Loving QueryParam Scanner

Posted At : September 18, 2008 7:30 AM | Posted By : Steve
Related Categories: ColdFusion

When the recent spat of SQL Injection attacks starting hitting the ColdFusion community recently, I actually felt pretty safe. I thought to myself, "I have been using <cfqueryparam> for years. Every query that I have written since before I started working for myself has been safe from SQL injection attacks."

In good news, I was right about that. What I failed to consider, however, was that I also maintain legacy code written by others. Including code written well before the use of <cfqueryparam> was popularized.

A client of mine recently asked me to edit some of the old code. In this case, the original author was requesting permission to modernize the application when I started working with them - five years ago. We actually have updated a lot of it, but some old code still remains.

When I looked at the code, I was first struck by what a nasty, tangled mess it is with <cfquery> scattered throughout the output code. I braced myself to ignore the mess. After all, I was only asked to make a small change - no real way to justify a major clean-up effort for that. Then I was it.

```
<cfquery name="foo" ...>
SELECT *
FROM mytable
WHERE myvalue = #URL.id#
</cfquery>
```

A vulnerability to SQL Injection. I can ignore messy code, but not a security problem. I called my client and quickly got approval to secure all of the queries.

I imagined searching through all of the messy code to find all of the vulnerable queries. It seemed like a nightmare. Fortunately, I remembered reading about **QueryParam Scanner**.

I downloaded it, copied the files to the server on which I wanted to check the code and opened up the newly created folder in my browser. With no more effort than that, it was working. I put in the path to the code and ran the search. After a few seconds, it reported all of the potential vulnerabilities that it found. While this had some false positives (#Val(URL.id)# not really being a vulnerability, for example), it still made the process quick and painless.

It provided the file names and line number of the vulnerabilities and even (I discovered a few minutes in), allowed me to see the cfquery statement directly from the report.

I don't want to think about how long that process would have taken without QueryParam Scanner.

So, thanks **Peter Boughton**!