

¿La banca online es vulnerable? (5ª Parte)

Banca Electrónica Nuevos Vectores de Ataque. 5º Parte.

RECOMENDACIONES FINALES

A la finalización de este informe, se hizo patente la necesidad de incluir en él un apartado con recomendaciones para solucionar los problemas que aquí se plantean. Si bien el objetivo de este documento no era otro que el de llevar al interesado en el tema a una reflexión sobre la seguridad de los sistemas de acreditación de banca online, sin caer en el tópico de facilitar un "recetario" de soluciones, parece interesante dedicar un breve espacio a presentar algunas propuestas sobre posibles soluciones.

Tal y como se desprende del presente informe, el principal problema de seguridad de las aplicaciones web es su propia "naturaleza", es decir, la falta de seguridad en el propio protocolo de comunicaciones (HTTP).

Si como parece que sucede en la actualidad, no hay más remedio que hacer uso de aplicaciones basadas en web en las operaciones de banca electrónica, para evitar los ataques descritos en este documento, de algún modo podrían emplearse las siguientes técnicas:

Usar páginas estáticas en los formularios de firma electrónica (2ª validación). Esto significa que en dichas páginas no se debería generar dinámicamente ningún contenido. Ello y a pesar de que en principio los datos "dinámicos" no dependen directamente de la entrada de datos del usuario, nunca se sabe.

Aparentemente, aplicar la recomendación anterior parece inviable, pues en los formularios de 2ª validación (firma) se deben reflejar todos los datos de la transacción que va a validar el usuario, y esa información es por definición, dependiente del propio usuario... Sin embargo, el problema de las partes de "información" que se generan dinámicamente no son el obstáculo en si mismo, sino el método de representación. De alguna manera en la mayoría de casos estamos ofreciendo al usuario una interfaz de "generación de código". Si, tal y como suena, porque al fin y al cabo, por muy filtrados que estén los datos, tanto a la entrada del usuario como a la salida cuando se renderiza la página, siempre existe la posibilidad de que los mecanismos de filtrado no sean efectivos al 100% y al final el atacante pueda inyectar su código.

Pero veamos ¿Lo que interesa es mostrar una información determinada para su confirmación? ¿Importa la manera de presentar esa información? ¿Porqué no presentarla como una imagen, por ejemplo? La idea sería que el formulario de firma (2ª validación) presentara todos los datos dinámicos en formato imagen (GIF, PNG, JPEG, etc.). Si no se genera código, sino una imagen, lo peor que puede suceder si el atacante consigue inyectar código maligno- es que éste aparezca en la imagen...

Esto sería realmente frustrante para el intruso ¿O no? Alguien podría pensar que este método puede servir como práctica general para evitar la inyección de código... No se piensa así y por una razón muy sencilla: Por la carga de CPU que debería soportar el servidor, para webs de alto contenido dinámico... y más aún, por la dificultad de integración con todas las aplicaciones.

En cualquier caso, para la situación concreta de aquellos formularios que requieran de firma (2ª validación), esta solución sí que se presenta como una opción a estudiar.

Si se quiere seguir utilizando el tradicional sistema de renderizado de páginas dinámicas que basan su seguridad en el filtrado de entrada de datos de usuario, no existe nada más original que lo que cualquier analista recomendaría: utilizar una metodología de programación segura. Otra opción son los cortafuegos de aplicación, aunque esto no siempre es posible por distintas razones: Coste, implementación y un mantenimiento que no está al alcance de todas las empresas.

Existen otros detalles, que van más allá de la programación segura. Son pequeños "trucos" o prácticas que pueden dificultar la labor intrusiva:

-Evitar referencias relativas y usar referencias absolutas. Esto hace la web menos portable, pero mas segura ya que evita los ataques basados en inyección de tag "base href".

-En los formularios, no realizar el POST a una URL que dependa de una variable. Si hace esto se expone a que una inyección de código modifique el valor de esa variable y se produzca un ataque de desvío de POST.

-Aunque parezca absurdo y cuando sea posible, definir las variables "delicadas" al principio y al final de la página. Sí, repetidas. Esto no evitaría una inyección y modificación de variable, pues el atacante aun puede redefinir las variables y "comentar" el resto de código. Pero en muchísimas ocasiones le complicaría la intrusión.

-Teclados virtuales: Utilizar un sistema del que no puedan obtenerse patrones. El teclado debe cambiar de posición y de tamaño cada vez que se invoca, las teclas deben cambiar de posición y de tamaño con cada invocación del mismo.

Se debería poner a disposición del usuario algún sistema "generador de caos", es decir un sistema por el cual el usuario pueda efectuar clics de ratón antes, durante o después de introducir su contraseña en el teclado virtual, y por tanto que dificulten la localización de coordenadas "válidas". Si utiliza teclado virtual con contraseñas estáticas, hay que tener especial cuidado de no enviar la contraseña en claro... Aunque utilice SSL. Es muy simple, hay una máxima en la explotación de vulnerabilidades de inyección de código en aplicaciones web: toda la información que es accesible para un script cargado desde la página atacada, lo es para el atacante. Consecuentemente, si la página objeto de estudio contiene un script que realiza una "traducción" de coordenadas a clave, el atacante puede que ya no necesite descodificar las coordenadas, sino simplemente acceder al valor de la variable que contiene esa clave que se va a enviar. ¿Cómo solucionar este problema?

Una manera, sería utilizar un sistema sincronizado entre el servidor y el cliente, de manera tal que el cliente enviara al servidor las coordenadas más un token que identifica al teclado generado, con el objeto de que el servidor pueda resolver las coordenadas para ese teclado en particular. El modo más sencillo de llevar a cabo ese cometido, es generando el teclado en el servidor y con formato de imagen. De esta manera el cliente no tendría "responsabilidad" alguna en la generación de dicho teclado, solo cargaría una imagen del tipo:

"teclado.5tyfghtysdr94r.jpeg", la cual contendría una serie de teclas dispersas aleatoriamente por toda su superficie. Entonces los clics del cliente serían capturados por el script cliente, y enviados al servidor de banca. Y lo que recibiría el servidor de banca es una serie de coordenadas que solo

tienen sentido para el teclado específico que ha generado, es decir, en nuestro ejemplo, el que se asocia al token: "5tyfghtysdr94r". Un atacante que consiguiera desviar el POST como en los ataques mostrados anteriormente, obtendría unas coordenadas y un token sin ningún sentido para él. Evidentemente, el "eslabón débil" de este esquema es la imagen que contiene el teclado, pues permite junto a las coordenadas en caso de capturarse descifrar la clave. Sin embargo no parece existir, un modo sencillo de forzar al usuario a "enviar" dicha imagen al atacante, y siempre pueden reducirse los posibles ataques a este esquema mediante técnicas cuya explicación no es el objeto de este informe.

Como puede observarse, a pesar de seguir siendo una serie de soluciones "imperfectas" son posibilidades originales, sencillas y elegantes que dotan a los sistemas de banca online de medidas adicionales de protección que dificultan en gran medida los ataques incluso de los intrusos mas decididos.

CONCLUSIONES

No se ha realizado un estudio en detalle para otro tipo de sistemas de autenticación de banca on-line, debido a la falta de un estándar para el resto de sistemas de autenticación de firma.

La conclusión mas inmediata que se desprende del presente informe es que los sistemas de firma de banca online, ya sea basados en "tokens" físicos, teclados virtuales, u otros, actualmente distan mucho de poder considerarse como soluciones robustas pues prácticamente todos adolecen del mismo fallo: que su seguridad se implementa -en gran medida- a través de un protocolo, http, que jamás se diseñó para ser seguro, sino solo funcional.

Hasta que llegue el día en que los mecanismos de verificación de la identidad del usuario de una aplicación web sean robustos, la banca on-line estará expuesta a toda serie de riesgos. ¿Se pueden eliminar completamente dichos riesgos? No es probable, pero si que se pueden reducir en gran medida mediante la realización de pruebas periódicas especializadas sobre los sistemas de seguridad empleados.

Si nos fijamos en la evolución de las metodologías de evaluación de la seguridad en los sistemas de información, observamos una tendencia a analizar cada vez con menor detalle las distintas tecnologías de protección. Ello es así porque a veces se confía ciegamente, en ciertos mecanismos clásicos, sin tener en cuenta que tan importante es un estudio de las particularidades de cada escenario concreto, como el del diseño en general y su específica implementación, y esto sea cual sea su origen y los Argumentos de Autoridad que lo defiendan.

[Informe 1ª parte del informe](#)

[Informe 2ª parte del informe](#)

[Informe 3ª parte del informe](#)

[Informe 4ª parte del informe](#)

[Informe 5ª parte del informe](#)

[Artículo completo](#)

Hugo Vázquez Caramés

(Director Técnico de PENTEST, Consultores de Seguridad Telemática)

<http://www.pentest.es>

Nota: La empresa española [PENTEST](#) tal vez sea una de las mejores del entorno europeo en la realización de Tests de Intrusión. Empresa que basa su éxito en la recluta para cada tipo de proyecto de los mejores "Pen-Testers" existentes en la problemática a auditar. Una vez seleccionados, forma un equipo de auditores o "Tiger Team" que pone al servicio del cliente. Normalmente los "Tiger Team" de [PENTEST](#) no solo son los mejores, sino también los más motivados pues trabajan a partir de la propia libertad de sus conocimientos y experiencia y de la que concede Pentest para el desarrollo de su función profesional.

Fecha artículo: 2006-06-20 20:00:00 - url artículo: <http://www.internautas.org/html/797.html>

Logos y marcas propiedad de sus respectivos autores.

Los comentarios son propiedad y responsabilidad de cada autor.

© 1998-2009 Asociación de Internautas - <http://www.internautas.org>

Inscrita en el Registro Nacional de Asociaciones con el número 164343 - e-mail: asociacion@internautas.org