

Sunday, January 10, 2010

SQLi Detection - Duh Moment

Not sure why it took me so long to figure out what I am sure is obvious to most other people who have thought about this, but it never clicked for me how to get anywhere near useful SQL Injection detection. The injection itself is trivial, of course, but determining whether it actually worked and weeding out false positives in an automated manner was something that seemed too hard.

During my run on Friday I had a Duh! moment on it. Annoyingly simple. Do it in 3 requests. Request #1 is a normal request. For example, "?id=1" in the URL. If the id is being passed to an SQL request it will return a single record or perhaps no record, it doesn't really matter. Now on request #2 do "?id=1 or 3=4", that is, inject a false 'OR' condition. If the output changes, we are done. Nothing to see here. However, if the output does not change we send request #3 with "?id=1 or 3=3" and if that output differs from request #2 then we have a potential SQLi situation. There are of course still chances of false positives (and negatives) with page stamps and such, but filtering out the response headers and html comments cuts down on that a bit. Add different combinations of single and double-quotes, like "?id=1'or'3='3" (without the double-quotes, of course) and it might be able to catch something.

The best thing about it is that it can slide into an existing scanner framework quite easily. If you have a base reference request, then it just adds a single request to the common case where the false 'OR' condition output does not match the base reference. You only need to do the true 'OR' condition request in case it does match.

Anybody have any other approaches?

Posted by Rasmus in Software at 18:44