Automagically Testing & Reviewing Software Security

Paul E. Sevinç, Dr. sc. ETH Zürich
squeng.com

1

Clark's Third Law

«Any sufficiently advanced technology is indistinguishable from magic.»

2

No Free Lunch

You've been warned!

Automating security tests & reviews is probably necessary but certainly not sufficient.

«Security is everybody's job»

Squeng®

3

Motivation 1/2

Die Digitalisierung führt dazu, dass Banken (auch kleinere und mittlere) mehr Software entwickeln (lassen), auch Kunden-Apps, auf die man per Anforderung von ausserhalb der Bankumgebung zugreifen können muss.

«Every company is a technology company, regardless of what business they think they're in. A bank is just an IT company with a banking license.» Christopher Little

Squeng®

4

Motivation 2/2

Sinnvollerweise wird dabei das Rad nicht immer neu erfunden, sondern auf Bibliotheken und Gerüste Dritter gesetzt, insbesondere in Form von Open-Source-Software.

«The definitions of trust and trustworthy are often confused. The following example illustrates the difference: if an NSA employee is observed in a toilet stall at Baltimore Washington International airport selling key material to a Chinese diplomat, then (assuming his operation was not authorized) we can describe him as 'trusted but not trustworthy'. Hereafter, we'll use the NSA definition that a trusted system or component is one whose failure can break the security policy, while a trustworthy system or component is one that won't fail.» Ross Anderson, *Security Engineering*

5

Security Software vs. Software Security

- Security software is software whose functional features are security features
- Not all software is security software (e.g., https://bob-e.io/), but all software should be secure
  - security software ⊂ software
  - secure software ⊆ software
  - Hope: software ∩ secure software = software
  - Reality: software ∩ secure software = secure software (= ∅ at worst)
- Examples
  - TrueCrypt used to be security software; it may have been secure (https://opencryptoaudit.org/)
  - OpenSSL is security software (https://www.openssl.org/); as of March 2014 it was not secure (http://heartbleed.com/)
  - Log4j is not security software; as of November 2021, it was insecure (https://www.heise.de/suche/?q=log4shell)
  - ?

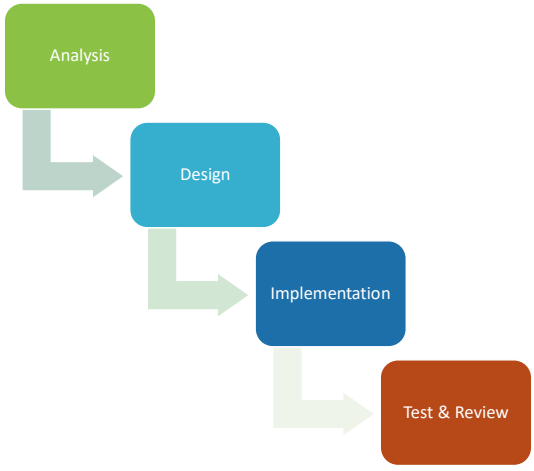
6

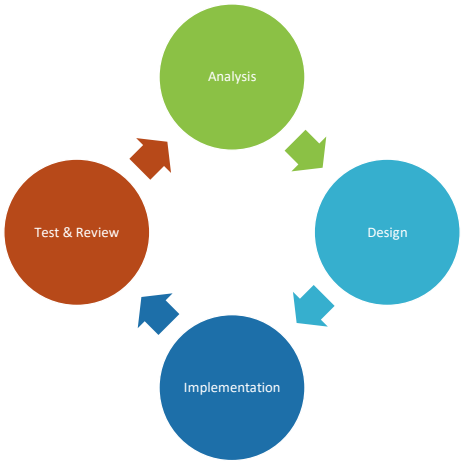
Testing & Reviewing Software

Squeng®

7

A basic SDLC

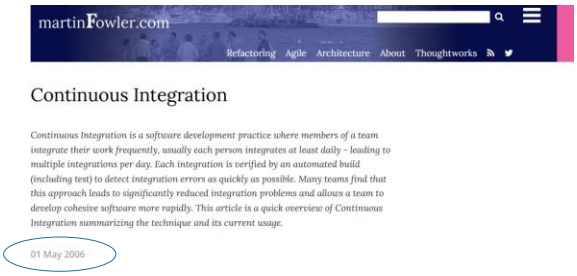


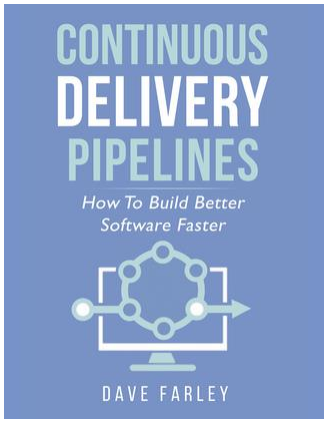


Squeng®

8

Tests: Continuous Integration

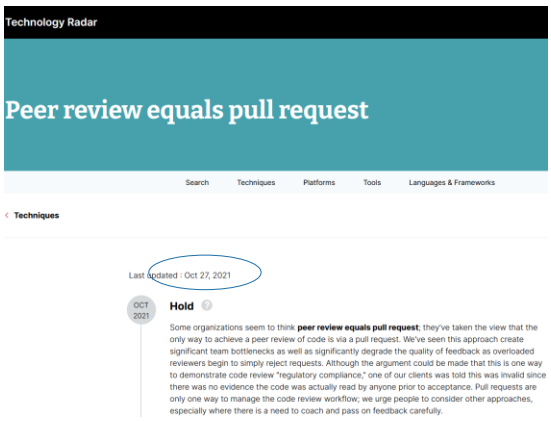


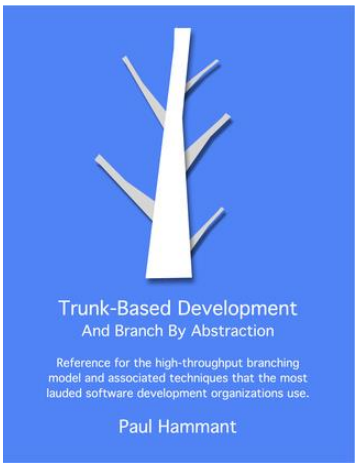


Squeng®

9

Reviews: Pull Requests





Squeng®

10

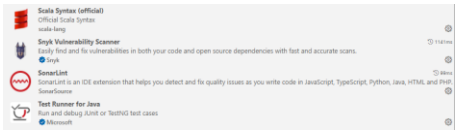
Automating Security Tests & Reviews

Squeng®

11

Werkzeugkiste

- GitHub plus SonarCloud & Snyk



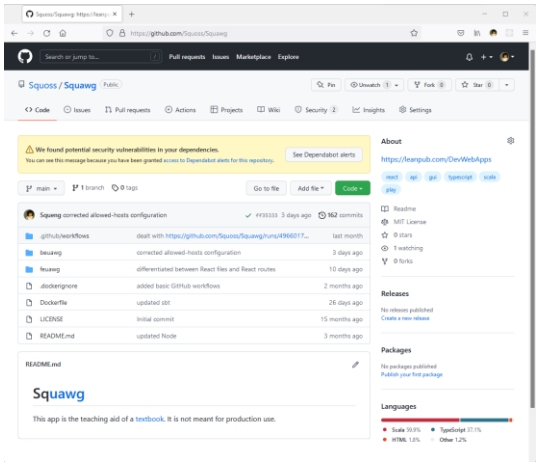
Bonus

- Habe weder zu Microsoft (GitHub, Visual Studio Code) noch zu SonarSource (SonarCloud, SonarLint) noch zu Snyk eine spezielle Beziehung (und hatte auch keine zu DeepCode).
- Es gibt zig Alternativen.

Squeng®

12

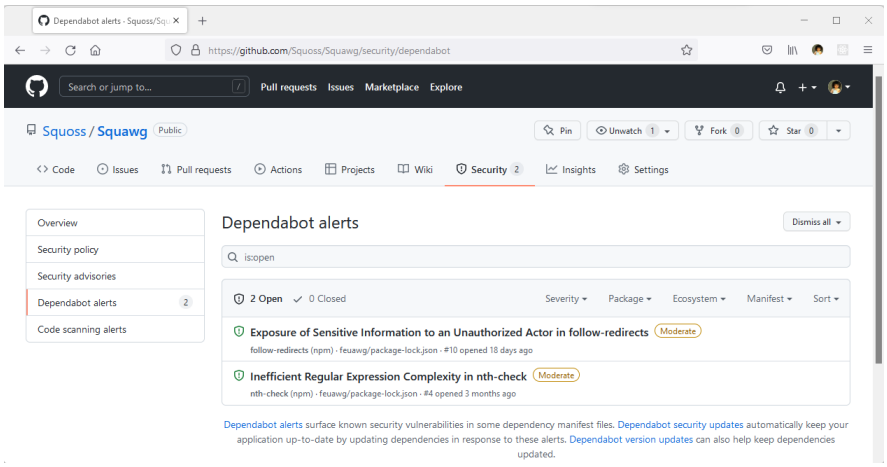
Project



Squeng®

13

Dependabot



Squeng®

14

Audit & Fix
(or ignore conciously & well-considered)

```
PS C:\Users\Paul\DevOps\Repositories\Squawg\feuawg> npm audit

                        === npm audit security report ===

# Run  npm update follow-redirects --depth 5  to resolve 1 vulnerability

  Moderate          Exposure of Sensitive Information to an Unauthorized Actor
                    in follow-redirects

  Package           follow-redirects

  Dependency of     react-scripts [dev]

  Path              react-scripts > webpack-dev-server > http-proxy-middleware >
                    http-proxy > follow-redirects

  More info         https://github.com/advisories/GHSA-pw2r-vq6v-hr8c


                                  Manual Review
  Dependency of     react-scripts [dev]

  Path              react-scripts > @svgr/webpack > @svgr/plugin-svgo > svgo >
                    css-select > nth-check

  More info         https://github.com/advisories/GHSA-rp65-9cf3-cjxr

found 2 moderate severity vulnerabilities in 1468 scanned packages
  run `npm audit fix` to fix 1 of them.
  1 vulnerability requires manual review. See the full report for details.
PS C:\Users\Paul\DevOps\Repositories\Squawg\feuawg> npm audit fix
npm WARN @apideck/better-ajv-errors@0.3.1 requires a peer of ajv@>=8 but none is installed. You must install peer dependencies yourself.

updated 1 package in 6.3s

166 packages are looking for funding
  run `npm fund` for details

fixed 1 of 2 vulnerabilities in 1468 scanned packages
  1 vulnerability required manual review and could not be updated
PS C:\Users\Paul\DevOps\Repositories\Squawg\feuawg>
```

Squeng®

15

Pipeline: CI and Continuous Delivery

Test

```yaml
name: Test
on:
  push:
    branches: [ '**' ]
  pull_request:
    branches: [ main ]
jobs:
  npm:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v2
    - name: Set up Node.js 16
      uses: actions/setup-node@v2
      with:
        node-version: '16'
        cache: 'npm'
        cache-dependency-path: feuawg/package-lock.json
    - name: Clean install with npm
      run: npm ci
      working-directory: feuawg
    - name: Test with npm
      run: npm test
      working-directory: feuawg
  sbt:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v2
    - name: Set up JDK 17
      uses: actions/setup-java@v2
      with:
        distribution: 'temurin'
        java-version: '17'
    - name: Test with sbt
      run: sbt test
      working-directory: beuawg
```

Deliver

```yaml
name: Deliver to Docker Hub

on:
  workflow_run:
    workflows: ["Scan"]
    types: [completed]
  workflow_dispatch:

jobs:
  deliver:
    runs-on: ubuntu-latest
    if: ${{ github.event_name == 'workflow_dispatch' || github.event.workflow_run.conclusion ==
'success' }}
    steps:
      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v1
      - name: Log in to Docker Hub
        uses: docker/login-action@v1
        with:
          username: ${{ secrets.DOCKER_HUB_USER }}
          password: ${{ secrets.DOCKER_HUB_TOKEN }}
      - name: Build and push
        id: docker_build
        uses: docker/build-push-action@v2
        with:
          push: true
          tags: squoss/squawg:latest
```

Squeng®

16

Pipeline: Continuous Deployment

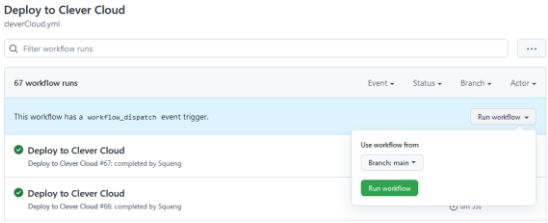
Deploy

```yaml
name: Deploy to Clever Cloud

on:
  workflow_run:
    workflows: ["Scan"]
    types: [completed]
  workflow_dispatch:

jobs:
  deploy:
    runs-on: ubuntu-latest
    if: ${{ github.event_name == 'workflow_dispatch' || github.event.workflow_run.conclusion == 'success' }}
    steps:
    - uses: actions/checkout@v2
      with:
        fetch-depth: 0
    - name: Set up Node.js 16
      uses: actions/setup-node@v2
      with:
        node-version: '16'
    - name: Install the Clever Tools
      run: npm install -g clever-tools
    - name: Link to the Clever Cloud account
      run: clever login --token ${{ secrets.CLEVER_TOKEN }} --secret ${{ secrets.CLEVER_SECRET }}
    - name: Link to the app
      run: clever link ${{ secrets.CLEVER_APP_ID }}
    - name: Deploy
      run: clever deploy
```

Dispatch



Squeng®

17

Pipeline: Scan

```yaml
name: Scan
on:
  workflow_run:
    workflows: ["Test"]
    types: [completed]
  schedule:
    - cron: '0 12 * * *'
jobs:
  Sonar:
    runs-on: ubuntu-latest
    if: ${{ github.event_name == 'schedule' || github.event.workflow_run.conclusion == 'success' }}
    steps:
      - uses: actions/checkout@v2
        with:
          fetch-depth: 0
      - name: back-end and front-end
        uses: sonarsource/sonarcloud-github-action@master
        with:
          args: >
            -Dsonar.organization=squoss
            -Dsonar.projectKey=Squoss_Squawg
            -Dsonar.sources=beuawg/app/,beuawg/reinraum/src/main/scala/,feuawg/src/
            -Dsonar.tests=beuawg/test/,beuawg/reinraum/src/test/scala/,feuawg/src/__tests__/
            -Dsonar.exclusions=feuawg/src/__tests__/**/*
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
          SONAR_TOKEN: ${{ secrets.SONAR_CLOUD_TOKEN }}
```

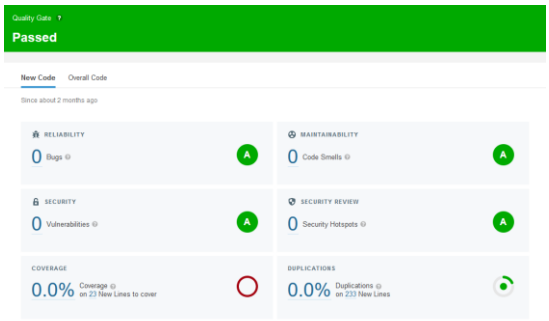
```yaml
Snyk:
  runs-on: ubuntu-latest
  if: ${{ github.event_name == 'schedule' || github.event.workflow_run.conclusion == 'success' }}
  steps:
    - uses: actions/checkout@v2
    - name: Run Snyk to check for TypeScript vulnerabilities
      continue-on-error: true
      id: coeTypeScript
      uses: snyk/actions/node@master
      env:
        SNYK_TOKEN: ${{ secrets.SNYK_AUTH_TOKEN }}
      with:
        args: |
          --sarif-file-output=feuawg.sarif
          --all-projects
          --exclude=beuawg
    - name: Upload result to GitHub Code Scanning
      uses: github/codeql-action/upload-sarif@v1
      with:
        sarif_file: feuawg.sarif
        category: feuawg
    - name: Run Snyk to check for Scala vulnerabilities
      continue-on-error: true
      id: coeScala
      uses: snyk/actions/scala@master
      env:
        SNYK_TOKEN: ${{ secrets.SNYK_AUTH_TOKEN }}
      with:
        args: |
          --sarif-file-output=beuawg.sarif
          --all-projects
          --exclude=feuawg
    - name: Upload result to GitHub Code Scanning
      uses: github/codeql-action/upload-sarif@v1
      with:
        sarif_file: beuawg.sarif
        category: beuawg
    - name: Check for failures
      if: ${{ steps.coeTypeScript.outcome != 'success' || steps.coeScala.outcome != 'success' }}
      run: exit 1
```

Squeng®

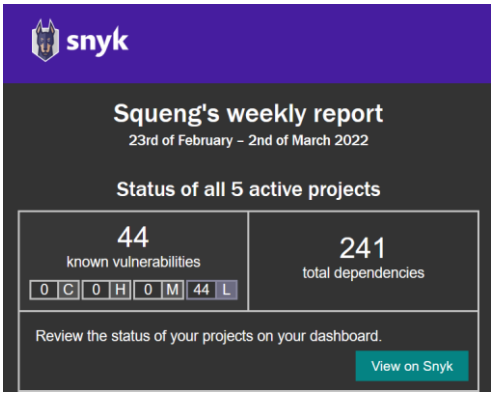
18

Results & Alerts

on a dashboard …



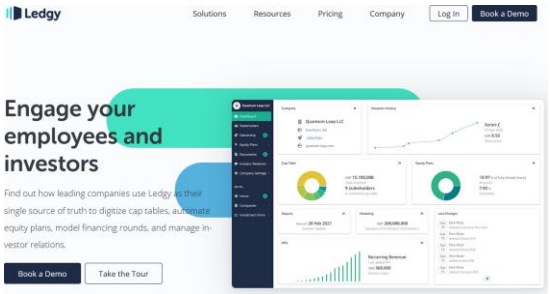
… or by e-mail

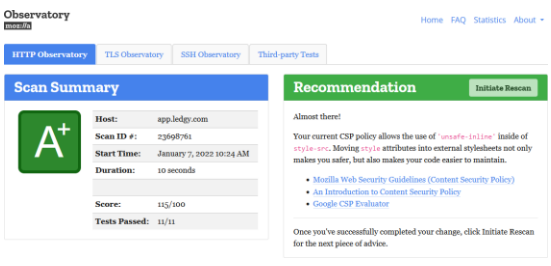


Squeng®

19

Trau, schau, wem!



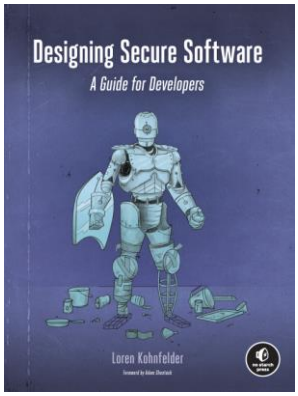


Squeng®

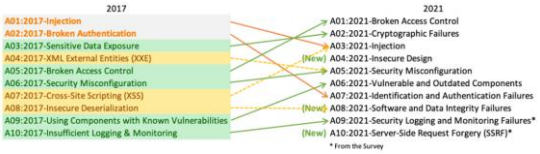
20

More Time for Features?

<<



OWASP Top 10



Squeng®

21

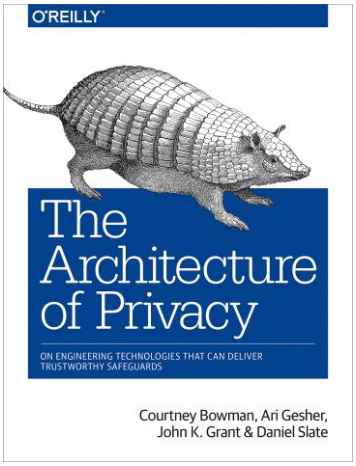
One more thing …

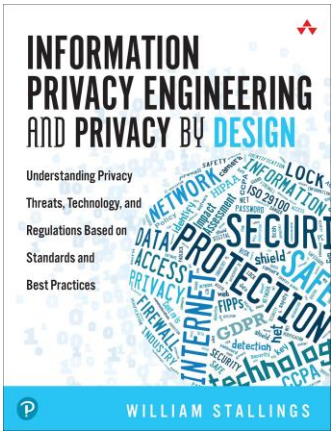
… or two

Squeng®

22

Privacy

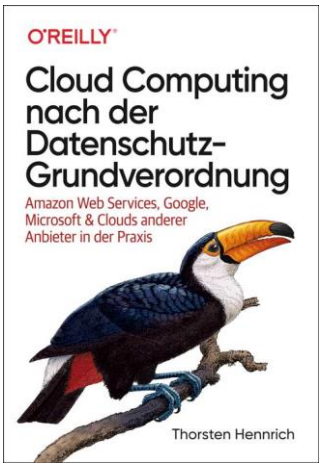




Squeng®

23

Data Protection



Squeng®

24