Breaking WEP

Julián Gonzalez Calderón

Abstract

El objetivo de este trabajo práctico es el de investigar el protocolo WEP y la razón de su vulnerabilidad

Contents

1 ¿Qué es WEP? **3**

2 ¿Cómo funciona WEP? **3**

3 ¿Cómo funciona RC4? **4**

4 Inseguridades de RC4 **4**
4.1 Debilidad de la Invariancia 4
4.2 Debilidad de clave relacionada 5

5 Detalles del *Known IV Attack* **5**

6 Aplicación del Ataque en WEP **6**

7 Preparación para el Ataque **7**

8 Obtención de Paquetes **7**

9 Resultados del Ataque **7**

10 Otras Inseguridades **7**

11 Evolución de WEP **8**

1 ¿Qué es WEP?

El sistema ***WEP*** es un sistema de cifrado opcional incluido en el estándar ***IEEE 802.11, edición 1999.*** Para proteger la confidencialidad de los datos intercambiados a través de un medio compartido.

Se utiliza una clave en común entre el remitente y el receptor. Se presume que esta clave ya fue enviada de forma segura por otro medio, independiente del ***IEEE 802.11***.

Debido a esta clave común, es un algoritmo simétrico, ya que la misma clave se utiliza para encriptar o desencriptar.

2 ¿Cómo funciona WEP?

Sea Data el segmento de datos que quiero enviar, el tamaño total del segmento concatenado será de n, en bytes.

El algoritmo parte de una semilla de encriptación, que se obtendrá al concatenar un vector de inicialización de 24 bytes que llamaremos IV, con la clave compartida de 40 bytes, que llamaremos SK. El operador $\|$ denota concatenación.

$$\text{Seed} = IV \| \text{SK}$$

La semilla, de un total de 64 bytes, será introducida en el algoritmo de encriptación ***RC4,*** el cual devolverá un flujo de claves de longitud n que se utilizará para encriptar el segmento de datos.

$$\text{Key Stream} = \text{RC4}(Seed)$$

Luego, se aplica una aplicación de ***XOR*** entre el segmento de datos y el flujo de claves para hallar el segmento cifrado.

$$\text{Cipher} = \text{Key Stream} \oplus \text{Data}$$

Finalmente, se concatena el vector de inicialización IV al segmento cifrado, y se envía a través del medio.

$$\text{Sent} = \text{Cipher} \| \text{IV}$$

Cuando el receptor recibe el paquete, extrae el vector de inicialización del segmento y lo introduce a ***RC4***, junto a la clave compartida, para hallar el mismo flujo de claves que utilizó el remitente.

Luego, se aplica una aplicación de ***XOR*** entre el segmento cifrado y el flujo de claves, obteniendo así el segmento original.

$$\text{Data} = \text{Cipher} \oplus \text{Key Stream}$$

3 ¿Cómo funciona RC4?

El protocolo ***RC4*** consta de dos algoritmos, extremadamente simples, que parten de una semilla dada. Definiremos N como la cantidad de permutaciones posibles de palabras de longitud n, comúnmente, tendremos que $N = 256$.

1. ***KSA - Key Scheduling Algorithm:*** Parte del vector de clave y crea un conjunto desordenado S de valores entre 0 y N. Esto se obtiene creando una permutación identidad de valores entre 0 y N e intercambiando elementos a partir de la clave.

   ```
   for(i = 0 to N-1)
   {
      S[i] = i;
   }
   j=0;
   for(i = 0 to N-1)
   {
      j = ( j + S[i] + K[i mod keylength] ) mod N;
      intercambia S[i] and S[j];
   }
   ```

2. ***PRGA - Pseudo-Random Generation Algorithm:*** A partir del vector de claves y el conjunto permutación, va devolviendo valores del flujo de claves. Inicialmente, $i, j = 0$.

   ```
   i = (i + 1) mod N;
   j = (j + S[i]) mod N;
   intercambia S[i] and S[j];
   t = (S[i] + S[j]) mod N;
   Exponer valor de S[t];
   ```

   Cada valor expuesto por el algoritmo será un octeto del flujo de claves, este podrá tener cualquier longitud que se desea.

4 Inseguridades de RC4

4.1 Debilidad de la Invariancia

El primer algoritmo de, ***KSA***, expone dos inseguridades significantes. La primera es la existencia de una gran clase de ***claves débiles***, en las cuales una pequeña parte de la llave determina un gran número de ***bits*** de la permutación inicial. Además, ***PRGA*** traduce estos patrones en la permutación inicial, en patrones en el prefijo del flujo de claves.

Esta afirmación permite distinguir fácilmente flujos de ***RC4*** de segmentos de datos aleatorios, debido a que las palabras iniciales contienen patrones fácilmente reconocibles. Cuando las claves son seleccionadas bajo una distribución uniforme, esta inclinación se atenúa, pero aún permite la construcción de un diferenciador eficiente. Además, los patrones se extienden a las primeras decenas de palabras, por lo que la eficiencia del diferenciador no disminuye si se descartan las primeras palabras.

4.2 Debilidad de clave relacionada

La segunda debilidad está relacionada con la vulnerabilidad de las claves, que aplica cuando parte de la clave presenta es expuesta al atacante. Cuando la misma porción de clave secreta está presente en diferentes valores expuestos, el atacante puede derivar la parte secreta al analizar la palabra inicial del flujo de claves.

Para hacer esto, el algoritmo se aprovecha de la posibilidad de encontrar claves tales que causen que el algoritmo entre en condición de ***resuelto***.

Sea S_i el vector permutación tras la iteración i de la permutación, definimos $X_i = S_i[1], Y_i = S_i[X]$. Si para algún punto de la permutación, se cumple que i es mayor o igual que $1, X_i, Y_i$, entonces diremos que el algoritmo está en condición de resuelto.

Cuando ocurre esto, aseguramos con probabilidad mayor a $e^{-3} \approx 0.05$ que los elementos $X_i, Y_i, S_i[Y_i]$ no volverán a participar en un intercambio. Luego, la primera palabra del flujo de claves estará dado por $S_i[X_i + S_i[Y_i]]$.

Si no se cumple esto, entonces los elementos volverán a participar en intercambios, haciendo que el valor resultante sea efectivamente aleatorio. Esto implica que si repetimos este análisis para muchos valores de escenarios resueltos, entonces el valor más probable será el correcto.

5 Detalles del *Known IV Attack*

Debido a como funciona el algoritmo de ***WEP***, únicamente estudiaremos el escenario en el que el vector de inicialización precede a la clave secreta compartida, aunque ambas situaciones son vulnerables al mismo tipo de ataque.

Sea IV el vector de inicialización conocido, de longitud I, y SK la clave secreta, de longitud $l - I$, entonces construiremos la semilla para ***RC4*** como $K = \text{IV}$ SK, con longitud l. Trataremos de derivar el valor de la palabra B de la clave secreta, esto es, la palabra $I + B$ de la semilla. Para que este ataque sea posible, el atacante debe poder obtener la primera palabra del flujo de claves. A esta primera palabra, la denominaremos Out.

Si se cumple que $S_I[1] < I$ y $S_I[1] + S_I[S_I[1]] = I + B$, entonces estaremos ante una condición resuelta tras la ronda $I + B$ con una probabilidad relativamente

alta: $p \approx e^{-\frac{2B}{N}}$. Luego, podremos tomar la suposición de que esto se cumplirá.

Por la condición de resuelto, tendremos que Out $= S_{I+B}[I+B]$ será cierto más del 5% de las veces. Al ser el valor más probable, lo asumiremos cierto. Luego, podremos predecir la palabra buscada, a partir de la siguiente fórmula:

$$K[I+B] = S^{-1}_{I+B-1}[\text{Out}] - j_{I+B-1} - S_{I+B-1}[I+B]$$

Algo importante a notar, es que este ataque requiere de una gran cantidad de paquetes encriptados con distintos IVs, por lo que si los ***hosts,*** en lugar de utilizar valores distintos cada vez, alternan entre dos o una pequeña cantidad valores distintos, entonces el ataque deja de funcionar.

Cabe notar que si ocurre esto, la red será susceptible a otro tipo de ataques. Debido a que es posible obtener el flujo de claves a través del cifrado y el valor desencriptado, si un ataque obtiene esto podrá desencriptar fácilmente todos los paquetes provenientes del mismo vector de inicialización, incluso sin conocer la clave.

6 Aplicación del Ataque en WEP

El ataque consiste en observar selectivamente vectores IV tal que podamos calcular la permutación en la iteración I sin necesitar la clave secreta. En este caso, I será 3. Debemos previamente conocer la primera palabra de flujo de claves.

Se necesitan aproximadamente 60 vectores de la forma $(A+3, N-1, V)$, donde A indica la cantidad de palabras conocidas de la clave secreta, y V tomará valores cualesquiera.

La elección de estos valores particulares, nos asegura que todas las permutaciones, desde la inicial hasta la ronda I son conocidos, lo que nos permite trabajar con el resultado de la anterior sección. Para cada vector válido observado, calcularemos sus permutaciones hasta la ronda I.

Si para alguno de los valores hallados, no se cumple la condición, entonces lo descartamos. Para los valores que no descartamos, podemos tomar las suposiciones indicadas en la anterior sección y derivar $K[A+3]$ con probabilidad $p > 0.05$. Al examinar una gran cantidad de resultados, un atacante puede derivar la clave correctamente con probabilidad $p > 0.5$.

Debemos tener en cuenta que este ataque es válida para todos los ataques que cumplan la condición propuesta, aunque estos no pertenezcan a la forma mencionada.

7 Preparación para el Ataque

Necesitamos dos elementos fundamentales para la ejecución de este ataque. La primera es alguna tarjeta o adaptador de red que nos permita escuchar y enviar paquetes de redes ***802.11***. A partir de un programa conocido como ***sniffer*** o analizador de paquetes, podremos visualizar todos los paquetes que son enviados a través del medio.

Estos programas configuran el adaptador de red en modo ***promiscuo***. Este permite que los ***frames*** no destinados a nuestro dispositivo (a partir de la dirección ***MAC***), no sean descartados.

El segundo elemento que necesitamos, es conocer la primera palabra de los segmentos de datos enviados, sin encriptar. Al conocer esto, y con el segmento cifrado, podremos determinar la primera palabra del flujo de claves.

8 Obtención de Paquetes

Como vimos, el método requiere de, inicialmente, recolectar un gran número de paquetes cifrados con distintos IVs. Una opción es esperar el tiempo necesario hasta recolectar los paquetes requeridos, pero existe un truco para acelerar el proceso.

Otra opción es la de tomar un paquete ya perteneciente a la red (encriptado con el mismo algoritmo), y reenviárselo repetidamente a uno de los nodos de la red. Esto también aumentará la carga de la red.

Recordemos que el ataque es un ataque pasivo, por lo que este paso no es necesario, pero altamente efectivo.

9 Resultados del Ataque

En ***2001***, ***Fluhrer, Mantin y Shamir*** tratarón de aplicar el ataque mencionado, para vulnerar una red con ***802.11*** autenticación ***WEP***.

A partir de lo mencionado, y con algunas modificaciones, pudieron vulnerar ***WEP*** utilizando un total de un millón de paquetes.

10 Otras Inseguridades

El ataque mencionado no es la única forma de vulnerar este protocolo, por ejemplo, veamos el ataque activo de inyección de paquetes.

El atacante debe conocer el contenido real de un paquete encriptado, de esta forma, podremos obtener el flujo de claves correspondiente. Luego, podremos encriptar cualquier paquete que nosotros queramos con este flujo de claves. De esta forma, lograremos introducir paquetes malignos en la red.

11 Evolución de WEP

A principios de 2001, se empezaron a identificar varias debilidades a partir de analistas criptográficos. Unos meses más tarde, el **IEEE** creó la corrección de seguridad ***802.11i*** para neutralizar esto. En ***2004***, finalmente, el estándar ***802.11i***, también conocido como ***WPA***, fue ratificado.