

# Protocolo ARP

Práctica 2 Planificación y Administración de Redes  
Gonzalo Tudela Chavero

# ÍNDICE

---

## CONTENIDO

---

EJERCICIO 1.....	1
1. Indica cómo funciona el protocolo ARP cuando dos hosts que se encuentran en el mismo segmento de red quieren comunicarse. Ten en cuenta todos los casos posibles. Para la explicación como máximo se permitirán 200 palabras.....	1
EJERCICIO 2.....	1
1. Indica el contenido de tu caché ARP, así como el comando empleado. En el caso de que no se encuentre vacía, vacíala indicando el comando necesario. Envía un paquete de tipo ICMP a tu puerta de enlace. Analiza los mensajes y paquetes intercambiados por ambos hosts con la ayuda de Wireshark y del filtro <i>arp    icmp</i> . Se pide:.....	1
a. Dirección de destino del mensaje ARP_REQUEST.....	2
b. Dirección de destino del paquete ICMP_REQUEST, así como su tipo.....	3
c. Dirección de destino del mensaje ARP_REPLY.....	4
d. Dirección de destino del paquete ICMP_REPLY, así como su tipo.....	4
e. Contenido de la caché ARP del host que ha originado los mensajes anteriores. ....	5
EJERCICIO 3.....	6
1. Simula con Packet Tracer el entorno del punto anterior.....	6
a. Muestra el contenido de las cachés ARP en el momento inicial, después de recibir el mensaje ARP_REQUEST y después de recibir el mensaje ARP_REPLY. ....	11
b. Usa filtros para sólo monitorizar los paquetes y mensajes relacionados con los protocolos ICMP y ARP.....	12
c. Muestra el contenido de los mensajes ARP.....	12
EJERCICIO 4.....	13
1. Descarga la herramienta de auditoría Cain y Abel ( <a href="https://softfamous.com/cain-abel/download/">https://softfamous.com/cain-abel/download/</a> ). Apoyándote en el analizador de tráfico de red, se pide:.....	13
a. Realiza un escaneo de tipo ARP y explica qué hace Cain y Abel para detectar los host dentro de un segmento de red. Ayúdate del analizador de tráfico de red. ....	13
b. Del escaneo anterior muestra un mensaje ARP_REQUEST y su correspondiente ARP_REPLY, así como el contenido de la pregunta y de la respuesta. Ayúdate del analizador de tráfico de red. ....	15
EJERCICIO 5.....	16
1. Utilizando la herramienta anterior, se pide:.....	16
a. Realiza un ataque Man In The Middle empleando para ello la técnica ARP_Poison. La idea es que el atacante se ponga en el medio de la víctima y de su puerta de enlace. Documenta todo el proceso anterior de uso de Caín y Abel. ....	16
c. Una vez que hayas envenenado la caché ARP de la víctima, indica cómo será el contenido de la caché ARP, así como la del host que tiene el rol de puerta de enlace. ....	18
d. Busca un formulario en Internet haciendo “dorking” que solicite los datos de acceso por HTTP. Desde la víctima introduce información en el formulario buscado y con el analizador de tráfico de red localiza los datos introducidos en el formulario por la víctima. Puedes hacer uso del filtro “http.request.method==POST”. ....	19
EJERCICIO 6.....	21

1. (Opcional). Desde Kali Linux ( <a href="https://www.kali.org/">https://www.kali.org/</a> ) realiza un ataque MiTM con ARP_Poisson utilizando la herramienta Ettercap ( <a href="https://www.ettercap-project.org/">https://www.ettercap-project.org/</a> ). Documenta y explica todo el proceso que has realizado.....	21
--	----

## ÍNDICE DE FIGURAS

---

Figura 1. Terminal, resultado del comando: <code>arp -a</code> .....	1
Figura 2. Terminal, resultados de <code>arp -d *</code> y <code>arp -a</code> .....	2
Figura 3. Wireshark, Asustek pregunta por la MAC de 192.168.1.1 a todos (broadcast) .....	2
Figura 4. Wireshark, Ping request y desglose del Frame. ....	3
Figura 5. Wireshark, Comtrend contesta a Asustek directamente con ARP_REPLY. ....	4
Figura 6. Wireshark, ICMP_REPLY. ....	4
Figura 7. Resultado de la caché ARP tras realizar las operaciones. ....	5
Figura 8. Pracket Tracer, configuración del escenario. ....	6
Figura 9. Packet Tracer, configuración 192.168.1.2 .....	6
Figura 10. Packet Tracer, configuración 192.168.1.1 .....	7
Figura 11. Packet Tracer, Ping a 192.168.1.1 un solo paquete. ....	7
Figura 12. Packet Tracer, ICMP a la espera de la resolución ARP. ....	8
Figura 13. Packet Tracer, 192.168.1.2 envía un ARP_REQUEST. ....	8
Figura 14. Packet Tracer, 192.168.1.1 recibe el ARP_REQUEST (broadcast) y responde con ARP_REPLY. ...	9
Figura 15. Packet Tracer, 192.168.1.1 ya conoce la MAC de 192.168.1.2. ....	9
Figura 16. Packet Tracer, en este punto se prepara el paquete ICMP con los datos obtenidos de la comunicación ARP. ....	10
Figura 17. Packet Tracer, paquete ICMP alcanza su destino y se prepara la respuesta. ....	10
Figura 18. Packet Tracer, 192.168.1.2 recibe la respuesta a su petición. ....	11
Figura 19. Packet Tracer, caché ARP de 192.168.1.2 tras recibir el ARP_REPLY. ....	11
Figura 20. Packet Tracer, Contenido de ARP_REQUEST. ....	12
Figura 21. Packet Tracer, Contenido de ARP_REPLY. ....	12
Figura 22. Cain & Abel, Aviso sobre TCP. ....	13
Figura 23. Cain&Abel, enviando peticiones ARP a todo el segmento de red. ....	14
Figura 24. Wireshark, escuchando el tráfico generado por Cain&Abel. ....	15
Figura 25. Powershell, cache ARP de la víctima (192.168.1.2). ....	16
Figura 26. VMware, Cache ARP y configuración IP de la VM que realiza el ataque. ....	16
Figura 27. Cain&Abel resultado ARP Scan. ....	17
Figura 28. Cain&Abel, resultado de la regla MITM. ....	17
Figura 29. Cache ARP de la víctima donde 192.168.1.1 tiene asignada la MAC del atacante. ....	18
Figura 30. CMD mostrando la MAC atacante y la MAC real de la puerta de enlace. ....	18
Figura 31. Cain&Abel, ataque MITM exitoso mostrando login y password introducido por la víctima en el formulario de la puerta de enlace. ....	19
Figura 32. <a href="http://cms.shopcherrycreek.com/admin/login.asp">http://cms.shopcherrycreek.com/admin/login.asp</a> .....	19
Figura 33. Wireshark filtro HTTP.REQUEST.METHOD=="POST" .....	20
Figura 34. Ettercap, Unified Sniffing y seleccion de interfaz de red. ....	21
Figura 35. Ettercap, mostrando la lista de HOSTS disponibles en mi segmento. ....	21
Figura 36. Ettercap, seleccionando la variante MITM y su configuración. ....	22
Figura 37. Ettercap, mostrando la comunicación entre los objetivos designados. ....	22

## EJERCICIO 1

**1. Indica cómo funciona el protocolo ARP cuando dos hosts que se encuentran en el mismo segmento de red quieren comunicarse. Ten en cuenta todos los casos posibles. Para la explicación como máximo se permitirán 200 palabras.**

El protocolo de la capa de enlace ARP se utiliza para encontrar la dirección MAC correspondiente a una IP, enviando una petición ARP a la dirección de red broadcast, (MAC ff:ff:ff:ff:ff:ff), esta petición llegara al resto de host, respondiendo únicamente el host por cuya IP se pregunta.

Escenarios:

1. Ambos hosts desconocen las MAC del otro así que el host que inicia la comunicación envía una petición ARP a la dirección de red broadcast, el host con la IP solicitada responde con su dirección MAC al solicitante directamente ya que la IP y MAC de origen vienen implícitas en la petición, estas son agregadas a su cache ARP y, finalmente el host solicitante recibe la respuesta haciendo lo mismo con su caché.
2. El host solicitante no tiene en su cache ARP la MAC del otro (ha sido eliminada), se repetirá todo el proceso del caso 1, exceptuando la actualización de la cache ARP del receptor puesto que este conoce al solicitante.
3. El host receptor no tiene en su cache ARP la MAC del solicitante, el inicio de la comunicación será directo al receptor evitándose un ARP broadcast, el receptor agrega a su cache ARP los datos del solicitante.

## EJERCICIO 2

**1. Indica el contenido de tu caché ARP, así como el comando empleado. En el caso de que no se encuentre vacía, vacíala indicando el comando necesario. Envía un paquete de tipo ICMP a tu puerta de enlace. Analiza los mensajes y paquetes intercambiados por ambos hosts con la ayuda de Wireshark y del filtro `arp || icmp`. Se pide:**

En la siguiente figura (1) vemos un terminal CMD en el que he ejecutado el comando `arp -a`, obteniendo la caché ARP del host en el que estoy trabajando.

```

C:\Windows\system32\cmd.exe

C:\>arp -a

Interfaz: 192.168.1.2 --- 0x7
Dirección de Internet    Dirección física    Tipo
192.168.1.1              f8-8e-85-69-ca-5f  dinámico
192.168.1.34             b0-fc-0d-20-c1-f8  dinámico
192.168.1.38             20-c6-eb-b5-57-ac  dinámico
192.168.1.255            ff-ff-ff-ff-ff-ff  estático
224.0.0.2                01-00-5e-00-00-02  estático
224.0.0.22               01-00-5e-00-00-16  estático
224.0.0.251              01-00-5e-00-00-fb  estático
224.0.0.252              01-00-5e-00-00-fc  estático
239.255.255.250          01-00-5e-7f-ff-fa  estático
255.255.255.255          ff-ff-ff-ff-ff-ff  estático

Interfaz: 192.168.193.1 --- 0x12
Dirección de Internet    Dirección física    Tipo
192.168.193.254          00-50-56-e6-14-14  dinámico
192.168.193.255          ff-ff-ff-ff-ff-ff  estático
224.0.0.2                01-00-5e-00-00-02  estático
224.0.0.22               01-00-5e-00-00-16  estático
224.0.0.251              01-00-5e-00-00-fb  estático
224.0.0.252              01-00-5e-00-00-fc  estático
239.255.255.250          01-00-5e-7f-ff-fa  estático
255.255.255.255          ff-ff-ff-ff-ff-ff  estático

Interfaz: 169.254.89.218 --- 0x16
Dirección de Internet    Dirección física    Tipo
169.254.255.255          ff-ff-ff-ff-ff-ff  estático
224.0.0.2                01-00-5e-00-00-02  estático
224.0.0.22               01-00-5e-00-00-16  estático
224.0.0.251              01-00-5e-00-00-fb  estático
224.0.0.252              01-00-5e-00-00-fc  estático
239.255.255.250          01-00-5e-7f-ff-fa  estático
255.255.255.255          ff-ff-ff-ff-ff-ff  estático

Interfaz: 192.168.85.1 --- 0x19
Dirección de Internet    Dirección física    Tipo
192.168.85.254           00-50-56-e0-f6-31  dinámico
192.168.85.255           ff-ff-ff-ff-ff-ff  estático
224.0.0.2                01-00-5e-00-00-02  estático
224.0.0.22               01-00-5e-00-00-16  estático
224.0.0.251              01-00-5e-00-00-fb  estático
224.0.0.252              01-00-5e-00-00-fc  estático
239.255.255.250          01-00-5e-7f-ff-fa  estático
255.255.255.255          ff-ff-ff-ff-ff-ff  estático

```

Figura 1. Terminal, resultado del comando: `arp -a`

En la siguiente figura (2) se puede comprobar el resultado del comando `arp -d *` el cual ha sido necesario realizar en una terminal con privilegios de administrador para poder ser ejecutado, para ver su resultado vuelvo a usar `arp -a` para ver comprobar los cambios en la caché.

```
C:\>arp -d *

C:\>arp -a

Interfaz: 192.168.1.2 --- 0x7
  Dirección de Internet      Dirección física      Tipo
192.168.1.255              ff-ff-ff-ff-ff-ff    estático
224.0.0.2                  01-00-5e-00-00-02    estático
224.0.0.22                 01-00-5e-00-00-16    estático

Interfaz: 192.168.193.1 --- 0x12
  Dirección de Internet      Dirección física      Tipo
192.168.193.255            ff-ff-ff-ff-ff-ff    estático
224.0.0.2                  01-00-5e-00-00-16    estático

Interfaz: 169.254.89.218 --- 0x16
  Dirección de Internet      Dirección física      Tipo
169.254.255.255            ff-ff-ff-ff-ff-ff    estático
224.0.0.2                  01-00-5e-00-00-16    estático

Interfaz: 192.168.85.1 --- 0x19
  Dirección de Internet      Dirección física      Tipo
192.168.85.255             ff-ff-ff-ff-ff-ff    estático
224.0.0.2                  01-00-5e-00-00-16    estático

C:\>
```

Figura 2. Terminal, resultados de `arp -d *` y `arp -a`

**a. Dirección de destino del mensaje ARP\_REQUEST.**

En la siguiente figura (3) se puede ver en la fila 1, como mi máquina con el NIC (Asustek) realiza un `ARP_REQUEST` a la dirección MAC de broadcast (ff:ff:ff:ff:ff:ff) solicitando la MAC de la puerta de enlace (Comtrend), ya que la caché no dispone de esa entrada.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	AsustekC_b4:63:b5	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.2
2	0.000216	Comtrend_69:ca:5f	AsustekC_b4:63:b5	ARP	60	192.168.1.1 is at f8:8e:85:69:ca:5f
3	0.000226	192.168.1.2	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=123/31488, ttl=128 (reply in 4)
4	0.000531	192.168.1.1	192.168.1.2	ICMP	74	Echo (ping) reply id=0x0001, seq=123/31488, ttl=64 (request in 3)

> Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0  
> Ethernet II, Src: AsustekC\_b4:63:b5 (e0:3f:49:b4:63:b5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
> Destination: Broadcast (ff:ff:ff:ff:ff:ff)  
> Source: AsustekC\_b4:63:b5 (e0:3f:49:b4:63:b5)  
> Type: ARP (0x0806)  
> Address Resolution Protocol (request)

Figura 3. Wireshark, Asustek pregunta por la MAC de 192.168.1.1 a todos (broadcast)

**b. Dirección de destino del paquete ICMP\_REQUEST, así como su tipo.**

En la siguiente figura (4) se ve como en la fila 3, se envía el paquete ICMP tipo 8 (*Echo request*) a la puerta de enlace, resulta interesante como *Wireshark* ordena la información y desglosa el *Frame* desde la capa más interna hasta llegar al protocolo *ICMP*.

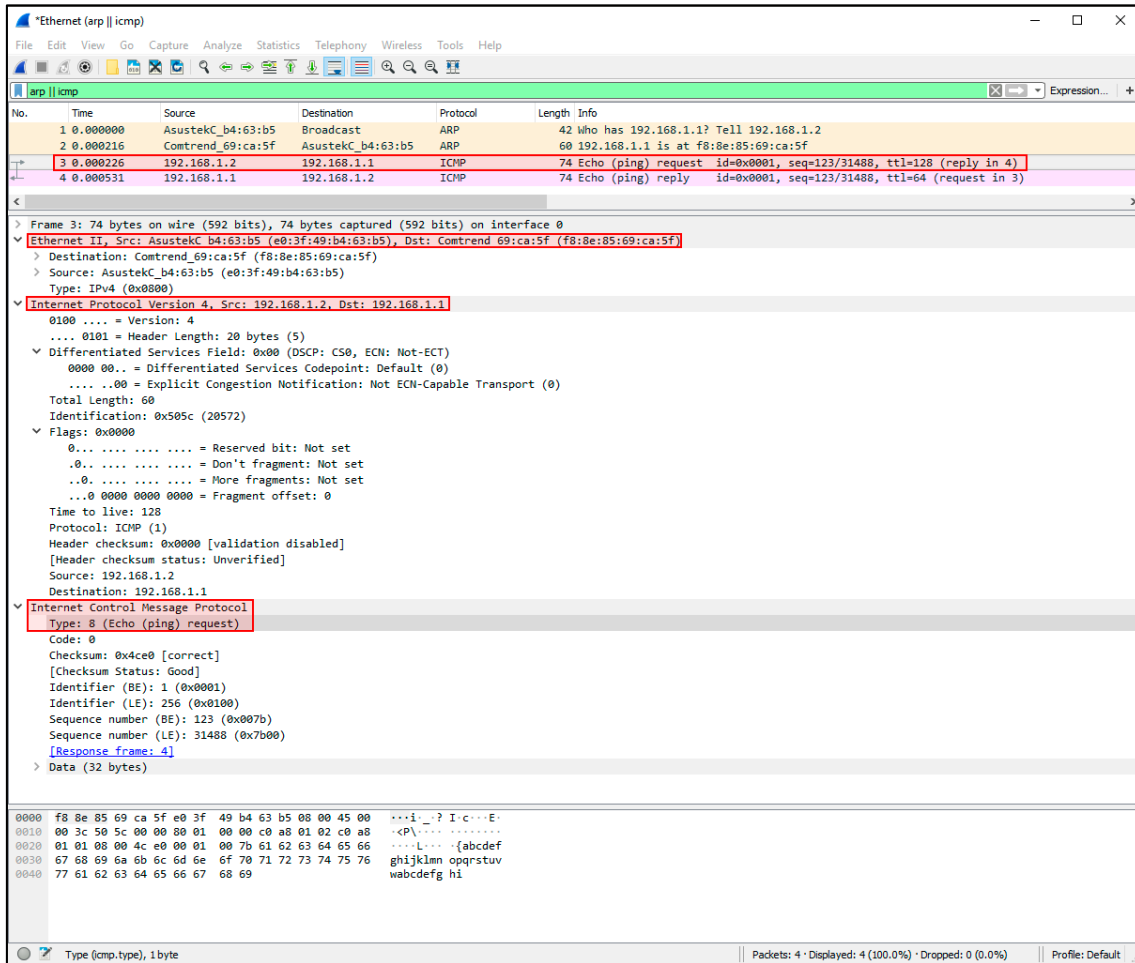


Figura 4. Wireshark, Ping request y desglose del Frame.

### c. Dirección de destino del mensaje ARP\_REPLY.

En la siguiente figura (5) de arriba a abajo se puede ver resaltada en rojo la fila 2, donde es enviado el paquete *ARP\_REPLY*, y más abajo se ha resaltado varias veces la fuente (*Src*) y el destino (*Dst*) del *ARP\_REPLY*.

Se aprecia que ya no es una respuesta broadcast, es dirigida al demandante. Puesto que el paquete *REQUEST* contenía la MAC de este.

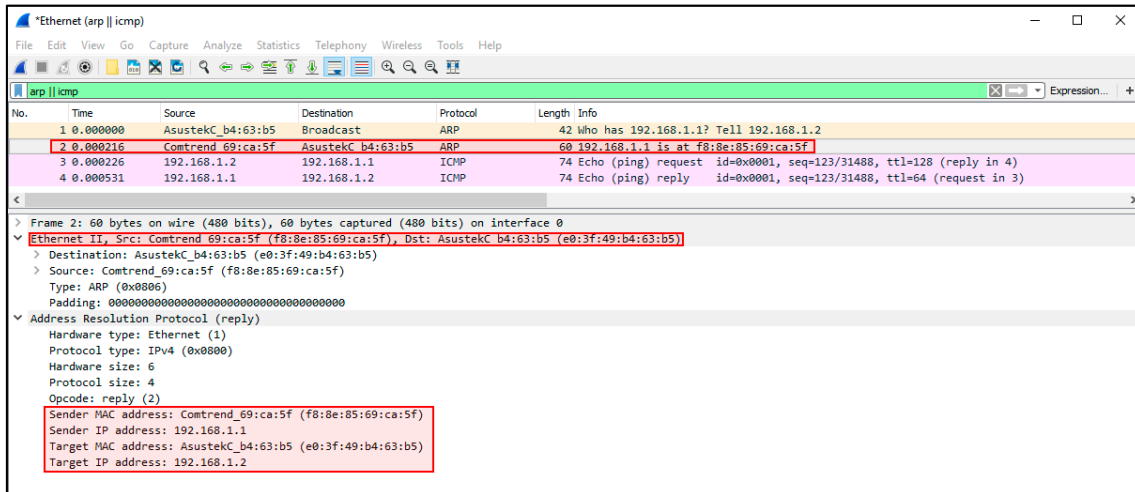


Figura 5. Wireshark, Comtrend contesta a Asustek directamente con *ARP\_REPLY*.

### d. Dirección de destino del paquete ICMP\_REPLY, así como su tipo.

En la siguiente figura (6) se puede ver como la puerta de enlace responde con un paquete *ICMP\_REPLY* tipo 0 (*Echo Reply*).

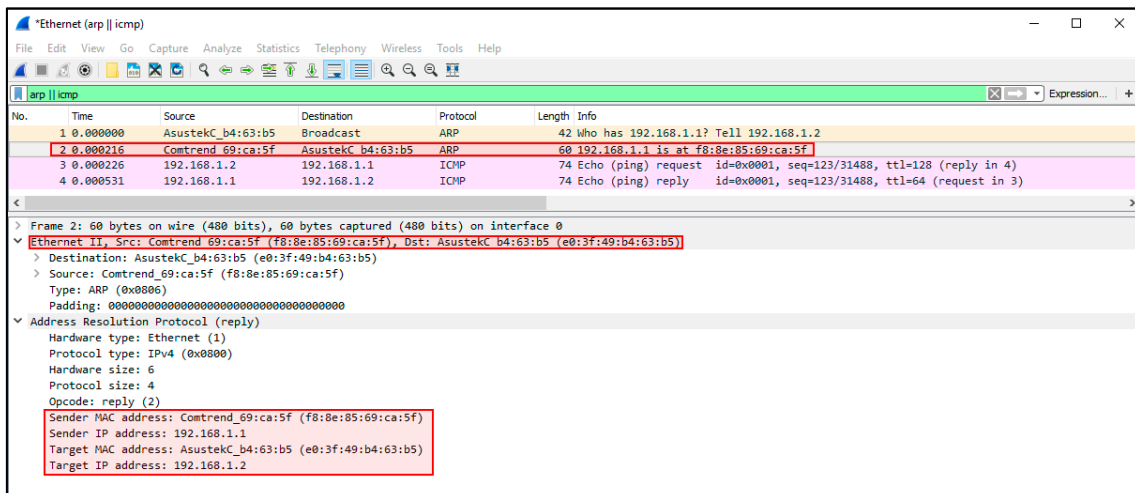


Figura 6. Wireshark, ICMP\_REPLY.

**e. Contenido de la caché ARP del host que ha originado los mensajes anteriores.**

En la siguiente figura (7) podemos ver como se elimina la caché ARP con el comando `arp -d *`, posteriormente realizo un `ping`, que en este caso le he añadido la opción `-n 1` para evitar un mayor número de entradas en *Wireshark*, ya que esta opción indica a *Ping* que solo envíe una petición.

```

C:\>arp -d *

C:\>ping 192.168.1.1 -n 1

Haciendo ping a 192.168.1.1 con 32 bytes de datos:
Respuesta desde 192.168.1.1: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.1.1:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\>arp -a

Interfaz: 192.168.1.2 --- 0x7
  Dirección de Internet      Dirección física      Tipo
  192.168.1.1                f8-8e-85-69-ca-5f    dinámico
  192.168.1.86               00-1b-a9-d1-4a-b0    dinámico
  192.168.1.255              ff-ff-ff-ff-ff-ff    estático
  224.0.0.2                  01-00-5e-00-00-02    estático
  224.0.0.9                  01-00-5e-00-00-09    estático
  224.0.0.22                 01-00-5e-00-00-16    estático
  224.0.0.251                01-00-5e-00-00-fb    estático
  224.0.0.252                01-00-5e-00-00-fc    estático
  239.255.255.250            01-00-5e-7f-ff-fa    estático

Interfaz: 192.168.193.1 --- 0x12
  Dirección de Internet      Dirección física      Tipo
  192.168.193.254            00-50-56-e6-14-14    dinámico
  192.168.193.255            ff-ff-ff-ff-ff-ff    estático
  224.0.0.2                  01-00-5e-00-00-02    estático
  224.0.0.9                  01-00-5e-00-00-09    estático
  224.0.0.22                 01-00-5e-00-00-16    estático
  224.0.0.251                01-00-5e-00-00-fb    estático
  224.0.0.252                01-00-5e-00-00-fc    estático
  239.255.255.250            01-00-5e-7f-ff-fa    estático

Interfaz: 169.254.89.218 --- 0x16
  Dirección de Internet      Dirección física      Tipo
  169.254.255.255            ff-ff-ff-ff-ff-ff    estático
  224.0.0.2                  01-00-5e-00-00-02    estático
  224.0.0.9                  01-00-5e-00-00-09    estático
  224.0.0.22                 01-00-5e-00-00-16    estático
  224.0.0.251                01-00-5e-00-00-fb    estático
  224.0.0.252                01-00-5e-00-00-fc    estático
  239.255.255.250            01-00-5e-7f-ff-fa    estático
  255.255.255.255            ff-ff-ff-ff-ff-ff    estático

Interfaz: 192.168.85.1 --- 0x19
  Dirección de Internet      Dirección física      Tipo
  192.168.85.254            00-50-56-e0-f6-31    dinámico
  192.168.85.255            ff-ff-ff-ff-ff-ff    estático
  224.0.0.2                  01-00-5e-00-00-02    estático
  224.0.0.9                  01-00-5e-00-00-09    estático
  224.0.0.22                 01-00-5e-00-00-16    estático
  224.0.0.251                01-00-5e-00-00-fb    estático
  224.0.0.252                01-00-5e-00-00-fc    estático
  239.255.255.250            01-00-5e-7f-ff-fa    estático

C:\>
  
```

Figura 7. Resultado de la caché ARP tras realizar las operaciones.



## EJERCICIO 3

### 1. Simula con Packet Tracer el entorno del punto anterior.

Para simular con *Packet Tracer* el entorno anterior creo 2 equipos y los configuro con las IP que teníamos en el ejercicio anterior, salvando la diferencia que uno de ellos era un *router* y ahora es un PC, lo que a efectos de esta prueba no supone ninguna diferencia.

En la siguiente figura (8), tras crear los 2 equipos y configurarles una IP, conecto las maquinas con un cable directo entre ellas, configuro *Packet Tracer* en modo simulación y selecciono el filtro para los paquetes *ICMP* y *ARP*.

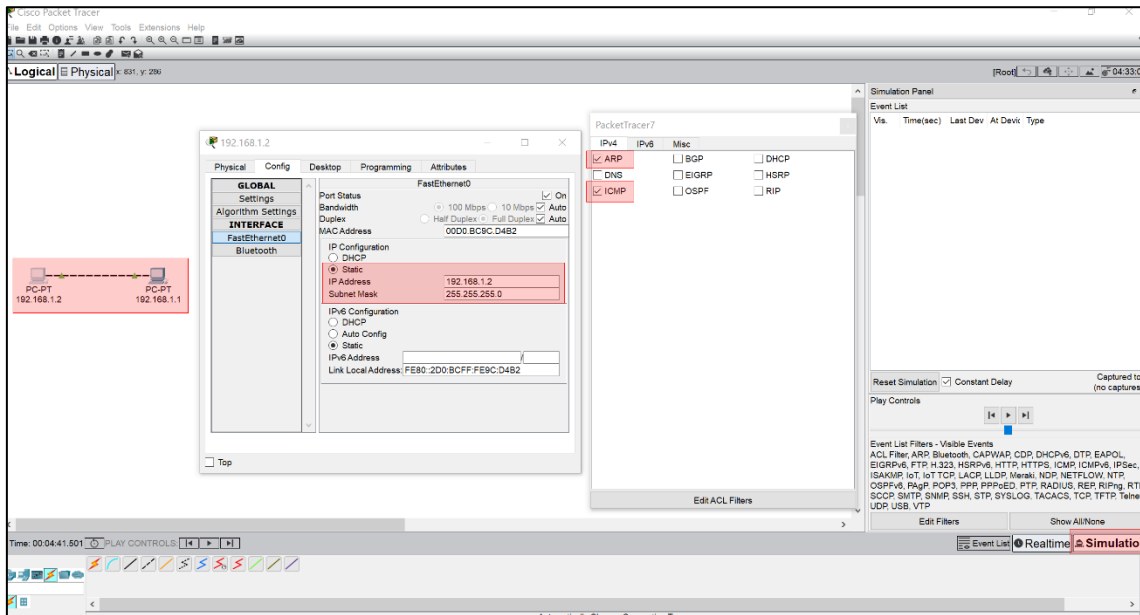


Figura 8. *Packet Tracer*, configuración del escenario.

Desde la consola del equipo 192.168.1.2, figura (9) compruebo que su configuración es correcta con *ipconfig*, también compruebo que su cache *ARP* está vacía, estas operaciones las realizo también con la máquina 192.168.1.1. figura (10).

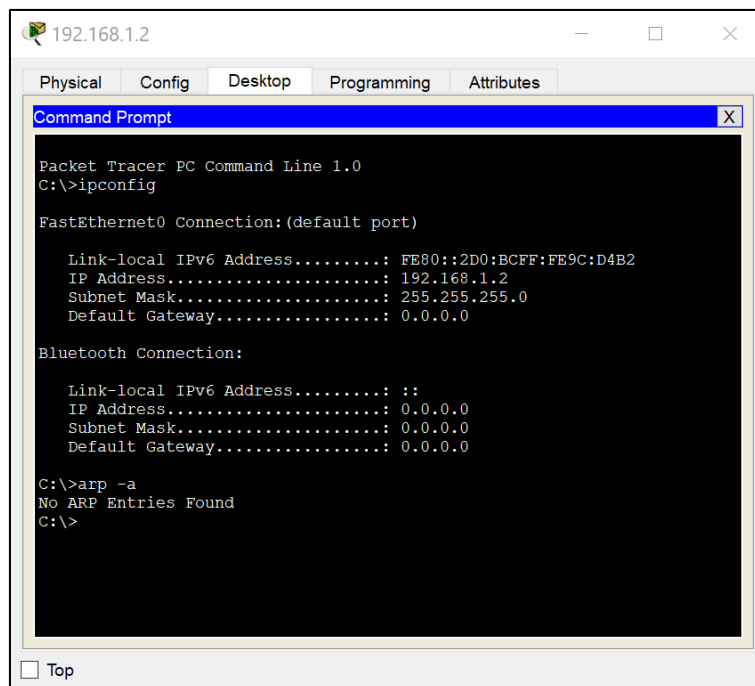


Figura 9. *Packet Tracer*, configuración 192.168.1.2

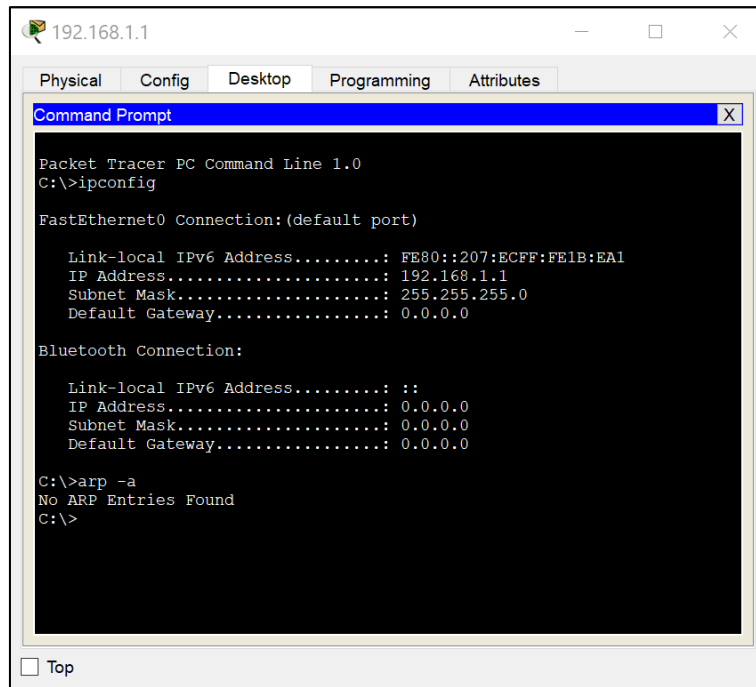


Figura 10. Packet Tracer, configuración 192.168.1.1

Ahora desde la consola, figura (11), de 192.168.1.2 realizo un (ping 192.168.1.1 -n 1) para enviar una sola petición.

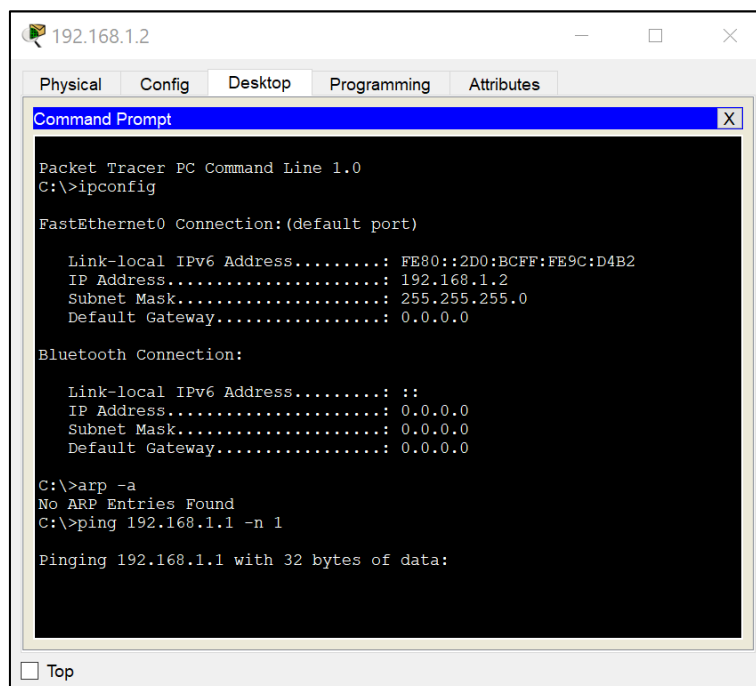


Figura 11. Packet Tracer, Ping a 192.168.1.1 un solo paquete.

Observamos en la figura (12) como el paquete *ICMP* se pone a la espera, ya que, primero se deben resolver las direcciones *MAC* de las IP con las que trabaja *ICMP* ya que este pertenece a otra capa del esquema *OSI* y necesita de *ARP* (capa inferior) para realizar este trabajo.

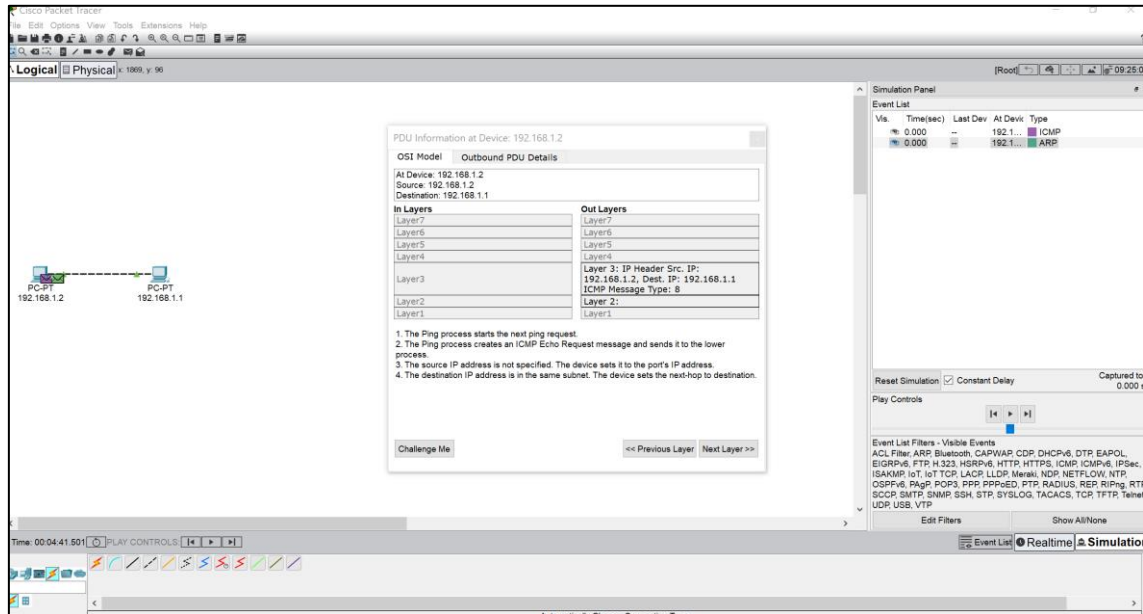


Figura 12. Packet Tracer, ICMP a la espera de la resolución ARP.

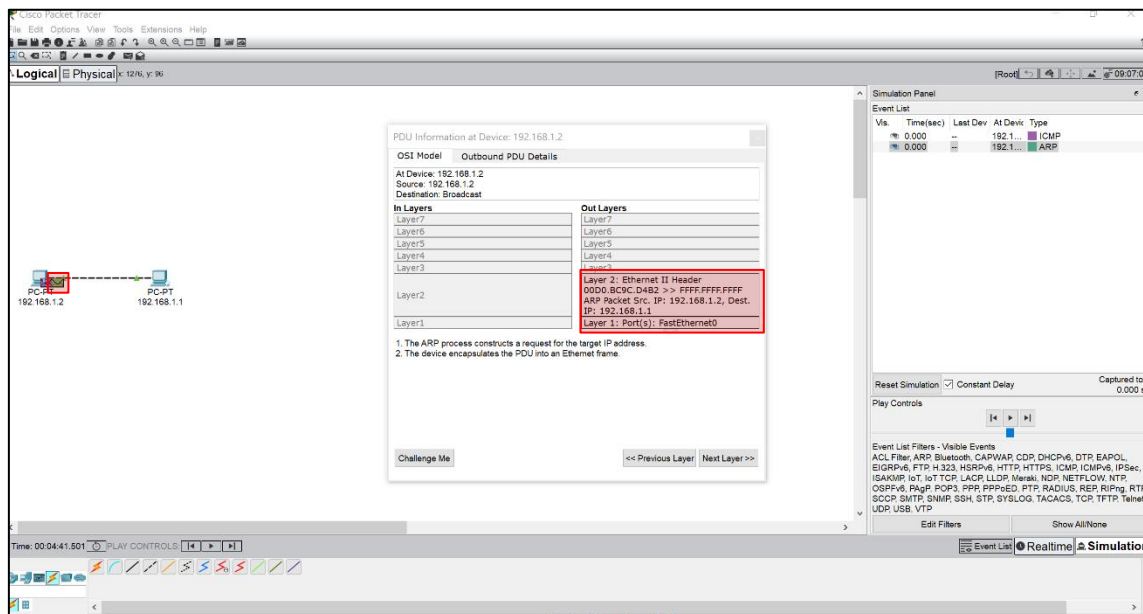


Figura 13. Packet Tracer, 192.168.1.2 envía un ARP\_REQUEST.

En la figura anterior (13) también podemos ver cómo 192.168.1.2 envía un paquete *ARP\_REQUEST* a la *MAC broadcast* (FFFF.FFFF.FFFF) preguntando por la *MAC* correspondiente a 192.168.1.1.

Finalmente 192.168.1.1 contesta a 192.168.1.2 directamente con un paquete *ARP\_REPLY* indicando su MAC, figura (14).

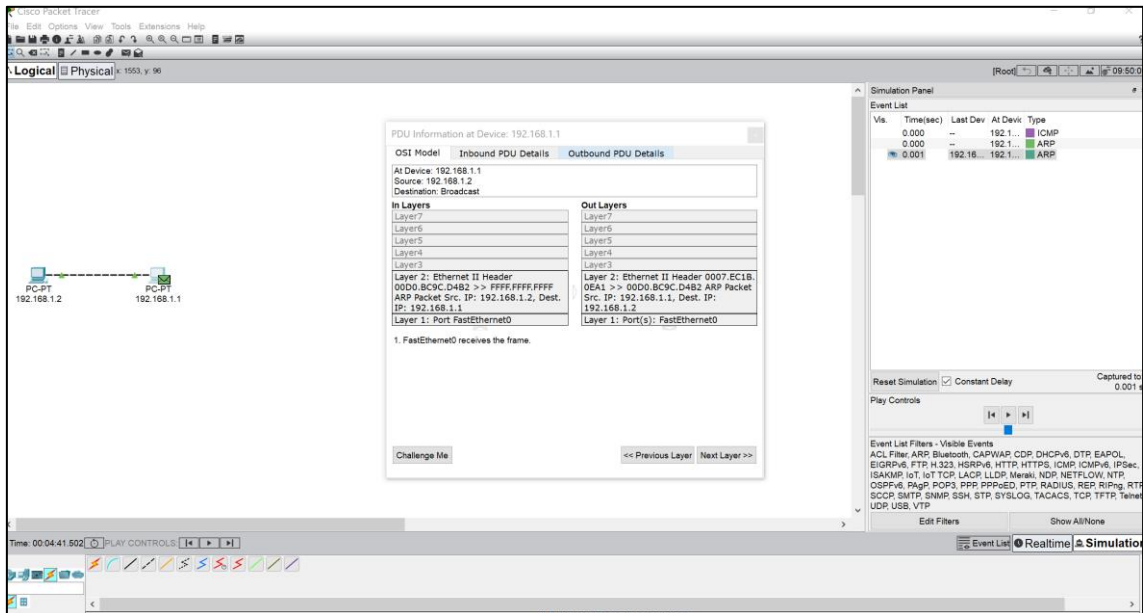


Figura 14. *Packet Tracer*, 192.168.1.1 recibe el ARP\_REQUEST (broadcast) y responde con ARP\_REPLY.

Como se observa en la figura (15) el equipo destino del ping 192.168.1.1 ya tiene en su caché ARP la IP y la MAC del equipo origen 192.168.1.2 y este también los del destino.

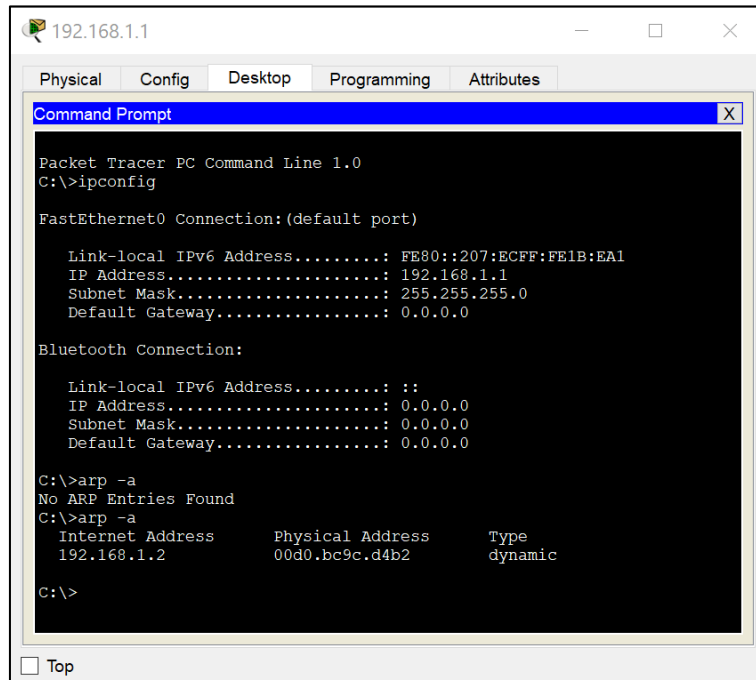


Figura 15. *Packet Tracer*, 192.168.1.1 ya conoce la MAC de 192.168.1.2

Ahora el paquete *ICMP* se monta con los datos que necesitaba, figura (16).

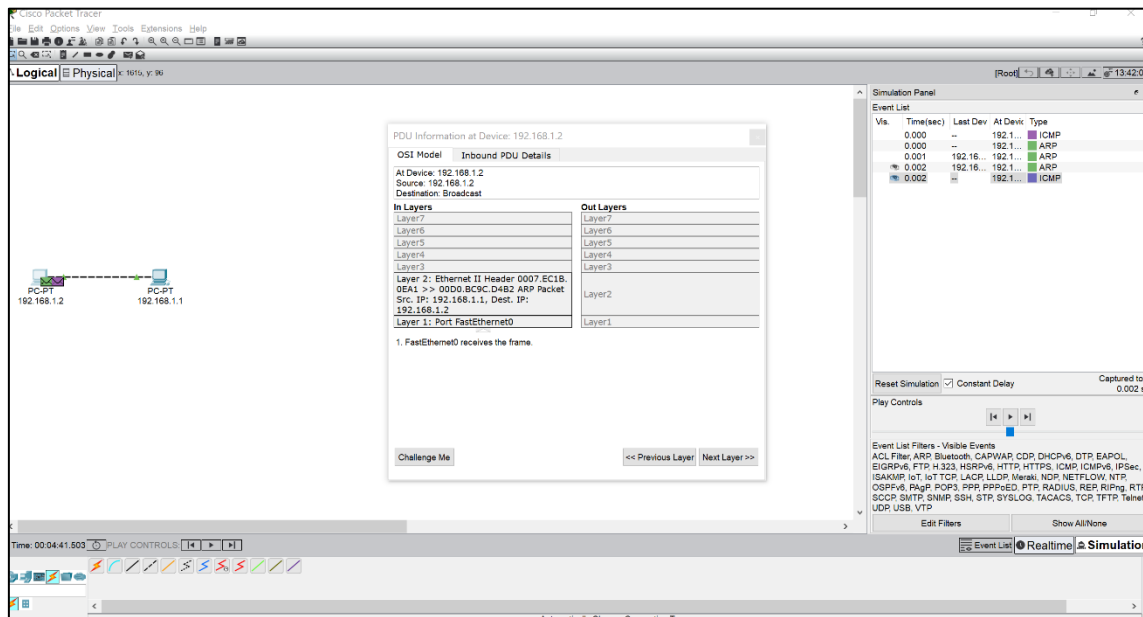


Figura 16. *Packet Tracer*, en este punto se prepara el paquete ICMP con los datos obtenidos de la comunicación ARP.

Finalmente se realiza la comunicación solicitada por el comando *ping*, figura (17).

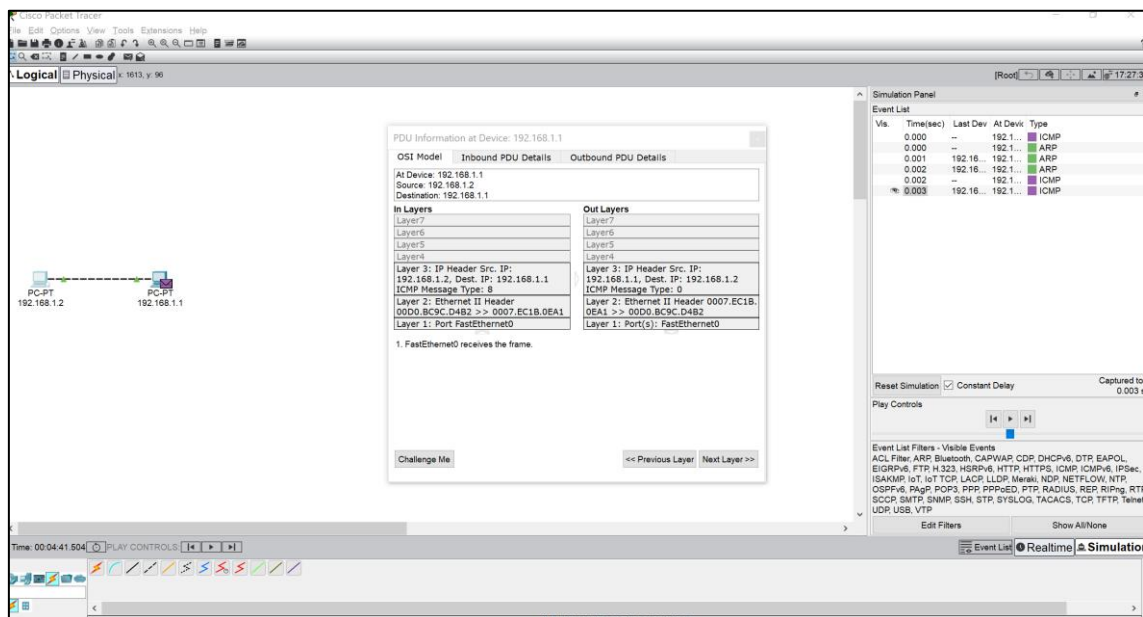


Figura 17. *Packet Tracer*, paquete ICMP alcanza su destino y se prepara la respuesta.

En esta figura (18) se puede ver como el paquete *ICMP* enviado por 192.168.1.1 es recibido por 192.168.1.2.

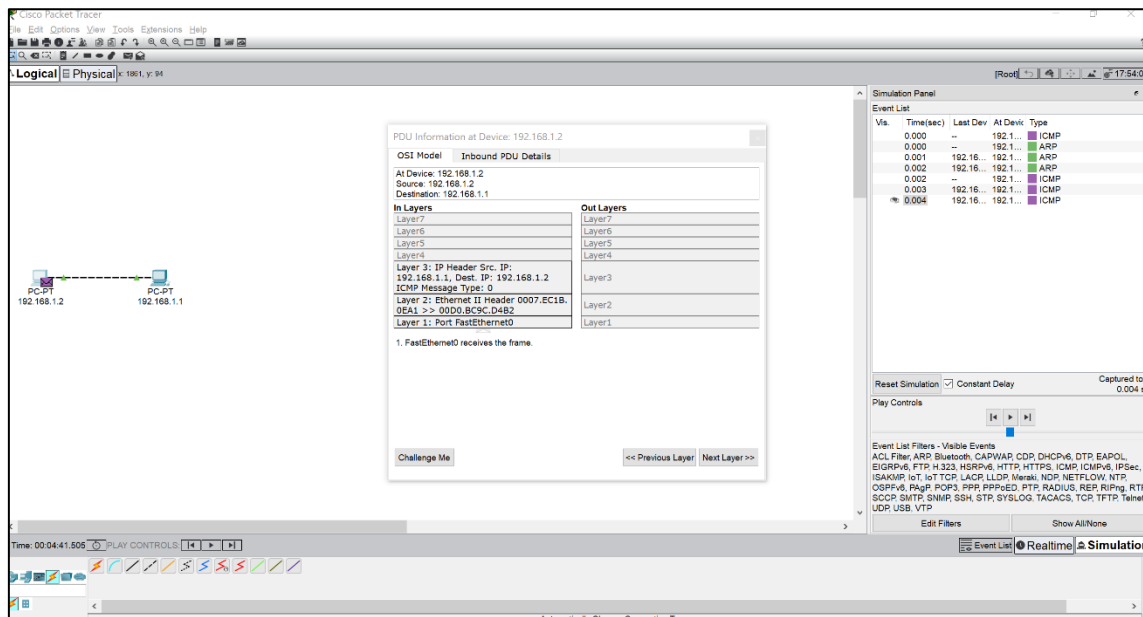


Figura 18. *Packet Tracer*, 192.168.1.2 recibe la respuesta a su petición.

**a. Muestra el contenido de las cachés ARP en el momento inicial, después de recibir el mensaje ARP\_REQUEST y después de recibir el mensaje ARP\_REPLY.**

La respuesta al momento inicial de esta pregunta aparece en las figuras siguientes (links):

Figura 9. *Packet Tracer*, configuración 192.168.1.2

Figura 10. *Packet Tracer*, configuración 192.168.1.1

El estado final tras recibir los mensajes aparece en las siguientes figuras (links):

Figura 15. *Packet Tracer*, 192.168.1.1 ya conoce la MAC de 192.168.1.2

En la siguiente figura ( ) se ve el estado de la cache ARP de 192.168.1.2 tras recibir el paquete ARP\_REPLY.

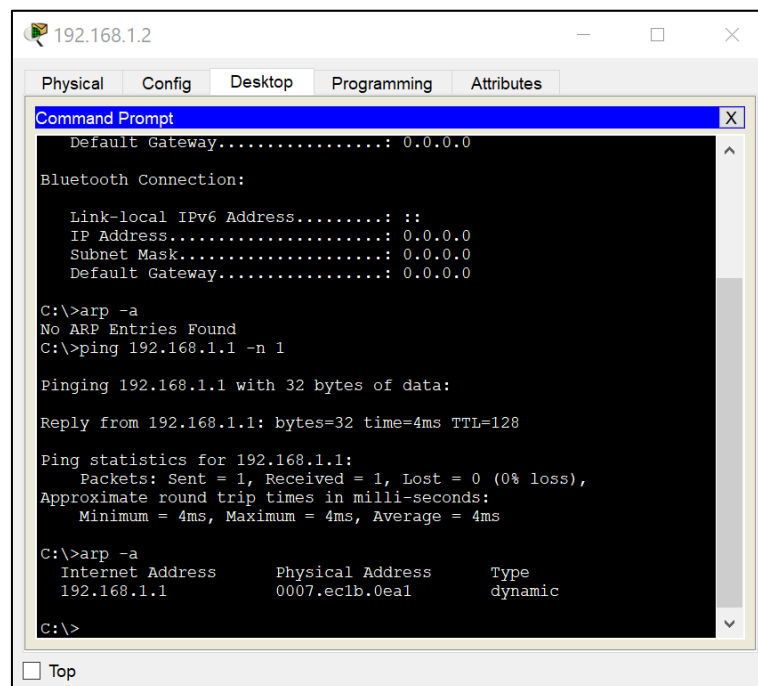


Figura 19. *Packet Tracer*, caché ARP de 192.168.1.2 tras recibir el ARP\_REPLY.

b. Usa filtros para sólo monitorizar los paquetes y mensajes relacionados con los protocolos ICMP y ARP.

Esta pregunta quedo respondida en la Figura 8. *Pracket Tracer*, configuración del escenario. (link). Donde se aprecia la ventana de filtros, siendo ARP e ICMP los únicos marcados.

c. Muestra el contenido de los mensajes ARP.

En las siguientes figuras se muestra el contenido de los paquetes ARP, tanto el ARP\_REQUEST (figura 20) como el ARP\_REPLY (figura 21).

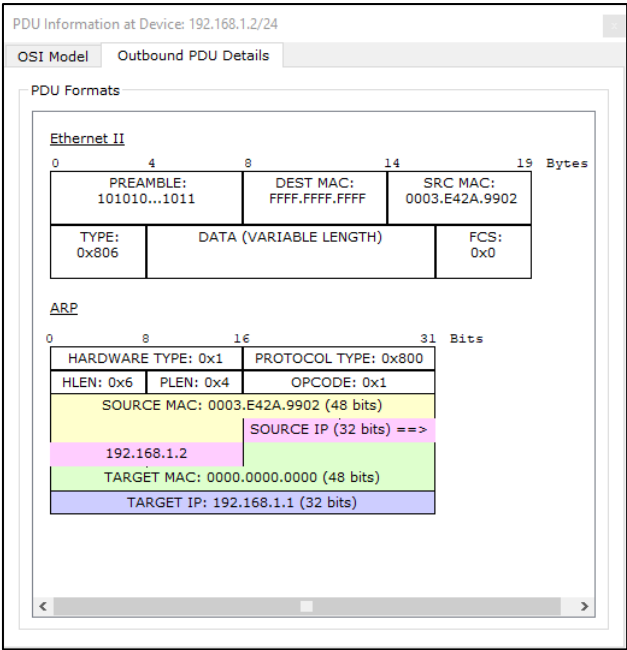


Figura 20. Packet Tracer, Contenido de ARP\_REQUEST.

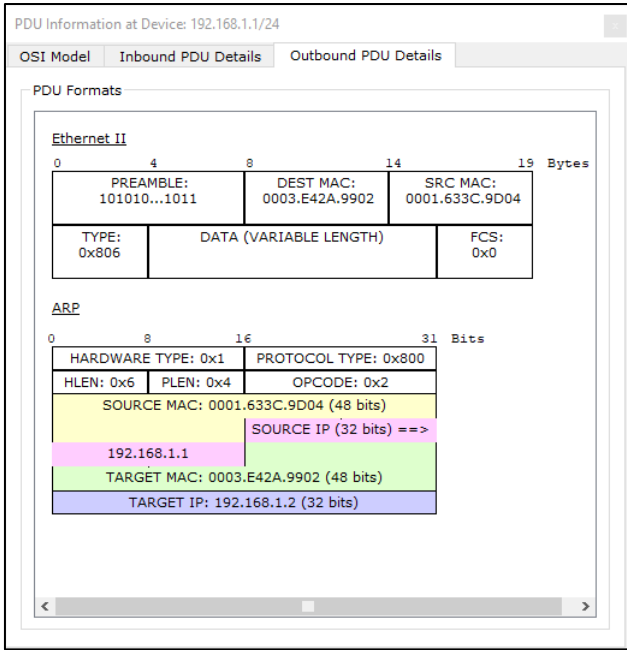


Figura 21. Packet Tracer, Contenido de ARP\_REPLY.

## EJERCICIO 4

1. Descarga la herramienta de auditoría Cain y Abel (<https://softfamous.com/cain-abel/download/>). Apoyándote en el analizador de tráfico de red, se pide:

- Realiza un escaneo de tipo ARP y explica qué hace Cain y Abel para detectar los host dentro de un segmento de red. Ayúdate del analizador de tráfico de red.

Para comenzar y por motivos de compatibilidad he decidido realizar esta parte en una máquina virtual con Windows XP Profesional con su adaptador de red configurado en modo bridge. Tras la instalación, el programa avisa de un posible problema (figura 22) en el manejo de los paquetes TCP/IP, y recomienda modificar la configuración sugiriendo un comando, que deshabilita la segmentación de paquetes TCP por el hardware ethernet, realizándose en la máquina (CPU).

Tras buscar información al respecto en internet encuentro el siguiente artículo.

(<http://www.peerwisdom.org/2013/04/03/large-send-offload-and-network-performance/>).

A continuación, se expone un pequeño resumen.

*Una característica de los NIC modernos es que ya que se permite construir un mensaje grande en la pila TCP/IP (hasta 64KB) el cual es enviado al adaptador de red, el cual se encargará de segmentarlo en paquetes de datos más pequeños conocidos como frames que podrán ser enviados a través del cable. Esto, libera a la CPU de tener que manejar el segmentado de mensajes grades TCP para que entren dentro del tamaño de frame especificado, lo que significa un mejor desempeño de la comunicación, en teoría.*

*Lo que en la realidad sucede es que todos los demás NIC (en el blog, Switches) de la red tienen que tener una configuración igual de tamaño de frame. Una máquina no puede enviar frames que sean mayores que la Unidad Máxima de Transmisión (MTU) soportada por el NIC de destino, ya que estos serán descartados y la NIC que los envía quedará a la espera de recibir la contestación lo que en tiempos de computación es mucho tiempo, así que la máquina que envía el paquete reintentará el envío haciendo una petición de transmisión y reconstruirá el mensaje TCP esta vez teniendo en cuenta la segmentación por si misma (CPU) ya que la máquina si puede preguntar por el MTU adecuado. Este diseño está pensado para evitar este tipo de problemas causados por descargar ese trabajo en el hardware del NIC, que como vemos no es capaz de preguntar por sí mismo por el tamaño adecuado de transmisión (MTU).*

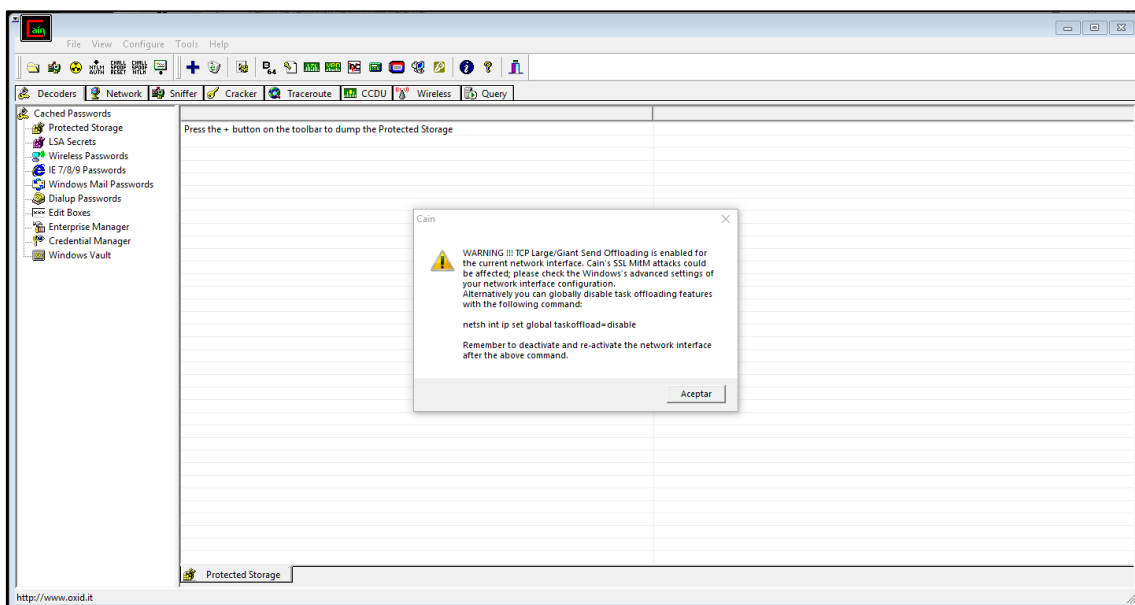


Figura 22. Cain&Abel, Aviso sobre TCP.



Dejando el problema sobre la pila TCP solucionado tras hacer los cambios que solicita el programa, procedemos a realizar una búsqueda de Host mediante el envío de paquetes ARP\_REQUEST (broadcast) a todos los posibles Host del segmento de red. (figura 23)

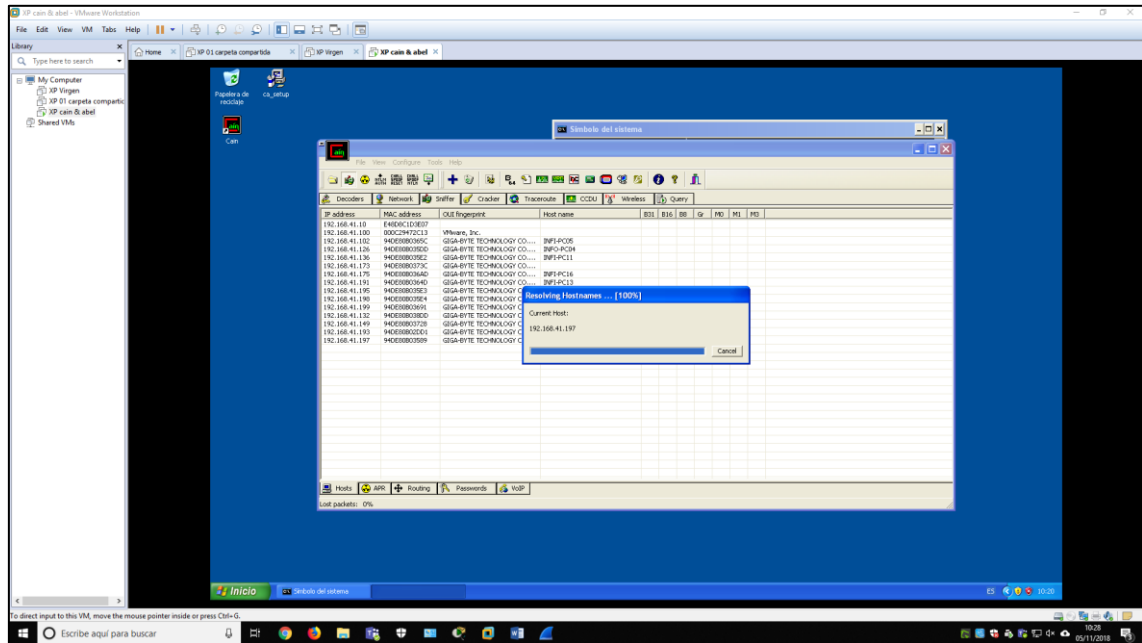


Figura 23. Cain&Abel, enviando peticiones ARP a todo el segmento de red.

En este caso, como se realiza en el aula la red es 192.168.41.x. lo que da como resultado la lista de todos aquellos equipos que están funcionando en el segmento. Esta forma de descubrir hosts en un segmento es ideal ya que los paquetes ARP pertenecen a la capa de enlace de datos y no son bloqueados.

Nótese que esto es solo aplicable al segmento de red ya que los paquetes *ARP\_REPLY* no viajan a otras redes ya que de ese trabajo se encargan otros dispositivos.

b. Del escaneo anterior muestra un mensaje *ARP\_REQUEST* y su correspondiente *ARP\_REPLY*, así como el contenido de la pregunta y de la respuesta. Ayúdate del analizador de tráfico de red.

En la siguiente figura (24) se puede ver una captura de Wireshark capturando los paquetes ARP enviados por *Cain & Abel* (192.168.41.103) para descubrir las maquinas que respondan a peticiones ARP en el segmento de red en el que me encuentro (192.168.41.x).

*¿Who has 192.168.41.3?* (quien tiene 192.168.41.3) *Tell 192.168.41.103* (decir a 192.168.41.103)

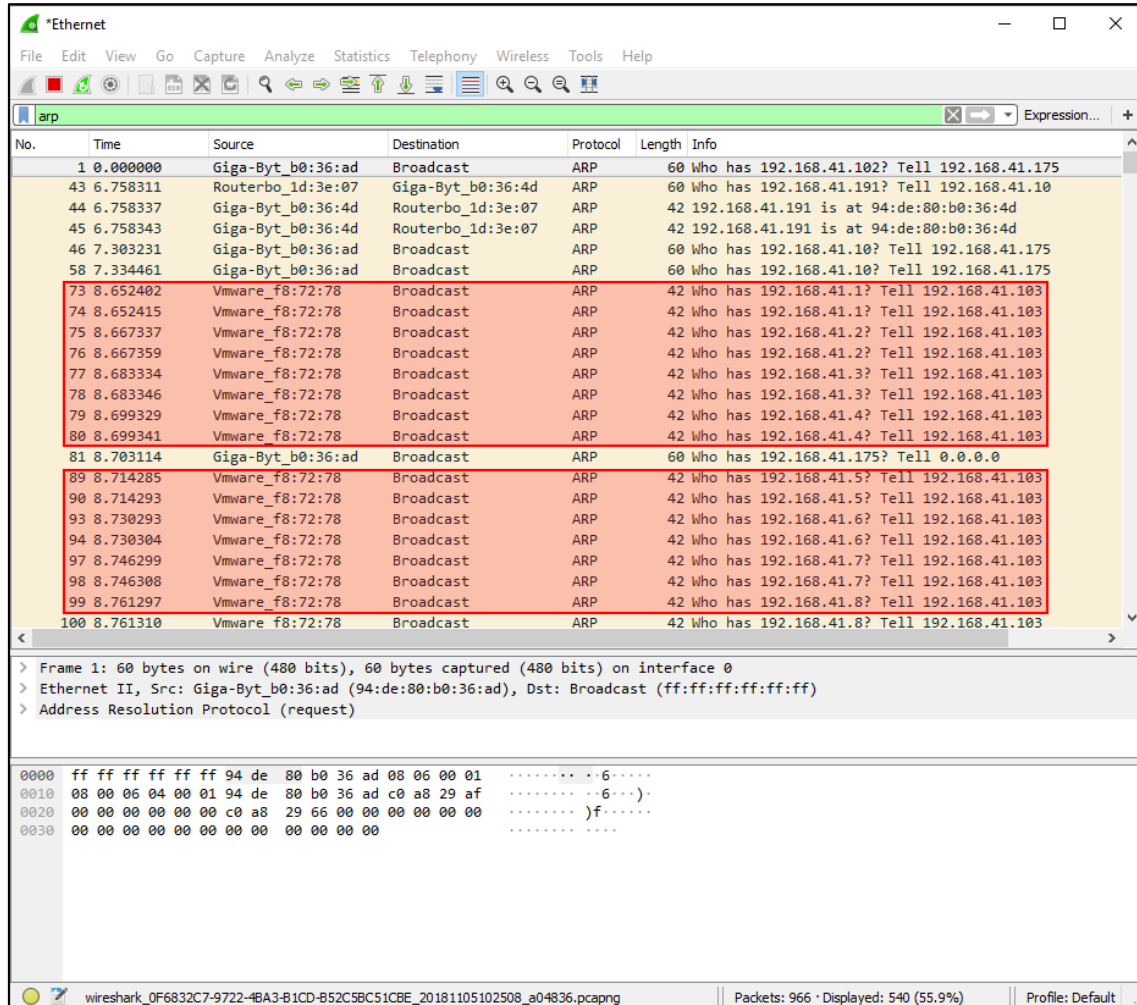


Figura 24. Wireshark, escuchando el tráfico generado por Cain&Abel.

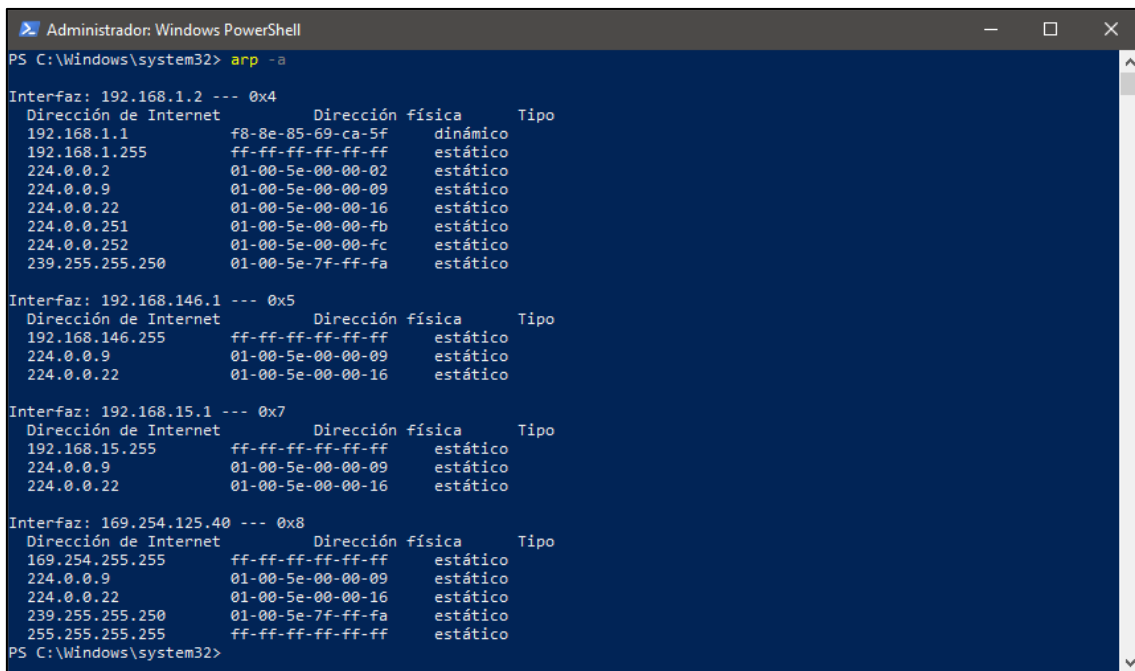
## EJERCICIO 5

### 1. Utilizando la herramienta anterior, se pide:

- Realiza un ataque *Man In The Middle* empleando para ello la técnica *ARP\_Poison*. La idea es que el atacante se ponga en el medio de la víctima y de su puerta de enlace. Documenta todo el proceso anterior de uso de *Cain y Abel*.

Para realizar este ataque lanzamos la MV en modo bridge y nos aseguramos que esta tiene una IP dentro del segmento de red de la víctima para poder realizar el ARP\_POISON, en mi caso voy a envenenar la cache ARP de mi maquina principal (192.168.1.2) para interceptar la comunicación entre esta y la puerta de enlace (192.168.1.1), así podremos ver la comunicación entre ambas consultando la web de configuración de la puerta de enlace ya que esta utiliza HTTP en lugar de HTTPS.

A continuación se muestra el contenido de la cache ARP de la víctima y del atacante, figuras (25) y (26).



```

Administrador: Windows PowerShell
PS C:\Windows\system32> arp -a

Interfaz: 192.168.1.2 --- 0x4
Dirección de Internet    Dirección física    Tipo
192.168.1.1              f8-8e-85-69-ca-5f  dinámico
192.168.1.255            ff-ff-ff-ff-ff-ff  estático
224.0.0.2                01-00-5e-00-00-02  estático
224.0.0.9                01-00-5e-00-00-09  estático
224.0.0.22              01-00-5e-00-00-16  estático
224.0.0.251              01-00-5e-00-00-fb  estático
224.0.0.252              01-00-5e-00-00-fc  estático
239.255.255.250          01-00-5e-7f-ff-fa  estático

Interfaz: 192.168.146.1 --- 0x5
Dirección de Internet    Dirección física    Tipo
192.168.146.255          ff-ff-ff-ff-ff-ff  estático
224.0.0.9                01-00-5e-00-00-09  estático
224.0.0.22              01-00-5e-00-00-16  estático

Interfaz: 192.168.15.1 --- 0x7
Dirección de Internet    Dirección física    Tipo
192.168.15.255           ff-ff-ff-ff-ff-ff  estático
224.0.0.9                01-00-5e-00-00-09  estático
224.0.0.22              01-00-5e-00-00-16  estático

Interfaz: 169.254.125.40 --- 0x8
Dirección de Internet    Dirección física    Tipo
169.254.255.255          ff-ff-ff-ff-ff-ff  estático
224.0.0.9                01-00-5e-00-00-09  estático
224.0.0.22              01-00-5e-00-00-16  estático
239.255.255.250          01-00-5e-7f-ff-fa  estático
255.255.255.255          ff-ff-ff-ff-ff-ff  estático
PS C:\Windows\system32>
  
```

Figura 25. Powershell, cache ARP de la víctima (192.168.1.2).

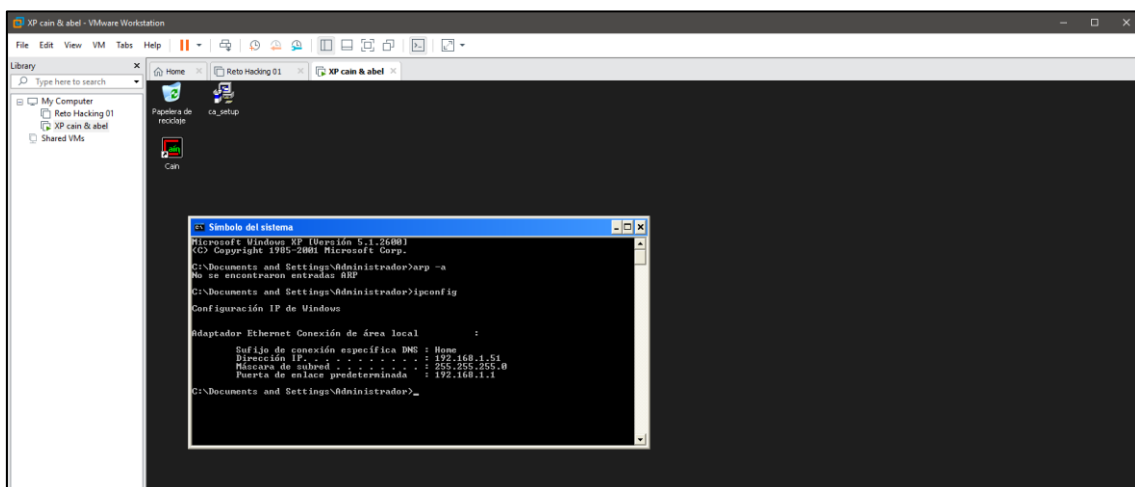


Figura 26. VMware, Cache ARP y configuración IP de la VM que realiza el ataque.

Paso 1:

Realizamos una exploración figura (27) de los hosts activos en la red mediante el envío de solicitudes *ARP*. Click en la imagen para *youtube* con el proceso.

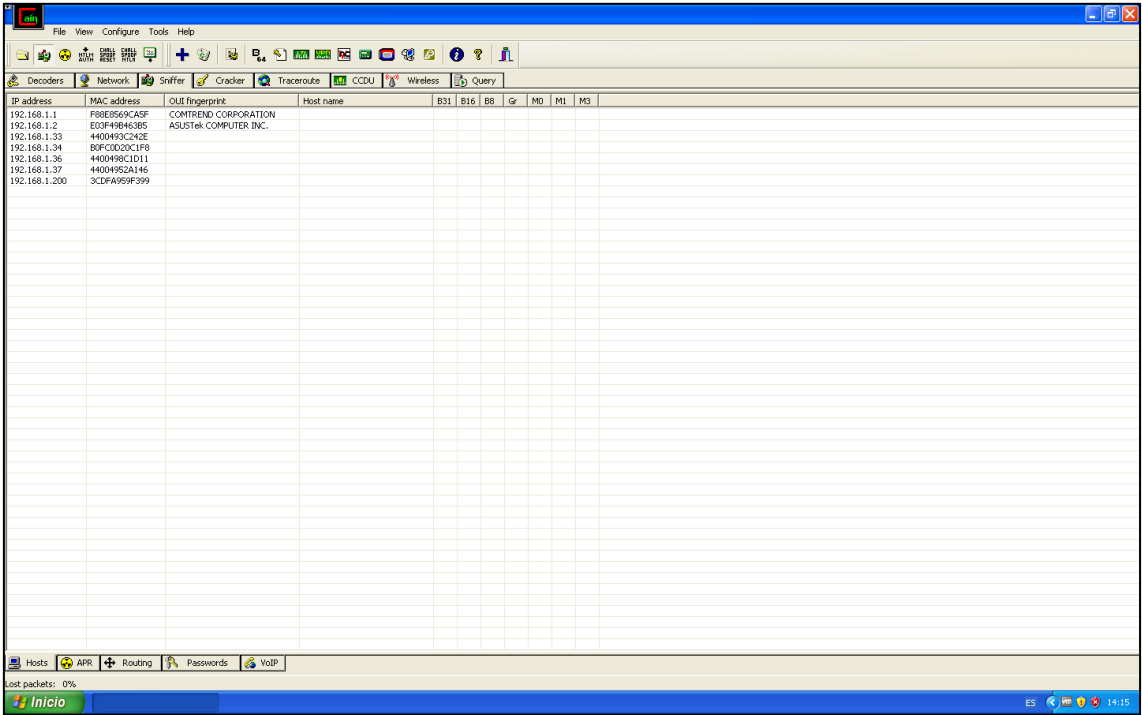


Figura 27. Cain&Abel resultado ARP Scan.

Paso 2:

Mediante la pestaña APR (*ARP Poison Routing*) establecemos la regla para el MITM, cuyo resultado aparece en la siguiente figura (28), click en la imagen para *youtube* con el proceso.

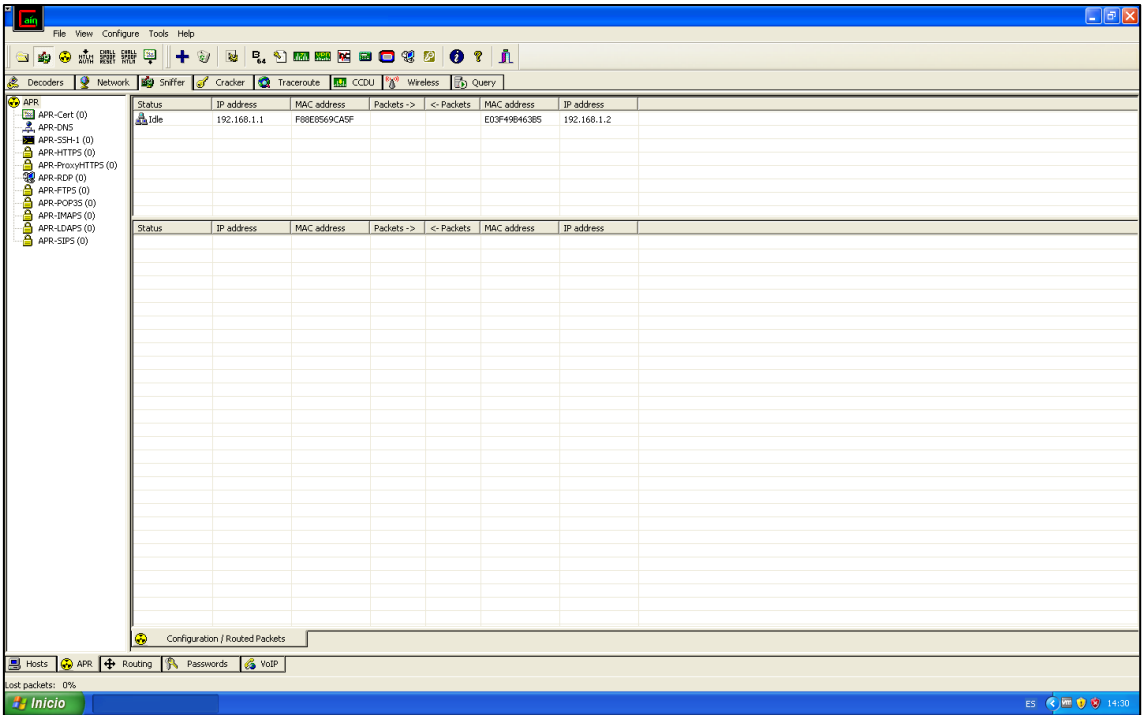
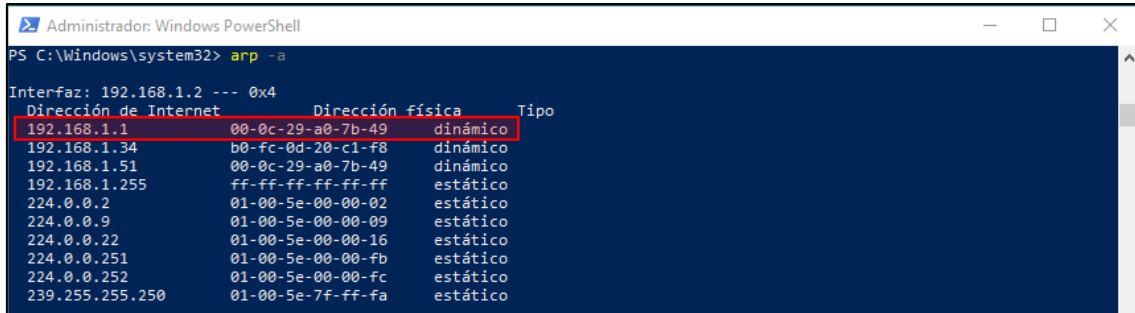


Figura 28. Cain&Abel, resultado de la regla MITM.

c. Una vez que hayas envenenado la caché ARP de la víctima, indica cómo será el contenido de la caché ARP, así como la del host que tiene el rol de puerta de enlace.

Ya que no puedo saber el contenido de la cache ARP de la puerta de enlace (es un *router* real) el cual no permite consultar esta información puedo demostrar también el efecto del ataque según la cache ARP de la víctima la cual relaciona la IP del la puerta de enlace (192.168.1.1) con la MAC del atacante (00:0C:29:A0:7B:49).



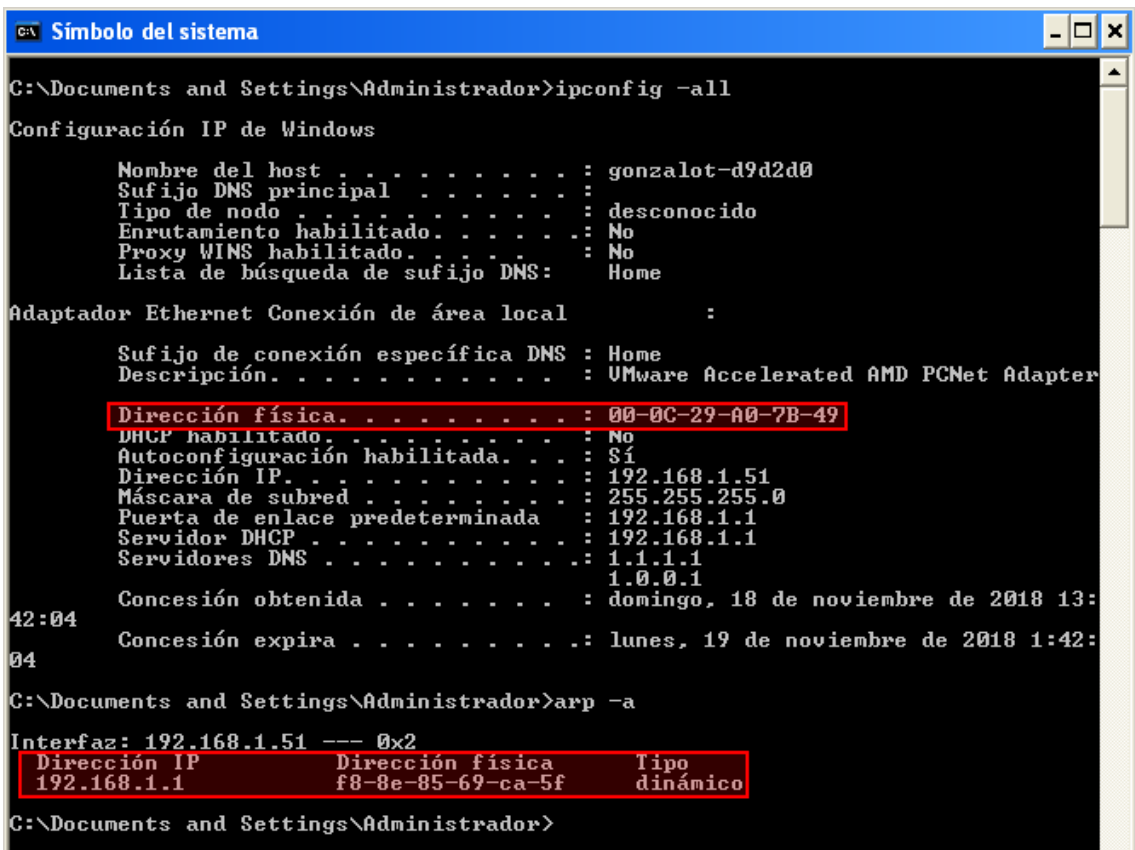
```

Administrador: Windows PowerShell
PS C:\Windows\system32> arp -a

Interfaz: 192.168.1.2 --- 0x4
Dirección de Internet    Dirección física    Tipo
-----
192.168.1.1              00-0c-29-a0-7b-49  dinámico
192.168.1.34             b0-fc-0d-20-c1-f8  dinámico
192.168.1.51             00-0c-29-a0-7b-49  dinámico
192.168.1.255           ff-ff-ff-ff-ff-ff  estático
224.0.0.2                01-00-5e-00-00-02  estático
224.0.0.9                01-00-5e-00-00-09  estático
224.0.0.22               01-00-5e-00-00-16  estático
224.0.0.251              01-00-5e-00-00-fb  estático
224.0.0.252              01-00-5e-00-00-fc  estático
239.255.255.250          01-00-5e-7f-ff-fa  estático
  
```

Figura 29. Cache ARP de la víctima donde 192.168.1.1 tiene asignada la MAC del atacante.

Aquí se muestra la configuración de la VM que está realizando el ataque, así como la MAC real que posee la puerta de enlace, la cual ha sido envenenada en la víctima.



```

C:\Documents and Settings\Administrador>ipconfig -all

Configuración IP de Windows

Nombre del host . . . . . : gonzalot-d9d2d0
Sufijo DNS principal . . . . . :
Tipo de nodo . . . . . : desconocido
Enrutamiento habilitado. . . . . : No
Proxy WINS habilitado. . . . . : No
Lista de búsqueda de sufijo DNS: Home

Adaptador Ethernet Conexión de área local :

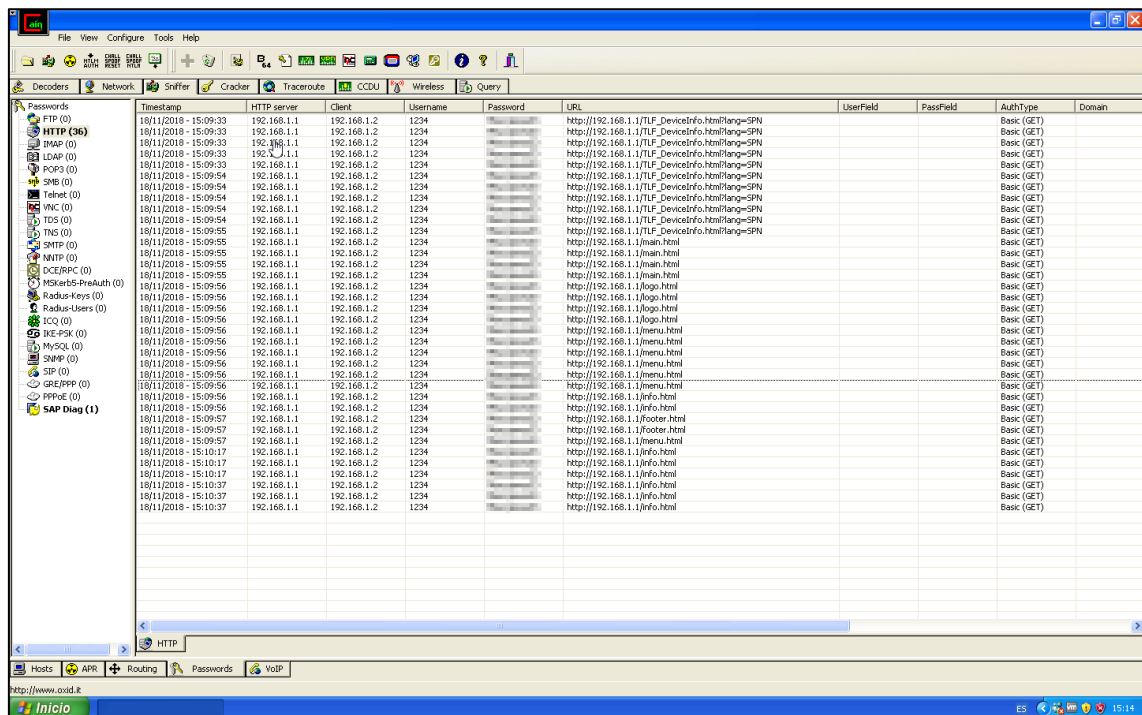
Sufijo de conexión específica DNS : Home
Descripción. . . . . : VMware Accelerated AMD PCNet Adapter
Dirección física. . . . . : 00-0C-29-A0-7B-49
DHCP habilitado. . . . . : No
Autoconfiguración habilitada. . . . . : Sí
Dirección IP. . . . . : 192.168.1.51
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada : 192.168.1.1
Servidor DHCP . . . . . : 192.168.1.1
Servidores DNS . . . . . : 1.1.1.1
                             1.0.0.1
Concesión obtenida . . . . . : domingo, 18 de noviembre de 2018 13:42:04
Concesión expira . . . . . : lunes, 19 de noviembre de 2018 1:42:04

C:\Documents and Settings\Administrador>arp -a

Interfaz: 192.168.1.51 --- 0x2
Dirección IP    Dirección física    Tipo
-----
192.168.1.1     f8-8e-85-69-ca-5f  dinámico
  
```

Figura 30. CMD mostrando la MAC atacante y la MAC real de la puerta de enlace.

Finalmente, y lo que seria el Paso 3 de la parte (a) de este ejercicio, muestro como Cain&Abel interceptan la comunicación e identifican el *login* y el *password* utilizado para acceder al router.



>

Figura 31. Cain&Abel, ataque MITM exitoso mostrando *login* y *password* introducido por la víctima en el formulario de la puerta de enlace.

- d. Busca un formulario en Internet haciendo “*dorking*” que solicite los datos de acceso por HTTP. Desde la víctima introduce información en el formulario buscado y con el analizador de tráfico de red localiza los datos introducidos en el formulario por la víctima. Puedes hacer uso del filtro “*http.request.method==POST*”.

Utilizando la búsqueda “*inurl:/admin/login.asp*” obtengo diversas páginas, escogemos una al azar que sea HTTP, de entre los resultados arrojados he escogido la siguiente:

<http://cms.shopcherrycreek.com/admin/login.asp>

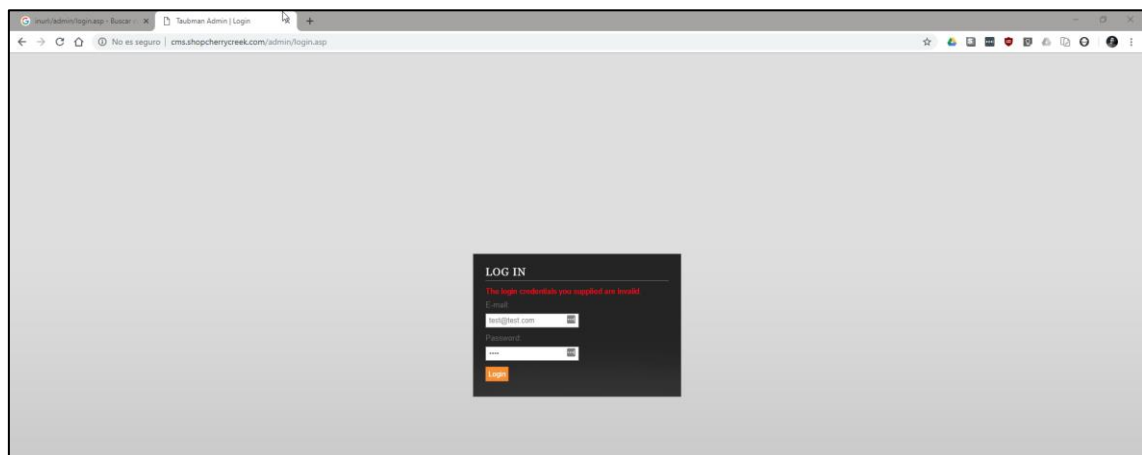


Figura 32. <http://cms.shopcherrycreek.com/admin/login.asp>

Tras introducir unos datos de ejemplo en el formulario, podemos ver como Wireshark muestra estos fácilmente.

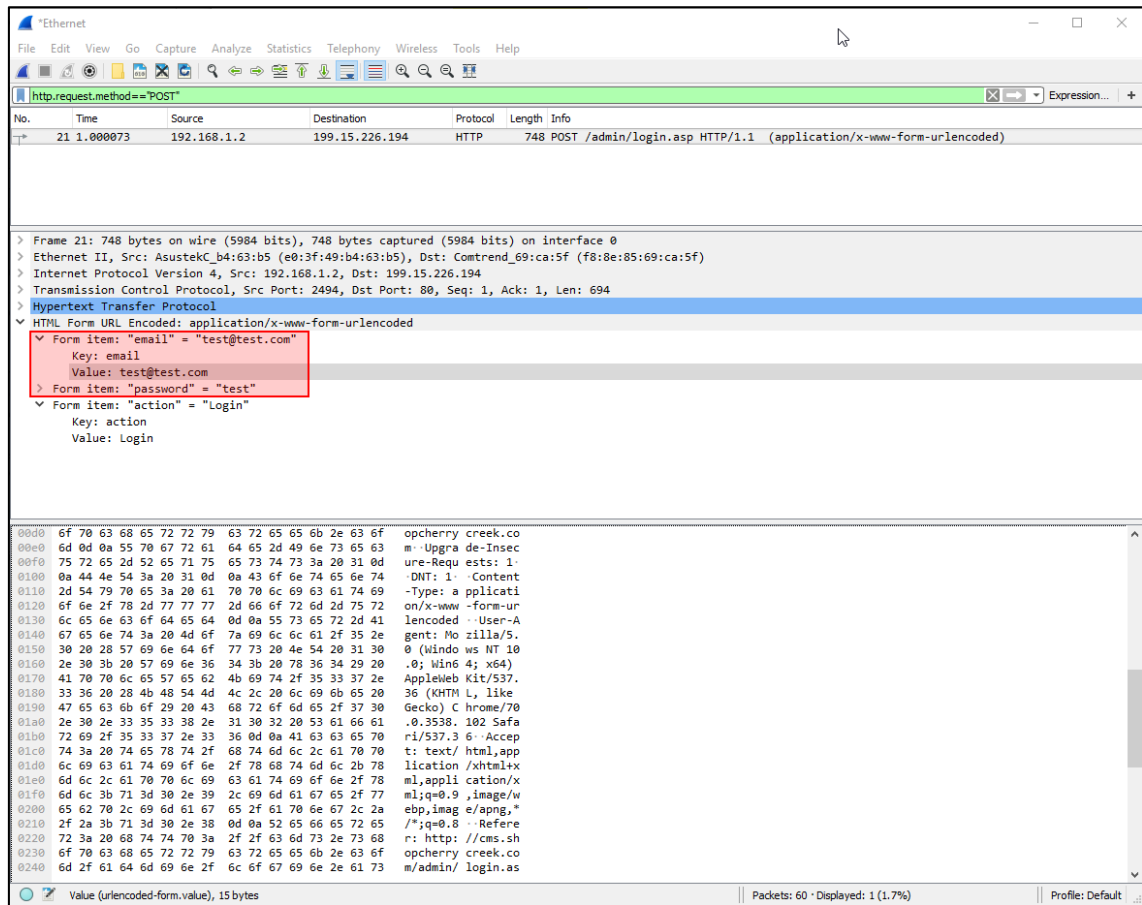


Figura 33. Wireshark filtro HTTP.REQUEST.METHOD=="POST"

## EJERCICIO 6

1. (Opcional). Desde Kali Linux (<https://www.kali.org/>) realiza un ataque *MiTM* con *ARP\_Poison* utilizando la herramienta *Ettercap* (<https://www.ettercap-project.org/>). Documenta y explica todo el proceso que has realizado.

Para realizar este ataque utilizando los mismos objetivos, mi maquina y la puerta de enlace, lanzo una VM con Kali Linux y posteriormente ejecuto *Ettercap*, cuyo funcionamiento para este ejemplo es muy parecido a Cain&Abel.

Una vez lanzado *Ettercap* en el menú *Sniff*, escogemos la opción *Unified Sniffing* que para este ejemplo servirá, la otra opción, *Bridged Sniffing* utiliza 2 interfaces de red para evitar la detección del ataque MITM, la cual queda fuera del ámbito de este ejercicio.

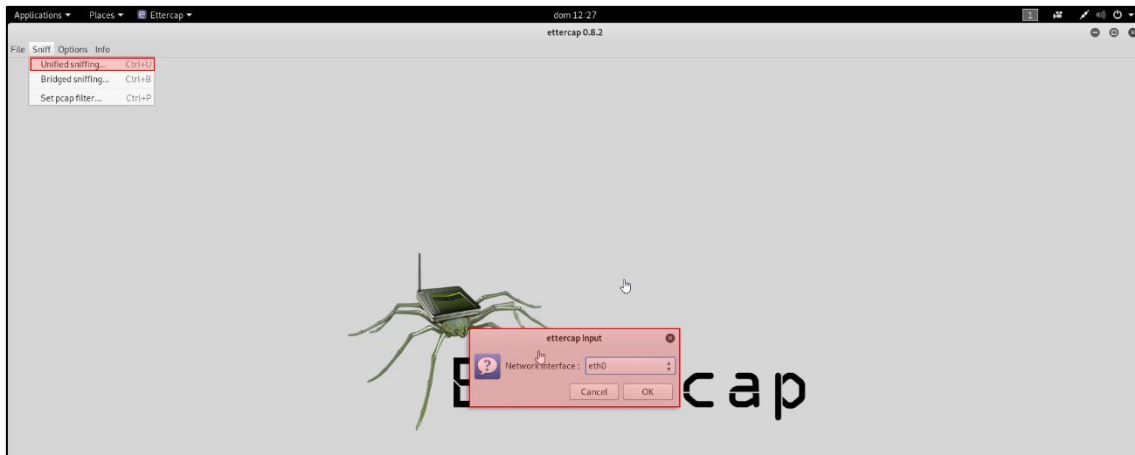


Figura 34. Ettercap, Unified Sniffing y seleccion de interfaz de red.

Tras esto pasamos a buscar los objetivos en mi segmento de red. Utilizando el menú *HOSTS* -> *Scan for HOSTS*, tras esto definimos los objetivos TARGET 1 y 2 con *click* derecho.

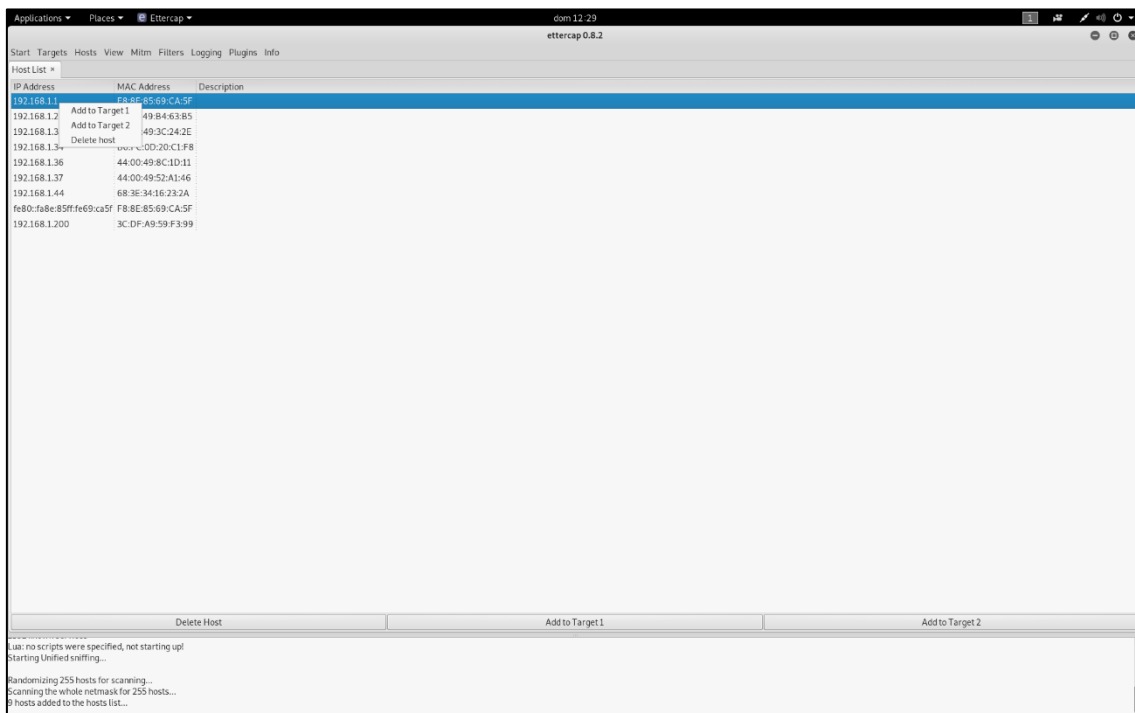


Figura 35. Ettercap, mostrando la lista de HOSTS disponibles en mi segmento.



Ahora en el menú MITM (*man in the middle*) escogemos la variante ARP *Poisoning*, y después en el cuadro de dialogo escogemos *Only poison one Way*. (solo envenenar en una dirección).

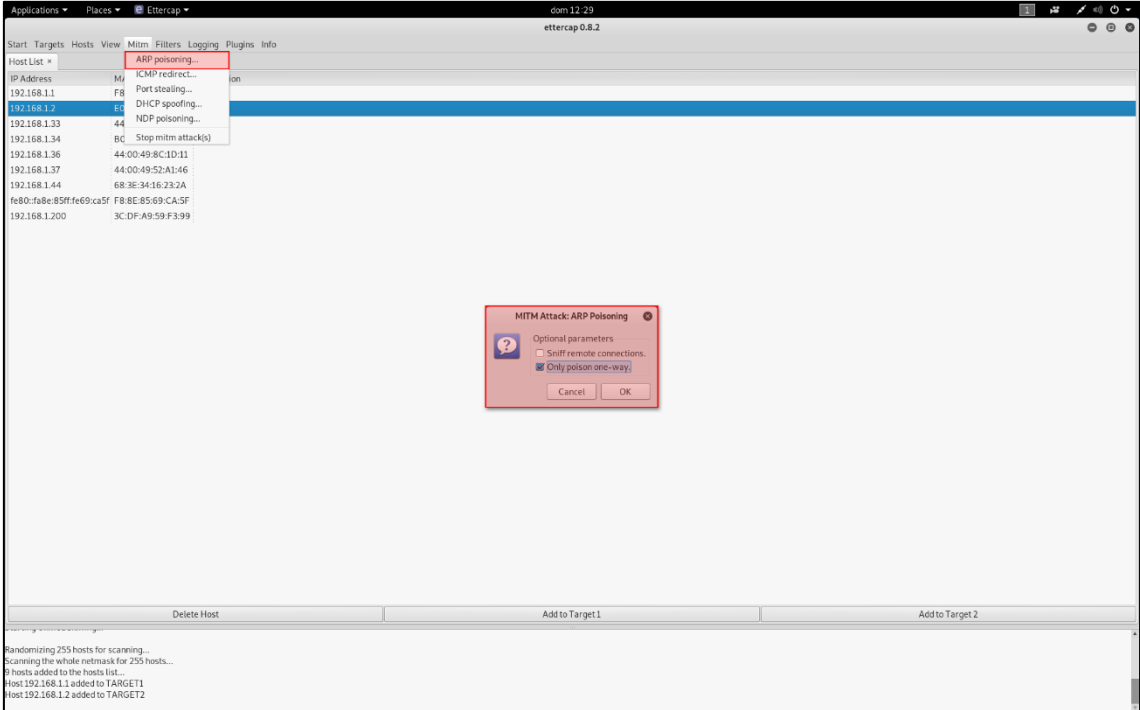


Figura 36. Ettercap, seleccionando la variante MITM y su configuración

Nada mas iniciar el ataque en el menú *Start*, en cuanto se genera tráfico entre ambos objetivos, *Ettercap* reconoce la información relevante y la muestra como se puede ver en la figura siguiente.

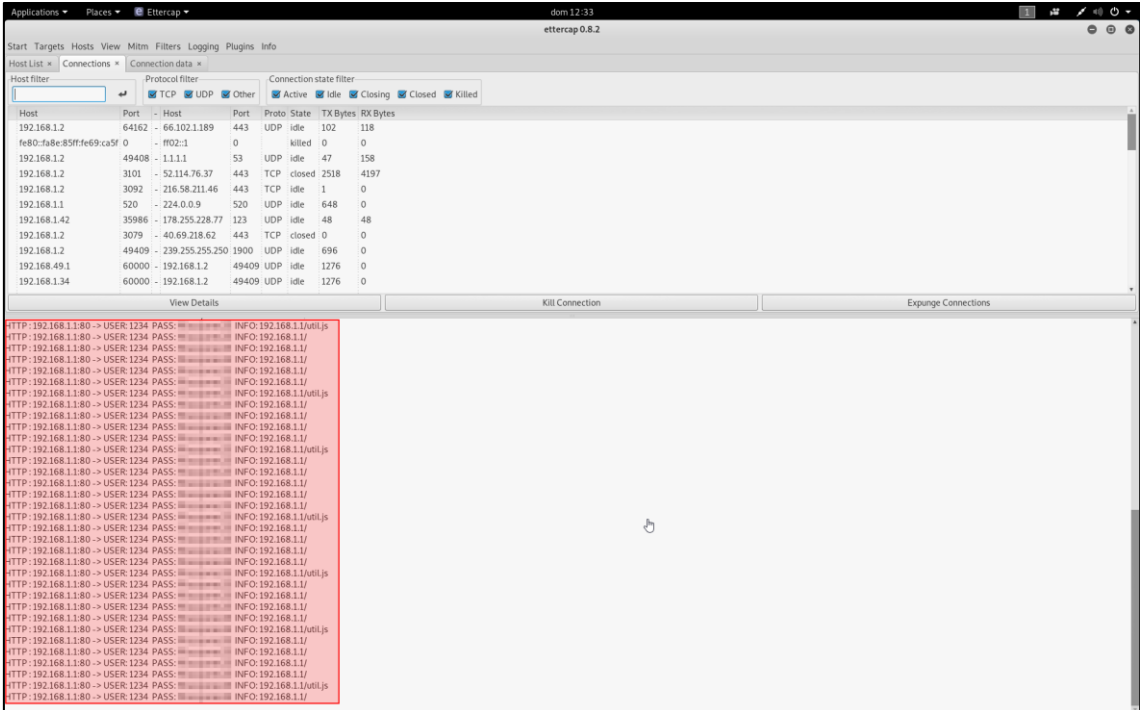


Figura 37. Ettercap, mostrando la comunicación entre los objetivos designados