

# fix(expressions): avoid helper eval in literal checks #7321

Merged ehsandeeep merged 2 commits into dev from dwiswant0/fix/expressions/avoid... last week

Conversation 7 Commits 2 Checks 20 Files changed 6

dwiswant0 commented 3 weeks ago • edited by coderabbitai bot Member

## Proposed changes

fix(expressions): avoid helper eval in literal checks

hasLiteralsOnly() currently evaluates helper expressions while deciding whether "{...}" contains unresolved variables, which makes validation paths run side-effectful helpers.

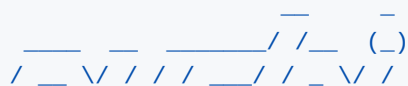
Just replace that runtime eval with a vars() len check so unresolved-variable detection literally stays literal (am I writing it right?), and of course side-effect free.

Also make Evaluate() return template-authored expression compile/eval errors instead of logging and then keep continuing, so malformed helper calls still fail in the rendering path.

Fixes #7320

## Proof

```
$ time ./bin/nuclei -t eval-while-validating.yaml -u "http://localhost:8088/" -var "fur it
```





## Checklist

- Pull request is created against the [dev](#) branch
- All checks passed (lint, unit/integration/regression tests etc.) with my changes
- I have added tests that prove my fix is effective or that my feature works
- I have added necessary documentation (if appropriate)

## Summary by CodeRabbit

- **Bug Fixes**

- Expression evaluation now fails immediately with clear error messages instead of skipping, and avoids leaking resolved secret values.
- Helper/template-like syntax embedded in resolved inputs is no longer executed.
- Request argument processing aborts correctly on argument-evaluation errors.

- **Tests**

- Added tests covering expression error handling, helper isolation, unresolved-variable behavior, and argument-evaluation failure handling.



**auto-assign** (bot) requested a review from **dogancanbakir** [3 weeks ago](#)

**coderrabbitai** (bot) commented [3 weeks ago](#) • edited ▾

Contributor

## Walkthrough

Expression evaluation now returns errors immediately on compile/eval failures and avoids executing expressions to determine "literals-only." Unresolved-marker replacement is gated by presence of unresolved markers and function tokens. Tests added to ensure helpers are not executed during unresolved-variable checks and to validate error messaging.

## Changes

Cohort / File(s)	Summary
<b>Evaluation &amp; Replacement</b> pkg/protocols/common/expressions/expressions.go	Evaluate now fails fast on compile/eval errors (returns wrapped error). Replacement may be set to unresolved-marker only if unresolved markers exist and the

Cohort / File(s)	Summary
	compiled expression contains govaluate function tokens; incremental placeholder replacement uses the computed replacement .
<b>Validation (literals-only check)</b> pkg/protocols/common/expressions/variables.go	hasLiteralsOnly treats nil as literals-only and determines literals-only by checking len(expr.Vars()) == 0 instead of executing the expression.
<b>Tests &amp; Test Helpers</b> pkg/protocols/common/expressions/expressions_test.go , pkg/protocols/common/expressions/variables_test.go	Added tests covering: helpers not executed during unresolved-variable checks, error messages for invalid template expressions (no leakage of resolved secret values), and behavior with marker-like substrings. Added withTestHelperFunction test helper.
<b>JS Request error path &amp; tests</b> pkg/protocols/javascript/js.go , pkg/protocols/javascript/js_test.go	Request.getArgsCopy now returns immediately on arg-evaluation error (records failure and increments failed requests). New test ensures ExecuteWithResults returns an evaluation error (no panic) when args contain invalid expression like {{base64()}} .

## Estimated code review effort

 4 (Complex) |  ~45 minutes

## Poem



I nibbled templates under moonlight bright,  
 Kept helpers sleeping, avoided a fright.  
 Errors call out, secrets tucked tight,  
 Markers stay markers, functions take flight.  
 Hooray for tidy burrows tonight! 🥕

▶ 🚦 Pre-merge checks | ✅ 4 | ❌ 1

▶ ✨ Finishing Touches

Comment [@coderabbitai help](#) to get the list of available commands and usage tips.

**dwiswant0** requested a review from **Ice3man543** [3 weeks ago](#)

**coderabbitai** (bot) reviewed [3 weeks ago](#)

[View reviewed changes](#)



**coderabbitai** (bot) left a comment

Contributor

**Actionable comments posted: 2**

▶ 🤖 Prompt for all review comments with AI agents

▶ ✨ Autofix (Beta)

▶ ⓘ Review info

> pkg/protocols/common/expressions/expressions.go (Outdated) Show resolved

> pkg/protocols/common/expressions/expressions.go (Outdated) Show resolved

fix(expressions): avoid helper eval in literal checks ❌ [a3f3c99](#)

**dwiswant0** force-pushed the [dwiswant0/fix/expressions/avoid-helper-eval-in-literal-checks](#) branch from **0734d48** to **a3f3c99** [3 weeks ago](#) [Compare](#)

**coderabbitai** (bot) reviewed [3 weeks ago](#)

[View reviewed changes](#)



**coderabbitai** (bot) left a comment

Contributor

- ▶ Nitpick comments (1)
- ▶ Prompt for all review comments with AI agents

---

- ▶ Review info



[fix\(javascript\): avoid nil map panic on arg eval failure](#) ...

✓ [64721d8](#)

**neo-by-projectdiscovery-dev** (bot) commented [3 weeks ago](#) • edited ▾

## Neo - PR Security Review

**No security issues found**

Comment [@pdneo help](#) for available commands. · [Open in Neo](#)



**coderabbitai** (bot) reviewed [3 weeks ago](#)

[View reviewed changes](#)



**coderabbitai** (bot) left a comment

Contributor

- ▶ Nitpick comments (1)
- ▶ Prompt for all review comments with AI agents

---

- ▶ Review info



**ehsandeep** approved these changes [last week](#)

[View reviewed changes](#)



**ehsandeep** merged commit **6c803c7** into `dev` [last week](#)

32 of 33 checks passed

[View details](#)



**ehsandeep** deleted the `dwiswant0/fix/expressions/avoid-helper-eval-in-literal-checks`

branch [last week](#)

[Sign up for free](#) to join this conversation on **GitHub**. Already have an account? [Sign in to comment](#)

### Reviewers



**coderabbitai[bot]**



**ehsandeep**



**dogancanbakir**



**Ice3man543**



### Assignees

No one assigned

### Labels

None yet

### Projects

None yet

### Milestone

No milestone

### Development

Successfully merging this pull request may close these issues.



**[BUG] validation executes DSL helpers while checking unresolved variables**

### 2 participants

