

Gestión de red inteligente con servidor de direccionamiento y control de acceso

1. Instalación del servidor de red y servicios básicos.

- **Instalar dnsmasq como servicio combinado de DHCP y DNS ligero.**
- **Configurar el servidor para que escuche solo la interfaz LAN.**
- **Asignar un rango IP específico por cada grupo:** Desarrollo, diseño y administración.
- **Establecer reserva de IPs por MAC según el tipo de dispositivo.**

2. Control de acceso básico.

- **Crear una lista blanca de dispositivos autorizados por MAC y rechazar los no registrados.**
- **Establecer nombres simbólicos para cada cliente usando DHCP+DNS local.**
- **Aplicar reglas en iptables:** Para permitir tráfico desde direcciones IP asignadas.

3. Integración con clientes y simulación de red.

- **Configurar varios clientes linux con NIC configurada en DHCP.**
- **Simular diferentes dispositivos por grupo**
- **Verificar que cada dispositivo reciba su IP asignada correctamente y acceda a la red.**

4. Supervisión de tráfico e identificación.

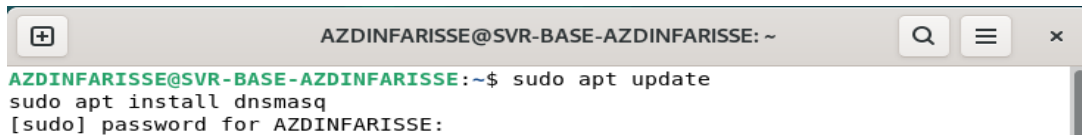
- **Instalar y configurar tcpdump:** Para capturar el tráfico DHCP/DNS y comprobar su comportamiento.
 - **Analizar los logs generados por dnsmasq y /var/log/syslog.**
 - **Usar arp-scan:** Para ver los dispositivos conectados a la red y su relación IP/MAC.
-

1. Instalación del servidor de red y servicios básicos.

Para instalar **dnsmasq**:

sudo apt update

sudo apt install dnsmasq



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~  
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo apt update  
sudo apt install dnsmasq  
[sudo] password for AZDINFARISSE:
```

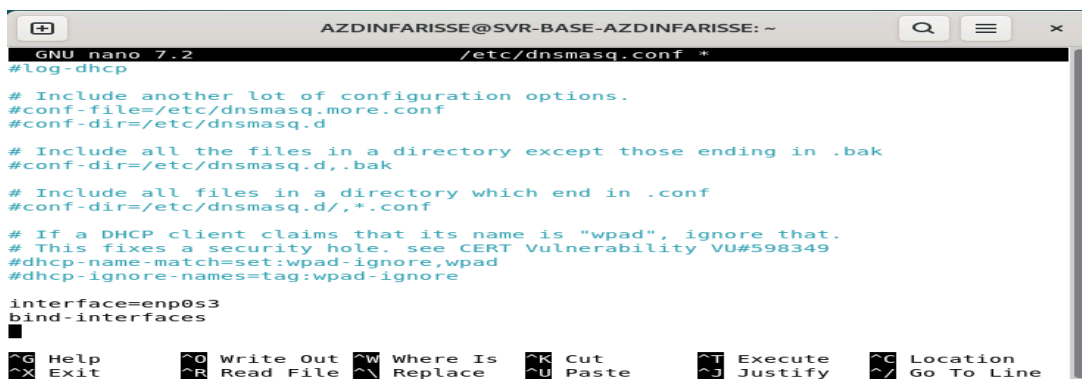
Para que escuche solo en lan introducimos el comando **ip a** y detectamos la interfaz LAN. En mi caso es la **enp0s3**. Ahora editamos el archivo de configuración de dnsmasq:

sudo nano /etc/dnsmasq.conf

Y le añadimos:

interface=enp0s3

bind-interfaces



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~  
GNU nano 7.2 /etc/dnsmasq.conf *  
#log-dhcp  
  
# Include another lot of configuration options.  
#conf-file=/etc/dnsmasq.more.conf  
#conf-dir=/etc/dnsmasq.d  
  
# Include all the files in a directory except those ending in .bak  
#conf-dir=/etc/dnsmasq.d,.bak  
  
# Include all files in a directory which end in .conf  
#conf-dir=/etc/dnsmasq.d/,*.conf  
  
# If a DHCP client claims that its name is "wpad", ignore that.  
# This fixes a security hole. see CERT Vulnerability VU#598349  
#dhcp-name-match=set:wpad-ignore,wpad  
#dhcp-ignore-names=tag:wpad-ignore  
  
interface=enp0s3  
bind-interfaces  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

Ahora para asignar un rango ip por cada departamento lo haremos añadiendo las siguientes líneas al archivo de dnsmasq.conf

Desarrollo:

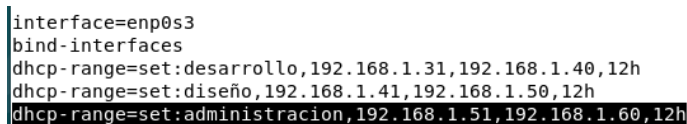
dhcp-range=set:desarrollo,192.168.1.31,192.168.1.40,12h

Diseño:

dhcp-range=set:diseño,192.168.1.41,192.168.1.50,12h

Administración:

dhcp-range=set:administracion,192.168.1.51,192.168.1.60,12h



```
interface=enp0s3  
bind-interfaces  
dhcp-range=set:desarrollo,192.168.1.31,192.168.1.40,12h  
dhcp-range=set:diseño,192.168.1.41,192.168.1.50,12h  
dhcp-range=set:administracion,192.168.1.51,192.168.1.60,12h
```

Para reservar IPs por MAC se añadiría al archivo de configuración las siguientes líneas:

PC de desarrollo:

dhcp-host=AA:BB:CC:11:22:33,192.168.1.32,pc-desarrollo-01

Impresora en administración:

dhcp-host=DD:EE:FF:44:55:66,192.168.1.52,imp-administracion-01

Cámara en diseño:

dhcp-host=77:88:99:AA:BB:CC,192.168.1.42,cam-diseño-01

```
dhcp-host=AA:BB:CC:11:22:33,192.168.1.32,pc-desarrollo-01
dhcp-host=77:88:99:AA:BB:CC,192.168.1.42,cam-diseño-01
dhcp-host=DD:EE:FF:44:55:66,192.168.1.52,imp-administracion-01
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
 ^X Exit ^R Read File ^N Replace ^U Paste ^J Justify ^_ Go To Line

2. Control de acceso básico.

Para crear una lista blanca de dispositivos autorizados (por MAC) y rechazar los no registrados, lo haremos añadiendo la siguiente línea a nuestro archivo

dnsmasq.conf:

dhcp-ignore=tag:!known

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~
GNU nano 7.2 /etc/dnsmasq.conf
#conf-dir=/etc/dnsmasq.d,.bak
# Include all files in a directory which end in .conf
#conf-dir=/etc/dnsmasq.d/*.conf
# If a DHCP client claims that its name is "wpad", ignore that.
# This fixes a security hole. see CERT Vulnerability VU#598349
#dhcp-name-match=set:wpad-ignore,wpad
#dhcp-ignore-names=tag:wpad-ignore

interface=enp0s3
bind-interfaces
dhcp-ignore=tag:!known
dhcp-range=set:desarrollo,192.168.1.31,192.168.1.40,12h
dhcp-range=set:diseño,192.168.1.41,192.168.1.50,12h
dhcp-range=set:administracion,192.168.1.51,192.168.1.60,12h

dhcp-host=AA:BB:CC:11:22:33,192.168.1.32,pc-desarrollo-01
dhcp-host=77:88:99:AA:BB:CC,192.168.1.42,cam-diseño-01
dhcp-host=DD:EE:FF:44:55:66,192.168.1.52,imp-administracion-01
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
 ^X Exit ^R Read File ^N Replace ^U Paste ^J Justify ^_ Go To Line

Los nombres simbólicos para cada cliente usando DHCP+DNS, ya los asignamos anteriormente al definir nombres en cada dhcp-host, por ejemplo:

dhcp-host=AA:BB:CC:11:22:33,192.168.1.32,pc-desarrollo-01

Para aplicar reglas en iptables para permitir tráfico solo desde direcciones ip asignadas lo haremos con los comandos:

sudo iptables -A INPUT -s 192.168.1.32 -j ACCEPT

sudo iptables -A INPUT -s 192.168.1.42 -j ACCEPT

sudo iptables -A INPUT -s 192.168.1.52 -j ACCEPT

Y ahora para bloquear el resto del tráfico:

sudo iptables -A INPUT -s 192.168.1.0/24 -j DROP

```

AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo iptables -A INPUT -s 192.168.1.32 -j ACCEPT
sudo iptables -A INPUT -s 192.168.1.42 -j ACCEPT
sudo iptables -A INPUT -s 192.168.1.52 -j ACCEPT
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo iptables -A INPUT -s 192.168.1.0/24 -j DROP
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ iptables -L
iptables v1.8.10 (nf_tables): Could not fetch rule set generation id: Permission denied (you
must be root)
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo iptables -L
Chain INPUT (policy DROP)
target prot opt source destination
ufw-before-logging-input all -- anywhere anywhere
ufw-before-input all -- anywhere anywhere
ufw-after-input all -- anywhere anywhere
ufw-after-logging-input all -- anywhere anywhere
ufw-reject-input all -- anywhere anywhere
ufw-track-input all -- anywhere anywhere
ACCEPT all -- 192.168.1.32 anywhere
ACCEPT all -- 192.168.1.42 anywhere
ACCEPT all -- 192.168.1.52 anywhere
DROP all -- 192.168.1.0/24 anywhere

```

3. Integración con clientes y simulación de red.

Ahora crearemos 1 cliente de linux para cada departamento y configuraremos su adaptador de red en red interna. En la máquina servidor añadimos un segundo adaptador de red y lo configuramos con red interna también.

Cada uno de estos clientes pertenece a los distintos departamentos anteriormente mencionados.

```

pc-desarrollo1 [Corriendo]-Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
vboxuser@Clientlinux:~$ sudo ip addr flush dev enp0s3
vboxuser@Clientlinux:~$ sudo ip netns exec vns1
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LFF/enp0s3/08:00:27:ec:1e:fd
Sending on LFF/enp0s3/08:00:27:ec:1e:fd
Sending on Socket/fallback
xid: warning: no network with usable MACBDR found for seed's uniqueness enforcement
xid: read init seed (0x0149f6a) built using gethostid
DHCPDISCOVER on enp0s3 to 255.255.255.255 port 67 interval 3 (xid=0x5f4b57)
OFFER of 192.168.100.52 from 192.168.100.1 (xid=0x0070b215)
DHCPREQUEST for 192.168.100.52 on enp0s3 to 255.255.255.255 port 67
ACK of 192.168.100.52 from 192.168.100.1 (xid=0x0070b215)
Setting LLNRP support level "yes" for "2", but the global support
bound to 192.168.100.52 -- renewal in 15786 seconds.
vboxuser@Clientlinux:~$

pc-diseño1 [Corriendo]-Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
vboxuser@Clientlinux:~$ sudo ip addr flush dev enp0s3
vboxuser@Clientlinux:~$ sudo ip netns exec vns2
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LFF/enp0s3/08:00:27:3d:4d:35
Sending on LFF/enp0s3/08:00:27:3d:4d:35
Sending on Socket/fallback
xid: warning: no network with usable MACBDR found for seed's uniqueness enforcement
xid: read init seed (0x0149f6a) built using gethostid
DHCPREQUEST for 192.168.100.42 on enp0s3 to 255.255.255.255 port 67 (xid=0x5f4b57)
OFFER of 192.168.100.42 from 192.168.100.1 (xid=0x0070b215)
DHCPREQUEST for 192.168.100.42 on enp0s3 to 255.255.255.255 port 67 (xid=0x152b0bc7)
ACK of 192.168.100.42 from 192.168.100.1 (xid=0x0070b215)
Setting LLNRP support level "yes" for "2", but the global support
bound to 192.168.100.42 -- renewal in 21565 seconds.
vboxuser@Clientlinux:~$

pc-administración1 [Corriendo]-Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
vboxuser@Clientlinux:~$ sudo ip addr flush dev enp0s3
vboxuser@Clientlinux:~$ sudo ip netns exec vns3
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LFF/enp0s3/08:00:27:12:c2:60
Sending on LFF/enp0s3/08:00:27:12:c2:60
Sending on Socket/fallback
xid: warning: no network with usable MACBDR found for seed's uniqueness enforcement
xid: read init seed (0x0149f6a) built using gethostid
DHCPDISCOVER on enp0s3 to 255.255.255.255 port 67 interval 3 (xid=0xe70b2b15)
OFFER of 192.168.100.52 from 192.168.100.1 (xid=0x0070b215)
DHCPREQUEST for 192.168.100.52 on enp0s3 to 255.255.255.255 port 67 (xid=0x152b0bc7)
ACK of 192.168.100.52 from 192.168.100.1 (xid=0x0070b215)
Setting LLNRP support level "yes" for "2", but the global support
bound to 192.168.100.52 -- renewal in 20671 seconds.
vboxuser@Clientlinux:~$

SVR-BASE-AZDINFARISSE [Corriendo]-Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Jul 8 08:50
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~
GNU nano 7.2 /etc/dnsmasq.conf

# If a DHCP client claims that its name is "wpad", ignore that.
# This fixes a security hole, see CERT Vulnerability VU#598349
#dhcp-name-match=wpad:ignore:wpad
#dhcp-ignore-names=wpad:ignore

interface=enp0s8
bind-interfaces
dhcp-range=tag:desarrollo,192.168.100.31,192.168.100.40,12h
dhcp-range=tag:diseño,192.168.100.41,192.168.100.50,12h
dhcp-range=tag:administración,192.168.100.51,192.168.100.60,12h
dhcp-host=08:00:27:ec:1e:fd,192.168.100.32,pc-desarrollo-01,set:desarrollo
dhcp-host=08:00:27:bd:dd:35,192.168.100.42,pc-diseño-01,set:diseño
dhcp-host=08:00:27:12:c2:60,192.168.100.52,pc-administración-01,set:administración

```

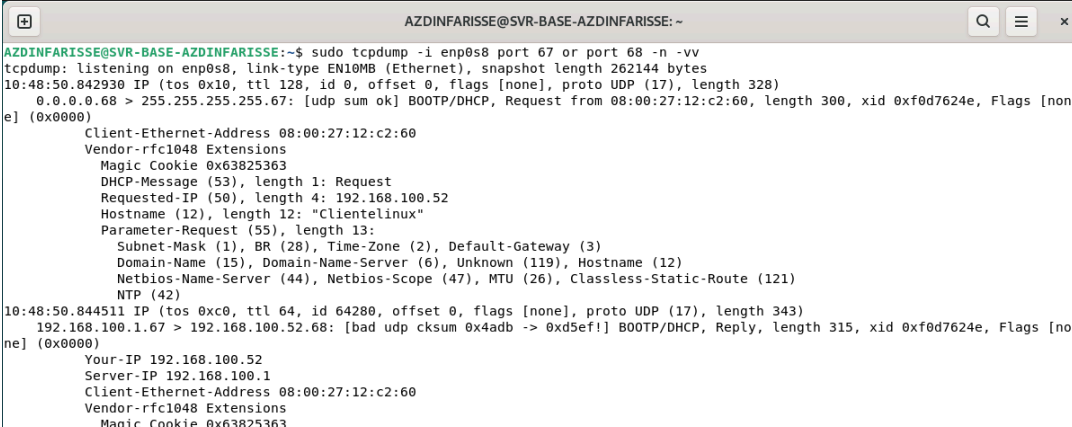
4. Supervisión de tráfico e identificación.

Para capturar el tráfico **DCHP** lo hacemos con el comando:

sudo tcpdump -i enp0s8 port 67 -n -vv

Y en cada máquina cliente introducimos el siguiente comando para que pida ip al servidor:

sudo dhclient -v enp0s3



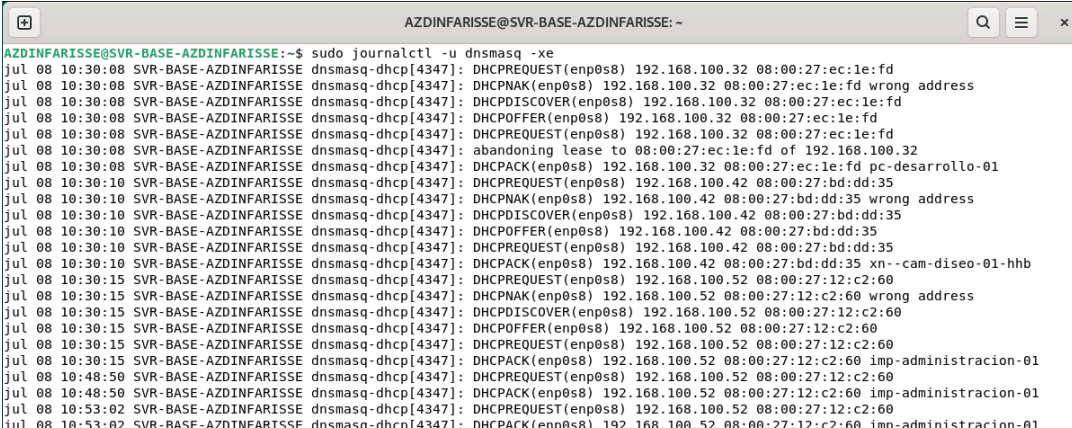
```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo tcpdump -i enp0s8 port 67 or port 68 -n -vv
tcpdump: listening on enp0s8, link-type EN10MB (Ethernet), snapshot length 262144 bytes
10:48:50.842930 IP (tos 0x10, ttl 128, id 0, offset 0, flags [none], proto UDP (17), length 328)
  0.0.0.0.68 > 255.255.255.255.67: [udp sum ok] BOOTP/DHCP, Request from 08:00:27:12:c2:60, length 300, xid 0xf0d7624e, Flags [none] (0x0000)
    Client-Ethernet-Address 08:00:27:12:c2:60
    Vendor-rfc1048 Extensions
    Magic Cookie 0x63825363
    DHCP-Message (53), length 1: Request
    Requested-IP (50), length 4: 192.168.100.52
    Hostname (12), length 12: "Clientlinux"
    Parameter-Request (55), length 13:
      Subnet-Mask (1), BR (28), Time-Zone (2), Default-Gateway (3)
      Domain-Name (15), Domain-Name-Server (6), Unknown (119), Hostname (12)
      Netbios-Name-Server (44), Netbios-Scope (47), MTU (26), Classless-Static-Route (121)
      NTP (42)
10:48:50.844511 IP (tos 0xc0, ttl 64, id 64280, offset 0, flags [none], proto UDP (17), length 343)
  192.168.100.1.67 > 192.168.100.52.68: [bad udp cksum 0x4adb -> 0xd5ef!] BOOTP/DHCP, Reply, length 315, xid 0xf0d7624e, Flags [none] (0x0000)
    Your-IP 192.168.100.52
    Server-IP 192.168.100.1
    Client-Ethernet-Address 08:00:27:12:c2:60
    Vendor-rfc1048 Extensions
    Magic Cookie 0x63825363
```

En esta imagen podemos observar que el cliente está pidiendo al servidor una ip, y que este le asigna la ip 192.168.100.52

Ahora vamos a analizar los logs creados por **dnsmasq** y **/var/log/syslog**.

Para analizar pos creados por dnsmasq lo hacemos con el comando:

sudo journalctl -u dnsmasq -xe



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo journalctl -u dnsmasq -xe
jul 08 10:30:08 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPREQUEST(enp0s8) 192.168.100.32 08:00:27:ec:1e:fd
jul 08 10:30:08 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPNAK(enp0s8) 192.168.100.32 08:00:27:ec:1e:fd wrong address
jul 08 10:30:08 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPDISCOVER(enp0s8) 192.168.100.32 08:00:27:ec:1e:fd
jul 08 10:30:08 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPPOFFER(enp0s8) 192.168.100.32 08:00:27:ec:1e:fd
jul 08 10:30:08 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPREQUEST(enp0s8) 192.168.100.32 08:00:27:ec:1e:fd
jul 08 10:30:08 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: abandoning lease to 08:00:27:ec:1e:fd of 192.168.100.32
jul 08 10:30:08 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPACK(enp0s8) 192.168.100.32 08:00:27:ec:1e:fd pc-desarrollo-01
jul 08 10:30:10 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPREQUEST(enp0s8) 192.168.100.42 08:00:27:bd:dd:35
jul 08 10:30:10 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPDISCOVER(enp0s8) 192.168.100.42 08:00:27:bd:dd:35
jul 08 10:30:10 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPPOFFER(enp0s8) 192.168.100.42 08:00:27:bd:dd:35
jul 08 10:30:10 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPREQUEST(enp0s8) 192.168.100.42 08:00:27:bd:dd:35
jul 08 10:30:10 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPACK(enp0s8) 192.168.100.42 08:00:27:bd:dd:35 xn--cam-diseo-01-hhb
jul 08 10:30:15 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPREQUEST(enp0s8) 192.168.100.52 08:00:27:12:c2:60
jul 08 10:30:15 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPNAK(enp0s8) 192.168.100.52 08:00:27:12:c2:60 wrong address
jul 08 10:30:15 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPDISCOVER(enp0s8) 192.168.100.52 08:00:27:12:c2:60
jul 08 10:30:15 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPPOFFER(enp0s8) 192.168.100.52 08:00:27:12:c2:60
jul 08 10:30:15 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPREQUEST(enp0s8) 192.168.100.52 08:00:27:12:c2:60
jul 08 10:30:15 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPACK(enp0s8) 192.168.100.52 08:00:27:12:c2:60 imp-administracion-01
jul 08 10:48:50 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPREQUEST(enp0s8) 192.168.100.52 08:00:27:12:c2:60
jul 08 10:48:50 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPACK(enp0s8) 192.168.100.52 08:00:27:12:c2:60 imp-administracion-01
jul 08 10:53:02 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPREQUEST(enp0s8) 192.168.100.52 08:00:27:12:c2:60
jul 08 10:53:02 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4347]: DHCPACK(enp0s8) 192.168.100.52 08:00:27:12:c2:60 imp-administracion-01
```

En la imagen vemos que el cliente ha pedido ip varias veces al servidor y este se la ha dado. Sale en varias ocasiones porque he estado haciendo pruebas. Siempre se le da la misma ip, ya que en el archivo **dnsmasq.conf** está configurado para que este cliente con la dirección MAC 08:00:27:12:c2:60 siempre obtenga esta ip.

Ahora los creados por `/var/log/syslog` con el comando:
sudo grep dnsmasq /var/log/syslog

```

AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~
2025-07-07T09:08:35.459222+02:00 SVR-BASE-AZDINFARISSE resolvconf[4742]: Dropped protocol specifier '.dnsmasq' from 'lo.dnsmasq'. Usin
'lo' (ifindex=1).
2025-07-07T09:08:35.460718+02:00 SVR-BASE-AZDINFARISSE dnsmasq[1193]: exiting on receipt of SIGTERM
2025-07-07T09:08:35.467343+02:00 SVR-BASE-AZDINFARISSE systemd[1]: dnsmasq.service: Deactivated successfully.
2025-07-07T09:08:35.467509+02:00 SVR-BASE-AZDINFARISSE systemd[1]: Stopped dnsmasq.service - dnsmasq - A lightweight DHCP and caching
NS server.
2025-07-07T09:08:35.475950+02:00 SVR-BASE-AZDINFARISSE systemd[1]: Starting dnsmasq.service - dnsmasq - A lightweight DHCP and caching
DNS server...
2025-07-07T09:08:35.491047+02:00 SVR-BASE-AZDINFARISSE dnsmasq[4760]: started, version 2.90 cachesize 150
2025-07-07T09:08:35.491109+02:00 SVR-BASE-AZDINFARISSE dnsmasq[4760]: compile time options: IPv6 GNU-getopt DBus no-UBus i18n IDN2 DHC
DHCPv6 no-Lua TFTP conntrack ipset nftset auth cryptohash DNSSEC loop-detect inotify dumpfile
2025-07-07T09:08:35.491136+02:00 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4760]: DHCP, IP range 192.168.100.51 -- 192.168.100.60, lease time
12h
2025-07-07T09:08:35.491214+02:00 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4760]: DHCP, IP range 192.168.100.41 -- 192.168.100.50, lease time
12h
2025-07-07T09:08:35.491256+02:00 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4760]: DHCP, IP range 192.168.100.31 -- 192.168.100.40, lease time
12h
2025-07-07T09:08:35.491310+02:00 SVR-BASE-AZDINFARISSE dnsmasq[4760]: read /etc/hosts - 3 names
2025-07-07T09:08:35.504105+02:00 SVR-BASE-AZDINFARISSE resolvconf[4768]: Dropped protocol specifier '.dnsmasq' from 'lo.dnsmasq'. Usin
'lo' (ifindex=1).
2025-07-07T09:08:35.505368+02:00 SVR-BASE-AZDINFARISSE systemd[1]: Started dnsmasq.service - dnsmasq - A lightweight DHCP and caching
NS server.
grep: /var/log/syslog: binary file matches

```

En esta imagen se puede observar las distintas configuraciones del archivo `dnsmasq.conf` y a la hora en la que el servicio ha sido lanzado.

Lo siguiente que hacemos es usar **arp-scan** para ver los dispositivos conectados a la red y su relación IP/MAC. Esto lo hacemos con el comando:
sudo arp-scan --interface=enp0s8 --localnet

```

AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~
2025-07-07T09:08:35.491286+02:00 SVR-BASE-AZDINFARISSE dnsmasq-dhcp[4760]: DHCP, sockets bound exclusively to interface enp0s8
2025-07-07T09:08:35.491310+02:00 SVR-BASE-AZDINFARISSE dnsmasq[4760]: read /etc/hosts - 3 names
2025-07-07T09:08:35.504105+02:00 SVR-BASE-AZDINFARISSE resolvconf[4768]: Dropped protocol specifier '.dnsmasq' from 'lo.dnsmasq'. Usin
'lo' (ifindex=1).
2025-07-07T09:08:35.505368+02:00 SVR-BASE-AZDINFARISSE systemd[1]: Started dnsmasq.service - dnsmasq - A lightweight DHCP and caching
NS server.
grep: /var/log/syslog: binary file matches
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo apt install arp-scan
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
arp-scan ya está en su versión más reciente (1.10.0-2build2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 52 no actualizados.
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo arp-scan --interface=enp0s8 --localnet
Interface: enp0s8, type: EN10MB, MAC: 08:00:27:e9:d8:c7, IPv4: 192.168.100.1
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.100.32 08:00:27:ec:1e:fd (Unknown)
192.168.100.42 08:00:27:bd:dd:35 (Unknown)
192.168.100.52 08:00:27:12:c2:60 (Unknown)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.833 seconds (139.66 hosts/sec). 3 responded

```

En la imagen podemos ver que tenemos los tres clientes conectados, cada uno con su respectiva IP perteneciente al departamento correspondiente y su dirección MAC. También se observa la ip de la propia máquina servidor.