

OWISAM-DI

Antonio Alonso García



Instalación de Scapy	3
Interfaz modo monitor	3
Código de herramienta OWISAM-DI	4
Comando para ejecutar OWISAM-DI	6
Resultado	6
Ver ayuda	7



Instalación de Scapy:

```
sudo apt update
```

```
python3 -m venv venv  
source venv/bin/activate
```

```
pip3 install scapy
```

```
(venv)(kali@kali)-[~/rtl8812au]  
$ pip3 install scapy  
Collecting scapy  
  Downloading scapy-2.6.1-py3-none-any.whl.metadata (5.6 kB)  
  Downloading scapy-2.6.1-py3-none-any.whl (2.4 MB)  
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.4/2.4 MB 36.0 MB/s eta 0:00:00  
Installing collected packages: scapy  
Successfully installed scapy-2.6.1  
  
(venv)(kali@kali)-[~/rtl8812au]
```

Interfaz modo monitor.

```
sudo ip link set wlan0 down  
sudo iw dev wlan0 set type monitor  
sudo ip link set wlan0 up
```

```
iwconfig
```

```
(venv)(kali@kali)-[~/rtl8812au]  
$ iwconfig  
lo          no wireless extensions.  
eth0        no wireless extensions.  
docker0     no wireless extensions.  
wlan0       unassociated  ESSID:""  Nickname:"<WIFI@REALTEK>"  
            Mode:Monitor  Frequency=2.437 GHz  Access Point: Not-Associated  
            Sensitivity:0/0  
            Retry:off   RTS thr:off   Fragment thr:off  
            Power Management:off  
            Link Quality:0  Signal level:0  Noise level:0  
            Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0  
            Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```



Código de herramienta OWISAM-DI.

```
#!/usr/bin/env python3
import json
import argparse
import time
import sys
import signal
from scapy.all import sniff, RadioTap
from scapy.layers.dot11 import Dot11, Dot11Beacon, Dot11ProbeReq

# Almacena APs y clientes
aps = {}      # BSSID -> {'ssid','channel','crypto','signal'}
clients = set() # MACs de clientes

# Maneja Ctrl+C para terminar el script inmediatamente
stop_capture = False
def handle_sigint(signum, frame):
    global stop_capture
    print("\n[!] Captura interrumpida por el usuario. Finalizando...")
    stop_capture = True

signal.signal(signal.SIGINT, handle_sigint)

# Callback para cada paquete capturado
def packet_handler(pkt):
    if pkt.haslayer(Dot11Beacon):
        bssid = pkt[Dot11].addr2
        stats = pkt[Dot11Beacon].network_stats()
        ssid = stats.get('ssid', "")
        channel = stats.get('channel', "")
        crypto = stats.get('crypto', [])
        if not isinstance(crypto, list):
            crypto = [str(crypto)]
        signal_dbm = None
        if pkt.haslayer(RadioTap) and hasattr(pkt[RadioTap], 'dBm_AntSignal'):
            signal_dbm = pkt[RadioTap].dBm_AntSignal
        prev = aps.get(bssid)
        if prev is None or (signal_dbm is not None and prev.get('signal', -999) < signal_dbm):
            aps[bssid] = {'ssid': ssid, 'channel': channel, 'crypto': crypto, 'signal': signal_dbm}

    elif pkt.haslayer(Dot11ProbeReq):
        client_mac = pkt[Dot11].addr2
        if client_mac:
```



```

clients.add(client_mac)

# Función principal
def main():
    parser = argparse.ArgumentParser(description='OWISAM-DI: Device Discovery')
    parser.add_argument('-i', '--interface', required=True, help='Interfaz en modo monitor')
    parser.add_argument('-o', '--output', required=True, help='Archivo JSON de salida')
    parser.add_argument('--wait-time', type=int, default=10,
                        help='Segundos para detectar primer paquete (0=infinito)')
    args = parser.parse_args()

    output_file = args.output

    print(f"[+] Esperando primer paquete en {args.interface} (timeout {args.wait_time}s)...")
    try:
        sniff(iface=args.interface, prn=packet_handler, store=False,
              timeout=(args.wait_time if args.wait_time > 0 else None))
    except KeyboardInterrupt:
        pass

    if not aps and not clients:
        print(f"[!] No se detectó ningún paquete tras {args.wait_time}s. Saliendo.")
        sys.exit(1)

    duration = 60
    end_time = time.time() + duration
    bar_length = 50
    print(f"[+] Paquetes detectados, iniciando captura de {duration}s...")

    while time.time() < end_time and not stop_capture:
        sniff(iface=args.interface, prn=packet_handler, store=False, timeout=1)
        elapsed = duration - (end_time - time.time())
        filled = int(bar_length * elapsed / duration)
        bar = '#' * filled + '-' * (bar_length - filled)
        sys.stdout.write(f"\r[+] Capturando: [{bar}] {int(elapsed)}/{duration}s")
        sys.stdout.flush()

    print() # nueva línea al terminar barra de progreso

# Generar y guardar resultados
results = {
    'access_points': [
        {'bssid': b, 'ssid': d['ssid'], 'channel': d['channel'],
         'crypto': d['crypto'], 'signal': d['signal']} for b, d in aps.items()
    ],

```



```

        'clients': list(clients)
    }
    with open(output_file, 'w') as f:
        json.dump(results, f, indent=2)
    print(f"[+] Resultados guardados en {output_file}")

if __name__ == '__main__':
    main()

```

```

(venv)(kali@kali)-[~/rtl8812au]
└─$ sudo python3 owisam_di.py --interface wlan0 --output resultados.json --wait-time 10
[+] Esperando primer paquete en wlan0 (timeout 10s)...
[+] Paquetes detectados, iniciando captura de 60s...
[+] Capturando: [#####] 60/60s
[+] Resultados guardados en resultados.json

(venv)(kali@kali)-[~/rtl8812au]
└─$ nano resultados.json

```

Comando para ejecutar OWISAM-DI

```

sudo python3 owisam_di.py
--interface wlan0
--output resultados.json
--wait-time 10

```

Resultado

```

GNU nano 8.4 resultados.json
[+] "access_points": [
    {
      "bssid": "34:57:60:d2:29:22",
      "ssid": "MOVISTAR_PLUS_8939",
      "channel": 100,
      "crypto": [
        "'WPA2/PSK'"
      ],
      "signal": -60
    },
    {
      "bssid": "9c:63:5b:2a:64:48",
      "ssid": "DIGIFIBRA-Rctd",
      "channel": 8,
      "crypto": [
        "'WPA2/PSK'"
      ],
      "signal": -58
    },
    {
      "bssid": "10:47:b3:d2:da:bc",
      "ssid": "Vivamovil-40F7A8",
      "channel": 11,
      "crypto": [
        "'WPA2/PSK', 'WPA/PSK'"
      ],
      "signal": -70
    },
    {
      "bssid": "va:c9:5a:c0:b8:ac",
      "ssid": "AT_401_RAC_056905_WW_b8ac",
      "channel": 11,
      "crypto": [
        "'WPA2/PSK'"
      ],
      "signal": -58
    }
  ],
  "clients": [
    "9a:06:f2:f3:c7:09",
    "fc:9c:91:8c:c1:1b",
    "c4:e9:89:2e:ca:2a",
    "9a:4c:d3:ce:b1:25"
  ]
}

```



Ver ayuda

python3 owisam_di.py -h

```
(venv)(kali@kali)-[~/rtl8812au]
$ python3 owisam_di.py -h
usage: owisam_di.py [-h] -i INTERFACE -o OUTPUT [--wait-time WAIT_TIME]

OWISAM-DI: Device Discovery

options:
  -h, --help            show this help message and exit
  -i, --interface INTERFACE
                        Interfaz en modo monitor
  -o, --output OUTPUT so Archivo JSON de salida
  --wait-time WAIT_TIME
                        Segundos para detectar primer paquete (0=infinito)

(venv)(kali@kali)-[~/rtl8812au]
```

