

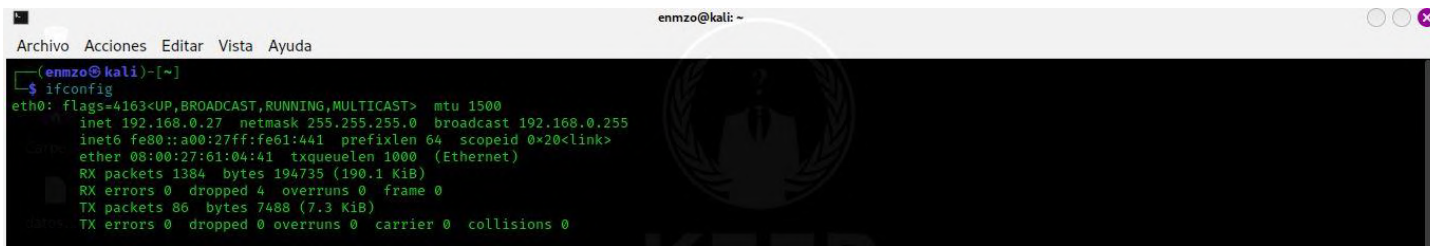
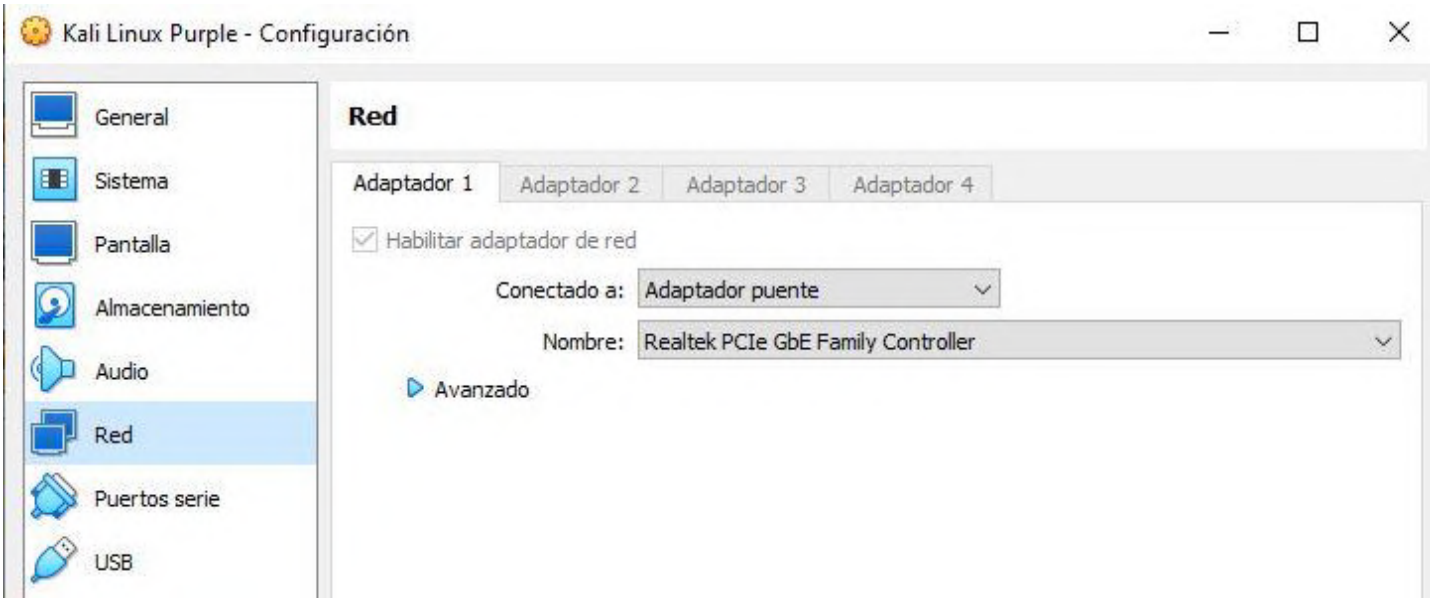
EJERCICIO FINAL - ATAQUES A INFRAESTRUCTURA DE SISTEMAS Y REDES

Definición de alcance y requisitos

- Suponed que tenemos un cliente (vosotros) y queréis conocer el estado de vuestra red.
- Vais a realizar vuestra primera "incursión" en entorno real, sin usar máquinas virtuales preparadas, y generaros un informe para vosotros mismos sobre lo que habéis hecho.
- Para realizar este ejercicio hay que tener en cuenta que los ataques y análisis van a realizarse en la propia red personal de cada uno, por lo que es necesario antes de nada, "pedir permiso" e "informar" al resto de usuarios de la red de los objetivos, horario para poder hacerlo, si el router está accesible para reiniciarlo (se puede quedar tonto en un ARP Spoofing si es de mala calidad), etc...
- DISCLAIMER - ¡¡¡Hacedlo con responsabilidad y cabeza!!! :-)

1. Configuración

- Configurar el tipo de red de Kali Linux como Bridge. De esta manera estará configurada como si fuera un equipo de la propia red. Comprobar que la IP asignada está en el rango de red del resto de equipos (móvil y máquina Host).



Host List x		
IP Address	MAC Address	Description
192.168.0.1	58:76:AC:20:23:20	
192.168.0.10	4C:17:44:8E:2D:6A	
192.168.0.11	F4:17:B8:FF:AA:1E	
192.168.0.13	E0:CC:F8:AC:A7:97	Android.local
192.168.0.14	CE:F9:A7:D4:17:15	
192.168.0.15	F8:5E:A0:51:DE:AC	
192.168.0.16	18:C0:4D:7E:43:CB	
192.168.0.17	F4:17:B8:FF:47:4A	
192.168.0.19	44:01:BB:37:A2:A6	es-e956b2e28460.local
192.168.0.21	F0:2F:74:1E:F9:EB	

2. Selección de objetivo

- Realizar una identificación de equipos de toda la red.
- Identificar equipos por la MAC Address (recordad que podemos sacar basándonos en la MAC el fabricante y por lo tanto acotar que equipos son). Elegir un equipo como objetivo.
 - Nota: Contad con que al menos en la red hay 4 equipos: Equipo 1)
 - Kali Linux en modo Bridge.
 - Equipo 2) Vuestra máquina Host.
 - Equipo 3) Un teléfono móvil en la misma red. Equipo 4)
 - Router.
 - El resto serán otros equipos conectados (aparte de éstos) que haya en la red.

```
C:\> Símbolo del sistema

Microsoft Windows [Versión 10.0.19045.2965]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Chrétien>ipconfig /all

Configuración IP de Windows

Nombre de host. . . . . : chretiensec
Sufijo DNS principal . . . . . :
Tipo de nodo. . . . . : híbrido
Enrutamiento IP habilitado. . . : no
Proxy WINS habilitado . . . . . : no

Adaptador de Ethernet Ethernet:

Sufijo DNS específico para la conexión. . :
Descripción . . . . . : Realtek PCIe GbE Family Controller
Dirección física. . . . . : F0-2F-74-1F-F9-EB
DHCP habilitado . . . . . : sí
Configuración automática habilitada . . . : sí
Vínculo: dirección IPv6 local. . . : fe80::ef27:7d2a:6e58:30a2%10(Preferido)
Dirección IPv4. . . . . : 192.168.0.21(Preferido)
Máscara de subred . . . . . : 255.255.255.0
Concesión obtenida. . . . . : jueves, 11 de mayo de 2023 9:07:41
La concesión expira . . . . . : sábado, 13 de mayo de 2023 8:27:14
Puerta de enlace predeterminada . . . . . : 192.168.0.1
Servidor DHCP . . . . . : 192.168.0.1
IAID DHCPv6 . . . . . : 133181300
DUID de cliente DHCPv6. . . . . : 00-01-00-01-2B-E2-21-85-F0-2F-74-1F-F9
Servidores DNS . . . . . : 192.168.0.1
```

```
Servidores DNS. . . . . : fec0:0:0:ffff::1%1
                        fec0:0:0:ffff::2%1
                        fec0:0:0:ffff::3%1
NetBIOS sobre TCP/IP. . . . . : habilitado
```

C:\Users\Chrétien>arp -a

Interfaz: 192.168.56.1 --- 0x8

Dirección de Internet	Dirección física	Tipo
192.168.56.255	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

Interfaz: 192.168.0.21 --- 0xa

Dirección de Internet	Dirección física	Tipo
192.168.0.1	58-76-ac-20-23-20	dinámico
192.168.0.15	f8-5e-a0-51-de-ac	dinámico
192.168.0.27	08-00-27-61-04-41	dinámico
192.168.0.255	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: 172.31.176.1 --- 0xf

Identificamos la MAC de KALI y del ROUTER.

Ettercap 0.8.3

Host List x

IP Address	MAC Address	Description
192.168.0.1	58:76:AC:20:23:20	
192.168.0.10	4C:17:44:8E:2D:6A	
192.168.0.11	F4:17:B8:FF:AA:1E	
192.168.0.13	E0:CC:F8:AC:A7:97	Android.local
192.168.0.14	CE:F9:A7:D4:17:15	
192.168.0.15	F8:5E:A0:51:DE:AC	
192.168.0.16	18:C0:4D:7E:43:CB	
192.168.0.17	F4:17:B8:FF:47:4A	
192.168.0.19	44:01:BB:37:A2:A6	es-e956b2e28460.local
192.168.0.21	F0:2F:74:1F:F9:FB	

Delete Host Add to Target

Lua: no scripts were specified, not starting up!
Starting Unified sniffing...

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
11 hosts added to the hosts list...

DHCP: [EE:9C:BD:3D:61:9A] REQUEST 192.168.0.26
DHCP: [7E:EB:9B:8F:F2:F2] REQUEST 192.168.0.18
DHCP: [7E:EB:9B:8F:F2:F2] REQUEST 192.168.0.18
DHCP: [7E:EB:9B:8F:F2:F2] REQUEST 192.168.0.18
DHCP: [7E:EB:9B:8F:F2:F2] REQUEST 192.168.0.18
HTTP: 207.182.142.2:80 -> USER: George_Michae! PASS: I'll_never_goNNa_dance_again INFO: anti-phishing
Host 192.168.0.13 added to TARGET1
Host 192.168.0.1 added to TARGET2

Añadimos como target 1 el objetivo ANDROID, y como target 2 nuestro router en Ettercap.

3. Análisis de vulnerabilidades - Exploración

- Realizar una identificación de sistema operativo de un equipo objetivo. (También es importante para validar el punto anterior y ver que equipos son). Realizar una
- identificación de servicios y puertos abiertos del objetivo.
- Realizar una identificación de versiones de servicios del objetivo.

```
(root@kali)-[/home/enmzo]
# nmap -Pn -O 192.168.0.13

Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-12 10:56 CEST
Nmap scan report for 192.168.0.13
Host is up (0.019s latency).
All 1000 scanned ports on 192.168.0.13 are in ignored states.
Not shown: 970 closed tcp ports (reset), 30 filtered tcp ports (no-response)
MAC Address: E0:CC:F8:AC:A7:97 (Xiaomi Communications)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.80 seconds
```

COMPROBAMOS EL ANDROID Y EFECTIVAMENTE, ES UN XIAOMI.

Con este comando realizamos una identificación de servicios y puertos abiertos del objetivo.

```
(root@kali)-[/home/enmzo]
# nmap -Pn -O 192.168.0.13

Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-12 10:56 CEST
Nmap scan report for 192.168.0.13
Host is up (0.019s latency).
All 1000 scanned ports on 192.168.0.13 are in ignored states.
Not shown: 970 closed tcp ports (reset), 30 filtered tcp ports (no-response)
MAC Address: E0:CC:F8:AC:A7:97 (Xiaomi Communications)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.80 seconds
```



Y nos sale que todos los puertos están cerrados, ahora vamos a probar con Nessus...

android

[Back to My Scans](#)

Hosts 0 Vulnerabilities 0 History 1

Search History 1 History

	Start Time	Last Scanned	Status	Scan Details
<input checked="" type="checkbox"/>	Current Today at 10:59 AM	N/A	 Running	Policy: Basic Network Scan Status: Running  Severity Base: CVSS v2.0 Scanner: Local Scanner Start: Today at 10:59 AM

Realizar una identificación de versiones de servicios del objetivo que lo haremos con el comando `-sV` en NMAP.

```
(root@kali)-[/home/enmzo]
# nmap -sV 192.168.0.13

Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-12 11:05 CEST
Nmap scan report for 192.168.0.13
Host is up (0.0090s latency).
All 1000 scanned ports on 192.168.0.13 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: E0:CC:F8:AC:A7:97 (Xiaomi Communications)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.28 seconds
```

4. Análisis de vulnerabilidades - Evaluación

- Realizar un análisis de vulnerabilidades con las herramientas utilizadas sobre el objetivo. Comprobar si hay
- alguna vulnerabilidad crítica (CVSS alto) sobre el objetivo.

El análisis lo hemos realizado en Nessus, por su mejor interfaz gráfica del dispositivo Android.

N

Hosts 1 Vulnerabilities 4 History 1

Filter Search Vulnerabilities 4 Vulnerabilities

Sev	CVSS	VPR	Name	Family	Count
INFO			Ethernet Card Manufacturer ...	Misc.	1
INFO			Ethernet MAC Addresses	General	1
INFO			ICMP Timestamp Request Re...	General	1
INFO			mDNS Detection (Local Netwo...	Service detection	1

Nos detecta 4 vulnerabilidades, pero ninguna que revista una gravedad entrañable. Realmente, son de muy bajo nivel y es buena noticia para el dispositivo Android o Xiaomi.

INFO

mDNS Detection (Local Network)

< >

Description

The remote service understands the Bonjour (also known as ZeroConf or mDNS) protocol, which allows anyone to uncover information from the remote host such as its operating system type and exact version, its hostname, and the list of services it is running.

This plugin attempts to discover mDNS used by hosts residing on the same network segment as Nessus.

Solution

Filter incoming traffic to UDP port 5353, if desired.

Output

```
Nessus was able to extract the following information :  
- mDNS hostname      : Android.local.
```

To see debug logs, please visit individual host

Plugin Details

Severity:	Info
ID:	66717
Version:	\$Revision: 1.1 \$
Type:	remote
Family:	Service detection
Published:	May 31, 2013
Modified:	May 31, 2013

Risk Information

Risk Factor: None

Vulnerabilities 5

INFO Ethernet MAC Addresses

Description

This plugin gathers MAC addresses discovered from both remote probing of the host (e.g. SNMP and Netbios) and from running local checks (e.g. ifconfig). It then consolidates the MAC addresses into a single, unique, and uniform list.

Output

The following is a consolidated list of detected MAC addresses:

- E0:CC:F8:AC:A7:97

To see debug logs, please visit individual host

Port ▲

Hosts

N/A

192.168.0.13

Plugin Details

Severity: Info

ID: 86420

Version: 1.6

Type: combined

Family: General

Published: October 16, 2015

Modified: May 13, 2020

Risk Information

Risk Factor: None

Aquí arriba he puesto dos vulnerabilidades detectadas, y como podemos comprobar ninguna reviste gravedad alguna.

También adjunto en classroom y no lo meto aquí por no saturar el documento, un informe en PDF de las vulnerabilidades de dicho dispositivo sacado por Nessus como nos enseñaron en clase.

5. Ataque MITM y captura/sniffing de tráfico

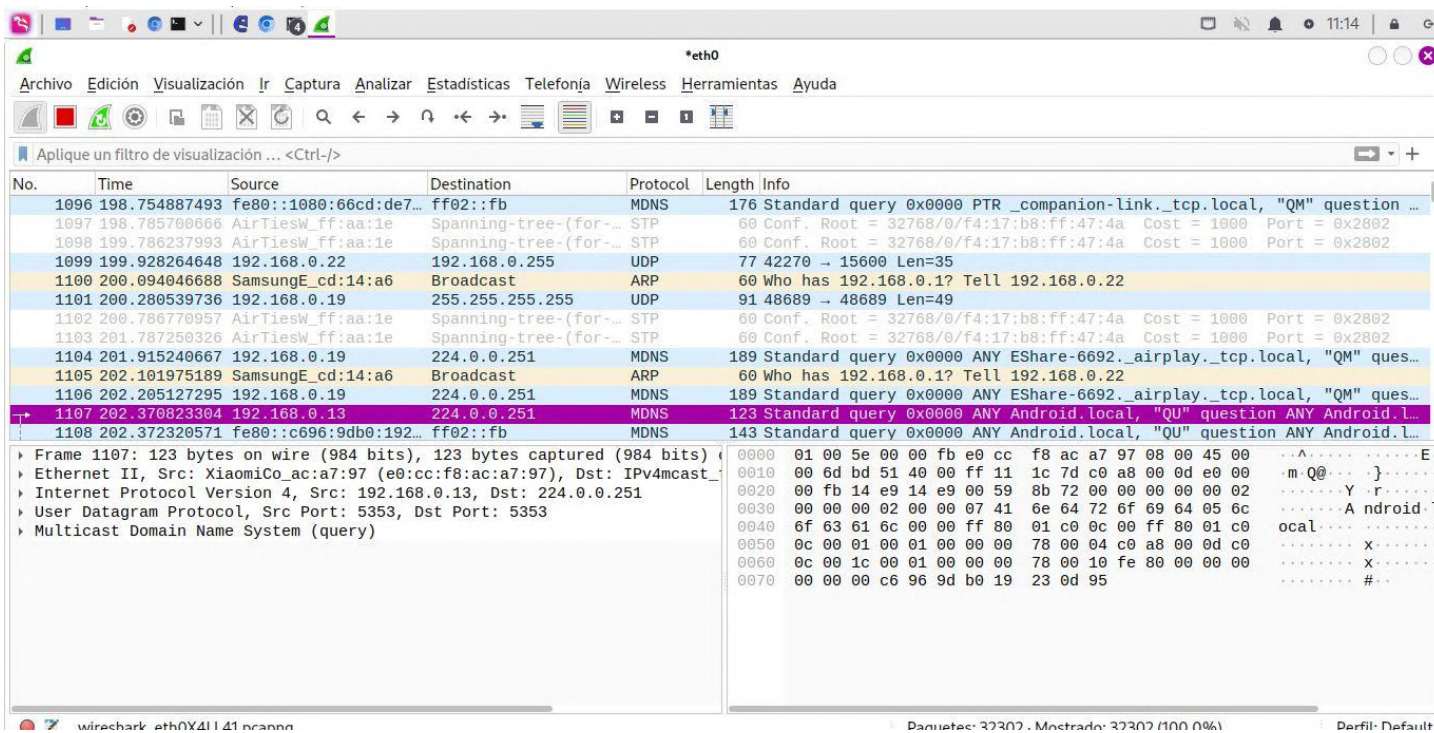
- Realizar un ataque MITM entre un equipo de la red y el router para capturar tráfico entre ellos, e intentar averiguar a qué servicios, IPs y webs se está accediendo.
- En caso que se utilice algún protocolo inseguro, es posible analizar la información más en detalle utilizando varias reglas en Wireshark, de forma "que sea tráfico ftp o tráfico telnet o tráfico http" para poder ayudarlos en el análisis.
- Ejemplo: Usuarios y contraseñas que se transmitan en "texto plano", hay en http, ftp y telnet entre otros, como hemos visto en clase.

El ataque que hemos realizado es al dispositivo Android o Xiaomi con ip: 192.168.0.13, lo primero que haremos es abrir Ettercap con el comando en la terminal de Linux : ettercap -G

Una vez abierto localizamos los dispositivos que queremos...

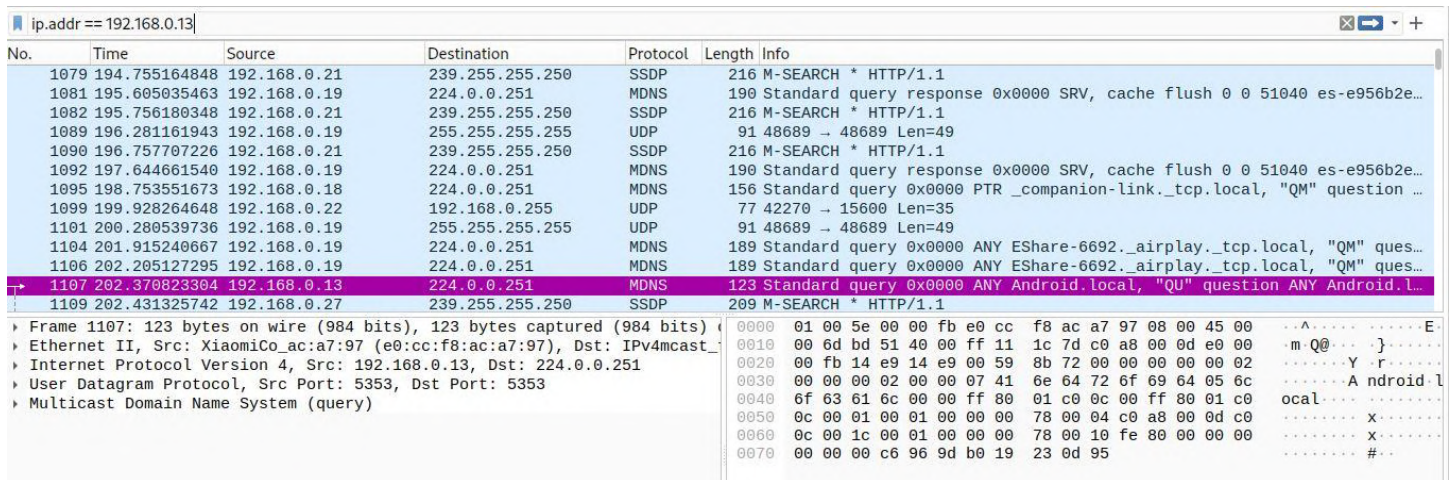
IP Address	MAC Address	Description
192.168.0.1	58:76:AC:20:23:20	
192.168.0.13	E0:CC:F8:AC:A7:97	Android.local

Y procedemos a abrir Wireshark para empezar a scanear el tráfico....



Ahora nos encontramos en un mar de datos, que se seguirán añadiendo y que no tienen mucho sentido así a simple vista, porque es buscar agua en el desierto.

Para ello utilizaremos un filtro de wireshark...



Al utilizar este filtro, filtraremos por la ip del dispositivo que nos interesa.

Por ejemplo podemos ver las búsquedas del dispositivo en la web...

844	158.018048329	192.168.0.18	224.0.0.251	MDNS	156 Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question ...
846	158.078815774	192.168.0.21	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1
848	158.113973668	192.168.0.18	224.0.0.251	MDNS	401 Standard query response 0x0000 TXT, cache flush PTR _rdlink._tcp.local...
850	158.471956790	192.168.0.20	224.0.0.251	MDNS	112 Standard query 0x0000 PTR _sleep-proxy._udp.local, "QU" question OPT
854	159.080243181	192.168.0.21	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1
855	159.474777614	192.168.0.20	224.0.0.251	MDNS	112 Standard query 0x0000 PTR _sleep-proxy._udp.local, "QM" question OPT
857	159.765532838	192.168.0.19	224.0.0.251	MDNS	805 Standard query response 0x0000 TXT, cache flush PTR _airplay._tcp.local...

[Group: Sequence]					0000	01 00 5e 7f ff fa f0 2f 74 1f f9 eb 08 00 45 00	..A.../ t....E
Request Method: M-SEARCH					0010	00 cb 36 67 00 00 01 11 d2 03 c0 a8 00 15 ef ff	..6g....
Request URI: *					0020	ff fa ec 8b 07 6c 00 b7 d9 30 4d 2d 53 45 41 52OM-SEAR
Request Version: HTTP/1.1					0030	43 48 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48	CH * HTTP/1.1..H
HOST: 239.255.255.250:1900\r\n					0040	4f 53 54 3a 20 32 33 39 2e 32 35 35 2e 32 35 35	OST: 239.255.255
MAN: "ssdp:discover"\r\n					0050	2e 32 35 30 3a 31 39 30 30 0d 0a 4d 41 4e 3a 20	.250:190 0..MAN:
MX: 1\r\n					0060	22 73 73 64 70 3a 64 69 73 63 6f 76 65 72 22 0d	"ssdp:discover".
ST: urn:dial-multiscreen-org:service:dial:1\r\n					0070	0a 4d 58 3a 20 31 0d 0a 53 54 3a 20 75 72 6e 3a	MX: 1..ST: urn:
USER-AGENT: Microsoft Edge/113.0.1774.35 Windows\r\n					0080	64 69 61 6c 2d 6d 75 6c 74 69 73 63 72 65 65 6e	dial-mul tiscreen
\r\n					0090	2d 6f 72 67 3a 73 65 72 76 69 63 65 3a 64 69 61	-org:ser vice:dia
[Full request URI: http://239.255.255.250:1900*]					00a0	6c 3a 31 0d 0a 55 53 45 52 2d 41 47 45 4e 54 3a	l:1..USE R-AGENT:
[HTTP request 1/4]					00b0	20 4d 69 63 72 6f 73 6f 66 74 20 45 64 67 65 2f	Microso ft Edge/
[Next request in frame: 854]					00c0	31 31 33 2e 30 2e 31 37 37 34 2e 33 35 20 57 69	113.0.17 74.35 Wi
					00d0	6e 64 6f 77 73 0d 0a 0d 0a	ndows...

Como podemos ver, M-Search y HTTP, son búsquedas que ha realizado el dispositivo.

Aquí abajo también podemos ver por ejemplo un payload de la red del dispositivo...

87	16.574835618	192.168.0.27	142.250.184.10	QUIC	72 Pr
88	16.575003122	192.168.0.27	142.250.184.10	QUIC	78 Pr
89	16.575026813	142.250.184.10	192.168.0.27	QUIC	231 Pr
90	16.583480476	142.250.184.10	192.168.0.27	QUIC	65 Pr
91	16.605227009	192.168.0.27	142.250.184.10	QUIC	74 Pr
94	17.252997202	192.168.0.19	224.0.0.251	MDNS	483 St
96	18.255248963	192.168.0.19	224.0.0.251	MDNS	142 St
97	18.545330430	192.168.0.19	224.0.0.251	MDNS	142 St
99	18.834448952	192.168.0.19	224.0.0.251	MDNS	142 St
100	19.117668332	192.168.0.19	224.0.0.251	MDNS	190 St

[Stream index: 7]					0000
▸ [Timestamps]					0010
UDP payload (36 bytes)					0020
▾ QUIC IETF					0030
▾ QUIC Connection information					0040
[Connection Number: 0]					
[Packet Length: 36]					
▾ QUIC Short Header DCID=ede7eb6ff97ad0d2					
0... .. = Header Form: Short Header (0)					
.1... .. = Fixed Bit: True					
..0... .. = Spin Bit: False					
Destination Connection ID: ede7eb6ff97ad0d2					
Remaining Payload: 790725470befb4e4fa971b1b17e3c589457e48f719cd891266					

- Un payload es en seguridad informática referida a amenazas de tipo exploit, payload es la parte del código del malware que realiza la acción maliciosa en el sistema.

También podemos ver los diversos ping que tiene o ha realizado el dispositivo o ha recibido...

28113	4036.6163323...	192.168.0.27	192.168.0.13	ICMP	60 Echo (ping) request	id=0x0001, seq=1/256, ttl=64 (reply in 28114)
28114	4036.6215335...	192.168.0.13	192.168.0.27	ICMP	60 Echo (ping) reply	id=0x0001, seq=1/256, ttl=64 (request in 28113)
28115	4036.7246164...	192.168.0.27	192.168.0.13	ICMP	296 Echo (ping) request	id=0x0001, seq=1/256, ttl=64 (reply in 28116)
28116	4036.7380491...	192.168.0.13	192.168.0.27	ICMP	296 Echo (ping) reply	id=0x0001, seq=1/256, ttl=64 (request in 28115)
28117	4036.8203124...	192.168.0.27	192.168.0.13	ICMP	296 Echo (ping) request	id=0x0001, seq=1/256, ttl=64 (reply in 28118)
28118	4036.8474866...	192.168.0.13	192.168.0.27	ICMP	296 Echo (ping) reply	id=0x0001, seq=1/256, ttl=64 (request in 28117)
28120	4036.9286764...	192.168.0.27	192.168.0.13	ICMP	296 Echo (ping) request	id=0x0001, seq=1/256, ttl=64 (reply in 28121)
28121	4036.9517638...	192.168.0.13	192.168.0.27	ICMP	296 Echo (ping) reply	id=0x0001, seq=1/256, ttl=64 (request in 28120)

Incluso y si no me equivoco hemos capturado una llamada de teléfono que imagino que será por wifi o whatsapp...

28136	4040.1637747...	192.168.0.27	192.168.0.13	IAX2	54 IAX, source call# 8,
28137	4040.1830453...	192.168.0.13	192.168.0.27	ICMP	90 Destination unreacha
28138	4040.2405831...	192.168.0.27	192.168.0.13	ICMP	60 Domain Name Request
28141	4041.3559705...	192.168.0.27	192.168.0.13	ICMP	60 Address mask request
28142	4041.5597353...	192.168.0.27	192.168.0.13	UDP	118 60803 → 4827 Len=76

[Protocols in frame: eth:ethertype:ip:udp:iax2]
[Coloring Rule Name: UDP]
[Coloring Rule String: udp]

▼ Ethernet II, Src: PcsCompu_61:04:41 (08:00:27:61:04:41), Dst: XiaomiCo_...
 ► Destination: XiaomiCo_ac:a7:97 (e0:cc:f8:ac:a7:97)
 ► Source: PcsCompu_61:04:41 (08:00:27:61:04:41)
 Type: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 192.168.0.27, Dst: 192.168.0.13
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 ► Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 40
 Identification: 0xb24a (45642)
 ► 010 = Flags: 0x2 Don't fragment

0000 e0 cc f8 ac a7 97
0010 00 28 b2 4a 40 00
0020 00 0d b7 c9 11 d9
0030 00 00 00 00 06 1e

Que como podemos ver corresponde a XiaomiCo.

También una llamada al DNS desde el dispositivo...

28146	4042.4153037...	192.168.0.13	192.168.0.27		
28147	4042.4646133...	192.168.0.27	192.168.0.13		
28148	4042.4702423...	192.168.0.27	192.168.0.13		
28149	4042.5760661...	192.168.0.13	192.168.0.27		
28150	4042.6248061...	192.168.0.27	192.168.0.13		

▼ 000. = Flags: 0x0
 ...0 0000 0000 0000 = Fragment Offset: 0
 Time to Live: 255
 Protocol: ICMP (1)
 Header Checksum: 0xa862 [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 192.168.0.27
 Destination Address: 192.168.0.13
▼ Internet Control Message Protocol
 Type: 37 (Domain Name Request)
 Code: 0
 Checksum: 0xdafd [correct]
 [Checksum Status: Good]

Con este filtro de abajo podemos filtrar si hemos capturado algún login o similar en el dispositivo, en el momento de nuestro ataque pero no hay.

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda					
ip.addr == 192.168.0.13 and http.request "POST"					
No.	Time	Source	Destination	Protocol	Length Info
28133	4039.5541670...	192.168.0.27	192.168.0.13	UDP	62 34509 → 65001 Len=20

Comunicaciones con el protocolo TCP por ejemplo, también podemos ver abajo...

28191	4053.0103390...	192.168.0.27	192.168.0.13	TCP
28192	4053.1193705...	192.168.0.13	192.168.0.27	TCP
28193	4053.1725034...	192.168.0.27	192.168.0.13	UDP
28194	4053.4740557...	192.168.0.27	192.168.0.13	L2TP
28195	4053.5276833...	192.168.0.13	192.168.0.27	ICMP
28196	4053.5287611...	192.168.0.27	192.168.0.13	TCP
28197	4053.5328590...	192.168.0.13	192.168.0.27	TCP
28198	4053.5333464...	192.168.0.27	192.168.0.13	UDP
28205	4054.8922330...	192.168.0.27	192.168.0.13	ISAKMP
28206	4055.0262049...	192.168.0.13	192.168.0.27	ICMP
28208	4055.2701105...	192.168.0.27	192.168.0.13	UDP
28209	4055.5718269...	192.168.0.27	192.168.0.13	UDP

Source Address: 192.168.0.27

Destination Address: 192.168.0.13

Transmission Control Protocol, Src Port: 3133, Dst Port: 64796, Seq: 1, Source Port: 3133
Destination Port: 64796
[Stream index: 7675]
[Conversation completeness: Incomplete (40)]
[TCP Segment Len: 1]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 538
[Next Sequence Number: 2 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 65383
0101 = Header Length: 20 bytes (5)

- El protocolo de control de transmisión (TCP) es, al igual que el protocolo UDP como el SCTP, un protocolo de Internet que está ubicado en la capa de transporte del modelo OSI. El objetivo del protocolo TCP es crear conexiones dentro de una red de datos compuesta por redes de computadoras para intercambiar datos.

Aquí abajo también podemos ver un intento de acceso a alguna web o servicio, pero está encriptado...

28291	4067.7575742...	192.168.0.27	192.168.0.13	UDP	67 39702 → 3708 L
28295	4068.3140589...	192.168.0.27	192.168.0.13	RADIUS	100 Access-Request
28296	4068.3717038...	192.168.0.13	192.168.0.27	ICMP	128 Destination un
28297	4068.3731919...	192.168.0.27	192.168.0.13	RIPv2	66 Request
28298	4068.3733102...	192.168.0.27	192.168.0.13	RIPv1	66 Request
28305	4069.5304239...	192.168.0.27	192.168.0.13	UDP	791 36294 → 3702 L
28308	4069.7501184...	192.168.0.13	192.168.0.27	ICMP	590 Destination un

[Stream index: 759]

▸ [Timestamps]

UDP payload (58 bytes)

▾ RADIUS Protocol

Code: Access-Request (1)

Packet identifier: 0x6c (108)

Length: 58

Authenticator: e0b8a0506bf6ad64f3cba6191025ca57

▾ Attribute Value Pairs

▸ AVP: t=User-Name(1) l=8 val=nessus

▸ AVP: t=User-Password(2) l=18 val=Encrypted

▸ AVP: t=NAS-IP-Address(4) l=6 val=192.168.0.13

▸ AVP: t=NAS-Port(5) l=6 val=1025

```
0000 e0 cc f8 ac
0010 00 56 72 26
0020 00 0d c9 fd
0030 a0 50 6b f6
0040 6e 65 73 73
0050 65 a2 e8 31
0060 00 00 04 01
```

- Según he estado mirando e informándome el protocolo define algunos AVP que deben estar presentes en los mensajes de petición para accounting en la sesión de un usuario.