MOPS-2010-005: ClanSphere MySQL Driver Generic SQL Injection Vulnerability

May 3rd, 2010

A generic SQL Injection vulnerability was discovered in the MySQL Driver of ClanSphere that allows exploiting a lot of otherwise safe SQL queries.

Affected versions

Affected is ClanSphere <= 2009.0.3

Credits

The vulnerability was discovered by Stefan Esser during the Month of PHP Security SQL Injecton Marathon.

About ClanSphere (translated)

ClanSphere is a modern and modular Web-CMS that allows to completely organise clans, clubs and other kinds of groups. It is a PHP script that is open source and free of costs for now. It is released under the new bsd license.

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 ClanSphere was also audited and within 30 minutes it was possible to find multiple SQL injection vulnerabilities. The first one is located in the Captcha generator. However the second one is a generic SQL injection vulnerability located in the MySQL database driver of ClanSphere. The offending code is the following.

```
function cs_sql_select($cs_file, $sql_table, $sql_select, $sql_where = 0, $sql_order = 0, $first = 0, $m
{
  if (!empty($cache) && $return = cs_cache_load($cache)) {
    return $return;
  }

  global $cs_db;
  settype($first, 'integer');
  settype($max, 'integer');
  $run = 0;
  $sql_where = str_replace('"', '', $sql_where);

  $sql_query = 'SELECT ' . $sql_select . ' FROM ' . $cs_db['prefix'] . '_' . $sql_table;
  if (!empty($sql_where)) {
    $sql_query .= ' WHERE ' . $sql_where;
  }
...
```

One can see here that when the SQL query is build all double quotes are just removed from the query. This is however dangerous because it causes problems with argument escaping. The easiest example is an escaped double quote at the end of a user supplied value. Because only the double quote is removed the escaping character \ stays and will be interpreted as the escaping of the following single quote that is supposed to terminate the string context. The SQL parser will interpret everything until the next single quote in the query as string data and the string data of the next value will be wrongly interpreted as SQL statement. Because of this all SQL SELECT statements are vulnerable to SQL injection if there are two user supplied values used after each other.

Example vulnerable statement:

$sql = "SELECT * FROM user WHERE username='$safeUser' AND pass='$safePass'";

Proof of concept, exploit or instructions to reproduce

No proof of concept exploit will be provided for this vulnerability.

Notes

This vulnerability did not affect the MySQLi, the PDO, the postgres, sqlite and sqlsrv database drivers.

The vendor already released a fixed version of ClanSphere 2009.3.1. However in his release announcement he claims that for MOPS we checked ClanSphere very deeply by using the german expression “auf Herz und Nieren geprÃ¼ft”. This is however not true. ClanSphere like any other application was only checked for less than 30 minutes. You should always remind that when looking at the vulnerabilities found during SQL Injection Marathon. Each of these vulnerabilities was found in less than 30 minutes so they were easy to find and therefore most probably long known by blackhats.