



CSRF Falsificando peticiones

Javier Garson
Desarrollador web
Abril 2016



OWASP

The Open Web Application Security Project



OWASP

The Open Web Application Security Project

Javier Garson Aparicio

Desarrollador web

- 4 años de experiencia en desarrollo web
- 2 años de experiencia haciendo pruebas de seguridad
- Aficionado por la electrónica
- Desarrollador frontend/backend JS
- Especializado en JS

lanstat@gmail

@lanstat

Fb.me/lanstat





OWASP

The Open Web Application Security Project

“CRSF (falsificación de petición en sitios cruzados) es una vulnerabilidad en un sitio web que permite a los atacantes forzar a las víctimas a ejecutar acciones sensibles en el sitio sin su conocimiento.”



OWASP

The Open Web Application Security Project

1. El objetivo es una operación sensible en la aplicación, ej. **UpdateSalary.php**, es posible falsificar la ejecución.
2. Las víctimas pueden ser **forzadas a ejecutar esta acción** a través de cualquier método para conseguir que se cargue un recurso automáticamente, ej. Tag img, script, acciones onload, submit, etc. Nota: las credenciales van en todas las peticiones!
3. Esto sucede **sin conocimiento de la víctima** porque las acciones son ejecutadas por el browser de la víctima, no por la víctima específicamente.



OWASP

The Open Web Application Security Project

Ejemplos de acciones sensibles:

- /EditDocument.php
- /Login.php
- /CreateAdmin.php
- /UpdateStatus/



OWASP

The Open Web Application Security Project

Forzando a la victima a ejecutar la acción (GET):

- `` (GET)



OWASP

The Open Web Application Security Project

Forzando a que la victima ejecute la acción (POST):

```
<html><head><title>BMC IDM Change PW CSRF PoC</title></head>

<body onload="document.getElementById('CSRF').submit()">

<form action="https://xxx.xxx.xxx.xxx/idm/password-manager/changePasswords.do";
method="post"
id="CSRF">

<input type="hidden" name="colChkbox_Tab1" value="CN=Test User,OU=User
Accounts,DC=corporate,DC=business,DC=com corporate Win2000" />
<input type="hidden" name="password" value="Abc123!" />
<input type="hidden" name="passwordAgain" value="Abc123!" />
<input type="hidden" name="selAccts" value="CN=user Name,OU=User
Accounts,DC=corporate,DC=business,DC=com corporate Win2000" />

</form></body></html>
```



OWASP

The Open Web Application Security Project

Ambos XSS y CSRF son capaces de abusar de conexiones confiables:

- En XSS el browser ejecutara JS malicioso **porque este puede ser generado por un sitio(origen) en el cual se confía.**



- En CSRF el servidor ejecuta acciones sensibles **porque este contenido fue enviado por un cliente que el servidor confía.**





OWASP

The Open Web Application Security Project

Validaciones

Criterios para identificar una
vulnerabilidad



OWASP

The Open Web Application Security Project

Si no se puede **cambiar** algo usando su vulnerabilidad CSRF, entonces usted no tiene una vulnerabilidad.

Ejemplos de cambios de estado:

- Actualizando una cuenta(nuevo password?)
- Transfiriendo fondos
- Cambiando los roles de un usuario
- Comprando un ítem
- Agregando un administrador al sistema



OWASP

The Open Web Application Security Project

Como verificar si existe un CSRF:

1. Configurar un proxy para observar el trafico
2. Iniciar sesión en el sitio a verificar
3. Ejecutar las funcionalidades del sitio, a través del browser
4. Observar las peticiones, buscando cambios de estado, sensibilidad o singularidad
5. Buscar por cualquier control adicional que pueda detener CSRF, como son CAPTCHA o autenticaciones adicionales
6. Cerrar sesión e iniciar sesión con un diferente set de credenciales
7. Solicitar las peticiones encontradas desde el nuevo contexto, y si se ejecutan exitosamente
8. Si la acción es ejecutada sin problemas, es casi seguro que es un CSRF
9. Recuerda que los problemas que los problemas pueden satisfacer los requerimientos de cambios de estado y sensibilidad. El unicidad no es suficiente.



OWASP

The Open Web Application Security Project

Métodos de defensa



OWASP

The Open Web Application Security Project

- La principal defensa contra CSRF es crear tokens de petición **únicos** que no puedan ser fácilmente generados por el atacante.
- Políticas de mismo origen
- Re autenticación
- CAPTCHAs puede ser también útil, para autenticaciones.



reCAPTCHA™

- WHAT IS reCAPTCHA
- GET reCAPTCHA
- PROTECT YOUR EMAIL
- MY ACCOUNT
- RESOURCES: DOCS & PLUGINS

reCAPTCHA IS A FREE ANTI-BOT SERVICE THAT HELPS DIGITIZE BOOKS.

steamboat train, from New
this **morning** ran off the track
New-London. Four cars plunged



→ LEARN HOW reCAPTCHA WORKS

USE reCAPTCHA ON YOUR SITE

-  **STRONG SECURITY**
-  **ACCESSIBLE TO BLIND USERS**
-  **30+ MILLION SERVED DAILY**

NEW See how accurate reCAPTCHA is at digitizing content!

Blog | About Us | Contact | FAQs | Terms
© 2013 Google, all rights reserved.



- **Solo aceptar peticiones POST**
 - Detiene ataques basados en links simples (IMG, frames, etc.)
 - Pero peticiones POST ocultas pueden ser creadas con frames, scripts, etc...
- **Obligar transacciones multi-pasos**
 - Los ataques CSRF cada paso en el orden requerido
- **Reescritura de URL**
 - General session id exposure in logs, cache, etc.

Ninguno de estos métodos funciona eficientemente contra ataques CSRF!



OWASP

The Open Web Application Security Project

Vectores de ataque



OWASP

The Open Web Application Security Project

- La clave para la defensa contra CSRF es que el atacante no tenga acceso a un token valido
- Pero con XSS presente el atacante puede forzar a la victima a hacer una petición al sitio, consumir el token y agregarlo a la petición CSRF



OWASP

The Open Web Application Security Project

- Usar inyección de código XSS persistente en un sitio vulnerable
- Crear un nuevo sitio internamente y atraer a los usuarios a visitar el sitio vía email, etc. (phishing)



OWASP

The Open Web Application Security Project

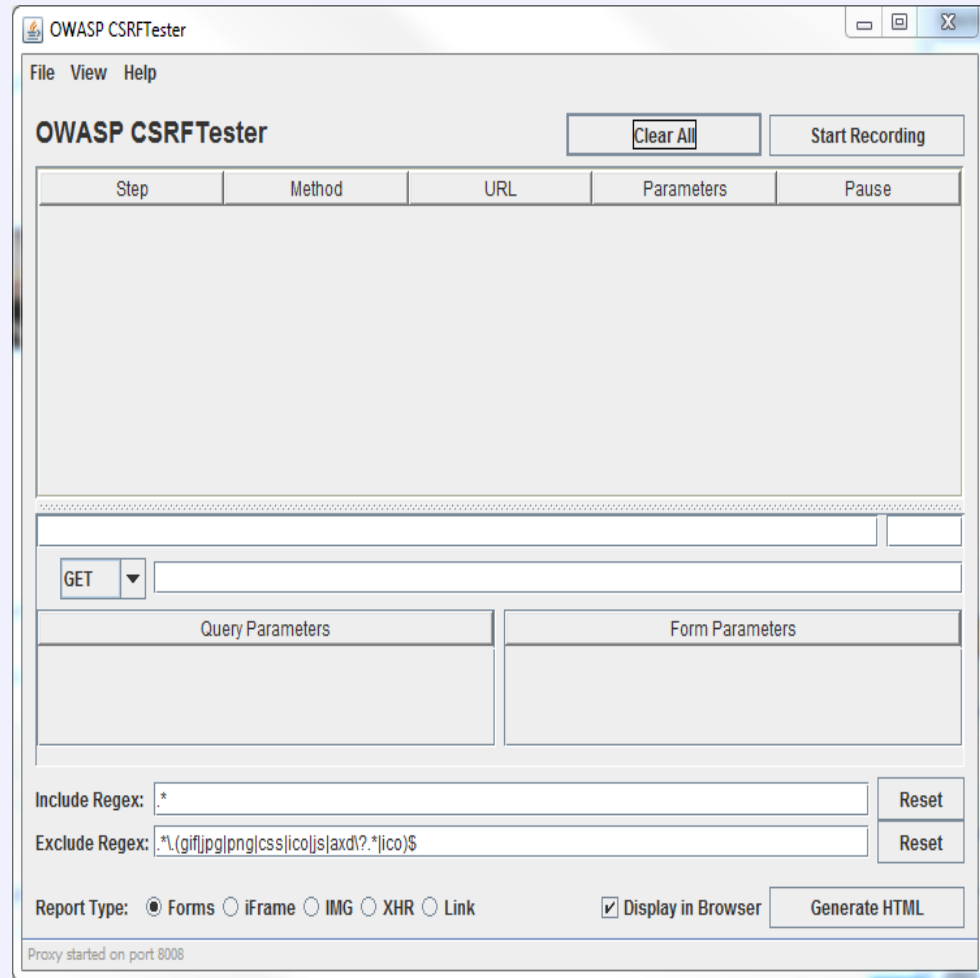
CSRF Tester



OWASP

The Open Web Application Security Project

- CSRF Tester es una herramienta de OWASP para crear código CSRF PoC
- Esta funciona capturando trafico sensible y después generando código PoC para puedas usar con otro contexto de usuario
- Escucha a través del puerto 8008





OWASP

The Open Web Application Security Project

- Envía tráfico a través de CSRF Tester como cualquier otro proxy
- Registra las acciones sensibles de un
- Entonces crea un reporte de un cierto tipo, Form, iFrame, IMG, XHR, Link
- Ese código es ahora un Poc para probar y ver si es un problema de CSRF

The screenshot shows the OWASP CSRFTester application window. At the top, there are 'File', 'View', and 'Help' menus. Below the menu is the title 'OWASP CSRFTester' and two buttons: 'Clear All' and 'Stop Recording'. A table lists several requests with columns for Step, Method, URL, Parameters, and Pause. Request 7 is highlighted in blue. Below the table, the details for Request 7 are shown, including the method (GET) and the URL (https://www.google.com.bo:443/gen_204). There are sections for Query Parameters and Form Parameters. At the bottom, there are fields for 'Include Regex' and 'Exclude Regex', and a 'Report Type' section with radio buttons for Forms, iFrame, IMG, XHR, and Link. A 'Generate HTML' button is also present.

Step	Method	URL	Parameters	Pause
Request 0	GET	http://gmail.com:80/		1360
Request 1	GET	http://google.com:80/		304
Request 2	GET	http://www.google.com...		303
Request 3	GET	https://www.google.co...		626
Request 4	GET	https://www.google.co...		156
Request 6	GET	https://www.google.co...		130
Request 7	GET	https://www.google.co...		148
Request 8	GET	https://www.google.co...		376

Request 7 details:

Method: GET
URL: https://www.google.com.bo:443/gen_204

Query Parameters:

- v=3
- s=webhp
- atyp=csi
- =lGIEV7vsCYaee_i5o9aF

Form Parameters:

Include Regex: *

Exclude Regex: *\.(gif|jpg|png|css|ico|js|axd|?.*|ico)\$

Report Type: Forms iFrame IMG XHR Link

Display in Browser



OWASP

The Open Web Application Security Project

Preguntas