

# Wild Wild WiFi: Dancing with wolves

*Chema Alonso ([chema@l1paths.com](mailto:chema@l1paths.com))*

*Pablo González ([pablo@l1paths.com](mailto:pablo@l1paths.com))*

*Ioseba Palop ([ioseba.palop@l1paths.com](mailto:ioseba.palop@l1paths.com))*

*Francisco Ramírez ([franciscojose.ramirezvicente@telefonica.com](mailto:franciscojose.ramirezvicente@telefonica.com))*

*Innovación y Laboratorio ([labs@l1paths.com](mailto:labs@l1paths.com))*

## Executive Summary

La seguridad en las redes *Wireless* ha estado siempre en observación en la comunidad de investigadores de seguridad informática. El presente trabajo proporciona una serie de mejoras e ideas prácticas que pueden ayudar a fortificar la seguridad de las redes *Wireless*. La implementación de cambios de contraseñas dinámicos y temporales, sumado a la posibilidad de hacer “*pinning*” con varios elementos característicos de las redes *Wireless* aumentan la seguridad de las redes.

## 1.- Antecedentes

La seguridad *Wireless* ha sido siempre un reto para los administradores de las organizaciones. Desde el nacimiento de éstas han tenido graves problemas de seguridad como la carencia de algoritmos de cifrado robustos, fallos en la implementación de protocolos seguros o la debilidad en la distribución de claves, entre otros ejemplos. El paper “*Living in the Jungle: Legitimate users in legitimate insecure Wireless networks*” recoge algunos de estos fallos de seguridad y propone una métrica de seguridad en redes inalámbricas. Esta métrica se encuentra en la herramienta *Mummy*.

El paper “*Living in the Jungle: Legitimate users in legitimate insecure Wireless networks*” proporciona información sobre cómo la compatibilidad hacia atrás, los costes que suponen las migraciones o la falta de conocimientos técnicos pueden proporcionar inseguridad en la red *Wireless*.

LIVING IN THE JUNGLE: LEGITIMATE USERS IN LEGITIMATE, INSECURE WIRELESS NETWORKS

1

Alejandro Martín, Rodolfo Bordón Villar, José María Alonso, Antonio Guzmán

### Living in the jungle: Legitimate users in legitimate, insecure wireless networks

**Abstract**— Security in wireless networks has been much debated in recent years. Although the general understanding of the technologies that provide secure networks has reached very high levels, the fact remains that the security of some networks currently in use is below standard. It is not at all unusual for a legitimate user to have to access a legitimate, insecure network. These connections multiply the risks involved in data transmission for legitimate users, since the security provided by the infrastructure is insufficient. This article describes the risks and protection options that a legitimate user of a legitimate, although insecure wireless network, can resort to. This document analyses the environments in which a legitimate user may be at risk, exposed to attacks from malicious network users, and the practices that help to increase security for your work within the network. A monitoring tool has been developed to provide assistance in this task, by allowing the user to monitor network activity, and thereby gaining greater security.

**Terms used**— WEP, WPA, WPA2, Computer security, Wireless network risks, TKIP, AES, Wireless network protection.

for this article and to serve as proof of concept. This instrument uses Centrino laptop computers with Intel Pro/Wireless 2200BG chipsets. These devices were chosen for their capacity to offer a clear presentation of the work carried out, although other technologies can be used for the same purpose.

#### II. WIRELESS NETWORK SECURITY MODELS

There are three security models currently co-existing in the real world: IEEE 802.11 (WEP [Wireless Equivalent Privacy]), WPA (Wireless Protected Access) and IEEE 802.11i (WPA2 [Wireless Protected Access 2]).

##### A. The IEEE 802.11 (WEP[2]) standard

Figura 1: “*Living in the Jungle: Legitimate users in legitimate, insecure Wireless networks*”

El anterior *paper* fue publicado en el año 2010 y durante estos años han surgido nuevas amenazas y vulnerabilidades. Recientemente, se ha publicado el ataque KRACK (*Key Reinstallation Attacks*), el cual llevaba a cabo la explotación de una vulnerabilidad en WPA2. La explotación de esta vulnerabilidad permitiría a un atacante, dentro del rango de la red *WiFi*, tener acceso a la información de la red. Este ataque ha supuesto un duro golpe al protocolo WPA2, el cual no tenía graves vulnerabilidades en su configuración PSK.

El investigador *Mathy Vanhoef*, de la Universidad de *KY Leuven*, mostró y detalló como el ataque podía llevarse a cabo contra cualquier plataforma: *Android*, *Linux*, *Windows*, *OpenBSD*, *Linksys*, etcétera.

El ataque se centraba en el *four-way handshake* del protocolo WPA2. Este procedimiento se utiliza para comprobar las credenciales, cuando el usuario intenta unirse a la red. Durante el proceso, se generan nuevas claves de cifrado, las cuales se añaden en este proceso para proteger la sesión del usuario que intenta la conexión. La

vulnerabilidad que explota KRACK permite al atacante manipular o repetir de nuevo el tercer mensaje del *four-way handshake*, permitiendo reinstalar la clave criptográfica que ya se ha utilizado. La clave criptográfica solo debía ser utilizada en una ocasión. Se puede obtener un mayor nivel de detalle en el *paper* publicado por *Mathy Vanhoef* denominado “

## Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2

Mathy Vanhoef  
imec-DistriNet, KU Leuven  
Mathy.Vanhoef@cs.kuleuven.be

Frank Piessens  
imec-DistriNet, KU Leuven  
Frank.Piessens@cs.kuleuven.be

### ABSTRACT

We introduce the key reinstallation attack. This attack abuses design or implementation flaws in cryptographic protocols to reinstall an already-in-use key. This resets the key's associated parameters such as transmit nonces and receive replay counters. Several types of cryptographic Wi-Fi handshakes are affected by the attack.

All protected Wi-Fi networks use the 4-way handshake to generate a fresh session key. So far, this 14-year-old handshake has remained free from attacks, and is even proven secure. However, we show that the 4-way handshake is vulnerable to a key reinstallation attack. Here, the adversary tricks a victim into reinstalling an already-in-use key. This is achieved by manipulating and replaying handshake messages. When reinstalling the key, associated parameters such as the incremental transmit packet number (nonce) and receive packet number (replay counter) are reset to their initial value. Our key reinstallation attack also breaks the PeerKey, group key, and Fast BSS Transition (FT) handshake. The impact depends on the handshake being attacked, and the data-confidentiality protocol in use. Simplified, against AES-CCMP an adversary can replay and decrypt (but not forge) packets. This makes it possible to hijack TCP streams and inject malicious data into them. Against WPA-TKIP and GCMP the impact is catastrophic: packets can be replayed,

work, we present design flaws in the 4-way handshake, and in related handshakes. Because we target these handshakes, both WPA- and WPA2-certified products are affected by our attacks.

The 4-way handshake provides mutual authentication and session key agreement. Together with (AES)-CCMP, a data-confidentiality and integrity protocol, it forms the foundation of the 802.11i amendment. Since its first introduction in 2003, under the name WPA, this core part of the 802.11i amendment has remained free from attacks. Indeed, the only currently known weaknesses of 802.11i are in (WPA-)TKIP [57, 66]. This data-confidentiality protocol was designed as a short-term solution to the broken WEP protocol. In other words, TKIP was never intended to be a long-term secure solution. Additionally, while several attacks against protected Wi-Fi networks were discovered over the years, these did not exploit flaws in 802.11i. Instead, attacks exploited flaws in Wi-Fi Protected Setup (WPS) [73], flawed drivers [13, 20], flawed random number generators [72], predictable pre-shared keys [45], insecure enterprise authentication [21], and so on. That no major weakness has been found in CCMP and the 4-way handshake, is not surprising. After all, both have been formally proven as secure [39, 42]. With this in mind, one might reasonably assume the design of the 4-way handshake is indeed secure.

Figura 2: “Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2”

El esquema del ataque sería el siguiente:

1. El atacante se intenta unir a la red *WiFi*.
2. Comienza el proceso *four-way handshake*.
3. Se negocia una nueva clave de cifrado en el paso tres, dentro del proceso *four-way handshake*.
4. No se envía la señal de ACK, la cual verifica que se ha recibido el mensaje del paso tres.
5. Este hecho fuerza que el punto de acceso retransmita de nuevo el paso tres varias veces.
6. Este proceso repetido varias veces, siempre instalará la misma clave de cifrado, por lo que se resetea a cero el *nonce*.
7. El atacante en este punto fuerza “reseteos” de tipo *nonce*, recolectándolos y repitiéndolos en el paso tres del *four-way handshake*.
8. Reutilizando estos *nonce* es posible repetir los paquetes y descifrarlos, ya que la misma clave de descifrado se utiliza con valores *nonce* que ya fueron utilizados en el pasado.

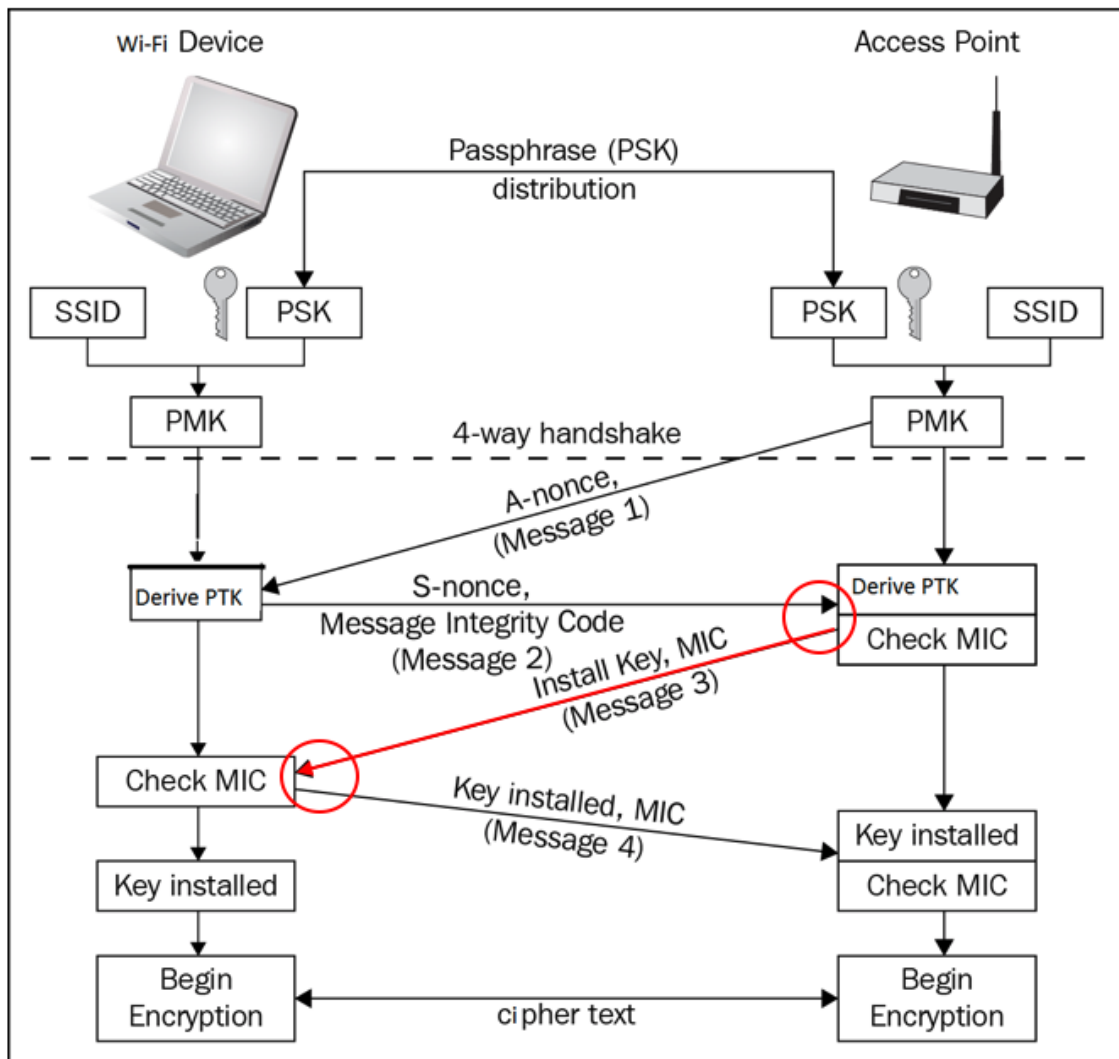


Figura 3: "Esquema de ataque KRACK"

### 1.1.- La nueva solución WPA3

La vulnerabilidad de KRACK fue parcheada por muchos proveedores y fabricantes, aunque, cómo se puede entender, quedarán muchos dispositivos que por mantenimiento o compatibilidad se quedarán con la vulnerabilidad en producción. La solución WPA3 se liberó a finales del año 2017.

Como se ha mencionado anteriormente, el protocolo WPA2 utiliza el conocido *four-way handshake* para llevar a cabo la autenticación. Con ataques como KRACK se ha evidenciado los problemas de seguridad que este mecanismo ofrece. El protocolo WPA3 implementará un nuevo mecanismo de *handshake* distinto a WPA2.

Además, el protocolo WPA3 contará con un nuevo cifrado de 192 bits, mejorando notablemente el utilizado por WPA2, que era de 128 bits.

### 2.- TOTP como parte del cifrado cambiante en la clave WiFi

En el año 2017 se llevó a cabo una investigación sobre cómo la temporalidad de una clave WiFi, mediante la utilización de un TOTP en la generación de dicha clave,

aportaba un valor a la seguridad de la red. El resultado de la investigación acabó en patente. Esta patente utiliza diferentes componentes, no solo el TOTP, como son la aleatoriedad y los rasgos biométricos del usuario.

La patente registrada en el año 2017 denominada “*A method and a system for encrypting wireless communications including authentication*” especifica un nuevo método para el cifrado y autenticación de los usuarios en entornos *Wireless*. La propuesta propone la utilización de un triple factor de seguridad: temporalidad baja en la gestión de claves, aleatoriedad en ciertas claves intermedias y la utilización de características biométricas del usuario para la generación de la clave. La suma de estos factores limitan en un gran valor la validez temporal de la clave y proporciona un mecanismo de autenticación periódica.

El esquema del trabajo, en un alto nivel de detalle, sería el siguiente:

1. En primer lugar, el dispositivo de red, el router, solicita al usuario el introducir sus rasgos biométricos. Para ello, el router dispone de un lector de huellas, reconocimiento facial o cualquier otro elemento que permita la lectura de rasgos biométricos.
2. El dispositivo de red almacena los datos y los convierte en elementos procesables.
3. El dispositivo de red genera una clave aleatoria.
4. El dispositivo de red genera un QRCode que el usuario debe leer con su dispositivo para conseguir el intercambio de la clave aleatoria.
5. El usuario lee el QRCode y obtiene la clave aleatoria.

En este instante, la tabla de almacenamiento de TOTP del dispositivo de red tiene un nuevo valor. El dispositivo almacenará los diferentes TOTP de cada usuario que tendrán acceso a la red. El dispositivo debe ir actualizando los diferentes valores en función de la temporalidad del TOTP. En la figura 4 se puede ver en detalle la siguiente parte.

6. El usuario solicita conectar con el dispositivo de red.
7. El dispositivo realiza el “*Authentication challenge*”.
8. El dispositivo del usuario solicita el rasgo biométrico al usuario en un instante de tiempo  $t$ . En caso de ser un reconocimiento facial, éste ocurre sin que el usuario tenga que realizar acción.
9. Se calcula un *secret* a través de la combinación de la característica biométrica extraída anteriormente y la cadena aleatoria. Este *secret* se genera de forma periódica en el dispositivo de red y es almacenado en su tabla de TOTP.
10. Se utiliza el *secret* anterior para combinarlo con el intervalo actual de tiempo. Se obtiene un  $K_c$ .
11. Se envía el tráfico cifrado con  $K_c$ .
12. El dispositivo de red identifica la MAC del cliente, asociándolo a su  $K_c$ .

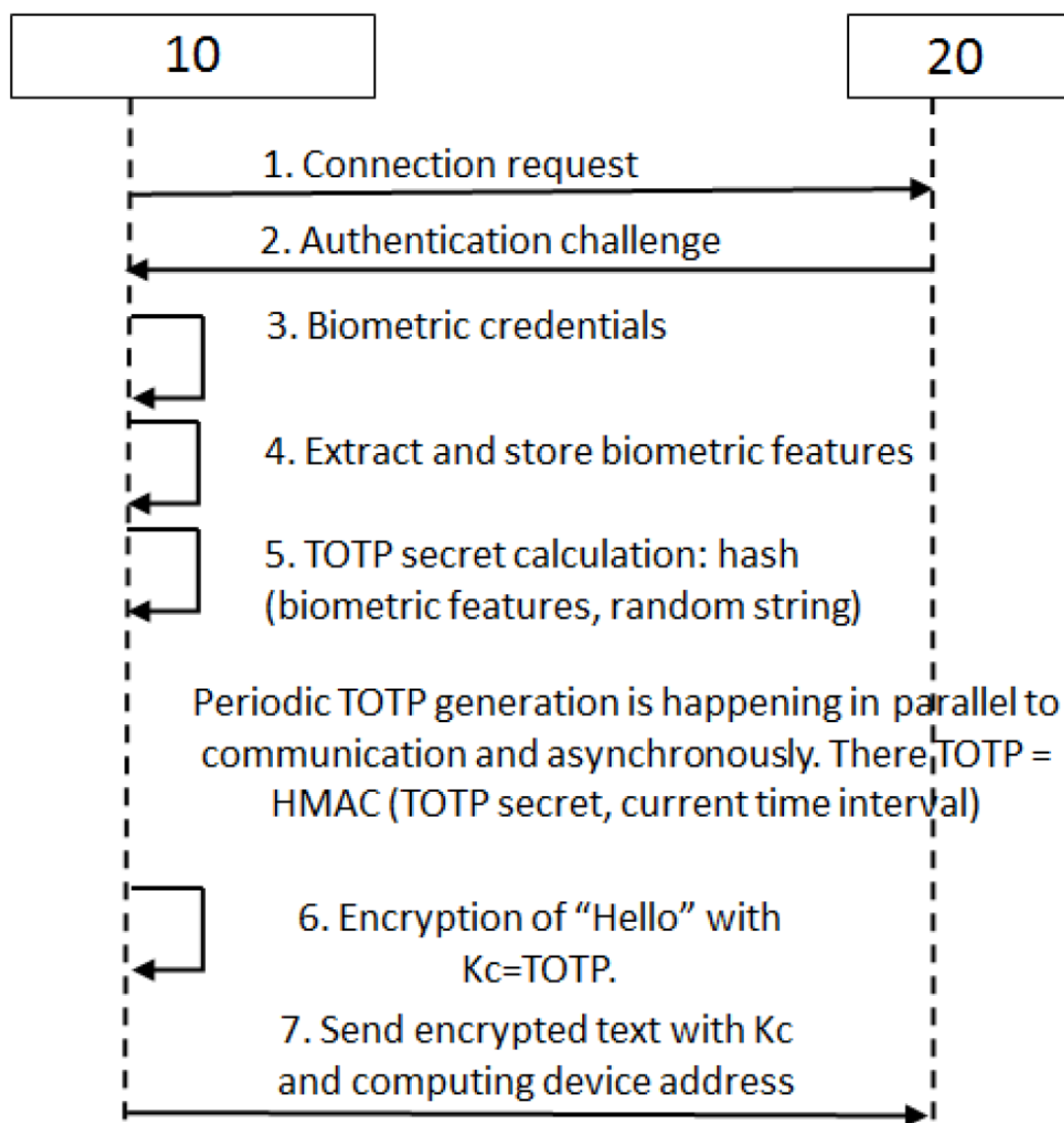


Figura 4: Creación de la Kc entre ambos dispositivos

Como parte de la investigación y patente anterior, se ha llevado a cabo una implementación parcial, basándose en la idea del cambio de clave temporal en una WiFi con protocolo WPA2. Los detalles de esta implementación se detallan en el siguiente apartado.

## 2.1.- Idea del uso de TOTP para generación de clave temporal en WPA2

La idea consiste en que el router y los dispositivos cliente asociados a la red utilicen un TOTP cambiante en un intervalo de tiempo definido. El TOTP que calculan el router y los dispositivos asociados es de 8 dígitos.

El TOTP generado formará parte de una clave que acabará siendo derivada de la unión del PSHK, *Pre-Shared Half Key*, y del propio TOTP. El PSHK es la parte aleatoria de la clave que se genera en el router y se debe configurar en el cliente. La generación de la clave en esta prueba de concepto es la siguiente:



$$Kc = Sha1( Sha1 (TOTP(8digits)) + Sha1 (PSHK))$$

La clave que se obtiene, denominada  $Kc$ , tiene una longitud de 40 caracteres. En este ejemplo el *charset* es  $[0-9,a-f]$ . La implementación está abierta a aplicar cualquier otro tipo de fórmula o método para derivar la clave de cifrado, pudiendo aplicar otros mecanismos más robustos con un mayor *charset*. Esta clave es utilizada para configurar la contraseña de la red *WiFi* y cambia a los  $X$  segundos, en función del tiempo que se haya configurado.

La distribución de la semilla del TOTP se realiza desde router hacia el resto de equipos y para ello existen dos vías:

- Generación del QRCode a través del portal web del router.
- Configuración manual en el cliente, introduciendo para ello el valor de la semilla que genera el TOTP y el valor asociado al PSHK.

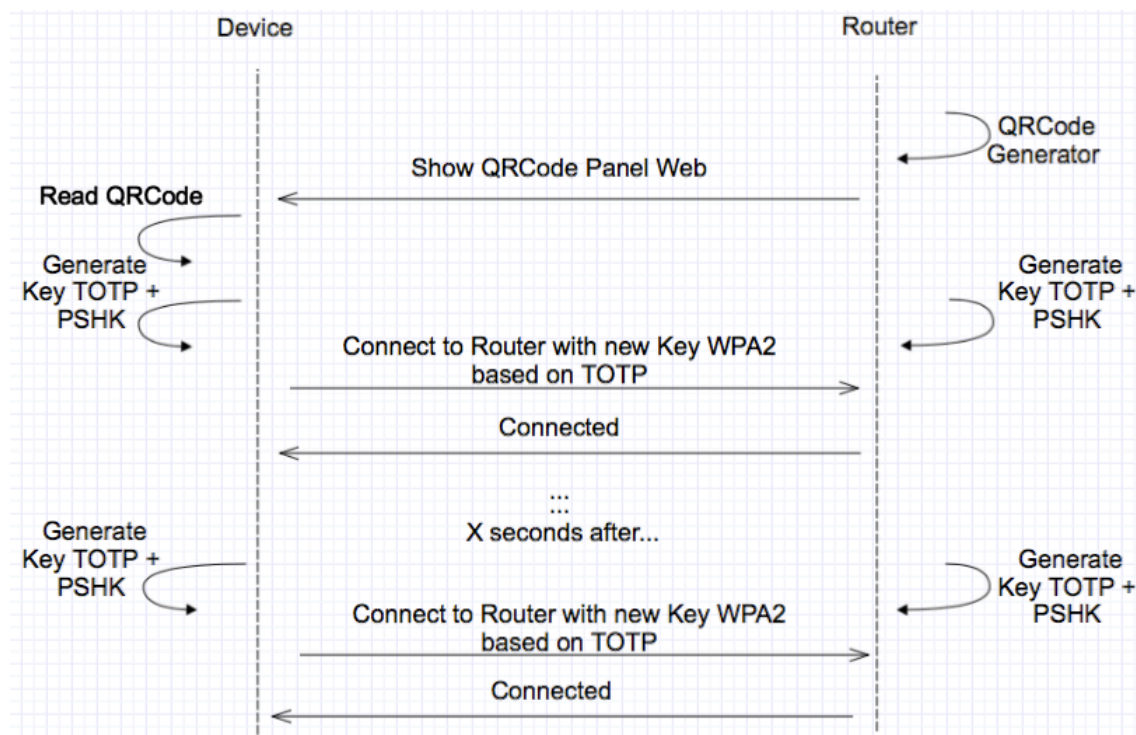


Figura 5: Esquema de generación y conexión con la clave basada en TOTP

### 3.- Implementación en OpenWRT y diferentes plataformas como cliente

Para llevar a cabo la idea de la utilización de un TOTP para la generación de una clave temporal, como parte de la implementación de la patente anterior, se decidió utilizar el *firmware OpenWRT* en su unión con *LEDE*. El router que albergue el *firmware* debe ser compatible. El router hará de servidor, mientras que se implementó un cliente en plataformas como *Android*, *Windows*, *macOS* y *GNU/Linux*.

En la siguiente dirección URL perteneciente al proyecto se puede visualizar la tabla de hardware compatible con OpenWRT: <https://openwrt.org/toh/start>. Para este trabajo se ha utilizado el modelo *Amper 26555* y el modelo *Comtrend WAP-5813n*.



Figura 6: Router utilizado. *Amper 26555*

Además, se ha modificado la aplicación LUCI de *OpenWRT* para que proporcione al usuario la posibilidad de configurar en el router todo lo necesario. En las opciones de configuración de la red *Wireless*, se puede encontrar el apartado de seguridad. Entre los protocolos de cifrado se encuentran los comunes WEP, WPA, WPA2 o red abierta. En esta implementación se incluye el tipo WPA2-TOTP.

Como se puede visualizar en la imagen, al seleccionar en el “*combo*” la opción WPA2-TOTP se puede elegir el intervalo en segundos del cambio de clave o generación del TOTP. Además, se puede generar un *secret*, que será la semilla para la generación del TOTP. Por último, el usuario puede generar una PSHK, la cual es la parte aleatoria que forma parte de la clave, la cual acabará derivada en  $K_c$ , tal y como se ha explicado en un punto anterior de este trabajo.



## Interface Configuration

General Setup

Wireless Security

MAC-Filter

Advanced Settings

Encryption

WPA2-TOTP

Step seconds

20

Secret

.....



Generate new secret

PSHK

.....



Generate new PSHK



Figura 7: Configuración WPA2-TOTP en el panel web de LUCI

### 3.1.- Implementación en un Router real con OpenWRT

Para llevar a cabo la implementación en un router real de *OpenWRT* y poder implementar el trabajo se ha necesitado realizar el “flasheo” de un dispositivo con este *firmware*. El procedimiento llevado a cabo se resume a continuación:

1. En primer lugar, se debe conectar el dispositivo vía cable *Ethernet*.
2. Asegurarse de que el firmware que se quiere ‘flashear’ es correcto con el modelo de hardware. El fichero se llamará *factory.bin*.
3. Hacer *login* en el panel web denominado LUA que ofrece el router. Se debe acceder a la pestaña *System* y luego al apartado *Flash Firmware*.
4. Espera mientras el dispositivo escribe la imagen del *firmware* en memoria. Esto puede tardar varios minutos. Al final del proceso, el dispositivo reiniciará automáticamente.
5. El nuevo *firmware* habrá sido instalado.

### Flash new firmware image

Upload a sysupgrade-compatible image here to replace the running firmware. Check "Keep settings" to retain the current configuration (requires a compatible firmware image).

Keep settings: ☒

Image:

Examinar...

No se ha seleccionado ningún archivo.

Flash image...

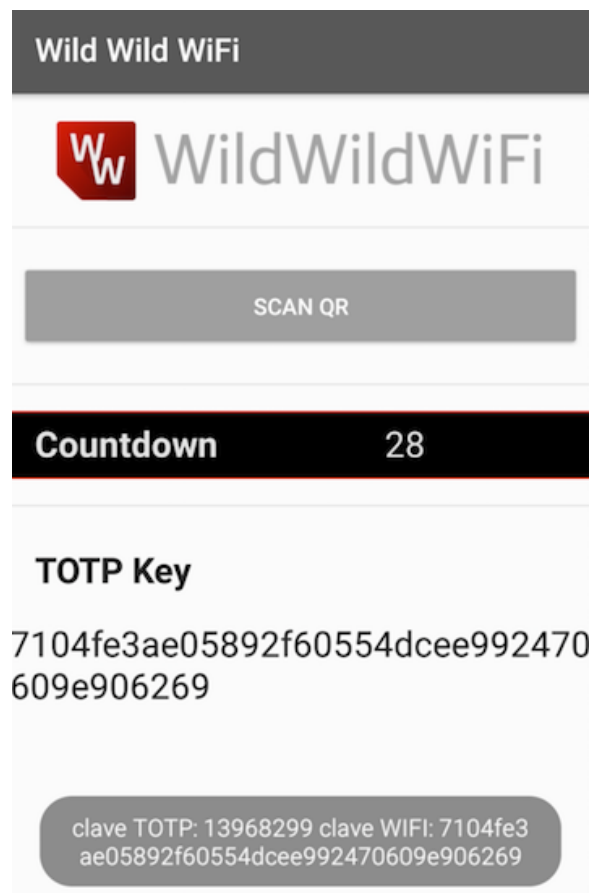
*Figura 8: Flash del nuevo Firmware*

### 3.2.- Caso 1: App Android con TOTP WiFi implementado

Se ha implementado una app para la plataforma Android con las siguientes características:

- La App dispone de un lector de *QRCode*.
- En el *QRCode*, el router inserta toda la información necesaria: semilla para la generación del TOTP, la clave *Pre-Shared Half Key*, el nombre del SSID y el intervalo en segundos de la generación del nuevo TOTP.
- En el instante que la app obtiene esta información puede hacer el cálculo de la clave derivada, basándose en el TOTP que se genera cada 'x' segundos y la clave *Pre-Shared Half Key*.

El aspecto que tiene la app es el que se puede visualizar en la siguiente imagen. Como se puede ver,



*Figura 9: Aspecto de la app WildWildWiFi para Android*

El aspecto visual de la app nos muestra la posibilidad de escáner un *QRCode*. Una vez escaneado muestra la semilla del TOTP y la contraseña actual, así como el tiempo de vida que le queda a la contraseña.

La conexión a la red *WiFi* con SSID concreto se realiza automáticamente cada 'x' segundos. De esta forma, el tiempo de vida de una contraseña de una red WPA2 viene definida por el intervalo que se haya configurado en el router. En la imagen 7, se puede visualizar la configuración que se puede realizar sobre la opción de seguridad WPA2-TOTP.

### 3.3.- Caso 2: Servicio en Windows 10 con TOTP WiFi implementado

Se ha implementado un servicio para los equipos *Microsoft Windows* con los que se pueda conectar el equipo a la configuración WPA2-TOTP. El servicio no tiene la posibilidad de leer un *QRCode*, como en el caso anterior, por lo que el usuario deberá insertar manualmente las diferentes opciones:

- SSID. Nombre de la red a la que se va a conectar con el protocolo WPA2-TOTP.
- El intervalo en segundos. Este intervalo mide el número de segundos que pasarán hasta generar un nuevo TOTP, el cual será utilizado en la obtención de una clave derivada que permitirá la conexión a la red *WiFi*.
- El campo *Secret Key* indica la semilla con la que se generará el TOTP en función del tiempo.
- El campo *PSHK*. Este campo es la clave que se compartirá entre ambos dispositivos y que será utilizada en la generación de la clave temporal de conexión a la red *WiFi*.

Como se puede visualizar en la imagen, el servicio proporciona una barra de tiempo en el cual se indica el tiempo que queda para finalizar la vida de la clave. Además, hay un campo denominado '*Current Password*' el cual muestra la contraseña actual que hay en vigor en la red.

Lógicamente, como prueba de concepto, los campos de contraseña y datos sensibles como la semilla del TOTP o el valor de la *PSHK* están visibles en todo momento. Si se quisiera aplicar la aplicación en un entorno real, se deberían proteger dichos valores.

El servicio se puede detener y arrancar en cualquier instante. Para algunas pruebas se ha utilizado un modo paranoico que permite el cambio de contraseñas cada 30-60 segundos. El uso recomendado de este tipo de solución sería crear un intervalo de tiempo mayor, por ejemplo, 60 minutos o, incluso, un cambio de contraseña al día. La opción, modo paranoico puede utilizarse en entornos críticos, aunque se puede considerar modo paranoico hasta un rango de minutos.



Figura 10: Aspecto del servicio ejecutándose en Windows 10

#### 4.- El concepto de SSID Pinning

La conexión entre un cliente y un punto de acceso *WiFi* viene regida, generalmente, por la debilidad de un nombre y un tipo de configuración de seguridad. En otras palabras, los atacantes podrán engañar a un dispositivo *WiFi* haciéndole creer que la red *WiFi* que él conoce se encuentre en el radio de acción. Esto se puede llevar a cabo a través del uso de un SSID falso que coincida con el real y un tipo de configuración de seguridad que coincida, incluyendo la clave de la red y si ésta tiene, con el del punto de acceso *WiFi* legítimo.

El SSID Pinning propone varias propiedades o combinación de éstas para “certificar” que el punto de acceso *WiFi* que se tiene en el radio de acción es quién dice ser. En el presente trabajo se han usado varios elementos para definirlo, susceptibles de mejora y ampliación en el futuro, que definirán la granularidad de la seguridad en el Pinning según el criterio y umbral que se defina sobre cada parámetro.

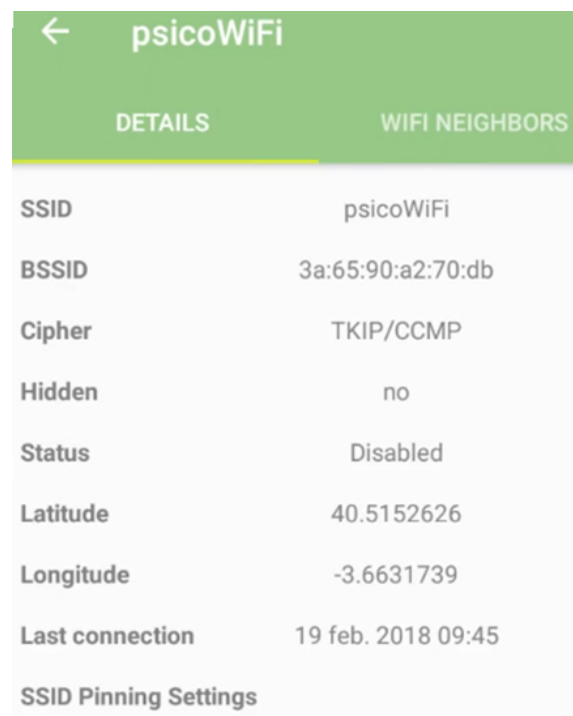
En la implementación experimental llevada a cabo para el presente trabajo se ha decidido que el usuario pueda escoger la granularidad, es decir, qué elementos son los que forman parte de su ‘Pinning’ al punto de acceso. Estas medidas pueden resultar realmente útiles en entornos del día a día, en el que un usuario se está moviendo y su dispositivo se enfrenta a diferentes puntos de acceso *WiFi* en el que, ante la posibilidad

de conectar automáticamente, se pueden terminar conectando. Si el dispositivo se conecta a un punto de acceso no legítimo se estará exponiendo la información de la comunicación, en algunos casos.

En resumen, el SSID Pinning propone ciertas características para poder “fijar” y “certificar” que el dispositivo no se conectará con un punto de acceso no legítimo sin que el usuario sea consciente de ello y por último, la granularidad permite al usuario escoger qué características forman parte de su SSID Pinning, siendo la unión de las varias el modo más robusto de prevención de conexiones no deseadas.

#### 4.1.- Caso 3: App Android con SSID Pinning implementado

En este caso, se presenta la *app* de *Android* que implementa el concepto de SSID Pinning. En esta *app* se crean automáticamente los perfiles *WiFi* a los que el usuario se conecta y éste puede configurar el nivel de granularidad de SSID Pinning quiere. El usuario puede escoger, en función de su necesidad o en función de los requisitos de la red, la configuración.



psicoWiFi	
DETAILS	WIFI NEIGHBORS
SSID	psicoWiFi
BSSID	3a:65:90:a2:70:db
Cipher	TKIP/CCMP
Hidden	no
Status	Disabled
Latitude	40.5152626
Longitude	-3.6631739
Last connection	19 feb. 2018 09:45
SSID Pinning Settings	

Figura 11: PsicoWiFi versión Android

#### 4.2.- Caso 4: Herramienta en Windows 10 con SSID Pinning implementado

El presente caso muestra el uso de la herramienta *PsicoWiFi* en el sistema operativo *Windows 10*. La herramienta permite configurar diferentes opciones para controlar la característica SSID Pinning.

En la imagen se puede visualizar el listado de redes *WiFi* vecinas asociadas a un perfil *WiFi* conocido. Este listado se actualiza cada vez que el dispositivo se conecta a una red

conocida. Además, la aplicación utiliza un número que permite identificar el umbral que podría considerar seguro para la configuración del pinning. Las características de protección para SSID Pinning que ofrece *PsicoWiFi* en la herramienta de escritorio pueden ser diferentes a las ofrecidas en su versión móvil.

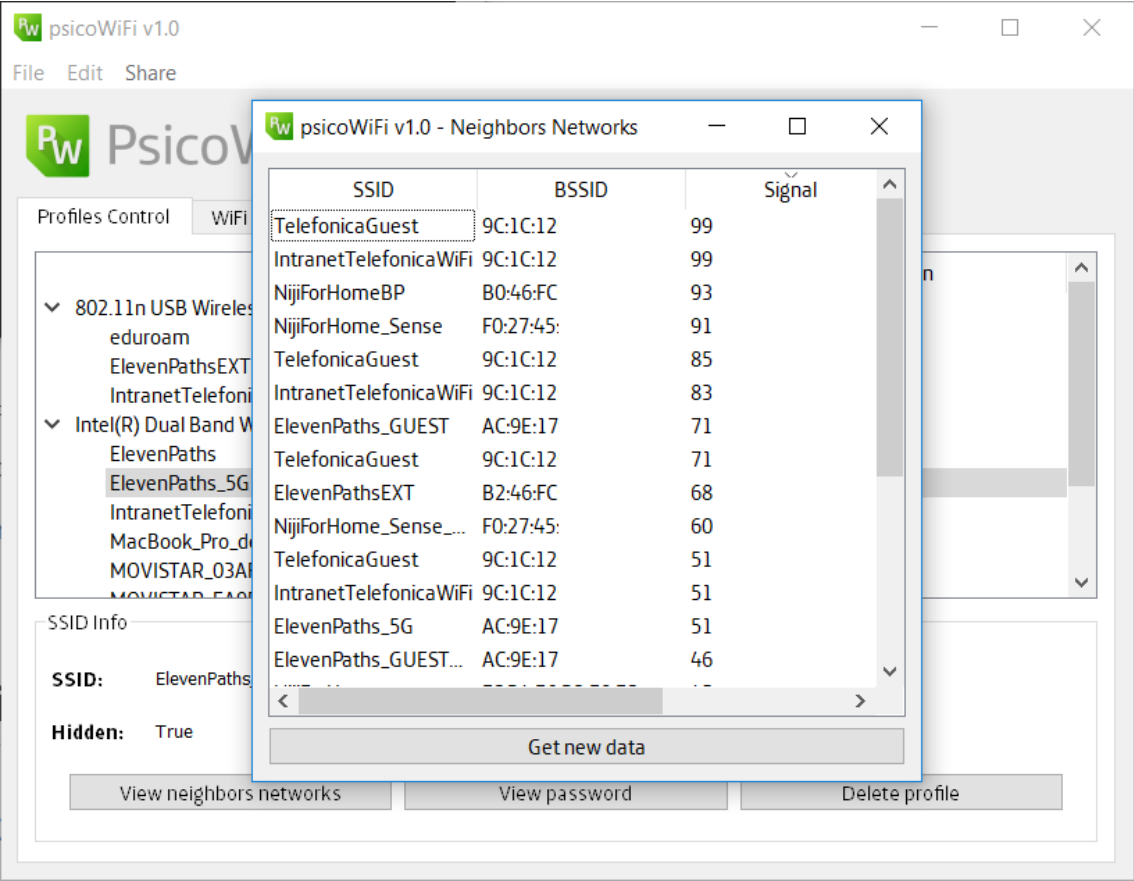


Figura 12: PsicoWiFi y el listado de redes WiFi vecinas

5.- Conclusiones

Actualmente siguen existiendo amenazas en los entornos *WiFi*. En este presente trabajo se han presentado varios conceptos que ayudan a incrementar la seguridad de las redes inalámbricas en entornos corporativos como entornos particulares. El uso de la componente temporal en el cambio de claves WPA2 basadas en el uso de un TOTP, minimiza exponencialmente el riesgo de exposición de una clave. Además, el uso del concepto SSID Pinning en dispositivos de escritorio o móviles ayudan a evitar conexiones no queridas a puntos de accesos falsificados.

El presente trabajo es un trabajo eminentemente práctico, por lo que podrá ser probado en cualquier instante por cualquier interesado.

## 6.- Referencias

- Paper: *"Living in the Jungle: Legitimate users in legitimate insecure Wireless networks"*. Chema Alonso, Rodolfo Bordón, Alejandro Martín, Antonio Guzmán. <https://www.slideshare.net/chemai64/viviendo-en-la-jungla-27121575>
- Instalación del firmware OpenWRT. [https://openwrt.org/docs/guide-quick-start/factory\\_installation](https://openwrt.org/docs/guide-quick-start/factory_installation)
- Listado de hardware compatible con OpenWRT. <https://openwrt.org/toh/start>
- Proyecto TOTP WiFi disponible en el Github de ElevenPaths: <https://github.com/elevenpaths>
- <https://www.elevenpaths.com/es/labsp/herramientas/index.html>