ITIM

PRACTICAL - 9

Name : Yagna Patel
Enrollment No. : 211621020
Batch : 61(CBA)

Tasks :

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.

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

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)

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)

5) Set your firewall into public zone

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

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.

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.

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

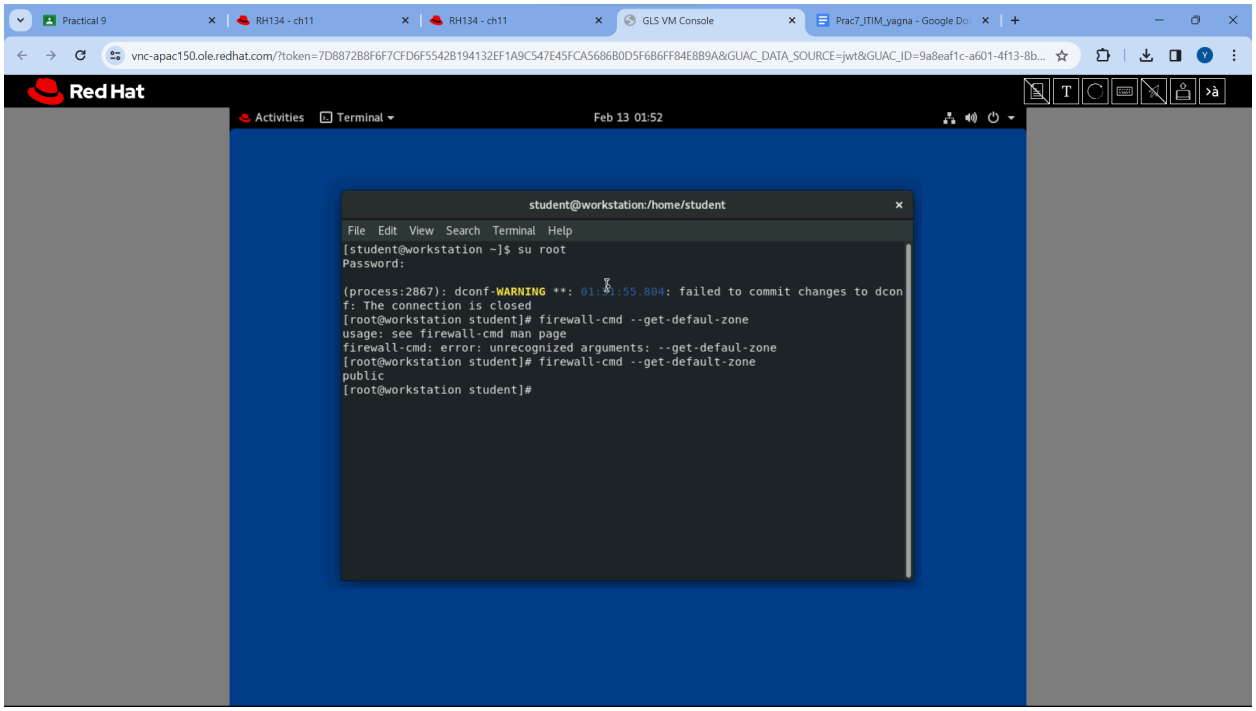
10) In your words provide the information about the semange command that you have used

previously to solve question 4.

Steps :

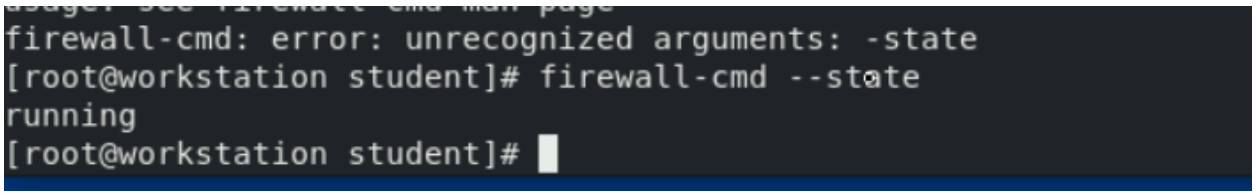
1)Find out your device is configured in which zone. Provide the default zone of your machine.

We can see it with command : firewall-cmd –get-default-zone



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

We can see it with command : firewall-cmd –state



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)

Login into servera

```
[student@workstation ~]$ lab netsecurity-firewalls-start
Error: unknown problem specified and not found, netsecurity-firewalls-start
[student@workstation ~]$ lab netsecurity-firewalls start

Starting lab.

Preparing servera for lab exercise work:

 · Check servera connectivity..................................  SUCCESS
 · Enable and start cockpit.socket on servera..................  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 08:19:05 2020 from 172.25.250.9
```

Install httpd and mod_ssl using sudo yum install httpd mod_ssl

```
[student@servera ~]$ sudo yum install httpd mod_ssl
[sudo] password for student:
Last metadata expiration check: 1 day, 9:22:21 ago on Tue 20 Feb 2024 01:27:50 AM EST.
Dependencies resolved.
================================================================================
 Package            Arch   Version     Repository                          Size
================================================================================
Installing:
 httpd              x86_64 2.4.37-21.module+el8.2.0+5008+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+5008+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+5008+cca404a3
                                        rhel-8.2-for-x86_64-appstream-rpms  36 k
 httpd-tools        x86_64 2.4.37-21.module+el8.2.0+5008+cca404a3
                                        rhel-8.2-for-x86_64-appstream-rpms 103 k
 mod_http2          x86_64 1.11.3-3.module+el8.2.0+4377+dc421495
                                        rhel-8.2-for-x86_64-appstream-rpms 158 k
 redhat-logos-httpd noarch 81.1-1.el8  rhel-8.2-for-x86_64-baseos-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
Enabling module streams:
 httpd                      2.4

Transaction Summary
================================================================================
```

create the **/var/www/html/index.html file**. Add one line of text that reads: I am

Servera. Start and enable the httpd service on your servera system

```
[student@servera ~]$ sudo bash -c \ "echo 'I am servera yagna.' > /var/www/html/index.html"
[student@servera ~]$ sudo systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd
/system/httpd.service.
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

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

On servera, make sure that the nftables service is masked and the firewalld

service is enabled and running. If it is not running make it run.

```
Last login: Wed Feb 21 10:49:33 2024 from 172.25.250.9
[student@servera ~]$ sudo systemctl status nftables
[sudo] password for student:
Sorry, try again.
[sudo] password for student:
● nftables.service - Netfilter Tables
   Loaded: loaded (/usr/lib/systemd/system/nftables.service; disabled; vendor preset: disab>
   Active: inactive (dead)
     Docs: man:nft(8)

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

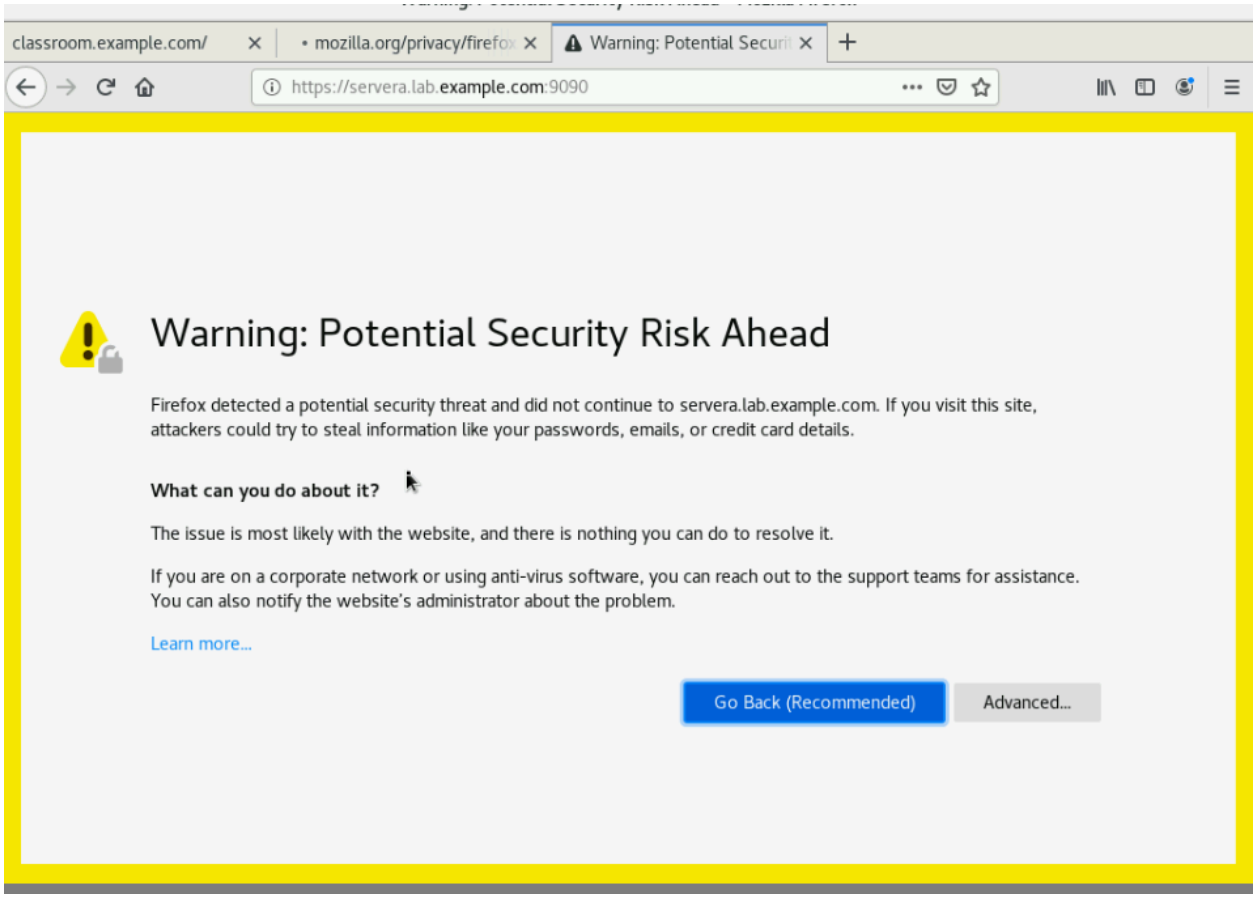
Verify that the status of the nftables service is masked. Using systemctl status nftables

Verify that the status of the firewalld service is enabled and running using systemctl status firewalld

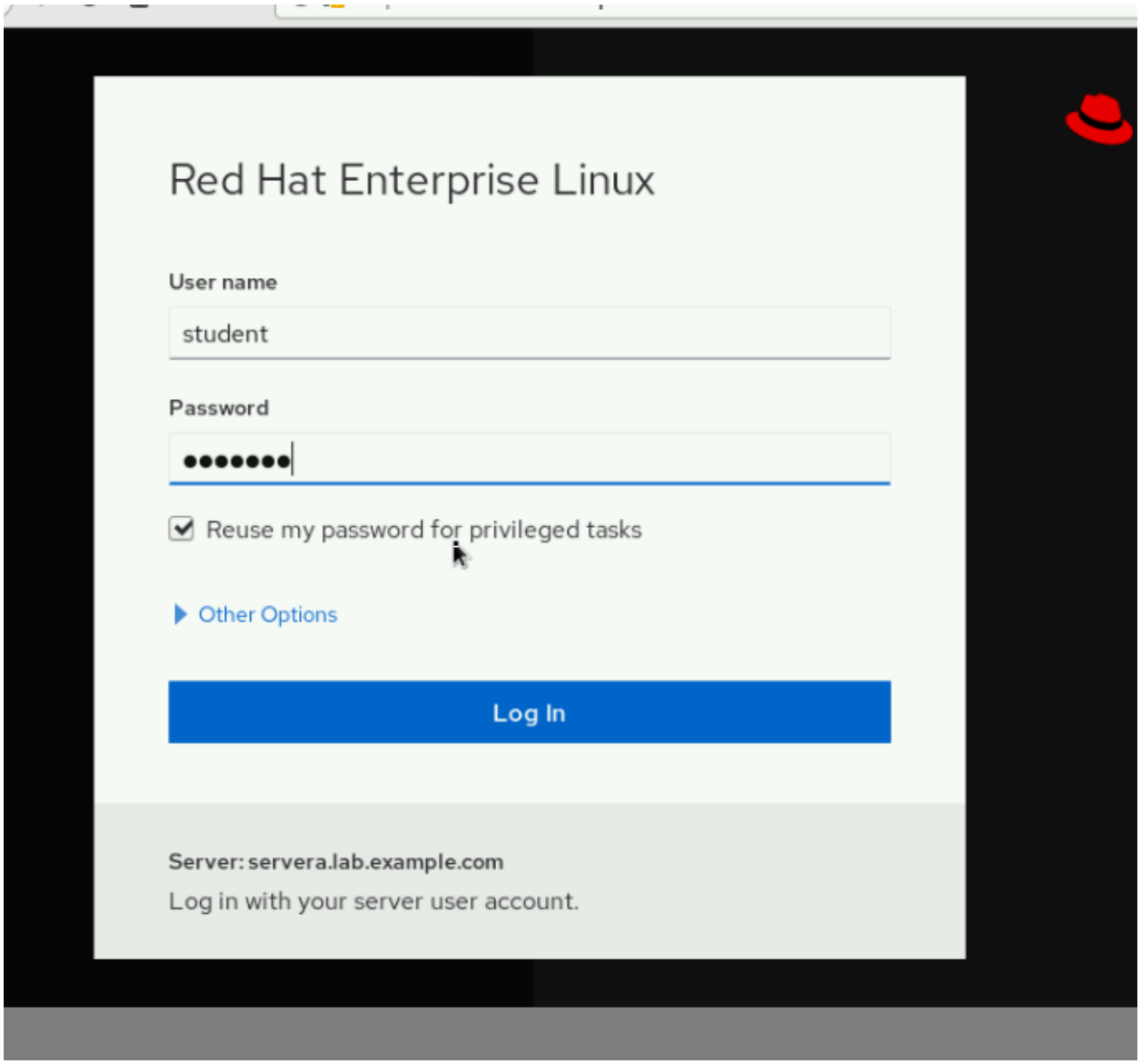
```
[student@servera ~]$ sudo systemctl status nftables
● nftables.service
   Loaded: masked (Reason: Unit nftables.service is masked.)
   Active: inactive (dead)
[student@servera ~]$
```

```
[student@servera ~]$ sudo systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabl>
   Active: active (running) since Wed 2024-02-21 10:44:17 EST; 14min ago
     Docs: man:firewalld(1)
 Main PID: 867 (firewalld)
    Tasks: 2 (limit: 11345)
   Memory: 31.3M
   CGroup: /system.slice/firewalld.service
           └─867 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid

Feb 21 10:44:16 servera.lab.example.com systemd[1]: Starting firewalld - dynamic firewall d>
Feb 21 10:44:17 servera.lab.example.com systemd[1]: Started firewalld - dynamic firewall da>
Feb 21 10:44:17 servera.lab.example.com firewalld[867]: WARNING: AllowZoneDrifting is enabl>

[student@servera ~]$
```

Open Firefox and browse to https://servera.lab.example.com:9090 to access the Web Console. Accept the self-signed certificate used by servera by adding an exception.

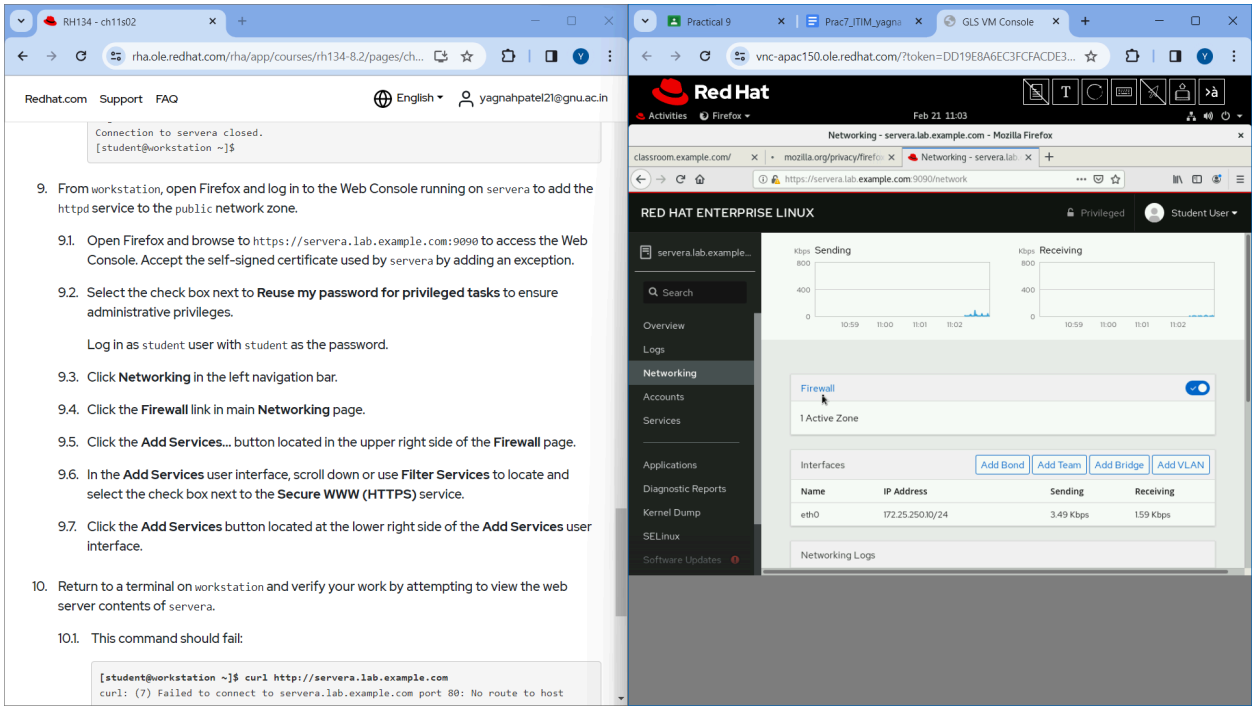


Log in as student user with student as the password.



->Click Networking in the left navigation bar.

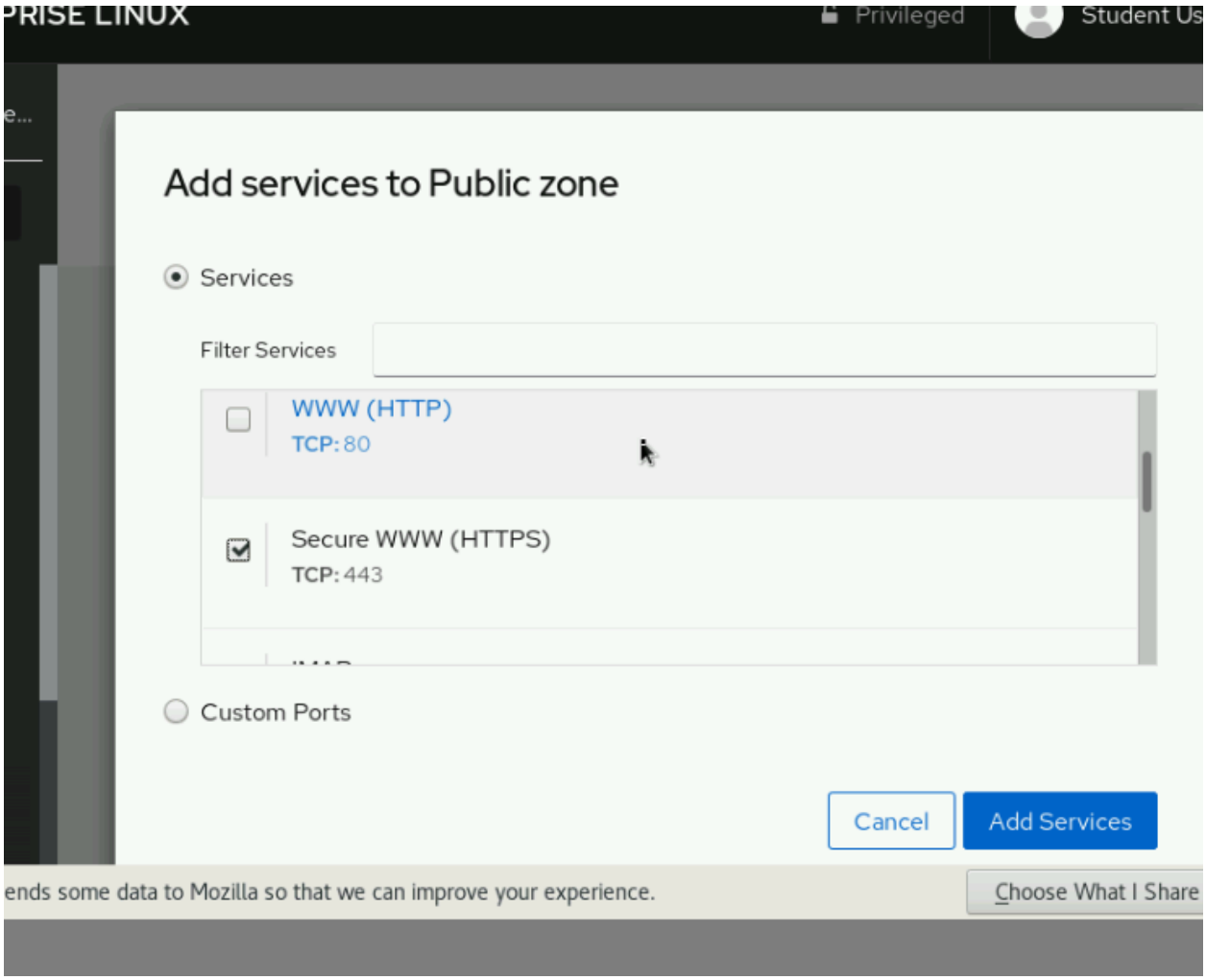
->Click the Firewall link in main Networking page.

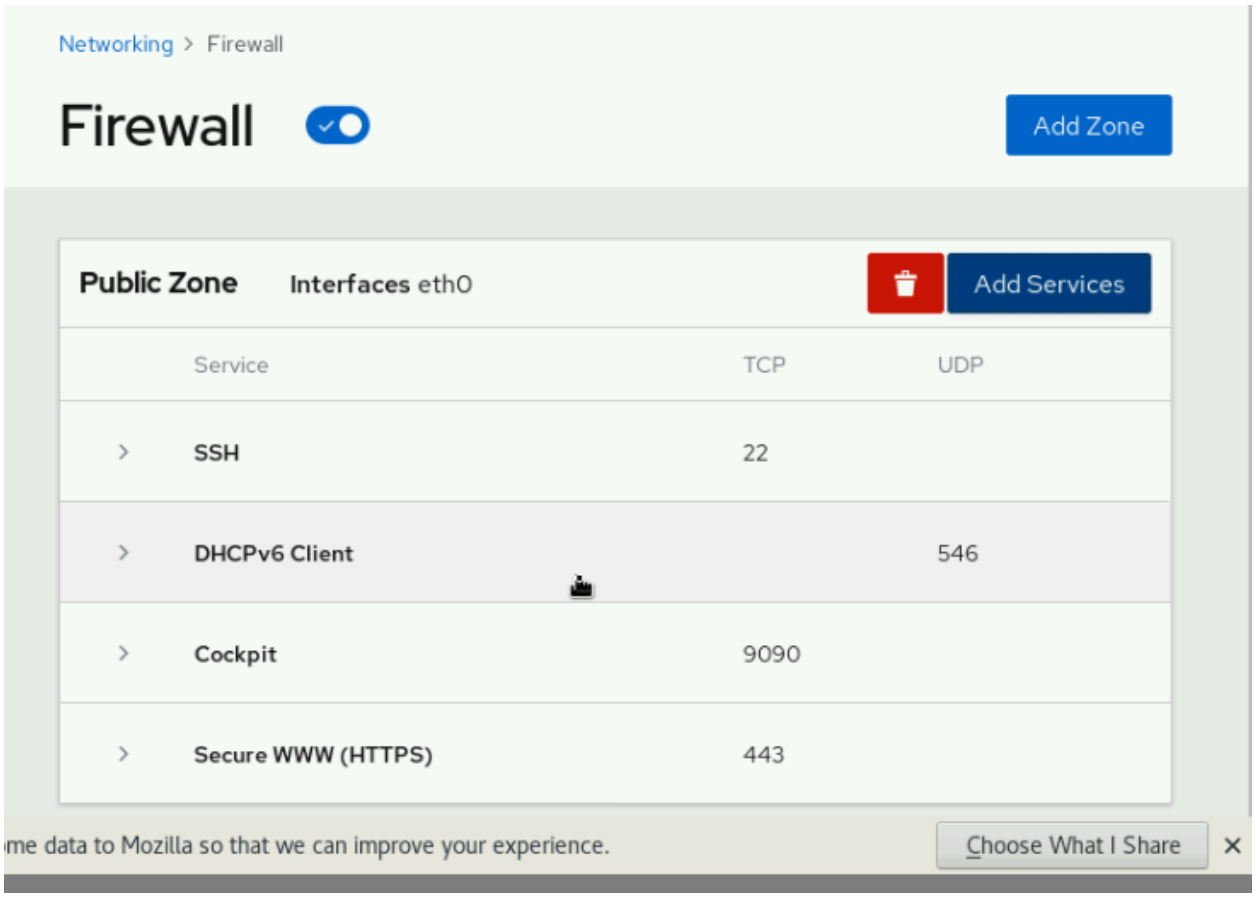


->Click the Add Services... button located in the upper right side of the Firewall page.

->In the Add Services user interface, scroll down or use Filter Services to locate and select the check box next to the Secure WWW (HTTPS) service.

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


PRISE LINUX
Privileged
Student Us
Add services to Public zone
Services
Filter Services
WWW (HTTP)
TCP: 80
Secure WWW (HTTPS)
TCP: 443
Custom Ports
Cancel
Add Services
ends some data to Mozilla so that we can improve your experience.
Choose What I Share



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)

Login in as student user at servera

```
root@servera:~
File  Edit  View  Search  Terminal  Help
[student@workstation ~]$ lab netsecurity-ports start

Starting lab.

Preparing workstation for lab exercise work:
 · Verifying that servera is running: .........................  SUCCESS
 · Ensuring package is installed: httpd........................  SUCCESS
 · Setting firewall rules: ....................................  SUCCESS
 · Set HTTP listening port: ...................................  SUCCESS
 · Setting the index.html file: ...............................  SUCCESS

[student@workstation ~]$ ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Wed Feb 21 10:55:28 2024 from 172.25.250.9
[student@servera ~]$ sudo -i
[sudo] password for student:
[root@servera ~]#
```

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.

Attempt to fix the web content problem by restarting the httpd service.

```
[root@servera ~]# systemctl restart httpd.service
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xe" for details.
[root@servera ~]# systemctl status -l httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since Wed 2024-02-21 11:09:17 EST; 20s ago
     Docs: man:httpd.service(8)
  Process: 25879 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAI>
 Main PID: 25879 (code=exited, status=1/FAILURE)
   Status: "Reading configuration..."

Feb 21 11:09:17 servera.lab.example.com systemd[1]: Starting The Apache HTTP Server...
Feb 21 11:09:17 servera.lab.example.com httpd[25879]: (13)Permission denied: AH00072: make_>
Feb 21 11:09:17 servera.lab.example.com httpd[25879]: (13)Permission denied: AH00072: make_>
Feb 21 11:09:17 servera.lab.example.com httpd[25879]: no listening sockets available, shutt>
Feb 21 11:09:17 servera.lab.example.com httpd[25879]: AH00015: Unable to open logs
Feb 21 11:09:17 servera.lab.example.com systemd[1]: httpd.service: Main process exited, cod>
Feb 21 11:09:17 servera.lab.example.com systemd[1]: httpd.service: Failed with result 'exit>
Feb 21 11:09:17 servera.lab.example.com systemd[1]: Failed to start The Apache HTTP Server.

[root@servera ~]#
```

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

```
[root@servera ~]# sudo sealert -a /var/log/audit/audit.log
100% done
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   ************************

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_managemen
t_port_t, jboss_messaging_port_t, ntop_port_t, puppet_port_t.

*****  Plugin catchall (1.49 confidence) suggests   **************************

If you believe that httpd should be allowed name_bind access on the port 82 tcp_socket by de
fault.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp


Additional Information:
Source Context                system_u:system_r:httpd_t:s0
Target Context                system_u:object_r:reserved_port_t:s0
Target Objects                port 82 [ tcp_socket ]
Source                        httpd
```

Use the semanage command to find an appropriate port type for port `82/TCP`.

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

Use the systemctl command to restart the `httpd.service` service. This command should succeed.

```
[root@servera ~]#
[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 ~]# semanage port -a -t http_port_t -p tcp 82
[root@servera ~]# systemctl restart httpd.service
[root@servera ~]#
```

Check if you can now access the web server running on port `82/TCP`. Use the curl command to access the web service from `servera`.

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

Use the firewall-cmd command to open port `82/TCP` in the permanent configuration for the default zone on the firewall on `servera`.Activate your firewall changes on `servera`.

```
[root@servera ~]# firewall-cmd --permanent --add-port=82/tcp
success
[root@servera ~]# firewall-cmd --reload
success
[root@servera ~]#
```

Use the curl command to access the web service from `workstation`.

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

5) Set your firewall into public zone using firewall-cmd –set-default-zone=public

```
File  Edit  View  Search  Terminal  Help
[root@workstation student]# firewall-cmd --set-default-zone=public
Warning: ZONE_ALREADY_SET: public
success
[root@workstation student]#
```

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.

To allow port 234 use command :

sudo firewall-cmd --zone=public --add-port=234/tcp --permanent

Than reload the firewall to apply a new configuration persistently

To check configuration command is firewall-cmd –list-all

```
[root@workstation student]# sudo firewall-cmd --zone=public --add-port=234/tcp --permanent
success
[root@workstation student]# sudo firewall-cmd --reload
success
[root@workstation student]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 234/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@workstation student]#
```

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.

For block service command is : sudo firewall-cmd --remove-service=cockpit --permanent

So i blocked cockpit service and make that configuration persistent with command : firewall-cmd --reload

Now if i'll try to access cockpit service , so i'll send http request via curl

```
[root@workstation student]# sudo firewall-cmd --remove-service=cockpit --permanent
success
[root@workstation student]# firewall-cmd --reload
success
[root@workstation student]# curl https://172.25.250.9:9009
curl: (7) Failed to connect to 172.25.250.9 port 9009: Connection refused
[root@workstation student]#
```

Question 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.

To disable firewall we have command : systemctl disable firewalld

To enable firewall again : systemctl enable firewalld

```
[root@workstation student]# systemctl disable firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@workstation student]# systemctl enable firewalld
Created symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service → /usr/lib/sys
temd/system/firewalld.service.
Created symlink /etc/systemd/system/multi-user.target.wants/firewalld.service → /usr/lib/sys
temd/system/firewalld.service.
[root@workstation student]#
```

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

Let's assume you want to allow traffic from the IP address 192.168.1.100

We use the command:

firewall-cmd --zone=public --add-source=192.168.1.100 --permanent

And we reload it

```
[root@workstation student]# sudo firewall-cmd --zone=public --add-source=192.168.1.100 --per
manent
success
[root@workstation student]# firewall-cmd --reload
success
[root@workstation student]#
```

```
[root@workstation student]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources: 192.168.1.100
  services: dhcpv6-client ssh
  ports: 234/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@workstation student]#
```

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

SELinux is a security mechanism that provides mandatory access controls to enhance the security of a Linux system. It restricts the actions that users, processes, and services can perform, based on defined security policies.

In this case, it's used because a custom web application running on a nonstandard port (port 82/TCP) isn't accessible.

To fix this, we tell SELinux (a security feature) that it's okay to allow access to port 82. We used this command:

sudo semanage port -a -t http_port_t -p tcp 82 // in this command

semanage port: This is the main semanage command used to manage port definitions.

-a: This option specifies that we are adding a new port definition.

-t http_port_t: This option specifies the SELinux type for the port. In this case, we're using http_port_t, which is the SELinux type associated with HTTP ports.

-p tcp: This option specifies the protocol for the port. In this case, we're specifying TCP.

82: This is the port number that we want to allow access to.