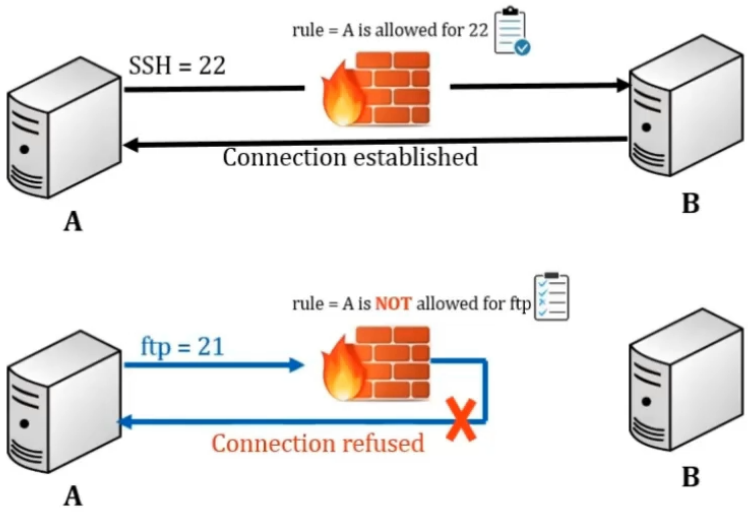
Managing network security (Firewall)

Linux #redhat #network #firewall

Introduction to Firewall

- **What is a Firewall?**
  - A wall that prevents the spread of fire
  - When data moves in and out of a server its packet information is tested against the firewall rules to see if it should be allowed or not
  - In simple words, a firewall is like a watchman, a bouncer, or a shield that has a set of rules given and based on that rule they decide who can enter and leave
  - There are 2 type of firewalls in IT
    - Software = Runs on operating system
      - This will be the focus on this note
    - Hardware = A dedicated appliance with firewall software
      - This kind of appliance is often found on the network side, the network team, or the one who actually maintains that type of firewall.



- You may see this firewall sitting in the middle, and you are probably thinking, "Where is it exactly sitting?". But in reality this firewall is sitting on server A, as well as there's another firewall that's also sitting on server B. Both servers A and B do have firewalls as well.
- Server A who wants to connect to server B could also have a rule whether this service is even allowed to leave the server A or not.
- When it goes to server B, the server B has its own firewall which would say "Hey, even though it came in, but let me see if this service is allowed to be accepted to my server".

Firewall (`firewalld`)

- `firewalld` works the same way as `iptables` but of course it has it own commands

    - `firewall-cmd`
        - Note the "firewall" word on this command does not include the "d" as in the name of the service.
- It has a few pre-defined service rules that are very easy to turn on and off
    - Services such as: NFS, NTP, HTTPD etc.
        - These services are already there in an `.xml` format, all you have to do is add those services into the `firewalld` table and the rules.
- `firewalld` also has the following:
    - **Table**
        - This is the one that has all the information about chain rules and targets
    - **Chains**
        - These are the ones that has outgoing, incoming, or forward traffic information.
    - **Rules**
        - The rules are associated to the chains, like if incoming traffic matches or does not match this rule, then what to do then comes the target.
    - **Targets**
        - The target would be either drop, reject, or accept the traffic.

Disable `iptables`

- You can run one or the other
    - `iptables` or `firewalld`
- Make sure `iptables` is stopped, disabled and mask before you start `firewalld`
    - `systemctl stop iptables`
    - `systemctl disable iptables`
    - `systemctl mask iptables`
        - Remember: By masking the service you or any other program won't start it accidentally.
    - In my RedHat 9 distribution, `iptables` was not reachable even when I installed the package.
        - Solved by running `dnf install iptables-services` to install the actual `iptables` service

Example using `systemctl` command:

```
[root@localhost ~]# systemctl status iptables
○ iptables.service
     Loaded: masked (Reason: Unit iptables.service is masked.)
     Active: inactive (dead)
```

- This is how the `iptables` service should look like after stopping, disabling, and masking it.

Check if `firewalld` package is installed

- Installed by default on RedHat and CentOS versions 7 and above.
- `rpm -qa | grep firewalld`
- If you do not find it, you can run the `dnf/yum` command to install it.

Example using `rpm -qa` command:

```
[root@localhost ~]# rpm -qa | grep firewalld
firewalld-filesystem-1.3.4-1.el9.noarch
```

```
firewalld-1.3.4-1.el9.noarch
```

- This is the package that should be installed in order to use `firewalld` service

Start `firewalld`

- `systemctl start firewalld`
- `systemctl enable firewalld`

Example using `systemctl` command:

```
[root@localhost ~]# systemctl unmask firewalld
```

- In case the service is not allowed to start you can use the `umask` option with `systemctl` to let the service start by you or any other service.

Check the rule of `firewalld`

- `firewall-cmd --list-all`
  - There are services predefined by `firewalld` and are written in `.xml` format, and you could simply add those services and remove them in one single command.

Example using `firewall-cmd` command:

```
[root@localhost ~]# firewall-cmd --list-all
```

Output:

```
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

- This is the table, it's a public table, and it has the services by default
  - `ssh`, `dhcpv6-client`
    - These services are already enabled, every time we start up, by default, ssh is allowed to come in for example.
- You could define ports or any specific rule that you have under 'rich rules'
- If you add something, you could run the command `firewall-cmd` followed by the correct syntax to add a service or port.

Get the listing of all services `firewalld` is aware of:

- `firewall-cmd --get-services`

Example using `firewall-cmd` command:

```
[root@localhost ~]# firewall-cmd --get-services
```

Output:

```
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit
ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin
bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine
checkmk-agent cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6
dhcpv6-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-
client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-
replication freeipa-trust ftp galera ganglia-client ganglia-master git gpsd grafana gre high-
availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isns
jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-
control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-
nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-
worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp
managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql
nbd nebula netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-
storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy
prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel
radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-
dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync
spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui syncthing-relay synergy
syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vdsm vnc-
server warpinator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-
discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-
server zerotier
```

- These are the predefined services, all of them separated by a space and they are written in alphabetical order.
- It doesn't mean that these services are already part of the firewall rules to allow all these services by default to be enable.
- This is telling `firewalld` "Hey, you have the service", all you have to do is run a command and tell which is the service you want to enable.

To make `firewalld` re-read the configuration added

- `firewall-cmd --reload`
  - Meaning if you added a rule in `firewalld` you will always have to run this command so your firewall process would know about the new rule or new information that you added into the table.

The `firewalld` has multiple zones, to get a list of all zones

- `firewall-cmd --get-zone`

Example using `firewall-cmd` command:

```
[root@localhost ~]# firewall-cmd --get-zones
```

Output:

```
block dmz drop external home internal nm-shared public trusted work
```

- You could use any zone that is required by your company or policies
- Most of the time it is on the public zone (this is the most used zone)

To get a list of active zones

- `firewall-cmd --get-active-zones`

Example using `firewall-cmd` command:

```
[root@localhost ~]# firewall-cmd --get-active-zones
```

Output:

```
public
  interfaces: enp0s3
```

- This is listing that the zone that is active in terms of firewall, where the firewall is getting all its rules, configuration, is public.
- The interface attached to that zone is 'enp0s3'
- Note it only returned the public zone

To get firewall rules for public zone

- `firewall-cmd --zone=public --list-all`

Example using `firewall-cmd` command:

```
[root@localhost ~]# firewall-cmd --zone=public --list-all
```

- When you only have one zone active, then you really don't have to specify the zone name.

Output:

```
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
```

```
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

- Note this is the same output from `firewall-cmd --list-all`
- This is the zone which has, by default, out of the box, if you are running the `firewalld` for the first time this is what you are gonna see.

Add a 3rd party service

- All services are pre-defined by `firewalld`. What if you want to add a 3rd party service?
- `/usr/lib/firewalld/services/allservices.xml`
  - All the pre-defined services are in this directory
- Simply `cp` any `.xml` file and change the service and port number

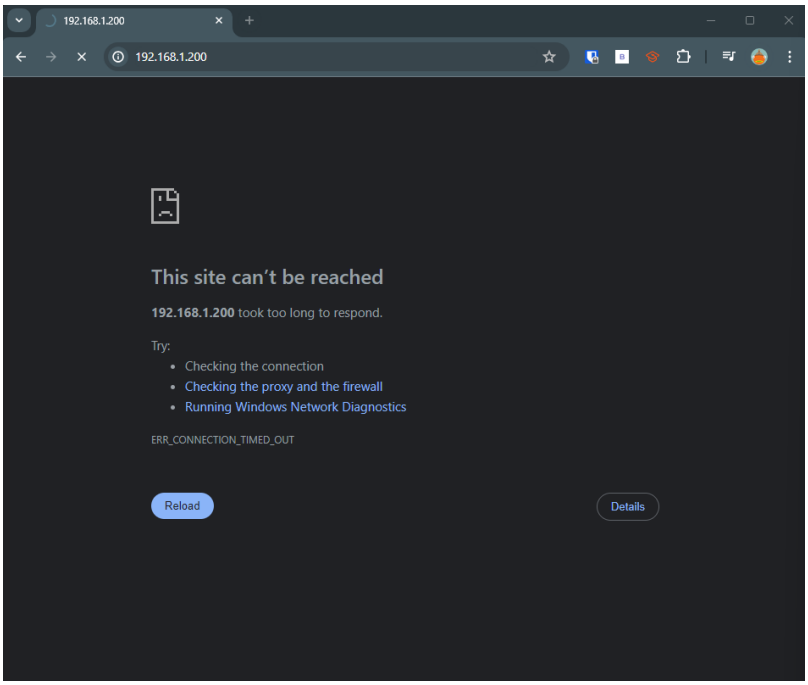
Output from `cat ssh.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>Secure Shell (SSH) is a protocol for logging into and executing commands on remote
machines. It provides secure encrypted communications. If you plan on accessing your machine remotely
via SSH over a firewalled interface, enable this option. You need the openssh-server package installed
for this option to be useful.</description>
  <port protocol="tcp" port="22"/>
</service>
```

- When you open an `.xml` file this is what it's gonna look like.
- There are many files in the `/usr/lib/firewalld/services` directory that probably have a longer description.
- When you `cat` any on the files in that directory that ends with `.xml` you'll see:
  1. You'll see first its version of XML
  2. Secondly you'll see what type of service is that (SSH)
  3. Then you'll see the entire information about the service
     - This includes the description of the service and of course the port information and the service name.

Add a service (http)

- `firewall-cmd --add-service=http`
- Right now when you run the `firewall-cmd --list-all` command to check the firewall enabled services you won't see the `http` service.
  - So if you have Firefox for example and you could open it up and you could try to access your HTTP service, you'll see that it will not be accessible.
  - But of course in order to check that, first you need to make sure your HTTPD service is running.
- To start the `http` service:
  - `systemctl status httpd`
  - `systemclt unmask httpd` (in case the service is masked)

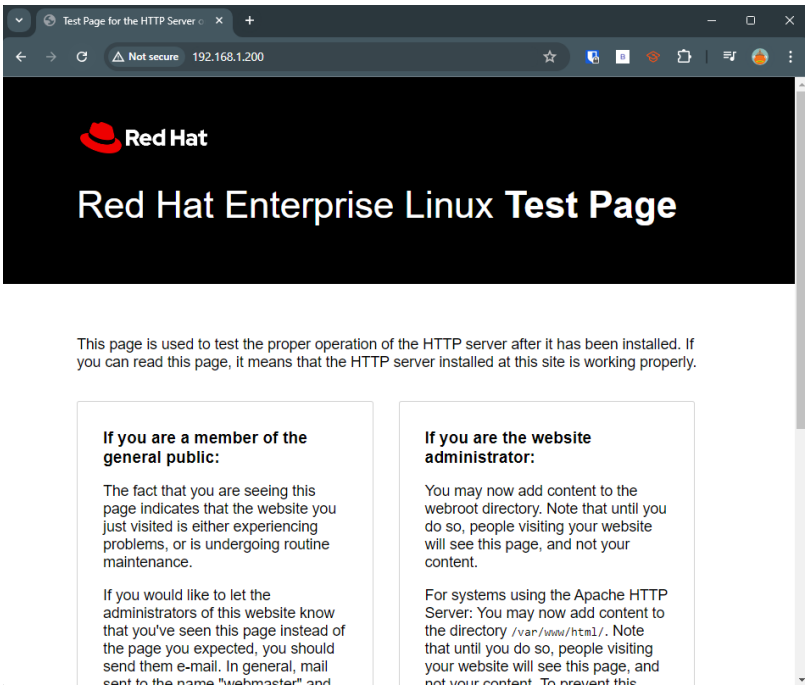
    - `systemctl start httpd`
- Once it is active you can go ahead and open your browser to make sure you can access your Apache server using HTTP protocol service.
    - 

        - The IP address of my machine is 192.168.1.200, so when I reload you will see it will say "This site can't be reached" because of time out.
- In order to get to your server through that HTTP service, you have to enable it at the firewall level.
- Run `firewall-cmd --add-service=http`

Example using `firewall-cmd` command:

```
[root@localhost ~]# `firewall-cmd --add-service=http`
success
```

- Now you can run `firewall-cmd --list-all` you will see in the services section that the `http` service has been added.
- Now if you go to your browser and refresh the screen:



- Now you will be able to see your splash page and it works perfectly fine.

Remove a service (http)

- `firewall-cmd --remove-service=http`
  - Note we only changed the keyword "add" by "remove"
- It will remove it, you could check the status by listing firewall table and you will see it's gone now.

To reload the `firewalld` configuration

- `firewall-cmd --reload`
- If you add any service to the firewall table just by using the `firewall-cmd --add-service` command without changing anything form the `firewalld` configuration and you run `--reload` command it will reload to its previous configuration set up, meaning you won't see the service you just added with the `-add-service` command under the firewall table after reloading the firewall configuration.
- It's kind of a way to flush whatever the temporary rules are in there
- If you want to add any rule permanently, you have to specify "permanent" in the end

To add or remove a service permanently

- `firewall-cmd --add-service=http --permanent`
  - This line will add the service permanently to that zone, which is the public zone right now.
- `firewalld-cmd --remove-service=http --permanent`
  - If you want to remove the service permanently you can run this line

To add a service that is not pre-defined by `firewalld` (3rd party)

- `/usr/lib/firewalld/services/allservices.xml`

- Simply `cp` any `.xml` file (`sap.xml`) and change the service and port number (32)
- `systemctl restart firewalld`
- `firewall-cmd --get-services` (to verify new service)
- `firewall-cmd --add-service=sap`

Travel to directory
Example using `cd` command:

```
[root@linuxtest ~]# cd /usr/lib/firewalld/services/
[root@linuxtest services]# ls -ltr
total 884
-rw-r--r--. 1 root root 242 Nov  6  2023 zerotier.xml
-rw-r--r--. 1 root root 315 Nov  6  2023 zabbix-server.xml
-rw-r--r--. 1 root root 314 Nov  6  2023 zabbix-agent.xml
-rw-r--r--. 1 root root 545 Nov  6  2023 xmpp-server.xml
...
```

- Once you are in the `/usr/lib/firewalld/services/` directory and you run `ls -l` command you'll see all those services that you get when you run the command `firewall-cmd --get-services`
  - All these services that you see, `firewalld` is pulling them from these files.
- In this example we are looking to add the SAP service.

Go ahead and take any file
Example using `cp` command:

```
[root@linuxtest services]# cp ssh.xml sap.xml
```

Output from `ls -l sap.xml`:

```
-rw-r--r-- 1 root root 463 Jul  9 22:39 sap.xml
```

Modify the file
Example using `vi` command:

```
[root@linuxtest services]# vi sap.xml
```

File editor:

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>Secure Shell (SSH) is a protocol for logging into and executing commands on remote
machines. It provides secure encrypted communications. If you plan on accessing your machine remotely
via SSH over a firewalled interface, enable this option. You need the openssh-server package installed
for this option to be useful.</description>
  <port protocol="tcp" port="22"/>
</service>
```

- Remove the short description and change it to SAP
- Remove everything in the description field

    - Remember you can Escape the INSERT mode and just keep pressing 'x' to delete character by character.
    - Add a new description for the service, in this example we will just add "This is a 3rd party app" as the new description
- The protocol in this case is TCP or whatever you wanted to specify, but you must change the port field to port '32'
    - Of course you need to know these things, whether it's a TCP, UDP, what port number is, what's the service
- Now you can go ahead and save that file.

The new file should look like this
File editor:

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SAP</short>
  <description>This is a 3rd party application service</description>
  <port protocol="tcp" port="32"/>
</service>
```

- You can always verify this by running `cat sap.xml` to confirm its contents
- if you want to see that SAP service in your listing, meaning when you run the `firewall-cmd --get-services` command and you wanted to see it there, then you would have to restart the `firewalld` service.

Restart `firewalld`
Example using `systemctl` command:

```
[root@linuxtest services]# systemctl restart firewalld
[root@linuxtest services]# firewall-cmd-get | grep sap
... sap ...
```

- Now when you run that `--get-services` you should be able to see your SAP service
- If you `grep` it, it will apear highlighted in red
- Now we could go ahead and add it as a firewall enabled service

Add the service to firewall configuration
Example using `firewall-cmd` command:

```
[root@linuxtest services]# firewall-cmd --add-service=sap
success
[root@linuxtest services]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client sap ssh
...
```

- When you add the service and you do a listing now you'll see the `sap` service added to the services section
- Now, if you're trying to add any other services that is not in that `.xml` file it will just simply fail, it's not gonna run.
    - For example trying to run `firewall-cmd --add-service=ironman` will return `Error: INVALID_SERVICE: ironman`
        - The reason is because it went into that `/usr/lib/firewalld/services directory`, it looked for the `ironman.xml` file, and it doesn't exist.
- This way you could just simply have to add or remove services that are defined in that directory
- Remember to run `firewall-cmd --reload` if you do not want this service to be added at the end.

To add a port

- `firewall-cmd –add-port=1110/tcp`

Example using `firewall-cmd` command:

```
[root@linuxtest services]# firewall-cmd --add-port=1110/tcp
success
[root@linuxtest services]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 1110/tcp
...
```

- You could pick whichever port that you wanted to open and specify the protocol.
- You will see it under ports when listing the firewall table.
- Remember that if you do not specify `--permanent`, it will go away the moment you run `--reload`.

To remove a port

- `firewall-cmd --remove-port=1110/tcp`
  - Works the same as the `--add-port` command but to remove.

To reject incoming traffic from an IP address

- This is kind of an scenario where you have someone coming into your machine with an IP address and you do not like that person or that machine and you just want to block any traffic from that IP address.
- You will run this as a rule.
- `firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.168.0.25" reject'`
  - Remember rich rules are shown at the bottom of `firewall-cmd --list-all`

Example using `firewall-cmd` command:

```
[root@linuxtest services]# firewall-cmd --add-rich-rule='rule family="ipv4" source
address="192.168.0.25" reject'
success
```

- Note the whole rule is enclosed by `''` (single quotes)
- You can verify this by listing the firewall table with `--list-all`
- To remove this rich rule you could press the Up arrow key to re-type the last command and change the 'add' keyword to 'remove'
  - You could also do `firewall-cmd --reload`

To block and unblock ICMP incoming traffic

- What if you do not want people to ping you.
- `firewall-cmd --add-icmp-block-inversion` (To block all incoming ping traffic)

Example using `firewall-cmd` command:

```
[root@linuxtest services]#  firewall-cmd --add-icmp-block-inversion
success
[root@linuxtest services]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: yes
...
```

- The ping comes on the protocol ICMP
- Note we can also verify we added the ping block with the same `firewall-cmd --list-all` command
- If you try to ping your machine IP from any other machine you will get a "host unreachable" message.
- You can revert this by running the same command but replacing the keyword 'add' with 'remove' or reloading the firewall

To block outgoing traffic to a specific website/IP address

- As an example, What if you want to block all the traffic that goes outside to Facebook?
- For that you'll have to find out what's the IP address of that page.
  - To find the IP of a page you have to run the command `host -t a [website]`
- `host -t a www.facebook.com` = find IP address
- `firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -d 31.13.71.36 -j DROP`

Find website IP
Example using `host -t` command:

```
[root@localhost]# host -t a www.facebook.com
www.facebook.com is an alias for star-mini.c10r.facebook.com.
star-mini.c10r.facebook.com has address 157.240.3.35
```

Try to ping that IP
Example using `ping` command:

```
[root@localhost ~]# ping 157.240.3.35
PING 157.240.3.35 (157.240.3.35) 56(84) bytes of data.
64 bytes from 157.240.3.35: icmp_seq=1 ttl=53 time=52.9 ms
64 bytes from 157.240.3.35: icmp_seq=2 ttl=53 time=54.1 ms
64 bytes from 157.240.3.35: icmp_seq=3 ttl=53 time=54.4 ms
^C
--- 157.240.3.35 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 52.943/53.821/54.438/0.637 ms
```

- Just to make sure that you could go out.

Block that IP outgoing traffic
Example using `firewall-cmd` command:

```
[root@localhost]# firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -d 157.240.3.35 -j DROP
success
```

- Note the `-j DROP` at the end is very important to complete our purpose

Ping that IP address again
Example using `ping` command:

```
[root@localhost ~]# ping 157.240.3.35
PING 157.240.3.35 (157.240.3.35) 56(84) bytes of data.
^C
--- 157.240.3.35 ping statistics ---
11 packets transmitted, 0 received, 100% packet loss, time 10221ms
```

- Now you are not permitted to go outside.
- You can go ahead and `--reload` your firewall to revert this rule
    - If you still are not able to ping the website IP even after you reload your firewall configuration you can go ahead and restart the `firewalld` service by running `systemctl resatart firewalld`, maybe it wasn't cached.

This is very close to the commands we run in `iptables`