Lab: Web – XSS (Cross Site Scripting)

# Problem 1: Use XSS Attack to steal messages being stored by *PersistentXSS* application

This problem is based upon the ***PersistentXSS*** and **StealMessage** web applications included in the lab zip file. Extract these files to Windows. Add both projects to NetBeans.

PersistentXSS is a basic web application that allows users to log in and post messages. Examine the code to determine the two (hard coded) accounts (abe/password and bob/password).

Try PersistentXSS to get a feel for how it works.

Try StealMessage to get a feel for how it works too.

The StealMessage application is going to be used to steal a copy of the messages posted by the PersistentXSS application. Abe is going to post a friendly message (“Hi Bob!”) along with an injection of some JavaScript into the messages in PersistentXSS. This JavaScript will update the "action" of the form so that it submits to the StealMessage application instead of the correct location. Bob is then going to log in and post a message. He is unaware that his messages are being copied into StealMessage.

Examine the messages.txt file in both \PersistentXSS\build\web\WEB-INF \StealMessage\build\web\WEB-INF

It is not a requirement for StealMessage to reside on the same server as PersistentXSS, but we use this configuration since it is easy to demonstrate.

What is the JavaScript injection exploit?

# Problem 2: Harden the application against the attack

Once you have successfully attacked the application as described in problem 1 you should make a copy of it called ***PersistentXSS\_NoStealMessages***. This version of the application should not allow the attack to occur because it validates user input and rejects input that contains anything that could be used maliciously against the application.

There are many ways that input can be checked to ensure it does not contain anything dangerous. Try to solve the problem in two different ways.

Upload your code to the discussions.