max/mdev = 33.234/38.238/40.938/3.545 ms  
server@ubuntu:~$
```

输入`su root`升级权限，然后输入`apt install openssh-server`

```
server@ubuntu:~$ su root  
Password:  
root@ubuntu:/home/server# apt install openssh-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  ncurses-term openssh-sftp-server ssh-import-id  
Suggested packages:  
  ssh-askpass rssh molly-guard monkeysphere  
The following NEW packages will be installed:  
  ncurses-term openssh-server openssh-sftp-server ssh-import-id  
0 upgraded, 4 newly installed, 0 to remove and 180 not upgraded.  
Need to get 633 kB of archives.  
After this operation, 5,136 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 ncurses-term all 6.0+20160213-1ubuntu1  
  9 kB  
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 openssh-sftp-server amd64 1:7.2p2-  
  4ubuntu2.10 [38.8 kB]  
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 openssh-server amd64 1:7.2p2-  
  4ubuntu2.10 [335 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 ssh-import-id all 5.5-0ubuntu1 [10.2  
  9 kB]  
Fetched 633 kB in 34s (18.5 kB/s)  
Preconfiguring packages ...  
Selecting previously unselected package ncurses-term.  
(Reading database ... 177377 files and directories currently installed.)
```

输入命令`service ssh start`开启服务

输入命令`ps -aux | grep ssh`出现`sshd`服务，启动成功

```
Processing triggers for libc-bin (2.19-0ubuntu2.10) ...  
root@ubuntu:/home/server# service ssh start  
root@ubuntu:/home/server# ps -aux | grep ssh  
root      3139  0.0  0.1  65512  5420 ?        Ss   Nov27   0:00 /usr/sbin/sshd -D  
root      3271  0.0  0.0   21292  1012 pts/2    S+   00:00   0:00 grep --color=auto ssh  
root@ubuntu:/home/server#
```

输入命令`ssh-keygen`，生成公钥和密钥，可以一路回车，就自动生成了

```
root@ubuntu:/home/server# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): test
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in test.
Your public key has been saved in test.pub.
The key fingerprint is:
SHA256:Ytwyw0C806q0hUs7YIA+TS0qe6LOIf7URWhwiwxyDQY root@ubuntu
The key's randomart image is:
+---[RSA 2048]-----+
|Eo=+  .|
|o=oo= o|
|o.o=o+  .|
|+..+oo..|
|+=o.  *.S|
|=+o....+|
|*+o.  .|
|*o=.  .|
|0*..  .|
+---[SHA256]-----+
root@ubuntu:/home/server#
```

输入命令`ssh localhost`，登录一下，检查是否一切ok，看到了之前生成的密钥，使用`cat`命令看一下

```

root@ubuntu: /home/server
File Edit View Search Terminal Help
server@ubuntu:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:GqCE19ImIE00KsBF+T62onFUVtcAmentovfPZxrq4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
server@localhost's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.15.0-112-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

188 packages can be updated.
153 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

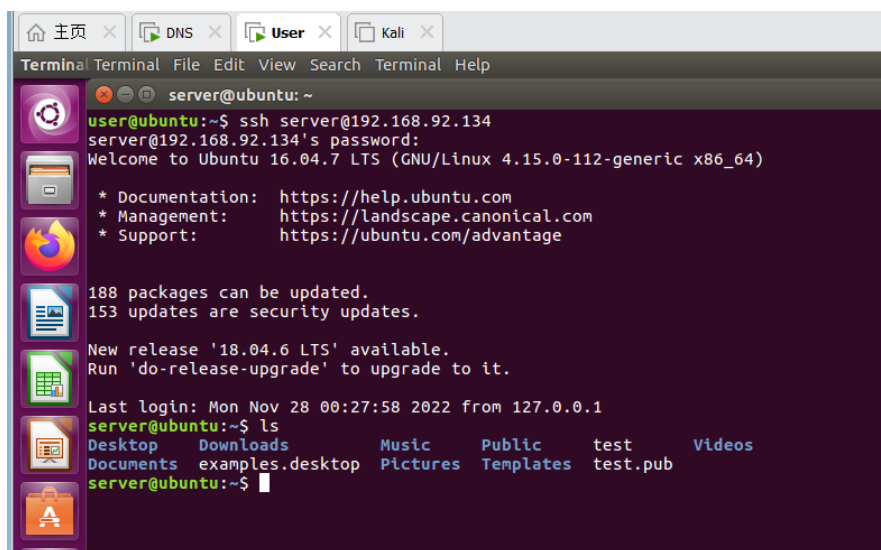
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

server@ubuntu:~$ ^C
server@ubuntu:~$ disconnect
disconnect: command not found
server@ubuntu:~$ cd root
-bash: cd: root: No such file or directory
server@ubuntu:~$ ls
Desktop  Documents  Downloads  examples.desktop  Music  Pictures  Public  Templates  test  test.pub  Videos
server@ubuntu:~$ cd test
-bash: cd: test: Not a directory
server@ubuntu:~$ cat test
cat: test: Permission denied
server@ubuntu:~$ su root
Password:
root@ubuntu:/home/server# cat test
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAzVGLYw31F5aEdm+/9bcvE5VA81VLT5oaM3e1+6aDwUuZj0Vq
Xmz3n7b1ja42ty2YNNdpPDH-/3AlVfk04701924200PwJAlVCfpEDfkyMtsI9Yv
JWxvN70V2frvLHdbfV9WfTgQfYHsp00FA2xg6uPgzaI0rn/6g9xhx7nPW7MoUp
A7osBy45zb0UPmptjZ7EVGwKtdp8fQynCo3dAlN3LRFvRwq4u6I8BqncFr4Kp/Z
ob4USPGxK/nrPp4nahBStC2WEysXKcyehs6ykp/Gch9N08HbUjRZ0H2ycdhnJ+Z0
eBcc9a1RF0yYrS1LLDGBdFLnxvKf/rTQcGLQIDAQABAIAAAHlCan5BwEL3kr9
N6hTCR3UQH6EAK08zbni0vni2b9F55IAVB0uq57QW8+Hf9rPK3mWlHRCt8Ts
3Zz0u7HujEBP+59U4jeRbnnH38qvy0h1eegs0odZBg108Mwv9F5zStQ6ehuv/0
NU4PROF+NFU9zVpZewZBCj+RdVCA7r4J00Rxn0eRFyjuM2aXlGjRfGEBQfKypET
dj48cOPC8L5n9qNMFD8jbfPfb1UgLVHxR0ZThody1P2KvVXZ86zchtJThZWB8W
GR952WdZB9gNX8EuaCRHPFr7YVgq2KEC1SFhu2L9uoQ5FhxVf5kndZtwa2Tms6F
QIW7dGEGYEA7MH06xt7+Whn+6Et2MG0xJZL4RKnRKv6gCJLe0Cj1qYAUlF4ugY
PetV0YSzZ9MKrKap7P+zgha2lAj/coISnRQmZ8nIhrtxyLSBM7P8b2guhBewHN+b
oG6lNk7VpScBkgosC0wctb171ZdA8gP/rmg+H5UPrnlCB0u2TcbCgYEAy7mo
UdpYvbn71X4nph//SR8lRPW/y5fXMRp0p9rt4Khg0LSZM0TdxjMwZPvgx1Z
11ebwCVrgnfMHDfVMKKQRWuJ9cfYahhb11LFMH70Ms5fAmnG18nacV209MGr3CN
EdKwAR75ZM920+77+s7Ikm3xPogT/EU4uA+fzScgYEA2F/785L40eTYe4D1KHKA
xEXtmL6mLk1g1lpdykG0Fe0vP5wYFrPp09JlRRJnVr7/QqJp8+eFq3sTW/9/AK
JY0xQnNOUgHX7KqLk0dHNgzAuf/PTQJMsnyUE2IRwWEXr1uXX+NIPF559yFH5Cc
1KZ20hAWBaseaAAHnqgaBjScgYB0MA0lRZ/CS5trAg/s0f1N5a9LTTudh9dgHRts
1AdvXPYEnCv45nBnLUkF02eZLpW0s1UvHkac0h0ak8B7TGrKv8Dvuk2gtLk5C
W016qE3Kc/6dRjCbVvX1z3tb1dsc7Fuj0DM8xaUFMB7un1WwE9WKGpTbncTail
5HNsJwK8gCNjek8gTk87NQv60cCYKZQfrVf+JeEXGyD4ehhE5pk7hkuKH/FOB1D
M8NEsRnbNYH0ga9RCdVMD0Zv0uutaTyB8r0CG8n4vCewKRnsQNTTLalLwGuuCuX
lRnkDpr/jVPT9jup1btMnolgiPDnsrgYpBE7b1Az0U0xht+lo8
-----END RSA PRIVATE KEY-----
root@ubuntu:/home/server# cat test.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQ8aWLLDFUvJor1b7/1ty8TLUDzVLPmhozd7X7poPBRrPRNpebPefuMNRja3LZg012k8mf78kCJV+TTjySL3bJm6g
JgJ+kQWMTJ2PCwJ1181Ze+E3sSXZ+uBUent9X1YVmaFB/1eynSpUDB0d4+DnoJquf/qD3GHuc9bsyhSkDuLwHlJnNs5Q+an2NnsS8bAQ0N2nx9DKckjdcCI3ctf+9Hc
JwGC3+vgq9nhvhtK8Ber+as+ndqEFJnzZYRlXcpzJ6GzrK5n8ZyH007dtsNfNqfbjX0ecn5nR4fwL2jVEU7JlVnUUsMYF0UuZe+sp/tNBWyt root@ubuntu
root@ubuntu:/home/server#

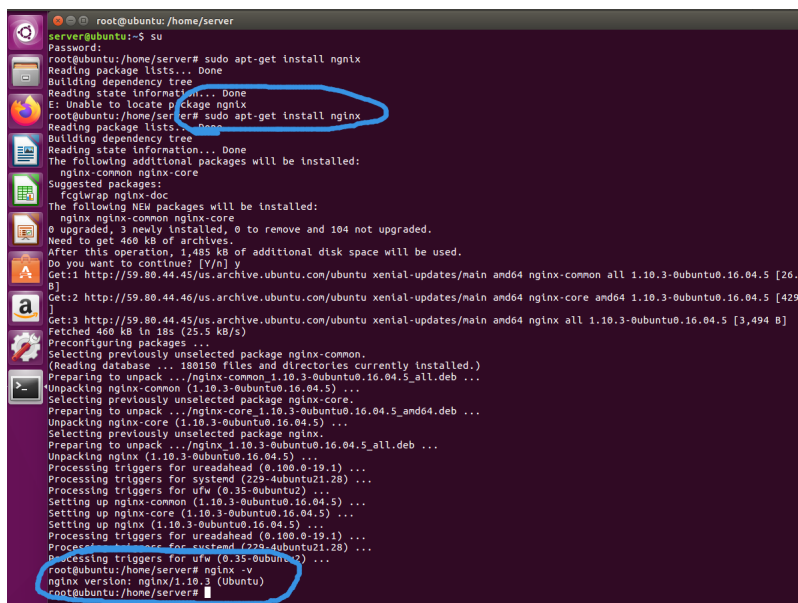
```

用本地的其他Ubuntu虚拟机模拟登录一下看看，没啥问题登录成功，说明ssh配置成功



2.2.2 Nginx配置

输入命令`sudo apt - get install nginx`安装nginx，输入`nginx - v`查看nginx的版本



3 实验步骤

3.1 实验一

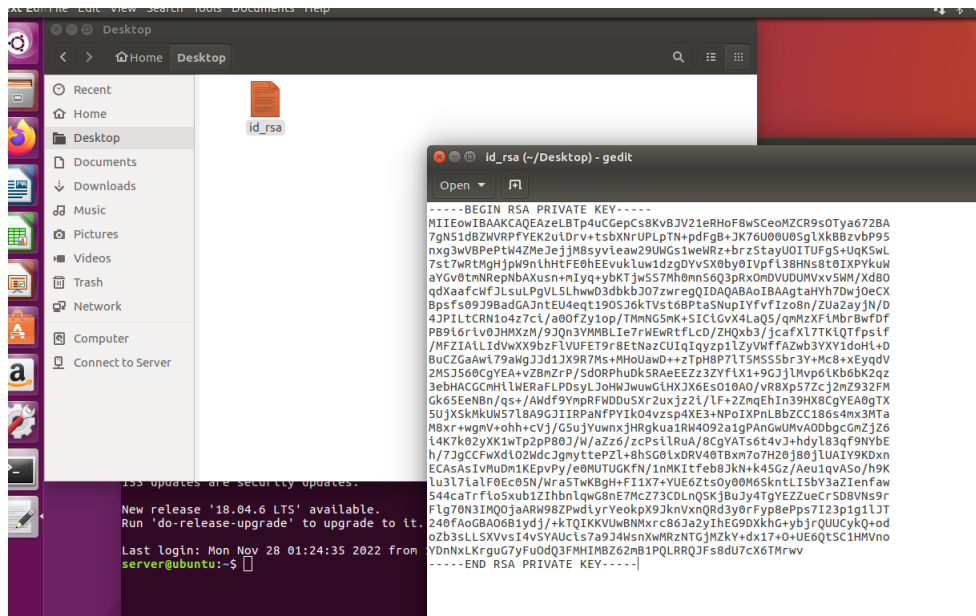
首先生成一对公私钥，我使用命令`ssh-keygen`，然后一路空格

```
server@ubuntu:~$ cd .ssh
server@ubuntu:~/.ssh$ ls
authorized_keys id_rsa id_rsa.pub known_hosts \.pub
server@ubuntu:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/server/.ssh/id_rsa):
/home/server/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/server/.ssh/id_rsa.
Your public key has been saved in /home/server/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Lx0J3eLWyKsjlEQcc0WrhFx2+FQb8qcaZ/DUjbEhxqg server@ubuntu
The key's randomart image is:
+---[RSA 2048]---+
| .00000= 0      |
| .00.=. = *     |
| .+. = + = .    |
| .+E* 0 =       |
| o.o.S X .      |
| .o. 0 0        |
| .. + =         |
| . =            |
| ...           |
+-----[SHA256]-----+
```

产生了密钥后，`ls`一下，然后将`id_rsa`复制一下，这个是密钥

```
+----[SHA256]-----+
server@ubuntu:~/.ssh$
server@ubuntu:~/.ssh$ ls
\ authorized_keys id_rsa id_rsa.pub known_hosts \.pub
server@ubuntu:~/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAXie/NP300suwuRcjIGlohiYdybfmLL8zDRDiJ79vRku45BCm
yAFIU0F6PddeYA9birasJ3zriRbEKJb+uY3+T0Jcjmz5s4ttBLNp61R/h97cMJH
Oq2Ka+wgCa2btzLFrBcIujtF4BqR8N5iIDPwLN4ALvMexJ3cqXe1rJdDjXfPS00f
FvA00igGIXyaeyRcWY2s4NSZJ+9pTMzEDMpCPn0WI2RCLiU8ER8ViLcLyj2M9wws
3ngreLVDNminXaQxdRSuGmi/sx/SKTyNSAo505wcknfaj0yJ+KeM1Er80q2u2Wl
hf6puj4pr2GgcPRdRDVKHEVpgvkdWa61TUa8AQIDAQABAoIBAGGRPo+MH443rxtj
rhMOFTSAUUsodnkMv0LJb+H8W4DkVFRN0qEWKJN94P/EJJ7t00gbc+chliPZBtrp
hiC9Lq0+DC7Ar65dD1/MFbtjFsUmoTQjJmLam5aKHDhHuC1eUNTSPorcForC6msM
N/CAiLi9Mp/ra5HqHLWA0P5bOF+dbPKrpxhE4nm5XH2wccHkiDYDY9EYfQ0dzQcq
zZWapEdQEgMWegJB+td0osbuq5WkdRwRAW0/0uhJemCCCLHpN/VYz4Y7w7eIFKjv
G7lFWkpn9ewQ5nqhItqc+lyPt+hjLSb0bzKRB22zjW0tIfgblWP2LQL/Wh7+sDmg
XsAQxMECgYEA+/HbX4m5DEz87S41ugcLImobWksdW5uBc23eTja0guGrU3IRRAHQ
+VBO2j15K8DMWv5GHgiDq7LWJU3bTJSNqM2/WatqmGLz/I0ShzbaQQordH67aCuz
eSFuTYDOV/QN6BgCU+CooVltsZX2pQ2L4Zc5FbhsApMfpHTIMB81AukCgYEAyVg/
1nKhjz67ZRyWGu1HdVN8ygCwTfc1Rpm29fjMoIrcJzLTr6k46AbiPlgzmsVuyxUH
UUJkvP3oJgpKf753TCuM+BjwGJCX3xJgS1JqGki8gyGndbLvd7ROZ192yAULrWN+
mWmc94/MERj6JTxePRnz3/hlx/VjcLA3B0hLUVKcgyBzS3fPna/wIB8yzw3JJtHo
kNQZnALG1EAG0cYc1s3+gxUrJ60dDA06+N0s7arQyfb6wRgHUHApGrSAd80epPC6
xr0qrLYmj00D0BYzCHgSgXvnYMOUZUNJcHIQx5wmBcqDZtFR3xaGYe0wWo83cypx
NQnrLazf9MHRPHUJgg55WQKBGu4ulbKd60C5ZGgLL1VfDwnOgaXDKqVKEUH+4T
ka8m7d8480adG+QPosXGPL3GeZNTBB+2jN0MX0LS6K7VwU/QRh/hx0wKtVcKiBd
7h5nwlrDC1pp7HK31ea6jXtQPas5ewcnntzYY06WdWapHmDEXNTS0dA0emMzhW1x
DzvpAoGBAKSRM10ckDpY4jR+ai0tUVvc0TqBnwT3eHZbBmhhHXvTQpUr0e7DAvDr
TYcT6vmLfIKXozsN8GjwpyNumY+N2hB399MRUJDJnE0bJIqPYpR1rkn7Rww//iYl
RXgrIkEo9v780PHgSQaium8/ZFnDuT6cokq4JXBtGH0+jikEMTEJ
-----END RSA PRIVATE KEY-----
server@ubuntu:~/.ssh$
```

把它复制到另一台虚拟机上，我是用的上一个实验的User虚拟机，在桌面建立一个文件，把密钥放进去，截图是用的上一个密钥



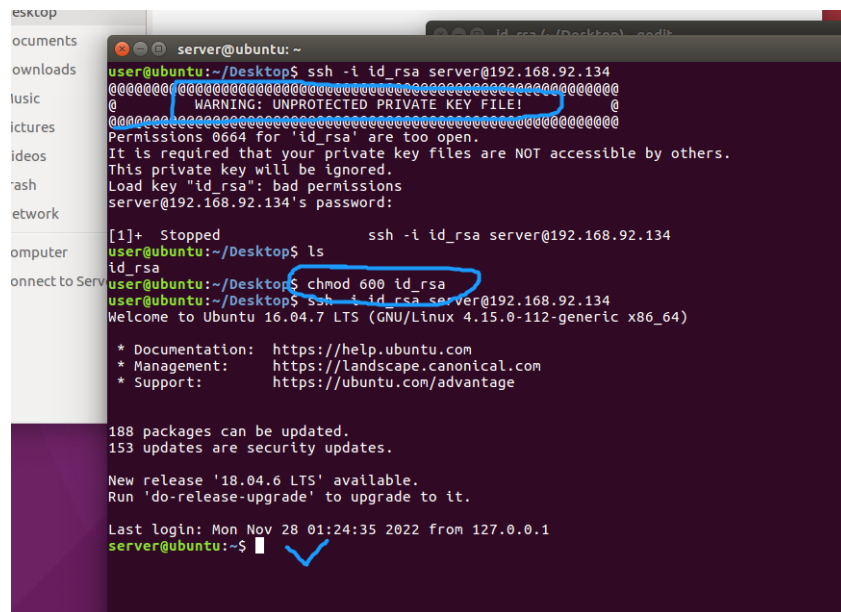
同时还要把公钥放进`authorized_keys`中，不然密钥是非法的，进不去，之前已经建立过了，不然需要`touch authorized_keys`建立`authorized_keys`文件

输入命令`cat id_rsa.pub >> authorized_keys`

输入命令`cat authorized_keys`查看里面的信息

```
-----END RSA PRIVATE KEY-----
server@ubuntu:~/.ssh$ cat id_rsa.pub >> authorized_keys
server@ubuntu:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBN4sFon141Z6Kzwg8ELybV5EegXsB1J6gk3H2v5P3rvvEduA3nV8F1ZVE9gg
Qra61Ou/62atzt2t0u1h3610AH4krvptTRTRKCVeQEHO9s/3nfGDf8Ue94+1bhkx4160Mzyzk+35pb1RvazXB5ZHP5uWk1f3Q4
*HNQWBL5SopLAvuy3v8Goyae0Lb2eEe0TSEqS+6SW70V30ANI9JfRVLQhW1+Lfwc23y3Qhc9LS5Zpqa/S2Y1F6k1sBe5yfoYJkr7
Jsp0PB3LsyHsdlpdlHE6YVNOQXG/LyZ9d3wop1dpp9xZ8kuy4s+BUvkuH0APd1uRsk7vPct68 server@ubuntu
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDLyboDAM1gt1vth46KpT8DB22Kzt50kuWU91y4K8KkhZKayH2KgQhv7RTnGmE
51NbbqLAFK1qwd1fvcVNTPEExL4rL21Clgzaan0KEUgK16fucB4v+xECLentvFjRT/31ZH9Y3ST51Zfg0lRF5Yw4J4d6mV5Cnu
XR2exclUShDQ6bjpg2j9fy3/vYvL/uanZHeNoovEgxBvbqMNYEH41bUuUJ2KN7+TtLAlKJ24FT6vHCFopZEPGD0p1eFP3cKLVEY+T
FB/+tVw6v08hyst172Fh0LSORjtrEPQARRxVjuJLFNnLS+e+gkVxvP0eKdQ8/I+1/v+k5SKsf4B server@ubuntu
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDLyboDAM1gt1vth46KpT8DB22Kzt50kuWU91y4K8KkhZKayH2KgQhv7RTnGmE
51NbbqLAFK1qwd1fvcVNTPEExL4rL21Clgzaan0KEUgK16fucB4v+xECLentvFjRT/31ZH9Y3ST51Zfg0lRF5Yw4J4d6mV5Cnu
XR2exclUShDQ6bjpg2j9fy3/vYvL/uanZHeNoovEgxBvbqMNYEH41bUuUJ2KN7+TtLAlKJ24FT6vHCFopZEPGD0p1eFP3cKLVEY+T
FB/+tVw6v08hyst172Fh0LSORjtrEPQARRxVjuJLFNnLS+e+gkVxvP0eKdQ8/I+1/v+k5SKsf4B server@ubuntu
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDQG3780/FQ6y7C5FyMgaKLG3gP3t+YuxZMNE0Inv29G57jKEKbIAUH0Xo9115gD
1UKtqwnfouJf5QolV65JfSPQ1yobPmz120Et02nrVH+H3twkC0rYpr7CAJrZu30UmsFwL600XGpHw3k1IM/As3gAu8X7Endypd7
W5t0R1CU9L7RNB8DQ6KAYHfJp7JfXblazg1Jkn72LHzMQMyKI+FRYJZE1UK7rHxMItyXKPZ30Czeect4tUW2aKddp0P1FK4aa+
+2H91P11iCjntn8y90PPT1n4p4y15vz5ra7YyWf/qn6PlvYaBw9F1EUocRhmC+R1ZrrVNRwB server@ubuntu
server@ubuntu:~/.ssh$
```

在虚拟机上输入命令`ssh -i id_rsa server@192.168.92.134`，会发现密钥不安全，因为太开放（too open），因此要输入命令`chmod 600 id_rsa`加上权限，在输入登录命令就能登录成功



```
server@ubuntu: ~
user@ubuntu:~/Desktop$ ssh -i id_rsa server@192.168.92.134
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@           WARNING: UNPROTECTED PRIVATE KEY FILE!           @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for 'id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "id_rsa": bad permissions
server@192.168.92.134's password:
[1]+  Stopped                  ssh -i id_rsa server@192.168.92.134
user@ubuntu:~/Desktop$ ls
id_rsa
user@ubuntu:~/Desktop$ chmod 600 id_rsa
user@ubuntu:~/Desktop$ ssh -i id_rsa server@192.168.92.134
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.15.0-112-generic x86_64)

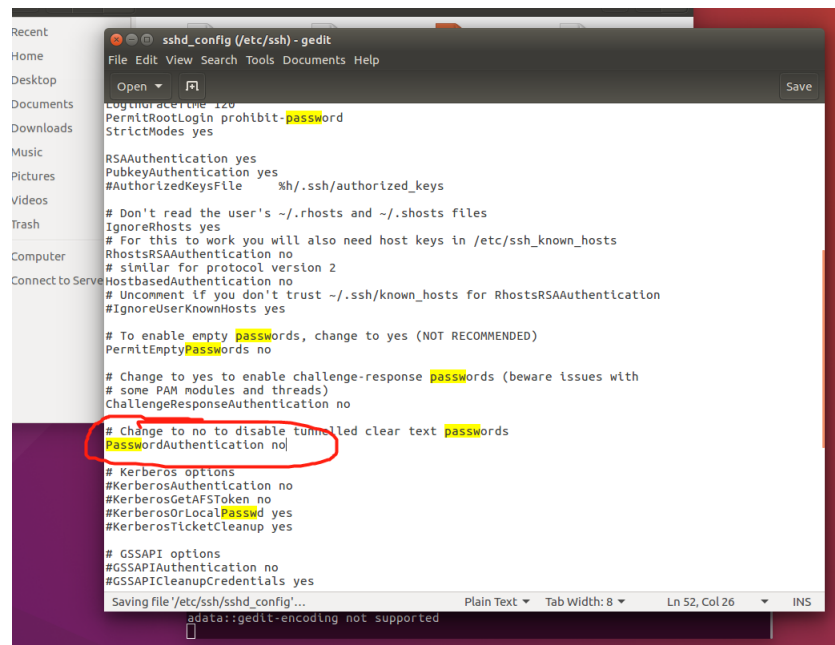
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

188 packages can be updated.
153 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Nov 28 01:24:35 2022 from 127.0.0.1
server@ubuntu:~$
```

然后是关掉密码登录，找到`sshd_config`和`ssh_config`文件，文件路径是`/etc/ssh/sshd_config`，然后改一下设置，输入命令`service sshd restart`重新启动ssh服务器。检查一下发现关掉了密码登录，不能利用密码登录，可以用密钥登录



```
user@ubuntu:~/Desktop$ ssh server@192.168.92.134
Permission denied (publickey).
user@ubuntu:~/Desktop$ ssh -i id_rsa server@192.168.92.134
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.15.0-112-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

188 packages can be updated.
153 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Nov 28 02:55:46 2022 from 192.168.92.1
server@ubuntu:~$
```

3.2 实验二

首先是申请CA证书，我使用的openssl创建本地https证书，模拟CA认证机构的创建证书流程，首先进入到nginx的根目录下，我的是/etc/nginx，然后升级到管理员权限，实验流程如下：

1、生成服务器私钥。

```
openssl genrsa -out server.key 1024
```

2、根据服务器私钥文件生成证书请求文件，这个文件中会包含申请人的一些信息，所以执行下面这行命令过程中需要用户在命令行输入一些用户信息，随便填写，一路回车即可

```
openssl req -new -key server.key -out server.csr
```

3、生成CA机构的私钥，命令和生成服务器私钥一样，只不过这是CA的私钥

```
openssl genrsa -out ca.key 1024
```

4、生成CA机构自己的证书申请文件

```
openssl req -new -key ca.key -out ca.csr
```

5、生成自签名证书，CA机构用自己的私钥和证书申请文件生成自己签名的证书，俗称自签名证书，这里可以理解为根证书。

```
openssl x509 -req -in ca.csr -signkey ca.key -out ca.crt
```

6、根据CA机构的自签名证书ca.crt或者叫根证书、CA机构的私钥ca.key、服务器的证书申请文件server.csr生成服务端证书。

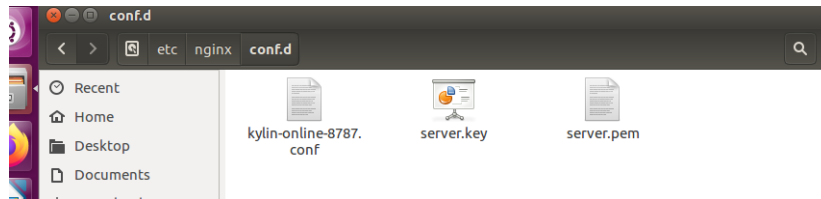
```
openssl x509 -req -CA ca.crt -CAkey ca.key -CAcreateserial -  
in server.csr -out server.crt
```

7、输入命令，将拿到的私钥server.key和证书server.crt复制到一起形成公钥server.pem

```
cat server.key server.crt > server.pem
```

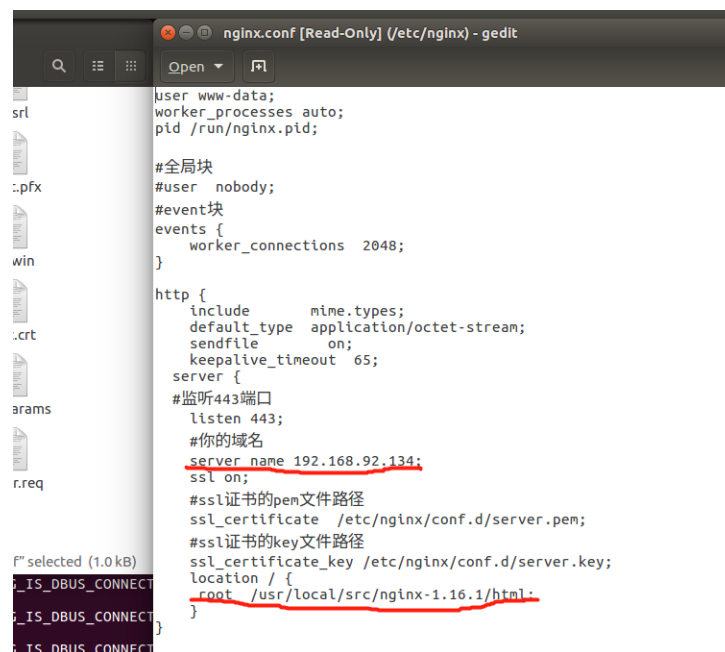
8、公钥server.pem和私钥server.key放入/etc/nginx/conf.d中

```
mv serve.pem/etc/nginx/conf.d
mv serve.key/etc/nginx/conf.d
```



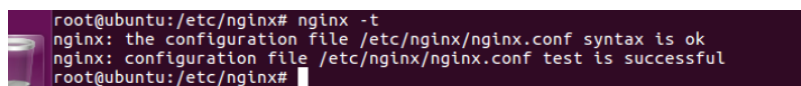
pem是公钥，key是私钥

9、将nginx配置文件nginx.conf里的http server服务修改为https server服务，nginx文件位置/etc/nginx/nginx.conf

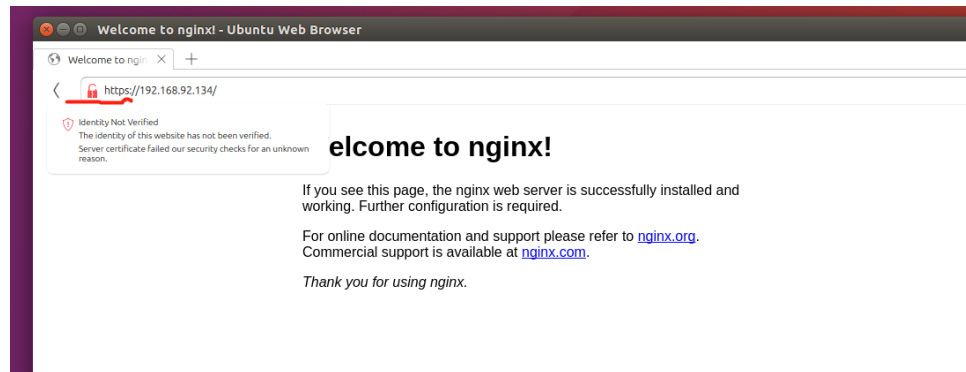


注意：这里需要在安全组中开放443端口

10、然后检查一下，输入命令nginx -t，显示success就表示没有问题，接着输入命令nginx -s reload重启nginx服务



11、接着在另一台虚拟机上，打开浏览器输入网址`https://192.169.92.134`，访问成功！



4 总结

通过本次实验，我了解到了CA证书颁发的详细过程，了解了SSL密钥登录的安全性以及实现方式，同时在配置Nginx与进行实验二的过程中遇到了很多的问题，最后终于解决了，收获颇丰！

5 参考文献

- [1]徐恪，李琦等，《网络空间安全原理与实践》，清华大学出版社
- [2]<https://www.cnblogs.com/ambition26/p/14077773.html>
- [3]<https://cloud.tencent.com/developer/article/1548350>
- [4]<https://blog.csdn.net/afreon/article/details/97142847>