

CY 5010: Foundations to Information Assurance

Lab 1: SSH Authentication

Part 1: Password Authentication

We successfully ran the ssh command to connect to the master VM using username and password provided as shown on figure 1.

1. We ran the `id` command to view User Identifier of our team which is 1046 and group Id (1048) which was assigned to team8 on the master server.
2. We also ran `ifconfig` command to view the configuration of network interfaces on the master server.

```
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.

team8@SP21CY5010master:~$ id
uid=1046(team8) gid=1048(team8) groups=1048(team8),1002(student)
team8@SP21CY5010master:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:aa:b7:61:9b txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.10.140 netmask 255.255.255.0 broadcast 10.0.10.255
    inet6 fe80::250:56ff:fe2f:45 prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:2f:00:45 txqueuelen 1000 (Ethernet)
    RX packets 223139 bytes 227108943 (227.1 MB)
    RX errors 0 dropped 208 overruns 0 frame 0
    TX packets 97051 bytes 154477148 (154.4 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 2 bytes 176 (176.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2 bytes 176 (176.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

team8@SP21CY5010master:~$
```

Figure 1

Part 2: Public Key Authentication

In part 1, we had to type in the password for authentication when we ran the ssh command. Thus, we now set up public key authentication in part 2. Exited the master VM and configured the following on our local VM.

1. Generated RSA private and public keys using the following command.
`ssh-keygen -t rsa -b 2048`
2. Then copied the public key we generated to the master VM. This will be used later for authentication when we connect to the master server using ssh.
`ssh-copied -i ~/ssh/id-rsa.pub team8@cy5010.ccs.neu.edu -p 17001`

- Now that we have the private key on our VM and public key on the master server, we tried to ssh to the master server using this key pair. We were able to successfully ssh without typing the password for authentication as seen on figure 2.

```

user@ubuntu:~$ ssh team8@cy5010.ccs.neu.edu -p 17001
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64)

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

System information as of Fri Jan 29 21:09:45 EST 2021

System load:  0.0               Processes:           107
Usage of /:   41.0% of 15.68GB   Users logged in:    1
Memory usage: 27%              IP address for ens160: 10.0.10.140
Swap usage:   0%               IP address for docker0: 172.17.0.1

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

   https://microk8s.io/high-availability

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

11 packages can be updated.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Fri Jan 29 19:30:37 2021 from 10.15.185.27

team8@SP21CY5010master:~$
team8@SP21CY5010master:~$

```

Figure 2

Part 3: Securely Copying Files To/From the Host/VM

In this part, we followed the instructions given from the lab tasks.

- Copied the **Test.jpg** file from the master server to our VM using the following command.
`scp -P 17001 team8@cy5001.ccs.neu.edu:/home/SharedFolder/Test.jpg .`
- Created **details.team** file using the following command that is also seen on figure 3.
`echo -e "CY5010 Spring 2021\n TEAM 8 \n +Sowmyashree Bevrur Manya Vankatesh\n +Vanessa Kibaja" > details.team`

```

*** System restart required ***
Last login: Sat Jan 30 18:43:52 2021 from 10.15.185.27
team8@SP21CY5010master:~$ cd /home/SharedFolder
team8@SP21CY5010master:/home/SharedFolder$ ls
Test.jpg
team8@SP21CY5010master:/home/SharedFolder$ exit
logout
Connection to cy5010.ccs.neu.edu closed.
user@ubuntu:~$ scp -P 17001 team8@cy5010.ccs.neu.edu:/home/SharedFolder/Test.jpg .
Test.jpg
user@ubuntu:~$ ls
Test.jpg
user@ubuntu:~$ echo -e "CY5010 Spring 2021\n TEAM #8 \n + Sowmyashree Bevur Mandya Venkatesh \n + Vanessa Kibaja" > details.team
user@ubuntu:~$ cat details.team
CY5010 Spring 2021
TEAM #8
+ Sowmyashree Bevur Mandya Venkatesh
+ Vanessa Kibaja
user@ubuntu:~$ _

```

Figure 3

- Appended the contents of `details.team` file to `Test.jpg` file using the following command.
`cat details.team >> Test.jpg`
- Generated and moved the sha-256 hash value of `Test.jpg` file to `hash.team` file using the following command as seen on figure 4.
`sha256sum Test.jpg > hash.team`

```

TEAM #8
+ Sowmyashree Bevur Mandya Venkatesh
+ Vanessa Kibaja
user@ubuntu:~$ sha256sum Test.jpg > hash.team
user@ubuntu:~$ cat hash.team
2a59c9f7ffc3f2933402aa15e1c34fc9efc38fc0d7b206c15c2803df2ef3f042 Test.jpg
user@ubuntu:~$ _

```

Figure 4

- Finally created a `lab1` directory on the master server (`mkdir lab1`) and moved all the files from step 1-4 in this directory as seen below.
`scp -P 17001 /home/user/hash.team team8@cy5010.ccs.neu.edu:/home/team8/lab1`
`scp -P 17001 /home/user/Test.jpg team8@cy5010.ccs.neu.edu:/home/team8/lab1`
`scp -P 17001 /home/user/details.team team8@cy5010.ccs.neu.edu:/home/team8/lab1`

```

details.team 100% 86 2.4KB/s 00:00
user@ubuntu:~$ scp -P 17001 /home/user/hash.team team8@cy5010.ccs.neu.edu:/home/team8/lab1
hash.team 100% 75 3.5KB/s 00:00
user@ubuntu:~$ ssh team8@cy5010.ccs.neu.edu -p 17001
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64)

```

Figure 5

```

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

System information as of Sat Jan 30 20:51:33 EST 2021

System load:  0.07          Processes:            109
Usage of /:   40.8% of 15.68GB Users logged in:      2
Memory usage: 29%          IP address for ens160: 10.0.10.140
Swap usage:   0%           IP address for docker0: 172.17.0.1

* Introducing self-healing high availability clusters in MicroK8s.
  Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

  https://microk8s.io/high-availability

* Canonical Livepatch is available for installation.
  - Reduce system reboots and improve kernel security. Activate at:
    https://ubuntu.com/livepatch

11 packages can be updated.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Sat Jan 30 20:38:12 2021 from 10.15.185.27
team8@SP21CY5010master:~$ cd /home/team8
team8@SP21CY5010master:/home/team8$ ls
lab1
team8@SP21CY5010master:/home/team8$ cd lab1
team8@SP21CY5010master:/home/team8/lab1$ ls
details.team  hash.team    Test.jpg
team8@SP21CY5010master:/home/team8/lab1$ _

```

Figure 6

Questions:

1. How can you debug on your SSH connection?

We can debug SSH by running it in verbose mode. Verbose option provides more in-depth detail on what the ssh command is doing like the status of connection, error message if the connection failed, etc. Hence, we can debug using the information obtained from verbose.

Verbose has 3 levels (-v, -vv, -vvv) in which level 3 (-vvv) provides more advance information than level 1 (-v). For our case, we can debug ssh using the following command.

```
ssh -v team8@cy5010.ccs.neu.edu
```

2. What is the other use-cases (functionalities) for SSH? Explain in brief.

Here are most common uses of SSH:

- SSH command itself is being used to securely connect two different machines with authentication of either a password or using private and public keys. This helps to eliminate intermission from Network sniffers. (Which was realized in this Lab Assignment)
- Securely copy files from source to destination using the scp command. Here the files are encrypted before transferred and decrypted as soon as it arrives at the destinations. This one-way process makes it easier for users to transfer their files without manually

encrypting and decrypting their files. (We also used this functionality in our Assignment)

- c. Port Forwarding/ Tunneling: SSH enables Tunneling. This functionality enables insecure applications like Ftp run in a much secure environment. SSH encrypts the entire data traffic hence creating an end to end encryption between the users.
- d. Access control: We can utilize SSH to not only connect to servers but also provide specific level-based access control. We can restrict the programs used by Users with specific access to the given servers. For example, we can disable email access to certain users.
- e. Secure remote command: A security administrator can use SSH to view all the active processes or opened ports on many devices on the same network using SSH. Since SSH can open a lot of connections to many server ports over a given network, as an administrator we can utilize this feature to execute a single command across all the given servers thus reducing the time of operations. For example, if we want to start all the apache servers on the given servers, we can simply do this using the above feature.

3. State the linux command to securely copy a file from your VM to the master server.

`scp -P 17001 /home/user/hash.team team8@cy5001.ccs.neu.edu:/home/team8/lab1`

We used the above command to copy hash.team file from the local VM onto the master server. /home/user/hash.team was the pathname for the file on the VM to be copied onto the master server. We used the scp command (`scp`) that connects to the master server team8@cy5001.ccs.neu.edu through port 17001 (`-P 17001`). The semi-colon (:) separates the server address and the file path of the destination directory.

4. Submit the screenshot of the ssh-keygen command and its output. Explain in your own words the command which you used to create a stronger key.

We ran the following command to generate the public and private keys from our VM.

`ssh-keygen -t rsa -b 2048`

The ssh-keygen command above, uses the -t option to select encryption algorithm (rsa in our case) and -b option for specifying the key size (here 2048 bits). It is recommended to use minimum of 2048 bit key size for rsa since it would take thousands of years to crack it.

We observed the following output for the command that we ran:

- After generating the key pairs, it asked for a file path to save the keys in and passphrase to use for authentication (home/.ssh/id_rsa directory). We decided to keep the passphrase empty.
- It also generated a key print. This is unique value to the key which is obtained by using a cryptographic hash function. SSH used SHA-256 to do the same.
- In addition to the Key print it also provided with key's randomart image.

```

user@ubuntu:~$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:BNJ9raBLutvf71hvhI/DDcGoRSwSOhc3foQmUcwjgaw user@ubuntu
The key's randomart image is:
+----[RSA 2048]-----+
|
|..o*0+o..
|oo==Xo+ .
|.o.*0*+.
|E oo..+ o
| o .So o
|.. . o .
|..*
|.. . 0+..+
|..... 0000.
+----[SHA256]-----+
user@ubuntu:~$

```

ssh-keygen

Here are the screen shots of the generated keys:

```

-----[SHA256]-----+
user@ubuntu:~$ ls -la
.  .bash_history  .bashrc  .local  .ssh
.. .bash_logout  .cache   .profile  .sudo_as_admin_successful
user@ubuntu:~$ cat ssh/id_rsa
cat: ssh/id_rsa: No such file or directory
user@ubuntu:~$ cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAg92xQ0QisSAiH6pz1U1vz1bEg+eqaq4u9IanLOHS+Re/35FC
rDMS10ISWjv3U21+axn1/knVtaIPW2whMKxdRrDxvThGUA/bDS9ykv1Hf0r1Yudi
r2A22ZGHJ1NuVz79Q43xRNYkYGu9J4pdZuHsLk/6EvGnm1WPhhSRI1IoGtd1xaW
xagkGICfG3vIhohJNMPG14yKd3sk3A90E96XcdU6T2uw429GgmjC8RQXQ8e/d8U
WVHCSdqbJabGIjzqZhbHqYjak6+iP0WPF5IRU3NSJeLIZ00dd2271c1Fwuy1wJH
y9783nks22awSBUJwhK1JtwE01NjX0x7mptGwIDAQABAoIBAD5McKLMuE3EiJqS
YYxiI3bbKk00KcSjgckFTvampK2GRfn34An70khWv9SXCQPMHn5/CQA4JLH30zN
nfrXgrmTPYdA0D13rTZDrs1f316b897s/uBFmGqsSTj+0amF4v0WJpqqKMMtUB
1PshZw109h0KcKQSwgXCi2DvLx1236Jx313f1DFkQNSBAeA/f71pSTsuJP/Vn02c
IaJAh65e6mHjBFuHIE271xXGF5FknblkpyjPFVS+vxv31kTSku3BUmB6JpRC8FN
/CKymDpyXw4E77AuSIpJ5zd3WurnURUr1cU+sg8LHJyPisg4KgPhtrtMwCtZw7q2
7K1pNYECgYEA421aQybcIqp6q04U/VXJn6hfPcTqUR1bHTb1vHF1bIzbbfUR7Hu8
4kcfxHScQ0Nra0/IRFPvaS07U6U1vxFIho186pu3rxQA+gn6IC82F2InaNsLXEG
H4Kf0WniSk+ovK2UX1S9nLxRUBDUXMkCQEYDTn2t0Fa1Tr0Xf/V3uCUcGYEAWKE6
StTGS13FjVukJlhzeJ9rtN6wie111dJrc4BY8uQM28RR1lnkqXURS90PY4m6TYe
AJhrHGUsV18SuRzB0LraDwriV5UIC0JhBH6H002ws7yS/PzXmt+3f1TQImuV+xit
voFL4JyYbECCBP8sm1HxLSxIoeJ0S604kSEV08CgYBVfRR13Enzkj10E+qWfFy9
H3G2ad1g2vXctMmGntuIGye+XVvAmBmpM+2oUkn9KhnF2DFLD3a4ArvARSJAUsVE
IdkqsSF12u9HEAScoELntwrJqzEQICyRaBtL93m67Gyo/V8JQIm8tBrM1uTCemz
ud4yVv+k0Y5UzVX1JHQFI1QKBgATTcVPamRPaT27oNjVE1aTgQkhy6DydzYCQ0egm
6ov10SQnHJP6GQ2kkrYJjrdL5cX3Ynkp1x0LhS9N7hy1t1/Fw1RMASdIHfU4K
ZhmUR82EH9q16CgBvk8tiIJbIsNx60X80E2JHBIP21YeQYImB0tHArFM1v5p/fCR
Kwf7AoGAKB2W2svetw2d2P9PYHX/sKukxcxShfyDa5xbU1e1+HtD/Fs2T+hvJcJA
dgNrvv12b1dzRou0J9oeaISwrz7F3gHTSk8sRJ0cckISkkV8n13zkeu8JzPeu1I
BEy8Ggrb0h8FEFE9K1oFh/NTiySD2q1zYI+31JK2ogC5Q2CAE=
-----END RSA PRIVATE KEY-----
user@ubuntu:~$

```

Private Key

```

-----END RSA PRIVATE KEY-----
user@ubuntu:~$ cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCr1nFA5CKxICifqnPVSW/0VsSD56pqr170hqcs4dL5F7fkUK8MxLU0VJa0/
bX5pefX+SdK1U92nCEwrF1GsPG90E2QD9sNL3Iq+Id86uV152KvYDb2KycmJ3BXPy1DjffE1iRgbd2P11m4ewuT/oS+A2aV2
eFJE1iig2N3XFpbFqCQYgJ8be+kGiGM0w8aLjKQPeYTCd3QT3pdx1TpPbC/jb0VyCaNzxGpdDx793xRa8dxJ2puMBsYiP0pmFs
gyNorr618Ny/khfTc1KN4shk4N13bZnvVvYUxBKXAKfL3vzecqxl1r8IFQnCeQumrATTU2nfThuanG0b user@ubuntu
user@ubuntu:~$

```

Public Key