fix(expressions): only eval template-authored expressions #7221

Merged **dwisiswant0** merged 2 commits into `dev` from `dwisiswant0/fix/expressions/only-…` on Mar 16

dwisiswant0 commented on Mar 14 • edited by coderabbitai bot — Member

Proposed changes

`expressions.Evaluate()` currently replaces placeholders first and then scans the substituted output for expressions which allows response-derived values, including extractor output, to be reinterpreted as DSL/helper syntax on a second pass, which then becomes more serious when env-backed variables are enabled.

So making this by collecting expressions from the original template text before placeholder substitution and then evaluating only those original expressions after normal replacement to keep trusted template-authored helpers working, including placeholders nested inside helper arguments while treating substituted response data as literal text.

Fixes #7210

Proof

```
$ go test -v -count 1 -run ^TestEvaluateDoesNotReinterpretResolvedValues$ ./pkg/protocc
=== RUN   TestEvaluateDoesNotReinterpretResolvedValues
=== RUN   TestEvaluateDoesNotReinterpretResolvedValues/helper_syntax_in_resolved_values_stays
```

```
=== RUN   TestEvaluateDoesNotReinterpretResolvedValues/resolved_values_cannot_access_other_va
=== RUN   TestEvaluateDoesNotReinterpretResolvedValues/template-authored_placeholders_inside_
--- PASS: TestEvaluateDoesNotReinterpretResolvedValues (0.00s)
    --- PASS: TestEvaluateDoesNotReinterpretResolvedValues/helper_syntax_in_resolved_values_s
    --- PASS: TestEvaluateDoesNotReinterpretResolvedValues/resolved_values_cannot_access_othe
    --- PASS: TestEvaluateDoesNotReinterpretResolvedValues/template-authored_placeholders_ins
PASS
ok      github.com/projectdiscovery/nuclei/v3/pkg/protocols/common/expressions  0.084s
```

Checklist

- [x] Pull request is created against the dev branch
- [x] All checks passed (lint, unit/integration/regression tests etc.) with my changes
- [x] 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 detection now runs before data replacement, preventing helper syntax and resolved values from being re-interpreted while preserving correct placeholder resolution.
- **Tests**
  - Added unit tests verifying resolved values remain literal and isolated from other variables.
  - Added an integration test validating literal reuse of response data in HTTP scenarios.

`fix(expressions): only eval template-authored expressions` ··· ✓ `75836b0`

dwisiswant0 requested a review from **Ice3man543** last month

auto-assign bot requested a review from **dogancanbakir** last month

neo-by-projectdiscovery-dev bot commented on Mar 14 • edited

Neo - PR Security Review

No security issues found

Highlights

- Commit `a334ad8` builds on the expression injection fix from `75836b0`
- Core security mechanism unchanged: expressions collected from original template BEFORE placeholder substitution
- Response-derived data continues to be treated as literal text, preventing DSL reinterpretation
- Test coverage validates the injection prevention works as intended

▶ **Hardening Notes**

Comment `@pdneo help` for available commands. · Open in Neo

coderabbitai bot commented on Mar 14 • edited Contributor

No actionable comments were generated in the recent review. 🎉

▶ ℹ️ Recent review info

Walkthrough

Moved expression discovery in Evaluate to run before replacement so expressions are collected from pre-replacement data, preventing recursive evaluation of DSL syntax in resolved values. Added unit and integration tests validating literal reuse and isolation of resolved values.

Changes

Cohort / File(s)	Summary
Expression Detection Timing `pkg/protocols/common/expressions/expressions.go`	Invoke FindExpressions earlier in Evaluat expressions are discovered from pre-repl data rather than after replacements; remo later duplicate discovery.
Expression Resolution Tests `pkg/protocols/common/expressions/expressions_test.go`	Added `TestEvaluateDoesNotReinterpretResol` with cases ensuring resolved response-d values are not reinterpreted as DSL/helpe that placeholders behave as expected.
Integration Test: HTTP `cmd/integration-test/http.go`	Added an integration test case and runne (`httpResponseDataLiteralReuse`) that s local server returning template-like conte

Cohort / File(s)	Summary
	asserts template execution does not re-ev response-derived DSL.
Module manifest `go.mod`	Module manifest updated (lines changed in diff).

Estimated code review effort

🎯 3 (Moderate) | ⏱️ ~22 minutes

Poem

> 🐰 I found expressions, neat and early,
> No more re-parsing things so curly.
> Response text stays as given, true,
> Helpers only run on what we knew.
> Hop, test, and ship — a calm review.

▶ 📟 Pre-merge checks | ✅ 4 | ❌ 1

▶ ✨ Finishing Touches

▶ 📝 Coding Plan

Comment `@coderabbitai help` to get the list of available commands and usage tips.

> Tip
>
> ▶ You can disable the changed files summary in the walkthrough.

`test: add resp data literal reuse test` ··· ✓ `a334ad8`

Ice3man543 approved these changes on Mar 16

View reviewed changes

Ice3man543 left a comment Member

lgtm!

dwisiswant0 merged commit `d221732` into `dev` on Mar 16 View details

20 checks passed

dwisiswant0 deleted the `dwisiswant0/fix/expressions/only-eval-template-authored-expressions` branch last month

Sign up for free **to join this conversation on GitHub**. Already have an account? Sign in to comment

Reviewers

Ice3man543 ✓

dogancanbakir ●

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

Disable recursive variable/DSL resolution from target response data

2 participants