The Coldfusion Hash() function decoded - kind of

Posted At : 23 January 2011 22:29 | Posted By : Shaun McCran
Related Categories: Security, Coldfusion, Best practices

I've always believed that using the hash() function in ColdFusion is a one way process. If I wanted to reverse a string I had to use encode() and decode(). The Adobe documentation states that "It is not possible to convert the hash result back to the source string" - Adobe Docs for Hash() .

Strictly speaking this is still true, but some bright spark has decided to host an MD5 string database and provide a lookup service.

The Hash() function has been around for a long time, in pre ColdFusion 7 versions of it you could not specify an algorithm, so you could only encode to MD5 standards.

In most cases the algorithm did not really matter too much. Most developers would have used hash() to store a password and perform real time character checks against the database values when a user submits a password string.

```
Hash(string [, algorithm [, encoding ]])

<cfset variables.encodedValue = hash('myPassword')>

<cfoutput>#variables.encodedValue#</cfoutput>

Results in:
DEB1536F480475F7D593219AA1AFD74C
```

You could 'in-effect' never actually get a password back again, only perform a check against it using other hash()-ed strings.

The site http://www.md5decrypter.com/ appears to be hosting a database containing '8,076,999 unique MD5 hashes'. I've tested over a dozen random strings and they have all been successfully returned from their search.

It is probably more of a legacy application issue, but it is definitely worth noting that you really should specify an algorithm type when using the hash() function now.

This really makes a good case for revisiting those 'old' applications that never get a budget for bringing up-to-date.