LIKE Wildcard Injection in Search.php — Unescaped % and _ Metacharacters Enable Broad Content Disclosure

Moderate **thorsten** published **GHSA-gcp9-5jc8-976x** last week

Package

php **thorsten/phpmyfaq** (Composer)

Affected versions

4.1.0

Patched versions

4.1.1

Description

Summary

The `searchCustomPages()` method in `phpmyfaq/src/phpMyFAQ/Search.php` uses `real_escape_string()` (via `escape()`) to sanitize the search term before embedding it in LIKE clauses. However, `real_escape_string()` does **not** escape SQL LIKE metacharacters `%` (match any sequence) and `_` (match any single character). An unauthenticated attacker can inject these wildcards into search queries, causing them to match unintended records — including content that was not meant to be surfaced — resulting in information disclosure.

Details

File: `phpmyfaq/src/phpMyFAQ/Search.php`, lines 226–240

Vulnerable code:

```
$escapedSearchTerm = $this->configuration->getDb()->escape($searchTerm);
$searchWords = explode(' ', $escapedSearchTerm);
$searchConditions = [];

foreach ($searchWords as $word) {
    if (strlen($word) <= 2) {
        continue;
    }
```

```
    $searchConditions[] = sprintf(
        "(page_title LIKE '%%%s%%' OR content LIKE '%%%s%%')",
        $word,
        $word
    );
}
```

`escape()` calls `mysqli::real_escape_string()`, which escapes characters like `'`, `\`, `NULL`, etc. — but explicitly does **not** escape `%` or `_`, as these are not SQL string delimiters. They are, however, LIKE pattern wildcards.

Attack vector:

A user submits a search term containing `_` or `%` as part of a 3+ character word (to bypass the `strlen <= 2` filter). Examples:

- Search for `a_b` → LIKE becomes `'%a_b%'` → `_` matches any single character, e.g. matches `"aXb"`, `"a1b"`, `"azb"` — broader than the literal string `a_b`
- Search for `te%t` → LIKE becomes `'%te%t%'` → matches `test`, `text`, `te12t`, etc.
- Search for `_%_` → LIKE becomes `'%_%_%'` → matches any record with at least one character, effectively dumping all custom pages

This allows an attacker to retrieve custom page content that would not appear in normal exact searches, bypassing intended search scope restrictions.

PoC

1. Navigate to the phpMyFAQ search page (accessible to unauthenticated users by default).
2. Submit a search query: `_%_` (underscore, percent, underscore — length 3, bypasses the `<= 2` filter).
3. The backend executes: `WHERE (page_title LIKE '%_%_%' OR content LIKE '%_%_%')`
4. This matches **all** custom pages with at least one character in title or content — returning content that would not appear for a specific search term.

Impact

- **Type:** LIKE Wildcard Injection → Information Disclosure (CWE-943 / CWE-20)
- **Severity:** Moderate
- **Authentication required:** None — search is publicly accessible
- **Affected component:** `searchCustomPages()` in `Search.php`; custom pages (faqcustompages table)
- **Impact:** Unauthenticated users can enumerate/disclose all custom page content regardless of the intended search term filter
- **Fix:** Escape `%` and `_` in LIKE search terms before interpolation:

```
$word = str_replace(['\\', '%', '_'], ['\\\\', '\\%', '\\_'], $word);
```

Or use parameterized queries with properly escaped LIKE values.

Severity

Moderate

CVE ID

CVE-2026-34973

Weaknesses

No CWEs

Credits

athuljayaram Reporter