



Enforcing Code & Security Standards with Semgrep

Colleen Dai | colleen@returntocorp.com

 [@ar2cdev](https://twitter.com/ar2cdev)

👁️ So you're doing a code review...

app/controllers/widget_controller.rb

85 +

86 + before_action :ensure_user



You 3 days ago Member



Hey this looks great, but we've stopped using `ensure_user()`, could you use `ensure_logged_in()` instead?

👁️ So you're doing a code review...

app/controllers/widget_controller.rb

```
85 +  
86 +      before_action :ensure_user
```



You 3 days ago Member



Hey this looks great, but we've stopped using `ensure_user()`, could you use `ensure_logged_in()` instead?

- Every code base has assumptions, requirements, coding standards
- Tools exist for language/framework generic checks
- What about code patterns **unique** to your project/org?

Semgrep tl;dr

- A customizable, lightweight, static analysis tool for finding bugs



Semgrep

Find bugs and enforce code standards.

Semgrep is a lightweight, offline, [open-source](#), static analysis tool. Run community rules or write your own in less than 5 minutes. Configure and run in 2 minutes. ⚡

Semgrep Trophy Case

CVEs			
CVE	Semgrep rule	Affected software	Description
CVE-2019-5479	javascript.lang.security.detect-non-literal-require	larbitbase-api < v0.5.5	An unintended require vulnerability in <v0.5.5 larvitbase-api may allow an attacker to load arbitrary non-production code (JavaScript file).
CVE-2020-8128	javascript.lang.security.detect-non-literal-require	jsreport < 2.5.0	An unintended require and server-side request forgery vulnerabilities in jsreport version 2.5.0 and earlier allow attackers to execute arbitrary code.
CVE-2020-8129	javascript.lang.security.detect-non-literal-require	script-manager < 0.8.6	An unintended require vulnerability in script-manager npm package version 0.8.6 and earlier may allow attackers to execute arbitrary code.
CVE-2020-7739	javascript.phantom.security.audit.phantom-injection	phantomjs-seo	This affects all versions of package phantomjs-seo. It is possible for an attacker to craft a url that will be passed to a PhantomJS instance allowing for an SSRF attack.

who is?

me:

Colleen Dai, security software engineer @ r2c
Graduated Stanford with B.S. of C.S., M.S. Stats



r2c:

We're an SF based static analysis startup on a mission to profoundly improve software security and reliability.



Outline

1. **Background**
2. `grep` and Abstract Syntax Trees (ASTs)
3. Semgrep Examples!
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](#)

static-analysis

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

Language Support |

Language	Status	Extensions	Tags
Go	GA ⓘ	.go	go, golang
Java	GA ⓘ	.java	java
JavaScript	GA ⓘ	.js, .jsx	js, jsx, javascript
JSON	GA ⓘ	.json	json, JSON, Json
Python	GA ⓘ	.py, .pyi	python, python2, python3, py
Ruby	GA ⓘ	.rb	ruby, rb
TypeScript	GA ⓘ	.ts, .tsx	ts, tsx, typescript
JSX	GA ⓘ	.js, .jsx	js, jsx, javascript
TSX	GA ⓘ	.ts, .tsx	ts, tsx, typescript
OCaml	alpha ⓘ	.ml, .mli	ocaml, ml
PHP	alpha ⓘ	.php	php
C	alpha ⓘ	.c	c
Generic (YAML, ERB, Jinja, etc.)	alpha ⓘ	*	generic
Rust	develop ⓘ	.rs	rust, Rust, rs
Lua	develop ⓘ	.lua	lua



Outline

1. Background
2. **grep and Abstract Syntax Trees (ASTs)**
3. Semgrep Examples!
4. Integration into CI/CD
5. Semgrep Rules Registry

grep, ASTs, and Semgrep

```
exec("ls")

exec(some_var)

exec (arg)

exec(
    bar
)

other_exec(foo)

// exec(foo)

print("exec(bar)")
```

✓ Easy - `exec\ (`

✓ Easy - `exec\ (`

⚠ Handle whitespace `exec\s*\ (`

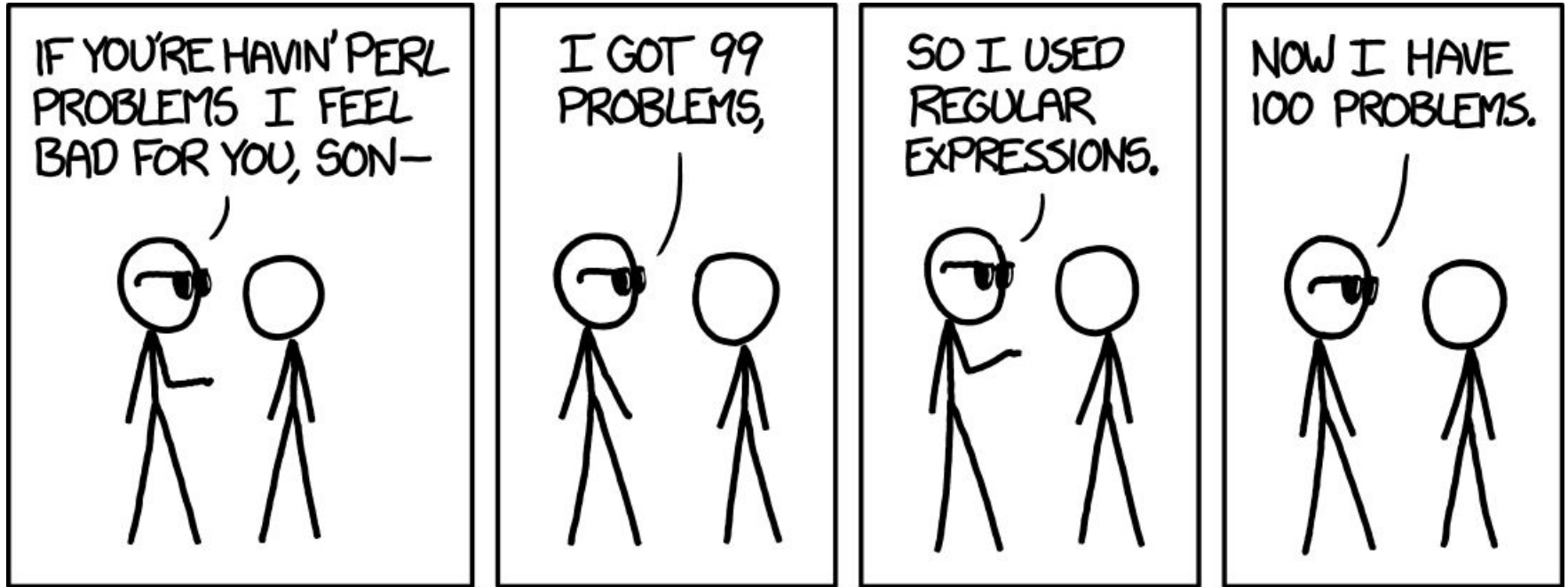
⚠ 😂 Handle whitespace/newlines

⚠ 😂😂 Method suffix matches `exec`

⚠ 😂😂 Is this a comment?

⚠ 😂😂 Is this a string literal?

xkcd 1171



Code is not a string, it's a tree



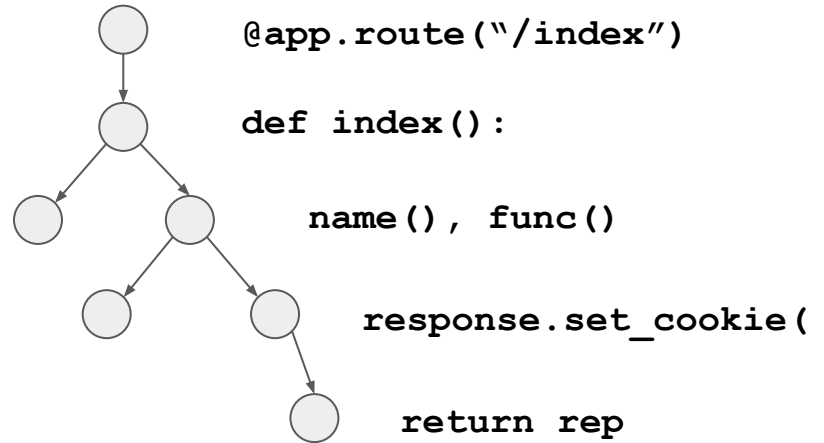
string

```
@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 `eval()`
in only 307 LOC 



```
yeonjuan Update: support globalThis (refs #12670) (#12774) 183e300 on Mar 17
20 contributors
307 lines (258 sloc) 9.24 KB
Raw Blame History
1 /**
2  * @fileoverview Rule to flag use of eval() statement
3  * @author Nicholas C. Zakas
4  */
5
6 "use strict";
7
8 -----
9 // Requirements
10 -----
11
12 const astUtils = require("../utils/ast-utils");
13
14 -----
15 // Helpers
16 -----
17
18 const candidatesOfGlobalObject = Object.freeze([
19   "global",
20   "window",
21   "globalThis"
22 ]);
23
24 /**
25  * Checks a given node is a Identifier node of the specified name.
26  * @param {ASTNode} node A node to check.
27  * @param {string} name A name to check.
28  * @returns {boolean} `true` if the node is a Identifier node of the name.
29  */
30 function isIdentifier(node, name) {
31   return node.type === "Identifier" && node.name === name;
32 }
33
34 /**
35  * Checks a given node is a Literal node of the specified string value.
36  * @param {ASTNode} node A node to check.
37  * @param {string} name A name to check.
38  * @returns {boolean} `true` if the node is a Literal node of the name.
39  */
40 function isConstant(node, name) {
41   switch (node.type) {
42     case "Literal":
43       return node.value === name;
```

<https://github.com/eslint/eslint/blob/master/lib/rules/no-eval.js>

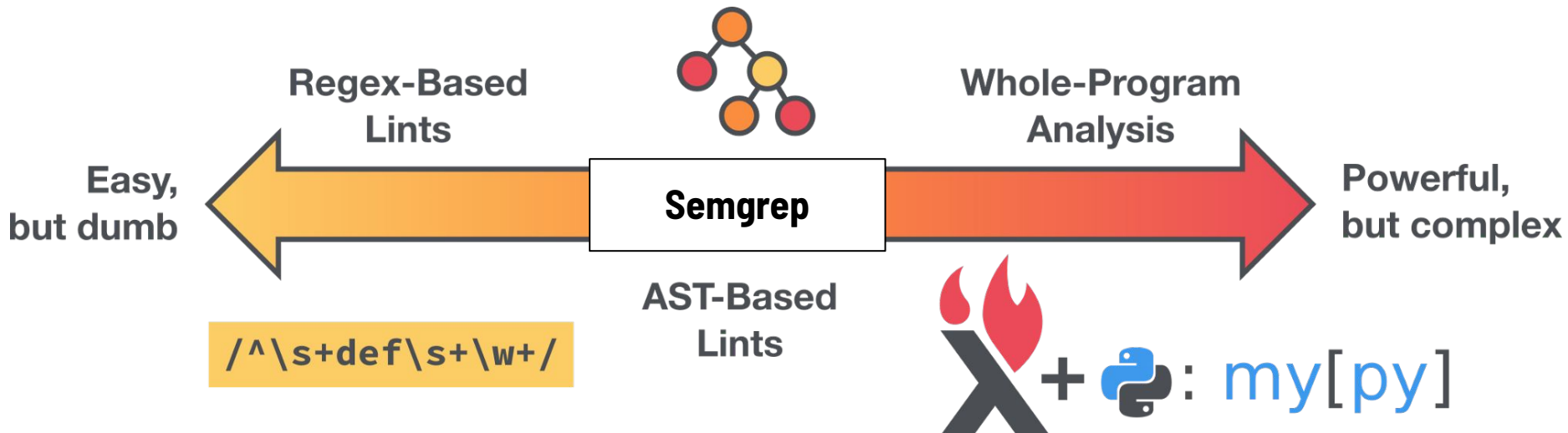


Static Analysis at Scale: An Instagram Story



Benjamin Woodruff [Follow](#)

Aug 15, 2019 · 13 min read



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

Semgrep lets you reason about your **analysis**
the way you reason about your **code**.

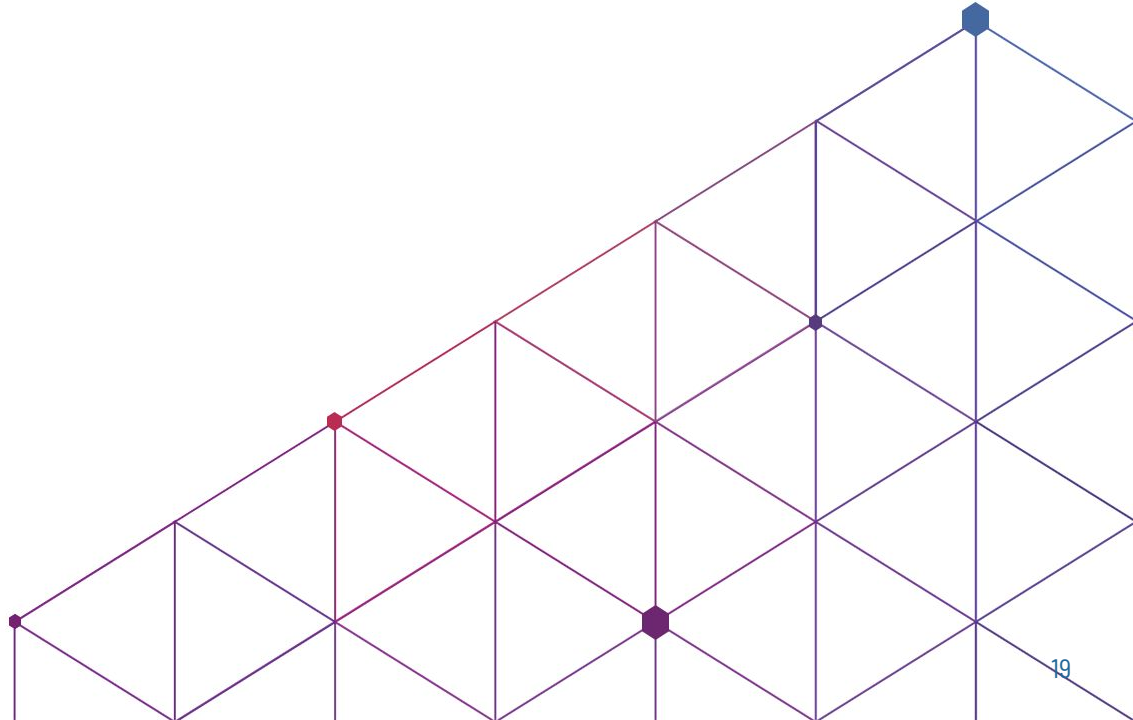
<https://r2c.dev/blog/2020/why-i-moved-to-semgrep-for-all-my-code-analysis/>

Outline

1. Background
2. `grep` and Abstract Syntax Trees (ASTs)
- 3. Semgrep Examples!**
4. Integration into CI/CD
5. Semgrep Rules Registry

Tutorials

1. Ellipsis ("...") operator
2. Metavariables
3. Composing Patterns
4. Advanced Features



Finding Insecure Functions: Node Exec

(... operator)

```
exec("ls");
```

⇒ <https://semgrep.live/Xnw>

Full Solution: <https://semgrep.live/1Kk>

Hard-coded Secrets, Constant String Arguments

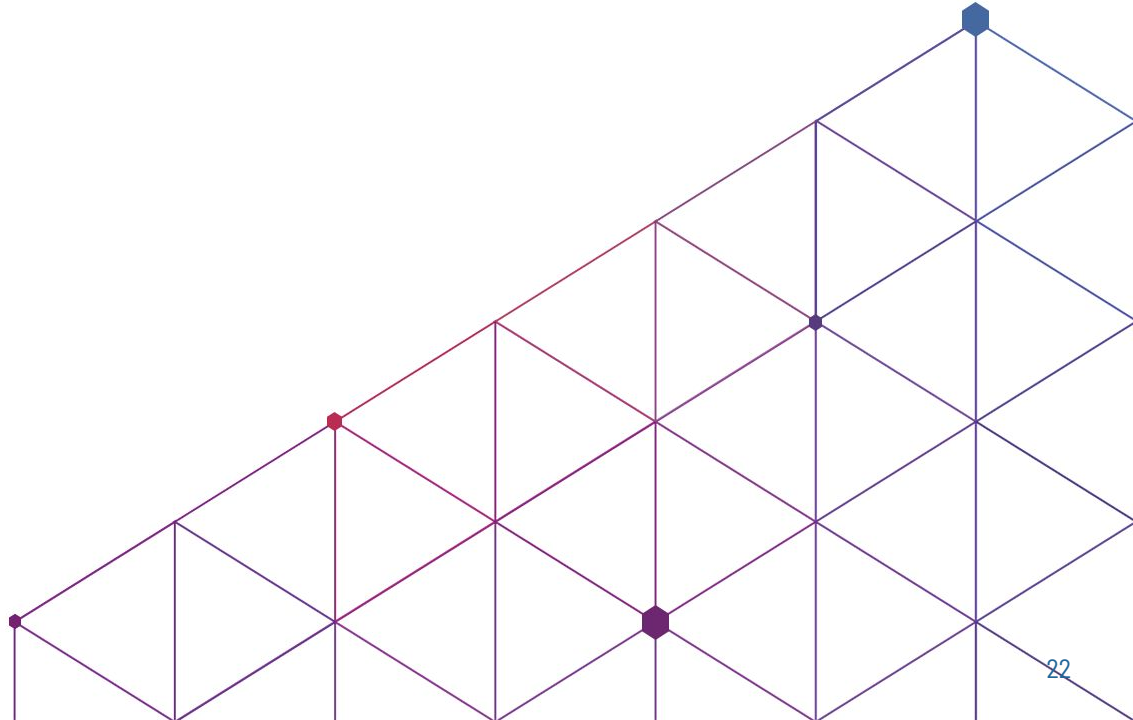
```
s3 = boto3.client(  
    "s3",  
    aws_secret_access_key = "abcd...",  
    aws_access_key_id = "AKIA...")
```

⇒ <https://semgrep.live/RG08/>

Full Solution: <https://semgrep.live/A89w/>

Tutorials

1. Ellipsis (“...”) operator
2. **Metavariables**
3. Composing Patterns
4. Advanced Features



Finding Uses of `unsafe`

(Metavariables)

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

⇒ <https://semgrep.live/nJNZ>

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


Path Traversal with send_file

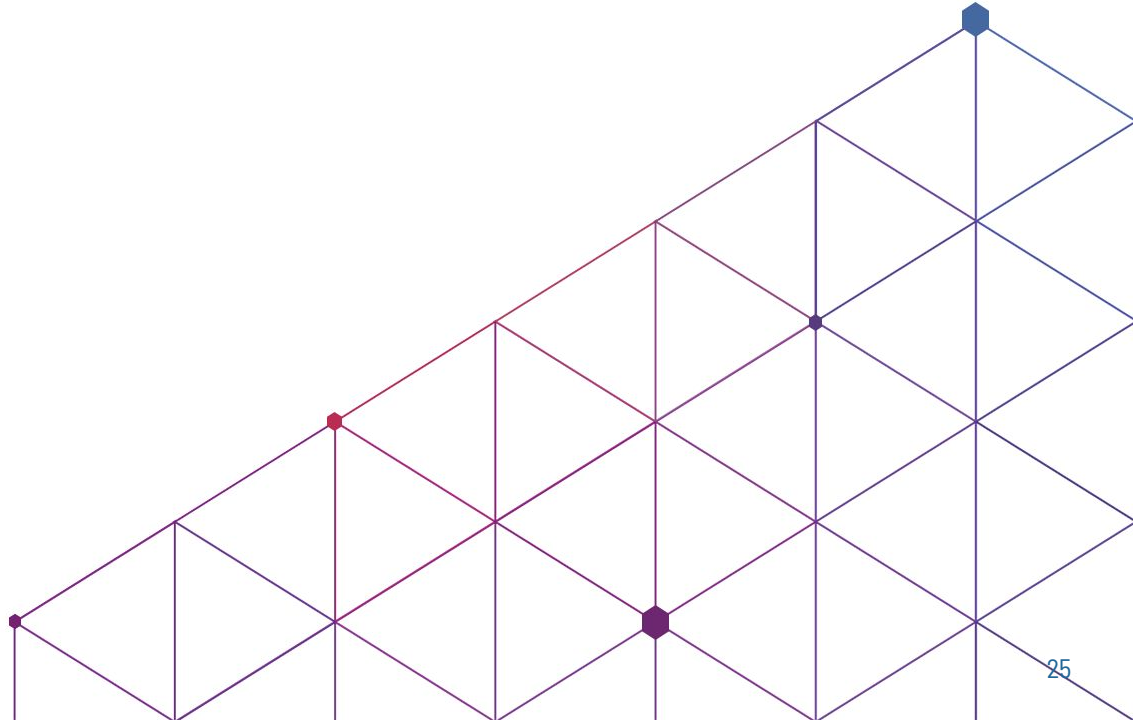
```
@app.route("/get_file/<filename>")
def get_file(filename):
    print("sending file", filename)
    return send_file(filename, as_attachment=True)
```

⇒ <https://semgrep.live/4bXx>

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

Tutorials

1. Ellipsis ("...") operator
2. Metavariables
3. Composing Patterns 
4. Advanced Features



Cookies 🍪

```
@app.route("/index")
def index():
    r = response.set_cookie("username", "drew")
    return r
```

⇒ <https://semgrep.live/8dJ>

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

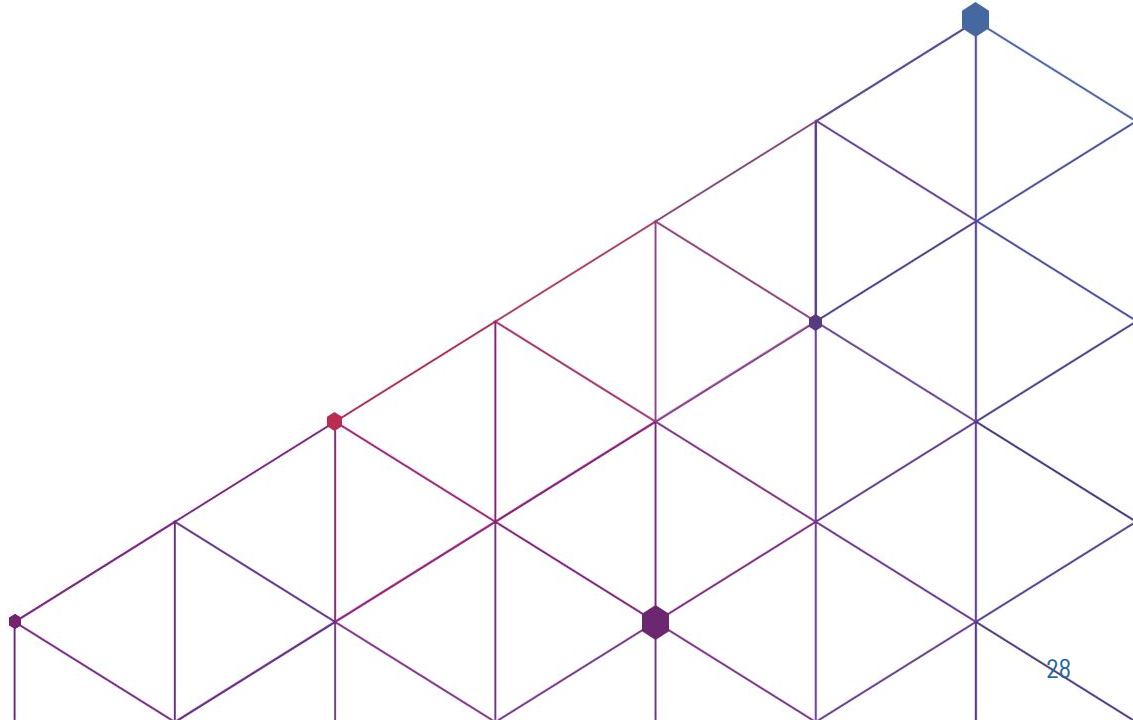
Finding Insecure SSL Configurations (Composing patterns)

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

⇒ <https://semgrep.dev/s/N4yN/>

Tutorials

1. Ellipsis (“...”) operator
2. Metavariables
3. Composing Patterns
4. Advanced Features 🧑🔬



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>

Insecure SSL Configuration

(Autofix)

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

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



Detect HTTP

(Inline String Regexes)

```
func bad1() {  
    req, err := http.NewRequest("GET", "http://example.com", nil)  
}
```

```
pattern: |  
    http.NewRequest(..., "=~/[hH][tT][tT][pP]://.*/", ...)
```

<https://semgrep.dev/editor?registry=problem-based-packs.insecure-transport.go-stdlib.http-customized-request>

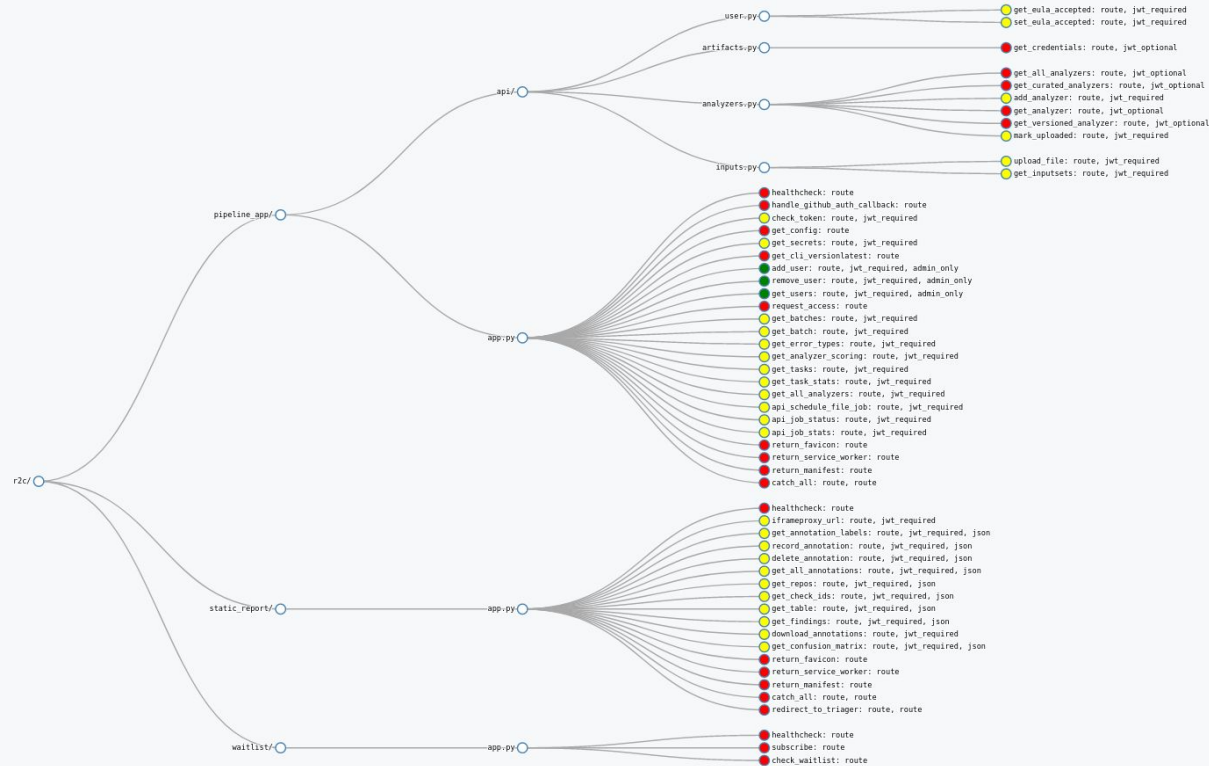
Know When New Routes Are Added

(Route Detection)

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

<https://semgrep.live/r6o1>

Semgrep application: code inventory



Terraform

(Generic Language Support)

```
resource "aws_s3_bucket" "b" {  
  bucket = "my-tf-test-bucket"  
  acl    = "public-read-write"  
  ...  
}
```

```
pattern: |  
  acl = "public-read-write"
```

<https://semgrep.dev/s/ne0Z/>

SEMGREP

Home

Getting started

Writing rules

Getting started

Syntax

Pattern syntax

Rule syntax

Examples

Pattern examples

Rule examples


Testing rules

Running rules

Ignoring findings

Managing CI policy

Integrations

Experiments 

STATS

Rules

Supported languages

Trophy case

HELP

FAQ

Support

Contributing

Docs » Syntax » Writing rules » Semgrep » Pattern syntax

Pattern syntax

Info

Getting started with rule writing? Try the [Semgrep Tutorial](#) 📖

This document describes Semgrep's pattern syntax. You can also see pattern [examples by language](#). In the command line, patterns are specified with the flag `--pattern` (or `-e`). Multiple coordinating patterns may be specified in a configuration file. See [rule syntax](#) for more information.

- [Expression matching](#)
- [String matching](#)
- [Ellipsis operator](#)
 - [Function calls](#)
 - [Method calls](#)
 - [Function definitions](#)
 - [Class definitions](#)
 - [Strings](#)
 - [Binary operations](#)
 - [Arrays](#)
 - [Conditionals and loops](#)
- [Metavariables](#)
- [Typed Metavariables](#)
- [Equivalences](#)
 - [Imports](#)
 - [Constants](#)
- [Deep expression operator](#)
- [Limitations](#)
 - [Statements types](#)
 - [Partial statements](#)

Search: Vulnerabilities



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

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

ID	Analyzer	Version	YAML URL	Created	Error Rate	Status
1131	dev/semgrep	0.14.0				94.7% complete
1130	dev/semgrep	0.14.0				99.8% complete
1129	dev/semgrep	0.14.0				99.7% complete
1128	dev/semgrep	0.14.0				99.8% complete
1127	dev/semgrep	0.14.0	github-1200-depends-on-flask/0.0.1	48 minutes ago	23.98% error rate	complete
1126	dev/semgrep	0.14.0	github-1200-depends-on-flask/0.0.1	54 minutes ago	71.31% error rate	complete
1125	dev/semgrep	0.14.0	github-1200-depends-on-flask/0.0.1	59 minutes ago	96.50% error rate	complete
1124	dev/semgrep	0.14.0	npm-github-1000-latest-2019-09-17/...	4 days ago	2.40% error rate	complete
1123	dev/semgrep	0.14.0	npm-latest-2019-08-26/0.0.2	4 days ago	0.00% error rate	starting up

Run Job

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

Filter by repositories, commit hashes, or checks:

filter by repos...

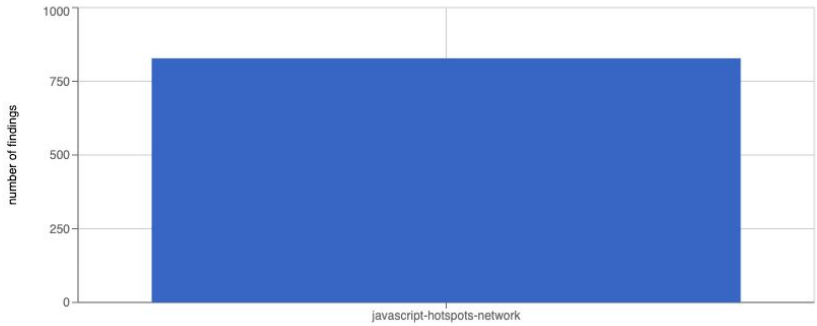
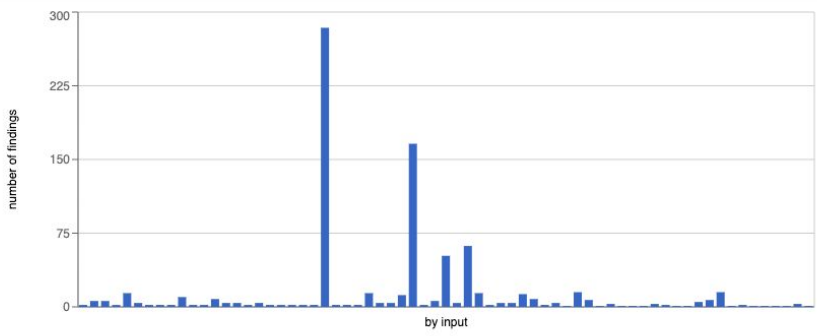
filter by commit hashes...

filter by check ids...

filter exclude path prefixes...

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/drivardan/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/chj/blended	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/rwjlblue/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	4ba93f8	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())
```


Outline

1. Background
2. `grep` and Abstract Syntax Trees (ASTs)
3. Semgrep Examples!
- 4. Integration into CI/CD**
5. Semgrep Rules Registry

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

Use in CI

`git pre-commit` [GitHub](#) [GitLab](#) [CircleCI](#) [AppVeyor](#) [Travis](#)

Add this snippet in your `.github/workflows/semgrep.yml`:

```
name: Semgrep
on: [push, pull_request]
jobs:
  semgrep:
    runs-on: ubuntu-latest
    name: Check
    steps:
      - uses: actions/checkout@master
```

Integrations

<p>▼ Linters on: pull_request</p>	<p>Linters / semgrep with managed policy failed 1 hour ago in 1m 25s</p>
<p>✓ super-linter</p>	<p>▶ ✓ Set up job</p>
<p>✓ pre-commit</p>	<p>▶ ✓ Pull returntocorp/semgrep-action:v1</p>
<p>✗ semgrep with managed policy</p>	<p>▶ ✓ Run actions/checkout@v1</p>
	<p>▼ ✗ Run returntocorp/semgrep-action@v1</p> <pre>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</pre> <p>▶ ✓ Complete job</p>

Outline

1. Background
2. `grep` and Abstract Syntax Trees (ASTs)
3. Semgrep Examples!
4. Integration into CI/CD
5. **Semgrep Rules Registry**

Community rule registry

semgrep.dev/registry ⇒ github.com/returntocorp/semgrep-rules

findsecbugs



by r2c

Selected rules from FindSecBugs, a security checker for Java, rewritten in Semgrep.

Java

gosec



by Ulzii Otgonbaatar

Selected rules from gosec, a security checker for Golang, rewritten in Semgrep.

Go

dgryski.semgrep-go



by Damian Gryski

Rules for finding odd Go code. See github.com/dgryski/semgrep-go to contribute.

Go

nodejsscan



by Ajin Abraham

Rules from the preeminent Node.js security scanner, NodeJSScan.

JavaScript



r2c

Go JavaScript Python

Default ruleset, by r2c

audit cookies correctness
crypto csrf injection security
spring xss xxe

r2c-ci

Go 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/r2c
```

Partnering with OWASP




- New partnership between Semgrep + OWASP [ASVS](#), [Cheat Sheets](#)
- **Goal:** Out of the box support for:
 - Verifying if your code is compliant with ASVS Level 1
 - Finding code that violates Cheat Sheets best practice recommendations

Want to get involved?  [Let's talk!](#) 🙌

Thanks to [Daniel Cuthbert](#), [Joe Bollen](#), [Rohit Salecha](#), and more

 [semgrep / rules-owasp-asvs](#)

 [OWASP / CheatSheetSeries](#)

<> Code  **Issues** 27  Pull requests 9  Actions  Projects 1

Update: Adding Semgrep Rules #457



Coming Soon

Improved language support, new languages

More rules! + prevention cheatsheets

Semgrep Community!

Centrally manage Semgrep on your repos!

VS Code extension (in beta)!

Tainting + constant propagation + speedups!

The screenshot displays the Semgrep web interface for managing policies. The top navigation bar includes 'Semgrep', 'Write', 'Explore', 'Manage', and 'Docs'. The main content area is titled 'Deployments' and features a sidebar with 'Projects', 'Policies', 'Actions', and 'Manage Access'. The 'Policies' section is active, showing a grid of policy cards: 'Web Apps' (10 items), 'Python Packages' (7 items), 'policy one' (1 item), and 'bandit' (1 item). Each card lists the projects it is used on. Below the grid, there is a 'Python Packages' section with a list of rules and their status. The rules include 'bandit', 'minusworld.python-insecure-transport-starter', 'r2c-CI', 'r2c-security-audit', and several python-specific rules. Each rule has a status indicator (e.g., 'notifying', 'blocking CI', 'not blocking CI') and an 'Edit Policy' link.

Semgrep Write Explore Manage Docs

Deployments

Projects
Policies
Actions
Manage Access

My Policies

[Delete Policy](#) [New Policy](#)

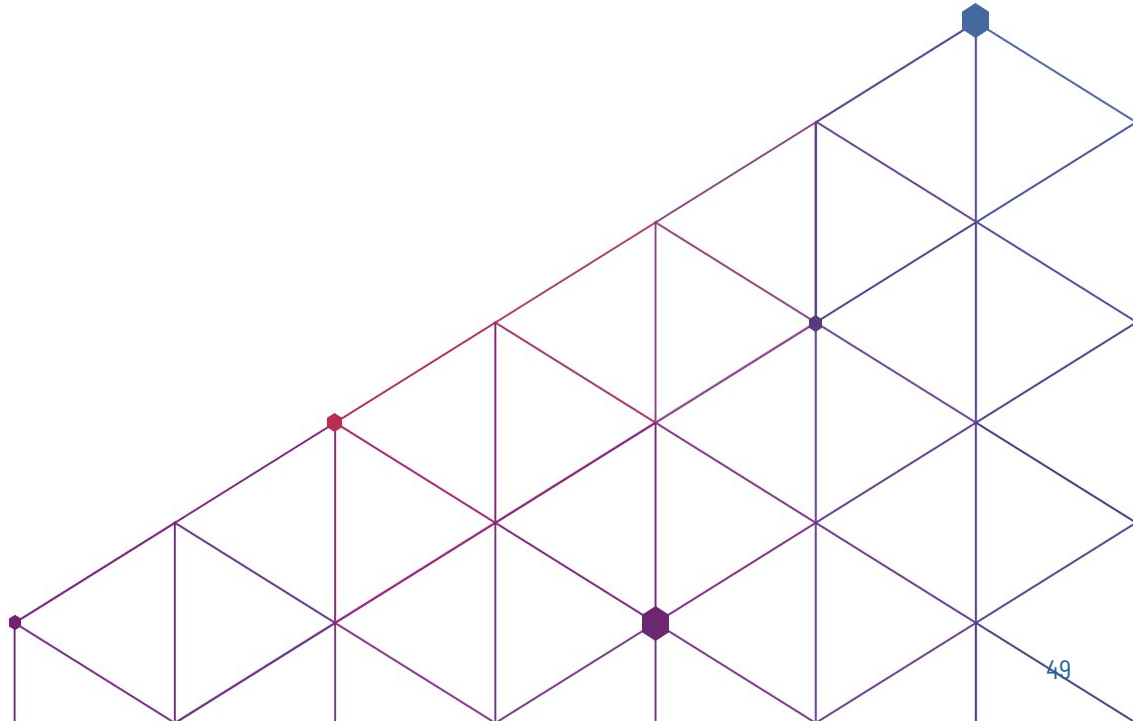
- Web Apps**
10 items
Used on: returncorp/semgrep-app, returncorp/echelon-backend and 4 more...
- Python Packages**
7 items
Used on: pallets/flask, semgrep and 2 more...
- policy one**
1 item
Used on: daghan/lets-be-bad-guys
- bandit**
1 item
Used on: chmccreeery/test, dormbase/dormbase
- default**
1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

Python Packages

[Edit Policy](#)

- RULESET **bandit** [Edit](#)
 - notifying blocking CI
- RULESET **minusworld.python-insecure-transport-starter** [Edit](#)
 - notifying not blocking CI
- > RULESET **r2c-CI** [Edit](#)
 - notifying blocking CI
- RULESET **r2c-security-audit** [Edit](#)
 - notifying not blocking CI
- RULE **python.attr.correctness.mutable-initializer.attr-mutable-initializer** [Edit](#)
 - notifying blocking CI
- RULE **python.lang.best-practice.pdb.python-debugger-found** [Edit](#)
 - notifying blocking CI
- RULE **python.lang.correctness.concurrent.uncaught-executor-exceptions** [Edit](#)
 - notifying blocking CI

Wrapping Up





Semgrep

lightweight static analysis for many languages

Locally:

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

Playground:

- semgrep.live





Semgrep

lightweight static analysis for many languages

Questions?

Colleen Dai | colleen@returntocorp.com

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

<https://r2c.dev/survey> ← plz :)

