Secure Computer Systems I: Lab 1

Ren Li Tianyao Ma Samuel Pettersson

January 28, 2014

Task 1: SQL injections

Listing 1: Here's a caption.

```
SELECT * FROM Customers
WHERE id = 4 or 1 = 1;
```

The statement shown in Listing 1 is the result of a SQL injection.

The following is a command with symbols with special meaning in LaTeX: grep "report\|open"

Exercises:

(a) Text goes here

(b)

(c)

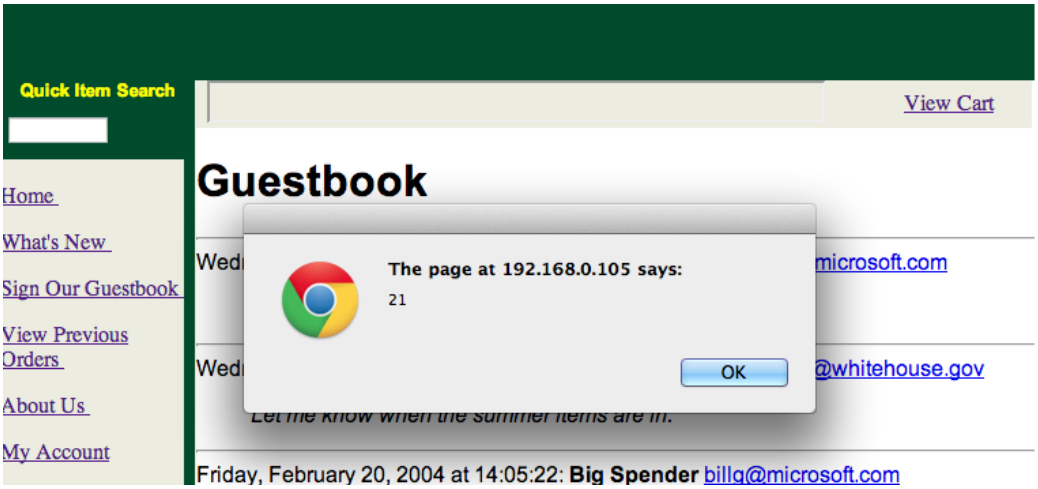
(d)

Task 2: XSS and CSRF

Exercises:

(a) The steps are as follows:

- Open BadStore in your browser.
- Click "Sign Our GuestBook" on the left side of the page.
- In the "Your Name" field, type whatever you want.
- In the "Email" field, type whatever you want.
- In the "Comments" field, type *<script>alert("1")</script>* .
- Click "Add Entry", the page should like this below:



Explanation: In the example, we write a piece of Javascript code in the comment area and submit it. It can work because the website has XSS vulnerability which enable us to insert and perform Javascript in the website. When it reloads, it will generate the current HTML code which contains the code we just injected. And it works as a result.

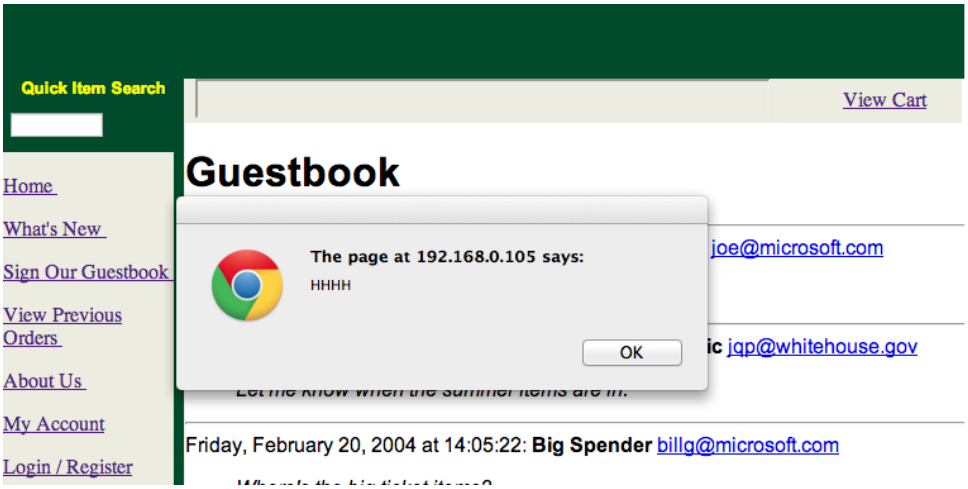
(b) To perform CSRF attacks, we need to create a custom web page contains deceived information. The function is to let user comment in the guestbook and receive countless alert. The key point is to guide victim to visit your page and click on the information. As the reason that we demonstrate it just as an example, we can suppose that the victim will click it. The custom page is so simple that it is just written as a demo. Below are the steps:

- Write a simple page containing these code:

```
<!DOCTYPE html>
<html>
<head>
        <title>iPhone Free!</title>
</head>
<body>
        <FORM METHOD="POST" ACTION="http://10.0.2.5/cgi-bin/badstore.cgi?action=doguestbook">
                <INPUT TYPE=hidden NAME=name value="<script>for(var␣i=0;i<10000;i++){alert('HHHH');}</script>" SIZE=30
                <INPUT TYPE=hidden NAME=email value="1" SIZE=40>
                <TEXTAREA style="display:none" NAME=comments COLS=60 ROWS=4 value="aaa"> </TEXTAREA>
                <INPUT TYPE=submit VALUE="Get␣Now!">
                <INPUT style="display:none" TYPE=reset></Center>
        </FORM>
</body>
</html>
```

- The page shows like this:

- Open BadStore and login in as an user.
- Click the "Get Now!" button in the custom page.
- The page will be redirected into BadStore and shows like this:



- The alert will repeat again and again as well as leaving many comments.
- Note: There might be some different performance in different browsers. i.e. Safari will operate the code after you quit it and reopen it.

Explanation: In the example, we use a self-written page to perform CSRF attack. As mentioned above, the key point is that the attacker need to persuade the victim to visit the deceived page which contains evil code. The code can work with following reasons. First, the code will perform action in the server (BadStore) such as submitting an order or transferring money to another account. Second, the code will work because the victim doesn't close the server website. To make it clear, the cookie of the server website still exists in the browser. As a result, when the server receives the deceived request, it will think that the request is made by the victim and perform related operation.

(c) The difference is that, in an XSS attack, attacker make the attack by inserting the Javascript code inside the site. Because the site do not have strict check on user input, the code can be planted into HTML which will be operated on loading. On the contrary, CSRF attack don't necessarily need Javascript. What count most is that attacker must acquire trust from user and let him to operate the

deceived request. Also, the user can't quit the page because the code will only work when the cookie of the target page stays. It's not the attacker himself that make it work.

(d) [**higher grades only**]

(e) [**higher grades only**]

Task 3: Authentication

Exercises:

(a)

(b)

(c) [**higher grades only**]

-
-

(d)

-
-