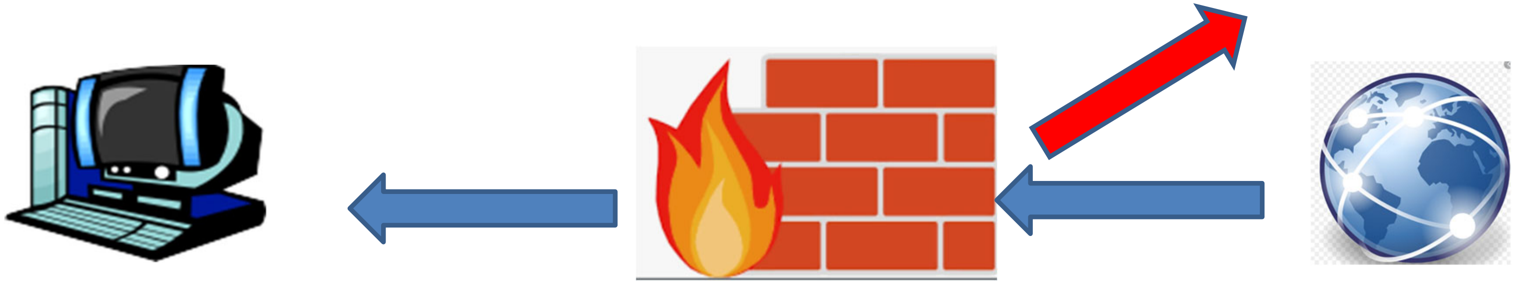
Security with IPTABLES

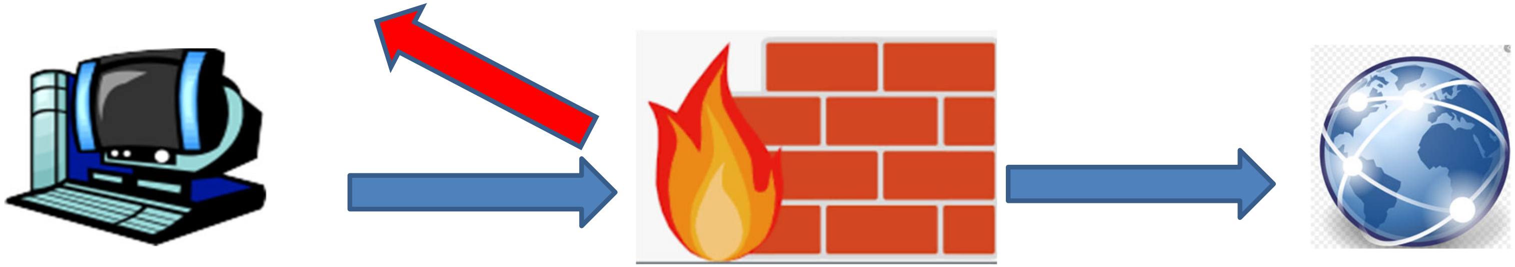
What is Packet Filtering? and Why Do I Need It?

- Blocking unwanted traffic from outside
  - By IP address (you may wish to trust specific hosts)
  - By destination port (allowing specific services, such as HTTP, but excluding all others)
  - By protocol type (e.g., disallowing all PINGs from outside)



What is Packet Filtering? and Why Do I Need It?

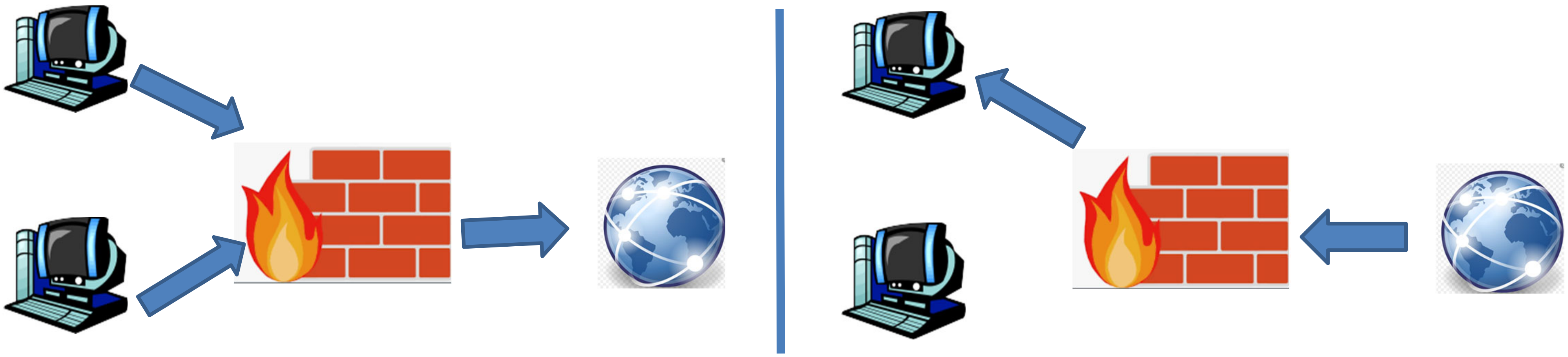
- Limiting access to Internet from certain hosts
  - By IP address (allow computer to access only specific printers, for example)
  - By destination port (allowing specific services, such as HTTP, but excluding all others)
  - Typically, political/management, not “security” per se



What is Packet Filtering? and Why Do I Need It?

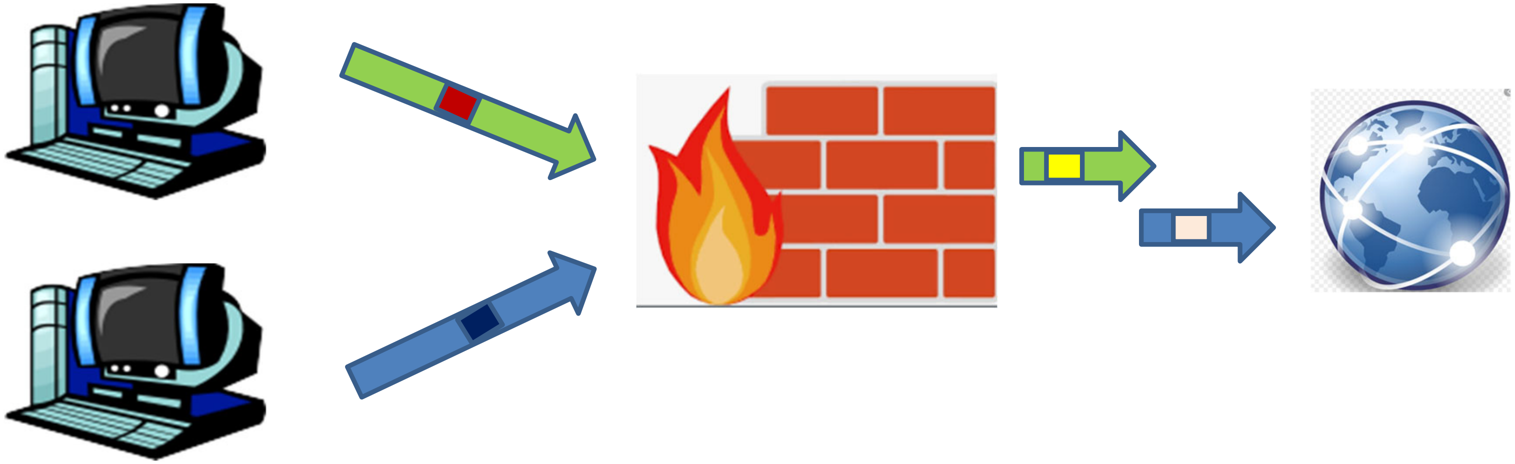
Network Address Translation (NAT):

- Sharing a single Internet address with multiple hosts from an internal LAN
- Redirecting specific inbound requests to selected internal hosts



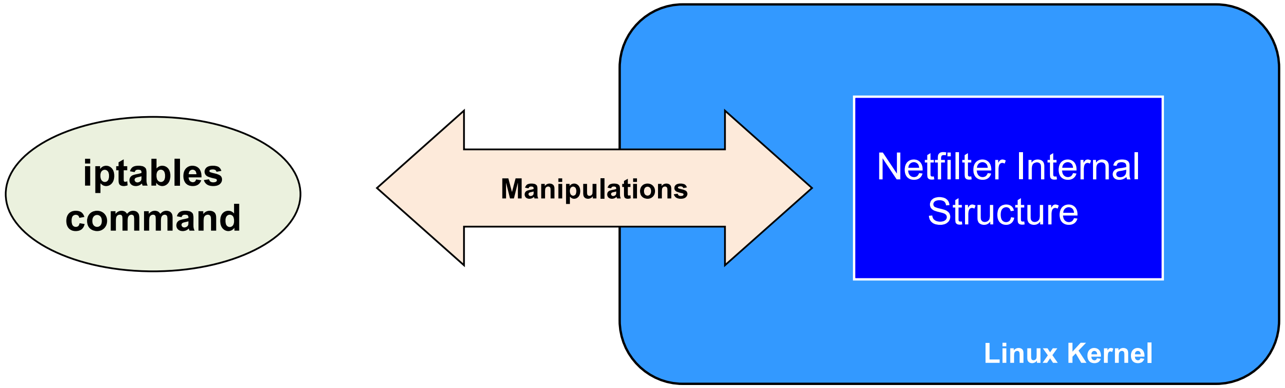
What is Packet Filtering? and Why Do I Need It?

- Rewriting attributes of packets as TTL …



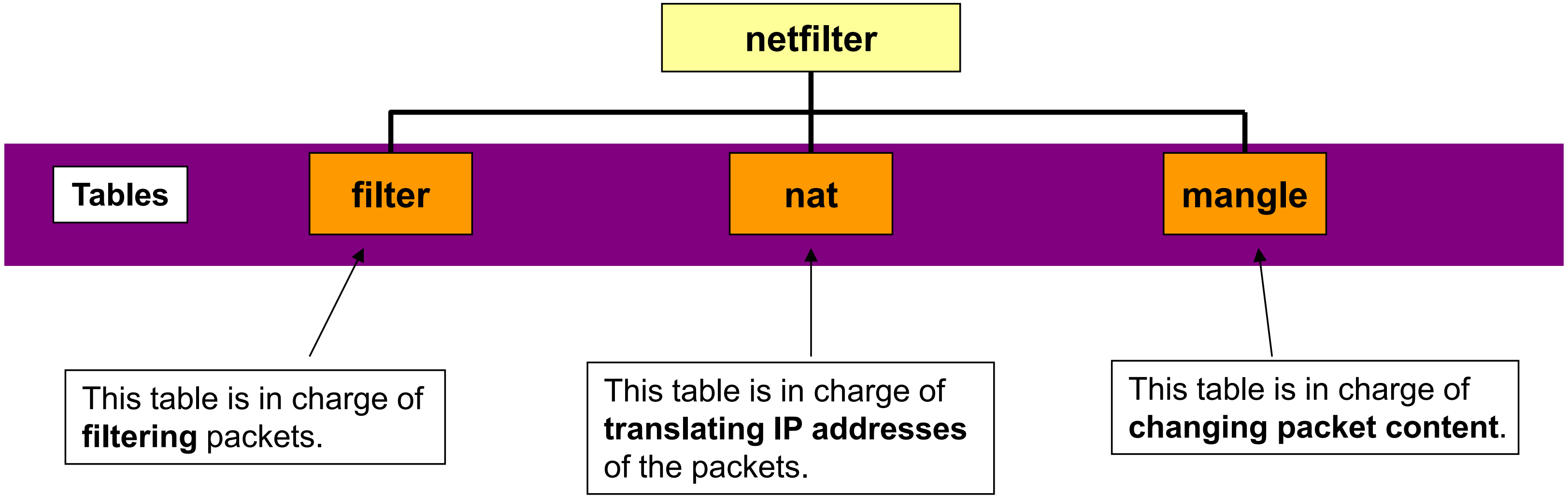
iptables

- **iptables** is a user-level program that controls the kernel-level network module called **netfilter**.



iptables – Tables and Chains

- Each function provided by the netfilter architecture is presented as a **table**.



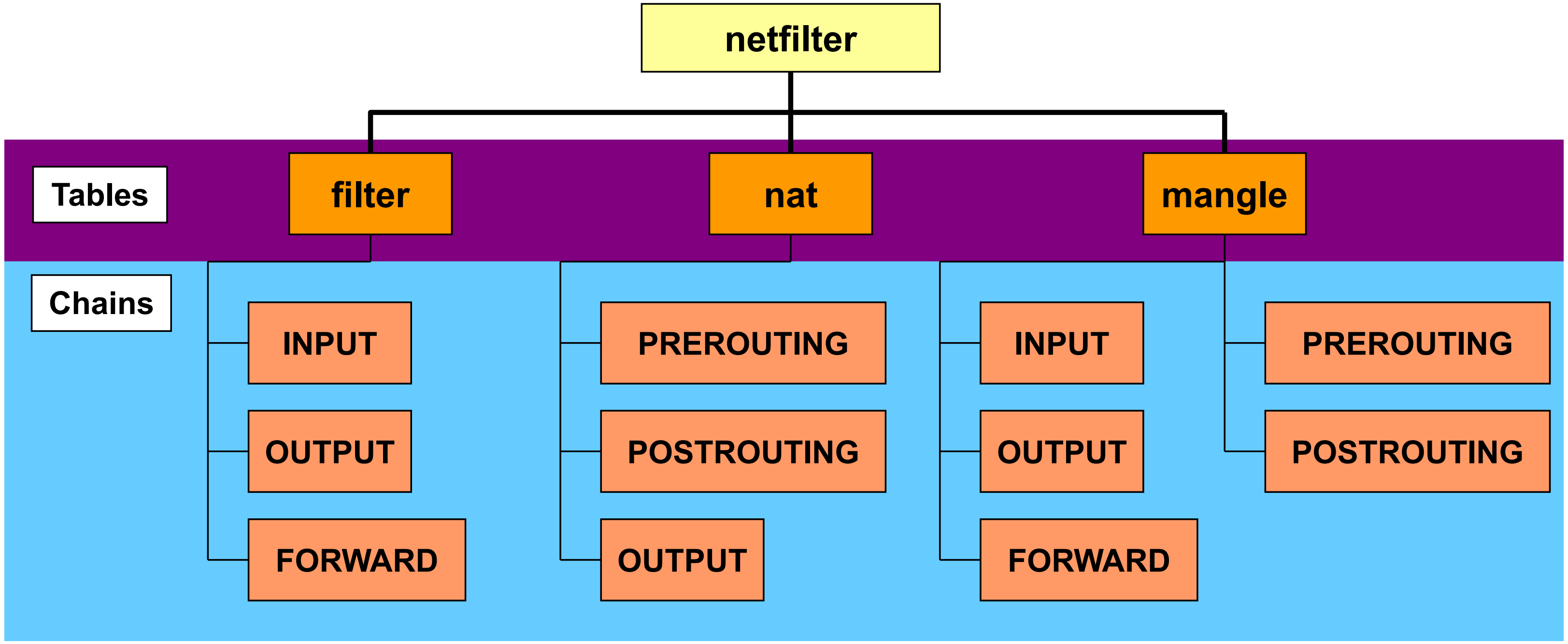
This table is in charge of **filtering** packets.

This table is in charge of **translating IP addresses** of the packets.

This table is in charge of **changing packet content**.

iptables – Tables and Chains

- Under each table, there are a set of **chains**.
  - Under each chain, you can assign a set of **rules**.

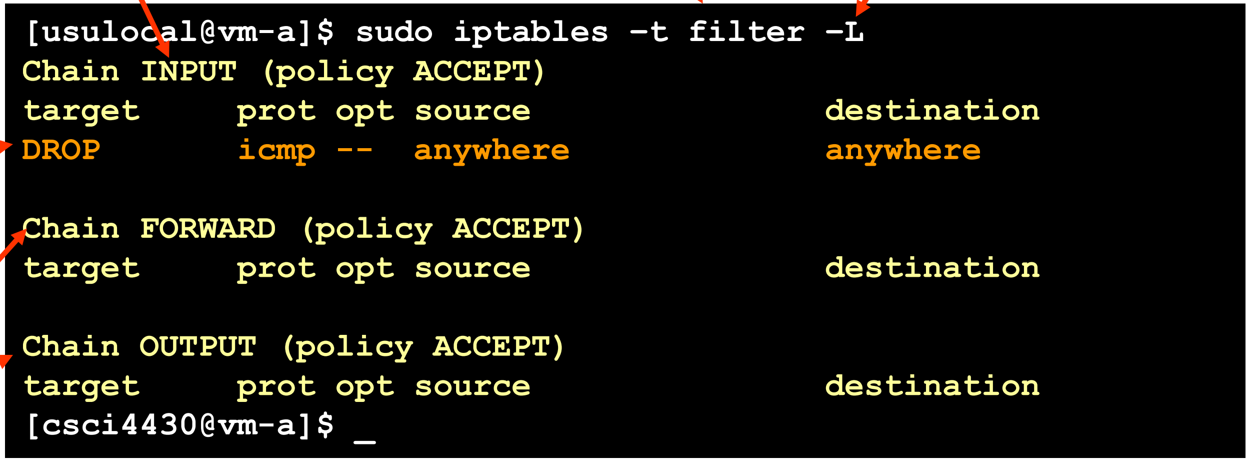


iptables – Rules

Chain name: **INPUT**

Table name: **filter**

The command: **list**



```
[usulocal@vm-a]$ sudo iptables -t filter -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[csci4430@vm-a]$ _
```

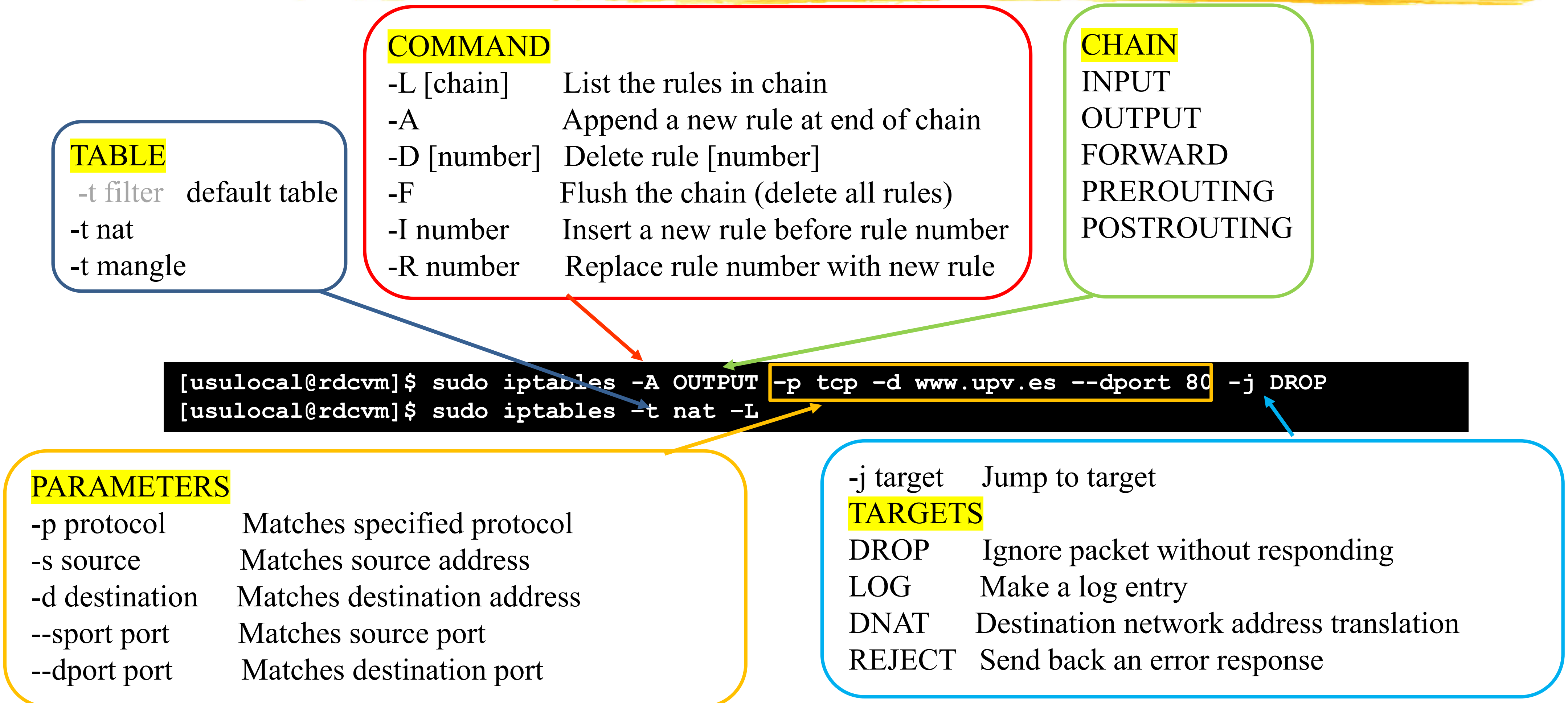
There is one rule set in the INPUT chain.

The other two chains.

The rule in the INPUT chain means:

When a packet with ICMP payload passes through the **INPUT hook**, **DROP** that packets, no matter it is **from anywhere** and **to anywhere**.

iptables – Rules



TABLE

-t filter	default table
-t nat	
-t mangle	

COMMAND

-L [chain]	List the rules in chain
-A	Append a new rule at end of chain
-D [number]	Delete rule [number]
-F	Flush the chain (delete all rules)
-I number	Insert a new rule before rule number
-R number	Replace rule number with new rule

CHAIN

INPUT
OUTPUT
FORWARD
PREROUTING
POSTROUTING

```
[usulocal@rdcvm]$ sudo iptables -A OUTPUT -p tcp -d www.upv.es --dport 80 -j DROP
[usulocal@rdcvm]$ sudo iptables -t nat -L
```

PARAMETERS

-p protocol	Matches specified protocol
-s source	Matches source address
-d destination	Matches destination address
--sport port	Matches source port
--dport port	Matches destination port

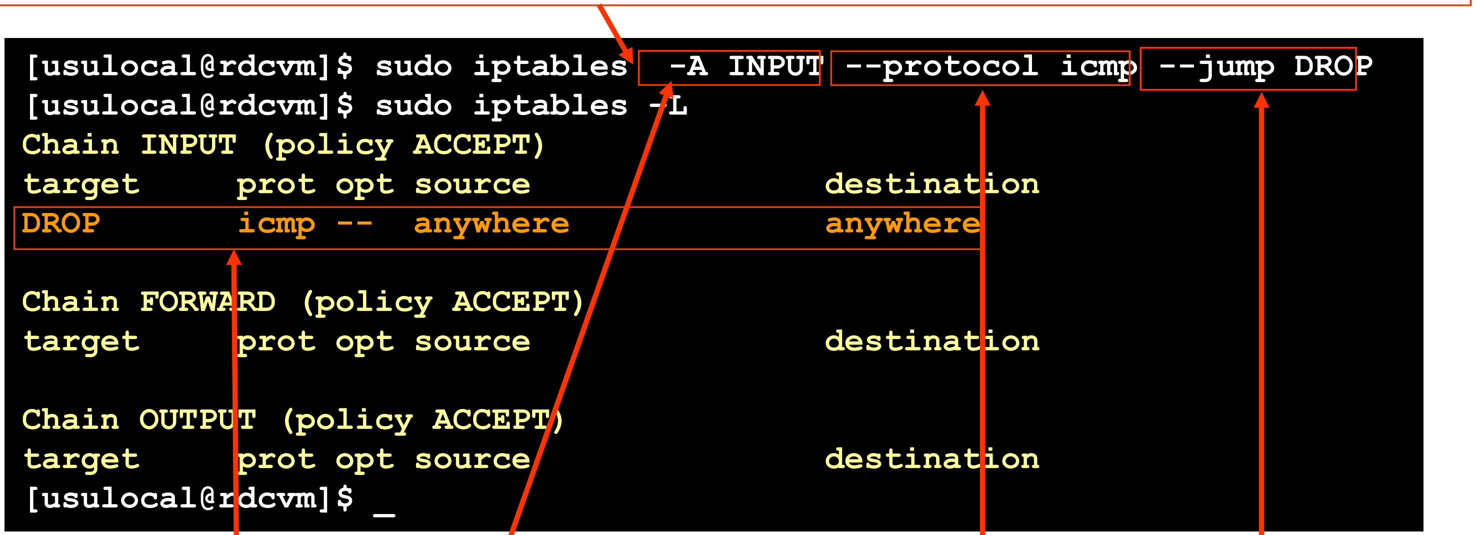
-j target Jump to target

TARGETS

DROP	Ignore packet without responding
LOG	Make a log entry
DNAT	Destination network address translation
REJECT	Send back an error response

iptables – Rules

Table **filter** is the default table (`-t filter`), therefore is not necessary to include it in the command line



```
[usulocal@rdcvm]$ sudo iptables  -A INPUT --protocol icmp --jump DROP
[usulocal@rdcvm]$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[usulocal@rdcvm]$ _
```

Add a new rule to the INPUT chain.

This entry shows that a new rule is added to the INPUT chain of the filter table successfully.

The **protocol** of the packets in which this rule is interested is **ICMP**.

If a packet
(1) passes through the INPUT hook, and
(2) is an ICMP packet,

then the packet **jumps to the target DROP** – **to discard the packet**.