

How To Prevent Disastrous SQL Injection Attacks

One of the biggest flaws in the PHP language is the fact that it allows for web developers to make very big mistakes in regards to security. One example of this is through SQL injections- an exploit that malicious users take advantage of when web developers don't accurately safeguard their application.

An SQL injection attack is, simply put, a vulnerability in the SQL query that programmers unwittingly leave wide open. When a web developer calls an SQL query, he or she will commonly forget to escape quotes that the user might input. Users might input text such as "MyVariable' OR 1=1--" ; this line will actually give the malicious user to your database!

As long as we can escape the quote that needs to be used in the injection, we can prevent any type of harm that may come to a web application. The first way to accomplish this is to simply use magic quotes. It should be noted that magic quotes are no longer supported as of PHP 6, and shouldn't be used. Instead, we leave SQL injection prevention up to a newer and more dependable command.

Using the "mysql_real_escape_string()" function will enable web developers to escape quotes properly. And unlike magic quotes, this function will only escape quotes that we need. Keep in mind that when using this function, it may be necessary to use the "stripslashes()" function to counteract the slashes that are being outputted as a result.

Another good way to prevent SQL injections is to simply restrict authority in SQL users where possible. For instance: it would be a good idea to create individual users that do specific things: such as create a table or update rows in the said table. This can help make the task of ruining one's hard work much harder for malicious web users, although it's a lot more work for webmasters (Although well worth it).

It should be noted that programs and web applications that stop SQL injections should not be obtained- since they commonly cost quite a bit of money. As long as webmasters take precautions with what they create, there should be no reason to spend hundreds of dollars on software that only makes use of escape characters and formatting data correctly. This type of application is created to con webmasters into buying something they don't need- so dont fall victim to them!

Closing Comments

There isn't much effort that needs to be exerted in order to declare a database safe from harm. All that is needed is a little prevention- which comes from avid usage of the function and design principles previously stated. It may also be a good idea to use SQL injection scanners on large web applications to cover holes that might not have been covered over the course of the development period.

About the Author

Learn more on [Click Here](#) and [Preventing Injection String](#).

Source: <http://www.zogol.com>