Task 1

- Create a firewall on your computer that prevents ping from working
- Capture on both computers using tcpdump; check WHY ping doesn't work
- Try different chains, 5 results:
  - INPUT on A: A ping B
  - INPUT on A: B ping A
  - OUTPUT on A: A ping B
  - OUTPUT on A: B ping A
  - FORWARD on A: anyone ping

Task 2

- Create a firewall that will drop TCP packets destined to a specific computer (your neighbours)

Task 3

- Create an internet with two subnets: on one subnet is a single PC1; and on the other subnet is two PCs (PC2 and PC3).
- PC1 should run a web server and SSH server.
- Create a firewall on the router that allows the following:
  - Any computer can connect to the web server on PC1;
  - Only PC2 can connect to the SSH server on PC1;
  - No computers can connect to any other servers (e.g. FTP, Email) on PC1.
  - PC1 can access servers on PC2 and PC3

Default Policy: DROP

- *Any computer can connect to the web server on PC1;*

```
-d PC1 -p tcp --dport 80 -j ACCEPT
-s PC1 -p tcp --sport 80 -j ACCEPT
```

Allow request to server and response from server

- *Only PC2 can connect to the SSH server on PC1;*

```
-s PC2 -d PC1 -p tcp --dport 22 -j ACCEPT
-d PC2 -s PC1 -p tcp --sport 22 -j ACCEPT
```

- *No computers can connect to any other servers (e.g. FTP, Email) on PC1.*
- *PC1 can access servers on PC2 and PC3*

```
-s PC1 -p tcp --dport 1:1024 -j ACCEPT
-d PC1 -p tcp --sport 1:1024 -j ACCEPT
```

Assume servers use only well-known ports 1 to 1024

Default Policy: DROP

- *Any computer can connect to the web server on PC1;*

```
-d PC1 -p tcp --dport 80 -j ACCEPT
-s PC1 -p tcp --sport 80 -j ACCEPT
```

- *Only PC2 can connect to the SSH server on PC1;*

```
-s PC2 -d PC1 -p tcp --dport 22 -j ACCEPT
-d PC2 -s PC1 -p tcp --sport 22 -j ACCEPT
```

- *No computers can connect to any other servers (e.g. FTP, Email) on PC1.*
- *PC1 can access servers on PC2 and PC3*

```
-s PC1 -p tcp --dport 1:1024 -j ACCEPT
-d PC1 -p tcp --sport 1:1024 -
```

PROBLEM
PC3 client uses port 234 to connect to PC1 SSH server on port 22

Using Port Ranges

- Although it is common for servers to use well-known ports (1-1024) and clients to use dynamic ports (>40,000), there is nothing to stop a malicious client to use any port number

Default Policy: DROP and SPI

- *Enable Stateful Packet Inspection:*

```
-m state --state ESTABLISHED,RELATED -j ACCEPT
```

Automatically accepts packets that are part of existing connection.

- *Any computer can connect to the web server on P*

```
-d PC1 -p tcp --dport 80 -j ACCEPT
```

- *Only PC2 can connect to the SSH server on PC1*

```
-s PC2 -d PC1 -p tcp --dport 22 -j ACCEPT
```

- *No computers can connect to any other servers (e.g. FTP, Email) on PC1.*
- *PC1 can access servers on PC2 and PC3*

```
-s PC1 -p tcp --dport 1:1024 -j ACC
```

Only need rules to accept the first request packet. SPI will handle all others