Humanidades, Tecnología y Ciencia, del Instituto Politécnico Nacional
Certificado de Reserva de Derechos al Uso Exclusivo del Título: 04-2010-0326124-14000-203 ISSN: 2007-1957

DESARROLLO DE UNA EXTENSIÓN PARA LA AUTENTICACIÓN USANDO CHAFFING AND WINNOWING SOBRE HTTP

Verónica Agustín Domínguez
Instituto Politécnico Nacional
vagustin@ipn.mx

Axel Ernesto Moreno Cervantes
Instituto Politécnico Nacional
axelernesto@gmail.com

Elia Tzindejhé Ramírez Martínez
Instituto Politécnico Nacional
etramirez@ipn.mx

Abstract

En este trabajo se presenta el desarrollo de un mecanismo de autenticación basado en la técnica de Chaffing and Winnowing en una aplicación web con inicio de sesión por contraseña haciendo uso de certificados digitales expedidos por una autoridad certificadora en un servidor autenticador, una extensión de Google Chrome del lado del usuario y una API del lado del servicio web para lograr que los usuarios puedan autenticarse de forma automática y segura.

Palabras clave: Método de autenticación, Chaffing and Winnowing, Extensiones de Google Chrome.

Hasta el año 2017 la autenticación por contraseña era la más utilizada en los servicios web por su facilidad de implementación, mantenimiento y usabilidad para el usuario. Sin embargo, a pesar de estos beneficios, no es tan segura como otros métodos y requiere la memorización de las contraseñas (Komarova & Menshchikov, 2017). Con el uso de contraseñas para autenticación han surgido herramientas para administración de éstas o el almacenado, directamente, en dispositivos físicos, lo que expone a los usuarios a perdidas de datos sensibles, robo de identidad o incluso robo de cuentas bancarias (Espinoza Madrigal, 2018). Por esta razón en el presente documento se propone una alternativa de autenticación rápida y segura para las aplicaciones web con autenticación por contraseña, para que los usuarios puedan acceder sin la necesidad de utilizar herramientas como las mencionadas anteriormente.

Chaffing and Winnowing

Chaffing and Winnowing es una técnica que logra confidencialidad sin usar ningún proceso de cifrado para el envío de datos sobre un canal inseguro. Fue creada por Ron Riviest y

Humanidades, Tecnología y Ciencia, del Instituto Politécnico Nacional
Certificado de Reserva de Derechos al Uso Exclusivo del Título: 04-2010-0326124-14000-203 ISSN: 2007-1957

publicada en un artículo en línea el 18 de marzo de 1998 (Rivest, 1998).

El objetivo de *Chaffing and Winnowing* es proporcionar privacidad en un entorno simétrico, desde un punto de vista de seguridad, este esquema debe tratarse simplemente como un esquema de cifrado simétrico.

En los procesos de cifrado se toma un mensaje en claro y al cifrarse se obtiene el mensaje cifrado, después el proceso de descifrado toma el mensaje cifrado y recupera el mensaje en claro, ambos procesos utilizan una llave secreta en común.

En el esquema *Chaffing and Winnowing* la "clave" de cifrado es el código de autenticación de mensaje (MAC). Este proceso no se implementa de manera habitual, pero debe existir o de lo contrario no se lograría la privacidad (Goldwasser & Micali, 1984) (Bellare & Boldyreva, 2000).

Rivest propone un esquema que consta de 3 partes principales: autenticación, chaffing y winnowing.

El proceso de autenticación consiste en descomponer el mensaje original en paquetes más pequeños y completar cada paquete con un código de autenticación de mensaje (MAC).

Chaffing consiste en agregar paquetes inválidos (*Chaff packets*).

Winnowing es el proceso de remover los paquetes inválidos para obtener el mensaje original en el texto plano.

En la Figura 1, se muestra el diagrama general del proceso completo de *Chaffing* and *Winnowing.*

Metodología

La metodología de desarrollo usada para este sistema fue la metodología de prototipos evolutivos con unas pequeñas adaptaciones a fin de acortar el tiempo de desarrollo de éste. La metodología de desarrollo usada tiene la ventaja de permitirnos realizar la implementación del sistema aún cuando los requerimientos no están completamente definidos. Del mismo modo nos brinda la posibilidad de utilizar fragmentos de programas existentes o aplicar herramientas que nos permitan generar rápidamente proyectos que funcionen y puedan evolucionar. En la figura 2 se muestra el ciclo de vida de esta metodología.

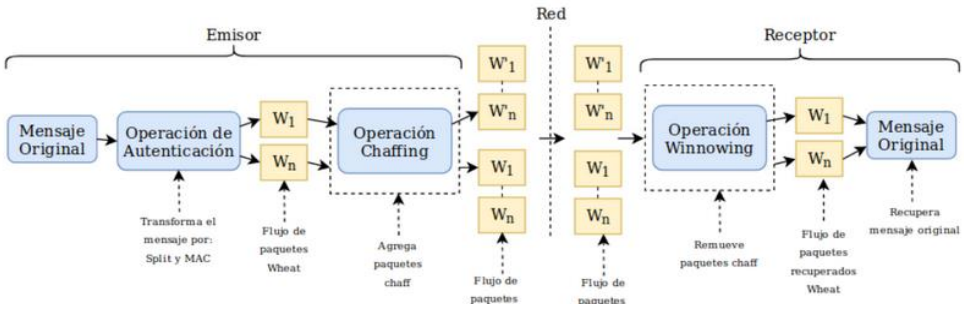

Figura 1 Diagrama general del proceso Chaffing and Winnowing

Humanidades, Tecnología y Ciencia, del Instituto Politécnico Nacional
Certificado de Reserva de Derechos al Uso Exclusivo del Título: 04-2010-0326124-14000-203 ISSN: 2007-1957

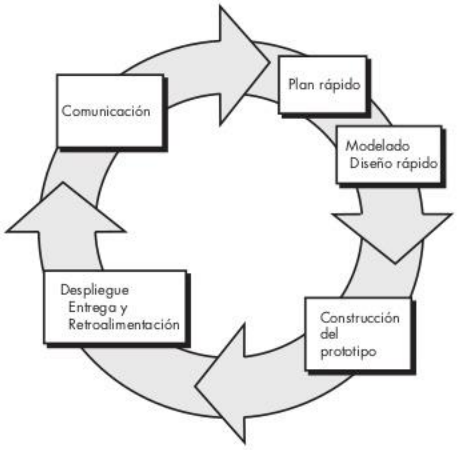

Figura 2. Metodología por prototipos evolutivos.

De manera breve se describirán las etapas de la metodología que se implementaron:

- Comunicación y plan rápido: se reunieron los requerimientos del prototipo, investigación sobre el desarrollo de extensiones para Chrome, sobre el algoritmo *Chaffing and Winnowing*, sobre Autoridades certificadoras, certificados digitales y sobre la implementación de un servidor Apache para su modificación.
- Modelado y diseño rápido: Modelado del algoritmo *Chaffing* y generación de patrones pasados en él, implementación de un módulo para lectura de peticiones GET de HTTP, implementación de una base de datos para almacenar certificados digitales y diseño de la arquitectura del sistema.
- Construcción del prototipo: implementación de una extensión para interpretar peticiones HTTP, implementación de una autoridad certificadora, implementación de un API para identificar tipos de peticiones HTTP.
- Despliegue, entrega y retroalimentación: Aquí se llevaron a cabo diversas pruebas unitarias para validar la funcionalidad del sistema.

El sistema propuesto se compone de 3 componentes que se comunican vía red.

1. Navegador Chrome con la extensión instalada
2. Servidor autentificador
3. Servidor web con API instalada

El usuario interactúa directamente con el primer componente, que consiste en la extensión creada y el navegador web que el usuario utiliza para hacer peticiones a diferentes servicios.

El segundo componente se encarga de generar certificados para cada usuario que se registre en la extensión. Con ayuda de una autoridad certificadora (AC) se garantiza la seguridad de los certificados.

El tercer componente es un servicio web con una Interfaz de Programación de Aplicaciones (API), que se encarga de reconocer las peticiones que se reciben con el método de autenticación propuesto, interpretar los datos y facilitar la información de autenticación al servicio.

La comunicación entre cada uno de los componentes se realiza mediante técnicas que permiten la confidencialidad de los datos.

La comunicación entre la extensión y el servidor autenticador se realiza a través de un socket seguro.

Para enviar el certificado, que viaja entre la extensión y el servicio web, éste se oculta mediante *Chaffing and Windowing*.

A continuación, se describe con mayor detalle el rol de cada componente en el sistema propuesto.

Componente 1: Extensión

La extensión en el navegador Google Chrome tiene la capacidad de interceptar las peticiones hechas por el usuario a través del navegador con el propósito de modificar dicha petición. La modificación se realiza siempre y cuando la extensión esté habilitada.

La modificación de la petición consiste en inyectar el certificado autenticador en el encabezado del protocolo, después de que se haya llevado acabo el proceso de *Chaffing*. Con el certificado incrustado en la petición, ésta es liberada para que salga a la red.

El certificado es único para cada usuario y se obtiene del componente 2 cuando el usuario inicia sesión en la extensión o bien cuando se registra, en caso de no tener una cuenta.

Cuando el usuario cierre sesión en la extensión, se eliminará el certificado de la maquina local del usuario.

Componente 2: Servidor Autentificador

En este componente se implementa un servidor autenticador en el que se crean y almacenan los certificados de los usuarios. Para acceder a este componente, los usuarios necesitan registrarse a través de la extensión en el componente 1.

La principal función de este componente es gestionar las cuentas y certificados en una base de datos, de tal forma que el componente 1 pueda comunicarse con éste, ya sea para generar un certificado a un nuevo usuario o para obtener el certificado de un usuario existente.

También tiene la capacidad de comunicarse con la API en el componente 3, para controlar la revocación de certificados.

Para la creación del certificado se utiliza la biblioteca OpenSSL, además, la comunicación entre extensión y servidor se hace bajo SSL/TSL.

Componente 3: API

En este componente se utiliza un servidor web de prueba para probar la funcionalidad de inicio de sesión con el método descrito en este trabajo.

La API realiza el proceso de *Winnowing* a la petición Http para obtener el certificado, así mismo, tiene comunicación con el componente 2 para verificar el estatus y validez del certificado.

Pruebas

Prototipo 1

Las pruebas se realizaron en una red local con un celular Huawei P20 cuya velocidad de conexión es de 70 Mb/s.

Para medir el tiempo de ejecución del proceso *Chaffing* se utilizó la función de JavaScript performance.now(), que mide el tiempo con una precisión en milisegundos, se obtuvo un tiempo total de 424.2925 ms, esta prueba incluyo las siguientes etapas: creación de patrón, proceso de *Chaffing* y cifrado de patrón.

Para el proceso *Winnowing* se utilizó la función de Java System.currentTimeMilis(), que también mide el tiempo con una precisión en milisegundos, se obtuvo un tiempo de 53.99 ms. La prueba incluyo la etapa de descifrado del patrón y el proceso de *Winnowing*.

Humanidades, Tecnología y Ciencia, del Instituto Politécnico Nacional
Certificado de Reserva de Derechos al Uso Exclusivo del Título: 04-2010-0326124-14000-203 ISSN: 2007-1957

Se realizaron un total de 100 pruebas del inicio de sesión a través de la solución propuesta. Se midió el tiempo que transcurre desde que el usuario da click en el botón de iniciar sesión en el servicio web, pasando por la intercepción de la petición por el componente 1, hasta la impresión de la respuesta del servicio web. Para medir el tiempo se utilizó la función de JavaScript llamada performance.now(), al igual que en la medición del tiempo en el proceso *Chaffing*. El tiempo promedio de los 100 inicios de sesión fue de 1101.68 ms.

Prototipo 2.

Las pruebas se realizaron en una red local creada con un celular Huawei P20 cuya velocidad de conexión es de 76 Mb/s.

Para el algoritmo *Chaffing*, que incluye las mismas etapas que las pruebas en el prototipo 1; creación del patrón, proceso de *Chaffing* y cifrado del patrón y utilizando la misma función de JavaScript, se obtuvo un tiempo de ejecución de 17.43 ms. La mejora respecto al tiempo obtenido en el prototipo 1 es de 416.8625 ms, esto se logro al disminuir el tamaño del patrón (que afecta directamente a la manera en que se crea y en que se recorre para hacer el proceso de *Chaffing*) y a la eliminación del cifrado por bloques AES.

Para el algoritmo *Winnowing*, midiendo el tiempo en el mismo proceso y etapas con la misma función de java, se obtuvo un tiempo de ejecución de 12.46 ms. La mejora obtenida respecto al prototipo 1 es de 41.53 ms, que se logró de la misma forma al disminuir el tamaño del patrón y a la eliminación del descifrado por bloques AES.

También se realizaron un total de 100 pruebas de inicio de sesión. El tiempo promedio de estas pruebas fue de 335.28 ms. La mejora obtenida respecto con el primer prototipo es de 766.4 ms, con lo que se comprueba que al disminuir el tamaño del patrón y eliminar el proceso de cifrado/descifrado por bloques AES, se acelera el proceso de autenticación con la solución propuesta.

Resultados

Tiempos de ejecución:

- **Algoritmo *Chaffing***: incluye la creación de patrón, proceso de chaffing y cifrado de patrón, el tiempo de ejecución fue de 17.43 ms.
- **Algoritmo *Winnowing***: incluye descifrado del patrón y proceso de winnowing. El tiempo de ejecución fue de 12.46 ms.
- **Inicio de sesión**: incluye el tiempo que transcurre desde que el usuario da click en el botón de iniciar sesión en el Servicio Web, es decir, la intercepción de la petición por parte del Componente I) hasta la impresión de la respuesta del ServicioWeb. El tiempo de ejecución fue de 335.28 ms.

Conclusiones

Gracias a que el API permitió conectar softwares, nos proporcionó un medio para la comunicación entre servicio web y autoridad autenticadora y por ende, llevar a cabo el proceso de *winnowing*. Por otro lado y gracias a la metodología de desarrollo de software utilizada fue posible terminar el desarrollo del prototipo en menos de un año que se tenía estimado realizarlo e incluso reducir el tamaño del patrón usado en el algoritmo chaffing que se manda hacia el servicio web

para mejorar el tiempo de ejecución de la aplicación.

También concluimos que no fue posible evitar por completo el uso de autenticación por usuario y contraseña. Pero se logró desarrollar una alternativa al inicio de sesión de los servicios web que deseen implementar este método de autenticación.

Referencias

Bellare, M., & Boldyreva, A. (2000). The Security of Chaffing and Winnowind. Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (págs. 517-530). Heidelberg, Berlin: Springer.

Espinoza Madrigal, C. C. (23 de Enero de 2018). DGTIC UNAM. Obtenido de Robo de indentidad y consecuencia sociales : https://www.seguridad.unam.mx/robo-de-identidad-y-consecuencias-sociales

Goldwasser, S., & Micali, S. (1984). Probabilistic encryption. Journal of Computer and System Sciences, 270-299.

Komarova, A., & Menshchikov, A. (2017). Comparison of Authentication Methods on Web Resources. Varna, Bulgaria: Springer.

Rivest, R. L. (1998). Chaffinf and Winnowing: Confidentiality without Encryption. Cryptobytes, 12-17.