Practical 11

Configuring SSH

Configuration of SSH (Secure Shell Remote Service) key-based Authentication

1) Check whether ssh package available.

```
[root@localhost ~]# rpm -q openssh
openssh-7.4p1-11.el7.x86_64
[root@localhost ~]#
```

2) From Client Machine access the server using command:
#ssh root@10.0.0.100
Type yes and you will get access to the Server from Client.

```
File  Edit  View  Search  Terminal  Help
[root@localhost ~]# ssh root@10.0.0.100
The authenticity of host '10.0.0.100 (10.0.0.100)' can't be established.
ECDSA key fingerprint is SHA256:U4V9Zr+NLJNg/HBXRYn1Dj7XB8didv1tGOpwRdaysd0.
ECDSA key fingerprint is MD5:ac:f2:d9:64:54:ba:12:7a:73:0f:3f:1c:60:b5:15:96.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.100' (ECDSA) to the list of known hosts.
root@10.0.0.100's password:
Last login: Mon Sep 30 15:35:37 2024
[root@localhost ~]#
```

3) Change to root directory. Give command **ssh-keygen -t dsa** to generate key. Press Enter when asked for passphrase.

```
[root@localhost ~]# ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
SHA256:QK/l98vyzyLeXev5SsNiv8yHUL6hqeET6LYHHqyaBQg root@localhost.localdomain
The key's randomart image is:
+---[DSA 1024]----+
|      .          |
|     . .         |
|E     . o        |
| . .   =      .  |
|  . . ..S..  o   |
|     .  =.....o  |
|      .+ o..++++.|
|     o. +.==+Xoo+|
|    o. ..++*=o@Bo|
+----[SHA256]-----+
[root@localhost ~]#
```

4) From Server Machine, change to root. Give command ls -a /root. There is .ssh directory, if it's not present create it using mkdir command.

```
[root@localhost ~]# ls -a
.                .bash_history  .bashrc  .cshrc                model       .mysql_history
..               .bash_logout   .cache   .dbus                 model2.sql  .rnd
anaconda-ks.cfg  .bash_profile  .config  initial-setup-ks.cfg  model.sql   .ssh
[root@localhost ~]#
```

5) Change to .ssh directory using **cd .ssh** command and make a new directory in the .ssh directory using **mkdir authorised_keys**. Then disable the firewall.

```
[root@localhost ~]# cd .ssh
[root@localhost .ssh]# mkdir authorised_keys
[root@localhost .ssh]# systemctl stop firewalld
[root@localhost .ssh]#
```

6) From Client Machine, secure copy using **scp /root/.ssh/id_rsa.pub root@10.0.0.100:/root/.ssh/authorised_keys** command. (to copy the public key of Client on the server)

```
[root@localhost ~]# scp /root/.ssh/id_dsa.pub 10.0.0.100:/root/.ssh/authorised_keys
root@10.0.0.100's password:
id_dsa.pub                                    100%  616   900.3KB/s   00:00
[root@localhost ~]#
```

7) From Server machine, give **ls authorised_keys** command to check the contents inside the .ssh directory.

```
[root@localhost .ssh]# ls authorised_keys
id_dsa.pub
[root@localhost .ssh]# _
```

8) In order to take login to Server Machine from client machine, go to client machine and give command **ssh root@10.0.0.100** Type the password and press Enter.

```
[root@localhost ~]# ssh root@10.0.0.100
root@10.0.0.100's password:
Last login: Mon Sep 30 15:37:45 2024 from 10.0.0.100
```

9) From Client machine give **who** command. The last entry will be of 10.0.0.50

```
[root@localhost ~]# who
student  :0           2024-09-30 13:15 (:0)
student  pts/0        2024-09-30 13:36 (:0)
root     pts/1        2024-09-30 15:33 (10.0.0.51)
root     pts/2        2024-09-30 15:37 (10.0.0.100)
root     pts/3        2024-09-30 15:55 (10.0.0.100)
```

10) Go to server and give exit command. It will close the connection.