



Published on [hacktimes.com](http://www.hacktimes.com) (<http://www.hacktimes.com>)

## Rompiendo el protocolo WPA con clave precompartida (PSK) y (TKIP) .. Acelerando el proceso.

By modlok

Creado 7 Jul 2006 - 16:45

En el artículo [anterior](#) [1] vimos como se podía romper el protocolo WPA con clave precompartida PSK (TKIP), a través de un ataque de fuerza bruta.

Recordamos una vez obtenida la clave del protocolo WPA-PSK(TKIP) las conclusiones que obtuvimos...

Como todo proceso aquí existen 2 problemas, que pasaremos a comentar:

- Tiempo: más de 5 horas
- Rate: 172 palabras/segundo

El tiempo que se tarda en completar un fichero de más de 4 millones de entradas, que no es mucho, es excesivo y es debido al rate o número de palabras verificadas por segundo.

Para acelerar este proceso es necesario precalcular el PMK, para ello podemos utilizar la utilidad **genpmk** de **Cowpatty**.

**Cowpatty** es una herramienta específica para la ruptura del protocolo WPA-PSK(TKIP) que podríamos haber empleado en la primera parte del artículo.

Teniendo en cuenta que la generación del PSK se realiza de la siguiente manera:

```
PSK = PMK = PBKDF2 (frase, SSID, longitud SSID 4096, 256)
```

El problema al que nos enfrentamos a la hora de atacar el protocolo WPA con diccionarios de texto plano es que, como se pudo observar, es un proceso exageradamente lento. Algunos pensarán que un ataque offline de más 5 horas es muy poco, pero teniendo en cuenta que el diccionario era de apenas 4 millones de entradas cambia un poco la perspectiva.

Como podemos observar en la fórmula anterior, el PMK esta compuesto por una serie de valores o atributos donde el SSID y la longitud del mismo entre otros son siempre los mismos para un mismo dispositivo.

Siendo esto así podremos precalcular un hash PMK donde solo se compare el valor cambiante, es decir, la clave.

Este proceso como el que se describe en el artículo anterior no esta exento de problemas ya que, como os habréis dado cuenta, es necesario generar el fichero de hash para cada ESSID, siendo imposible la reutilización del mismo.

Por ello, la generación de un fichero de hash para un ESSID no genérico o por defecto es más larga y costosa que el propio ataque a través de un diccionario de texto plano. ¿Donde esta la gracia entonces? Es la pregunta que muchos se estarán haciendo en estos momentos, pero la realidad es otra si tenemos en cuenta los miles de puntos de acceso que vienen con el ESSID por defecto, ya sea del fabricante o del ISP. Entonces ¿y si encontramos o nos ponemos a generar ficheros de hash con nombres de ESSID por defecto?

Probablemente nunca podremos recopilar todos los ESSID por defecto y menos aún generarlos con un diccionario de más de 50 millones de entradas, tardaríamos toda una vida y parte de la otra. De ahí la importancia de compartir este tipo de ficheros.

A continuación mostraremos como generar los ficheros de hash y hacer uso de las mismas. En nuestro caso, y por seguir el hilo del artículo anterior, precalcularemos el PMK para cada una de las entradas del diccionario con el **ESSID HACKTIMES**.

- Generar el fichero:

```
# genpmk -f ModlokHacktimesDICCsr.txt -d HACKTIMES_COW -s  
HACKTIMES
```

Opción -f: Diccionario.

Opción -d: Nombre del fichero de hash que se generará.

Opción -s: El Essid.

- Ejecución del ataque:

```
# cowpatty -r pskcrack-01.cap -d HACKTIMES_COW -s HACKTIMES  
Opción -r: Fichero *.cap donde capturamos el handshake con el  
airodump.  
Opción -d: Nombre del fichero generado con la utilidad genpmk.  
Opción -s: El Essid.
```

```
Collected all necessary data to mount crack against  
passphrase.
```

```
Starting dictionary attack. Please be patient.
```

```
key no. 100000: ÔÁÓÐÄÊÏÔØÑ
```

```
key no. 200000: after-written
```

```
...
```

```
The PSK is "ZinedineZidane".
```

```
3586552 passphrases tested in 50.24 seconds: 71385.42  
passphrases/second
```

Las previsiones han sido mejores que las que esperábamos, que eran de unas 20000 palabras por segundo. Como se puede observar ha testado 3 millones y medio de claves en 50 segundos, muy por debajo de las más de 5 horas de la vez anterior.

### Enlaces relacionados:

[http://www.remote-exploit.org/index.php/Main\\_Page](http://www.remote-exploit.org/index.php/Main_Page)  
<http://madwifi.org/>  
<http://www.wirelessve.org/>  
<http://wrzwaldo.org/hash-tables/>  
<http://www.renderlab.net/projects/WPA-tables/>  
[http://www.churchofwifi.org/Project\\_Display.asp?PID=87&S=wpa](http://www.churchofwifi.org/Project_Display.asp?PID=87&S=wpa)

---

**Source URL:**

<http://www.hacktimes.com/?q=node/35>

**Links:**

[1] <http://www.hacktimes.com/?q=node/34>