Universidad de Alcalá

Ataques a TLS desde 2012 - POODLE Detalles técnicos

Máster en ciberseguridad

Asignatura: Comunicaciones seguras

Kevin van Liebergen, Daniel López, Diego Espinosa, Miguel Eduardo Ortiz, Alejandro Rodriguez, Marco Santacroce

2021

1 Abstract

Este apartado se ha realizado entre los integrantes Alejandro Rodríguez y Kevin van Liebergen después de realizar un sorteo aleatorio para designar a los distintos campos que cada persona tiene que trabajar.

Para empezar con los detalles técnicos, en primer lugar nos hicimos con el Whitepaper [1] para abordar el problema de la manera más técnica posible, y seguidamente leímos fuentes secundarias (artículos o vídeos) para analizar más detalles que se omite en el Whitepaper y poder abordar un conocimiento global sobre la vulnerabilidad.

Por último, se va a desarrollar el modo de funcionamiento del ataque junto con un ejemplo práctico para explicarlo de la manera más clara y legible posible.

2 Introducción

POODLE es un ataque que se aprovecha de una vulnerabilidad en SSL (protocolo de cifrado que actúa sobre la capa de transporte, cifrando la capa de aplicación), siendo su CVE-2014-3566 [1]. Esta vulnerabilidad, descubierta en 2014 por el grupo de analistas de seguridad de Google (aún no se acuñó el término `Project Zero`), revolucionó y dejó obsoleto completamente a SSL v3.0 de una vez, sin embargo por temas de compatibilidad en servidores y navegadores aún se sigue utilizando, siendo su uso completamente no recomendable.

POODLE permite poder filtrar información confidencial de las peticiones que se realizan al servidor, pudiendo conocer datos sensibles como cookies, contraseñas y demás.

POODLE (Padding Oracle On Downgraded Legacy Encryption) se trata de una vulnerabilidad que afecta a servicios que utilizan el cifrado SSL en su versión 3.0, además existen implementaciones sobre este tipo de ataque que afectan a los cifrados TLS 1.0, 1.1 Y 1.2 [2].

Este ataque, debido a la dificultad que tiene para explotarlo, tiene una puntuación en la versión 3 de CVSS un 3.4 y en la versión 2 de CVSS un 4.3 [3]. Esto es debido a que el atacante necesita tener acceso a la red de la víctima para poder realizar un downgrade.

3 Modo de funcionamiento

Cuando el cliente (navegador) se conecta con el servidor, envía un mensaje con una lista de todos los posibles cifrados que admite junto con sus versiones.

[1] https://nvd.nist.gov/vuln/detail/CVE-2014-3566
[2] https://nvd.nist.gov/vuln/detail/CVE-2019-6593
[3] https://nvd.nist.gov/vuln/detail/CVE-2014-3566

El atacante puede modificar dicha lista eliminando los cifrados más seguros, dejando como el algoritmo más seguro que admite el navegador la versión 3.0 de SSL. Sin embargo es posible realizar dowgngrades legítimos debido a temas de conexión.

SSL v3.0 utiliza un cifrado de flujo RC4 o un cifrado de bloques CBC (Cipher-block chaining), POODLE ataca a cifrados que utilizan CBC (AES, DES, etc), en el que como se muestra en la imagen 1, el valor de cada bloque cifrado depende del anterior, y lo mismo para el descifrado.

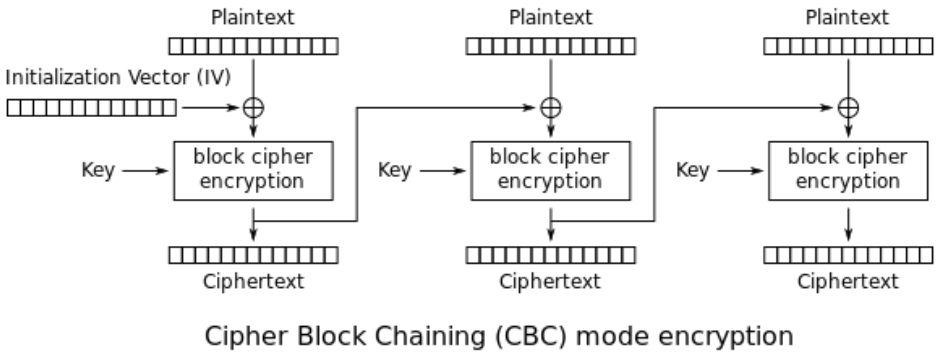

Figura 1: Modo de cifrado CBC [2]

Y la manera matemática de cifrar y descifrar cualquier mensaje se realiza mediante las siguientes ecuaciones:

$$C_i = E_k(P_i \oplus P_{i-1}), C_0 = IV$$
$$P_i = D_k(C_i) \oplus C_{i-1}, C_0 = IV$$

Siendo i en C_i el número de bloque que corresponde el mensaje una vez divididos entre el tamaño del bloque.

Lo interesante de este algoritmo es que utiliza como cifrado de autenticación MAC-then-encrypt (MtE), cuando se va a enviar un mensaje en texto plano se realiza lo siguiente [3]:

Primero se calcula la MAC del texto a enviar, después se añade dicha MAC al final del paquete (el tamaño de la MAC en SSL 3.0 CBC es de 20 bytes), y por último se añade el padding para que la longitud de todo el texto (plano + MAC + padding) sea múltiplo del tamaño de cada bloque (8 o 16 bytes), por último se cifra todo el texto y se envía al servidor como se muestra en la figura 2.

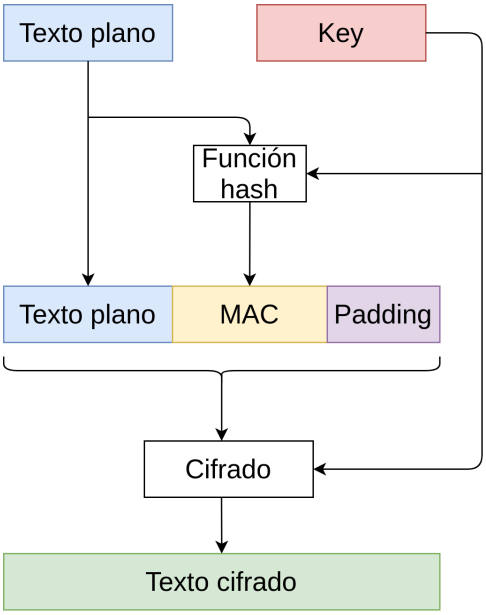

Figura 2: Estructura Mac-then-Encrypt

Como se muestra en la imagen anterior, los últimos bytes no ofrecen integridad en el mensaje enviado, allí es donde se encuentra la debilidad y donde podemos romper con la confidencialidad de dicho mensaje. Sin embargo para explotar el ataque es necesario comprender dos aspectos importantes.

- Padding

  Mediante el `Padding`, se comprueba el último byte para conocer los bytes que existen de relleno, es decir este byte indica la longitud del relleno en número de bytes, dicho número debe rondar entre `00` y `L-1` siendo `L` la longitud del bloque (pudiendo ser 8 o 16), cuando el servidor descifra el mensaje y lee este último byte, ignora los bytes de relleno que existen en el mensaje y comienza a comprobar la MAC del mensaje para asegurar la integridad del mensaje, por lo que no verifica que los bytes de relleno del mensaje pueda contener información (que en un principio no debería).

  Entonces, es posible que exista un bloque que únicamente tenga padding, si el bloque es de 16 bytes el valor del último byte del bloque debe de ser 15.

- Padding Oracle

  El `Padding Oracle` es una situación en el que el atacante conoce o puede adivinar por qué los datos que se han enviado al servidor son rechazados, si es debido a que el padding es incorrecto o la MAC.

4 Anatomía del ataque

La idea del ataque es conseguir que el último bloque se encuentre relleno de padding, para que, copiando el bloque de información sensible que queremos descifrar al último bloque que inicialmente se encuentra como hemos dicho relleno de padding podamos descifrar el último byte si el servidor no devuelve error (*Padding Oracle*), para conseguir los demás bytes en texto plano desplazamos el mensaje, esto se hace de la siguiente manera:

1. Acceso a la red

   Como se ha comentado el atacante debe conseguir realizar un person-in-the-middle para poder realizar un downgrade de la negociación entre el navegador del cliente con el servidor para forzar a utilizar el cifrado SSL 3.0.

2. Ejecución del JavaScript

   El atacante debe conseguir insertar un código malicioso JavaScript en el navegador del cliente para que pueda realizar múltiples solicitudes al servidor, de manera que añada caracteres extra para conseguir que exista un bloque solamente de relleno, como se ha comentado en la sección 3.

3. Conocimiento del bloque con información sensible

   El atacante debe conocer en qué número de bloque contiene la información sensible (cookies, contraseña), por ejemplo en el segundo bloque.

4. Copiado del bloque sensible al último bloque

   En este caso se copia el segundo bloque cifrado al último bloque de todo el mensaje cifrado, siendo la estructura del mensaje legítimo la que se muestra en la imagen 3, con 20 bytes de MAC y un bloque entero de padding (que se ha modificado por el JavaScript incrustado por el navegador).

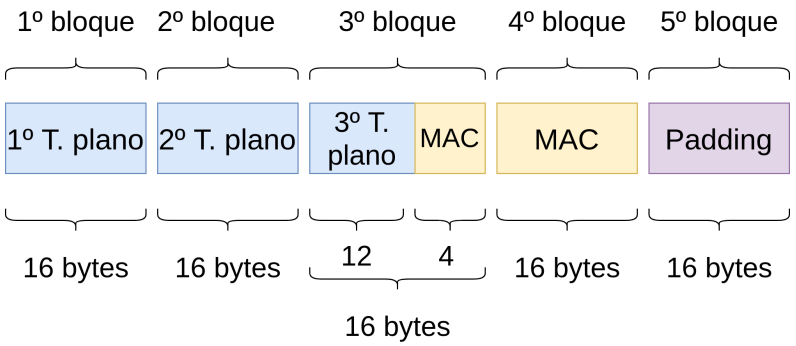

Figura 3: Estructura de un envío legítimo

Para posteriormente cifrarlo como aparece en la figura 4, como se muestra, un cifrado de bloque ocasiona un mensaje cifrado del mismo tamaño que el texto plano.

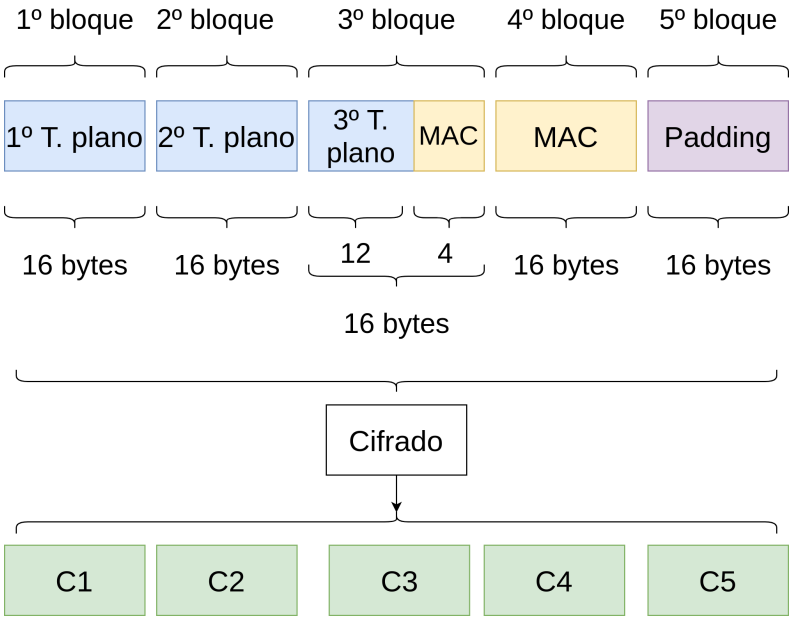

Figura 4: Envío de un texto cifrado legítimo

Sabiendo que el bloque 2 contiene un bloque sensible (cookies, contraseñas, etc), el código JavaScript copia el segundo bloque cifrado hacia el último (donde legítimamente contendría padding), como se muestra en la imagen 5

Figura 5: Mensaje cifrado manipulado

5. Respuesta del servidor

   Para que la solicitud se haya realizado con éxito, el servidor no debe de responder con ningún mensaje de error (Padding Oracle), para ello es necesario que el último byte contenga el valor '1 5' para que el servidor se crea que es un bloque entero de padding.

   - Si el valor del último byte es 16 o mayor, el servidor al recibirlo contestará con un mensaje de error debido a un padding incorrecto (Si el tamaño del bloque de 16, solo espera recibir un padding entre

0 y 14 bytes más el byte que define el tamaño del padding; en total 0 y 15).

- Si el valor del último byte es 14 o menor, el servidor no descartará leer los bytes de padding correspondientes y comenzará a leer la MAC con los primeros bytes erróneos.
- Por ello es necesario que el último byte del último bloque sea 15. Si el atacante realiza la misma petición, el texto cifrado será aleatorizado debido a los vectores de inicialización IV, por lo que la probabilidad de que el byte obtenga el valor '1 5', 0001 0101 en binario es de $2^8 = 256$ posibilidades, por lo que de media se tardarán 256 peticiones.

6. Descifrado

   Si el servidor no devuelve un error significa que se ha descifrado el texto y el último byte del último bloque tiene el valor 15.

   Como se ha comentado anteriormente, la ecuación para descifrar en modo CBC es la siguiente:

   $$P_i = D_k(C_i) \oplus C_{i-1}$$

   El atacante puede averiguar el último byte del texto plano que corresponde al segundo bloque siguiendo la siguiente ecuación:

   $$D_k(C_i)[15] \oplus C_{n-1}[15] = 15 \Longrightarrow D_k(C_i)[15] = 15 \oplus C_{n-1}[15]$$

   Siendo i el número de bloque copiado, en este caso 2, y n la posición del último bloque del mensaje, en este caso 5:

   $$D_k(C_2)[15] \oplus C_4[15] = 15 \Longrightarrow D_k(C_2)[15] = 15 \oplus C_4[15]$$

   Por lo que despejando P podemos hallarlo como:

   $$P_i[15] = 15 \oplus C_{n-1}[15] \oplus C_{i-1}[15]$$

   Volviendo a sustituir i por el valor del bloque, 2:

   $$P_2[15] = 15 \oplus C_4[15] \oplus C_1[15]$$

   Y como conocemos todos los valores de la derecha es posible saber el texto plano del último byte del segundo bloque (el que contiene la cookie).

7. Repetición del proceso rotando

   El atacante procede a averiguar el próximo byte realizando la conexión un byte más larga, para que se desplace el siguiente byte del segundo bloque, y así poder repetir el proceso anterior.

Bibliografía

[1] B. Möller, T. Duong, and K. Kotowicz, "This POODLE Bites: Exploiting The SSL 3.0 Fallback," p. 4.

[2] "Modos de operación de una unidad de cifrado por bloques," Apr. 2020, page Version ID: 125126320. [Online]. Available: https://es.wikipedia.org/w/index.php?title=Modos_de_operaci%C3%B3n_de_una_unidad_de_cifrado_por_bloques&oldid=125126320

[3] "What Is the POODLE Attack?" Jun. 2020. [Online]. Available: https://www.acunetix.com/blog/web-security-zone/what-is-poodle-attack/