Firewalls

Tables, Chains, and NAT

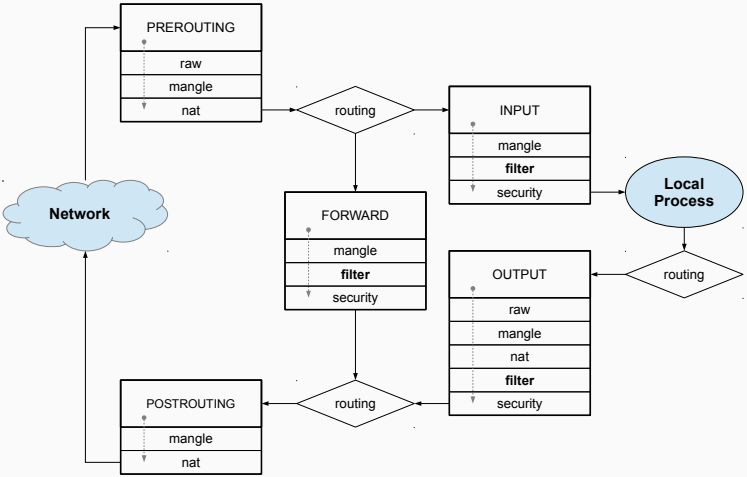
Miguel Frade & Francisco Santos

POLITÉCNICO DE LEIRIA
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Tables and Chains

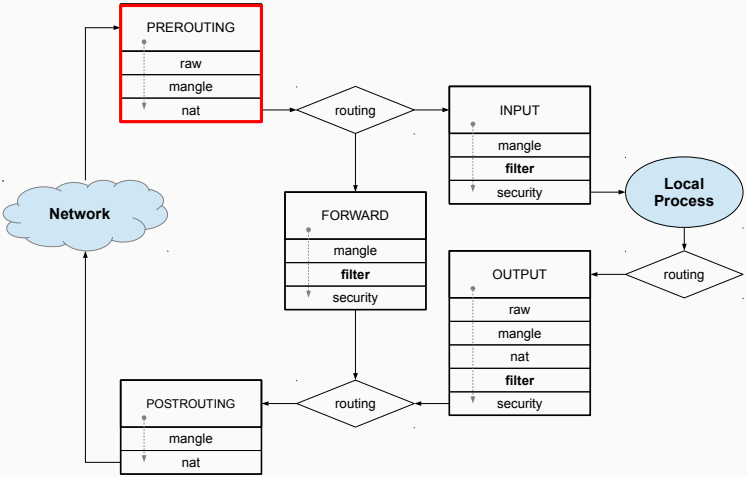
`iptables`

- packets journey across chains and tables



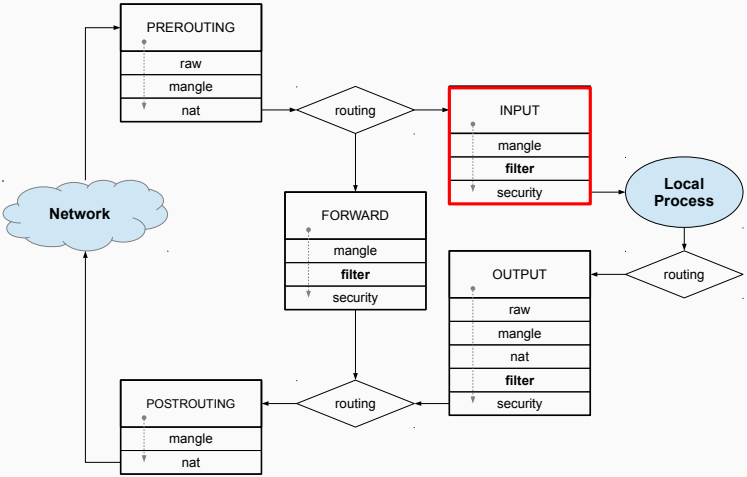
Default Chains

- `PREROUTING` – applied before packets are routed



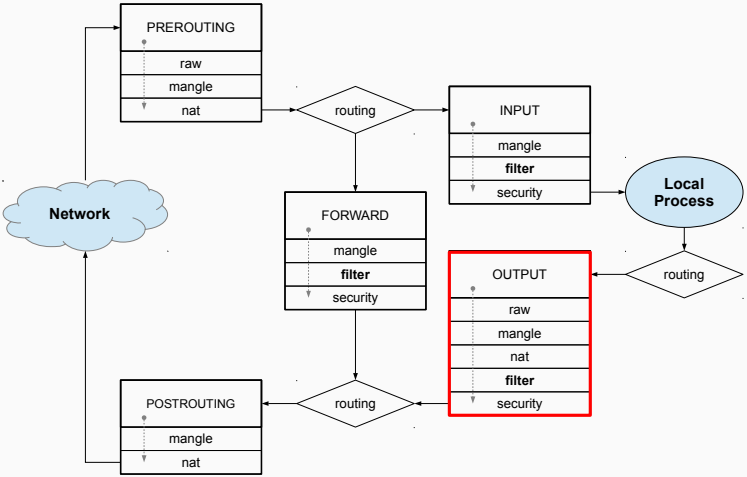
Default Chains

- `PREROUTING` – applied before packets are routed
- `INPUT` – applied for incoming packets into a process



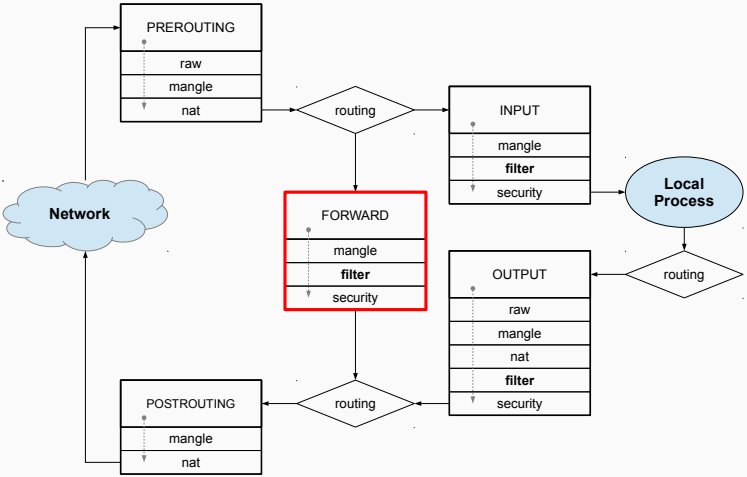
Default Chains

- `PREROUTING` – applied before packets are routed
- `INPUT` – applied for incoming packets into a process
- `OUTPUT` – applied for outgoing packets from a process



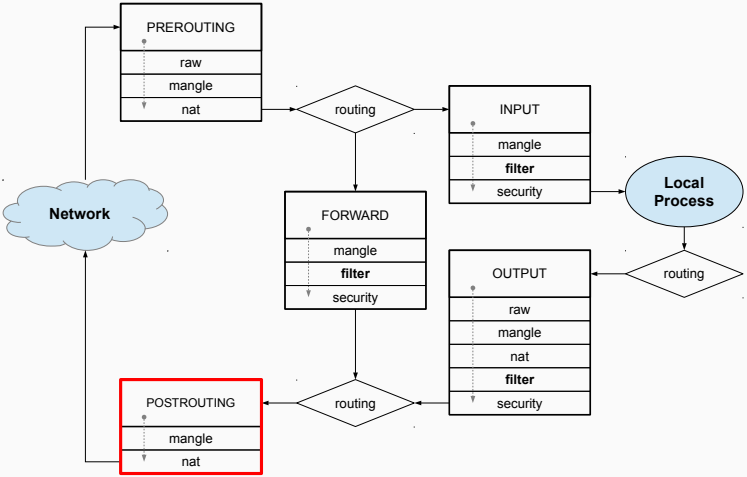
Default Chains

- `PREROUTING` – applied before packets are routed
- `INPUT` – applied for incoming packets into a process
- `OUTPUT` – applied for outgoing packets from a process
- `FORWARD` – applied for routed packets



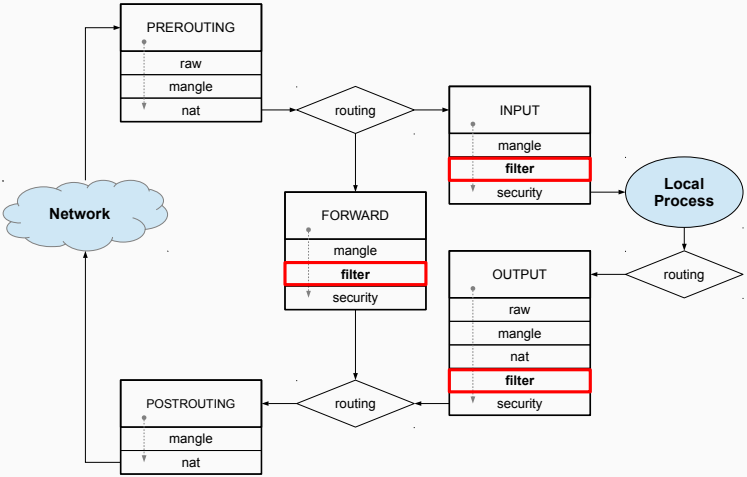
Default Chains

- `PREROUTING` – applied before packets are routed
- `INPUT` – applied for incoming packets into a process
- `OUTPUT` – applied for outgoing packets from a process
- `FORWARD` – applied for routed packets
- `POSTROUTING` – applied after packets are routed



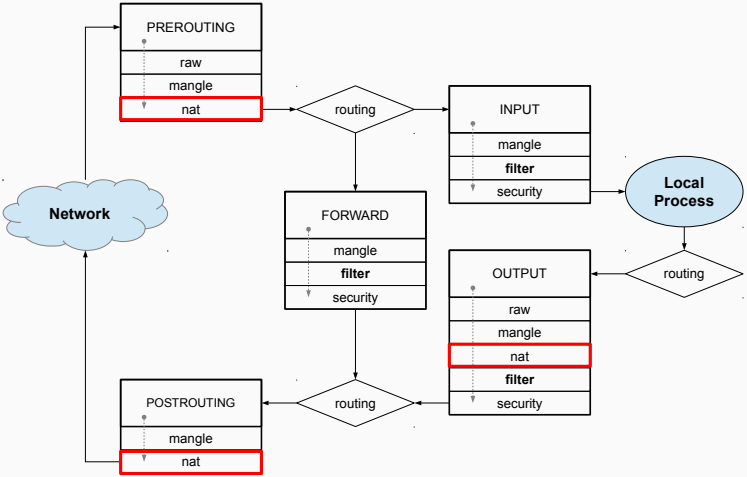
Tables

- `filter` – where the filter rules should be placed, it is the default table when the option `-t` is omitted: `iptables [-t <table>]`



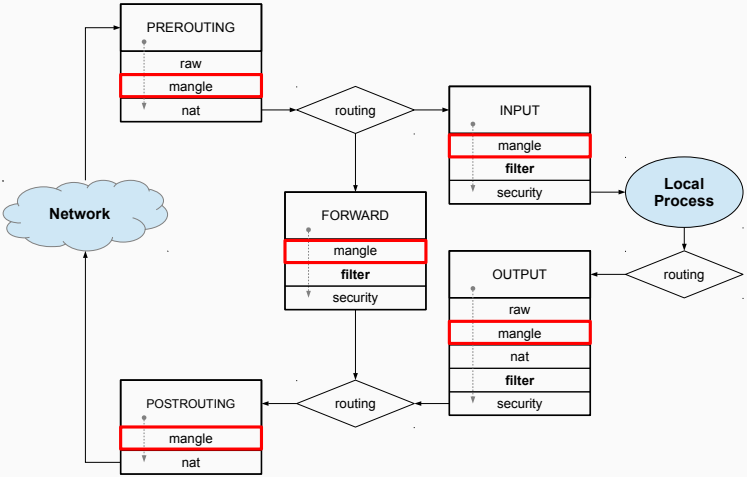
Tables

- `filter` – where the filter rules should be placed, it is the default table when the option `-t` is omitted: `iptables [-t <table>]`
- `nat` – to perform *Network Address Translation (NAT)*



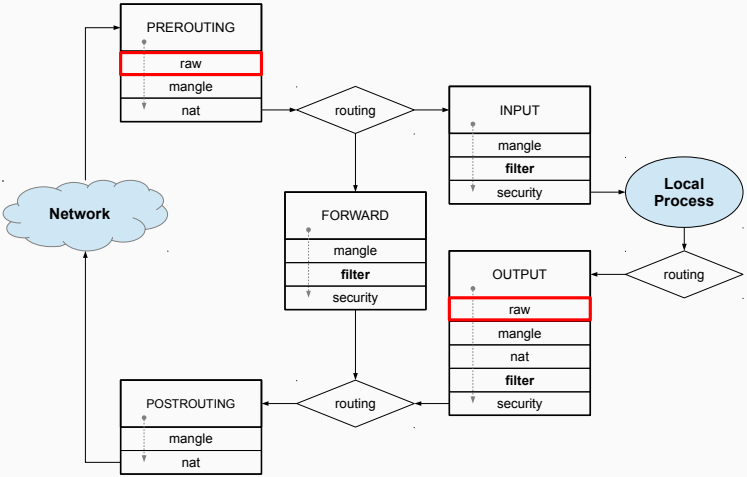
Tables

- `filter` – where the filter rules should be placed, it is the default table when the option `-t` is omitted: `iptables [-t <table>]`
- `nat` – to perform *Network Address Translation (NAT)*
- `mangle` – to make specialized changes to the packages, namely `TOS` and `TTL`



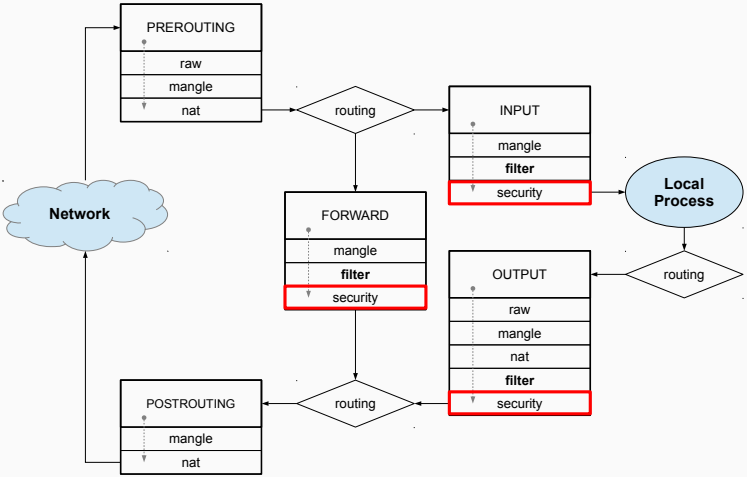
Tables

- `filter` – where the filter rules should be placed, it is the default table when the option `-t` is omitted: `iptables [-t <table>]`
- `nat` – to perform *Network Address Translation (NAT)*
- `mangle` – to make specialized changes to the packages, namely `TOS` and `TTL`
- `raw` – to configure exceptions to the connection tracking system



Tables

- `filter` – where the filter rules should be placed, it is the default table when the option `-t` is omitted: `iptables [-t <table>]`
- `nat` – to perform *Network Address Translation (NAT)*
- `mangle` – to make specialized changes to the packages, namely `TOS` and `TTL`
- `raw` – to configure exceptions to the connection tracking system
- `security` – for mandatory access control with S.E. Linux



Add rules

- `filter` table

```
# using the default table
$IPT -A OUTPUT ...

# or specifying the table
$IPT -t filter -A OUTPUT ...
```

- other tables

```
$IPT -t raw -A OUTPUT ...
$IPT -t mangle -A OUTPUT ...
$IPT -t nat -A OUTPUT ...
$IPT -t security -A OUTPUT ...
```

Add rules

- `filter` table

```
# using the default table
$IPT -A OUTPUT ...

# or specifying the table
$IPT -t filter -A OUTPUT ...
```

- other tables

```
$IPT -t raw -A OUTPUT ...
$IPT -t mangle -A OUTPUT ...
$IPT -t nat -A OUTPUT ...
$IPT -t security -A OUTPUT ...
```

Delete all rules

- `filter` table

```
# using the default table
$IPT -F

# or specifying the table
$IPT -t filter -F
```

- other tables

```
$IPT -t raw -F
$IPT -t mangle -F
$IPT -t nat -F
$IPT -t security -F
```

Add rules

- `filter` table

```
# using the default table
$IPT -A OUTPUT ...

# or specifying the table
$IPT -t filter -A OUTPUT ...
```

- other tables

```
$IPT -t raw -A OUTPUT ...
$IPT -t mangle -A OUTPUT ...
$IPT -t nat -A OUTPUT ...
$IPT -t security -A OUTPUT ...
```

Delete all rules

- `filter` table

```
# using the default table
$IPT -F

# or specifying the table
$IPT -t filter -F
```

- other tables

```
$IPT -t raw -F
$IPT -t mangle -F
$IPT -t nat -F
$IPT -t security -F
```

Delete custom chains

- `filter` table

```
# using the default table
$IPT -X  # chains

# or specifying the table
$IPT -t filter -X
```

- other tables

```
$IPT -t raw -X
$IPT -t mangle -X
$IPT -t nat -X
$IPT -t security -X
```

Network Address Translation

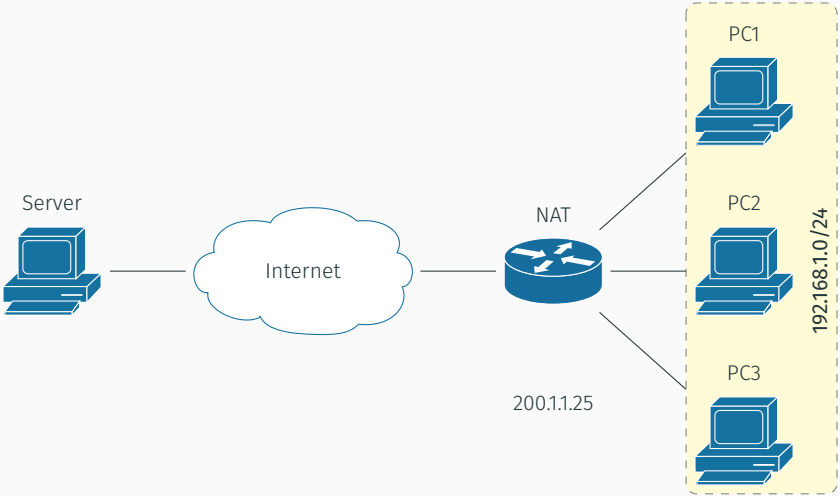
NETWORK ADDRESS TRANSLATION

Allows multiple computers to share the same IP

- changes the source and/or destination IP addresses
- recalculates the checksum of the packets
- two types of NAT
  - *Source Network Address Translation* (SNAT)
  - *Destination Network Address Translation* (DNAT)
  - these names are not universal and there are other definitions, for more info ▸ Wikipedia: NAT

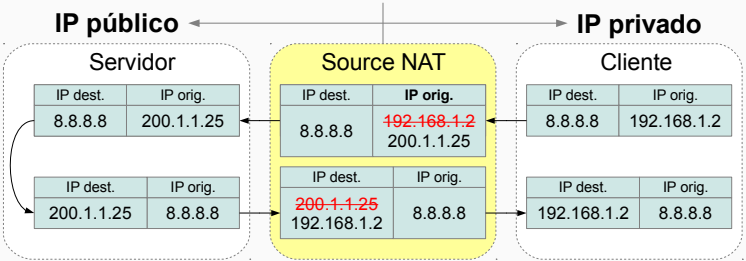
NAT

- private network with several PCs
- one public IP address shared



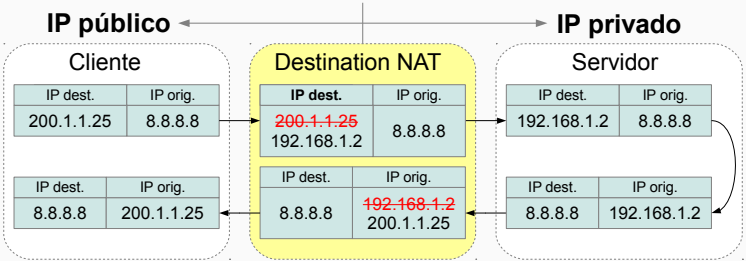
Source NAT

- the client PC has a private IP address
- client PC needs to connect to a server with a public IP address
- the router performs NAT by changing the source IP of the client with the router's public IP



Destination NAT

- the client PC has a public IP address
- client needs to connect to a server behind a private IP address
- the router performs NAT by changing the destination IP of the router with the server's private IP



Target `-j SNAT`

- only valid in the `NAT` table of the `POSTROUTING` chain
- allows to specify the source IP address that must be changed to
- supports the following options:
  - `--to-source ipaddr[-ipaddr][:port[-port]]`
    IP address to map source to, and optionally a port range can be specified if `-p tcp` or `-p udp` is present
  - `--random`
    for random port mapping
  - `--persistent`
    for persistent port mapping for the same client

Target `-j MASQUERADE`

- only valid in the `NAT` table of the `POSTROUTING` chain
- similar to `SNAT`
  - it should be used only when the client is configured with `DHCP`
  - if the client has a static IP address `-j SNAT` should be used (it's faster)

Target `-j MASQUERADE`

- only valid in the `NAT` table of the `POSTROUTING` chain
- similar to `SNAT`
  - it should be used only when the client is configured with `DHCP`
  - if the client has a static IP address `-j SNAT` should be used (it's faster)
- supports the following options:
  - `--to-ports <port>[-<port>]`
    port range to map to
  - `--random`
    randomize source port

Example of rules created by VirtualBox to allow NAT to its virtual machines

```
$IPT -A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -p tcp -j MASQUERADE
↪  --to-ports 1024-65535
$IPT -A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -p udp -j MASQUERADE
↪  --to-ports 1024-65535
```

Target `-j DNAT`

- only valid in the `NAT` table of the `PREROUTING` and `OUTPUT` chains
- allows to specify the destination IP address that must be changed to
- supports the following options:
  - `--to-destination [<ipaddr>[-<ipaddr>]][:port[-port]]`
    IP address to map destionation to, and optionally a port range can be specified if `-p tcp` or `-p udp` is present
  - `--random`
    for random port mapping
  - `--persistent`
    for persistent port mapping

Target `-j REDIRECT`

- only valid in the `NAT` table of the `PREROUTING` and `OUTPUT` chains
- redirects packets to a different TCP, or UDP port that are sent to the same computer that is running `iptables`
- can be used to implement transparent proxies
- supports the following options:
  - `--to-ports <port>[-<port>]`
    port range to map to
  - `--random`
    for random port mapping

Target `-j REDIRECT`

- only valid in the `NAT` table of the `PREROUTING` and `OUTPUT` chains
- redirects packets to a different TCP, or UDP port that are sent to the same computer that is running `iptables`
- can be used to implement transparent proxies
- supports the following options:
  - `--to-ports <port>[-<port>]`
    port range to map to
  - `--random`
    for random port mapping

Example

```
$IPT -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
$IPT -t nat -A PREROUTING -p udp --dport 80 -j REDIRECT --to-port 8080
```

Configure sNAT/MASQUERADE on a computer

- two network cards, *e. g.* `eth0` connected to Internet and `eth1` connected to the LAN
- enable packet forwarding in the kernel

```
Terminal
root@linux:~# sysctl -w net.ipv4.ip_forward=1
root@linux:~# sysctl -w net.ipv6.conf.all.forwarding=1
```

- then configure `iptables`

```
# configure source NAT with masquerade
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# Forward all packets LAN -> INTERNET
iptables -A FORWARD -i eth1 -o eth0 -m state --state NEW -j ACCEPT
# allow all established connections
iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Configure dNAT on a computer

- two network cards, *e. g.* `eth0` connected to Internet and `eth1` connected to the LAN
- enable packet forwarding in the kernel (see previous slide)
- then configure `iptables`

```
# configure destination NAT
PublicIP=200.1.1.25
PrivateIP=192.168.1.1
iptables -t nat -A PREROUTING --dst $PublicIP -p tcp --dport 22 -j DNAT --to-destination
↪  $PrivateIP

# Forward SSH packets Internet -> LAN
iptables -A FORWARD -i eth0 -o eth1 -p tcp --dport ssh -m state --state NEW -j ACCEPT
# allow all established connections
iptables -A FORWARD -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Exercise

EXERCISE

1. Redirect packets from port 2022 to 22 on your firewall

Questions?

- Online documentation

```
Terminal
user@linux:~$ iptables -j SNAT -h
user@linux:~$ iptables -j MASQUERADE -h
user@linux:~$ iptables -j DNAT -h
user@linux:~$ iptables -j REDIRECT -h
```