

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 {
          [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
```

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

srenatus commented on Apr 5

Contributor

Author

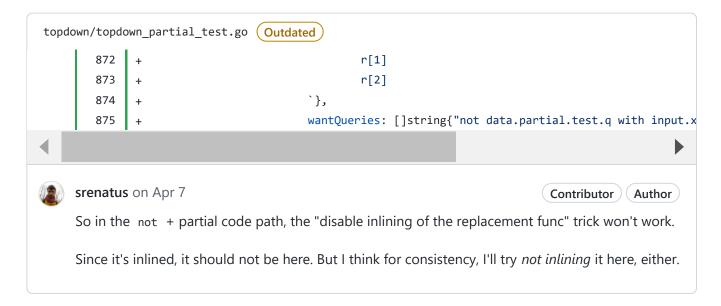
I think we can do this simpler than c16e679. Mocked funcs shouldn't be affected by mocking.

note: 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



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

Compare

johanfylling reviewed on Apr 8

View changes

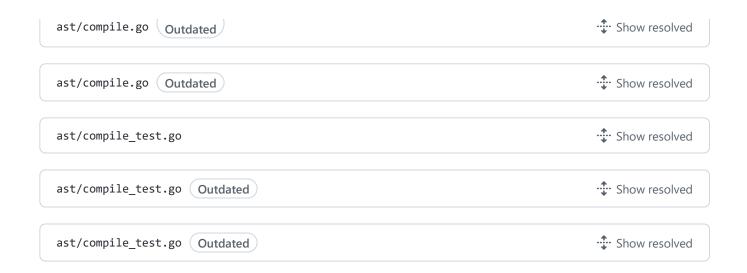


Contributor

Partial review. Looking good 👍 🙂

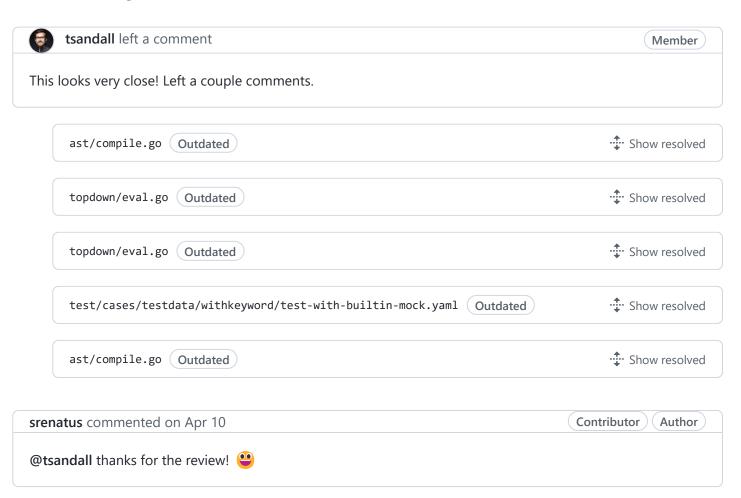






tsandall reviewed on Apr 10

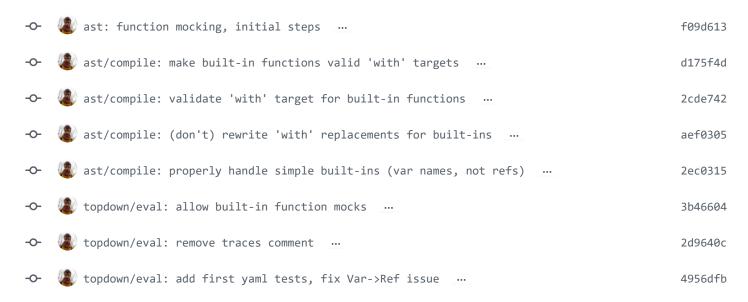
View changes



- 🕻 🏖 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



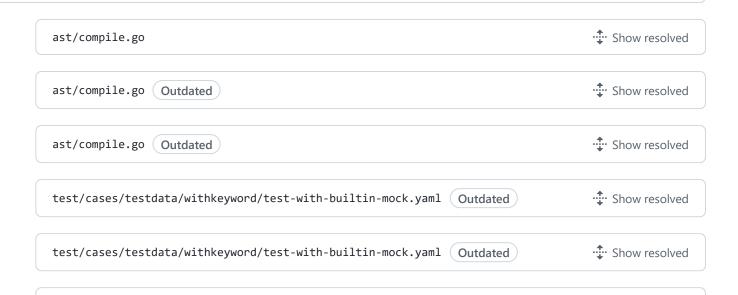
40 hidden items
Load more...

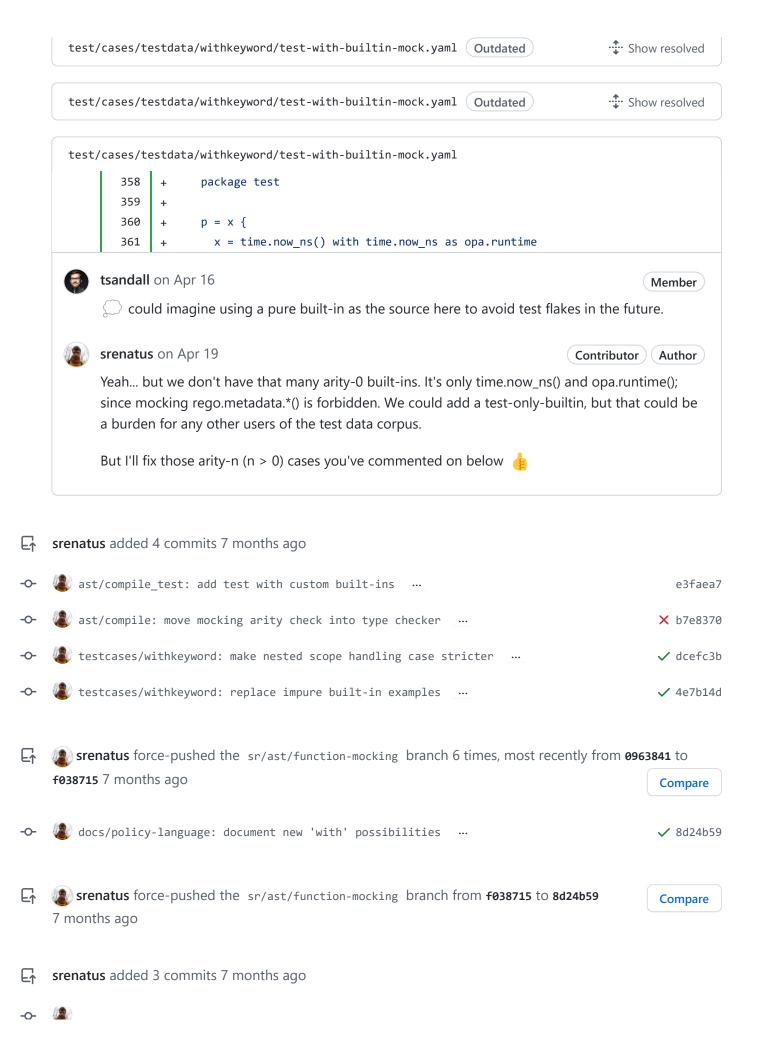
tsandall reviewed on Apr 16

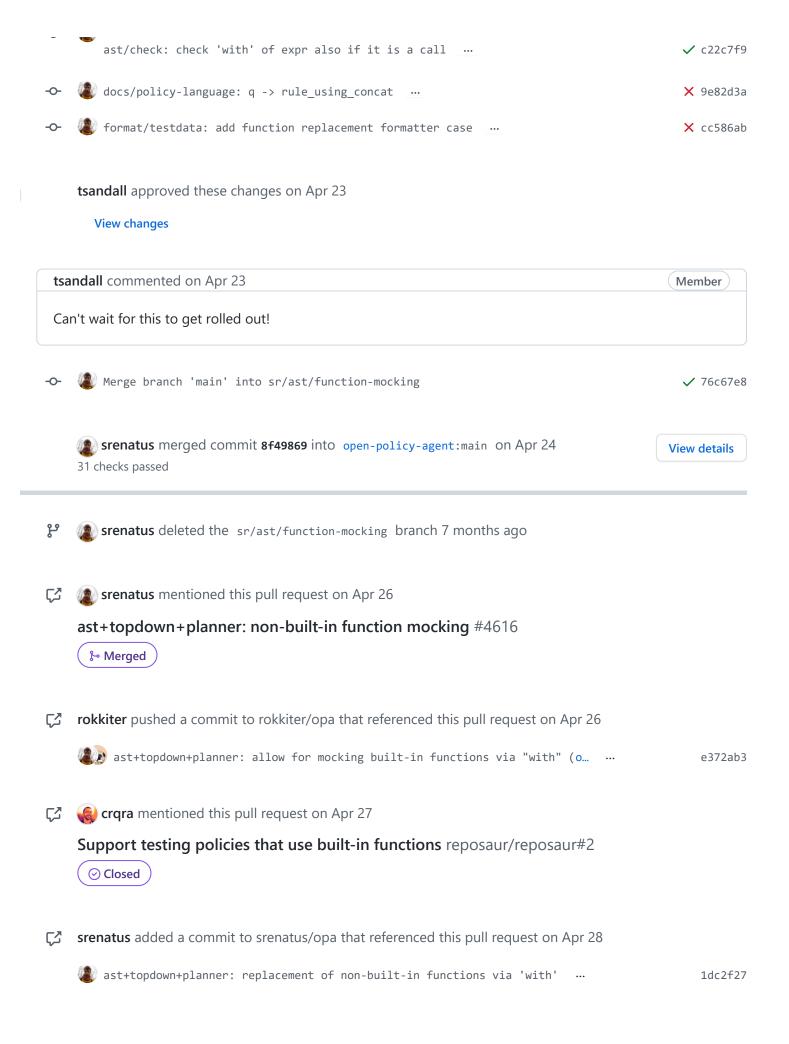
View changes



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.







ÇŽ	srenatus added a commit that referenced this pull request on Apr 28	
	ast+topdown+planner: replacement of non-built-in functions via 'with' (🗸 70	e50293
ÇŽ	damienjburks added a commit to damienjburks/opa that referenced this pull request on May 17	
	finalizing changes for formatting with sprintf	c1c289
ÇŽ	damienjburks added a commit to damienjburks/opa that referenced this pull request on May 17	
	# This is a combination of 38 commits di	208414
ÇŽ	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	
ÇŽ	github-actions bot mentioned this pull request on Sep 14	
	github.com/open-policy-agent/opa: CVE-2022-36085 aquasecurity/trivy#2880	
Revie		
	ohanfylling sandall	√
Assign	inees	
No on	ne assigned	
Labels		
None	e yet	
Project None		
	···	
Milest		
ino mi	nilestone	

Development

Successfully merging this pull request may close these issues.

None yet

3 participants





