http://www.matarowireless.net

(In)seguridad en redes 802.11b

Pau Oliva Fora <pof@eslack.org>

V1.3.1 - Marzo 2003

Última versión del documento y la presentación que lo acompaña:
http://pof.eslack.org/wireless/

Nota del Autor

Este documento pretende reunir unos conceptos básicos sobre seguridad o inseguridad en redes wireless, según se mire.

La idea de escribir este documento surgió gracias al profesor Pere Barberán de la *Universidad Politécnica de Mataró*, que aceptó mi propuesta sobre este tema como trabajo para la asignatura de *Ingeniería de Redes* de tercer curso de Ingeniería Técnica de Telecomunicaciones, especialidad en Telemática.

La presentación que acompaña al documento, ha sido realizada en base a éste para la conferencia que realicé en el *Primer Seminario de Wireless en la EUPMT* organizado por la comunidad *Mataró Wireless*, de la cual formo parte.

Desde aquí me gustaría agradecer la ayuda prestada por todos los miembros de las comunidades *Mataró Wireless* y *Vinaròs Wireless*, a Fatuo por iniciarme en el mundo del 802.11b, al Jefe del Departamento de Telemática de la EUPMT Leonard Janer que presentó la parte de soluciones en la conferencia sobre seguridad del seminario, y a Laia Albaladejo, sin la cual todo esto no habría sido posible.

Para cualquier comentario sobre el documento, se puede contactar conmigo vía e-mail en la dirección <pof@eslack.org>.

Índice

1. 802.11b en breve
Conceptos básicos para entender el resto del documento.
1.1 Topologías:
1.1.1 Modo Ad-Hoc
1.1.2 Modo Infraestructura
1.2 ESSID
1.3 BEACON FRAMES
1.4 WEP
1.5 Medidas de seguridad

2. Como funciona WEP
Conceptos básicos para entender las vulnerabilidades
2.1 Llaves
2.2 Encriptación
2.3 Desencriptación

3. Proceso de conexión a una WLAN
Conceptos básicos para entender las vulnerabilidades
3.1 Mecanismos de autenticación
3.1.1 Open System Autentication
3.1.2 Shared Key Autentication

4. Vulnerabilidades
4.1 Deficiencias en la encriptación WEP
4.2.1 Características lineares de CRC32
4.2.2 MIC Independiente de la llave
4.2.3 Tamaño de IV demasiado corto
4.2.4 Reutilización de IV
4.2 Deficiencias en el método de autenticación Shared Key

5. Ataques
5.1 Ataques al WEP
5.1.1 Ataque de fuerza bruta
5.1.2 Ataque inductivo Arbaugh
5.1.3 Debilidades en el algoritmo key Scheduling de RC4
5.2 Ataques a redes wireless
5.2.1 Romper ACL's basados en MAC
5.2.2 Ataque de Denegación de Servicio (DoS)
5.2.3 Descubrir ESSID ocultados
5.2.4 Ataque Man in the middle
5.2.5 Ataque ARP Poisoning

6. Conclusiones

7. Bibliografía

Apéndice A: Tarjetas wi-fi: modos y chipsets
Apéndice B: Encontrar redes Wireless

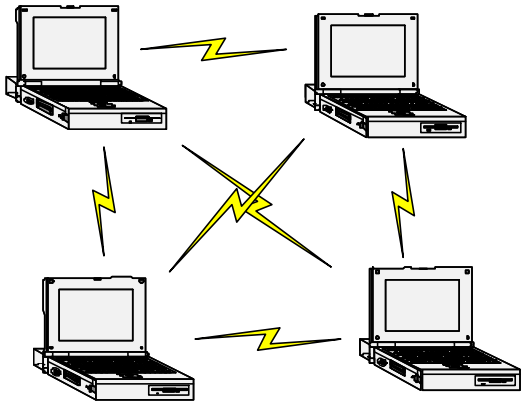
1. 802.11b en breve

Conceptos básicos para entender el resto del documento

1.1 Un vistazo a las distintas topologías que puede adoptar una red Wireless

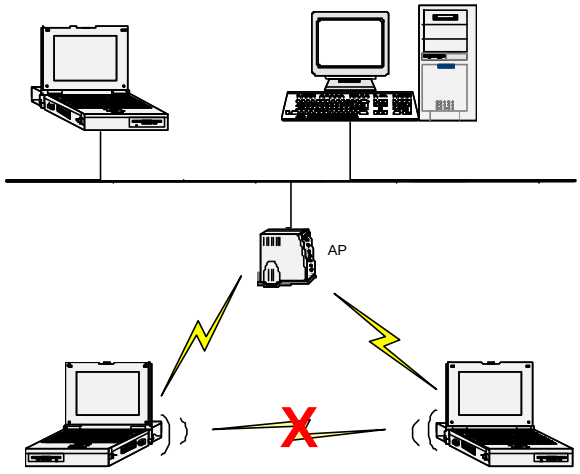
1.1.1 Modo Ad-Hoc

Esta topología se caracteriza por que no hay Punto de Acceso (AP), las estaciones se comunican directamente entre si (peer-to-peer), de esta manera el área de cobertura está limitada por el alcance de cada estación individual.



1.1.2 Modo Infraestructura

Como mínimo se dispone de un Punto de Acceso (AP), las estaciones wireless no se pueden comunicar directamente, todos los datos deben pasar a través del AP. Todas las estaciones deben ser capaces de “ver” al AP.



La mayoría de las redes wireless que podemos encontrar en las empresas utilizan modo infraestructura con uno o más Puntos de Acceso. El AP actúa como un HUB en una LAN, redistribuye los datos hacia todas las estaciones.

1.2 ESSID

Cada red wireless tiene un ESSID (Extended Service Set Identifier), que la identifica.
El ESSID consta de cómo máximo 32 caracteres y es *case-sensitive*. Es necesario conocer el ESSID del AP para poder formar parte de la red wireless, es decir, el ESSID configurado en el dispositivo móvil tiene que concordar con el ESSID del AP.

1.3 BEACON FRAMES

Los Puntos de Acceso mandan constantemente anuncios de la red, para que los clientes móviles puedan detectar su presencia y conectarse a la red wireless. Estos "anuncios" son conocidos como BEACON FRAMES, si esnifamos las tramas de una red wireless podremos ver que normalmente el AP manda el ESSID de la red en los BEACON FRAMES, aunque esto se puede deshabilitar por software en la mayoría de los AP que se comercializan actualmente.

1.4 WEP

Las redes Wireless (WLANs) son de por sí más inseguras que las redes con cables, ya que el medio físico utilizado para la transmisión de datos son las ondas electromagnéticas. Para proteger los datos que se envían a través de las WLANs, el estándar 802.11b define el uso del protocolo WEP (Wired Equivalent Privacy). WEP intenta proveer de la seguridad de una red con cables a una red Wireless, encriptando los datos que viajan sobre las ondas radioeléctricas en las dos capas más bajas del modelo OSI (capa física y capa de enlace).

El protocolo WEP está basado en el algoritmo de encriptación RC4, y utiliza claves de 64bits o de 128bits. En realidad son de 40 y 104 bits, ya que los otros 24 bits van en el paquete como Vector de Inicialización (IV). Se utiliza un checksum para prevenir que se inyecten paquetes spoofeados. Más adelante veremos más a fondo como funciona la encriptación WEP.

1.5 Medidas de seguridad

Las medidas de seguridad más comúnmente utilizadas en las redes wireless son las siguientes:

- ACL's basadas en MAC's: sólo permitir la comunicación con el AP a las direcciones MAC que el AP conoce (hay que añadir las MACs de los clientes que pueden tener acceso a la WLAN a mano en el AP).

- No emitir Beacon Frames (o emitirlos sin el ESSID)

- Utilizar WEP para cifrar los datos.

2. Como funciona WEP

Conceptos básicos para entender las vulnerabilidades

WEP utiliza el algoritmo RC4 para la encriptación con llaves de 64 bits, aunque existe también la posibilidad de utilizar llaves de 128 bits. Veremos que en realidad son 40 y 104 bits, ya que los otros 24 van en el paquete como Vector de Inicialización (IV).

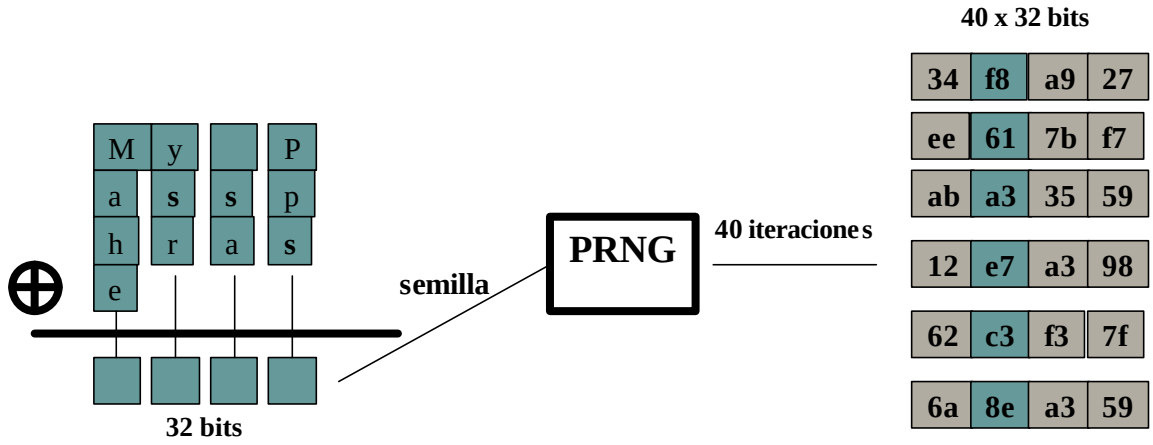
2.1 Llaves

La llave de 40 ó 104 bits, se genera a partir de una clave (passphrase) estática de forma automática, aunque existe software que permite introducir esta llave manualmente.

La clave o passphrase debe ser conocida por todos los clientes que quieran conectarse a la red wireless que utiliza WEP, esto implica que muchas veces se utilice una clave fácil de recordar y que no se cambie de forma frecuente.

A partir de la clave o passphrase se generan 4 llaves de 40 bits, sólo una de ellas se utilizará para la encriptación WEP.

Este es el proceso que se realiza para generar las llaves:



Se hace una operación XOR con la cadena ASCII (*My Passphrase*) que queda transformada en una semilla de 32 bits que utilizará el generador de números pseudo-aleatorios (PRNG) para generar 40 cadenas de 32 bits cada una.

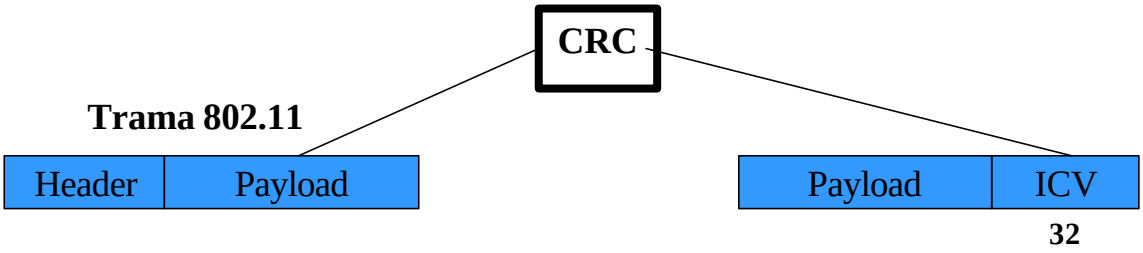
Se toma un bit de cada una de las 40 cadenas generadas por el PRNG para construir una llave y se generan 4 llaves de 40 bits.

De estas 4 llaves sólo se utilizará una para realizar la encriptación WEP como veremos a continuación.

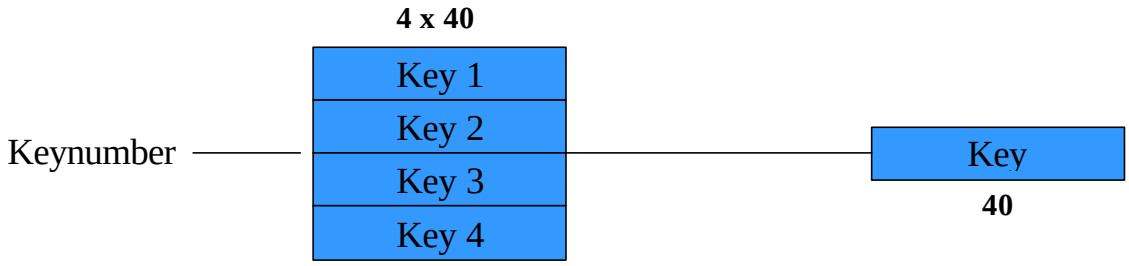
2.2 Encriptación

Para generar una trama encriptada con WEP se sigue el siguiente proceso:

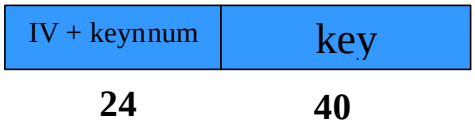
Partimos de la trama que se quiere enviar. Esta trama sin cifrar está compuesta por una cabecera (Header) y contiene unos datos (Payload). El primer paso es calcular el CRC de 32 bits del payload de la trama que se quiere enviar. El CRC es un algoritmo que genera un identificador único del payload en concreto, que nos servirá para verificar que el payload recibido es el mismo que el enviado, ya que el resultado del CRC será el mismo. Añadimos este CRC a la trama como **valor de chequeo de integridad** (ICV: Integrity Check Value):



Por otra parte seleccionamos una llave de 40 bits, de las 4 llaves posibles:



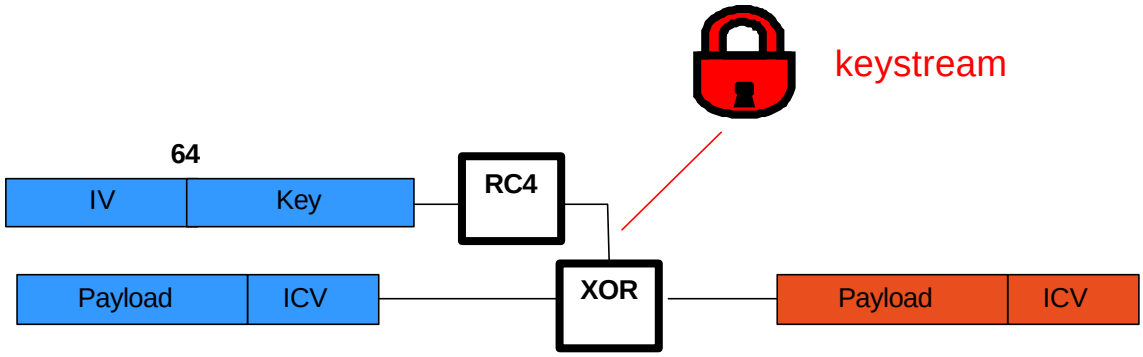
Y añadimos el **Vector de Inicialización** (IV) de 24 bits al principio de la llave seleccionada:



El IV es simplemente un contador que suele ir cambiando de valor a medida que vamos generando tramas, aunque según el estándar 802.11b también puede ser siempre cero. Con el IV de 24 bits y la llave de 40 conseguimos los 64 bits de llave total que utilizaremos para encriptar la trama. En el caso de utilizar encriptación de 128 bits tendríamos 24 bits de IV y 104 de llave.

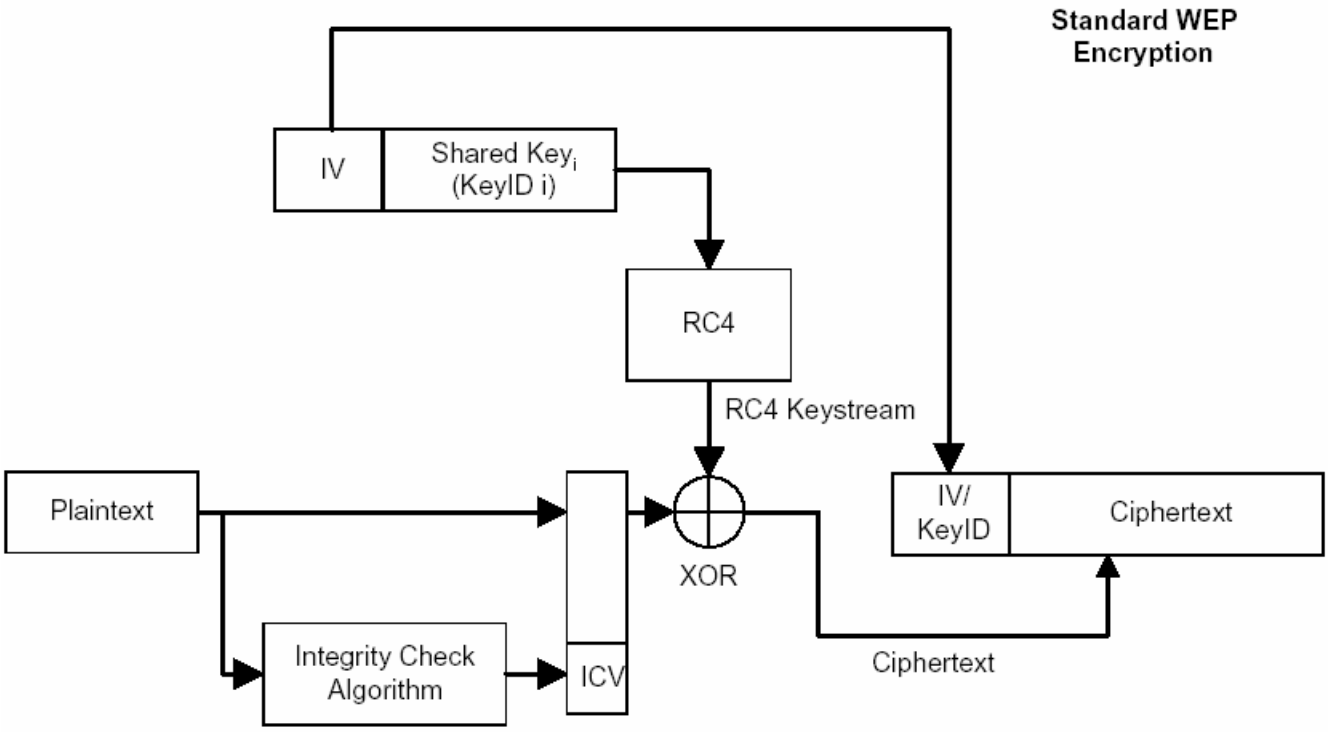
Llegado a este punto, aplicamos el algoritmo RC4 al conjunto IV+Key y conseguiremos el keystream o flujo de llave. Realizando una operación XOR con este keystream y el conjunto Payload+ICV obtendremos el Payload+ICV cifrado, este proceso puede verse en el siguiente grafico.

Se utiliza el IV y la llave para encriptar el Payload + ICV:



Después añadimos la cabecera y el IV+Keynumber sin cifrar. Así queda la trama definitiva lista para ser enviada:

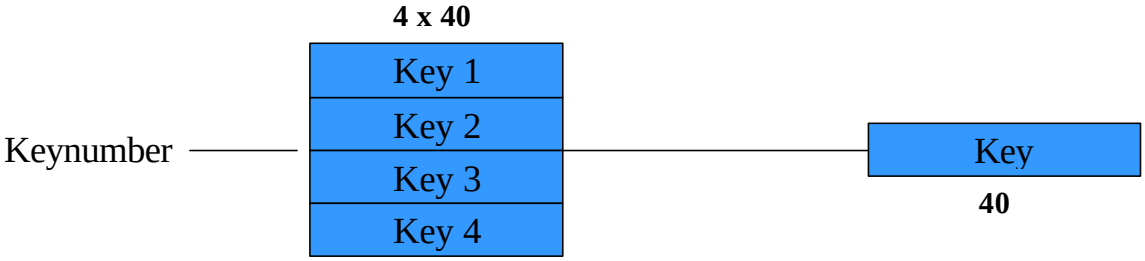
El proceso de encriptación en conjunto se ve resumido en este esquema:



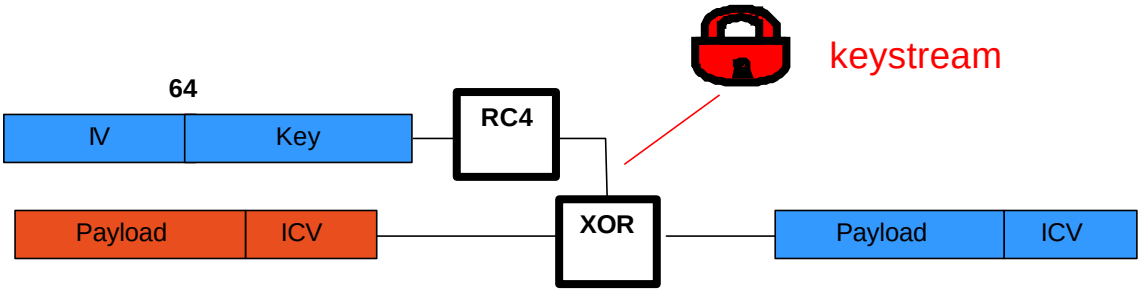
2.3 Desencriptación

Ahora vamos a ver el proceso que se realiza para desencriptar una trama encriptada con WEP:

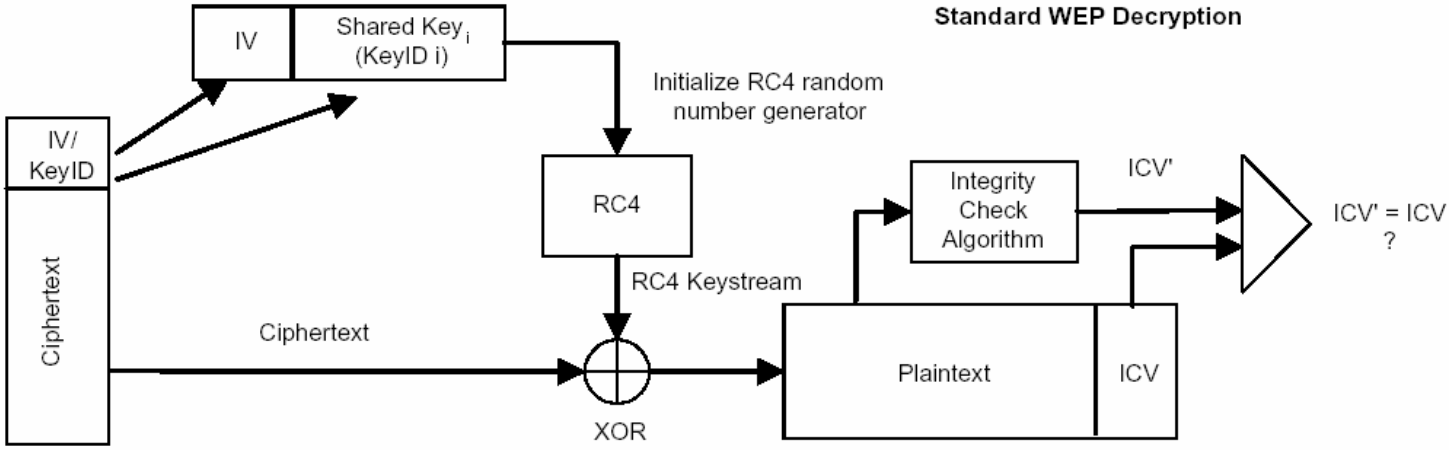
Se utiliza el número de llave que aparece en claro en la trama cifrada junto con el IV para seleccionar la llave que se ha utilizado para cifrar la trama:



Se añade el IV al principio de la llave seleccionada, consiguiendo así los 64 bits de llave. Aplicando RC4 a esta llave obtenemos el keystream válido para obtener la trama en claro (plaintext) realizando una XOR con el Payload+ICV cifrados y la llave completa como se describe a continuación.

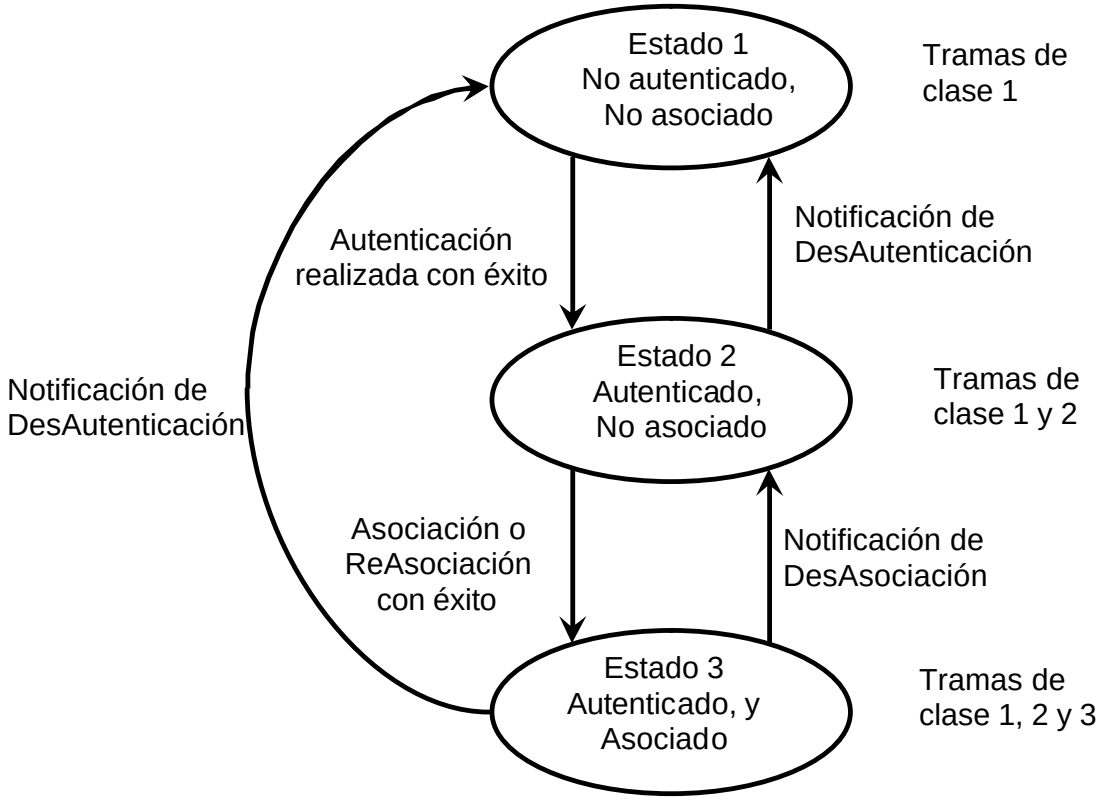


Una vez obtenido el plaintext, se vuelve a calcular el ICV del payload obtenido y se compara con el original. El proceso completo puede verse en el siguiente esquema:



3. Conexión a una WLAN

El siguiente diagrama de estados muestra los pasos que debe realizar un cliente para asociarse con un AP:



El proceso de asociación tiene dos pasos, envueltos en 3 estados:

1. No autenticado y no asociado
2. Autenticado y no asociado
3. Autenticado y asociado

En la transición por los diferentes estados, ambas partes (cliente y AP) intercambian mensajes llamados “management frames”.

El proceso que realiza un cliente wireless para encontrar y asociarse con un AP es el siguiente:

Los AP transmiten BEACON FRAMES cada cierto intervalo de tiempo fijo. Para asociarse con un AP y unirse a una red en modo infraestructura, un cliente escucha en busca de BEACON FRAMES para identificar Puntos de Acceso. El cliente también puede enviar una trama “PROVE REQUEST” que contenga un ESSID determinado para ver si le responde un AP que tenga el mismo ESSID.

Después de identificar al AP, el cliente y el AP realizan autenticación mutua intercambiando varios *management frames* como parte del proceso. Hay varios mecanismos de autenticación posibles que veremos con más detalle un poco más adelante.

Después de una autenticación realizada con éxito, el cliente pasa a estar en el segundo estado (autenticado y no asociado). Para llegar al tercer estado (autenticado y asociado) el cliente debe mandar una trama "ASSOCIATION REQUEST" y el AP debe contestar con una trama "ASSOCIATION RESPONSE", entonces el cliente se convierte en un host más de la red wireless y ya está listo para enviar y recibir datos de la red.

3.1 Mecanismos de autenticación

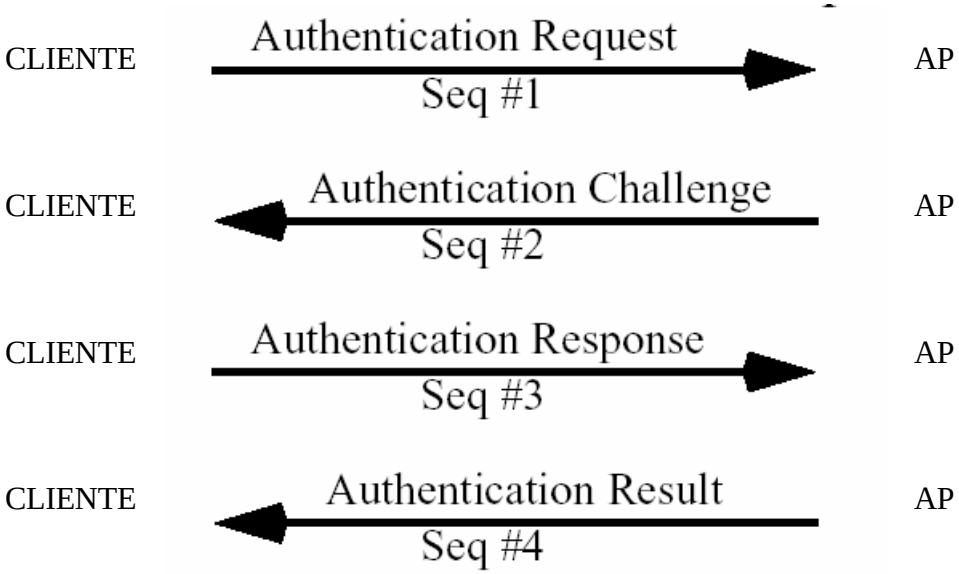
3.1.1 Open System Authentication

Open system authentication es el protocolo de autenticación por defecto para 802.11b. Como su nombre indica, este método autentica a cualquier cliente que pide ser autenticado. Es un proceso de autenticación NULO, las tramas se mandan en texto plano aunque esté activado el cifrado WEP.

3.1.2 Shared Key Authentication

Este mecanismo utiliza una clave secreta compartida, que conocen cliente y AP.

El siguiente esquema muestra el proceso de autenticación descrito a continuación:



La estación que quiere autenticarse (cliente), envía una trama AUTHENTICATION REQUEST indicando que quiere utilizar una "clave compartida". El destinatario (AP) contesta enviando una trama que contiene 128 octetos de texto (desafío) al cliente.

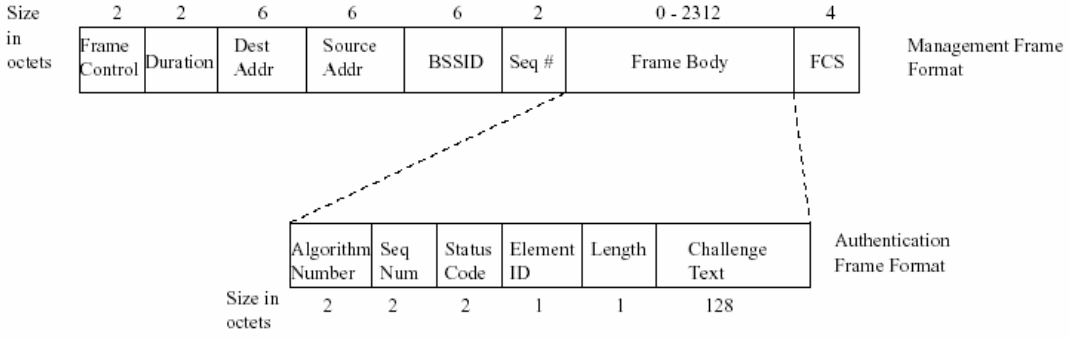
El texto del desafío se genera utilizando el PRNG (generador de números pseudo-aleatorios de WEP) con la clave compartida y un vector de inicialización (IV) aleatorio.

Una vez el cliente recibe la trama, copia el contenido del texto de desafío en el payload de una nueva trama, que encripta con WEP utilizando la clave compartida (*passphrase*) y añade un nuevo IV (elegido por el cliente). Una vez construida esta nueva trama encriptada, el cliente la envía al AP, y éste desencripta la trama recibida y comprueba que:

- El ICV (Integrity Check Value) sea valido (CRC de 32 bits).
- El texto de desafío concuerde con el enviado en el primer mensaje.

Si la comprobación es correcta, se produce la autenticación del cliente con el AP y entonces se vuelve a repetir el proceso pero esta vez el primero que manda la trama con el AUTHENTICATION REQUEST es el AP. De esta manera se asegura una autenticación mutua.

En la siguiente figura se muestra el formato de una trama de autenticación. Este formato es utilizado para todos los mensajes de autenticación:



Si el campo "Status Code" tiene valor '0' indica que la autenticación ha sido realizada con éxito, si no contiene un código de error.
El campo "Element identifier" indica que la trama contiene el texto de desafío.
El campo "Length" indica la longitud del texto de desafío y está fijado a 128 bits.
El campo "Challenge text" incluye el texto de desafío aleatorio.

La siguiente tabla muestra los posibles valores de los campos y cuando está presente el texto de desafío, según el número de secuencia (Seq #) del mensaje:

Sequence Number	Status Code	Challenge Text	Se usa WEP
1	Reservado	No presente	No
2	Status	Presente	No
3	Reservado	Presente	Si
4	Status	No presente	No

4. Vulnerabilidades

4.1 Deficiencias en la encriptación WEP

4.1.1 Características lineares de CRC32

Esta vulnerabilidad fue demostrada teóricamente por Nikita Borisov, Ian Goldberg y David Wagner (Universidad de Berkeley).

Como hemos visto anteriormente, el campo ICV (Integrity Check Value) de una trama encriptada con WEP contiene un valor utilizado para verificar la integridad del mensaje. Esto provee de un mecanismo de autenticación de mensajes a WEP, por lo tanto el receptor aceptará el mensaje si el ICV es válido. El ICV se genera simplemente haciendo un CRC (Cyclic Redundancy Check) de 32 bits, del payload de la trama. Este mecanismo tiene dos graves problemas:

- Los CRCs son independientes de la llave utilizada y del IV (veremos este problema más a fondo en el apartado 4.1.2)
- Los CRCs son lineares: $CRC(m \oplus k) = CRC(m) \oplus CRC(k)$

Debido a que los CRCs son lineares, se puede generar un ICV valido ya que el CRC se combina con una operación XOR que también es linear y esto permite hacer el *'bit flipping*' como veremos a continuación:

- Un atacante debe interceptar un mensaje m (conocido o no) y modificarlo de forma conocida para producir m':

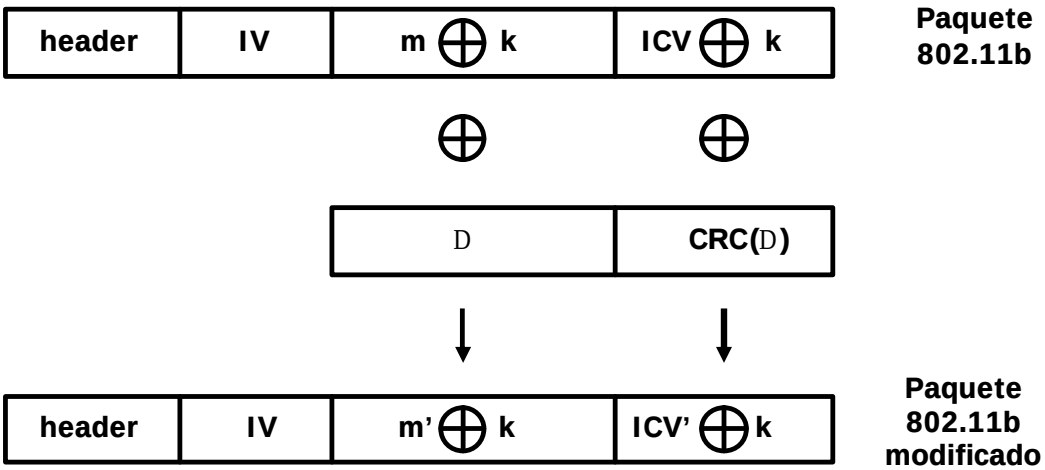
$$\mathbf{m' = m} \text{ Å } D$$

- Como el CRC-32 es linear, puede generar un nuevo ICV' a partir del ICV de m:

$$\mathbf{IC' = IC} \text{ Å } \mathbf{h(}D\mathbf{)}$$

- ICV' será valido para el nuevo cyphertext c'

$$\mathbf{c' = c} \text{ Å } D = \mathbf{k} \text{ Å } \mathbf{(m} \text{ Å } D\mathbf{) = k} \text{ Å } \mathbf{m'}$$



4.1.2 MIC Independiente de la llave

Esta vulnerabilidad fue demostrada teóricamente por David Wagner (Universidad de Berkeley).

Esta vulnerabilidad en WEP es conocida en inglés como “Lack of keyed MIC”: Ausencia de mecanismo de chequeo de integridad del mensaje (MIC) dependiente de la llave.

- El MIC que utiliza WEP es un simple CRC-32 calculado a partir del payload, por lo tanto no depende de la llave ni del IV.

Esta debilidad en la encriptación da lugar a que conocido el plaintext de un solo paquete encriptado con WEP sea posible inyectar paquetes a la red.

Esto es posible de la siguiente manera:

- El atacante captura un paquete **c = m** Å **k** donde m es conocido
  (por ejemplo, el atacante envía un e-mail a la victima)

- El atacante recupera el flujo pseudo-aleatorio **k = c** Å **m** para el IV concreto del paquete

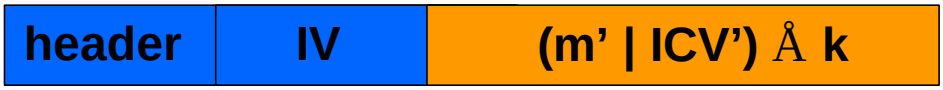
- Supongamos que el atacante quiere inyectar un mensaje m’, debe realizar lo siguiente:

  ICV’ = CRC32(m’)

  - El atacante ya puede ensamblar la parte encriptada del paquete:

    c = (m’|ICV’) Å **k**

- El atacante obtiene un paquete válido y listo para ser inyectado a la red:



4.1.3 Tamaño de IV demasiado corto

Otra de las deficiencias del protocolo viene dada por la corta longitud del campo IV en las tramas 802.11b. El vector de inicialización (IV) tiene sólo 24 bits de longitud y aparece en claro (sin encriptar).
Matemáticamente sólo hay 2^24 (16.777.216) posibles valores de IV. Aunque esto pueda parecer mucho, 16 millones de paquetes pueden generarse en pocas horas en una red wireless con tráfico intenso:

Un punto de acceso que constantemente envíe paquetes de 1500 bytes (MTU) a 11Mbps, acabará con todo el espacio de IV disponible después de 1500*8/(11*10^6)*2^24 = ~1800 segundos, o 5 horas. Este tiempo puede ser incluso más pequeño si la MTU es menor que 1500.

La corta longitud del IV, hace que éste se repita frecuentemente y de lugar a la deficiencia del protocolo que veremos a continuación, basada en la posibilidad de realizar ataques estadísticos para recuperar el plaintext gracias a la reutilización del IV.

4.1.4 Reutilización de IV

Esta vulnerabilidad fue demostrada teóricamente por David Wagner (Universidad de Berkeley). Se basa en que WEP no utiliza el algoritmo RC4 "con cuidado": el Vector de Inicialización se repite frecuentemente. Se pueden hacer ataques estadísticos contra cyphertexts con el mismo IV.

Si un IV se repite, se pone en riesgo la confidencialidad

Supongamos que *P, P'* son dos plaintexts encriptados con el mismo IV.
Supongamos $Z = RC4(key, IV)$; entonces los dos ciphertexts son $C = P \oplus Z$ y $C' = P' \oplus Z$.
Nótese que $C \oplus C' = (P \oplus Z) \oplus (P' \oplus Z) = (Z \oplus Z) \oplus (P \oplus P') = P \oplus P'$ por lo que la XOR de ambos plaintexts es conocida. Si hay redundancia, se pueden descubrir ambos plaintexts. Si podemos adivinar un plaintext, el otro puede también ser descubierto estadísticamente de forma trivial, así que si RC4 no se usa con cuidado, se vuelve inseguro

WEP no usa RC4 con cuidado

¡El estándar 802.11 especifica que cambiar el IV en cada paquete es opcional!
El IV normalmente es un contador que empieza con valor cero y se va incrementando de uno en uno, por lo tanto:

- Rebotar causa la reutilización de IV's
- Sólo hay 16 millones de IV's posibles, así que después de interceptar suficientes paquetes, seguro que hay IV's repetidos

Un atacante capaz de escuchar el tráfico 802.11 puede descifrar ciphertexts interceptados incluso sin conocer la clave.

4.2 Deficiencias en el método de autenticación Shared Key

El método de autenticación *Shared Key Autentication* descrito anteriormente se puede explotar fácilmente mediante un ataque pasivo:

El atacante captura el segundo y el tercer *management messages* de una autenticación mutua (*Authentication Challenge* y *Authentication Response*). El segundo mensaje contiene el texto de desafío en claro, y el tercer mensaje contiene el desafío encriptado con la clave compartida. Como el atacante conoce el desafío aleatorio (plaintext, P), el desafío encriptado (cyphertext, C), y el IV público, el atacante puede deducir el flujo pseudo-aleatorio (keystream) producido usando WEP utilizando la siguiente ecuación:

$$WEP_{PR}^{K,IV} = C \oplus P$$

El tamaño del keystream será el tamaño de la trama de autenticación, ya que todos los elementos de la trama son conocidos: número de algoritmo, número de secuencia, status code, element id, longitud, y el texto de desafío. Además, todos los elementos excepto el texto de desafío son los mismos para TODAS las *Authentication Responses*.

El atacante tiene por lo tanto todos los elementos para autenticarse con éxito sin conocer la clave secreta compartida K. El atacante envía un *Authentication Request* al AP con el que se quiere asociar. El AP contesta con un texto de desafío en claro. El atacante entonces, coge el texto de desafio aleatorio, R, y el flujo pseudo-aleatorio WEP $^{k,IV}{}_{PR}$ y genera el cuerpo de una trama *Authentiation Response* válido, realizando una operación XOR con los dos valores. El atacante entonces debe crear un nuevo ICV valido aprovechando la vulnerabilidad de *Características lineares de CRC32*. Una vez creado el nuevo ICV, el atacante acaba de completar la trama de *Authentication Response* y la envía, de esta manera se asocia con el AP y se une a la red.

Con este proceso el atacante sólo esta autenticado, pero todavía no puede utilizar la red. Como el atacante no conoce la clave compartida, para poder utilizar la red debe implementar algún ataque al protocolo WEP.

5. Ataques

5.1 Ataques al WEP

5.1.1 Ataque de fuerza bruta

La semilla de 32 bits que utiliza el PRNG es obtenida a partir de la passphrase. La passphrase normalmente contiene caracteres ASCII, por lo cual el bit más alto de cada carácter siempre es cero. El resultado de la operación XOR de estos bits también es cero y esto provoca una reducción de la entropía de la fuente, es decir, las semillas sólo podrán ir desde 00:00:00:00 hasta 7F:7F:7F:7F en lugar de hasta FF:FF:FF:FF.

El uso del PRNG con esta semilla también reduce la entropía. De la semilla de 32 bits sólo utilizan los bits del 16 al 23. El generador es un generador lineal congruente (LGC: linear congruential generator) de módulo 2^32, esto provoca que los bits mas bajos sean "menos aleatorios" que los altos, es decir, el bit 0 tiene una longitud de ciclo de 2^1, el bit 1 de 2^2, el bit 2 de 2^3, etc. La longitud de ciclo del resultado será por tanto 2^24. Con esta longitud de ciclo sólo las semillas que vayan de 00:00:00:00 a 00:FF:FF:FF producirán llaves únicas.

Como las semillas sólo llegan hasta 7F:7F:7F:7F y la última semilla que tiene en cuenta el PRNG es 00:FF:FF:FF, sólo necesitamos considerar las semillas desde 00:00:00:00 hasta 00:7F:7F:7F por lo que la entropía total queda reducida a 21 bits.

El conocimiento de estos datos nos permite hacer ataques de fuerza bruta contra la encriptación WEP generando llaves de forma secuencial utilizando las semillas desde 00:00:00:00 hasta 00:7F:7F:7F. Utilizando este proceso, un procesador PIII a 500MHZ tardaría aproximadamente 210 días en encontrar la llave, aunque se puede usar computación en paralelo para obtener la llave en un tiempo más razonable.

También existe la posibilidad de utilizar un diccionario para generar sólo las semillas de las palabras (o frases) que aparezcan en el diccionario, con lo que si la passphrase utilizada está en el diccionario conseguiríamos reducir sustancialmente el tiempo necesario para encontrarla.

5.1.2 Ataque Inductivo Arbaugh

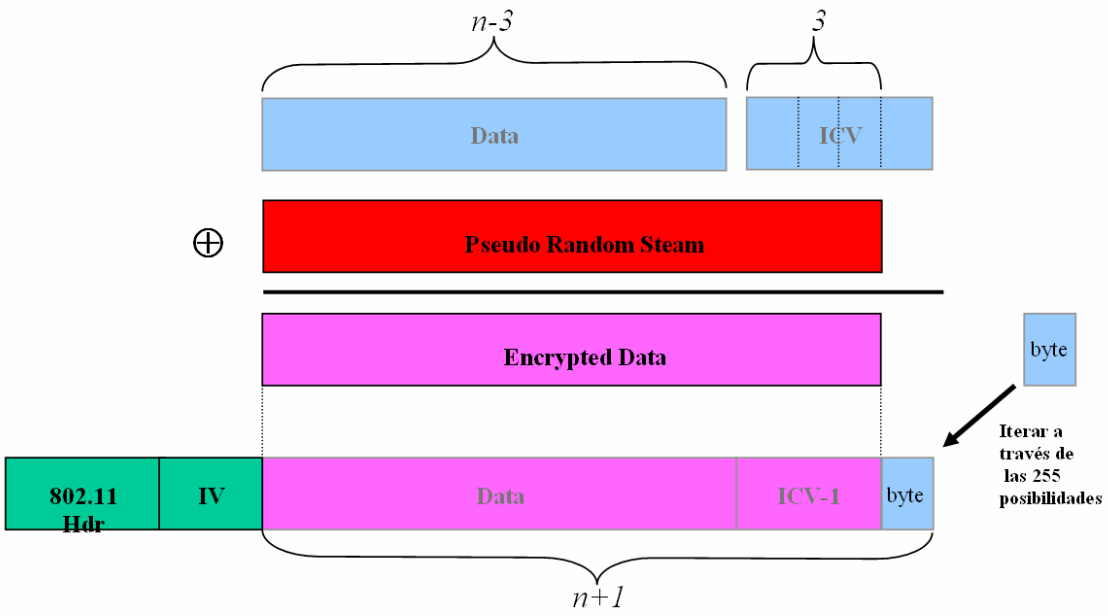
Este ataque fue demostrado teóricamente por William A. Arbaugh (Universidad de Maryland).

Se basa en explotar la vulnerabilidad de MIC independiente de la llave aprovechando también la redundancia de información producida por el CRC.

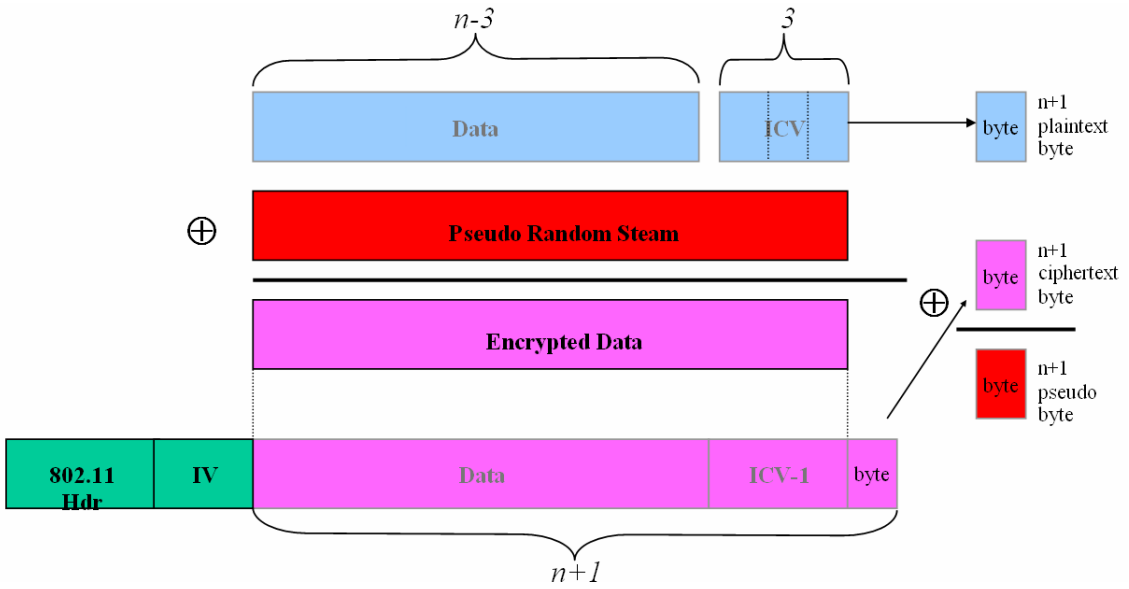
Para realizar el ataque hay que conocer parte del plaintext que viaja encriptado en una trama, que podemos obtener por ejemplo identificando mensajes "*DHCPDISCOVER*" de los que conocemos que la cabecera IP tendrá como origen 0.0.0.0 y como destino 255.255.255.255 y tienen longitud fija. Una vez identificada la trama con el mensaje "*DHCPDISCOVER*" realizamos una XOR del plaintext conocido con el cyphertext que hemos recibido, obteniendo así n (en este caso 24) bytes del keystream para el IV concreto del paquete.

Una vez tengamos estos 24 bytes conocidos del keystream hay que generar un paquete de tamaño n-3, es decir 24-3 = 21 bytes de longitud. Este paquete debe ser algo de lo que podamos esperar una respuesta, por ejemplo un ping o un ARP Request.

Calculamos el ICV del paquete generado y añadimos sólo los primeros 3 bytes del ICV que hemos calculado. Realizamos una XOR con el resto del keystream añadiendo el último byte del ICV en el byte n+1 (al final del paquete) tratando de adivinar el siguiente byte del keystream tal y como se muestra en la figura:



Una vez generado el paquete completo lo enviamos y esperamos una respuesta (echo reply, ARP reply…), si no hay respuesta tendremos que ir probando las 255 posibilidades restantes modificando el último byte (n+1). Si hay respuesta podemos afirmar que el byte n+1 era el último byte del ICV, así que tenemos un plaintext que concuerda con el cyphertext y que a su vez nos da el byte n+1 del keystream que es lo que nos interesa. Realizando este proceso repetidas veces obtendremos el keystream completo.



Asumiendo que un atacante puede realizar aproximadamente 100 pruebas por segundo, tardaría una media de 36 minutos en encontrar un keystream completo de 1500 bytes valido para un IV determinado.

Una vez tenemos un keystream entero, los $2^{24} - 1$ restantes son fáciles de obtener:

El atacante tiene que volver a generar un paquete del cual se le devuelva una respuesta, (lo mejor es enviar broadcast pings, así recibimos múltiples respuestas por cada paquete que enviamos). El atacante conoce el plaintext de la respuesta y el que responde cada vez enviará el paquete con un IV diferente, así es posible construir una tabla de keystreams completos para cada IV que el atacante puede utilizar para descifrar el tráfico encriptado con WEP en tiempo real.

El atacante necesita almacenar 1500 bytes de keystream por cada IV, por lo que la tabla ocuparía 2^{24}x1500 = 24GB y tardaría una media de 30 horas en construir la tabla. Si el ataque se realiza en paralelo 4 hosts atacantes tardarían 7,5 horas y 8 hosts atacantes 3.75 horas.

Cuando el atacante recibe un paquete mira en la tabla a que keystream corresponde el IV recibido y hace una XOR del keystream con el cyphertext del paquete para obtener el plaintext.

5.1.3 Debilidades en el algoritmo key Scheduling de RC4

Scott Fluhrer, Itsik Mantin y Adi Shamir publicaron en Agosto del 2001 la demostración teórica de la vulnerabilidad más devastadora de las existentes hasta ahora en la encriptación WEP.

Su trabajo, de cierta complejidad matemática, se puede encontrar en:
http://www.eyetap.org/~rguerra/toronto2001/rc4_ksaproc.pdf

Demostraron que usando sólo la primera palabra de un keystream, podían obtener información de la clave secreta compartida. Se buscan IVs que causen que no haya información de la llave en el keystream. Los autores llamaron a esta condición "*resolved condition*" o condición resuelta. Cada uno de estos paquetes resueltos sólo tiene ausencia de información de un byte de la llave, y este byte debe ser adivinado correctamente para que el siguiente paquete pueda ofrecer información del siguiente byte de la llave. Para realizar el ataque más rápidamente sólo se buscan los IVs débiles que cumplen esta condición. Hay una posibilidad del 5% de adivinar el byte de la llave correctamente cuando encontramos un paquete resuelto (con un IV débil). Pero como hay gran cantidad de paquetes resueltos viajando por la red, las posibilidades son aún mayores.

Adam Stubblefield, un trabajador de AT&T Labs, fue la primera persona que implementó este ataque con éxito. Añadió que en el tráfico IP se añade una cabecera 802.2 extra, y esto hace que el ataque sea más sencillo de implementar, ya que cada paquete IP tiene el mismo primer byte de plaintext. Para realizar el ataque con éxito, durante la primera fase del ataque, los primeros pocos bytes deben ser adivinados correctamente. Stubblefield utilizo dos métodos para conseguirlo.

El primer método es apuntar los paquetes resueltos para disminuir las posibles combinaciones de bytes de la llave. Se puede comprobar si las llaves son correctas mediante el ICV de los paquetes desencriptados.

El segundo método se centra en la manera en que se distribuyen las llaves WEP. Se supone que el usuario introducirá una clave fácil de recordar en el software de configuración. Una llave fácil de recordar debe contener caracteres ASCII. Comprobando si los bytes de la llave concuerdan con caracteres ASCII cómo letras o símbolos etc. Las posibilidades de adivinar la llave correcta aumentan.

Cuando se han recolectado suficientes IVs débiles para un valor concreto de un byte de la llave, el análisis estadístico muestra una tendencia hacia un valor en particular para ese byte de la llave. Se le da una puntuación a cada una de las 256 posibilidades según la probabilidad de ser el valor correcto.

La llave se intenta adivinar a partir de los valores con mayor puntuación en el análisis estadístico (Hay un 95% de posibilidades de que un IV no revele información sobre un byte de la llave!).

Los IV's debiles no estan distribuidos de forma lineal a través del espacio de IV's.

El número de paquetes que necesitamos recolectar antes de descubrir un byte de la llave varía en función de en que valor se encuentre el contador de IV’s de las tarjetas que estemos monitorizando. Hay 9.000 IV's débiles en los 16 millones de IV's posibles.

¿Cuántos paquetes encriptados necesitamos recolectar para crackear la llave WEP?

- La mayoría de las llaves pueden ser adivinadas después de encontrar aproximadamente 2000 paquetes resueltos
- Algunas llaves requieren que capturemos incluso más de 4000 paquetes resueltos

Podremos adivinar la llave después de recolectar de 5 a 10 millones de paquetes encriptados.

Poco después de que el trabajo realizado por estos tres autores y la vulnerabilidad práctica de Stubblefield fueran publicados, aparecieron dos herramientas en Internet que implementan totalmente el ataque:

- Wepcrack: http://wepcrack.sourceforge.net/

- Airsnort: http://airsnort.shmoo.com/

5.2 Ataques a redes wireless

Vista la manera romper la encriptación WEP ya no debería ser un problema para nosotros, por eso en la implementación de los ataques que vamos a ver a continuación no vamos a hablar de WEP ya que si la WLAN que estamos "auditando" tiene encriptación WEP ya disponemos de las herramientas necesarias para obtener la clave y por tanto, podremos realizar los distintos ataques tanto si existe encriptación WEP como si no.

5.2.1 Romper ACL's basados en MAC

Una de las medidas más comunes que se utilizan para securizar una red wireless es restringir las máquinas que podrán comunicarse con el Punto de Acceso haciendo filtrado por dirección MAC en éste. Para esto se suele crear una tabla en el punto de acceso que contiene todas las MACs de los clientes que están autorizados para conectar.

Aunque esto pueda parecer una medida de seguridad efectiva, no lo es, ya que es muy fácil cambiar la dirección MAC que aparece en los paquetes que un cliente envía, y hacernos pasar por uno de los equipos que si que tienen acceso a la red.

Para llevar a cabo el ataque basta con esnifar durante un momento el tráfico y fijarnos en la MAC de cualquiera de los clientes, sólo hace falta que nos pongamos su misma MAC y ya habremos saltado la restricción. Esto es sencillo de implementar, por ejemplo en el sistema operativo Linux se puede realizar con el comando *ifconfig* dependiendo del tipo de tarjeta que tengamos. También existen otras utilidades para cambiar la MAC como por ejemplo *setmac*.

Hay que tener en cuenta que si hay dos máquinas en la red con la misma dirección MAC podemos tener problemas, aunque generalmente en las redes wireless esto no suele ser un problema muy grave ya que el Punto de Acceso no puede distinguir que verdaderamente hay dos máquinas con la misma MAC. De todas formas, si queremos podemos "anular" a la máquina que le hemos "robado" la dirección MAC. Para hacer esto, debemos implementar un ataque de Denegación de Servicio, como el que veremos seguidamente.

5.2.2 Ataque de Denegación de Servicio (DoS)

Para realizar este ataque basta con esnifar durante un momento la red y ver cual es la dirección MAC del Punto de Acceso. Una vez conocemos su MAC, nos la ponemos y actuamos como si fuéramos nosotros mismos el AP. Lo único que tenemos que hacer para denegarle el servicio a un cliente es mandarle continuamente notificaciones (*management frames*) de desasociación o desautenticación. Si en lugar de a un solo cliente queremos denegar el servicio a todos los clientes de la WLAN, mandamos estas tramas a la dirección MAC de broadcast.

Existen varias herramientas para realizar este ataque, las más comunes para el sistema operativo Linux son:

- wlan-jack: perteneciente a las utilidades air-jack, presentadas en la concentración Black Hat 2002 en Las Vegas, se puede encontrar en http://802.11ninja.net.

- dassoc: envia tramas de desasociacion, herramienta desarrollada por @stake (antes L0pht), se puede encontrar en http://www.atstake.com.

5.2.3 Descubrir ESSID ocultados

Como hemos comentado anteriormente, para que un cliente y un AP se puedan comunicar, ambos deben tener configurado el mismo ESSID, es decir, deben pertenecer a la misma red wireless.

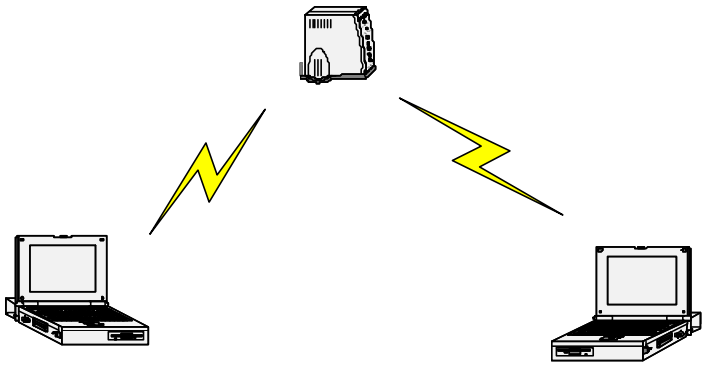
Una medida de seguridad bastante común es “ocultar” el ESID, es decir, hacer que el AP no mande BEACON FRAMES, o en su defecto no incluya el ESSID en éstos.

En este caso, para descubrir el ESSID deberíamos esnifar y esperar a que un cliente se conectara, y veríamos el ESSID en la trama PROVE REQUEST del cliente (en el caso de que no se manden BEACON FRAMES), o en la trama PROVE RESPONSE del AP. Pero también podemos “provocar” la desconexión de un cliente, utilizando el mismo método que en el ataque DoS, pero mandando sólo una trama de desasociación o de desautenticación en lugar de mandarlas repetidamente, es decir, nos ponemos la dirección física del AP y mandamos una trama DEAUTH o DISASSOC a la dirección MAC del cliente (o a la de broadcast), entonces el cliente intentará volver a asociarse o autenticarse, con lo que podremos ver el ESSID en los *management frames*.

Para implementar el ataque podemos usar la herramienta essid-jack, que también pertenece al paquete de utilidades air-jack para Linux (http://802.11ninja.net).

5.2.4 Ataque Man in the middle

El ataque de *Man in the middle*, también conocido como *Monkey in the middle* consiste en convencer al cliente (la victima) de que el host que hay en el medio (el atacante) es el AP, y hacer lo contrario con el AP, es decir, hacerle creer al AP que el atacante es el cliente.



WLAN antes del ataque

Para realizar este ataque, primero debemos esnifar para obtener:

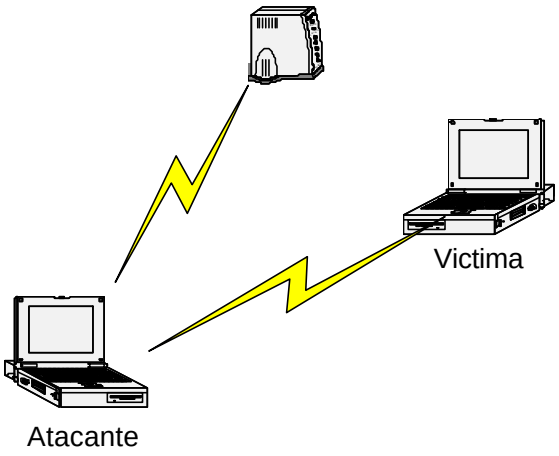
- El ESSID de la red (si esta ocultado, usaremos el método anterior)
- La dirección MAC del AP
- La dirección MAC de la victima

Una vez conocemos estos datos, utilizamos el mismo método que en el ataque DoS, para desautenticar a la victima del AP real, es decir, el atacante spoofea su MAC haciéndose pasar por el AP y manda tramas DEAUTH a la victima. La tarjeta wi-fi de la victima empezará entonces a escanear canales en busca de un AP para poderse autenticar, y ahí es donde entra en juego el atacante.

El atacante hace creer a la victima que él es el AP real, utilizando la misma MAC y el mismo ESSID que el AP al que la victima estaba autenticada anteriormente, pero operando por un canal distinto. Para realizar esto la tarjeta wi-fi del atacante debe estar en modo **master** (véase apéndice A).

Por otra parte, el atacante debe asociarse con el AP real, utilizando la dirección MAC de la victima.

De esta manera hemos conseguido insertar al atacante entre la victima y el AP, veamos como quedaría la WLAN después de realizar el ataque.



WLAN después del ataque

De esta manera todos los datos que viajan entre la victima y el AP pasan a través del atacante. Como el ataque ha sido realizado a nivel de enlace (nivel 2), el atacante puede ver, capturar e incluso modificar las tramas en los niveles superiores del modelo OSI.

Es muy fácil implementar este tipo de ataques utilizando el driver air-jack con la herramienta monkey-jack.

Hay que tener en cuenta que muchas soluciones de seguridad están pensadas asumiendo que las capas 1 y 2 son seguras, esto como hemos visto es incierto para las redes wireless y por tanto el uso de según que tipo de solución podría no ser adecuado para estas redes. Hay que ir con mucho cuidado sobre todo en implementaciones de VPN que no realizan las comprobaciones necesarias de autenticación para protegerse de ataques *Man in the middle* en redes wireless.

5.2.5 Ataque ARP Poisoning

El *ARP cache poisoning* es un ataque que sólo se puede llevar a cabo cuando el atacante está conectado a la misma LAN lógica que las victimas, limitando su efectividad a redes conectadas con switches, hubs y bridges, pero no routers. La mayoría de los Puntos de Acceso 802.11b actúan como bridges transparentes de capa 2, lo que permite que los paquetes ARP pasen de la red wireless hacia la LAN donde está conectado el AP y viceversa. Esto permite que se ejecuten ataques de *ARP cache poisoning* contra sistemas que están situados detrás del Punto de Acceso, como por ejemplo servidores conectados a un switch en una LAN a los que se pueda acceder a través de la WLAN.

Vamos a ver el ejemplo para entender mejor la idea:

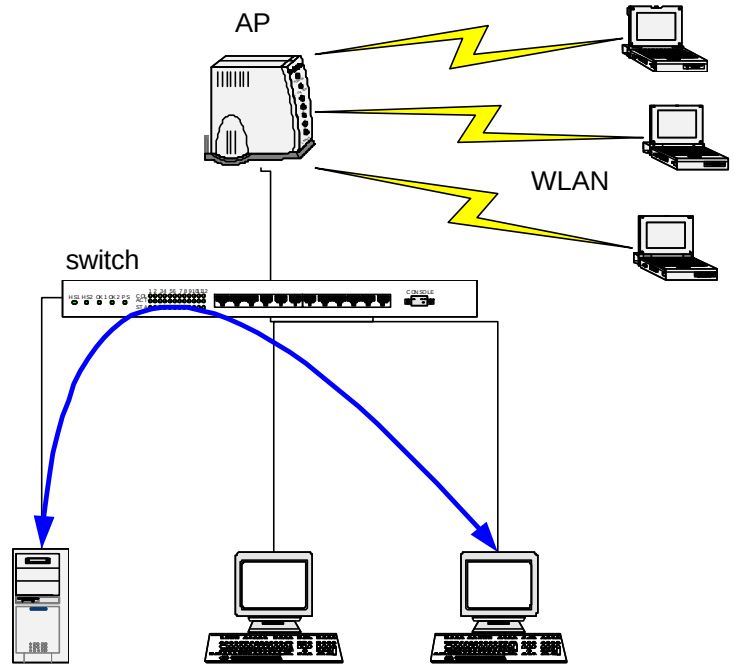


Tabla ARP – PC 1

Dirección IP	Dirección MAC
192.168.1.2	3B:AF:44:F8:2C:B3
192.168.1.3	3C:2A:4A:59:1A:7A

Tabla ARP – PC 3

Dirección IP	Dirección MAC
192.168.1.2	3B:AF:44:F8:2C:B3
192.168.1.1	6A:8F:5E:99:2B:78

Tabla ARP – PC 2

Dirección IP	Dirección MAC
192.168.1.1	6A:8F:5E:99:2B:78
192.168.1.3	3C:2A:4A:59:1A:7A

El servidor PC 1 se comunica con PC 3 a través del switch, si un atacante desde la WLAN envenena la tabla de ARP's de PC 1 y de PC 3 podrá realizar un ataque del tipo *Man in the Middle* situándose entre los dos hosts de la red con cables.

Así es como se efectuaría la comunicación después del ataque:

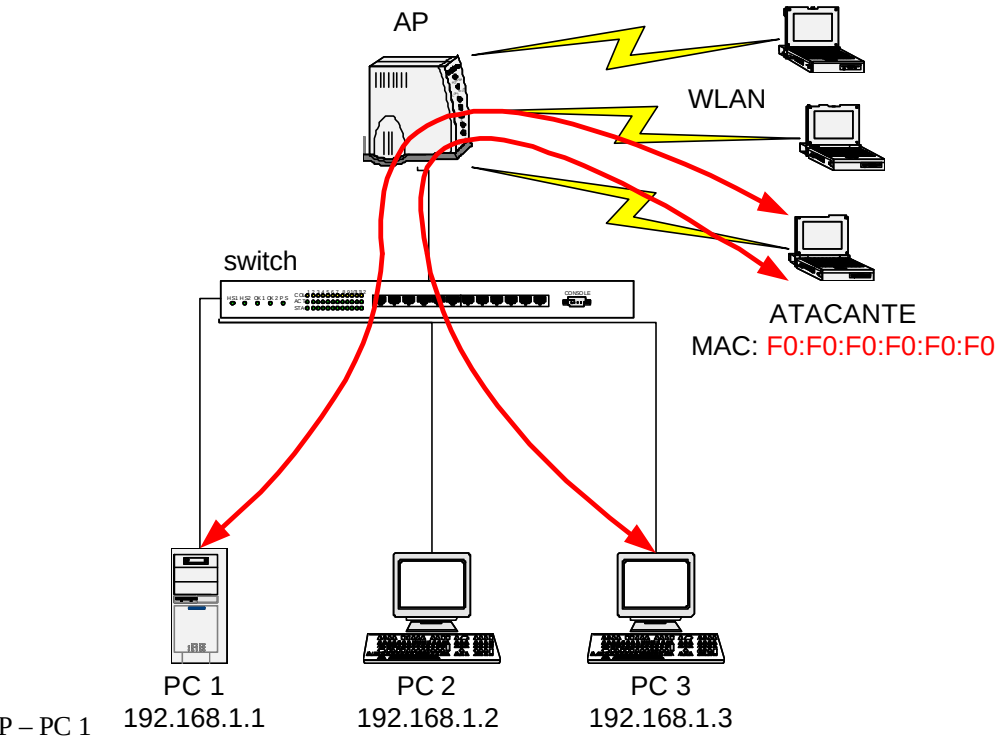


Tabla ARP – PC 1

Dirección IP	Dirección MAC
192.168.1.2	3B:AF:44:F8:2C:B3
192.168.1.3	F0:F0:F0:F0:F0:F0

Tabla ARP – PC 3

Dirección IP	Dirección MAC
192.168.1.2	3B:AF:44:F8:2C:B3
192.168.1.1	F0:F0:F0:F0:F0:F0

Tabla ARP – PC 2

Dirección IP	Dirección MAC
192.168.1.1	6A:8F:5E:99:2B:78
192.168.1.3	3C:2A:4A:59:1A:7A

El atacante manda paquetes *ARP REPLY* a PC 2 diciendo que la dirección IP de PC 1 la tiene la MAC del atacante, de esta manera consigue "envenenar" la caché de ARP's de PC 2. Luego realiza la misma operación atacando a PC 1 y haciéndole creer que la dirección IP de PC 2 la tiene también su propia MAC.

Como ARP es un protocolo *stateless*, PC 1 y PC 2 actualizan su caché de acuerdo a la información que el atacante ha inyectado a la red.

Como el switch y el AP forman parte del mismo dominio de broadcast, los paquetes ARP pasan de la red wireless a la red con cables sin ningún problema.

Para realizar el ataque ARP Poisoning, existen múltiples herramientas en Internet, ya que este ataque no es específico de las redes wireless, la más famosa es el sniffer Ettercap (http://ettercap.sourceforge.net).

Podríamos frenar este ataque creando dos VLAN's en el switch, una para la boca a la que está conectado el AP y la otra para el resto de máquinas. Otra forma de frenarlo sería utilizando tablas de ARP estáticas.

6. Conclusiones

Confidencialidad
Tu red es vulnerable desde una distancia de 10 kilómetros.
Todo tu tráfico puede ser desencriptado fácilmente.

Control de Acceso
Cualquier persona puede unirse a tu red cuando quiera.
Seguramente incluso a tu red interna.

Integridad
Todo tu tráfico es vulnerable a ser modificado y re-enviado.
Hackeo tu servidor DHCP – todo tu tráfico se enruta ahora a través de mi portátil.

Fiabilidad
Tu red puede venirse abajo en cualquier momento.

Hoy en día las redes wireless se están implantando en muchas empresas españolas, esta tecnología está ahora en fase de crecimiento y va evolucionando cada vez más rápido.

Muy pronto los ordenadores portátiles se venderán con tarjetas wi-fi integradas, de hecho muchas PDA's ya se venden con estas tarjetas. Cada vez más la tecnología 802.11 va a ir ganando terreno a las redes con cables y finalmente las redes domésticas van a convertirse en redes inalámbricas, los equipos de sobremesa también se venderán con tarjetas wireless integradas.

Si no conocemos las vulnerabilidades de estas redes es imposible que seamos capaces de securizarlas correctamente, proporcionando el grado de seguridad telemática apropiado a una tecnología tan vulnerable como esta. Por esto, es muy importante preocuparse por la seguridad en las redes inalámbricas y conocer a fondo sus detalles.

Un atacante ya no requiere acceso físico a la red, las redes wireless están muchas veces situadas detrás de los firewalls, son redes muy susceptibles a ataques de Man In The Middle y Denegaciones de Servicio (DoS) y el atacante puede marcharse con relativa impunidad.

Evidentemente existen soluciones a todos los ataques planteados en este documento, aunque esto queda fuera de la índole del mismo. No descarto la inclusión de un nuevo apartado con soluciones en un futuro.

7. Bibliografía

IEEE 802.11Working Group.
http://grouper.ieee.org/groups/802/11/index.html

N. Borisov, I. Goldberg, y D. Wagner, “Intercepting Mobile Communications: The Insecurity of 802.11.”, Enero 2001.
http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html

C. Klaus, “ISS Wireless LAN 802.11b Security FAQ”, Octubre 2002.
http://www.iss.net/wireless/WLAN_FAQ.php

W. Arbaugh, N. Shankar, y Y. Wan. “Your 802.11 wireless network has no clothes”, Marzo 2001.
http://www.cs.umd.edu/~waa/wireless.pdf

W. Arbaugh. “An inductive chosen plaintext attack against WEP/WEP2”. Documento IEEE 802.11-01/230, Marzo 2001.
http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/0-230.zip

S. Fluhrer, I. Mantin, y A. Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4”, Agosto 2001.
http://www.eyetap.org/~rguerra/toronto2001/rc4_ksaproc.pdf

A. Stubblefield, J. Ioannidis, y A. Rubin, “Using the Fluhrer, Mantin, and Shamir Attack to Break WEP”, Agosto 2001.
http://www.cs.rice.edu/~astubble/wep/wep_attack.html

J. Walker, “Unsafe at any key size: an analysis of the WEP encapsulation”. Documento IEEE 802.11-00/362, Octubre 2000.
http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/0-362.zip

R. Baird y M. Lynn, “Advanced 802.11 Attack”, Black Hat 2002, Julio 2002.
http://802.11ninja.net/bh2002.ppt

B. Fleck y J. Dimov. "Wireless Access Points and ARP Poisoning: Wireless vulnerabilities that expose the wired network", Septiembre 2001.
http://www.cigitallabs.com/resources/papers/download/arppoison.pdf

M. Barrie. “Computer and Network security: Wireless Security”, Octubre 2002.
http://www.ee.usyd.edu.au/~mattb/2002/lectures/5610_lecture19.ppt

Apéndice A

Tarjetas Wi-Fi: modos y chipsets

Chipsets

Las tarjetas Wi-Fi (802.11b) utilizan un chipset u otro según fabricante. Es posible que el mismo fabricante comercialice tarjetas con chipsets distintos, para hacernos una idea estos son los chipsets más comunes:

- Hermes (Lucent)
  - Lucent / Agere / **Orinoco**
    - Orinoco, Avaya, Compaq, Lucent

- **Prism** 2 / 2.5 / 3 (Intersil)
    - D-Link, Linksys, Netgear, SMC, USR, Conceptronic

  El firmware tiene propiedades de AP, por tanto es posible configurar la tarjeta para que trabaje como si fuera un Punto de Acceso (modo Master).

- Airo (Aironet)
  - Cisco

- TI ACX100 (Texas Instruments)
  - 3Com / USR, D-Link, Wisecom, Eusso, Linksys (WAP11 v2.2)

Modos

Los modos más comunes que aceptan las tarjetas wi-fi son Ad-Hoc, Managed, Master, y monitor.

- **Ad-Hoc**: conectar dos PC's sin AP

- **Managed**: Tarjeta asociada con un AP

- **Master**: La tarjeta trabaja como un AP
  - HostAP: http://hostap.epitest.fi/
  - IEEE 802.1X
  - Dynamic WEP rekeying
  - RADIUS Accounting
  - RADIUS-based ACL for IEEE 802.11 authentication
  - Minimal IAPP (IEEE 802.11f)

  SÓLO FUNCIONA CON CHIPSET PRISM

- **Monitor**: Permite capturar paquetes sin asociarse a un AP o a una red ad-hoc.
  - Monitoriza un canal especifico sin transmitir paquetes
  - La tarjeta no mira los CRC's de los paquetes
  - **No** es lo mismo que el modo promiscuo

    Chipset PRISM: Sin problemas
    Chipset Orinoco: Parche http://airsnort.shmoo.com/orinocoinfo.html

Para realizar varios de los ataques descritos en este documento hace falta configurar la tarjeta para que funcione en modo **Master**, es decir, la tarjeta actúa como si fuera un Punto de Acceso.

Apéndice B

Encontrar redes wireless

Material necesario

- Ordenador portátil o PDA
- Tarjeta Wi-Fi con firmware adecuado.
- Programa o driver que permita poner la tarjeta en modo monitor
- Sniffer

Otros materiales adicionales

- Antena direccional o omnidireccional
- GPS
- Equipo electrógeno
- Mochila
- Auriculares
- Medio de transporte (coche, patines, bicicleta…)

Proceso a seguir

Antes de salir en busca de una red wireless, hay que configurar el equipo que nos permitirá detectar la red.

Se debe poner la tarjeta wi-fi en modo **monitor**, este modo es parecido al modo "promiscuo" de las tarjetas ethernet convencionales, lo que hace es dejar la tarjeta "a la escucha" por la frecuencia utilizada en 802.11b (2,4 GHz).
El método para poner la tarjeta en modo monitor es distinto para cada sistema operativo, y para tipo de tarjeta (según chipset), referirse al apéndice A para más información.

Explicaremos el método a seguir para las tarjetas Prism y Orinoco utilizando el sistema operativo GNU Linux, si se dispone de otro chipset u otro sistema operativo, es recomendable buscar esta información por Internet (ojo, esto no es posible con todos los chipsets!).

Instalar wireless-tools:
http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html

Instalar pcmcia-cs:
http://pcmcia-cs.sourceforge.net/

Si la tarjeta tiene el chipset Orinoco hay que parchear el pcmcia-cs para poder ponerla en modo monitor, el parche se puede encontrar en la siguiente página:

Orinoco Monitor Mode Patch Page
http://airsnort.shmoo.com/orinocoinfo.html

Una vez cargados los módulos de la tarjeta (provistos por el pcmcia-cs), hay que utilizar el comando “iwpriv” (provisto por las wireless-tools) para ponerla en modo monitor, con la siguiente sintaxis:

```
# iwpriv wlan0 monitor 1 #canal
```

Una vez tenemos la tarjeta en modo monitor, hay que instalar un sniffer que nos permita capturar las tramas wireless. Los más comunes son los siguientes:

Sistema Operativo Linux:
Kismet: http://www.kismetwireless.net/
Airsnort: http://airsnort.shmoo.com/
Ethereal: http://www.ethereal.com/

Sistema Operativo Windows:
Airopeek: http://www.wildpackets.com/products/airopeek
NetStumbler: http://www.netstumbler.com/

En este punto, ya estamos listos para salir a la calle en busca de una red wireless. Es aconsejable desplazarse a poca velocidad, moverse cerca de los edificios y hacerlo preferiblemente en horario laboral.

Según el medio de transporte que utilicemos, esta práctica se denomina de la siguiente manera:

- WarWalking: Andando
- WarSkating: En patines
- WarCycling: En bicicleta o ciclomotor

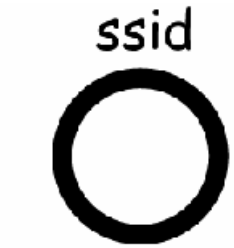
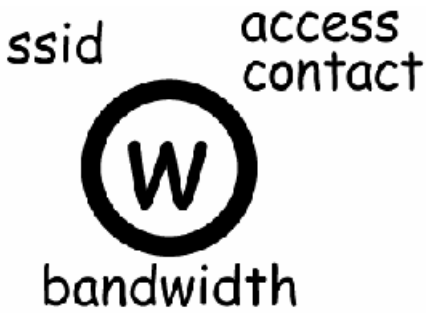
El sniffer más comodo para estas prácticas es el NetStumbler, ya que emite un tono por la salida de audio cuando detecta una red wireless. Se suele poner el portátil en una mochila, con los auriculares conectados, y el monitor en modo Stand-by (para ahorrar batería).

- WarDriving: Coche

Podemos utilizar el kismet o el Airsnort, que permiten comunicación directa con dispositivos GPS. Cuando el sniffer detecta una WLAN, guarda un registro con toda la información que ha podido obtener de la red mientras hemos tenido cobertura, si disponemos de GPS podemos saber en que posición exacta estaba situada la red y que área de cobertura tenía.

Existen otros métodos como el WarFlying, utilizando un avión como medio de transporte.

Una vez hayamos encontrado una red wireless, es aconsejable marcarla con los datos que conozcamos, para que otra persona pueda localizarla fácilmente si en el momento que ve la marca no dispone del material necesario para conectar a la red wireless. Esta práctica es conocida como WarChalking y consiste en realizar una marca con tiza en una pared de la zona donde haya cobertura. La convención de símbolos utilizada es la siguiente:

SÍMBOLO	SIGNIFICADO
	Nodo Abierto
	Nodo cerrado
	Nodo con WEP

Se puede encontrar más información sobre WarChalking en
http://www.warchalking.org