Taller de Scapy

Teoría de las Comunicaciones

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Abril 2019

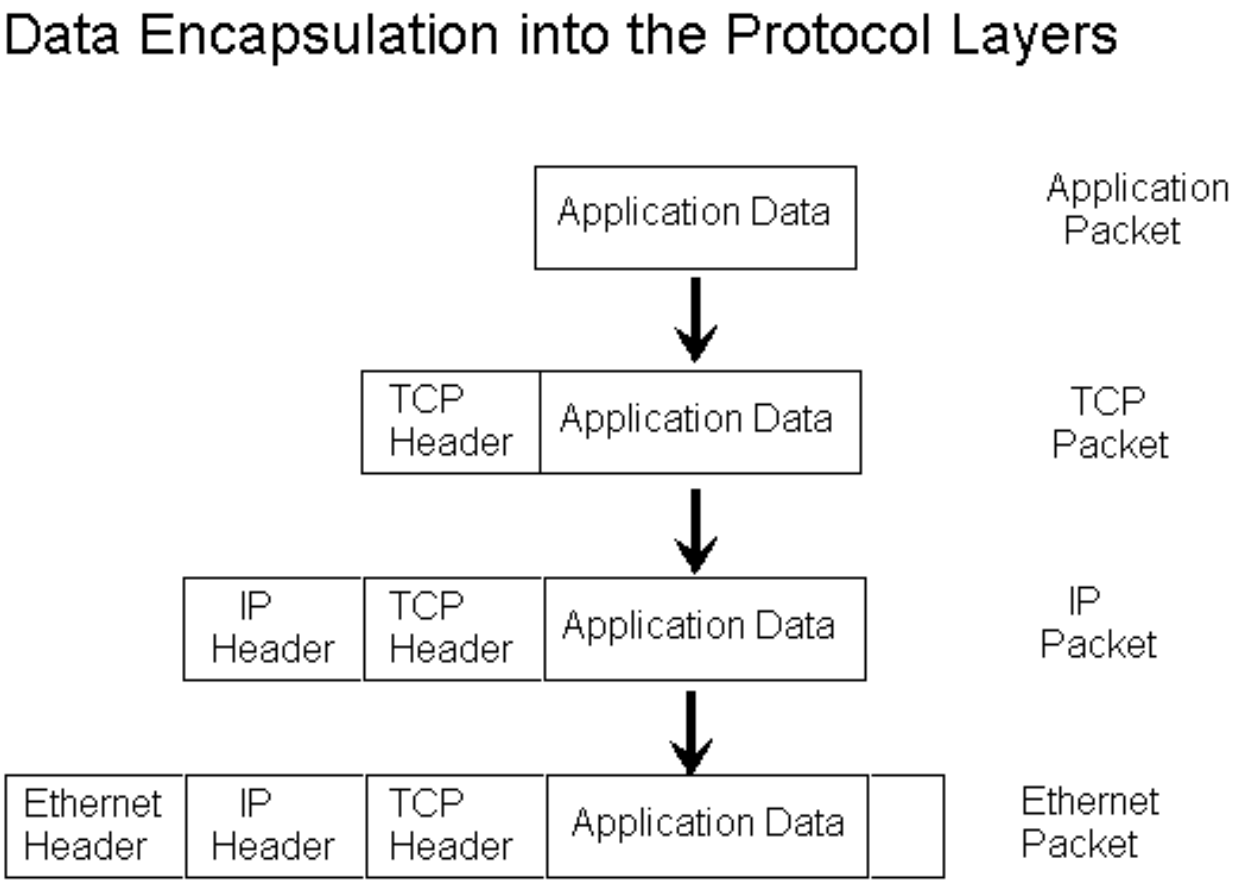
Agenda

1. Introducción
   - Encabezados - Tramas y Paquetes
2. ARP
   - ¿Qué es ARP?
   - Encabezado
3. Scapy
   - Instalación
   - Paquetes en Scapy
4. Sniffing
   - Definiciones
   - Escenarios
   - Con Scapy
   - Posibilidades
   - Bonus
   - Creando paquetes
   - Más material

Agenda

1. Introducción
   - Encabezados - Tramas y Paquetes
2. ARP
   - ¿Qué es ARP?
   - Encabezado
3. Scapy
   - Instalación
   - Paquetes en Scapy
4. Sniffing
   - Definiciones
   - Escenarios
   - Con Scapy
   - Posibilidades
   - Bonus
   - Creando paquetes
   - Más material

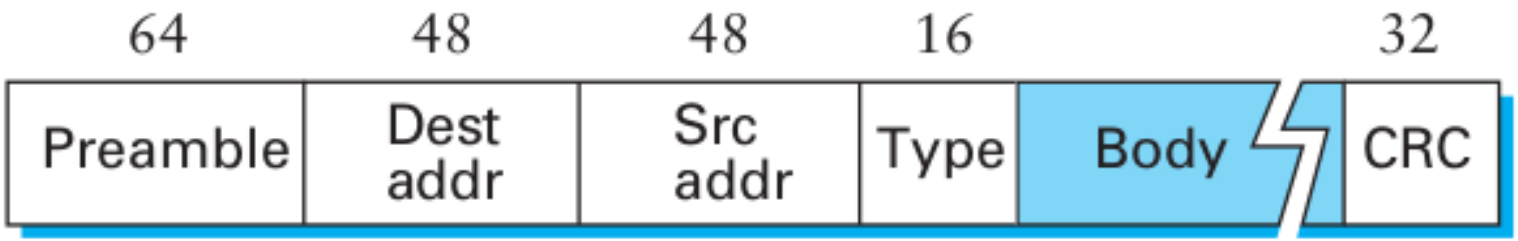
Encapsulamiento



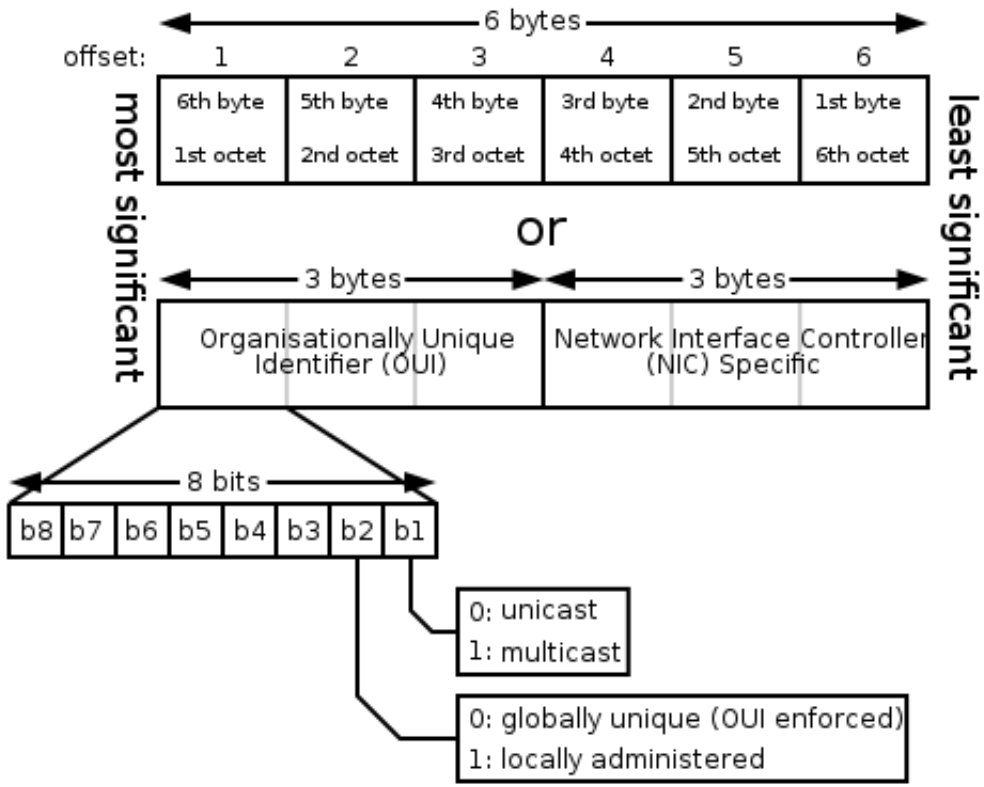
Ejemplo captura Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
7	3.378592453	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
8	3.378609597	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
181	43.926159451	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
182	43.926181716	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
541	80.368730565	LgElectr_6e:00:00_Rabbit	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.39
1528	82.006955666	LgElectr_6e:00:00_Rabbit	Broadcast	ARP	42	Gratuitous ARP for 192.168.1.39 (Request)
2959	83.952682184	LgElectr_6e:00:00_Rabbit	Broadcast	ARP	42	Gratuitous ARP for 192.168.1.39 (Request)
3183	84.267498285	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
3184	84.267509869	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
6762	124.591709421	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
6763	124.591721378	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
6934	165.753903358	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
6935	165.753915173	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
7037	188.194923996	Hormiga	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.34
7089	207.366528859	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
7090	207.366541161	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
7188	240.528362452	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7189	241.443871067	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7190	242.467896088	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7193	244.721005959	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7198	245.744995959	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7215	247.700268704	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
7216	247.700280541	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
7218	248.919217174	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7225	249.943213119	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7236	250.967527291	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7247	253.220136294	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7256	254.244159582	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7267	255.165824278	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7273	257.418670191	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7291	258.442669837	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7292	259.466710620	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7311	261.719582124	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7312	261.725751189	SamsungE_c9:00:00_Líquenes	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.38
7317	262.649721561	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7319	263.665513125	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1

Ethernet



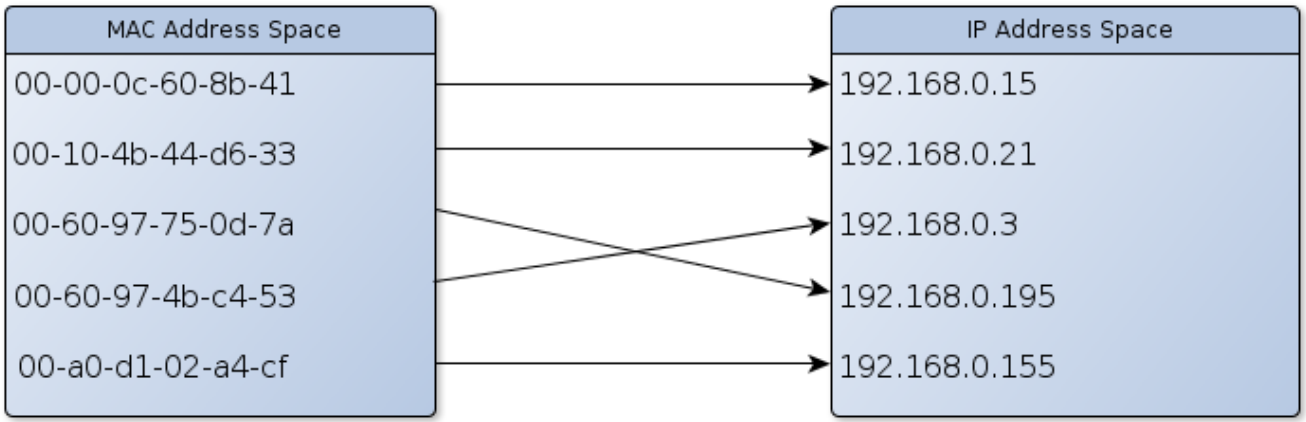
Ethernet - MAC Address



Agenda

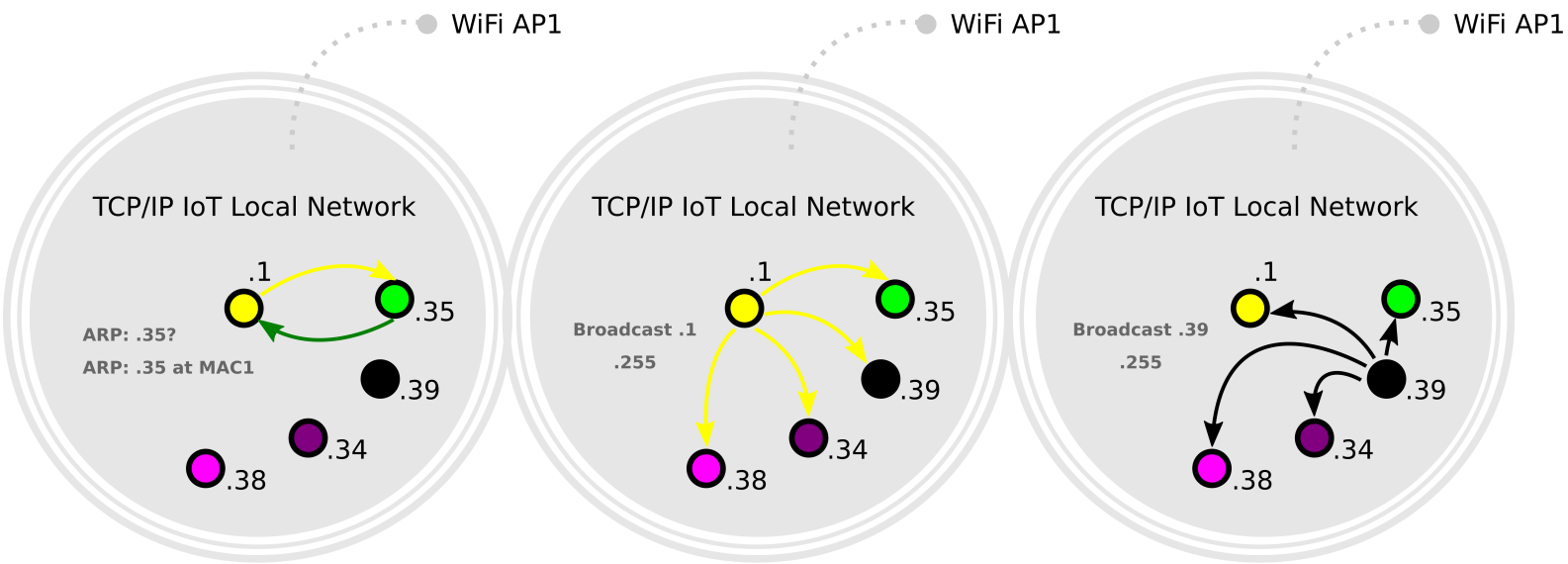
1. Introducción
   - Encabezados - Tramas y Paquetes
2. ARP
   - ¿Qué es ARP?
   - Encabezado
3. Scapy
   - Instalación
   - Paquetes en Scapy
4. Sniffing
   - Definiciones
   - Escenarios
   - Con Scapy
   - Posibilidades
   - Bonus
   - Creando paquetes
   - Más material

¿Y las direcciones IPs?



Ethernet - MAC Address - ARP (rfc826)

ARP permite construir tablas para traducir una dirección **A** en un espacio de direcciones de un protocolo **P** a direcciones **Ethernet** de 48 bits.



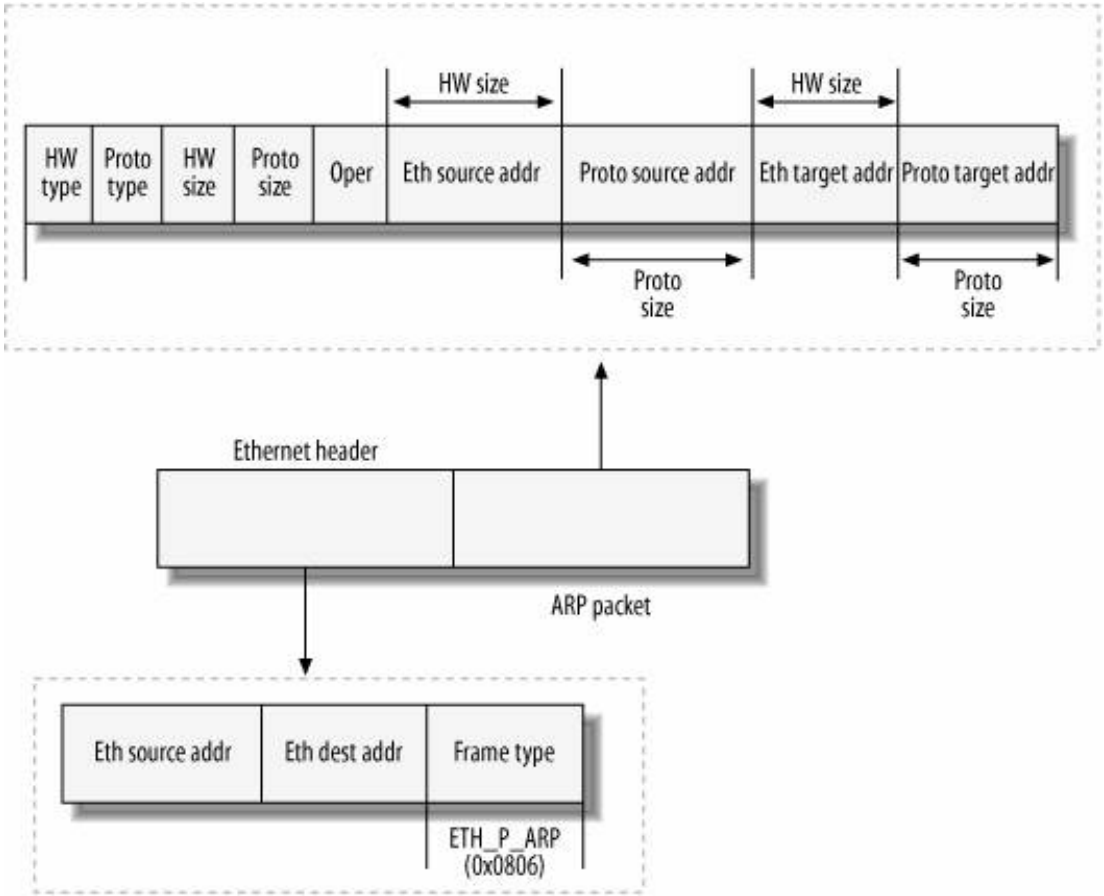
¿Qué es ARP?

- La sigla: *Address Resolution Protocol*.
- Es un protocolo que, en esencia, permite mapear direcciones de nivel de red a direcciones físicas.
- Clave e indispensable en el funcionamiento de las redes modernas.
- Especificado en el RFC 826 (circa 1982).
- No está limitado a IP + Ethernet: la especificación es general.

Tecnicismos varios

- La pregunta ARP consiste en un mensaje **broadcast** sobre la red local.
  - Recordar que no se propaga más allá de la red local!
- La respuesta, en cambio, es **unicast**.
- Optimización: se implementa una caché para guardar las direcciones resueltas (o conocidas).
  - Las entradas se agregan al resolver o bien al observar un pedido de otra máquina.
  - Cada entrada tiene un tiempo de expiración para evitar problemas.

Pormenores del paquete



Pormenores del paquete (cont.)

- El campo **Oper** puede tomar los valores 1 (`who-has`) o 2 (`reply`).
- Observar que la cantidad de bits asignada a las direcciones depende del valor que tomen los campos **HW size** y **Proto size**.
- Dichos campos tienen un largo de 8 bits (i.e., direcciones con un máximo de $2^8 - 1 = 255$ bits).
- **HW type** y **Proto type** indican los protocolos de nivel de enlace y de nivel de red respectivamente involucrados en la comunicación.

Ejemplo captura filtrando protocolo ARP

No.	Time	Source	Destination	Protocol	Length	Info
7	3.378592453	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
8	3.378609597	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
181	43.926159451	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
182	43.926181716	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
541	80.368730565	LgElectr_6e:00:00_Rabbit	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.39
1528	82.006955666	LgElectr_6e:00:00_Rabbit	Broadcast	ARP	42	Gratuitous ARP for 192.168.1.39 (Request)
2959	83.952682184	LgElectr_6e:00:00_Rabbit	Broadcast	ARP	42	Gratuitous ARP for 192.168.1.39 (Request)
3183	84.267498285	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
3184	84.267509869	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
6762	124.591709421	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
6763	124.591721378	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
6934	165.753903358	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
6935	165.753915173	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
7037	188.194923996	Hormiga	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.34
7089	207.366528859	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
7090	207.366541161	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
7188	240.528362452	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7189	241.443871067	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7190	242.467896088	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7193	244.721005959	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7198	245.744995959	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7215	247.700268704	Shenzhen_29:00:00_AP1	Azurewav_6a:00:00_Grillo	ARP	42	Who has 192.168.1.35? Tell 192.168.1.1
7216	247.700280541	Azurewav_6a:00:00_Grillo	Shenzhen_29:00:00_AP1	ARP	42	192.168.1.35 is at MAC1
7218	248.919217174	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7225	249.943213119	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7236	250.967527291	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7247	253.220136294	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7256	254.244159582	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7267	255.165824278	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7273	257.418670191	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7291	258.442669837	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7292	259.466710620	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7311	261.719582124	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7312	261.725751189	SamsungE_c9:00:00_Líquenes	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.38
7317	262.649721561	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1
7319	263.665513125	Shenzhen_29:00:00_AP1	Broadcast	ARP	42	Who has 192.168.1.39? Tell 192.168.1.1

Agenda

1 Introducción
- Encabezados - Tramas y Paquetes

2 ARP
- ¿Qué es ARP?
- Encabezado

3 Scapy
- Instalación
- Paquetes en Scapy

4 Sniffing
- Definiciones
- Escenarios
- Con Scapy
- Posibilidades
- Bonus
- Creando paquetes
- Más material

¿Qué es Scapy?

¿Qué es Scapy?

- Scapy es un programa de manipulación de paquetes.
- Puede crear y descifrar paquetes de un gran número de protocolos.
- Puede enviar paquetes, capturarlos, analizarlos, unir pedidos con respuestas, y mucho más.
- Amplia funcionalidad que permite reemplazar otras herramientas (nmap, arping, tcpdump, etc.).
- Multiplataforma, libre, abierto, gratis y hecho en python
- Más info $\Rightarrow$ `http://www.secdev.org/projects/scapy/`

Cómo instalar Scapy

Para python 2.*:

```
pip install scapy
sudo apt-get install python-scapy
```

Para python 3.*:

```
pip3 install scapy
sudo apt-get install python3-scapy
```

Paquetes en Scapy

```
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 8c:10:d4:94:e7:b5
  type      = 0x806
###[ ARP ]###
     hwtype    = 0x1
     ptype     = 0x800
     hwlen     = 6
     plen      = 4
     op        = who-has
     hwsrc     = 8c:10:d4:94:e7:b5
     psrc      = 0.0.0.0
     hwdst     = 00:00:00:00:00:00
     pdst      = 169.254.8.9
###[ Padding ]###
        load      = '\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00Td\x85e'


In [52]:
```

Tenemos una lista de esto... ¿qué hacemos?

Agenda

1. Introducción
   - Encabezados - Tramas y Paquetes
2. ARP
   - ¿Qué es ARP?
   - Encabezado
3. Scapy
   - Instalación
   - Paquetes en Scapy
4. Sniffing
   - Definiciones
   - Escenarios
   - Con Scapy
   - Posibilidades
   - Bonus
   - Creando paquetes
   - Más material

Algunas definiciones

- ¿NIC? Network Interface Controller (wlan0, eth0, lo, prueben haciendo ifconfig).

```
$ ifconfig
eth0:   flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
ether 3c:92:0e:33:4b:01  txqueuelen 1000  (Ethernet)
RX packets 0  bytes 0 (0.0 B)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 0  bytes 0 (0.0 B)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Algunas definiciones, cont.

Modo promiscuo

Los paquetes con MAC destino ajena no se descartan. Suben hasta el kernel para que podamos consumir las tramas. **Igual veríamos mensajes broadcast, multicast y unicast.**

Modo monitor

Permite capturar tráfico por medio del Wireless NIC, estando o no asociados con el AP o la red Ad-Hoc. En este modo se puede escuchar todo el tráfico de una red wireless.

Sudo

(No) tener permisos de root es la raíz de todos los problemas.

Escenarios

Local

- loopback
- eth, wlan, etc

Red local

- Atrás de un hub. Todos los mensajes se floodean.
- Atrás de un switch. No podemos ver mensajes ajenos. (Salvo que...)

Escuchando tráfico con Scapy

```
#! /usr/bin/env python3
from scapy.all import *

def monitor_callback(pkt):
    print(pkt.show())

if __name__ == '__main__':
  packets = sniff(prn=monitor_callback,
    iface="wlan0", filter="arp", count=1000)
```

Escuchando tráfico con Scapy

```
#! /usr/bin/env python3
from scapy.all import *

def monitor_callback(pkt):
    print(pkt.show())

if __name__ == '__main__':
  packets = sniff(prn=monitor_callback,
    iface="wlan0", filter="arp", count=1000)
```

Más información sobre los parámetros posibles, y valores por defecto: help(sniff)

Referencias

- RFC 826 (ARP) `http://tools.ietf.org/html/rfc826`
- Wireshark (página web oficial) `http://www.wireshark.org`
- Scapy (página web oficial) `http://www.secdev.org/projects/scapy/`
- Scapy Doc `https://scapy.readthedocs.io/en/latest/`
- Berkeley Packet Filter `http://biot.com/capstats/bpf.html`
- Tutoriales de Scapy `https://thepacketgeek.com/scapy-p-01-scapy-introduction-and-overview/`
- Troubleshooting modos NIC `https://www.wireshark.org/faq.html#q6.1`