Firewall Configuration Cheatsheet

A comprehensive cheat sheet covering essential firewall configurations, rules, and best practices for various operating systems and network environments. This guide provides quick references and examples to help secure your systems effectively.

Firewall Fundamentals

Basic Concepts

What is a Firewall?

A network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. It acts as a barrier between a trusted internal network and an untrusted external network, such as the Internet.

Types of Firewalls:

- **Hardware Firewalls:** Physical devices that protect the entire network.
- **Software Firewalls:** Applications installed on individual machines protecting that specific system.

Key Functions:

- **Packet Filtering:** Examining network packets and allowing or blocking them based on source/destination IP addresses, ports, and protocols.
- **Stateful Inspection:** Tracking the state of network connections and making decisions based on the context of those connections.
- **Proxy Service:** Intermediating network connections to hide internal IP addresses and provide additional security.

Default Policy:

Firewalls operate based on either:

- **Default Deny:** Block all traffic unless explicitly allowed.
- **Default Allow:** Allow all traffic unless explicitly blocked.

Default Deny is generally more secure.

Firewall Rule Components

Source IP Address	The IP address or address range from which the traffic originates.
Destination IP Address	The IP address or address range to which the traffic is directed.
Source Port	The port number from which the traffic originates.
Destination Port	The port number to which the traffic is directed.
Protocol	The communication protocol used (e.g., TCP, UDP, ICMP).
Action	The action to take when a rule matches (e.g., ALLOW, DENY, REJECT).

iptables (Linux)

iptables Commands

`iptables -L`	List all current rules in all tables.
`iptables -t <table_name> -L`	List rules in a specific table (e.g., `filter`, `nat`, `mangle`).
`iptables -A <chain_name> <rule>`	Append a new rule to the end of a chain (e.g., `INPUT`, `OUTPUT`, `FORWARD`).
`iptables -I <chain_name> <rule>`	Insert a new rule at the beginning of a chain.
`iptables -D <chain_name> <rule_number>`	Delete a rule by its number in the chain. Use `iptables -L --line-numbers` to see line numbers.
`iptables -F`	Flush all rules in the current table.
`iptables -X`	Delete a user-defined chain.
`iptables -P <chain_name> <target>`	Set the default policy for a chain (e.g., `ACCEPT`, `DROP`).
`iptables -S`	Display all rules in iptables using the command syntax.

Example iptables Rules

Allow SSH traffic:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Allow HTTP traffic:

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Allow HTTPS traffic:

```
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

Drop all ICMP traffic:

```
iptables -A INPUT -p icmp -j DROP
```

Allow established and related connections:

```
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Drop all other incoming traffic (Default Deny):

```
iptables -A INPUT -j DROP
```

Saving iptables Rules

To save iptables rules on Debian/Ubuntu:

```
sudo apt-get install iptables-persistent
sudo netfilter-persistent save
```

To save iptables rules on CentOS/RHEL:

```
sudo yum install iptables-services
sudo systemctl enable iptables
sudo systemctl start iptables
sudo iptables-save > /etc/sysconfig/iptables
```

firewalld (Linux)

firewalld Basics

firewalld is a dynamic firewall management tool with support for network/firewall zones to define the trust level of network connections.

Key Concepts:

- **Zones:** Predefined sets of rules (e.g., `public`, `private`, `trusted`).
- **Services:** Predefined configurations for common network services (e.g., `http`, `https`, `ssh`).
- **Ports:** Specific TCP or UDP ports to open.

firewalld Commands

`sudo firewall-cmd --state`	Check the status of firewalld.
`sudo firewall-cmd --get-default-zone`	Get the default zone.
`sudo firewall-cmd --set-default-zone=<zone>`	Set the default zone (e.g., `public`).
`sudo firewall-cmd --get-active-zones`	List active zones.
`sudo firewall-cmd --zone=<zone> --list-all`	List all settings for a zone.
`sudo firewall-cmd --list-services`	List all available services.
`sudo firewall-cmd --zone=<zone> --add-service=<service> --permanent`	Add a service to a zone permanently.
`sudo firewall-cmd --zone=<zone> --remove-service=<service> --permanent`	Remove a service from a zone permanently.
`sudo firewall-cmd --zone=<zone> --add-port=<port>/<protocol> --permanent`	Add a port to a zone permanently.
`sudo firewall-cmd --reload`	Reload firewalld to apply changes.

Example firewalld Configurations

Allow SSH traffic in the public zone:

```
sudo firewall-cmd --zone=public --add-service=ssh --permanent
sudo firewall-cmd --reload
```

Allow HTTP and HTTPS traffic in the public zone:

```
sudo firewall-cmd --zone=public --add-service=http --permanent
sudo firewall-cmd --zone=public --add-service=https --permanent
sudo firewall-cmd --reload
```

Allow a custom port (e.g., 8080) in the public zone:

```
sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
sudo firewall-cmd --reload
```

Remove a service (e.g., http) from the public zone:

```
sudo firewall-cmd --zone=public --remove-service=http --permanent
sudo firewall-cmd --reload
```

ufw (Ubuntu Firewall)

ufw Basics

ufw (Uncomplicated Firewall) is a user-friendly frontend for iptables, designed to simplify firewall management.

ufw provides a command-line interface for managing firewall rules, making it easier to configure common firewall settings.

ufw Commands

`sudo ufw enable`	Enable the firewall.
`sudo ufw disable`	Disable the firewall.
`sudo ufw status`	Check the status of the firewall.
`sudo ufw default deny incoming`	Set the default incoming policy to deny.
`sudo ufw default allow outgoing`	Set the default outgoing policy to allow.
`sudo ufw allow <port>`	Allow traffic on a specific port.
`sudo ufw deny <port>`	Deny traffic on a specific port.
`sudo ufw allow <service>`	Allow traffic for a specific service (e.g., ssh, http, https).
`sudo ufw delete allow <rule>`	Delete a specific rule.
`sudo ufw reload`	Reload the firewall to apply changes.

Example ufw Configurations

Allow SSH traffic:

```
sudo ufw allow ssh
```

Allow HTTP traffic:

```
sudo ufw allow http
```

Allow HTTPS traffic:

```
sudo ufw allow https
```

Allow traffic on port 8080:

```
sudo ufw allow 8080
```

Deny traffic on port 25:

```
sudo ufw deny 25
```

Delete a rule allowing port 8080:

```
sudo ufw delete allow 8080
```