**HARSHAL BHOLE, ADESH PADWAL, ASHWIT REDDY**

Problem 2:

Task 1:

In this task we modified Alice brief description and inserted this code <script>alert('XSS');</script> and we got the following output.

Graphical user interface

Description automatically generated

Task 2:

In this we embed a JavaScript program in your Elgg profile, such that when another user views your profile, the user’s cookies will be displayed in the alert window and we got this code.

A screenshot of a computer screen

Description automatically generated

Task 3:

In this task, the attacker wants the JavaScript code to send the cookies to himself/herself. To achieve this, the malicious JavaScript code needs to send an HTTP request to the attacker, with the cookies appended to the request.

This script will send an HTTP GET request to the Attacker machines IP address at port location 7777. The Attacker machine needs to have a server listening on this port in order to receive the data (the cookie) being sent to that port.

We open a new terminal and use netcat (nc for short) to set up the server. The attack was successful and Samy now has Alice’s session cookie.

We did the steps given and obtained the output like this:

A screenshot of a computer

Description automatically generated

Task4:

In this task, we need to find out what information is sent in the HTTP request sent out when a user adds Samy as a friend on the Elgg site. First, we remove last task’s script from Samy’s profile and save the changes. Then we log in as Charlie, go to Samy’s profile, pull up the web development network tool.

In a HTTP GET request, the data being sent is attached to the URL. The parameters are placed after the ‘?’ in the URL and separated by ‘&’. We need to create a GET request to http://www.xsslabelgg.com/action/frineds/add with the parameters friend=47 (which is Samy’s GUID), \_\_elgg\_ts and \_\_elgg\_token. The \_\_elgg\_ts and \_\_elgg\_token don’t need to be repeated, and their values are obtained dynamically using elgg.security.token.\_\_elgg\_ts and elgg.security.token.\_\_elgg\_token. Using this we built a code on JS and posted it in the about of profile and got following:

Before executing code:

Graphical user interface, text, application

Description automatically generated

After execution:

A screenshot of a computer

Description automatically generated

We can see that Samy is added to the friendlist successfully.

Task 5:

For this task we need to figure out what type of HTTP request is used when a user edits their profile, and what data is sent along with that request. We begin by editing Samy’s profile while using the web developer network tool to view the HTTP request sent.

Using the information we have gathered, we create the malicious JavaScript program for performing the attack by editing Samy’s profile, removing the ‘Test’ from his brief description, and then copy and paste the script into his About me section and save the changes:

Before execution:

Graphical user interface, text, application

Description automatically generated

After execution:

A screenshot of a computer

Description automatically generated

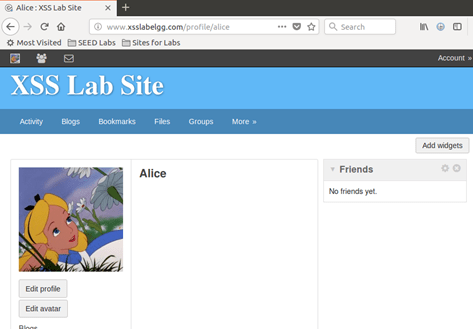
The attack was successful.

Task 6:

For this task we need to combine the things that were accomplished in the last task: add Samy to the victim’s friends list and edit the victim’s profile. This task also requires making the script self-propagating, meaning that it will copy itself to the victim’s profile so the victim also becomes an attacker, and anyone who views the victim’s profile will also become a victim of the attack.

We will be using the DOM approach to accomplish the self-propagation. The lab description shows an example of this approach under the task 6 section. we will use the information we have already gathered to create a HTTP GET request for adding Samy to the victim’s friends list, and to create a HTTP POST request for editing the victim’s profile. The HTTP POST request will not only add ‘Samy is my hero’ to the Brief description but will also add a copy of the script to the About me section.

Alice’s profile before viewing samy’s account:



After:

A screenshot of a computer

Description automatically generated

Boby,s profile before visiting Alice:

Graphical user interface, application, Word

Description automatically generated

After: Graphical user interface, website

Description automatically generated