



# Detect complex code patterns using semantic grep

Bence Nagy | [bence@r2c.dev](mailto:bence@r2c.dev)

 [@r2cdev](https://twitter.com/r2cdev)

# tl;dw – This Talk

- Secure code is hard
- Static analysis tools are too noisy / too slow
- grep isn't expressive enough
- Need something, fast, code-aware, flexible, powerful... **open source!**



[Semgrep](#): Fast and syntax-aware semantic code pattern search for many languages: like grep but for code

## Use Semgrep to:

- Search: Find security bugs
- Guard: Enforce best practices
- Monitor: Get notifications about new matches
- Migrate: Refactor code easily

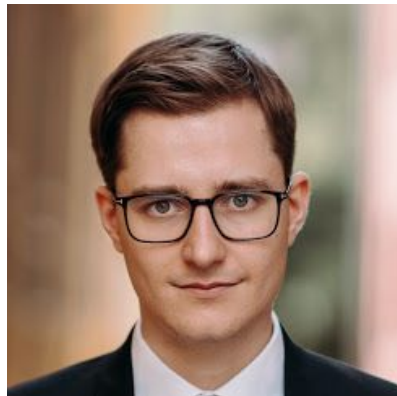
# \$ whois

@underyx (Bence Nagy) engineer @ r2c

previously at:

Astroscreen (information warfare)

Kiwi.com (travel)








# \$ getent group r2c


We're an SF based code analysis startup.

Mission: profoundly improve code security & reliability



# Outline

1. A 60 second history 
2. Trees.  *(well... syntax trees)*
3. Learning Semgrep! 
4. Integration into CI/CD 
5. Semgrep Rules Registry 

 **returntocorp / semgrep**

Watch 35

Unstar 2k

Fork 77

<> Code

Issues 168

Pull requests 6

Actions

Security

...

develop







Go to file

Add file

Code

About



	emjin Update pattern-from-code ...	18 seconds ago	1,327
	.circleci	Use new python rule to detect wro...	17 days ago
	.github	add basic metrics for semgrep-co...	6 days ago
	.vscode	add pre-commit	8 months ago
	docs	release changes	2 days ago
	ocaml-tree-sit...	use latest ocaml-tree-sitter and nf...	7 days ago

Lightweight static analysis for many languages. Find bug variants with patterns that look like source code.

 [semgrep.dev](https://semgrep.dev)

static-analysis

[github.com/returntocorp/semgrep](https://github.com/returntocorp/semgrep)

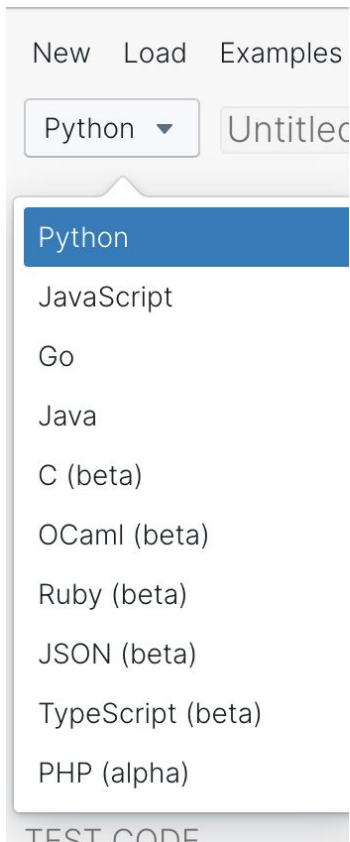
# Semgrep, Est. 2009



First version of Semgrep (sgrep/pfff) was written at Facebook circa 2009 and was used to enforce nearly 1000 rules!

The original author, Yoann Padioleau ([@aryx](#)), joined r2c last year. Yoann was the first static analysis hire at Facebook and previously PhD @ Inria, contributor to [coccinelle.lip6.fr](http://coccinelle.lip6.fr)


# Language Support



# License

Branch: **develop** ▾ **semgrep / LICENSE**

---



returntocorp/semgrep is licensed under the  
**GNU Lesser General Public License v2.1**

Primarily used for software libraries, the GNU LGPL requires that derived works be licensed under the same license, but works that only link to it do not fall under this restriction. There are two commonly used versions of the GNU LGPL.

**Permissions**

- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Private use

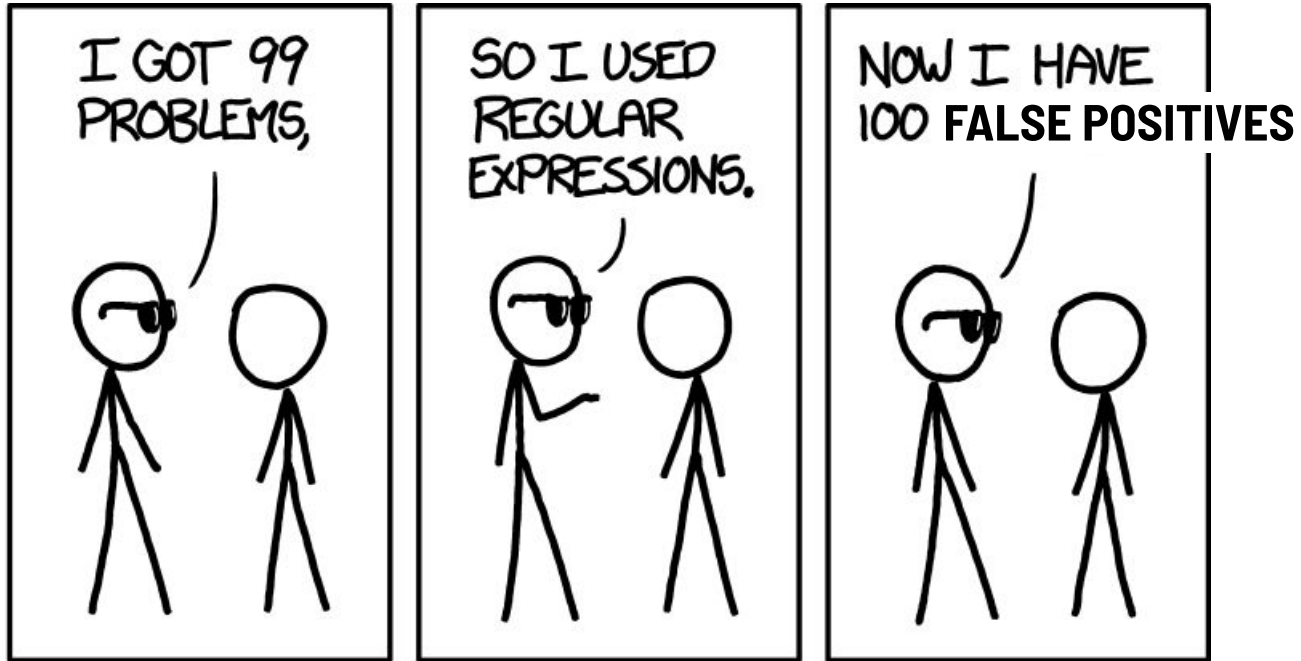
---

This is not legal advice. [Learn more about repository licenses.](#)



# `grep` and Abstract Syntax Trees (ASTs)

# xkcd 1171



# Code is not a string, it's a tree



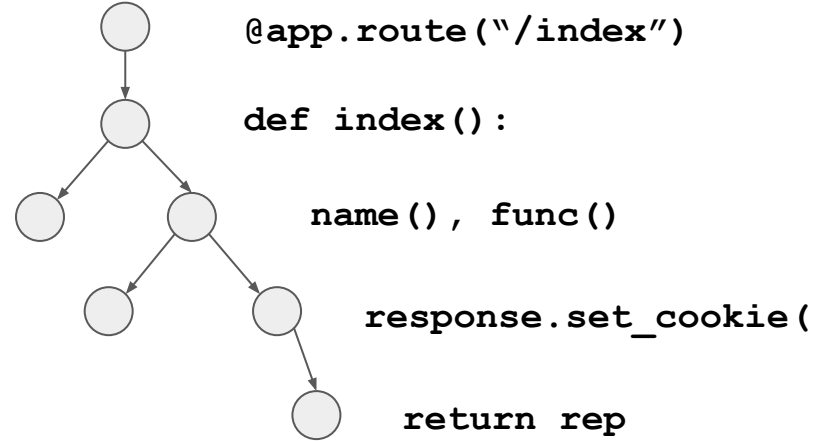
**string**

```
@app.route("/index")
def index():
    rep = response.set_cookie(name(),
    secure=False, s=func())
    return rep
```

**!=**



**tree**



# Tree Matching

- Many tree matching tools: Gosec, Golint, Bandit, Dlint, ESLint, Flake8, Pylint, RuboCop, TSLint, and more!
- Have to become an **expert in every AST syntax** for every language your team uses
- Need **programming language expertise** to cover all idioms: languages have “more than one way to do it”
- **Commercial SAST tools?**
  - Complicated
  - Slow (not CI friendly)
  - Expensive

Find calls to old  
crypto in 94 LOC



```
94 lines (81 loc) 3.15 KB
1 // (c) Copyright 2016 Hewlett Packard Enterprise Development LP
2 //
3 // Licensed under the Apache License, Version 2.0 (the "License");
4 // you may not use this file except in compliance with the License.
5 // You may obtain a copy of the license at
6 //
7 // http://www.apache.org/licenses/LICENSE-2.0
8 //
9 // Unless required by applicable law or agreed to in writing, software
10 // distributed under the License is distributed on an "AS IS" BASIS,
11 // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 // See the License for the specific language governing permissions and
13 // limitations under the License.
14
15 package rules
16
17 import (
18     "go/ast"
19     "strings"
20
21     "github.com/securego/gosec/v2"
22 )
23
24 type blocklistedImport struct {
25     gosec.Metadata
26     blocklisted map[string]string
27 }
28
29 func unquote(original string) string {
30     copy := strings.TrimFunc(original,
31         func(r rune) bool {
32             return strings.ContainsAny(string(r), "\"'")
33         })
34 }
35
36 func (r blocklistedImport) ID() string {
37     return r.Metadata.ID
38 }
39
40 func (r blocklistedImport) MatchIn ast.Node, c gosec.Context (gosec.Issue, error) {
41     if node, ok := r.Last.ImportSpec(); ok {
42         if description, ok := r.blocklisted[unquote(node.Path.Value)]; ok {
43             return gosec.NewIssue(c, node, r.ID(), description, r.Severity, r.Confidence), nil
44         }
45     }
46     return nil, nil
47 }
48
49 // NewBlocklistedImports reports when a blocklisted import is being used.
50 // Typically when a deprecated technology is being used.
51 func NewBlocklistedImports(conf gosec.Config, blocklist map[string]string) (gosec.Rule, []ast.Node) {
52     return blocklistedImport{
53         Metadata: gosec.Metadata{
54             ID: "blocklist",
55             Severity: gosec.High,
56             Confidence: gosec.High,
57         },
58         blocklisted: blocklist,
59     }, []ast.Node{ast.ImportSpec}
60 }
61
62 // NewBlocklistedImportDES fails if DES is imported
63 func NewBlocklistedImportDES(id string, conf gosec.Config) (gosec.Rule, []ast.Node) {
64     return NewBlocklistedImports(id, conf, map[string]string{
65         "crypto/des": "Blocklisted import crypto/des: weak cryptographic primitive",
66     })
67 }
68
69 // NewBlocklistedImportDES fails if DES is imported
70 func NewBlocklistedImportDES(id string, conf gosec.Config) (gosec.Rule, []ast.Node) {
71     return NewBlocklistedImports(id, conf, map[string]string{
72         "crypto/des": "Blocklisted import crypto/des: weak cryptographic primitive",
73     })
74 }
75
76 // NewBlocklistedImportMD5 fails if MD5 is imported
77 func NewBlocklistedImportMD5(id string, conf gosec.Config) (gosec.Rule, []ast.Node) {
78     return NewBlocklistedImports(id, conf, map[string]string{
79         "crypto/md5": "Blocklisted import crypto/md5: weak cryptographic primitive",
80     })
81 }
82
83 // NewBlocklistedImportSHA1 fails if SHA1 is imported
84 func NewBlocklistedImportSHA1(id string, conf gosec.Config) (gosec.Rule, []ast.Node) {
85     return NewBlocklistedImports(id, conf, map[string]string{
86         "crypto/sha1": "Blocklisted import crypto/sha1: weak cryptographic primitive",
87     })
88 }
89
90 // NewBlocklistedImportSHA256 fails if SHA256 is imported
91 func NewBlocklistedImportSHA256(id string, conf gosec.Config) (gosec.Rule, []ast.Node) {
92     return NewBlocklistedImports(id, conf, map[string]string{
93         "crypto/sha256": "Blocklisted import crypto/sha256: weak cryptographic primitive",
94     })
95 }
```

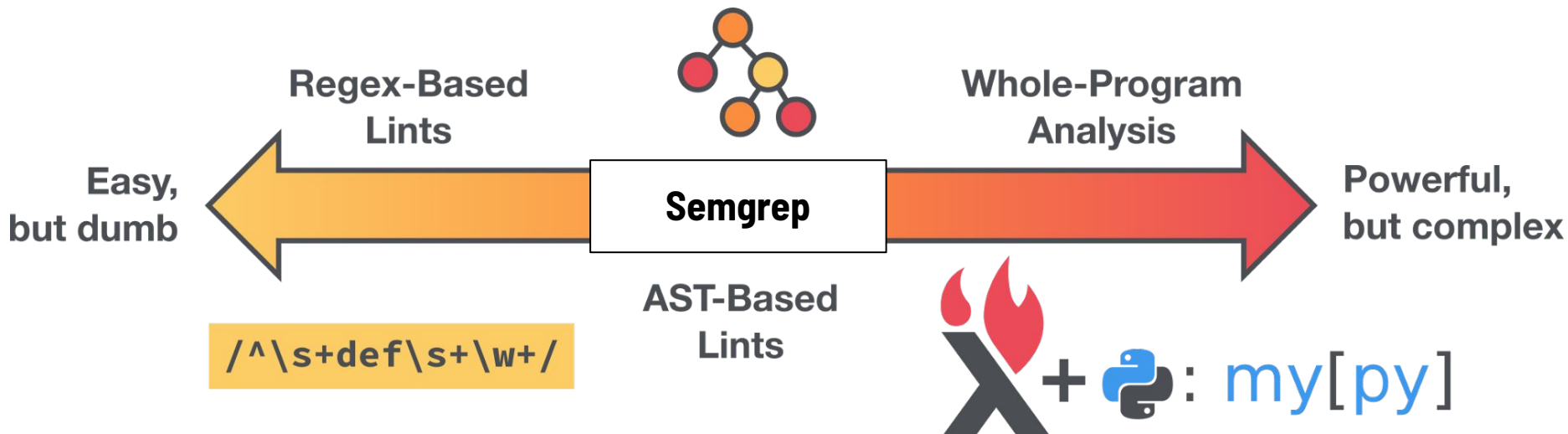
<https://github.com/securego/gosec/blob/master/rules/blocklist.go#L39-L46>

# Static Analysis at Scale: An Instagram Story



Benjamin Woodruff [Follow](#)

Aug 15, 2019 · 13 min read



<https://instagram-engineering.com/static-analysis-at-scale-an-instagram-story-8f498ab71a0c>

Semgrep:

reason about **analysis** like you reason about **code**

write **eval(...)** to match **eval(request)**

<https://r2c.dev/blog/2020/why-i-moved-to-semgrep-for-all-my-code-analysis/>

# Demos

## 1. Overview

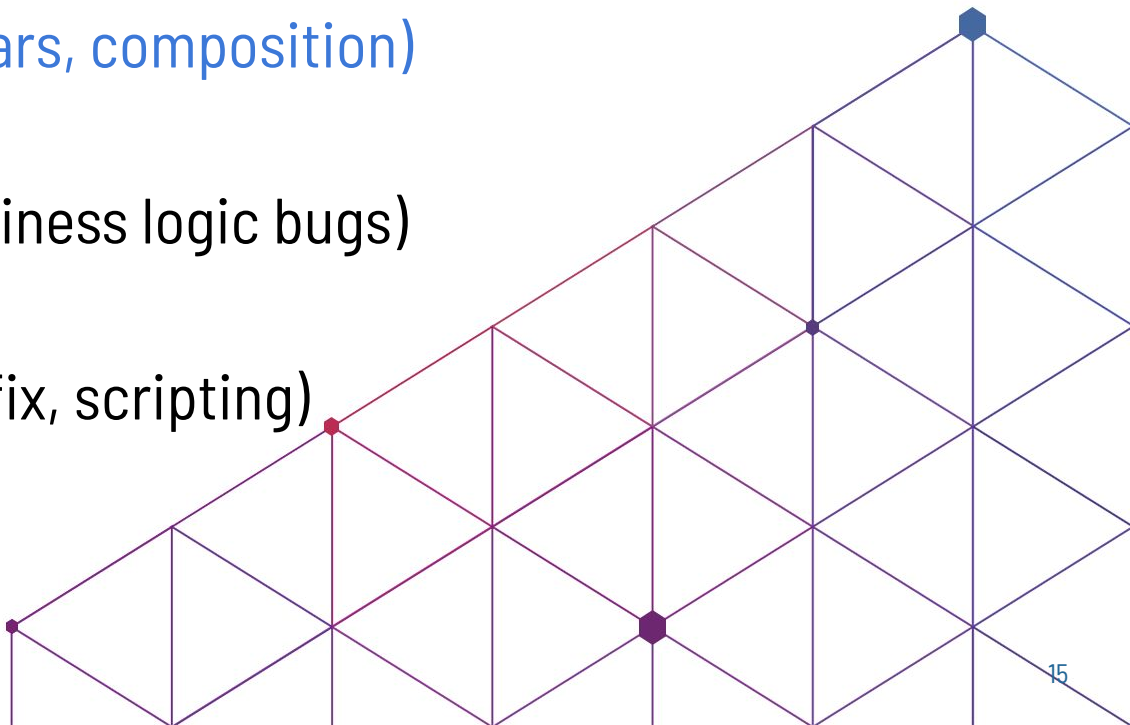
(The `...` operator, metavariables, composition)

## 2. Semgrep In Practice

(for antipatterns and business logic bugs)

## 3. Advanced Features

(Extracting Routes, autofix, scripting)



## Finding Banned or Deprecated Functions: RC4 (... operator)

```
c, err := rc4.NewCipher(key)
d, err := rc4.NewCipher(otherKey)
e, err := rc4.NewCipher (    key    )

// rc4.NewCipher(key)
fmt.Println("rc4.NewCipher(key)")
```

⇒ <https://semgrep.dev/s/10Bx>

Full Solution: <https://semgrep.live/X5g4> | [docs](#)



# Finding Uses of `unsafe`

(Metavariables)

```
unsafe.Pointer(intPtr)  
unsafe.Sizeof(intArray[0])
```

⇒ <https://semgrep.dev/s/7gZe/>

Full Solution: <https://semgrep.dev/s/ErXL/>

## Finding Insecure SSL Configurations (Field/Param matching)

```
&tls.Config{  
    KeyLogWriter: w,  
    MinVersion:  tls.VersionSSL30,  
    Rand:  randSource{}  
}
```

⇒ <https://semgrep.live/Pewp>

Full Solution: <https://semgrep.live/4b9x>

# Finding Insecure SSL Configurations (Composing patterns)

```
&tls.Config{  
    KeyLogWriter: w,  
    MinVersion:  tls.VersionSSL30,  
    Rand:  randSource{},  
    InsecureSkipVerify: true,  
}
```

⇒ <https://semgrep.live/s/DbYd>



# Configuration Files

This document describes `semgrep` configuration files and provides rule examples. Configuration files are specified with the `--config` (or `-f`) flag. A single [YAML](#) file or a directory of files ending in `.yaml` or `.yml` may be specified. Each configuration file must match the [schema](#).

For more information on the `--config` flag see [other configuration options](#).

Contents:

- [Simple Example](#)
- [Other Configuration Options](#)
- [Schema](#)
- [Operators](#)
  - [pattern](#)
  - [patterns](#)
  - [pattern-either](#)
  - [pattern-regex](#)
  - [pattern-not](#)
  - [pattern-inside](#)
  - [pattern-not-inside](#)
  - [pattern-where-python](#)
- [Metavariable Matching](#)
  - [Metavariables in Logical ANDs](#)
  - [Metavariables in Logical ORs](#)
  - [Metavariables in Complex Logic](#)

<https://github.com/returntocorp/semgrep/blob/develop/docs/configuration-files.md>

# Using Hardcoded Secret for JWT

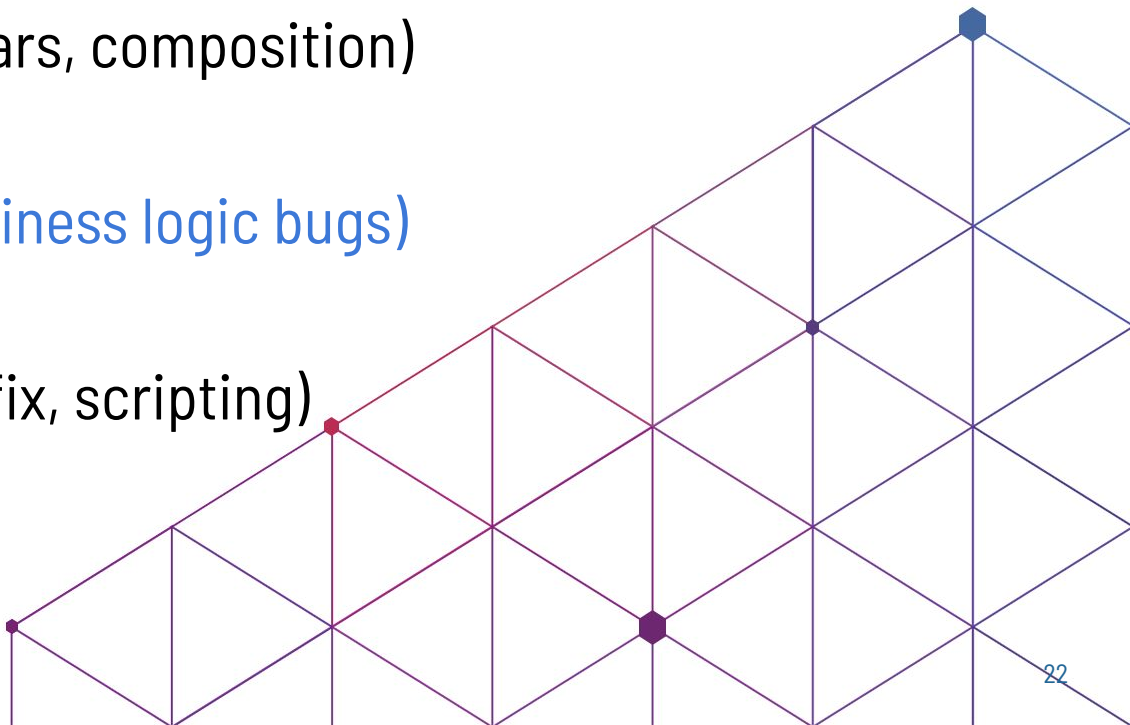
```
var jwtKey = []byte("my_secret_key")  
  
token := jwt.NewWithClaims(jwt.SigningMethodHS256, claims)  
  
tokenString, err := token.SignedString(jwtKey)
```

⇒ <https://semgrep.live/0oZB>

Full Solution: <https://semgrep.live/WAbL>

# Demos

1. Overview  
(The `...` operator, metavariables, composition)
2. **Semgrep In Practice**  
(for antipatterns and business logic bugs)
3. Advanced Features  
(Extracting Routes, autofix, scripting)



## Hidden Goroutines ([blog post](#))

(Antipatterns)

```
//  
// Antipattern  
//  
func Cleanup() {  
    go func() {  
        // ...  
    }()  
}
```

Cleanup()

```
//  
// Better  
//  
func Cleanup() {  
    // ...  
}
```

go Cleanup()

⇒ <https://semgrep.live/9A4z>

## Order of API Calls Must be Enforced (Business Logic)

```
/*  
 * In this financial trading application, every transaction  
 * MUST be verified before it is made  
 *  
 * Specifically: verify_transaction() must be called on a transaction  
 * object before that object is passed to make_transaction()  
 */
```

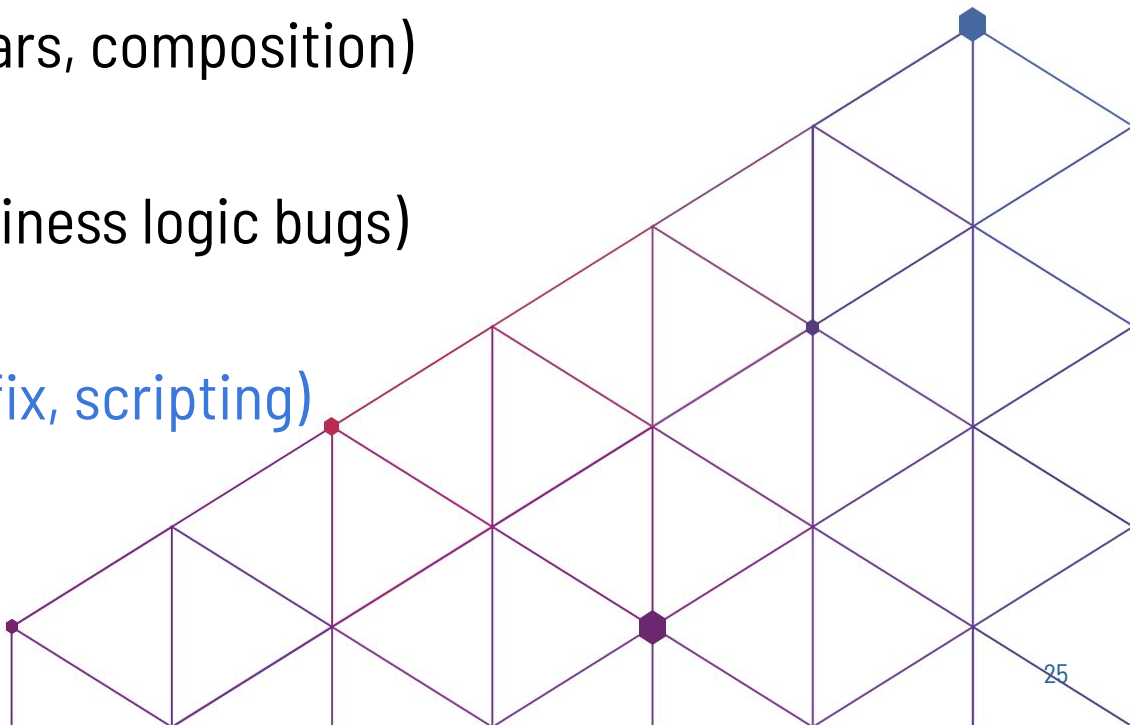
⇒ <https://semgrep.live/6JqL>

Full Solution: <https://semgrep.live/oqZ6>



# Demos

1. Overview  
(The `...` operator, metavariables, composition)
2. Semgrep In Practice  
(for antipatterns and business logic bugs)
3. **Advanced Features**  
(Extracting Routes, autofix, scripting)



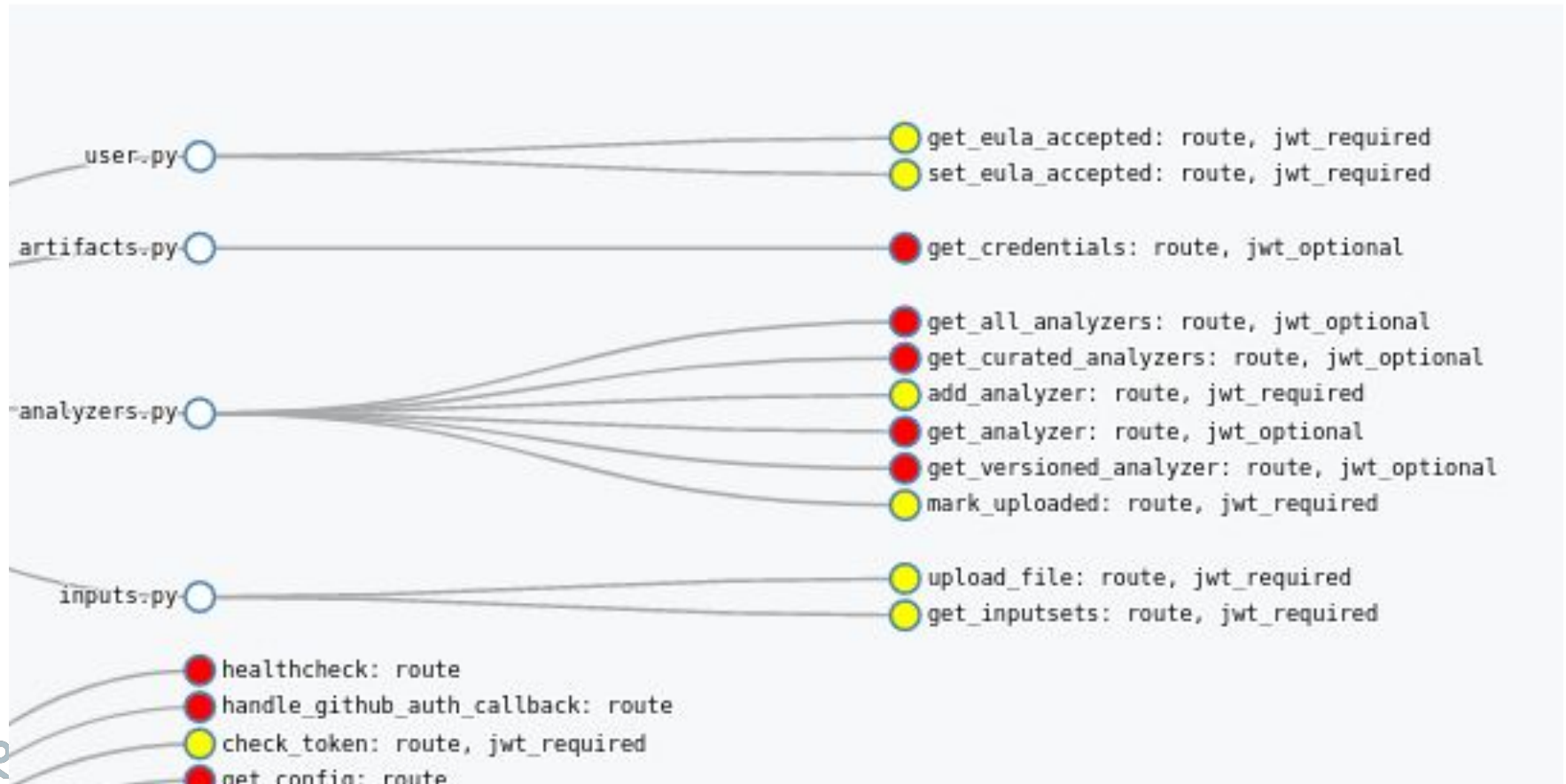
## Know When New Routes Are Added

([Gorilla Toolkit](#))

```
func (a *App) initializeRoutes() {  
    a.Router.HandleFunc("/products",  
                        a.getProducts).Methods("GET")  
}
```

<https://semgrep.dev/s/r6o1>

# Semgrep application: code inventory



## Autofix - Insecure SSL Configuration

```
&tls.Config{  
    KeyLogWriter: w,  
    MinVersion:  tls.VersionSSL30,  
    Rand:  randSource{}  
}
```

<https://semgrep.dev/s/xxyA/>

# Scripting

your code here

```
rules:
  - id: use-decimalfield-for-money
    patterns:
      - pattern-inside: |
          class $M(...):
              ...
      - pattern: $F = django.db.models.FloatField(...)
      - pattern-where-python: 'price' in vars['$F']
      - message: "Found a FloatField used for variable $F. Use
        DecimalField for currency fields to avoid float-rounding errors."
    languages: [python]
    severity: ERROR
```

**\* requires a flag:** --dangerously-allow-arbitrary-code-execution-from-rules

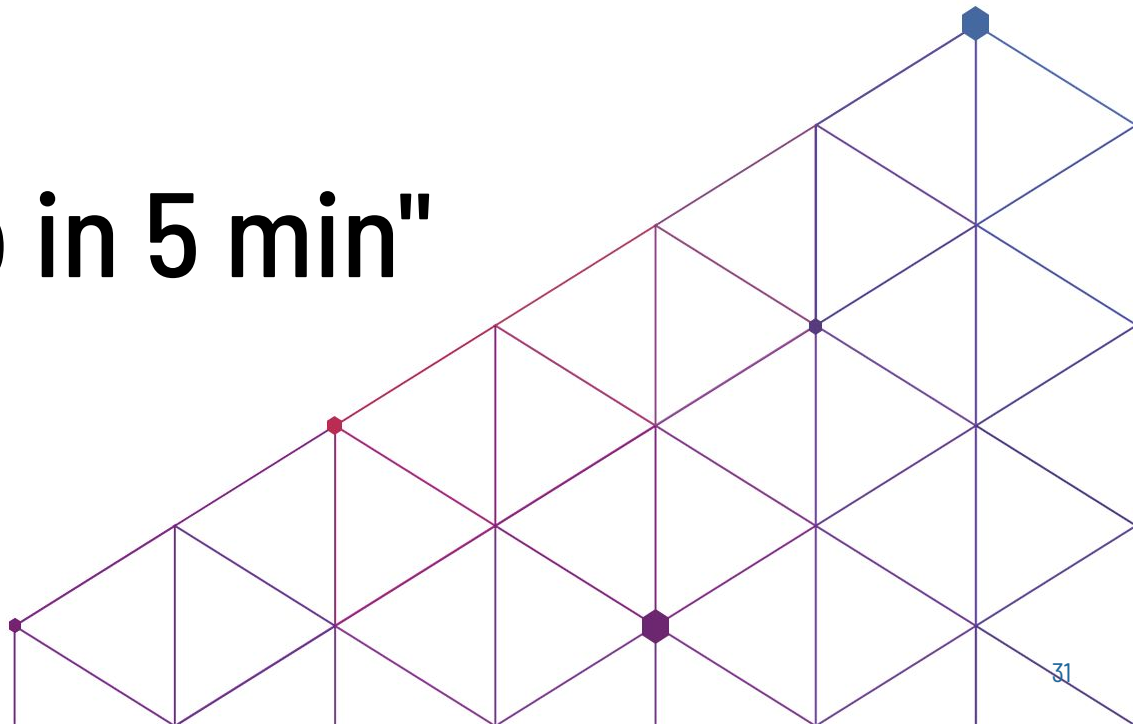
## Use of Weak RSA Key

```
// Insufficient bit size  
pvk, err := rsa.GenerateKey(rand.Reader, 1024)  
  
// Sufficiently large bit size  
pvk, err := rsa.GenerateKey(rand.Reader, 2048)
```

⇒ <https://semgrep.dev/s/zdRI>

Full Solution: <https://semgrep.dev/s/zdRI> | [docs](#)

recap, a.k.a.  
"learn semgrep in 5 min"



# #1 Code equivalence (**semantic grep**)

```
$X == $X
```

Will match

```
(a+b != a+b) # <=> !(a+b==a+b)
```

```
foo(kwd1=1, kwd2=2, ...)
```

Will match

```
foo(kwd2=2, kwd1=1, kwd3=3)
```

```
subprocess.open(...)
```

Will match

```
from subprocess import open as  
sub_open  
  
result = sub_open("ls")
```

```
import foo.bar
```

Will match

```
from foo import bar
```

- **semgrep** knows about the semantics of the language, so one pattern can match variations of equivalent code (constant propagation! <https://semgrep.live/4K5>)



## #2: '...' ellipsis operator

```
foo(...,5)
```

Will match

```
foo("...")
```

Will match

```
$V = get()  
...  
eval($V)
```

Will match

```
foo(1,2,3,4,5)  
foo(5)
```

```
foo("whatever sequence of chars")
```

```
user_data = get()  
print("do stuff")  
foobar()  
eval(user_data)
```

'...' can match sequences of:

- Arguments, parameters
- Characters
- Statements

## #3 Metavariables (part 1)

```
foo($X,2)
```

Will match

```
foo(1,2)
```

```
if $E:  
    foo()
```

Will match

```
if x > 2:  
    foo()
```

```
if $X > $Y:  
    $S
```

Will match

```
if var > 2:  
    return 1
```

```
$F(1,2)
```

Will match

```
foo(1,2)
```

- **Metavariables** start with a \$ (\$X, \$Y, \$WHATEVER), contain uppercase ASCII characters
- **Matches:**
  - Expressions (including arguments)
  - Statements
  - Names (functions, fields, etc.)

## #3 Metavariables (part 2)

```
$X == $X
```

Will match

```
if $E:  
    $S  
else:  
    $S
```

Will match

```
$V = open()  
close($V)
```

Will match

```
if (a+b == a+b) :
```

```
if x > 2:  
    foo()  
    bar()  
else:  
    foo()  
    bar()
```

```
myfile = open()  
close(myfile)
```

You can reuse the same metavariable: **semgrep** enforces **equality constraint**

# Awesome Use Cases

## Search your code

- Vulnerabilities
- Audit security hotspots
- Extract routes
- Codify domain knowledge

## Guard your code

- Secure defaults
- Banned APIs
- Best- and required-practices
- Configuration file auditing

## Upgrade your code

- Migrate from deprecated APIs
- Apply automatic fixes

# Search: Vulnerabilities



Rule name **2Zz5** rename

Single Pattern docs Multi-Pattern (YAML) docs

Saved Rules Example Rules

## Matches (1)

Run with docker image: 0.14.0 in 0.6s (show steps)

### 6: Detected path traversal with filename

flask-path-traversal ERROR

```
rules:
- id: flask-path-traversal
  message: Detected path traversal with $FILENAME
  pattern: |
    @APP.route(...)
    def $FUNC(..., $FILENAME, ...):
      ...
      open(<... $FILENAME ...>, ...)
  severity: ERROR
  languages:
  - python
```

Code Snippet In Python

Example Code

```
1 from flask import Flask, send_file, make_response
2 import os
3
4 app = Flask()
5
6 @app.route("/get_file/<filename>")
7 def get_file(filename):
8     print("getting file", filename)
9     return make_response(open(os.path.join("/tmp", filename), 'r').read())
10
11
12 @app.route("/")
13 def get_index():
14     return send_file("index.html")
```

### More Options

- Run locally
- Scan My Code
- Add to CI
- Run on r2c platform
- Add True Positive Tests
- Advanced Options

Scan lots of code

Save & Run

saving lets you share this rule, run it locally or in CI, and more

```
@$APP.route(...)  
def $FUNC(..., $FILENAME, ...):  
    ...  
    open(<... $FILENAME ...>, ...)
```

<https://semgrep.live/2Zz5/>

Start a job

Jobs

1131 dev/semgrep 0.14.0

1130 dev/semgrep 0.14.0

1129 dev/semgrep 0.14.0

1128 dev/semgrep 0.14.0

1127 dev/semgrep 0.14.0 github-1200-depends-on-flask/0.0.1 yamurl:https://semgrep.live/c/Kx5d 48 minutes ago 23.98% error rate

1126 dev/semgrep 0.14.0 github-1200-depends-on-flask/0.0.1 yamurl:https://semgrep.live/c/Kx5d 54 minutes ago 71.31% error rate

1125 dev/semgrep 0.14.0 github-1200-depends-on-flask/0.0.1 yamurl:https://semgrep.live/c/Kx5d 59 minutes ago 96.50% error rate

1124 dev/semgrep npm-github-1000-latest-2019-09-17/ yamurl:https://semgrep.live/c/colleen... 4 days ago 2.40% error rate

1123 dev/semgrep npm-latest-2019-08-26/0.0.2 yamurl:https://semgrep.live/c/colleen... 4 days ago 0.00% error rate

Analyzer

dev/semgrep 0.14.0

Change

Step 3: Select your parameters (optional)

Step 2: Select your input set

Select input set

flask

depends-on-flask(0.0.1) 39.9k

github-1200-depends-on-flask(0.0.1) 1.2k

github-flask-talisman(0.0.1) 37

top1k-flask-github(0.0.1) 1k

Run Job





Filter by repositories, commit hashes, or checks:

🔍

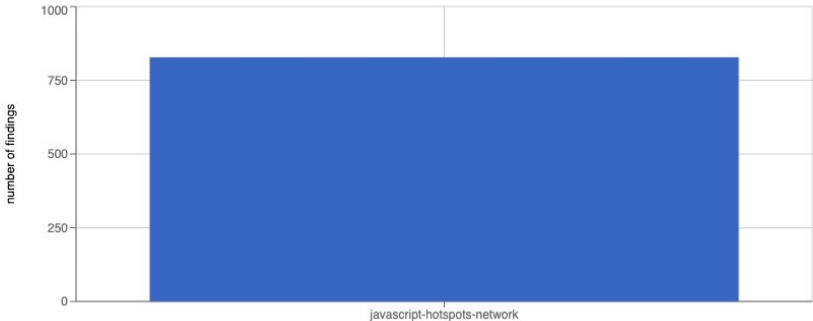
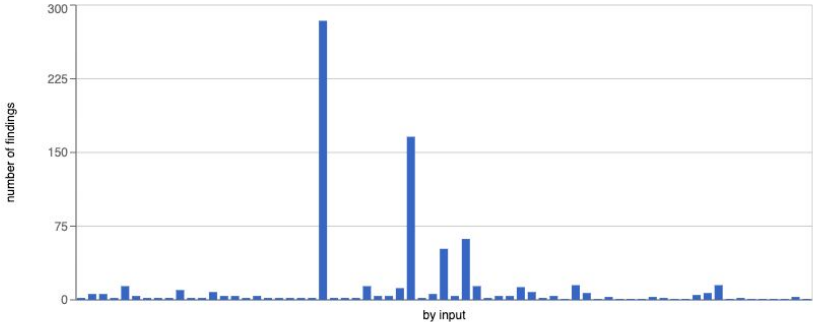
🔍

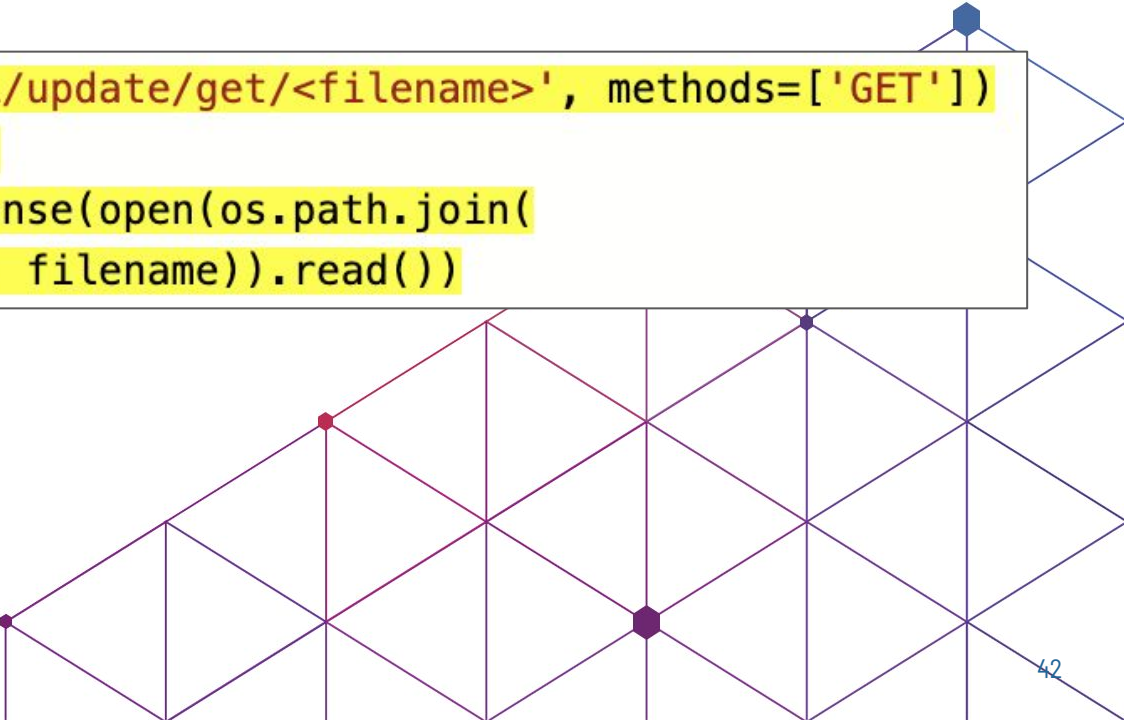
🔍

🔍

☐ Only Severity = ERROR

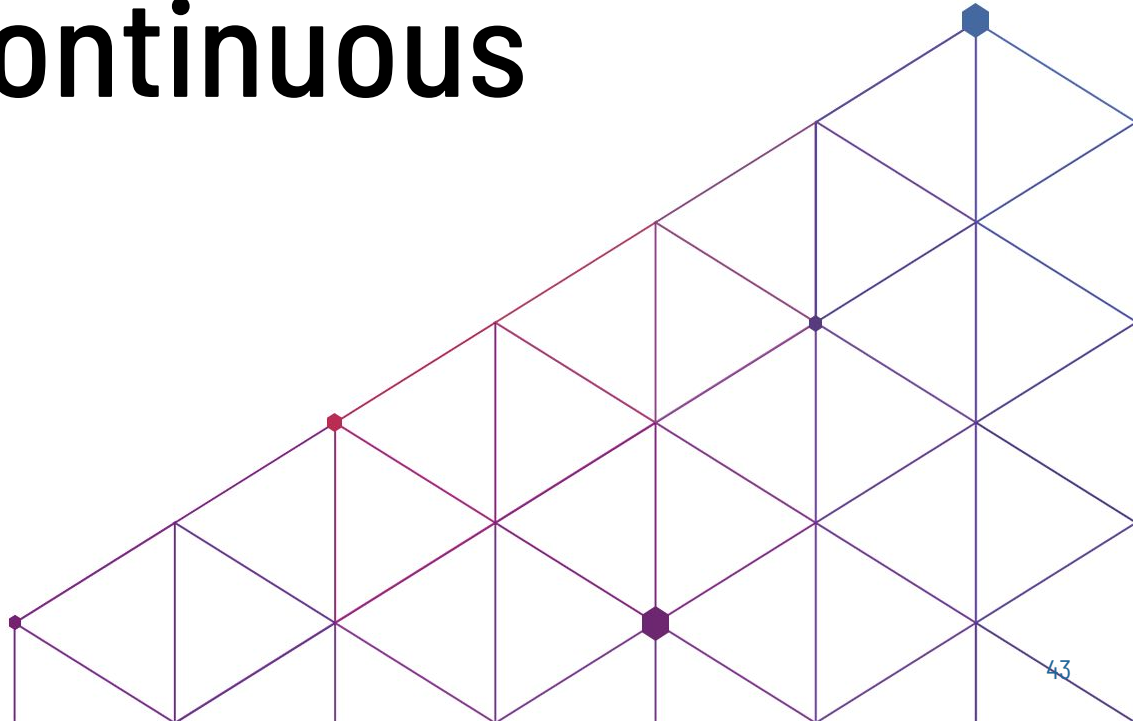
	Repository	Commit	Findings	Annotations	Action			
1	https://github.com/caolan/nodeunit	cd773a2	2		results			
2	https://github.com/trentm/node-bunyan	fe31b83	6		results			
3	https://github.com/kangax/html-minifier	51ce10f	6		results			
4	https://github.com/aheckmann/gm	e715cbd	2		results			
5	https://github.com/felixge/node-formidable	d23e560	14		results			
6	https://github.com/driverdan/node-XMLHttpRequest	97966e4	4		results			
7	https://github.com/defunctzombie/zuul	0a5644c	2		results			
8	https://github.com/intesso/connect-livereload	7c6ca1f	2		results			
9	https://github.com/chjj/blessed	eab243f	2		results			
10	https://github.com/request/request	212570b	10		results			
11	https://github.com/expressjs/express	e1b45eb	2		results			
12	https://github.com/moment/moment	13a61b2	2		results			
13	https://github.com/facebook/react	d862f0e	8		results			
14	https://github.com/gruntjs/grunt-contrib-watch	fc8458e	4		results			
15	https://github.com/karma-runner/karma	6235e68	4		results			
16	https://github.com/mishoo/UglifyJS2	70bb304	2		results			
17	https://github.com/webpack/webpack-dev-server	9d1c6d2	4		results			
18	https://github.com/jsdom/jsdom	699ed6b	2		results			
19	https://github.com/ember-cli/ember-cli	d6bbe89	2		results			
20	https://github.com/rwylblue/ember-cli-inject-live-reload	5a37c1d	2		results			
21	https://github.com/NodeRedis/node_redis	a60261d	2		results			
22	https://github.com/visionmedia/superagent	67a5eee	2		results			
23	https://github.com/hock/hock	f6e319d	284		results			
24	https://github.com/senchalabs/connect	fa8916e	2		results			
25	https://github.com/BrowserSync/browser-sync	2191369	2		results			
26	https://github.com/facebook/jest	bc0f55d	2		results			
27	https://github.com/aws/aws-sdk-js	c9ac802	14		results			
28	https://github.com/websockets/ws	08c6c8b	4		results			
29	https://github.com/koajs/koa	817b498	4		results			
30	https://github.com/nodejs/readable-stream	4ba93ff	12		results			





```
50 @frontend.route('/api/update/get/<filename>', methods=['GET'])
51 def getZip(filename):
52     return make_response(open(os.path.join(
53         TEMPLATE_DIR, filename)).read())
```

# Guard with Continuous Integration



# Integrations

- Enforce secure defaults + secure frameworks at CI time
  - Easy to add to CI as either a Docker container or Linux binary
  - JSON output → easy to integrate with other systems

# Integrations - GitLab CI

```
3  
4  semgrep:  
5    image: returntocorp/semgrep-agent:v1  
6    script:  
7      - python -m semgrep_agent --config https://semgrep.dev/p/flask  
8
```

# Integrations - GitLab CI

✓ Linters  
on: pull\_request

✓ super-linter

✓ pre-commit

✗ semgrep with managed policy

Linters / semgrep with managed policy  
failed 1 hour ago in 1m 25s

▶ ✓ Set up job

▶ ✓ Pull returntocorp/semgrep-action:v1

▶ ✓ Run actions/checkout@v1

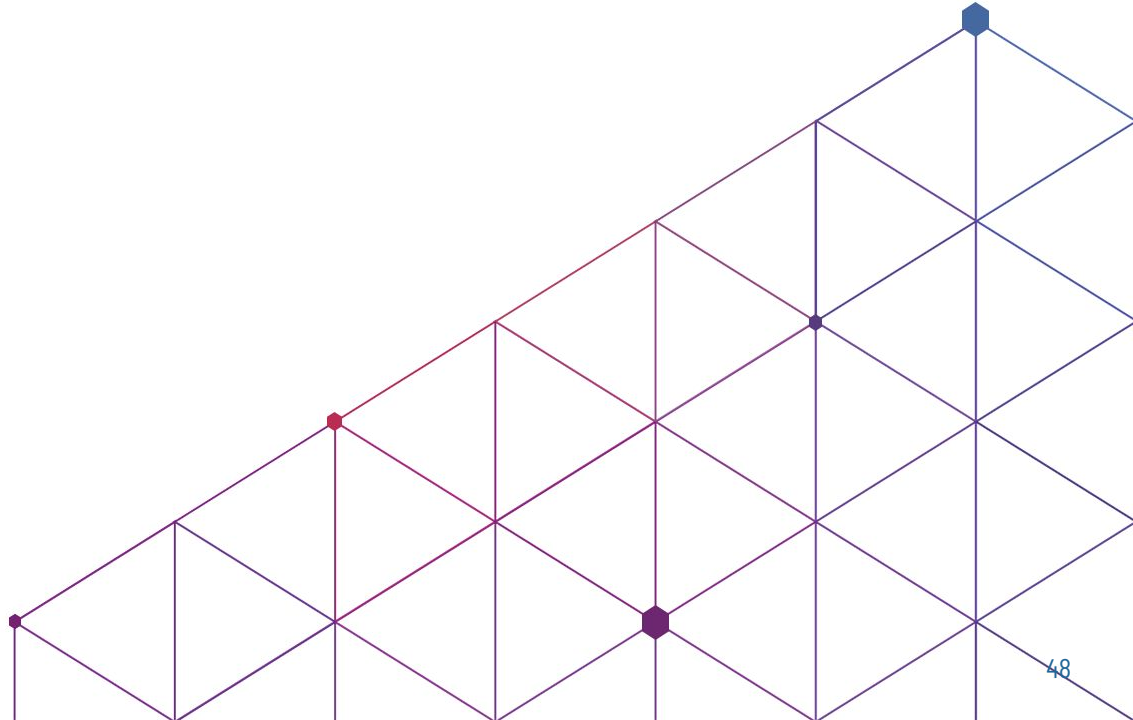
▼ ✗ Run returntocorp/semgrep-action@v1

```
GITHUB_EVENT_NAME -e GITHUB_SERVER_URL -e GITHUB_API_URL -e GITHUB_GRAPHQL_URL
-e ACTIONS_RUNTIME_URL -e ACTIONS_RUNTIME_TOKEN -e ACTIONS_CACHE_URL -e GITHUB_
"/home/runner/work/_temp/_github_home":"/github/home" -v "/home/runner/work/_te
returntocorp/semgrep-action:v1
6 === detecting environment
7 | versions      - semgrep 0.17.0 on Python 3.8.5
8 | environment   - running in github-actions, triggering event is 'pull_request'
9 | semgrep.dev   - logged in as deployment #1
10 === setting up agent configuration
11 | using semgrep rules configured on the web UI
12 | using default path ignore rules of common test and dependency directories
13 | adding further path ignore rules configured on the web UI
14 | looking at 1 changed path
15 | found 1 file in the paths to be scanned
16 === looking for current issues in 1 file
17 | 1 current issue found
18 === looking for pre-existing issues in 1 file
19 | 1 pre-existing issue found
20 python.flask.security.injection.path-traversal-open.path-traversal-open
21      .py:459
22
23 459| open(path).readlines(), mimetype="text/plain"
24
25     = Found request data in a call to 'open'. Ensure the request data is
26       validated or sanitized, otherwise it could result in path traversal
27       attacks.
28
29 === exiting with failing status
```

▶ ✓ Complete job

```
10  === setting up agent configuration
11  | using semgrep rules configured on the web UI
12  | using default path ignore rules of common test and dependency directories
13  | adding further path ignore rules configured on the web UI
14  | looking at 1 changed path
15  | found 1 file in the paths to be scanned
16  === looking for current issues in 1 file
17  | 1 current issue found
18  === looking for pre-existing issues in 1 file
19  | 1 pre-existing issue found
20  python.flask.security.injection.path-traversal-open.path-traversal-open
21  [REDACTED].py:459
22  |
23  459 | open(path).readlines(), mimetype="text/plain"
24  |
25  = Found request data in a call to 'open'. Ensure the request data is
26  validated or sanitized, otherwise it could result in path traversal
27  attacks.
28
29  === exiting with failing status
```

# registry





# Community rule registry

## community participation

- 700+ rules under development by r2c + community
- **NodeJsScan v4 is powered by semgrep!**
- [Gosec](#) and [find-sec-bugs](#) checks have been ported - no compilation required 👍
- Rule ideas contributed by Django co-creator
- Suggestions by Flask team
- Independent security researchers via HackerOne & elsewhere

go.otto.security.audit.dangerous-execution.dangerous-execution +

go.grpc.security.grpc-server-insecure-connection.grpc-server-insecure-connection +

Found an insecure gRPC connection. This allows for a connection without encryption to this server. A malicious attacker could tamper with the gRPC message, which could compromise the machine.

Example

No examples found

References

[github.com/dgryski/semgrep-go](https://github.com/dgryski/semgrep-go)

semgrep rules by  
Damian Gryski,  
(author of [Go-Perfbook](#))

23 lines (23 sloc) | 980 Bytes

```
1 rules:
2   - id: use-math-bits
3     patterns:
4       - pattern-either:
5         - pattern: $X >> $N | $X << (8 - $N)
6         - pattern: $X << $N | $X >> (8 - $N)
7         - pattern: $X >> (8 - $N) | $X << $N
8         - pattern: $X << (8 - $N) | $X >> $N
9         - pattern: $X >> $N | $X << (16 - $N)
10        - pattern: $X << $N | $X >> (16 - $N)
11        - pattern: $X >> $N | $X << $N
12        - pattern: $X << $N | $X >> $N
13        - pattern: $X >> (32 - $N)
14        - pattern: $X << (32 - $N)
```

95 lines (95 sloc) | 2.71 KB

```
1 rules:
2   - id: odd-sequence-ifs
3     patterns:
4       - pattern-either:
5         - pattern: |
6           if $X { return ... }
7           if $X { ... }
8         - pattern: |
9           if ! $X { return ... }
10          if $X { ... }
11        - pattern: |
12          if $X { return ... }
13          if ! $X { ... }
```

11 lines (11 sloc) | 323 Bytes

```
1 rules:
2   - id: odd-compound-expression
3     patterns:
4       - pattern-either:
5         - pattern: $X += $X + $Y
6         - pattern: $X += $X - $Y
7         - pattern: $X -= $X + $Y
```

# Community rule registry

[semgrep.dev/registry](https://semgrep.dev/registry) ⇒ [github.com/returntocorp/semgrep-rules](https://github.com/returntocorp/semgrep-rules)

```
$ brew install semgrep  
$ semgrep --config=<url>
```



### r2c

Go Java JavaScript Python

Default ruleset, by r2c

audit cookies correctness  
crypto csrf injection security  
spring xss xxe

### r2c-ci

Go Java JavaScript Python

Scan for runtime errors, logic bus, and high-confidence security vulnerabilities....

CI cookies correctness crypto  
csrf injection security spring  
xss xxe logic logic bugs

### r2c-security-audit

Ruby JavaScript Go Java C

Scan code for potential security issues that require additional review. Recommended for tea...

security audit xxe injection  
deserialization xss jwt csrf  
crypto

## Languages and Frameworks

Get security coverage for the languages and frameworks you use.

### minusworld.ruby-all

### python

Python

Default ruleset for Python, by r2c

security correctness

### javascript

JavaScript

Default ruleset for JavaScript, by r2c

security correctness

```
$ semgrep --config=https://semgrep.dev/p/python
```

# Coming Soon

1000 rules!

Semgrep Community!

Centrally manage Semgrep on your repos!

Tainting (intrafile)

```
eval ($X:<user_data>)
```

The screenshot displays the Semgrep web interface. At the top, there's a navigation bar with 'Semgrep', 'Write', 'Explore', 'Manage', and 'Docs'. The main section is titled 'Deployments' and has a sidebar with 'Projects', 'Policies' (selected), 'Actions', and 'Manage Access'. The 'My Policies' section shows four policy cards: 'Web Apps' (10 items), 'Python Packages' (7 items), 'policy one' (1 item), and 'bandit' (1 item). Each card lists 'Used on:' with repository names. Below these is a 'default' policy card. To the right of the 'My Policies' section is an 'FAQ' box with links like 'What is a project?', 'What is a policy?', 'How do I add to my policy?', 'How do I remove from my policy?', and 'How do I edit individual rules within a ruleset?'. The 'Python Packages' section at the bottom lists several rulesets and rules with their status (notifying, blocking, or not blocking CI).

Policy	Items	Used on:
Web Apps	10 items	returntocorp/semgrep-app, returntocorp/echelon-backend and 4 more...
Python Packages	7 items	pallets/flask, semgrep and 2 more...
policy one	1 item	daghan/lets-be-bad-guys
bandit	1 item	chmcreery/test, dormbase/dormbase
default	1 item	returntocorp/infrastructure, returntocorp/cli and 5 more...

Ruleset	Rule	Status
bandit	minusworld.python-insecure-transport-starter	notifying, blocking CI
	r2c-CI	notifying, not blocking CI
r2c-security-audit	python.attr.correctness.mutable-initializer.attr.mutable-initializer	notifying, blocking CI
	python.lang.best-practice.pdb.python-debugger-found	notifying, blocking CI
python.lang.correctness.concurrent.uncaught-executor-exceptions		notifying, blocking CI



# Semgrep

lightweight static analysis for many languages

Locally:

1. `(pip|brew) install semgrep`
2. `semgrep --config=r2c`

Bence Nagy | [bence@r2c.dev](mailto:bence@r2c.dev)

[r2c.dev](https://r2c.dev) | [@r2cdev](https://twitter.com/r2cdev)

<https://r2c.dev/survey> ← plz :)

Online editor:

- [semgrep.live](https://semgrep.live)

