Session Six – Web Input Vulnerabilities

This session features an introduction to exploitation of web servers through input-based attacks. Web servers need to accept legitimate traffic over HTTP and this leaves them potentially vulnerable to user input that may cause the website to have unexpected behaviour.

We will be looking at a couple of key ways to exploit this through SQL Injection and Cross Site Scripting, which will provide the basic skills for an open ended web exploitation challenge in a future session.

Disclaimer This is a massive area of cyber security and the tasks mentioned here merely provide an oversight into the general area.

Part 0 – (Optional) Background Reading

For those with no experience in HTML/Javascript a good overview to the key points can be found here. While beyond the scope of this session, the free tutorials on CodeCadamy here are excellent for getting to grips with the languages involved.

Part 1 – SQL Injection

Structured Query Language (SQL) is a language used to query databases. Many web servers will take user input and use it to construct a query to the database in this language. By changing our input to contain embedded statements in this language we can take advantage of this if the server is not adequately protected.
Head to This SQL demo and work through the steps to understand the practical application of this type of attack. You will need to be able to carry out SQL based attacks for future CyberBulls practicals.

Part 2 – Cross Site Scripting (XSS) Challenge

The front end (display side) of a web site can also be vulnerable to input based attacks, especially from Javascript code execution, known as XSS. This typically arises anywhere where user input will later be displayed on screen, such as a comment post etc. This could be as simple as inputting html tags such as <script></script> with malicious code inside that will then be ran when the input is rendered.

Take on the XSS Challenge and see how far you can get – this gets pretty fiddly pretty quickly so major props to anyone who is able to exploit all 6 levels.

Part 4 – Extensions

Getting through all levels of the XSS challenge should be pretty hard in itself, but if you fancy an extra challenge try doing it without viewing any of the source code. This better simulates trying to exploit a web server for real as you may not be able to see what code it is running on the server side and as such a bit of educated guessing is required to work out what makes the site tick.