Taller de Scapy

Teoría de las Comunicaciones

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

04.04.2018

Agenda

1. Introducción
   - Scapy
   - Instalación
2. Sniffing
   - Definiciones
   - Escenarios
   - Con scapy
3. Tramas y Paquetes
   - Encabezados
   - Scapy
   - Más allá
4. ARP
   - Introducción
   - Encabezado
   - Posibilidades

Agenda

1. Introducción
   - Scapy
   - Instalación
2. Sniffing
   - Definiciones
   - Escenarios
   - Con scapy
3. Tramas y Paquetes
   - Encabezados
   - Scapy
   - Más allá
4. ARP
   - Introducción
   - Encabezado
   - Posibilidades

¿Qué es Scapy?

¿Qué es Scapy?

- Scapy es un programa de manipulación de paquetes.

¿Qué es Scapy?

- Scapy es un programa de manipulación de paquetes.
- Puede crear y descifrar paquetes de un gran número de protocolos.

¿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.

¿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.).

¿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

¿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 instalarlo

Cómo instalarlo

Para python 2.*:

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

Cómo instalarlo

Para python 2.*:

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

Para python 3.*:

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

Agenda

1. Introducción
   - Scapy
   - Instalación
2. Sniffing
   - Definiciones
   - Escenarios
   - Con scapy
3. Tramas y Paquetes
   - Encabezados
   - Scapy
   - Más allá
4. ARP
   - Introducción
   - Encabezado
   - Posibilidades

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

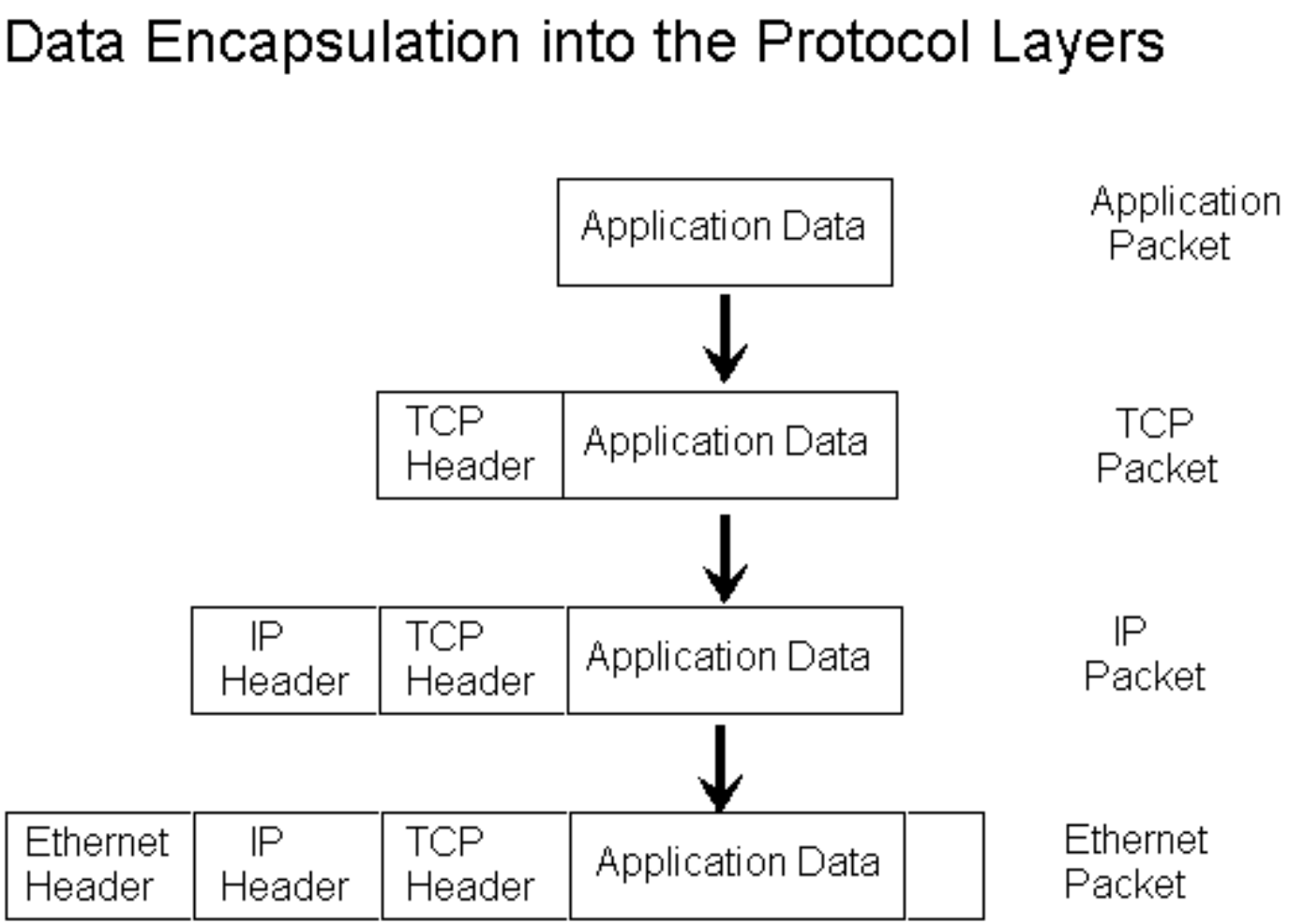
```
#! /usr/bin/env python3
from scapy.all import *

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)

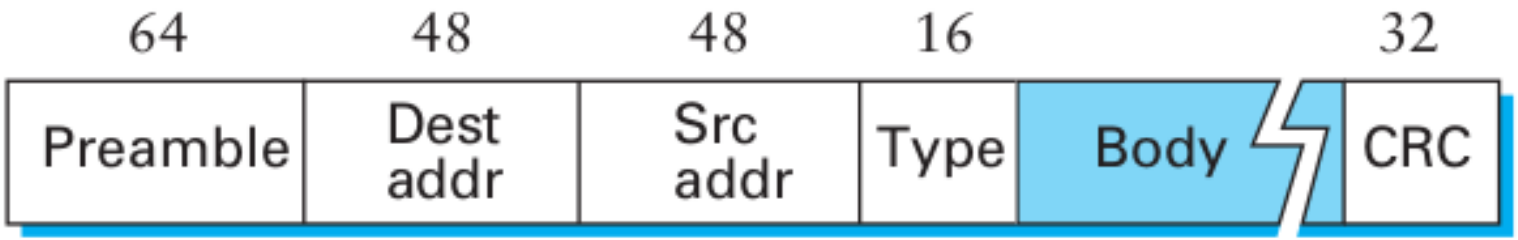
Agenda

1. Introducción
   - Scapy
   - Instalación
2. Sniffing
   - Definiciones
   - Escenarios
   - Con scapy
3. Tramas y Paquetes
   - Encabezados
   - Scapy
   - Más allá
4. ARP
   - Introducción
   - Encabezado
   - Posibilidades

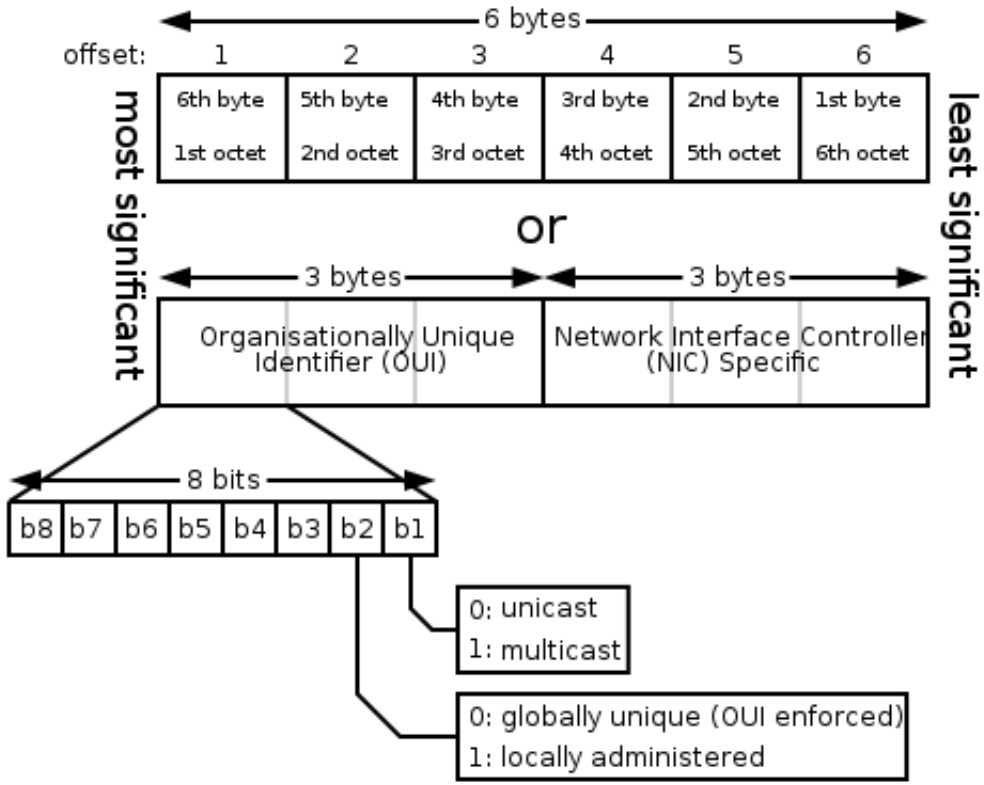
Encapsulamiento



Ethernet



Ethernet - MAC Address

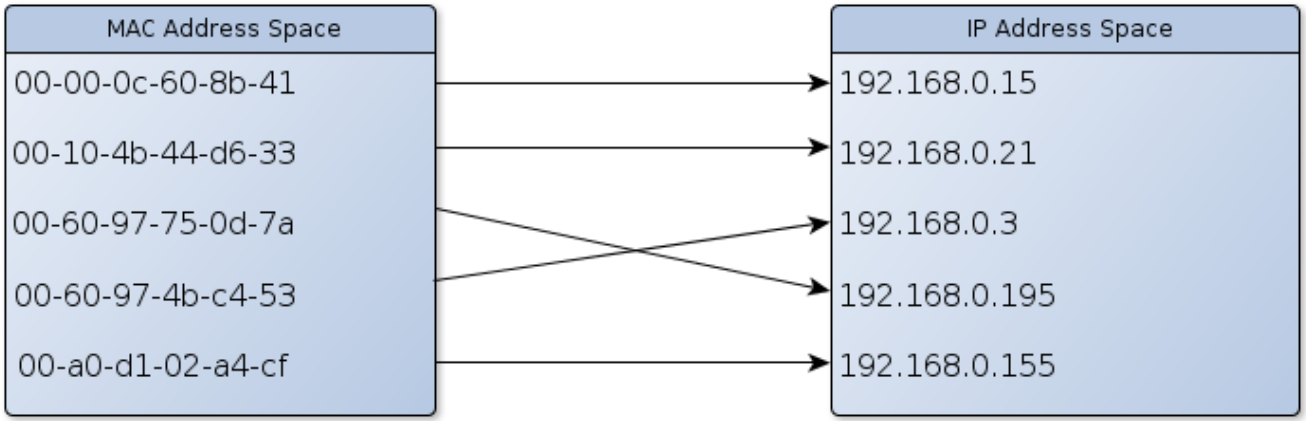


Paquetes en Scapy

```
###[ 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\x00\x00\x00\x00\x00Td\x85e'


In [52]:
```

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

¿Y las direcciones IPs?



Agenda

1. Introducción
   - Scapy
   - Instalación
2. Sniffing
   - Definiciones
   - Escenarios
   - Con scapy
3. Tramas y Paquetes
   - Encabezados
   - Scapy
   - Más allá
4. ARP
   - Introducción
   - Encabezado
   - Posibilidades

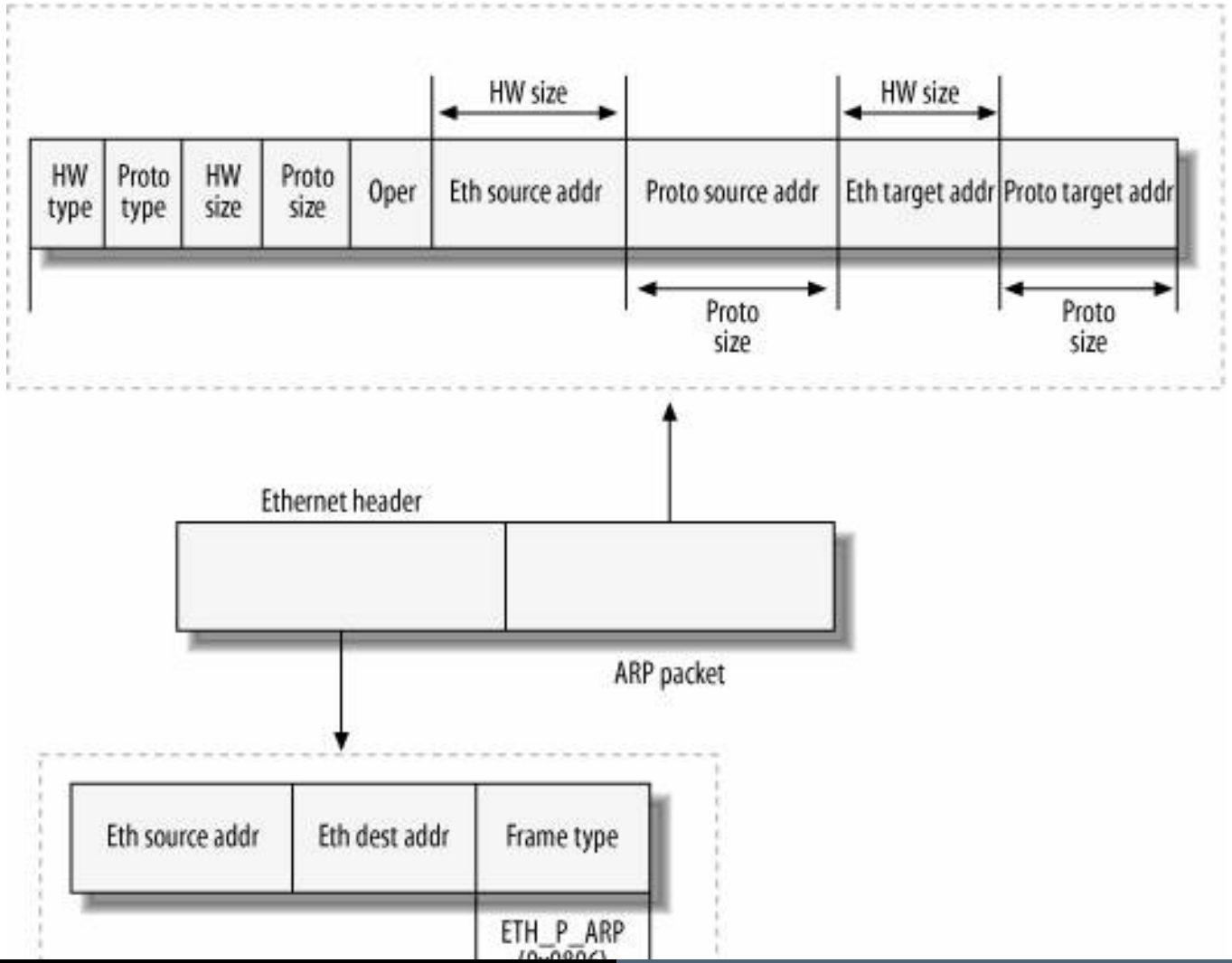
¿Qué es ARP?

- 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.

- De lo anterior se desprende que ARP es un protocolo **sin estado** y **sin seguridad**.
- La técnica de ARP spoofing se apoya precisamente en estas características.
- Idea: una máquina envía de la nada una respuesta ARP mapeando una IP objetivo con su propia MAC.
- $\Rightarrow$ todo el tráfico destinado a dicha IP va a ser recibido por ella.
- Otro ataque: MAC flooding.
- Idea: llenar la tabla ARP de un switch, para que entre en un modo a prueba de fallos
- $\Rightarrow$ =¿ Empieza a actuar como un hub, y sniffeo tranquilo.

Agenda

1. Introducción
   - Scapy
   - Instalación
2. Sniffing
   - Definiciones
   - Escenarios
   - Con scapy
3. Tramas y Paquetes
   - Encabezados
   - Scapy
   - Más allá
4. ARP
   - Introducción
   - Encabezado
   - Posibilidades

Transmitiendo

```
#!/usr/bin/env python3

import sys
from scapy.all import sr1,IP,ICMP

p=sr1(IP(dst=sys.argv[1])/ICMP())
if p:
    p.show()
```

Ejecutar este script, con una IP como argumento.

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`