CS4238 Homework 2: Cross-Site Scripting (XSS) Attack

The development of this document is funded by the National Science Foundation's Course, Curriculum, and Laboratory Improvement (CCLI) program under Award No. 0618680 and 0231122.

1 Instructions

Due date & time: 27 September 2021, 23:59 SGT. This is an **individual** project. You **MUST** finish the implementation and report independently.

Please convert your report into **PDF** format, and upload it into the LumiNUS workbin before the given deadline. Please include your Matric number in the submitted filename, and the first page of the submission.

2 Project Overview and Goal

Cross-site scripting (XSS) is a type of vulnerability commonly found in Web applications. This vulnerability makes it possible for attackers to inject malicious code (e.g. JavaScript programs) into victim's Web browser. Using this malicious code, the attackers can, among others, steal the victim's credentials, such as cookies. The access control policies (i.e., the same origin policy) employed by the browser to protect those credentials can thus be bypassed by exploiting the XSS vulnerability. Vulnerabilities of this kind can potentially lead to large-scale attacks. In fact, XSS continues to be listed among the top three in the OWASP Top 10 Most Critical Web Application Security Risks [1].

To demonstrate what attackers can do by exploiting XSS vulnerabilities, we have set up a Web-based message board using `phpBB`. We modified this message board software to introduce an XSS vulnerability, which allows users to post any arbitrary message to the board, including JavaScript programs. You need to **exploit this vulnerability** by posting some malicious messages to the message board. Users who view these messages will become victims. The attackers' **final goal** is to post forged messages for the victims.

3 Homework Environment

For this assignment, you will need to set up a network using your VirtualBox, and install the necessary software components on the created VMs.

Network Setup. You can use VirtualBox to create a *NAT network*, to which a few VMS are attached. VirtualBox supports *NAT networking* mode, which is different from *NAT host* mode, since version 4.3 (`http://www.oracle.com/us/corporate/press/2033376`). The networking mode allows you to create a private network, which makes use of private IP addresses, but gives Internet access to the hosts.

At least two VMs are suggested: one machine is used as a Web server, and another machine is used for attacking as well as the victim.

You will need to map some domain names used in this assignment to their respective IP addresses. Since we won't use DNS here, we will just manually modify the `/etc/hosts` files of the relevant machines. For **the Web server**, you will need to add the following line in the starting section of its `/etc/hosts`:

```
127.0.0.1                www.xsslabphpbb.com
```

If you use 2 VMs, with one VM as **both your client's and attacker's machines**, you will need to add the following lines in the VM's `/etc/hosts`:

```
<Web-server-IP-address>    www.xsslabphpbb.com
127.0.0.1                  attacker.com
```

If you use 3 VMs instead, add the following lines in **the client machine's** `/etc/hosts`:

```
<Web-server-IP-address>    www.xsslabphpbb.com
<attacker-IP-address>      attacker.com
```

Then, add the following in **the attacker machine's** `/etc/hosts`:

```
<Web-server-IP-address>    www.xsslabphpbb.com
```

Firefox web browser. You will need to have Firefox web browser on all the VMs. For the attacker's browser, you will use the Developer Tools built in Firefox to inspect the HTTP requests and responses as demonstrated in Lab 4 week 6. Or you are free to use any browser extension you are comfortable with, but need to state clearly in your submission.

Apache web server. The Apache Web server can be installed as follows:

```
% sudo apt-get install apache2
```

If you already installed the Web server, but it is not started by default, you can start the web server using one of the following two commands:

```
% sudo apache2ctl start
or
% sudo service apache2 start
```

`phpBB` **message board Web application.** Now you need set up the `phpBB`. First you need make your system support php5 and mysql:

To install mysql server, you can follow the instruction at `https://www.fosstechnix.com/how-to-install-mysql-5-7-on-ubuntu-20-04-lts/`, you can skip step 4 and the rest of it.

To install php5 and mysql support, run the following command:

```
% sudo apt-get install php5.6 php5.6-mysql
```

Before starting install phpBB forum, we need create a database for it to store related information later. So let's create a database named '`xss_phpbb_db`' and create an account who have full access privilege on this database, with name '`apache`' and password '`apache`':

(Login mysql as root, the password is set when you install mysql-server. If it is null, just press enter.)

```
% mysql -u root -p
mysql> create database xss_phpbb_db;
mysql> GRANT ALL PRIVILEGES ON xss_phpbb_db.* TO 'apache'@'localhost'
       IDENTIFIED BY 'apache' WITH GRANT OPTION;
```

Now we start phpBB installation. Create a folder `XSSLabPhpbb` under the directory `/var/www`, download the file `https://www.comp.nus.edu.sg/%7Ecs4238/downloads/XSSLabPhpbb.tar.gz`, and unpack its files into the `XSSLabPhpbb` directory.

If you haven't done so, map the Web server's domain name `www.xsslabphpbb.com` to this machine's local IP address (`127.0.0.1`) by modifying the Web server's `/etc/hosts` file. Add the following line in the starting section of `/etc/hosts`:

```
127.0.0.1       www.xsslabphpbb.com
```

Next, configure the file `/etc/apache2/site-available/000-default.conf` and add the following lines (in **bold**) at the bottom (before </VirtualHost>):

```
<VirtualHost *:80>
```

ServerName www.XSSLabPhpbb.com
DocumentRoot /var/www/XSSLabPhpbb

```
</VirtualHost>
```

Now you can access the phpBB using the following URL (the Apache server needs to be started first):

```
http://www.xsslabphpbb.com
```

It will show you the phpBB forum installation page. Fill the information according to Figure 1.

Click 'Start Install' button, then choose 'Just send the file to me and I'll FTP it manually'. Download the file `config.php`, and copy it into `/var/www/XSSLabPhpbb`. Delete the sub-directories `install` and `contrib` in `/var/www/XSSLabPhpbb`. Now visit the page `http://www.xsslabphpbb.com` again, you should be able to view the phpBB forum.

Other software. Some of the tasks require some basic familiarity with JavaScript. To complete Task 3, you may need a utility to watch incoming requests on a particular TCP port of the attacker's machine. The utility `nc` (netcat) can be used for this purpose, e.g., `nc -l 8080`.

4 Assignment Tasks

Please complete the following series of tasks for a maximum of 100 marks. Again, please refer to Section 5 below for guidance on what will be looked for in your report.

4.1 Task 1: Posting a Malicious Message to Display an Alert Window (20 marks)

The objective of this task is to post a malicious message containing JavaScript that will display an alert window showing the string "`Hello Web!`". In your crafted message, the JavaScript should be provided along with the user comments.

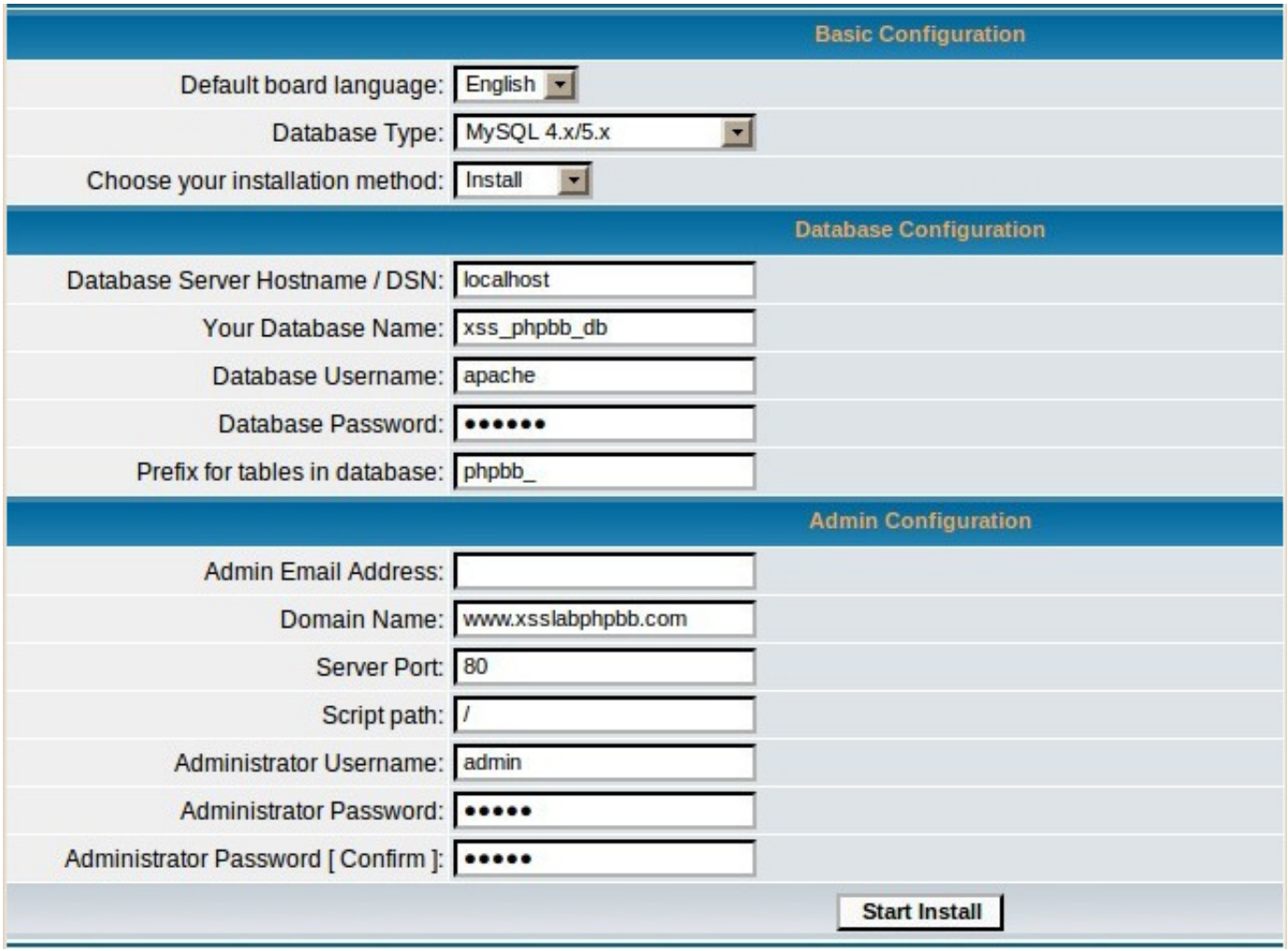

Figure 1: phpBB installation page

4.2 Task 2: Posting a Malicious Message to Display Cookies (10 marks)

The objective of this task is to post a malicious message containing a JavaScript code onto the message board, such that whenever a user views this message, the user's cookies will be printed out. For your information, the JavaScript statement `document.cookie` retrieves the value of the cookie of the current Web session.

4.3 Task 3: Stealing Cookies from the Victim's Machine (20 marks)

In the previous task, the malicious JavaScript code prints out the user's cookies. In this task, the attacker wants the JavaScript code to send the cookies to the himself/herself. To achieve this, the malicious JavaScript code can send an HTTP request to the attacker, with the cookies appended to the request. Tha attacker can do this by having the malicious JavaScript insert an `<img>` tag with its `src` attribute set to the URL of the attacker's destination. An example is: `<img src="http://attacker.com:5555/abcd">`. When the JavaScript inserts the `img` tag, the browser will try to load the image from the given URL. As a result, the process ends up sending an HTTP GET request to the attacker's website. The JavaScript given above

```
http://www.xsslabphpbb.com/posting.php

POST /posting.php HTTP/1.1
Host: www.xsslabphpbb.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686;
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.xsslabphpbb.com/posting.php?mode=newtopic&f=1
Cookie: phpbb2mysql_data=......;phpbb2mysql_sid=......
Content-Type: application/x-www-form-urlencoded
Content-Length: 376
subject=<Content of the message>


HTTP/1.x 200 OK
Date: Thu, 11 Jun 2009 19:43:15 GMT
Server: Apache/2.2.11 (Ubuntu) PHP/5.2.6-3
X-Powered-By: PHP/5.2.6-3ubuntu4.1
Set-Cookie: phpbb2mysql_data=XXXXXXXXXXX; expires=Fri, GMT; path=/
Set-Cookie: phpbb2mysql_sid=YYYYYYYYY; path=/
Set-Cookie: phpbb2mysql_t=XXXXXXXXXXX; path=/
Cache-Control: private, pre-check=0, post-check=0, max-age=0
Expires: 0
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3904
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
```

Figure 2: Sample screenshot of the HTTP request and response

will send the request for `abcd` to the port 5555 of the server `attacker.com`. On that port, the attacker has a TCP server that simply prints out the request it receives. The server can be achieved by the `nc` utility.

4.4 Task 4: Impersonating the Victim User using the Stolen Cookies (35 marks)

After stealing the victim's cookies, the attacker can do whatever the victim user can do to the `phpBB` Web server, including posting a new message in the victim's name, delete the victim's post, etc. In this task, we will write a program that forges a message post on behalf of the victim.

To forge a message post, we should first analyze how `phpBB` works in terms of posting messages. More specifically, our goal is to figure out what are sent to the server when a user posts a message. Firefox's Developer Tools can help us with this. It can display the contents of any HTTP request message sent from the browser. From the contents, we can thus identify all the the parameters of the message. A screenshot of the HTTP request and response sample is given in Figure 2.

Once understanding what the HTTP request for message posting looks like, the attacker can send out the same HTTP request by his/her own programs. The `phpBB` server cannot distinguish whether the request is

sent out by the user's browser or by the attacker's programs. As long as the attacker sets all the parameters correctly, the server will accept and process the message-posting HTTP request. This task can be done by a C or Java program, a Python script, utilities such `nc`, `curl`, or the add-on like `TamperData`.

If you want to write your own program, it consists of the following steps:

1. Opens a connection to Web server;
2. Sets the necessary HTTP header information;
3. Sends the request to Web server;
4. Gets the response from Web server.

Limitation: Note that the forged message post should be sent from *the same host* as the host used by the victim (i.e. user connected to the Web forum). This is because `phpBB` uses both IP address and the cookies for session management. If the attacker generates the forged message post from a different host, the IP address of the forged packet and the victim's IP address would differ. Therefore, the forged message post would be rejected by the `phpBB` server, despite the fact that the forged message carries the correct cookie information. Hence, for this task, you will be asked to just send the message from the victim's host.

4.5 Task 5: Addressing the Limitation of Victim Impersonation Attack (15 marks)

Given the limitation of Task 4, your task here is to discuss ways of addressing the limitation. That is, discuss how the attacker may find workaround(s) that will allow him/her to send the forged message from his/her own machine instead of the victim's. First, you can assume a scenario whereby the victim's machine and host's machine are on the same network segment so that sniffing will work. Subsequently, you can consider a more difficult but general case, where the attacker is on a different network from that of the victim user. For this task, you will thus need to apply your network-attack knowledge that you have learnt before.

5 Grading Criteria

Please find below the grading criteria for this homework. Do follow them when writing your report.

General notes:

1. All necessary code should be supplied.
2. Please provide details using `Firefox Developer Tool`, `Wireshark`, and/or screenshots and code.
3. Where it says "screenshot" below, it may be read as "screenshots" if necessary.
4. You also need to provide explanation to the observations that are interesting or surprising.
5. All explanations should be sufficiently detailed.

Task 1 (20 marks):

1. Include the screenshot of the alert window, and the message including JavaScript code.

Task 2 (10 marks):

1. Include the screenshot of the shown cookies, and the message including JavasSript code.

Task 3 (20 marks):

1. You need to write a program or use a tool to receive the data sent by the attacker. Describe and explain the used program/tool.
2. Include the screenshot showing the stolen cookies on the attacker side, and the employed Javascript code.

Task 4 (35 marks):

1. Explain the importance of the HTTP request components in forging a message.
2. Include the screenshot and description of how to post a message on behalf of the victim.

Task 5 (15 marks):

1. Explain a possible workaround that enables an attack on the same-network scenario.
2. Explain possible ways to address the general case of different-network scenario.

References

[1] OWASP Top 10 for 2017 Release Candidate. Available at the following URL: `https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#tab=OWASP_Top_10_for_2017_Release_Candidate`.