DÍA 13: CONFIGURACIÓN Y ADMINISTRACIÓN DE UN SERVIDOR DHCP EN LINUX

Por Daniel Ariza
04/07/2025

Fase 1: Instalación del servidor de red y servicios básicos

- **Instalar dnsmasq como servicio combinado de DHCP y DNS ligero.**

PASO 1: Instalar dnsmasq

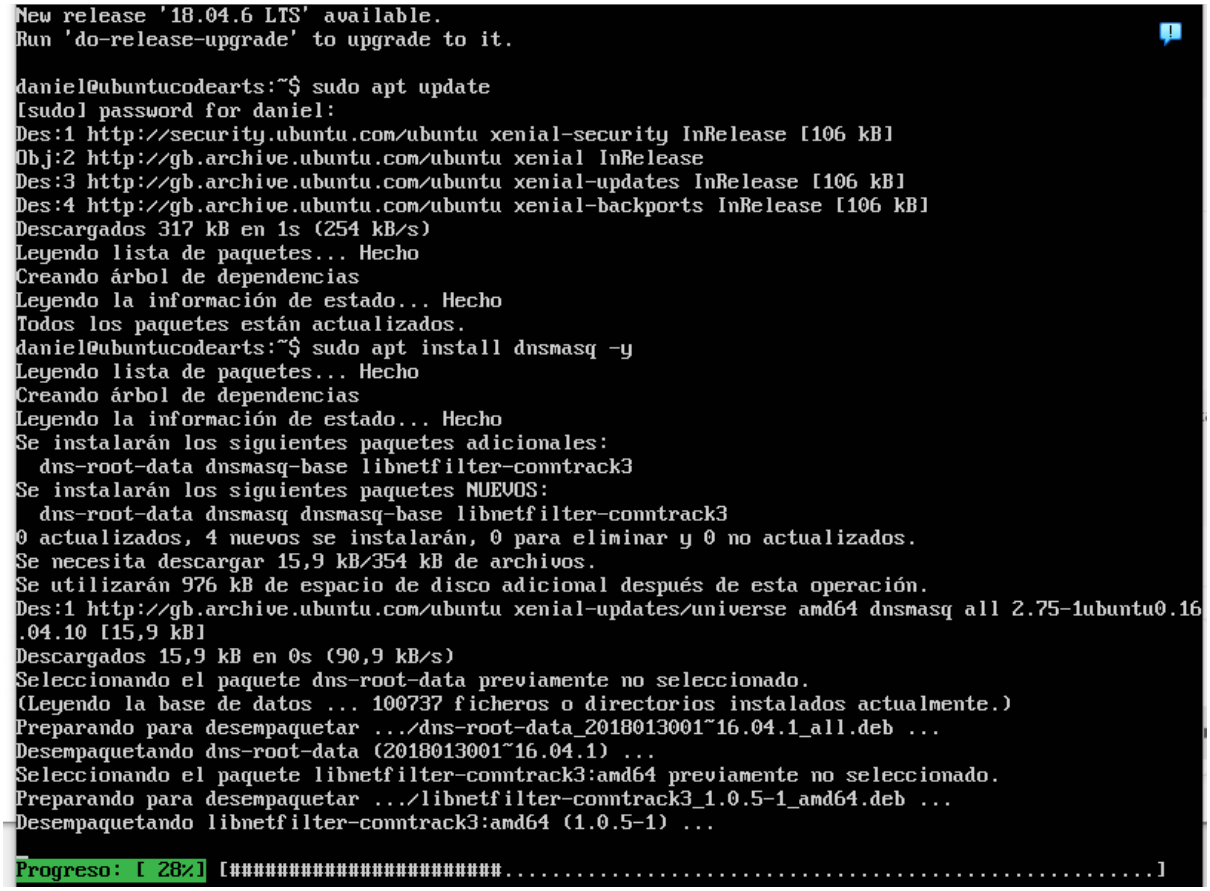
Abre tu terminal y ejecuta:

```
sudo apt update
sudo apt install dnsmasq -y
```

Una vez instalado, verifica que el servicio se ha iniciado:

```
systemctl status dnsmasq
```

Si todo va bien, debería mostrar algo como: active (running)



PASO 2: Detener el servicio mientras lo configuramos

sudo systemctl stop dnsmasq

PASO 3: Hacer copia de seguridad del archivo original

sudo cp /etc/dnsmasq.conf /etc/dnsmasq.conf.original

PASO 4: Editar la configuración principal

Editamos el archivo principal:

sudo nano /etc/dnsmasq.conf

Dentro del archivo, puedes dejar solo estas líneas activas (borra el resto o comenta con #):

# INTERFAZ INTERNA
interface=ens33 # Sustituye por la interfaz de red interna

DESACTIVAR DNS EN OTRAS INTERFACES
bind-interfaces

RANGO DE IPs QUE ASIGNARÁ POR DHCP
dhcp-range=192.168.100.50,192.168.100.150,12h

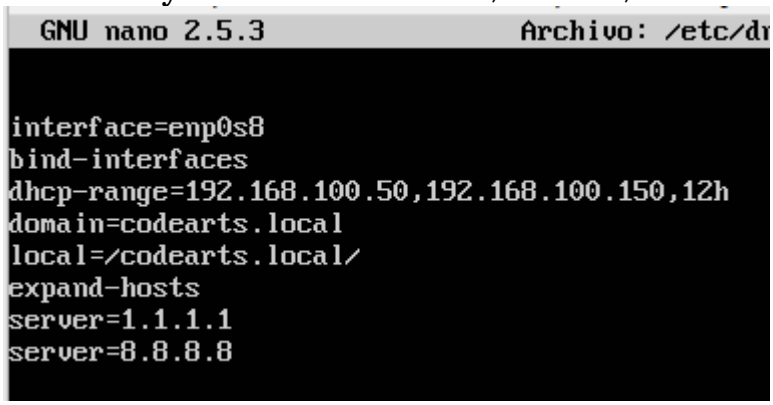
# DIRECCIÓN DEL SERVIDOR DNS LOCAL
domain=lab.local
local=/lab.local/
expand-hosts

# FORWARD DNS (opcional, para salir a Internet)
server=1.1.1.1
server=8.8.8.8

Notas:

- Sustituye enp0s8 por el nombre real de tu interfaz interna (usa ip a para verlo).

- Puedes cambiar codearts.local por el dominio que estés usando.
  Guarda y cierra: Ctrl + O, Enter, Ctrl + X



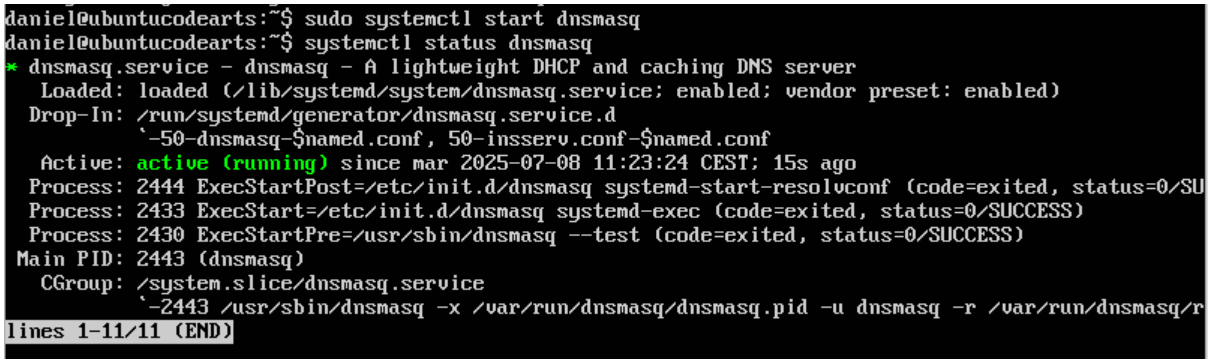
Paso 4: Reiniciar dnsmasq

Después de editar:

sudo systemctl restart dnsmasq

Y confirma que está funcionando:

systemctl status dnsmasq



PASO 5 : Verificar que dnsmasq está sirviendo DHCP

Podemos usar estos comandos:

cat /var/lib/misc/dnsmasq.leases

Esto te mostrará los dispositivos que han recibido IP. En cuanto conectes un cliente en red interna con DHCP activado, debería aparecer aquí.

- **Configurar el servidor para que escuche solo en la interfaz LAN.**

Paso 1: Identificar el nombre exacto de la interfaz LAN

Ejecuta: ip a

Busca la interfaz que tenga una IP del tipo 192.168.100.x (según tu red interna anterior). Por ejemplo, puede llamarse enp0s8, eth0, ens33, etc. En nuestro caso es enp0s8.

Apunta ese nombre exacto, porque lo usaremos en la configuración.

```
daniel@ubuntucodearts:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:ee:ee:8c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.40/24 brd 192.168.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feee:ee8c/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:65:33:ec brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.1/24 brd 192.168.100.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe65:33ec/64 scope link
       valid_lft forever preferred_lft forever
daniel@ubuntucodearts:~$
```

Paso 2: Editar /etc/dnsmasq.conf para que escuche solo en esa interfaz

Ejecuta:

sudo nano /etc/dnsmasq.conf

Y asegúrate de que estén estas dos líneas, ajustando el nombre de la interfaz:

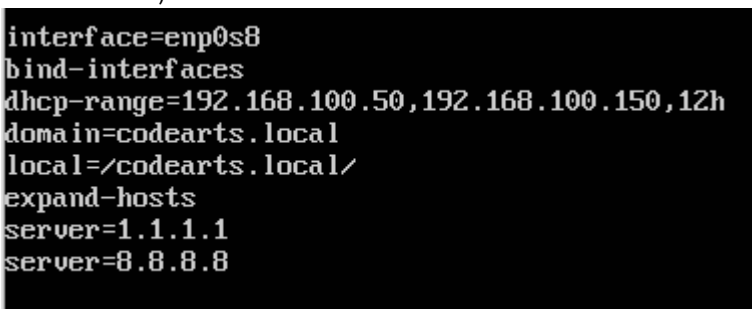
interface=enp0s8 # Reemplaza por el nombre de tu interfaz interna

bind-interfaces

¿Qué hacen estas líneas?

- interface= indica en qué interfaz debe escuchar dnsmasq.

- bind-interfaces fuerza a que solo escuche en esa interfaz, incluso si el sistema tiene varias.



Paso 3: Reiniciar dnsmasq

sudo systemctl restart dnsmasq

Comprueba que esté activo: sudo systemctl status dnsmasq

Paso 4 (opcional): Verificar con netstat que solo escucha en LAN

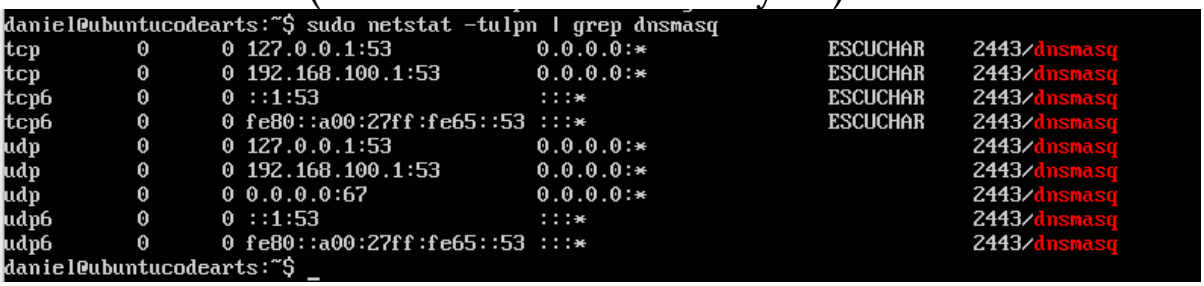
Instala net-tools si no lo tienes:

sudo apt install net-tools

Luego:

sudo netstat -tulpn | grep dnsmasq

Deberías ver que dnsmasq está escuchando solo en la IP asociada a LAN (como 192.168.100.1:53 y 67).



- Está escuchando en localhost (127.0.0.1) correcto para uso interno del sistema.
- Está escuchando en LAN interna 192.168.100.1 lo que esperábamos.
- NO está escuchando en la interfaz externa (192.168.1.40) perfecto, porque limitamos el servicio con interface=enp0s8 y bind-interfaces.

- **Asignar un rango IP específico por cada grupo/departamento (ej: desarrollo, diseño, administración).**

Estructura de prueba

Rol	Sistema	Función
Servidor DHCP/DNS	Ubuntu Server Core (VM)	dnsmasq configurado
Cliente 1	Windows Server Core (VM)	Obtiene IP por DHCP
Cliente 2	Ubuntu Desktop (host real)	Obtiene IP por DHCP

Ventajas de este enfoque

- Pruebas multiplataforma reales (Linux y Windows como clientes).

- Puedes observar cómo se comporta dnsmasq con distintos clientes.
- Te permite capturar MACs reales, asignar IPs por grupo y comprobar nombres DNS locales.

- Perfecto para seguir aprendiendo con lógica de red.

PASO 1: Obtener la MAC de los clientes

A. Cliente: Windows Server Core (VM)

En la VM de Windows Server Core, ejecuta:

Get-NetAdapter

Te devolverá algo como:

Name : Ethernet

InterfaceDescription : Intel(R) PRO/1000 MT...

MacAddress : 00-0C-29-XX-XX-XX

Status : Up

Anótala (cámbiale los - por : al estilo Linux), por ejemplo:

00:0C:29:AB:CD:EF

B. Cliente: Ubuntu Desktop (host real)

En la terminal del host, ejecuta:

ip a

Busca la interfaz que esté conectada a la red interna. Aparecerá algo como:

3: enp3s0: <...>

link/ether 34:23:87:ab:cd:ef brd ff:ff:ff:ff:ff:ff

Copia la dirección MAC:

PASO 2: Editar /etc/dnsmasq.conf y añadir asignaciones por grupo

Ahora en tu Ubuntu Server Core, edita:

sudo nano /etc/dnsmasq.conf

Y añade al final algo como esto (reemplaza las MACs reales por las que acabas de obtener):

Rango por defecto para cualquier cliente no identificado

dhcp-range=192.168.100.120,192.168.100.150,12h

=== GRUPO DESARROLLO ===

dhcp-host=00:0C:29:AB:CD:EF,set:desarrollo

dhcp-range=tag:desarrollo,192.168.100.50,192.168.100.69,12h

=== GRUPO DISEÑO ===

dhcp-host=34:23:87:AB:CD:EF,set:diseno

dhcp-range=tag:diseno,192.168.100.70,192.168.100.89,12h

Explicación:

- dhcp-host=MAC,set:grupo → etiqueta al cliente según su MAC.

- dhcp-range=tag:grupo,... → define el rango IP que solo ese grupo puede recibir.
  Guarda y cierra: Ctrl + O, Enter, Ctrl + X.

```
interface=enp0s8
bind-interfaces
dhcp-range=192.168.100.50,192.168.100.150,12h
domain=codearts.local
local=/codearts.local/
expand-hosts
server=1.1.1.1
server=8.8.8.8

# Rango por defecto para cualquier cliente no identificado
dhcp-range=192.168.100.120,192.168.100.150,12h

# Grupo desarrollo
dhcp-host=08:00:27:89:C2:6E,set:desarrollo
dhcp-range=tag:desarrollo,192.168.100.50,192.168.100.69,12h

# Grupo diseno
dhcp-host=74:d4:35:b5:99:05,set:diseno
dhcp-range=tag:diseno,192.168.100.70,192.168.100.89,12h
```

PASO 3: Reiniciar dnsmasq

sudo systemctl restart dnsmasq

PASO 4: Comprobar que se ha asignado una IP con el rango establecido a Windows Server.

Ejecuta en Windows Server: ipconfig

```
Adaptador de Ethernet Ethernet 2:

   Sufijo DNS específico para la conexión. . : codearts.local
   Vínculo: dirección IPv6 local. . . : fe80::9d9a:36de:a7f9:aba0%7
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.100.64
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.100.1
PS C:\Users\daniel>
```

Vemos que tiene la IP en el rango establecido en Ubuntu Server.

Ejecuta en Ubuntu Server:

cat /var/lib/misc/dnsmasq.leases

```
daniel@ubuntucodearts:~$ cat /var/lib/misc/dnsmasq.leases
1752022424 08:00:27:89:c2:6e 192.168.100.64 WIN-SERVER-DANIELARIZA 01:08:00:27:89:c2:6e
daniel@ubuntucodearts:~$ _
```

dnsmasq está funcionando perfectamente como servidor DHCP.

- **Establecer reserva de IPs por MAC según el tipo de dispositivo (PC, impresoras, cámaras IP...).**

Vamos a hacer una reserva de IP por Mac para mi Windows Server. Añadimos esto al archivo /etc/dnsmasq.conf:

dhcp-host=08:00:27:89:C2:6E,pc-desarrollo-,192.168.100.65

```
dhcp-host=08:00:27:89:C2:6E,pc-desarrollo,192.168.100.65
```

Reiniciar dnsmasq para aplicar cambios:

sudo systemctl restart dnsmasq

Desde el cliente Windows server ejecutamos ipconfig

```
Adaptador de Ethernet Ethernet 2:

   Sufijo DNS específico para la conexión. . : codearts.local
   Vínculo: dirección IPv6 local. . . : fe80::9d9a:36de:a7f9:aba0%7
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.100.65
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.100.1
PS C:\Users\daniel> _
```

Aquí vemos como la ip reservada es la correcta.

Fase 2: Control de acceso básico

- **Crear una lista blanca de dispositivos autorizados (por MAC) y rechazar los no registrados.**

Definimos las MAC autorizadas usando dhcp-host=MAC,....

Activamos la opción dhcp-ignore=tag:!autorizado para bloquear todo lo que no tenga la etiqueta autorizado.
1. Editar /etc/dnsmasq.conf con sudo nano /etc/dnsmasq.conf
2. Añade la siguiente estructura:

# Activar DHCP en la interfaz interna
interface=enp0s8

Lista blanca: solo dispositivos etiquetados como "autorizado"
dhcp-ignore=tag:!autorizado

Rango general (solo se aplica si el dispositivo tiene la etiqueta)
dhcp-range=tag:autorizado,192.168.100.50,192.168.100.99,12h

Dispositivos autorizados
dhcp-host=08:00:27:89:C2:6E,set:autorizado,192.168.100.65,pc-desarrollo

```
# Lista blanca: solo dispositivos etiquetados como "autorizado"
dhcp-ignore=tag: !autorizado

# Rango general (solo se aplica si el dispositivo tiene la etiqueta)
dhcp-range=tag:autorizado,192.168.100.50,192.168.100.99,12h

# Dispositivos autorizados
dhcp-host=08:00:27:89:C2:6E,set:autorizado,192.168.100.65,pc-desarrollo
```

set:autorizado etiqueta cada MAC como permitida.
dhcp-ignore=tag:!autorizado **ignora cualquier MAC no etiquetada**.
3. Reinicia dnsmasq para aplicar los cambios:
sudo systemctl restart dnsmasq

- **Establecer nombres simbólicos para cada cliente usando DHCP+DNS local.**

Editamos el archivo /etc/dnsmasq.conf e incluimos la siguiente línea:

```
#Lista blanca con nombres simbolicos
dhcp-host=08:00:27:89:C2:6E,pc-desarrollo,192.168.100.65
```

Estás diciendo:

> A la MAC 08:00:27:89:C2:6E le das la IP 192.168.100.65
> Y le asignas el hostname pc-desarrollo
> dnsmasq también lo registrará para resolución DNS

Reiniciamos el servicio.

```
daniel@ubuntucodearts:~$ ping pc-desarrollo
PING pc-desarrollo (192.168.100.65) 56(84) bytes of data.
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=1 ttl=128 time=0.532 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=2 ttl=128 time=0.366 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=3 ttl=128 time=0.487 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=4 ttl=128 time=0.350 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=5 ttl=128 time=0.496 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=6 ttl=128 time=0.505 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=7 ttl=128 time=0.378 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=8 ttl=128 time=0.372 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=9 ttl=128 time=0.394 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=10 ttl=128 time=0.366 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=11 ttl=128 time=0.324 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=12 ttl=128 time=0.319 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=13 ttl=128 time=0.353 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=14 ttl=128 time=0.314 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=15 ttl=128 time=0.359 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=16 ttl=128 time=0.380 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=17 ttl=128 time=0.313 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=18 ttl=128 time=0.290 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=19 ttl=128 time=0.622 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=20 ttl=128 time=0.440 ms
64 bytes from pc-desarrollo.codearts.local (192.168.100.65): icmp_seq=21 ttl=128 time=0.514 ms
```

Ahora gracias al nombre simbólico establecido podemos referirnos al cliente por su nombre sin necesidad de escribir siempre la ip.

- **Aplicar reglas en iptables para permitir tráfico solo desde direcciones IP asignadas.**

1. Verifica tu interfaz de red interna

ip a | grep enp

```
daniel@ubuntucodearts:~$ ip a | grep enp
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    inet 192.168.1.40/24 brd 192.168.1.255 scope global enp0s3
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    inet 192.168.100.1/24 brd 192.168.100.255 scope global enp0s8
daniel@ubuntucodearts:~$
```

2. Limpia las reglas actuales (solo si es un entorno de pruebas)

sudo iptables -F

sudo iptables -X

3. Acepta conexiones desde las IPs autorizadas (ej. DHCP de dnsmasq)

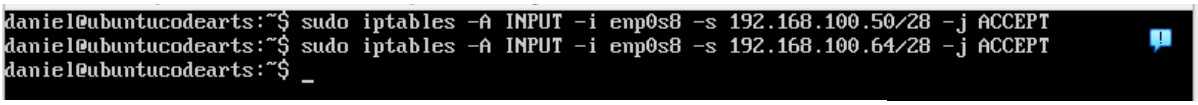
Permitir tráfico de clientes autorizados

sudo iptables -A INPUT -i enp0s8 -s 192.168.100.50/28 -j ACCEPT

sudo iptables -A INPUT -i enp0s8 -s 192.168.100.64/28 -j ACCEPT

Explicación:

192.168.100.50/28 cubre de .50 a .63
192.168.100.64/28 cubre de .64 a .79
Puedes añadir más rangos si es necesario (como .80 a .99 → 192.168.100.80/28)

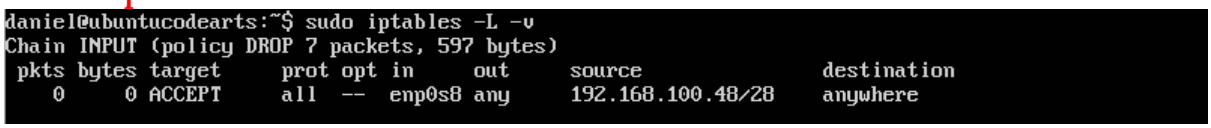


4. Rechaza el resto del tráfico en esa interfaz

sudo iptables -A INPUT -i enp0s8 -j DROP

5. Verifica las reglas activas

sudo iptables -L -v



Guardar reglas para que persistan

Instala el paquete iptables-persistent (si aún no lo tienes):

sudo apt install iptables-persistent

Y guarda las reglas activas:

sudo netfilter-persistent save

Fase 3: Integración con clientes y simulación de red

- **Configurar varios clientes Linux simulados (o virtuales) con NIC configurada en DHCP.**

En nuestro caso configuraremos nuestro Windows Server que es el que tenemos disponible virtualizado.

PASO 1: Ver interfaces de red disponibles

En tu máquina Windows Server Core, abre PowerShell y ejecuta:

```
Get-NetAdapter
```

Te devolverá algo como:

```
Name                 InterfaceDescription                Status
MacAddress

Ethernet             Intel(R) PRO/1000 MT                Up
08-00-27-89-C2-6E
```

Apunta el nombre exacto del adaptador (por ejemplo, Ethernet).

PASO 2: Habilitar DHCP en IPv4 para esa interfaz

Reemplaza Ethernet por el nombre real de tu adaptador:

```
Set-NetIPInterface -InterfaceAlias "Ethernet 2" -Dhcp
Enabled
```

Set-DnsClientServerAddress -InterfaceAlias "Ethernet 2" -ResetServerAddresses

```
PS C:\Users\daniel> Get-NetAdapter

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
Ethernet 2                Intel(R) PRO/1000 MT Desktop Adapter #2       7 Up           08-00-27-89-C2-6E         1 Gbps
Ethernet                  Intel(R) PRO/1000 MT Desktop Adapter          3 Up           08-00-27-31-48-EE         1 Gbps


PS C:\Users\daniel> Set-NetIPInterface -InterfaceAlias "Ethernet 2" -Dhcp Enabled
PS C:\Users\daniel> Set-DnsClientServerAddress -InterfaceAlias "Ethernet 2" -ResetServerAddresses
```

Esto activa DHCP tanto para IP como para servidores DNS.

PASO 3: Renovar la IP asignada

ipconfig /release

ipconfig /renew

PASO 4: Comprobar que recibió IP del servidor dnsmasq

ipconfig

```
PS C:\Users\daniel> ipconfig

Configuración IP de Windows


Adaptador de Ethernet Ethernet:

   Sufijo DNS específico para la conexión. . : home
   Vínculo: dirección IPv6 local. . . : fe80::f953:a86d:9493:b03a%3
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.1.96
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.1.1

Adaptador de Ethernet Ethernet 2:

   Sufijo DNS específico para la conexión. . : codearts.local
   Vínculo: dirección IPv6 local. . . : fe80::9d9a:36de:a7f9:aba0%7
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.100.65
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.100.1
PS C:\Users\daniel>
```

Busca una dirección del rango 192.168.100.X. Si configuraste reserva por MAC, por ejemplo:

IPv4 Address. . . . . : 192.168.100.65

Default Gateway . . . . . : 192.168.100.1

PASO 5: Verificar que el DNS también está funcionando
nslookup ubuntu-host

(Suponiendo que hayas definido en dnsmasq que 192.168.100.10 es ubuntu-host)

También puedes hacer ping por nombre simbólico: ping ubuntu-host

```
PS C:\Users\daniel> Clear-DnsClientCache
PS C:\Users\daniel> nslookup pc-desarrollo
Servidor:  UnKnown
Address:  192.168.100.1

*** UnKnown no encuentra pc-desarrollo: Non-existent domain
PS C:\Users\daniel> nslookup pc-desarrollo.codearts.local
Servidor:  UnKnown
Address:  192.168.100.1

Nombre:  pc-desarrollo.codearts.local
Address:  192.168.100.65

PS C:\Users\daniel> ping pc-desarrollo

Haciendo ping a pc-desarrollo.codearts.local [192.168.100.65] con 32 bytes de datos:
Respuesta desde 192.168.100.65: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.100.65: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.100.65: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.100.65: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.100.65:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
PS C:\Users\daniel>
```

- **Simular diferentes dispositivos por grupo (3 por grupo mínimo).**

Este punto lo hemos realizado solo con un Windows Server que es el que tenemos virtualizado y disponible.

- **Verificar que cada dispositivo reciba su IP asignada correctamente y acceda a red.**

Ya está realizado, hemos comprobado en las capturas que recibe su ip de forma correcta y que tiene acceso a la red.

Fase 4: Supervisión de tráfico e identificación

- **Instalar y configurar tcpdump para capturar el tráfico DHCP/DNS y comprobar su comportamiento.**

En mi caso tcpdump ya está instalado y teniendo en cuenta que nuestra interfaz de red es enp0s8 vamos a:

Capturar tráfico DHCP

DHCP usa puerto UDP 67 (servidor) y 68 (cliente)

Para ver solo tráfico DHCP:

sudo tcpdump -i enp0s8 port 67 or port 68 -n -vv

Qué significa:

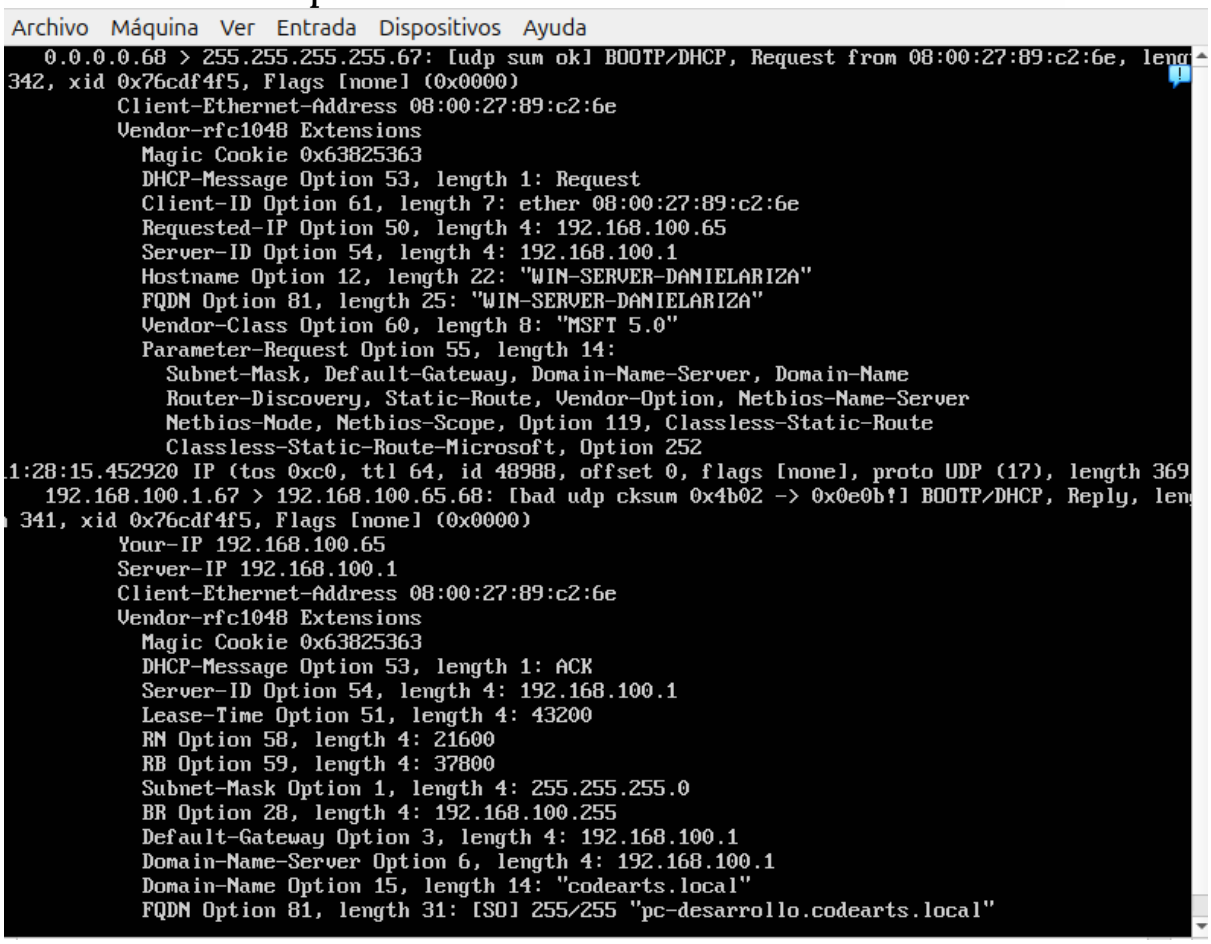
- -i enp0s8: escucha en esa interfaz
- port 67 or port 68: filtra solo paquetes DHCP
- -n: no resuelve IPs a nombres
- -vv: muy verbose (más detalle)

Lanza este comando y, desde tu Windows Server, haz:

ipconfig /release

ipconfig /renew

Y verás en tiempo real cómo se hace la solicitud de IP



Capturar tráfico DNS

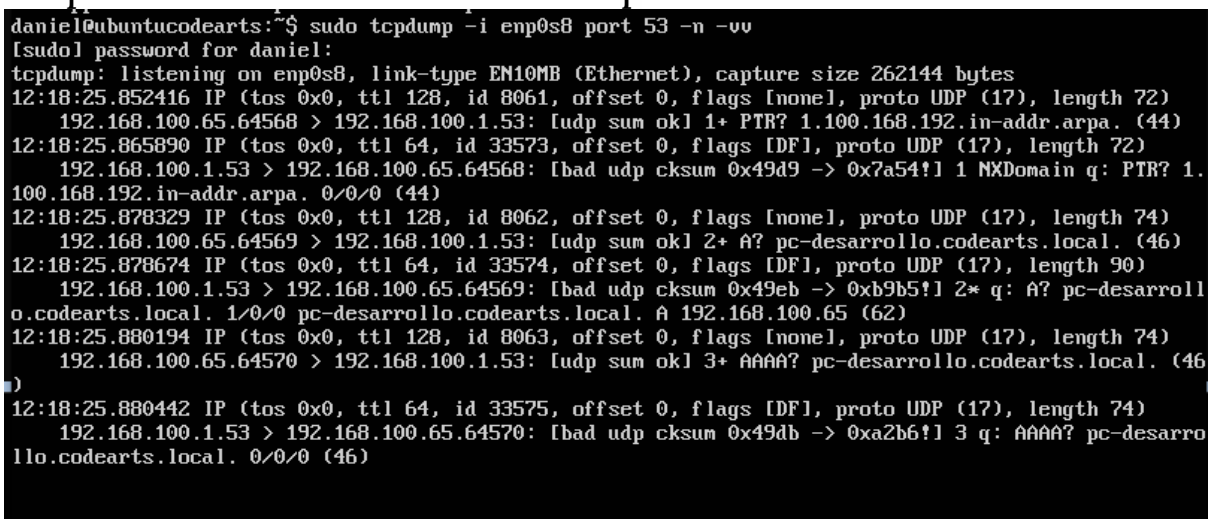
DNS usa el puerto UDP 53:

sudo tcpdump -i enp0s8 port 53 -n -vv

Y en Windows:

nslookup pc-desarrollo

Deberías ver la solicitud DNS saliendo del cliente y la respuesta del servidor dnsmasq.



- **Analizar los logs generados por dnsmasq y /var/log/syslog.**

PASO 1: Asegúrate de que dnsmasq genera logs

Abre el archivo de configuración:

sudo nano /etc/dnsmasq.conf

Y verifica (o añade) esta línea para asegurar que dnsmasq registre actividad detallada:

log-queries

log-dhcp

PASO 2: Revisar logs de dnsmasq en /var/log/syslog

Si no has definido un log separado, por defecto dnsmasq escribe en el log general del sistema:

sudo journalctl -u dnsmasq.service -f

Esto mostrará **solo los logs de** dnsmasq **en tiempo real** (filtrados por servicio).

- **Usar arp-scan para ver los dispositivos conectados a la red y su relación IP/MAC.**

PASO 1: Instalar arp-scan

sudo apt update

sudo apt install arp-scan -y

Verifica la instalación:

arp-scan --version

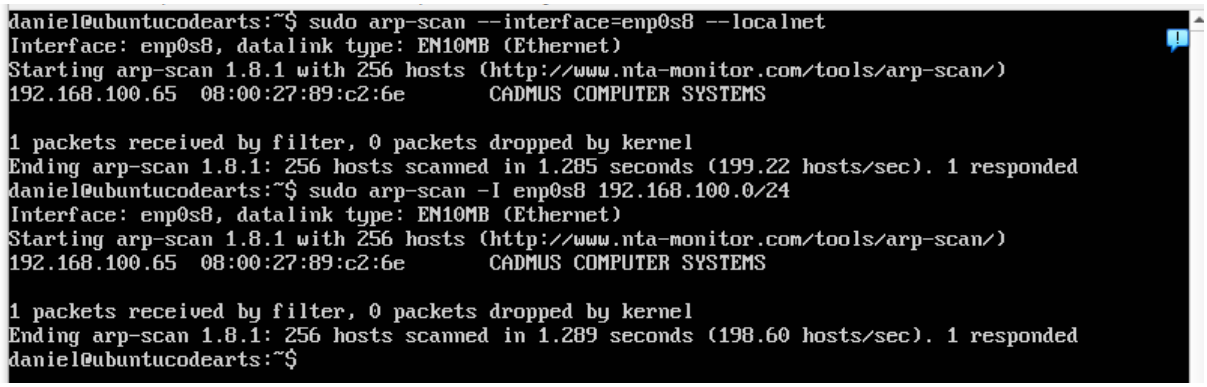
PASO 2: Escanear la red local

Ejecuta:

sudo arp-scan --interface=enp0s8 --localnet

Esto buscará en la subred a la que pertenece enp0s8.
También puedes forzarlo con el rango:

sudo arp-scan -I enp0s8 192.168.100.0/24



Interpretación

- Solo un dispositivo respondió en la red: el cliente Windows Server (con la MAC que definimos en dnsmasq.conf).
- IP = 192.168.100.65 ➜ correcta según reserva
- MAC = 08:00:27:89:c2:6e ➜ correcta según dnsmasq.conf
- Fabricante: CADMUS COMPUTER SYSTEMS = asignación genérica usada por VirtualBox