

¿La banca online es vulnerable? (2ª Parte)

Banca Electrónica Nuevos Vectores de Ataque. 2º Parte.

VECTOR DE ATAQUE PRICIPAL Lógica del mecanismo de protección (2ª validación o firma)

Anteriormente hemos apuntado conceptualmente que es posible secuestrar una sesión http mediante técnicas de "Cross Site Scripting" o "Inyección de Código".

Para proseguir, supondremos que un atacante ya ha superado el primer nivel de validación de una entidad bancaria mediante el secuestro de una sesión, y por lo tanto tiene acceso a la misma sesión que su víctima.

Tal y como ya se ha descrito, para cada operación de riesgo se requiere una validación de 2º nivel o firma. Esta validación no establece un contexto de nueva sesión susceptible de ser secuestrada. En definitiva, si un atacante consigue saltarse el primer nivel mediante un secuestro de sesión, aun necesita conocer los datos de autenticación del 2º nivel, los cuales, para mayor complicación, en muchos casos, ni tan siquiera son estáticos, sino que pueden tener valores distintos en cada ocasión ("tokens", tarjetas de coordenadas, etc). Sin embargo una cosa si parece posible: si el atacante que ya tiene acceso a la sesión de su víctima es capaz de interceptar los datos de dicha 2ª validación o firma, para una transacción en concreto, y conoce el estado de la sesión de su víctima, en principio, no parece existir impedimento técnico para que el atacante pueda suplantar a la víctima en dicho paso de la transacción.

El "talón de Aquiles" del sistema de firma.

El proceso de secuestro de sesión http para burlar el primer nivel es bastante simple, solo se necesita que aparezca el típico fallo de programación por el cual no se valida la entrada de datos en algún formulario, o una inyección de código en cualquier página que pueda ser explotada externamente. Debido a que la aplicación misma realiza el seguimiento de sesión en toda la zona que requiere 1ª validación, un problema de "Inyección de Código" probablemente le permitiría a un atacante obtener sin problema el estado de la sesión. Sin embargo para burlar el segundo nivel solo existen ciertos puntos interesantes para el ataque de "Inyección de Código": las páginas que muestran el formulario para introducir la 2ª validación (firma). ¿Qué se conseguiría con la inyección de código en una de dichas páginas?

Pues, si dicha inyección de código es controlable por un atacante, éste puede modificar por completo el comportamiento de la página que solicita la 2ª validación (firma). El problema que aparece en este punto es cómo controlar externamente dicha inyección. Pongamos un ejemplo: supongamos que el atacante puede conocer el estado de la sesión de la víctima y por tanto, puede realizar peticiones validas suplantando su identidad. El atacante además, descubre que una de las páginas donde se pide 2ª validación (firma) es susceptible de inyección de código.

Consecuentemente, aquí se plantean dos casos:

Primero: La inyección se puede realizar externamente .

Definiremos como inyección externa a aquella que se puede realizar de alguna de las siguientes

maneras:

- Desde fuera de la aplicación misma
- Desde otra sesión válida (distinto usuario)

En este primer caso (inyección externa), el atacante ya tiene el elemento que le faltaba, por tanto puede modificar el comportamiento de dicha página, es decir inyectar un formulario falso, redireccionar la petición http, etc. El atacante está en condiciones de obtener el "token" y puede usarlo para realizar una transacción. Este caso no es muy común pero de cualquier modo, podemos afirmar que hablamos del "típico" problema de inyección de código.

Segundo: La inyección se puede realizar internamente, es decir se puede realizar la inyección, pero solo desde la misma sesión del usuario. Aunque parezca extraño, es un caso harto común en aplicaciones web. Son aquellos problemas de inyección de código, en los que solo el propio usuario puede llevar a cabo la inyección. Estos casos, usualmente se catalogan como no peligrosos, pues no parece que tenga sentido auto inyectarse código en la propia sesión.... Pues bien, veamos un escenario en el que este tipo de inyecciones pueden ser muy útiles para un intruso.

Inyecciones de código internas

Es precisamente en este contexto el que aparece un nuevo vector de ataque que hasta el momento no se había contemplado, pues hasta ahora, nadie encontraba sentido al hecho de poder realizar una inyección de código desde la propia sesión del usuario. ¿Por qué? Porque las aplicaciones web con zonas restringidas a usuarios registrados normalmente solo disponen de un nivel de validación seguido luego de un seguimiento de sesión convencional. Así pues el esfuerzo de los intrusos hasta el presente solo se centraba en poder controlar su inyección de código desde el exterior porque una vez conseguida la sesión del usuario ya tenían acceso a toda la aplicación. Es precisamente en el contexto de banca on-line -donde existe un segundo nivel de validación- en el que las inyecciones en la propia sesión cobran una importancia vital. Podrán comprobar los usuarios de banca on-line más curiosos, que generalmente, es posible inyectar código desde la propia sesión con relativa facilidad. ¿Por qué? A caso porque las auditorias de seguridad se centran, principalmente, en comprobar las aplicaciones de banca, desde la zona pública, más que desde la zona privada. No obstante y en principio, no existe impedimento para que el intruso cree una cuenta bancaria en una entidad, solicite acceso online y posteriormente estudie el comportamiento de la web, desde la zona de usuarios registrados...

NUEVO VECTOR DE ATAQUE

Como se ha acaba de apuntar, aparentemente no tiene sentido inyectar código en la propia sesión, excepto en algunos casos. Supongamos el siguiente escenario: un intruso ha sido capaz de secuestrar una sesión de un usuario válido. El intruso sabe de la existencia de un problema de inyección de código en la zona privada, es decir, una vez validado -esto puede obtenerlo tras haber realizado un estudio previo desde otra cuenta o en el mismo momento del secuestro, desde la propia sesión de la víctima-. Si el intruso está en la misma sesión que la víctima y puede autoinyectarse código ¿Se ejecutará dicho código en el navegador de la víctima?

Existen al menos dos casos:

- 1) La inyección es de tipo estático o permanente : es decir, una vez inyectado el código en una

determinada parte de la aplicación web, este código queda almacenado en una base de datos.

2) La inyección es de tipo dinámico o no permanente : o sea, la inyección tiene un tiempo de vida limitado y no se almacena al menos permanentemente en ningún sitio (es importante este matiz).

De acuerdo con lo comentado anteriormente, el caso más interesante para un atacante es poder inyectar código EXACTAMENTE en el paso del formulario donde se exige la 2ª autenticación (firma) para poder manipular el comportamiento de dicho formulario. No es común encontrar inyecciones de tipo estático o permanentes en ese paso del formulario pero de existir, el juego para el atacante prácticamente habría terminado.

El caso de inyecciones de tipo dinámico o no permanente , son menos frecuentes aún pero no por ello despreciables. Partiendo de que la página del formulario correspondiente a la segunda validación es dinámica, pues normalmente se deben reflejar los datos del usuario, su nombre, apellidos, cuentas, o datos del propio perfil (preferencias, etc), existe un caso que es especialmente curioso y delicado al mismo tiempo: estamos hablando de aquellos sistemas o aplicaciones que mantienen una conciencia del paso del formulario, o del paso de la sesión en el que se encuentra el usuario, de manera que impiden que éste pueda enviar mediante distintas peticiones, información distinta para el mismo paso de la sesión. El efecto que observará el usuario común es que en un determinado paso del formulario - donde realizamos el POST- no es posible utilizar el navegador para retroceder y modificar los datos del formulario. Esto significa que el usuario no puede modificar estos datos mediante la repetición del POST pues la aplicación web devuelve la misma respuesta que ya se obtuvo anteriormente para ese mismo paso.

Este comportamiento, que se ha bautizado como el efecto formulario tozudo , es muy peligroso por las siguientes razones:

- 1) Puede permitir ataques man-in-the-middle en sistemas OTP (One Time Password).
- 2) Puede permitir ataques man-in-the-middle en sistemas que usan "tokens" (RSA Secur-ID, etc).
- 3) En general, puede permitir ataques man-in-the-middle en sistemas de segunda validación (firma) como los usados en banca electrónica.
- 4) Puede permitir burlar los sistemas de seguimiento de sesión http más avanzados, basados en testigos o pseudo-números de secuencia que se intercambian en la capa de aplicación (http).
- 5) Es un nuevo vector de ataque y es, conceptualmente hablando, bastante simple de combinar con otros ataques ya conocidos.

Continúa:

[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](#)

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.

[Artículo completo](#)

Fecha artículo: 2006-06-20 20:00:04 - url artículo: <http://www.internautas.org/html/799.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