Make Your Own Network Security Tools With Scapy

Temario

- Introducción a Scapy
- Sniffing ARP
- ARP Discovery
- Man-In-The-Middle (MITM)
- ARP Spoofing
- IP Forwarding
- Sniffing HTTP
- Modificación de Tráfico HTTP

Introducción a Scapy

1. Es una librería de Python (2.7.x y 3.4+)
2. Permite enviar, escuchar, analizar y crear paquetes de red
3. Soporta un modo de trabajo interactivo y tambíen por scripts
4. Principalmente, hace dos cosas: envía paquetes, recibe respuestas
5. No interpreta, decodifica (ej: Puerto abierto vs TCP SYN/ACK)

Introducción a Scapy

"You’re free to put any value you want in any field you want and stack them like you want. *You’re an adult after all."* From: Scapy Official Docs

Inicio

```
# scapy3

                     aSPY//YASa
             apyyyyCY//////////YCa       |
            sY//////YSpcs  scpCY//Pp     | Welcome to Scapy
 ayp ayyyyyyySCP//Pp           syY//C    | Version 2.4.0
 AYAsAYYYYYYYY///Ps              cY//S   |
         pCCCCY//p          cSSps y//Y   | https://github.com/secdev/scapy
         SPPPP///a          pP///AC//Y   |
              A//A            cyP////C   | Have fun!
              p///Ac            sC///a   |
              P////YCpc           A//A   | Craft packets like it is your last
       scccccp///pSP///p          p//Y   | day on earth.
      sY/////////y  caa           S//P   |                     -- Lao-Tze
       cayCyayP//Ya              pY/Ya   |
        sY/PsY////YCc           aC//Yp
         sc  sccaCY//PCypaapyCP//YSs
                  spCPY//////YPSps
                       ccaacs
                                       using IPython 5.5.0
>>>
```

Nuestro Primer Paquete

```
:::py3
>>> l3 = IP(dst='8.8.4.4')
>>> l4 = ICMP()
>>> sr1(l3/l4)

Begin emission:
.Finished sending 1 packets.
*
Received 2 packets, got 1 answers, remaining 0 packets
<IP  version=4 ihl=5 tos=0x0 len=28 id=11010 flags= frag=0 ttl=63 proto=icmp chksum=0x34c1
src=8.8.4.4 dst=10.0.2.15 options=[] |<ICMP  type=echo-reply code=0 chksum=0xffff id=0x0
seq=0x0 |<Padding  load='\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00' |>>
```

¿Qué es un "layer"?

```
:::py3
>>> l3.show()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= hopopt
  chksum= None
  src= 10.0.2.15
  dst= 8.8.4.4
  \options\

>>> l4.show()
###[ ICMP ]###
  type= echo-request
  code= 0
  chksum= None
  id= 0x0
  seq= 0x0
```

¿Qué es un "layer"? (2)

```
:::py3
>>> l3.show2()
###[ IP ]###
  version= 4
  ihl= 5
  tos= 0x0
  len= 20
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= hopopt
  chksum= 0x5ecb
  src= 10.0.2.15
  dst= 8.8.4.4
  \options\

>>> l4.show2()
###[ ICMP ]###
  type= echo-request
  code= 0
  chksum= 0xf7ff
  id= 0x0
  seq= 0x0
```

Generando varios paquetes

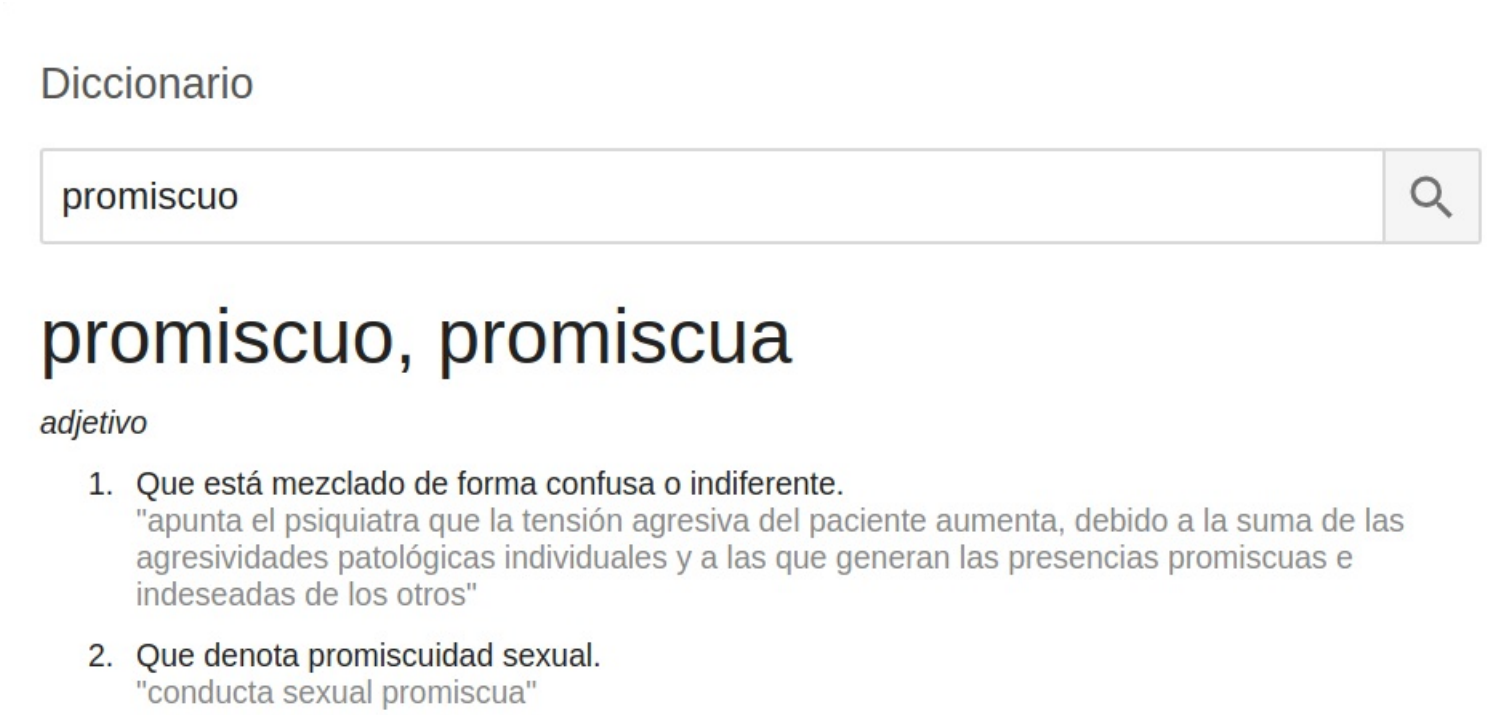
```
:::py3
>>> l3 = IP(dst='8.8.4.4/30')
>>> ans,unans = sr(l3/l4, timeout=3)
Begin emission:
.Finished sending 4 packets.
*
Received 2 packets, got 1 answers, remaining 3 packets
>>> ans
<Results: TCP:0 UDP:0 ICMP:1 Other:0>
>>> unans
<Unanswered: TCP:0 UDP:0 ICMP:3 Other:0>
```

No-Respondidos vs Respondidos

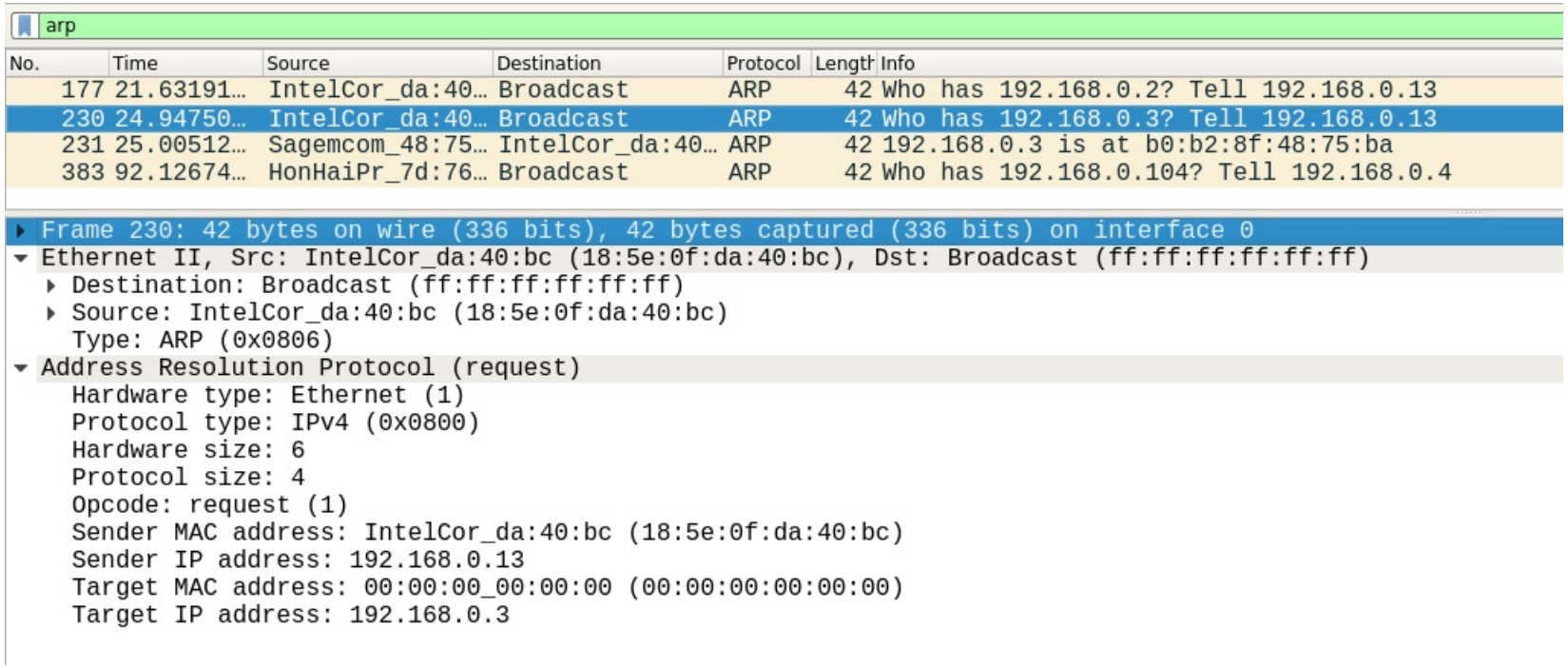
```
:::py3
>>> for pkt in unans:
...:    print(pkt.summary())
...:
IP / ICMP 10.0.2.15 > 8.8.4.5 echo-request 0
IP / ICMP 10.0.2.15 > 8.8.4.6 echo-request 0
IP / ICMP 10.0.2.15 > 8.8.4.7 echo-request 0

>>> for s,pkt in ans:
...:    print(s.summary(), '|', pkt.summary())
...:
IP / ICMP 10.0.2.15 > 8.8.4.4 echo-request 0 | IP / ICMP 8.8.4.4 > 10.0.2.15 echo-reply 0 / Paddin
```

Sniffing

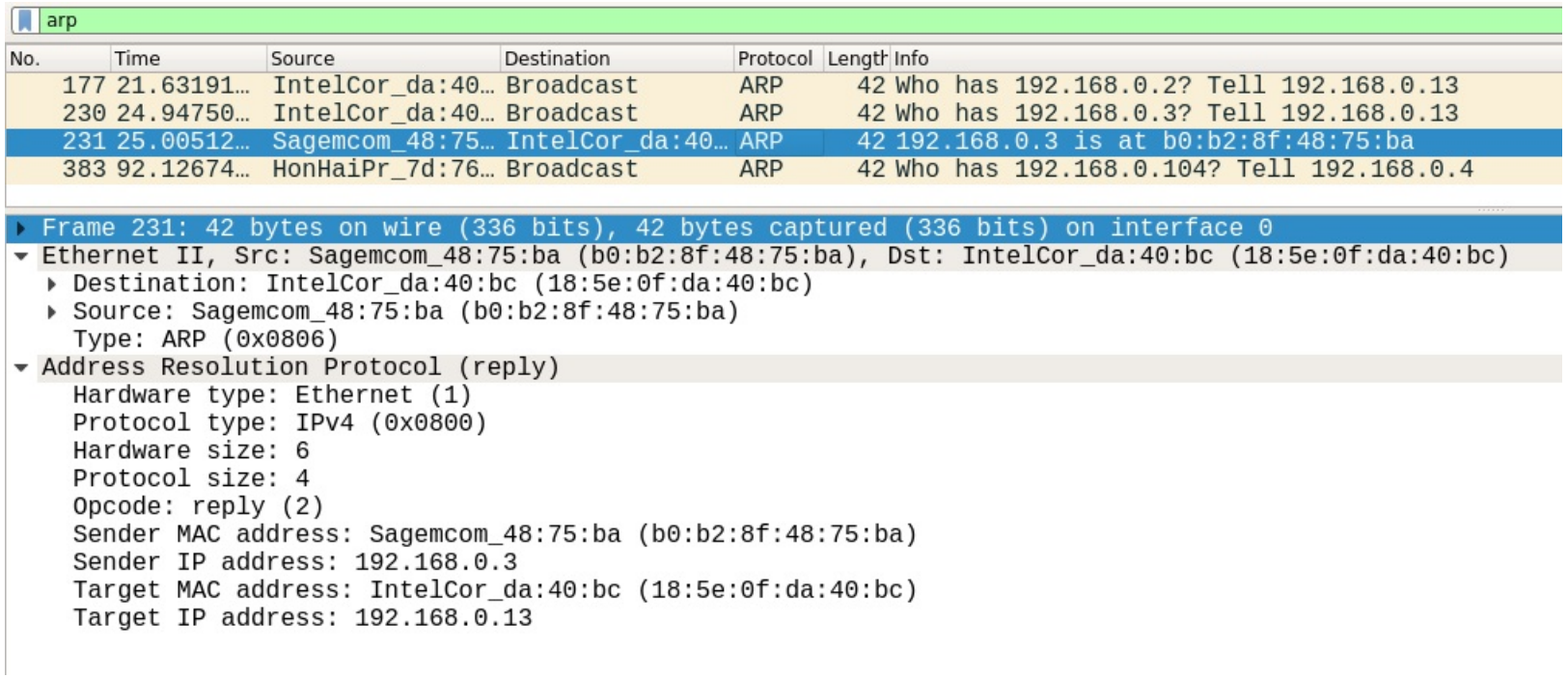
Placa en estado promiscuo



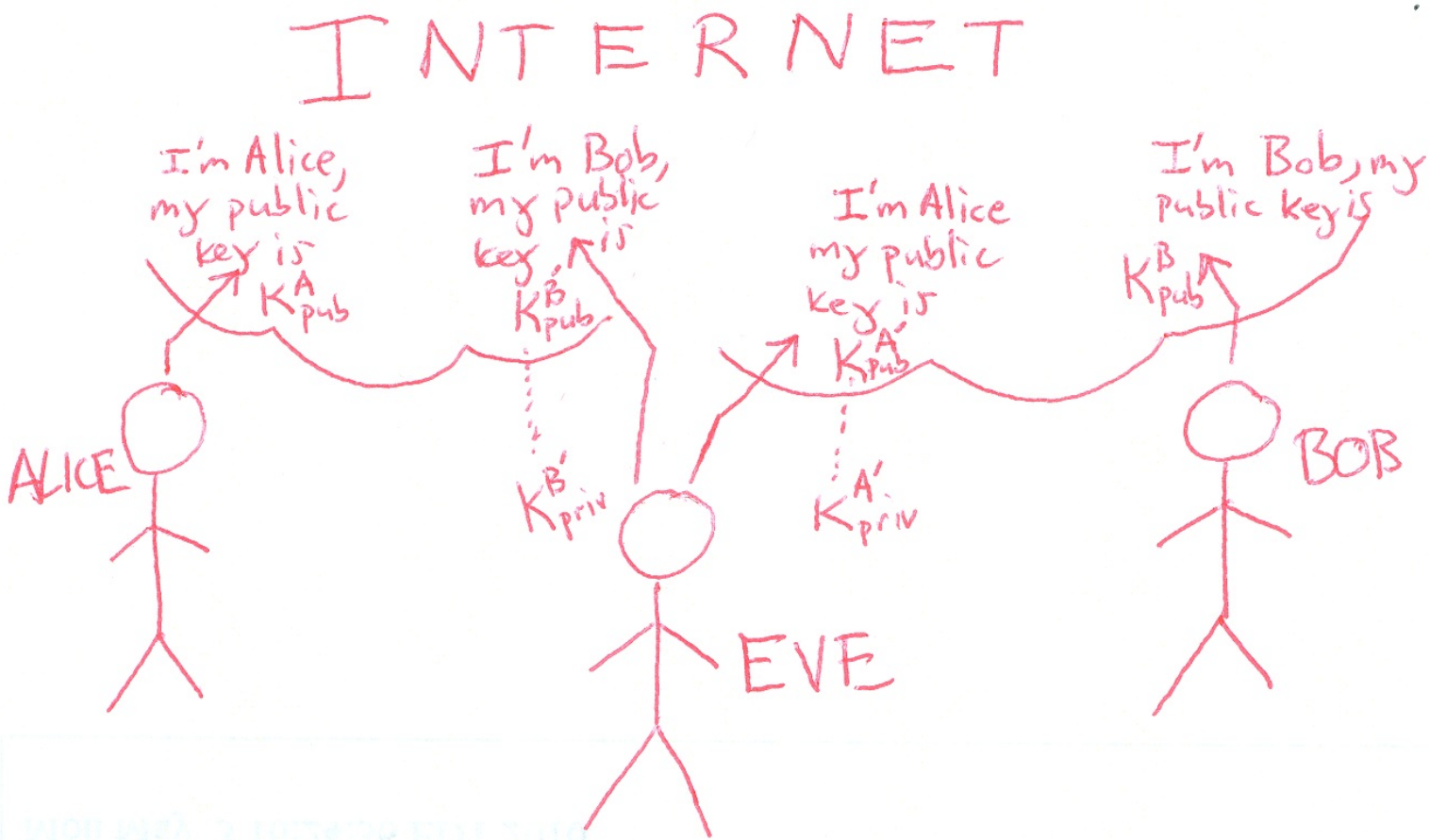
ARP Discovery



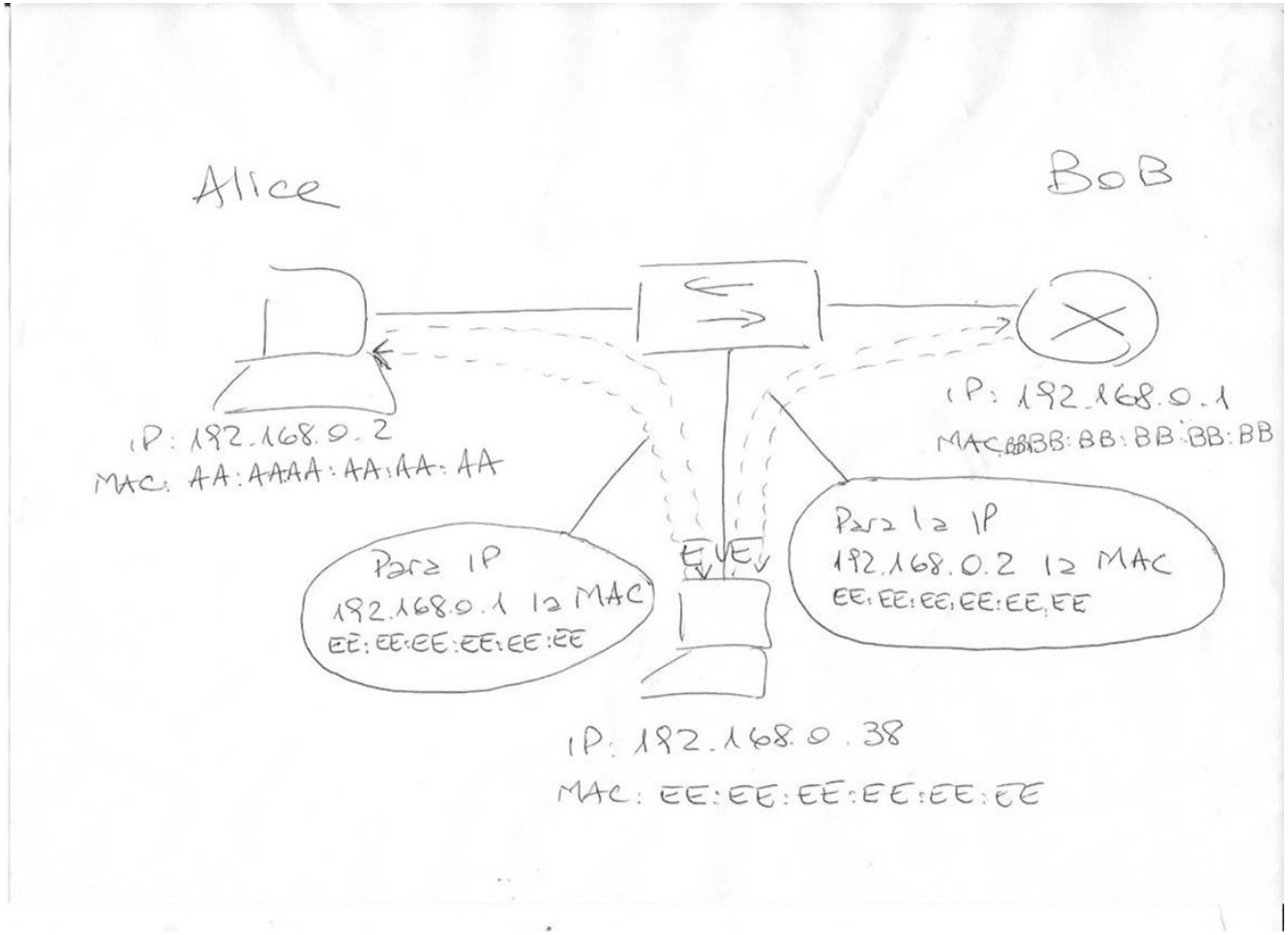
ARP Discovery (2)



Man-In-The-Middle (MITM)



ARP Spoofing



Modificación de Tráfico HTTP