**Assignment 4**

Team members:

Chanip Chong (012548263)

Eugene Clewlow (003416986)

**Stored XSS Attack**

I performed this code by posting a segment of JavaScript code on a website message or posting where everyone is able to see them and trigger the code when they enter the URL. Basically, the malicious code will be stored in the server’s database exactly as the user input if the server doesn’t do anything to prevent this stored XSS attack. When other user enter the URL which contains a public element that get/show the malicious mode from database, the user’s browser will also execute this malicious code, so, the attack will succeed.

**Low level:**

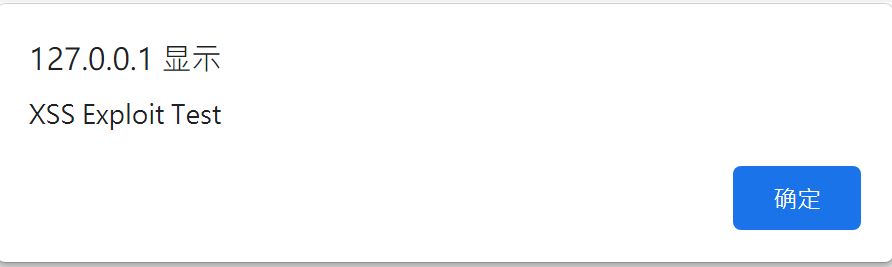
I posted a new message to the forum or the website with the code below:

Name: abc

Message:

<script>alert("XSS Exploit Test")</script> .

This JavaScript code will be interpreted by the browser every time when someone open the URL, the alert box will be popped up immediately. Attackers can use other code to achieve their goals too.

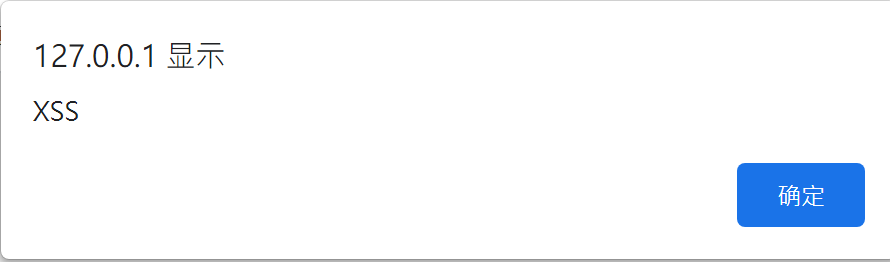
****

**Medium level:**

When I switch to medium level. This method doesn’t work anymore. I tried <Script>alert("XSS Exploit Test")</Script> and <Scrip<Script>t>alert("Test")</Scrip</Script>t>, but they all fail. The system check the <script> tag in Message field and delete them before storing into the database.

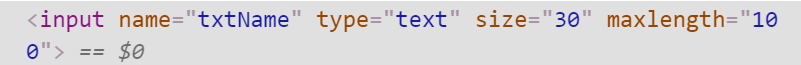
I changed to input the malicious code to the Name field, and it succeeded with the following code:

<scr<script>ipt>alert("XSS")</script>



Obviously, the server (medium security level )doesn’t check <script> tags in Name field.

In order to input this long code to the Name field, I needed to open the developer tools in Chrome and enlarge the max length of input.



**High level:**

When I switch to high level. The above methods don’t work at all, because the server will block every <script> tags in any field; Therefore, I tried use HTML tags to inject the malicious code in the Name field:

<img src= x onError=alert("XXX")>



Since the server only block <script> tags, we can expect that HTML, CSS are possible to achieve this attack too.

reference: <http://www.computersecuritystudent.com/SECURITY_TOOLS/DVWA/DVWAv107/lesson9/index.html>

<https://www.youtube.com/watch?v=ivvTrTie16I>