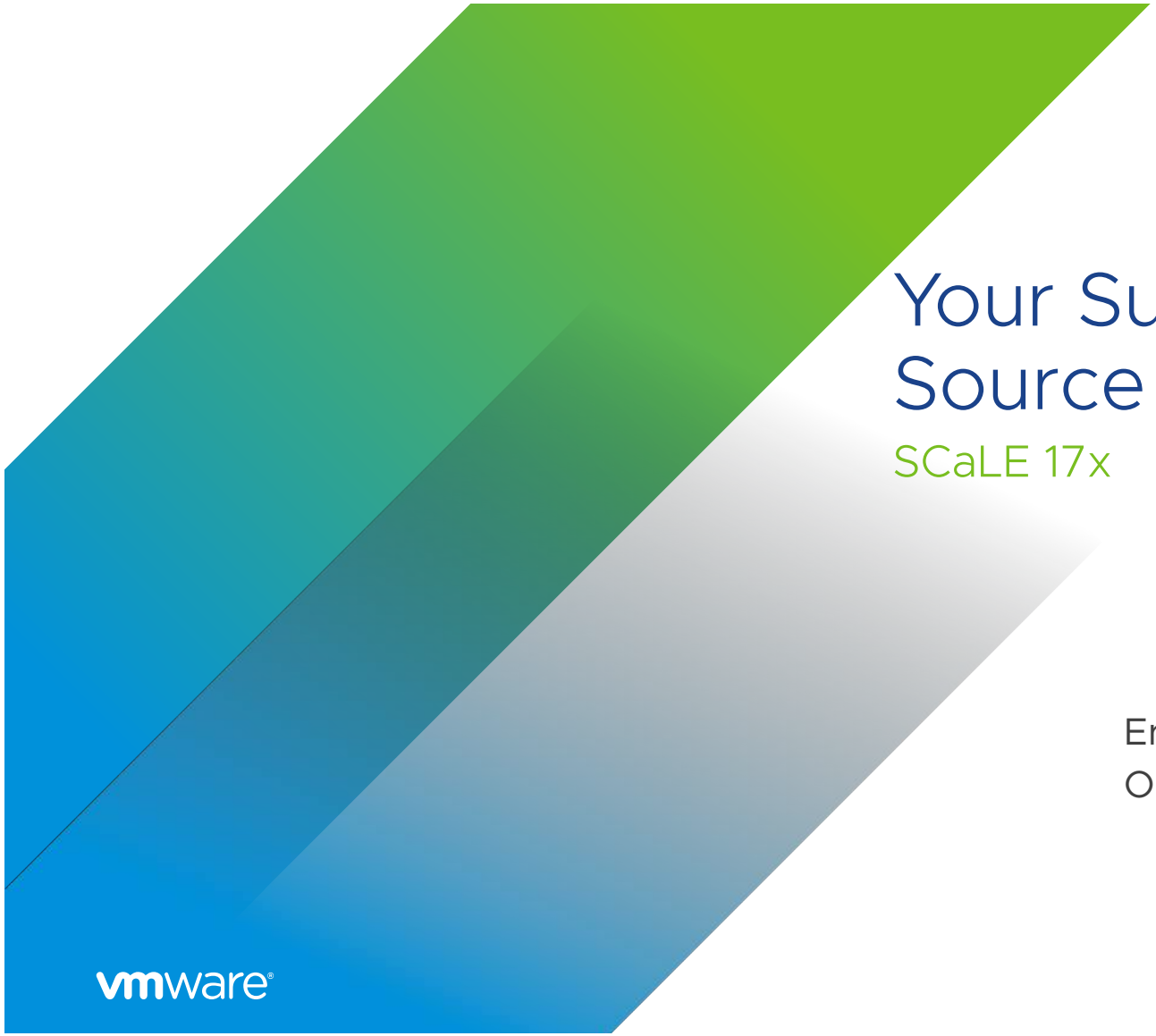


Your Suite of Open Source Security Linters

SCaLE 17x

Eric Brown
Open Source Engineering Manager

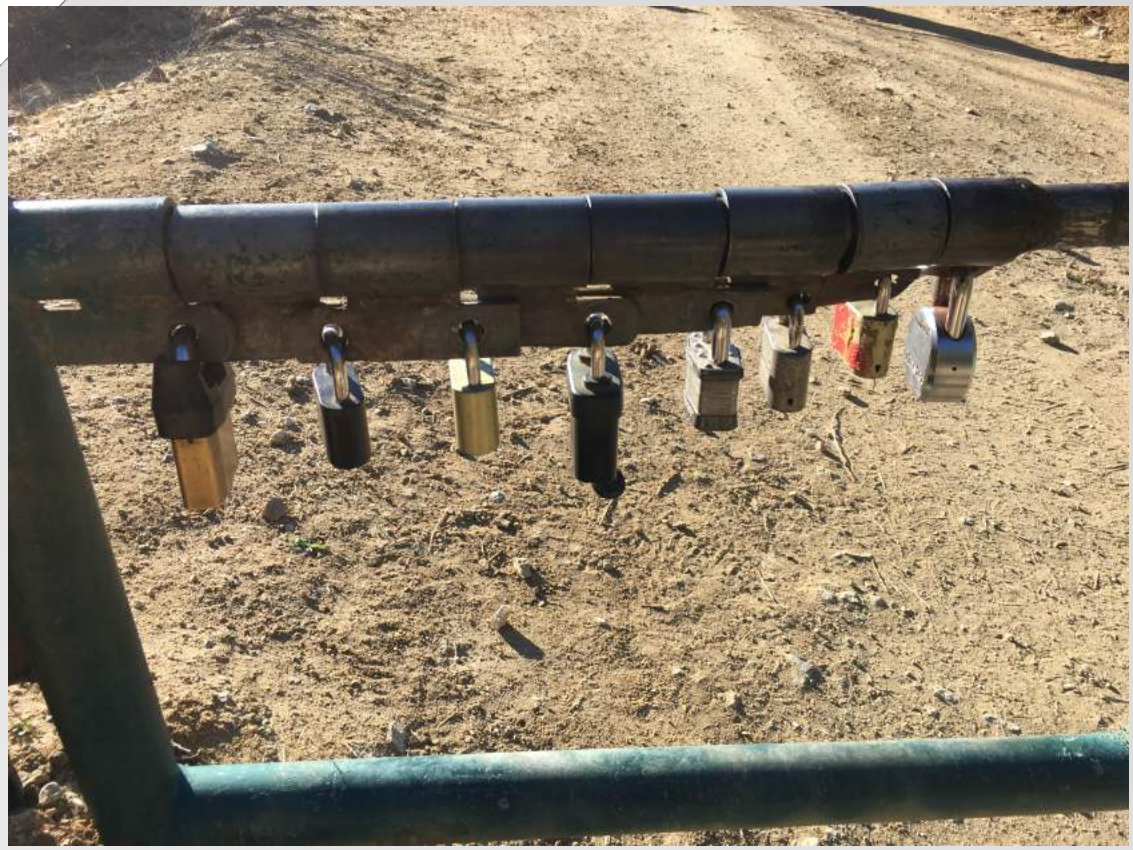
March 10th, 2019

vmware

Agenda

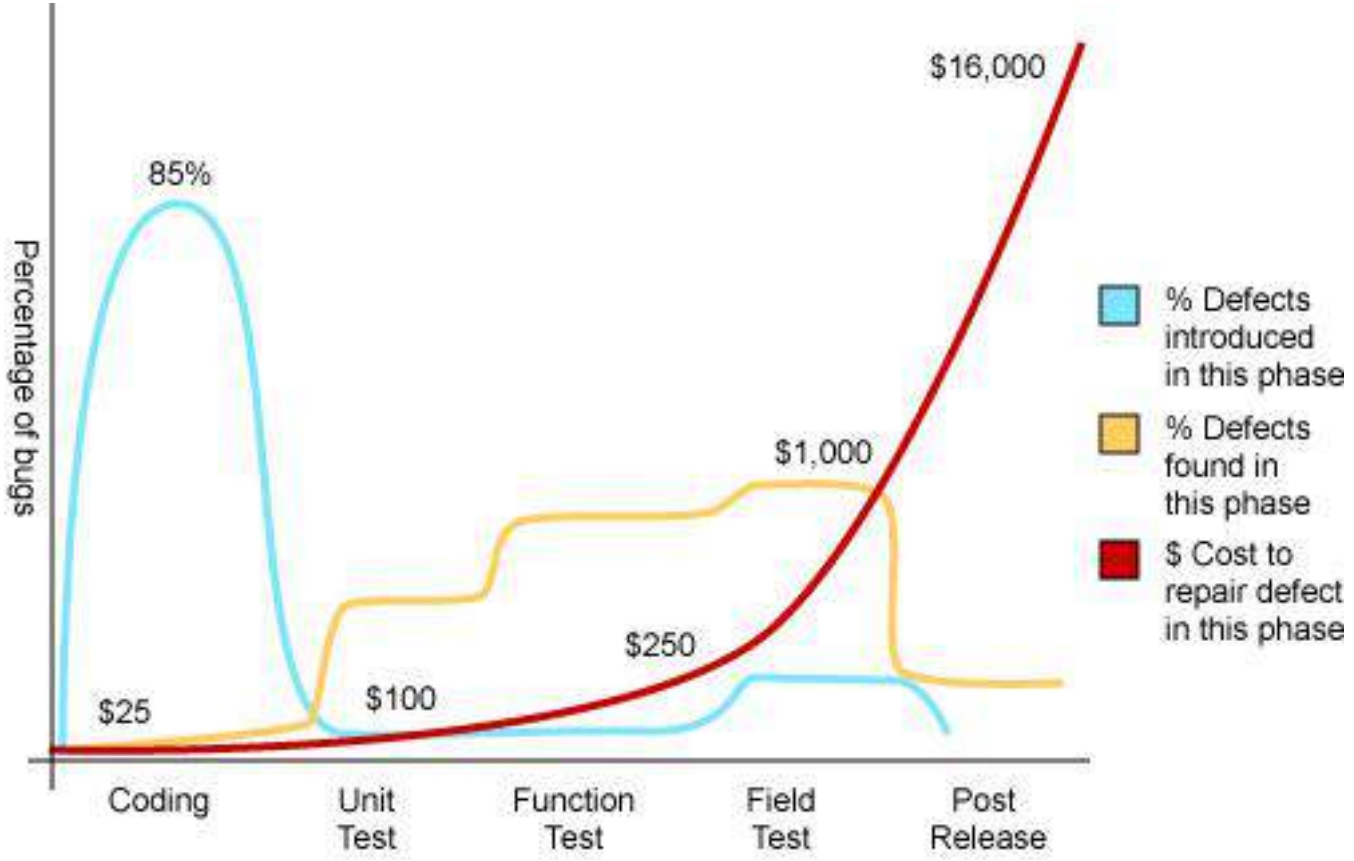
- Why is this important?
- What is static code analysis?
- Tools available
  - Python
  - Golang
  - JavaScript
  - Others
  - GitHub

Moar Security



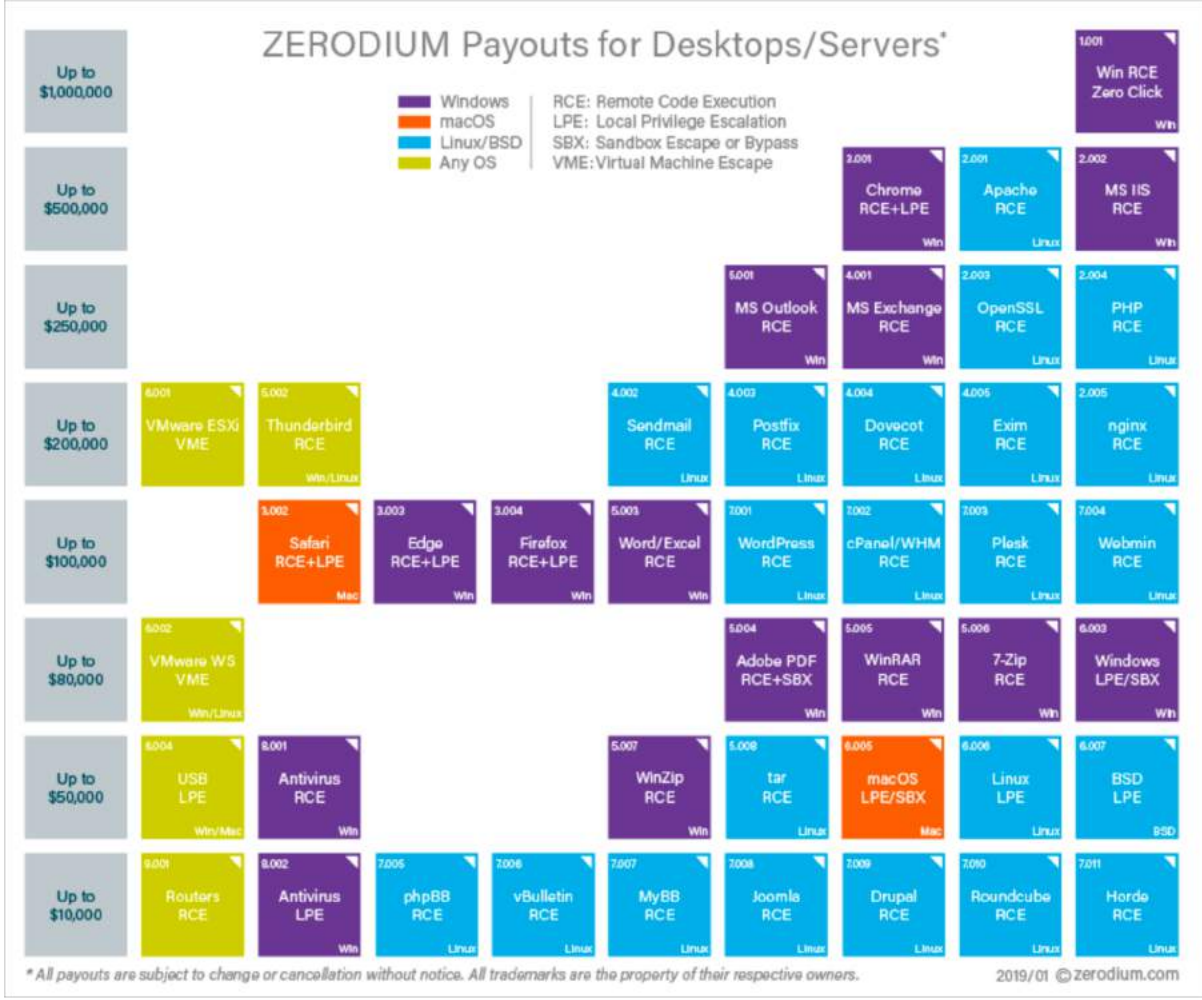
vmware®

Cost to Fix Bugs

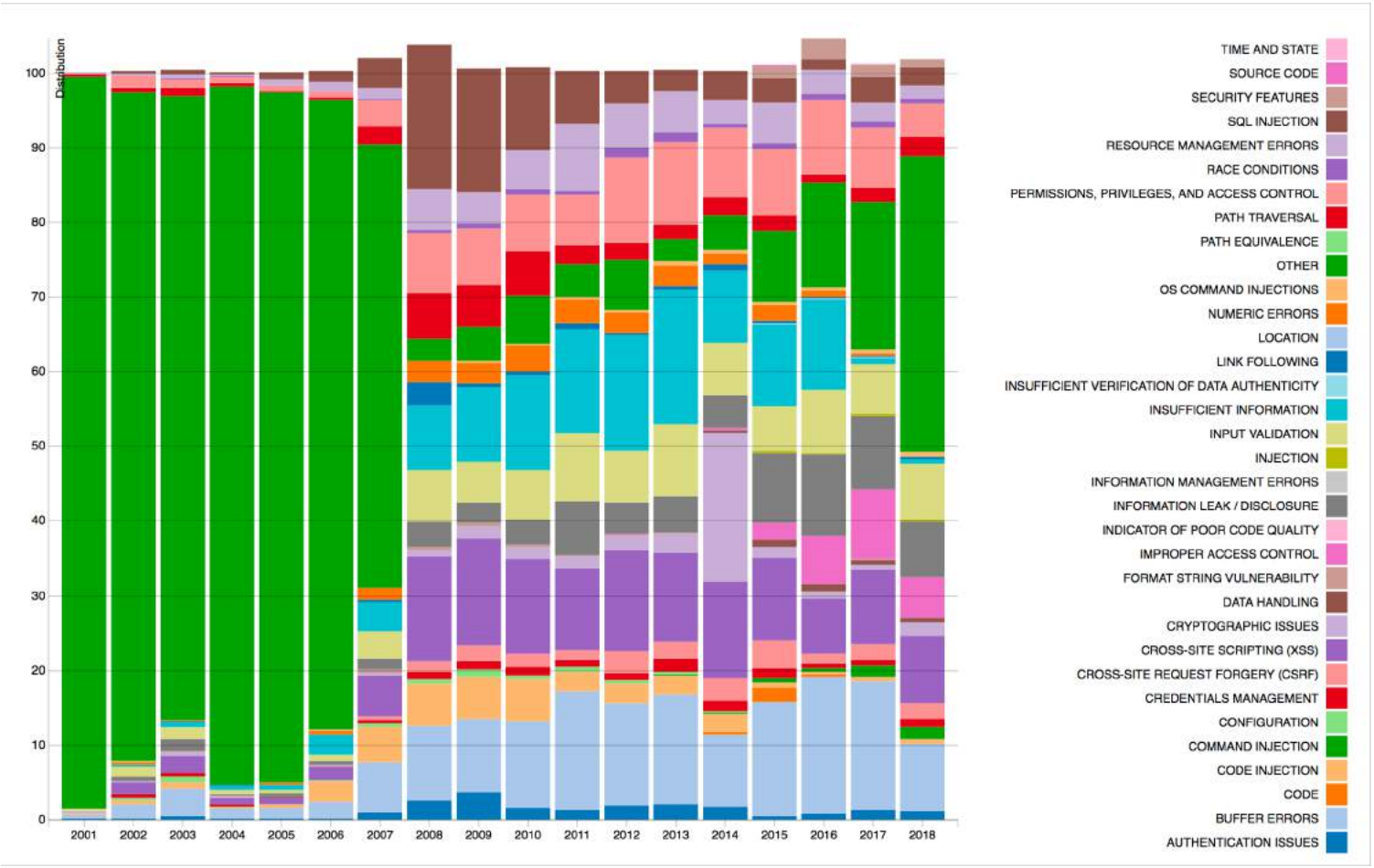


Source: Applied Software Measurement, Capers Jones, 1996

Business Impact $$$



Types of Security Bugs



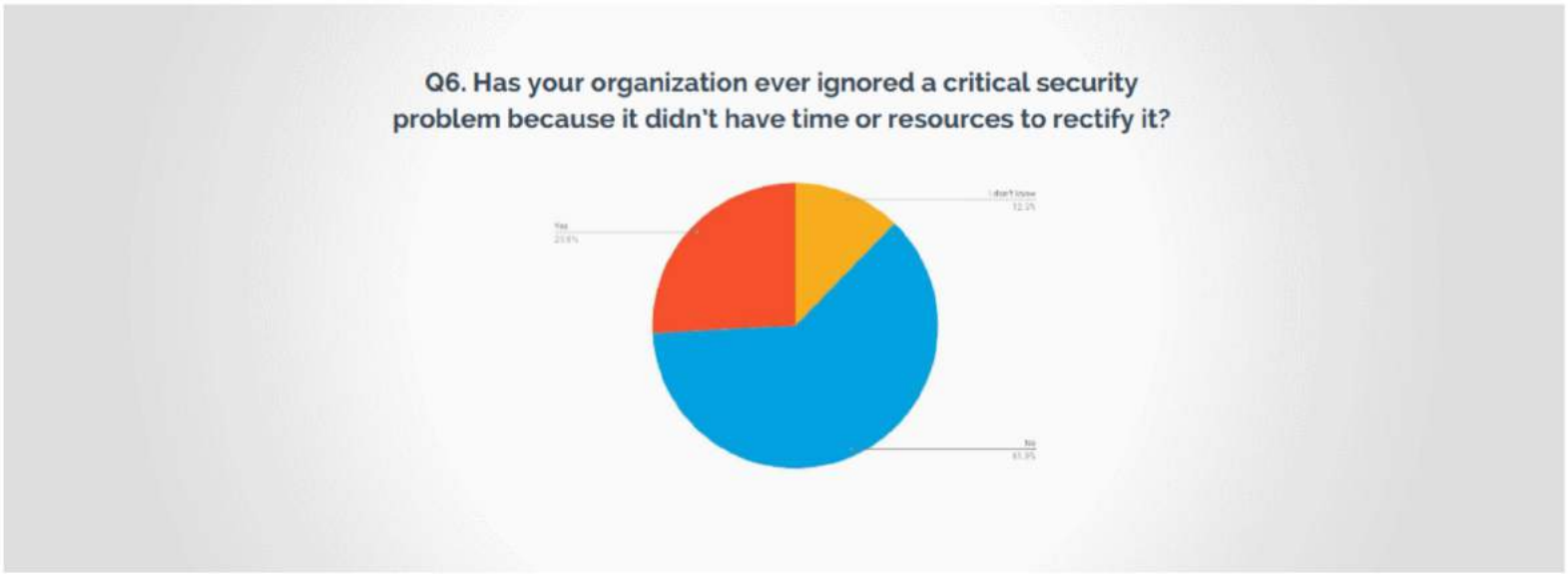
Source: National Institute of Standards and Technology

What's the Struggle?

26% of Companies Ignore Security Bugs Because They Don't Have the Time to Fix Them

By Catalin Cimpanu

May 10, 2018 10:00 AM 3



Source: https://www.bleepingcomputer.com/news/security/26-percent-of-companies-ignore-security-bugs-because-they-don-t-have-the-time-to-fix-them/

How to Catch Security Bugs Earlier

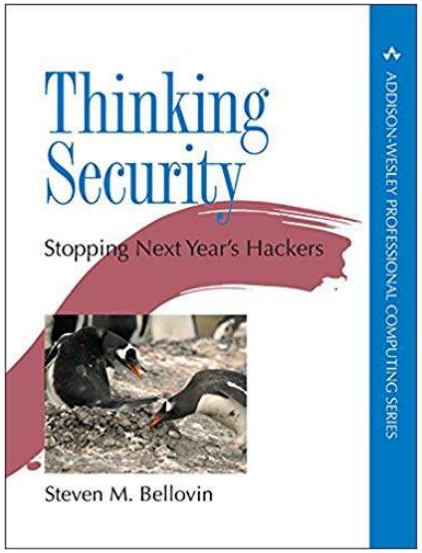
Education

How to Catch Security Bugs Earlier

Education

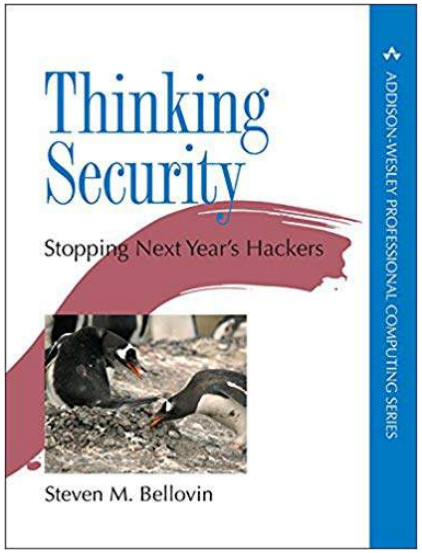
How to Catch Security Bugs Earlier

Education



How to Catch Security Bugs Earlier

Education



vmware

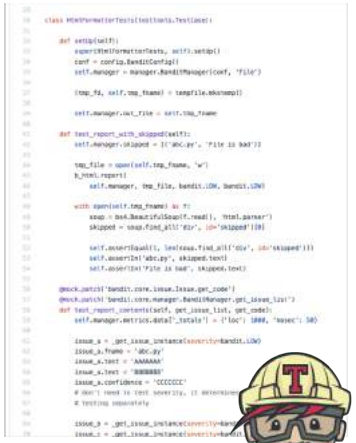
How to Catch Security Bugs Earlier

Tooling

How to Catch Security Bugs Earlier

Tooling

- Unit testing

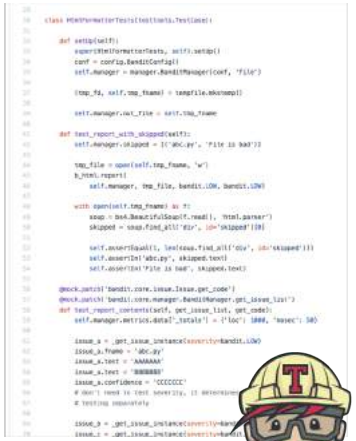


Travis CI

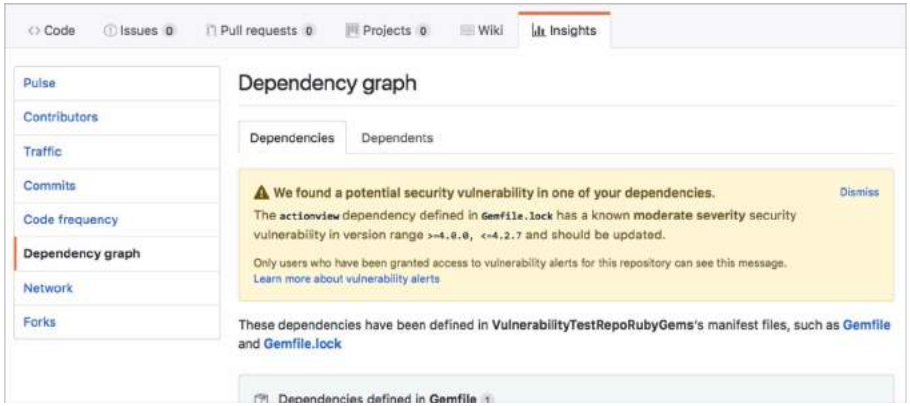
How to Catch Security Bugs Earlier

- Unit testing
- Dependency checking

Tooling



Travis CI

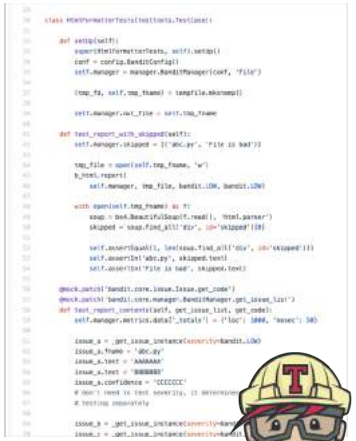


vmware

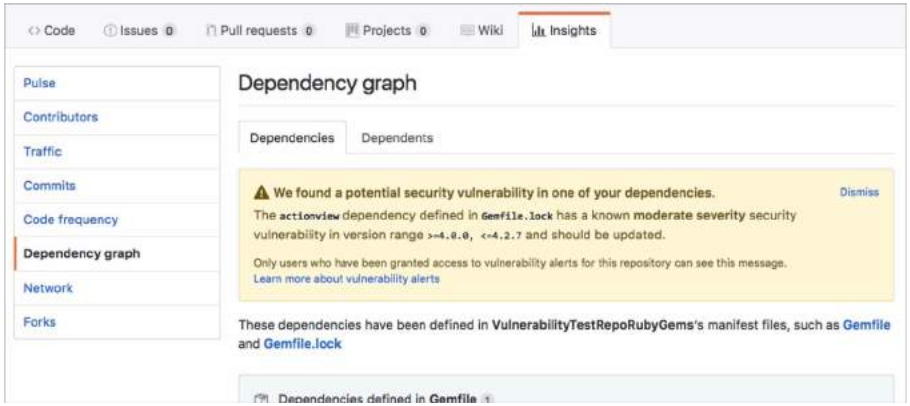
How to Catch Security Bugs Earlier

- Unit testing
- Dependency checking
- Static Code Analysis

Tooling



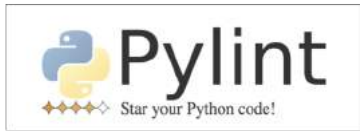
Travis CI



vmware

What is a Linter?

- Also known as Static Code Analysis tools
- Scans source code statically without running or building the code
- Linters focus on finding common mistakes
- Typically make use of a compiler's AST (abstract syntax tree)
- Popular examples:



When to Run Linters?

- When coding
  - Plugin to your favorite IDE
  - Fix as you code
- As part of a CI (continuous integration) system
  - Catch mistakes when code is pushed to a repository for review
  - Travis-CI, Circle CI, Jenkins, etc.

Downsides ☹️

- False positives
- Supplements, not replaces, human code review
- Extra time
  - Time to interpret the results
  - Time spent by CI to scan code

Bandit

Your Python Security Linter

vmware

What is Bandit?

- For Python
- Project started early 2015
- Originally designed for OpenStack but now part of the Python Code Quality Authority
- Python 2.7, 3.5, 3.6, 3.7 compatible
- Runs on Linux and macOS
- Easy to write new plugins
- Low resource requirements
- Runs quickly

Bandit: Issues It Finds

- Use of assert
- Hardcoded passwords
- Command injection
- Insecure temporary file usage
- Promiscuous file permissions
- Usage of unsafe functions/libraries
- Binding to all interfaces
- Weak cryptography
- Bad SSL versions
- Requests without certificate validation
- Use of insecure protocols

Bandit: Formatters

CSV

HTML

JSON

Text

XML

YAML

Custom

Bandit: Other Features

- Filtering on severity and confidence levels
- Allows creation of profiles to scan a subset of plugin tests
- Adjust lines of context shown in output
- Group results by file or vulnerability
- Output delta reports of previous scans
- Allows marking false positives using the "*# nosec"* comment

Bandit: Config

GitHub, Inc. (US) | https://github.com/Netflix-Skunkworks/historical/blob/master/tox.ini

```
59
60  [testenv:bandit]
61  basepython = python3
62  skip_install = true
63  deps =
64      bandit
65  commands =
66      bandit --ini tox.ini -r historical
67
68  [bandit]
69  skips = B101
```

Bandit: Config

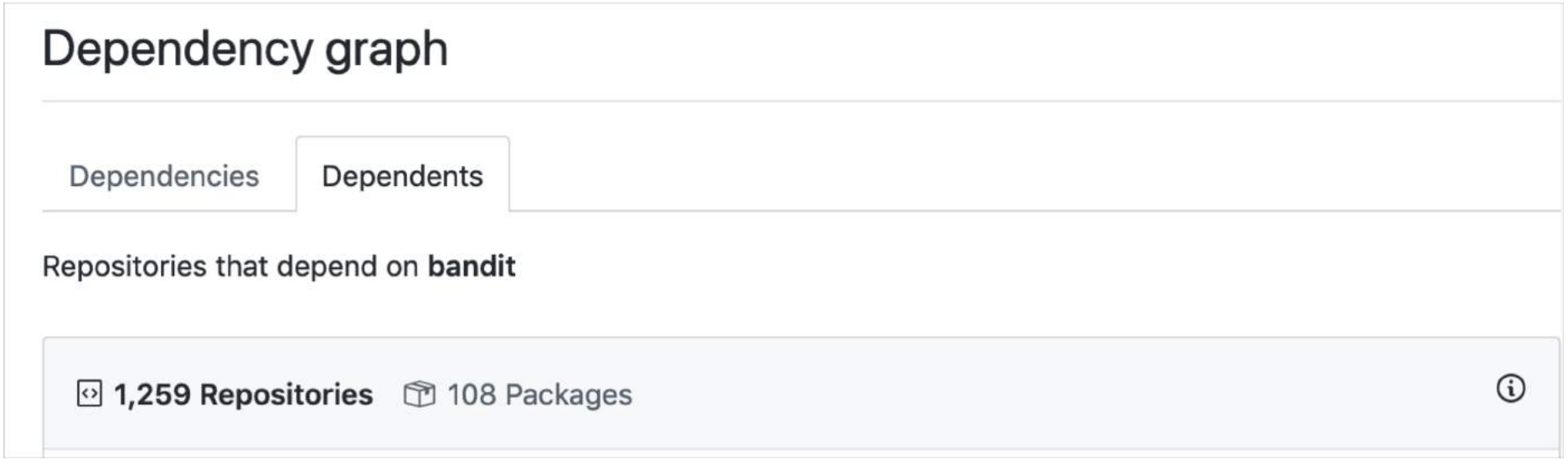
```
browne-a01:workspace browne$ bandit-config-generator -o bandit.yaml
[ INFO]: Successfully wrote profile: bandit.yaml
```

demo1.py | ssh_no_host_key_verification.py | bandit.yaml

```
348     - subprocess.run
349   subprocess_without_shell_equals_true:
350     no_shell:
351     - os.execl
352     - os.execle
353     - os.execlp
354     - os.execlpe
355     - os.execv
356     - os.execve
363     - os.spawnv
364     - os.spawnve
365     - os.spawnvp
366     - os.spawnvpe
367     - os.startfile
368     shell:
369     - os.system
370     - os.popen
371     - os.popen2
372     - os.popen3
373     - os.popen4
374     - popen2.popen2
375     - popen2.popen3
376     - popen2.popen4
377     - popen2.Popen3
378     - popen2.Popen4
379     - commands.getoutput
380     - commands.getstatusoutput
381     subprocess:
382     - subprocess.Popen
383     - subprocess.call
384     - subprocess.check_call
385     - subprocess.check_output
386     - subprocess.run
387   try_except_continue:
388     check_typed_exception: false
389   try_except_pass:
390     check_typed_exception: false
391   weak_cryptographic_key:
392     weak_key_size_dsa_high: 1024
393     weak_key_size_dsa_medium: 2048
394     weak_key_size_ec_high: 160
395     weak_key_size_ec_medium: 224
396     weak_key_size_rsa_high: 1024
397     weak_key_size_rsa_medium: 2048
```

vmware

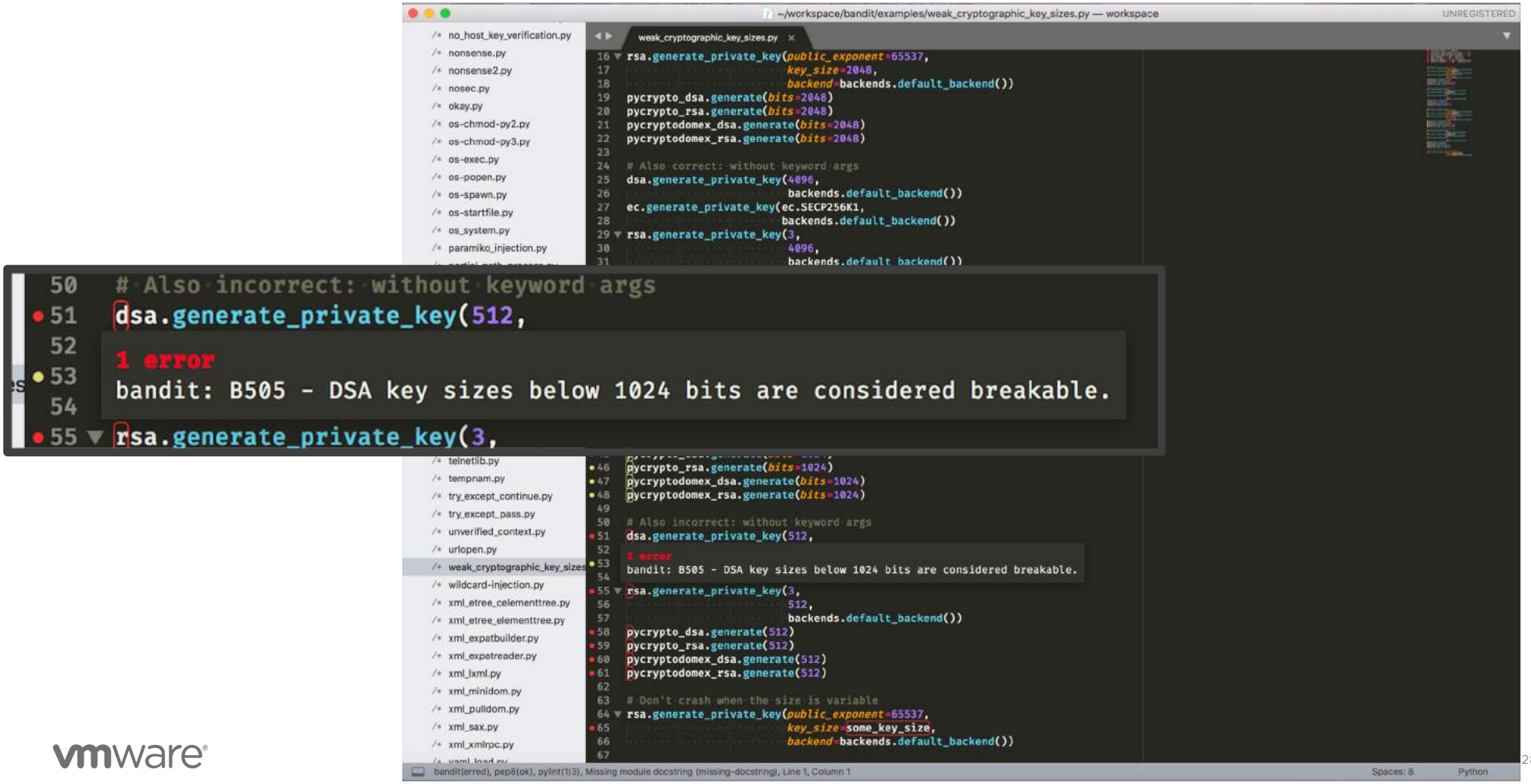
Bandit: Dependents



Bandit: Integrations

- SublimeLinter-bandit – Sublime Text linter plugin
- flake8-bandit – Flake8 plugin
- pyreportcard – Report card of Python projects quality
- bandit-plugin – Hudson/Jenkins plugin
- vscode-python – Plugin for Visual Studio Code * Added Feb 26th!

Bandit: Sublime Text Linter



Package Control

Search

cmd+shift+p

BROWSE

SublimeLinter-bandit

by SublimeLinter ST3

Only works with Sublime Text 3

SublimeLinter plugin for Python, using bandit

LABELS linting, SublimeLinter, python, security

Details

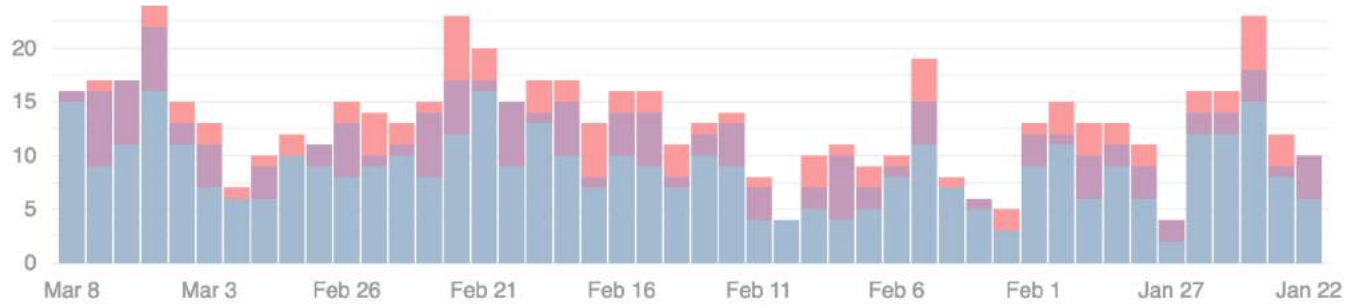
VERSION 1.1.1
HOMEPAGE github.com
ISSUES github.com

MODIFIED 12 months ago

LAST SEEN 1 hour ago
FIRST SEEN 2 years ago

Installs

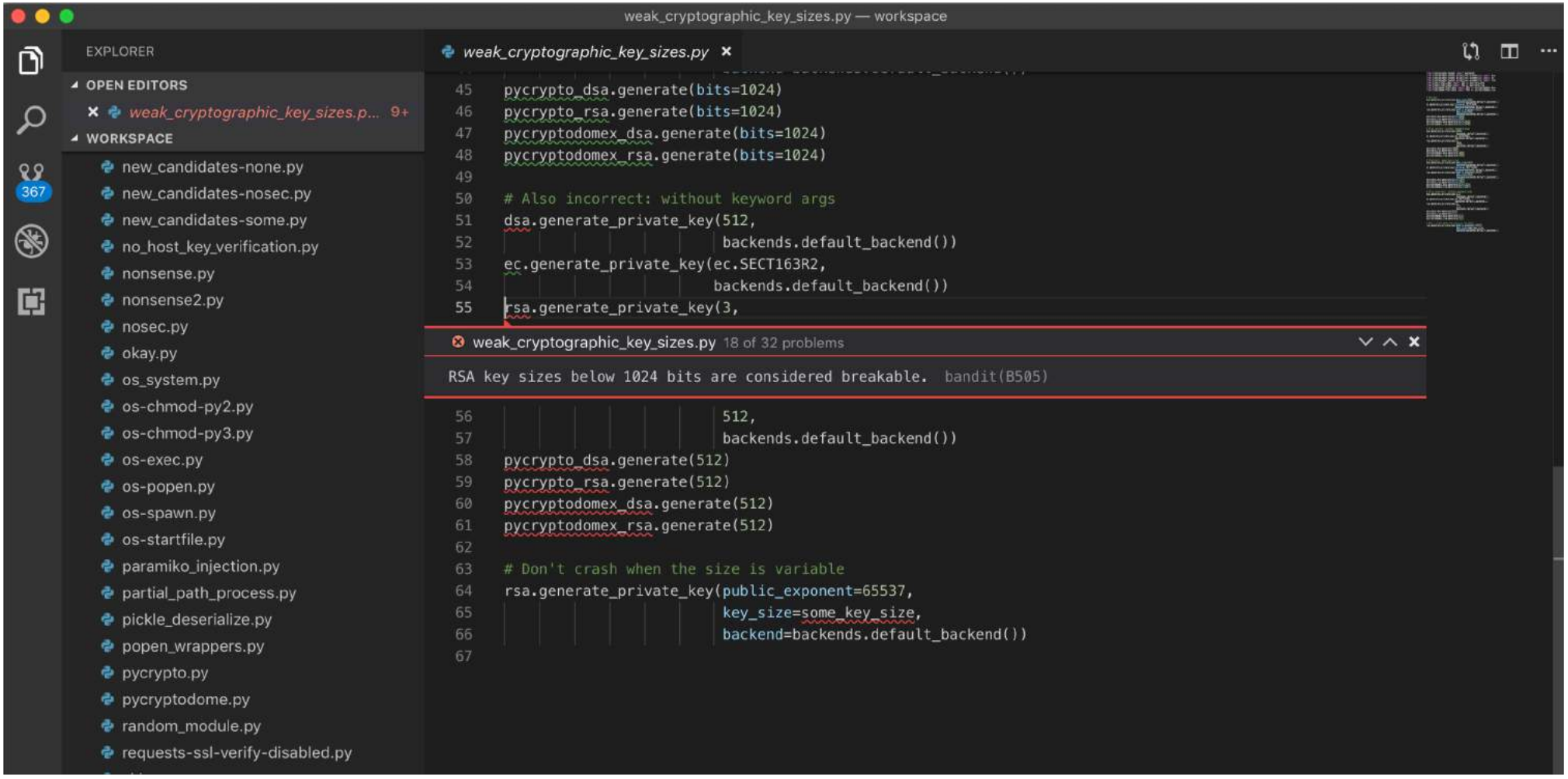
TOTAL 15K WIN 10K OS X 3K LINUX 2K



Readme

SOURCE raw.githubusercontent.com

Bandit: VSCode



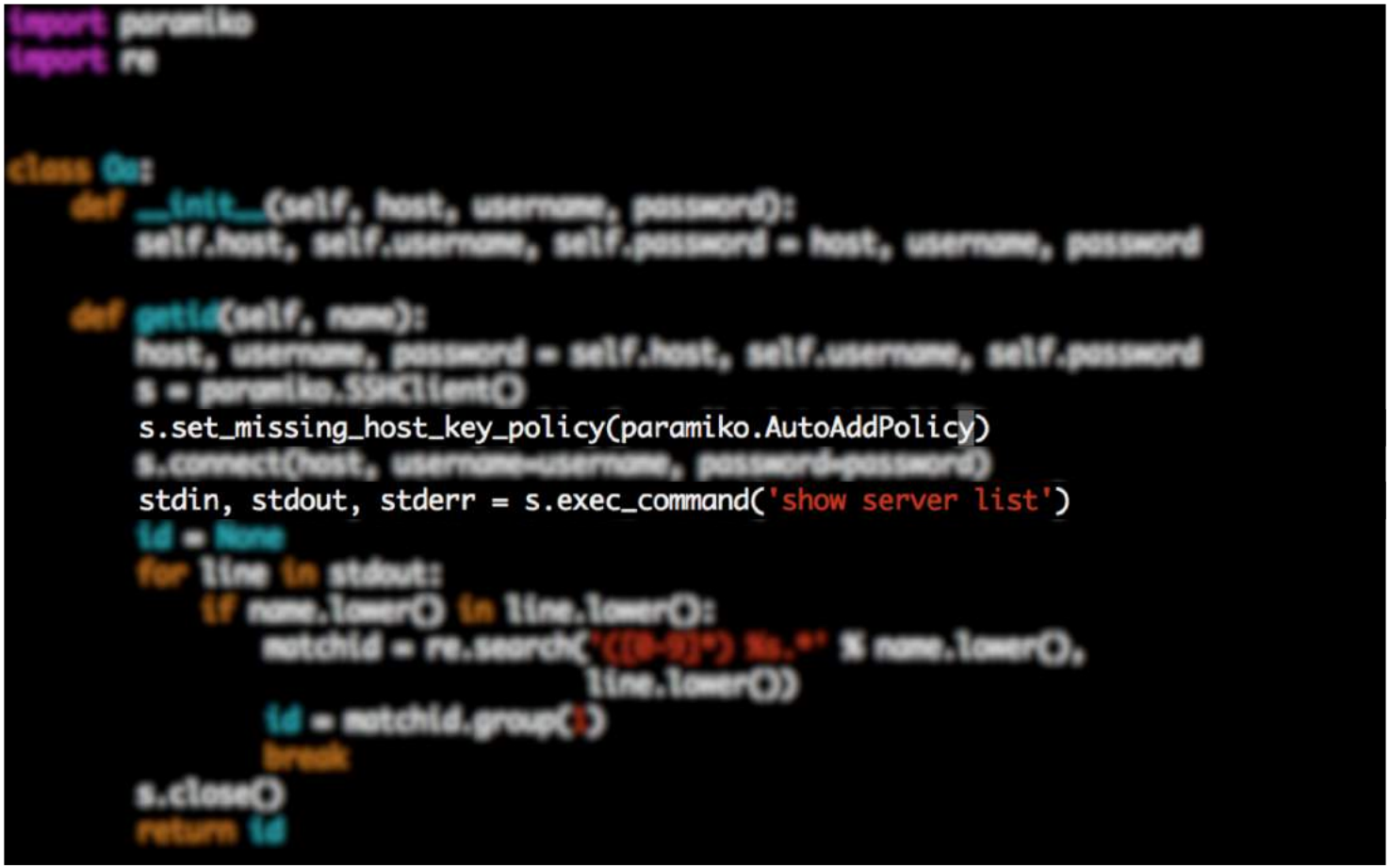
Bandit in Action

```
import paramiko
import re


class Oa:
    def __init__(self, host, username, password):
        self.host, self.username, self.password = host, username, password

    def getid(self, name):
        host, username, password = self.host, self.username, self.password
        s = paramiko.SSHClient()
        s.set_missing_host_key_policy(paramiko.AutoAddPolicy)
        s.connect(host, username=username, password=password)
        stdin, stdout, stderr = s.exec_command('show server list')
        id = None
        for line in stdout:
            if name.lower() in line.lower():
                matchid = re.search('([0-9]*) %s.*' % name.lower(),
                                    line.lower())
                id = matchid.group(1)
                break
        s.close()
        return id
```

Bandit in Action

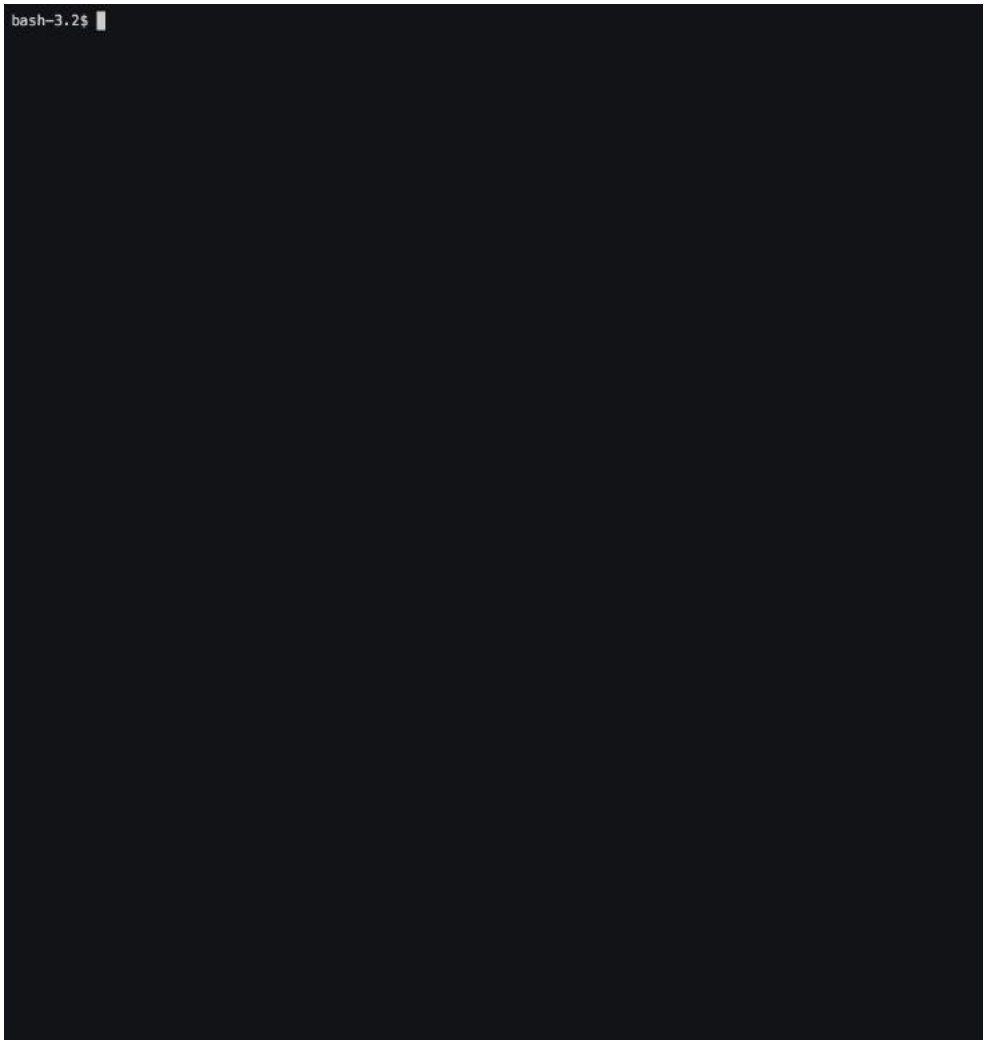


```
import paramiko
import re


class Os:
    def __init__(self, host, username, password):
        self.host, self.username, self.password = host, username, password

    def getid(self, name):
        host, username, password = self.host, self.username, self.password
        s = paramiko.SSHClient()
        s.set_missing_host_key_policy(paramiko.AutoAddPolicy)
        s.connect(host, username=username, password=password)
        stdin, stdout, stderr = s.exec_command('show server list')
        id = None
        for line in stdout:
            if name.lower() in line.lower():
                matchid = re.search("([0-9]*) %s.*" % name.lower(),
                                    line.lower())
                id = matchid.group(1)
                break
        s.close()
        return id
```

Bandit in Action



Bandit in Action

Contributing to Bandit

- https://github.com/PyCQA/bandit
- Core maintainers:
  - Myself – ericwb
  - Ian StapleTon Cordasco – sigmavirus24
  - Gage Hugo – ghugo
  - Luke Hinds – lukehinds
- Ideas:
  - Documentation could be improved
  - Integrations with more IDEs
  - Fix suggestions
    - Quick Fix - VSCode
    - Suggested changes - GitHub

Gosec

Your Golang Security Linter

vmware

What is Gosec?

- For Golang
- Project started mid 2016
- Started by one of the creators of Bandit
- Runs on Linux, macOS, and Windows
- Low resource requirements
- Runs quickly
- Pluggable

Gosec: Issues It Finds

- Hardcoded credentials
- Binding to all interfaces
- SQL injection
- Command injection
- Insecure temporary file usage
- Promiscuous file permissions
- Usage of unsafe functions/libraries
- Weak cryptography
- Bad TLS/SSL versions
- Ignoring host keys

Gosec: Formatters

CSV

HTML

JSON

Text

XML

YAML

Gosec: Other Features

- Filtering on severity levels
- Allows creation of profiles to scan a subset of plugin tests
- Group results by severity
- Allows marking false positives using the "*# nosec"* comment
- Tool to generate TLS rules according to Mozilla recommendations
  - https://statics.tls.security.mozilla.org/server-side-tls-conf.json
- Gotchas:
  - Only scans projects in your $GOPATH

Gosec: Integrations

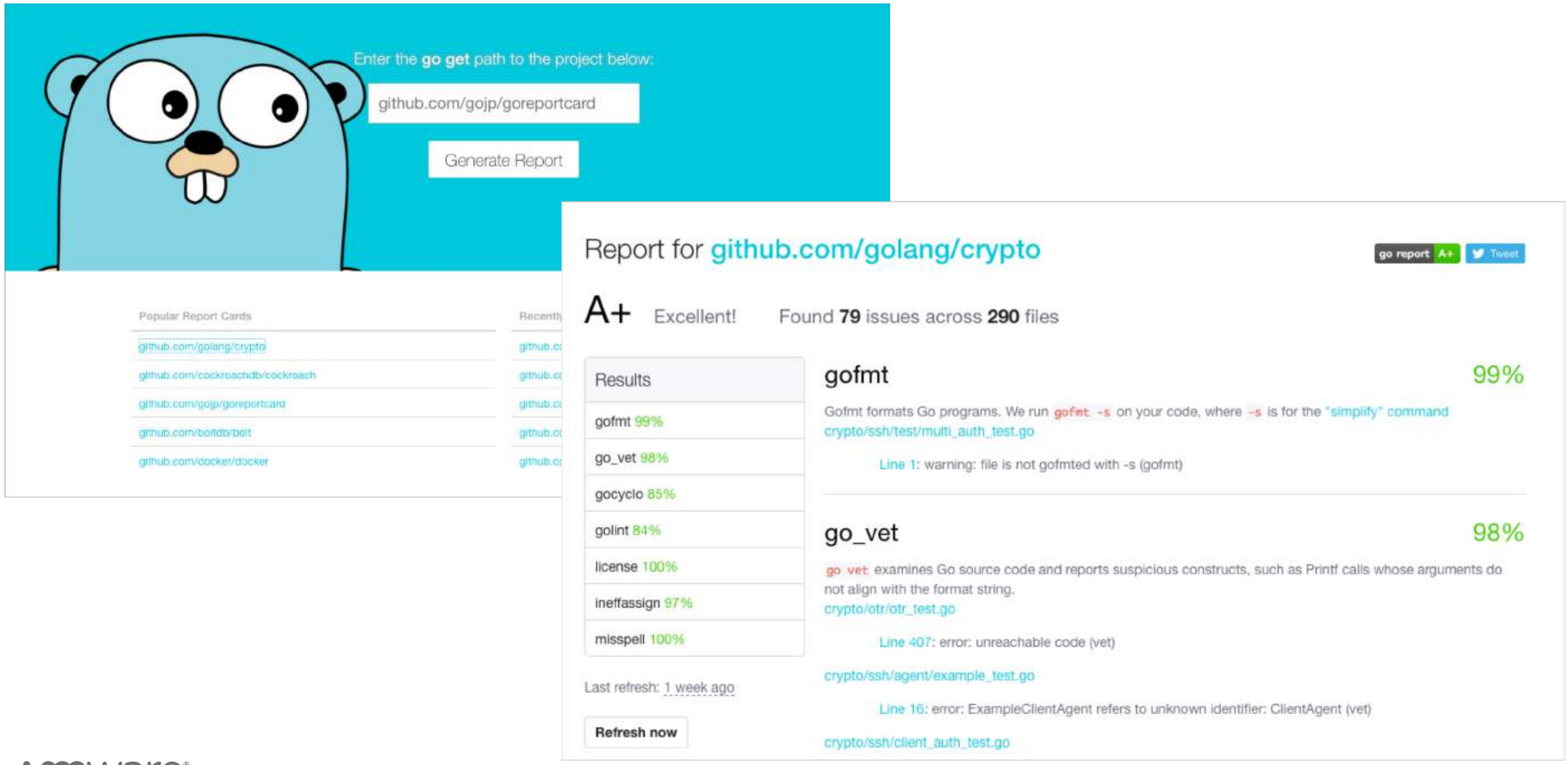
- SublimeLinter-contrib-gometallinter – Sublime Text plugin
- SublimeLinter-golangcilint – Sublime Text plugin
- gometallinter – collection of linters
- golangcilint – collection of linters
- 

– via gometallinter or golangci-lint plugin
- go-plus – Atom via golangci-lint plugin

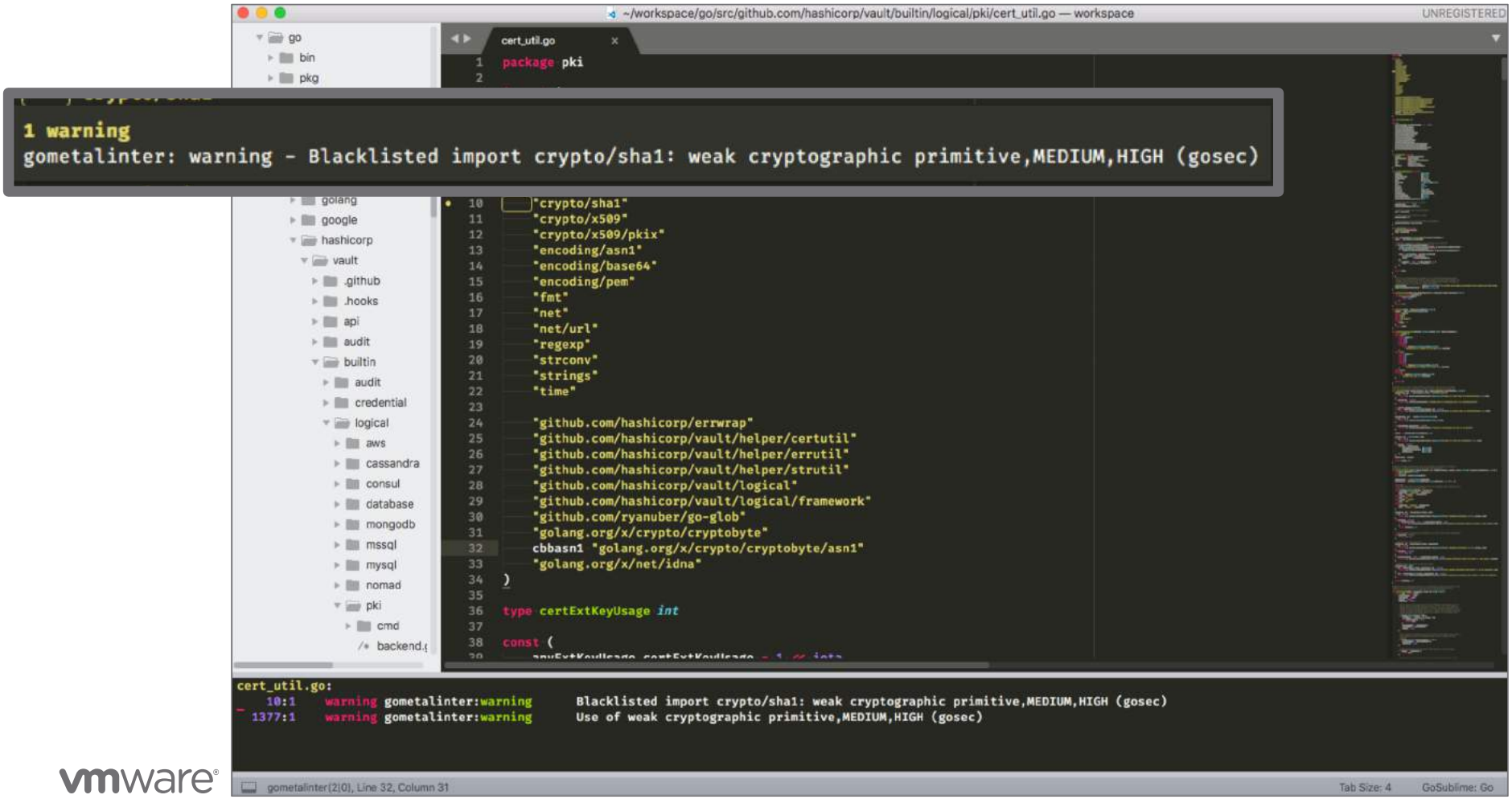
Wishlist:

- Go Report Card

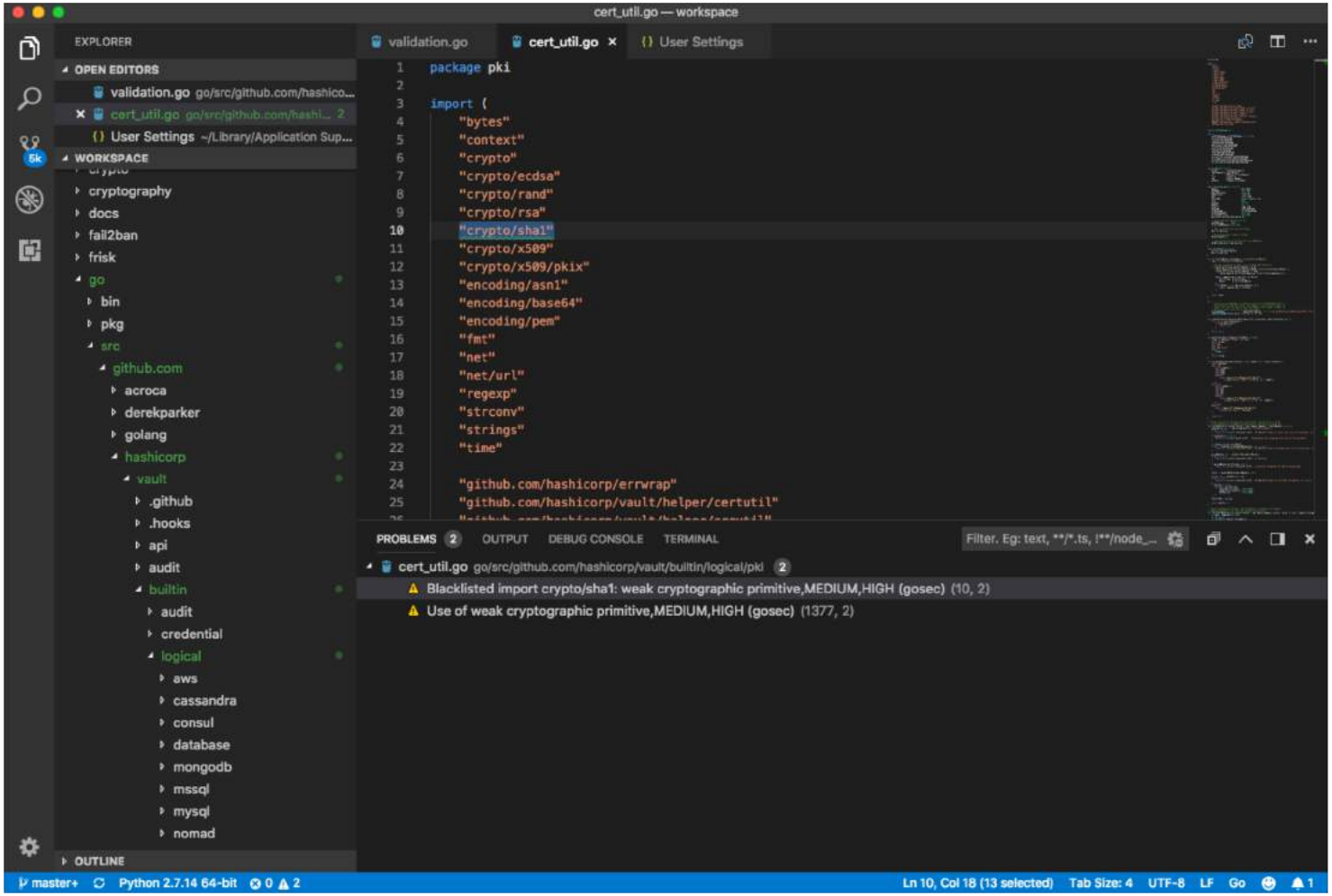
Gosec: Go Report Card



Gosec: Sublime Text Linter



Gosec: Visual Studio Code



Gosec in Action

```
package main

import (
    "crypto/rand"
    "crypto/rsa"
    "fmt"
)

func main() {
    reader := rand.Reader
    key, _ := rsa.GenerateKey(reader, 512)
    publicKey := key.PublicKey
    fmt.Println("Public key: ", publicKey)
}
```

Gosec in Action

Gosec in Action

Gosec: TLS rules

```
browne-a01:go browne$ bin/tlsconfig
browne-a01:go browne$
```

```
browne-a01:ericwb browne$ go generate ./...
browne-a01:ericwb browne$
```

```go
package rules

import (
    "go/ast"

    "github.com/securego/gosec"
)

// NewModernTLSCheck creates a check for Modern TLS ciphers
// DO NOT EDIT - generated by tlsconfig tool
func NewModernTLSCheck(id string, conf gosec.Config) (gosec.Rule, []ast.Node) {
    return &insecureConfigTLS{
        MetaData:     gosec.MetaData{ID: id},
        requiredType: "crypto/tls.Config",
        MinVersion:   0x0303,
        MaxVersion:   0x0303,
        goodCiphers: []string{
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305",
            "TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305",
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
        },
    }, []ast.Node{(*ast.CompositeLit)(nil)}
}

// NewIntermediateTLSCheck creates a check for Intermediate TLS ciphers
// DO NOT EDIT - generated by tlsconfig tool
func NewIntermediateTLSCheck(id string, conf gosec.Config) (gosec.Rule, []ast.Node) {
    return &insecureConfigTLS{
        MetaData:     gosec.MetaData{ID: id},
        requiredType: "crypto/tls.Config",
        MinVersion:   0x0301,
        MaxVersion:   0x0303,
        goodCiphers: []string{
            "TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305",
            "TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305",
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_DHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_DHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384",
```

Gosec: Contributing

- https://github.com/securego/gosec
- Ideas:
  - Documentation could use some work
  - More integrations with IDEs

eslint-plugin-security

Your JavaScript Security Linter

vmware

What is ESLint-Security-Plugin?

- A security plugin for ESLint
- Project started mid 2015
- Runs on Linux, macOS, and Windows
- Ignore false positives with `// eslint-disable-line`
- Configure via .eslintrc file
- Supports auto-fixing

eslint-plugin-security: Formatters

- So many!
  - Checkstyle
  - Codeframe
  - Compact
  - HTML
  - Jslint-xml
  - JSON
  - Junit
  - Stylish (default)
  - Table
  - Tap
  - Unix
  - visualstudio

eslint-plugin-security: Issues It Finds

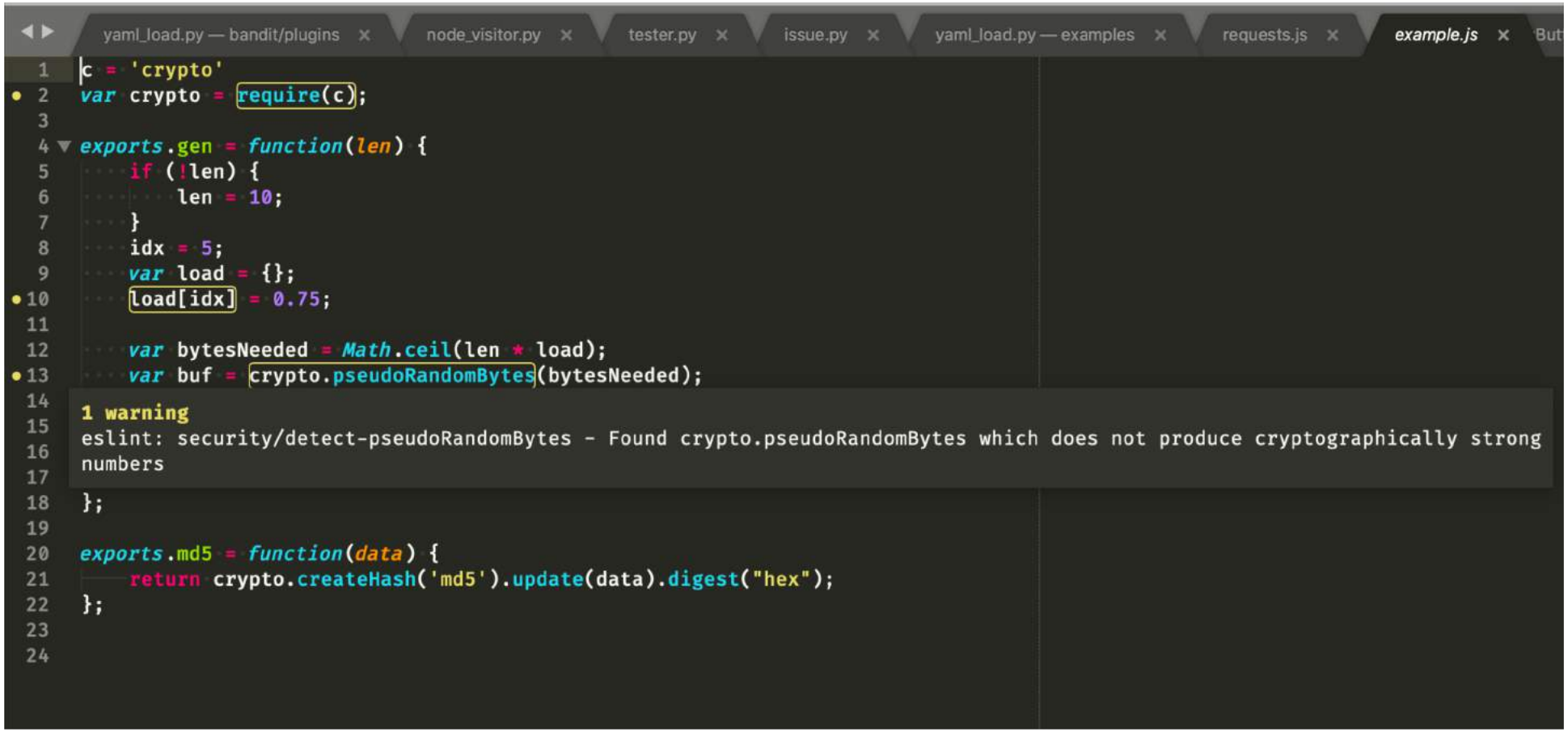
- Unsafe regex
- Buffer with `noAssert` flag
- Instances of `child_process` and `exec()`
- Disabled escape markup
- `eval()` with an expression
- CSRF middleware setup before method-override
- Variable in filename argument of `fs`
- Use of `RegExp(variable)`
- Use of `require(variable)`
- Object injection
- Possible timing attacks
- Use of `pseudoRandomBytes()`

eslint-plugin-security : Integrations

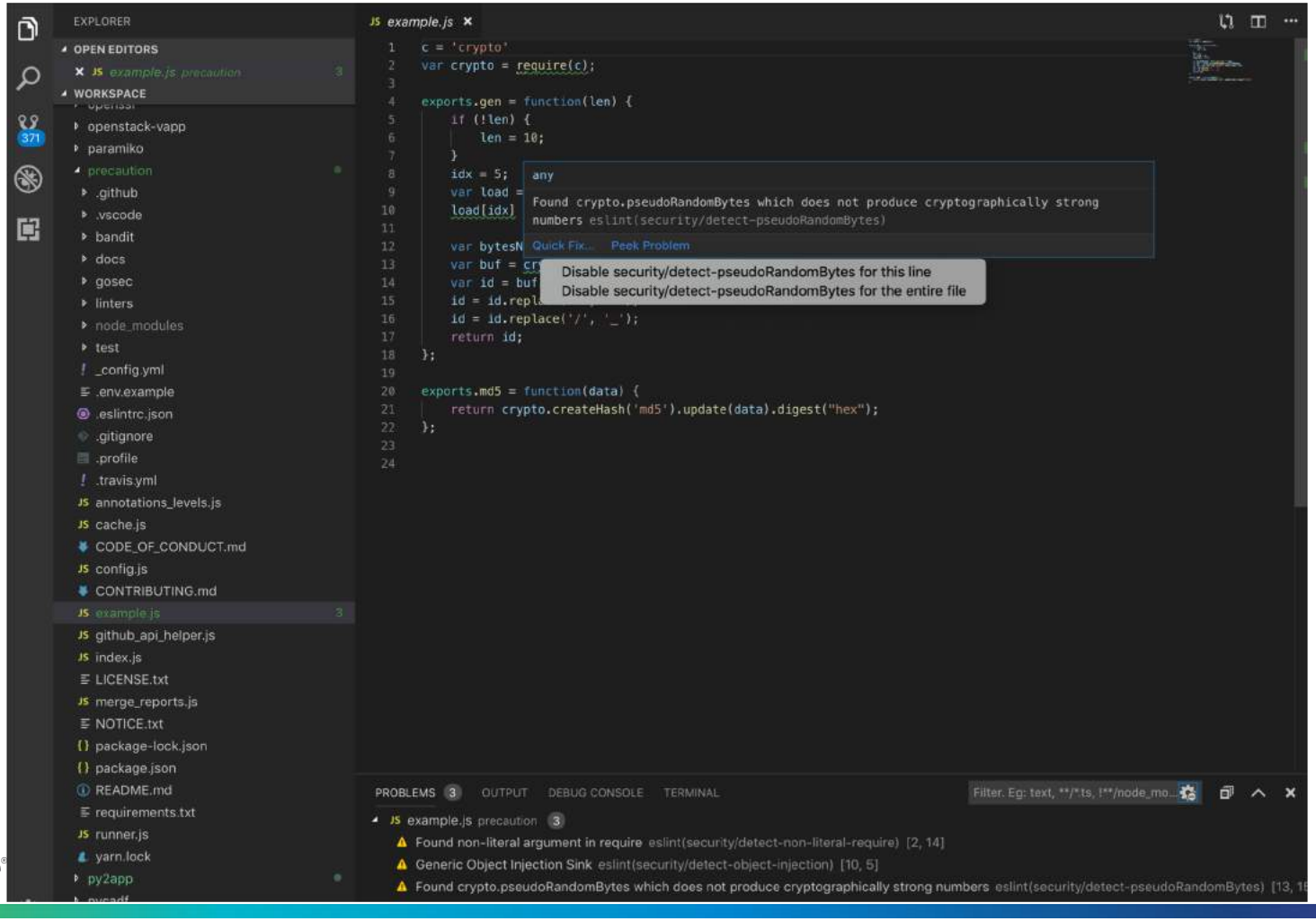
- SublimeLinter-eslint - Sublime Text plugin
- vscode-eslint – 

- linter-eslint – Atom plugin

eslint-plugin-security: Sublime Text Linter



eslint-plugin-security: VSCode



eslint-plugin-security in Action

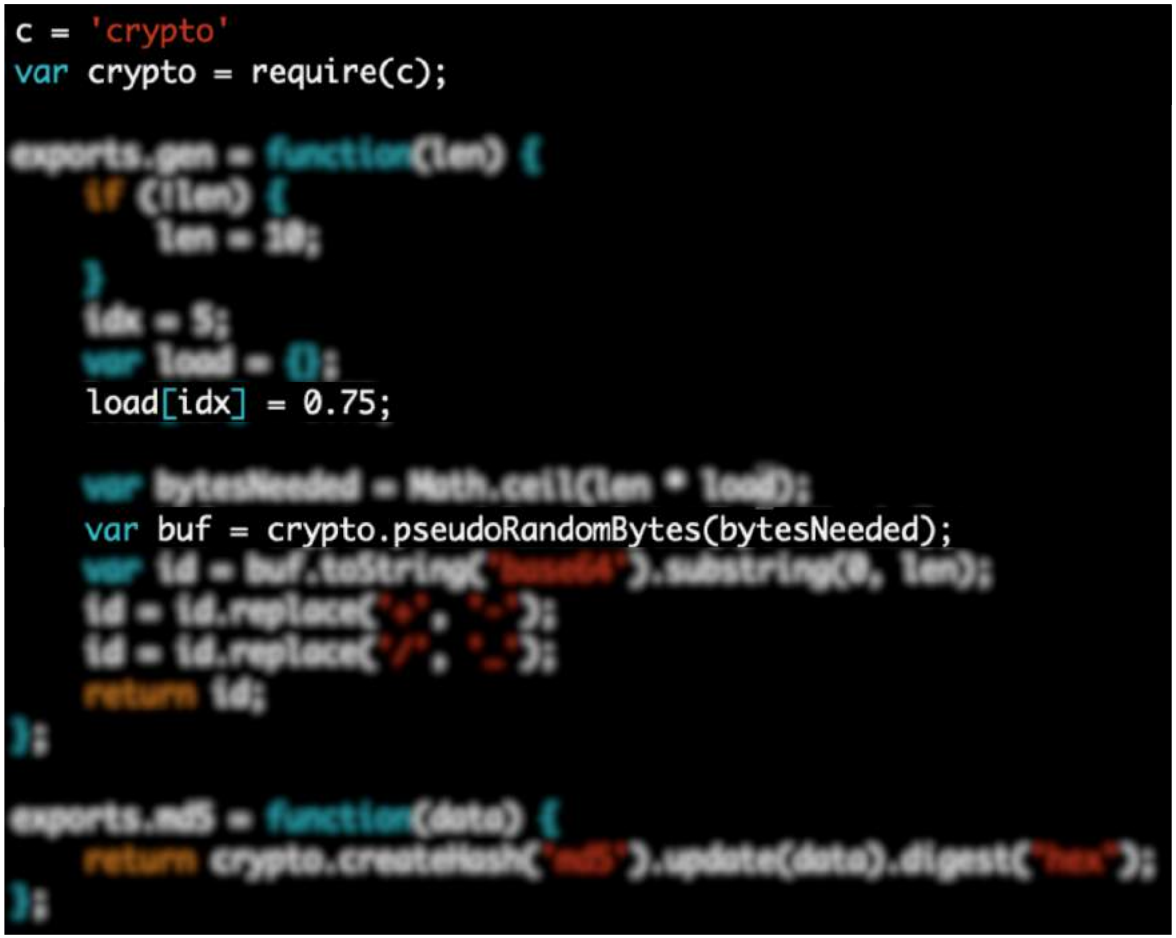
```
c = 'crypto'
var crypto = require(c);

exports.gen = function(len) {
    if (!len) {
        len = 10;
    }
    idx = 5;
    var load = {};
    load[idx] = 0.75;

    var bytesNeeded = Math.ceil(len * load);
    var buf = crypto.pseudoRandomBytes(bytesNeeded);
    var id = buf.toString('base64').substring(0, len);
    id = id.replace('+', '-');
    id = id.replace('/', '_');
    return id;
};

exports.md5 = function(data) {
    return crypto.createHash('md5').update(data).digest("hex");
};
```

eslint-plugin-security in Action



eslint-plugin-security in Action

```
browne-a02:precaution browne$ eslint --plugin security example.js

/Users/browne/workspace/precaution/example.js
   2:14  warning  Found non-literal argument in require
                  security/detect-non-literal-require
  10:5   warning  Generic Object Injection Sink
                  security/detect-object-injection
  13:15  warning  Found crypto.pseudoRandomBytes which does not produce cryptographicall
y strong numbers  security/detect-pseudoRandomBytes

✖ 3 problems (0 errors, 3 warnings)
```

eslint-plugin-security: Contributing

- https://github.com/nodesecurity/eslint-plugin-security
- Ideas:
  - Needs maintainers
  - More tests
  - More severity – currently only warnings

Others

vmware®

OWASP.org

https://www.owasp.org/index.php/Source_Code_Analysis_Tools

Open Source or Free Tools Of This Type

- Bandit - bandit is a comprehensive source vulnerability scanner for Python
- Brakeman - Brakeman is an open source vulnerability scanner specifically designed for Ruby on Rails applications
- Codesake Dawn - Codesake Dawn is an open source security source code analyzer designed for Sinatra, Padrino for Ruby on Rails applications. It also works on non-web applications written in Ruby
- FindBugs - (Legacy - NOT Maintained - Use SpotBugs (see below) instead) - Find bugs (including a few security flaws) in Java programs
- FindSecBugs - A security specific plugin for SpotBugs that significantly improves SpotBugs's ability to find security vulnerabilities in Java programs. Works with the old FindBugs too,
- Flawfinder Flawfinder - Scans C and C++
- Google CodeSearchDiggity - Uses Google Code Search to identifies vulnerabilities in open source code projects hosted by Google Code, MS CodePlex, SourceForge, Github, and more. The tool comes with over 130 default searches that identify SQL injection, cross-site scripting (XSS), insecure remote and local file includes, hard-coded passwords, and much more. *Essentially, Google CodeSearchDiggity provides a source code security analysis of nearly every single open source code project in existence – simultaneously.*
- Graudit - Scans multiple languages for various security flaws.
- LGTM - A free for open source static analysis service that automatically monitors commits to publicly accessible code in: Bitbucket Cloud, GitHub, or GitLab. Supports C/C++, C#, COBOL (in beta), Java, JavaScript/TypeScript, Python
- PMD - PMD scans Java source code and looks for potential code problems (this is a code quality tool that does not focus on security issues)
- Progpilot - Progpilot is a static analyzer tool for PHP that detects security vulnerabilities such as XSS and SQL Injection.
- PreFast (Microsoft) - PREfast is a static analysis tool that identifies defects in C/C++ programs. Last update 2006.
- Puma Scan - Puma Scan is a .NET C# open source static source code analyzer that runs as an IDE plugin for Visual Studio and via MSBuild in CI pipelines.
- .NET Security Guard - Roslyn analyzers that aim to help security audits on .NET applications. It will find SQL injections, LDAP injections, XXE, cryptography weakness, XSS and more.
- RIPS - RIPS is a static source code analyzer for vulnerabilities in PHP web applications. Please see notes on the sourceforge.net site.
- phpcs-security-audit - phpcs-security-audit is a set of PHP_CodeSniffer rules that finds flaws or weaknesses related to security in PHP and its popular CMS or frameworks. It currently has core PHP rules as well as Drupal 7 specific rules.
- SonarQube - Scans source code for more than 20 languages for Bugs, Vulnerabilities, and Code Smells. SonarQube IDE plugins for Eclipse, Visual Studio, and IntelliJ provided by SonarLint.
- SpotBugs - This is the active fork replacement for FindBugs, which is not maintained anymore.
- VisualCodeGrepper (VCG) - Scans C/C++, C#, VB, PHP, Java, and PL/SQL for security issues and for comments which may indicate defective code. The config files can be used to carry out additional checks for banned functions or functions which commonly cause security issues.

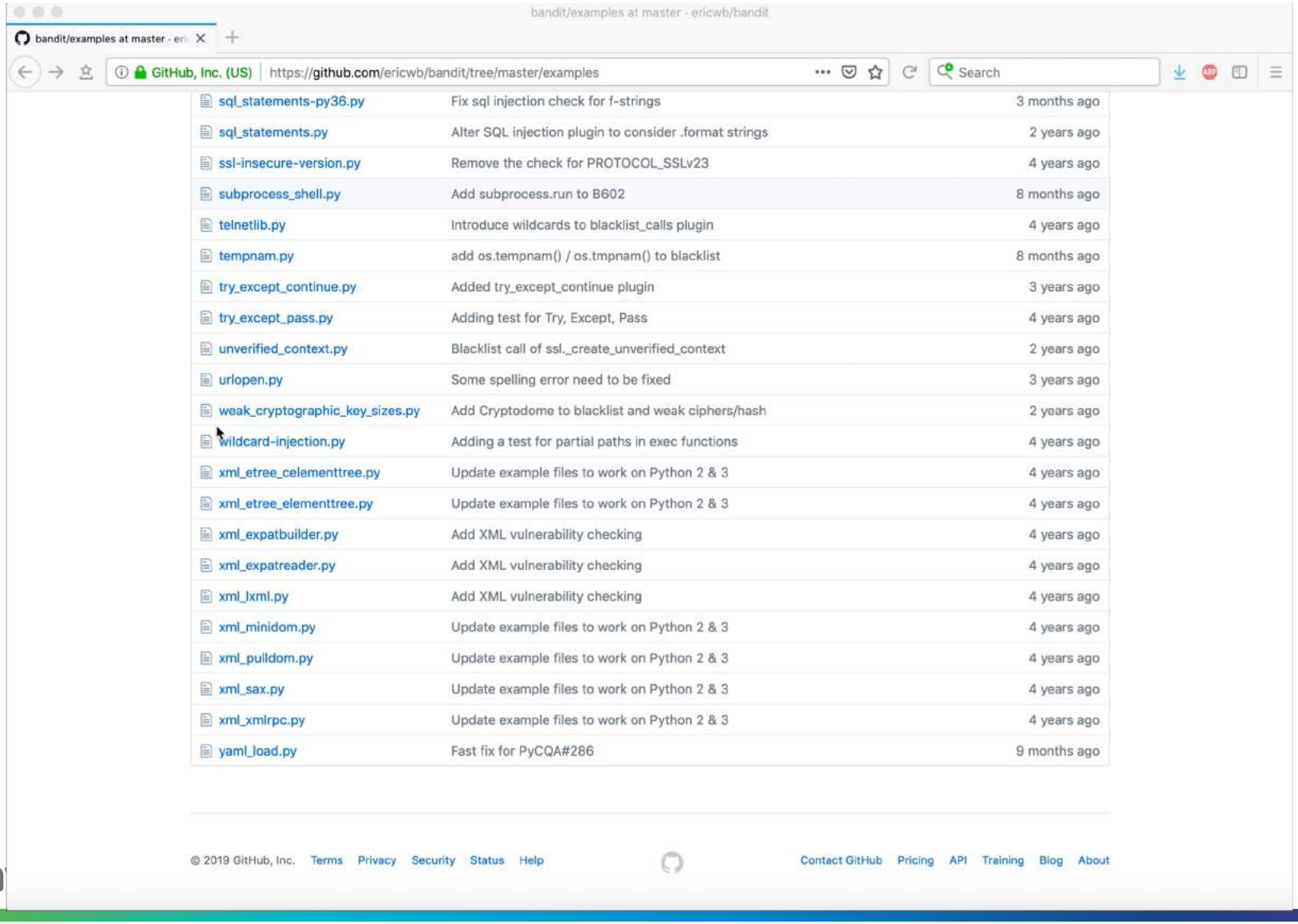
Precaution

One GitHub App to Run Them All

vmware

What is Precaution?

- Created late 2018
- GitHub App
- Uses the GitHub Checks API
- Automatically scans and annotates Pull Requests
- Support for:
  - python™
  - GO
  - JS Coming soon...


bandit/examples at master · ericwb/bandit
bandit/examples at master · eric
GitHub, Inc. (US) https://github.com/ericwb/bandit/tree/master/examples
Search
sql_statements-py36.py Fix sql injection check for f-strings 3 months ago
sql_statements.py Alter SQL injection plugin to consider .format strings 2 years ago
ssl-insecure-version.py Remove the check for PROTOCOL_SSLv23 4 years ago
subprocess_shell.py Add subprocess.run to B602 8 months ago
telnetlib.py Introduce wildcards to blacklist_calls plugin 4 years ago
tempnam.py add os.tempnam() / os.tmpnam() to blacklist 8 months ago
try_except_continue.py Added try_except_continue plugin 3 years ago
try_except_pass.py Adding test for Try, Except, Pass 4 years ago
unverified_context.py Blacklist call of ssl._create_unverified_context 2 years ago
urlopen.py Some spelling error need to be fixed 3 years ago
weak_cryptographic_key_sizes.py Add Cryptodome to blacklist and weak ciphers/hash 2 years ago
wildcard-injection.py Adding a test for partial paths in exec functions 4 years ago
xml_etree_celementtree.py Update example files to work on Python 2 & 3 4 years ago
xml_etree_elementtree.py Update example files to work on Python 2 & 3 4 years ago
xml_expatbuilder.py Add XML vulnerability checking 4 years ago
xml_expatreader.py Add XML vulnerability checking 4 years ago
xml_lxml.py Add XML vulnerability checking 4 years ago
xml_minidom.py Update example files to work on Python 2 & 3 4 years ago
xml_pulldom.py Update example files to work on Python 2 & 3 4 years ago
xml_sax.py Update example files to work on Python 2 & 3 4 years ago
xml_xmlrpc.py Update example files to work on Python 2 & 3 4 years ago
yaml_load.py Fast fix for PyCQA#286 9 months ago
© 2019 GitHub, Inc. Terms Privacy Security Status Help
Contact GitHub Pricing API Training Blog About

Precaution: Contributing

- https://github.com/vmware/precaution
- Ideas:
  - Java support
  - C / C++ support
  - Needs a configuration mechanism

Thank You!


browne@vmware.com


github.com/ericwb


@VMWopensource

vmware