# Detect complex code patterns using semantic grep

Drew Dennison | r2c.dev

@r2cdev @drewdennison

## whois?

**me:**

Drew Dennison, co-founder @ r2c

BS in CompSci from MIT, ex-Palantir dev

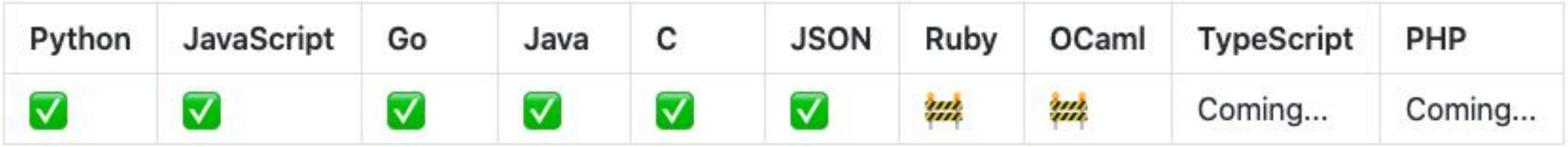
**r2c**

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

# tl;dr

- Computers excel at consistently applying human experience
- The coding future has a “paved road” of default-safe frameworks (i.e. React or ORMs)
- `grep` and static analysis tools can be slow, complicated, or noisy
- We need a tool that is fast, easy to use, precise, multi-lingual, and open source!

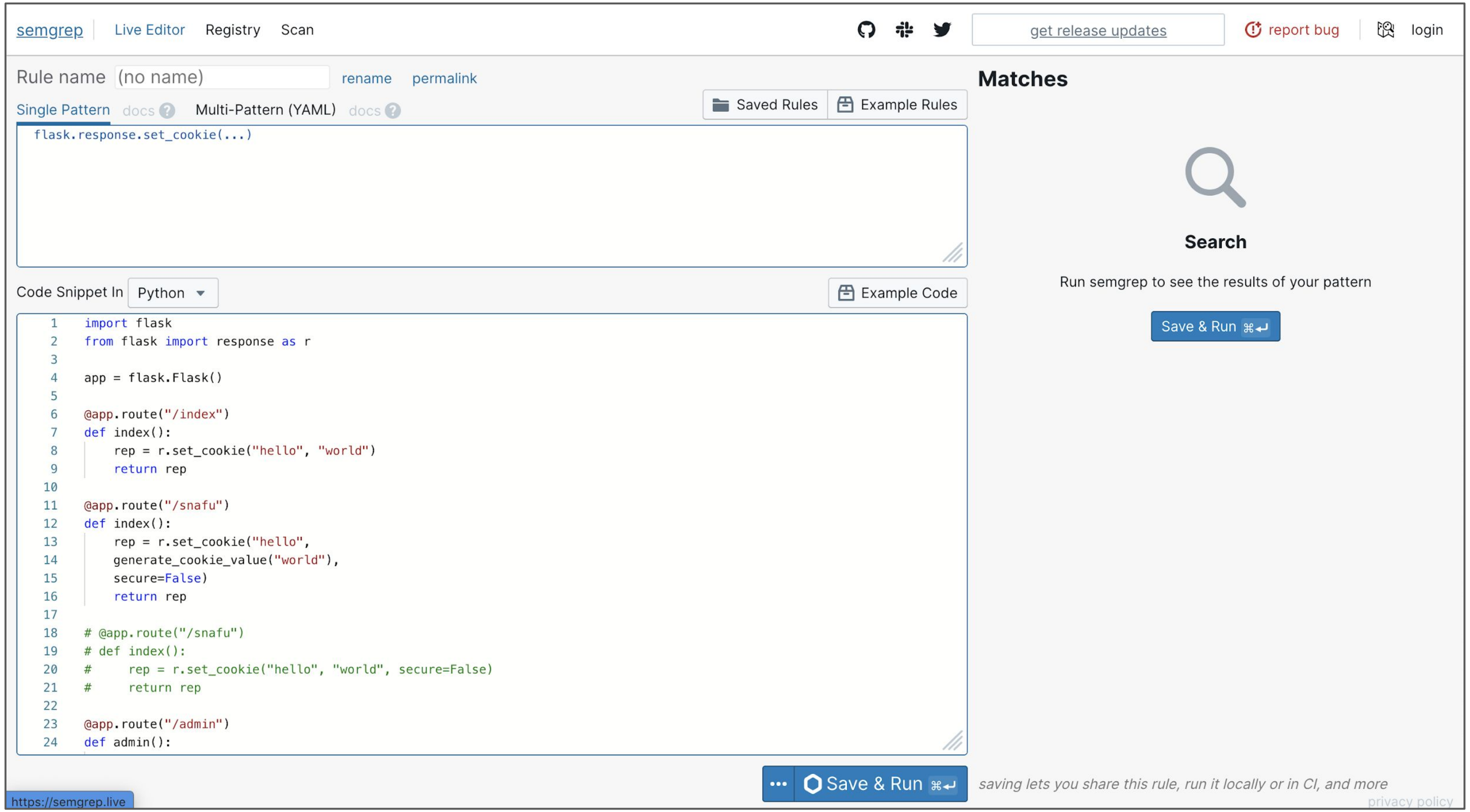
Semgrep: Lightweight static analysis for many languages. Find bug variants with patterns that look like source code.



| Python | JavaScript | Go | Java | C | JSON | Ruby | OCaml | TypeScript | PHP |
|---|---|---|---|---|---|---|---|---|---|
| ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | 🚧 | 🚧 | Coming... | Coming... |

## Outline

1. **Demo**: Dive right in!
2. **Tech Talk**: `grep` and Abstract Syntax Trees (ASTs)
3. **In Practice:** Your rules and community rule packs



```
flask.response.set_cookie(...)
```

```python
import flask
from flask import response as r

app = flask.Flask()

@app.route("/index")
def index():
    rep = r.set_cookie("hello", "world")
    return rep

@app.route("/snafu")
def index():
    rep = r.set_cookie("hello",
    generate_cookie_value("world"),
    secure=False)
    return rep

# @app.route("/snafu")
# def index():
#     rep = r.set_cookie("hello", "world", secure=False)
#     return rep

@app.route("/admin")
def admin():
```

## Online Editor: semgrep.live

## Node Exec

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

⇒ https://semgrep.live/Xnw

**Full Solution:** https://semgrep.live/1Kk

## Finding Uses of `unsafe`

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

⇒ https://semgrep.live/nJNZ

**Full Solution:** https://semgrep.live/ZqLp

## Injection Sending 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](https://semgrep.live/4bXx)

**Full Solution:** [https://semgrep.live/Pevp](https://semgrep.live/Pevp)

## Cookies 🍪

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

⇒ https://semgrep.live/8dJ

**Full Solution:** https://semgrep.live/vWX

## Autofix - API Deprecation or Misuse

```
func main() {
   http.HandleFunc(“/index”, Handler)
   http.ListenAndServe(“:80”, nil)
}
```

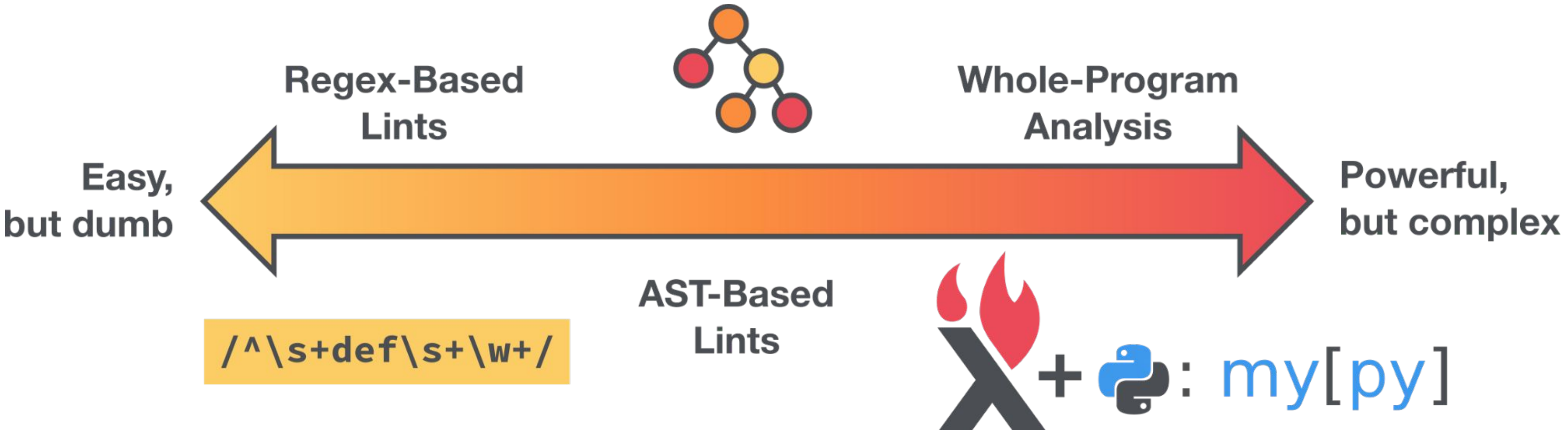
https://semgrep.live/L18G

**Solution:** https://semgrep.live/Or8R

more tutorial goodness at

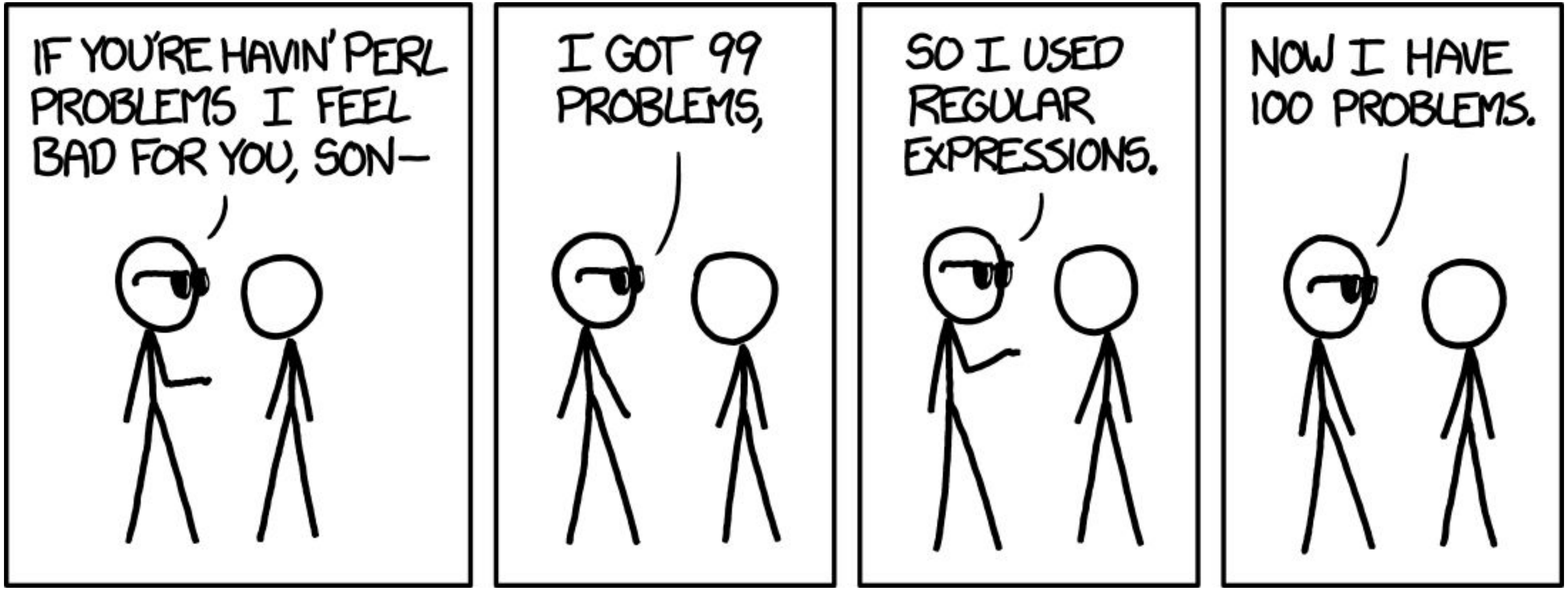
semgrep.live/learn

## Tech Talk



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

## xkcd 1171


IF YOU'RE HAVIN' PERL PROBLEMS I FEEL BAD FOR YOU, SON—
I GOT 99 PROBLEMS,
SO I USED REGULAR EXPRESSIONS.
NOW I HAVE 100 PROBLEMS.

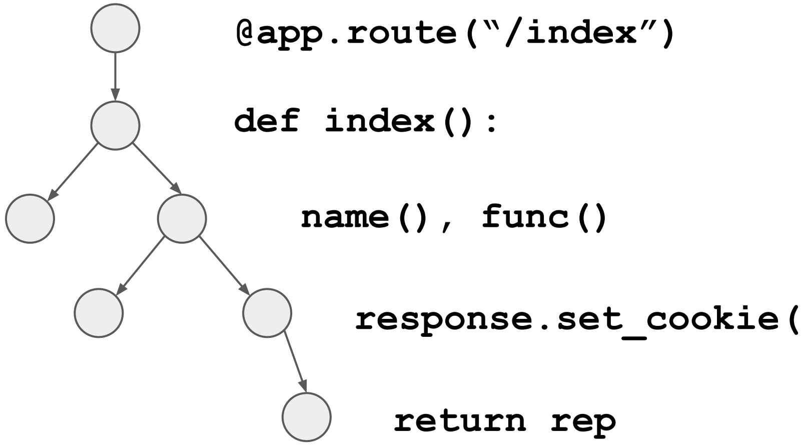
## Code is not a string, it’s a tree

**string**

**!=**

**tree**

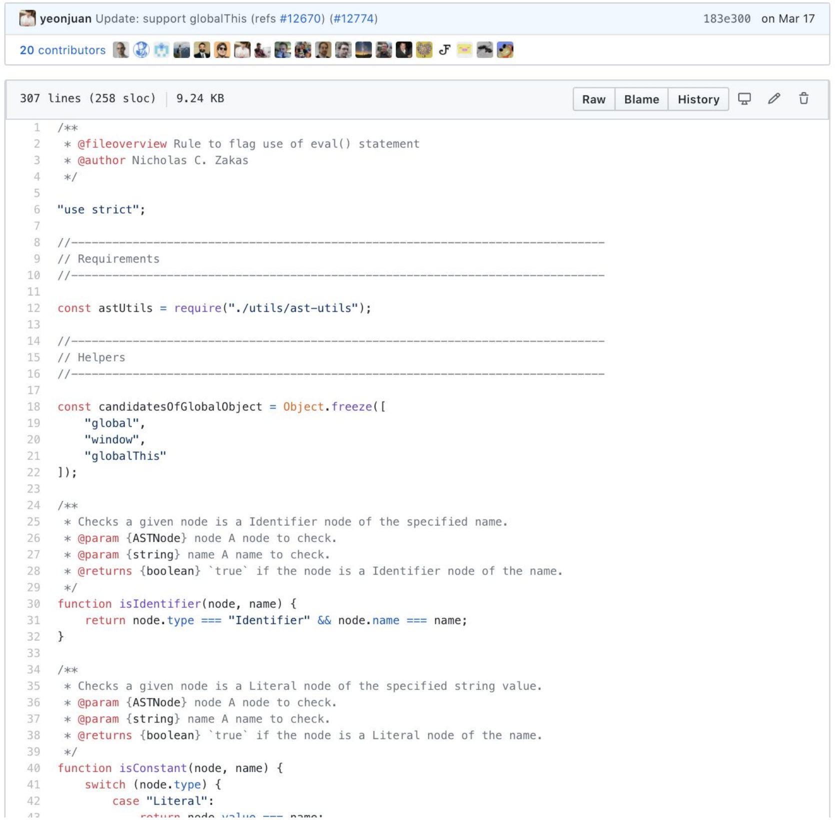
```
@app.route("/index")
def index():
    rep = response.set_cookie(name(),
secure=False, s=func())
    return rep
```



## Tree Matching 🌲

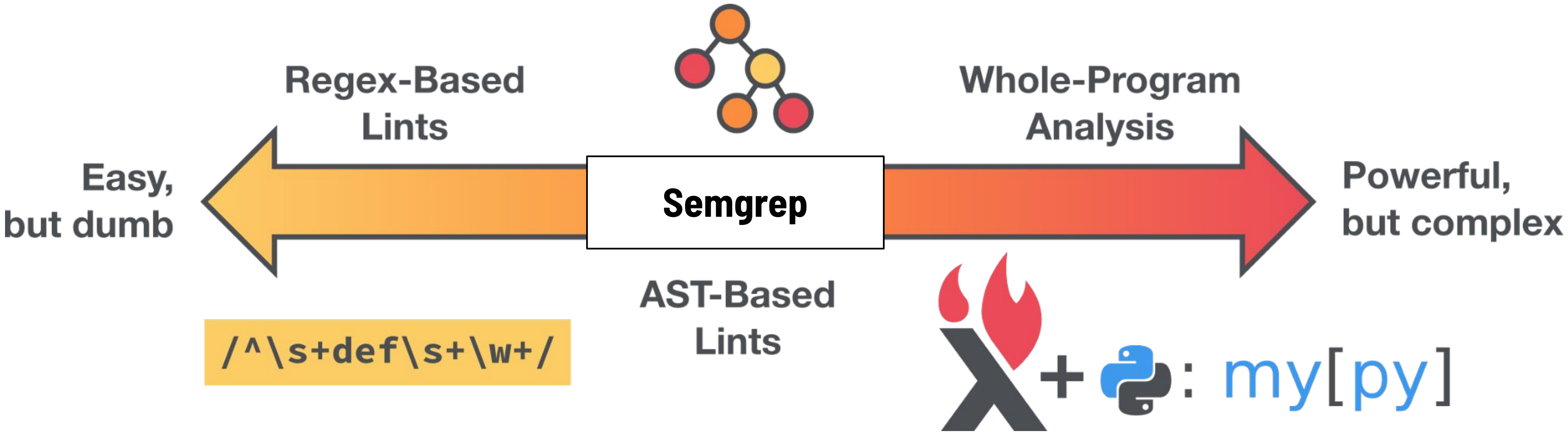
- Many tree matching tools: Gosec, Golint, Bandit, Dlint, ESLint, Flake8, Pylint, RuboCop, TSLint, and more!
- Have to become an **expert in every AST syntax** for every language your team uses
- Need **programming language expertise** to cover all idioms: languages have "more than one way to do it"
- **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

```
/**
 * @fileoverview Rule to flag use of eval() statement
 * @author Nicholas C. Zakas
 */

"use strict";

//------------------------------------------------------------------------------
// Requirements
//------------------------------------------------------------------------------

const astUtils = require("./utils/ast-utils");

//------------------------------------------------------------------------------
// Helpers
//------------------------------------------------------------------------------

const candidatesOfGlobalObject = Object.freeze([
    "global",
    "window",
    "globalThis"
]);

/**
 * Checks a given node is a Identifier node of the specified name.
 * @param {ASTNode} node A node to check.
 * @param {string} name A name to check.
 * @returns {boolean} `true` if the node is a Identifier node of the name.
 */
function isIdentifier(node, name) {
    return node.type === "Identifier" && node.name === name;
}

/**
 * Checks a given node is a Literal node of the specified string value.
 * @param {ASTNode} node A node to check.
 * @param {string} name A name to check.
 * @returns {boolean} `true` if the node is a Literal node of the name.
 */
function isConstant(node, name) {
    switch (node.type) {
        case "Literal":
            return node.value === name;
```

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



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

## In Practice

## How I write rules

1. From recent **post mortems**: what code issues contributed to it?
2. [XYZ] is a (security, performance, other) **library that everyone should use**, but they don't consistently.
3. When you review code, what changes do you frequently ask for?
4. What vulnerability classes from **bug bounty submissions** reoccur (or appear in different places of the codebase)?
5. Are there **eng / perf patterns**? Consistent exception handlers?
6. What issues were caused by **misconfigurations** in **Infrastructure-as-Code** files (JSON)?
7. What are some "**invariants**" that should hold about your code - things that should always or never be true (e.g. every admin route checks if user is admin)?
8. What methods/APIs are **deprecated** and you're trying to move away from?

## Order of API Calls Must be Enforced

```
/*
* 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

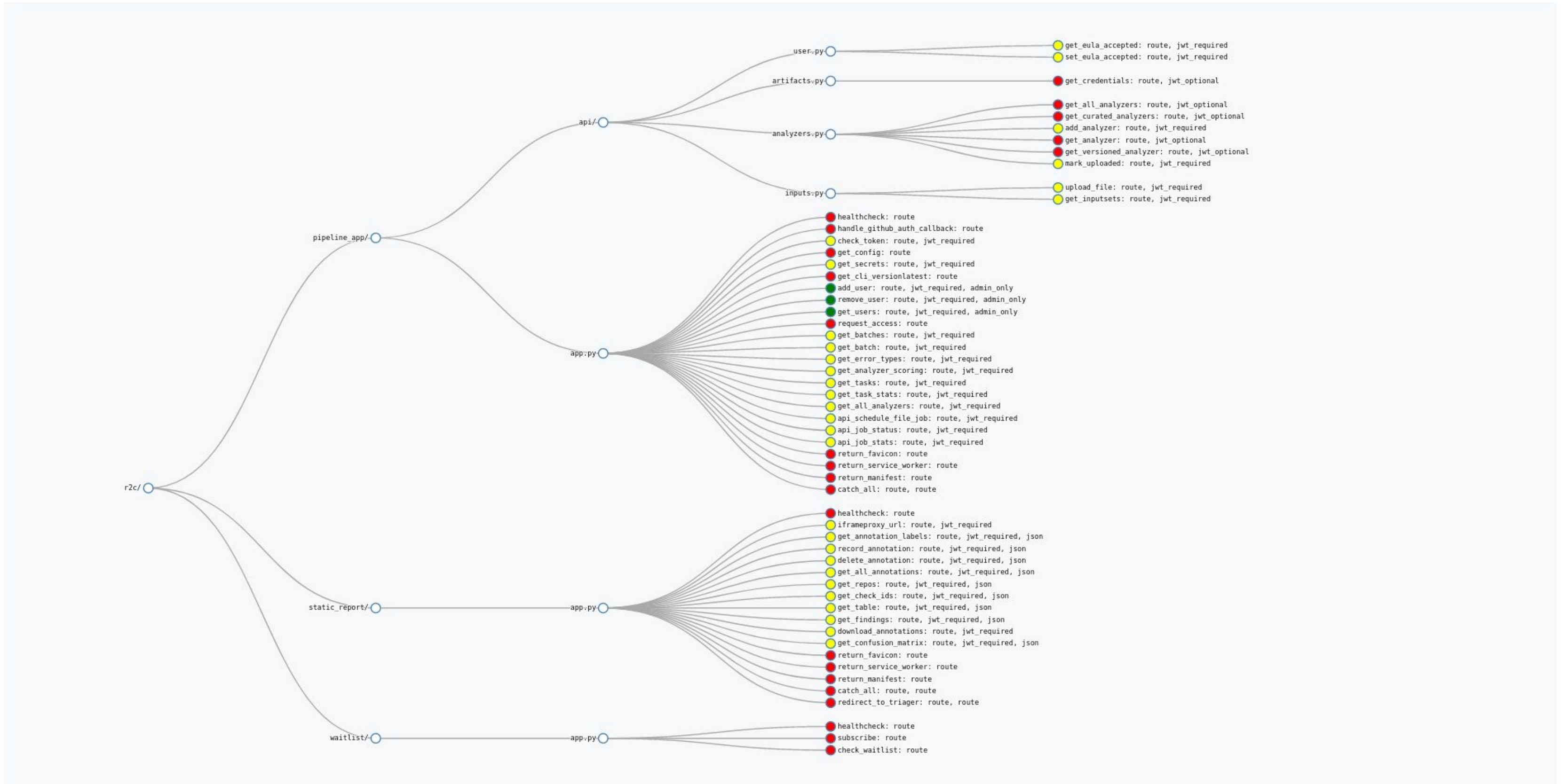
## Know When New Routes Are Added (Gorilla Toolkit)

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

https://semgrep.live/bwzd

**Solution:** https://semgrep.live/r6o1

# Semgrep application: code inventory



r2c/
pipeline_app/
api/
user.py
get_eula_accepted: route, jwt_required
set_eula_accepted: route, jwt_required
artifacts.py
get_credentials: route, jwt_optional
analyzers.py
get_all_analyzers: route, jwt_optional
get_curated_analyzers: route, jwt_optional
add_analyzer: route, jwt_required
get_analyzer: route, jwt_optional
get_versioned_analyzer: route, jwt_optional
mark_uploaded: route, jwt_required
inputs.py
upload_file: route, jwt_required
get_inputsets: route, jwt_required
app.py
healthcheck: route
handle_github_auth_callback: route
check_token: route, jwt_required
get_config: route
get_secrets: route, jwt_required
get_cli_versionlatest: route
add_user: route, jwt_required, admin_only
remove_user: route, jwt_required, admin_only
get_users: route, jwt_required, admin_only
request_access: route
get_batches: route, jwt_required
get_batch: route, jwt_required
get_error_types: route, jwt_required
get_analyzer_scoring: route, jwt_required
get_tasks: route, jwt_required
get_task_stats: route, jwt_required
get_all_analyzers: route, jwt_required
api_schedule_file_job: route, jwt_required
api_job_status: route, jwt_required
api_job_stats: route, jwt_required
return_favicon: route
return_service_worker: route
return_manifest: route
catch_all: route, route
static_report/
app.py
healthcheck: route
iframeproxy_url: route, jwt_required
get_annotation_labels: route, jwt_required, json
record_annotation: route, jwt_required, json
delete_annotation: route, jwt_required, json
get_all_annotations: route, jwt_required, json
get_repos: route, jwt_required, json
get_check_ids: route, jwt_required, json
get_table: route, jwt_required, json
get_findings: route, jwt_required, json
download_annotations: route, jwt_required
get_confusion_matrix: route, jwt_required, json
return_favicon: route
return_service_worker: route
return_manifest: route
catch_all: route, route
redirect_to_triager: route, route
waitlist/
app.py
healthcheck: route
subscribe: route
check_waitlist: route

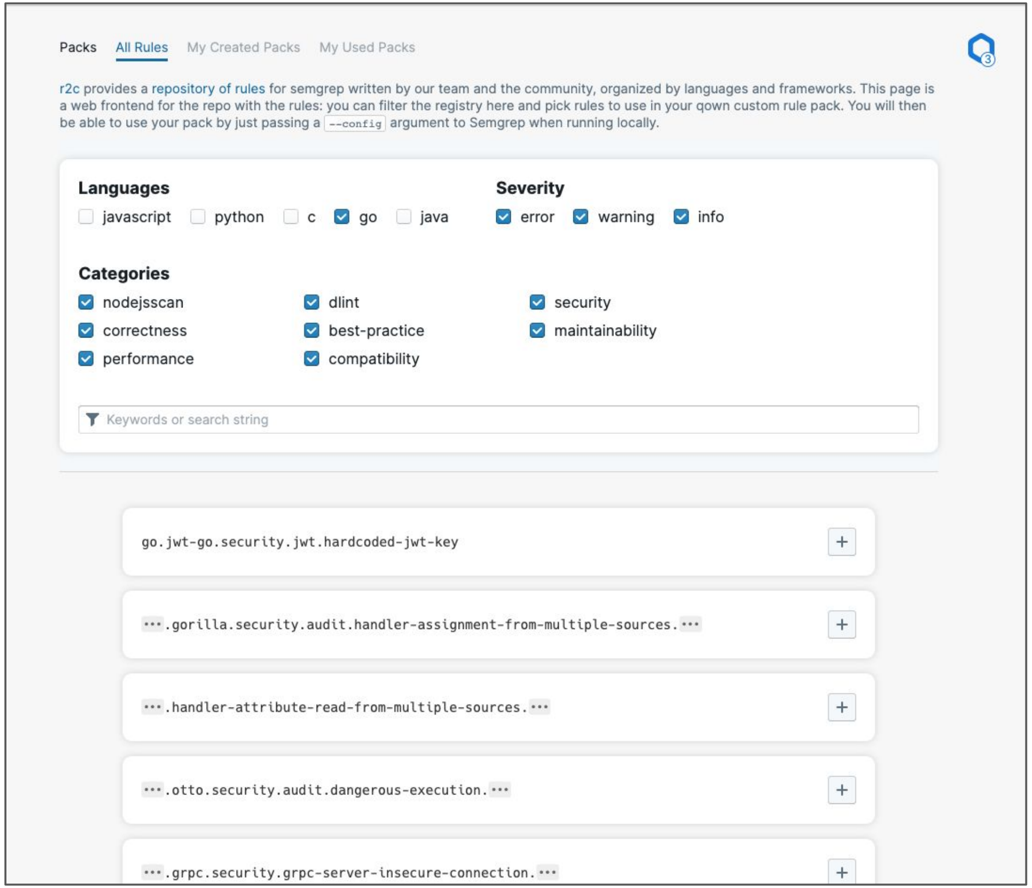
## Use of Weak RSA Key

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

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

⇒ https://semgrep.live/zdRl

**Full Solution:** https://semgrep.live/zdRl | **docs**

semgrep.live/**registry**



Packs All Rules My Created Packs My Used Packs

r2c provides a repository of rules for semgrep written by our team and the community, organized by languages and frameworks. This page is a web frontend for the repo with the rules: you can filter the registry here and pick rules to use in your qown custom rule pack. You will then be able to use your pack by just passing a `--config` argument to Semgrep when running locally.

**Languages**
☐ javascript ☐ python ☐ c ✓ go ☐ java

**Severity**
✓ error ✓ warning ✓ info

**Categories**

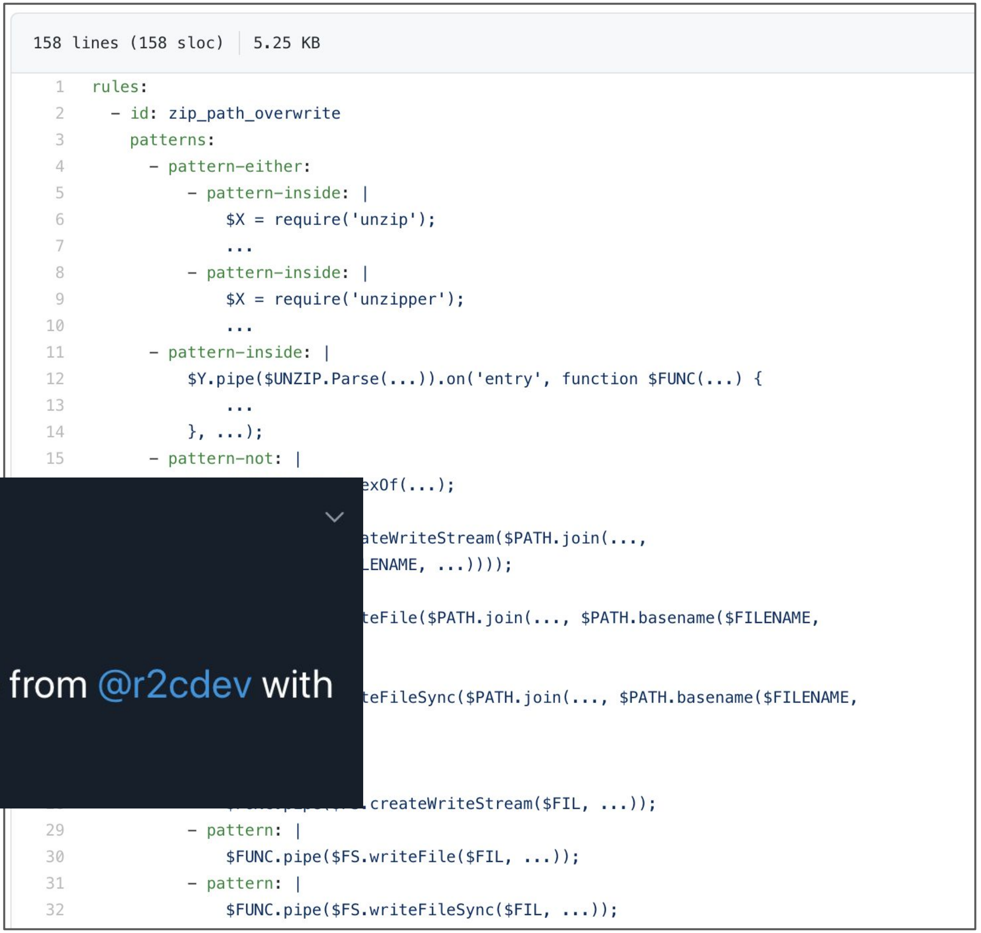
| | | |
|---|---|---|
| ✓ nodejsscan | ✓ dlint | ✓ security |
| ✓ correctness | ✓ best-practice | ✓ maintainability |
| ✓ performance | ✓ compatibility | |

Keywords or search string

```
go.jwt-go.security.jwt.hardcoded-jwt-key
•••.gorilla.security.audit.handler-assignment-from-multiple-sources.•••
•••.handler-attribute-read-from-multiple-sources.•••
•••.otto.security.audit.dangerous-execution.•••
•••.grpc.security.grpc-server-insecure-connection.•••
```

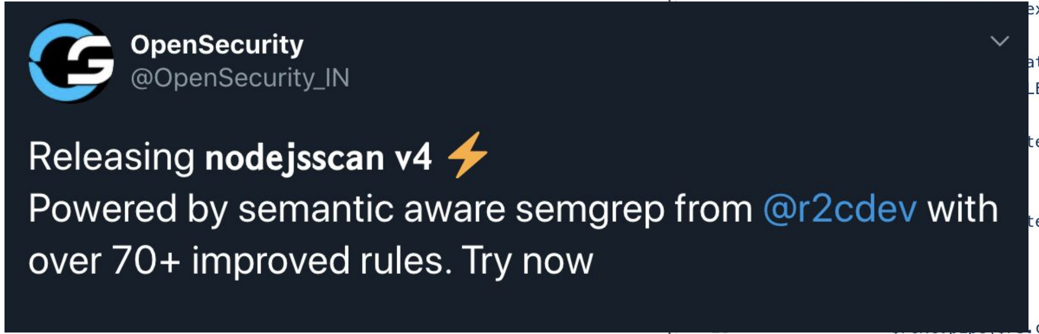
# NodeJsScan

# v4+ powered by Semgrep



158 lines (158 sloc) | 5.25 KB

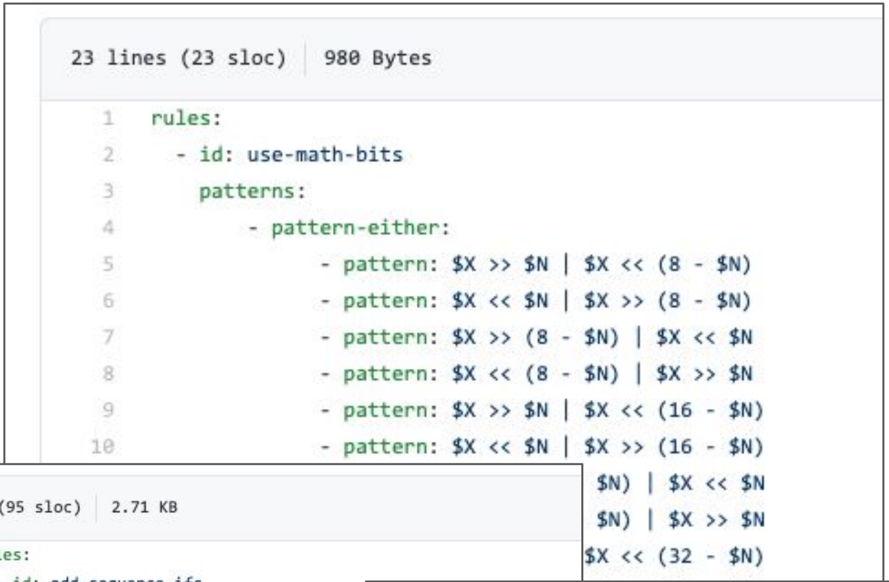
```
rules:
  - id: zip_path_overwrite
    patterns:
      - pattern-either:
          - pattern-inside: |
              $X = require('unzip');
              ...
          - pattern-inside: |
              $X = require('unzipper');
              ...
      - pattern-inside: |
          $Y.pipe($UNZIP.Parse(...)).on('entry', function $FUNC(...) {
              ...
          }, ...);
      - pattern-not: |
                          exOf(...);

                          ateWriteStream($PATH.join(...,
                          ENAME, ...))));

                          teFile($PATH.join(..., $PATH.basename($FILENAME,

                          teFileSync($PATH.join(..., $PATH.basename($FILENAME,

                          createWriteStream($FIL, ...));
          - pattern: |
              $FUNC.pipe($FS.writeFile($FIL, ...));
          - pattern: |
              $FUNC.pipe($FS.writeFileSync($FIL, ...));
```



OpenSecurity
@OpenSecurity_IN

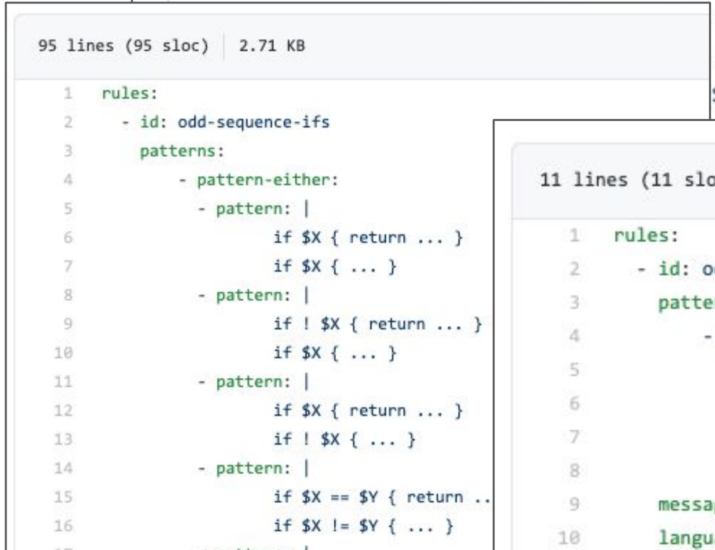
Releasing **nodejsscan v4** ⚡
Powered by semantic aware semgrep from @r2cdev with over 70+ improved rules. Try now

semgrep rules by Damian Gryski, author of **Go-Perfbook**



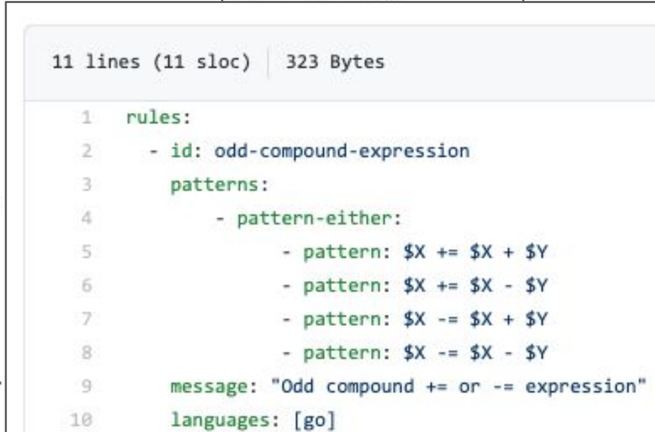
23 lines (23 sloc) | 980 Bytes

```
rules:
  - id: use-math-bits
    patterns:
      - pattern-either:
          - pattern: $X >> $N | $X << (8 - $N)
          - pattern: $X << $N | $X >> (8 - $N)
          - pattern: $X >> (8 - $N) | $X << $N
          - pattern: $X << (8 - $N) | $X >> $N
          - pattern: $X >> $N | $X << (16 - $N)
          - pattern: $X << $N | $X >> (16 - $N)
                                   $N) | $X << $N
                                   $N) | $X >> $N
                                  $X << (32 - $N)
```



95 lines (95 sloc) | 2.71 KB

```
rules:
  - id: odd-sequence-ifs
    patterns:
      - pattern-either:
          - pattern: |
              if $X { return ... }
              if $X { ... }
          - pattern: |
              if ! $X { return ... }
              if $X { ... }
          - pattern: |
              if $X { return ... }
              if ! $X { ... }
          - pattern: |
              if $X == $Y { return ..
              if $X != $Y { ... }
```



11 lines (11 sloc) | 323 Bytes

```
rules:
  - id: odd-compound-expression
    patterns:
      - pattern-either:
          - pattern: $X += $X + $Y
          - pattern: $X += $X - $Y
          - pattern: $X -= $X + $Y
          - pattern: $X -= $X - $Y
    message: "Odd compound += or -= expression"
    languages: [go]
```

# 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

**Watch your code**

- See fixes over time

# Semgrep

lightweight static analysis for many languages

1. semgrep.live/learn
2. `$ (brew or pip) install semgrep`
3. `$ semgrep --config=r2c .`
4. https://r2c.dev/survey pls :)

Drew Dennison | r2c.dev

@r2cdev @drewdennison