Operating System Security

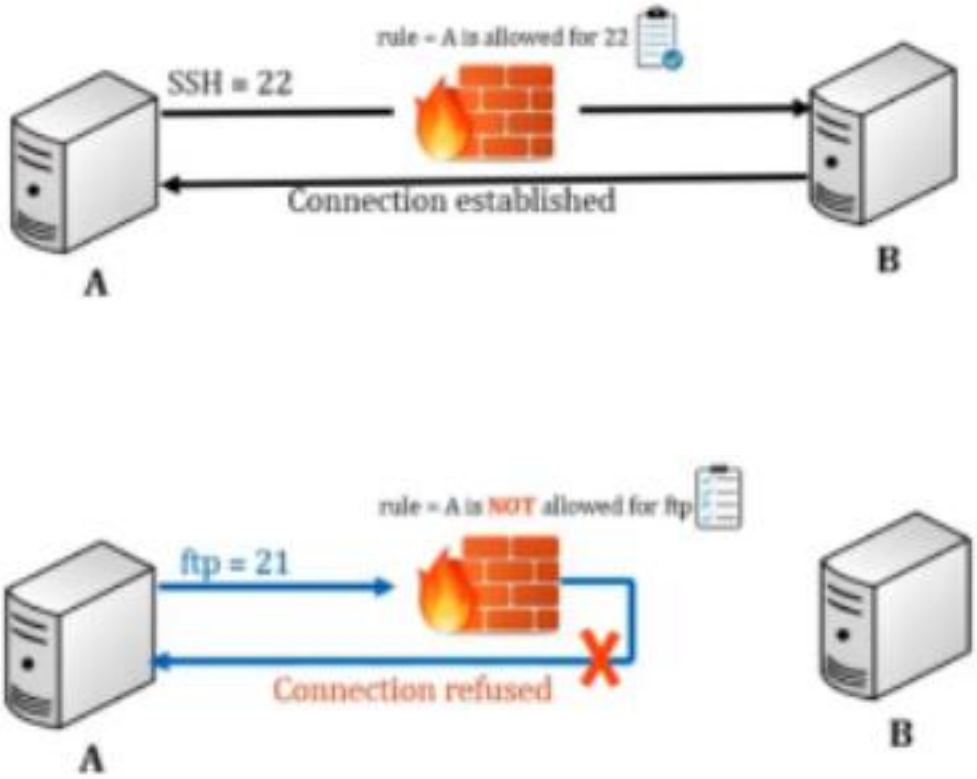
#07 - Securing Linux System Network

Herman Kabetta

Introduction to Firewall

- What is Firewall
  - A wall that prevents the spread of fire
  - When data moves in and out of a server its packet information is tested against the firewall rules to see if it should be allowed or not
  - In simple words, a firewall is like a watchman, a bouncer, or a shield that has a set of rules given and based on that rule they decide who can enter and leave
  - There are 2 type of firewalls in IT
    - Software = Runs on operating system
    - Hardware = A dedicated appliance with firewall software

Introduction to Firewall



Firewall (iptables)

- There are 2 tools to manage firewall in most of the Linux distributions
  - iptables = For older Linux versions but still widely used
  - firewalld = For newer versions like 7 and up

- You can run one or the other
  - In this lecture we will work with **iptables** to manage firewall

  - Before working with iptables make sure firewalld is not running and disable it
    - `service OR systemctl stop firewalld` = To stop the service
    - `systemctl disable firewalld` = To prevent from starting at boot time
    - `systemctl mask firewalld` = To prevent it from running by other programs

  - Now check if you have iptables-services package installed
    - `rpm -qa | grep iptables-services`
    - `yum install iptables-services` - *If not installed then*

  - Start the service
    - `systemctl start iptables`
    - `systemctl enable iptables`

  - To check the iptables rules
    - `iptables -L`

  - To flush iptables.
    - `iptables -F`

Firewall (iptables)

- The function of iptables tool is packet filtering
- The packet filtering mechanism is organized into three different kinds of structures: **tables,** chains and targets

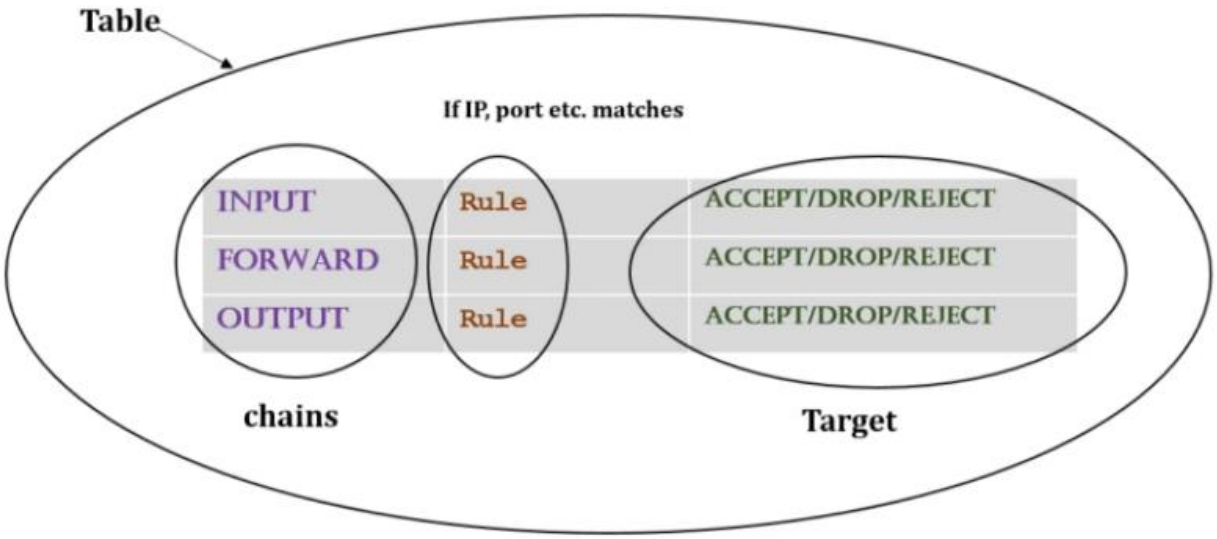
1. **`tables`** = table is something that allows you to process packets in specific ways. There are 4 different types of tables, **filter**, mangle, nat and raw

2. **`chains`** = The chains are attached to tables, These chains allow you to inspect traffic at various points. There are 3 main chains used in iptables
   - INPUT = incoming traffic
   - FORWARD = going to a router, from one device to another
   - OUTPUT = outgoing traffic
     - chains allow you to filter traffic by adding rules to them
     - Rule = if traffic is coming from `192.168.1.35` then go to defined target

3. **`targets`** = target decides the fate of a packet, such as allowing or rejecting it. There are 3 different type of targets
   - ACCEPT = connection accepted
   - REJECT = Send reject response
   - DROP = drop connection without sending any response

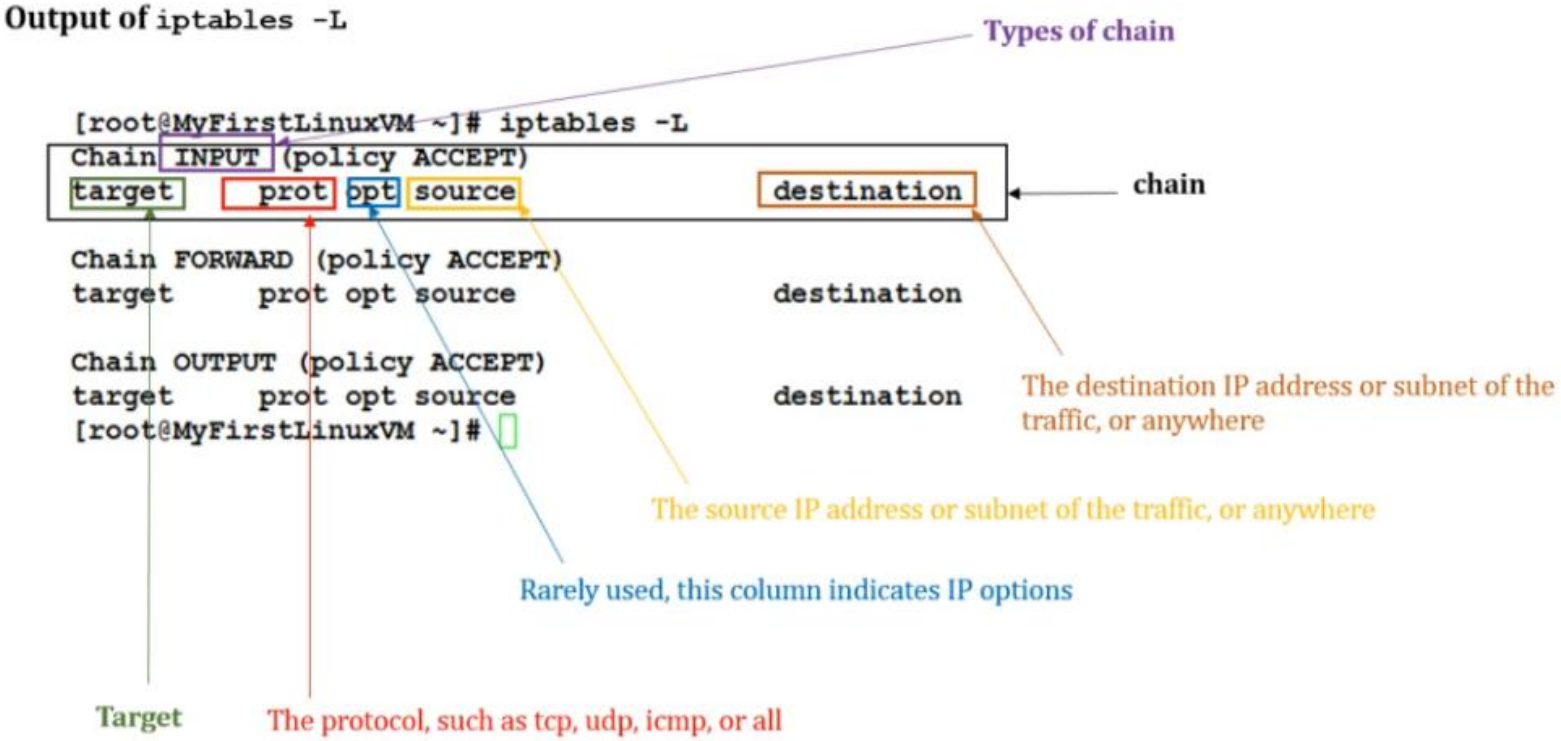
Firewall (iptables)

Let's draw it out:



- To check the iptables rules
  - `iptables -L`

Firewall (iptables)



Firewall (iptables)

- Drop all traffic coming from a specific IP (192.168.0.25)
  - `iptables -A INPUT -s 192.168.0.25 -j DROP`
- Drop all traffic coming from a range of IPs (192.168.0.0)
  - `iptables -A INPUT -s 192.168.0.0/24 -j DROP`
- List all rules in a table by line numbers
  - `iptables -L --line-numbers`
- Delete a specific rule by line number
  - `iptables -D INPUT 1`
- To flush the entire chain
  - `iptables -F`
- To block a specific protocol with rejection (e.g. ICMP)
  - `iptables -A INPUT -p icmp -j REJECT`
- To block a specific protocol without rejection (e.g. ICMP)
  - `iptables -A INPUT -p icmp -j DROP`
- To block a specific port # (e.g. http port 80)
  - `iptables -A INPUT -p tcp --dport 80 -j DROP`

Firewall (iptables)

Practical:

- Block connection to a network interface
  - `iptables -A INPUT -i enps03 -s 192.168.0.25 -j DROP`
- Drop all traffic going to www.facebook.com
  - `host -t a www.facebook.com` = find IP address
  - `iptables -A OUTPUT -d 31.13.71.36 -j DROP`
- Block all outgoing traffic to a network range
  - `iptables -A OUTPUT -d 31.13.71.0/24 -j DROP`
- Block all incoming traffic except SSH
  - `iptables -A INPUT -p tcp --dport 22 -j ACCEPT`
  - `iptables -P INPUT DROP`
- After making all the changes save the iptables. Again make sure firewalld is not running
  - `iptables-save` = The file is save in /etc/sysconfig/iptables
- iptables saved file can also be restored
  - `iptables-restore /LOCATION/FILENAME`
- By default everything is logged in
  - `/var/log/messages`

IMPORTANT: The iptables read the rules in sequence

- DROP first then it will drop all without going to the next one
- So make sure to ACCEPT first with –I option instead of -A

Firewall (Firewalld)

- Firewalld works the same way as iptables but of course it has it own commands
  - `firewall-cmd`
- It has a few pre-defined service rules that are very easy to turn on and off
  - `Services such as: NFS, NTP, HTTPD etc.`
- Firewalld also has the following:
  - `Table`
  - `Chains`
  - `Rules`
  - `Targets`

Firewall (Firewalld)

- You can run one or the other
  - iptables or firewalld
- Make sure iptables is stopped, disabled and mask
  - `systemctl stop iptables`
  - `systemctl disable iptables`
  - `systemctl mask iptables`
- Now check if filewalld package is installed
  - `rpm -qa | grep firewalld`
- Start firewalld
  - `systemctl start/enable firewalld`
- Check the rule of firewalld
  - `firewall-cmd --list-all`
- Get the listing of all services firewalld is aware of:
  - `firewall-cmd --get-services`
- To make firewalld re-read the configuration added
  - `firewall-cmd --reload`

Firewall (Firewalld)

- The firewalld has multiple zone, to get a list of all zones
  - `firewall-cmd --get-zones`
- To get a list of active zones
  - `firewall-cmd --get-active-zones`
- To get firewall rules for public zone
  - `firewall-cmd --zone=public --list-all`
    OR
  - `firewall-cmd --list-all`
- All services are pre-defined by firewalld. What if you want to add a 3rd party service
  - `/usr/lib/firewalld/services/allservices.xml`
  - `Simply cp any .xml file and change the service and port number`

Firewall (Firewalld)

- To add a service (http)
  - `firewall-cmd --add-service=http`
- To remove a service
  - `firewall-cmd --remove-service=http`
- To reload the firewalld configuration
  - `firewall-cmd --reload`
- To add or remove a service permanently
  - `firewall-cmd --add-service=http --permanent`
  - `firewall-cmd --remove-service=http --permanent`
- To add a service that is not pre-defined by firewalld
  - `/usr/lib/firewalld/services/allservices.xml`
  - `Simply cp any .xml file sap.xml and change the service and port number (32)`
  - `systemctl restart firewalld`
  - `firewall-cmd --get-services (to verify new service)`
  - `Firewall-cmd --add-service=sap`

Firewall (Firewalld)

- To add a port
  - `firewall-cmd --add-port=1110/tcp`
- To remove a port
  - `firewall-cmd --remove-port=1110/tcp`
- To reject incoming traffic from an IP address
  - `firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.168.0.25" reject'`
- To block and unblock ICMP incoming traffic
  - `firewall-cmd --add-icmp-block-inversion`
  - `firewall-cmd --remove-icmp-block-inversion`
- To block outgoing traffic to a specific website/IP address
  - `host -t a www.facebook.com` = find IP address
  - `firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -d 31.13.71.36 -j DROP`

Encrypt Incoming and Outgoing Traffic

- What is data encryption?
  - Encryption is essentially a code used to hide the contents of a message or data
  - Data encryption is a security method where information is encoded and can only be accessed or decrypted by a user or a program with the correct encryption key

- Encryption method?
  - ssh vs. telnet
  - sftp vs. ftp
  - scp vs. cp
  - https vs. http
  - You can also mount remote server file system or your own home directory using special sshfs and fuse tool

SSH vs. Telnet

- SSH is a secure way to connect to a remote server
- Telnet connection does not encrypt data across the wire

- Check status of telnet service
  - `systemctl status telnet.socket`

- Stop telnet service
  - `systemctl stop telnet.socket`
  - `service xinetd start (older version)`

- Disable telnet service
  - `systemctl disable telnet.socket`
  - `chkconfig telnet off` *(older version)*

- Remove telnet package
  - `rpm -qa | grep telnet`
  - `rpm -e telnet-server.xxx`
  - `rpm -e xinetd` *(older version)*