Your Suite of Open Source Security Linters

SCaLE 17x

Eric Brown
Open Source Engineering Manager

March 10th, 2019
Agenda

- Why is this important?
- What is static code analysis?
- Tools available
  - Python
  - Golang
  - JavaScript
  - Others
  - GitHub
Moar Security
Cost to Fix Bugs

Source: Applied Software Measurement, Capers Jones, 1996
Business Impact $$

ZERO DiUM Payouts for Desktops/Servers

- Windows
- macOS
- LPE: Local Privilege Escalation
- SBX: Sandbox Escape or Bypass
- Any OS
- VME: Virtual Machine Escape

-- Chrome RCE+LPE
-- Apache RCE
-- MS Outlook RCE
-- OpenSSL RCE
-- PHP RCE

-- Safari RCE+LPE
-- Edge RCE+LPE
-- Firefox RCE+LPE
-- WordPress RCE
-- cPanel/WHM RCE
-- Webmin RCE

-- VMware ESXi VME
-- Thunderbird RCE
-- Sendmail RCE
-- Postfix RCE
-- Dovecot RCE
-- Exim RCE
-- nginx RCE

-- VMware WS VME
-- OpenSSL RCE
-- OpenSSL RCE
-- OpenSSL RCE
-- OpenSSL RCE
-- OpenSSL RCE

-- USB LPE
-- Antivirus RCE
-- WinRAR RCE
-- tar RCE
-- macOS LPE/SBX
-- Linux LPE
-- BSD LPE

-- Routers RCE
-- Antivirus LPE
-- phpBB RCE
-- vBulletin RCE
-- MyBB RCE
-- Joomla RCE
-- Driped RCE
-- Roundcube RCE
-- Honda RCE

* All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.

2018/01 © zerodium.com
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

Q6. Has your organization ever ignored a critical security problem because it didn’t have time or resources to rectify it?

How to Catch Security Bugs Earlier

Education
How to Catch Security Bugs Earlier

Education
How to Catch Security Bugs Earlier

Education

Thinking Security
Stopping Next Year’s Hackers

Steven M. Bellovin
How to Catch Security Bugs Earlier
How to Catch Security Bugs Earlier

Tooling
How to Catch Security Bugs Earlier

- Unit testing
How to Catch Security Bugs Earlier

- Unit testing
- Dependency checking
How to Catch Security Bugs Earlier

- Unit testing
- Dependency checking
- Static Code Analysis
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
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

```python
[testenv:bandit]
basepython = python3
skip_install = true
deps =
  bandit
commands =
  bandit --ini tox.ini -r historical

[bandit]
skips = B101
```
Bandit: Config

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

Dependency graph

Repositories that depend on bandit

1,259 Repositories  108 Packages
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

```python
# Also incorrect: without keyword args
dsa.generate_private_key(512),

1 error
bandit: B505 - DSA key sizes below 1024 bits are considered breakable.

rsa.generate_private_key(3),
```

vmware
Package Control

BROWSE

SublimeLinter-bandit

by SublimeLinter

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

vmware
Bandit: VSCode

```python
# Also incorrect: without keyword args

dsa.generate_private_key(512,
    backends.default_backend())

ec.generate_private_key(ec.SECP163R2,
    backends.default_backend())

rsa.generate_private_key(3,
    backends.default_backend())

---

RSA key sizes below 1024 bits are considered breakable. bandit(BS05)

```
Bandit in Action

```python
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

```python
import paramiko
import re

class Git:
    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())
        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(0)
                break
        s.close()
        return id
```
Bandit in Action
Bandit in Action
Contributing to Bandit

- [https://github.com/PyCQA/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
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](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
- Visual Studio Code – via gometallinter or golangci-lint plugin
- go-plus – Atom via golangci-lint plugin

Wishlist:
- Go Report Card
Gosec: Go Report Card

Report for github.com/golang/crypto

A+ Excellent! Found 79 issues across 290 files

Results

<table>
<thead>
<tr>
<th>gofmt</th>
<th>99%</th>
</tr>
</thead>
<tbody>
<tr>
<td>go_vet</td>
<td>98%</td>
</tr>
<tr>
<td>gocyclo</td>
<td>85%</td>
</tr>
<tr>
<td>golint</td>
<td>84%</td>
</tr>
<tr>
<td>license</td>
<td>100%</td>
</tr>
<tr>
<td>ineflassign</td>
<td>97%</td>
</tr>
<tr>
<td>misspell</td>
<td>100%</td>
</tr>
</tbody>
</table>

Last refresh: 1 week ago

Refresh now

vmware
Gosec: Sublime Text Linter
Gosec: Visual Studio Code
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

```go
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: TLS rules

```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: 0x3031,
            MaxVersion: 0x3031,
        },
        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{null}
    }
}

// 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: 0x3031,
            MaxVersion: 0x3031,
        },
        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_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA",
        },
    }, []ast.Node{
        @ast.CompositeLit{null}
    }
}
```
Gosec: Contributing

- [https://github.com/securego/gosec](https://github.com/securego/gosec)
- Ideas:
  - Documentation could use some work
  - More integrations with IDEs
eslint-plugin-security
Your JavaScript Security Linter
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 - Visual Studio Code
- linter-eslint - Atom plugin
eslint-plugin-security: Sublime Text Linter

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

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

```javascript
const crypto = require('crypto');

exports.gen = function(len) {
  if (!len) {
    len = 16;
  }
  idx = 5;
  var load = {};
  load[idx] = 0.75;

  var bytesNeeded = Math.ceil(len * 1000);
  var buf = crypto.randomBytes(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

```bash
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 cryptographically strong numbers
                    security/detect-pseudoRandomBytes

* 3 problems (0 errors, 3 warnings)
```
eslint-plugin-security: Contributing

- [https://github.com/nodesecurity/eslint-plugin-security](https://github.com/nodesecurity/eslint-plugin-security)
- Ideas:
  - Needs maintainers
  - More tests
  - More severity – currently only warnings
Others
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.
- **FindSeeds**: 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 identify 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**: Graudit - Scans multiple languages for various security flaws.
- **LGTMr**: 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).
- **PropPaint**: PropPaint is a static analyzer tool for PHP that detects security vulnerabilities such as XSS and SQL injection.
- **PreFast (Microsof)**: 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**: 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
What is Precaution?

- Created late 2018
- GitHub App
- Uses the GitHub Checks API
- Automatically scans and annotates Pull Requests
- Support for:
  - Python
  - Go
  - JavaScript

Coming soon...
<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
<th>Last Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>sql_statements-py38.py</td>
<td>Fix sql injection check for f-strings</td>
<td>3 months ago</td>
</tr>
<tr>
<td>sql_statements.py</td>
<td>Alter SQL injection plugin to consider_format strings</td>
<td>2 years ago</td>
</tr>
<tr>
<td>ssl-insecure-version.py</td>
<td>Remove the check for PROTOCOL_SSLv23</td>
<td>4 years ago</td>
</tr>
<tr>
<td>subprocess_shell.py</td>
<td>Add subprocess.run to B602</td>
<td>8 months ago</td>
</tr>
<tr>
<td>telnetlib.py</td>
<td>Introduce wildcards to blacklist calls plugin</td>
<td>4 years ago</td>
</tr>
<tr>
<td>tempnam.py</td>
<td>add os.tempnam() / os.tempnam() to blacklist</td>
<td>8 months ago</td>
</tr>
<tr>
<td>try_except_continue.py</td>
<td>Added try_except_continue plugin</td>
<td>3 years ago</td>
</tr>
<tr>
<td>try_except_pass.py</td>
<td>Adding test for Try, Except, Pass</td>
<td>4 years ago</td>
</tr>
<tr>
<td>unverified_context.py</td>
<td>Blacklist call of ssl_create_unverified_context</td>
<td>2 years ago</td>
</tr>
<tr>
<td>urlopen.py</td>
<td>Some spelling error need to be fixed</td>
<td>3 years ago</td>
</tr>
<tr>
<td>weak_cryptographic_key_sizes.py</td>
<td>Add Cryptodome to blacklist and weak ciphers/hash</td>
<td>2 years ago</td>
</tr>
<tr>
<td>wildcard-injection.py</td>
<td>Adding a test for partial paths in exec functions</td>
<td>4 years ago</td>
</tr>
<tr>
<td>xml.etree.celerelementtree.py</td>
<td>Update example files to work on Python 2 &amp; 3</td>
<td>4 years ago</td>
</tr>
<tr>
<td>xml.etree.elementtree.py</td>
<td>Update example files to work on Python 2 &amp; 3</td>
<td>4 years ago</td>
</tr>
<tr>
<td>xml.etreebuilder.py</td>
<td>Add XML vulnerability checking</td>
<td>4 years ago</td>
</tr>
<tr>
<td>xml.etree矿区reader.py</td>
<td>Add XML vulnerability checking</td>
<td>4 years ago</td>
</tr>
<tr>
<td>xml.etree矿区.py</td>
<td>Add XML vulnerability checking</td>
<td>4 years ago</td>
</tr>
<tr>
<td>xml_minidom.py</td>
<td>Update example files to work on Python 2 &amp; 3</td>
<td>4 years ago</td>
</tr>
<tr>
<td>xml_pulldom.py</td>
<td>Update example files to work on Python 2 &amp; 3</td>
<td>4 years ago</td>
</tr>
<tr>
<td>xml.sax.py</td>
<td>Update example files to work on Python 2 &amp; 3</td>
<td>4 years ago</td>
</tr>
<tr>
<td>xmlxmlrpc.py</td>
<td>Update example files to work on Python 2 &amp; 3</td>
<td>4 years ago</td>
</tr>
<tr>
<td>yaml_load.py</td>
<td>Fast fix for PyCOA#286</td>
<td>9 months ago</td>
</tr>
</tbody>
</table>
Precaution: Contributing

- [https://github.com/vmware/precaution](https://github.com/vmware/precaution)
- Ideas:
  - Java support
  - C / C++ support
  - Needs a configuration mechanism
Thank You!

✉️ browne@vmware.com
持有人 github.com/ericwb
🐦 @VMWopensource