Name:

Red Hat Lab – Chapter 10

BC

Use **Red Hat Lab Environment** to complete the lab. Issue the following commands immediately before step 1:

```
history -c
history -w
```

These commands should be repeated for each user@machine prompt. See boxes below.

Paste a screenshot in the box below of the command output from the command below. Include the command itself in the screenshot:

`lab grade ssh-review`

```
[student@workstation ~]$ lab grade ssh-review

Grading lab.

 · Checking lab systems ..................................... SUCCESS
 · Verifying correct SSH key-based authentication configuration ..... SUCCESS
 · Verifying prevention of root user SSH login to serverb ........... SUCCESS
 · Verifying prevention of SSH password-based login to serverb ..... SUCCESS

Overall lab grade: PASS
```

Issue the command `history` after the last step for each user@machine prompt. Copy the command itself and the full history of commands. Paste in the boxes below:

production1@servera

```
[production1@servera ~]$ history
    1  ssh root@serverb
    2  ssh production2@serverb
    3  ssh production1@serverb
    4  history
```

root@serverb

```
[root@serverb ~]# history
    1  systemctl status psacct
    2  systemctl start  psacct
    3  systemctl is-active psacct
    4  systemctl enable psacct
    5  systemctl is-enabled psacct
    6  systemctl status rsyslog
    7  systemctl stop rsyslog
    8  systemctl is-active rsyslog
    9  systemctl disable rsyslog
   10  systemctl is-enabled rsyslog
   11  systemctl reboot
   12  history
   13  exit
   14  w --from
   15  exit
   16  vim /etc/ssh/sshd_config
   17  systemctl reload sshd.service
   18  vim /etc/ssh/sshd_config
   19  systemctl reload sshd
   20  cat /etc/ssh/sshd_config
   21  history
```

Red Hat Lab – Chapter 11

Use **Red Hat Lab Environment** to complete the lab. Issue the following commands immediately before step 1:

```
history -c
history -w
```

These commands should be repeated for each user@machine prompt. See boxes below.

Paste a screenshot in the box below of the command output from the command below. Include the command itself in the screenshot:

`lab grade l0g-review`

```
[student@workstation ~]$ lab start logs-review

Starting lab.

 · Checking lab systems ..................................................... SUCCESS
 · Ensuring chrony is installed on serverb ................................. FAIL
    - Command did not exit with the expected code
    - Expected: 0, Received: 255
    - Cannot continue starting lab
```

Issue the command `history` after the last step for student@serverb. Paste a screenshot of the history in the proper box below. Include the command itself and the full history of commands.

student@serverb

```
[student@serverb ~]$ history
    1  sudo -i
    2  history
    3  sudo -i
    4  exit
    5  w --from
    6  exit
    7  su - operator1
    8  exit
    9  su - operator1
   10  exit
   11  su - operator2
   12  ssh root@servera
   13  su - operator2
   14  exit
   15  chrony
   16  yum install chrony
   17  sudo yum install chrony
   18  yum update
   19  sudo yum update
   20  exit
   21  tzselect
   22  sudo timedatectl set-timezone America/Jamaica
   23  timedatectl
   24  date
   25  date -d "-30 minutes"
   26  journalctl --since 04:32:00 --until 05:02:00
   27  sudo vim /etc/rsyslog.d/auth-errors.conf
   28  sudo systemctl restart rsyslog
   29  logger -p authpriv.alert "Logging test authpriv.alert"
   30  sudo tail /var/log/auth-errors
   31  history
```

Lab Manual

Use the VirtualBox RHELv9 virtual machine for this lab. Do not use the Red Hat Lab Environment. Issue the following commands in the Terminal window before starting the lab on the next page:

```
history -c
history -w
```

Repeat these commands for root@RHELv8 if necessary.

Paste the results of the history command in the box at the end of the lab.

Lab 15: Using SSH

In this lab, the RHELv9 VM will be considered as a powerful machine to act as the SSH server with IP address 10.0.0.1. The Ubuntu VM will be a thin client with limited resources and will act as the SSH client with IP address 10.0.0.2.

The Ubuntu client wishes to connect to the powerful RHELv9 server. The Ubuntu client will access programs that will run on the RHELv9 server but will see them displayed on is Ubuntu display. The RHELv9 server will provide the CPU and RAM resources to run the programs.

In this scenario, the RHELv9 VM is the server or remote machine. The Ubuntu VM is the client or local machine (where the user is actually sitting).

1. Check to see if the ssh client is installed on the Ubuntu machine. Just type **ssh** at a terminal. If you see a help screen, then ssh is installed.

2. Check to see if the ssh server is installed on the RHELv9 machine. Type **service sshd status** to see if the ssh server is installed and active. To activate the service, use **service sshd start**. If the service is not installed, run **yum install openssh-server**.

3. For convenience, rename both VMs as follows:
    a. RHELv9 VM: **sudo hostname RHELv**9**-remote-server-10.0.0.1**
    b. Ubuntu VM: **sudo hostname Ubuntu-local-client-10.0.0.2**

4. Before continuing, change the IP addresses of both VMs to match what is indicated above.

5. Now try to establish an ssh session between the Ubuntu client and the RHELv9 server: **ssh 10.0.0.1**
    a. Since we don't have the server's public key, we can't verify we are truly talking to the server. We will take a risk and say yes to continue connecting. The server's public key will be added our list of known hosts and will now be trusted. The server's public key will be stored in *~/.ssh/known_hosts*. You may verify this by using **cat ~/.ssh/known_hosts**.

6. Type hostname to see that you are truly logged into the RHELv9 server. Then type **exit** to close the connection.

7. Try to connect again using **ssh 10.0.0.1**. This time the server will identify us by our password, but we authenticate the server with its public key from step 5.

8. The SSH client configuration file is */etc/ssh/ssh_config*. The server configuration file is */etc/ssh/sshd_config*. Edit the server configuration file to allow X11Forwarding. This activates SSH's tunneling features which will allow us to run X programs (GUI) on the server from our client VM. Remove the # sign at the beginning of the line to uncomment it.

9. We authenticate the server using its public key. The server may authenticate the client using the client's public key, but first we need to create one. Create both and RSA and DSA pair of keys (private/public): **ssh-keygen -t rsa** and **ssh-keygen -t dsa**.

10. Verify the client keys have been created: **ls ~/.ssh**

11. Check the permissions of the keys using **ls -l ~/.ssh**. Note that only the owner may read/write to the private keys, everyone is able to read the public keys.

12. Now let's send the client's public key (Ubuntu) to the server:
    sudo ssh-copy-id -i ~/.ssh/id_rsa.pub student@10.0.0.1

13. Display the client's RSA public key: **cat ~/.ssh/id_rsa.pub**

14. Log into the server: **ssh 10.0.0.1** (you'll need to type the passphrase if you used one the first time)

15. Now that you are logged into the server, display the RSA public key from Ubuntu and compare to the one displayed in step 13: **cat ~/.ssh/authorized_keys**

16. Now let's try to run a GUI program on the server while seeing it on our client. Do the following:

    a. Ubuntu client: enable remote access by adding **enable=yes** under the **[xdmcp]** section of **/etc/gdm3/custom.conf**

    b. Ubuntu: Tell Ubuntu to accept for display in its X server data that originates on the RHELv9 server: **xhost +10.0.0.1**

    c. Ubuntu: Determine the name of your display: **xdpyinfo** followed by **echo $DISPLAY**

    d. Ubuntu: **ssh student@10.0.0.1 -X**

    e. RHELv9: **export DISPLAY=Ubuntu:0** (use the name of your display from step c)

    f. RHELv9: check to make sure the edit in step 8 was completed within the RHELv9 server's sshd_config file

    g. Ubuntu: Type **hostname** to verify you are logged into the RHELv9 server from the Ubuntu client machine.

    h. Run **/usr/bin/baobab** which is the *Disk Usage Analyzer* to see a GUI program on your Ubuntu display that is really running on the RHELv9 server.

    i. When finished, type **exit** to return to your Ubuntu local display.

    j. Run **xhost -10.0.0.1** to remove from the access control list.

17. Submit your command history for grading.

```
student@RHELv8:~$ history
    1  history -w
    2  hostname
    3  exit
    4  sudo vim /etc/ssh/sshd_config
    5  ssh-keygen -t rsa
    6  ssh-keygen -t dsa
    7  ls ~/.ssh
    8  ls -l ~/.ssh
    9  cd /home/student/.ssh
   10  ls
   11  rm id_dsa
   12  rm id_rsa
   13  ls
   14  rm id_dsa.pub
   15  rm id_rsa.pub
   16  ls
   17  exit
   18  sudo vim /etc/ssh/sshd_config
   19  ssh-keygen -t rsa
   20  ssh-keygen -t dsa
   21  ls ~/.ssh
   22  ls -l ~/.ssh
   23  sudo ssh-copy-id -i ~/.ssh/id_rsa.pub student@10.0.0.1
   24  cat ~/.ssh/id_rsa.pub
   25  exit
   26  cat ~/.ssh/authorized_keys
   27  exit
   28  export DISPLAY=Ubuntu:0
   29  cat /etc/ssh/sshd_config
   30  sudo cat /etc/ssh/sshd_config
   31  sudo vim /etc/ssh/sshd_config
   32  hostname
   33  /usr/bin/baobab
   34  xhost -10
   35  xhost -10.0.0.1
   36  history

student@ubuntu:~$ history
    1  history -w
    2  ssh
    3  sudo hostname Ubuntu-local-client-10.0.0.2
    4  ip a
    5  ifconfig ens34 10.0.0.2/24
    6  sudo ifconfig ens34 10.0.0.2/24
    7  sudo vim /etc/network/interfaces
    8  sudo vi /etc/network/interfaces
    9  sudo nano /etc/network/interfaces
   10  ip a
   11  cd /etc/netplan
   12  ls
   13  cat 01-network-manager-all.yaml
```

Lab 16: Logs and NTP

1. Display the last ten lines add to the messages log file.
2. View journal data.
3. Display log entries for the past week.
4. Display log entries for the current system boot.
5. Display an overview of current time settings.
6. Set the timezone to the time in California, USA.
7. Reset the timexone to the time in Missouri, USA.

```
student@RHELv8:~$ history
    1  history -w
    2  tail -10 /var/log/messages
    3  sudo tail -10 /var/log/messages
    4  journalctl
    5  sudo ls /var/log/journal
    6  journalctl --since "2022-07-05" --until "2022-07-12"
    7  journalctl -b
    8  journalctl --list-boots
    9  timedatectl
   10  timedatectl set-timezone America/Los_Angeles
   11  timedatectl list-timezones
   12  timedatectl set-timezone America/Chicago
   13  timedatectl list-timezones
   14  history
```