**1) Breve introducción: ¿qué hace un firewall?**

Un firewall es un dispositivo o servicio que controla el tráfico de red entre interfaces (p. ej. LAN ↔ WAN) aplicando políticas: puede **permitir**, **denegar**, **registrar** o **redirigir** tráfico según atributos (IP origen/destino, puerto, protocolo, tiempo, estado de la conexión, aplicación). Hay firewalls packet-filter (iptables/nftables), stateful (mantienen estados TCP), y de capa aplicación (proxys, WAF) que entienden dominios/HTTP/HTTPS.

**Puntos claves**:

* El orden importa: la primera regla que coincide se aplica.
* Normalmente hay políticas por defecto (DROP o ACCEPT). Recomiendo: **default DROP (denegar)** y crear reglas explícitas de permit.
* Para bloquear por **dominio** (ej. www.ejemplo.com) muchas veces necesitas un proxy (Squid) o DNS filtering; bloquear solo por IP no siempre basta (CDNs, múltiples IPs).
* Para streaming/radios muchas veces usan puertos comunes y/o protocolos (HTTP stream, Icecast/Shoutcast, RTSP, RTP), por eso conviene combinar bloqueo de puertos + bloqueo por Host/DNS/proxy.

**2) Asignación de IPs (ejemplo)**

Red LAN: 192.168.1.0/24  
Firewall/gateway: 192.168.1.1  
Computadoras (10): 192.168.1.101 → 192.168.1.110

* **Admin (puede hacer todo)**: 192.168.1.101
* **Resto (9 máquinas)**:
  + **Permitidas internet (mitad del resto)** — 4 máquinas: 192.168.1.102 — 192.168.1.105
  + **No permitidas internet (la otra mitad)** — 5 máquinas: 192.168.1.106 — 192.168.1.110

(Ajusta los rangos si querés otra partición; uso estos para ejemplos.)

**3) Políticas generales recomendadas**

* Política por defecto: INPUT (local firewall) = ACCEPT (si es solo gateway, limitar INPUT), pero FORWARD (tránsito LAN→WAN) = **DROP** por defecto y crear reglas ACCEPT específicas.
* Habilitar NAT (masquerade) para los hosts permitidos si el firewall hace de gateway.
* Habilitar logging para reglas DROP/REJECT importantes.

**4) Reglas concretas (ejemplos en iptables)**

Asumo firewall Linux (iptables) con interfaz LAN eth1 (192.168.1.0/24) y WAN eth0. Ajusta nombres de interfaz a tu caso.

Primero, reglas básicas (reset y política por defecto para ejemplo):

# limpiar

iptables -F

iptables -t nat -F

iptables -X

# Políticas por defecto

iptables -P INPUT ACCEPT

iptables -P FORWARD DROP # muy importante: denegamos por defecto tránsito LAN->WAN

iptables -P OUTPUT ACCEPT

**Regla previa: permitir tráfico establecido (stateful)**

# Permitir respuestas de conexiones ya establecidas

iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

**1) "Una puede hacer todo" — permitir todo para admin 192.168.1.101**

# Permitir todo desde admin hacia internet (FORWARD)

iptables -A FORWARD -s 192.168.1.101 -o eth0 -j ACCEPT

# Si el firewall hace NAT:

iptables -t nat -A POSTROUTING -s 192.168.1.101 -o eth0 -j MASQUERADE

Esto permite cualquier protocolo/puerto desde la IP admin hacia WAN.

**2) "Bloquear una página web" — ejemplo: bloquear www.ejemplo.com**

**Nota importante:** si la web usa CDN o múltiples IPs, bloquear una sola IP puede no bastar. Para HTTP/HTTPS la forma fiable es usar **proxy (Squid)** y ACL por dominio o usar DNS filtering. Si igualmente querés bloquear por IP (rápido):

1. Resolver la IP de www.ejemplo.com (ej. 93.184.216.34) y luego:

# Bloquear acceso a la IP destino para toda la LAN

iptables -A FORWARD -s 192.168.1.0/24 -d 93.184.216.34 -j REJECT

1. Si querés que la **prohibición no aplique al admin**, permití admin antes (ya lo hicimos). Para que afecte solo a las que navegan: agregar -s 192.168.1.102/30 etc. (o usar listas).

**Opción mejor (recomendada): Squid + ACL por dominio**

* Configurás Squid como proxy transparente o explícito y en squid.conf:

acl bloquear\_pagina dstdomain .ejemplo.com

http\_access deny bloquear\_pagina

**3) "Bloquear la radio por Internet" — enfoque práctico**

Internet radio suele usar: **HTTP streams (puerto 80/443)**, **Shoutcast/Icecast (puertos 8000/8001)**, **RTSP (554)**, **RTP (puertos dinámicos UDP)**. Bloquear todo audio streaming es complicado con solo iptables, pero se puede:

**Ejemplo iptables: bloquear puertos comunes de streaming**

# Bloquear shoutcast/icecast típicos

iptables -A FORWARD -p tcp --dport 8000:8001 -s 192.168.1.0/24 -j REJECT

# Bloquear RTSP

iptables -A FORWARD -p tcp --dport 554 -s 192.168.1.0/24 -j REJECT

# Bloquear RTMP (Flash/streaming)

iptables -A FORWARD -p tcp --dport 1935 -s 192.168.1.0/24 -j REJECT

# Bloquear tráfico RTP/UDP (ejemplo rango común)

iptables -A FORWARD -p udp --dport 10000:20000 -s 192.168.1.0/24 -j REJECT

**Limitación:** muchos streams usan HTTP(S) por puerto 80/443 => no se pueden distinguir por puerto. Por eso la solución robusta es:

* **Proxy con filtrado por MIME/Content-Type** (bloquear audio/\*), o
* **DNS filtering** para bloquear dominios de radio, o
* **IDS/Layer7** (p. ej. nftables + deep packet inspection), o appliances UTM.

En Squid se puede bloquear por MIME o por URL patterns.

**4) "Del resto de compus mitad puede navegar, la otra mitad no"**

Recordemos nuestro plan:

* Permitidas: 192.168.1.102 — 192.168.1.105 (4 PCs)
* Denegadas: 192.168.1.106 — 192.168.1.110 (5 PCs)

**Reglas iptables**

Permitir navegación (masquerade + forward) para el grupo permitido:

# Permitir FORWARD para el bloque permitido

iptables -A FORWARD -s 192.168.1.102/30 -o eth0 -j ACCEPT

# y NAT para ese rango

iptables -t nat -A POSTROUTING -s 192.168.1.102/30 -o eth0 -j MASQUERADE

# Permitir también la IP 192.168.1.105 (si quedó fuera del /30)

iptables -A FORWARD -s 192.168.1.105 -o eth0 -j ACCEPT

iptables -t nat -A POSTROUTING -s 192.168.1.105 -o eth0 -j MASQUERADE

(En el ejemplo arriba uso /30 para 4 IPs; si preferís declarar IP por IP, lo hacés con varias líneas.)

Denegar todo acceso a Internet para las IPs bloqueadas:

# Rechazar FORWARD para las IPs no permitidas

iptables -A FORWARD -s 192.168.1.106 -o eth0 -j REJECT

iptables -A FORWARD -s 192.168.1.107 -o eth0 -j REJECT

iptables -A FORWARD -s 192.168.1.108 -o eth0 -j REJECT

iptables -A FORWARD -s 192.168.1.109 -o eth0 -j REJECT

iptables -A FORWARD -s 192.168.1.110 -o eth0 -j REJECT

**Importante:** las reglas ACCEPT para admin y permitidos deben ir antes de estas REJECT. El orden que propuse respeta eso (poné las reglas de permit antes de las de reject).

**5) Regla adicional (la que creo yo): Restricción horaria a redes sociales**

Propuesta: **Bloquear el acceso a redes sociales (dominios)** durante horario laboral (p. ej. 9:00–17:00) para todas las máquinas excepto el admin.

**En iptables (módulo time) — ejemplo bloque horario por IP destino (suponiendo IP conocida)**

# Ejemplo: bloquear acceso a 151.101.1.69 (ej. una IP de ejemplo de red social) de 09:00 a 17:00

iptables -A FORWARD -s 192.168.1.0/24 -d 151.101.1.69 -m time --timestart 09:00 --timestop 17:00 --weekdays Mon,Tue,Wed,Thu,Fri -j REJECT

**Limitación:** las redes sociales usan múltiples IPs/CDNs => no es eficiente solo por IP. Mejor usar **proxy (Squid) + ACL por dominio** y aplicar horario:

# squid.conf (ejemplo conceptual)

acl redes\_sociales dstdomain .facebook.com .twitter.com .instagram.com

acl horario\_trabajo time M T W Th F 09:00-17:00

http\_access deny redes\_sociales horario\_trabajo

http\_access allow localhost

Esto bloquea por dominio en ese horario y permite al admin si querés añadiendo excepciones por IP.

**5) Extras prácticos: logging, pruebas y mantenimiento**

* **Logging**: agregar reglas de LOG antes de REJECT para auditar intentos:

iptables -A FORWARD -s 192.168.1.0/24 -d 93.184.216.34 -j LOG --log-prefix "BLOQUEO\_PAG: "

iptables -A FORWARD -s 192.168.1.0/24 -d 93.184.216.34 -j REJECT

* **Testing**: desde cada PC:
  + ping 8.8.8.8 (¿responde?) — test de conectividad IP
  + curl -I https://www.ejemplo.com (¿llega? si está bloqueada por dominio, debería fallar)
  + Revisar logs del firewall (/var/log/syslog o dmesg según configuración)
* **Persistencia**: en Linux, guardar reglas (iptables-save/restore) o usar herramientas como ufw/firewalld o scripts de arranque.
* **Mejoras**: usar nftables para sintaxis más moderna; usar pfSense/OPNsense para GUI, Squid para filtrado por dominio/MIME, y optionally un servicio DNS con filtrado (Pi-hole).

**6) Cómo se redactan las reglas (guía práctica de redacción)**

1. **Definir objetivo** (qué querés permitir o bloquear, a quién, cuándo).
2. **Identificar atributos**: IP origen, IP destino o dominios, puertos, protocolo (tcp/udp), interfaz, estado (NEW/ESTABLISHED), tiempo.
3. **Orden lógico**: primeramente permitir excepciones (ej. admin), luego permitir grupos autorizados, luego bloquear lo restante. Siempre dejar paso para las respuestas (RELATED,ESTABLISHED).
4. **Especificidad sobre generalidad**: reglas específicas antes de reglas generales. Ej.: -s 192.168.1.101 -j ACCEPT antes de -s 192.168.1.0/24 -j REJECT.
5. **Documentar** cada regla (comentarios en scripts o descripciones en GUI).
6. **Probar** y **monitorear**: logs y tests automatizados.
7. **Mantener**: actualizar si cambian IPs, servicios o se detecta falsos positivos.

**7) Ejemplo de archivo de reglas (script mínimo)**

#!/bin/bash

# firewall-init.sh - ejemplo simple

# Interfaces

LAN\_IF="eth1"

WAN\_IF="eth0"

LAN\_NET="192.168.1.0/24"

# flush

iptables -F

iptables -t nat -F

# default policies

iptables -P FORWARD DROP

iptables -P INPUT ACCEPT

iptables -P OUTPUT ACCEPT

# stateful allow

iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

# admin all

iptables -A FORWARD -s 192.168.1.101 -o $WAN\_IF -j ACCEPT

iptables -t nat -A POSTROUTING -s 192.168.1.101 -o $WAN\_IF -j MASQUERADE

# allowed group (102-105)

for ip in 192.168.1.102 192.168.1.103 192.168.1.104 192.168.1.105; do

iptables -A FORWARD -s $ip -o $WAN\_IF -j ACCEPT

iptables -t nat -A POSTROUTING -s $ip -o $WAN\_IF -j MASQUERADE

done

# deny others (106-110)

for ip in 192.168.1.106 192.168.1.107 192.168.1.108 192.168.1.109 192.168.1.110; do

iptables -A FORWARD -s $ip -o $WAN\_IF -j REJECT

done

# block example webpage by IP (ej: 93.184.216.34)

iptables -A FORWARD -s $LAN\_NET -d 93.184.216.34 -j REJECT

# block streaming ports (ejemplos)

iptables -A FORWARD -p tcp --dport 8000:8001 -s $LAN\_NET -j REJECT

iptables -A FORWARD -p tcp --dport 1935 -s $LAN\_NET -j REJECT

iptables -A FORWARD -p tcp --dport 554 -s $LAN\_NET -j REJECT

iptables -A FORWARD -p udp --dport 10000:20000 -s $LAN\_NET -j REJECT

**8) Resumen rápido por punto pedido**

1. **Una puede hacer todo**: regla ACCEPT para IP admin (192.168.1.101) y NAT.
2. **Bloquear una página web**: ideal via proxy (Squid ACL); por IP con iptables -d <IP> -j REJECT (limitación: CDNs).
3. **Bloquear radio por internet**: bloquear puertos típicos streaming + usar proxy/DNS para mejores resultados.
4. **Mitad/no mitad acceso a internet**: permitir solo a 192.168.1.102-105, rechazar 192.168.1.106-110.
5. **Regla propia**: bloqueo horario a redes sociales vía Squid ACL por dominio y time, o iptables -m time (menos preciso).