

## Unknown Title

---



Tú dijiste:

que hace? iptables -A INPUT -j DROP

ChatGPT Plus

El comando:

bash

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

hace lo siguiente:

### 🔍 Desglose por partes

- **iptables**: herramienta para gestionar el cortafuegos en Linux.
- **-A INPUT**: *Append* → añade una regla **al final** de la cadena **INPUT**.
- **-j DROP**: *jump* → la acción es **descartar el paquete** silenciosamente.

## 🚫 Qué significa en la práctica

👉 Bloquea todo el tráfico entrante que llegue a esa regla y no haya sido aceptado antes.

- El paquete **se descarta sin avisar** (no se responde al emisor).
- El equipo se vuelve “invisible” para ese tráfico.
- Es más estricto que REJECT, que sí devuelve un mensaje de error.

## ⚠️ Muy importante

Si esta regla se ejecuta **sin reglas ACCEPT antes**, puede provocar que:

- Pierdas acceso por **SSH**
- No funcione **HTTP / FTP / cualquier servicio**

Ejemplo peligroso:

bash

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

👉 Resultado: **no entra nada**, ni siquiera desde tu propia red.

## ✓ Uso correcto típico

Primero permites lo necesario, **y al final bloqueas el resto**:

bash

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT      # SSH  
iptables -A INPUT -p tcp --dport 80 -j ACCEPT      # HTTP  
iptables -A INPUT -j DROP                          # Todo lo demás
```

Siquiero, te explico la diferencia exacta entre **DROP vs REJECT** o cómo afecta al **orden de las reglas** (muy típico en exámenes de ASIR 😊).

Tú dijiste:

que hace este script de iptables? #!/bin/bash  
iptables -A INPUT -p tcp --dport 80 -j ACCEPT  
iptables -A INPUT -p tcp --dport 443 -j ACCEPT  
sudo iptables -A INPUT -p tcp --dport 21 -j ACCEPT  
sudo iptables -A INPUT -p tcp --dport 56123 -j ACCEPT  
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT  
iptables -A INPUT -j DROP

Este **script de iptables** configura un **cortafuegos básico** que **solo permite ciertos servicios** y **bloquea todo lo demás**.

Vamos paso a paso 

---

## 1 Cabecera del script

bash

```
#!/bin/bash
```

Indica que el script debe ejecutarse con **bash**.

(Ojo: *hay un pequeño error en tu texto, debería ser #!/bin/bash, sin espacio*)

---

## 2 Reglas que permiten tráfico entrante (INPUT)

### Web (HTTP y HTTPS)

bash

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT  
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

- Permite acceder al **servidor web**:
    - Puerto **80** → HTTP
    - Puerto **443** → HTTPS
- 

### FTP

bash

```
sudo iptables -A INPUT -p tcp --dport 21 -j ACCEPT  
sudo iptables -A INPUT -p tcp --dport 56123 -j ACCEPT
```

- Permite:
  - **21** → FTP (canal de control)
  - **56123** → puerto alto típico de **FTP pasivo**

 Necesario para que las transferencias FTP funcionen correctamente.

---

### SSH

bash

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

- Permite acceso remoto por **SSH** desde el exterior.

### 3 Regla final: bloqueo total

bash

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

- **Descarta todo el tráfico entrante** que no haya sido aceptado antes.
- No responde al emisor (modo silencioso).
- Es una política de “**denegar todo excepto lo permitido**”.

## Comportamiento global del script

### Permite solo:

- Web (80, 443)
- FTP (21 y 56123)
- SSH (22)

### Bloquea:

- Cualquier otro puerto o protocolo (ping, SMTP, MySQL, etc.)

## Detalles importantes (muy de examen)

### ◆ Orden de las reglas

iptables evalúa **de arriba hacia abajo**.

Si el DROP estuviera arriba → **no funcionaría nada**.

### ◆ No permite tráfico local ni conexiones establecidas

Falta algo típico:

bash

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  
iptables -A INPUT -i lo -j ACCEPT
```

Sin esto:

- Respuestas a conexiones salientes pueden fallar
  - Programas locales pueden no funcionar bien
- 

## Conclusión

👉 Es un **firewall restrictivo** para un **servidor web + FTP + SSH**, muy típico en prácticas de **ASIR / Seguridad**.

Si quieres, te lo **corrijo y mejoró** para que quede **perfecto y “de 10” para examen** o te explico **qué pasaría si se elimina una línea concreta**.

Tú dijiste:

corrije y mejora

ChatGPT Plus

Perfecto 👍

Te dejo el **script corregido y mejorado**, con **buenas prácticas**, y después te explico **por qué es mejor** (tal como lo pedirían en ASIR).

---

## Script IPTABLES corregido y mejorado

bash

```
#!/bin/bash

# Limpiar reglas anteriores
iptables -F
iptables -X

# Política por defecto: denegar todo
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# Permitir tráfico local (loopback)
iptables -A INPUT -i lo -j ACCEPT

# Permitir conexiones ya establecidas o relacionadas
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# -----
# Servicios permitidos
# -----



# HTTP y HTTPS
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT

# FTP (control + modo pasivo)
iptables -A INPUT -p tcp --dport 21      -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 56123   -m state --state NEW -j ACCEPT

# SSH
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT

# (Opcional) permitir ping
# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

## 🧠 ¿Qué mejoras se han hecho?

### 1 Shebang corregido

bash

```
#!/bin/bash
```

✓ Ahora el script se ejecuta correctamente.

### 2 Limpieza previa del firewall

bash

```
iptables -F
iptables -X
```

✓ Evita conflictos con reglas antiguas.

### 3 Política por defecto segura

bash

```
iptables -P INPUT DROP
```

✓ Mucho mejor que un DROP al final:

- Es más claro
  - Más profesional
  - Más típico de examen
- 

## 4 Permitir loopback

bash

```
iptables -A INPUT -i lo -j ACCEPT
```

✓ Necesario para:

- Servicios locales
  - MySQL local
  - Apache/PHP
  - Scripts del sistema
- 

## 5 Permitir conexiones establecidas

bash

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

✓ Sin esto:

- No recibes respuestas a conexiones salientes
  - Internet “parece no funcionar”
- 

## 6 Uso correcto de NEW

bash

```
-m state --state NEW
```

✓ Acepta solo nuevas conexiones a los puertos permitidos

✓ Más seguro y más limpio

---

## 7 No usar sudo dentro del script

✓ El script debe ejecutarse como root:

bash

```
sudo ./firewall.sh
```

## 🔒 Comportamiento final del firewall

✓ Permite:

- HTTP / HTTPS
- FTP (activo + pasivo)
- SSH
- Tráfico local
- Respuestas a conexiones

✗ Bloquea:

- Todo lo demás (SMTP, MySQL externo, escaneos, etc.)

## 🎓 Frase perfecta para examen

*“Se aplica una política restrictiva, permitiendo únicamente los servicios necesarios y bloqueando el resto del tráfico entrante.”*

Si quieres, te hago:

- versión **mínima para examen**
- versión con **limitación de SSH**
- versión con **logs**
- versión **iptables-save compatible**

Dime 😊

Is this conversation helpful so far?

ChatGPT puede cometer errores. Considera verificar la información importante. Ver [preferencias de cookies](#).