

# Network Security Systems

A brief overview of some basic network security systems.

## Iptables and Firewalls

From Wikipedia:

*A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.*

*A firewall typically establishes a barrier between a trusted network and an untrusted network.*

## Firewall examples

Firewalls are one of the most basic types of network security. AWS calls them Security Groups and they must be applied to every instance and network related entity.

ExampleSecurityGroup:

Type: 'AWS::EC2::SecurityGroup'

Properties:

SecurityGroupIngress:

- IpProtocol: tcp  
FromPort: '22'  
ToPort: '22'  
CidrIp: 130.108.0.0/16
- IpProtocol: tcp  
FromPort: '1025'  
ToPort: '2048'  
CidrIp: 130.108.0.0/16
- IpProtocol: -1  
FromPort: '-1'  
ToPort: '-1'  
CidrIp: 10.0.0.0/16

## Firewall Types

### ▶ Packet filters

- ▶ **Stateless** filters check basic information like port number, IP address, protocol, etc. of a packet to take action.
- ▶ **Stateful** filters maintain records of connections passing through, can determine if a packet is the start of a new connection or a part of an existing connection. In iptables the four trackable states are **NEW**, **ESTABLISHED**, **RELATED** and **INVALID**.

### ▶ Application Layer

- ▶ works more like a proxy, can “understand” applications and protocols.
- ▶ frequently inspects contents of traffic.

## Firewall Policies

- ▶ **Blacklist** approach (default-allow)
  - ▶ All packets are allowed through except those that fit the defined blacklist.
  - ▶ Pros: flexible, less likely to impact services.
  - ▶ Cons: unexpected malicious traffic makes it through.
  - ▶ Lab 3: Task 3
- ▶ **Whitelist** approach (default-deny)
  - ▶ All packets are dropped or rejected unless specifically allowed.
  - ▶ Pros: safer network
  - ▶ Cons: unexpected important traffic gets dropped or rejected.
  - ▶ Lab 3: Task 4. . .

## Firewall Rules

Firewalls take one of three actions on a packet.

- ▶ *Allow* lets the packet pass through.
- ▶ *Drop* prevents the packet from passing but is silent about it.
- ▶ *Reject* prevents the packet from passing but notifies the sender that it is being blocked.

iptables specifically has multiple chains of rules that apply, here are the 5 predetermined ones:

- ▶ PREROUTING are rules applied to all packets before any routing decisions are made.
- ▶ INPUT are rules applied on packets that will be delivered locally.
- ▶ FORWARD are rules applying to packets that have been routed but are not for locally delivery
- ▶ OUTPUT are packets sent by the local machine.
- ▶ POSTROUTING again are all packets but after they have been through all previous relevant chains.

## iptables tables

Each of the chains defined previously can be applied to different tables depending on intent for the packet.

For packets that are destined to be filtered (allowed through or dropped) rules go in the `filter` table.

For packets destined to be altered there is a `nat` table.

For custom packet manipulation there is a `mangle` table.

## NAT (for l337 gamers and programmers)

NAT stands for Network Address Translation

This is what allows your non-routable private network

192.168.1.0/24 access to the internet.

Lets say you want to make a connection to google.com:80

Your computer

Router

```
=====
|           |           |           |
| port 31746 |=====>|           |
|           |           |           |
=====
```



Router		www.google.com
=====		=====
port 21283	====>	port 80
=====		=====

Router		www.google.com
=====		=====
port 21283	<====	port 80
=====		=====

## Open NAT

All traffic coming in to your Router on a given port gets sent to your computer.

Your computer		Router	
=====		=====	www.google.com:80
			www.google.com:443
port 31746	<===	port 21283	<=== serverfault.com:80
			fbi.gov:32188
=====		=====	botnet.cn:11288

## Moderate NAT

Only traffic from a specified *host* gets sent back through to your PC.

Your computer

Router

=====

=====

|                    |

|                    |

| port 31746 |

|<====

| port 21283 |

|<====

|                    |

|                    |

=====

=====

www.google.com:80

www.google.com:443

x (rejected) server1

x (rejected) fbi.gov

x (rejected) botnet

## Strict NAT

Only responses from the requested *host* **and** *port* are sent back to your pc.

Your computer

Router

=====

=====

www.google.com:80

| | | |  
| port 31746 |<===| port 21283|<===

x (rejected) www.google.com:80

x (rejected) server1.example.com:80

| | | |

x (rejected) fbi.gov:80

=====

=====

x (rejected) botnet.com:80

## IPtables info

iptables guide iptables man page cool NAT info I used

## IPtables notes for lab 3

- ▶ iptables do not persist through a reboot (so if you lock yourself out you can reboot to get back in)
- ▶ iptables can be annoying
- ▶ dropping all inbound connections (without any exceptions) will break everything