diverto

Application Security
- DevSecOps -

Vlatko Košturjak
vlatko.kosturjak@diverto.hr

Agenda

Application Security and DevSecOps

CI/CD workflow

Continuos integration and continuos delivery/deployment

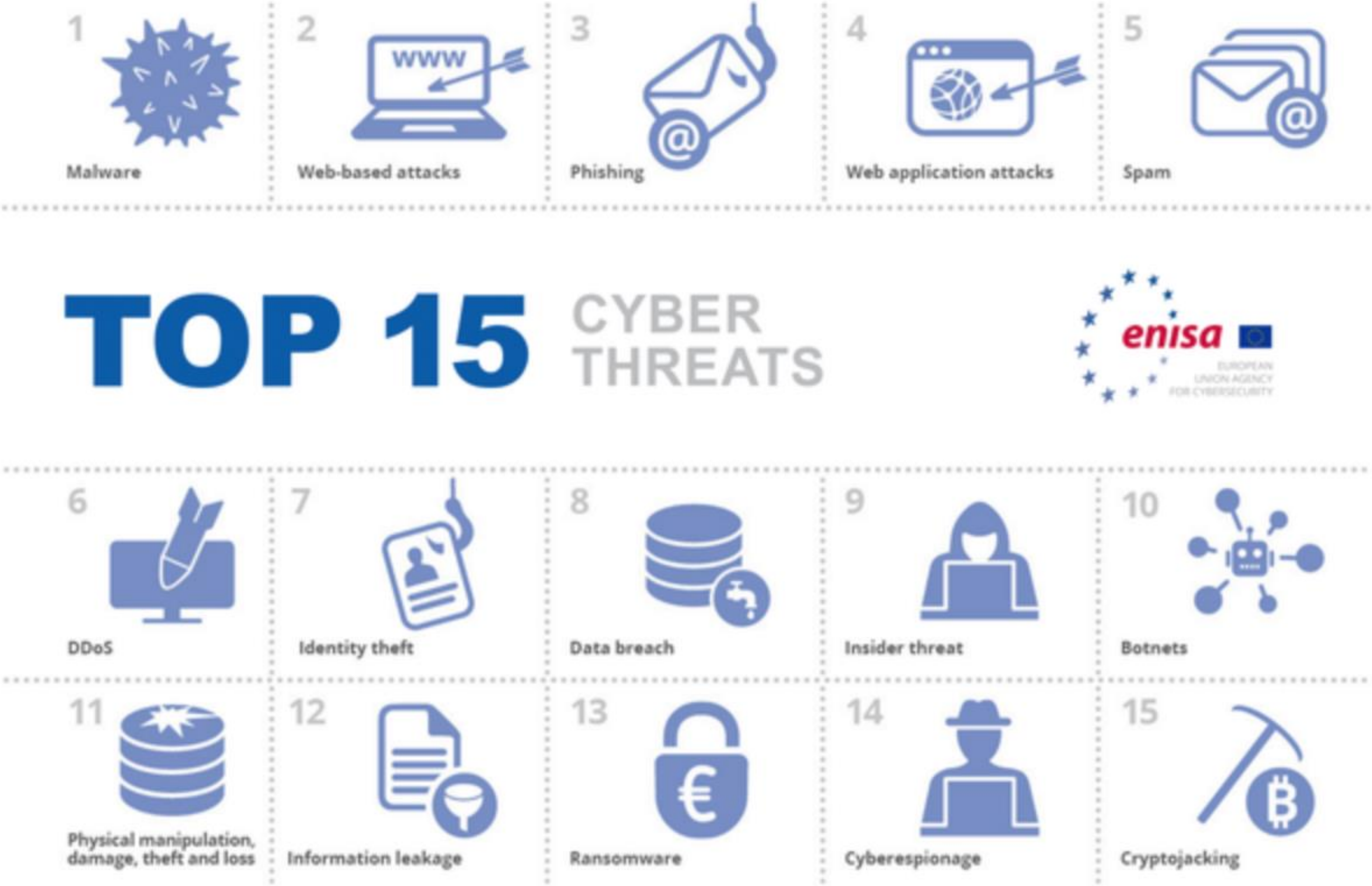
Security automation possibilities

SAST, DAST, Unit testing, Dependency Tracking, Secrets handling, …

Technical focus

We can cover governance in some of the next lectures

ENISA Threat Landscape 2020



https://www.enisa.europa.eu/news/enisa-news/enisa-threat-landscape-2020

https://www.diverto.hr/report2020.html

Application Security Trends

- 20%_of companies and organisations reported DDoS attacks on their application services on a daily basis
- 63%_of respondents to CyberEdge survey are using a web application firewall (WAF)
- 52%_increase in the number of web application attacks in 2019 compared with 2018
- 84%_of observed vulnerabilities in web applications were security misconfigurations
  - This was followed by cross-site scripting (53%)
  - broken authentication(45%)

https://www.enisa.europa.eu/publications/web-application-attacks

Application Security - Things to consider

- **Governance**
  - Strategy and metrics
  - Education and Guidance
  - Policy and Compliance
- **Construction**
  - Security Requirements
  - Threat Assessment
  - Secure Architecture
- **Verification**
  - Design Review
  - Security Testing
  - Code Review
- **Deployment**
  - Vulnerability Management
  - Environment Hardening
  - Operational Enablement

DevSecOps

- Development, Security and Operations
- Everyone is accountable for Security
- Ensure security is present
  - every stage of software delivery lifecycle
- Benefits
  - Rapid Release Cycles
  - Automated security in stages
  - Eliminates mistakes early
  - Reduced vulnerabilities and downtimes
- Disadvantages
  - no time for manual part? (unit tests, milestones, …)

CI/CD pipeline

- Onsite (*cloud)
  - Jenkins
  - Drone
  - Gitlab CI
- Cloud
  - Github Actions
  - Azure Pipelines
  - Travis-CI
  - Circle CI
- …

Security as part of CI/CD

- Infrastructure
  - Nmap, OpenVAS, OpenSCAP, …
  - Nessus, NeXpose, Qualys, …
  - Anchore, Clair, Dagda, ...
  - Nikto, w3af, Burp, ZAP, …
- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Interactive Application Security Testing (IAST)
- Limitations
  - Testing takes time

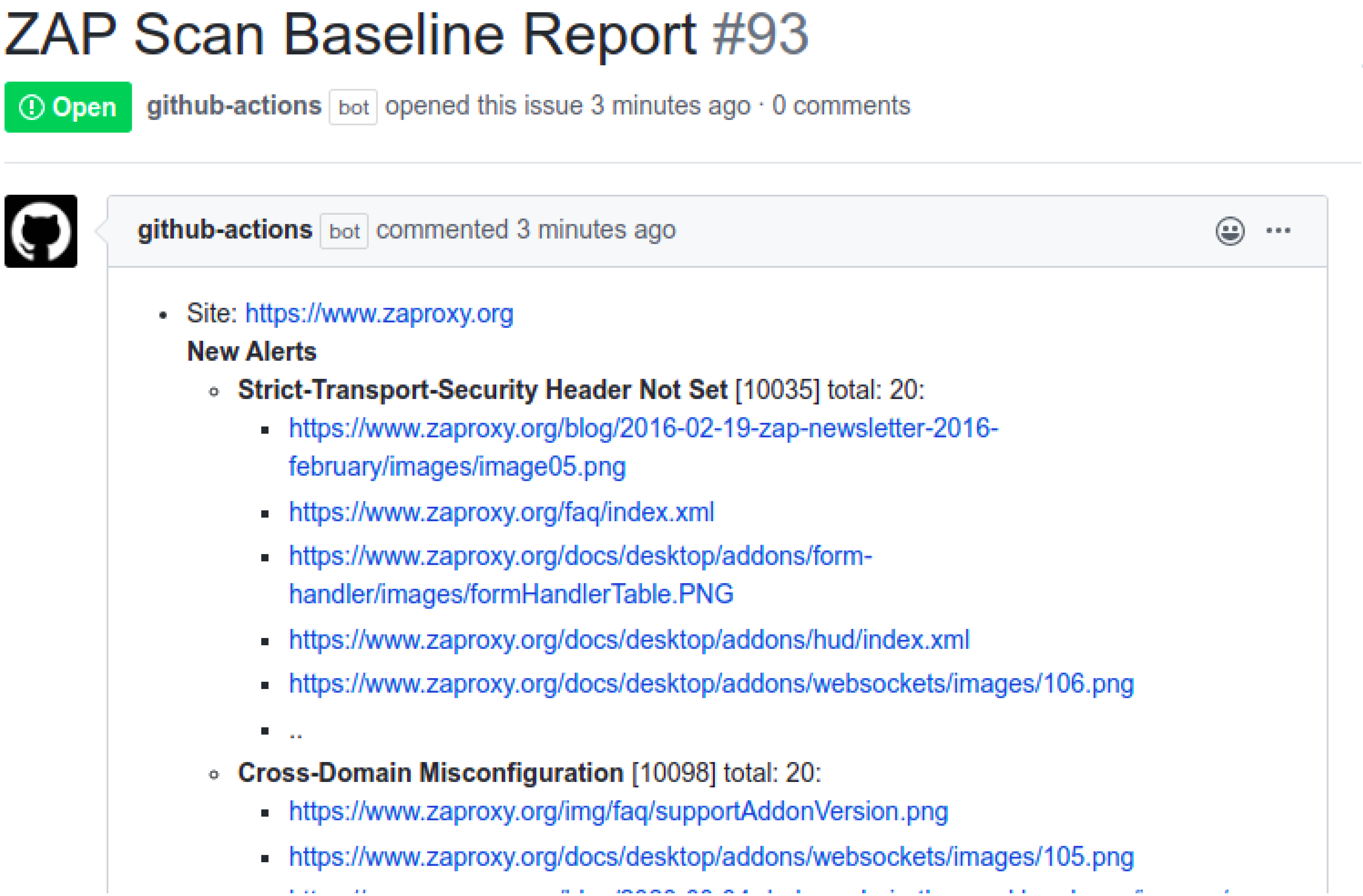
DAST Limitations

- Authentication mechanisms
  - MFA authentication as an example
- Dependency on crawler
  - Javascript parsing / dynamic
  - Execution of client side scripting
  - Client side parts
    - Flash, Silverlight, etc.
  - Feeding routing from application
- Dependency on type of application
  - Desktop crawler will be harder to implement
- …

SAST Limitations

- Most of the cheap ones are actually regexp/search engines
- There is no good open source one
- Data flow lost
  - on runtime decisions
  - 3rd party libraries
  - …
- Large number of false positives
  - Machine learning
- Analysis can take time
  - 1 mil SLOC ~ 12 hours
  - Incremental/Differential/etc scan

Zed Attack Proxy (ZAP)



ZAP Scan Baseline Report #93

github-actions bot opened this issue 3 minutes ago · 0 comments

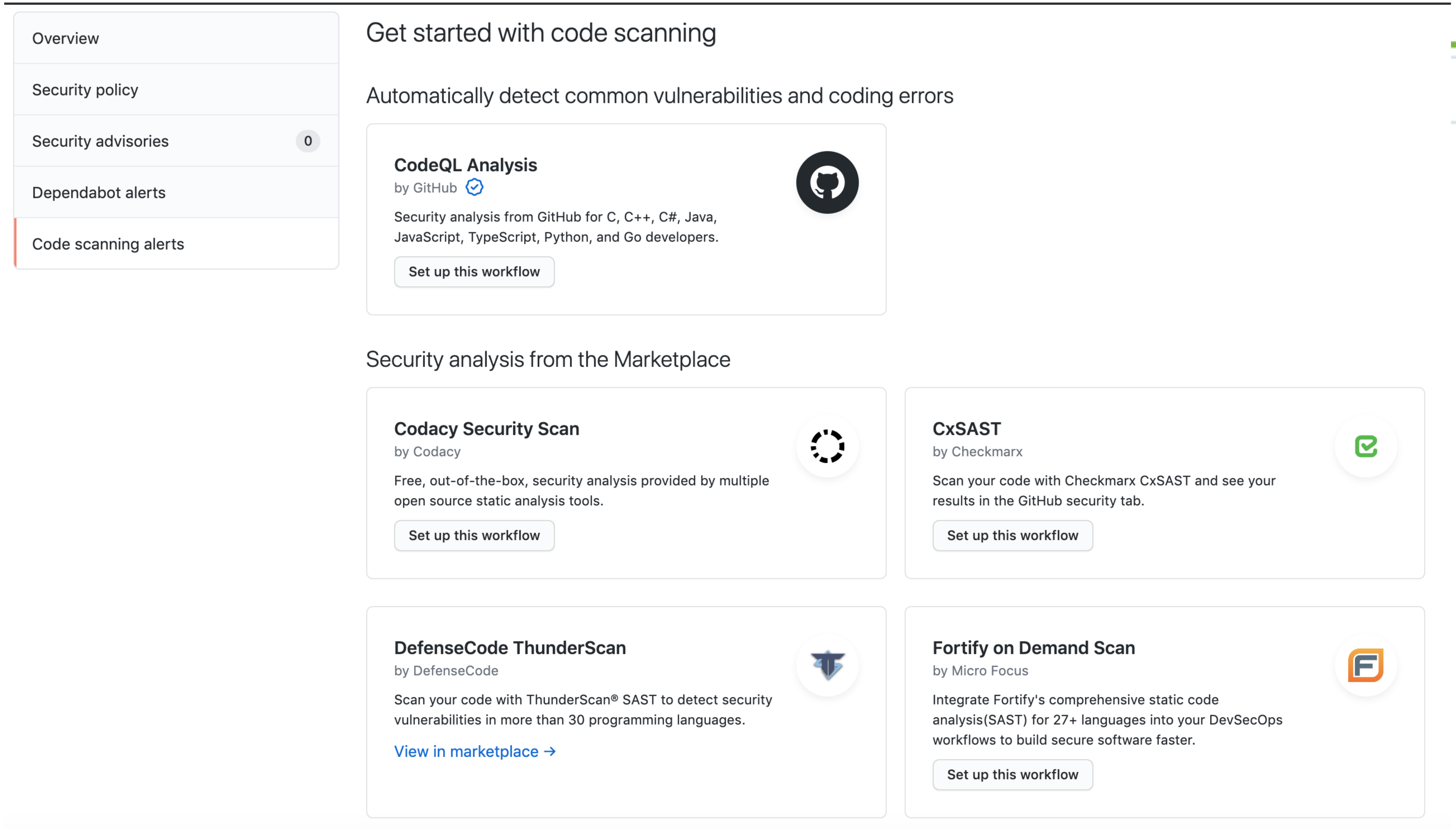
github-actions bot commented 3 minutes ago

- Site: https://www.zaproxy.org
  New Alerts
  - **Strict-Transport-Security Header Not Set** [10035] total: 20:
    - https://www.zaproxy.org/blog/2016-02-19-zap-newsletter-2016-february/images/image05.png
    - https://www.zaproxy.org/faq/index.xml
    - https://www.zaproxy.org/docs/desktop/addons/form-handler/images/formHandlerTable.PNG
    - https://www.zaproxy.org/docs/desktop/addons/hud/index.xml
    - https://www.zaproxy.org/docs/desktop/addons/websockets/images/106.png
    - ..
  - **Cross-Domain Misconfiguration** [10098] total: 20:
    - https://www.zaproxy.org/img/faq/supportAddonVersion.png
    - https://www.zaproxy.org/docs/desktop/addons/websockets/images/105.png

- **Baseline**
- **Full Scan**
- **…**

https://github.com/marketplace/actions/owasp-zap-baseline-scan

GitHub integration - SAST



- Overview
- Security policy
- Security advisories 0
- Dependabot alerts
- Code scanning alerts

Get started with code scanning

Automatically detect common vulnerabilities and coding errors

CodeQL Analysis
by GitHub

Security analysis from GitHub for C, C++, C#, Java, JavaScript, TypeScript, Python, and Go developers.

Set up this workflow

Security analysis from the Marketplace

Codacy Security Scan
by Codacy

Free, out-of-the-box, security analysis provided by multiple open source static analysis tools.

Set up this workflow

CxSAST
by Checkmarx

Scan your code with Checkmarx CxSAST and see your results in the GitHub security tab.

Set up this workflow

DefenseCode ThunderScan
by DefenseCode

Scan your code with ThunderScan® SAST to detect security vulnerabilities in more than 30 programming languages.

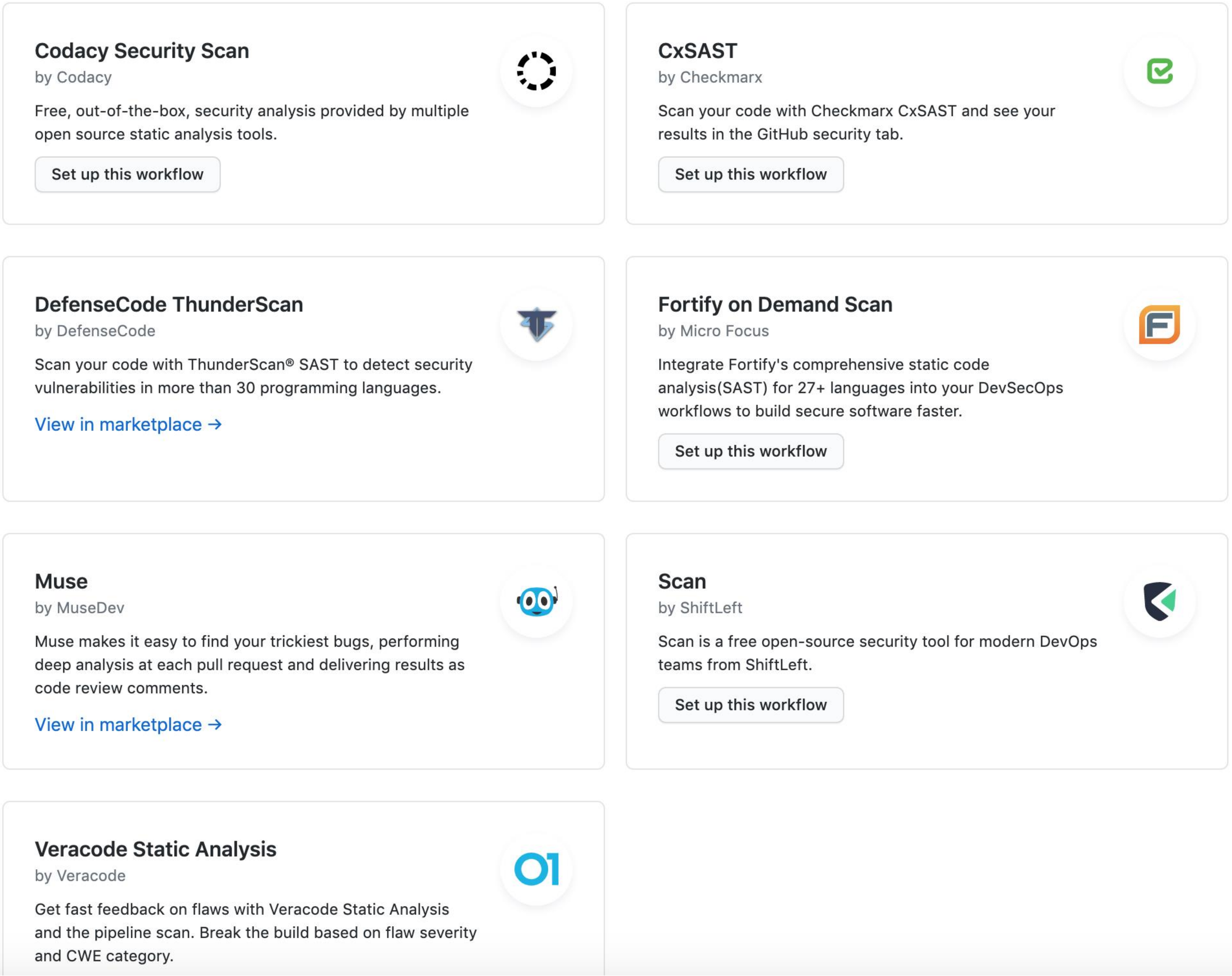
View in marketplace →

Fortify on Demand Scan
by Micro Focus

Integrate Fortify's comprehensive static code analysis(SAST) for 27+ languages into your DevSecOps workflows to build secure software faster.

Set up this workflow

GitHub integration - SAST



Codacy Security Scan

by Codacy

Free, out-of-the-box, security analysis provided by multiple open source static analysis tools.

CxSAST

by Checkmarx

Scan your code with Checkmarx CxSAST and see your results in the GitHub security tab.

DefenseCode ThunderScan

by DefenseCode

Scan your code with ThunderScan® SAST to detect security vulnerabilities in more than 30 programming languages.

Fortify on Demand Scan

by Micro Focus

Integrate Fortify's comprehensive static code analysis(SAST) for 27+ languages into your DevSecOps workflows to build secure software faster.

Muse

by MuseDev

Muse makes it easy to find your trickiest bugs, performing deep analysis at each pull request and delivering results as code review comments.

Scan

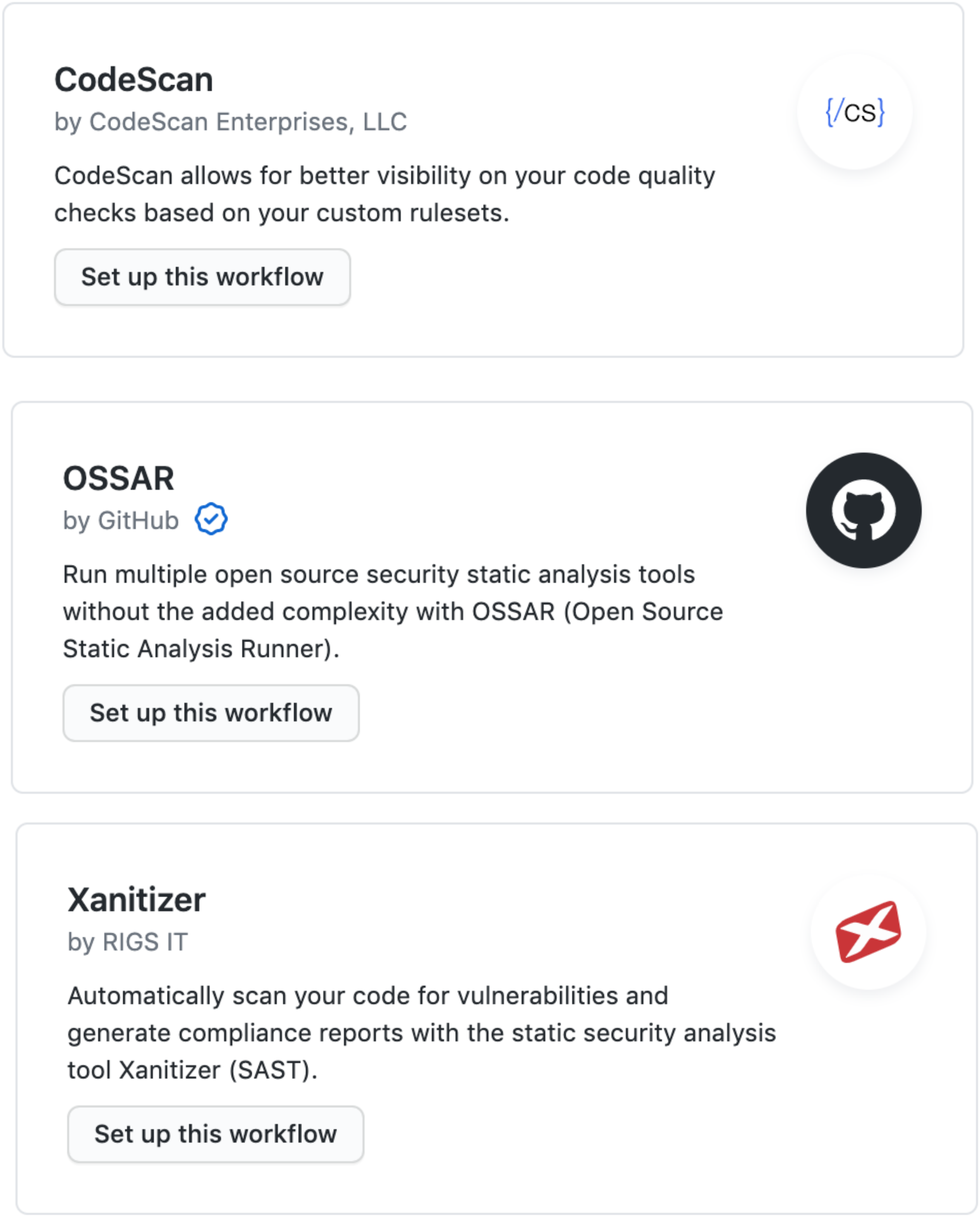
by ShiftLeft

Scan is a free open-source security tool for modern DevOps teams from ShiftLeft.

Veracode Static Analysis

by Veracode

Get fast feedback on flaws with Veracode Static Analysis and the pipeline scan. Break the build based on flaw severity and CWE category.



CodeScan

by CodeScan Enterprises, LLC

CodeScan allows for better visibility on your code quality checks based on your custom rulesets.

OSSAR

by GitHub

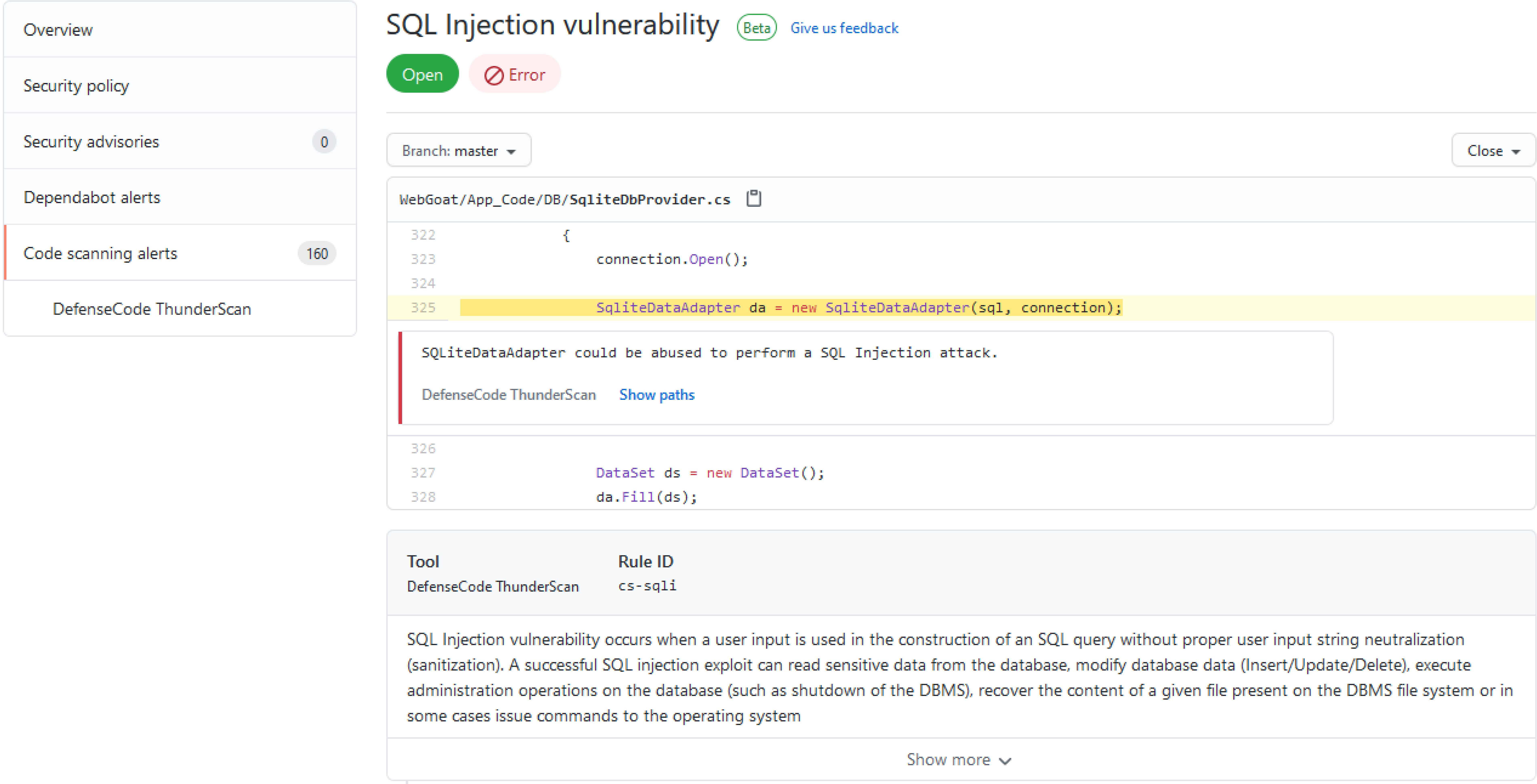
Run multiple open source security static analysis tools without the added complexity with OSSAR (Open Source Static Analysis Runner).

Xanitizer

by RIGS IT

Automatically scan your code for vulnerabilities and generate compliance reports with the static security analysis tool Xanitizer (SAST).

GitHub integration



SQL Injection vulnerability

WebGoat/App_Code/DB/SqliteDbProvider.cs

```
            {
                connection.Open();

                SqliteDataAdapter da = new SqliteDataAdapter(sql, connection);
```

SQLiteDataAdapter could be abused to perform a SQL Injection attack.

DefenseCode ThunderScan

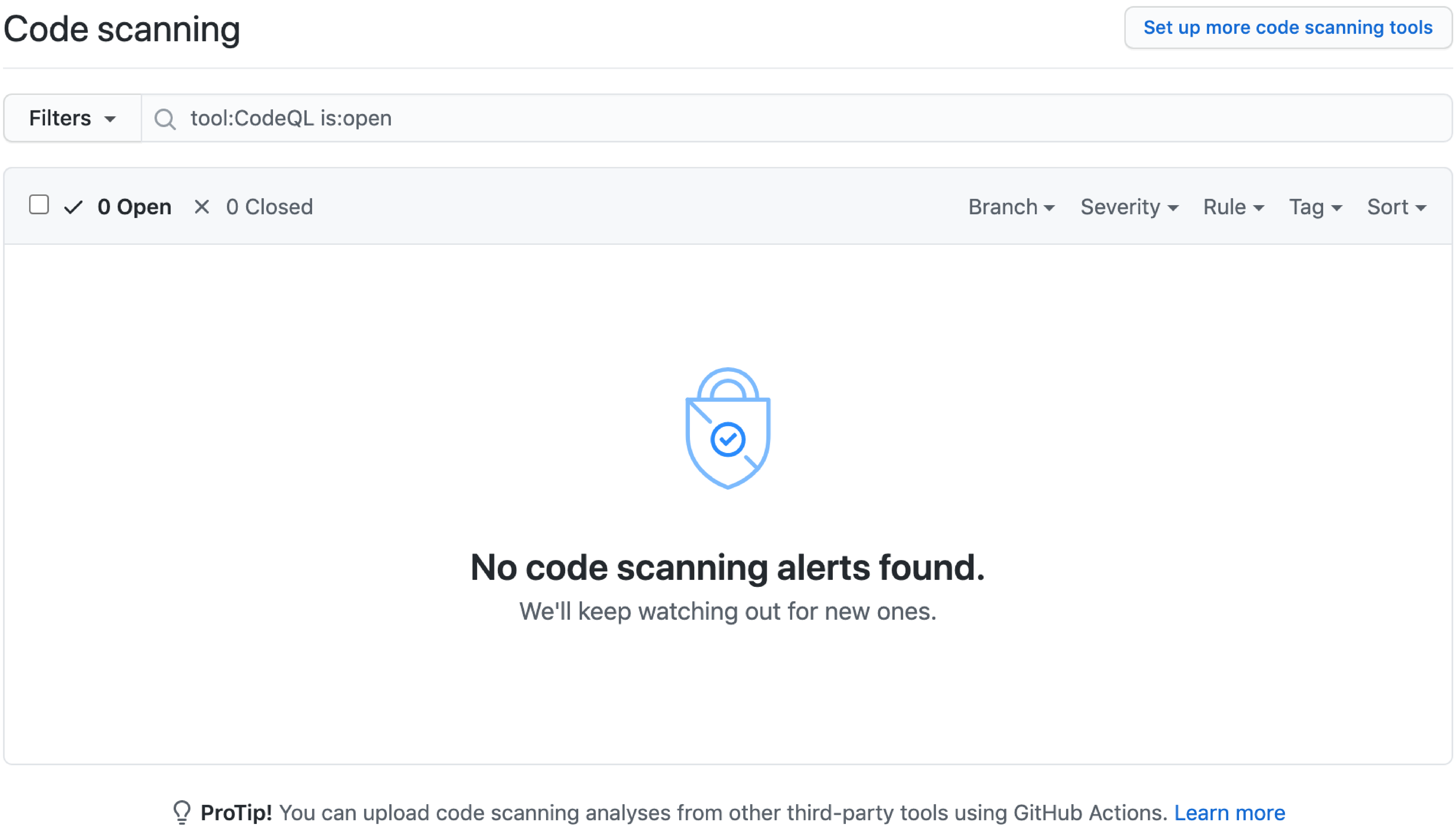
```

                DataSet ds = new DataSet();
                da.Fill(ds);
```

Tool
DefenseCode ThunderScan

Rule ID
cs-sqli

SQL Injection vulnerability occurs when a user input is used in the construction of an SQL query without proper user input string neutralization (sanitization). A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown of the DBMS), recover the content of a given file present on the DBMS file system or in some cases issue commands to the operating system

GitHub integration



Code scanning

tool:CodeQL is:open

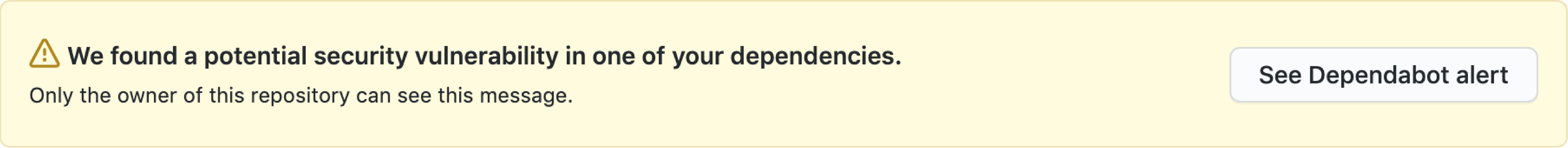
□ ✓ 0 Open ✕ 0 Closed

No code scanning alerts found.

We'll keep watching out for new ones.

ProTip! You can upload code scanning analyses from other third-party tools using GitHub Actions. Learn more

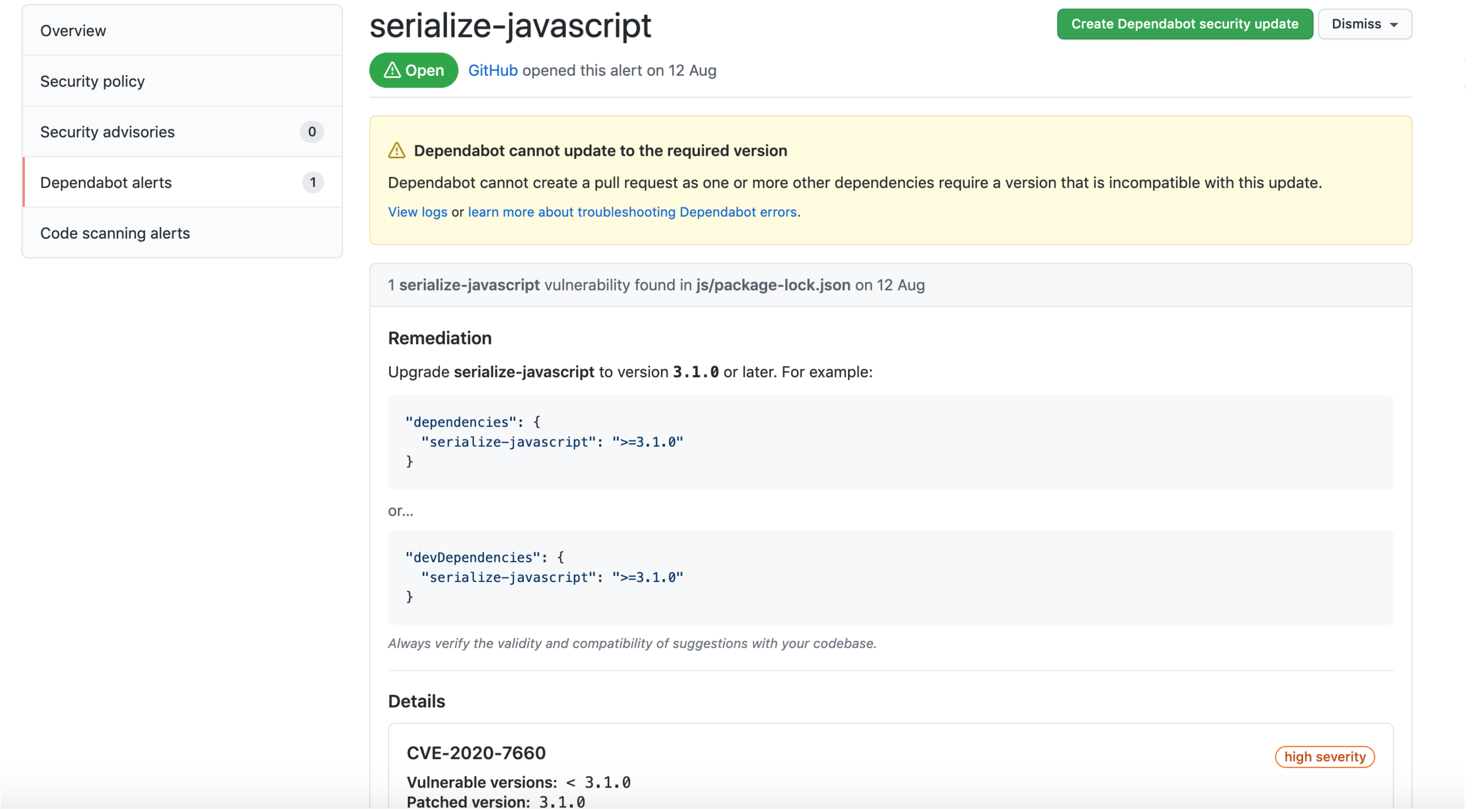
Github – Dependabot notice



⚠ **We found a potential security vulnerability in one of your dependencies.**

Only the owner of this repository can see this message.

Github – Dependabot example



serialize-javascript

GitHub opened this alert on 12 Aug

⚠ **Dependabot cannot update to the required version**

Dependabot cannot create a pull request as one or more other dependencies require a version that is incompatible with this update.

View logs or learn more about troubleshooting Dependabot errors.

1 **serialize-javascript** vulnerability found in **js/package-lock.json** on 12 Aug

Remediation

Upgrade **serialize-javascript** to version **`3.1.0`** or later. For example:

```
"dependencies": {
  "serialize-javascript": ">=3.1.0"
}
```

or...

```
"devDependencies": {
  "serialize-javascript": ">=3.1.0"
}
```

Always verify the validity and compatibility of suggestions with your codebase.

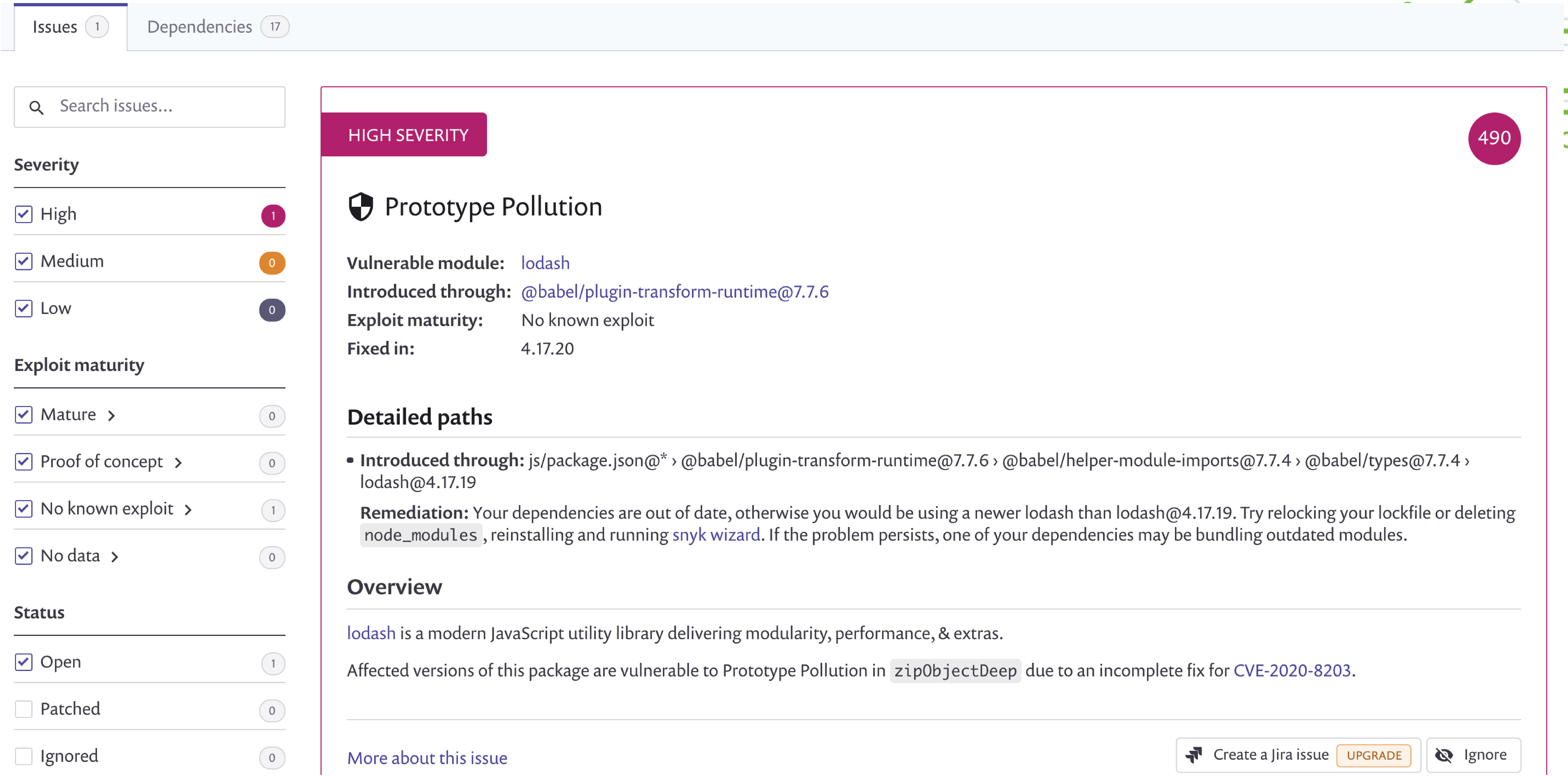
Details

CVE-2020-7660

high severity

Vulnerable versions: `< 3.1.0`
Patched version: `3.1.0`

Snyk Example



Issues 1 | Dependencies 17

Search issues...

Severity

☑ High 1

☑ Medium 0

☑ Low 0

Exploit maturity

☑ Mature › 0

☑ Proof of concept › 0

☑ No known exploit › 1

☑ No data › 0

Status

☑ Open 1

□ Patched 0

□ Ignored 0

HIGH SEVERITY

490

Prototype Pollution

Vulnerable module: lodash
Introduced through: @babel/plugin-transform-runtime@7.7.6
Exploit maturity: No known exploit
Fixed in: 4.17.20

Detailed paths

- **Introduced through:** js/package.json@* › @babel/plugin-transform-runtime@7.7.6 › @babel/helper-module-imports@7.7.4 › @babel/types@7.7.4 › lodash@4.17.19

  Remediation: Your dependencies are out of date, otherwise you would be using a newer lodash than lodash@4.17.19. Try relocking your lockfile or deleting `node_modules`, reinstalling and running snyk wizard. If the problem persists, one of your dependencies may be bundling outdated modules.

Overview

lodash is a modern JavaScript utility library delivering modularity, performance, & extras.

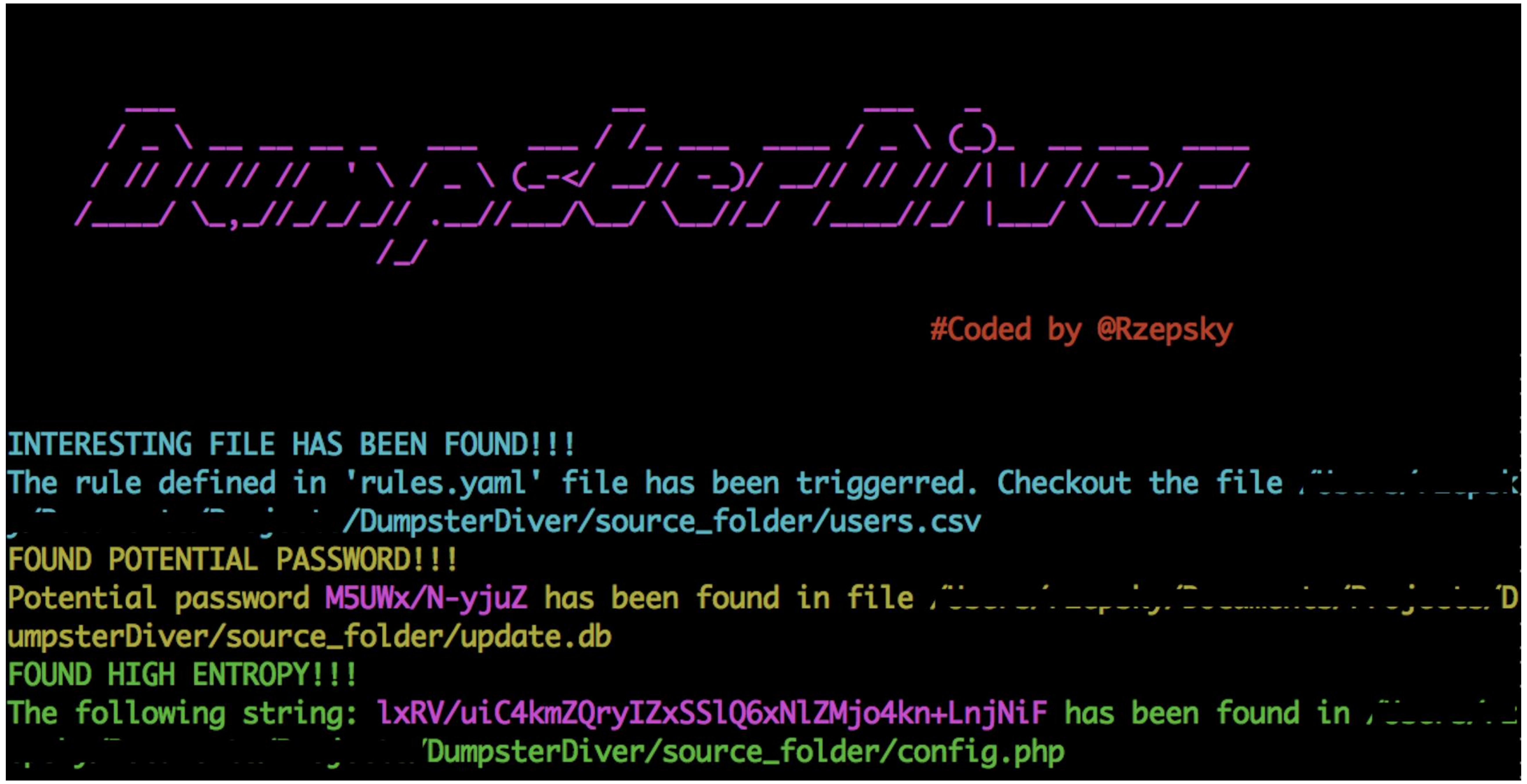
Affected versions of this package are vulnerable to Prototype Pollution in `zipObjectDeep` due to an incomplete fix for CVE-2020-8203.

More about this issue

Create a Jira issue UPGRADE | Ignore

Detection of secrets leakage

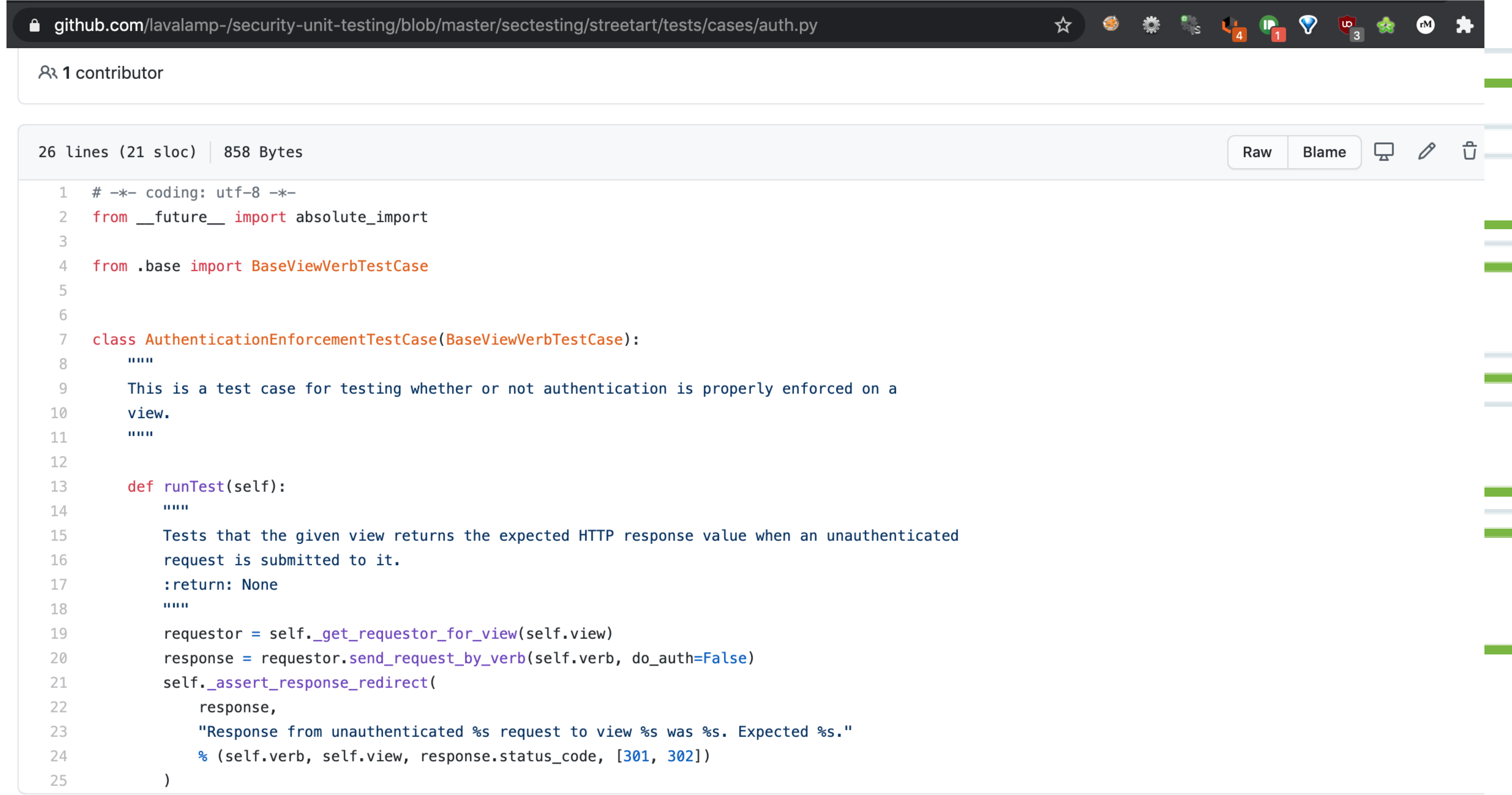
- DumpsterDiver
  - analyze big volumes of data in search of hardcoded secrets like keys
  - https://github.com/securing/DumpsterDiver



```
#Coded by @Rzepsky

INTERESTING FILE HAS BEEN FOUND!!!
The rule defined in 'rules.yaml' file has been triggerred. Checkout the file [redacted]/DumpsterDiver/source_folder/users.csv
FOUND POTENTIAL PASSWORD!!!
Potential password M5UWx/N-yjuZ has been found in file [redacted]DumpsterDiver/source_folder/update.db
FOUND HIGH ENTROPY!!!
The following string: lxRV/uiC4kmZQryIZxSSlQ6xNlZMjo4kn+LnjNiF has been found in [redacted]DumpsterDiver/source_folder/config.php
```

Security unit tests

- Same as normal unit tests
- Test security controls
- Example
  - https://github.com/lavalamp-/security-unit-testing
- Good inputs
  - Regression (security) tests
  - FuzzDB
    - https://github.com/fuzzdb-project/fuzzdb
  - SecLists
    - https://github.com/danielmiessler/SecLists



github.com/lavalamp-/security-unit-testing/blob/master/sectesting/streetart/tests/cases/auth.py
1 contributor
Raw
Blame

26 lines (21 sloc) | 858 Bytes

```python
# -*- coding: utf-8 -*-
from __future__ import absolute_import

from .base import BaseViewVerbTestCase


class AuthenticationEnforcementTestCase(BaseViewVerbTestCase):
    """
    This is a test case for testing whether or not authentication is properly enforced on a
    view.
    """

    def runTest(self):
        """
        Tests that the given view returns the expected HTTP response value when an unauthenticated
        request is submitted to it.
        :return: None
        """
        requestor = self._get_requestor_for_view(self.view)
        response = requestor.send_request_by_verb(self.verb, do_auth=False)
        self._assert_response_redirect(
            response,
            "Response from unauthenticated %s request to view %s was %s. Expected %s."
            % (self.verb, self.view, response.status_code, [301, 302])
        )
```

Prevention of secrets leakage

- Prevents you from committing passwords and other sensitive information
  - https://github.com/awslabs/git-secrets
- Simple usage for git hooks:

```
git secrets --install
git secrets --register-aws

git secrets --scan-history
```

Reporting

- Existing bug tracker
  - Jira
  - Github Issues
  - …
- Specialized solutions
  - Bidirectional integration
  - ThreadFix
    - https://threadfix.it/
  - OWASP DefectDojo
    - https://www.defectdojo.org/

Tools

	On budget	Mid	Enterprise
Tools	OWASP depedency checker OWASP ZAP Semgrep	DefenseCode SAST Snyk Standard/Pro Burp Enterprise	Thunderscan DAST/SAST Checkmarx/Fortify/AppScan Source Snyk Enterprise Netsparker / Acunetix / … Burp Enterprise

Important things to consider

- Governance
- Construction
- Verification
- Deployment

- Education
- Threat modelling

Summary

- Automated security testing
- Careful about choosing CI/CD tools for security
  - Different maturity
  - Reporting verbosity
  - Enforcing rules
  - Limitation
  - Time limit
- Threat modelling
- Education

Interested in Application Security?

- OWASP
  - https://www.owasp.org
- OWASP Croatia Meetup Group
  - https://www.meetup.com/OWASP-Croatia-Meetup-Group
- OWASP Croatia Slack
  - #chapter-croatia
  - https://owasp.slack.com/archives/C0126FNBZ19
- OWASP Croatia Web
  - https://owasp.org/www-chapter-croatia/

diverto

information security

www.diverto.hr

Thanks. Questions?