**Lesson 4**

**Network and Service Access Controls (firewalld and Fail2ban)**

**Contents**

[1. Overview of the netfilter, iptables and the firewalld 2](#_Toc85140663)

[2. Adding and removing a service in firewalld for packet filtering 4](#_Toc85140664)

[3. Adding Rich Rules 6](#_Toc85140665)

[4. Logging Rules 11](#_Toc85140666)

[5. Limiting frequency of packets to discourage port scanning 14](#_Toc85140667)

[6. Reverse Proxy via Network Address Translation (NAT) 18](#_Toc85140668)

[7. Controlling outgoing traffic by using the direct interface 24](#_Toc85140669)

[8. Protecting from brute force attack to sshd 27](#_Toc85140670)

1. Overview of the netfilter, iptables and the firewalld

netfilter is the packet filtering framework inside the Linux 2.4.x and later kernel series. This framework enables packet filtering, network address [and port] translation (NA[P]T) and other packet mangling.

iptables is a widely used firewall tool that interfaces with the kernel’s netfilter packet filtering framework. The successor of iptables, nftables, has been released since 2014 too.

firewalld is like the iptables service which talks to the netfilter framework in the kernel through the same interface, using the iptables command. firewalld is installed and enabled in Oracle Linux 8 by default. There are two main differences between firewalld and iptables.

1. firewalld can change the firewall rules/settings during normal system operation without existing connections being lost.

2. firewalld provides both command line and GUI interfaces to the system administrators.

Ref:<https://netfilter.org/>

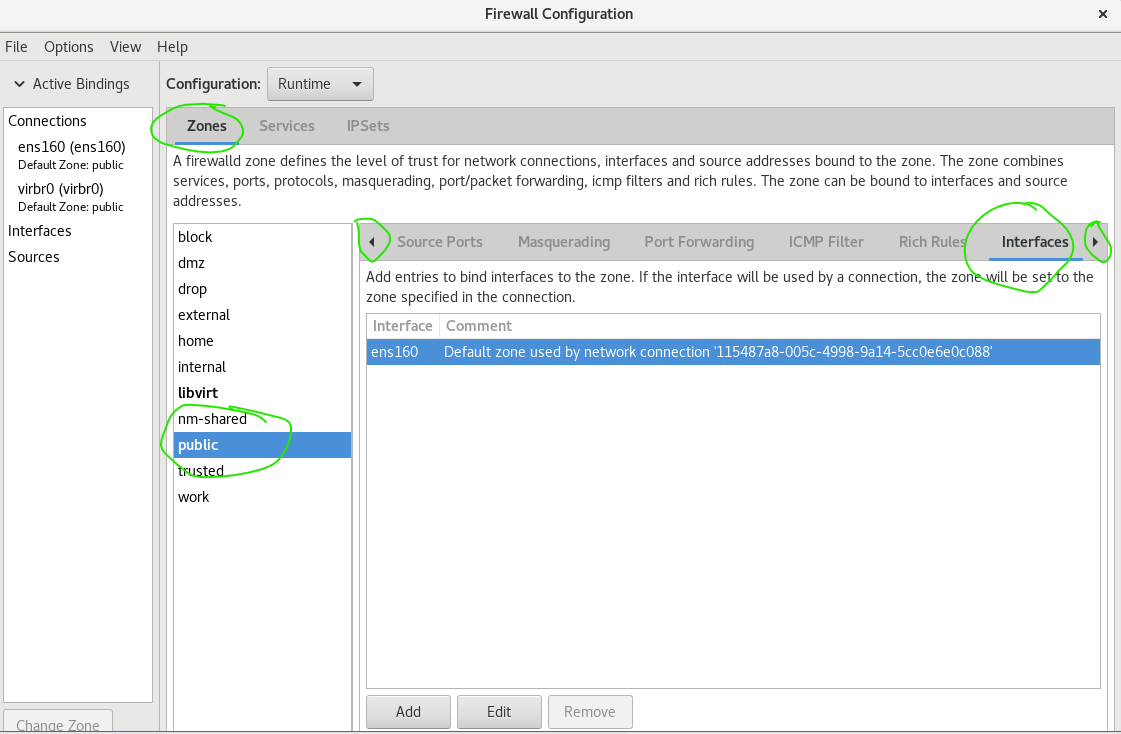
**Exploring the zones and pre-defined services of firewalld**

On any system:

1. Start the Firewalld GUI by going to Applications, Sundry, Firewall, or by running “firewall-config”.

firewall-config

1. Click on the Zones tab to view the zones. Click on the Interfaces tab to view the network interfaces that are currently in each zone. (see following diagram)



The default zone is highlighted in blue

1. You can also view the list of the zones and the details of them using one of the following command lines.

firewall-cmd --get-zones

firewall-cmd --info-zone=<zone name>

firewall-cmd --list-all-zones

1. View the default zone and its details

firewall-cmd --get–default-zone

firewall-cmd --info-zone=`firewall-cmd --get-default-zone`

Text

Description automatically generated

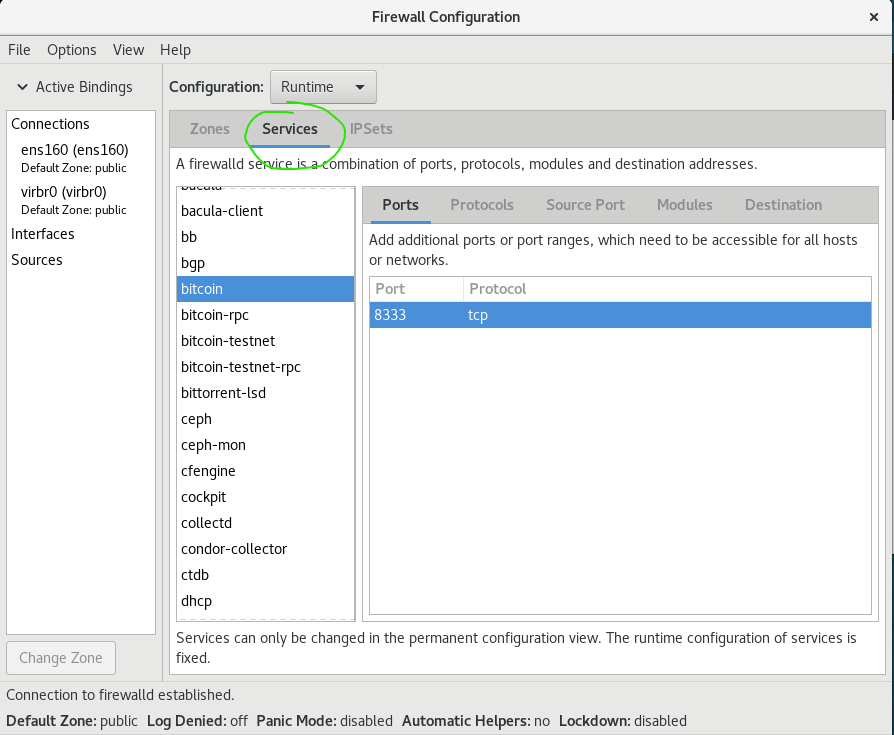
The above output reports the default zone name, included interface, and the current services that are allowed to pass through the firewall.

Take note: the `firewall-cmd --get-default-zone` is enclosed in a pair of back quote(`).

This is a simple shortcut to extract the output of a Linux command and use it as a command line argument for another Linux command.

1. In the Firewall GUI, click on the Services tab (the one beside the Zones tab) (see following diagram)

Scroll through the services listed to see which port and protocol have been specified for each of the specified services.



Your Task: Find out the ports/protocols for https, ftp , ipsec and dns.

e.g. http: 80/tcp

https: \_443/tcp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ftp: \_\_\_\_21/tcp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ipsec: \_\_\_4500/tcp , 500/udp , 4500/udp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dns: \_\_\_\_53/tcp , 53/udp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Click back on the Zones tab.
2. Adding and removing a service in firewalld for packet filtering

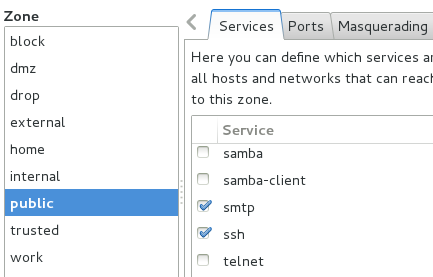
Firewall configuration is actually the management the filtering rules. By default, all the ports are filtered (i.e. not allow to pass through the firewall) for the interfaces that belong to public zone. One simple way to allow certain ports (with their corresponding protocols) to be opened is to add in the exception rules by the network services names.

On any system:

1. In the Firewall GUI, check that for Configuration, “Runtime” is selected. (see following diagram)



1. While in the public zone, click on the Services tab to view the available services. Check any one of the services that have not been enabled yet. (see following diagram)



Check any one of the services that have not been enabled yet

1. The change is applied immediately. List the services that are currently allowed through the firewall.

firewall-cmd –-list-services

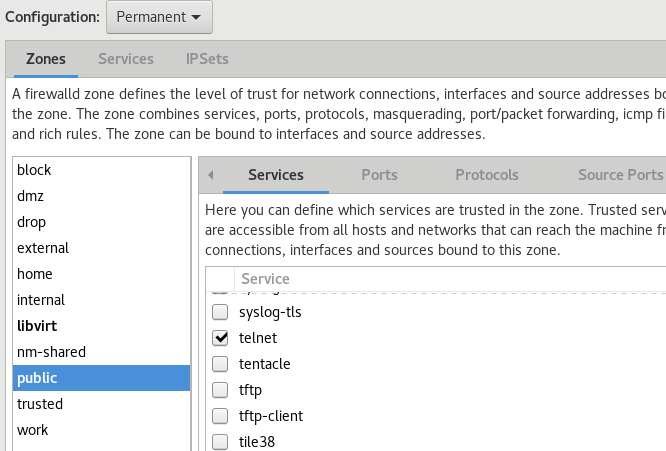
Text

Description automatically generated with medium confidence

1. In the Firewall GUI, change the Configuration to “Permanent”. (see following diagram)



1. While in the public zone, click on the Services tab to view the available services. Check any one of the services that have not been enabled yet. (see following diagram)

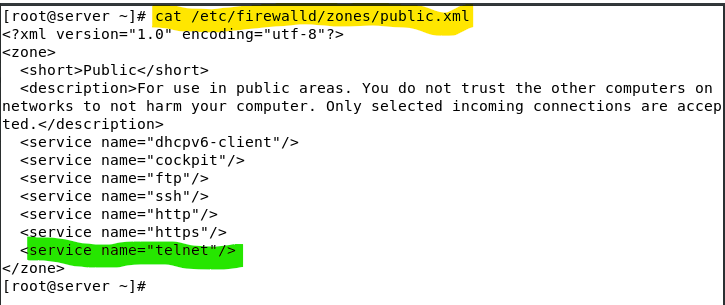


Check any one of the services that have not been enabled yet

1. As the Configuration is set to "Permanent", the change is not applied immediately, but is saved to a file, to be applied the next time the firewall is started.

The change is saved to the public zone file. View this file to verify that the service you just added is listed.

cat /etc/firewalld/zones/public.xml



1. Reload the firewall, through the GUI (Options menu->Reload Firewall).
2. Remove the service that you just added permanently, using the command line

firewall-cmd --permanent --zone=public --remove-service=telnet

1. Reload the firewall through the command line

firewall-cmd --reload

Your task: Based on the firewall access control, try to control the block and allow configuration at your **server** then use your client to test and verify the operations.

* Initial tests: verify you can access to the ftp and http services run on the server from the client.
* At the server side, block any external clients to access to the ftp service using firewall-cmd only.
* At the client side, verify that you cannot access to the ftp that runs on the server.
* At the server side, block any external clients to access to the http service using firewall-cmd only.
* At the server side, verify that you can still access to the http service that runs on the server itself.
* At the client side, verify that you cannot access to the http that runs on the server.
* At the server side, revert the firewall settings to the initial state.
* At the client side, verify you can again access to the ftp and http services run on the server.
* That's all.

1. Adding Rich Rules

Configure the firewall using rich rules allows **more specific** control beyond the ports and protocols.

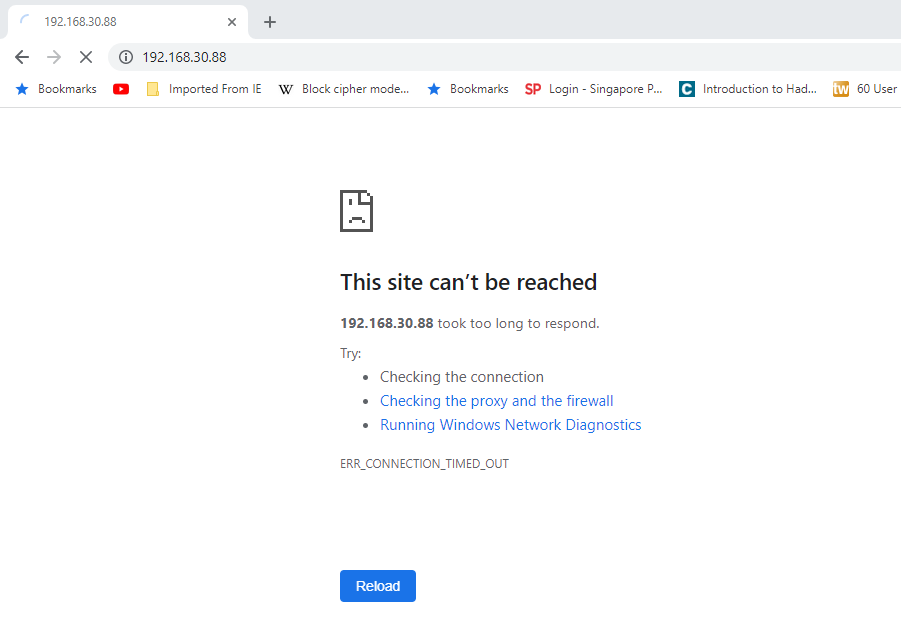
On server:

1. Check that the Apache Web Server is running and that you can browse the default web pages from the client system using http or https.
2. Redo the same test from your base PC (or your own notebook) browser. Ensure both of your client and your base PC can browse your server web page.
3. In the Firewall GUI, uncheck the services http and https. (see the following diagram)

Graphical user interface, text, application

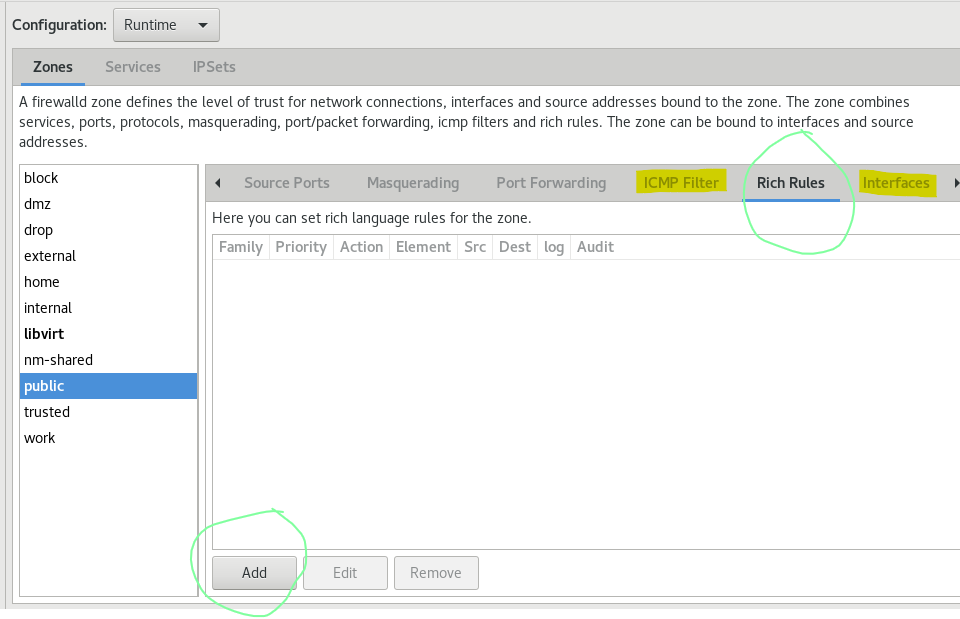
Description automatically generated

1. Now, verify that neither your client nor your base PC can browse the server web page (Try with both of the http and https protocols).

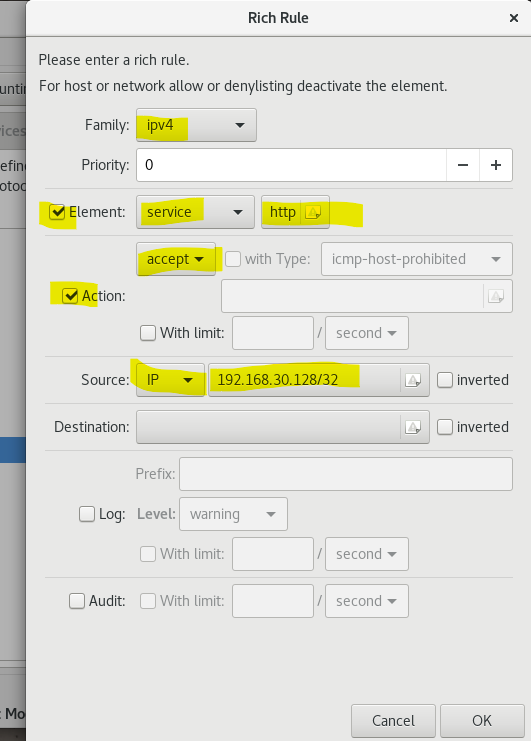


(To ensure that your browser is not displaying the cached web page. You may need to restart the browser and reload the web page to clear the cache.)

1. Use the Firewall GUI to add a rich rule (hint: look for a tab with the name 'rich rule' and press the Add button).



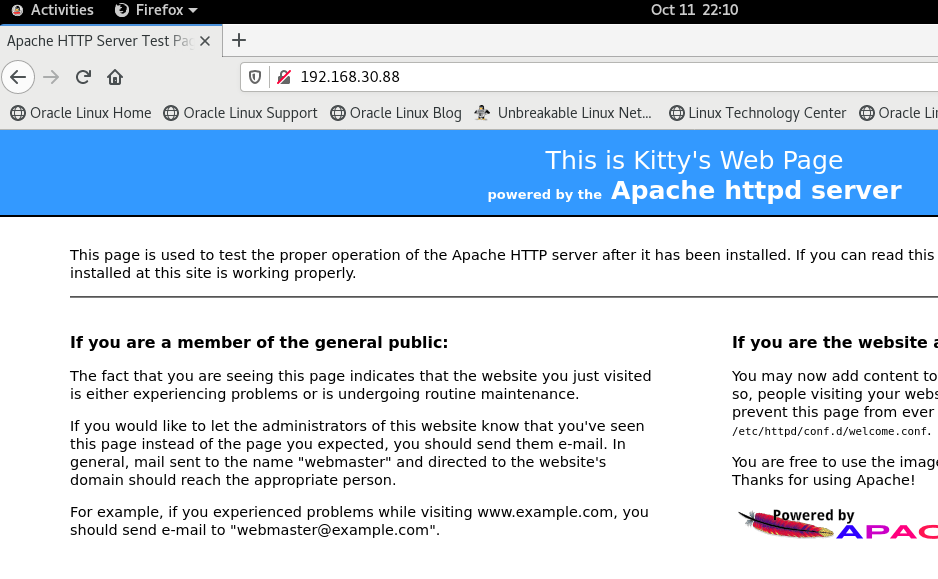
1. Define your new rich rule to accept HTTP traffic from only your client. (see following diagram for a sample of the rich rule)



Use the IP address of your client

(Take note that, you may have only added the above rich rule for runtime only.)

1. Test that your client can browse a web page from your server (using http only) but your Base PC cannot access it (Note : you may need to clear the browser cache history).



You will now try adding a rich rule using command line.

On server:

1. Add the following permanent rich rule that will allow any client in your subnet to connect to the FTP service on your server.

Type at the command line: (Note: type in all in one single command line.)

firewall-cmd --permanent --zone=public --add-rich-rule='rule family=ipv4 service name=https source address=192.168.126.129 accept'

Change to your subnet segment (e.g. 192.168.30.0/24)

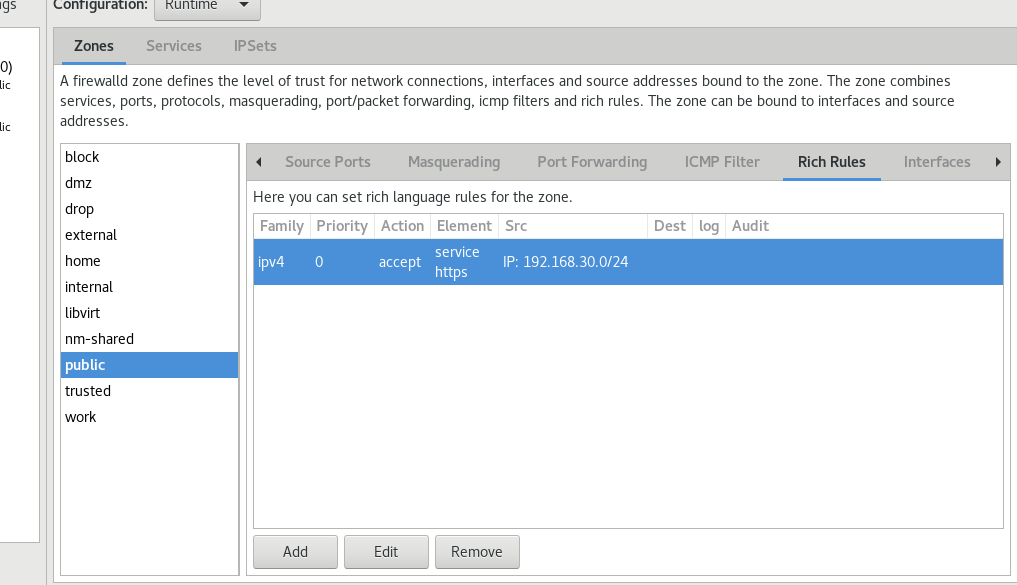
1. Reload the firewall. Type:

firewall-cmd --reload

A picture containing graphical user interface

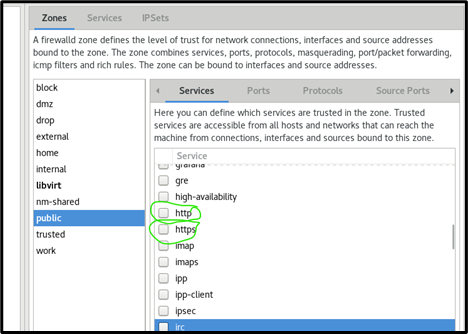
Description automatically generated

1. Using the Firewall GUI, check that the rich rule you just added is listed under the Rich Rules tab. (Note: You can refresh the rich rules tab by clicking at other tab then return to the rich rules tab.)



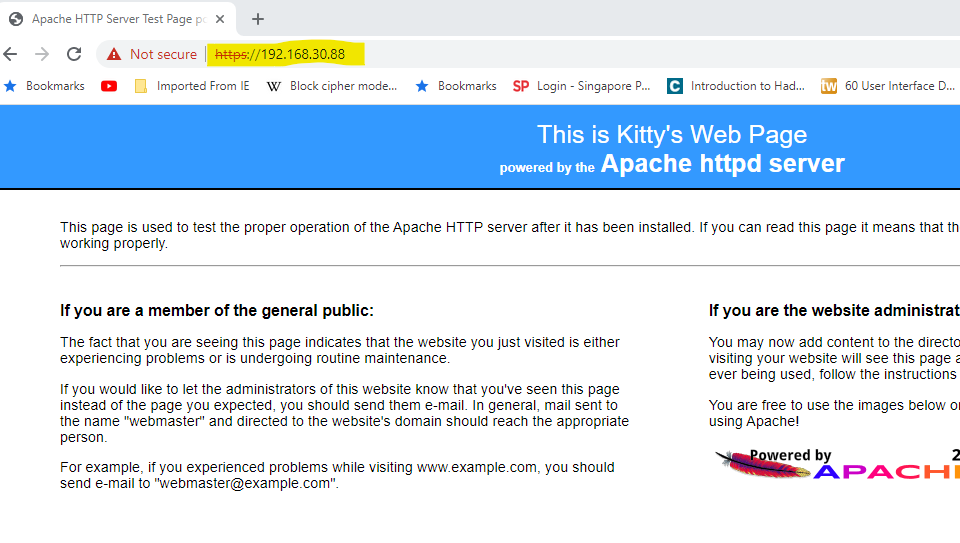
You may find the previous http related rule is gone and only the newly created https related rule is in the tab. (If this is the case, can you explain the reason ? Or you may ask your tutor for the answer.)

1. Switch to the service tab, ensure both of the http and https service entries are unchecked:



On client and On your Host PC:

1. Test that you can access your server web page with https protocol now.



1. Logging Rules

You will now add logging options to the rich rules.

Caveat: Oracle Linux is pre-installed with gnome box (a virtualization platform) and it also enable the virbr0-nic interface ( A virtual network interface) and the virbr0 device (A virtual bridge.). These two virtual devices may randomly inject annoying warning logging messages to the firewall logs. (probably due to the incomplete configuration of the gnome box). To ensure the warning messages will not confuse the following exercise. It's strongly recommended to stop and disable the virtualization service and the virbr0 related virtual devices by the following three commands at your server:

systemctl stop libvirtd

systemctl disable libvirtd

ip link set virbr0 down

A screenshot of a computer

Description automatically generated with medium confidence

( Before reboot, the virbr0 device is still shown in the output of nmcli device command.)

**You also need to reboot the server VM.**

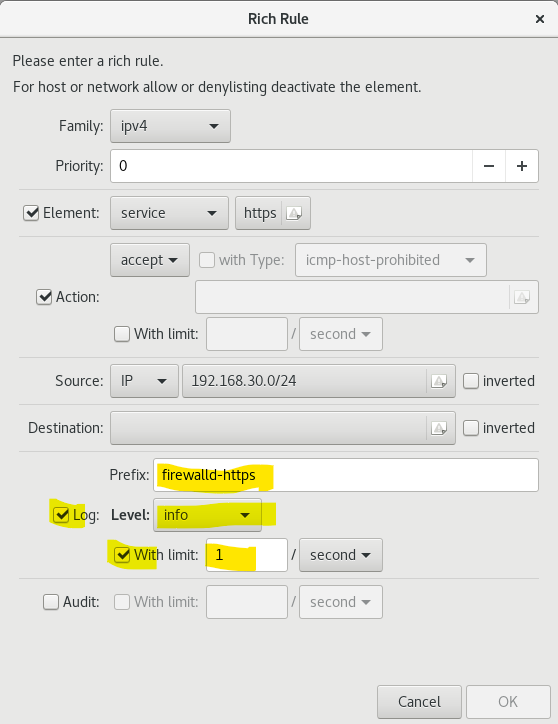
On server:

1. Verify you only have the loopback (lo) and ens160 devices :

nmcli device



1. In the Firewall GUI, under Rich Rules, **edit** the rich rule that you added for the https service.
2. Enable Log, and for the Level, choose **info**. (see following diagram)

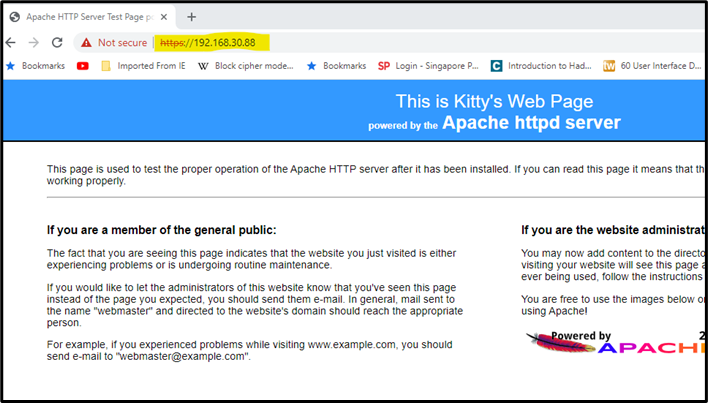


(Limit field set to 1 / second is to limit up to maximum 1 log per second.)

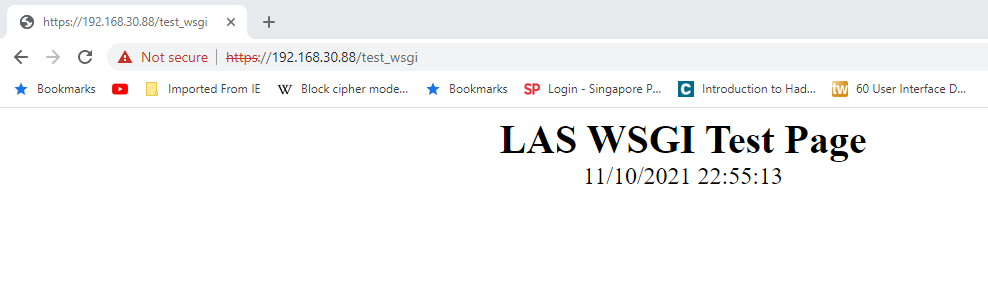
On client or On your host PC:

1. Use a web browser to access to a few different web pages of server.

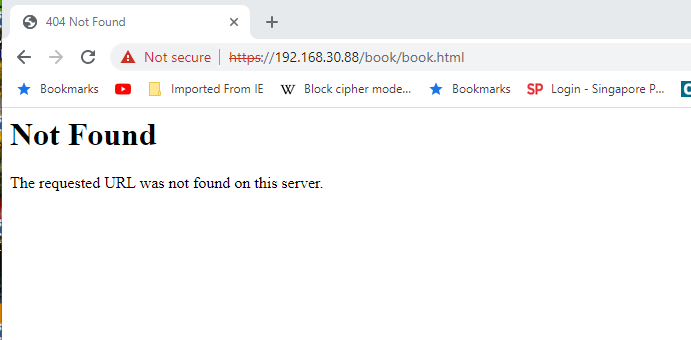
And include some invalid path. All these browsing attempts will trigger the logging operations.



(I have reloaded the default page.)



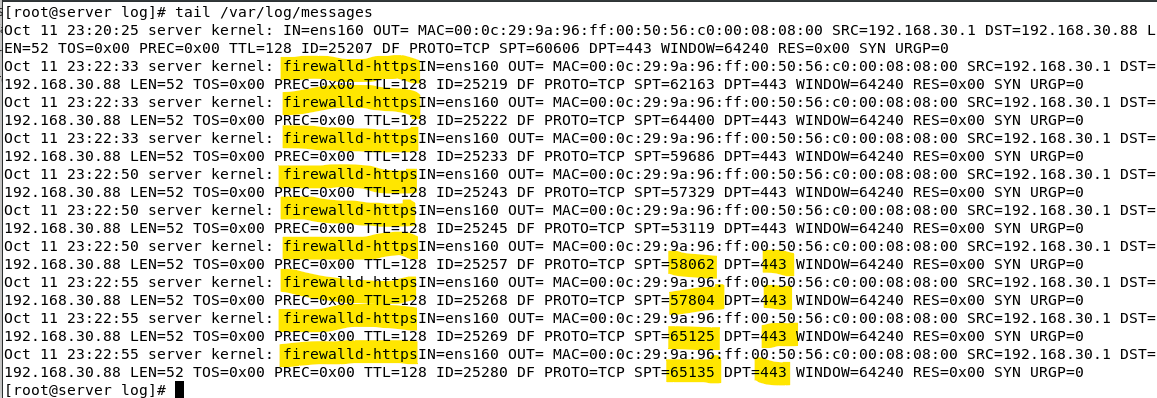
(I have visited the test\_wsgi page.)



(I have visited the /book/book.html page, but it is not found.)

On server:

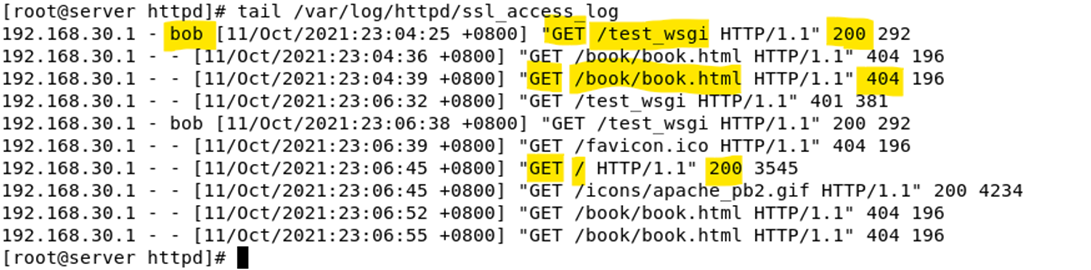
1. View the end of /var/log/messages to see the logged packet. Note the information captured from the packet (eg source address, source port, dest address, dest port, SYN flag of the packet).



(/var/log/messages collects many log entries from the kernel directly. The prefix we have set earlier in the rich rules will help us to locate the firewalld related log entries.)

Take note that, the above is a firewall log (not the httpd server access log). It is generated at the moment the firewall has allowed a https connection. **Whether the connection leads to a valid page or invalid page is not known at this moment.**

If you check out the /var/log/httpd/ssl\_access\_log you will find the higher level and useful information:



(At the httpd access log, you can tell whether the https request is successful or not.)

**Remove all the rich rules from the firewalld after you have completed the exercises.**

1. Limiting frequency of packets to discourage port scanning (STOP HERE)

On server:

1. Remove all the rich rules you have set in the earlier exercise.
2. Allow http and ftp service.
3. Reload the firewall.

On client:

1. Login as root. Run a SYN scan against your server using nmap. Install the nmap package if it is not yet installed.

nmap –sS <*serverIP>*

Depending on the configuration of the firewall on your server, you should see that some ports are opened.

To be able to compare the current and later port scanning result, rerun the nmap command within the time command:

time nmap -sS <serverIP>

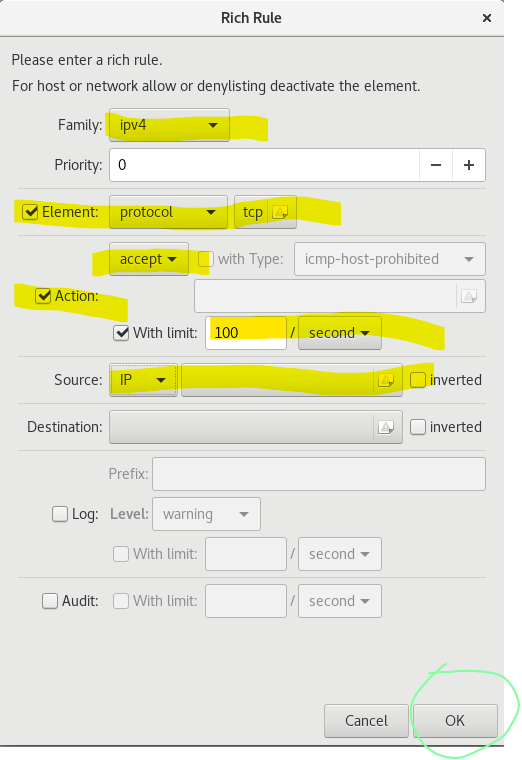
Text

Description automatically generated

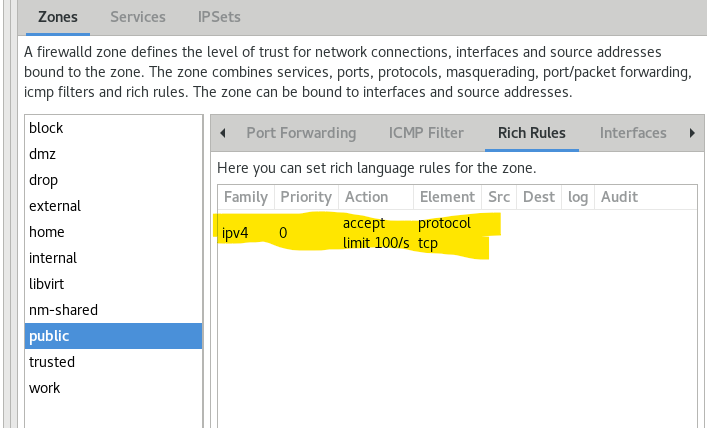
Based on the sample, it took about 20 seconds to complete the SYNC port scan at this point. (You may run a few times to take the average result.)

On server:

1. In the Firewall GUI, add a Rich Rule that will limit 100 new TCP connections per second (see following diagram)



Note : By adding this rule, you are allowing all TCP connections to your server.



The above is the content of the Rich Rules Tab after applying the rule.

On client:

1. Run the SYN scan against your server again. This time, the scan takes a longer time to complete. And there is one more open port, 9090 , being detected correctly, as all the TCP ports are opened now. At the same time, there are a lot of filtered ports (that nmap cannot determine their states) being reported. These confusing results are causing by the connection limits\* set on the firewall.

Text

Description automatically generated

The total scanning time also increased to 1 minutes 37 seconds!

With this+, we can 'discourage' port scanning activities.

\*Note on the SYNC packet responses:

In terms of network ports, they can only be in two states: open or closed.

When nmap reports its scanning result, it may report more then these two states.

There are 3 basic outcomes for a SYNC packet request of a connection:

a - The connection is made successfully, the receiving host return an ACK packet to the requesting host. Based on this, nmap reports the port is 'open' and there is no firewall to block this connection.

b - The connection attempt is failed because the receiving host returns an RST packet. Based on this, nmap reports the port is 'closed' and there is no service listening on the destinated port. It also implies there is a firewall setup between the sending host and the target host. However the firewall does not block this port.

c - The connection attempt is also failed since there is no ACK nor RST packet returns. The receiving host just drops (or discard) the incoming packet (Due to the firewall denial). This case will cause the request host to retry a few time until a timeout to determine that the port is not reachable (Due to firewall blockage.). The nmap will exclude the associated port from the output list.

When we set the TCP is accepted while with a limited number of packet (e.g. Maximum 100 per second.) it will confuse the nmap as sometimes for the same port the response is with either ACK or RST but sometimes there is no response. It will trigger nmap to retry more time. For these cases, nmap will list the port as filtered.

+Note on the discouragement:

There are two discourage factors:

a - The scanning time taken is much longer than the normal expected one. The above example is only applying the port scan on one target host. Imagine if the adversary is planning on scanning a network.

b - The misleading result may put off the interests for the adversary for further attempts.

On server:

1. Stop the firewalld. Type:

STOP HERE

systemctl stop firewalld

On client:

1. Run the SYN scan against your server again to find out how long (or short) it takes to complete the scanner when there is no firewall in place.

Text

Description automatically generated

There are only 6 open ports. Amazingly, nmap does not report all the other ports are 'closed' (though they are.)

On server:

1. Restart the firewalled and clear the Rich Rule you have added earlier so that the rest of the practical is not affected
2. Reverse Proxy via Network Address Translation (NAT)

Instead of apply NAT to allow internal users to share a public IP address for outgoing network operations.

You will now implement Network Address Translation (NAT) through firewalld to achieve a 'reverse proxy' feature. By deploying reverse proxy, you can reduce the attack surface of your real server by not letting it to face public access directly.

The Web Server is installed on your server image, which will be "hidden" from the outside world behind your client image. External users only connect to the client, which acts as the public interface. The client will forward the user’s requests to other systems such as your server image.

Public-facing computer

(client image)

External users

Web Server (server image)

These systems are “hidden” behind the Public-facing computer

User’s web requests forwarded to Web Server

On server:

1. Start the Apache server if it is not running yet.

On client:

1. Stop the firewall on the client.

systemctl stop firewalld

1. Make sure the Apache server is not running on the client.

systemctl stop httpd

systemctl disable httpd

1. Ensure there is no program listening on port 80.

netstat –tunpa | grep 'LISTEN'

and/or

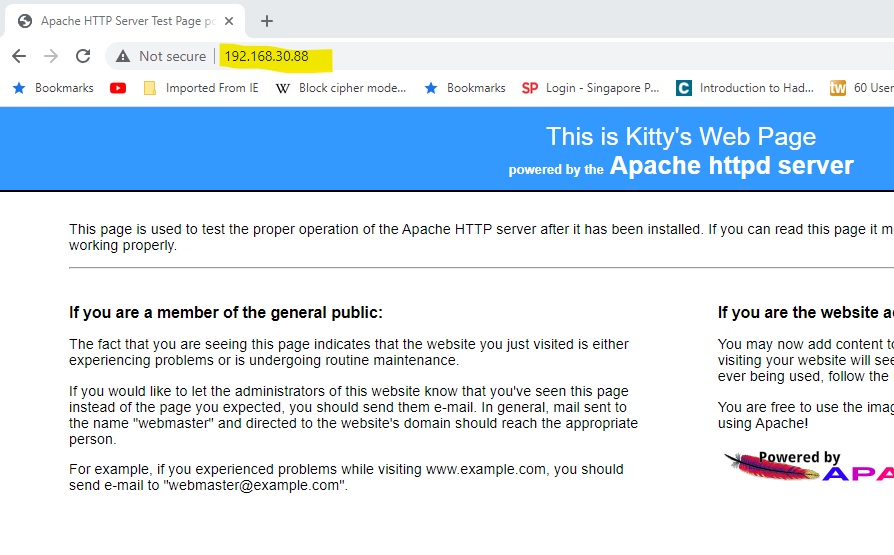
netstat -tunap | grep 'LISTEN' | grep 80

Calendar

Description automatically generated

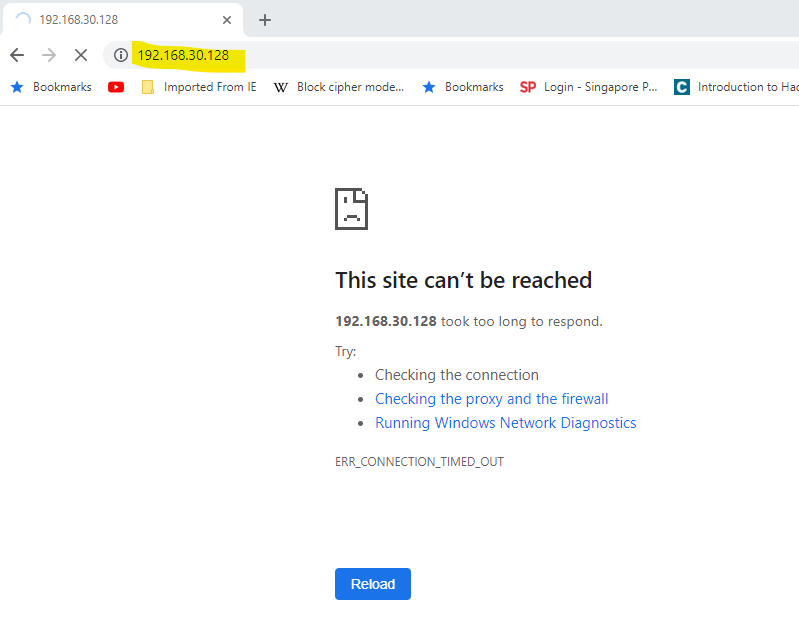
On Base PC:

1. Browse to http://<*server\_ip>* to see the web page of your server image. (you may need to adjust the firewall on your server image to allow the Base PC to connect to the Apache web server on the server image)



(Note: The server IP of the VM in the above is 192.168.30.88.)

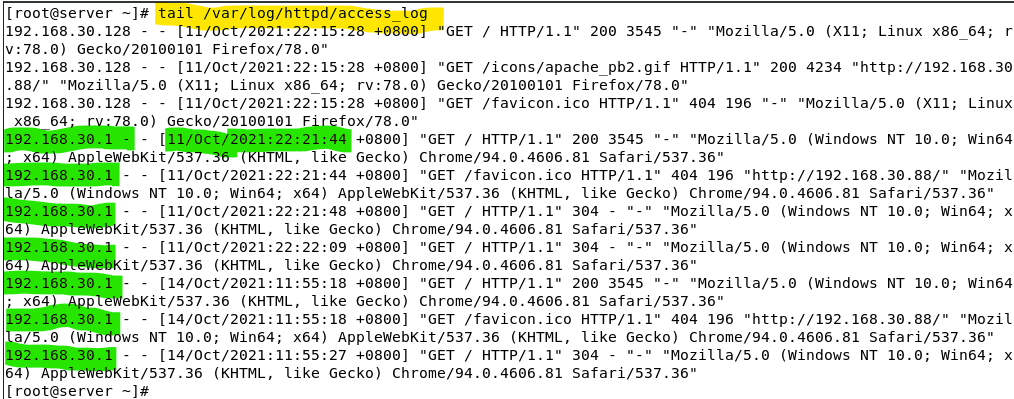
1. Browse to http://<*client\_ip>*. You should get an error message as there is no web server running on your client image.



(Note: The client IP of the VM in the above is 192.168.30.128.)

On server:

1. Check the latest content (log entires) of the /var/log/httpd/access\_log. You should be able to find the log entries that reflect the http requests from your Base PC to the apache server. (hint: tail /var/log/httpd/access\_log)



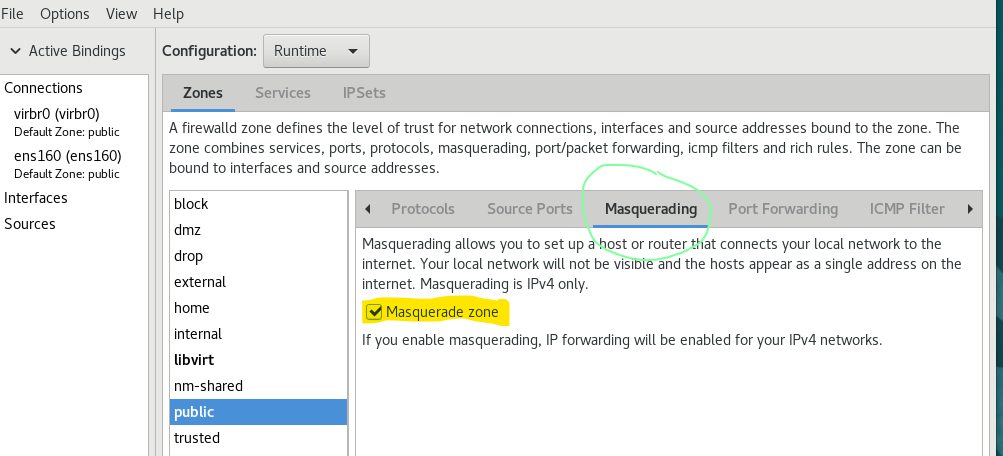
In the above, 192.168.30.1 is the ip address of the Base PC (This is the VMnet default setting. ). You should verify the date/time of the log entry too.

On client:

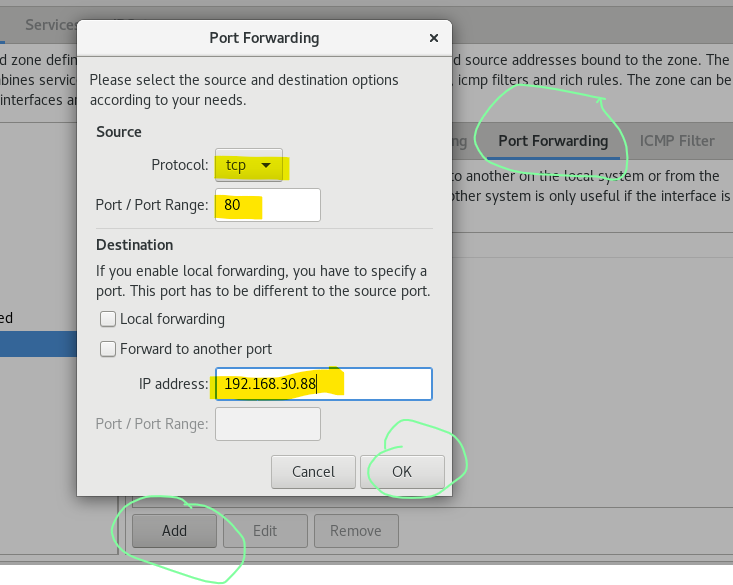
1. Enable the firewall on the client.

systemctl start firewalld

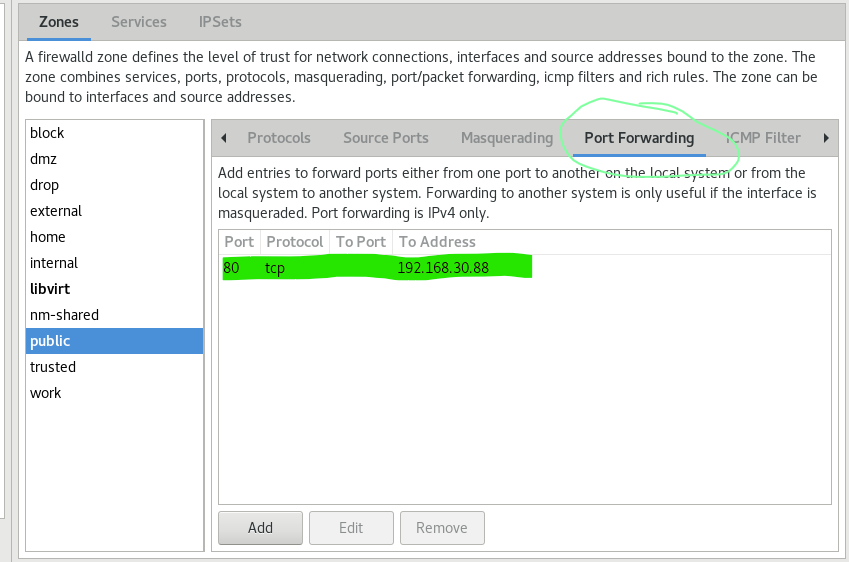
1. In the Firewall GUI, enable http service port (to allow incoming http request, even your client is not having a web server running).
2. In the Firewall GUI, enable IP masquerading from the Masquerading tab, and check Enable Masquerade. (see following diagram). This means all outgoing packets will be modified to have the same source IP as the client network interface card.



1. Next, move to the next tab, the Port Forwarding tab, and add the following entry to **forward** all incoming packets going to the Port 80 on your client to the Port 80 on your server (see following diagram)

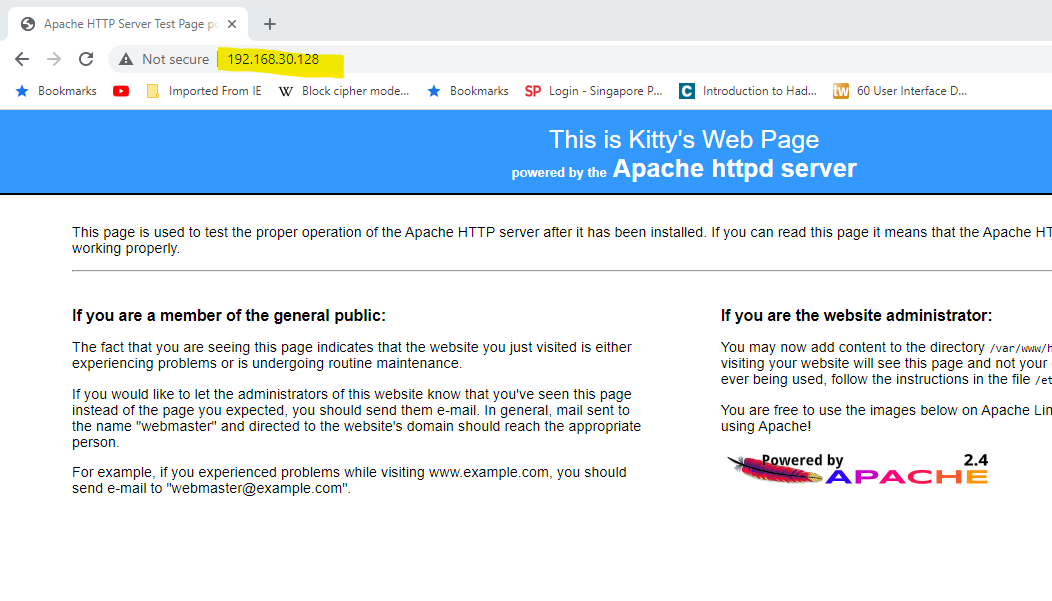


Change to your server IP

****

On Base PC:

1. Browse to http://*client\_ip*. Even though you are connecting to your client IP, your request has been forwarded to the server so you will see the web page of your server image. The server which receives the request would think the request is sent by the client (instead of from your base PC).



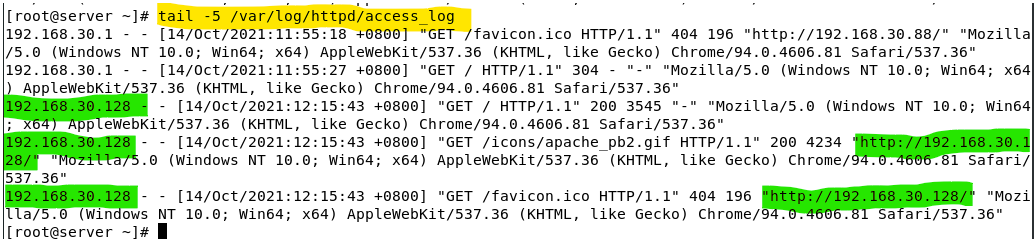
This is the IP address of the client.

On server:

1. You can verify the above by examining the /var/log/httpd/access\_log

You will find the access entries with the source IP of your client (instead of the Base PC) IP.

Moreover, the target destination addresses are also set t your client IP, because the browser was really trying to access to the Client IP. It was only the port mapping feature of the firewalld, mapped the request to your Server.

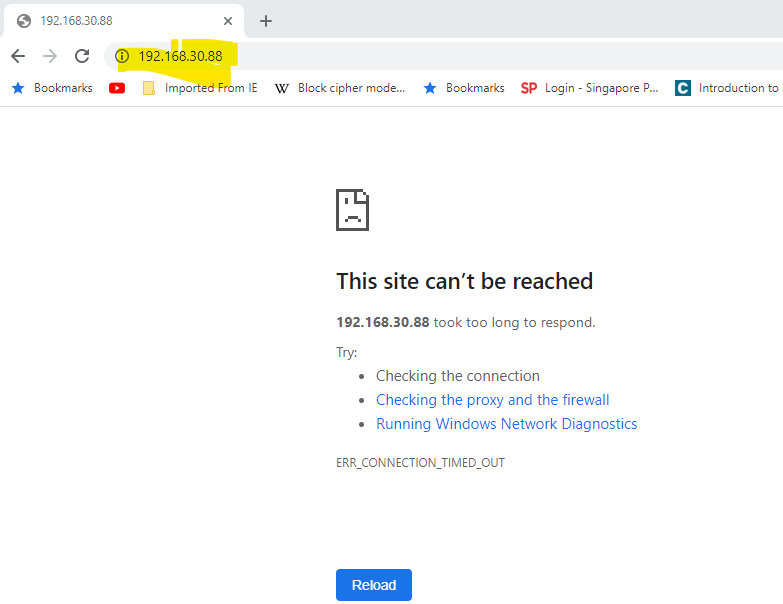


1. On your Server, change the firewall setting, to add a rich rule that rejects http access from everyone except your client.

Graphical user interface, application

Description automatically generated

1. On your base PC, verify that, you cannot access to your server web page directly. The only way to access to the web page is via the reverse proxy that set at your client VM.

****

Change to your server IP

**Note:** From the above exercise, you should learn that , the reject action of a Rich Rule overrides the normal allow rule.

Reset the firewalls of your systems

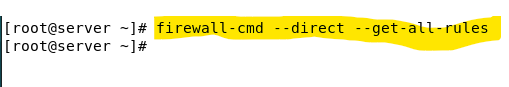
1. Reload the firewalls on both client and server to clear your runtime configuration.
2. Controlling outgoing traffic by using the direct interface

Firewalld is excellent in filtering the incoming traffic, but it is not able to control the outgoing traffic. If we want to create rules to control the outgoing traffic, to disallow connections to certain external IP addresses, we must add rules using the direct interface.

On server:

1. Check that you can browse an external website using HTTPS (eg https://www.bbc.com)
2. Use the following to list your current rules that have been defined for the direct interface.

firewall-cmd --direct --get-all-rules



( The above sample shows there is no direct rule found in your system.)

1. Add the following rule to block all outgoing ipv4 traffic through the direct interface. This rule has priority level set to 99. Rules with priority level 0 will be matched first. Level 99 is considered a quite low priority.

firewall-cmd --direct --add-rule ipv4 filter OUTPUT 99 -j DROP

1. Check that you can no longer browse external websites nor any ipv4 related outgoing traffic.

Text

Description automatically generated

1. Add the following rule that has higher priority to allow outgoing icmp traffic.
   1. firewall-cmd --direct --add-rule ipv4 filter OUTPUT 2 -p icmp -j ACCEPT

You can verify the effect by issuing a ping to 8.8.8.8

Text

Description automatically generated

1. Add the following rules to allow outgoing packets that belong to a connection that is already established

firewall-cmd --direct --add-rule ipv4 filter OUTPUT 3 -m state --state ESTABLISHED,RELATED -j ACCEPT

(This rule is particularly useful to allow a server to reply to its clients' requests)

1. Add the following two rules to allow outgoing traffic to HTTPS websites (**TCP Port 443**) and queries to DNS servers (**UDP Port 53**)

firewall-cmd --direct --add-rule ipv4 filter OUTPUT 4 -p tcp --dport=443 -j ACCEPT

firewall-cmd --direct --add-rule ipv4 filter OUTPUT 5 -p udp --dport=53 -j ACCEPT

and one more if you want to access to HTTP sites

firewall-cmd --direct --add-rule ipv4 filter OUTPUT 6 -p tcp --dport=80 -j ACCEPT

Note: Although we set each of added rules with a unique priority. The system allows rules to set with the same priority level.

1. Check that you can browse an external website using HTTPS (eg www.bbc.com).

Note that you may not be able to browse to external websites using HTTP (eg http://securitystartshere.org) if you have not allowed (have you ?) outgoing packets to the HTTP port.

1. Check that you cannot do a ssh to your client as your outgoing rules only allow HTTPS and DNS connections.
2. Caveat - Try to tell the difference of :

firewall-cmd --direct --get-all-rules

vs

firewall-cmd –list-all

Be careful when using rules through direct interface as these rules are not listed through the command **firewall-cmd --list-all**. You can view or edit them in the Firewall GUI under View->Direct Configuration.

To make the rules permanent, you need to add the **--permanent** option. Permanent direct interface rules are stored in /etc/firewalld/direct.xml.

1. To remove the rules using direct interface:

firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 2 -p icmp -j ACCEPT

firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 3 -m state --state ESTABLISHED,RELATED -j ACCEPT

firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 4 -p tcp --dport=443 -j ACCEPT

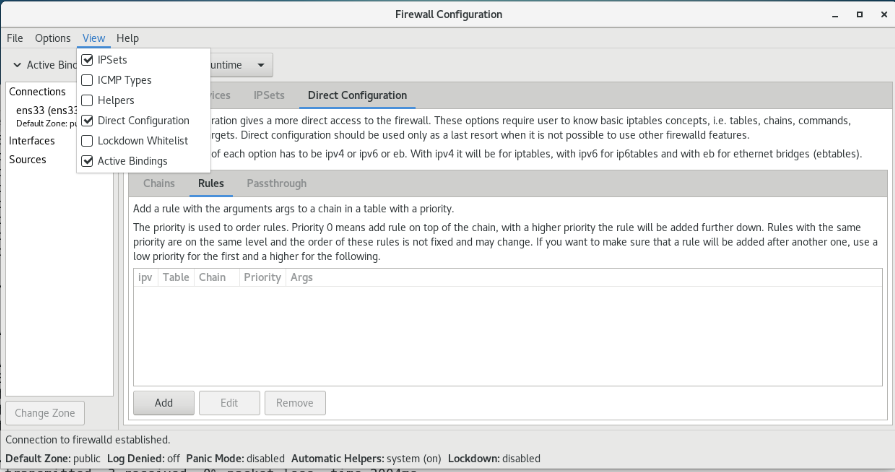
firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 5 -p udp --dport=53 -j ACCEPT

firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 6 -p tcp --dport=80 -j ACCEPT

firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 99 -j DROP

or

you may use the Firewall GUI to remove all the direct interface rules. (This requires you to enable the viewing of the Direct Configuration).



Graphical user interface, text, application, email

Description automatically generated

1. Protecting from brute force attack to sshd

ssh is a common way to enable remote secured shell access for Linux / Unix administrators. However, once the sshd service is enabled, it may be subjected to brute force attack.

One way to avoid this attack is to limit the number of failed login attempts.

We will install Fail2ban-firewalld package to our server to implement such brute force attack counter measure. Fail2ban package works together with firewalld.

Ref: ~https://devops.ionos.com/tutorials/install-fail2ban-on-centos-7-to-protect-ssh-via-firewalld/

~https://www.howtoforge.com/tutorial/how-to-install-fail2ban-on-centos/

Introduction of epel-release yum repository:

This repository provides Extra Packages for Enterprise Linux (EPEL) is a special interest group (SIG) from the Fedora Project that provides a set of additional packages for RHEL (and CentOS, and others) from the Fedora sources.

Ref: ~https://www.redhat.com/en/blog/whats-epel-and-how-do-i-use-it

On Server:

1. We need to add in epel-release repository to the system. Type:

dnf install oracle-epel-release-el8

1. Check if fail2ban has been installed yet.

dnf info fail2ban-firewalld

or

dnf list installed | grep fail2ban-firewalld

1. If it has not been installed run dnf update then install fail2ban package.

dnf update

dnf install fail2ban-firewalld

(Note: we only need fail2ban-firewalld for our setup. To install the full set of fail2ban, use dnf install fail2ban)

1. The default configuration folder of fail2ban is at /etc/fail2ban. The main configuration file that is related for banning access is named jail.conf. We can also add in additional 'jail' configuration settings by adding .local files into the /etc/fail2ban/jail.d subfolder.

Create a new text file 'sshd.local' into the /etc/fail2ban/jail.d subfolder.

Add the following lines into the file to define our 'jail' settings:

[sshd]

enabled = true

port = ssh

# you comment out the following if you only want to log the failed login.

banaction = iptables-multiport

logpath = %(sshd\_log)s

maxretry = 3

# connection will be banned for 600 seconds when tried more than 3 times

bantime = 600

1. Start the fail2ban service now. Type in:

systemctl start fail2ban

1. Verify if fail2ban service is running. Type in:

systemctl status fail2ban

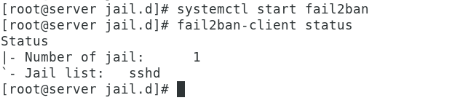
Text

Description automatically generated

(Note that the fail2an service is running but it is not enabled. It implies it will not automatically start in the next boot up.)

1. Verify if fail2ban is running and the sshd.jail is set. Type in:

fail2ban-client status



(The above shows that the sshd jail is ready.)

1. To check whether there is any banned ip. Type:

fail2ban-client status sshd

Graphical user interface, text, application, email

Description automatically generated

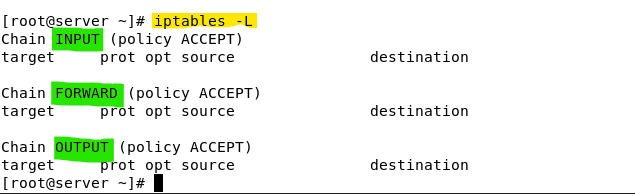
1. Now the fail2ban is activated and ready to block connections from blacklisted IP addresses. The blacklist IP addressess will be defined in a customised iptables chain\* which can be used to define iptables direct rules to deny / accept a set of IP addresses. Verify if your system has any iptables chain objects, type in:

iptables -L

or

iptables -L <a known iptable chain name>

Note: There is no customised iptable chain defined in the system at the beginning even the fail2ban has been started.



(Note: INPUT, FORWARD, and OUTPUT are system build-in default chains.)

On Client:

1. Verify if fail2ban is guarding your sshd from brute force attack by trying ssh to your server with wrong password.



In the above session, the first ssh session to server from client is successful. The second attempt uses all wrong password then triggers the fail2ban effect.

On Server: (within 5 minutes)

1. Verify if a new iptables chain has been defined by fail2ban and if there are any direct rule in the chain to block your client IP from accessing to the server.

Graphical user interface, application

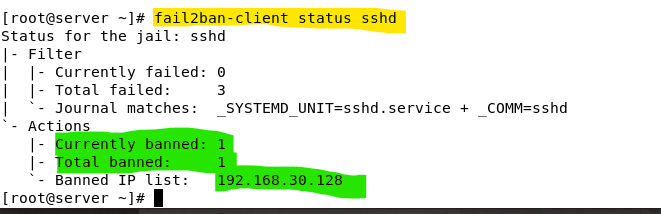
Description automatically generated with medium confidence

As depicted at the above, the name of the new chain is f2b-sshd. A new rule is added to the default INPUT chain to channel all the incoming ssh traffic to f2b-sshd chain.

In the definition of the f2b-sshd chain, we can see the client IP has been blocked by a REJECT rule. The last rule in the f2b-ssh chain is to return the flow to the INPUT chain.

1. Verify the fail2ban-client reports the same set of information. type:

fail2ban-client status sshd



1. Verify if the /var/log/secure file has logged the invalid ssh access attempts :

cat /var/log/secure | grep -i 'failed password'

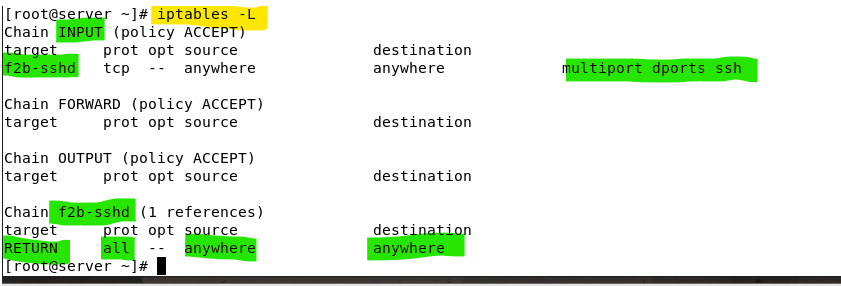
Text

Description automatically generated

(Note: grep with -i switch is to allow non-case sensitive matching)

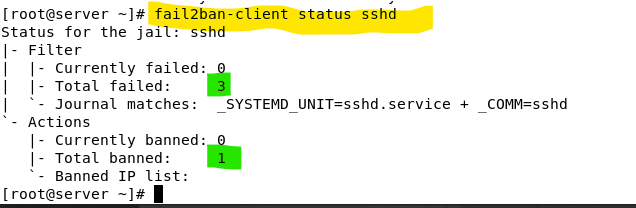
For you information, the working concept of fail2ban is based on the log content to determine there are failed attempts made on various services. SSH is one of them, based on the similar approach, administrator can use fail2ban to monitor and ban access on web service, email service and etc ….

1. Wait for 5 minutes or longer and verify what will happen to the f2b-sshd Chain and the iptable rule:



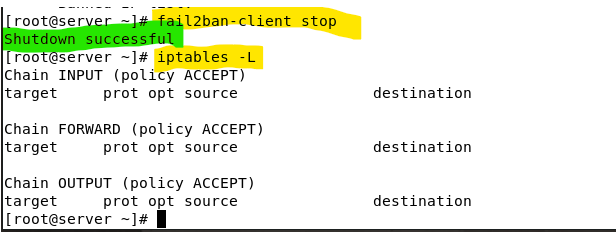
As shown , the iptables chains are remaining. But there are no more REJECT rules.

1. Rerun the fail2ban-client status sshd:



1. Reset your server setting by stopping the fail2ban **gracefully**. Type:

fail2ban-client stop



As shown, fail2ban-client will remove the customised f2b-sshd chain from the iptables.

**Additional Reference:**

Home page of the firewalld - <https://firewalld.org/>

firewall-cmd Cheat Sheet - https://cheatography.com/mikael-leberre/cheat-sheets/firewall-cmd/

Running a quick NMAP scan - <https://www.redhat.com/sysadmin/quick-nmap-inventory>

Port Scanning Basics - <https://nmap.org/book/man-port-scanning-basics.html>

Linux security: Protect your systems with fail2ban - https://www.redhat.com/sysadmin/protect-systems-fail2ban

*End of Practical*