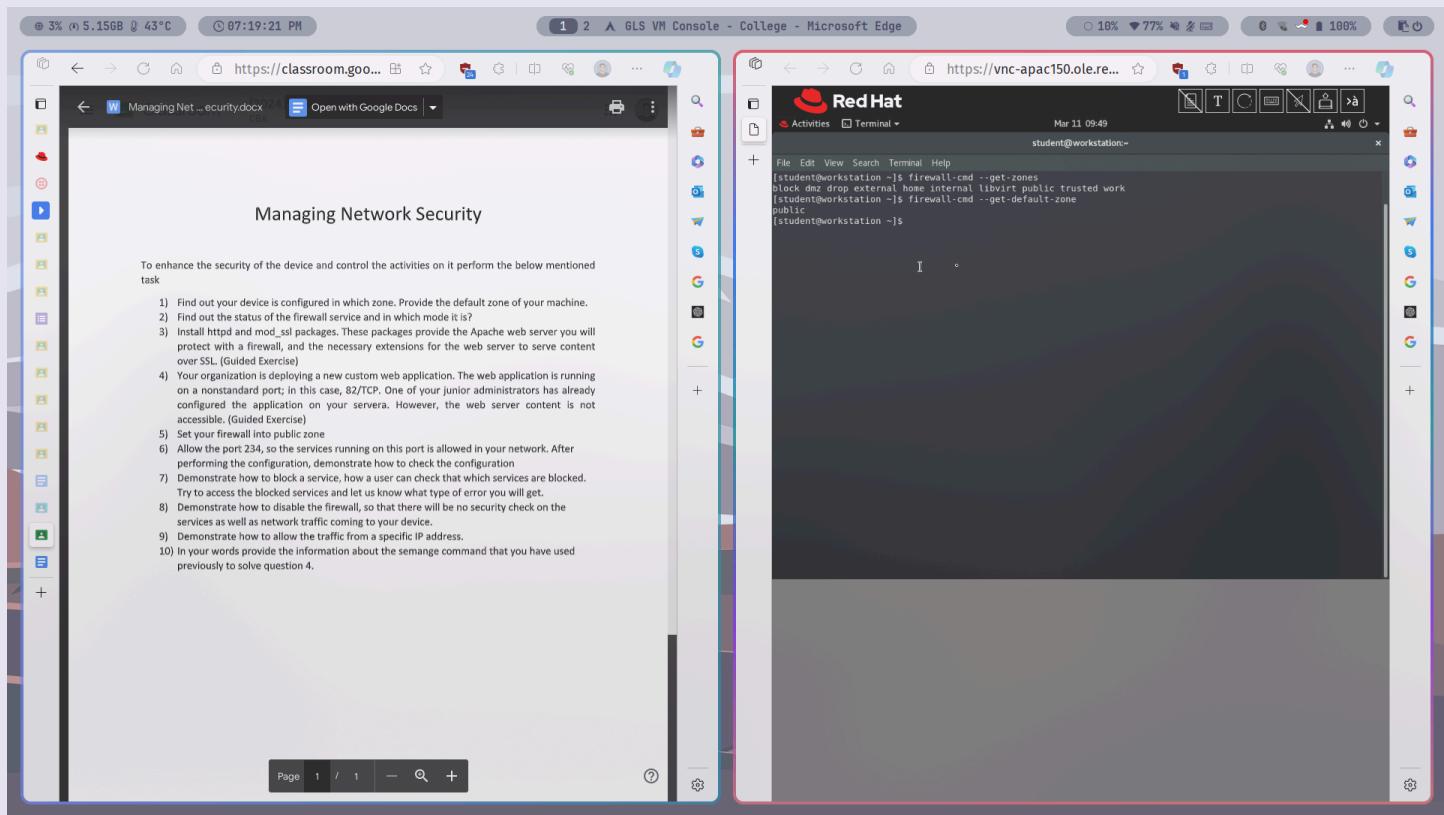


Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

Aim : To enhance the security of the device and control the activities on it perform the below mentioned task

1) Find out your device is configured in which zone. Provide the default zone of your machine.

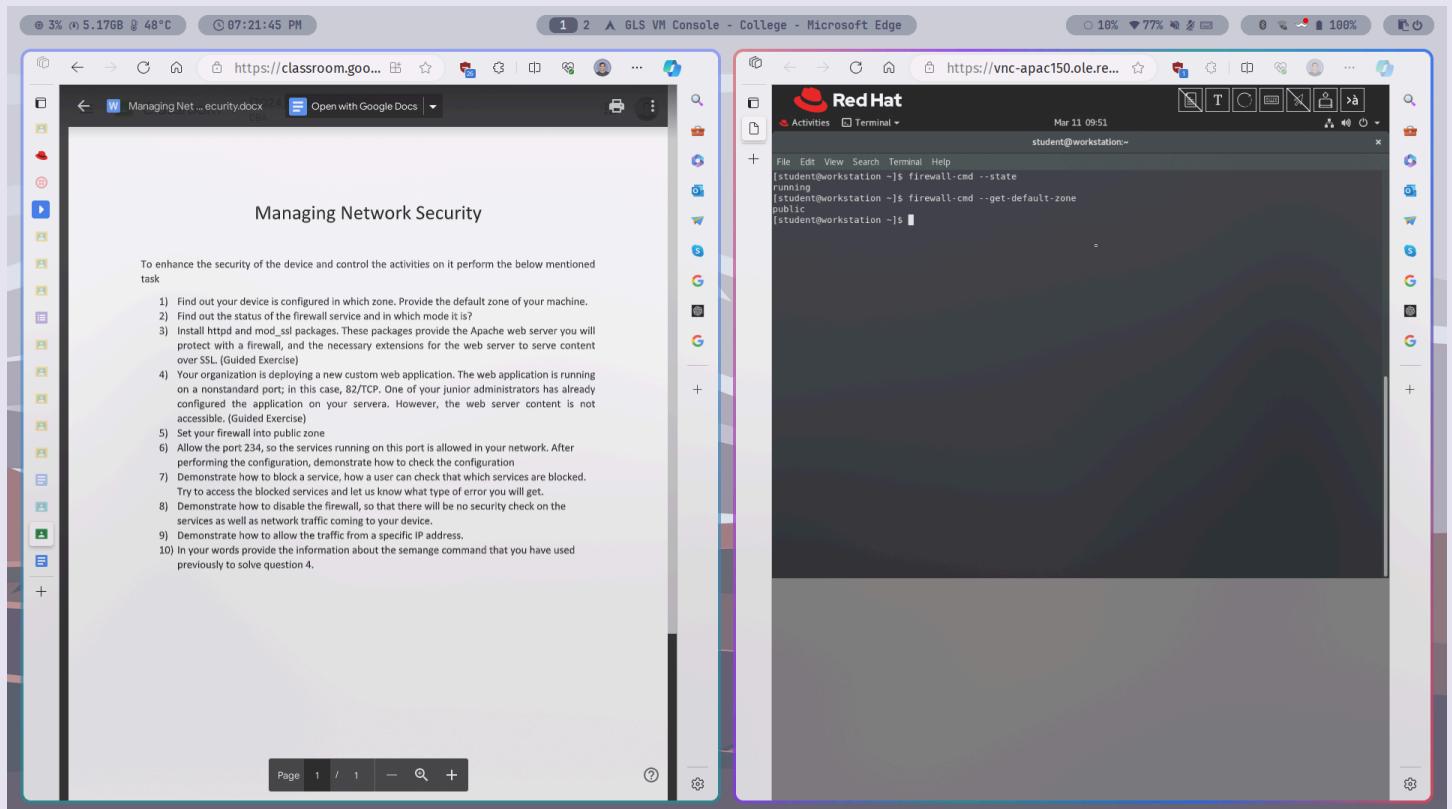


Commands :

- **firewall-cmd –get-zones** (to list all available zone options for firewall)
- **firewall-cmd –get-default-zone** (to get the default zone of device's firewall configuration)

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

2) Find out the status of the firewall service and in which mode it is?



Command : **firewall-cmd –state**

Currently, the firewall daemon is in ***running*** state/mode.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 9

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

3) Install httpd and mod_ssl packages. These packages provide the Apache web server you will protect with a firewall, and the necessary extensions for the web server to serve content over SSL. (Guided Exercise)

Steps :

- a. Run the lab setup script and login to servera via ssh. Thereafter, install httpd and mod_ssl

The screenshot shows a dual-monitor setup. The left monitor displays the Red Hat Academy website for a 'Guided Exercise: Managing Server Firewalls'. The right monitor shows a terminal window titled 'Red Hat' with the command 'lab netsecurity-firewalls start' being run. The terminal output shows the installation of httpd and mod_ssl packages.

```
[student@workstation ~]$ lab netsecurity-firewalls start
Starting lab.
Preparing server for lab exercise work:
  - Check server connectivity..... SUCCESS
  - Enable and start cockpit.socket on server..... SUCCESS
[student@workstation ~]$ ssh student@servera
web console: https://servera.lab.example.com:9090/ or https://172.25.250.10:9090/
This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register
Last login: Tue Sep 1 09:19:05 2020 from 172.25.250.9
[student@servera ~]$ sudo yum install httpd mod_ssl
[sudo] password for student:
Red Hat Enterprise Linux 8.2 BaseOS (dvd)
Red Hat Enterprise Linux 8.2 AppStream (dvd)
Dependencies resolved.
=====
Package           Arch      Version            Repository        Size
Installing:
httpd           x86_64    2.4.37-21.module+el8.2.0+50008+cca404a3   rhel-8.2-for-x86_64-appstream-rpms 1.4 M
mod_ssl         x86_64    1:2.4.37-21.module+el8.2.0+50008+cca404a3   rhel-8.2-for-x86_64-appstream-rpms 132 k
Installing Dependencies:
apr              x86_64    1.6.3-9.el8          rhel-8.2-for-x86_64-appstream-rpms 125 k
apr-util        x86_64    1.6.1-6.el8          rhel-8.2-for-x86_64-appstream-rpms 105 k
httpd-filesystem noarch   2.4.37-21.module+el8.2.0+50008+cca404a3   rhel-8.2-for-x86_64-appstream-rpms 36 k
httpd-tools     x86_64    2.4.37-21.module+el8.2.0+50008+cca404a3   rhel-8.2-for-x86_64-appstream-rpms 103 k
mod_http        x86_64    2.4.37-21.module+el8.2.0+43777+dc421495   rhel-8.2-for-x86_64-appstream-rpms 158 k
redhat-logos-ht  x86_64    0.1.1-1.el8          rhel-8.2-for-x86_64-basics-rpms 26 k
Installing Weak Dependencies:
apr-util-bdb    x86_64    1.6.1-6.el8          rhel-8.2-for-x86_64-appstream-rpms 25 k
apr-util-openssl x86_64    1.6.1-6.el8          rhel-8.2-for-x86_64-appstream-rpms 27 k
```

Command : **sudo yum install httpd mod_ssl** (to install the apache server and SSL modules packages using yum package manager with root privileges)

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 9

b. Create index.html file inside apache root and enable and start the http daemon

The image shows a Microsoft Edge browser window with two tabs. The left tab displays a Red Hat Academy guide for setting up Apache on a server. The right tab shows a terminal session on a Red Hat server. The terminal session includes the following commands:

```
[student@servera ~]$ sudo bash -c "echo 'I am servera.' > /var/www/html/index.html"
[sudo] password for student: student
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
[student@servera ~]$ sudo systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service.
[student@servera ~]$
```

Commands :

- **sudo bash -c “echo ‘I am servera’ > /var/www/html/index.html”** (to append text to the index.html file (create if not exists) of path specified, this command echo is run via bash shell with -c option which specifies to accept the whole string as command)
- **sudo systemctl enable --now httpd.service** (to enable http daemon service and start now only)

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

- c. Now, exit from servera and try to reach servera site from workstation via curl, it should give an error because it is not configured and firewall isn't allowing it

The screenshot shows a dual-monitor setup. The left monitor displays a Microsoft Edge browser window titled 'GLS VM Console - College - Microsoft Edge'. The URL is <https://rha.ole.redhat.com/>. The page content is a Red Hat Academy course, specifically 'Red Hat Academy Version 15.9'. The right monitor shows a Red Hat terminal window titled 'student@workstation'. The terminal output shows several commands being run:

- [student@servera ~]\$ exit
- logout
- Connection to servera closed.
- [student@workstation ~]\$
- 6. From workstation, attempt to access your web server on servera using both the cleartext port 80/TCP and the SSL encapsulated port 443/TCP. Both attempts should fail:
 - This command should fail:

```
[student@workstation ~]$ curl http://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 80: No route to host
```
 - This command should also fail:

```
[student@workstation ~]$ curl -k https://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 443: No route to host
```
- 7. Log in to servera as the student user.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```
- 8. On servera, make sure that the nftables service is masked and the firewalld service is enabled and running.

```
[student@servera ~]$ sudo systemctl status nftables
[sudo] password for student:
● nftables.service - Netfilter Tables
  Loaded: loaded (/usr/lib/systemd/system/nftables.service)
  Active: disabled; vendor preset: disabled
    Docs: man:nft(8)
```
- 81. Determine whether the status of the nftables service is masked.

```
[student@servera ~]$ sudo systemctl status nftables
[sudo] password for student:
● nftables.service - Netfilter Tables
  Loaded: loaded (/usr/lib/systemd/system/nftables.service)
  Active: disabled; vendor preset: disabled
    Docs: man:nft(8)
```

Commands :

- **curl <http://servera.lab.example.com>**
- **curl -k <http://servera.lab.example.com>**

(both for accessing the given URL via curl command, but -k for allowing insecure connections also)

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

d. Now, login to servera and check the status of nftables service

The screenshot shows a dual-monitor setup. The left monitor displays a Red Hat Academy page with a sidebar containing links like 'Access Courses and Assignments', 'Access Student Resources', 'Give Feedback', and 'Get Support'. The main content area shows steps for a practical exercise:

7. Log in to servera as the student user.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```
8. On servera, make sure that the nftables service is masked and the firewalld service is enabled and running.
- 8.1. Determine whether the status of the nftables service is masked.

```
[student@servera ~]$ sudo systemctl status nftables
[sudo] password for student:
● nftables.service - Netfilter Tables
  Loaded: loaded (/usr/lib/systemd/system/nftables.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
    Docs: man:nft(8)

The results show that nftables is disabled and inactive but not masked. Run the following command to mask the service.
```
- 8.2. Verify that the status of the nftables service is masked.

```
[student@servera ~]$ sudo systemctl status nftables
● nftables.service
  Loaded: masked (Reason: Unit nftables.service is masked.)
  Active: inactive (dead)
```
- 8.3. Verify that the status of the firewalld service is enabled and running.

```
[student@servera ~]$ sudo systemctl status firewalld
● firewalld.service - Firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2019-05-22 15:36:02
```

The right monitor shows a terminal window titled 'Red Hat' with the command 'sudo systemctl status nftables' run. The output shows the nftables service is masked and inactive. The terminal also displays a message about Red Hat Insights registration.

Command : **sudo systemctl status nftables**

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

e. Mask the Netfilter tables service

The screenshot shows a Microsoft Edge browser window with two tabs open. The left tab displays a Red Hat Academy guide titled "Mask the Netfilter tables service". It includes steps 7 through 8.3, with code snippets for each step. Step 7 shows the command [student@workstation ~]\$ ssh student@servera. Step 8.1 shows [student@servera ~]\$ sudo systemctl status nftables. Step 8.2 shows [student@servera ~]\$ sudo systemctl mask nftables. Step 8.3 shows [student@servera ~]\$ sudo systemctl status firewalld. The right tab shows a terminal session on a Red Hat server. The user logs in and runs the commands from the guide. The terminal output shows the nftables service being masked and the firewalld service being enabled.

```
[student@workstation ~]$ ssh student@servera
[student@servera ~]$ sudo systemctl status nftables
[student@servera ~]$ sudo systemctl mask nftables
[student@servera ~]$ sudo systemctl status firewalld
```

```
Last login: Mon Mar 11 09:37:18 2024 from 172.25.250.9
[student@servera ~]$ sudo systemctl status nftables
[sudo] password for student:
● nftables.service - Netfilter Tables
  Loaded: loaded (/usr/lib/systemd/system/nftables.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
    Docs: man:nft(8)

[student@servera ~]$ sudo systemctl mask nftables
Created symlink /etc/systemd/system/nftables.service → /dev/null.

[student@servera ~]$ sudo systemctl status nftables
● nftables.service - Netfilter Tables
  Loaded: masked (Reason: Unit nftables.service is masked.)
  Active: inactive (dead)
    Docs: man:nft(8)
```

Commands :

- **sudo systemctl mask nftables** (to mask the service given)
- **sudo systemctl status nftables** (to check its status)

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 9

- f. Now check the status of firewall daemon and it would be in running state and exit from ssh session

The screenshot shows a dual-monitor setup. The left monitor displays the Red Hat Academy website at <https://rha.ole.redhat.com>. The right monitor displays a terminal window titled "GLS VM Console - College - Microsoft Edge" showing a Linux command-line interface.

In the terminal window, the user runs the command `sudo systemctl status firewalld`. The output shows:

```
[student@servera ~]$ sudo systemctl status firewalld
● firewalld.service - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2019-05-22 15:36:02
  CDT; 5min ago
    Docs: man:firewalld(1)
   Main PID: 703 (firewalld)
     Tasks: 2 (limit: 11405)
    Memory: 29.8M
       CGroup: /system.slice/firewalld.service
               └─703 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nrepid

May 22 15:36:01 servera.lab.example.com systemd[1]: Starting firewalld - dynamic firewall daemon...
May 22 15:36:02 servera.lab.example.com systemd[1]: Started firewalld - dynamic firewall daemon.
```

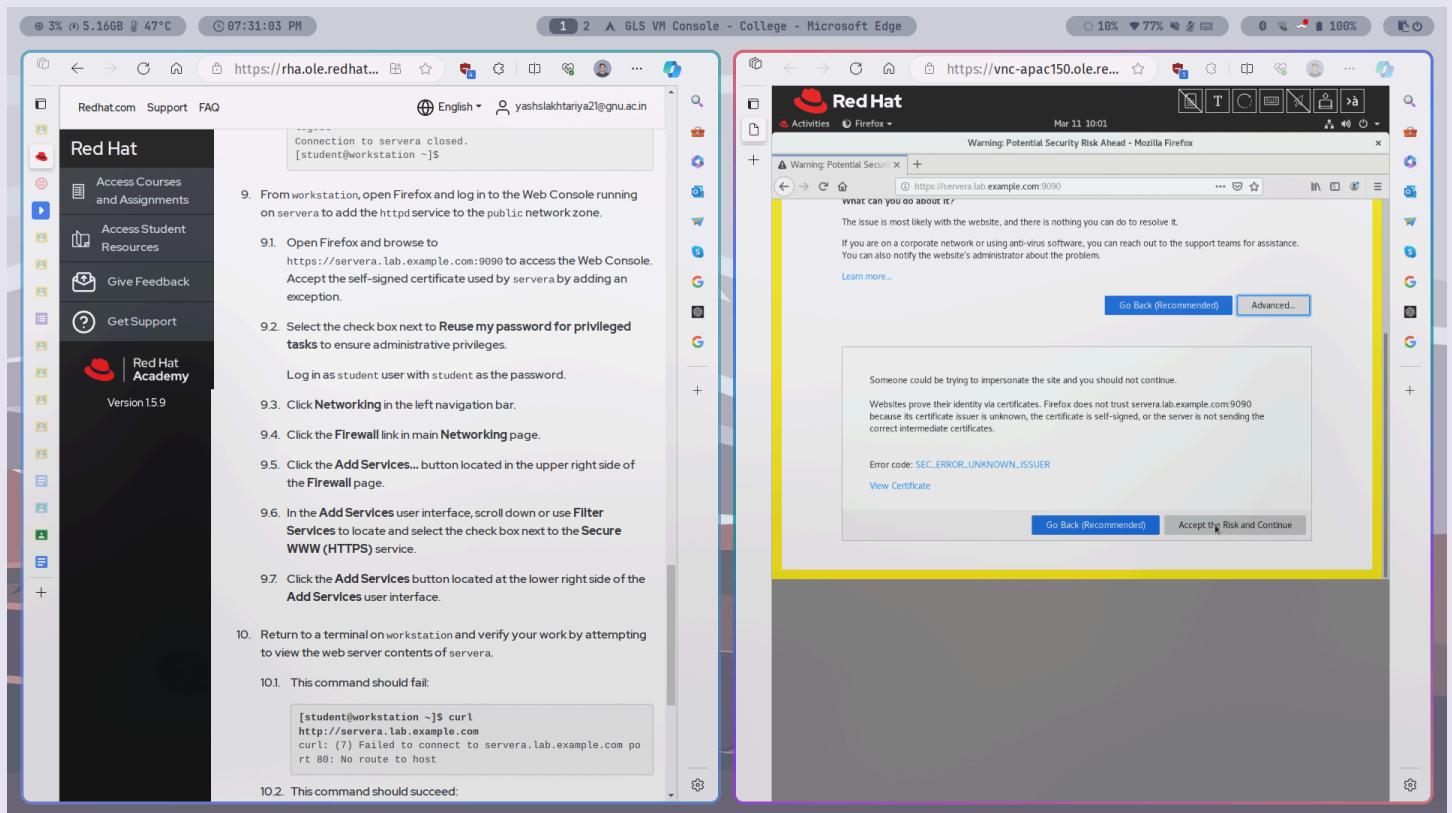
After checking the status, the user exits the terminal session with `exit`.

On the left monitor, the Red Hat Academy interface includes sections for "Access Courses and Assignments", "Access Student Resources", "Give Feedback", and "Get Support". It also features a "Red Hat Academy" logo and "Version 15.9".

Command : **`sudo systemctl status firewalld`**

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

g. Now, go to browser and accepting the self-signed certificate, go to servera lab URL with port 9090



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

h. Login as student with password student and Reuse password forprivileged tasks option enabled

The screenshot shows a dual-monitor setup. The left monitor displays a Microsoft Edge browser window titled 'GLS VM Console - College - Microsoft Edge' with the URL <https://rha.ole.redhat.com>. The page content is a step-by-step guide for a practical exercise, starting with a terminal message: 'Connection to servera closed. [student@workstation ~]\$'. The steps are numbered 9 through 10.2. Step 9: 'From workstation, open Firefox and log in to the Web Console running on servera to add the httpd service to the public network zone.' Step 10.1: 'This command should fail.' It shows a terminal command: '[student@workstation ~]\$ curl http://servera.lab.example.com'. Step 10.2: 'This command should succeed:' It shows a terminal command: '[student@workstation ~]\$ curl http://servera.lab.example.com'. The right monitor displays a Mozilla Firefox browser window titled 'servera.lab.example.com - Mozilla Firefox' with the URL <https://servera.lab.example.com:9090>. The page is a Red Hat Enterprise Linux login screen. The 'User name' field is filled with 'student' and the 'Password' field contains 'student'. A checked checkbox labeled 'Reuse my password for privileged tasks' is visible. Below the login form, it says 'Server: servera.lab.example.com' and 'Log in with your server user account.'

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 9

i. In Networking, tab go to Firewall and click Add service button

The image shows two Microsoft Edge browser windows side-by-side.

Left Window (Red Hat Academy): This window displays a list of steps for a practical exercise. The steps are numbered 9 through 10.2. Step 9 describes logging into the RHEL Web Console and opening the Firewall page. Step 10.1 shows a terminal command failing to connect to port 80. Step 10.2 shows a terminal command succeeding.

Right Window (RHEL Web Console Firewall): This window shows the 'Networking > Firewall' interface. It lists three services: SSH (TCP 22), DHCPv6 Client (TCP 546), and Cockpit (TCP 9090). There is a red 'Add Services' button at the top right.

Service	TCP	UDP
SSH	22	
DHCPv6 Client	546	
Cockpit	9090	

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

j. Search for and add Secure WWW service

The screenshot displays two Microsoft Edge windows side-by-side. The left window shows the Red Hat Academy interface, version 15.9, with a sidebar containing links like 'Access Courses and Assignments', 'Access Student Resources', 'Give Feedback', and 'Get Support'. The main content area shows a terminal window with the command 'curl http://servera.lab.example.com' failing due to a 'No route to host'. The right window shows the Red Hat Enterprise Linux Web Console interface, specifically the Firewall configuration page. A modal dialog titled 'Add services to Public zone' is open, showing the 'Services' tab with 'Secure WWW (HTTPS)' selected under 'Filter Services'. The status bar at the bottom of the right window indicates 'Privileged' and 'Student User'.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

k. Now, curl command with secure connection will fail but -k (insecure) will succeed.

The image shows a split-screen view. On the left is a Microsoft Edge browser window displaying a Red Hat Academy exercise titled "WWW (HTTPS) service". The exercise steps include clicking "Add Services", returning to the terminal, and attempting curl commands. One command fails due to SSL certificate issues, while another succeeds using the -k option. On the right is a Linux desktop environment terminal window titled "Red Hat" showing the same curl commands being run, with the output matching the browser's results.

WWW (HTTPS) service.

9.7. Click the Add Services button located at the lower right side of the Add Services user interface.

10. Return to a terminal on workstation and verify your work by attempting to view the web server contents of servera.

10.1. This command should fail:

```
[student@workstation ~]$ curl https://servera.lab.example.com
curl: (60) SSL certificate problem: self signed certificate in certificate chain
More details here: https://curl.haxx.se/docs/sslcerts.html
curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
```

10.2. This command should succeed:

```
[student@workstation ~]$ curl -k
https://servera.lab.example.com
I am servera
```

Note

If you use Firefox to connect to the web server, it will prompt for verification of the host certificate if it successfully gets past the firewall.

Finish

On workstation, run the lab netsecurity-firewalls finish script to complete this exercise.

```
[student@workstation ~]$ lab netsecurity-firewalls finish
```

This concludes the guided exercise.

[Previous](#) [Next](#)

Red Hat Privacy Policy Red Hat Training Policies Terms of Use All policies and guidelines

File Edit View Search Terminal Help

[student@workstation ~]\$ curl https://servera.lab.example.com
curl: (60) SSL certificate problem: self signed certificate in certificate chain
More details here: https://curl.haxx.se/docs/sslcerts.html
curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

[student@workstation ~]\$ curl -k https://servera.lab.example.com
I am servera

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

l. Finish the lab and exit from ssh

The image shows a Microsoft Edge browser window with two tabs open. The left tab displays a Red Hat Academy exercise titled "netsecurity-firewalls". It contains several steps and command-line examples. The right tab shows a terminal session on a Red Hat workstation (student@workstation). The terminal output includes commands like curl, lab netsecurity-firewalls finish, and lab finished, along with success messages.

Red Hat Academy Exercise (Left Window):

- 97. Click the Add Services button located at the lower right side of the Add Services user interface.
- 10. Return to a terminal on workstation and verify your work by attempting to view the web server contents of servera.
- 10.1. This command should fail:

```
[student@workstation ~]$ curl http://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 80: No route to host
```
- 10.2. This command should succeed:

```
[student@workstation ~]$ curl -k https://servera.lab.example.com
I am servera.
```

Note:
If you use Firefox to connect to the web server, it will prompt for verification of the host certificate if it successfully gets past the firewall.

Finish:
On workstation, run the `lab netsecurity-firewalls finish` script to complete this exercise.

```
[student@workstation ~]$ lab netsecurity-firewalls finish
```

This concludes the guided exercise.

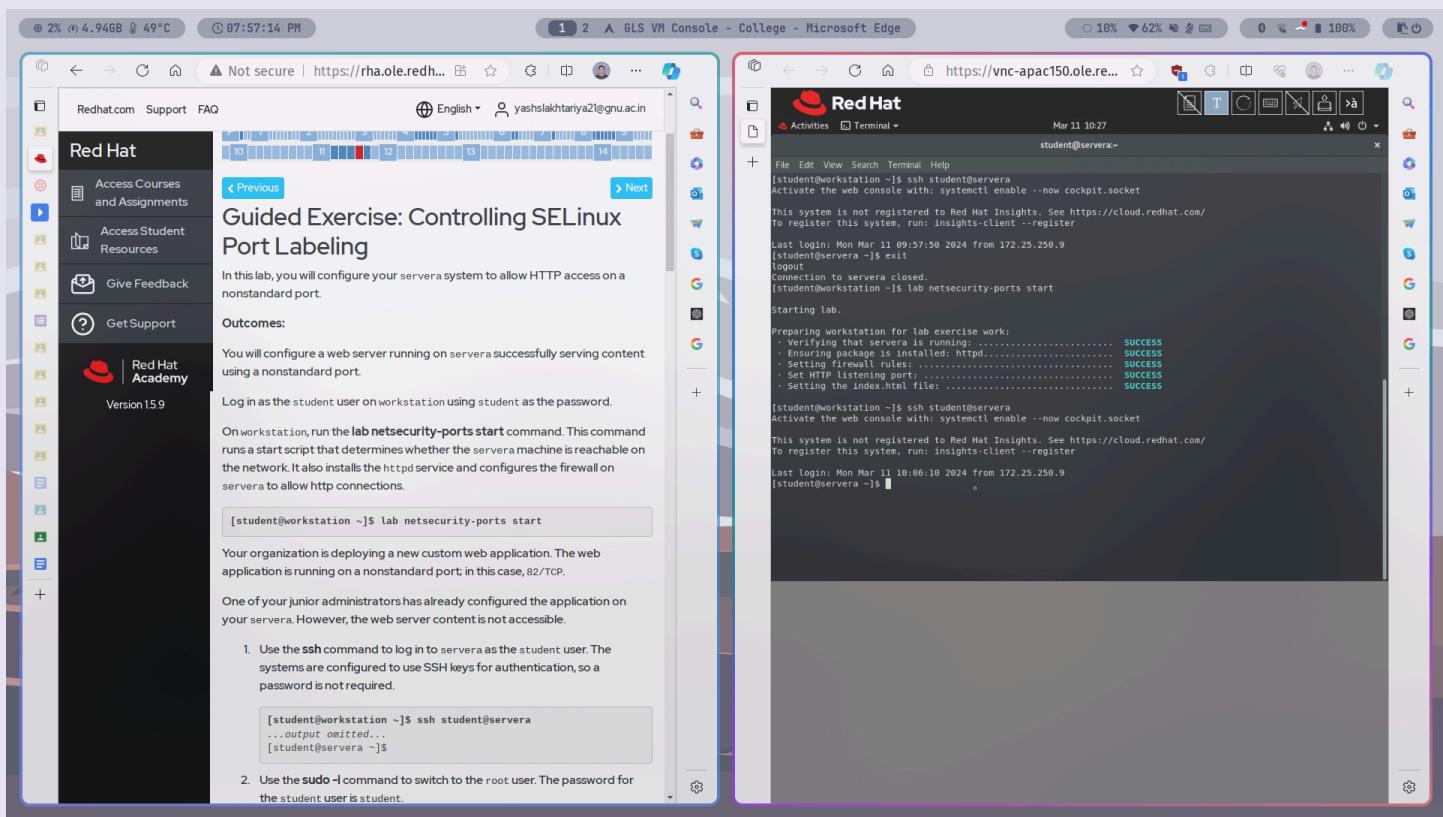
Terminal Session (Right Window):

```
File Edit View Search Terminal Help
[student@workstation ~]$ curl https://servera.lab.example.com
curl: (60) SSL certificate problem: self signed certificate in certificate chain
More details here: https://curl.haxx.se/docs/sslcerts.html
curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
[student@workstation ~]$ curl -k https://servera.lab.example.com
I am servera.
[student@workstation ~]$ lab netsecurity-firewalls finish
Cleaning up the lab on servera:
  * Remove httpd from servera..... SUCCESS
  * Reset firewall to defaults..... SUCCESS
Lab finished.
[student@workstation ~]$
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

4) Your organization is deploying a new custom web application. The web application is running on a nonstandard port; in this case, 82/TCP. One of your junior administrators has already configured the application on your servera. However, the web server content is not accessible. (Guided Exercise)

a. Start the lab and login to servera via ssh and switch to root user

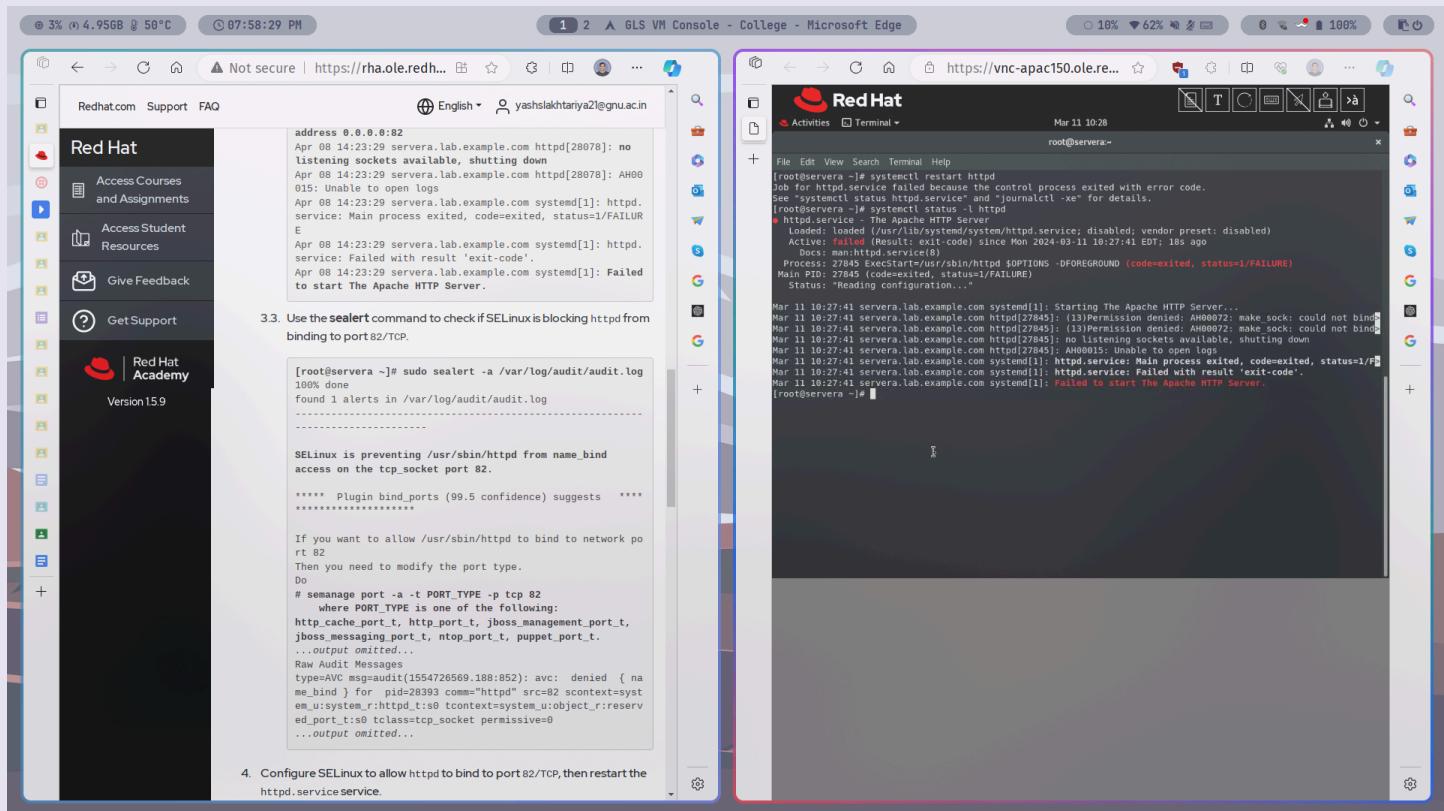


Commands :

- lab netsecurity-ports start
- ssh student@servera
- sudo -i

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

b. Restart httpd service and check the status, it should fail



The screenshot shows a Microsoft Edge browser window with two tabs. The left tab displays the Red Hat Academy website, which includes a sidebar with links like 'Access Courses and Assignments', 'Give Feedback', and 'Get Support'. The main content area shows a command-line interface (CLI) session. The right tab shows a terminal window titled 'Red Hat' with the URL 'https://vnc-apac150.ole.re...'. The terminal output shows the following commands and their results:

```
[root@servera ~]# systemctl restart httpd
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" for details.
[root@servera ~]# systemctl status -l httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since Mon 2024-03-11 10:27:41 EDT; 18s ago
     Docs: man:systemd-service(5)
   Process: 27845 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAILURE)
 Main PID: 27845 (code=exited, status=1/FAILURE)
   Status: "Reading configuration..."

Mar 11 10:27:41 servera.lab.example.com systemd[1]: Starting The Apache HTTP Server...
Mar 11 10:27:41 servera.lab.example.com httpd[27845]: (13)Permission denied: AH00072: make_sock: could not bind
Mar 11 10:27:41 servera.lab.example.com httpd[27845]: (13)Permission denied: AH00072: make_sock: could not bind
Mar 11 10:27:41 servera.lab.example.com httpd[27845]: no listening sockets available, shutting down
Mar 11 10:27:41 servera.lab.example.com httpd[27845]: AH0001: terminating
Mar 11 10:27:41 servera.lab.example.com systemd[1]: httpd.service: Main process exited, code=exited, status=1/FAILURE
Mar 11 10:27:41 servera.lab.example.com systemd[1]: httpd.service: Failed with result 'exit-code'.
Mar 11 10:27:41 servera.lab.example.com systemd[1]: Failed to start The Apache HTTP Server.
[root@servera ~]#
```

Commands :

- **systemctl restart httpd.service**
- **systemctl status -l httpd.service** (-l stands for giving full output, without elipsizing unit names)

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 9

- c. Use the sealert command to check if SELinux is blocking httpd from binding to port 82/TCP.

The screenshot shows a dual-monitor setup. The left monitor displays a Red Hat Academy course page titled "SELinux is preventing /usr/sbin/httpd from name_bind access on the tcp_socket port 82." It includes instructions to use semanage to allow httpd to bind to port 82. The right monitor shows a terminal window with the command "sudo sealert -a /var/log/audit/audit.log" run as root. The output shows SELinux preventing httpd from binding to port 82 and suggests using semanage to allow it.

```
[root@server ~]# sudo sealert -a /var/log/audit/audit.log
100% done
found 1 alerts in /var/log/audit/audit.log

SELinux is preventing httpd from name bind access on the tcp_socket port 82.
***** Plugin bind_ports (99.5 confidence) suggests *****
***** Plugin catchall (1.49 confidence) suggests *****

If you want to allow httpd to bind to network port 82
then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 82
    where PORT_TYPE is one of the following:
http_cache_port_t, http_port_t, jboss_management_port_t,
jboss_messaging_port_t, ntop_port_t, puppet_port_t.
...output omitted...
Raw Audit Messages
type=AVC msg=audit(1584726569.188:852): avc: denied { name_bind } for pid=28393 comm="httpd" src=82 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket permissive=0
...output omitted...

[root@server ~]# semanage port -l | grep http
http_cache_port_t          tcp    8080, 8118, 8123,
10001-10010
http_cache_port_t          udp    3130
http_port_t                 tcp    80, 81, 443, 488,
8008, 8009, 8443, 9000
pegasus_http_port_t         tcp    5988
pegasus_https_port_t        tcp    5999

http_port_t contains the default HTTP ports, 80/TCP and 443/TCP.
This is the correct port type for the web server.

42. Use the semanage command to assign port 82/TCP the http_port_t type.
```

Command : **sudo sealert -a /var/log/audit/audit.log**

Explanation : It records the results of the analysis and signals any clients which have attached to the setroubleshootd daemon that a new alert has been seen. sealert can be run in either a GUI mode or a command line mode. In both instances sealert runs as a user process with the privileges associated with the user. The option **-a** stands for scanning the log file and analyse it. (Only available for SELinux)

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

d. Use the semanage command to find an appropriate port type for port 82/TCP and restart httpd daemon

The screenshot shows a dual-terminal session on a Red Hat server. The left terminal displays a web browser window for 'Red Hat Academy' showing a guide on SELinux port management. The right terminal is a root shell where the user performs the following steps:

- Configures SELinux to allow httpd to bind to port 82/TCP, then restarts the httpd.service service.
- Uses the **semanage** command to find an appropriate port type for port 82/TCP.

```
[root@servera ~]# semanage port -l | grep http
http_cache_port_t          tcp    8080, 8118, 8123,
10001-10010
http_cache_port_t          udp    3130
http_port_t                tcp    80, 81, 443, 488,
8008, 8009, 8443, 9000
pegasus_http_port_t        tcp    5988
pegasus_https_port_t       tcp    5989
```

This is the correct port type for the web server.
- Uses the **semanage** command to assign port 82/TCP the **http_port_t** type.

```
[root@servera ~]# semanage port -a -t http_port_t -p tcp
82
```
- Uses the **systemctl** command to restart the **httpd.service** service. This command should succeed.

```
[root@servera ~]# systemctl restart httpd.service
```
- Checks if the web server is now accessible on port 82/TCP. Uses the **curl** command to access the web service from a workstation.

```
[root@servera ~]# curl http://servera.lab.example.com:82
Hello
```
- In a different terminal window, checks whether you can access the new web service from workstation. Uses the **curl** command to access the web service from workstation.

```
[root@servera ~]# curl http://servera.lab.example.com:82
Hello
```

Raw Audit Messages and Hash output from the right terminal:

```
Raw Audit Messages
type=AVC  avc=auditif(l710167261.779:985: avc: denied { name bind } for pid=27845 comm="httpd" src=82 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket permissive=0

Hash: httpd,httpd_t,reserved_port_t,tcp_socket,name_bind

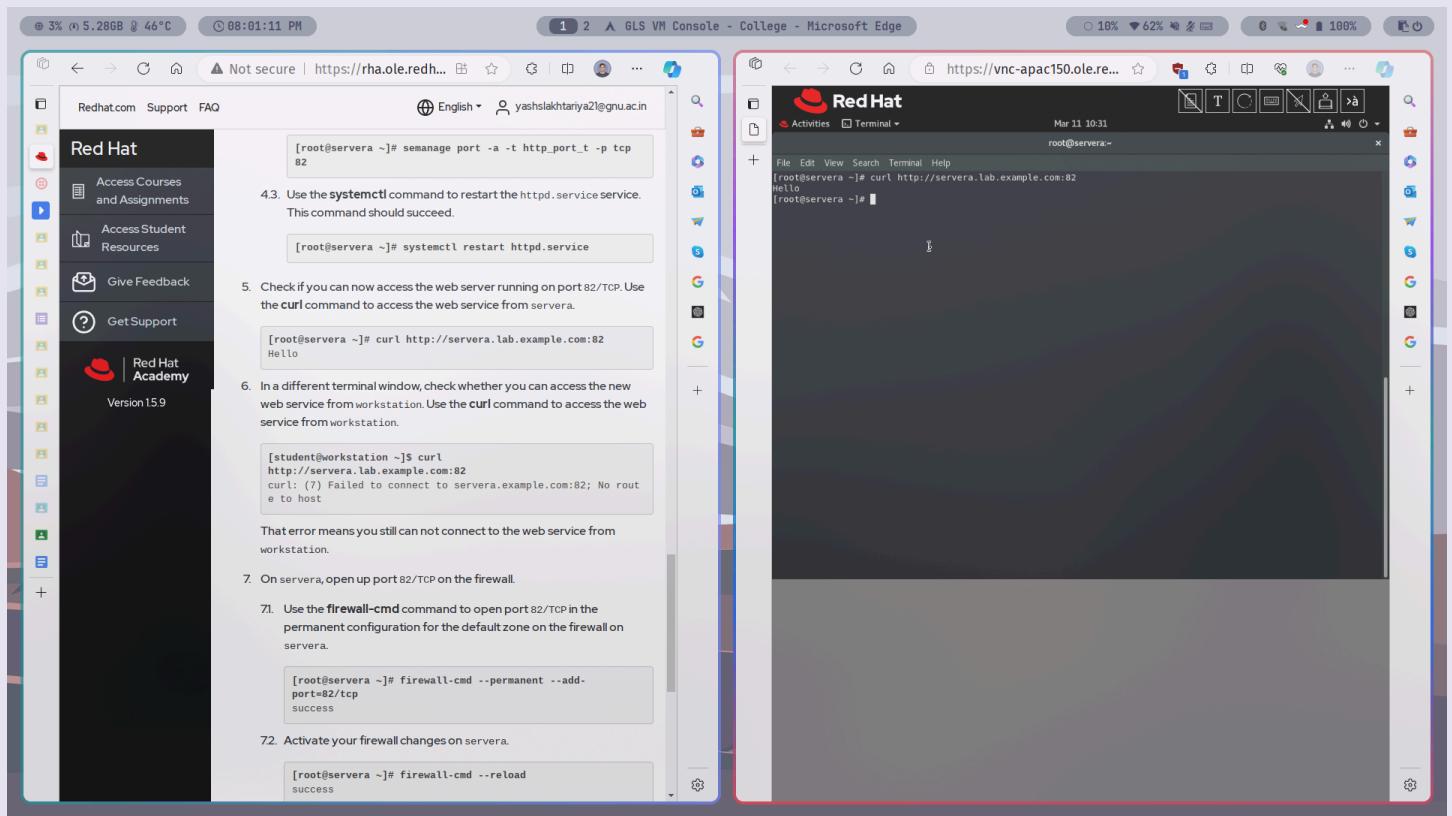
[root@servera ~]# semanage port -l | grep http
http_cache_port_t          tcp    8080, 8118, 8123, 10001-10010
http_cache_port_t          udp    3130
http_port_t                tcp    80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t        tcp    5988
pegasus_https_port_t       tcp    5989
[root@servera ~]#
[root@servera ~]# semanage port -a -t http_port_t -p tcp 82
[root@servera ~]#
[root@servera ~]# systemctl restart httpd
[root@servera ~]#
```

Commands :

- **semanage port -a -t http_port_t -p tcp 82** (SELinux manage command, port to manage port and **-a** stands for add object, **-t** for object type mentioned here (**http_port_t**) and **-p** for protocol of specified port, here **tcp 82**)
- **systemctl restart httpd**

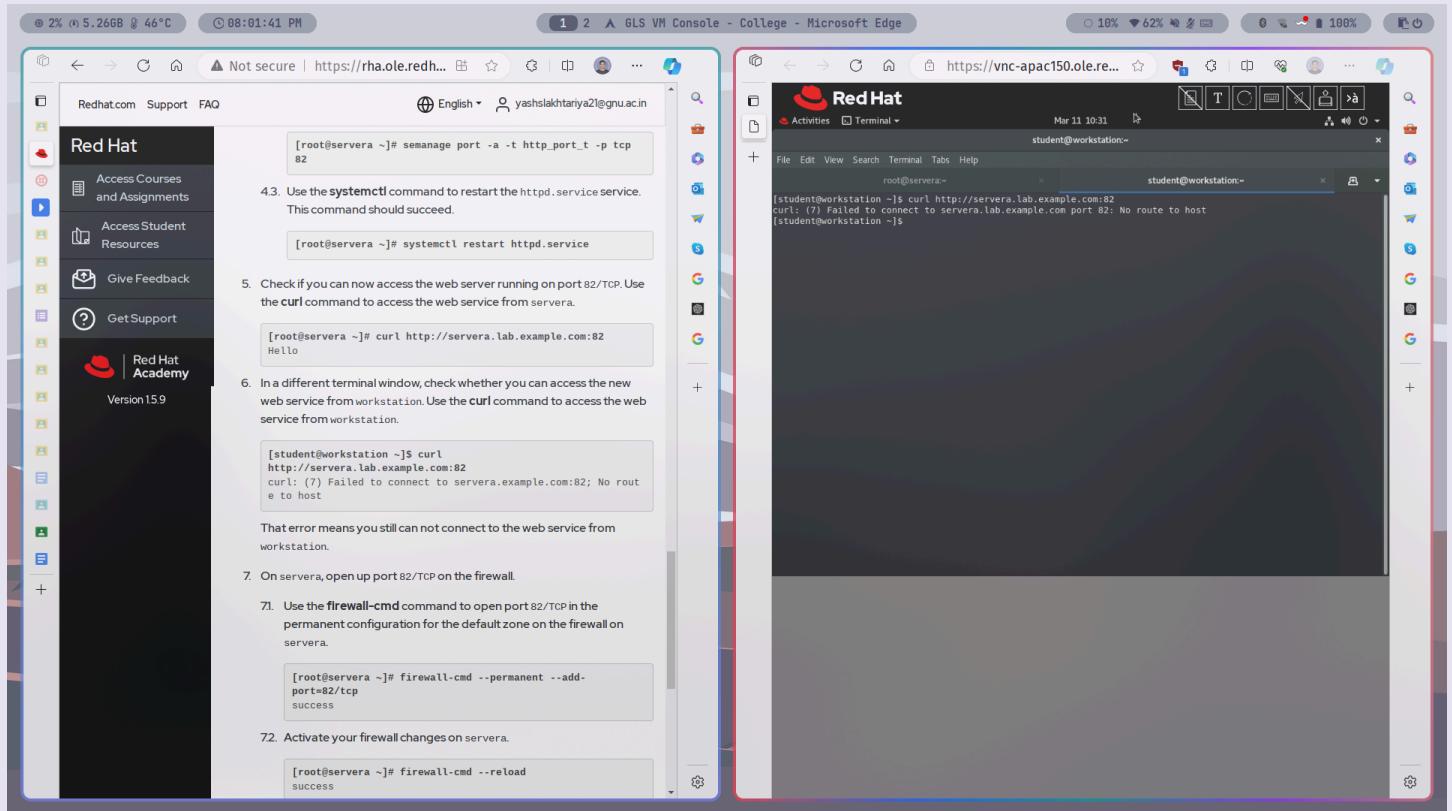
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

e. Now, use curl command to test connection with servera lab URL through port 82



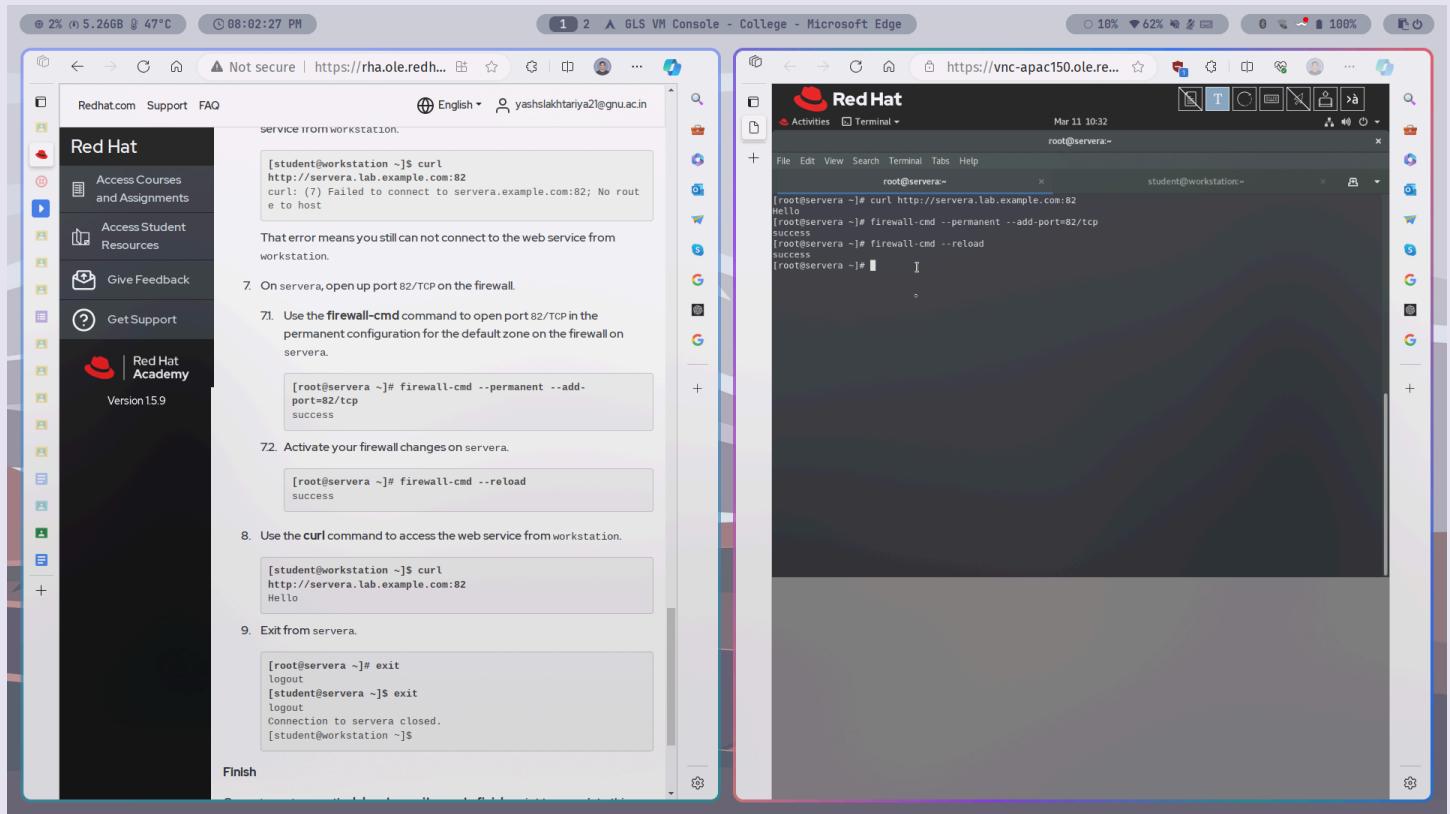
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

f. But, it would fail if tested from workstation



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

g. Through firewall daemon allow permanently tcp connections over port 82 and reload firewall daemon



Commands :

- **firewall-cmd --permanent --add-port=82/tcp** (to add port 82 and allow tcp connections permanently)
- **firewall-cmd --reload** (to reload the firewall daemon)

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

h. Now, from workstation also curl command will work for testing connection over tcp port 82

The screenshot shows a Microsoft Edge browser window with two tabs open. The left tab displays a Red Hat Academy exercise titled "netsecurity-ports". The exercise includes several steps:

- Step 7.2: [root@servera ~]# firewall-cmd --permanent --add-port=82/tcp
success
- Step 8: Activate your firewall changes on servera.
- Step 8. Use the curl command to access the web service from workstation.
[student@workstation ~]\$ curl http://servera.lab.example.com:82
Hello
- Step 9: Exit from servera.
[root@servera ~]# exit
logout
[student@servera ~]\$ exit
logout
Connection to servera closed.
[student@workstation ~]\$

Below these steps, there is a "Finish" section with instructions to run the "lab netsecurity-ports finish" script on the workstation. The right tab shows a terminal window on a Red Hat workstation with the following session:

```
[root@servera-] student@workstation-  
[student@workstation ~]$ curl http://servera.lab.example.com:82  
curl: (7) Failed to connect to servera.lab.example.com port 82: No route to host  
[student@workstation ~]$ [student@workstation ~]$ curl http://servera.lab.example.com:80  
Hello  
[student@workstation ~]$
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

i. Finish the lab and exit from ssh

The image shows a Microsoft Edge browser window with two tabs open. The left tab is a Red Hat Academy exercise titled "servera". It contains several terminal command boxes:

- [root@servera ~]# firewall-cmd --permanent --add-port=82/tcp
success
- 72. Activate your firewall changes on servera.
[root@servera ~]# firewall-cmd --reload
success
- 8. Use the curl command to access the web service from workstation.
[student@workstation ~]\$ curl http://servera.lab.example.com:82
Hello
- 9. Exit from servera.
[root@servera ~]# exit
logout
[student@servera ~]\$ exit
logout
Connection to servera closed.
[student@workstation ~]\$

Below these commands is a "Finish" section with instructions to run "lab netsecurity-ports finish" on the workstation.

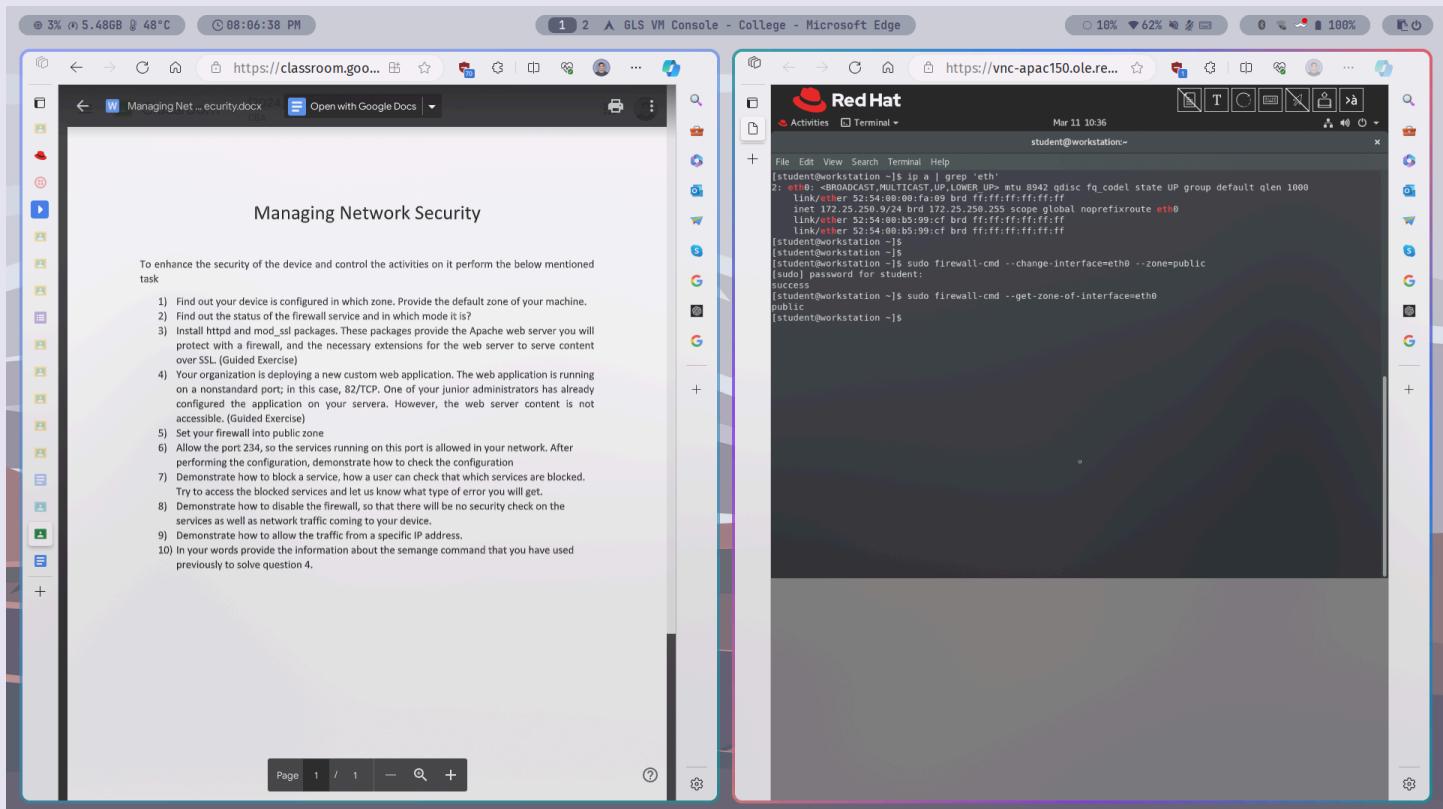
The right tab is a Red Hat terminal session titled "student@workstation". It shows the output of the "lab netsecurity-ports finish" command:

```
File Edit View Search Terminal Help
[student@workstation ~]$ lab netsecurity-ports finish
Cleaning up the lab on servera:
: Cleaning up firewall rules: ..... SUCCESS
: Removing the /etc/httpd: ..... SUCCESS
: Removing the /var/www/html/index.html: ..... SUCCESS
: Removing Firewall roles: ..... SUCCESS
: Removing port 82 from PORT_TYPE: ..... SUCCESS

Lab finished.
[student@workstation ~]$
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

5) Set your firewall into public zone

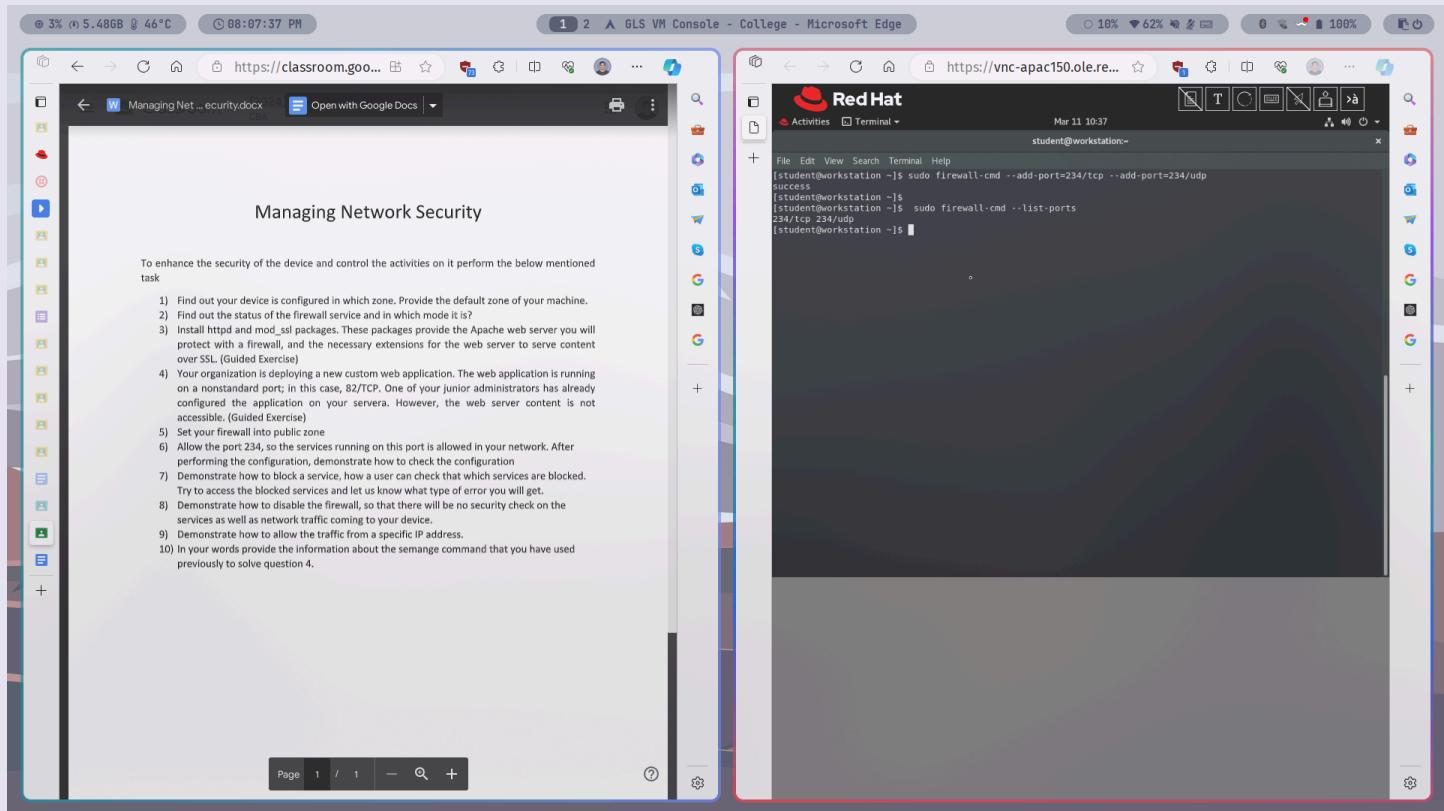


Commands :

- **ip a | grep 'eth'** (to get lines containing eth in output of ip a, which is similar to ifconfig, for showing interface configurations)
- **sudo firewall-cmd --change-interface=eth0 --zone=public** (to change firewall on eth0 interface to public zone)
- **sudo firewall-cmd --get-zone-of-interface=eth0** (to check the zone of specified (eth0) interface)

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

6) Allow the port 234, so the services running on this port is allowed in your network. After performing the configuration, demonstrate how to check the configuration



Commands :

- **sudo firewall-cmd --add-port=234/tcp --add-port=234/udp** (to add port 234 for tcp and udp connections)
- **sudo firewall-cmd --list-ports** (to check list of added ports)

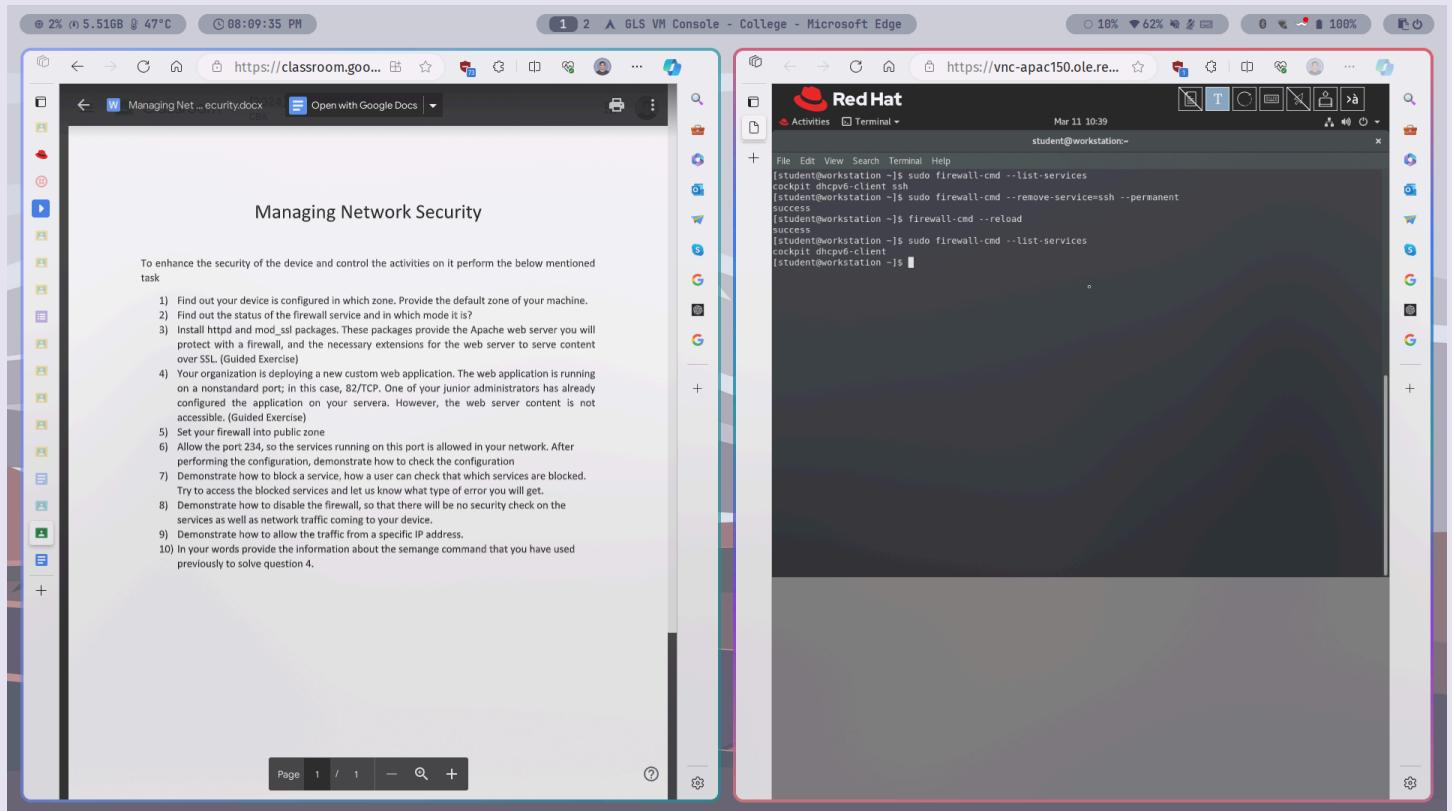
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 9

7) Demonstrate how to block a service, how a user can check that which services are blocked. Try to access the blocked services and let us know what type of error you will get.

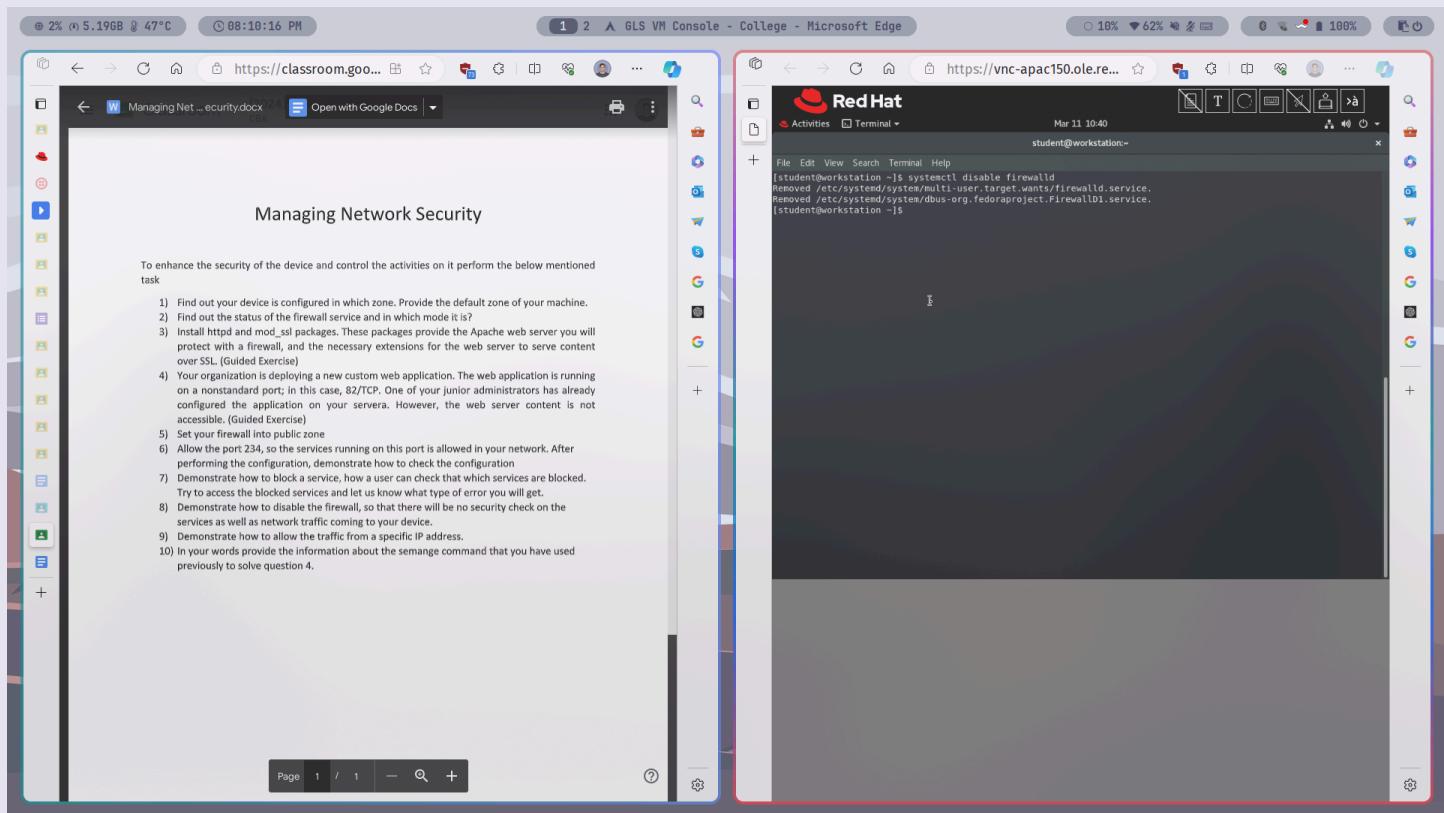


Commands :

- **sudo firewall-cmd --list-services** (to list firewall services)
- **sudo firewall-cmd --remove-service=ssh --permanent** (to remove ssh service permanently)
- **sudo firewall-cmd --reload** (to reload firewall daemon)

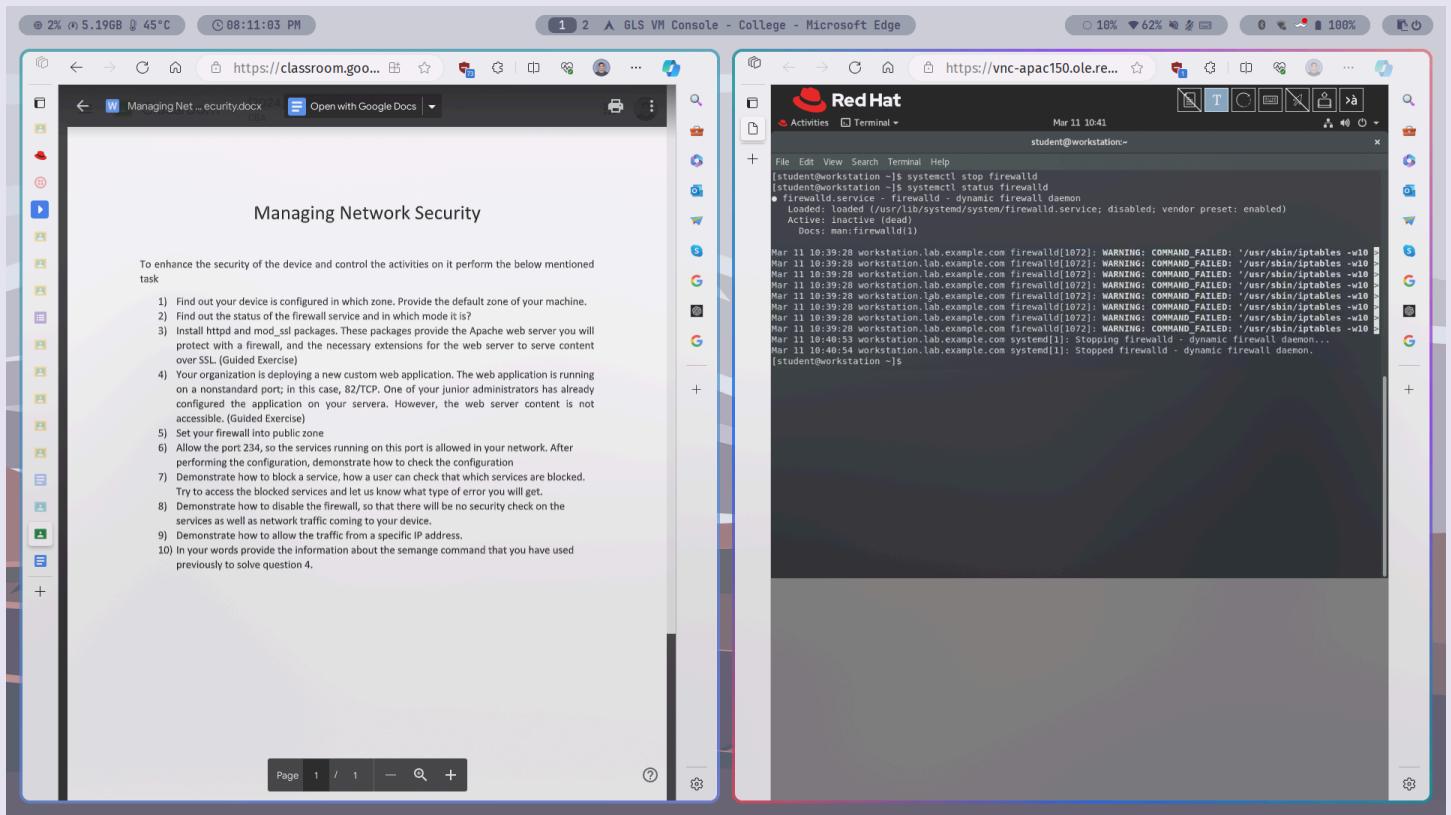
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

8) Demonstrate how to disable the firewall, so that there will be no security check on the services as well as network traffic coming to your device.



Command : **systemctl disable firewalld**

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

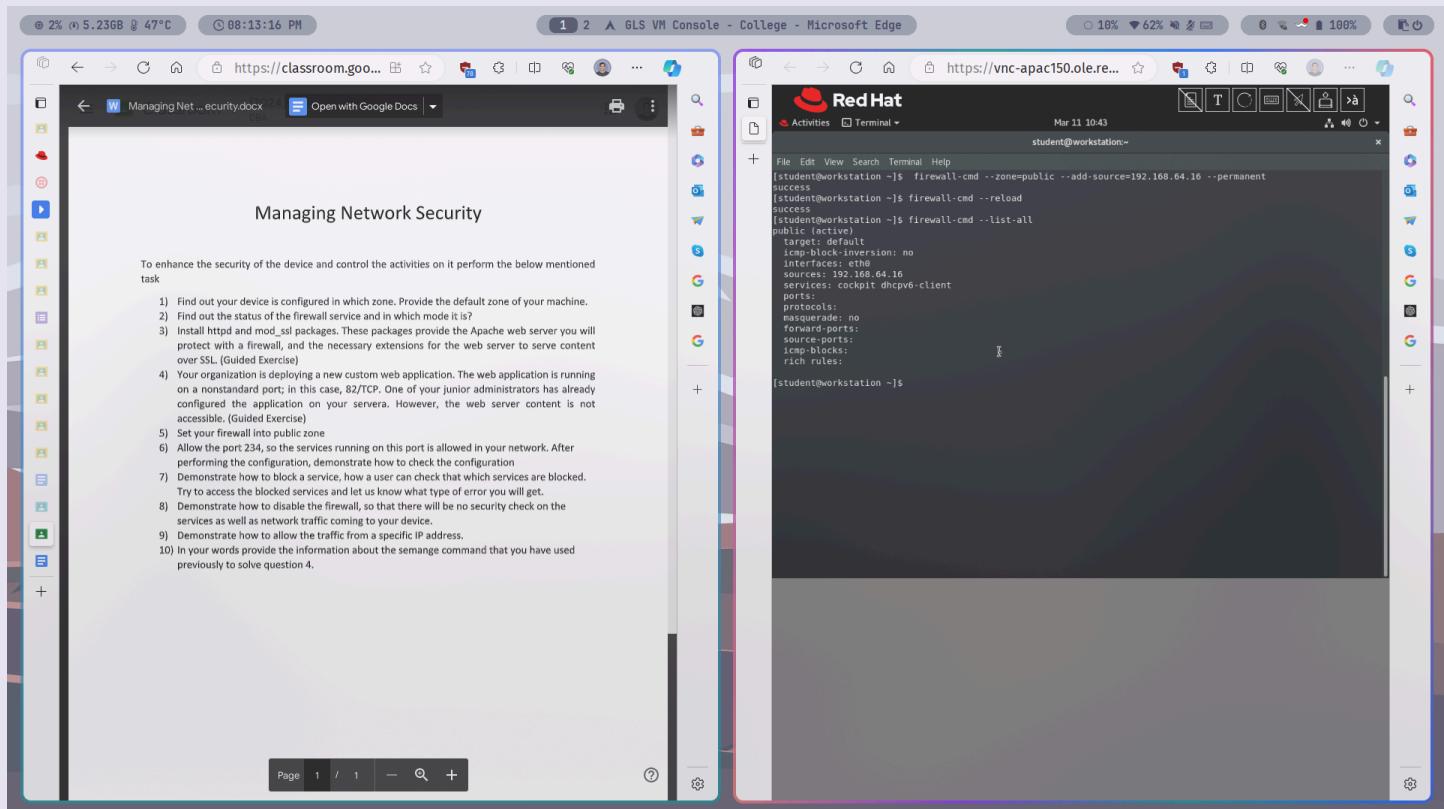


Commands :

- **systemctl stop firewalld**
- **systemctl status firewalld**

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 9

9) Demonstrate how to allow the traffic from a specific IP address.



Commands :

- **firewall-cmd --zone=public --add-source=192.168.64.16 --permanent** (to add source of IP specified and allow it permanently in public zone)
- **firewall-cmd --reload** (to reload the firewall dameon)
- **firewall-cmd --list-all** (to list all details of firewall)

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 9

10) In your words provide the information about the semange command that you have used previously to solve question 4.

→ `semanage` is a handy tool for managing security policies in SELinux environments. It lets you tweak settings like file permissions and port labeling easily. With it, you can add, delete, or change rules to keep your system secure. `semanage` simplifies SELinux configuration, making it accessible without diving into complex policy files. It's a key tool for maintaining system integrity in SELinux setups.

→ Here, specifically, it is used for port policies