Instantly share code, notes, and snippets.

Stacksmashers101 / sqli to rce

Created last year

☆ Star

<> Code -o- Revisions 1

<> sqli to rce

```
Injection attacks occur when data is sent to an interpreter which contain unintended commands with the data that are run by the interpreter
the same can be performed in b2evolution CMS 7.2.3 in the User Registration section, leading to remote code execution via SQL Injection (SQ
In the earlier Database chapter you saw the use of the cfqueryparam tag. It is one of the simplest steps you can take to help prevent SQL i

<cfparam name="url.state" default="" />
<cfparam name="url.orderby" default="LASTNAME" />
<cfquery name="request.listing" datasource="cfartgallery">
SELECT      FIRSTNAME, LASTNAME, EMAIL, THEPASSWORD, ADDRESS
FROM        table
WHERE       1=1
<cfif Len(url.state)>
AND         STATE = <cfqueryparam cfsqltype="cf_sql_varchar" value="#url.state#" />
</cfif>
ORDER BY    #url.orderby#
</cfquery>

By validating the URL parameters against a list of values we know to be good while changing all references from the URL scoped variables to

<cfscript>
    // list of valid values
    variables.validStates = "CA,CO,DC,FL,GA,NM,NY,OK,SD";
// check what was passed against list of valid values
if ( StructKeyExists(url, "state") AND ListFind(variables.validStates, url.state) ) {
    variables.state = url.state;
} else {
    variables.state = "";
}

if ( StructKeyExists(url, "orderby") AND ListFind(variables.validOrderBys, url.orderby) ) {
    variables.orderby = url.orderby;
} else {
    variables.orderby = "LASTNAME";
}

</cfscript>

While the above code removes the SQL injection, it can be made better. First, remove the hard coded list of States and reuse the query used
```