

Homework - [Lecture 5] - DevSecOps

Первым делом, поскольку не могу использовать shared раннеры, придется зарегистрировать локальный раннер. Для этого буду использовать Ubuntu в VB, при регистрации задаю executor docker, default image взял ubuntu:latest. Можно было через shell, ставить тулзы standalone, но так выходит проще, поэтому работаем с образами.

Ссылка на BenchmarkJava: <https://gitlab.com/xokage/BenchmarkJava/-/tree/master>

(<https://gitlab.com/xokage/BenchmarkJava/-/tree/master>)

Ссылка на security-pipeline:

<https://gitlab.com/xokage/security-pipeline> (<https://gitlab.com/xokage/security-pipeline>)

Gitleaks

Поиск секретов в глубину. В данном репозитории порядка 473 коммитов, почему то долго сканится. В итоге, я полностью отсканил репозиторий с дефолтными правилами, можно по идее дописать toml файл и найти больше. Вышло 3 уязвимости, первые две - это мои коммиты с RSA ключом. И один я нашел токен от Sonar Qube.

```

23 From https://gitlab.com/xokage/BenchmarkJava
24 * branch      master      -> FETCH_HEAD
25 * branch      test-004    -> FETCH_HEAD
26 $ git checkout $CI_COMMIT_REF_NAME
27 Branch 'test-004' set up to track remote branch 'test-004' from 'origin'.
28 Switched to a new branch 'test-004'
29 $ gitleaks detect --source $CI_PROJECT_DIR --verbose --report-path $SECRETS_ARTIFACT --log-opts="--all" || LEAK_EXIT_CODE=$?
30
31  \
32  | o
33  o ||
34  ||  gitleaks
35  {
36    "Description": "SSH private key",
37    "StartLine": 1,
38    "EndLine": 1,
39    "StartColumn": 1,
40    "EndColumn": 35,
41    "Match": "-----BEGIN OPENSSH PRIVATE KEY-----",
42    "Secret": "-----BEGIN OPENSSH PRIVATE KEY-----",
43    "File": "file",
44    "Commit": "47e780835e7b2d8313a3ac59324cb5e6e44f93c2",
45    "Entropy": 0,
46    "Author": "Andrei Prokhorov",
47    "Email": "pro-andruhaa@yandex.ru",
48    "Date": "2022-03-27T14:15:31Z",
49    "Message": "Add new file",
50    "Tags": [],
51    "RuleID": "OPENSsh-PK"
52  }
53  {
54    "Description": "SSH private key",
55    "StartLine": 1,
56    "EndLine": 1,
57    "StartColumn": 1,
58    "EndColumn": 35,
59    "Match": "-----BEGIN OPENSSH PRIVATE KEY-----",
60    "Secret": "-----BEGIN OPENSSH PRIVATE KEY-----",
61    "File": "file",
62    "Commit": "3ffba729c7c01cf0dc72929e7846ef9a5d1c2cb1",
63    "Entropy": 0,
64    "Author": "Andrei Prokhorov",
65    "Email": "pro-andruhaa@yandex.ru",
66    "Date": "2022-03-27T14:32:03Z",
67    "Message": "Add new file",
68    "Tags": [],
69    "RuleID": "OPENSsh-PK"
70  }
71  {
72    "Description": "Generic API Key",
73    "StartLine": 7,
74    "EndLine": 7,
75    "StartColumn": 8,
76    "EndColumn": 55,
77    "Match": "token=\"291e50b66b5f5198e6ac8d49bf1857ac3a3a8fc8\"",
78    "Secret": "291e50b66b5f5198e6ac8d49bf1857ac3a3a8fc8",
79    "File": "scripts/runSonarQube.sh",
80    "Commit": "bf0cfd19159c9a3852d589379583675c1b8a5a5",
81    "Entropy": 3.858695,
82    "Author": "Sascha Knoop",
83    "Email": "github@darkspirit510.de",
84    "Date": "2021-09-18T20:39:18Z",
85    "Message": "more preconditions checks",
86    "Tags": [],
87    "RuleID": "generic-api-key"
88  }
89 2:54PM MRN leaks found: 3
90 2:54PM INF scan completed in 22m26.4845981s
91 $ if [ -x $LEAK_EXIT_CODE ] # collapsed multi-line command
92 Secrets has been found
93 Uploading artifacts for successful job
94 Uploading artifacts...
95 gitleaks_all.json: found 1 matching files and directories
96 Uploading artifacts as "archive" to coordinator... 201 Created id=2255725869 responseStatus=201 Created token=InH5bVn6
97 Cleaning up project directory and file based variables
101 Job succeeded

```

Для визуализации+распарсить отчет, чтобы можно было комфортно исследовать, использовал DD. Создал енгејдж и импотнул туда json.

test F

OverviewComponentsMetricsEngagements1Findings2EndpointsBenchmarksSettings

Engagements / All Engagements

Active Engagements (1)

Showing entries 1 to 1 of 1

Column visibilityCopyExcelCSVPDFPrint

Search:

Name	Type	Lead	Date	Length	Tests	Active (Verified)	Mitigated	Accepted	All	Duplicates
Gitleaks	Interactive	(admin)	27th March - 3rd April	7 days	1	2 (2)	0	0	2	0

Showing entries 1 to 1 of 1

Page Size

Paused Engagements (0)

No paused engagements found.

Closed Engagements (0)

No closed engagements found.

test F

OverviewComponentsMetricsEngagements1Findings2EndpointsBenchmarksSettings

Engagements / Gitleaks / Gitleaks Scan / Test

Gitleaks Scan

Updated 19 minutes ago, Created 19 minutes ago

Engagement	Environment	Dates	Updated	Progress	Version	Reimports
Gitleaks	Test	March 27, 2022 - March 27, 2022	March 27, 2022	100%		1

Import History (1)

Groups (0)

Findings (2)

Critical: 0, High: 2, Medium: 0, Low: 0, Info: 0, Total: 2 Findings

Showing entries 1 to 2 of 2

Column visibilityCopyExcelCSVPDFPrint

Search:

Severity	Name	CWE	CVE	Date	Age	SLA	Reporter	Status	Group
High	Hard Coded SSH Private Key Found in File ^{CP}	798		March 27, 2022	0	30	(admin)	Active, Verified	N
High	Hard Coded Generic API Key Found in scripts/runSonarQube.sh ^{CP}	798		March 27, 2022	0	30	(admin)	Active, Verified	N

Showing entries 1 to 2 of 2

Page Size

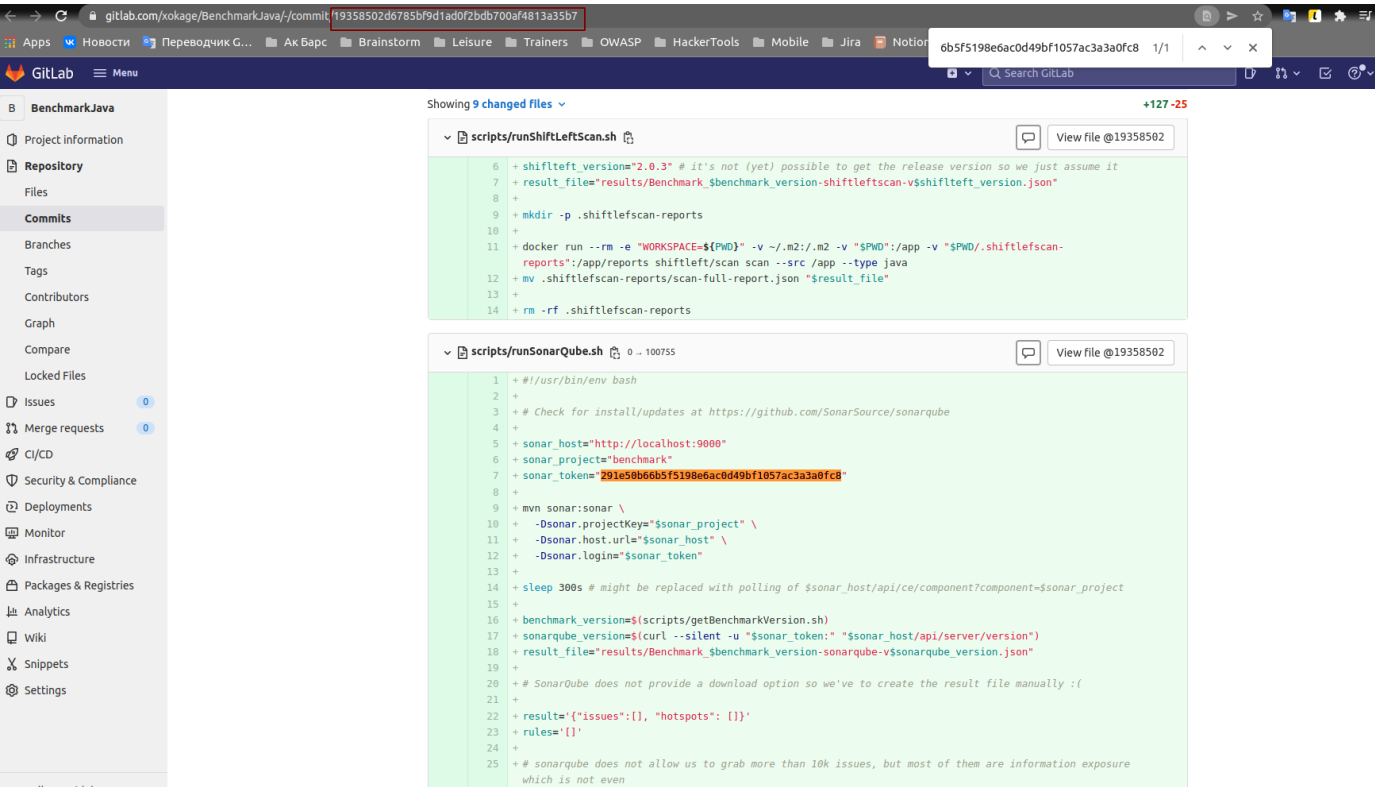
Potential Findings

Add a potential finding...

+ Add Potential Finding

В конце можно будет приложить полный отчет по всем инструментам.

Правда если сравнивать хэш коммита, то по нему не находится. Токен был захардкожен в другом коммите:




Предыдущие версии гитликса отдавали сразу ссылку в отчете.

Dependency Check

Здесь все достаточно просто, единственное я выгрузил два отчета, один html, другой xml для Defect Dojo.

Отчет показывает 3 Средних уязвимости:



DEPENDENCY-CHECK

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's discretion. The copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

[How to read the report](#) | [Suppressing false positives](#) | [Getting Help: github issues](#)

[Sponsor](#)

Project: BenchmarkJava

Scan Information ([show all](#)):

- dependency-check version: 7.0.1
- Report Generated On: Sun, 27 Mar 2022 16:33:54 GMT
- Dependencies Scanned: 6 (6 unique)
- Vulnerable Dependencies: 3
- Vulnerabilities Found: 14
- Vulnerabilities Suppressed: 0
- ...

Summary

Display: [Showing Vulnerable Dependencies \(click to show all\)](#)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
bootstrap.js		pkg:javascript/bootstrap@3.3.4	MEDIUM	5		3
bootstrap.min.js		pkg:javascript/bootstrap@3.3.4	MEDIUM	5		3
jquery.min.js		pkg:javascript/jquery@2.1.4	MEDIUM	4		3

Dependencies

bootstrap.js

Также закину в DefectDojo для общего отчета.

14 cve было найдено:

Findings (14) Critical: 0, High: 0, Medium: 14, Low: 0, Info: 0, Total: 14 Findings

Showing entries 1 to 14 of 14

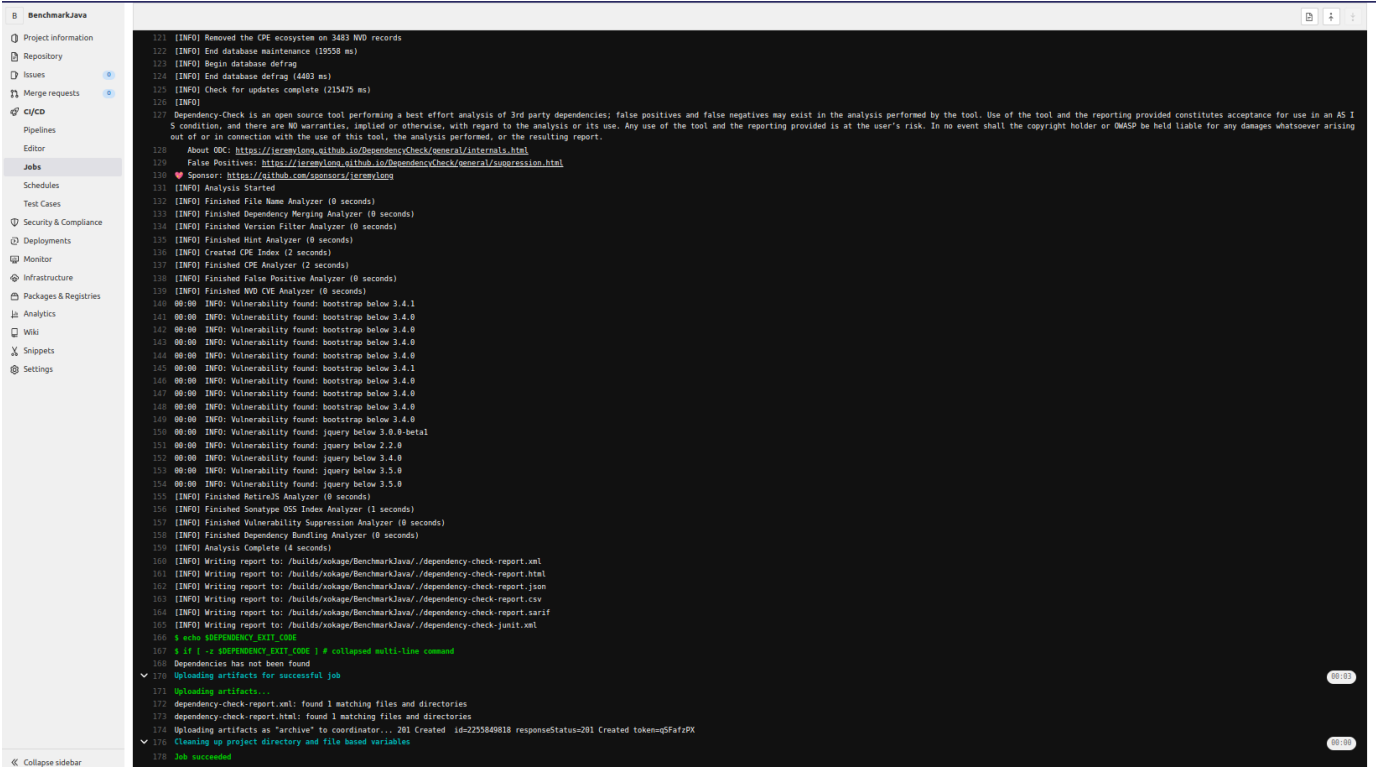
Page Size

Column visibilityCopyExcelCSVPDFPrint

Search:

<input type="checkbox"/>			Severity	Name	CWE	CVE	Date	Age	SLA	Reporter	Status	Group
<input type="checkbox"/>			Medium	bootstrap:3.3.4 CVE-2016-10735	79	CVE-2016-10735	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	bootstrap:3.3.4 CVE-2018-14040	79	CVE-2018-14040	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	bootstrap:3.3.4 CVE-2018-14041	79	CVE-2018-14041	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	bootstrap:3.3.4 CVE-2018-14042	79	CVE-2018-14042	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	bootstrap:3.3.4 CVE-2019-8331	79	CVE-2019-8331	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	bootstrap:3.3.4 CVE-2016-10735	79	CVE-2016-10735	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	bootstrap:3.3.4 CVE-2018-14040	79	CVE-2018-14040	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	bootstrap:3.3.4 CVE-2018-14041	79	CVE-2018-14041	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	bootstrap:3.3.4 CVE-2018-14042	79	CVE-2018-14042	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	bootstrap:3.3.4 CVE-2019-8331	79	CVE-2019-8331	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	jquery:2.1.4 CVE-2015-9251	79	CVE-2015-9251	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	jquery:2.1.4 CVE-2019-11358	79	CVE-2019-11358	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	jquery:2.1.4 CVE-2020-11022	79	CVE-2020-11022	March 27, 2022	0		(admin)	Active, Verified	N
<input type="checkbox"/>			Medium	jquery:2.1.4 CVE-2020-11023	79	CVE-2020-11023	March 27, 2022	0		(admin)	Active, Verified	N

Часть лога:



Впринципе можно создавать енгејджды и выгружать отчеты в DD через API, встроить в общий скрипт.

Semgrep SAST

Здесь нужно найти наилучший по результатам находок набор правил.
Золотую середину пришлось искать долго, путем перебора всех доступных наборов правил. Вышел вот такой список:

FROM Repository:

r2c-best-practices 9

ci 289

jwt 0

r2c 1071

xss 1721

test 0

java 3747

nginx 0

docker 3

r2c-ci 42 слабый DES алгоритм

default 5645

mobsfscan 1473

r2c-bug-scan 7

owasp-top-ten 3847

sql-injection 178

security-audit 3790

command-injection 1 небезопасная рефлексия

r2c-security-audit 3790

insecure-transport 1 используется версия TLSv1

R:

r/java.lang 3969

...

Можно брать просто правила по отдельности, но это очень долго, в наборах те же правила. Лучше всего конечно написать самому правила, что тоже очень долго. Делая вывод из таблицы выше, берем нужные правила в одну строку скрипта.

Вышло 56 штук, 2 DD посчитал за дубликаты:

Kics

Здесь все достаточно просто. Запускаем образ в контейнере, указываем папку с исходниками и куда выгружать отчет. Отчет я выбрал типа json, чтобы скормить DefectDojo.

В DD:

Kics нашел очень много уязвимостей, даже DD не смог столько распарсить, а только добавил в очередь. Понятно, что здесь тоже нужно что то придумать с правилами.

Еще попробовал прикрутить для поиска секретов в глубину trufflehog, но он очень медленный и находит много уязвимостей, тут тоже надо писать правила под него.

Отчет из ДД приложу в репозиторий гитхаба.