CSRF Falsificando peticiones

Javier Garson
Desarrollador web
Abril 2016

OWASP
The Open Web Application Security Project

Acerca de mi

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 victimas 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 victimas 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 victima porque las acciones son ejecutadas por el browser de la victima, no por la victima específicamente.

Ejemplos de acciones sensibles:

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

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

- (GET)

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="colChkbx_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>
```

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

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 autentificaciones 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 autentificaciones.

CAPTCHA

OWASP
The Open Web Application Security Project

Defensas que no funcionan

OWASP
The Open Web Application Security Project

- Solo aceptar peticiones POST
  - Detiene ataques basados en links simples (IMG, frames, etc.)
  - Pero peticiones POST ocultas puedes 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

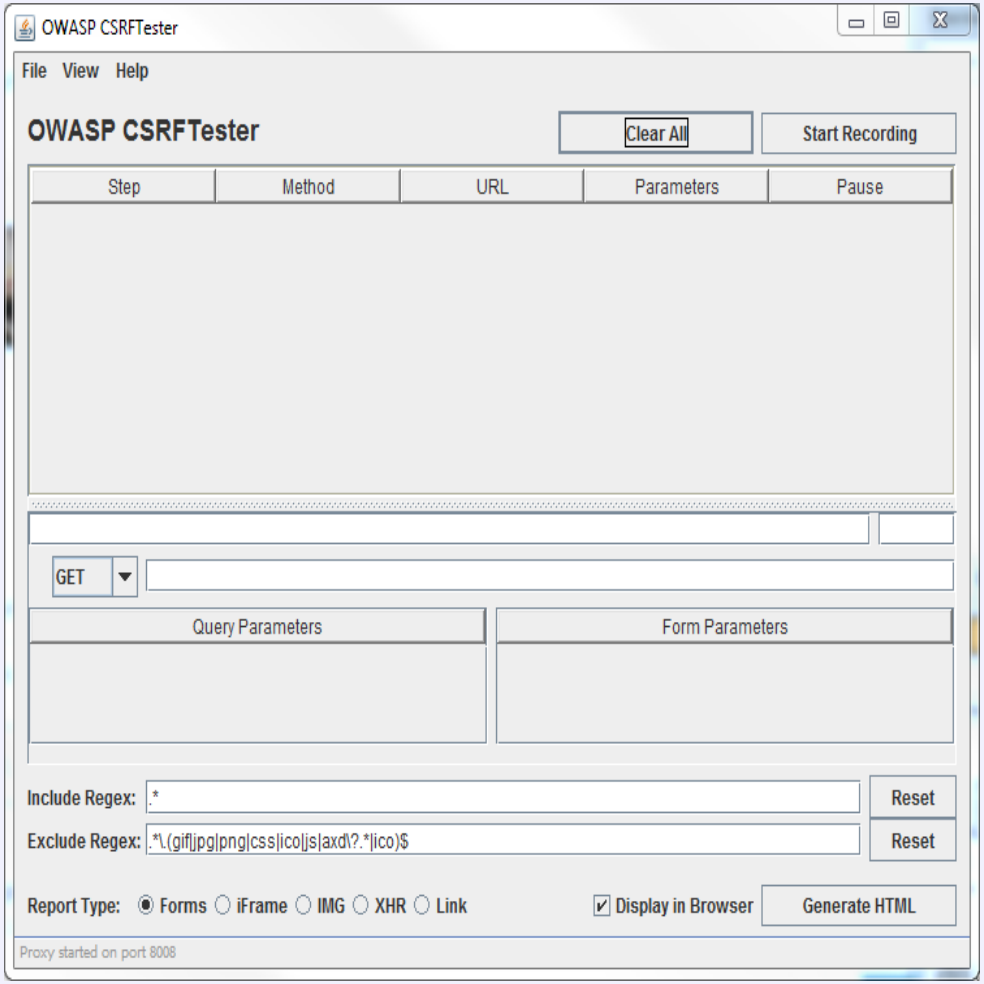
- Usar inyección de código XSS persistente en un sitio vulnerable

- Crear un nuevo sitio internamente y atraer a los usuarios a visitar el sitio vía email, etc. (phishing)

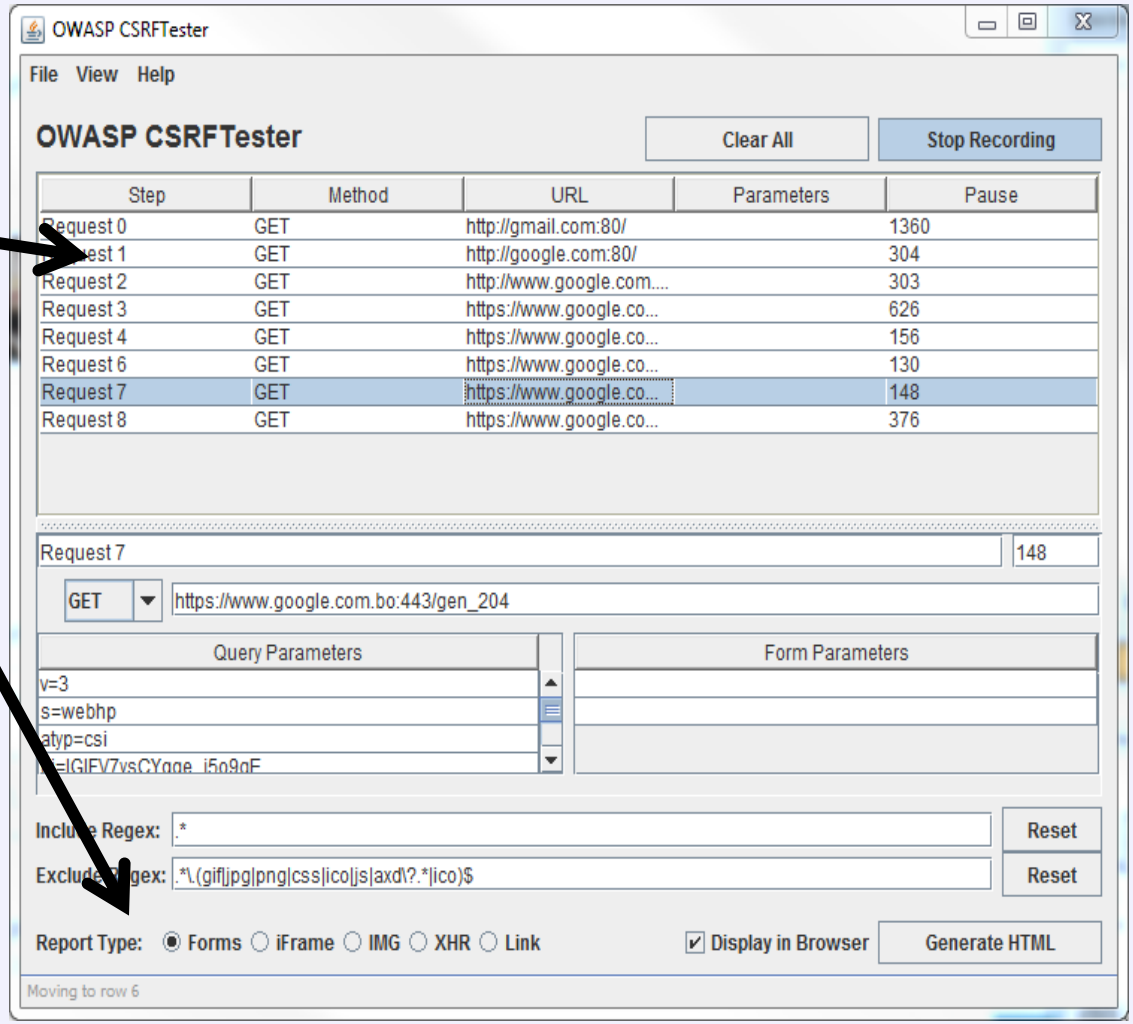
OWASP
The Open Web Application Security Project

CSRF Tester

- 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



- Envía trafico 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



OWASP
The Open Web Application Security Project

Preguntas