Assignment Setup
This assignment will utilize the SEED virtual machine image, information on obtaining and running this image can be found in Assignment 2.

You'll need two Firefox plugins to help you interact with the browser and its communications during these labs. Download and install both of these plugins if they are not already installed in your SEED VM.

1) **Firebug** – This is tool that can be used to help debug JavaScript and HTML running on a page (along with other function). It can be found here (`https://addons.mozilla.org/en-US/firefox/addon/firebug/`).

2) **LiveHTTPHeaders** – This tool can be used to monitor the HTTP headers for all requests and response between the browser and webserver. It can be obtained here (`https://addons.mozilla.org/en-US/firefox/addon/live-http-headers/`).

For this assignment we'll use the `http:///www.xsslabelgg.com` website on the SEED VM. Test that this works by opening Firefox and then browsing this site. Get familiar with how to view friends, add friends, remove friends, and updating you profile. We'll use the following user accounts, verify that you can log in to each of these users. Make sure you remove all friends before proceeding.

Users:	(Username	–	Password)
Samy	(samy	–	seedsamy)
Alice	(alice	–	seedalice)
Boby	(boby	–	seedboby)

Part A. Basic XSS
The part will perform a basic XSS attack, just to get you familiar with the concept. We'll focus on a stored XSS vulnerability in the Brief Description value in the Edit Profile page. Login with Samy, and then go to update your profile. Once you've located this field insert the following value and the select Save. Once the page reloads you should see an alert box pop-up.

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

and save the profile change. You should see the alert box pop-up after you hit save, with Samy's session token displayed

Question 1) What is Samy's session token?

Instead of directly injecting the JavaScript to the profile, now add it the web server by creating the file `/var/www/lab5_xss.js` and adding the value `alert(document.cookie).` Next you'll

have to change the injected JavaScript to load script from the address `http://127.0.0.1/lab5_xss.js`, rather than directly having the script in the `www.xsslabelgg.com` site.

> **Question 2)** What is the result HTML we must add to the Brief Description field in order to execute the now remotely hosted JavaScript?

Part B. Automatically Add Friends

For this part we'll change the code in `lab5_xss.js` so that it automatically adds a user to Samy's friends list, instead of just displaying the user's cookies. For this we need to spoof the correct HTTP GET request that is sent by the browser when a user adds a new friend. The following JavaScript can be used to send an `XMLHTTPRequest` to `http://www.xsslabelgg.com`, however we need to define the specific page and variables. Use the JavaScript below as the base for the new `lab5_xss.js` file, however, you'll need to find the correct value for `<ADDFRIENDPAGE>` and `<VARIABLES>`.

```
url="http://www.xsslabelgg.com/<ADDFRIENDPAGE>?<VARIABLES>";
var Ajax1=null;
Ajax1=new XMLHttpRequest();
Ajax1.open("GET",url,true);
Ajax1.setRequestHeader("Host","www.xsslabelgg.com");
Ajax1.setRequestHeader("Keep-Alive","300");
Ajax1.setRequestHeader("Connection","keep-alive");
Ajax1.setRequestHeader("Cookie",document.cookie);
Ajax1.send();
```

Next we need to use the Firefox `HTTP Live Headers` plugin to find the correct values for `<ADDFRIENDPAGE>` and `<VARIABLES>`. In Firefox, click `Tools -> HTTP Live Headers`. Now monitor the HTTP messages while adding Alice as Samy's friend.

> **Question 3)** What is the HTTP GET message URL used to add a new friend to a profile?

> **Question 4)** What are the 3 variables that accompany this message? Explain the purpose of each.

Because not all of the values in Question 4 are static, we need to go through the page's JavaScript to find the correct values, we can then add them to `<VARIABLES>`. Open the Firebug tool (`Tools -> Web Developer -> Firebug`) and view the `script` tab. Can you find currently defined JavaScript variables defined for these values?

> **Question 5)** Provide the resulting value of `<ADDFRIENDPAGE>` and `<VARIABLES>` that we can use to send automatically send a friend request when you visit Samy's page. Verify this works by ensuring Alice and Samy are not friend and then log-in as Alice and visit Samy's page. Alice should automatically be added to Samy's friends list.

Go in and remove all Samy's friends. Now log in as Alice and visit Samy's profile and then refresh the page. You should see now see Samy listed as Alice's friend.

Part C. Self-Propagating Worm

In the previously section we showed how we could spoof a message to automatically add Samy as a user's friend when they visited his page. In this part we will make this malicious script also propagate to other user's pages so that everyone will eventually be Samy's friends.

First, we need to look at the HTTP POST request used to update a user's profile, so that we can spoof our own update profile request whenever they visit Samy's page. Login as Alice and update her page, use `HTTP Live Headers` to view the HTTP POST information during this process.

> **Question 6)** What is the URL used to submit the profile update? Also, provide the HTTP POST variables.

Now we need to add a second `XMLHTTPRequest` to `lab5_xss.js` which will spoof a profile update HTTP POST and propagate the malicious script to whatever user views that page. Use the following code template (*Note: this is in addition to Part B*). Update the `<PROFILEUPDATEPAGE>` value based on the results of the previous question.

```
var Ajax2=null;
Ajax2=new XMLHttpRequest();
Ajax2.open("POST","http://www.xsslabelgg.com/<PROFILEUPDATEPAGE>
",true);
Ajax2.setRequestHeader("Host","www.xsslabelgg.com");
Ajax2.setRequestHeader("Keep-Alive","300");
Ajax2.setRequestHeader("Connection","keep-alive");
Ajax2.setRequestHeader("Cookie",document.cookie);
Ajax2.setRequestHeader("Content-Type","application/x-www-form-
urlencoded");
Ajax2.send(<HTTPPOSTVARIABLES>);
```

Now we just need to find the appropriate value for `<HTTPPOSTVARIABLES>`. Fortunately, we can reuse most of the HTTP POST variables discovered in Question 6). However, we need unique values for the 2 CSRF protection tokens, along with the viewing users' `name` and `guid`. To find these, we can use Firebug to search through the page's `script` tab. These are located in JSON objects, read the following page to learn more about this (`http://www.w3schools.com/js/js_json.asp`).

> **Question 7)** Provide the resulting 4 JavaScript variables that we need to include in our POST.

Last, but not least, we need also update the `Brief Description's` POST variable so we can add our propagating JavaScript to the unsuspecting user's profile page.

> **Question 8)** What value do we need in the `Brief Description` POST variable to propagate the malicious JavaScript? Note: you should use the JavaScript `escape()` function when adding HTML within HTTP POST variables.

Finally, we should have working Javascript malware that can propagate to other users, with each infected users adding Samy to their friend's list. Test to make sure it works. First remove all of Samy's friends. Now log-in as Alice and visit Samy's page, you should notice that Alice and Samy are now friend. Now log-in as Boby and visit Alice's page. You should not notice that now Boby and Samy are now friends (even though Boby hasn't visited Samy's page).

> **Question 9)** Provide the resulting `lab5_xss.js` file.