

IES Valle Inclán



IPTABLES

CARLOS GONZÁLEZ MARTÍN

Contenido

1.	Configuración de red.....	3
2.	Cambiar nombre a la maquina	3
3.	Instalación de paquetes	4
4.	Iptables.....	4
5.	Comprobaciones iniciales	4
6.	Creación del archivo de configuración	6
7.	Modificación del archivo de configuración	6
8.	1ª comprobación.....	7
9.	1ª regla iptables	8
10.	2ª regla iptables.....	9
11.	3ª regla iptables.....	10
12.	Script finalizado.....	11
13.	Bastión	12
14.	Comprobación del cliente	13
15.	Conclusión	14

1. Configuración de red

Vamos a poner la maquina en adaptador puente para así conectarnos a los diferentes servicios que están en la máquina.

```
root@debian-12:~# ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:86:d4:5b brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.131/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 86089sec preferred_lft 86089sec
    inet6 2a0c:5a80:5506:ea00:a00:27ff:fe86:d45b/64 scope global dynamic mngtmpaddr
        valid_lft forever preferred_lft forever
    inet6 fe80:a00:27ff:fe86:d45b/64 scope link
        valid_lft forever preferred_lft forever
root@debian-12:~#
```

2. Cambiar nombre a la maquina

Mediante el siguiente comando cambiamos el nombre a la máquina.

```
root@debian-12:~# hostnamectl set-hostname iptables
root@debian-12:~# exit_
```

Una vez que le demos a exit e iniciamos sesión veremos que ha cambiado el nombre a la máquina.

```
iptables login: root
Password:
Linux iptables 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan  9 19:00:39 CET 2025 on tty1
root@iptables:~#
root@iptables:~#
root@iptables:~#
root@iptables:~#
root@iptables:~# _
```

3. Instalación de paquetes

Ahora instalaremos el paquete iptables y los servicios como ssh y apache2, ping no lo instalaremos por que viene de serie habilitado en debían.

```
root@iptables:~# apt update ; apt install iptables ssh apache2 -y
Des:1 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Des:2 http://deb.debian.org/debian bookworm InRelease [151 kB]
Des:3 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Des:4 http://security.debian.org/debian-security bookworm-security/main Sources [133 kB]
Des:5 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [239 kB]
Des:6 http://security.debian.org/debian-security bookworm-security/main Translation-en [141 kB]
Des:7 http://deb.debian.org/debian bookworm/non-free-firmware Sources [6,444 B]
Des:8 http://deb.debian.org/debian bookworm/non-free-firmware amd64 Packages [59,464 kB]
```

4. Iptables

Ahora vamos a comprobar si hay alguna regla de iptables habilitada.

```
Procesando disparadores para iptables-bin (2:1.6.0-3+deb12u8) ...
root@iptables:~# iptables -L -n -v
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@iptables:~# _
```

5. Comprobaciones iniciales

Ahora vamos a conectarnos con los diferentes servicios que tiene el servidor que hemos instalado anteriormente.



192.168.1.131

Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
```

Ahora vamos a comprobamos mediante ssh.

```
C:\Users\carlo>ssh usuario@192.168.1.131
The authenticity of host '192.168.1.131 (192.168.1.131)' can't be established.
ED25519 key fingerprint is SHA256:GfU/X6wIOZdPacogiPSgu4Ve8eVn+pF0gaDVcZl+DQI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.131' (ED25519) to the list of known hosts.
usuario@192.168.1.131's password:
Linux iptables 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Sep 23 09:06:25 2024
usuario@iptables:~$
```

Ahora vamos a hacer ping.

```
C:\Users\carlo>ping 192.168.1.131

Haciendo ping a 192.168.1.131 con 32 bytes de datos:
Respuesta desde 192.168.1.131: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.131: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.131: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.131: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.1.131:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\carlo>
```

6. Creación del archivo de configuración

Ahora crearemos el archivo de configuración de iptables.

```
root@iptables:~# touch iptables.sh
root@iptables:~# ls -la
total 28
drwx----- 4 root root 4096 ene  9 19:54 .
drwxr-xr-x 18 root root 4096 sep 11 12:01 ..
-rw----- 1 root root  85 ene  9 19:14 .bash_history
-rw-r--r-- 1 root root 3525 sep 16 10:09 .bashrc
-rw-r--r-- 1 root root   0 ene  9 19:54 iptables.sh
drwxr-xr-x 3 root root 4096 sep 11 12:06 .local
-rw-r--r-- 1 root root 161 jul  9 2019 .profile
drwx----- 2 root root 4096 sep 11 12:00 .ssh
root@iptables:~#
```

Ahora le daremos permisos de ejecución al archivo creado anteriormente.

```
root@iptables:~# chmod +x iptables.sh
root@iptables:~# ls -la
total 28
drwx----- 4 root root 4096 ene  9 19:54 .
drwxr-xr-x 18 root root 4096 sep 11 12:01 ..
-rw----- 1 root root  85 ene  9 19:14 .bash_history
-rw-r--r-- 1 root root 3525 sep 16 10:09 .bashrc
-rwxr-xr-x 1 root root   0 ene  9 19:54 iptables.sh
drwxr-xr-x 3 root root 4096 sep 11 12:06 .local
-rw-r--r-- 1 root root 161 jul  9 2019 .profile
drwx----- 2 root root 4096 sep 11 12:00 .ssh
root@iptables:~#
```

7. Modificación del archivo de configuración

Abrimos el archivo de configuración y haremos un borrado de reglas.

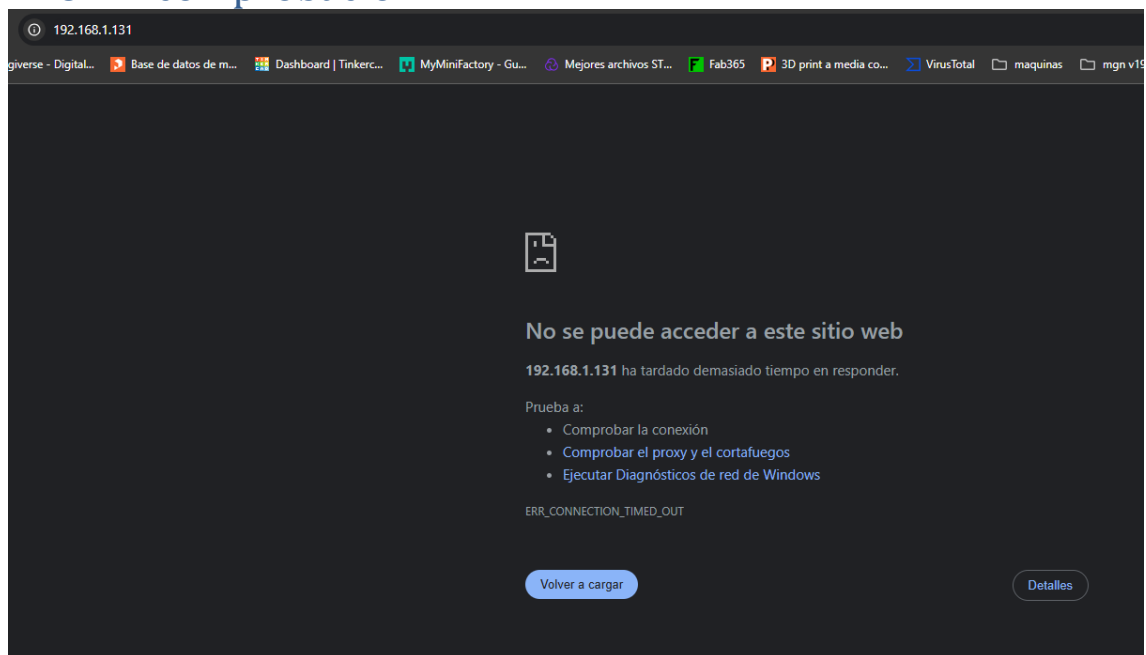
```
GNU nano 7.2
1 #!/bin/bash
2
3 # Borrado de reglas #
4 iptables -F
5 iptables -X
6 iptables -t nat -F
7
```

Ahora como vamos a denegar todo el trafico entrante pondremos las siguientes reglas.

```
GNU nano 7.2
1 #!/bin/bash
2
3 # Borrado de reglas #
4 iptables -F
5 iptables -X
6 iptables -t nat -F
7
8 # Denegar todo el trafico #
9 iptables -P INPUT DROP
10 iptables -P OUTPUT DROP
11 iptables -P FORWARD DROP
12
```

Una vez que ejecutemos el script vamos a comprobar si nos deniega todo.

8. 1ª comprobación



Ahora vamos a comprobar con un ping.

```
C:\Users\carlo>ping 192.168.1.131

Haciendo ping a 192.168.1.131 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.1.131:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
    (100% perdidos),

C:\Users\carlo>
```

Ahora vamos a conectarnos por ssh.

```
C:\Users\carlo>ssh usuario@192.168.1.131
ssh: connect to host 192.168.1.131 port 22: Connection timed out

C:\Users\carlo>
```

9. 1ª regla iptables

Ahora lo que haremos será crear la primera regla de iptables que acepte el ping de la maquina real.

```
12
13 # aceptar ping #
14 iptables -A INPUT -i enp0s3 -s 192.168.1.92 -p icmp --icmp-type echo-request -j ACCEPT
15 iptables -A OUTPUT -d 192.168.1.92 -p icmp --icmp-type echo-reply -j ACCEPT
16
```


Ahora haremos la ejecución del script y comprobaremos si funciona en la maquina real.

```
C:\Users\carlo>ping 192.168.1.131

Haciendo ping a 192.168.1.131 con 32 bytes de datos:
Respuesta desde 192.168.1.131: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.131: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.131: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.131: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.1.131:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ahora vamos a comprobar con otro equipo si nos acepta el ping.

```
C:\Users\carlo>ping 192.168.1.131

Haciendo ping a 192.168.1.131 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.1.131:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
    (100% perdidos),
```

10. 2ª regla iptables

Ahora lo que haremos será permitir solo conexiones por ssh de la maquina real.

```
16
17 #   aceptar ssh   #
18 iptables -A INPUT -s 192.168.1.92 -p tcp --dport 22 -j ACCEPT
19 iptables -A OUTPUT -d 192.168.1.92 -p tcp -j ACCEPT
20
21
```

Ahora vamos a comprobar con la maquina real.

```
C:\Users\carlo>ssh usuario@192.168.1.131
usuario@192.168.1.131's password:
Linux iptables 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan  9 19:30:32 2025 from 192.168.1.92
usuario@iptables:~$
```

Ahora vamos a comprobar con otro equipo de la red.

```
C:\Users\carlo>ssh usuario@192.168.1.131
ssh: connect to host 192.168.1.131 port 22: Connection timed out
```

11. 3ª regla iptables

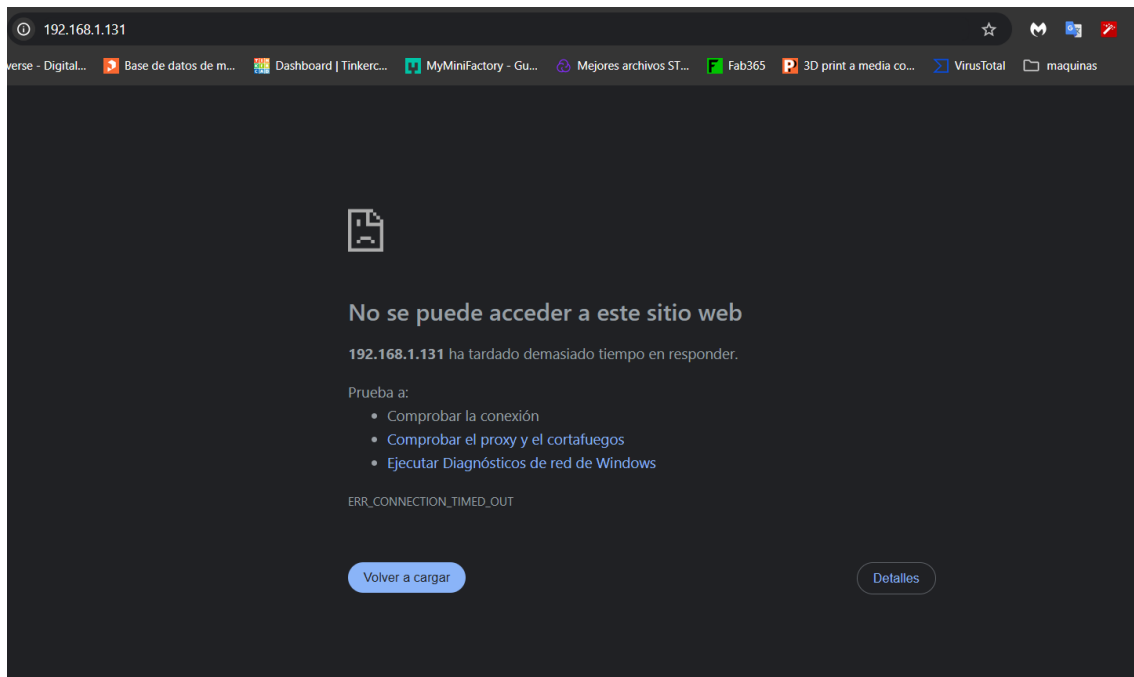
Ahora lo que haremos será aceptar que podamos ver la pagina web en la maquina real.

```
21 # aceptar web #
22 iptables -A INPUT -s 192.168.1.92 -p tcp --dport 80 -j ACCEPT
23 iptables -A OUTPUT -d 192.168.1.92 -p tcp -j ACCEPT
24
```

Ejecutamos el script y comprobaremos la maquina real.



Usando otro equipo comprobaremos.



12. Script finalizado

Ahora vemos todas las reglas que hemos creado.

```
GNU nano 7.2 iptables.sh
1 #!/bin/bash
2
3 # Borrado de reglas #
4 iptables -F
5 iptables -X
6 iptables -t nat -F
7
8 # Denegar todo el trafico #
9 iptables -P INPUT DROP
10 iptables -P OUTPUT DROP
11 iptables -P FORWARD DROP
12
13 # aceptar ping #
14 iptables -A INPUT -i enp0s3 -s 192.168.1.92 -p icmp --icmp-type echo-request -j ACCEPT
15 iptables -A OUTPUT -d 192.168.1.92 -p icmp --icmp-type echo-reply -j ACCEPT
16
17 # aceptar ssh #
18 iptables -A INPUT -s 192.168.1.92 -p tcp --dport 22 -j ACCEPT
19 iptables -A OUTPUT -d 192.168.1.92 -p tcp -j ACCEPT
20
21 # aceptar web #
22 iptables -A INPUT -s 192.168.1.92 -p tcp --dport 80 -j ACCEPT
23 iptables -A OUTPUT -d 192.168.1.92 -p tcp -j ACCEPT
24
25
```

Ahora vamos a hacer un iptables -L -n -v y vemos las reglas creadas.

```

root@iptables:~# iptables -L -n -v
Chain INPUT (policy DROP 6 packets, 596 bytes)
 pkts bytes target     prot opt in     out     source               destination           icmptype
  0      0 ACCEPT     1    --  enp0s3 *      192.168.1.92         0.0.0.0/0             icmp type 8
  0      0 ACCEPT     6    --  *      *      192.168.1.92         0.0.0.0/0             tcp dpt:22
  0      0 ACCEPT     6    --  *      *      192.168.1.92         0.0.0.0/0             tcp dpt:80

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

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination           icmptype
  0      0 ACCEPT     1    --  *      *      0.0.0.0/0            192.168.1.92          icmp type 0
  0      0 ACCEPT     6    --  *      *      0.0.0.0/0            192.168.1.92
  0      0 ACCEPT     6    --  *      *      0.0.0.0/0            192.168.1.92
root@iptables:~#

```

13. Bastión

Ahora lo que haremos será usar este equipo como bastión, para que el cliente que esta en una interfaz en modo red interna pueda comunicarse con el exterior.

Este es el archivo de configuración de enrutador.

```

GNU nano 7.2 enrutado.sh
1 #!/bin/bash
2
3 echo "1" > /proc/sys/net/ipv4/ip_forward
4 iptables -A FORWARD -j ACCEPT
5 iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.0.0/24 -j MASQUERADE
6

```

Ahora tenemos que poner la segunda interfaz en red interna y escribir las IPs.

```

GNU nano 7.2 /etc/network/interfaces
1 # This file describes the network interfaces available on your system
2 # and how to activate them. For more information, see interfaces(5).
3
4 source /etc/network/interfaces.d/*
5
6 # The loopback network interface
7 auto lo
8 iface lo inet loopback
9
10 # The primary network interface
11 allow-hotplug enp0s3 enp0s8
12 iface enp0s3 inet dhcp
13
14 iface enp0s8 inet static
15     address 192.168.0.224
16     netmask 255.255.255.0
17

```

14. Comprobación del cliente

Ahora vamos a configurar las IPs en el cliente y probaremos ya que anteriormente ejecutamos el script.

Obtener una dirección IP automáticamente

☒ Usar la siguiente dirección IP:

Dirección IP: 192 . 168 . 0 . 20

Máscara de subred: 255 . 255 . 255 . 0

Puerta de enlace predeterminada: 192 . 168 . 0 . 224

Obtener la dirección del servidor DNS automáticamente

☒ Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido: 8 . 8 . 8 . 8

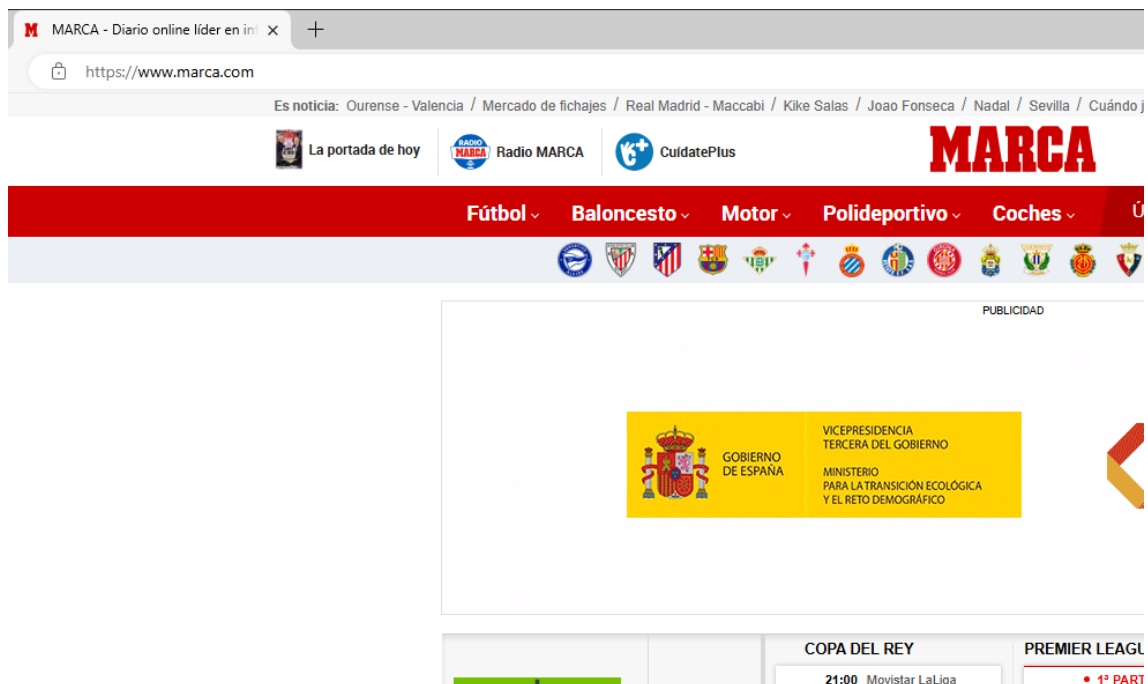
Servidor DNS alternativo: . . .

☐ Validar configuración al salir

Opciones avanzadas...

Aceptar Cancelar

Ahora vamos al navegador y buscamos alguna pagina web, en mi caso es marca.com.



15. Conclusión

Al implementar reglas en iptables para denegar el tráfico HTTP, SSH y ICMP, se ha logrado un primer nivel de protección contra ataques comunes.

La creación de una cadena personalizada para permitir el acceso al equipo 192.168.1.92 a través de SSH ha demostrado la potencia de iptables para establecer políticas de acceso granulares. Durante la práctica, se observó la importancia del orden de las reglas y la necesidad de realizar pruebas exhaustivas para evitar bloquear el tráfico legítimo.

Esta experiencia ha consolidado mis conocimientos sobre iptables y me ha permitido comprender la importancia de una configuración precisa y segura del firewall.