**Lab1-Report**

57118116 陈煜

Task 1：Observing HTTP Request.

在Elgg上截取的﻿HTTP GET请求如下：

Host: www.csrflabelgg.com

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0

Accept: \*/\*

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: http://www.csrflabelgg.com/

Cookie: Elgg=dlme29kju0s29tp16i3f0632j3

Connection: keep-alive

截取的HTTP POST请求如下：

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/admin/edit

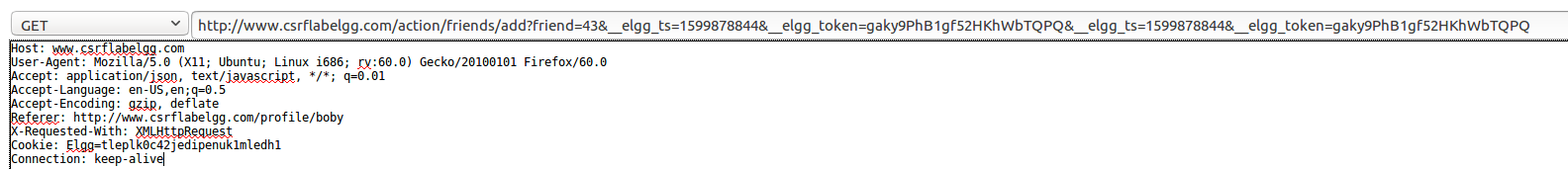
Cookie: Elgg=idpg165m6ngtjdltgmfo3l9hi5

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Task 2: CSRF Attacks using GET Request

进入Admin的账号，点击添加Boby为好友，捕获到的GET请求如下：



其中URL为：

﻿<http://www.csrflabelgg.com/action/friends/add?friend=43&__elgg_ts=1599878844&__elgg_token=gaky9PhB1gf52HKhWbTQPQ&__elgg_ts=1599878844&__elgg_token=gaky9PhB1gf52HKhWbTQPQ>

其他信息为：

Host: www.csrflabelgg.com

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0

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=ir5ar3ol02c6029gm0k3kqqa77

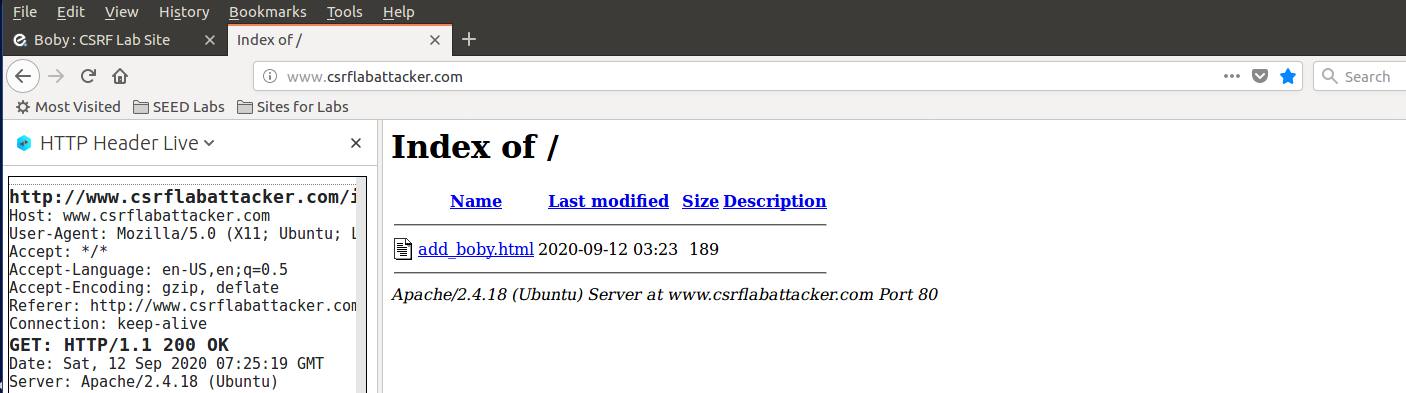
Connection: keep-alive

通过URL可以得知Boby的friend对应的参数为friend=43。

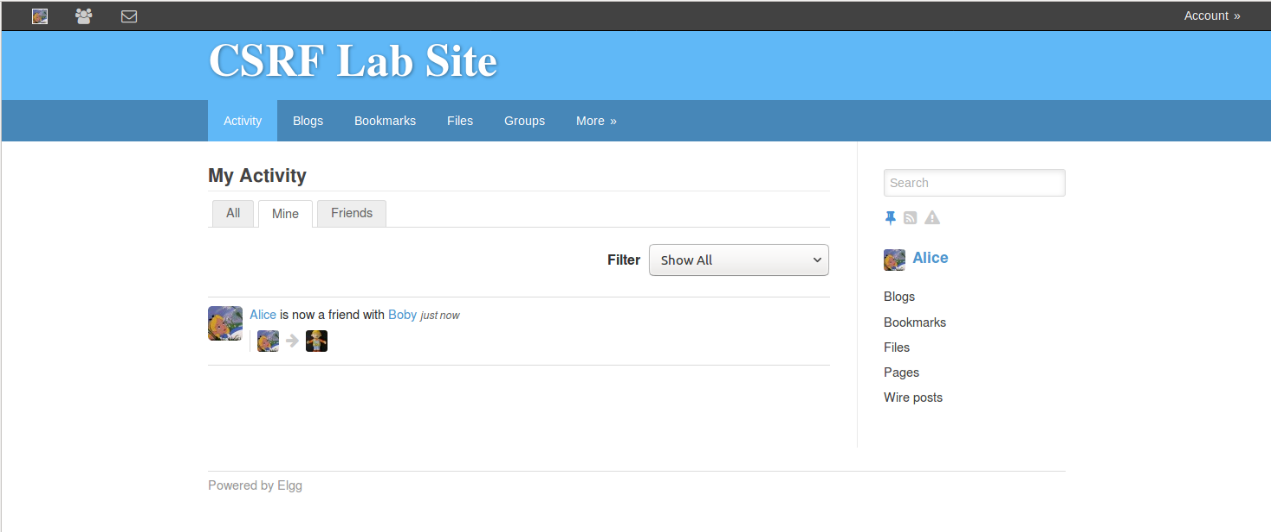
在/var/www/CSRF/Attacker中添加如下add\_boby.html文件：



打开[www.csrflabattacker.com](http://www.csrflabattacker.com)，界面如下：



Alice进入[www.csrflabattacker.com](http://www.csrflabattacker.com)之后，点击html链接，再回到[www.csrflabelgg.com](http://www.csrflabelgg.com)，发现已经添加Boby为好友：



攻击成功。

Task 3: CSRF Attack using POST Request

进入Boby的账户，将自己的profile修改成“Boby is my hero.”，得到的POST请求为：

<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/boby/edit

Content-Type: application/x-www-form-urlencoded

Content-Length: 507

Cookie: Elgg=hms2ij5oi9j7eg3885k01g9281

Connection: keep-alive

Upgrade-Insecure-Requests: 1

<http://www.csrflabelgg.com/profile/boby>

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

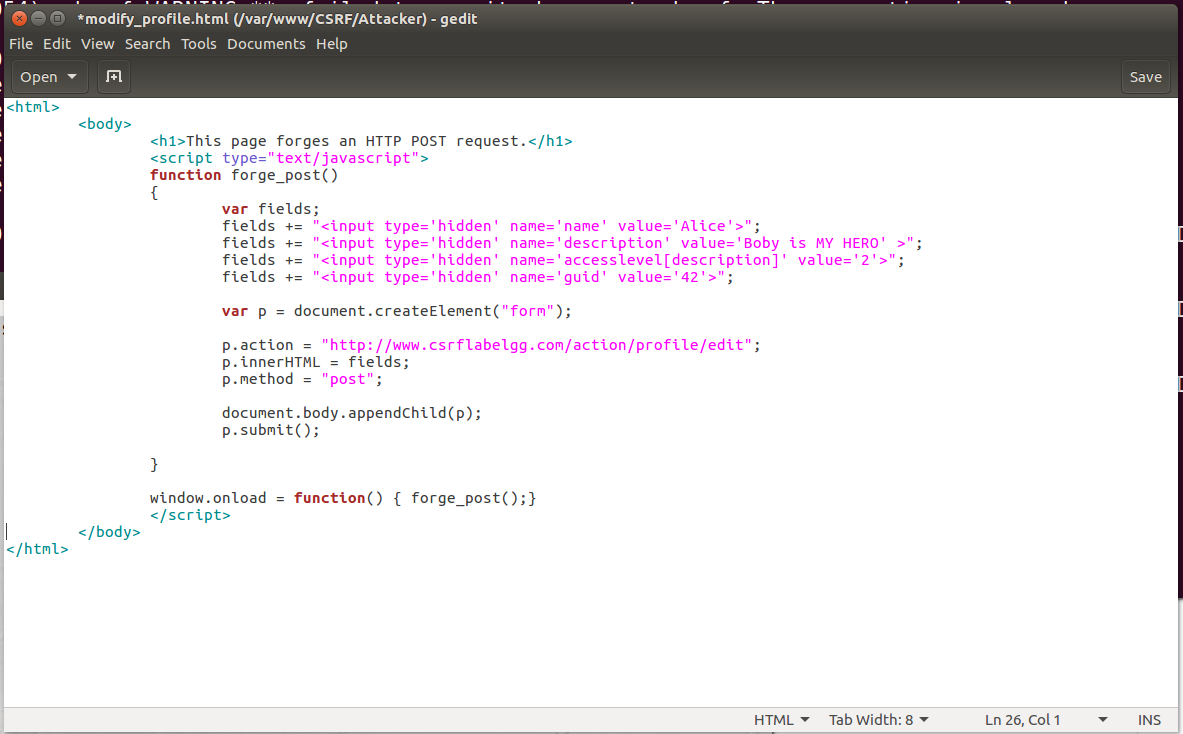
Referer: http://www.csrflabelgg.com/profile/boby/edit

Cookie: Elgg=hms2ij5oi9j7eg3885k01g9281

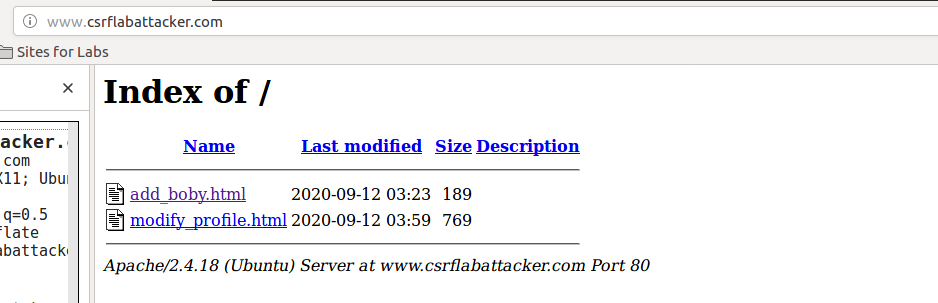
Connection: keep-alive

Upgrade-Insecure-Requests: 1

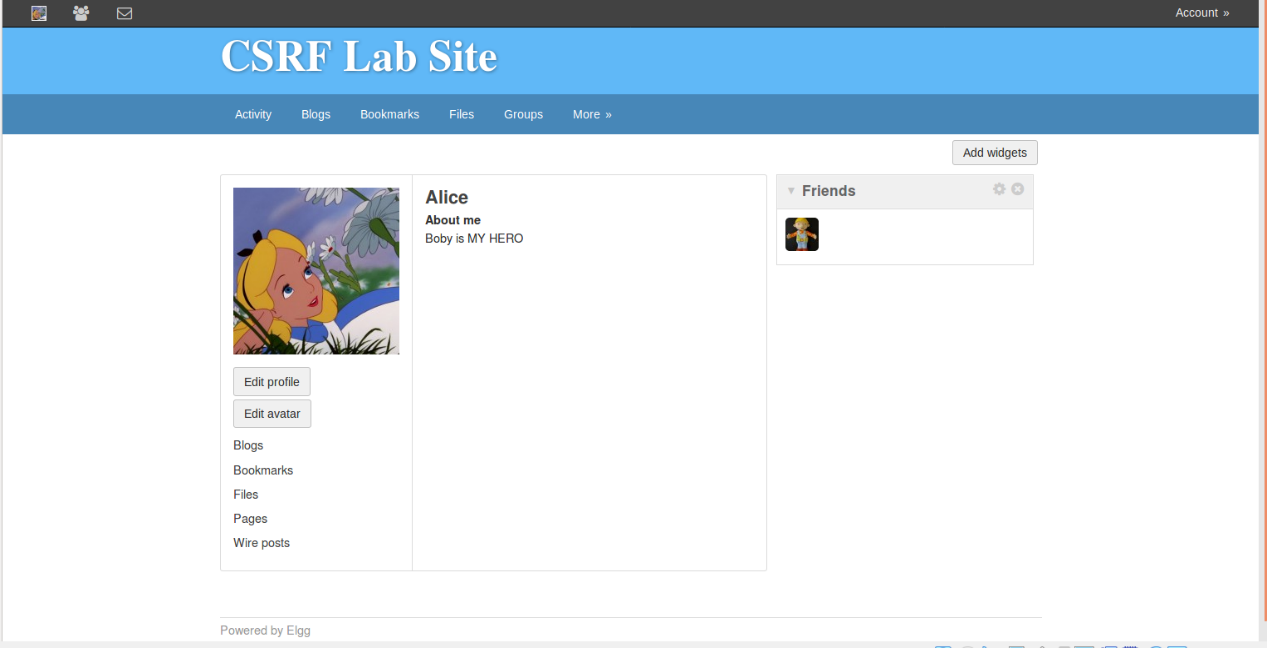
在var/www/CSRF/Attacker上创建modify\_profile.html文件，内容如下：



登录Alice的账户，进入[www.csrflabattacker.com](http://www.csrflabattacker.com)，界面如下：



回到[www.csrflabelgg.com](http://www.csrflabelgg.com)，发现Alice的profile被修改：



• Question 1: The forged HTTP request needs Alice’s user id (guid) to work properly. If Boby targets Alice specifically, before the attack, he can find ways to get Alice’s user id. Boby does not know Alice’s Elgg password, so he cannot log into Alice’s account to get the information. Please describe how Boby can solve this problem.

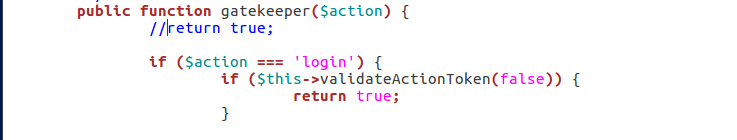
Boby可以通过添加Alice为好友，查看对应的HTTP GET请求，进而得到Alice的guid为42。

• Question 2: If Boby would like to launch the attack to anybody who visits his malicious web page. In this case, he does not know who is visiting the web page beforehand. Can he still launch the CSRF attack to modify the victim’s Elgg profile? Please explain.

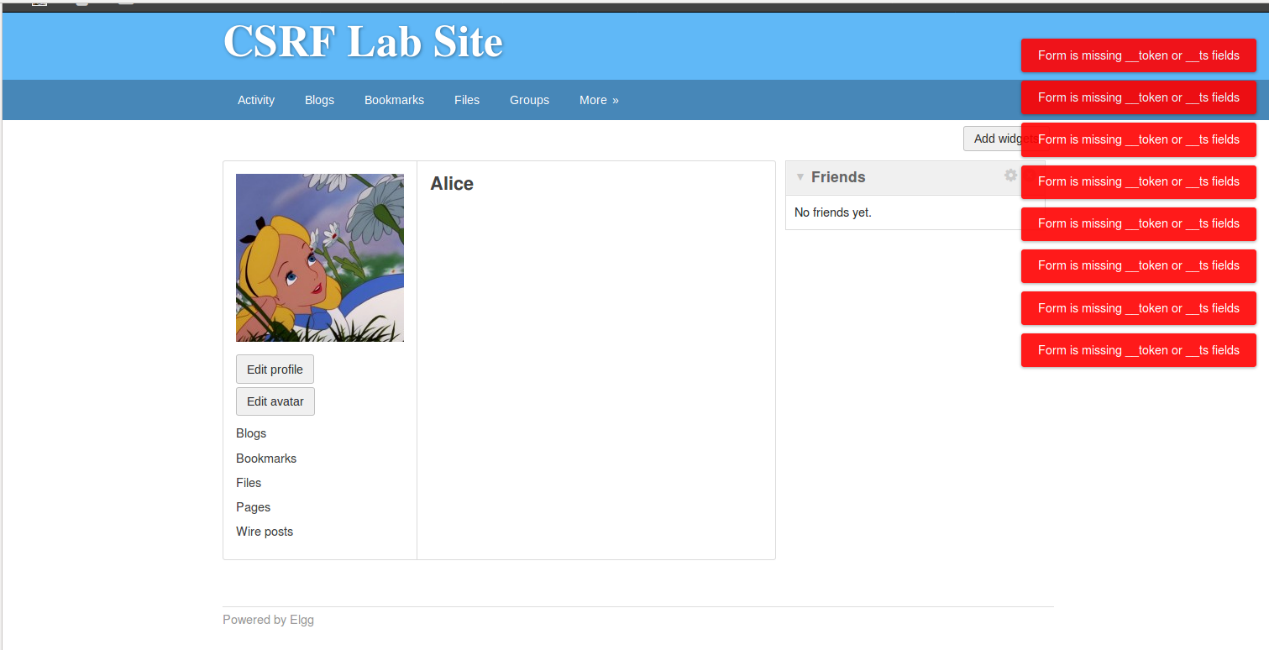
可以。Boby可以编写能自动获取访问该页面的用户对应的guid的JS代码，并将自动获取的guid加入html文件中。

Task 4: Implementing a countermeasure for Elgg

将/var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg下的ActionsService.php的函数gatekeeper()中的第一句话注释掉：



进入Alice的账户，移除Boby，并将profile清空。再次进入[www.csrflabattacker.com](http://www.csrflabattacker.com)，发现攻击不成功：



Elgg会在HTTP请求主体中添加secret-token和时间戳，分别为\_\_elgg\_token和\_\_elgg\_ts。Elgg的token是一个哈希值，会被加在web页面中。而不同源的网页不能访问这一页面的内容，因而恶意网站无法通过读取web页面上的\_\_elgg\_token来获取正确的token。攻击失败。