


[New issue](#)[Jump to bottom](#)

compiler: allow for mocking built-in functions via "with"

#4540

[Merged](#)

srenatus merged 53 commits into [open-policy-agent:main](#) from [srenatus:sr/ast/function-mocking](#)  on Apr 24

[Conversation 39](#)[Commits 53](#)[Checks 30](#)[Files changed 17](#)

srenatus commented on Apr 4 • edited ▼

[Contributor](#)

With this change, we're able to mock built-in functions in tests 🎉

```
package test

import future.keywords.in

allow {
    payload.owner == input.name
}

claims[k] = v {
    resp := http.send({"method": "GET", "body": input}).body
    some k, v in resp
}

payload = p {
    some p
    [true, _, p] = io.jwt.decode_verify(claims.jwt, {"cert": data.verification.cert, "iss": "issu
}

pass_resp = {"body": {"jwt": "myjot"}}

deny_resp = {"body": ""}

mock_http_send(req) = pass_resp {
    req.body.name == "alice"
} else = deny_resp {
    true
}
```

```
mock_decode_verify("myjot", _) = [true, {}, {"owner": "alice"}]


test_allow {
  allow with data.verification.cert as "cert"
    with input.name as "alice"
    with http.send as mock_http_send
    with io.jwt.decode_verify as mock_decode_verify
}
```

⚠ Mocking user-defined (rego) functions is the next slice of work 🚧

Fixes part of [#4449](#)

TODOs

- ☒ validate with target
- ☒ suppress rewriting
- ☒ adapt the var visitor to skip the ref call head for `SkipRefCallHead` in `*With targets`
- ☒ evaluate
 - ☒ deal with var in topdown or fix the var -> ref rewriting;
 - ☒ deal with stack overflow when mock func calls orig func
- ☒ partial eval
- ☒ document
- ☒ wasm
- ☒ forbid mocking relations (for now, "walk" and potentially custom relations)
- ☒ replacing builtins with builtins and other values
 - ☒ topdown
 - ☒ planner/wasm
 - ☒ update docs

🔗  **srenatus** force-pushed the `sr/ast/function-mocking` branch 2 times, most recently from **2627da9** to **83dec92** 8 months ago

[Compare](#)

🗨  **srenatus** mentioned this pull request on Apr 5

mocking function responses via "with" statements #4449

🔒 Closed

🔗  **srenatus** force-pushed the `sr/ast/function-mocking` branch 3 times, most recently from **70ab9f9** to

c16e679 8 months ago

Compare

srenatus commented on Apr 5

Contributor

Author

I think we can do this simpler than c16e679 . Mocked funcs shouldn't be affected by mocking. 🤔



srenatus force-pushed the sr/ast/function-mocking branch from 1d9ee44 to 804e27b 8 months ago

Compare



srenatus marked this pull request as ready for review 8 months ago



srenatus commented on Apr 7

[View changes](#)

topdown/topdown_partial_test.go

Outdated

```
872 | +                                r[1]
873 | +                                r[2]
874 | +                                `},
875 | +                                wantQueries: []string{"not data.partial.test.q with input.x"}
```



srenatus on Apr 7

Contributor

Author

So in the not + partial code path, the "disable inlining of the replacement func" trick won't work.

Since it's inlined, it should not be here. But I think for consistency, I'll try *not inlining* it here, either.



srenatus force-pushed the sr/ast/function-mocking branch from a79c1a1 to a4a1e94 8 months ago

Compare



johanfylling reviewed on Apr 8

[View changes](#)



johanfylling left a comment

Contributor

Partial review. Looking good 👍 😊

ast/compile.go **Outdated**

⌵ Show resolved

ast/compile.go **Outdated**

⌵ Show resolved

ast/compile_test.go

⌵ Show resolved

ast/compile_test.go **Outdated**

⌵ Show resolved

ast/compile_test.go **Outdated**

⌵ Show resolved

👁 **tsandall** reviewed on Apr 10

[View changes](#)



tsandall left a comment

Member

This looks very close! Left a couple comments.

ast/compile.go **Outdated**

⌵ Show resolved

topdown/eval.go **Outdated**

⌵ Show resolved

topdown/eval.go **Outdated**

⌵ Show resolved

test/cases/testdata/withkeyword/test-with-builtin-mock.yaml **Outdated**

⌵ Show resolved

ast/compile.go **Outdated**

⌵ Show resolved

srenatus commented on Apr 10

Contributor

Author

















@tsandall thanks for the review! 😊

🔧 **srenatus** marked this pull request as draft 8 months ago

🔀 **srenatus** force-pushed the `sr/ast/function-mocking` branch 7 times, most recently from **37111df** to **f9647a8** 8 months ago

[Compare](#)

 **srenatus** added 8 commits 8 months ago

-   ast: function mocking, initial steps ... f09d613
-   ast/compile: make built-in functions valid 'with' targets ... d175f4d
-   ast/compile: validate 'with' target for built-in functions ... 2cde742
-   ast/compile: (don't) rewrite 'with' replacements for built-ins ... aef0305
-   ast/compile: properly handle simple built-ins (var names, not refs) ... 2ec0315
-   topdown/eval: allow built-in function mocks ... 3b46604
-   topdown/eval: remove traces comment ... 2d9640c
-   topdown/eval: add first yaml tests, fix Var->Ref issue ... 4956dfb

40 hidden items

[Load more...](#)

 **tsandall** reviewed on Apr 16

[View changes](#)



tsandall left a comment

Member

This looks great--it feels very close. Left a few comments on the implementation and test cases. It would be nice to get the docs updates written up and included in this PR--it'll be important to explain the scoping in that section in the language guide.

ast/compile.go

 Show resolved

ast/compile.go

Outdated

 Show resolved

ast/compile.go

Outdated

 Show resolved

test/cases/testdata/withkeyword/test-with-builtin-mock.yaml

Outdated

 Show resolved

test/cases/testdata/withkeyword/test-with-builtin-mock.yaml

Outdated

 Show resolved

test/cases/testdata/withkeyword/test-with-builtin-mock.yaml

Outdated

Show resolved

test/cases/testdata/withkeyword/test-with-builtin-mock.yaml

Outdated

Show resolved

test/cases/testdata/withkeyword/test-with-builtin-mock.yaml

```
358 + package test
359 +
360 + p = x {
361 + x = time.now_ns() with time.now_ns as opa.runtime
```



tsandall on Apr 16

Member

could imagine using a pure built-in as the source here to avoid test flakes in the future.



srenatus on Apr 19

Contributor

Author

Yeah... but we don't have that many arity-0 built-ins. It's only `time.now_ns()` and `opa.runtime()`; since mocking `rego.metadata.*()` is forbidden. We could add a test-only-builtin, but that could be a burden for any other users of the test data corpus.

But I'll fix those arity-n ($n > 0$) cases you've commented on below 👍

srenatus added 4 commits 7 months ago

- ast/compile_test: add test with custom built-ins ... e3faea7
- ast/compile: move mocking arity check into type checker ... ❌ b7e8370
- testcases/withkeyword: make nested scope handling case stricter ... ✅ dcefc3b
- testcases/withkeyword: replace impure built-in examples ... ✅ 4e7b14d

srenatus force-pushed the sr/ast/function-mocking branch 6 times, most recently from 0963841 to f038715 7 months ago

Compare




- docs/policy-language: document new 'with' possibilities ... ✅ 8d24b59

srenatus force-pushed the sr/ast/function-mocking branch from f038715 to 8d24b59 7 months ago

Compare

srenatus added 3 commits 7 months ago



-  ast/check: check 'with' of expr also if it is a call ... ✓ c22c7f9
-  docs/policy-language: q -> rule_using_concat ... ✗ 9e82d3a
-  format/testdata: add function replacement formatter case ... ✗ cc586ab

tsandall approved these changes on Apr 23


[View changes](#)

tsandall commented on Apr 23

Member

Can't wait for this to get rolled out!

-  Merge branch 'main' into sr/ast/function-mocking ✓ 76c67e8

 srenatus merged commit **8f49869** into [open-policy-agent:main](#) on Apr 24
31 checks passed

[View details](#)

  srenatus deleted the sr/ast/function-mocking branch 7 months ago

  srenatus mentioned this pull request on Apr 26

ast+topdown+planner: non-built-in function mocking #4616

 Merged

 rokkiter pushed a commit to rokkiter/opa that referenced this pull request on Apr 26

 ast+topdown+planner: allow for mocking built-in functions via "with" (o... e372ab3


  crqra mentioned this pull request on Apr 27

Support testing policies that use built-in functions reposaur/reposaur#2

 Closed

 srenatus added a commit to srenatus/opa that referenced this pull request on Apr 28

 ast+topdown+planner: replacement of non-built-in functions via 'with' ... 1dc2f27

 **srenatus** added a commit that referenced this pull request on Apr 28

 **ast+topdown+planner:** replacement of non-built-in functions via 'with' (... ..) ✓ 7e50293

 **damienjburks** added a commit to damienjburks/opa that referenced this pull request on May 17

 **finalizing** changes for formatting with sprintf ... 1c1c289

 **damienjburks** added a commit to damienjburks/opa that referenced this pull request on May 17

 **#** This is a combination of 38 commits. ... d208414

  **GoVulnBot** mentioned this pull request on Sep 8

x/vulndb: potential Go vuln in github.com/open-policy-agent/opa: CVE-2022-36085
golang/vulndb#978

✓ Closed

  **github-actions** bot mentioned this pull request on Sep 14

github.com/open-policy-agent/opa: CVE-2022-36085 aquasecurity/trivy#2880

✓ Closed

Reviewers

 **johanfylling**



 **tsandall**



Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

3 participants

