**CEG4430/6430 Homework 2**

**(undergraduate students)**

(30 Points)

1. Please read the source code for index.php hosted in [www.testexample.com](http://www.testexample.com) (i.e., [www.testexample.com/index.php](http://www.testexample.com/index.php) ). A benign example would be: [www.testexample.com/index.php?name=adam](http://www.testexample.com/index.php?name=adam).

<?php

$name = $\_GET['name'];

echo "Welcome $name<br>";

echo "Have Fun!";

?>

1. **Question:** Which one of the following terms most precisely describe the vulnerability for this index page? (5 Points) and Please justify. (5 Points)
   1. Persistent XSS
   2. Non-persistent XSS
   3. SQL injection

Answer : Based on the code that is presented, it appears that the type of attack would most likely represent a Non-persistent XSS attack. To clarify, by looking at the differences between the code written for examples of Persistent XSS and Non-persistent XSS attacks, it proves that a Non-persistent XSS protocol is what is being used. While Persistent XSS is used to attack a large group as a target, this represents more like targeting a single entity as a target and this is what is used for a Non-persistent XSS attack.

1. **Question:** Please create an attack to send a victim’s cookie with [www.testexample.com](http://www.testexample.com) to [www.attack.com/collect.php](http://www.attack.com/collect.php) (controlled by an attacker). Let us assume that Javascript is enabled in the victim’s browser and “var x = document.cookie;” can be used to read the cookie. Feel free to write pseduo code if you are not familiar with Javascript syntax. But make sure your input is legible. (15 Points)

Answer : Spoudo Code for Result:

<Start\_Of\_JS\_Script> Cookie\_Attack\_Function()<End\_Of\_JS\_Script>

Here I have a JS function called Cookie\_Attack(); This finction is hosted on an attackers site ([www.attack.com/collect.php](http://www.attack.com/collect.php)) this site is hosted on the same donaim as the victum site ([www.testexample.com](http://www.testexample.com)). Then apone activating my Cookie\_Attack function, this allows the victum site cookies to be copied and data sent over to the attacker website to be collected and read to obtain any information that the cookie contains.

1. Explain how the heartbleed vulnerability occurred? Is the implementation of the SSL/TLS vulnerable or the protocol itself vulnerable? And why? (5 Points)

Answer : The heartbleed bug that at the time of release effected 17% (around half a million) of the worlds secure websites, seemed to be caused from the implementation of the SSL/TLS protocol. The vulnerability itself occurred from the OpenSSL library having a problem with memory buffers.

Link to source regarding Heartbleed (Always cite sources of info): <https://www.csoonline.com/article/3223203/what-is-the-heartbleed-bug-how-does-it-work-and-how-was-it-fixed.html>

From this link, I found that the Heartbleed bug occurred from a problem with memory buffers and the length of data being copied. Specifically, when the OpenSSL Library would copy data over, there was no check to ever see if the length of data that was copied was the same as the length of data from where it was copied from. This is what created the memory overflow and the heartbleed took effect.