

Relatório de Ferramentas utilizadas para scans de segurança

Introdução

Este documento detalha a seleção estratégica de ferramentas para integrar práticas de segurança (Sec) ao ciclo de vida de desenvolvimento e operações (DevOps), estabelecendo uma esteira DevSecOps robusta.

O objetivo é automatizar a detecção de vulnerabilidades o mais cedo possível ("Shift-Left Security"), cobrindo diferentes vetores de ataque: segredos expostos, vulnerabilidades no código-fonte, vulnerabilidades em dependências de terceiros (Supply Chain) e configurações inseguras de contêineres.

As ferramentas selecionadas—Gitleaks, SonarQube (SonarCloud), Trivy e OWASP Dependency Tracker—foram escolhidas por sua complementaridade, capacidade de automação em CI/CD e cobertura abrangente das principais áreas de risco.

Mapeamento das Ferramentas no Ciclo DevSecOps

A estratégia visa cobrir as seguintes etapas do pipeline:

1. **Pré-Commit / Commit (Desenvolvimento):** Detecção de segredos (Gitleaks).
2. **Build / Integração Contínua (CI):**
 - Análise Estática de Segurança (SAST) e Qualidade de Código (SonarQube).
 - Análise de Composição de Software (SCA) e Scan de Imagens (Trivy).
3. **Monitoramento / Pós-Deploy:** Gerenciamento contínuo de vulnerabilidades (Dependency Tracker).

Detalhamento das ferramentas utilizadas

Secret Scanning

Ferramenta: Gitleaks + Trivy

Descrição da ferramenta

O Gitleaks é uma ferramenta de linha de comando que varre repositórios Git (incluindo todo o histórico de commits) em busca de segredos hardcoded, como chaves de API, senhas, tokens de acesso e chaves privadas.

Posicionamento no Pipeline

- **Ideal:** Hooks de Pré-Commit (impedindo o desenvolvedor de comitar o segredo).
- **Obrigatório:** Pipeline de CI (garantindo que nenhum segredo chegue ao repositório central).

Motivo da Escolha

- **Prevenção Primária:** Vazamento de credenciais é uma das falhas de segurança mais comuns e impactantes. O Gitleaks aborda esse risco na origem.
- **Velocidade e Eficiência:** É extremamente rápido e utiliza regras de alta entropia e expressões regulares para identificar segredos com precisão.
- **Abrangência Histórica:** Diferente de outras ferramentas que olham apenas o código atual, o Gitleaks varre o histórico. Um segredo comitado há 6 meses, mesmo que removido hoje, ainda está no histórico do Git e é um risco.
- **Facilidade de Integração:** Sendo um binário simples, é trivial adicionar em scripts de pre-commit e em qualquer sistema de CI (GitHub Actions, GitLab CI, Jenkins).

Evidência de execução

The screenshot displays a GitHub Actions workflow run for the repository 'test: add secrets for scanning demo'. The workflow is named '01-secret-scanning' and is triggered on 'pull_request'. The job 'Secret Scanning - trivy' is highlighted in blue. The job log shows the following steps:

```
1 Current runner version: '2.209.0'
2 Runner name: 'arc-runner-set-514t-runner-zvrx8'
3 Runner group name: 'default'
4 Machine name: 'arc-runner-set-514t-runner-zvrx8'
5 ► GITHUB_TOKEN Permissions
6 Secret source: Actions
7 Prepare workflow directory
8 Prepare all required actions
9 Getting action download info
10 Download action repository 'actions/checkout@v5' (SHA:08c6903cd8c8fde910a37f88322edc7b5dd907a8)
11 Download action repository 'aquasecurity/trivy-action@0.33.1' (SHA:b6643a29fecd7734b3597bc8acba98b03d337f8)
12 Download action repository 'github/codeql-action@v3' (SHA:d198d7fab99a7f38b5ce57ce70d4942944f006e)
13 Download action repository 'gitleaks/gitleaks-action@v2' (SHA:ff98106e4c7b26c287b24eaf42907106329070c7)
14 Getting action download info
15 Download action repository 'aquasecurity/setup-trivy@6c2c5e321ed9123bda567646e2f96565e34abe1' (SHA:66c2c5e321ed9123bda567646e2f96565e34abe1)
16 Download action repository 'actions/cache@409d5f644dc74513175e3cd807122d44860800' (SHA:0409d5f644dc74513175e3cd807122d44860800)
17 Getting action download info
18 Download action repository 'actions/checkout@08c6903cd8c8fde910a37f88322edc7b5dd907a8' (SHA:08c6903cd8c8fde910a37f88322edc7b5dd907a8)
```

Vulnerabilidades encontradas

← 01-secret-scanning

ci: add secret scanning #1

Summary

Jobs

Secret Scanning - trivy

Secret Scanning - gitleaks

Run details

Usage

Workflow file

Re-run triggered now

doug-cpp → 9f0e3bf main

Status

In progress

Total duration

=

Artifacts

1

01-secret-scanning.yaml

on: push

Matrix: secret-scanning

Secret Scanning - gitleaks 15s

Secret Scanning - trivy 46s

Secret Scanning - trivy summary

Trivy Secret Scanning Results

Target	Rule ID	Category	Severity	Start Line	End Line	Match
.env	gitlab-pat	GitLab	CRITICAL	1	1	REDACTED

Some checks were not successful

4 failing, 1 in progress, 2 successful checks

4 failing checks

01-secret-scanning / Secret Scanning - gitleaks (pull_request) Failing after 1m

01-secret-scanning / Secret Scanning - trivy (pull_request) Failing after 53s

02-sca / Vulnerability Scan (pull_request) Failing after 1m

Code scanning results / Trivy Failing after 4s — 22 new alerts including 6 critical severity security vulne...

1 in progress check

02-sca / SBOM Generation (pull_request) Started 2 minutes ago — This check has started...

2 successful checks

02-sca / Dependency Graph (pull_request) Successful in 3m

No conflicts with base branch

Merging can be performed automatically.

Merge pull request

You can also merge this with the command line. [View command line instructions.](#)

Add a comment

Write

Preview

Add your comment here...

Markdown is supported

Paste, drop, or click to add files

2 references

The job failed because Trivy detected secrets in your repository and the workflow is configured to fail when secrets are found (Trivy runs with exit-code: 1). To resolve this:

1. Review the Trivy output (trivy-results.json and trivy-results.sarif) to identify files and lines where secrets were flagged.

2. Remove or rotate any sensitive data found. Replace hardcoded secrets with environment variables or use secret management tools.

3. After cleaning, commit your changes.

You can adjust the strictness by modifying the workflow, but the secure solution is to eliminate secrets from your codebase.

Relevant workflow: [.github/workflows/01-secret-scanning.yaml \(ref\)](#)

Ask anything

+ Add repositories, files, and spaces

```

17:21:40 ~/school/proj-final (test-secret-scanning*) $ git add .
(base)
17:21:47 ~/school/proj-final (test-secret-scanning*) $ git commit -m 'test: add dummy secrets do trigger gitleaks'
[test-secret-scanning b1348a1] test: add dummy secrets do trigger gitleaks
1 file changed, 18 insertions(+)
create mode 100644 my-tokens.env
(base)
17:22:27 ~/school/proj-final (test-secret-scanning) $ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 681 bytes | 681.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote: error: GH013: Repository rule violations found for refs/heads/test-secret-scanning.
remote:
remote: - GITHUB PUSH PROTECTION
remote:
remote: _____
remote: |
remote: | Resolve the following violations before pushing again
remote: |
remote: | - Push cannot contain secrets
remote: |
remote: | (?) Learn how to resolve a blocked push
remote: | https://docs.github.com/code-security/secret-scanning/working-with-secret-scanning-and-push-protection/working-
remote: |
remote: | (?) This repository does not have Secret Scanning enabled, but is eligible. Enable Secret Scanning to view and
remote: | Visit the repository settings page, https://github.com/doug-cpp-devsecops/proj-final/settings/security\_analysis
remote: |
remote: | _____ Slack Incoming Webhook URL _____
remote: | locations:
remote: |   - commit: b1348a13c526ec2c6984f5db54285b9fd52ce48
remote: |     path: my-tokens.env:9
remote: |
remote: | (?) To push, remove secret from commit(s) or follow this URL to allow the secret.
remote: | https://github.com/doug-cpp-devsecops/proj-final/security/secret-scanning/unblock-secret/34WlmHkzZAKxsPyaB5et
remote: |
remote: | To github.com:doug-cpp-devsecops/proj-final.git
remote: | [remote rejected] test-secret-scanning -> test-secret-scanning (push declined due to repository rule violations)
error: failed to push some refs to 'github.com:doug-cpp-devsecops/proj-final.git'
(base)

```



Push blocked because a secret was detected

! [Secret scanning](#) found a **Slack Incoming Webhook URL** secret in your attempted push.

Allowing this secret risks exposure. Instead, consider [removing the secret from your commit and commit history](#).

Exposing this secret can allow someone to:

- Verify the identity of this **Slack Incoming Webhook URL** secret
- Know which resources this secret can access
- Act on behalf of the secret's owner

Push protection is enabled for your account. To disable, visit your [code security settings](#).

Allow me to expose this secret

Correção da vulnerabilidade

```
gemen
2
17:01:14 ~/school/proj-final (main) $ git checkout -b test-secret-scanning
Switched to a new branch 'test-secret-scanning'
(base)
17:03:17 ~/school/proj-final (test-secret-scanning) $ cat > test-secrets.env << 'EOF'
AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
GITHUB_TOKEN="ghp_1234567890abcdefghijklmnopqrstuvwxyz"
DATABASE_PASSWORD="super-secret-password-123"
EOF
(base)
17:04:11 ~/school/proj-final (test-secret-scanning*) $ git status
On branch test-secret-scanning
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test-secrets.env

nothing added to commit but untracked files present (use "git add" to track)
(base)
17:04:15 ~/school/proj-final (test-secret-scanning*) $ git add .
(base)
17:04:27 ~/school/proj-final (test-secret-scanning*) $ git commit -m 'test: add secrets for scanning demo'
[test-secret-scanning 87flc90] test: add secrets for scanning demo
1 file changed, 3 insertions(+)
 create mode 100644 test-secrets.env
(base)
17:04:52 ~/school/proj-final (test-secret-scanning) $ git push
```



Secret allowed

You can now push this secret to the repository.

SCA (Software Composition Analysis)

Ferramenta: Dependency Track

Descrição da ferramenta

O Dependency Tracker (DT) é uma plataforma de análise de composição. Aonde seu foco é *gerenciar e monitorar* componentes de software ao longo do tempo.

Posicionamento no Pipeline

- Etapa de Pós-Build (CI) ou Pós-Deploy (CD). O pipeline deve gerar um SBOM e enviá-lo para o Dependency Tracker.

Motivo da Escolha

- **Visão Centralizada e Contínua:** O Trivy e o SonarQube nos dizem se o *build atual* é seguro. O Dependency Tracker nos diz se os *projetos em produção* estão seguros. Ele centraliza o inventário de componentes de *todos* os projetos da organização.
- **Monitoramento de "Novas" CVEs:** Esta é sua maior força. Uma biblioteca pode estar segura hoje, mas uma nova CVE pode ser descoberta para ela amanhã. O Dependency Tracker monitora ativamente os componentes e alerta proativamente quando um componente *já em produção* se torna vulnerável.
- **Análise de Risco de Licença:** Além de segurança, ele analisa as licenças de software (ex: GPL, MIT, Apache) das dependências, identificando riscos legais ou de conformidade.
- **Foco em SBOM (Supply Chain Security):** A plataforma é construída em torno do conceito de SBOM (como CycloneDX ou SPDX), que é o padrão emergente da indústria para transparência e segurança da cadeia de suprimentos de software.

Evidência de execução

Summary

Jobs

SBOM Generation

Dependency Graph

Vulnerability Scan

Run details

Usage

Workflow file

SBOM Generation

succeeded 3 minutes ago in 24s

Generate SBOM File with Syft

16s

Upload SBOM to Dependency Track

8s

Post Checkout code

9s

Complete job

0s

Search logs

23 [debug] http_download(url=https://github.com/anchore/syft/releases/download/v1.33.0/syft_1.33.0_checksums.txt)

24 [debug]

25 http_download(url=https://github.com/anchore/syft/releases/download/v1.33.0/syft_1.33.0_linux_amd64.tar.gz)

26 [info] installed /home/runner/_work/_temp/0ebb7b82-7aff-48c2-a910-1f224bd64b8f_syft/syft

27 /home/runner/_work/_temp/0ebb7b82-7aff-48c2-a910-1f224bd64b8f_syft/syft

28 [info] scan dir:.. -o cyclonedx

29 Executing Syft...

30 [0000] WARN no explicit name and version provided for directory source, deriving artifact ID from the given path (which is not ideal)

31 SBOM scan completed in: 13.506s

32 ----- Uploading workflow artifacts -----

33 /tmp/sbom-action-KqRLd6/sbom-syft.json

1 Run DependencyTrack/gh-upload-sbom@v3

2 Reading BOM: sbom-syft.json...

3 Uploading to Dependency-Track server dependencytrack-dependency-track-api-server.dependencytrack...

4 Finished uploading BOM to Dependency-Track server.

1 Post job cleanup.

2 /usr/bin/git version

3 git version 2.51.0

4 Temporarily overriding HOME='/home/runner/_work/_temp/83d3c5ee-9cb2-4d85-a898-47bbd1233ac0' before making global git config changes

5 Adding repository directory to the temporary git global config as a safe directory

6 /usr/bin/git config --global --add safe.directory /home/runner/_work/proj-final/proj-final

7 /usr/bin/git config --local --name-only --get-regexp core.sshCommand

8 /usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'core.sshCommand' && git config --local --unset-all 'core.sshCommand' || :"

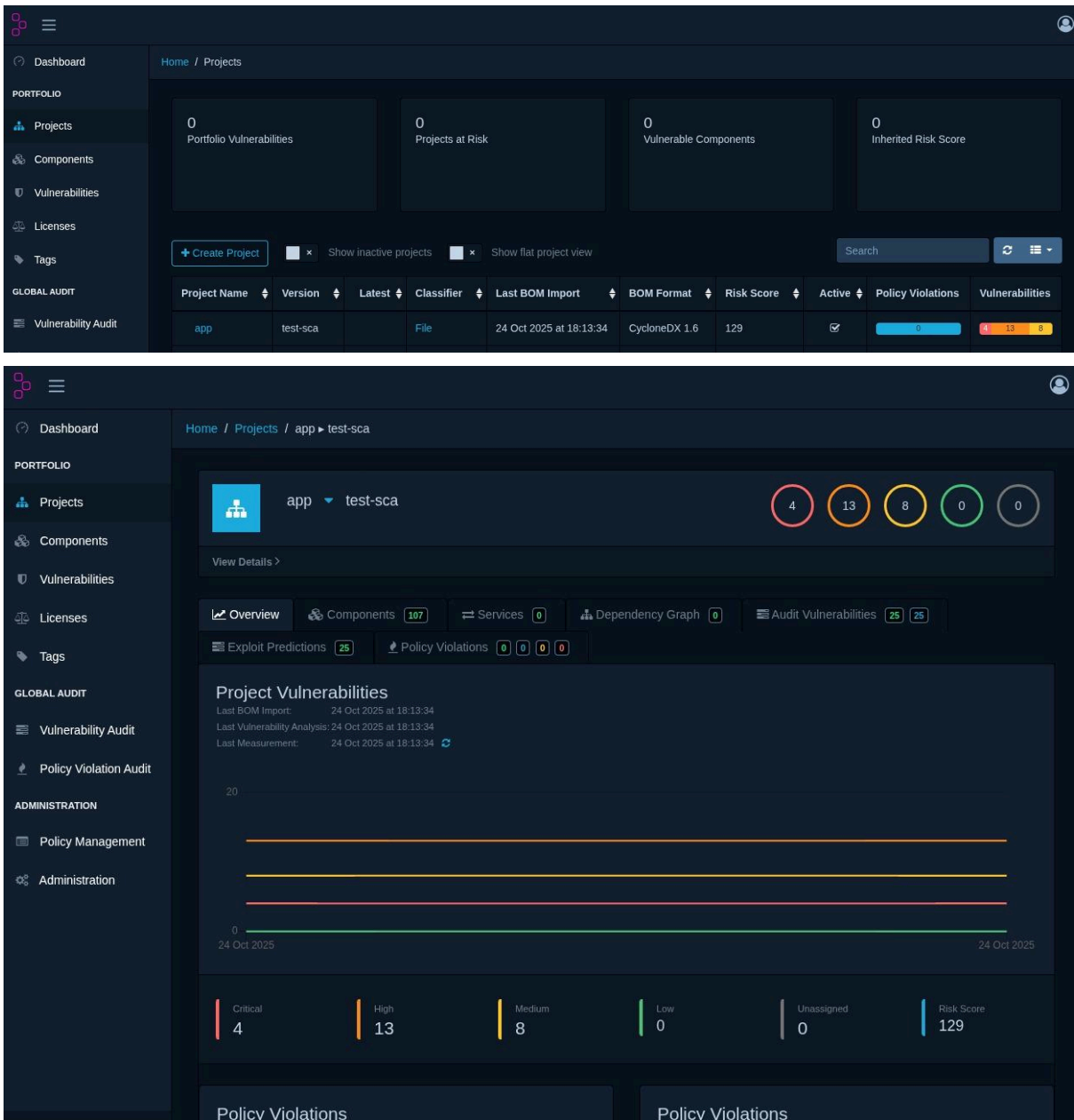
9 /usr/bin/git config --local --name-only --get-regexp http.https://github.com/.extraheader

10 http.https://github.com/.extraheader

11 /usr/bin/git config --local --unset-all http.https://github.com/.extraheader

12 /usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'http.https://github.com/.extraheader' && git config --local --unset-all 'http.https://github.com/.extraheader' || :"

Vulnerabilidades encontradas



	Component	Version	Group	Vulnerability	Severity	Analyzer	Attributed On	Analysis	Suppre
>	axios	0.21.1		NVD CVE-2025-27152	High	NVD	24 Oct 2025	-	
>	axios	0.21.1		NVD CVE-2025-58754	High	NVD	24 Oct 2025	-	
>	axios	0.21.1		NVD CVE-2024-57965	Critical	NVD	24 Oct 2025	-	
>	axios	0.21.1		NVD CVE-2021-3749	High	NVD	24 Oct 2025	-	
>	body-parser	1.13.3		NVD CVE-2024-45590	High	NVD	24 Oct 2025	-	
>	debug	2.2.0		NVD CVE-2017-16137	Medium	NVD	24 Oct 2025	-	
>	debug	2.2.0		NVD CVE-2017-20165	High	NVD	24 Oct 2025	-	
>	ejs	0.8.8		NVD CVE-2017-1000188	Medium	NVD	24 Oct 2025	-	
>	ejs	0.8.8		NVD CVE-2017-1000189	High	NVD	24 Oct 2025	-	
>	ejs	0.8.8		NVD CVE-2017-1000228	Critical	NVD	24 Oct 2025	-	

SAST (Static Analysis)

Ferramenta: SonarQube (SonarCloud)

O SonarQube (auto-hospedado) ou SonarCloud (SaaS) realiza uma análise estática profunda do código-fonte. Ele identifica não apenas "Code Smells" (má qualidade) e "Bugs", mas também "Security Hotspots" e "Vulnerabilities" (falhas de segurança).

Posicionamento no Pipeline

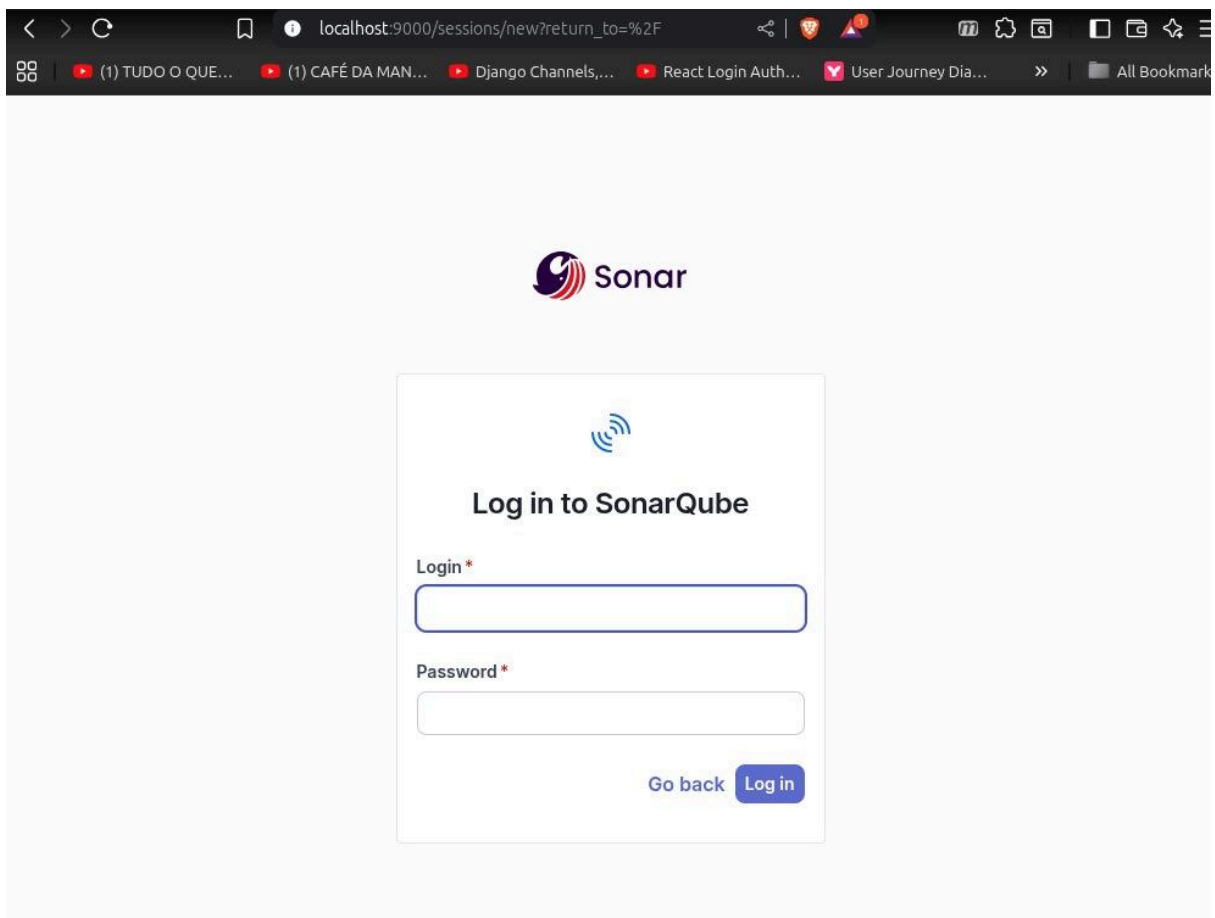
- Pipeline de CI, integrado ao Pull Request (PR) / Merge Request (MR).

Motivo da Escolha

- **Análise de Lógica de Negócio:** Enquanto outras ferramentas focam em dependências, o SonarQube foca no *código que a equipe escreve*. Ele é essencial para encontrar falhas como SQL Injection, Cross-Site Scripting (XSS), e lógica de controle de acesso insegura.

- **Quality Gates:** Permite a definição de "Portões de Qualidade" (Quality Gates) que podem *falhar o build* se novos problemas de segurança ou qualidade forem introduzidos, garantindo um padrão mínimo antes da mesclagem de código.
- **Gerenciamento de Dívida Técnica:** Fornece um dashboard claro sobre a saúde do projeto, ajudando a priorizar a correção de vulnerabilidades e a gerenciar a dívida técnica de segurança ao longo do tempo.
- **Ampla Cobertura de Linguagens:** Suporta uma vasta gama de linguagens de programação, tornando-se um padrão de mercado.

Evidência de execução



Update your password

⚠ This account should not use the default password.

Enter a new password

Old Password *

.....

Password *

.....

Confirm Password *

.....

Update

New Code

Overall Code

Security

3 Open issues

E

Reliability

0 Open issues

A

Maintainability

267 Open issues

A

Accepted issues

0

Valid Issues that were not fixed

B

Coverage

0.0%

On 781 lines to cover.

C

Duplications

0.0%

On 4.4k lines.

D

Security Hotspots

15

E

Vulnerabilidades encontradas

My Issues All

Filters

Looking for Bugs, Vulnerabilities, or Code Smells? If your team prefers working with these types, change it in the [settings](#)

Issues in new code

Software Quality

Security 3

Reliability 0

Maintainability 267

Severity

Blocker 4

High 217

Medium 32

Low 17

Open Not assigned L5 5min effort 14 minutes ago

Unexpected var, use let or const instead.

Maintainability High

es2015 bad-practice

Open Not assigned L6 5min effort 14 minutes ago

Unexpected var, use let or const instead.

Maintainability High

es2015 bad-practice

Open Not assigned L7 5min effort 14 minutes ago

Unexpected var, use let or const instead.

Maintainability High

es2015 bad-practice

Open Not assigned L8 5min effort 14 minutes ago

Unexpected var, use let or const instead.

Maintainability High

es2015 bad-practice

SonarQube community

Projects Issues Rules Quality Profiles Quality Gates Administration More

Search ? A

DevSecOps App Bind project / main

Overview Issues Security Hotspots Code Measures Activity

Project Settings Project Information

My Issues All

Filters

Looking for Bugs, Vulnerabilities, or Code Smells? If your team prefers working with these types, change it in the [settings](#)

Issues in new code

Software Quality

Security 3

Reliability 0

Maintainability 1

Severity

Blocker 4

High 217

Medium 32

Low 17

Bulk Change

Select issues

Navigate to issue

4 issues 1h 32min effort

app/config.js

Make sure this PostgreSQL database password gets changed and removed from the code.

Security Blocker

cwe

Open Not assigned L4 30min effort 15 minutes ago

Make sure this PostgreSQL database password gets changed and removed from the code.

Security Blocker

cwe

Open Not assigned L12 30min effort 15 minutes ago

Make sure this PostgreSQL database password gets changed and removed from the code.

Security Blocker

cwe

Open Not assigned L20 30min effort 15 minutes ago

app/public/js/freewall.js

Add the "let", "const" or "var" keyword to this declaration of "Sitem" to make it explicit.

Intentionality

0.0% Security Hotspots Reviewed

Make sure that using this pseudorandom number generator is safe here.

Make sure that using this pseudorandom number generator is safe here.

Make sure that using this pseudorandom number generator is safe here.

Review priority: Low

Insecure Configuration 1

Make sure creating this cookie without the "secure" flag is safe.

1 extra location

Location 1

Others 3

15 of 15 shown

Where	What	Assess	How	Activity
-------	------	--------	-----	----------

```

33 app.set('views', path.join(__dirname, 'views'));
34 app.set('view engine', 'ejs');
35
36 // uncomment after placing your favicon in /public
37 app.use(logger('combined', {stream: accessLogStream}));
38 app.use(bodyParser());
39 app.use(bodyParser.json());
40 app.use(bodyParser.urlencoded({ extended: true }));
41 app.use(cookieParser());
42 app.use(express.static(path.join(__dirname, 'public')));
43 app.use(session({
44
45   secret: 'hasddfilhpaf78h78032h780g780fg780asg780dsbovnucbuyvqy',
46   cookie: {
47     secure: false,
48     maxAge: 9999999999
49   }
50 }));
51
52 /*
53  * Routes config
54  */
55 app.use('', products);
56 app.use('', login);
57

```

Make sure creating this cookie without the "secure" flag is safe.

0.0% Security Hotspots Reviewed

Permission 2

The "node" image runs with "root" as the default user. Make sure it is safe here.

Copying recursively might inadvertently add sensitive data to the container. Make sure it is safe here.

Weak Cryptography 6

Make sure that using this pseudorandom number generator is safe here.

Make sure that using this pseudorandom number generator is safe here.

Make sure that using this pseudorandom number generator is safe here.

Make sure that using this pseudorandom number generator is safe here.

Make sure that using this pseudorandom number generator is safe here.

Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review

Where is the risk?	What's the risk?	Assess the risk	How can I fix it?	Activity
--------------------	------------------	-----------------	-------------------	----------

app/dummy.js

Open in IDE

```

10 {
11   "username": "roberto",
12   "password": "asdfpiuw981"
13 }
14 ],
15
16 "products": [
17 {
18   "name": "My public privacy",
19   "description": "Grant privacy in public to watch your favorite programs",
20   "price": parseInt(Math.random() * 100),
21
22   "image": "product_1.jpg"
23 },
24 {
25   "name": "The USB rocket",
26   "description": "Be happy with your USB rocket. Functionality: none. Usability: none. The best choice!",
27   "price": parseInt(Math.random() * 100),
28   "image": "product_2.jpg"
29 }
30 ]
31 }
32

```

Make sure that using this pseudorandom number generator is safe here.

SonarQube community

Projects Issues Rules Quality Profiles Quality Gates Administration More

DevSecOps App Bind project / main

Overview Issues Security Hotspots Code Measures Activity Project Settings Project Information

My Issues All

Filters Clear All Filters

Looking for Bugs, Vulnerabilities, or Code Smells? If your team prefers working with these types, change it in the [settings](#)

Issues in new code

Software Quality

Security	0
Reliability	0
Maintainability	17

Severity

Blocker	4
High	217
Medium	32
Low	17

Bulk Change Select issues Navigate to issue 17 issues 1h 35min effort

app/Dockerfile

Merge this RUN instruction with the consecutive ones. Consistency No tags

Maintainability Low

Open Not assigned L7 - 5min effort - 16 minutes ago

app/model/init_db.js

Expected a 'for-of' loop instead of a 'for' loop with this simple iteration. Consistency clumsy

Maintainability Low

Open Not assigned L22 - 5min effort - 16 minutes ago

app/public/js/freewall.js

'webkitTransition' is deprecated. Consistency cwe type-dependent

Maintainability Low

Open Not assigned L804 - 15min effort - 16 minutes ago

Detalhes das vulnerabilidades

0.0% Security Hotspots Reviewed

Authentication

2

Review this potentially hard-coded password.

Review this potentially hard-coded password.

Denial of Service (DoS)

1

Make sure the regex used here, which is vulnerable to super-linear runtime due to backtracking, cannot lead to denial of service.

Permission

2

The "node" image runs with "root" as the default user. Make sure it is safe here.

Copying recursively might inadvertently add sensitive data to the container. Make sure it is safe here.

Weak Cryptography

6

due to backtracking, cannot lead to denial of service.

Using slow regular expressions is security-sensitive `javascript:S5852`

Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review

Where is the risk?

What's the risk?

Assess the risk

How can I fix it?

Activity

app/routes/products.js

Open in IDE

```
110         address: params.address,
111         ship_date: params.ship_date,
112         phone: params.phone,
113         product_id: params.product_id,
114         product_name: params.product_name,
115         username: req.session.user_name,
116         price: params.price.substr(0, params.price.length - 1) // remove "€" symbol
117     }
118
119     // Check mail format
120     var re =
121
122     /^(?=[a-zA-Z0-9_!@#$%^&*~`{|}~\.\-\/\+?]{1,254})$/
```

Make sure the regex used here, which is vulnerable to super-linear runtime due to backtracking, cannot lead to denial of service.

```
121     if (!re.test(cart.mail)){
122         throw new Error("Invalid mail format");
123     }
```

0.0% Security Hotspots Reviewed ⓘ

Make sure that using this pseudorandom number generator is safe here.

Review priority: Low

Insecure Configuration

1

Make sure creating this cookie without the "secure" flag is safe.

1 extra location

Others

3

This framework implicitly discloses version information by default. Make sure it is safe here.

Omitting "--ignore-scripts" can lead to the execution of shell scripts. Make sure it is safe here.

Make sure automatically installing recommended packages is safe here.

15 of 15 shown

Where is the risk?What's the risk?Assess the riskHow can I fix it?Activity

app/Dockerfile

Open in IDE

5ENV STAGE "DOCKER"

6

7RUN apt-get update && apt-get install -y netcat

8

9# Build app folders

10RUN mkdir /app

11WORKDIR /app

12

13# Install depends

14COPY package.json /app/

15RUN npm install

16

17# Bundle code

18COPY . /app

19

20EXPOSE 3000

21

22CMD ["npm", "start"]

23

Omitting "--ignore-scripts" can lead to the execution of shell scripts. Make sure it is safe here.

0.0% Security Hotspots Reviewed

Make sure that using this pseudorandom number generator is safe here.

Review priority: Low

Insecure Configuration

Make sure creating this cookie without the "secure" flag is safe.

1 extra location

Others

This framework implicitly discloses version information by default. Make sure it is safe here.

Omitting "--ignore-scripts" can lead to the execution of shell scripts. Make sure it is safe here.

Make sure automatically installing recommended packages is safe here.

15 of 15 shown

Where is the risk? What's the risk? Assess the risk How can I fix it? Activity

app/Dockerfile Open in IDE

```
5 ENV STAGE "DOCKER"
6
7 RUN apt-get update && apt-get install -y netcat
8
9 # Build app folders
10 RUN mkdir /app
11 WORKDIR /app
12
13 # Install depends
14 COPY package.json /app/
15 RUN npm install
16
17 # Bundle code
18 COPY . /app
19
20 EXPOSE 3000
21
22 CMD [ "npm", "start" ]
23
```

Omitting "--ignore-scripts" can lead to the execution of shell scripts. Make sure it is safe here.

0.0% Security Hotspots Reviewed

To review Acknowledged Fixed Safe

15 Security Hotspots to review

Review priority: High

Authentication

Review this potentially hard-coded password.

Review this potentially hard-coded password.

Denial of Service (DoS)

Make sure the regex used here, which is vulnerable to super-linear runtime due to backtracking, cannot lead to denial of service.

Permission

The "node" image runs with "root" as the default user. Make sure it is safe here.

Review this potentially hard-coded password.

Hard-coded passwords are security-sensitive `javascript:S2068`

Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review

Where is the risk? What's the risk? Assess the risk How can I fix it? Activity

Copy the file path to the clipboard

app/dummy.js Open in IDE

```
1 // This file contains dummy information data
2
3 var dummy_info = {
4   // Customer module configs
5   "users": [
6     {
7       "username": "admin",
8       "password": "admin"
9     },
10    {
11      "username": "roberto",
12      "password": "asdfpluw981"
13    }
14  ],
15
```

Review this potentially hard-coded password.

Conclusão

Nenhuma das ferramentas listadas é redundante, ou seja, elas são complementares e cobrem diferentes estágios e tipos de risco:

- **GitLeaks** protege contra o vazamento de *credenciais*.
- **SonarQube** protege contra falhas no *código-fonte escrito pela equipe* (SAST).
- **Dependency Tracker** protege contra vulnerabilidades conhecidas no *supply chain* (SCA) de forma *contínua e centralizada*, mesmo após o deploy.

A adoção conjunta dessas quatro ferramentas fornece uma cobertura de segurança automatizada, multicamada e alinhada com as práticas de "Shift-Left". Elas permitem que a equipe de desenvolvimento identifique e corrija falhas de segurança (sejam elas segredos, bugs de lógica ou dependências vulneráveis) de forma rápida e eficiente dentro do fluxo de trabalho de CI/CD.