

Ejercicios sencillos de iptables en Linux

Ejercicio 1 – Permitir SSH en el puerto 22

🎯 **Objetivo:** Asegurar que puedes conectarte por SSH desde tu equipo Windows o desde Kali a sí misma.

🔧 **Comando:**

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

```
~/Documents/box > sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT  
[sudo] password for kali:
```

🔧 **Cómo probarlo:**

Desde Kali:

ssh [kali@localhost](#)

```
~/Documents/box > ssh kali@localhost  
kali@localhost's password:  
Last login: Tue Sep 30 12:50:50 CEST 2025 from ::1 on ssh  
Linux kali 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1k  
The programs included with the Kali GNU/Linux system are free software
```

Desde Windows (PowerShell o CMD):

ssh kali@192.168.1.55

✅ Si la conexión es exitosa, la regla funciona.

```
PS C:\Users\2-DAW> ssh kali@192.168.0.21  
ssh: connect to host 192.168.0.21 port 22: Connection refused  
PS C:\Users\2-DAW> ssh kali@192.168.0.41  
The authenticity of host '192.168.0.41 (192.168.0.41)' can't be established.  
ED25519 key fingerprint is SHA256:ZfpsKoPyRh/gaHIXWnSRiBggjAH0VqVxBh9TYI69b2w.  
This host key is known by the following other names/addresses:  
C:\Users\2-DAW/.ssh/known_hosts:1: 192.168.0.78  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.0.41' (ED25519) to the list of known hosts.  
kali@192.168.0.41's password:  
Last login: Tue Sep 30 14:20:36 CEST 2025 from ::1 on ssh  
Linux kali 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) x86_64  
  
The programs included with the Kali GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Tue Sep 30 14:20:36 2025 from ::1  
~
```

♦ Ejercicio 2 – Bloquear el puerto 23 (Telnet)

🎯 Objetivo: Simular bloqueo de servicios no utilizados como Telnet.

🔧 Comando:

sudo iptables -A INPUT -p tcp --dport 23 -j DROP

🔧 Cómo probarlo:

En Kali, install Telnet:

sudo apt install telnet -y

Ejecuta:

telnet localhost 23

❌ Verás que no responde (se cuelga), porque el firewall lo está bloqueando.

```
~ > sudo iptables -A INPUT -p tcp --dport 23 -j DROP
~ > telnet localhost 23
Trying ::1...
Connection failed: Connection refused
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused
```

♦ Ejercicio 3 – Permitir tráfico HTTP (puerto 80)

🎯 Objetivo: Habilitar el acceso web desde el navegador de Windows al servidor Apache de Kali.

🔧 Comando:

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

🔧 Cómo probarlo:

Instala Apache en Kali:

sudo apt install apache2 -y

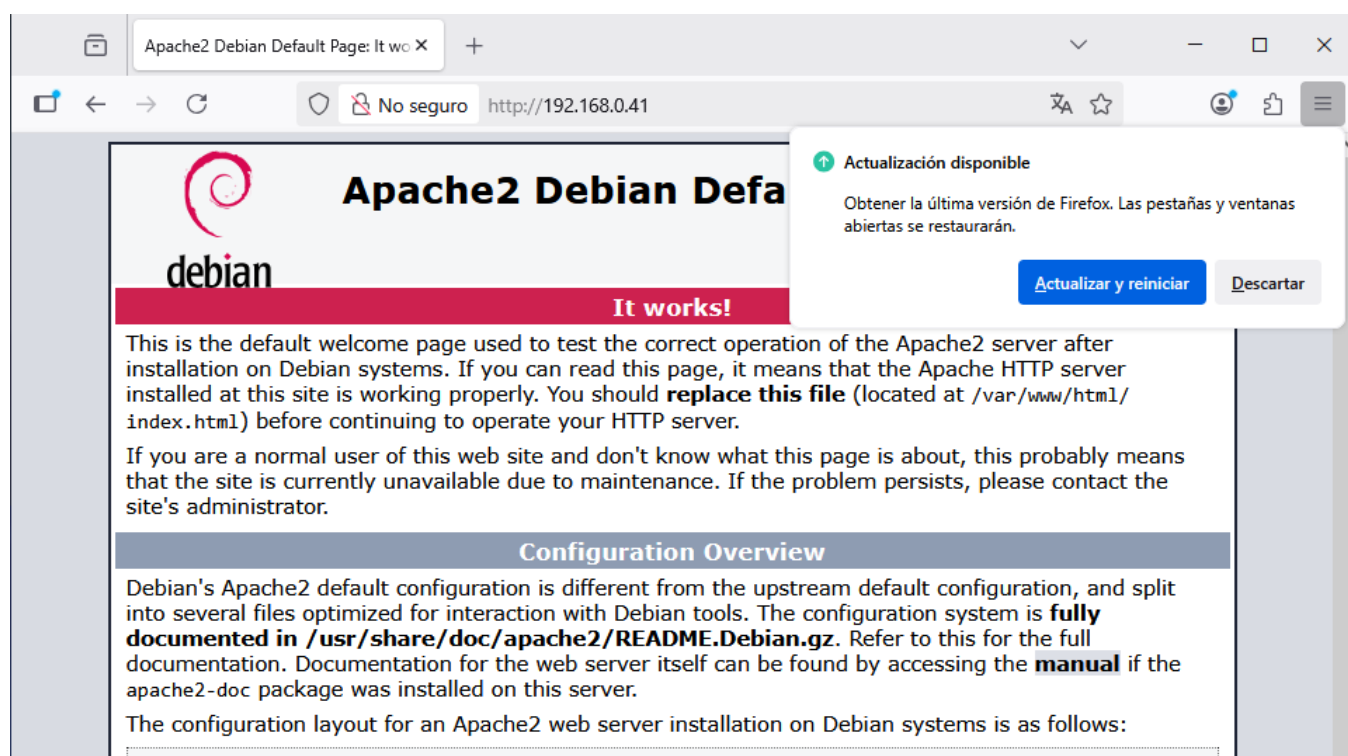
```
~ > sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
~ > sudo apt install apache2 -y
apache2 is already the newest version (2.4.65-3+b1).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 43
```

http://192.168.1.55

```
~ > curl localhost:80

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//
dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; cha
    <title>Apache2 Debian Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }

    body, html {
      padding: 3px 3px 3px 3px;
```



🎯 **Objetivo:** Ver qué reglas están en uso y cuántas veces se han aplicado.

```
sudo iptables -L -n -v
```

```

~ > sudo iptables -L -n -v
Chain INPUT (policy DROP 315 packets, 259K bytes)
 pkts bytes target     prot opt in     out     source    destination
38196 121M ufw-before-logging-input all -- *      *       0.0.0.0/0 0.0.0.0/0
38196 121M ufw-before-input all -- *      *       0.0.0.0/0 0.0.0.0/0
8987 866K ufw-after-input all -- *      *       0.0.0.0/0 0.0.0.0/0
403 264K ufw-after-logging-input all -- *      *       0.0.0.0/0 0.0.0.0/0
403 264K ufw-reject-input all -- *      *       0.0.0.0/0 0.0.0.0/0
403 264K ufw-track-input all -- *      *       0.0.0.0/0 0.0.0.0/0
1 52 ACCEPT tcp -- *      *       0.0.0.0/0 0.0.0.0/0 tcp dpt:22
0 0 DROP tcp -- *      *       0.0.0.0/0 0.0.0.0/0 tcp dpt:23
0 0 ACCEPT tcp -- *      *       0.0.0.0/0 0.0.0.0/0 tcp dpt:23
0 0 ACCEPT tcp -- *      *       0.0.0.0/0 0.0.0.0/0 tcp dpt:23
87 4524 ACCEPT tcp -- *      *       0.0.0.0/0 0.0.0.0/0 tcp dpt:80

Chain FORWARD (policy DROP 221 packets, 849K bytes)
 pkts bytes target     prot opt in     out     source    destination
233 864K DOCKER-USER all -- *      *       0.0.0.0/0 0.0.0.0/0
233 864K DOCKER-FORWARD all -- *      *       0.0.0.0/0 0.0.0.0/0
221 849K ufw-before-logging-forward all -- *      *       0.0.0.0/0 0.0.0.0/0
221 849K ufw-before-forward all -- *      *       0.0.0.0/0 0.0.0.0/0
221 849K ufw-after-forward all -- *      *       0.0.0.0/0 0.0.0.0/0
221 849K ufw-after-logging-forward all -- *      *       0.0.0.0/0 0.0.0.0/0

```

🔍 Verás columnas como:

pkts → paquetes que han coincidido con esa regla

bytes → tráfico acumulado

target → lo que hace (ACCEPT, DROP...)

dpt: → puerto destino (como 22 o 80)

♦ Ejercicio 5 – Borrar todas las reglas

🎯 Objetivo: Dejar el sistema sin restricciones para reiniciar las pruebas.

🔧 Comando:

sudo iptables -F

✅ Después de esto, todo el tráfico está permitido (por defecto en Kali). Puedes verificarlo intentando Telnet o SSH sin reglas previas.

```

~ > sudo iptables -F

~ > sudo iptables -L -n -v
Chain INPUT (policy DROP 327 packets, 261K bytes)
 pkts bytes target     prot opt in     out     source    destination

Chain FORWARD (policy DROP 221 packets, 849K bytes)
 pkts bytes target     prot opt in     out     source    destination

Chain OUTPUT (policy ACCEPT 1096 packets, 43196 bytes)
 pkts bytes target     prot opt in     out     source    destination

Chain DOCKER (0 references)

```

♦ Ejercicio 6 – Crear y usar una cadena personalizada

🎯 Objetivo: Organizar las reglas en una cadena propia (por ejemplo, "MI_CADENA").

🔧 Comandos:

sudo iptables -N MI_CADENA

sudo iptables -A INPUT -j MI_CADENA

sudo iptables -A MI_CADENA -p tcp --dport 443 -j ACCEPT

```
~ > sudo iptables -N MI_CADENA
~ > sudo iptables -A INPUT -j MI_CADENA
```

```
~ > sudo iptables -A MI_CADENA -p tcp --dport 443 -j ACCEPT
```

🔧 Cómo probarlo:

Activa un servicio en el puerto 443 (HTTPS) o prueba con curl si tienes configurado algo:

curl -v https://localhost

✅ Verás que el tráfico pasa por tu cadena personalizada.

```
~ > curl -v https://localhost
* Host localhost:443 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:443 ...
* connect to ::1 port 443 from ::1 port 52210 failed: Connection refused
*   Trying 127.0.0.1:443 ...
^C

~ > curl -v http://localhost
* Host localhost:80 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:80 ...
* Connected to localhost (::1) port 80
* using HTTP/1.x
> GET / HTTP/1.1
> Host: localhost
> User-Agent: curl/8.15.0
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Date: Tue, 30 Sep 2025 12:41:54 GMT
< Server: Apache/2.4.65 (Debian)
< Last-Modified: Thu, 29 May 2025 19:23:17 GMT
< ETag: "29cf-6364b3ad32860"
< Accept-Ranges: bytes
< Content-Length: 10703
< Vary: Accept-Encoding
< Content-Type: text/html
<
```

♦ Ejercicio 7 – Registrar intentos de conexión a SSH en logs

🎯 **Objetivo:** Ver en los logs del sistema cada vez que alguien intenta usar SSH.

🔧 **Comando:**

sudo iptables -A INPUT -p tcp --dport 22 -j LOG --log-prefix "Intento SSH: "

🔧 **Cómo probarlo:**

Conéctate a Kali desde Windows o desde otra terminal:

ssh kali@localhost

En otra ventana de terminal, ejecuta:

sudo journalctl -f

✅ Verás una línea con el prefijo Intento SSH: indicando que se ha detectado un intento de conexión.

```
~ > sudo iptables -A INPUT -p tcp --dport 22 -j LOG --log-prefix "Intento SSH: "
~ > sudo journalctl -f
Sep 30 14:44:36 kali systemd[1]: Started session-59.scope - Session 59 of User kali.
Sep 30 14:44:40 kali sshd-session[18476]: syslogin_perform_logout: logout() returned an error
Sep 30 14:44:40 kali sshd-session[18484]: Received disconnect from ::1 port 54696:11: disconnected by user
Sep 30 14:44:40 kali sshd-session[18484]: Disconnected from user kali ::1 port 54696
Sep 30 14:44:40 kali sshd-session[18476]: pam_unix(sshd:session): session closed for user kali
Sep 30 14:44:40 kali systemd[1]: session-59.scope: Deactivated successfully.
Sep 30 14:44:40 kali systemd-logind[601]: Session 59 logged out. Waiting for processes to exit.
Sep 30 14:44:40 kali systemd-logind[601]: Removed session 59.
Sep 30 14:44:48 kali sudo[18540]:      kali : TTY=pts/4 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/journalctl -f
Sep 30 14:44:48 kali sudo[18540]: pam_unix(sudo:session): session opened for user root(uid=0) by kali(uid=1000)
```

```
~ > cat /var/log/syslog | grep Intento
2025-09-30T14:44:17.253462+02:00 kali kernel: Intento SSH: IN=eth0 OUT= MAC=08:00:27:d1:f8:5d:d8:43:ae:44:e2:25:08:0
0 SRC=192.168.0.21 DST=192.168.0.41 LEN=52 TOS=0x00 PREC=0x00 TTL=128 ID=55398 DF PROTO=TCP SPT=26227 DPT=22 WINDOW=
64240 RES=0x00 SYN URGP=0
2025-09-30T14:44:18.254480+02:00 kali kernel: Intento SSH: IN=eth0 OUT= MAC=08:00:27:d1:f8:5d:d8:43:ae:44:e2:25:08:0
0 SRC=192.168.0.21 DST=192.168.0.41 LEN=52 TOS=0x00 PREC=0x00 TTL=128 ID=55399 DF PROTO=TCP SPT=26227 DPT=22 WINDOW=
64240 RES=0x00 SYN URGP=0
2025-09-30T14:44:20.254578+02:00 kali kernel: Intento SSH: IN=eth0 OUT= MAC=08:00:27:d1:f8:5d:d8:43:ae:44:e2:25:08:0
0 SRC=192.168.0.21 DST=192.168.0.41 LEN=52 TOS=0x00 PREC=0x00 TTL=128 ID=55400 DF PROTO=TCP SPT=26227 DPT=22 WINDOW=
64240 RES=0x00 SYN URGP=0
2025-09-30T14:44:24.254122+02:00 kali kernel: Intento SSH: IN=eth0 OUT= MAC=08:00:27:d1:f8:5d:d8:43:ae:44:e2:25:08:0
0 SRC=192.168.0.21 DST=192.168.0.41 LEN=52 TOS=0x00 PREC=0x00 TTL=128 ID=55401 DF PROTO=TCP SPT=26227 DPT=22 WINDOW=
64240 RES=0x00 SYN URGP=0
2025-09-30T14:44:32.254756+02:00 kali kernel: Intento SSH: IN=eth0 OUT= MAC=08:00:27:d1:f8:5d:d8:43:ae:44:e2:25:08:0
0 SRC=192.168.0.21 DST=192.168.0.41 LEN=52 TOS=0x00 PREC=0x00 TTL=128 ID=55402 DF PROTO=TCP SPT=26227 DPT=22 WINDOW=
64240 RES=0x00 SYN URGP=0
```