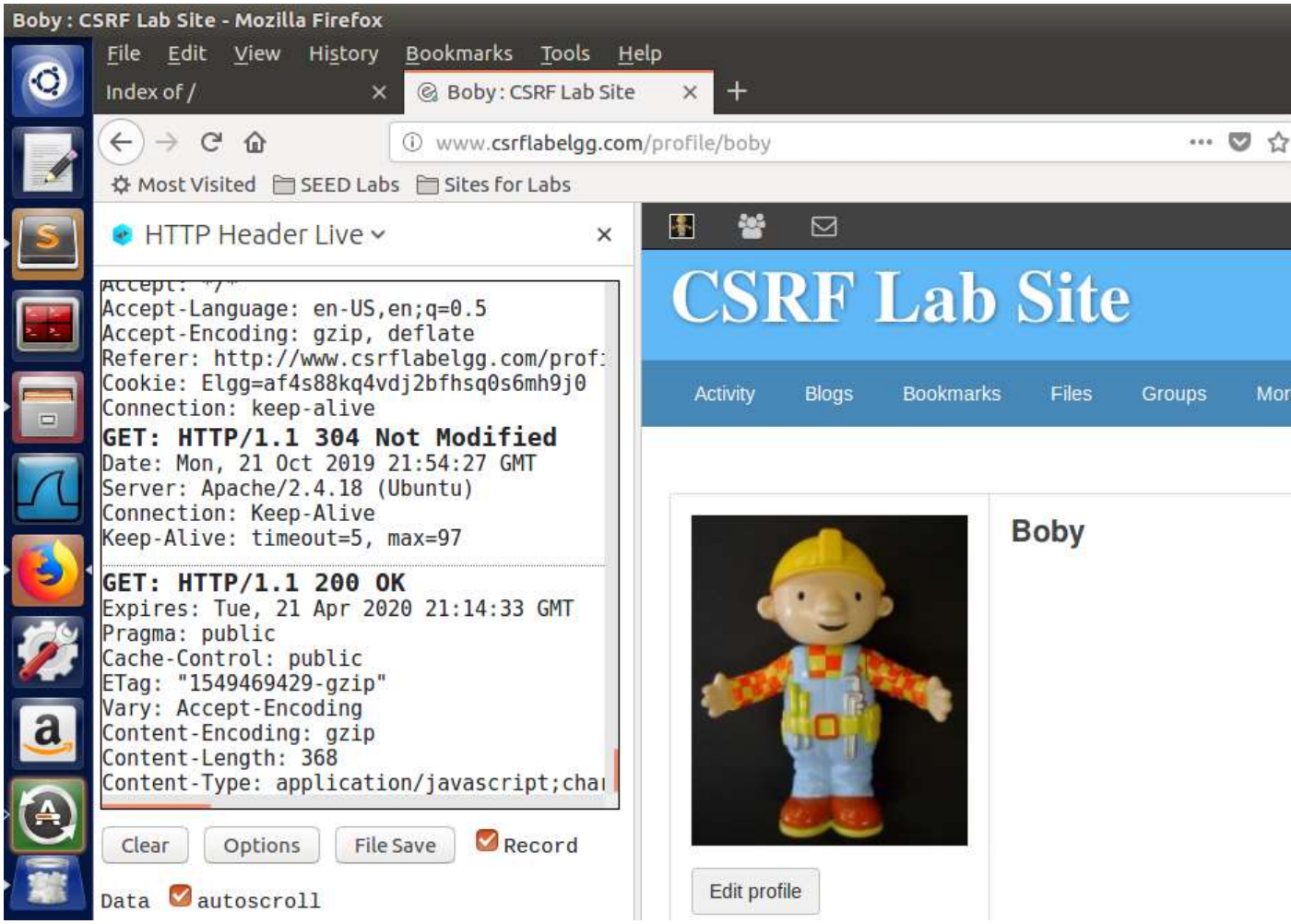
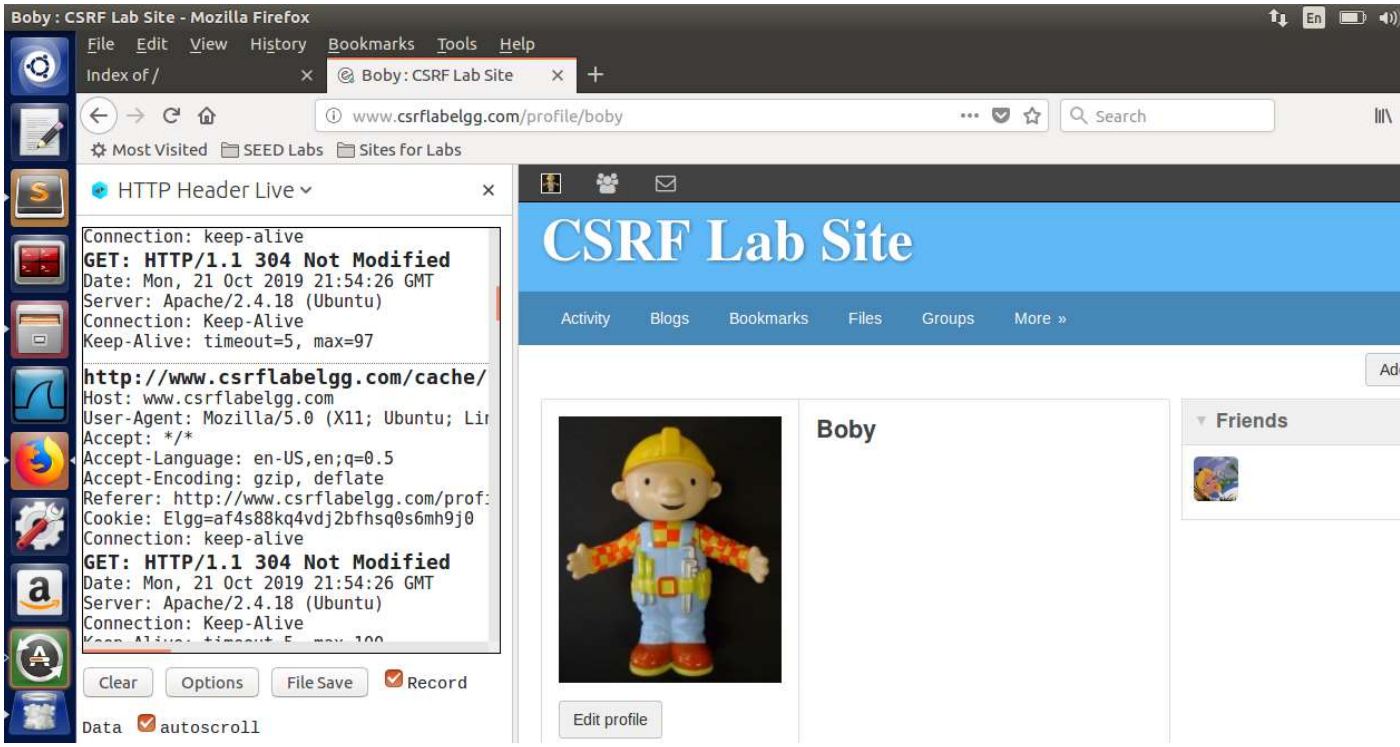
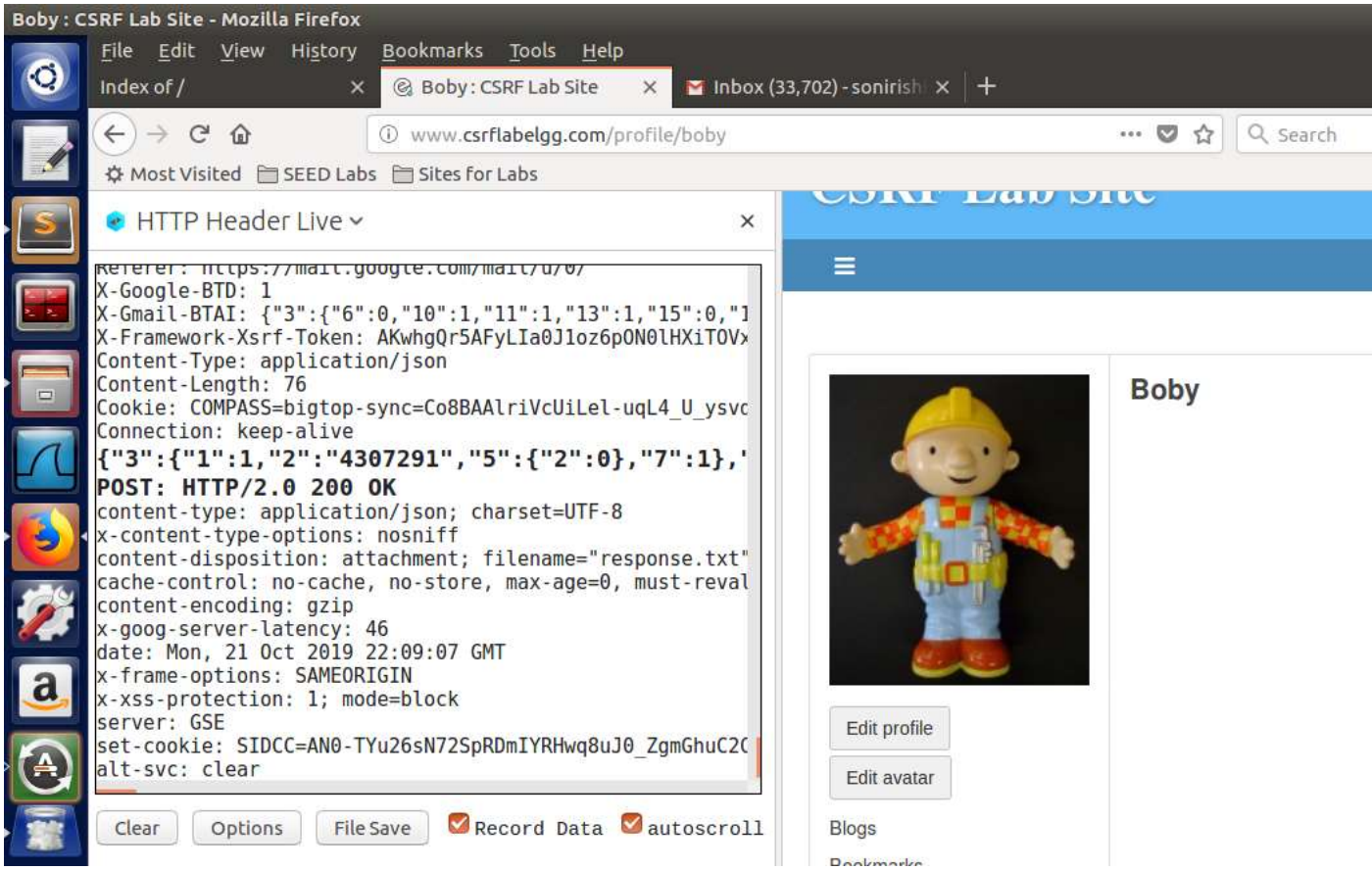
Assignment 07

RISHI KUMAR SONI 1001774020

Task 1: Observing HTTP Request.



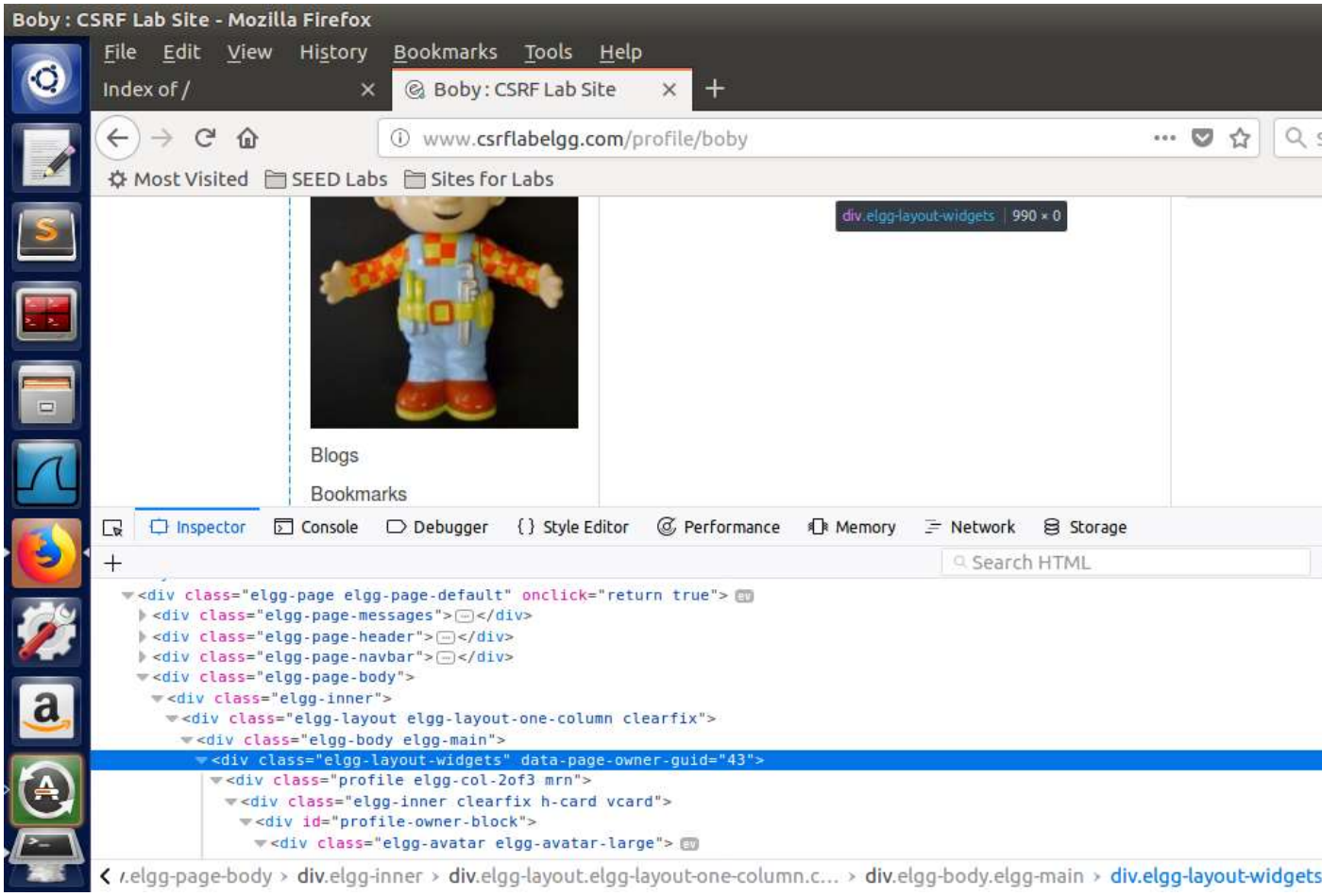

Boby : CSRF Lab Site - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Index of /
Boby : CSRF Lab Site
www.csrflabelgg.com/profile/boby
Search
Most Visited
SEED Labs
Sites for Labs
HTTP Header Live
Connection: keep-alive
GET: HTTP/1.1 304 Not Modified
Date: Mon, 21 Oct 2019 21:54:26 GMT
Server: Apache/2.4.18 (Ubuntu)
Connection: Keep-Alive
Keep-Alive: timeout=5, max=97
http://www.csrflabelgg.com/cache/
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/prof
Cookie: Elgg=af4s88kq4vdj2bfhsq0s6mh9j0
Connection: keep-alive
GET: HTTP/1.1 304 Not Modified
Date: Mon, 21 Oct 2019 21:54:26 GMT
Server: Apache/2.4.18 (Ubuntu)
Connection: Keep-Alive
Clear
Options
File Save
Record
Data
autoscroll
CSRF Lab Site
Activity
Blogs
Bookmarks
Files
Groups
More »
Boby
Friends
Edit profile


Boby : CSRF Lab Site - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Index of /
Boby : CSRF Lab Site
Inbox (33,702) - sonirish
www.csrflabelgg.com/profile/boby
Search
Most Visited
SEED Labs
Sites for Labs
HTTP Header Live
X-Google-BTD: 1
X-Gmail-BTAI: {"3":{"6":0,"10":1,"11":1,"13":1,"15":0,"1
X-Framework-Xsrf-Token: AKwhgQr5AFyLIa0J1oz6pON0lHXiTOV
Content-Type: application/json
Content-Length: 76
Cookie: COMPASS=bigtop-sync=Co8BAAlriVcUiLel-uqL4_U_ysvc
Connection: keep-alive
{"3":{"1":1,"2":"4307291","5":{"2":0},"7":1},'
POST: HTTP/2.0 200 OK
content-type: application/json; charset=UTF-8
x-content-type-options: nosniff
content-disposition: attachment; filename="response.txt"
cache-control: no-cache, no-store, max-age=0, must-reval
content-encoding: gzip
x-goog-server-latency: 46
date: Mon, 21 Oct 2019 22:09:07 GMT
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
server: GSE
set-cookie: SIDCC=AN0-TYu26sN72SpRDmIYRHwq8uJ0_ZgmGhuC2C
alt-svc: clear
Clear
Options
File Save
Record Data
autoscroll
Boby
Edit profile
Edit avatar
Blogs

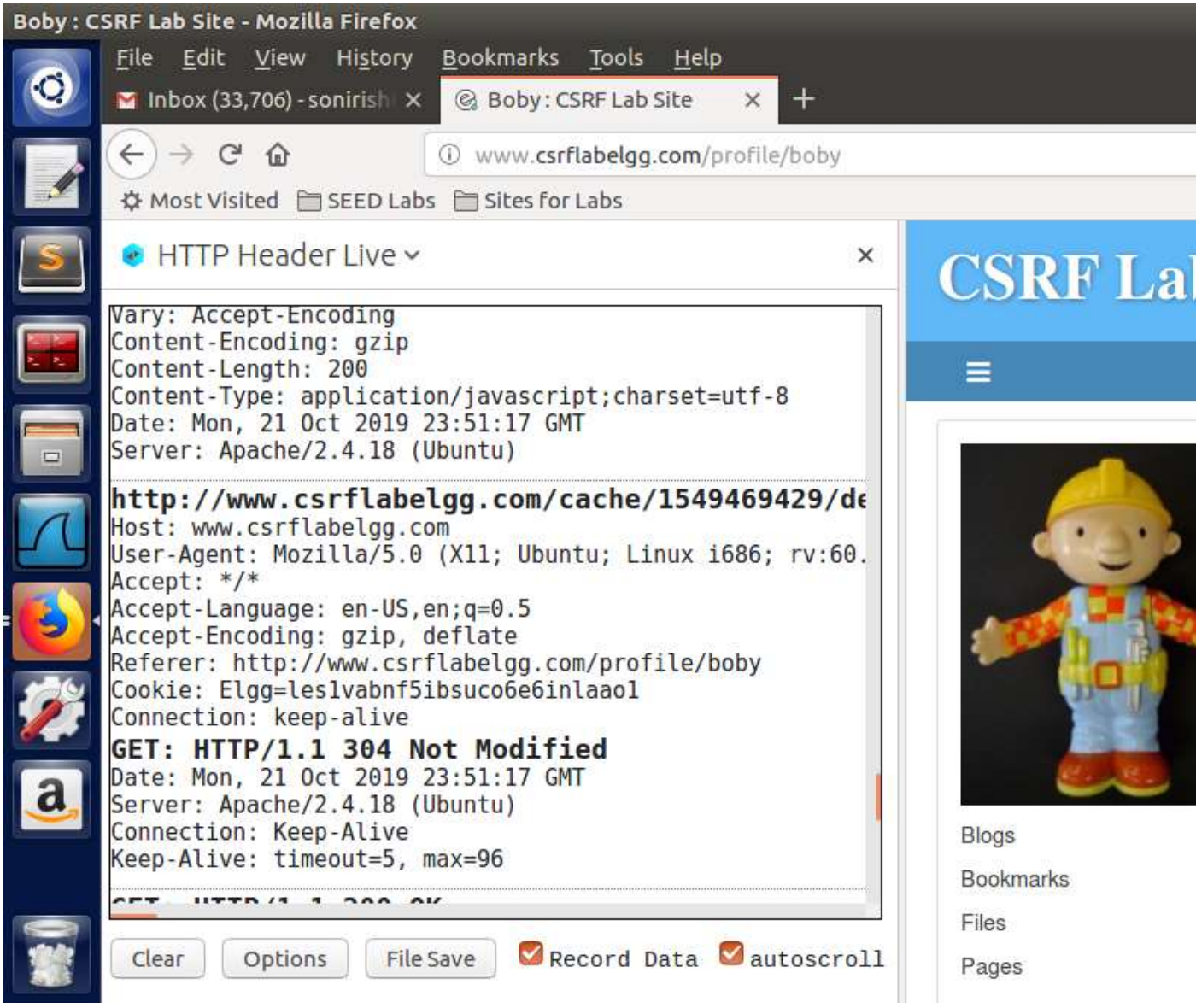
Observation

This is the Live HTTPHeader when we inspect through the Boby Page. Some HTML tags such as img, iframe, frame and form have no restriction on the URL that can be used in the attributes.HTML tags can also be used in the GET and Post request. We can forge by passing some of the attributes in it.

Task 2: CSRF Attack using GET Request

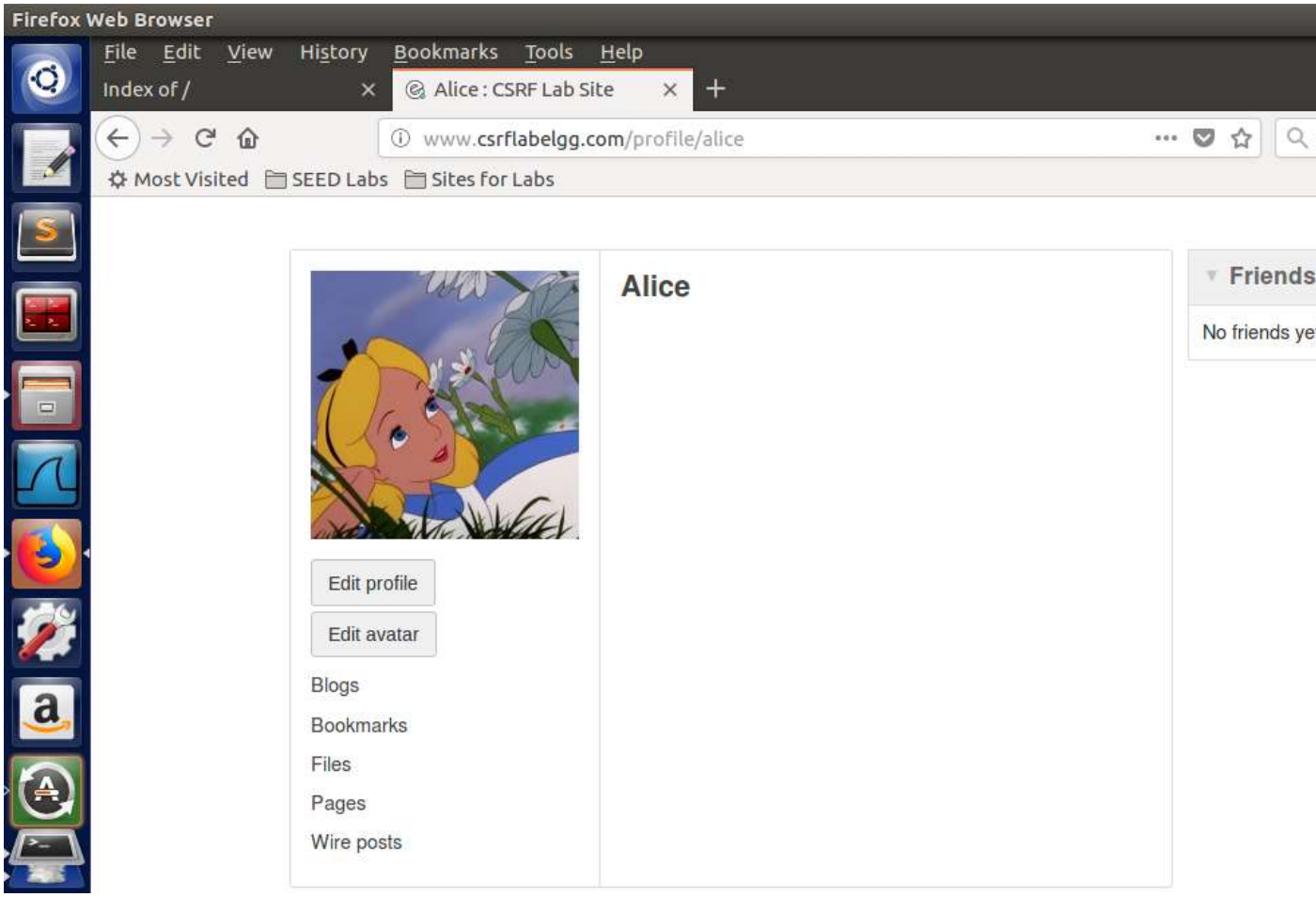


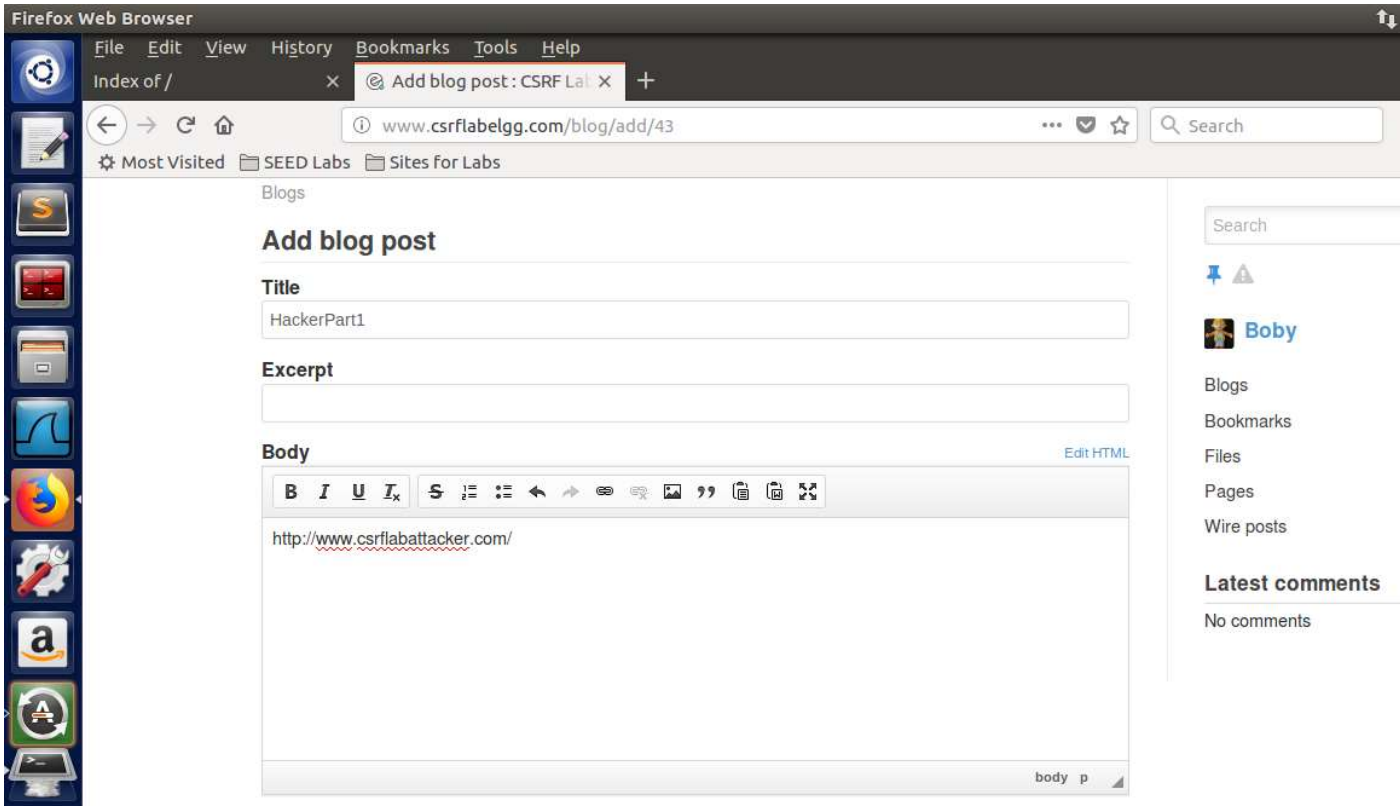
We are using the inspect element of Firefox to find out the user id of the attacker Boby.This User Id: 43.



Observation

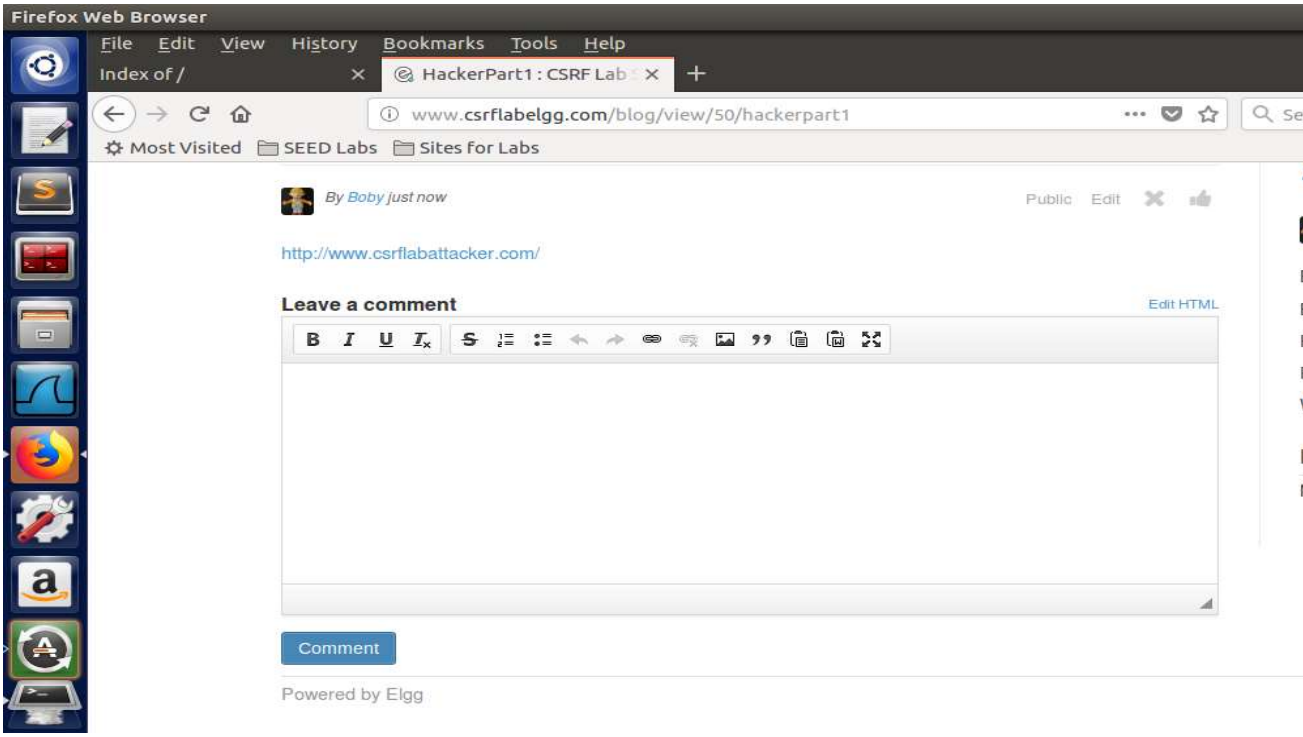
This is the LiveHTTPHeader when Boby adds Alice as his friend. He uses this as reference to construct the malicious url which adds him as a friend in Alice's account without Alice Knowing.





Observation

Boby creates a blog post with the malicious url in the body. When Alice clicks on this url, Boby gets added into Alice Friend list.



Bobby creates a blog post named “Hacker Part1”. Which content malicious code in it.

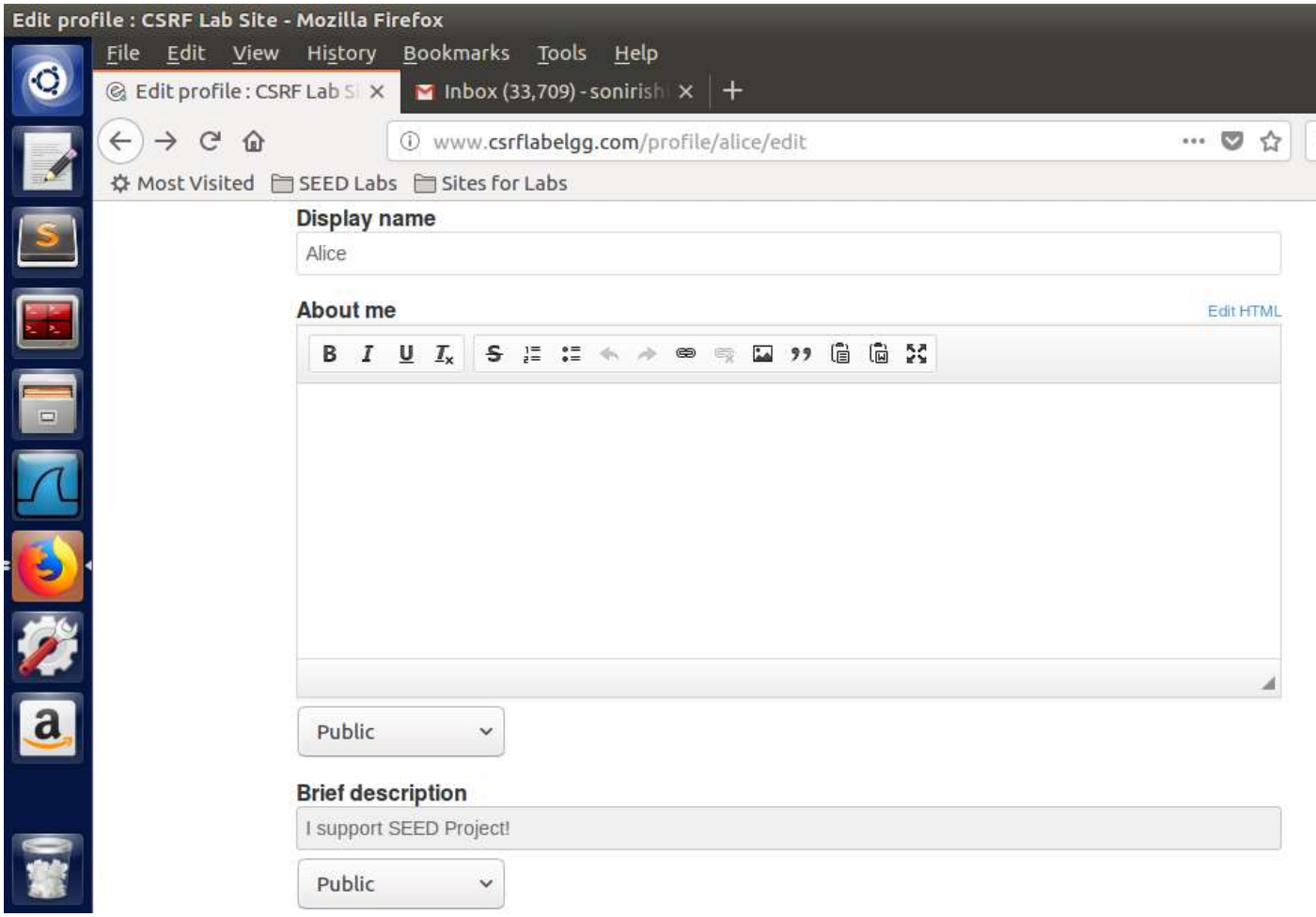
Observation

After Clicking on the blog url, Boby is added in the Alice's Friend List.

Explanation:

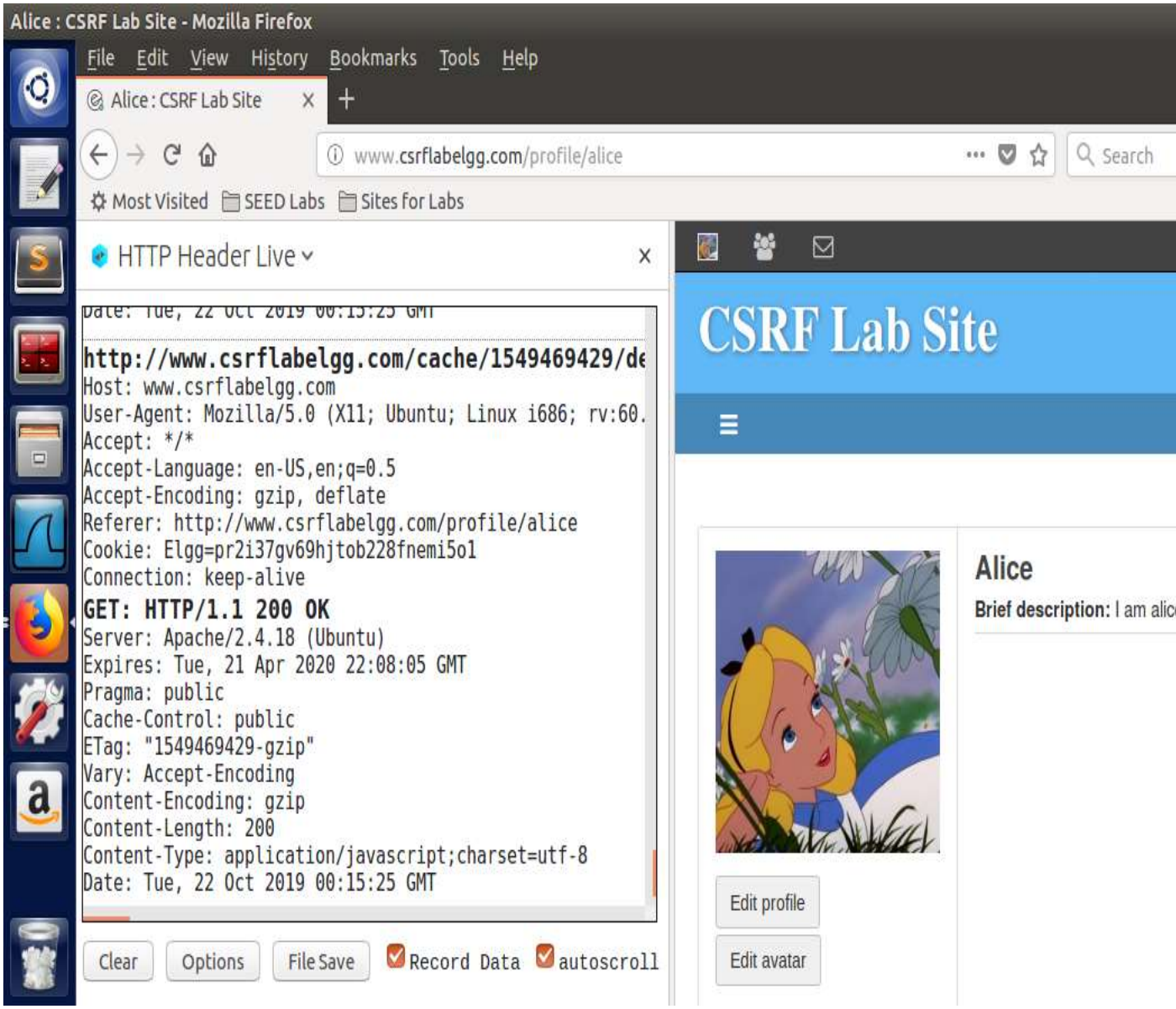
This is a Cross site request forgery attack where GET request is used to add Boby into Alice's friend list. Here we have a trusted site www.csrflabelgg.com, a user Alice logged into the trusteed site and Malicious website www.csrflabelgg.com created by Boby. So first, member' s page, inspect the element using Firefox and finds his id. Next he constructs the malicious url so that he can generate the GET request that adds him to Alice's friend list. So, Boby create a malicious url that he created. He sends this webpage to blog. So when Alice clicks on the link, boby gets added as a friend in Alice friend list.

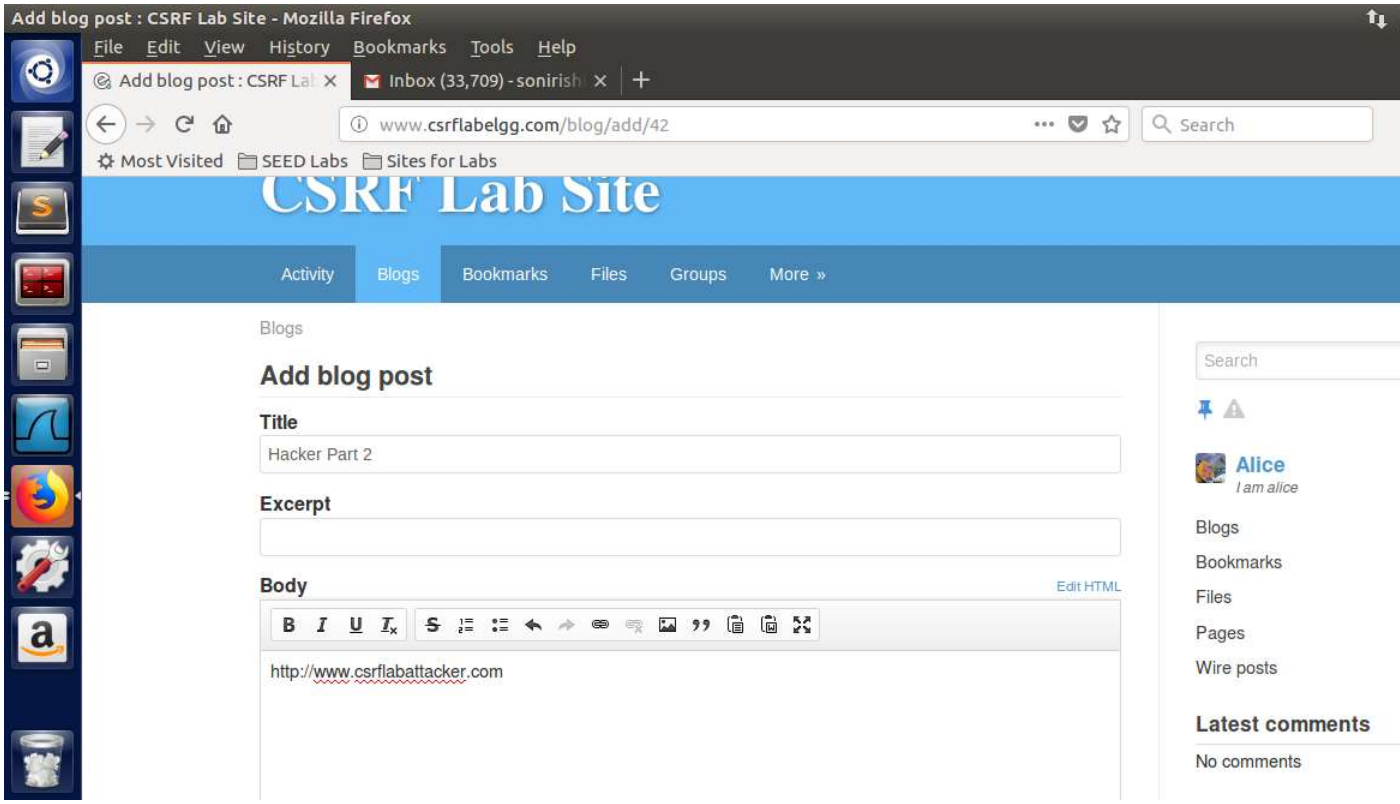
Task 3: CSRF Attack using POST Request



Observation

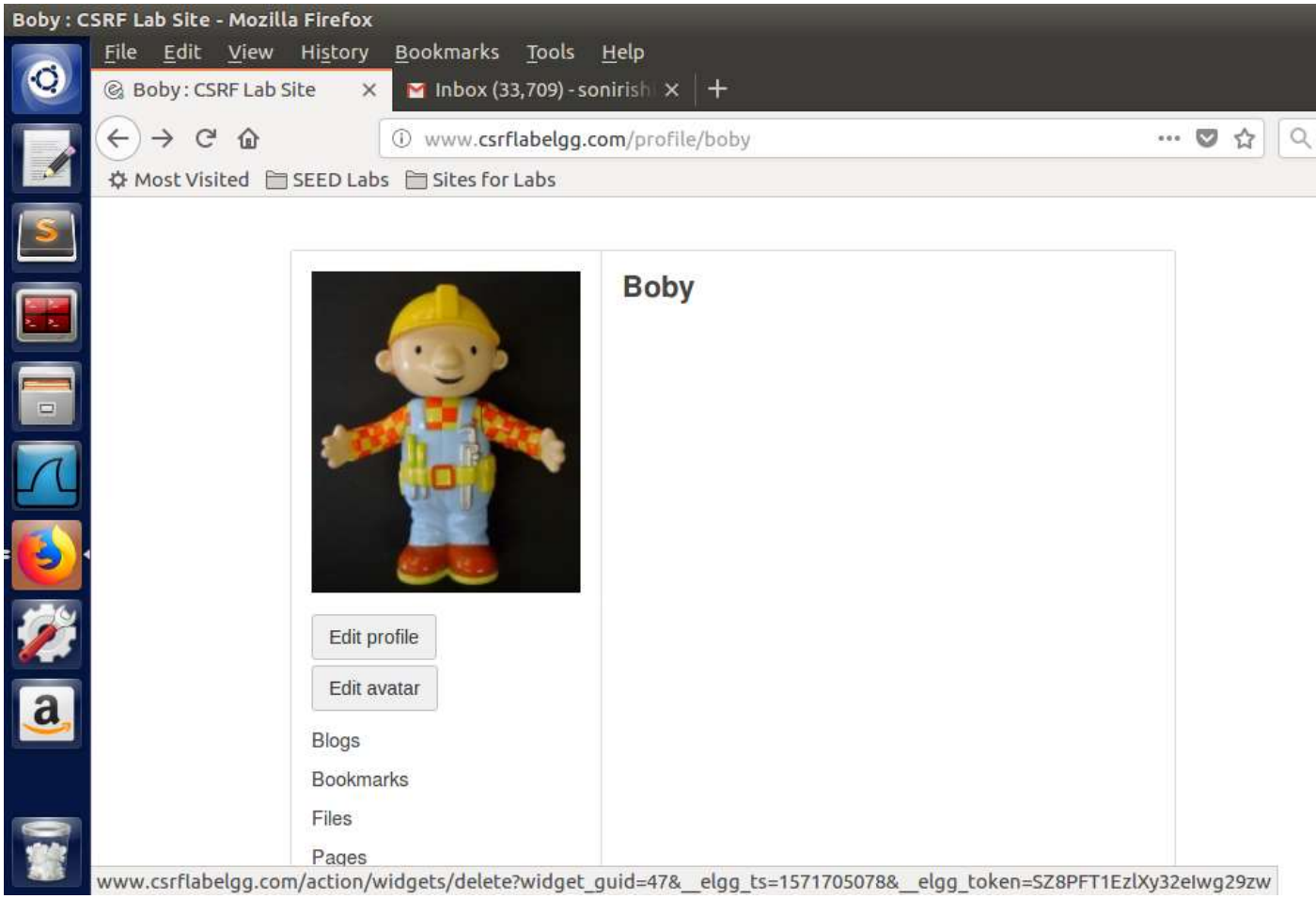
Alice edits her own profile with the description " I support SEED project " observe it using LiveHTTPHeader so that she can craft the malicious request.


Alice : CSRF Lab Site - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Alice : CSRF Lab Site
www.csrflabelgg.com/profile/alice
Search
Most Visited SEED Labs Sites for Labs
HTTP Header Live
http://www.csrflabelgg.com/cache/1549469429/de
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/profile/alice
Cookie: Elgg=pr2i37gv69hjtob228fnemi5o1
Connection: keep-alive
GET: HTTP/1.1 200 OK
Server: Apache/2.4.18 (Ubuntu)
Expires: Tue, 21 Apr 2020 22:08:05 GMT
Pragma: public
Cache-Control: public
ETag: "1549469429-gzip"
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 200
Content-Type: application/javascript;charset=utf-8
Date: Tue, 22 Oct 2019 00:15:25 GMT
Clear
Options
File Save
Record Data
autoscroll
CSRF Lab Site
Alice
Brief description: I am alic
Edit profile
Edit avatar



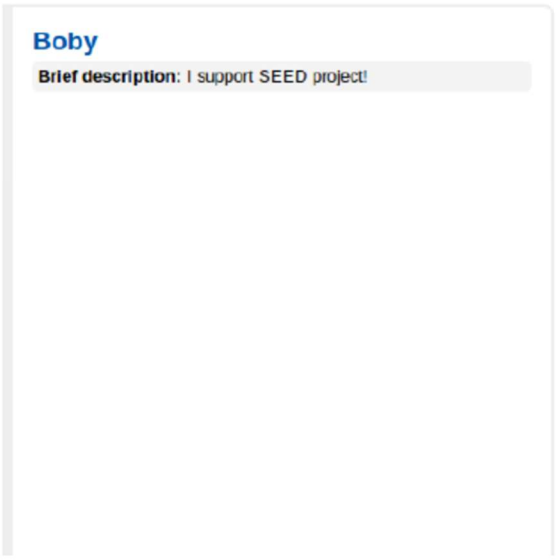
Observation

Alice creates a blog post with malicious url so that when Boby clicks on it, his description is modified.



Observation

This was before clicking on the malicious url.



Observation

Description on Boby’s profile is modified after clicking on this link.

Explanation:

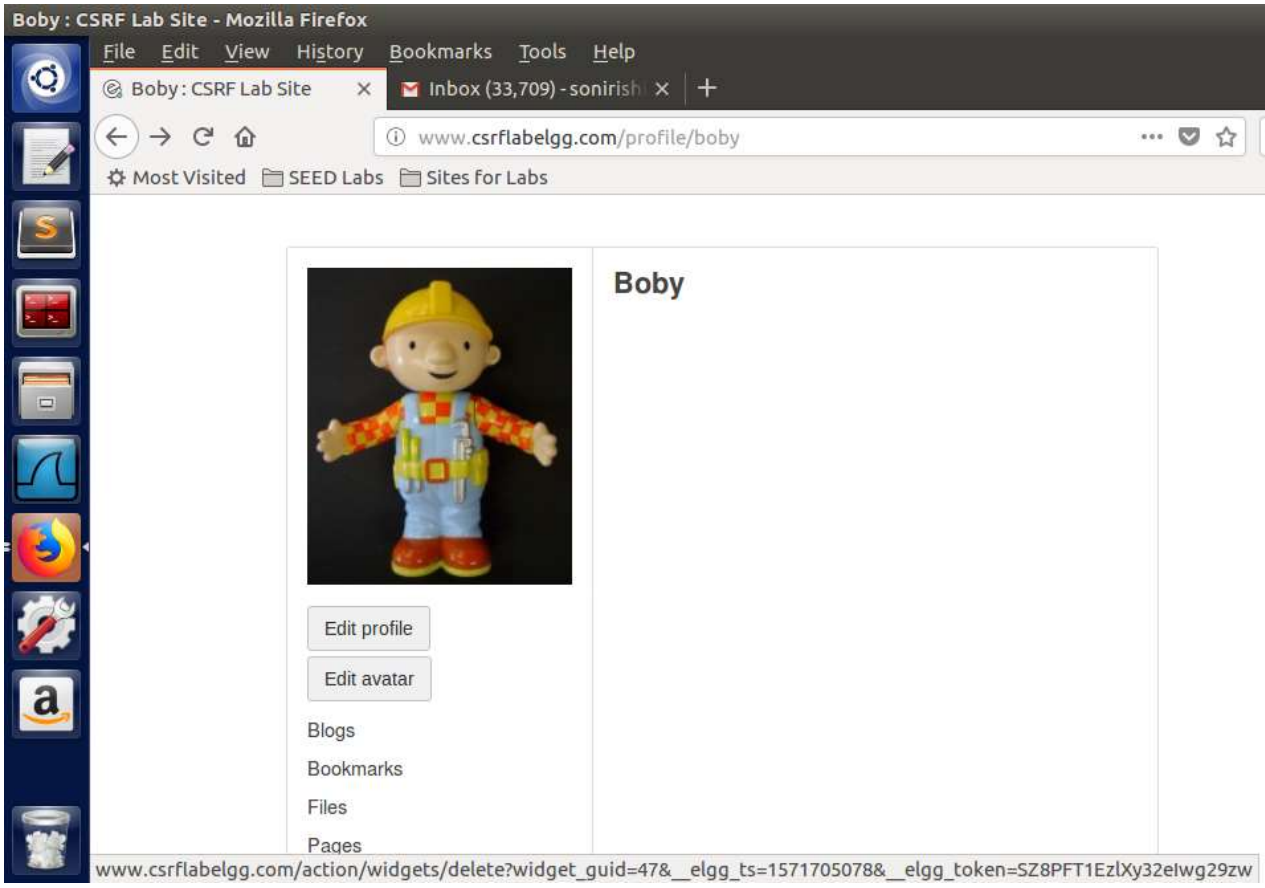
Since data must be sent using Post method for this attack. This is a cross request Forgery where post request is used to modify content of Boby Profile. Here we have a site csrflabelgg.com. This time it is created by Alice. Alice has to get the Boby id (which already available from the above task). Next she should construct the url so that she can generate the post request to the server which recreates the form submission of the profile page with changed content. When Boby click the the generated link the Boby profile get updated and the same thing appear on the Boby page which is written by the Alice.

Answer 1: Alice can find Boby's if by inspecting the member page of the elgg site using firebug.

Answer2. Alice is unable to launch the attack with anyone who visit her malicious page since the user id if every user is different and only when the user id of the logged in user and the user id specified in the webpage matches and out attack succeed. The attack takes place if the user id is specified in the webpage that has an active session with elgg and visit webpage and by changing this id (user id), we can perform the attack on the user also.

Task 4: Understanding phpBB's Countermeasures

Observation



Explanation:

Try to perform the same attack with the Countermeasure Turn on the attack failed. We can see the description is not modified. The countermeasures are to send 2 fields. The first is timestamp and the second one is Unique token along with each request. When the countermeasures are turned on, it compares the values with the session (valid session) with the session of the user. It fails if we perform the attack with the countermeasures because it identifies it as a cross site request and not a request from the user.