•Implicit Deny

All traffic is implicitly blocked unless you explicitly allow it

•Implicit Allow

All traffic is implicitly allowed unless you explicitly block it

•Rules may be implicit or explicit:

Implicit

oFirewall has this rule be default

oNo logging occurs

Explicit

oThis rule must be created

oLogging occurs

•Implicit Allow Policy to accept all traffic by default

•Add explicit rules allowing certain traffic to enter

•Add explicit deny at bottom of rule chain to reject all other traffic that isn’t explicitly allowed in the prior rules

**We need to ensure that the firewall doesn’t accept new inbound traffic and only accepts traffic in response to an established connection**

•Stateless (Static):

Evaluates contents of each packet but doesn’t keep track of the connection state

Similar to an Access Control List (ACL)

Low overhead / High throughput

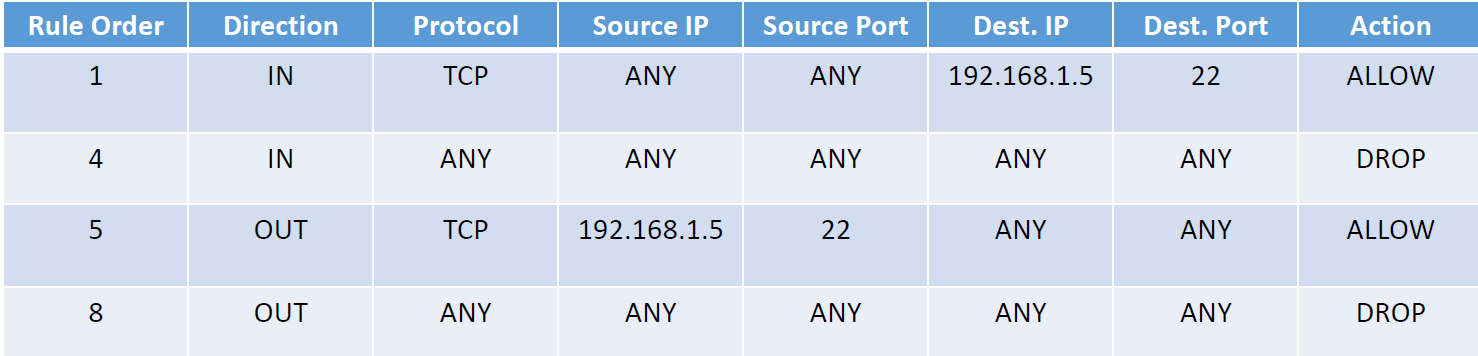
Inexpensive

•Stateful (Dynamic):

Keeps track of connection state in order to make decisions

Higher overhead / Lower throughput

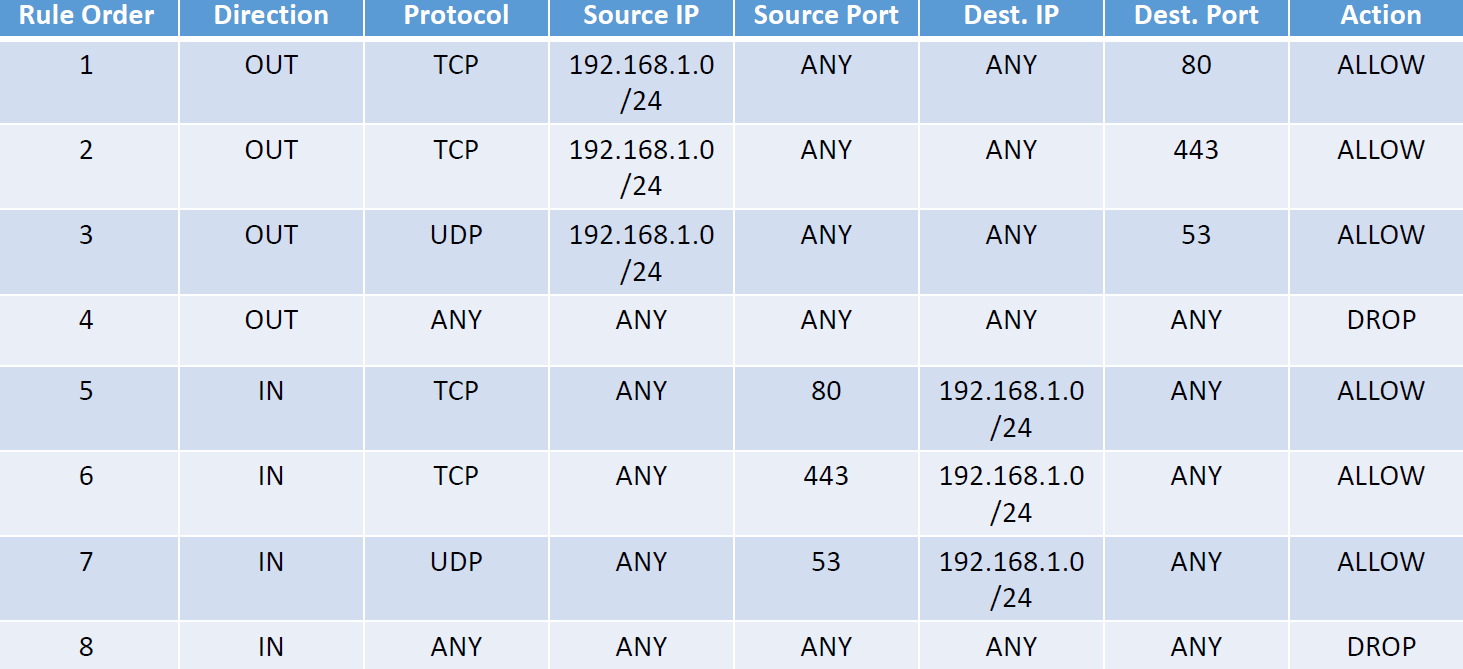
More costly



The SSH Server at 192.168.1.5:

Accepts inbound connections on its port 22 from any IP address and port on the Internet

Allows outbound communications emanating from port 22 to any IP address and port on the Internet



•All clients on the subnet 192.168.1.0/24:

Can connect to any web server on the Internet via ports 80 and 443

Can query any DNS server on the Internet via port 53

Can receive replies from any server on the Internet that uses ports 80 and 443

Can receive responses from any DNS server on the Internet that uses port 53

•Hierarchy is:

Tables (lower case name)

Chains (upper case name)

Rules

**•4 Tables:**

**Filter Table**

**O Default firewall table you will use the most to filter packets**

•Filter Table has three chains:

INPUT

O Packets coming in

OUTPUT

O Packets going out

FORWARD

O Packets routed through two network interfaces on the local server. Example: eth0 to eth1

**NAT Table**

**O For routing of packets**

**Mangle Table**

**O Special packet alteration**

**Raw Table**

**O For configuration exceptions**

•Rules have actions.

•Main actions are:

**ACCEPT**

O Packet is accepted through the firewall into the system.

**REJECT**

O Packet is rejected and ICMP or TCP response is returned to sender.

**DROP**

O Packet is rejected and no response is returned.

**•Create firewall rules that:**

**•Allow only established connections on ports 443, 80, and 53 for inbound traffic**

**•Allow new and established connections on ports 443, 80, and 53 for outbound traffic**

• **sudo iptables -t filter --list**

Shows Filter table and three chains.

They are all blank which means there are no rules set by default.

•We will focus on the Filter table but to see the rules for other tables use:

**sudo iptables –t nat --list**

**sudo iptables –t mangle --list**

**sudo iptables –t raw --list**

•An easier way to see the rules for the default filter table (which we will be using) is with:

**sudo iptables -L**

•Let’s now set an explicit DROP for all incoming traffic

•**sudo iptables –A INPUT -ieth0 -j DROP**



Allow inbound and outbound traffic for:

oTCP ports 80, 443

oUDP port 53

•**sudo iptables -A OUTPUT -o eth0 -p tcp –dport 443 -m conntrack –ctstate NEW,ESTABLISHED -j ACCEPT**

•**sudo iptables -A OUTPUT -o eth0 -p tcp –dport 80 -m conntrack —ctstate NEW,ESTABLISHED -j ACCEPT**

•**sudo iptables -A OUTPUT -o eth0 -p udp--dport53 –m conntrack –ctstate NEW,ESTABLISHED -j ACCEPT**

•**sudo iptables-A INPUT -ieth0 -p tcp --sport 443 -m conntrack –ctstate ESTABLISHED -j ACCEPT**

•**sudo iptables -A INPUT -ieth0 -p tcp --sport 80 -m conntrack –ctstate ESTABLISHED -j ACCEPT**

•**sudo iptables -A INPUT -ieth0 -p udp --sport 53 –m conntrack –ctstate ESTABLISHED -j ACCEPT**



**Could not work, because the first command – drop all ip addresses**

•Delete the Explicit DROP and add it again at the bottom of the INPUT chain.

**sudo iptables -D INPUT 1**

**sudo iptables -I INPUT 4 –ieth0 -j DROP**





**•Instead of adding a DROP rule at the end of the chain, you could change the default policy from IMPLICIT ACCEPT to DROP**

**sudo iptables –P INPUT ACCEPT**

•When you reboot your rules are not persistent.

•Save your rules in the Documents folder:

**sudo iptables -save > iptables.dump**

•Now let’s flush (delete) all of the rules.

**sudo iptables –F**

•Now let’s reload the rules:

**sudo iptables -restore < iptables.dump**