Day 41

Exploitation Analyst

Firewalls and TCP wrappers:

Iptables:

What is a Firewall?

Firewalls are security systems that control incoming and outgoing network traffic based on rules.

- **Purpose**: Block unauthorized access, allow trusted connections.
- **Types**:
  - Host-based (on a single machine, e.g., ufw, iptables).
  - Network-based (on routers/appliances).
- **Modes**:
  - Whitelist → allow only what's specified.
  - Blacklist → block certain traffic, allow rest.

A firewall is hardware or software?

A firewall can be both:

- **Hardware firewall** → a physical device (e.g., in routers, network appliances) that filters traffic between networks.
- **Software firewall** → runs on an OS (e.g., ufw, iptables, Windows Firewall) to filter traffic on that machine.

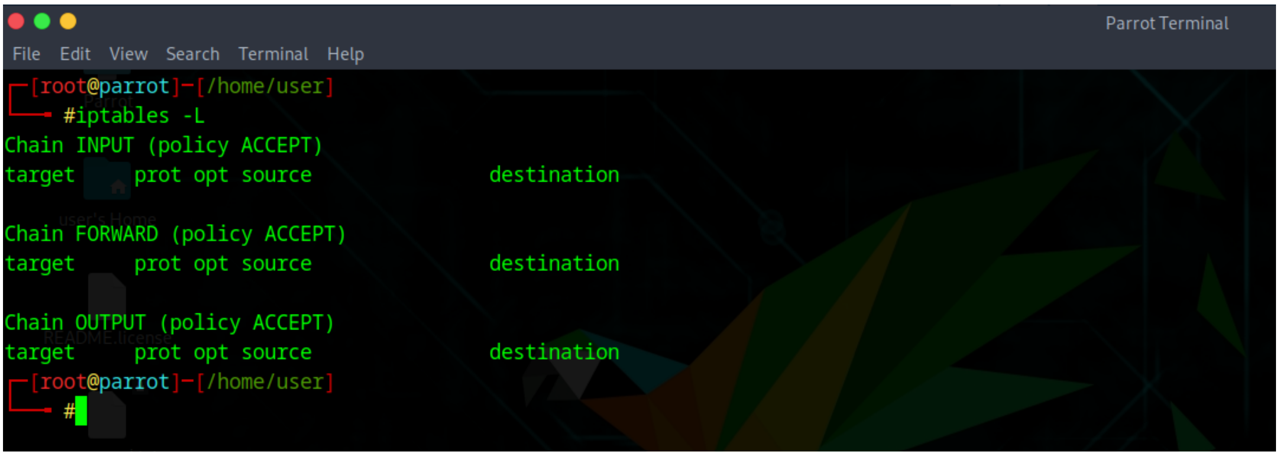
Analysing Iptables:

Command:

iptables -L

lists all current firewall rules in the filter table (default table).

- Shows chains: INPUT, FORWARD, OUTPUT.
- Displays rules, targets (ACCEPT, DROP), protocols, sources, destinations.



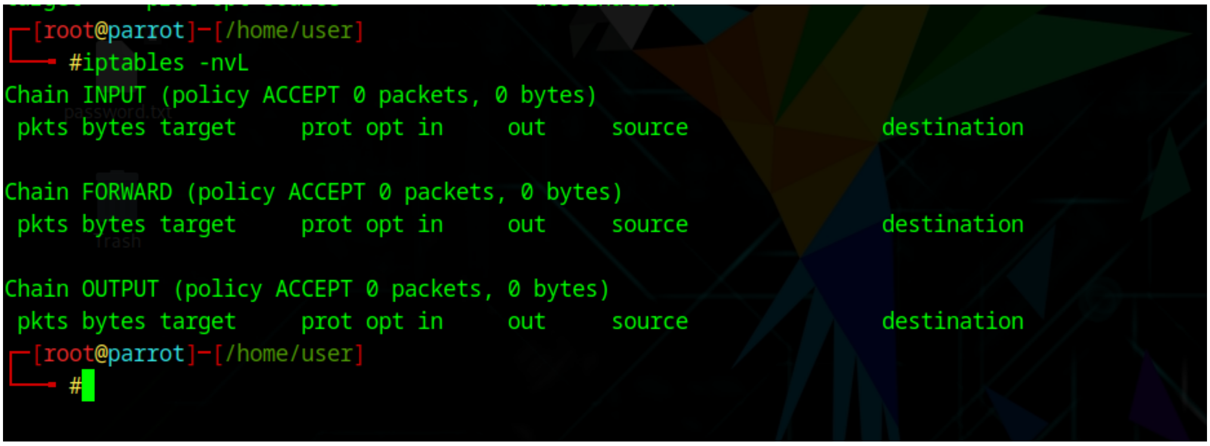
```
┌─[root@parrot]─[/home/user]
└──╼ #iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
┌─[root@parrot]─[/home/user]
└──╼ #
```

The output above shows that the INPUT, FORWARD, and OUTPUT chains exist, but there are no custom rules defined for them. The default policy for each chain is set to ACCEPT, which means all incoming connections, all forwarded traffic, and all outgoing traffic are currently allowed without restriction. In other words, your firewall is completely open and not blocking any traffic.

Command:

iptables -nvL

- -n → shows IP addresses and ports as numbers (faster, no DNS lookup).
- -v → shows detailed info like packet counts, byte counts, and interface.
- -L → lists rules.



```
┌─[root@parrot]─[/home/user]
└──╼ #iptables -nvL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
┌─[root@parrot]─[/home/user]
└──╼ #
```

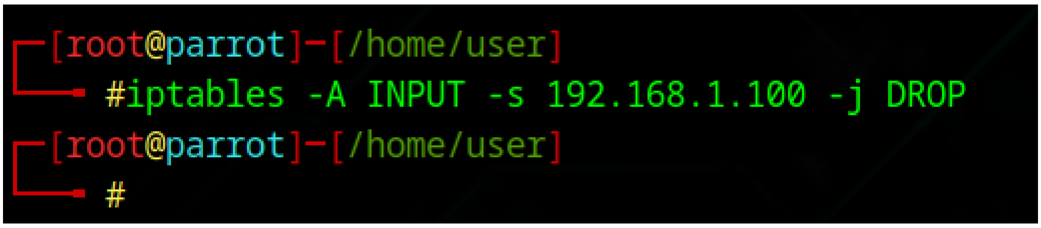
This output shown above means your firewall has no active filtering rules.

- Chains (INPUT, FORWARD, OUTPUT) exist, but each shows 0 packets, 0 bytes, meaning no traffic has been processed by iptables yet.
- Policy = ACCEPT → all traffic is allowed by default.
- No targets (like DROP or REJECT) are set.

Command: to block/ drop input / traffic from an IP.

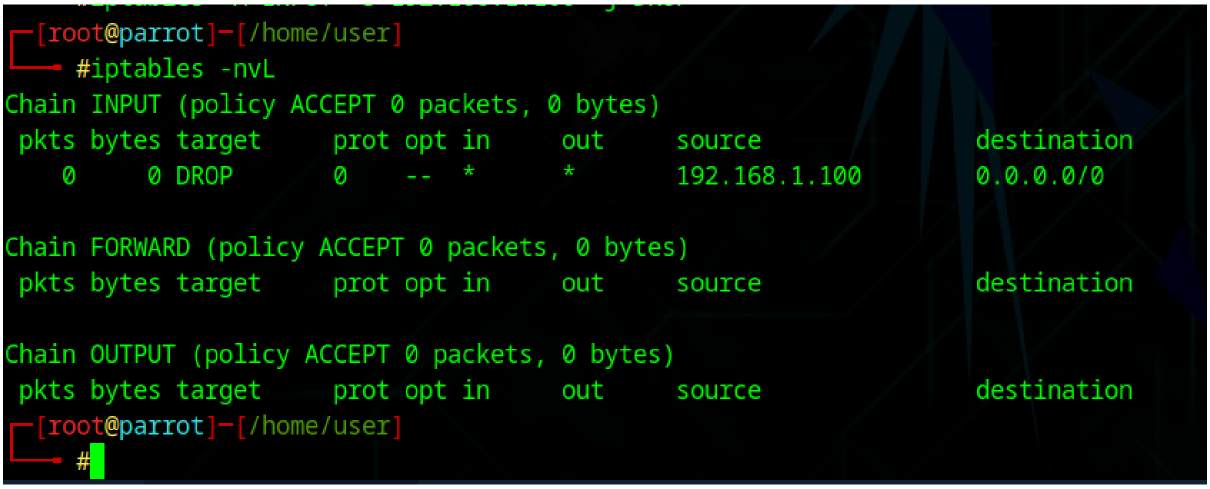
sudo iptables -A INPUT -s <IP_ADDRESS> -j DROP

- -A INPUT → appends rule to the INPUT chain.
- -s <IP> → source IP to match.
- -j DROP → silently drops packets (no response).



```
┌─[root@parrot]─[/home/user]
└──╼ #iptables -A INPUT -s 192.168.1.100 -j DROP
┌─[root@parrot]─[/home/user]
└──╼ #
```

To validate check the iptables:



```
┌─[root@parrot]─[/home/user]
└──╼ #iptables -nvL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       0    --  *      *       192.168.1.100        0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
┌─[root@parrot]─[/home/user]
└──╼ #
```

In the INPUT chain, you now see:

DROP ... 192.168.1.100 → 0.0.0.0/0

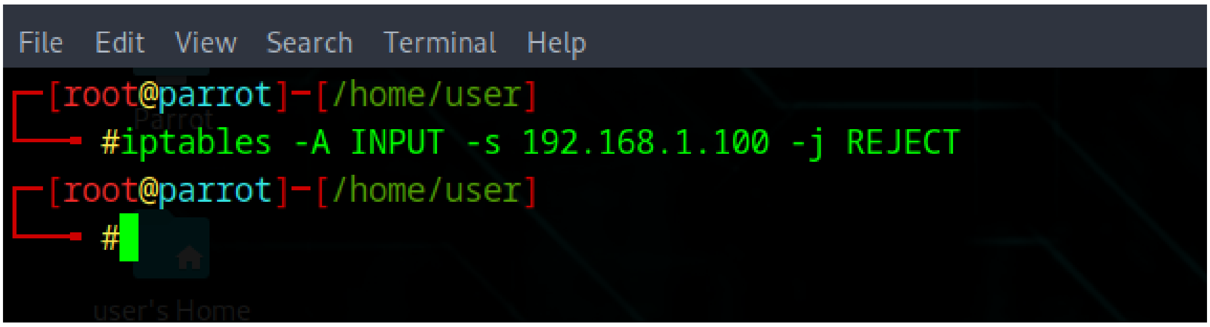
This means any traffic from 192.168.1.100 to your system will be dropped silently.

- The counters (0 pkts, 0 bytes) are still zero because no packets from that IP have hit the firewall since you added the rule. They'll increase if/when that IP actually sends traffic.

Note: Right now, the block is active, but it will reset after reboot unless you save it.

Command:

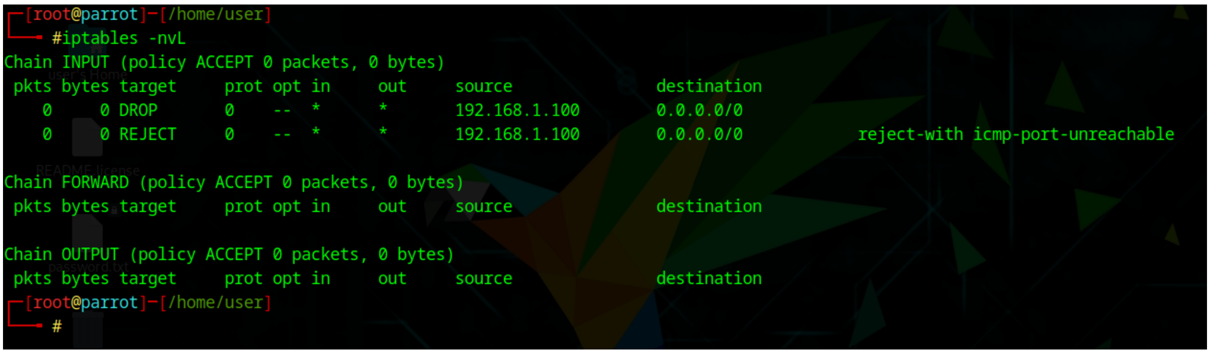
sudo iptables -A INPUT -s <IP_ADDRESS> -j REJECT



```
┌─[root@parrot]─[/home/user]
└──╼ #iptables -A INPUT -s 192.168.1.100 -j REJECT
┌─[root@parrot]─[/home/user]
└──╼ #
```

REJECT → actively refuses the packet, sending back an error (e.g., ICMP *port unreachable*).

Cross check:



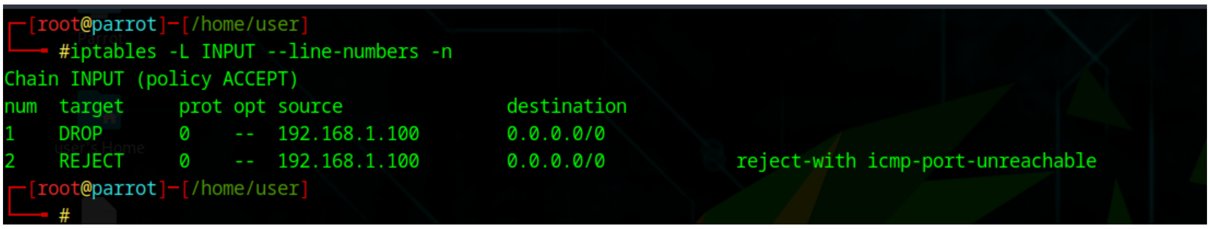
```
┌─[root@parrot]─[/home/user]
└──╼ #iptables -nvL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       0    --  *      *       192.168.1.100        0.0.0.0/0
    0     0 REJECT     0    --  *      *       192.168.1.100        0.0.0.0/0            reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
┌─[root@parrot]─[/home/user]
└──╼ #
```

Note:

1. Use **DROP** when you want to be stealthy (attacker doesn't know if a firewall is there).
2. Use **REJECT** when you want to clearly deny access (common in internal networks where users should know they're blocked).

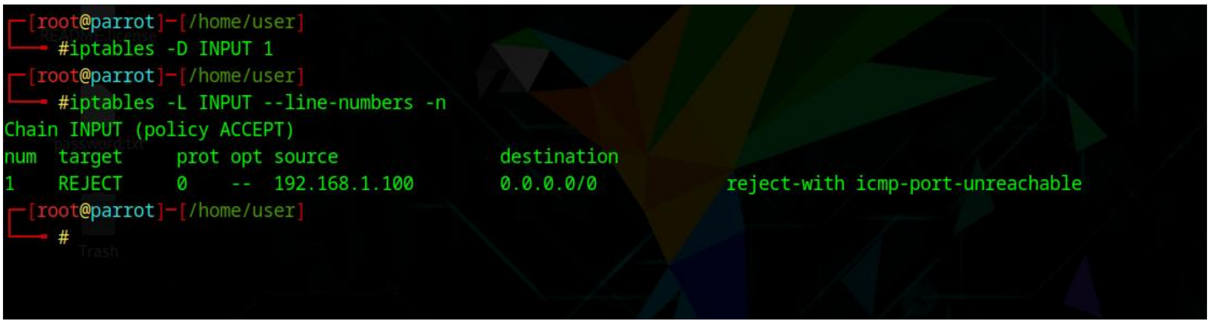
Deleting IP listed in IP tables:
Command:

> ***sudo iptables -L INPUT --line-numbers -n***
>
> ***sudo iptables -D INPUT 1***



```
┌─[root@parrot]─[/home/user]
└──╼ #iptables -L INPUT --line-numbers -n
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       0    --  192.168.1.100        0.0.0.0/0
2    REJECT     0    --  192.168.1.100        0.0.0.0/0            reject-with icmp-port-unreachable
┌─[root@parrot]─[/home/user]
└──╼ #
```

This shows the INPUT chain has two rules. Rule 1 drops all traffic from 192.168.1.100 silently, while rule 2 rejects traffic from the same IP with an ICMP "port unreachable" message. Since rules are checked top-down, the DROP rule takes effect first, making the REJECT rule unused.



```
┌─[root@parrot]─[/home/user]
└──╼ #iptables -D INPUT 1
┌─[root@parrot]─[/home/user]
└──╼ #iptables -L INPUT --line-numbers -n
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     0    --  192.168.1.100        0.0.0.0/0            reject-with icmp-port-unreachable
┌─[root@parrot]─[/home/user]
└──╼ #
```

Now the DROP rule is removed, and only one rule remains: rule 1, which rejects traffic from 192.168.1.100 by sending back an ICMP "port unreachable" message. So instead of being silently ignored, that IP now gets an explicit rejection response.

What actually happens in background when we uses, these commands:

Here's what happens in the background for each command:

iptables -nvL
This queries the kernel's netfilter framework and lists all active firewall rules in memory. The -n flag avoids DNS lookups, and -v shows packet/byte counters. No changes are made; it's just reading rules.

sudo iptables -A INPUT -s <IP_ADDRESS> -j DROP
This appends a new rule to the INPUT chain. When a packet arrives from the given IP, the kernel checks the chain and silently discards the packet without replying. The rule is stored in the kernel's netfilter tables.

sudo iptables -A INPUT -s <IP_ADDRESS> -j REJECT
This appends another rule after the previous one. Packets from that IP now hit the DROP rule first,

but if DROP is removed, they reach this REJECT rule, which discards them while sending back an ICMP “port unreachable” response.

sudo iptables -L INPUT --line-numbers -n
This lists only the INPUT chain with line numbers, allowing you to identify each rule’s exact position. Internally, iptables fetches the chain data from the kernel and adds numbering for easy deletion or editing.

sudo iptables -D INPUT 1
This deletes the first rule in the INPUT chain by its position number. The kernel’s netfilter table is updated in memory, removing that entry, so it no longer affects packet filtering.

--The End--