BRUTEFORCE WPS CON INFLATOR

Por Sergio Baamonde Álvarez

En este artículo trataremos cómo aprovecharnos de una vulnerabilidad en el protocolo WPS para obtener la clave de seguridad WPA de una red inalámbrica con WPS activado. Para ello, utilizaremos la distribución Xiaopan y la aplicación Inflator que implementa a su vez la aplicación wps-reaver, desarrollada para este tipo de ataques.

BRUTEFORCE WPS CON INFLATOR

Por Sergio Baamonde Álvarez

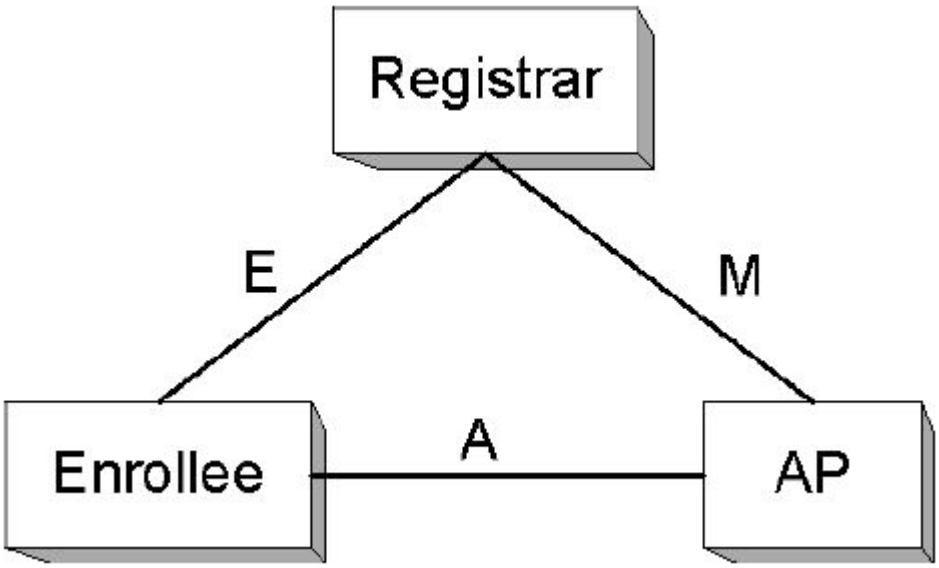
INTRODUCCIÓN

¿Qué es WPS?

WPS son las siglas de *Wi-Fi Protected Setup*. Es un estándar de 2007 para facilitar la creación de redes WLAN a gente con pocos conocimientos sobre seguridad inalámbrica, ya que es posible que se sientan intimidados ante la abundancia de opciones de seguridad disponibles en una red inalámbrica.

En sí mismo, WPS no es un mecanismo de seguridad, sino que simplifica la conexión de los dispositivos a la red al no tener que introducir la contraseña de la red.

Estructura



- **Registrar** (matriculador): es el dispositivo que puede crear o anular las credenciales en la red. Puede serlo o bien el punto de acceso (estando integrado en él) o cualquier otra estación o dispositivo de la red. Es posible la existencia de más de un *registrar* en una red.
- **Enrollee** (matriculado): es el dispositivo que solicita el acceso a la red.
- **Authenticator** (autenticador): es el punto de acceso funcionando de proxy entre el *Registrar* y el *Enrollee*.

Tipos de WPS

- **PIN:** existen 2 formas de conectarse mediante PIN. Una de ellas, la más generalizada, consiste en que el dispositivo que desea conectarse proporcione un número PIN (definiremos más adelante su estructura) el cual debe ser introducido en el "representante" de la red (normalmente el punto de acceso). Alternativamente, el punto de acceso puede proporcionar un PIN que deberá introducir el cliente en su dispositivo. Esta forma de acceso es obligatoria para todos los productos WPS.
- **PBC:** el usuario debe pulsar un botón (físico o virtual) en el punto de acceso y en el nuevo dispositivo. Este modo es obligatorio para los puntos de acceso y opcional para los dispositivos que deseen conectarse.
- **NFC:** el usuario debe aproximar el dispositivo al punto de acceso para permitir una comunicación NFC (*Near Field Communication*) entre ambos. Se presupone que a un dispositivo cercano físicamente al punto de acceso se le permitirá el susodicho acceso.
- **USB:** las credenciales se transmiten mediante una memoria flash.

VULNERABILIDAD

Introducción

Descubierta en Diciembre del 2011 por Stefan Viehböck, permite a un atacante recuperar el PIN WPS de la red y por ende, la clave pre-compartida de la red (sea WPA o WPA2) con un ataque de fuerza bruta en un tiempo reducido. La única solución efectiva es deshabilitar el acceso por WPS.

Esta vulnerabilidad se basa en los mensajes de entendimiento entre el *registrar* y el *enrollee* cuando intentan validar un PIN. El protocolo de registro WPS establece una serie de intercambio de mensajes EAP (*Extensible Authentication Protocol*) de la siguiente manera:

```
Enrollee -> Registrar: M1 = Version || N1 || Description || PKE
Enrollee <- Registrar: M2 = Version || N1 || N2 || Description || PKR [ ||
ConfigData ] || HMAC_AuthKey(M1 || M2*)
Enrollee -> Registrar: M3 = Version || N2 || E-Hash1 || E-Hash2 ||
HMAC_AuthKey(M2 || M3*)
Enrollee <- Registrar: M4 = Version || N1 || R-Hash1 || R-Hash2 ||
ENC_KeyWrapKey(R-S1) || HMAC_AuthKey (M3 || M4*)
Enrollee -> Registrar: M5 = Version || N2 || ENC_KeyWrapKey(E-S1) ||
HMAC_AuthKey (M4 || M5*)
Enrollee <- Registrar: M6 = Version || N1 || ENC_KeyWrapKey(R-S2) ||
HMAC_AuthKey (M5 || M6*)
Enrollee -> Registrar: M7 = Version || N2 || ENC_KeyWrapKey(E-S2
[||ConfigData]) || HMAC_AuthKey (M6 || M7*)
Enrollee <- Registrar: M8 = Version || N1 || [ ENC_KeyWrapKey(ConfigData) ] ||
HMAC_AuthKey (M7 || M8*)
```

Tras explicar que R-Hash1 y R-Hash2 son comprobaciones previas hechas por el *registrar* para demostrar el conocimiento de las 2 mitades del PIN del *enrollee*, no entraremos a describir qué significa cada parámetro de este protocolo, sino que pasaremos a enunciar la vulnerabilidad en base a él:

Descripción

Esta vulnerabilidad se basa en la autentificación por PIN, ya que simplemente con conocerlo podemos acceder a la red (ya que al descubrir el PIN accedemos a la clave WPA/WPA2), por lo que con un ataque de fuerza bruta tendríamos acceso.

El PIN es un número de 8 cifras decimales, de las cuales la 8ª es un checksum de las otras 7 de la siguiente forma (implementación en C, esta función devuelve el dígito checksum):

```
unsigned int wps_pin_checksum(unsigned int pin)
{
	unsigned int accum = 0;
	while (pin) {
		accum += 3 * (pin % 10);
		pin /= 10;
		accum += pin % 10;
		pin /= 10;
	}

	return (10 - accum % 10) % 10;
}
```

Por lo cual, si quisiéramos realizar un ataque por fuerza bruta, tendríamos 10^7 = 10000000 posibilidades, lo cual sería impracticable (ya que el tiempo medio por comprobación de PIN suele estar entre los 0,5-3 segundos, tardaríamos 348 días en dar con él en el peor de los casos.

Sin embargo, la vulnerabilidad se basa en el protocolo de registro anteriormente descrito, ya que se dan las siguientes circunstancias:

- Si el protocolo de registro falla en algún punto, el *registrar* enviará un mensaje NACK.
- Si el atacante recibe un mensaje NACK después de enviar el M4, sabrá que la primera mitad del PIN es incorrecta (ya que como hemos descrito antes, R-Hash1 comprueba que la primera mitad del PIN sea correcta, si recibimos un NACK es que este no es el caso).
- Si el atacante recibe un NACK después de enviar M6, sabe que la segunda mitad del PIN es incorrecta (análogamente al mensaje M4).

Este método reduce las posibilidades máximas a 10^4+10^3 = 11000, ya que solamente tenemos que comprobar la primera mitad del PIN y, cuando sepamos que es correcta (cuando recibamos M5 en lugar de NACK), comprobar la segunda mitad del PIN (de la cual solo necesitamos 3 dígitos, ya que el 4º es el checksum anteriormente mencionado).

Aprovechamiento

Una herramienta llamada wps-reaver ha sido desarrollada para permitir atacar por fuerza bruta a dispositivos con WPS activado. Su funcionamiento es muy sencillo: primero realiza un ataque de fuerza bruta contra la primera mitad del PIN y cuando la encuentra, pasa a realizarlo contra la segunda mitad del PIN.

En este report pasaremos a explicar cómo obtener una clave WPA mediante el uso de una versión de wps-reaver llamada **Inflator** que además de contener el programa wps-reaver, también contiene los programas airmon-ng (necesario para activar/desactivar el modo monitor de una tarjeta de red inalámbrica) y airodump-ng (para capturar paquetería de red inalámbrica), con lo que la inicialización de la tarjeta para poder empezar el ataque se hace de manera mucho más rápida y sencilla.

XIAOPAN E INFLATOR

Introducción

Utilizaremos la versión 0.4.7.2 de la distribución Xiaopanbasada en Tiny Core Linux que puede ser descargada desde http://sourceforge.net/projects/xiaopanos/files/XIAOPAN%200.4.7.2.iso/download . Xiaopan contiene una serie de aplicaciones orientadas al análisis de vulnerabilidades en redes inalámbricas.

Como tarjeta de red emplearemos una AlfaTools AWUS036H con chipset RTL8187, siendo este chipset uno de los más empleados a la hora de analizar vulnerabilidades en redes WLAN.

Por último, como hemos dicho, emplearemos la herramienta Inflator para automatizar el proceso de wps-reaver a través de una sencilla interfaz gráfica.

El modo de empleo será arrancando el PC desde esta distribución, aunque cabe esperar que si se ejecuta desde una máquina virtual el resultado obtenido sea el mismo. Se ha preferido hacer de esta manera para evitar problemas de configuración del dispositivo inalámbrico o interferencias con otros programas/servicios del sistema operativo host.

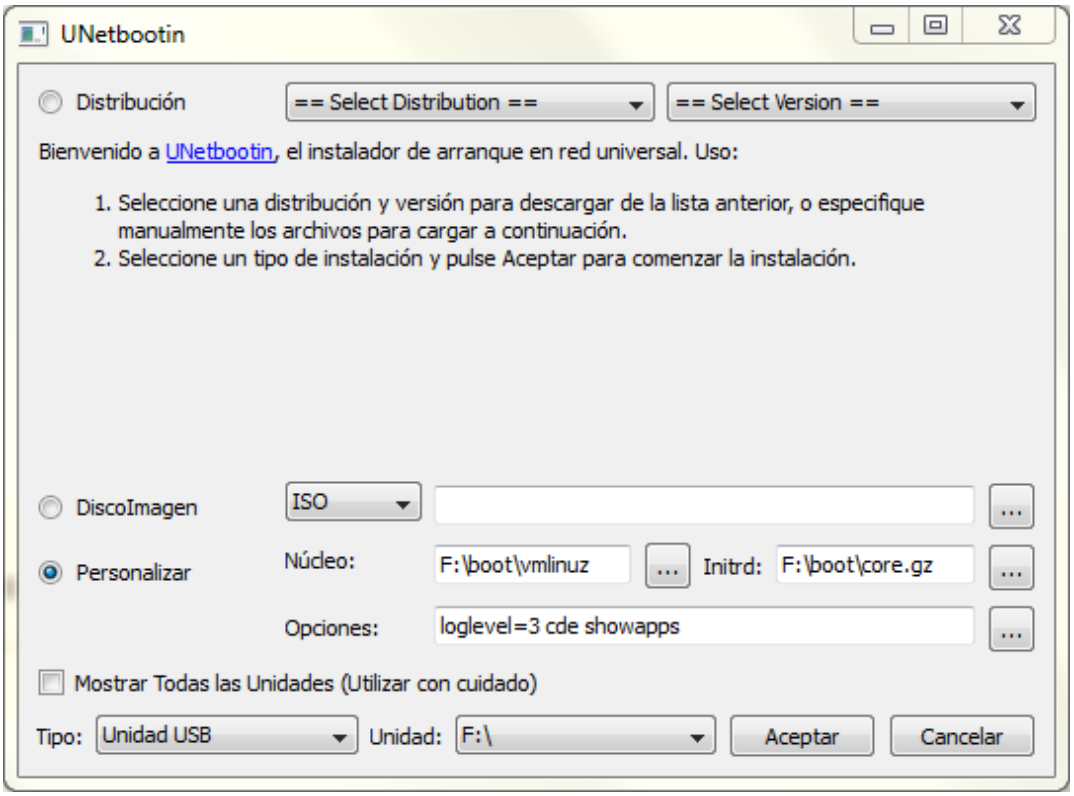
Carga del sistema operativo

Tras descargar la imagen ISO de la distribución, procedemos a extraer los archivos (mediante alguna herramienta como 7-Zip: http://www.7-zip.org/) en la memoria USB que queramos emplear para la carga de Xiaopan. Previamente la habremos formateado con formato FAT32.

Para conseguir que la memoria arranque el sistema operativo, emplearemos la herramienta Unetbootin disponible en http://unetbootin.sourceforge.net/ . Hemos tenido que emplear la versión 408 ya que la última versión en el momento de escribir estas líneas no da soporte para configurar los parámetros de boot.

Procedemos ahora a seleccionar como kernel el archivo boot/vmlinuz, como initrd el archivo boot/core.gz y en opciones abriremos el archivo boot/isolinux/isolinux.cfg, todo esto desde la raíz de la memoria USB. Con estas opciones debería bastar, si se requieren más parámetros de configuración o existe algún problema de arranque podemos acceder a http://xiaopan.co/run-from-usb/ para más detalles.

Tras tener los parámetros configurados, la pantalla debería quedar así:



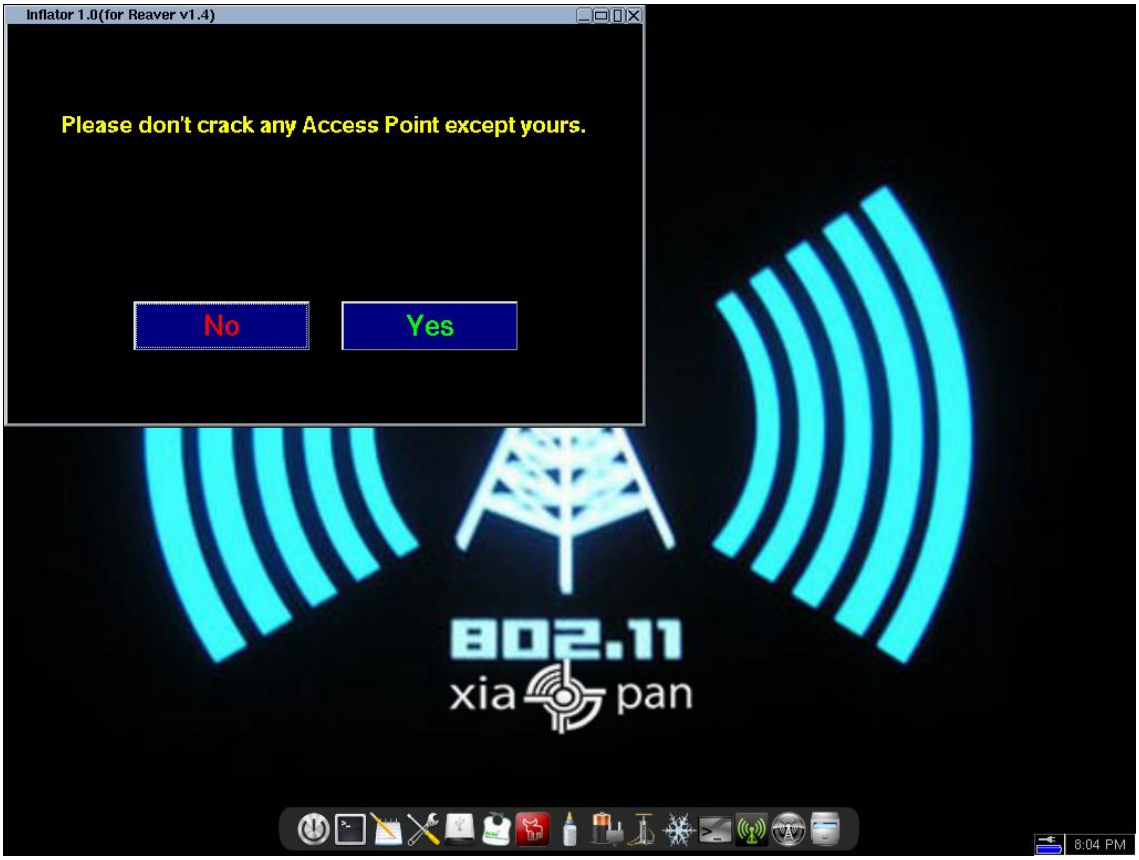
Obviamente seleccionaremos en la lista desplegable Unidad la letra correspondiente a la memoria USB. Tras esto, pulsaremos Aceptar y ya tendremos la memoria lista para el arranque.

Tras reiniciar el ordenador, procederemos a abrir el menú de arranque (en nuestro caso con la tecla F8) y seleccionar el arranque desde la memoria flash. Nos aparecerá una ventana para arrancar nuestra distribución, pulsaremos la tecla Enter para hacerlo.

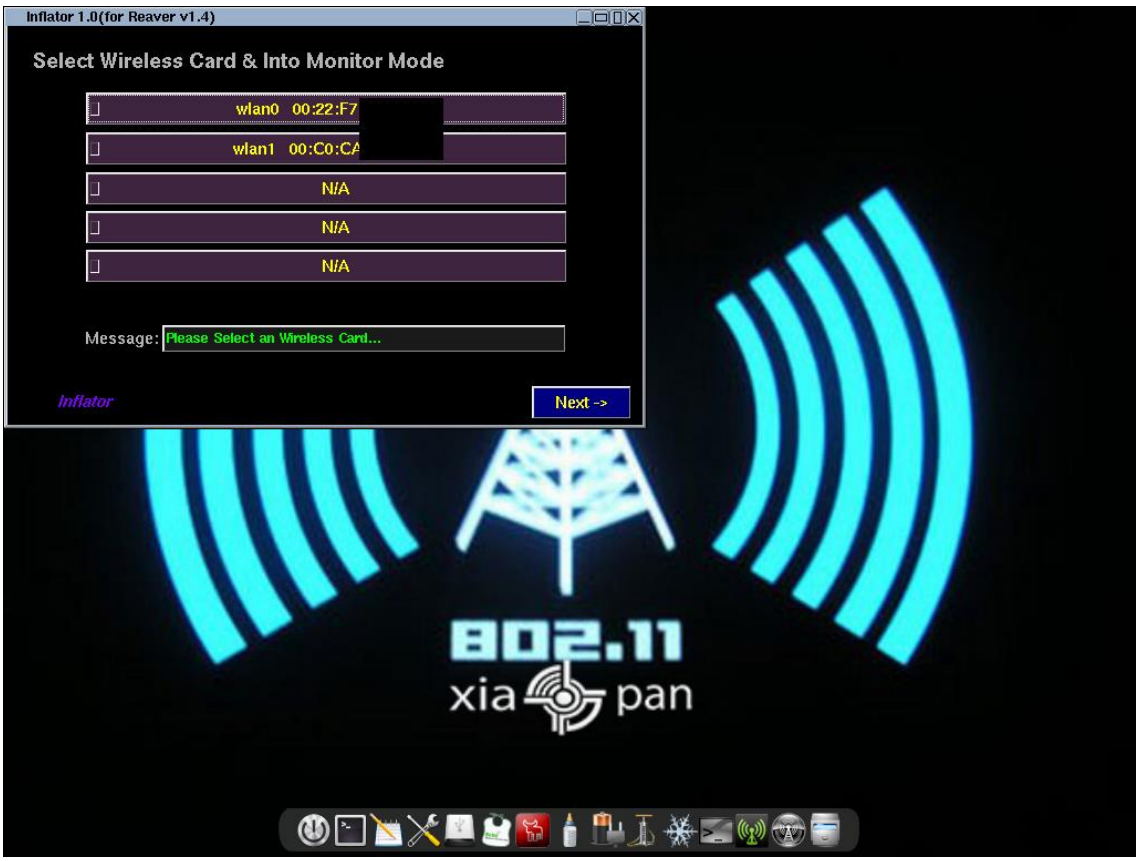
Tras arrancar el sistema, nos recibirá el escritorio de Xiaopan:

Inflator

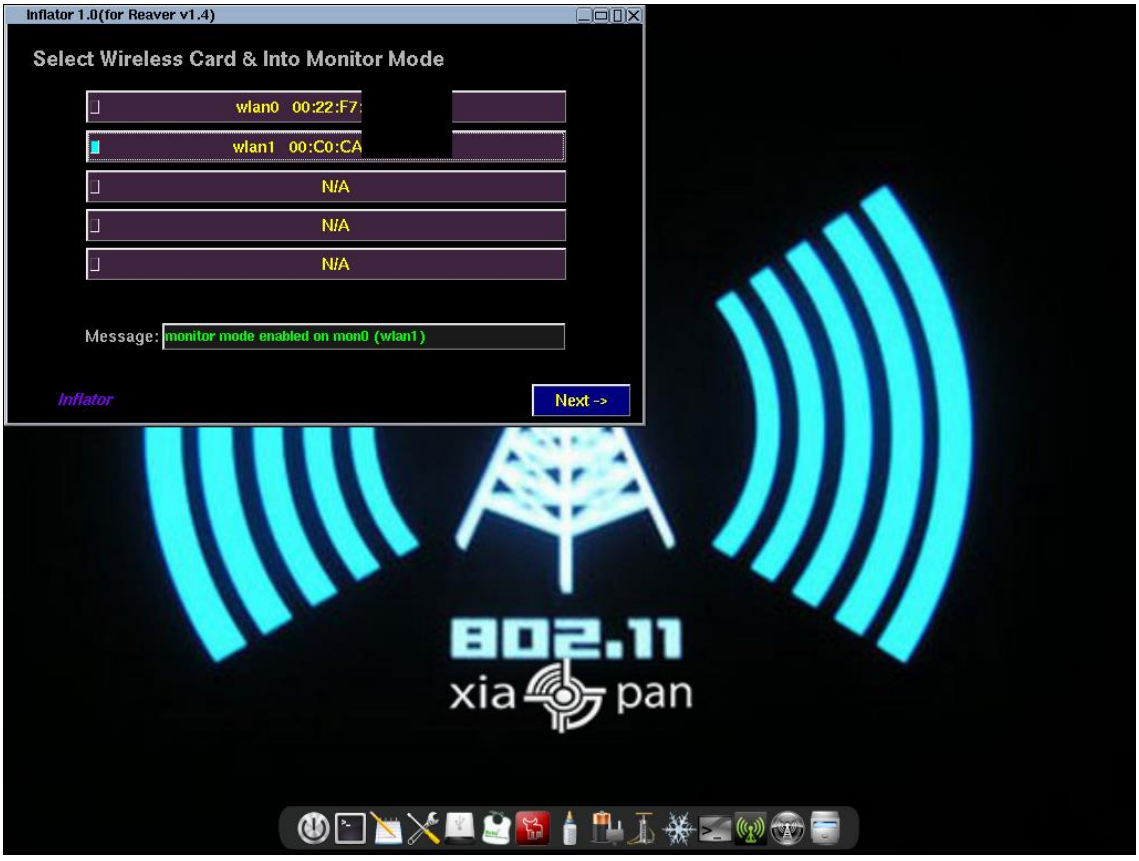
Procedemos ahora a abrir la aplicación Inflator haciendo click en el icono con aspecto de *inflador* de aire. Aparecerá la siguiente ventana, aceptamos las condiciones pulsando Yes:



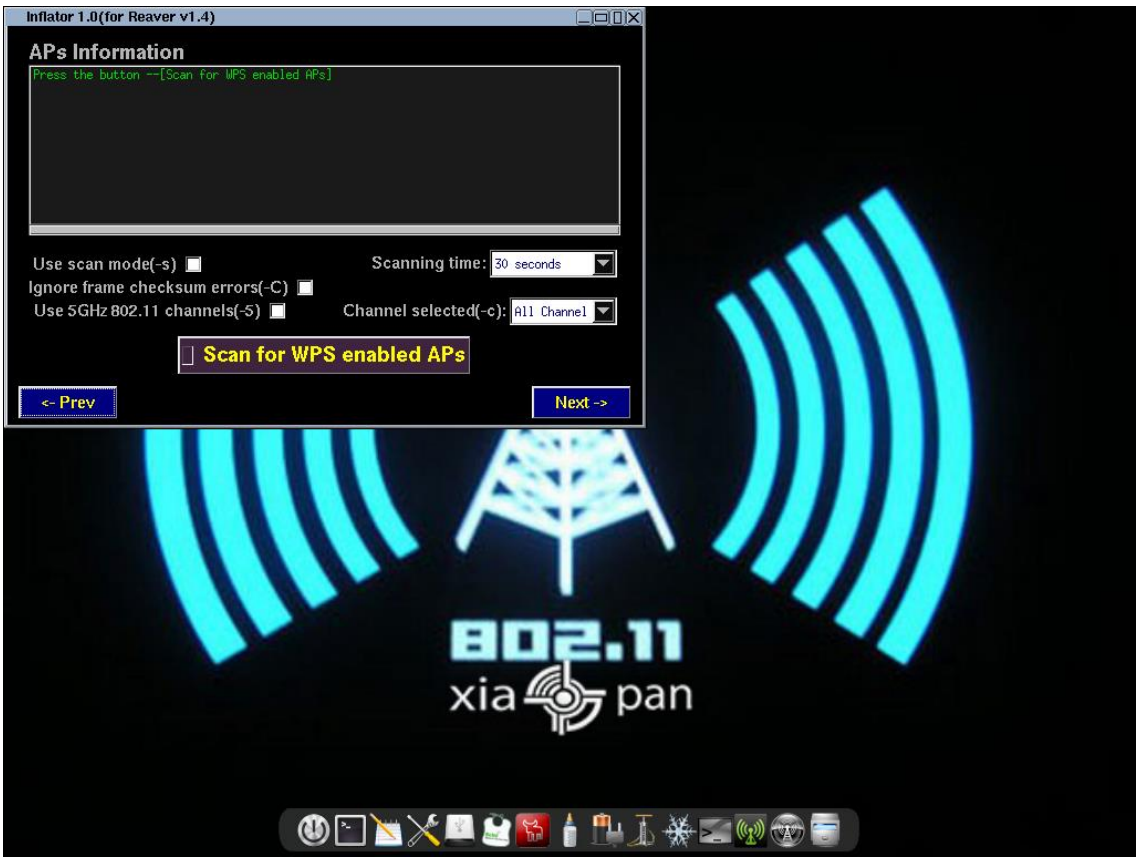
Nos aparecerá ahora una lista con las interfaces de red inalámbrica de las que disponemos. En este caso wlan0 corresponde a una tarjeta de red PCI por lo que escogeremos wlan1 que es el adaptador de red USB:



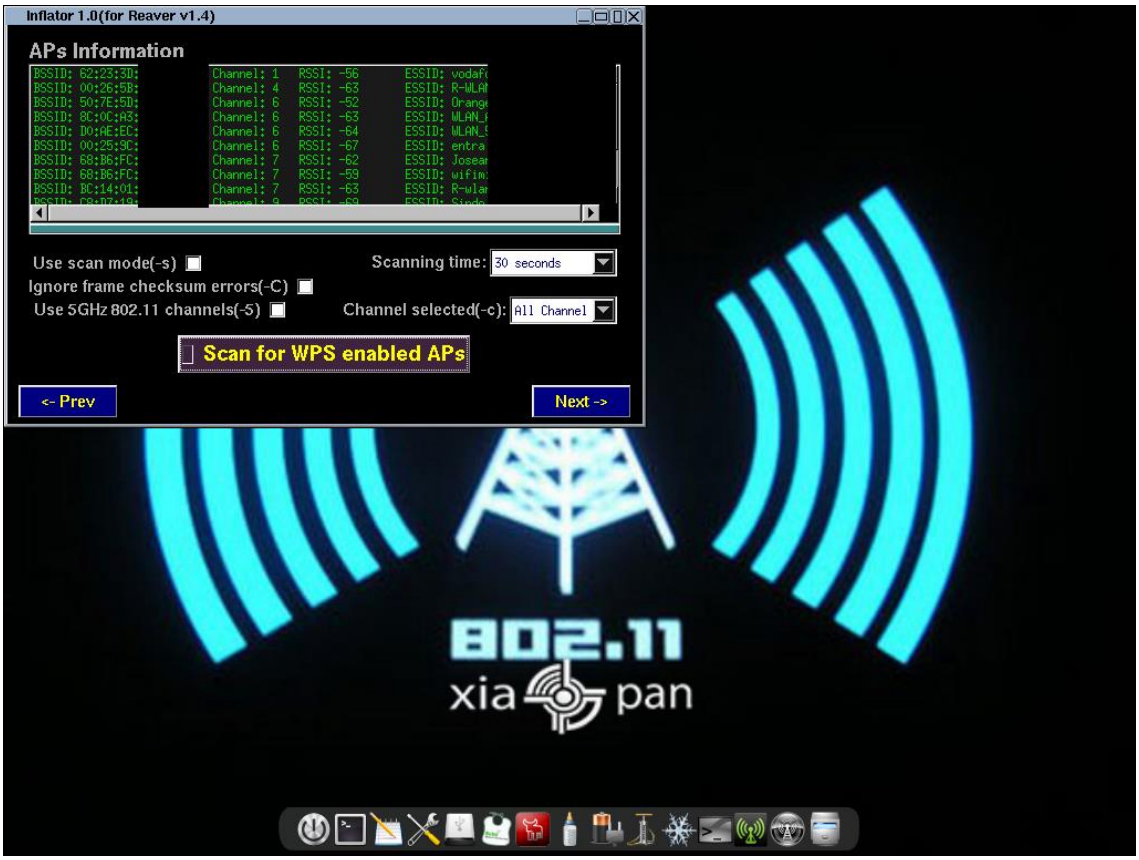
Tras seleccionarla, airmon-ng tratará de activar el modo monitor en nuestro dispositivo. Si todo va bien, aparecerá un mensaje indicando que el modo monitor está activado. Pulsaremos Next a continuación:



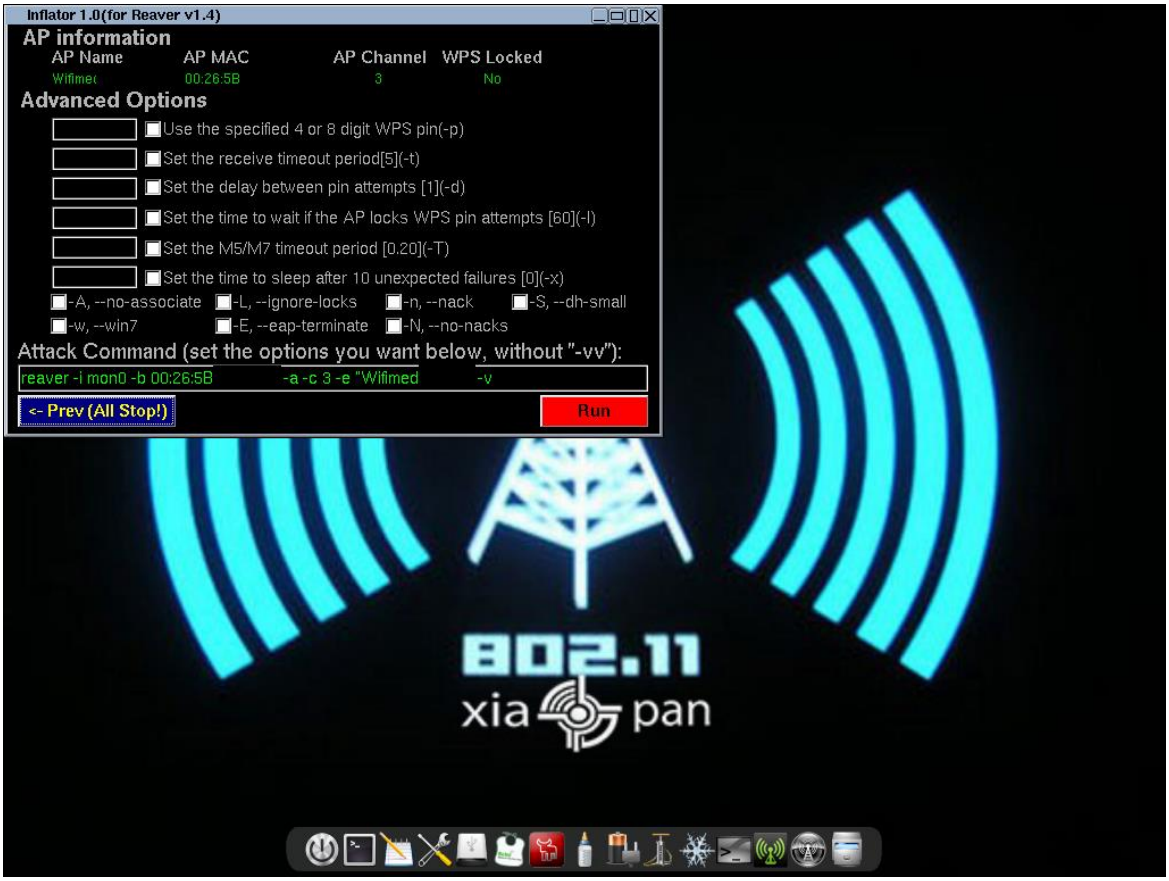
Aparecerá una ventana para escanear las redes inalámbricas que tienen el acceso por WPS activado. Hemos dejado los parámetros de configuración por defecto:



Tras pulsar el botón de Scan y esperar el tiempo correspondiente, nos aparecerán las redes con WPS:



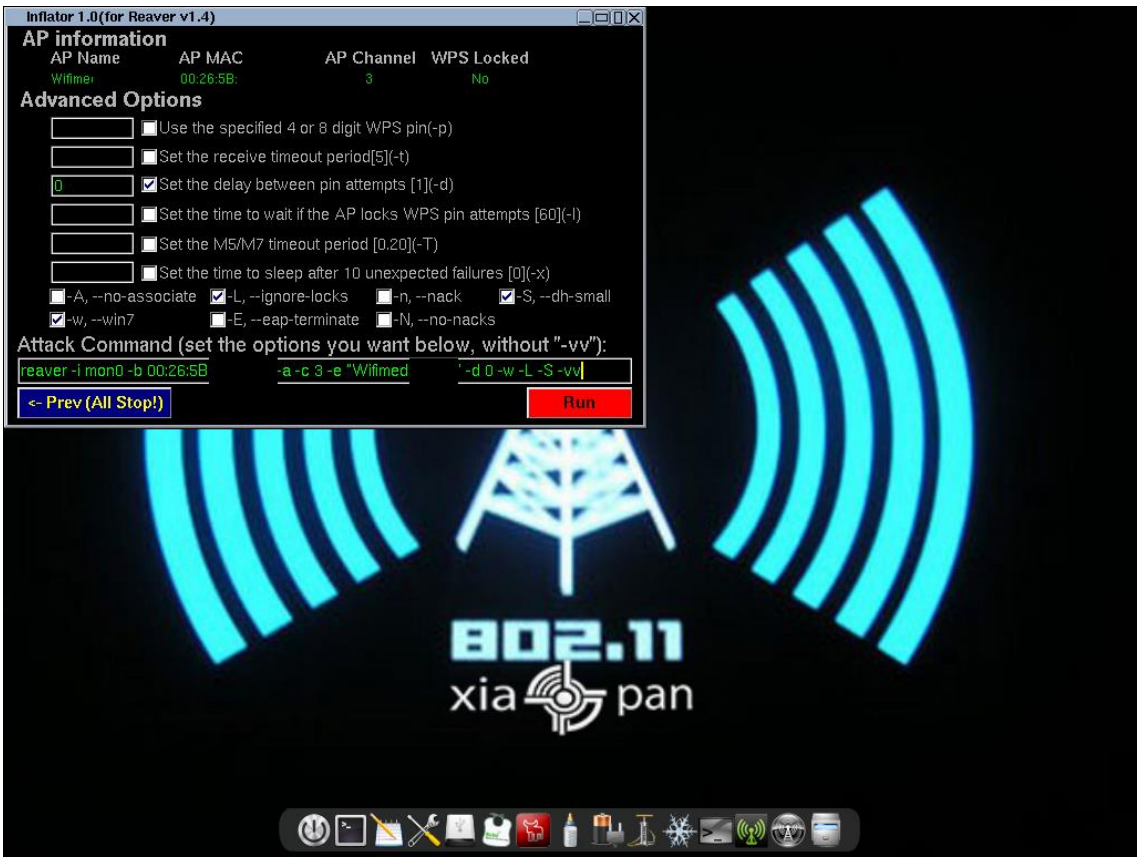
Por cuestiones prácticas, probaremos el ataque en las redes con menor RSSI (es decir, más próximas a la tarjeta de red) ya que la distancia y las interferencias pueden influir en el éxito del ataque. Escogemos una red y le damos a Next:



Aquí podemos configurar una serie de parámetros. Si se desea información sobre todos ellos podemos consultar la ayuda del programa (con reaver --help en una terminal). En esta demostración hemos usado los siguientes:

- -d 0 para que no haya delay alguno entre intentos de PIN. Puede dar problemas en algunos routers que no permitan recibir más de X solicitudes en unidad de tiempo, pero hemos optado por emplearlo para mayor rapidez
- -w simula el comportamiento de Windows 7 y se supone que puede ayudar en el ataque a algunos routers.
- -L ignora el estado de bloqueo que envíe el router. Se activa para proseguir con el ataque aún en el caso de que se reporte un bloqueo, para evitar bloqueos con falso positivo.
- -S genera claves DH más pequeñas, con lo cual ahorramos tiempo de CPU a la hora de generar dichas claves y obtenemos un ataque más rápido.
- Añadiremos manualmente una v al parámetro -v para obtener -vv (doble verbose) y poder leer más información del proceso de ataque.

Tras configurar estos parámetros (-d 0 solo se activará si primero escribimos el 0 en la casilla correspondiente y luego activamos el checkbox) pulsaremos Run y el ataque procederá a ejecutarse:



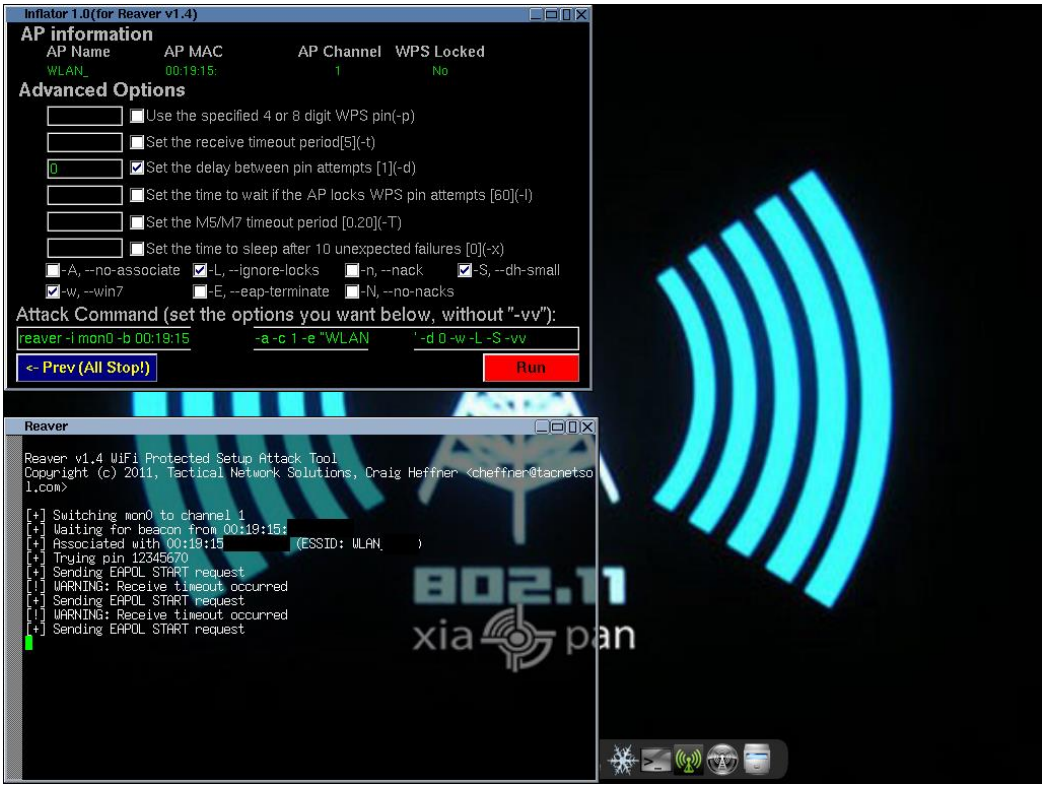
Ataque

El ataque no es perfecto debido a una serie de factores:

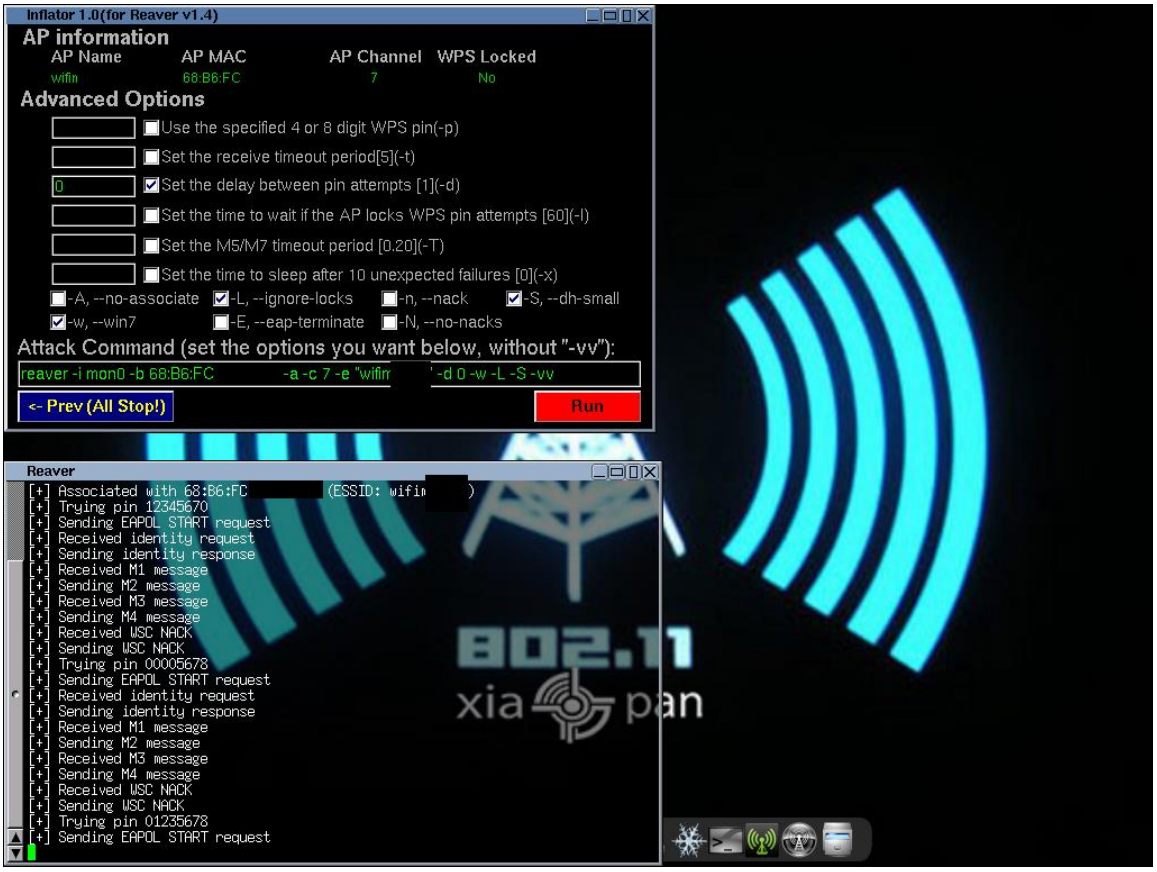
- Distancia entre el router y la tarjeta de red.
- Router con mecanismos de bloqueo cuando recibe muchas solicitudes WPS erróneas.
- Router que no soporta la carga de solicitudes y se bloquea (estaríamos haciendo un DoS).

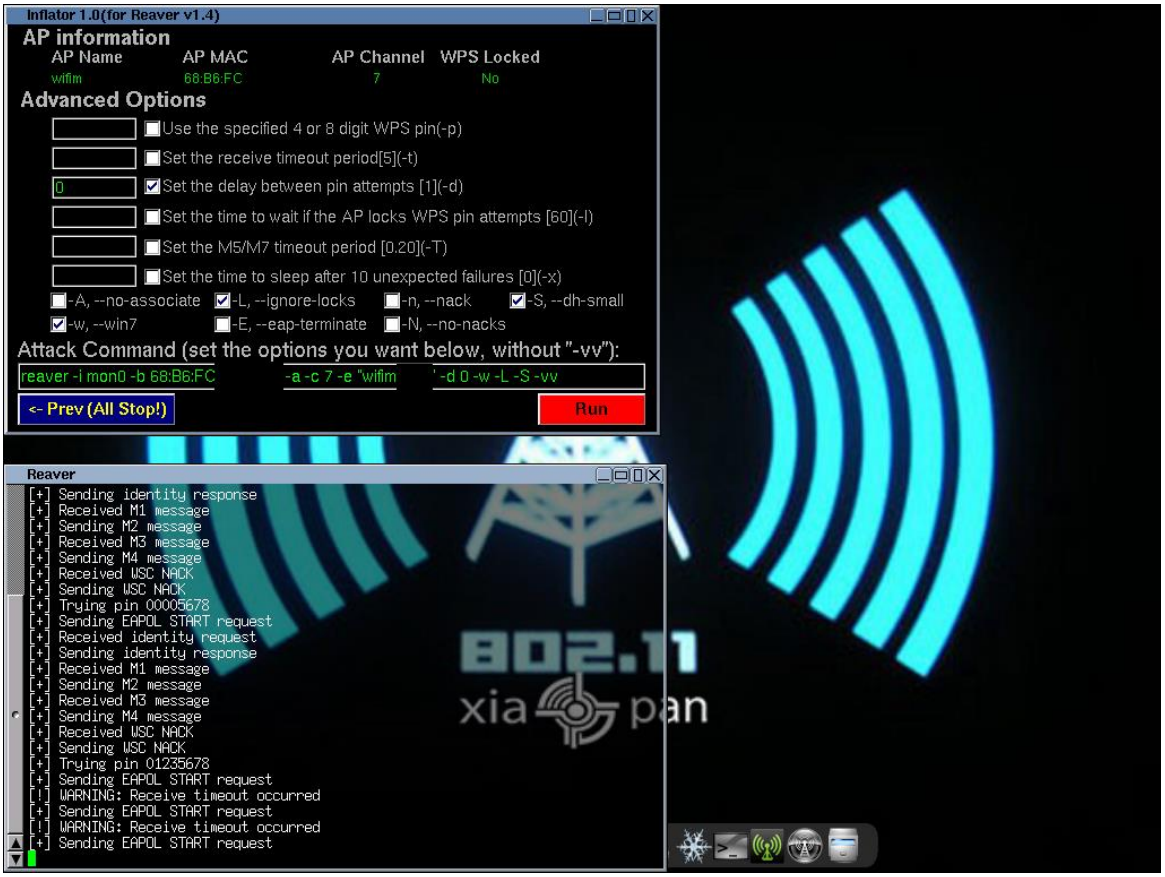
Por lo que hay que observar los resultados que obtenemos a la hora de hacer el ataque y averiguar si estamos teniendo éxito o no.

En la primera de las siguientes capturas, la tarjeta ni siquiera consigue asociarse al AP. En la segunda, se asocia pero no conseguimos establecer comunicación WPS:

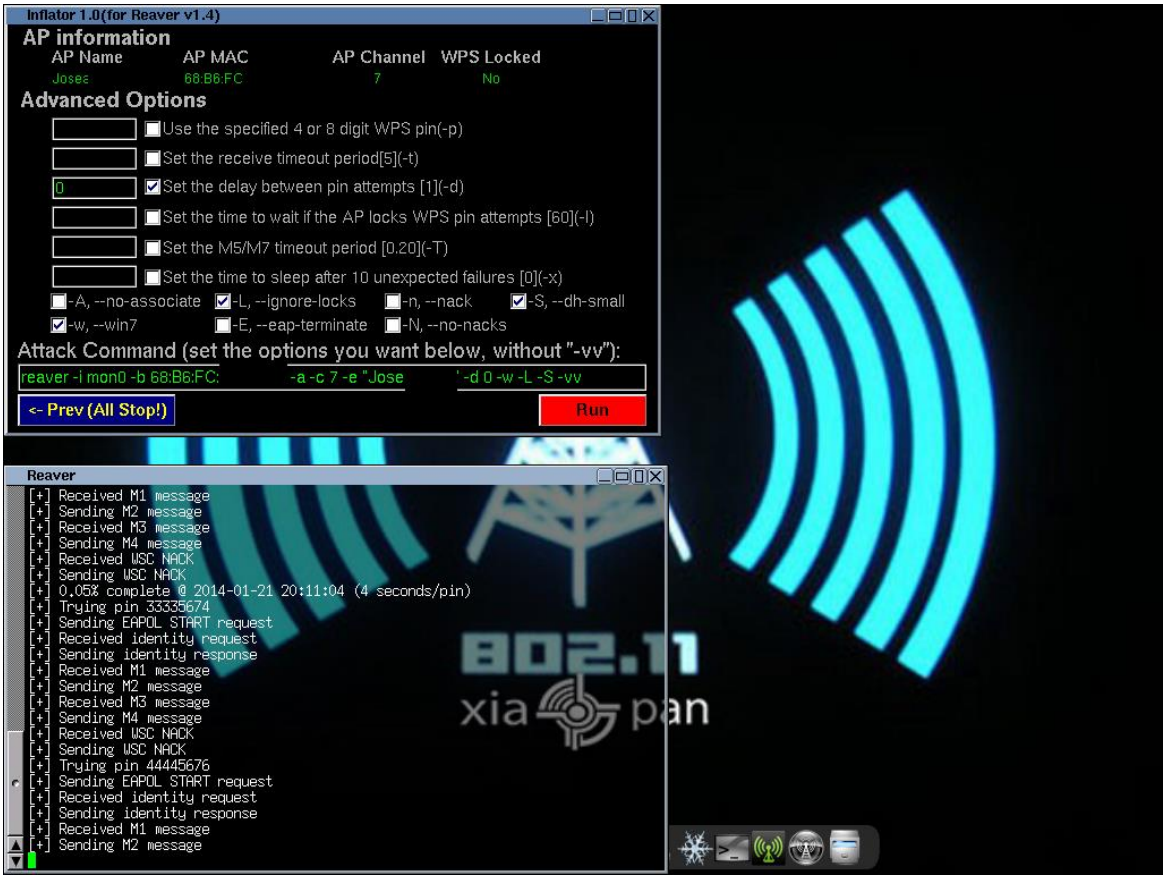


En las siguientes capturas el ataque comienza con éxito (podemos ver que empieza a comprobar varios PIN y al recibir un NACK tras enviar el mensaje M4 comprueba el siguiente) pero que llega un momento que no conseguimos conectar con el router (se ha bloqueado, posiblemente debido a la avalancha de solicitudes):





Si el ataque está teniendo éxito, veremos el PIN que se está comprobando en cada momento y de cada cierto tiempo información sobre el porcentaje de claves probadas y la velocidad media por segundo:

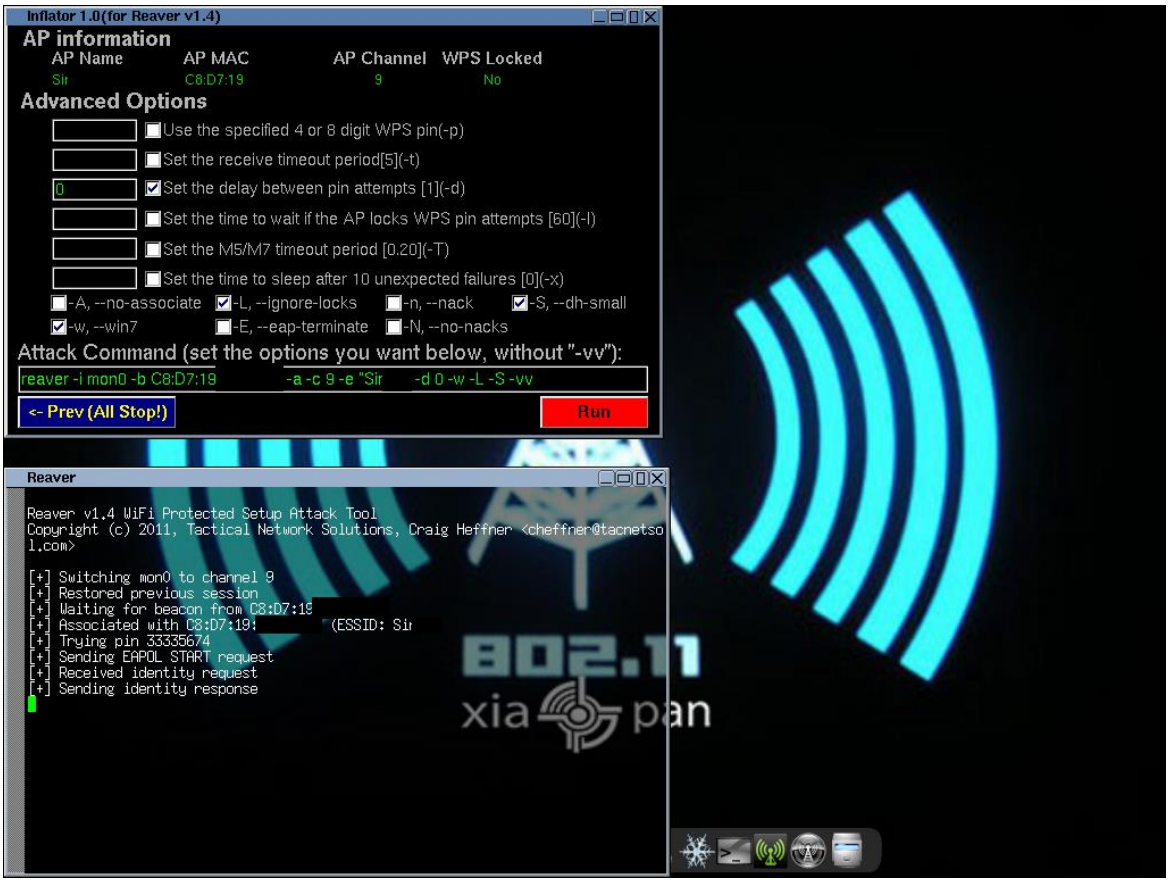


Si deseamos parar el ataque y continuarlo en otro momento, tendremos que hacer Ctrl+C en la terminal de reaver, lo cual guardará nuestro progreso en /usr/local/etc/reaver en un archivo de la forma XXXXXXXXXXXX.wpc donde las X's es la MAC del AP. Este archivo puede ser copiado y movido a otro dispositivo para ser empleado con reaver (bien sea independientemente, en otra distribución, etc), siempre que esté ubicado en la carpeta correspondiente:

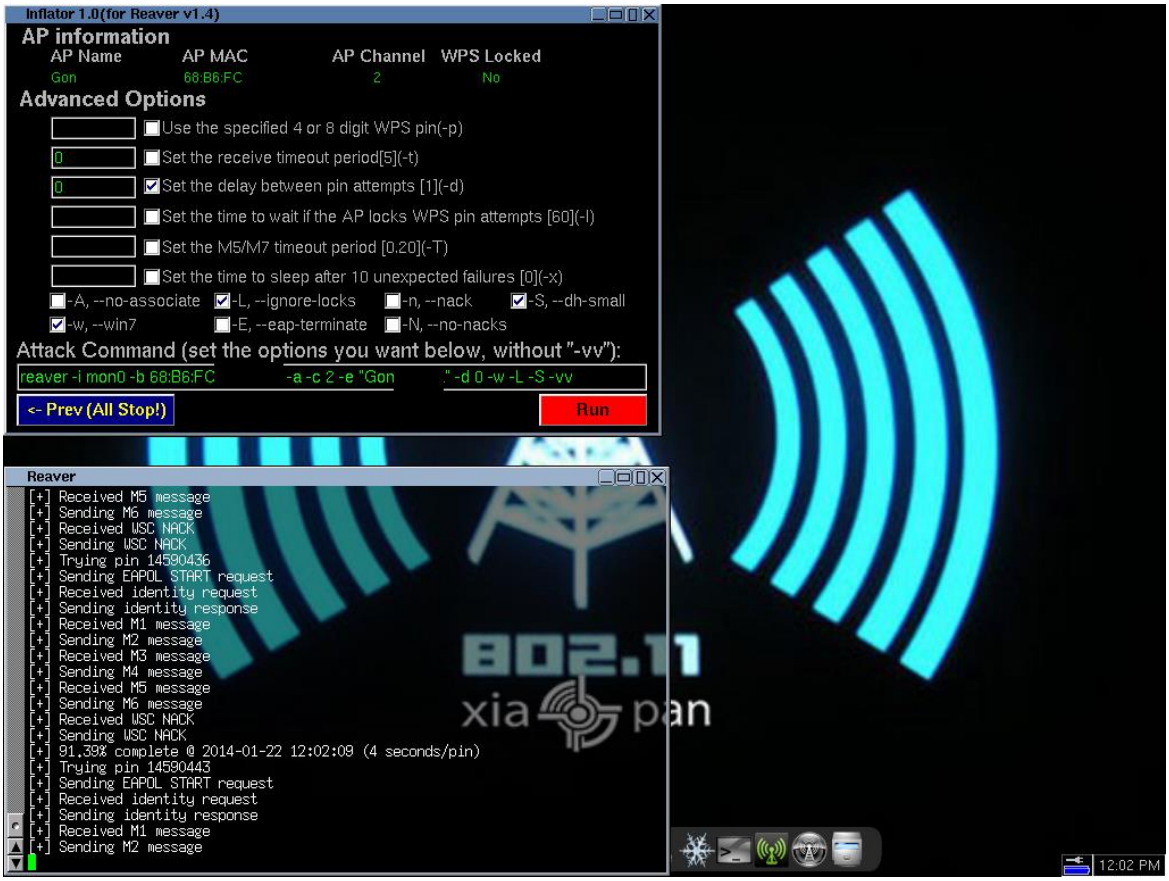
Este archivo de guardado se compone de una serie de números que indican la última primera mitad del PIN probada con éxito, la última segunda mitad del PIN probada con éxito y una lista ordenada de los PINs a probar (en algunas versiones de reaver esta lista se genera aleatoriamente, en nuestro caso está ordenada de menor a mayor con salvedad de algunos patrones de PIN muy comunes).

Si deseamos restaurar la sesión, Inflator se encarga de buscar sesiones antiguas en la carpeta de sesiones anteriormente mencionada, y si es así, restaurarlas al hacer click en Run. Sin embargo, en otras versiones de reaver, si se desea restaurar el progreso de un ataque anterior, debe ser indicado el archivo .wpc manualmente con el parámetro --session.

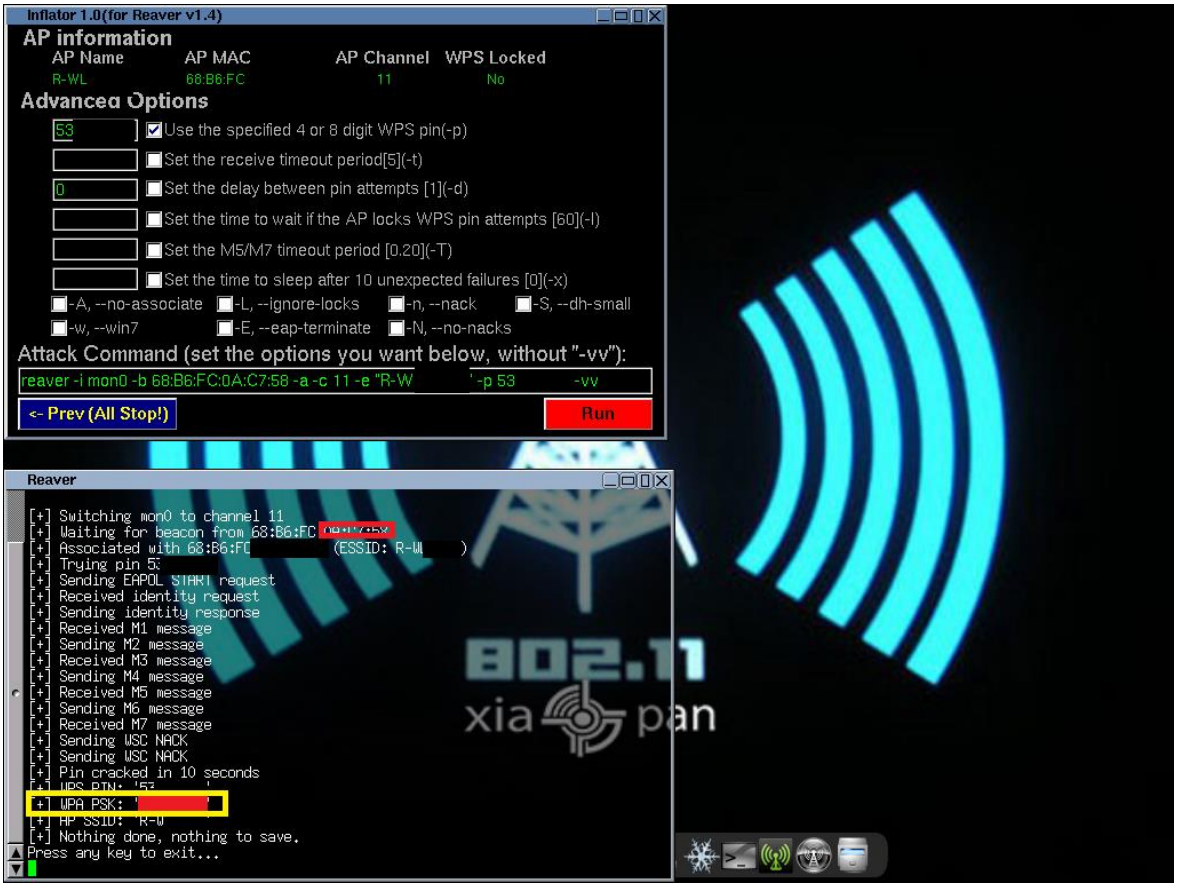
Como podemos comprobar en la siguiente imagen, el ataque ha sido restaurado de una sesión anterior por lo que continuamos comprobando los PINs a partir del 3333 en la primera mitad:



Si se ha hallado la primera mitad del PIN, comenzaremos a recibir mensajes M5 y procederemos a comprobar la segunda mitad, con lo cual el porcentaje de progreso subirá al 90.9% (ya que solamente quedan 1000 claves por probar de las 11000 originales):



El ataque continuaría así hasta que recibamos un mensaje M7, con lo que ya habremos averiguado el PIN y podremos obtener la clave WPA. Con propósitos ilustrativos hemos aprovechado el parámetro -p que nos permite introducir un PIN de 4 u 8 cifras (para comprobar la primera mitad o ambas) y seguir el ataque desde ese punto. En este caso, hemos aprovechado el conocimiento del PIN de una red para demostrar qué pasaría en el momento de recibir un mensaje M7: obtención del PIN, la contraseña WPA en texto plano y el éxito del ataque:



FUENTES

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

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

http://briolidz.wordpress.com/2012/01/10/wi-fi-protected-setup-wps/

http://ftp.netbsd.org/pub/NetBSD/NetBSD-release-6/src/external/bsd/wpa/dist/src/wps/wps_common.c

https://code.google.com/p/reaver-wps/source/browse/trunk/src/wpscrack.c

http://xiaopan.co/about/

http://xiaopan.co/run-from-usb/