Target Users: Second/Third year students

What I’m teaching: Chains, TCP, UDP, HTTP, SSH, syntax of IPTables, Ports of protocols, Connection States

Common commands to teach:

Allow all web traffic: sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT

Drop all incoming packets: sudo iptables -A INPUT -j DROP

Accept incoming connections by default: iptables --policy INPUT ACCEPT

**Accept** – Allow the connection.

**Drop** – Drop the connection, act like it never happened. This is best if you don’t want the source to realize your system exists.

**Reject** – Don’t allow the connection but send back an error. This is best if you don’t want a source to connect to your system, but you want them to know that your firewall blocked them.

Block all connections from an IP:

iptables -A INPUT -s 10.10.10.10 -j DROP

Block all connections in a range of IP:

iptables -A INPUT -s 10.10.10.0/24 -j DROP

Block all SSH connections from an IP:

iptables -A INPUT -p tcp --dport ssh -s 10.10.10.10 -j DROP

As we mentioned earlier, a lot of protocols are going to require two-way communication. For example, if you want to allow SSH connections to your system, the input and output chains are going to need a rule added to them. But, what if you only want SSH coming into your system to be allowed? Won’t adding a rule to the output chain also allow outgoing SSH attempts?

That’s where connection states come in, which give you the capability you’d need to allow two way communication but only allow one way connections to be established. Take a look at this example, where SSH connections FROM 10.10.10.10 are permitted, but SSH connections TO 10.10.10.10 are not. However, the system is permitted to send back information over SSH as long as the session has already been established, which makes SSH communication possible between these two hosts.

iptables -A INPUT -p tcp --dport ssh -s 10.10.10.10 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -p tcp --sport 22 -d 10.10.10.10 -m state --state ESTABLISHED -j ACCEPT

**Level Stucture:**

**First Level:**

Basic Overview of Input and Output table and what packet is

To win: Do a default DROP on all incoming traffic

What’s taught: how to change default policy, what the tables are, what the purpose of IP tables is, what a packet is

**Second Level:**



Basic overview over IP Addresses (enough to know firewalls)

To win: Block all connections from one IP address, Allow the other

What’s taught: Basic overview of what an IP address is and how its structures etc.

**Third Level:**



Introduce TCP protocol.

To win: Allow connections on TCP.

What’s taught: Basics of TCP

**Fourth Level:**



Introduce HTTP and the concept of ports

To win: Allow HTTP traffic in and out of the network

What’s taught: HTTP and ports

**Fifth Level**:



Introduce SSH and why it needs two-way communication to work

To win: Allow SSH from Charlie

What’s taught: SSH

**Sixth Level**:



Introduce UDP and why its used even though it’s less secure

To win: Block HTTP from Bob and allow UDP in and out for Charlie

What’s taught: UDP and its usage

**Seventh Level**:



Combination of TCP, HTTP and SSH.

To win: Allow TCP to and from Charlie, allow Bob to SSH into Alice, and allow all HTTP in and out of Alice

What’s taught: Revision of protocols

**Eighth Level**:



How to delete a rule

To win: Delete one rule to allow Alice to send HTTP packets to Charlie

What’s taught: Rule deletion

* In all of these, there will be different variations of default policies. The tables will always be visible e.t.c.