

# Firewall

- [Firewall iptables script](#)
- [iptables](#)
- [Firewalld](#)

# Firewall iptables script

```
# Interfaces
WAN="ens3"
LAN="ens9"

#ifconfig $LAN up
#ifconfig $LAN 192.168.1.1 netmask 255.255.255.0

echo 1 > /proc/sys/net/ipv4/ip_forward
sysctl -w net.ipv4.ip_forward=1

iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -X

# Default to drop packets
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Allow all local loopback traffic
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Allow output on $WAN and $LAN if. Allow input on $LAN if.
iptables -A INPUT -i $LAN -j ACCEPT
iptables -A OUTPUT -o $WAN -j ACCEPT
iptables -A OUTPUT -o $LAN -j ACCEPT

iptables -A INPUT -p tcp -i $WAN --dport 22 -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -o $LAN -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -i $LAN -o $WAN -j ACCEPT
iptables -t nat -A POSTROUTING -o $WAN -j MASQUERADE
```

```
# Allow ICMP echo reply/echo request/destination unreachable/time exceeded
```

```
iptables -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
```

```
iptables -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
```

```
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

```
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

```
# WWW
```

```
iptables -t nat -A PREROUTING -p tcp -i $WAN -m multiport --dports 80,443 -j DNAT --to 10.1.1.11
```

```
iptables -A FORWARD -p tcp -i $WAN -o $LAN -d 10.1.1.11 -m multiport --dports 80,443 -j ACCEPT
```

```
exit 0 #report success
```

# iptables

## iptables arguments

-t = table, -X = del chain, -i = interface

## Deleting a line:

```
iptables -L --line-numbers  
iptables -D (CHAIN) (LINE NUMBER)
```

## Nating:

example for FTP NAT:

```
iptables -t nat -A PREROUTING -p tcp --dport 21 -j DNAT --to-destination 192.168.1.100:21  
iptables -t nat -A PREROUTING -p tcp --dport 49152:65534 -j DNAT --to-destination 192.168.1.100:49152-65534
```

to check a nat rule:

```
iptables -t nat -nvL
```

## masquerade traffic from an IP to another host

Enable ip forwarding

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Then, we will add a rule telling to forward the traffic on port 1111 to ip 2.2.2.2 on port 1111:

```
iptables -t nat -A PREROUTING -p tcp --dport 1111 -j DNAT --to-destination 2.2.2.2:1111
```

and finally, we ask IPtables to masquerade:

```
iptables -t nat -A POSTROUTING -j MASQUERADE
```

Optionally, you could only redirect the traffic from a specific source/network with, for a host only:

```
iptables -t nat -A PREROUTING -s 192.168.1.1 -p tcp --dport 1111 -j DNAT --to-destination 2.2.2.2:1111
```

or for a whole network

```
iptables -t nat -A PREROUTING -s 192.168.1.0/24 -p tcp --dport 1111 -j DNAT --to-destination 2.2.2.2:1111
```

that's it, now the traffic to port 1111 will be redirected to IP 2.2.2.2 .

If you go on host 2.2.2.2, you should see a lot of traffic coming from the host doing the redirection.

# Firewalld

## Zones

Pre-defined zones within firewalld are:

- **drop**: The lowest level of trust. All incoming connections are dropped without reply and only outgoing connections are possible.
- **block**: Similar to the above, but instead of simply dropping connections, incoming requests are rejected with an icmp-host-prohibited or icmp6-adm-prohibited message.
- **public**: Represents public, untrusted networks. You don't trust other computers but may allow selected incoming connections on a case-by-case basis.
- **external**: External networks in the event that you are using the firewall as your gateway. It is configured for NAT masquerading so that your internal network remains private but reachable.
- **internal**: The other side of the external zone, used for the internal portion of a gateway. The computers are fairly trustworthy and some additional services are available.
- **dmz**: Used for computers located in a DMZ (isolated computers that will not have access to the rest of your network). Only certain incoming connections are allowed.
- **work**: Used for work machines. Trust most of the computers in the network. A few more services might be allowed.
- **home**: A home environment. It generally implies that you trust most of the other computers and that a few more services will be accepted.
- **trusted**: Trust all of the machines in the network. The most open of the available options and should be used sparingly.

Verify what zone is used by default

```
firewall-cmd --get-default-zone
```

Verify what zones are active

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

View all info for default zone

```
firewall-cmd --list-all
```

List pre-defined zones and custom zone names

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

View all information for a specific zone

```
firewall-cmd --permanent --zone= home --list-all
```

Change default zone

```
firewall-cmd --set-default-zone= home
```

Adding a service to a zone

First it is recommended to not add `--permanent` and to test if the service is reachable, if it works add the `--permanent`

```
firewall-cmd --zone=public --permanent --add-service=http
```

Removing/Denying a service

```
firewall-cmd --zone=public --permanent --remove-service=http
```

List services

```
firewall-cmd --zone=public --permanent --list-services
```

Removing/Denying a port

```
firewall-cmd --zone=public --permanent --remove-port=12345/tcp
```

To add a custom port

```
firewall-cmd --zone=public --permanent --add-port=8096/tcp
```

Add a port range

```
firewall-cmd --zone=public --permanent --add-port=4990-4999/udp
```

Check if port is added

```
firewall-cmd --list-ports
```

Services are simply collections of ports with an associated name and description, the simplest way to add a port to a service would be to copy the xml file and change the definition/port number.

```
cp /usr/lib/firewalld/services/service.xml /etc/firewalld/services/example.xml
```

Then reload

```
firewall-cmd --reload && firewall-cmd --get-services
```

# Creating Your Own Zones

```
firewall-cmd --permanent --new-zone=my_zone  
firewall-cmd --reload  
firewall-cmd --zone=my_zone --add-service=ssh  
firewall-cmd --zone=my_zone --change-interface=eth0
```

Then add the zone to your `/etc/sysconfig/network-scripts/ifcfg-eth0`

```
## ZONE=my_zone
```

```
systemctl restart network  
systemctl restart firewallld
```

And check if it works

```
firewall-cmd --zone=my_zone --list-services
```

## Port Forwarding

Forward traffic coming from 80 to 12345

```
firewall-cmd --zone="public" --add-forward-port=port=80:proto=tcp:toport=12345
```

To forward a port to a different server:

Forwards traffic from local port 80 to port 8080 on a *remote server* located at the IP address: 123.456.78.9.

```
firewall-cmd --zone=public --add-masquerade  
firewall-cmd --zone="public" --add-forward-port=port=80:proto=tcp:toport=8080:toaddr=123.456.78.9
```

If you need to remove it

```
sudo firewall-cmd --zone=public --remove-masquerade
```



# Rich Rules

Allow all IPv4 traffic from host 192.168.0.14.

```
firewall-cmd --zone=public --add-rich-rule 'rule family="ipv4" source address=192.168.0.14 accept'
```

Deny IPv4 traffic over TCP from host 192.168.1.10 to port 22.

```
firewall-cmd --zone=public --add-rich-rule 'rule family="ipv4" source address="192.168.1.10" port port=22 protocol=tcp deny'
```

Allow IPv4 traffic over TCP from host 10.1.0.3 to port 80, and forward it locally to port 6532.

```
firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source address=10.1.0.3 forward-port port=80 protocol=tcp to-port=6532 accept'
```

Forward all IPv4 traffic on port 80 to port 8080 on host 172.31.4.2 (masquerade should be active on the zone).

```
firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 forward-port port=80 protocol=tcp to-port=8080 to-addr=172.31.4.2 masquerade accept'
```

To list your current Rich Rules:

```
firewall-cmd --list-rich-rules
```