

**Дипломная работа по профессии  
«Специалист по информационной  
безопасности»**

**Track DevSecOps**

**Студент группы SIB-13**

**Ярмоленко Дмитрий Владимирович**

**2023**

# ОГЛАВЛЕНИЕ

Задача.....	3
Исходные данные .....	3
Этапы проектирования .....	4
Этап 1. CI/CD .....	5
Этап 2. SAST.....	8
Этап 3. DAST .....	11
Этап 4. Security Checks .....	16
Выводы .....	21
Приложения: .....	22

## Задача

**Ваша задача** — создать безопасный пайплайн для open-source проекта. Он должен включать в себя статический анализатор, динамический анализатор, чекеры безопасности, Security Gateway и документацию процесса.

Вы должны сами выбрать проект, для которого будет выстроен пайплайн. Платформа для организации CI/CD тоже на ваш выбор. Рекомендуем взять за основу GitLab CI/CD, GitHub Actions, CircleCI. Сервер для разворачивания даёт дипломный руководитель или методист в виде VPS.

## Исходные данные

1. Требования к проекту. Проект должен быть с открытым исходным кодом, представлять из себя веб-сервис или сайт с функционалом, использованием баз данных или кеша. Например, можно взять за основу [Defect Dojo](#) или [CMS, Netlify-CMS](#).
2. Требования к покрытию проекта тестами безопасности. Проект должен проверяться на наличие уязвимостей в коде. Ни один язык программирования или фреймворк не должны быть пропущены для конкретного проекта. Весь процесс должен быть задокументирован и описан с аналитикой выбора инструментов и зон роста.

## **Этапы проектирования**

### **Этап 1. CI/CD**

Критерии достижения:

1. Настроенный пайплайн по сборке и доставке программного обеспечения.
2. Использование облачных сервисов для раскатки.
3. Хорошо задокументированный процесс.

### **Этап 2. SAST**

Критерии достижения:

1. Покрытие кода проверками.
2. Успешные проверки во время сборки.
3. Выгрузка результатов в CI или систему менеджмента уязвимостей.

### **Этап 3. DAST**

Критерии достижения:

1. Покрытие сервиса проверками.
2. Успешные сканы по всем имеющимся методам.
3. Выгрузка результатов в CI или систему менеджмента уязвимостей.

### **Этап 4. Security Checks**

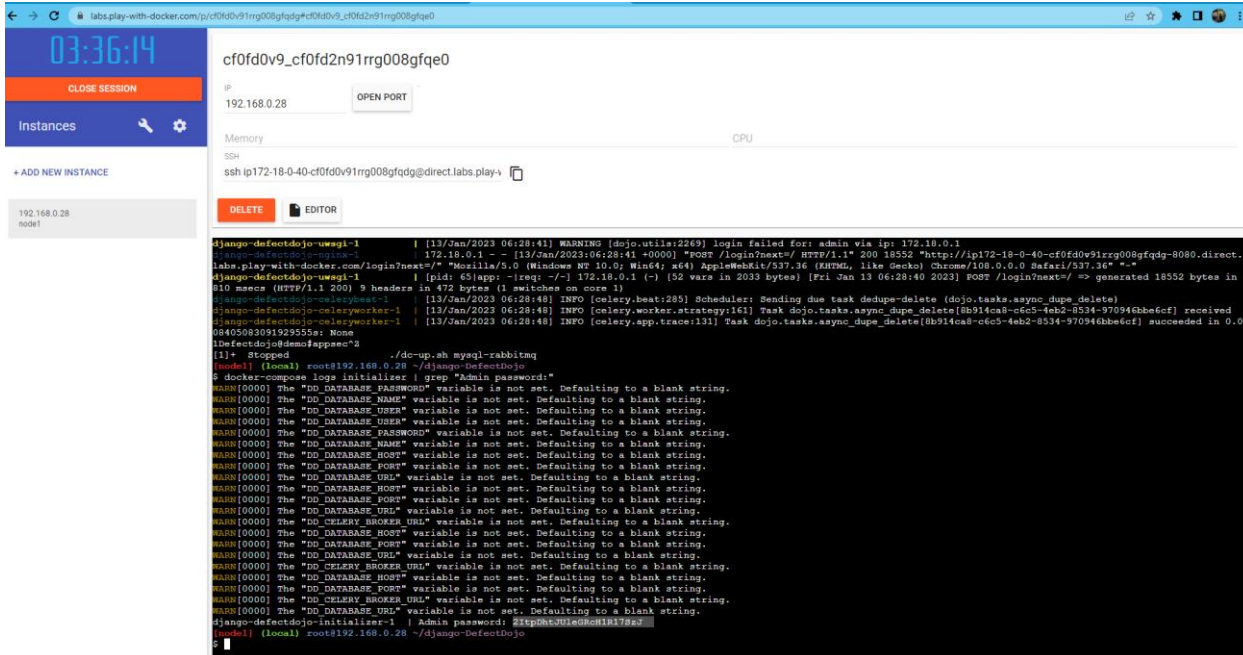
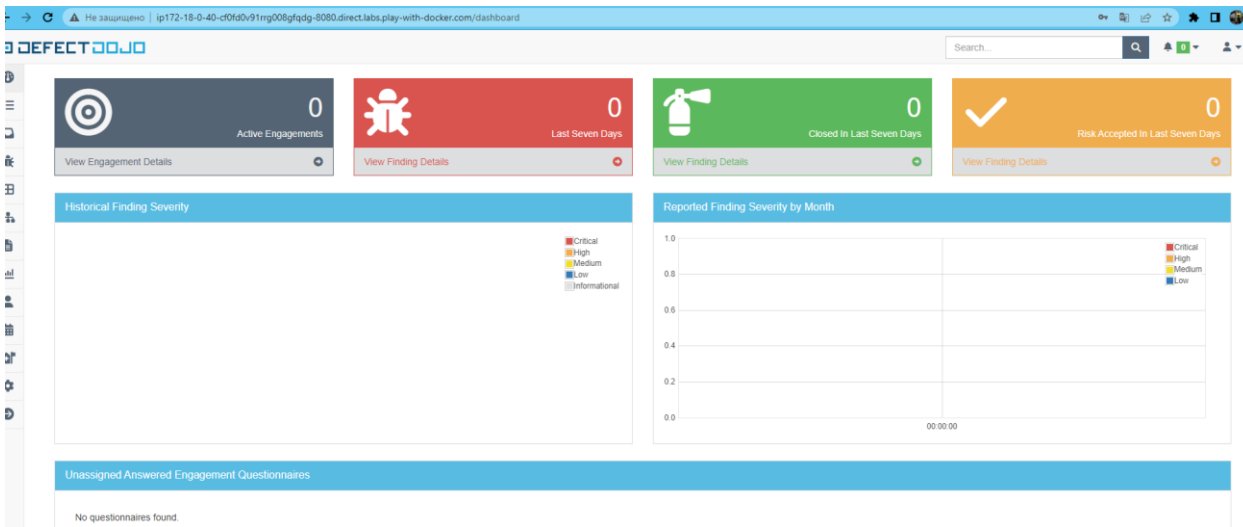
Критерии достижения:

1. Проверка репозиториев на секреты.
2. Проверка конфигурации или образов.

## Этап 1. CI/CD

Задача CI/CD — сократить время, необходимое для доставки ПО пользователям, без ущерба для качества. Чтобы этого добиться, необходимо регулярно проверять наличие изменений, тщательно их тестировать и быстро обрабатывать полученную обратную связь, чтобы внедрять изменения как можно чаще.

## Запуск приложения в PlayDocker.



Делаем форк в GitLab, на нем будем тестировать build и deploy.

fix

3 jobs for **master** in 4 minutes and 22 seconds (queued for 2 seconds)

**latest**

**b25885d5**

No related merge requests found.

Pipeline Needs Jobs **3** Tests **0**

Group jobs by Stage Job dependencies

Version	Build	Deploy
Generate Version	Built Docker Images	Deploy

Деплой:

```
1 Running with gitlab-runner 12.8.0 (1b659122)
2   on Docker runner 2 on Linux PMRAdoUk
3 Using Docker executor with image mid-registry.moduldev.ru/deploy-tools/ansible:6.0 ...
4 Authenticating with credentials from /root/.docker/config.json
5 Using locally found image version due to if-not-present pull policy
6 Using docker image sha256:af56baf013df65be8a1e52e8793ce62f09a4494303d3db10284e9935cd228795 for mid-registry.moduldev.ru/deploy-tools/ansible:6.0 ...
7 Authenticating with credentials from /root/.docker/config.json
8 Running on runner-PMRAdoUk-project-1476-concurrent-0 via urudc5ap090.brc.local...
9 Authenticating with credentials from /root/.docker/config.json
10 Fetching changes with git depth set to 20...
11 Reinitialized existing Git repository in /builds/poroshkinaa/django-DefectDojo/.git/
12 Checking out 3585c742 as master...
13 Removing build.env
14 Removing gl-sast-report.json
15 Skipping Git submodules setup
16 Authenticating with credentials from /root/.docker/config.json
17 Authenticating with credentials from /root/.docker/config.json
18 Downloading artifacts for Built Docker Images (3472880)...
19 Downloading artifacts from coordinator... ok id=3472880 responseStatus=200 OK token=JBT9xsHf
20 Authenticating with credentials from /root/.docker/config.json
21 $ eval $(ssh-agent -s)
22 Agent pid 11
23 $ (echo "$DEVOPS_SSH_PRIVATE_KEY" | base64 --decode) | ssh-add -
24 Identity added: (stdin) (root@urudc5ap349.brc.local)
25 $ mkdir -p ~/.ssh
26 $ chmod 700 ~/.ssh
27 $ echo "$DEVOPS_SSH_PRIVATE_KEY" > ~/.ssh/known_hosts
28 $ chmod 644 ~/.ssh/known_hosts
29 $ ansible-playbook -i devops/deploy/inventories/${STAGE_NAME} devops/deploy/deploy_palybook.yml --extra-vars "docker_registry=${DOCKER_REGISTRY}" --extra-vars "docker_project_na
me=${DOCKER_PROJECT_NAME}" --extra-vars "version=${BUILD_VERSION}" --extra-vars "serial_count=2" --vault-password-file ${key_vault}
```

```

35 PLAY [dojo_service] *****
36 Thursday 19 January 2023 07:18:52 +0000 (0:00:00.052) 0:00:00.052 *****
37 TASK [deploy_compose : Create dir for settings.d and keys] *****
38 ok: [172.21.22.184] => (item=docker/extra_settings)
39 Thursday 19 January 2023 07:18:55 +0000 (0:00:02.904) 0:00:02.956 *****
40 TASK [deploy_compose : copy tempaltes] *****
41 changed: [172.21.22.184] => (item={'src': 'docker_compose.j2', 'dest': 'docker-compose.yml'})
42 ok: [172.21.22.184] => (item={'src': 'docker_compose_env.j2', 'dest': 'docker_compose_env.env'})
43 Thursday 19 January 2023 07:18:57 +0000 (0:00:01.983) 0:00:04.940 *****
44 TASK [deploy_compose : Deploy "dojo_service"] *****
45 changed: [172.21.22.184]
46 Thursday 19 January 2023 07:19:46 +0000 (0:00:49.175) 0:00:54.116 *****
47 TASK [deploy_compose : Removing all inactive docker images] *****
48 changed: [172.21.22.184]
49 PLAY RECAP *****
50 172.21.22.184 : ok=4 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
51 Thursday 19 January 2023 07:19:57 +0000 (0:00:10.407) 0:01:04.523 *****
52 =====
53 deploy_compose : Deploy "dojo_service" ----- 49.18s
54 deploy_compose : Removing all inactive docker images ----- 10.41s
55 deploy_compose : Create dir for settings.d and keys ----- 2.90s
56 deploy_compose : copy tempaltes ----- 1.98s
✓ 59 Authenticating with credentials from /root/.docker/config.json
✓ 61 Authenticating with credentials from /root/.docker/config.json
63 Job succeeded

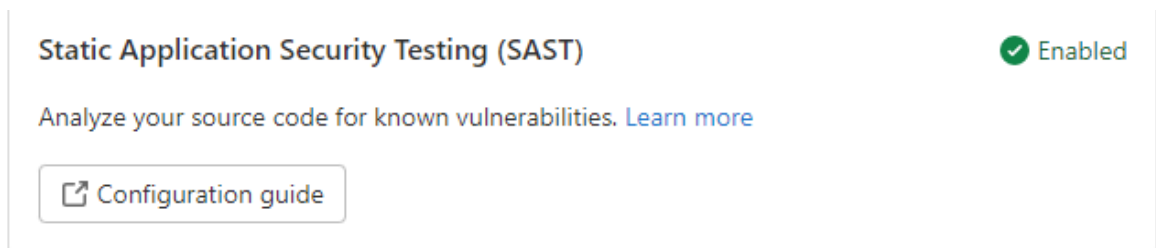
```

## Этап 2. SAST

**Основная задача SAST** – преодолеть разрыв между разработкой и безопасностью.

В качестве плюсов SAST можно выделить:

- возможность интеграции статического анализа в процесс разработки;
- автоматическое выявление критических уязвимостей, таких как переполнение буфера, SQL-инъекция, межсайтовый скриптинг (XSS) и других;
- и самое классное – указание на точное расположение подозрительного фрагмента кода, что особенно актуально для крупных проектов с сотнями тысяч и миллионами строк кода.
- В Gitlab можно использовать встроенный механизм проверки, который доступен без подписки Ultimate.

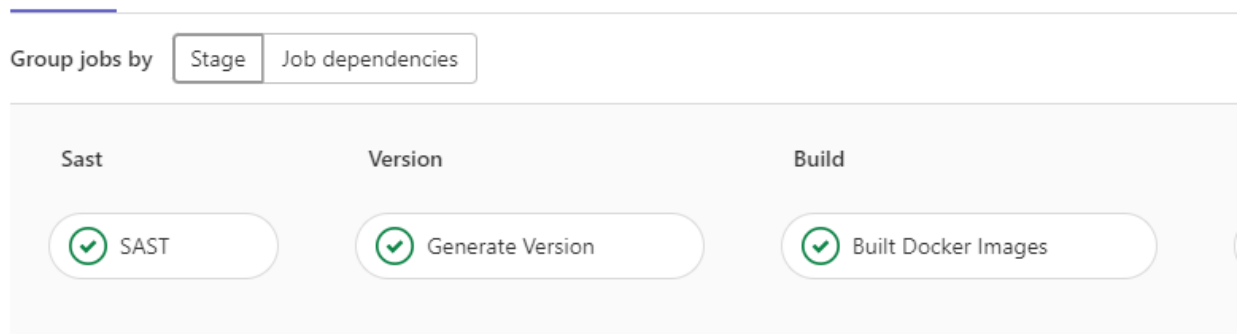


Видим отработку задания.

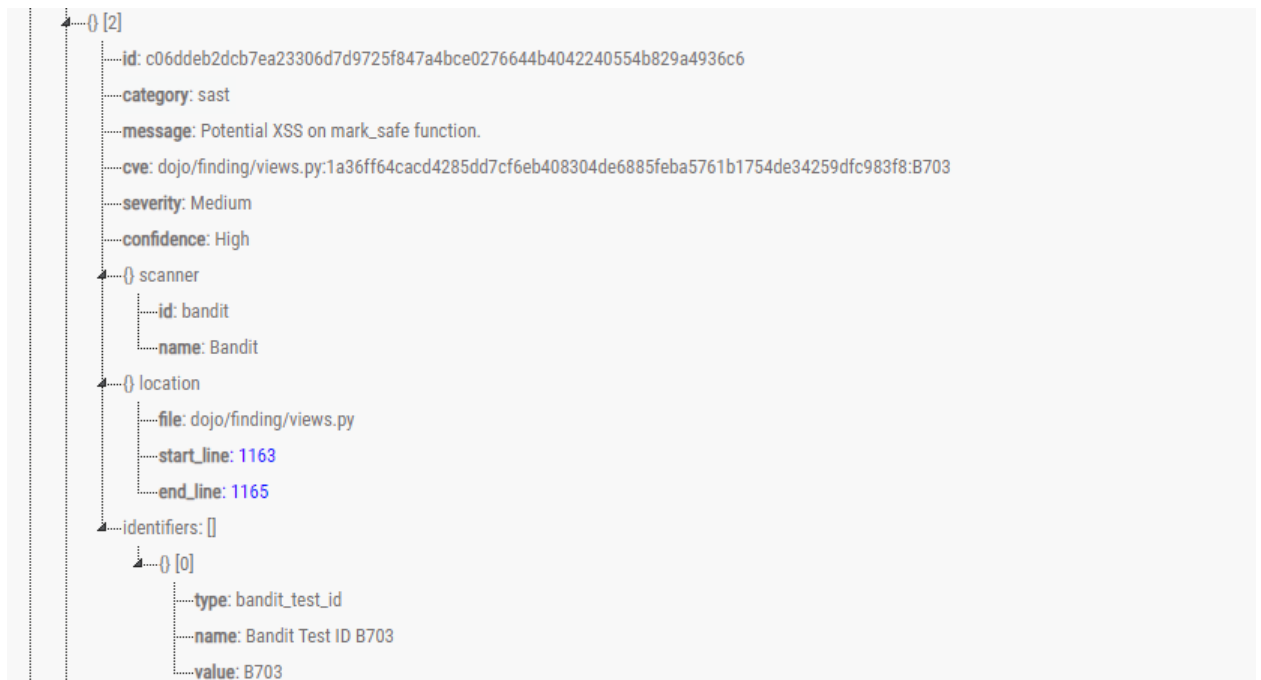
```
1 Running with gitlab-runner 12.8.0 (1b659122)
2   on Docker runner 1 on Linux hjrj9k_z
✓ 3 Using Docker executor with image registry.gitlab.com/security-products/bandit:2 ...
4 Using locally found image version due to if-not-present pull policy
5 Using docker image sha256:e6c6b290a8d8b0e2d37dc2e79a861f06dc820e4a8d2f37068636241c2e980d3c for registry.gitlab.com/security-products/bandit:2 ...
✓ 7 Authenticating with credentials from /root/.docker/config.json
8 Running on runner-hjrj9k_z-project-1476-concurrent-0 via urudc5ap089.brc.local...
✓ 10 Authenticating with credentials from /root/.docker/config.json
11 Fetching changes with git depth set to 20...
12 Reinitialized existing Git repository in /builds/poroshkinaa/django-DefectDojo/.git/
13 From https://gitlab.moduldev.ru/poroshkinaa/django-DefectDojo
14 * [new ref]      refs/pipelines/2361647 -> refs/pipelines/2361647
15   f9e893b..dafcf34 master      -> origin/master
16 Checking out dafcf34a as master...
17 Skipping Git submodules setup
✓ 19 Authenticating with credentials from /root/.docker/config.json
✓ 21 Authenticating with credentials from /root/.docker/config.json
23 $ /analyzer run
24 [INFO] [Bandit] [2023-01-18T07:01:55Z] ► GitLab Bandit analyzer v2.12.6
25 [INFO] [Bandit] [2023-01-18T07:01:55Z] ► Detecting project
26 [INFO] [Bandit] [2023-01-18T07:01:55Z] ► Found relevant files in project, analyzing entire repository
27 [INFO] [Bandit] [2023-01-18T07:01:55Z] ► Running analyzer
28 [INFO] [Bandit] [2023-01-18T07:03:01Z] ► Creating report
✓ 31 Authenticating with credentials from /root/.docker/config.json
✓ 33 Authenticating with credentials from /root/.docker/config.json
34 Uploading artifacts...
35 ./gl-sast-report.json: found 1 matching files
36 Uploading artifacts to coordinator... ok      id=3469423 responseStatus=201 Created token=w5e-vaK8
38 Job succeeded
```

После выполнения.





В качестве артефактов получаем выгрузку в формате JSON:



Dependabot.

Довольно качественный инструмент, который отслеживает уязвимости в зависимостях кода и открывает запросы на их обновление до минимально необходимой версии. Вполне приемлемый пользовательский интерфейс позволяет оперативно выявлять и устранять найденные уязвимости.

Overview

Reporting

Policy

Advisories

Vulnerability alerts

Dependabot

Code scanning

Secret scanning

Dependabot alerts

Configure

is:open

<input type="checkbox"/>	9 Open	1 Closed	Package	Ecosystem	Manifest	Severity	Sort
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Time-of-check Time-of-use (TOCTOU) Race Condition in league/flysystem	Critical	#8 opened 28 minutes ago • Detected in league/flysystem (Composer) • composer.lock		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Non-constant time comparison in UriSigner	High	#10 opened 28 minutes ago • Detected in symfony/http-kernel (Composer) • composer.lock		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Improper Input Validation in Laravel	High	#7 opened 28 minutes ago • Detected in laravel/framework (Composer) • composer.lock		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SQL Server LIMIT / OFFSET SQL Injection in laravel/framework and illuminate/database	High	#6 opened 28 minutes ago • Detected in laravel/framework (Composer) • composer.lock		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Unexpected database bindings	High	#5 opened 28 minutes ago • Detected in laravel/framework (Composer) • composer.lock		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Query Binding Exploitation	High			

Code scanning

Overview

Reporting

Policy

Advisories

Vulnerability alerts

Dependabot

Code scanning

Secret scanning

Code scanning

Add scanning tool

Latest scan	Branch	Workflow	Lines scanned	Duration	Result
now	master	CodeQL	85.6k	2s	8 alerts

is:open branch:master

<input type="checkbox"/>	62 Open	0 Closed	Tool	Branch	Rule	Severity	Sort
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Use of a broken or weak cryptographic hashing algorithm on sensitive data	High	master	#61 opened now • Detected by CodeQL in dojo/.../trufflehog3/parser.py:127	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Use of a broken or weak cryptographic hashing algorithm on sensitive data	High	master	#60 opened now • Detected by CodeQL in dojo/.../gitileaks/parser.py:127	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Use of a broken or weak cryptographic hashing algorithm on sensitive data	High	master	#59 opened now • Detected by CodeQL in dojo/.../ggshield/parser.py:105	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Use of a broken or weak cryptographic hashing algorithm on sensitive data	High	master	#58 opened now • Detected by CodeQL in dojo/.../bugcrowd/parser.py:76	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Clear-text logging of sensitive information	High	master	#57 opened now • Detected by CodeQL in dojo/remote_user.py:41	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Clear-text logging of sensitive information	High	master	#56 opened now • Detected by CodeQL in dojo/remote_user.py:25	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Incomplete URL substring sanitization	High	master	#55 opened now • Detected by CodeQL in dojo/models.py:2864	

## Этап 3. DAST

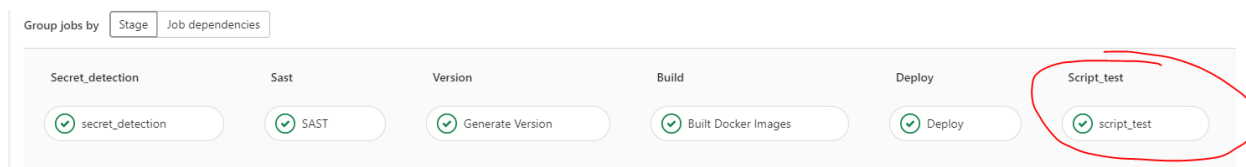
Динамическое тестирование безопасности приложений имитирует вредоносные атаки, которые используют распространенные уязвимости.

Основная задача DAST — выявить ошибки до того, как их обнаружит злоумышленник. Такие инструменты ищут уязвимые области, проверяя точки доступа и имитируя взаимодействие с пользователем.

DAST позволяет разработчикам выявлять недостатки, вызванные внедрениями кода (например, внедрение кода на веб-страницу) или связанные с некорректной настройкой (например, аутентификация с пустым паролем).

### Преимущества DAST:

- В отличие от SAST, он позволяет разработчикам обнаруживать проблемы во время выполнения кода. Это могут быть недостатки аутентификации и настройки сети, либо проблемы, возникающие только после входа в систему;
- DAST находит ошибки, возникающие при работе пользователя с приложением;
- Позволяет разработчикам тестировать приложение и выявлять недостатки, которые не были обнаружены обычными тестами;
- DAST не привязан к языкам программирования.



```

304 http://172.21.22.184:8080/login?next=/product/add (200)
305 PASS: User Controllable JavaScript Event (XSS) [10043]
306 PASS: Open Redirect [10028]
307 PASS: Username Hash Found [10057]
308 PASS: ViewState [10032]
309 PASS: X-AspNet-Version Response Header [10061]
310 PASS: X-Backend-Server Header Information Leak [10039]
311 SKIP: X-ChromeLogger-Data (XCOLD) Header Information Leak [10052]
312 WARN: X-Content-Type-Options Header Missing [10021] x 101
313 http://172.21.22.184:8080/static/bootstrap/dist/js/bootstrap.min.js (200)
314 http://172.21.22.184:8080/static/chosen-js/chosen.jquery.min.js (200)
315 http://172.21.22.184:8080/static/bootstrap-select/dist/js/bootstrap-select.min.js (200)
316 http://172.21.22.184:8080/static/metismenu/dist/metisMenu.min.js (200)
317 http://172.21.22.184:8080/static/bootstrap-select/dist/css/bootstrap-select.min.css (200)
318 PASS: X-Debug-Token Information Leak [10056]
319 PASS: Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) [10037]
320 WARN: Vulnerable JS Library (Powered by Retire.js) [10003] x 4
321 http://172.21.22.184:8080/static/components-jqueryui/jquery-ui.min.js (200)
322 http://172.21.22.184:8080/static/startbootstrap-sb-admin-2/bower_components/bootstrap/dist/js/bootstrap.min.js (200)
323 http://172.21.22.184:8080/static/startbootstrap-sb-admin-2/bower_components/jquery/dist/jquery.min.js (200)
324 http://172.21.22.184:8080/static/startbootstrap-sb-admin-2/bower_components/datatables/media/js/jquery.dataTables.min.js (200)
325 SUMMARY - PASS: 37 | WARN: 11 | SKIP: 12
✓ 327 Uploading artifacts for successful job
328 Uploading artifacts...
329 WARNING: /output: no matching files
330 ERROR: No files to upload
✓ 332 Cleaning up project directory and file based variables
334 Job succeeded

```

Для поиска уязвимостей я так же пробовал использовать sonarcloud

★
Dmitriy-yarmolenko / [django-DefectDojo](#)
NEW PUBLIC
Not computed

Last analysis: 1/16/2023, 10:37 AM • 696k Lines of Code • XML, HTML, ...

E 478
Bugs

E 1
Vulnerabilities

E 0.0%
Hotspots Reviewed

A 3.5k
Code Smells

3.1%
Duplications

Quality Gate ?

## Passed

Reliability
0 Bugs ?
A

Maintainability
0 Code Smells ?
A

Security
0 Vulnerabilities ?
A

Security Review
0 Security Hotspots ?
A

Coverage

A few extra steps are needed for SonarCloud to analyze your code coverage
[Setup coverage analysis](#)

Duplications
0.7% Estimated after merge

Dmitriy-yarmolenko > django-DefectDojo > master

SummaryIssuesSecurity HotspotsMeasuresCodeActivity

696k Lines of CodeLast analysis 9 minutes ago178dab97Create sonarqube.yml

Quality Gate

Not computed

The Quality Gate helps you see if your New Code is deployable or not.

Set New Code Definition

Reliability

478 Bugs

E

Maintainability

3.5k Code Smells

A

Security

1 Vulnerabilities

E

Security Review

197 Security Hotspots0.0% Reviewed

E

Coverage

A few extra steps are needed for SonarCloud to analyze your code coverage

Setup coverage analysis

Duplications

3.1% Duplications

Делаем запрос на исправление:

SummaryIssuesSecurity HotspotsMeasuresCodeActivity

Filters

Clear All Filters

Type

Bug8

Vulnerability0

Code Smell137

Ctrl + click to add to selection

Severity

Blocker8

Minor11

Critical33

Info0

Major85

Resolution

Bulk Change 137 Issue(s)

to select issues

to navigate

1 / 137 issues

7h 52min effort

app/Http/Controllers/EventController.php

Either split this list into multiple lines, aligned at column "12" or put all arguments on line "78".

No tags

Code Smell

Minor

Confirmed

Not assigned

1min effort · 8 years ago

app/Http/Controllers/PageController.php

Define and throw a dedicated exception instead of using a generic one.

No tags

Code Smell

Major

Confirmed

Not assigned

20min effort · 7 years ago

app/Http/Controllers/PostController.php

Remove the literal "false" boolean value.

No tags

Code Smell

Minor

Confirmed

Not assigned

5min effort · 9 years ago

app/Http/Controllers/PostController.php

Either split this list into multiple lines, aligned at column "12" or put all arguments on line "78".

No tags

Code Smell

Minor

Confirmed

Not assigned

1min effort · 8 years ago

Filters

is:pr is:open

Labels10

Milestones0

New pull request

1 Open

0 Closed

Author

Label

Projects

Milestones

Reviews

Assignee

Sort

Bump symfony/http-foundation from 2.7.3 to 2.7.51

dependencies

#1 opened 24 minutes ago by dependabot

1

ProTip!

Follow long discussions with comments:>50.

Bumps [symfony/http-foundation](#) from 2.7.3 to 2.7.51.

► Commits

compatibility | unknown

Dependabot will resolve any conflicts with this PR as long as you don't alter it yourself. You can also trigger a rebase manually by commenting `@dependabot rebase`.

► Dependabot commands and options

Bump [symfony/http-foundation](#) from 2.7.3 to 2.7.51 ...

Verified

✓ 9c240f6

dependabot (bot) added the [dependencies](#) label 25 minutes ago



sonarcloud (bot) commented 20 minutes ago



Kudos, SonarCloud Quality Gate passed! **Passed**

**A** 0 Bugs

**A** 0 Vulnerabilities

**A** 0 Security Hotspots

**A** 0 Code Smells

No Coverage information

No Duplication information

Add more commits by pushing to the `dependabot/composer/symfony/http-foundation-2.7.51` branch on [Dmitriy-yarmolenko/CMS](#).



Require approval from specific reviewers before merging

[Branch protection rules](#) ensure specific people approve pull requests before they're merged.

Add rule



All checks have passed

2 successful checks

[Show all checks](#)

# Можно выбрать и другие инструменты в GitHub

## Choose a workflow

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and [set up a workflow yourself](#) →

### Categories

- Automation
- Continuous integration
- Deployment
- Security**
- Pages

Q Search workflows

Found 64 workflows

CodeQL Analysis

By GitHub

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

Configure

Code scanning

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.

Configure

Code scanning

Codacy Security Scan

By Codacy

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

Configure

Code scanning

Dependency Review

By GitHub Actions

Scans Pull Requests on each push for the introduction and/or resolution of vulnerable dependencies to the repository

Configure

Dependency review

APIsec Scan

By APIsec

APIsec provides the industry's only automated and continuous API testing platform that uncovers security vulnerabilities and logic flaws in APIs.

Configure

Code scanning

Checkmarx

By Checkmarx

Beat vulnerabilities with more secure code. Scan your code with Checkmarx One and see results in the GitHub code scanning.

Configure

Code scanning

CxSAST

By Checkmarx

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

Configure

Code scanning

DevSkim

By Microsoft CST-E

DevSkim is a security linter that highlights common security issues in source code.

Configure

Code scanning

EthicalCheck

By APIsec

EthicalCheck provides the industry's only free & automated API security testing service that uncovers security vulnerabilities using OWASP API list.

Configure


Code scanning

## Этап 4. Security Checks

### Secret Detection

✓ Enabled

Analyze your source code and git history for secrets. [Learn more](#)

 Configuration guide

### Secret\_detection

### Sast

✓ secret\_detection



✓ SAST



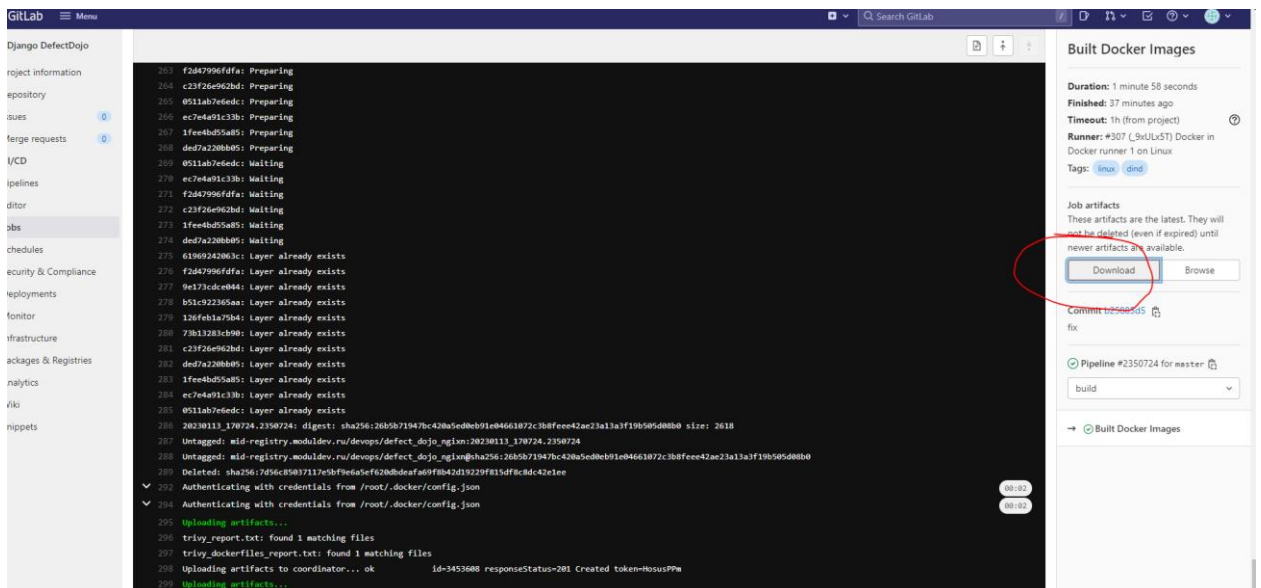
```
1 Running with gitlab-runner 14.10.1 (f761588f)
2   on security scan runner slAHQBL
3   Preparing the "docker" executor
4 Using Docker executor with image registry.gitlab.com/security-products/secrets:3 ...
5 Using locally found image version due to "if-not-present" pull policy
6 Using docker image sha256:f390210539d9224965764b376a06660c481d104f00e0043c7209d8396fe51ef1 for registry.gitlab.com/security-products/secrets:3 with digest registry.gitlab.com/se
curity-products/secrets@sha256:e9f2df058eda3a86987db272fc29aa72b7de95754dda2873e94ddc74c4a3d340 ...
7   Preparing environment
8   Running on runner-slanhqbl-project-1476-concurrent-0 via 90a8e40833b4...
9   Getting source from Git repository
10  Fetching changes with git depth set to 50...
11  Reinitialized existing Git repository in /builds/poroshkinaa/django-DefectDojo/.git/
12  Checking out 3585c742 as master...
13  Removing gl-secret-detection-report.json
14  Skipping Git submodules setup
15  Executing "step_script" stage of the job script
16  Using docker image sha256:f390210539d9224965764b376a06660c481d104f00e0043c7209d8396fe51ef1 for registry.gitlab.com/security-products/secrets:3 with digest registry.gitlab.com/se
curity-products/secrets@sha256:e9f2df058eda3a86987db272fc29aa72b7de95754dda2873e94ddc74c4a3d340 ...
17  $ if [ -n "$CI_COMMIT_TAG" ]; then echo "Skipping Secret Detection for tags. No code changes have occurred."; exit 0; fi
18  $ if [ "$SECRET_DETECTION_HISTORIC_SCAN" == "true" ]; then echo "Running Secret Detection Historic Scan"; /analyzer run; exit; fi
19  $ if [ "$CI_COMMIT_BRANCH" == "$CI_DEFAULT_BRANCH" ]; then echo "Running Secret Detection on default branch."; /analyzer run; exit; fi
20  Running Secret Detection on default branch.
```

```
24 [INFO] [secrets] [2023-01-19T06:55:47Z] ► GitLab secrets analyzer v3.27.1
25 [INFO] [secrets] [2023-01-19T06:55:47Z] ► Detecting project
26 [INFO] [secrets] [2023-01-19T06:55:47Z] ► Found relevant files in project, analyzing entire repository
27 [INFO] [secrets] [2023-01-19T06:55:47Z] ► Running analyzer
28 [INFO] [secrets] [2023-01-19T06:55:47Z] ►
29 [INFO] [secrets] [2023-01-19T06:55:47Z] ► o
30 [INFO] [secrets] [2023-01-19T06:55:47Z] ► | \
31 [INFO] [secrets] [2023-01-19T06:55:47Z] ► | o
32 [INFO] [secrets] [2023-01-19T06:55:47Z] ► o
33 [INFO] [secrets] [2023-01-19T06:55:47Z] ► gitleaks
34 [INFO] [secrets] [2023-01-19T06:55:47Z] ►
35 [INFO] [secrets] [2023-01-19T06:59:52Z] ► 6:59AM INF scan completed in 4m4.962691106s
36 [INFO] [secrets] [2023-01-19T06:59:52Z] ► 6:59AM WRN leaks found: 587
37 [INFO] [secrets] [2023-01-19T06:59:52Z] ► Creating report
38 Uploading artifacts for successful job
39 Uploading artifacts...
40 gl-secret-detection-report.json: found 1 matching files and directories
41 Uploading artifacts as "secret_detection" to coordinator... 201 Created id=3472877 responseStatus=201 Created token=VjXtai7J
42 Cleaning up project directory and file based variables
43 Job succeeded
```





Trivy показал нам в выгрузке найденные уязвимости и секреты.



Имя

trivy\_dockerfiles\_report

trivy\_report

trivy_report - Блокнот					
Файл Правка Формат Вид Справка					
mid-registry.moduldev.ru/devops/defect_dojo:20230113_170724.2350724 (debian 11.6)					
Total: 191 (UNKNOWN: 0, LOW: 116, MEDIUM: 33, HIGH: 37, CRITICAL: 5)					
Library	Vulnerability	Severity	Installed Version	Fixed Version	Title
apt	CVE-2011-3374	LOW	2.2.4		It was found that apt-key in apt, all versions, do correctly... <a href="https://avd.aquasec.com/nvd/cve-2011-3374">https://avd.aquasec.com/nvd/cve-2011-3374</a>
bash	CVE-2022-3715	CRITICAL	5.1-2+deb11u1		bash: a heap-buffer-overflow in valid_parameter_tr <a href="https://avd.aquasec.com/nvd/cve-2022-3715">https://avd.aquasec.com/nvd/cve-2022-3715</a>
bind9-dnsutils	CVE-2022-2881	HIGH	1:9.16.33-1~deb11u1		bind: buffer overread in statistics channel code <a href="https://avd.aquasec.com/nvd/cve-2022-2881">https://avd.aquasec.com/nvd/cve-2022-2881</a>
bind9-host					
bind9-libs					
bsdutils	CVE-2022-0563	LOW	2.36.1-8+deb11u1		util-linux: partial disclosure of arbitrary files and chsh when compiled... <a href="https://avd.aquasec.com/nvd/cve-2022-0563">https://avd.aquasec.com/nvd/cve-2022-0563</a>
coreutils	CVE-2016-2781		8.32-4		coreutils: Non-privileged session can escape to th session in chroot <a href="https://avd.aquasec.com/nvd/cve-2016-2781">https://avd.aquasec.com/nvd/cve-2016-2781</a>
coreutils	CVE-2017-18018	LOW	8.32-4		coreutils: race condition vulnerability in chown a <a href="https://avd.aquasec.com/nvd/cve-2017-18018">https://avd.aquasec.com/nvd/cve-2017-18018</a>
dnsutils	CVE-2022-2881	HIGH	1:9.16.33-1~deb11u1		bind: buffer overread in statistics channel code <a href="https://avd.aquasec.com/nvd/cve-2022-2881">https://avd.aquasec.com/nvd/cve-2022-2881</a>
e2fsprogs	CVE-2022-1304		1.46.2-2		e2fsprogs: out-of-bounds read/write via crafted fi <a href="https://avd.aquasec.com/nvd/cve-2022-1304">https://avd.aquasec.com/nvd/cve-2022-1304</a>
git	CVE-2022-24765	HIGH	1:2.30.2-1		git: On multi-user machines Git users might find t unexpectedly in a... <a href="https://avd.aquasec.com/nvd/cve-2022-24765">https://avd.aquasec.com/nvd/cve-2022-24765</a>

Секреты:

/app/unittests/scans/rusty\_hog/gottingenhog\_many\_vulns.json (secrets)

=====

Total: 2 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 2)

CRITICAL: AWS (aws-access-key-id)

=====

AWS Access Key ID

=====

/app/unittests/scans/rusty\_hog/gottingenhog\_many\_vulns.json:31

29     {  
30         "stringsFound": [  
31             "\*\*\*\*\*"  
32         ],

=====

CRITICAL: Stripe (stripe-secret-token)

=====

Stripe Secret Key

=====

/app/unittests/scans/rusty\_hog/gottingenhog\_many\_vulns.json:67

65     {  
66         "stringsFound": [  
67             "\*\*\*\*\*"  
68         ],

# Проверка на секреты в GitHub:

Overview

Reporting

Policy

Advisories

Vulnerability alerts

Dependabot

Code scanning

Secret scanning6

Secret scanning alerts

is:open

6 Open0 Closed

Secret typePro

☐

Google Cloud Private Key ID

ab2fe11c32b572c5ce72f0ca432d8...

#6 opened 3 minutes ago • Detected secret in tests/test-doj-sheets-NONEXIS...5

☐

Stripe API Key

sk\_live\_12345678901234567890a...

#5 opened 3 minutes ago • Detected secret in unittests/.../rusty\_hog/gottingenhog\_many\_vulns....67

☐

Google API Key

AIzaSyD9fw34edJfoEtEbymL9RXvL...

#4 opened 3 minutes ago • Detected secret in dojo/.../mobsf/android.json:4485

☐

Google API Key

AIzaSyBsZ1p5hn1TGLRNCJQAN-lNt...

#3 opened 3 minutes ago • Detected secret in dojo/.../mobsf/android.json:3842

☐

Mailchimp API Key

38c47f19e349153fa963bb3b3212f...

#2 opened 3 minutes ago • Detected secret in dojo/.../gitleaks/data\_many.json:16

☐

Slack API Token

xoxb-242897902580-JxwtAa0Y5NS...

#1 opened 3 minutes ago • Detected secret in dojo/utls.py:1035






## **Выводы:**

В данном проекте была продемонстрирована реализация построения безопасного пайплайна для open-source проекта. В нашем случае мы использовали [DefectDojo](#).

Основной инструмент я использовал Gitlab версии 14.10.2-ee и отдельные виртуальные машины, развернутые на инфраструктуре организации. В качестве виртуальной машины использовалась ОС на базе Ubuntu 22 версии.

Gitlab позволил сделать полноценный процесс сборки, тестирования и деплоя. Встроенными механизмами подключил Static Application Security Testing (SAST), Secret Detection, DAST (script\_test). В качестве DAST был использован встроенный инструмент, В качестве альтернативы возможно использование SonarCube Вариант альтернативных решений security можно использовать библиотеку GitHub, в которой имеется немало хороших реализаций. Security Checks так же было задействовано.

## Приложения:

1. SAST – артефакты  gl-sast-report.json
2. Артефакты сборки - trivy\_report  trivy\_report.txt
3. Gitlab-ci.yml -  gitlab-ci.yml
4. BUILD\_VERSION=20230118\_070140.2361647 -  build.env
5. Артефакты secret detection -  gl-secret-detection-report.json