Software Architecture Documentation

Assignment #2
Non-persistent Cross-Site Scripting

Course	Computer Communication
Professor	Bongjun Choi Prof.
Major	Computer Science and Engineering
Student ID	20172655
Name	Kangsan Lee
Submission Date	June 11, 2021.

1. Introduction and Background

Chapter 11 (The World Wide Web), Listing 11-2 [1] shows an example bank application that uses the Flask framework. This insecure payment application is prone to various scripting attacks, including the non-persistent cross-site scripting attack, where an attacker can build a URL whose query parameter includes a JavaScript [2] that makes automatic payment.

2. Logical View and Process View

- The example is an implementation of a code to generate a URL that can also display a custom message on the green box (indicated by flash) of the index page upon successful payment process
(as discussed on our lecture note p67).

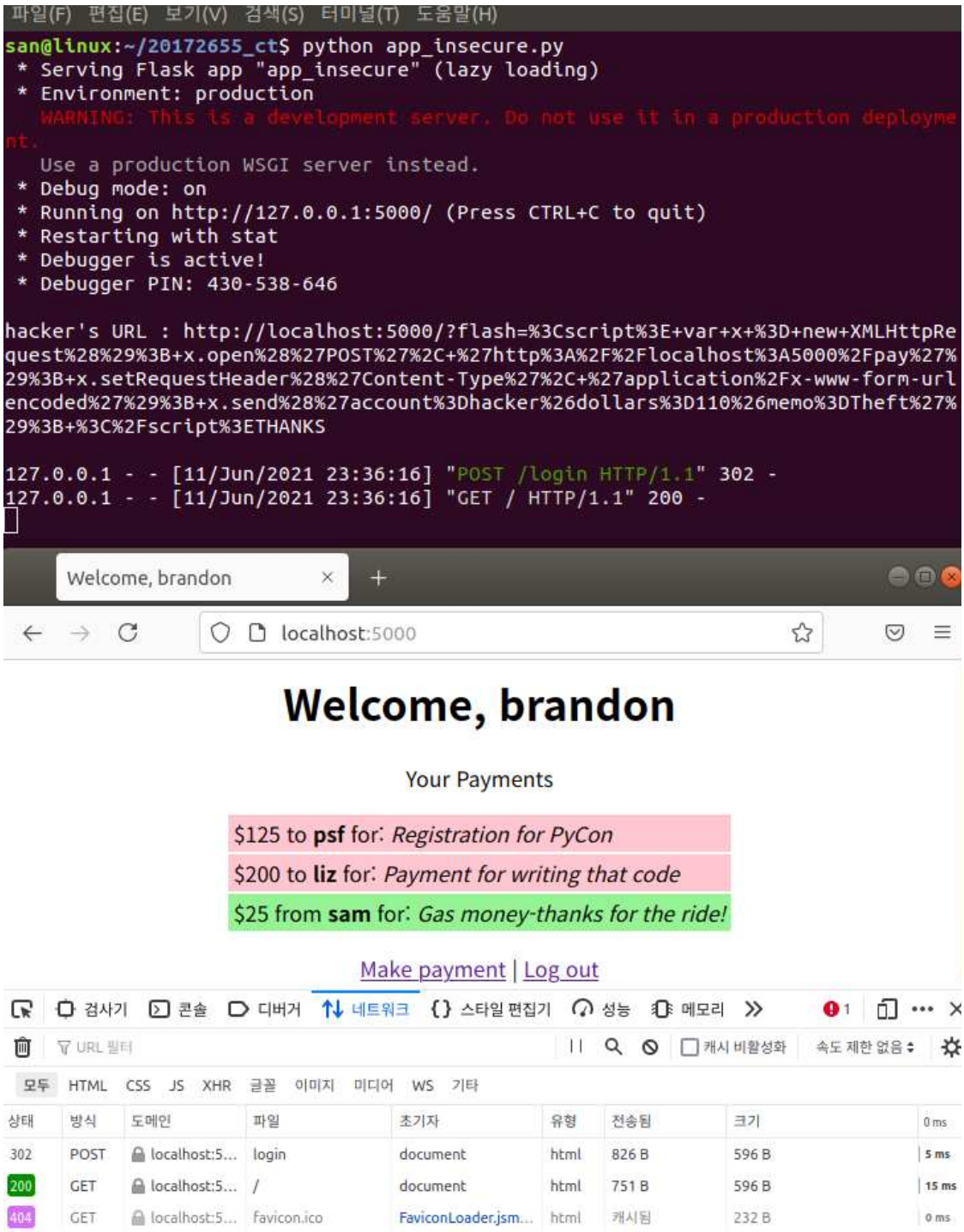
Specification:
1. Input:
- Attacker (named hacker) generates a URL (called attacker-generated URL) using Python code that opens a script file and then prints a URL that can be used to launch such an attack. Your code should look something like the one shown on p. 66 of our lecture note.

2. Method:
- Login as brandon (a target user).
- brandon visits the attacker-generated URL.
- brandon is redirected to the index.html that displays a custom message chosen by the attacker (e.g., Thanks, brandon) to become less suspicious.

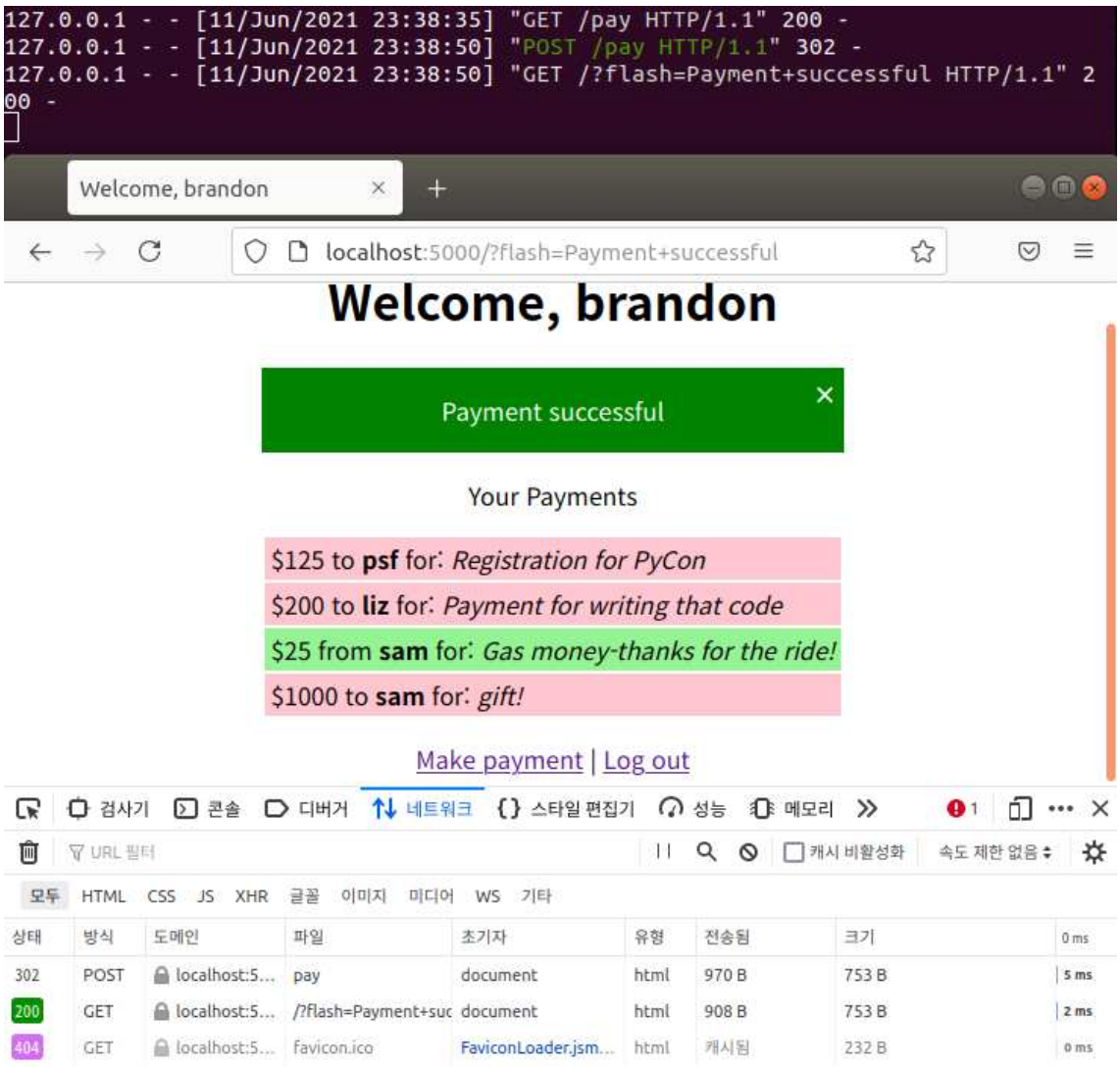
3. Output:
- Verify that brandon has unwillingly made payments to the hacker.

3. Test Cases

- Start : if success to log-in, print attacker's URL on terminal so that Brandon could visit the attacker-generated URL.



- Case 1 : Make payment without attacker's URL
  -> It works normally



- Case 2 : Make payment by visiting attacker's URL
  -> Brandon has unwillingly made payments to the hacker.
  -> The message of the green box displays a custom message chosen by the attacker.