PCS Lab 4: Cross-Site Request Forgery Instructions

Instructions

Read and follow the SEED lab instructions below, and answer the questions in the next page.

SEED Lab Instruction on CSRF:
http://www.cis.syr.edu/~wedu/seed/Labs_16.04/Web/Web_CSRF_Elgg/Web_CSRF_Elgg.pdf

Some useful materials:

- Background Info: http://www.cis.syr.edu/~wedu/seed/Labs_16.04/Web/Web_CSRF_Elgg

Submission

Due: April 17, 11:00 PM (EST)

Late submissions are accepted with 50% grading penalty within 24 hours of the due.
Submissions that are late for more than 24 hours will NOT be accepted.

Please submit your solution in PDF or image on https://www.gradescope.com/courses/38640.
Please choose the right page for your answer to every question.

Questions

Question 1: Please provide the names and NetIDs of your collaborator (up to 2). If you finished the lab alone, write None.
Collaboration Policy

1. **You can optionally form study groups consisting of up to 3 people per group (including yourself).**
2. **Everybody should write individually by themselves, and submit the lab reports separately. DO NOT copy each other's lab reports, or show your lab reports to anyone other than the course staff, or read other students' lab reports.**

Question 2. Please use the HTTP Header Live extension to capture an HTTP GET request and an HTTP POST request in Elgg, as described in "Task 1: Observing HTTP Request". Identify and describe the parameters of the requests.

Question 3: Please implement "Task 2: CSRF Attack using GET Request" and answer the following questions:

3.1 What are the URL and parameters of the GET request for sending friend requests?.
3.2 Construct a web page such that it automatically adds Boby as a friend when Alice visits it, without even making any click on the page. Please attach the source code for this web page.
3.3 Please describe your observation with screenshots that show your attack is successful (e.g. using HTTP Live Header).

Question 4: Please implement "Task 3: CSRF Attack using POST Request" and answer the following questions:

4.1 What are the parameters of the POST request for editing profiles?.
4.2 Construct a web-page such that it automatically changes Alice's profile's brief description to "Boby is my Hero" when Alice visits it, without even making any click on the page. Please attach the source code for this web page.
4.3 Please describe your observation with screenshots that show your attack is successful (e.g. using HTTP Live Header).
4.4 How can Boby find Alice's user id (guid) without her credentials? How can Boby find his own user id?
4.5 If Boby would like to launch the attack to anybody who visits his malicious web page. In this case, he does not know who is visiting the web page beforehand. Can he still launch the CSRF attack to modify the victim's Elgg profile? Please explain.

Question 5: "Task 4: Implementing a countermeasure for Elgg"

5.1 Please turn on the countermeasures and execute the CSRF attack again. Describe your observations and explain how the countermeasures work in brief.
5.2 Please explain why the attacker cannot send these secret tokens in the CSRF attack; what prevents them from finding out the secret tokens from the web page??