



DevSecOps на час



Андрей Моисеев
DevSecOps
Cloud Native Security

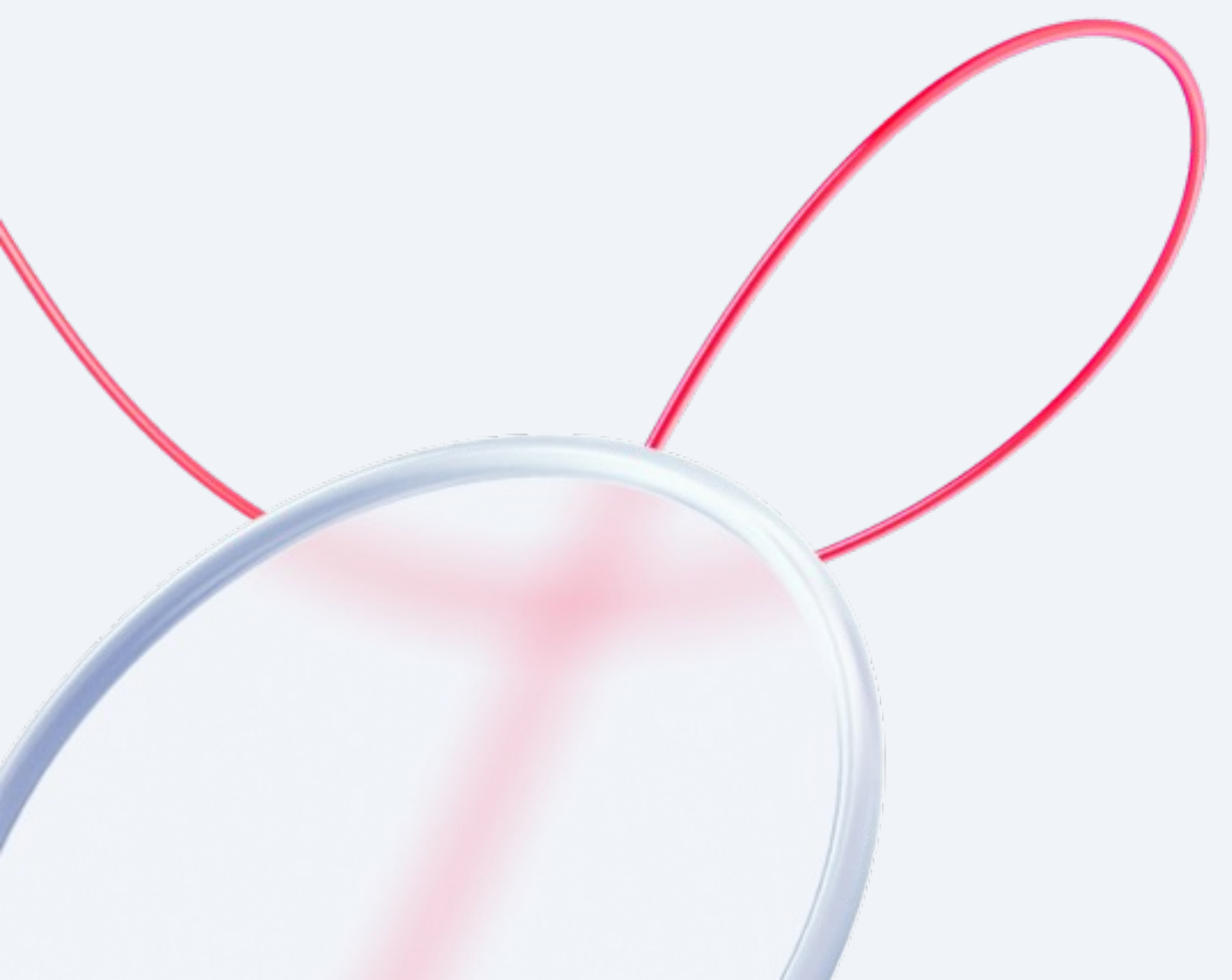


M W
S

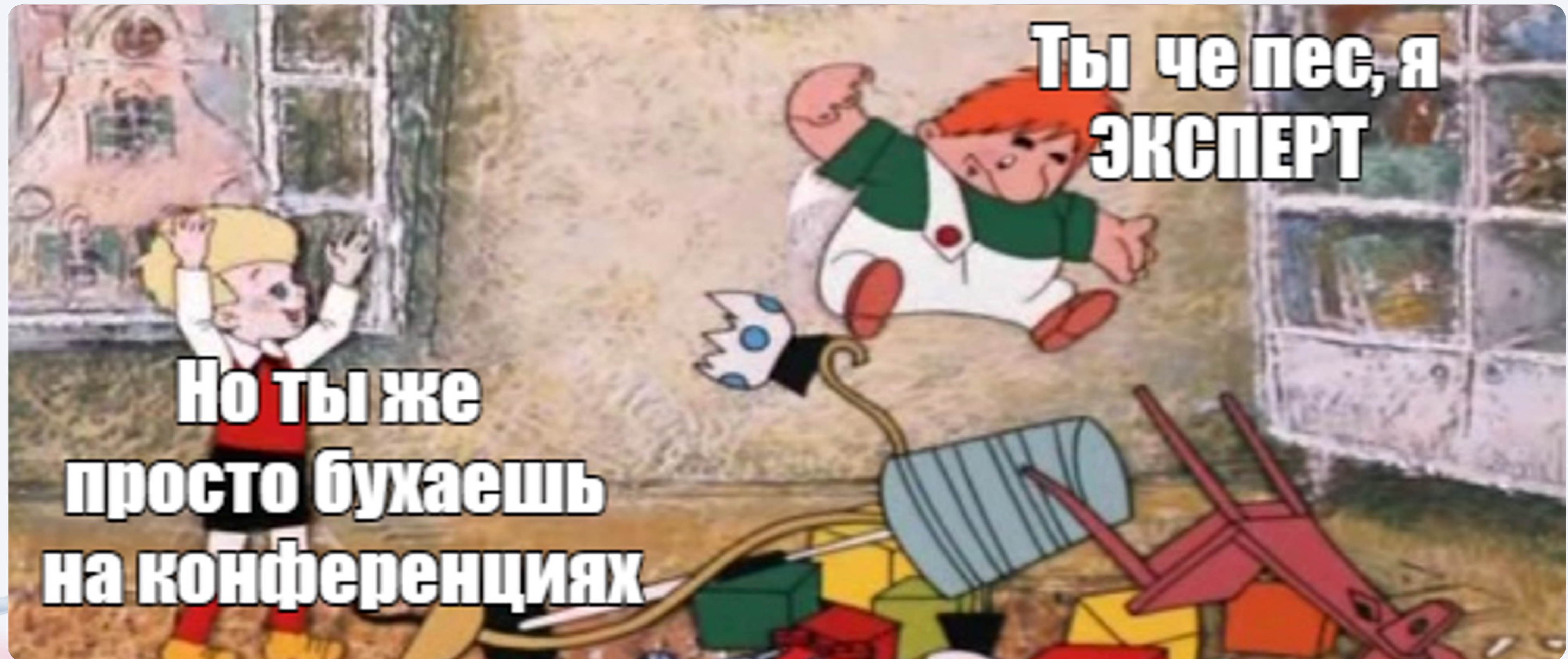
WHOAMI

**Андрей
Моисеев**

DevSecOps
Cloud Native Security



Чем занимается DevSecOps?



Чем занимается DevSecOps?

01 Не даёт катить фиши



Чем занимается DevSecOps?

01 Не даёт катить фичи

02 Создаёт проблемы



Чем занимается DevSecOps?

01 Не даёт катить фичи

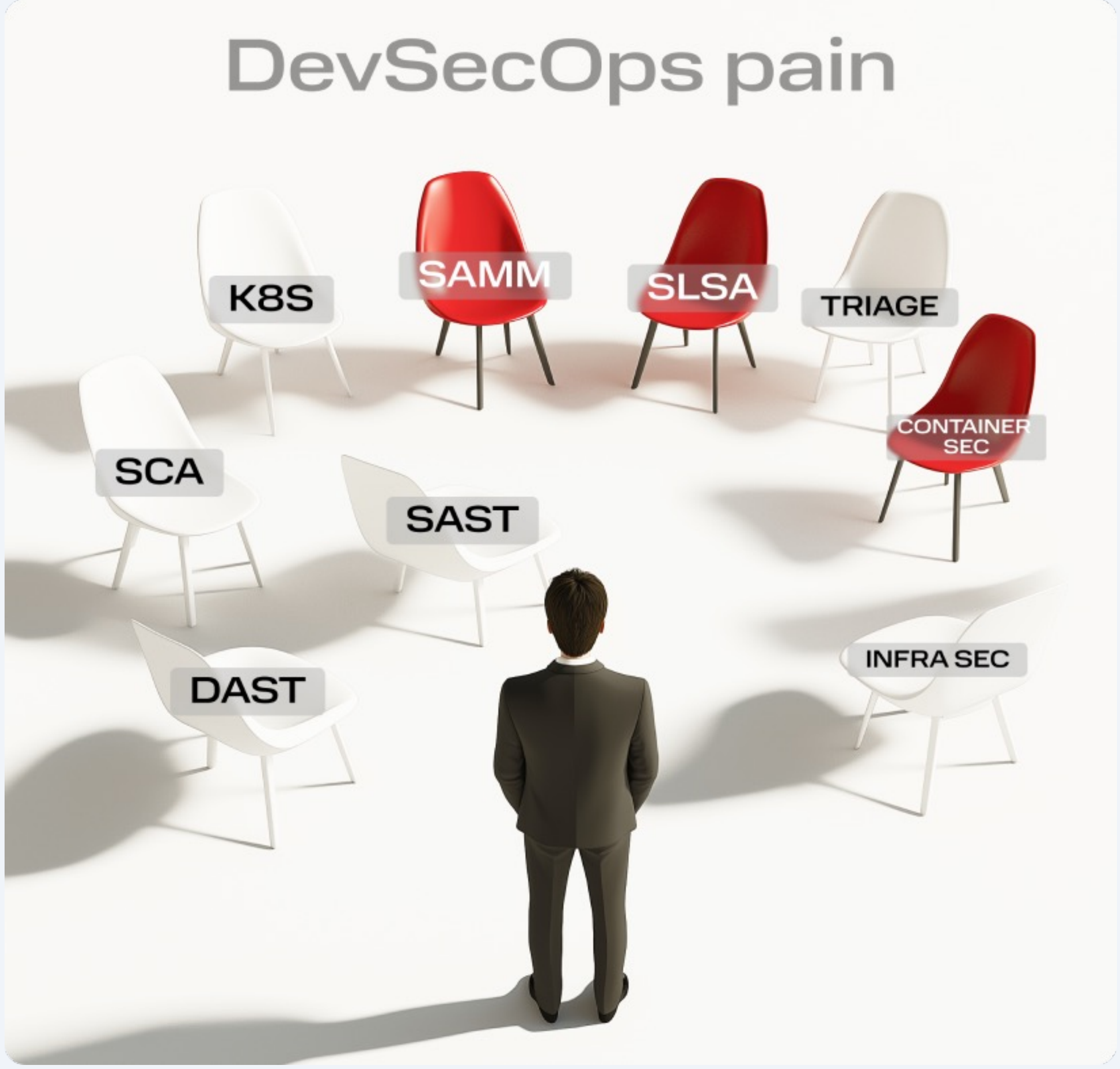
02 Создаёт проблемы

03 Развивает безопасность



Проблема one-man-team

01 Иллюзия выбора



Проблема one-man-team

01 Иллюзия выбора

02 И один в поле воин



Проблема one-man-team

01 Иллюзия выбора

02 И один в поле воин

03 Нужно все и сразу

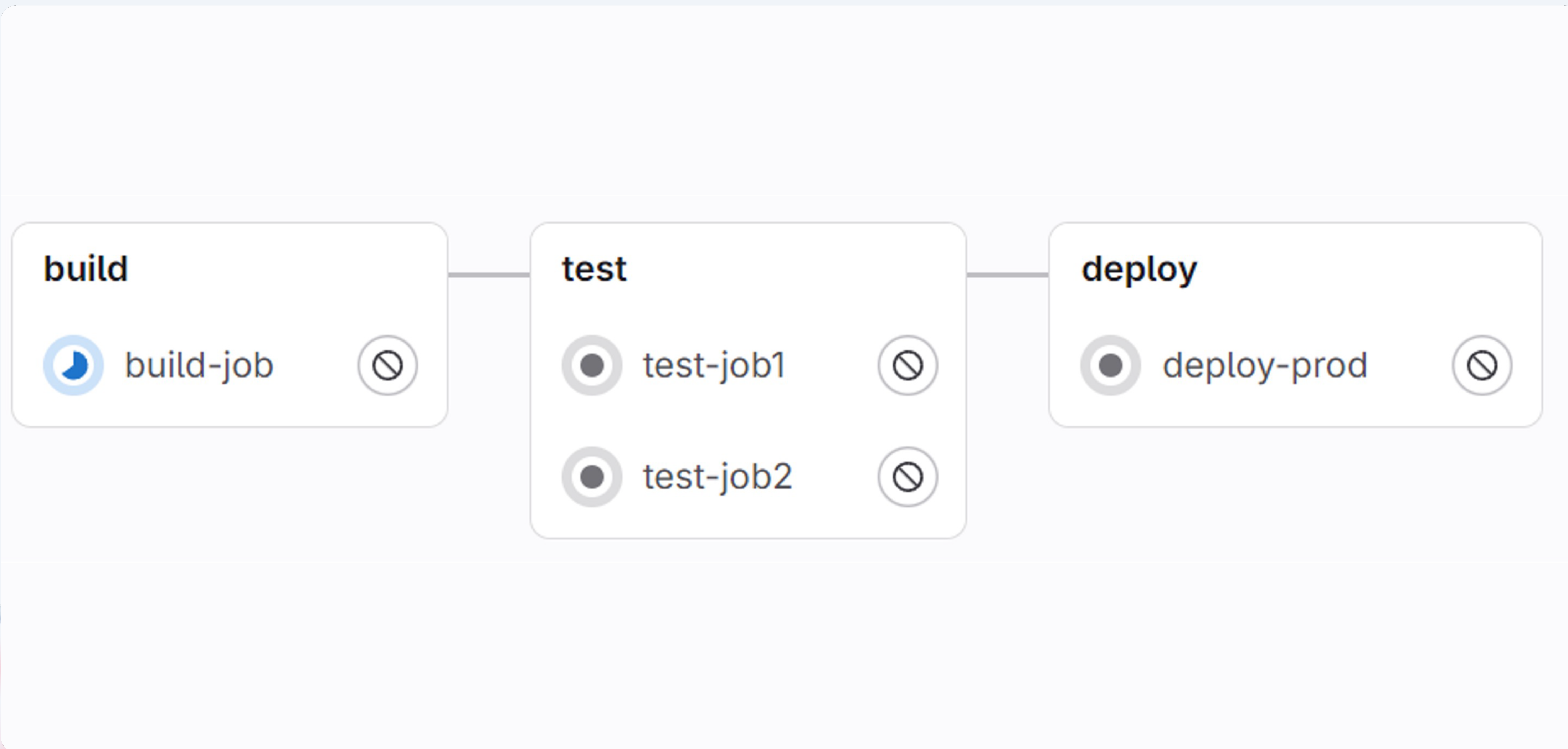


M W
S

Начинаем!

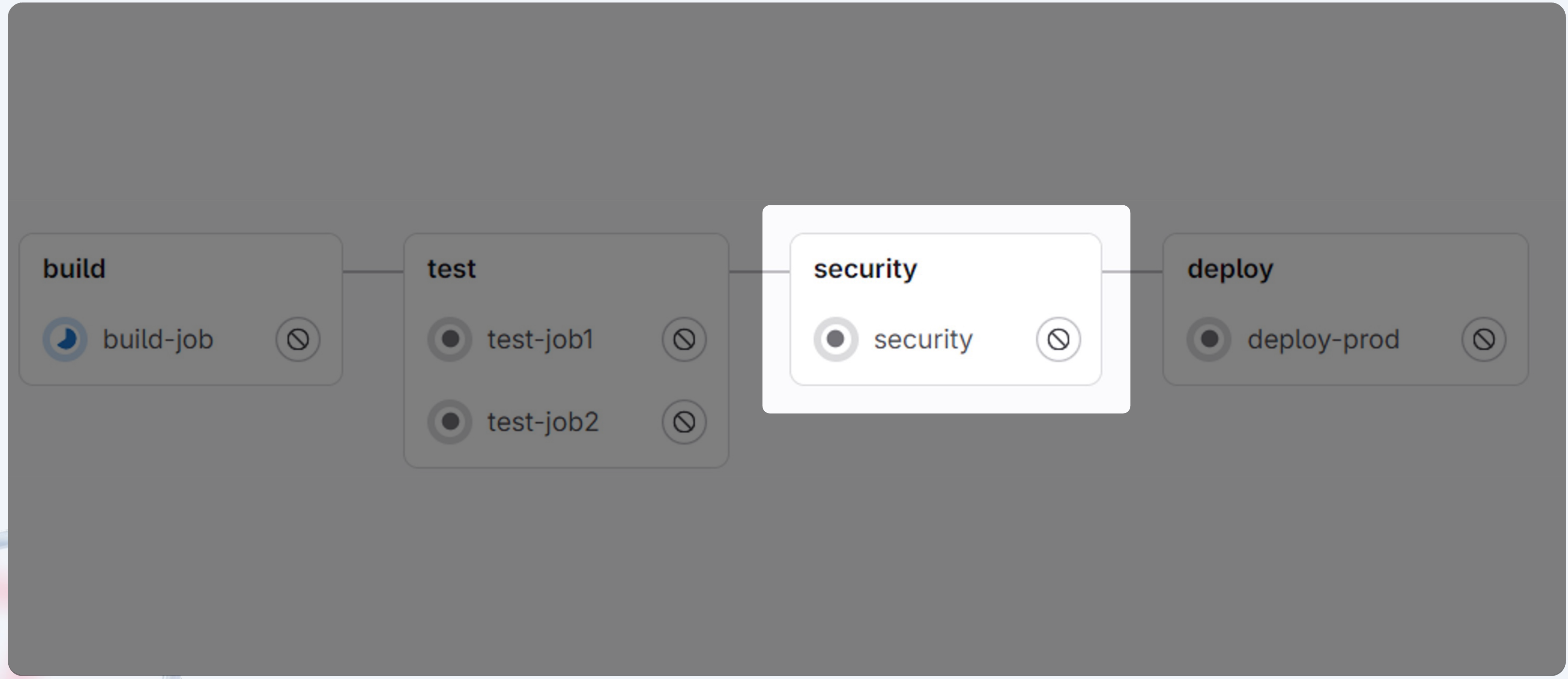


Базовый CI/CD



Базовый CI/CD

Нет security stage!



Зачем security

01 Собственный код может
содержать уязвимости



Зачем security

01 Собственный код может содержать уязвимости

02 Заимствованный код, тоже может содержать уязвимости



Зачем security

01 Собственный код может содержать уязвимости

02 Заимствованный код, тоже может содержать уязвимости

03 Окружение тоже может содержать уязвимости



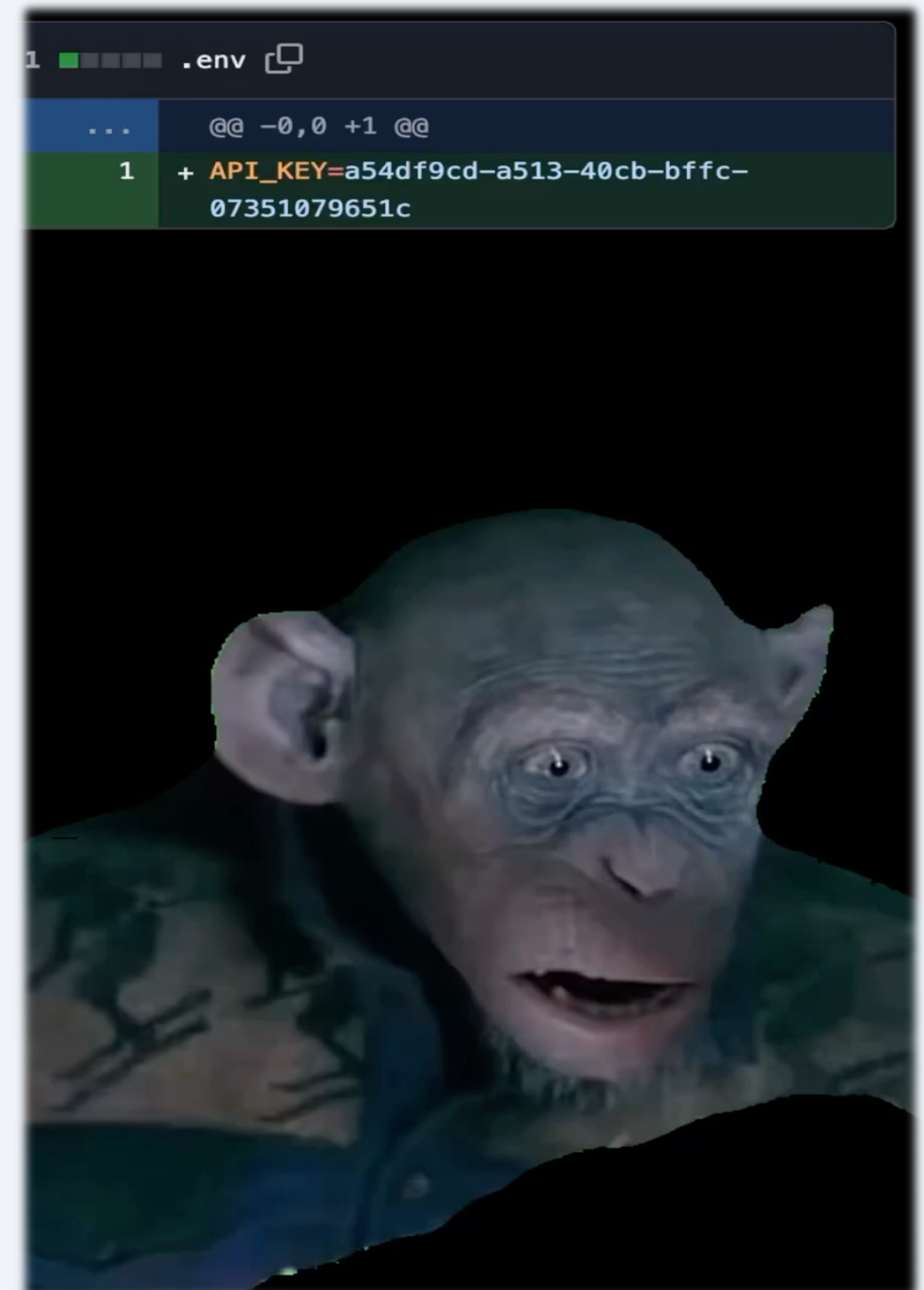
Зачем security

01 Собственный код может содержать уязвимости

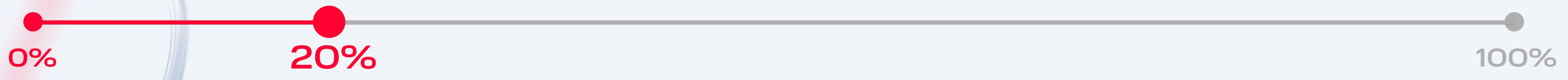
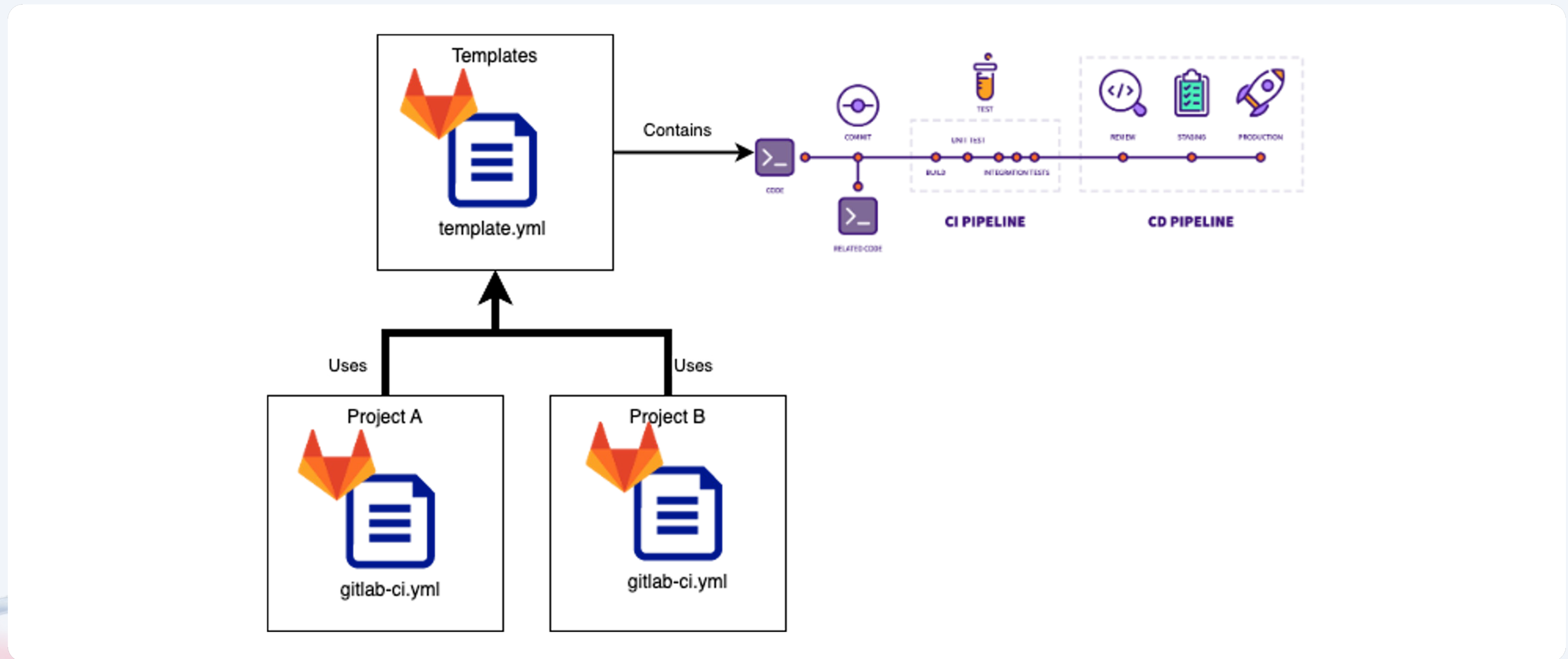
02 Заимствованный код, тоже может содержать уязвимости

03 Окружение тоже может содержать уязвимости

04 Человеческий фактор



Gitlab CI Security templates



Gitlab CI Security templates

SAST

```
stages:
```









- build
- test *#needed for template*
- security
- deploy

```
include:
```

- template: Jobs/SAST.gitlab-ci.yml

Gitlab CI Security templates

SAST

Pipeline	Created by	Stages
<p>chore: added code #1427725416  release1  fcc56342</p> <p> latest</p>		<p> </p> <div data-bbox="2215 1116 3148 1416"><p>Stage: test</p><p> semgrep-sast </p></div>

Gitlab CI Security templates

Остальные шаблоны

```
stages:
```

- build
- test *#needed for template*
- security
- deploy

```
include:
```

- template: Jobs/SAST.gitlab-ci.yml
- template: Jobs/Container-Scanning.gitlab-ci.yml
- template: Jobs/Secret-Detection.gitlab-ci.yml
- template: Jobs/SAST-IaC.latest.gitlab-ci.yml
- template: Jobs/Dependency-Scanning.gitlab-ci.yml

Gitlab CI Security templates

Остальные шаблоны

Status	Pipeline	Created by	Stages
Running	chore: added sec templates: #1427730434 release1 - f4d163fd latest		
Passed 00:01:16 4 minutes ago	chore: added code #1427725416 release1 - fcc56342		
Passed	chore: rest		

Stage: test









- container_scanning
- kics-iac-sast
- semgrep-sast
- secret_detection

Gitlab CI Security templates

Результаты

Artifacts

Total artifacts size 688.05 KiB

<input type="checkbox"/>	Artifacts	Job	Size	Created	
<input type="checkbox"/>	▼ 2 files	 kics-iac-sast #1427730434 f4d163fd release1	38.28 KiB	28 minutes ago	  
<input type="checkbox"/>	gl-sast-report.json <code>sast</code>			34.58 KiB	 
<input type="checkbox"/>	job.log <code>trace</code>			3.70 KiB	 

Gitlab CI Security templates

Результаты

gl-sast-report (10).json

C: > Users > Work > Downloads > gl-sast-report (10).json > scan > scanner

```
1  {
2    "version": "15.1.4",
3    "vulnerabilities": [
4      {
5        "id": "1123c58da1b9278254607db97bc4ece1b65614e38851126eca9c1ce652a1cc4a",
6        "category": "sast",
7        "name": "Missing User Instruction",
8        "description": "A user should be specified in the dockerfile, otherwise the image will run as root",
9        "cve": "kics_id:fd54f200-402c-4333-a5a4-36ef6709af2f:1:0",
10       "severity": "Critical",
11       "scanner": {
12         "id": "kics",
13         "name": "kics"
14       },
15       "location": {
```

Gitlab CI Security templates. Profit

01 Подключается в одну строку

Gitlab CI Security templates. Profit

01 Подключается в одну строку

02 Не требует тонкой настройки

Gitlab CI Security templates. Profit

01 Подключается в одну строку

02 Не требует тонкой настройки

03 Есть для разных сканеров

Gitlab CI Security templates. Profit

✔ Подключается в одну строку

✔ Не требует тонкой настройки

✔ Есть для разных сканеров

❗ **НО ВСЕГДА ЕСТЬ НЮАНС**

YES,



BUT



Проблемы шаблонов

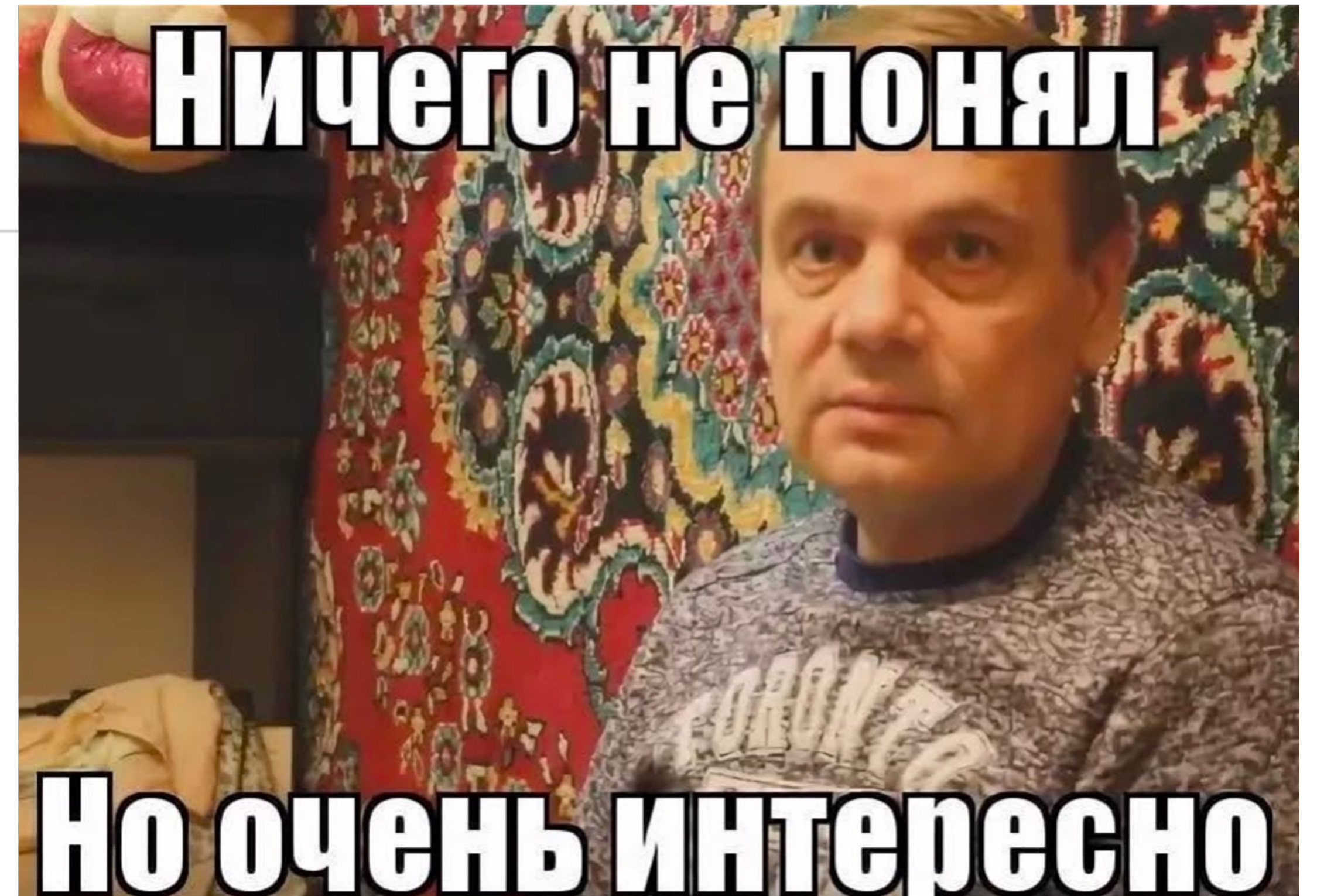
⚠ Может не работать из коробки

```
16 $ gtcs scan
17 [INFO] [2024-08-26 17:11:05 +0000] [container-scanning] > Remediation is disabled; /builds/off.moiseev.andrey/devoops/Dockerfile ca
nnot be found. Have you set `GIT_STRATEGY` and
18 `CS_DOCKERFILE_PATH`?
19 See https://docs.gitlab.com/ee/user/application\_security/container\_scanning/#solutions-for-vulnerabilities-auto-remediation
20 [INFO] [2024-08-26 17:11:06 +0000] [container-scanning] > Scanning container from registry registry.gitlab.com/off.moiseev.andrey/d
evoops/release1:f4d163fd7638f6b12b792e3f852eae2d762ec842 for vulnerabilities with severity level UNKNOWN or higher, with gcs 7.3.5 an
d Trivy Version: 0.52.1, advisories updated at 2024-08-26T12:12:59+00:00
21 [ERROR] [2024-08-26 17:11:07 +0000] [container-scanning] > Scanner has not created a file with results (tmp.json)
```

S Проблемы шаблонов

⚠ Может не работать из коробки

⚠ Все-таки придется разобраться в чужом коде



Gitlab CI Security templates. Выводы

01 Неуправляемый флоу сканеров



s Gitlab CI Security templates. Выводы

01 Неуправляемый флоу сканеров

02 Зачастую сложная реализация



s Gitlab CI Security templates. Выводы

01 Неуправляемый флоу сканеров

02 Зачастую сложная реализация

03 Нужен кастом под себя

s Gitlab CI Security templates. Выводы

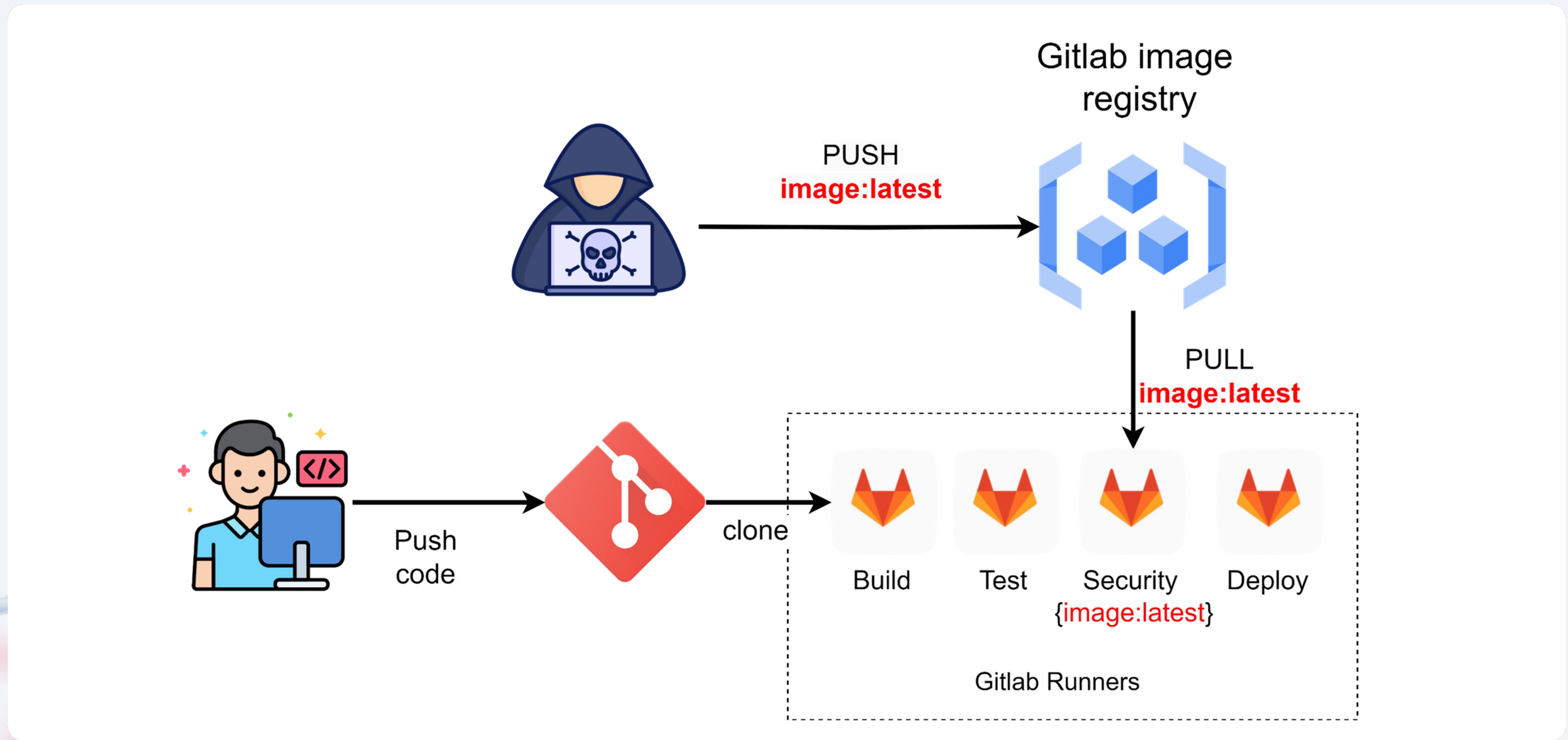
01 Неуправляемый флоу сканеров

02 Зачастую сложная реализация

03 Нужен кастом под себя

04 Прямая зависимость от gitlab registry

Dependency confusion

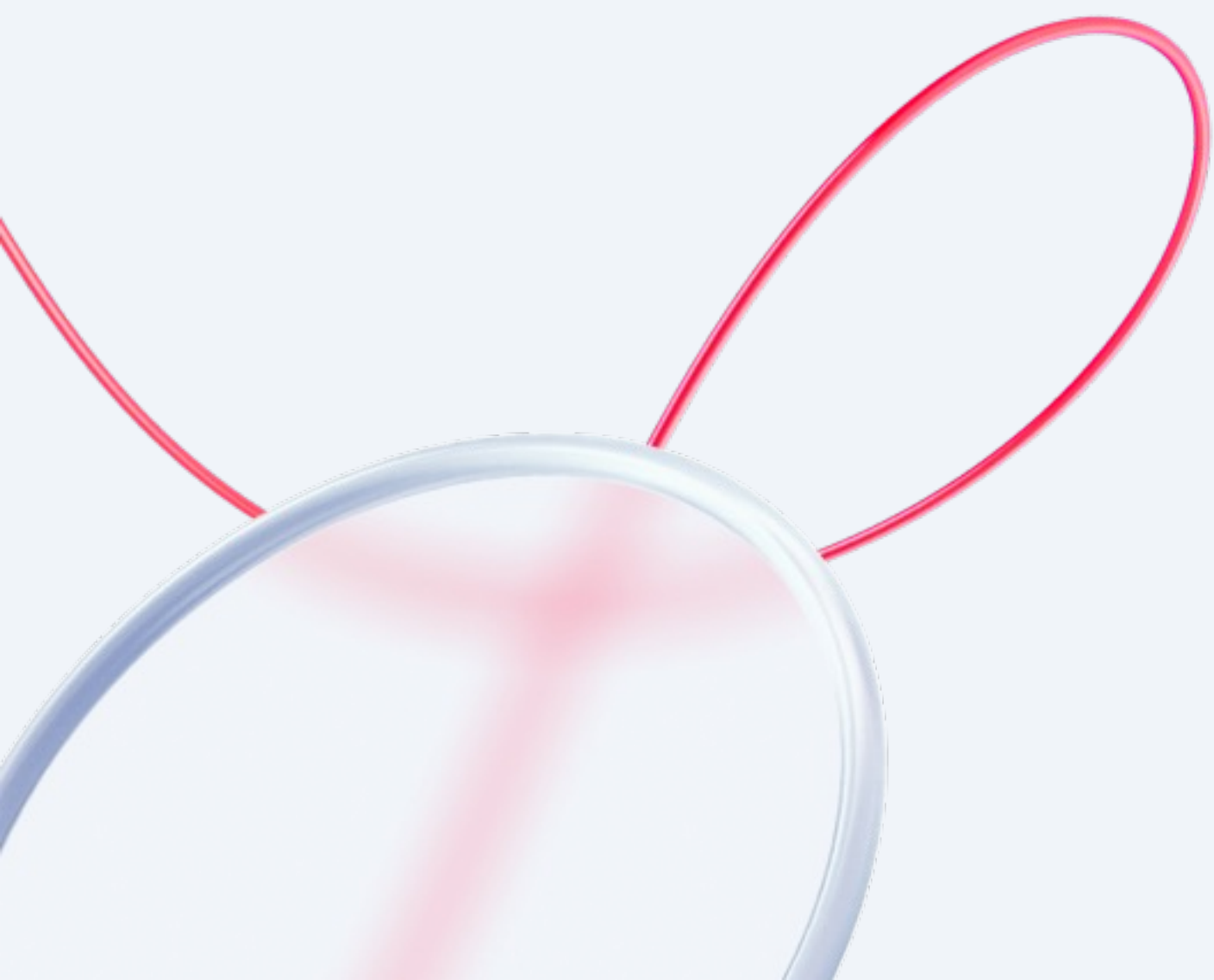


Dependency confusion

```
.ds-analyzer:  
  extends: dependency_scanning  
  allow_failure: true  
  variables:  
    # DS_ANALYZER_IMAGE is an undocumented variable used internally to allow QA to  
    # override the analyzer image with a custom value. This may be subject to change or  
    # breakage across GitLab releases.  
    DS_ANALYZER_IMAGE: "$SECURE_ANALYZERS_PREFIX/$DS_ANALYZER_NAME:$DS_MAJOR_VERSION"  
    # DS_ANALYZER_NAME is an undocumented variable used in job definitions  
    # to inject the analyzer name in the image name.  
    DS_ANALYZER_NAME: ""  
  image:  
    name: "$DS_ANALYZER_IMAGE$DS_IMAGE_SUFFIX"
```

M W
S

Что делаем?




Создаём свой шаблон





Наш шаблон

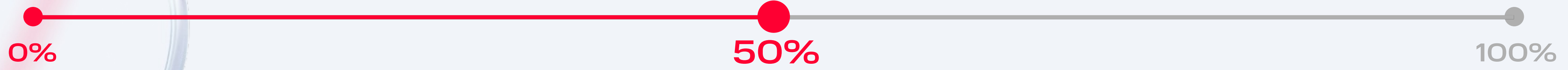
andrey moiseev / Security-template

main security-template / .gitlab / templates / .securiry-template.yml

 chore: init
Andrey Moiseev authored just now

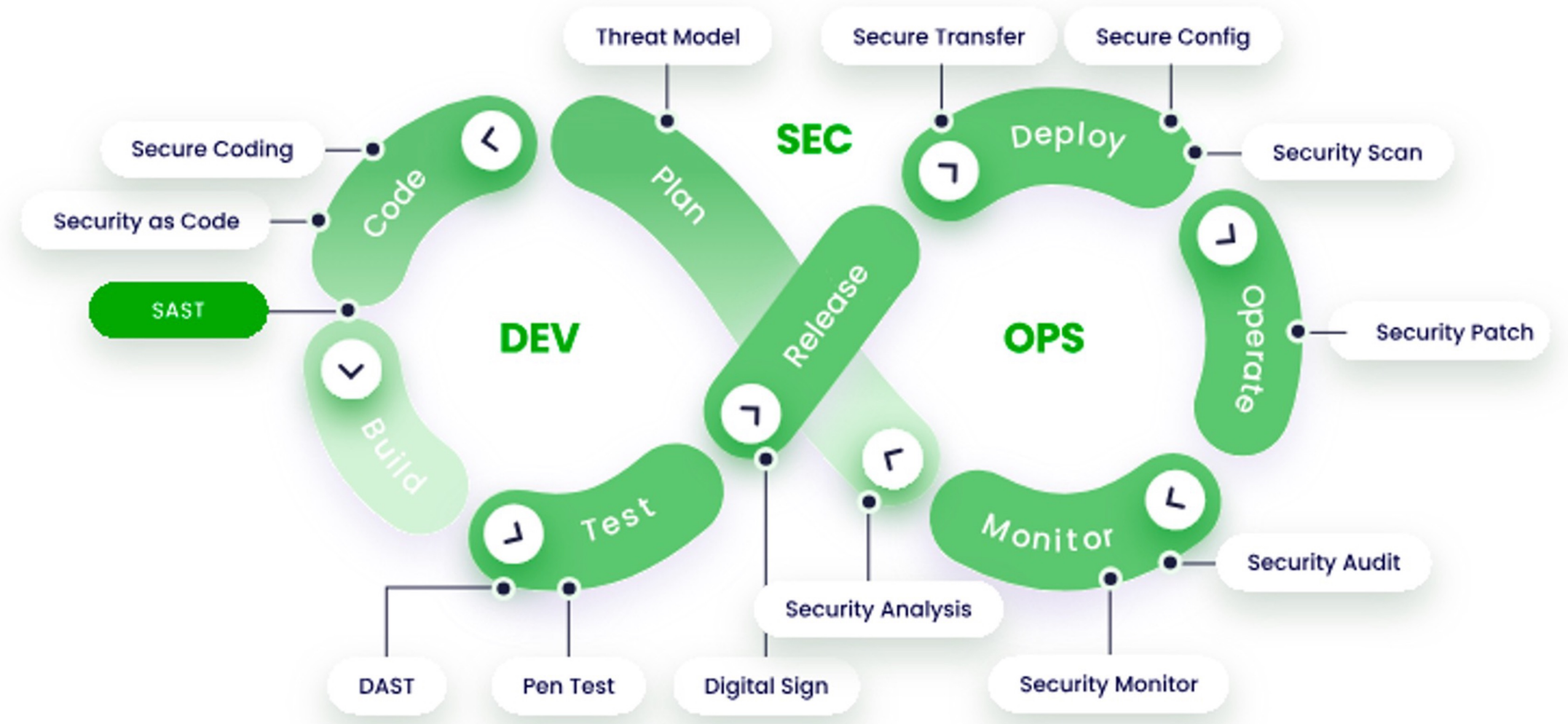
 .securiry-template.yml  317 B

```
1 # Security jobs
2 variables:
3
4
5 sast:
6   allow_failure: true
7   stage: security
8
9 secret_detection:
10  allow_failure: true
11  stage: security
```



Что нам нужно

01 Перечень security практик, которые внедряем



Что нам нужно

01 Перечень security практик, которые внедряем

02 Собственные образы для сканеров

The screenshot shows a project admin interface for 'public-security-tools'. At the top, there is a navigation bar with tabs: Summary, Repositories (selected), Members, Labels, Scanner, P2P Preheat, Policy, and Robot Account. Below the navigation bar, there is a 'DELETE' button. The main content is a table with the following data:

<input type="checkbox"/>	Name	Artifacts
<input type="checkbox"/>	public-security-tools/cdxgen	1
<input type="checkbox"/>	public-security-tools/semgrep	1
<input type="checkbox"/>	public-security-tools/gemnasium	2
<input type="checkbox"/>	public-security-tools/dockle	1
<input type="checkbox"/>	public-security-tools/kics	1
<input type="checkbox"/>	public-security-tools/nodejsscan	2

Что нам нужно

01 Перечень security практик,
которые внедряем

02 Собственные образы
для сканеров

03 Внутреннее registry,
чтобы их хранить




Что нам нужно

01 Перечень security практик, которые внедряем

02 Собственные образы для сканеров

03 Внутреннее registry, чтобы их хранить

04 Реализация security job

```
3 ▶ public class Main {  
4 ▶     public static void main(String[] args)  
5         System.out.println("Hello world!")  
6  
7      //TODO Add some useful code here  
8
```

Перечень практик

01 Secret management — GitLeaks



OWASP Top 10:2021

A01:2021 – Broken Access Control

Перечень практик

01 Secret management — GitLeaks

02 SCA — Gemnasium

☰ OWASP Top 10:2021

A06:2021 – Vulnerable and Outdated Components

Перечень практик

01 Secret management — GitLeaks

02 SCA — Gemnasium

03 SAST — nodejsscan

04 Additional SAST — Semgrep

05 IAC SAST — Kics

☰ OWASP Top 10:2021

A01:2021 – Broken Access Control

☰ OWASP Top 10:2021

A02:2021 – Cryptographic Failures

☰ OWASP Top 10:2021

A05:2021 – Security Misconfiguration

Secret management



Хакер

<https://xakep.ru>

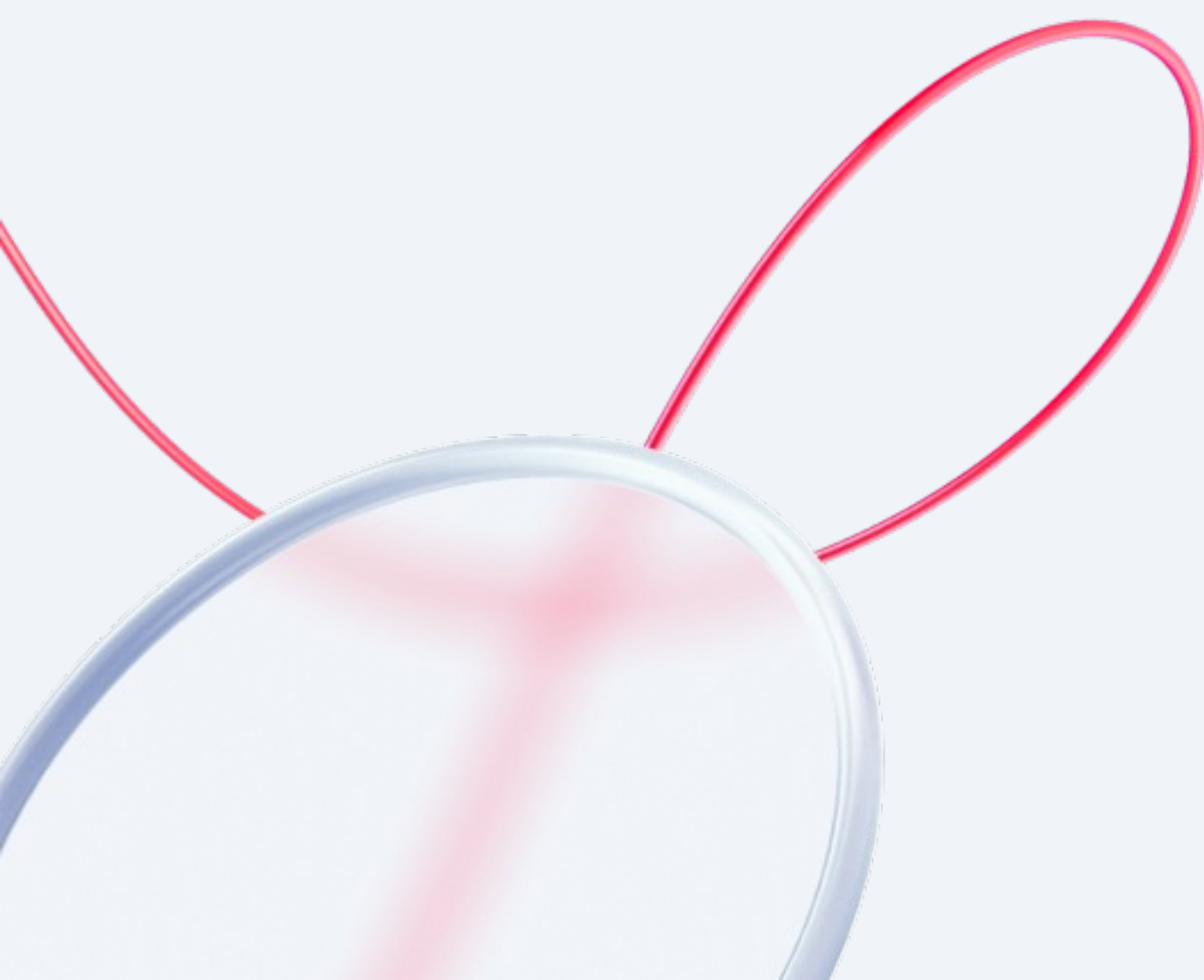
Автомобили Kia можно было взломать удаленно, зная

3 дня назад — «Эти атаки могли выполняться удаленно на любом оборудованном




Kia

Nokia



Secret management

 Хакер
<https://xakep.ru>

Автомобили Kia можно было взломать удаленно, зная
3 дня назад — «Эти атаки могли выполняться удаленно на любом оборудованном»

```
POST /apps/services/owners/apigwServlet.html HTTP/2  
Host: owners.kia.com  
Httpmethod: GET  
Apiurl: /door/unlock  
Servicetype: postLoginCustomer  
Cookie: JSESSIONID=SESSION_TOKEN;
```



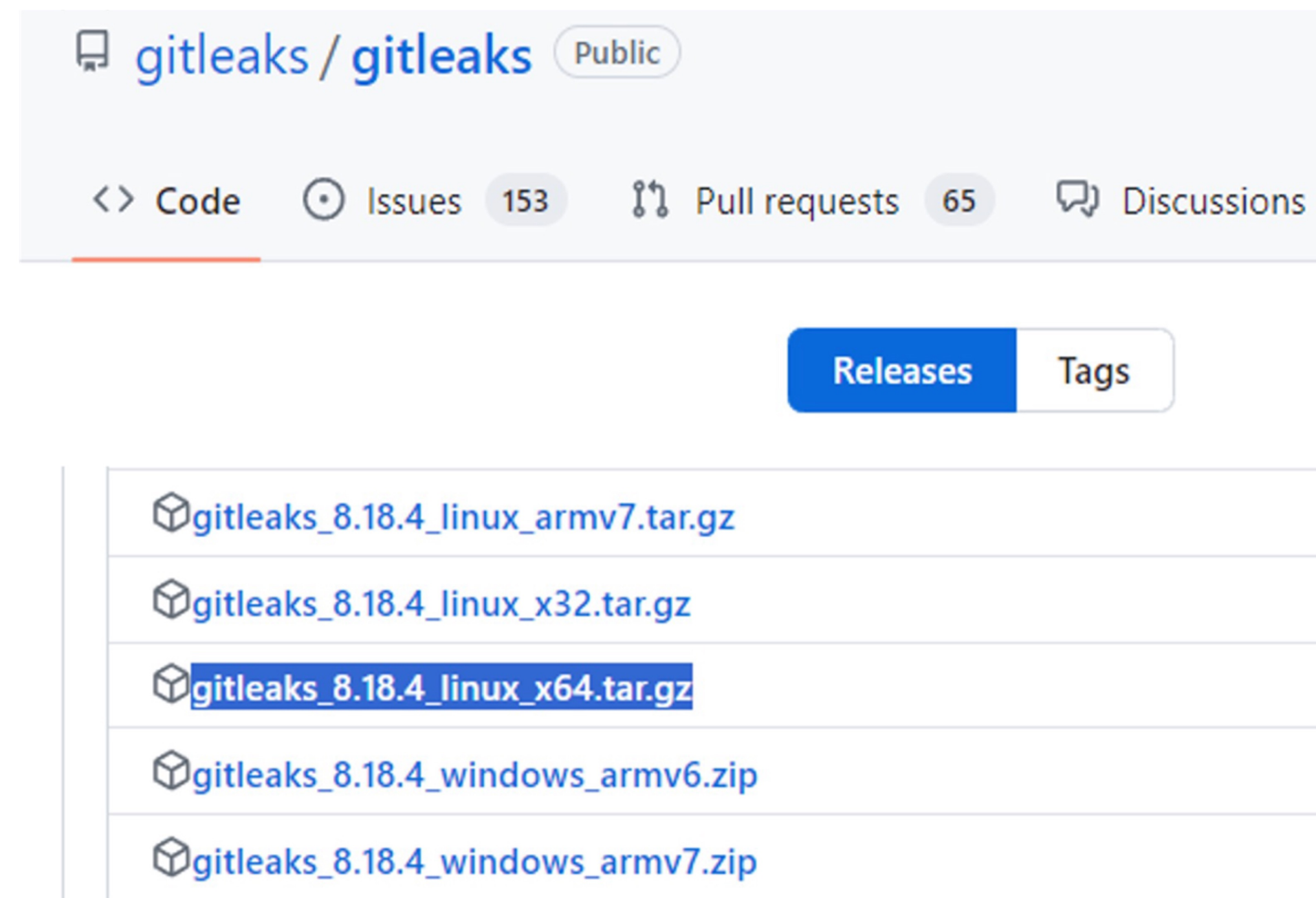
Kia

Nokia



s Secret management

01 Берем сорсы
сканера с github.com



gitleaks / gitleaks Public

<> Code Issues 153 Pull requests 65 Discussions

Releases Tags

- gitleaks_8.18.4_linux_armv7.tar.gz
- gitleaks_8.18.4_linux_x32.tar.gz
- gitleaks_8.18.4_linux_x64.tar.gz**
- gitleaks_8.18.4_windows_armv6.zip
- gitleaks_8.18.4_windows_armv7.zip

s Secret management

01 Берем сорсы
сканера с github.com

02 Пишем свой докерфайл

Secret management

```
1 FROM alpine:3.18.2
2
3 ARG GITLEAKS_VERSION=8.18.1
4 ARG GITLEAKS_SHA256="3e157a26081e296d4cb94ef0d87441c9afc5f392cb02957656dd5cf7aaf6c9"
```

Secret management

```
1 FROM alpine:3.18.2
2
3 ARG GITLEAKS_VERSION=8.18.1
4 ARG GITLEAKS_SHA256="3e157a26081e296d4cb94ef0d87441c9afc5f392cb02957656dd5cf7aaf6c9"
5
6 RUN adduser -D gitleaks && \
7 apk update && apk upgrade --no-cache && \
8 apk add --no-cache git curl &&\
9 wget https://github.com/gitleaks/gitleaks/releases/download/v${GITLEAKS_VERSION}/gitleaks_${GITLEAKS_VERSION}_
10 sha256sum /tmp/gitleaks.tar.gz > sha256sum.txt && \
```

Secret management

```
1 FROM alpine:3.18.2
2
3 ARG GITLEAKS_VERSION=8.18.1
4 ARG GITLEAKS_SHA256="3e157a26081e296d4cb94ef0d87441c9afc5f392cb02957656dd5cfeb7aaf6c9"
5
6 RUN adduser -D gitleaks && \
7 apk update && apk upgrade --no-cache && \
8 apk add --no-cache git curl &&\
9 wget https://github.com/gitleaks/gitleaks/releases/download/v${GITLEAKS_VERSION}/gitleaks_${GITLEAKS_VERSION}_
10 sha256sum /tmp/gitleaks.tar.gz > sha256sum.txt && \
11 cat sha256sum.txt && \
12 echo ${GITLEAKS_SHA256} && \
13 echo "${GITLEAKS_SHA256} /tmp/gitleaks.tar.gz" | sha256sum -c && \
14 tar xf /tmp/gitleaks.tar.gz -C /tmp gitleaks && \
15 rm -f /tmp/gitleaks.tar.gz && \
16 mv /tmp/gitleaks /usr/local/bin/gitleaks
```

Secret management

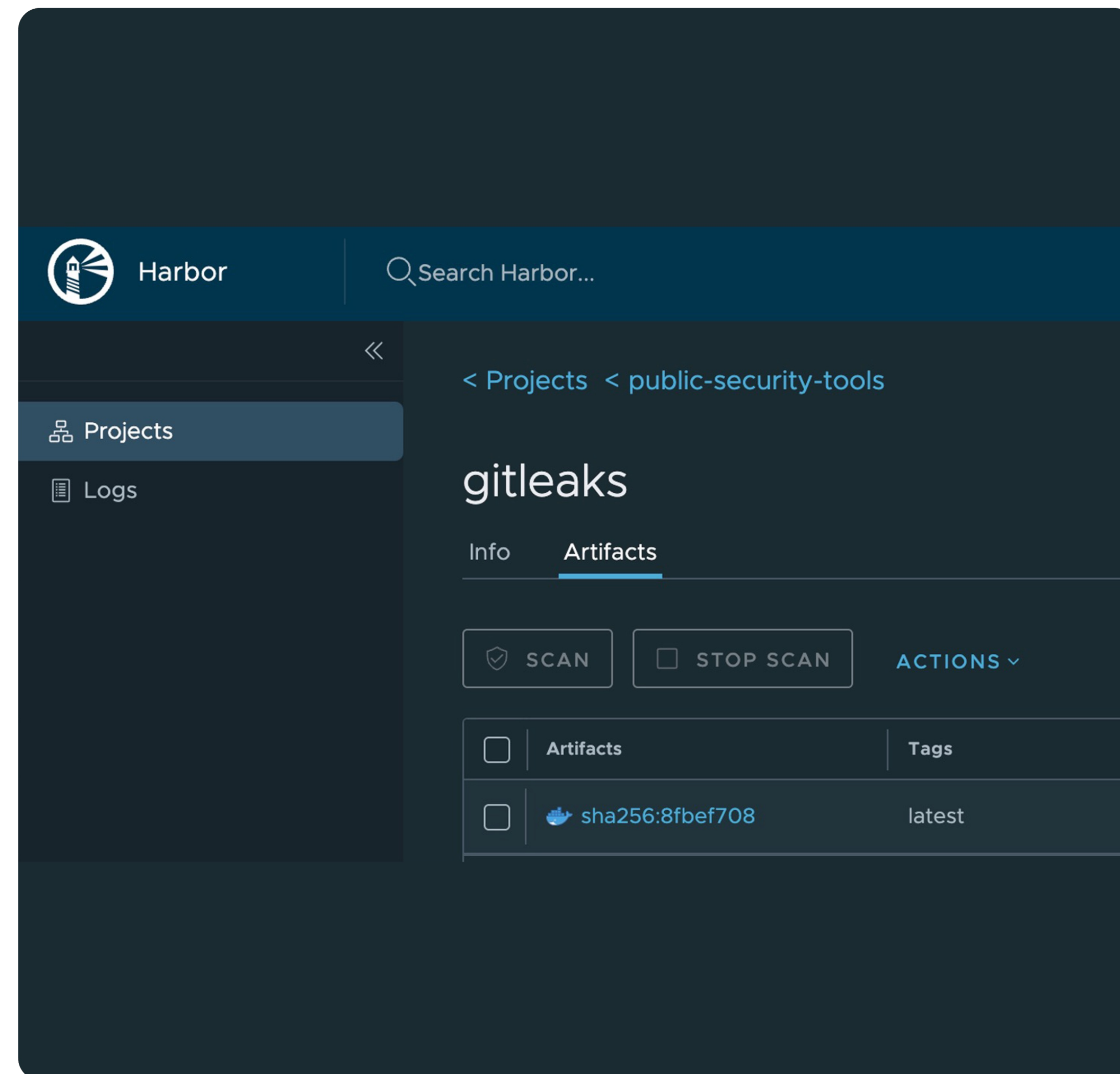
```
1 FROM alpine:3.18.2
2
3 ARG GITLEAKS_VERSION=8.18.1
4 ARG GITLEAKS_SHA256="3e157a26081e296d4cb94ef0d87441c9afc5f392cb02957656dd5cfeb7aaf6c9"
5
6 RUN adduser -D gitleaks && \
7 apk update && apk upgrade --no-cache && \
8 apk add --no-cache git curl &&\
9 wget https://github.com/gitleaks/gitleaks/releases/download/v${GITLEAKS_VERSION}/gitleaks_${GITLEAKS_VERSION}_
10 sha256sum /tmp/gitleaks.tar.gz > sha256sum.txt && \
11 cat sha256sum.txt && \
12 echo ${GITLEAKS_SHA256} && \
13 echo "${GITLEAKS_SHA256} /tmp/gitleaks.tar.gz" | sha256sum -c && \
14 tar xf /tmp/gitleaks.tar.gz -C /tmp gitleaks && \
15 rm -f /tmp/gitleaks.tar.gz && \
16 mv /tmp/gitleaks /usr/local/bin/gitleaks
17
18 USER gitleaks
19 WORKDIR /home/gitleaks
20
21 RUN git config --global --add safe.directory '*'
22
23 ENTRYPOINT [""]
```

s Secret management

01 Берем сорсы сканера с github.com

02 Пишем свой докерфайл

03 Пушим во внутренний режистри



И конечно YAML

```
! security-template.yml
1  secret_detection:
2    stage: security
3    image:
4      name: "registry.resources.cloud.mts.ru/public-security-tools/gitleaks"
5    variables:
6      GIT_DEPTH: "50"
7      SECRET_DETECTION_HISTORIC_SCAN: "true"
8    services: []
9    allow_failure: true
10   tags:
11     - devops-docker
12   script:
13     - gitleaks detect --exit-code "0" --source . --log-opts="--all" --report-path gitleaks-results.json
14   artifacts:
15     paths:
16       - gitleaks-*.*
```

Secret management

chore: added security

🕒 2 jobs for [security](#)
in 9 seconds and was queued for 2 seconds

🚩 latest

🔗 [83492022](#) 📄

🔗 No related merge requests found.

Pipeline Needs Jobs 2 Tests 0

security

✅ secret_detection 🔄

Соберём остальные сканеры

master ▾ devoops_sectools / + ▾

Find file Web IDE ↓ Clone ▾

README Add LICENSE Add CHANGELOG Add CONTRIBUTING Add Kubernetes cluster Set up CI/CD Add Wiki Configure Integrations

Name	Last commit	Last update
📁 dockle	chore: sec	2 minutes ago
📁 gemnasium	chore: sec	2 minutes ago
📁 gitleaks	chore: sec	2 minutes ago
📁 kics	chore: sec	2 minutes ago
📁 kubesecc	chore: sec	2 minutes ago
📁 nodejsscan	chore: sec	2 minutes ago
📁 security-code-scan	chore: sec	2 minutes ago
📁 semgrep	chore: sec	2 minutes ago
📄 README.md	Initial commit	14 hours ago

Допишем нужные джобы

```
.cyclonedx-reports:  
  artifacts:  
    paths:  
      - "**/gl-sbom-*.cdx.json"  
      - gl-dependency-scanning-report.json  
  reports:  
    cyclonedx: "**/gl-sbom-*.cdx.json"  
  
gemnasium-dependency_scanning:  
  extends:  
    - .ds-analyzer  
    - .cyclonedx-reports  
  variables:  
    DS_ANALYZER_NAME: "gemnasium"  
    GEMNASIUM_LIBRARY_SCAN_ENABLED: "true"
```

Допишем нужные джобы

```
112 nodejs-scan-sast:
113   allow_failure: true
114   stage: security
115   tags:
116     - devoops-gitleaks
117   image:
118     name: "registry.resources.cloud.mts.ru/public-security-tools/nodejsscan"
119   script:
120     - njsscan . --json -o ${PWD}/nodejs-results.json
121   artifacts:
122     paths:
123       - nodejs-results.json
124
125
126 kics-iac-sast:
127   allow_failure: true
128   stage: security
129   image:
130     name: "registry.resources.cloud.mts.ru/public-security-tools/kics"
131     entrypoint: [""]
132   tags:
133     - devoops-gitleaks
134   script:
135     - kics scan -s -p ${PWD} -o ./ --disable-secrets --report-formats json -o ${PWD} --output-name kics-results
136   artifacts:
137     name: kics-results.json
138     paths:
139       - kics-results.json
140   when: always
```

Еще немного YAML



```
81 semgrep:
82   stage: security
83   image:
84     name: "registry.resources.cloud.mts.ru/public-security-tools/semgrep"
85   artifacts:
86     paths:
87       - gl-sast-report.json
88   tags:
89     - devoops-gitleaks
90   variables:
91     SEARCH_MAX_DEPTH: '4'
92   rules:
93     - if: true
94       exists:
95         - '**/*.py'
96         - '**/*.js'
97         - '**/*.jsx'
```

M W
S

Ну вы поняли













Ура – шаблон работает!

```
 .gitlab-ci.yml  123 bytes  
1 stages:  
2   - build  
3   - test #needed for template  
4   - security  
5   - deploy  
6  
7 include:  
8   - local: 'security-template.yml'  
9  
10
```

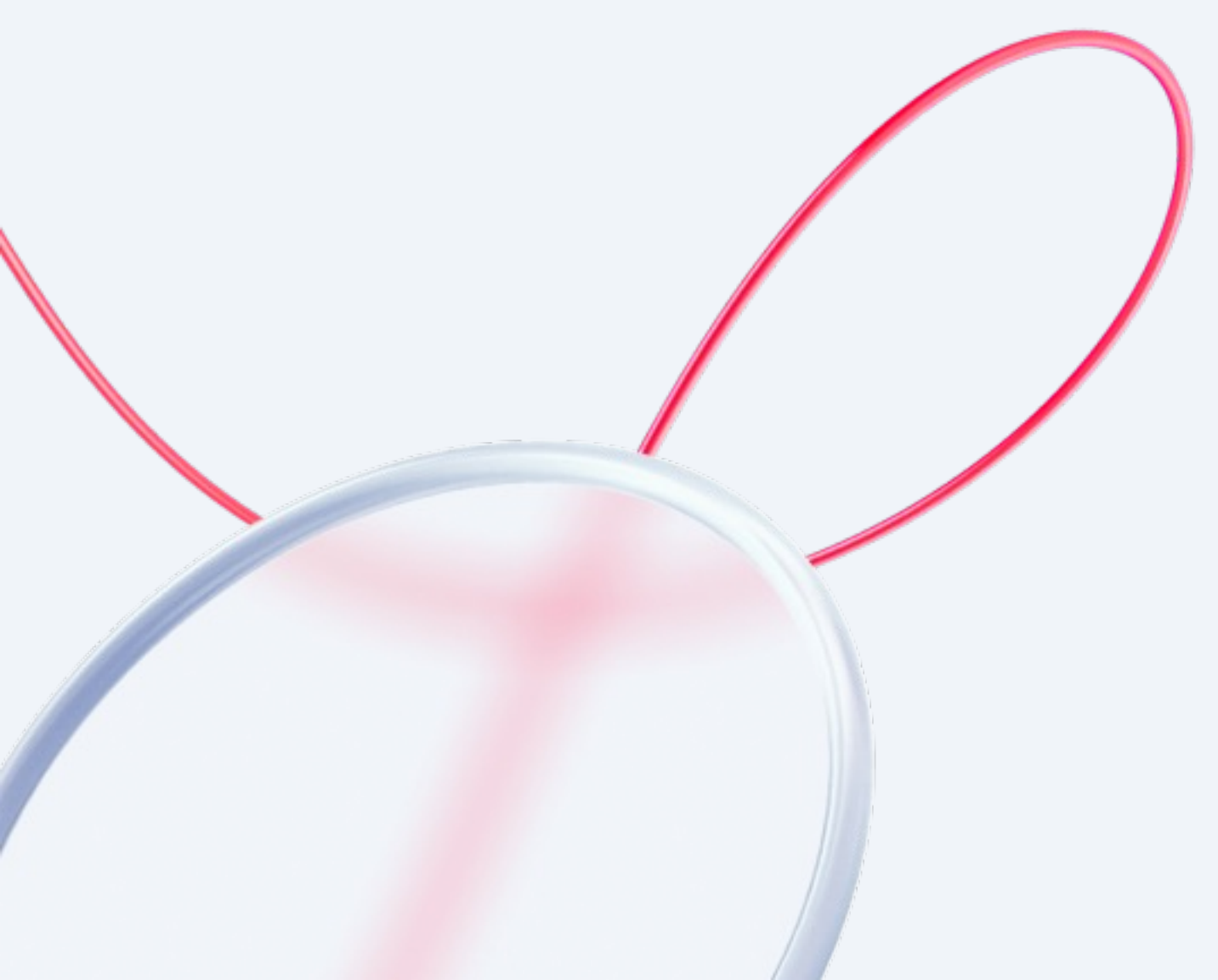
Pipeline Needs Jobs **5** Tests **0**

security

-  gemnasium-dependency_scanning 
-  kics-iac-sast 
-  nodejs-scan-sast 
-  secret_detection 
-  semgrep 

M W
S

Но есть проблемы



Проблемы


01 Всё помещается в gitlab artifacts

❗ failed Job #489455 in pipeline #78059 for 7277a482 from security by  Андрей Моисеев 3 minutes ago

Artifacts

✅ passed Job #489454 in pipeline #78059 for 7277a482 from security by  Андрей Моисеев 4 minutes ago


Name

 kics-results.json

Artifacts

Name

Size

 nodejs-results.json

74

Проблемы

01 Всё помещается в gitlab artifacts

02 Результаты AS IS

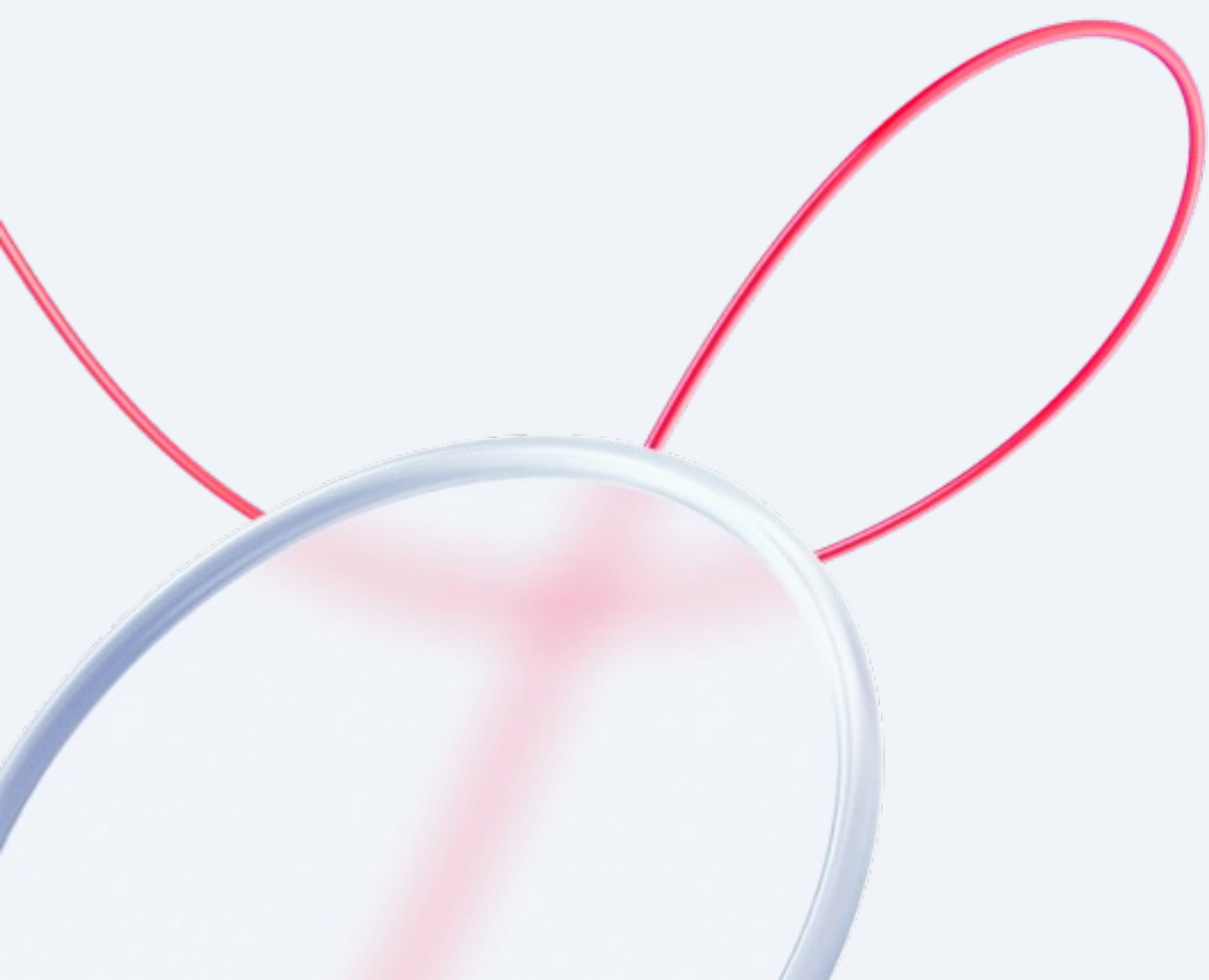
```
1013 "query_name": "Type Has Invalid Keyword (v3)",
1014 "query_id": "a9228976-10cf-4b5f-b902-9e962aad037a",
1015 "query_url": "https://swagger.io/specification/#schema-object",
1016 "severity": "INFO",
1017 "platform": "OpenAPI",
1018 "category": "Structure and Semantics",
1019 "experimental": false,
1020 "description": "Schema Object define type should not use a keyword of another type",
1021 "description_id": "d7b8c860",
1022 "files": [
1023   {
1024     "file_name": "juice-shop-17.1.1/swagger.yml",
1025     "similarity_id": "b55ac523ed1086f992b2166d0c88b52df245bc9f4cb0f792d992c9d5e2acb8f1",
1026     "line": 38,
1027     "issue_type": "IncorrectValue",
1028     "search_key": "components.schemas.OrderConfirmation.properties.orderNo.uniqueItems"
```

Проблемы

01 Всё помещается в gitlab artifacts

02 Результаты AS IS

03 Много файндингов, пайплайнов



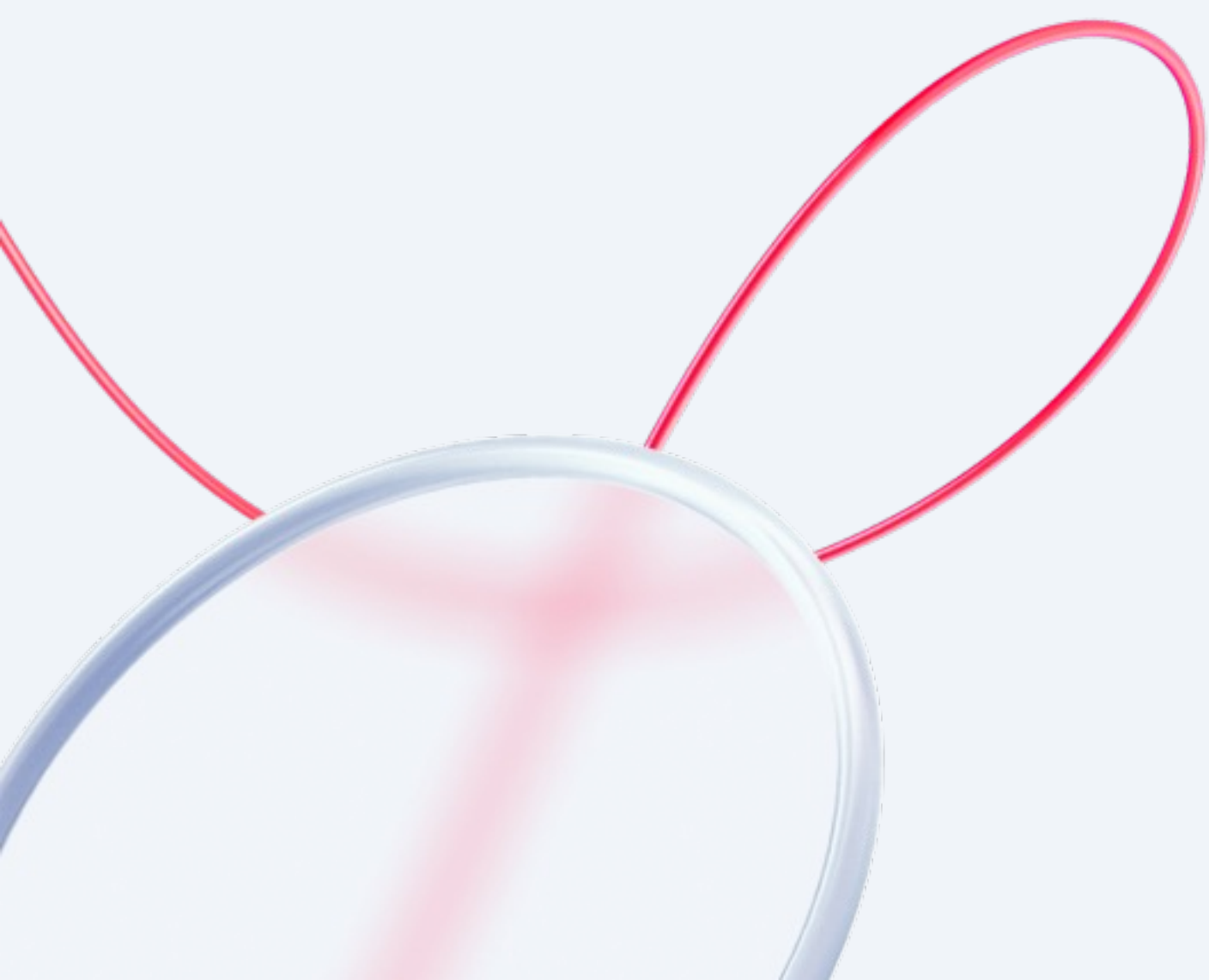
Проблемы

01 Всё помещается в gitlab artifacts

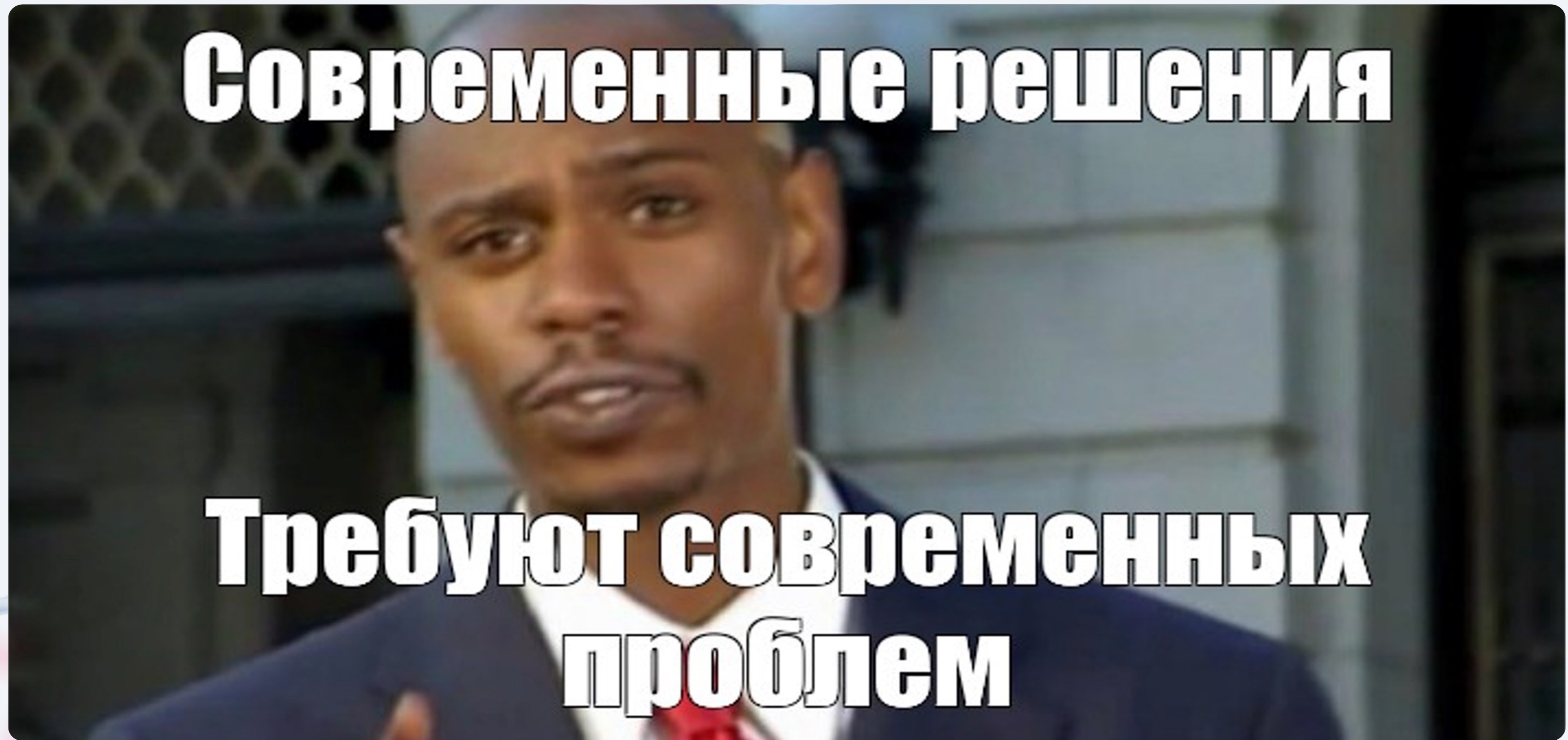
02 Результаты AS IS

03 Много файндингов, пайплайнов

04 Разработчики когда видят отчет →



Проблемы надо решать



Современные решения

**Требуют современных
проблем**

Что предпримем?

01 Используем минимальные наборы правил

 [semgrep / semgrep-rules](#) Public

[Code](#) [Issues 46](#) [Pull requests 2](#)



Что предпримем?

01 Используем минимальные наборы правил

02 Реализуем триаж



0%

70%

100%

S Что предпримем?

01 Используем минимальные наборы правил

02 Реализуем триаж

03 Согласуем SLA с разработчиками



0%

70%

100%

Что предпримем?

01 Используем минимальные наборы правил

02 Реализуем триаж

03 Согласуем SLA с разработчиками

04 Создадим удобный дашборд



Defect dojo

The screenshot displays the Defect Dojo Pro dashboard with the following components:

- Navigation Menu:** Includes sections for DASHBOARD, MANAGE (Products, Engagements, Tests, Findings, Endpoints), IMPORT (Add Findings), PLUGINS (License Manager, API Connectors), and BETA (Changelog).
- Summary Cards:** A grid of 12 cards showing key metrics:
 - Passing Products: 0
 - Active Engagements: 2
 - Last 7 Days: 1770
 - Closed In Last 7 Days: 0
 - Accepted In Last 7 Days: 0
 - Failing Products: 0
 - Approaching SLA Violation: 0
 - Active Critical Findings: 0
 - Active High Findings: 1
 - Active Medium Findings: 1769
 - Active Low Findings: 0
 - Not Scanned in 3 Months: 0
- Historical Finding Severity:** A pie chart showing 100% of findings are of Medium severity.
- Reported Finding Severity:** A bar chart showing 1769 findings, all of which are of Medium severity.

0%

95% 100%

И снова YAML Development

```
defectdojo.template.yml 5.09 KiB
1 variables:
2   DEFECTDOJO_ENGAGEMENT_PERIOD: 7
3   DEFECTDOJO_ENGAGEMENT_STATUS: "Not Started"
4   DEFECTDOJO_ENGAGEMENT_BUILD_SERVER: "null"
5   DEFECTDOJO_ENGAGEMENT_SOURCE_CODE_MANAGEMENT_SERVER: "null"
6   DEFECTDOJO_ENGAGEMENT_ORCHESTRATION_ENGINE: "null"
7   DEFECTDOJO_ENGAGEMENT_DEDUPLICATION_ON_ENGAGEMENT: "false"
8   DEFECTDOJO_ENGAGEMENT_THREAT_MODEL: "true"
9   DEFECTDOJO_ENGAGEMENT_API_TEST: "true"
10  DEFECTDOJO_ENGAGEMENT_PEN_TEST: "true"
11  DEFECTDOJO_ENGAGEMENT_CHECK_LIST: "true"
12  DEFECTDOJO_NOT_ON_MASTER: "false"
13  DEFECTDOJO_SCAN_MINIMUM_SEVERITY: "Info"
14  DEFECTDOJO_SCAN_ACTIVE: "true"
15  DEFECTDOJO_SCAN_VERIFIED: "true"
16  DEFECTDOJO_SCAN_CLOSE_OLD_FINDINGS: "true"
17  DEFECTDOJO_SCAN_PUSH_TO_JIRA: "false"
```

И снова YAML Development

Напишем джобу для создания engagement

```
36 script:
37   - |
38     ENGAGEMENTID=`curl -k --fail --location --request POST "${DEFECTDOJO_URL}/api/v2/engagements/" \
39       --header "Authorization: Token ${DEFECTDOJO_TOKEN}" \
40       --header 'Content-Type: application/json' \
41       --data-raw "{
42         \"tags\": [\"GITLAB-CI\"],
43         \"name\": \"#${CI_PIPELINE_ID}\",
44         \"description\": \"${CI_COMMIT_DESCRIPTION}\",
45         \"version\": \"${CI_COMMIT_REF_NAME}\",
46         \"first_contacted\": \"${TODAY}\",
47         \"target_start\": \"${TODAY}\",
```

И снова YAML Development

Напишем джобу для загрузки отчета в engagement

```
96  script:
97    - |
98      curl -k --fail --location --request POST "${DEFECTDOJO_URL}/api/v2/import-scan/" \
99        --header "Authorization: Token ${DEFECTDOJO_TOKEN}" \
100       --form "scan_date=\"${TODAY}\"" \
101       --form "minimum_severity=\"${DEFECTDOJO_SCAN_MINIMUM_SEVERITY}\"" \
102       --form "active=\"${DEFECTDOJO_SCAN_ACTIVE}\"" \
103       --form "verified=\"${DEFECTDOJO_SCAN_VERIFIED}\"" \
104       --form "scan_type=\"${DEFECTDOJO_SCAN_TYPE}\"" \
105       --form "engagement=\"${DEFECTDOJO_ENGAGEMENTID}\"" \
106       --form "file=@${DEFECTDOJO_SCAN_FILE}" \
107       --form "close_old_findings=\"${DEFECTDOJO_SCAN_CLOSE_OLD_FINDINGS}\"" \
108       --form "push_to_jira=\"${DEFECTDOJO_SCAN_PUSH_TO_JIRA}\"" \
109       --form "test_type=\"${DEFECTDOJO_SCAN_TEST_TYPE}\"" \
110       --form "environment=\"${DEFECTDOJO_SCAN_ENVIRONMENT}\""
```

И снова YAML Development

Напишем интеграции с нашими сканерами

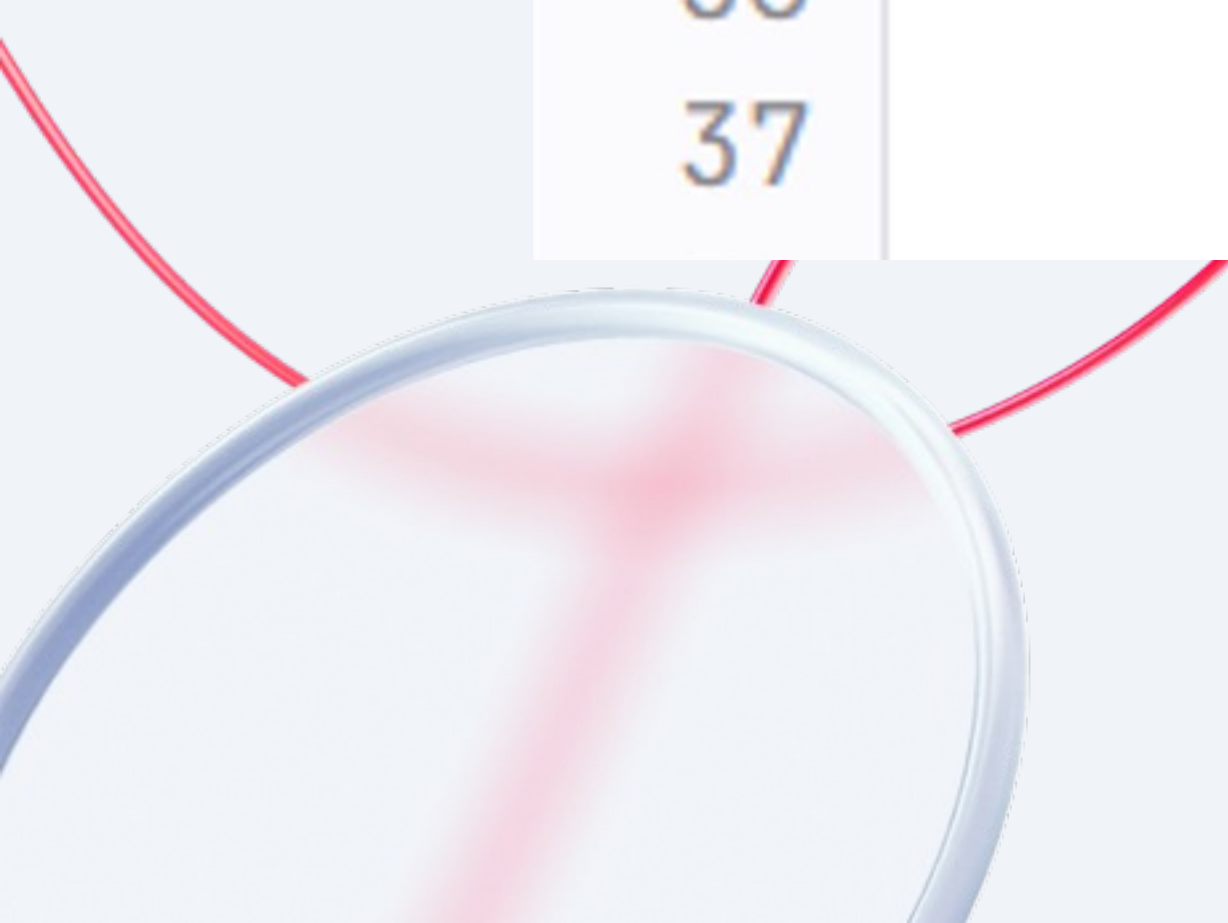
defectdojopublish.template.yml 7.50 KiB

Blame

```
1 gemnasium-defectdojo:
2   extends: .defectdojo_upload
3   needs: ["defectdojo_create_engagement", "gemnasium-dependency_scanning"]
4   variables:
5     DEFECTDOJO_SCAN_TYPE: "GitLab Dependency Scanning Report" #DD param used to know report type
6     DEFECTDOJO_SCAN_TEST_TYPE: "GitLab-CI Gemnasium"
7     DEFECTDOJO_SCAN_FILE: "gl-dependency-scanning-report.json"
8   rules:
9     - if: $SEC_DISABLED
10      when: never
11     - if: $SEC_ENABLE_BRANCHES
12      when: always
```

И снова YAML Development

```
31 kics-defectdojo:  
32   extends: .defectdojo_upload  
33   needs: ["defectdojo_create_engagement", "kics-iac-sast"]  
34   variables:  
35     DEFECTDOJO_SCAN_TYPE: "KICS Scan"  
36     DEFECTDOJO_SCAN_TEST_TYPE: "GitLab-CI KICS Scan"  
37     DEFECTDOJO_SCAN_FILE: "./kics-results.json"
```

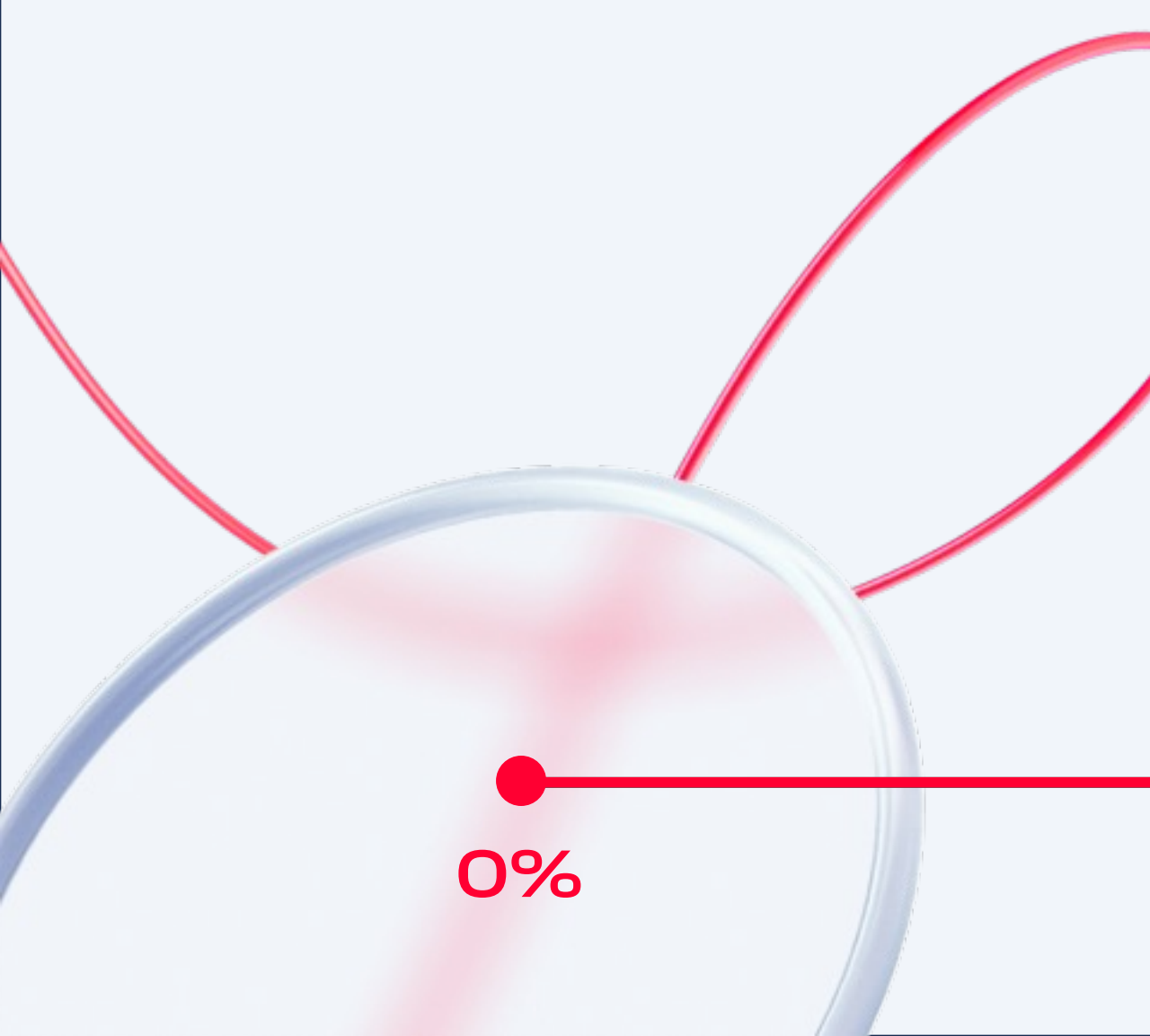


Всё работает

Title / Type	Date	Lead	Total Findings	Active (Verified)	Mitigated	Duplicates	Notes	Reimports
⋮ Dependency Check Scan	Sept. 11, 2024 - Sept. 11, 2024		77	77 (77)	0	0		0
⋮ Dockle Scan	Sept. 11, 2024 - Sept. 11, 2024		2	2 (2)	0	0		0
⋮ Dockle Scan	Sept. 11, 2024 - Sept. 11, 2024		2	2 (2)	0	0		0
⋮ GitLab Dependency Scanning Report	Sept. 11, 2024 - Sept. 11, 2024		55	55 (55)	0	0		0
⋮ GitLab Secret Detection Report	Sept. 11, 2024 - Sept. 11, 2024		1	1 (1)	0	0		0
⋮ Semgrep Scan (GitLab SAST Report)	Sept. 11, 2024 - Sept. 11, 2024		53	53 (53)	0	0		0
⋮ Trivy Scan (GitLab SAST Report)	Sept. 11, 2024 - Sept. 11, 2024		45	45 (45)	0	0		0
⋮ Trivy Scan (GitLab SAST Report)	Sept. 11, 2024 - Sept. 11, 2024		138	6 (6)	132	0		0
⋮ kics Scan (GitLab SAST Report)	Sept. 11, 2024 - Sept. 11, 2024		217	217 (217)	0	0		0
⋮ njsscan Scan (GitLab SAST Report)	Sept. 11, 2024 - Sept. 11, 2024		35	35 (35)	0	0		0

Showing entries 1 to 10 of 10

Page Size ▾

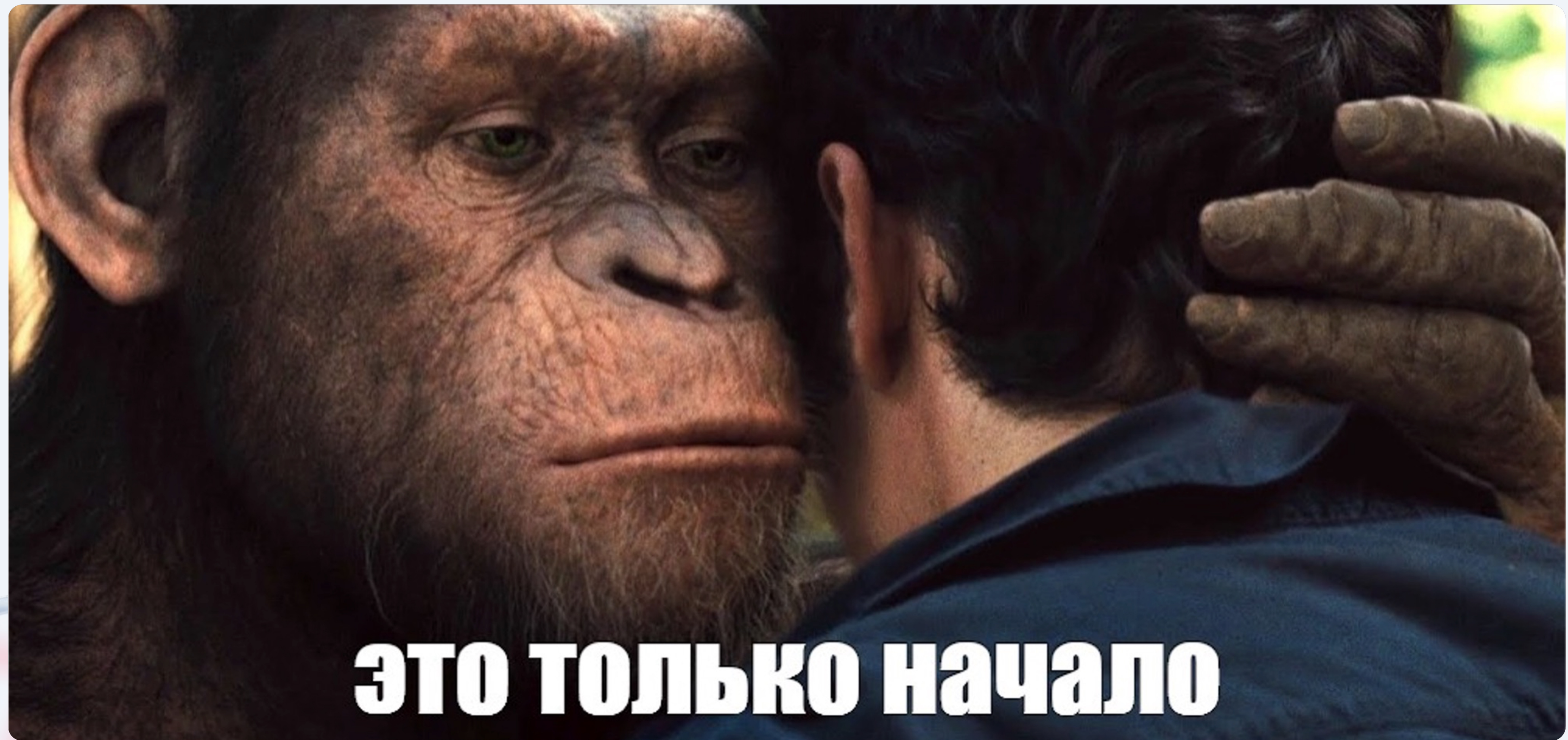


0%

100%

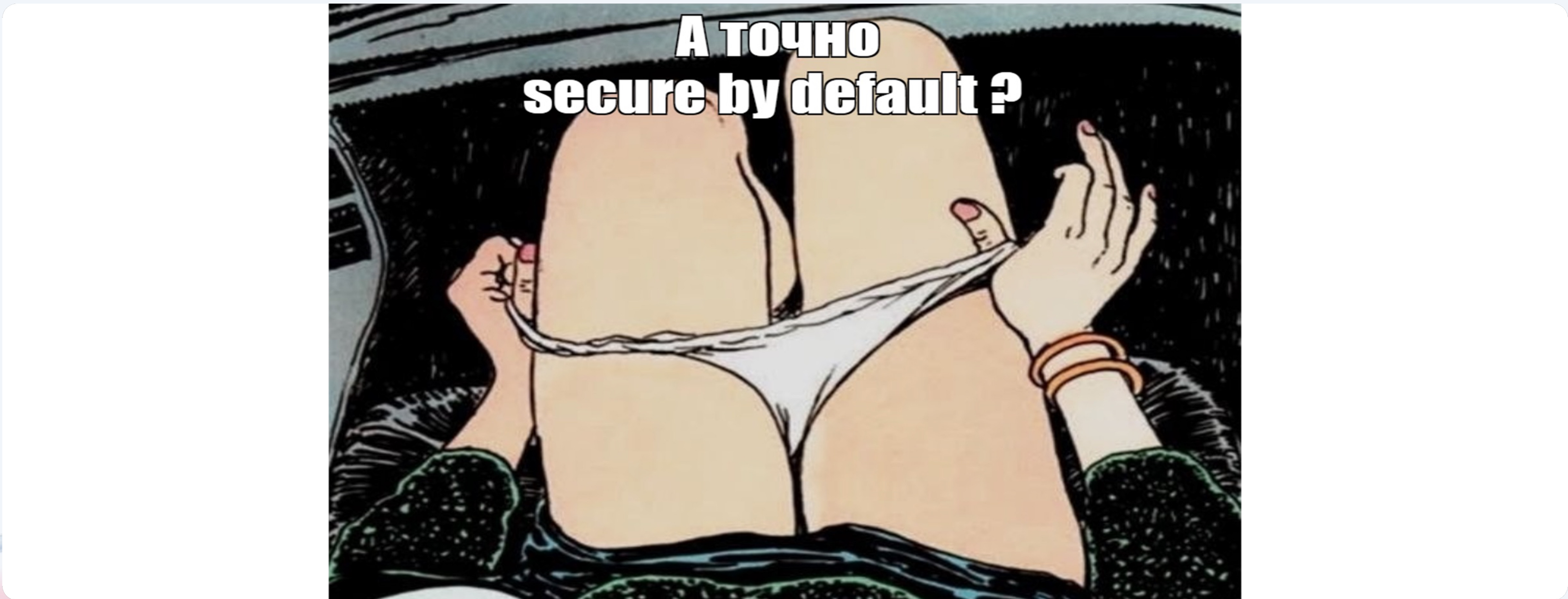
M W
S

И всё?



ЭТО ТОЛЬКО НАЧАЛО

Secure by default



0%

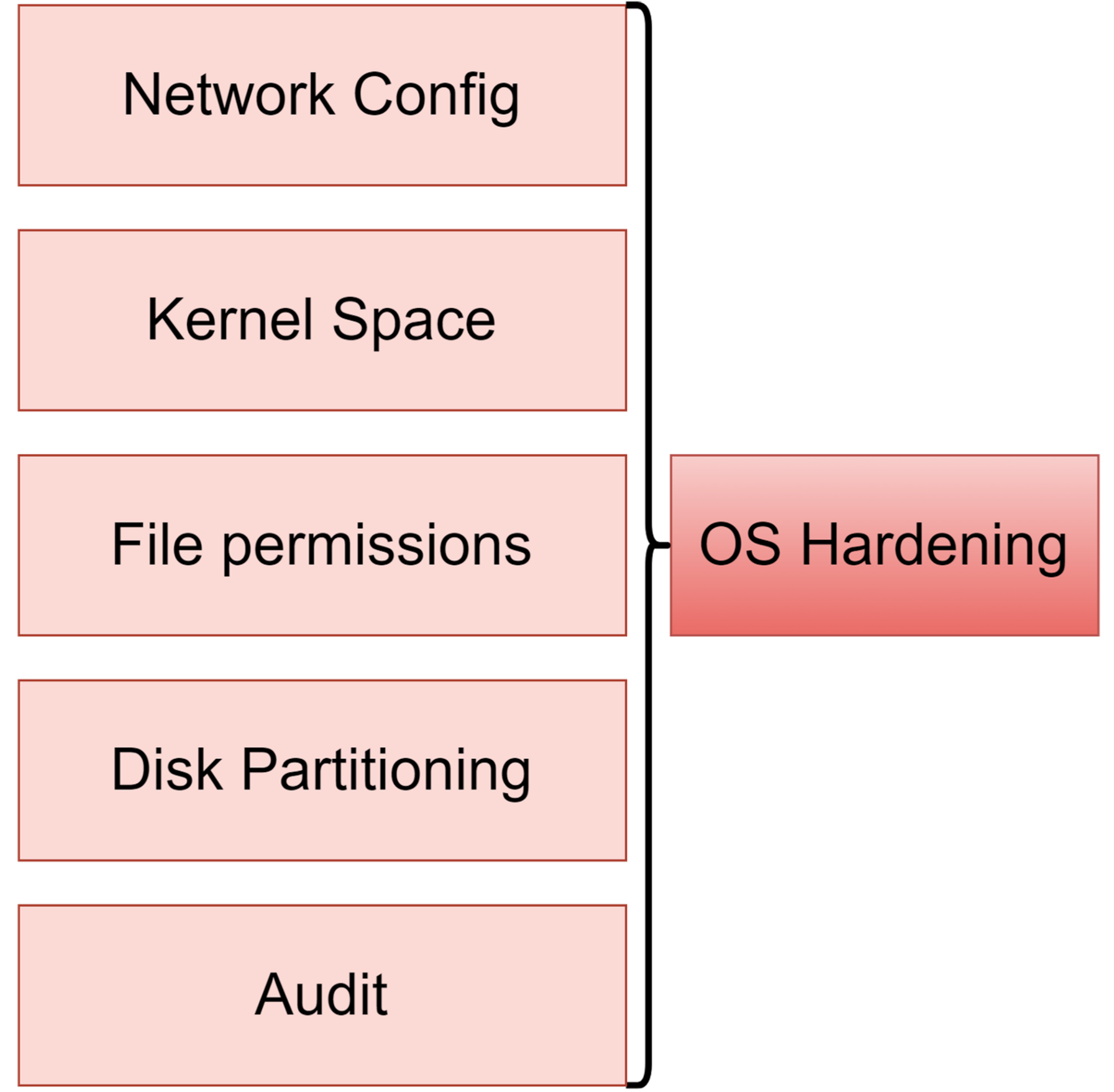
100%

110%

Secure by default

Hardening

01 Операционной системы



Secure by default

Hardening

01 Операционной системы



OpenSCAP
BASE

<https://github.com/OpenSCAP/openscap>

Secure by default

Hardening

01 Операционной системы

02 Контейнеров

Container Runtime

Permissions

Trusted images

Audit

Container
Hardening

```
graph LR; CR[Container Runtime]; P[Permissions]; TI[Trusted images]; A[Audit]; CH[Container Hardening]; CR --- CH; P --- CH; TI --- CH; A --- CH;
```

Secure by default

Hardening

01 Операционной системы

02 Контейнеров



<https://github.com/aquasecurity/trivy>

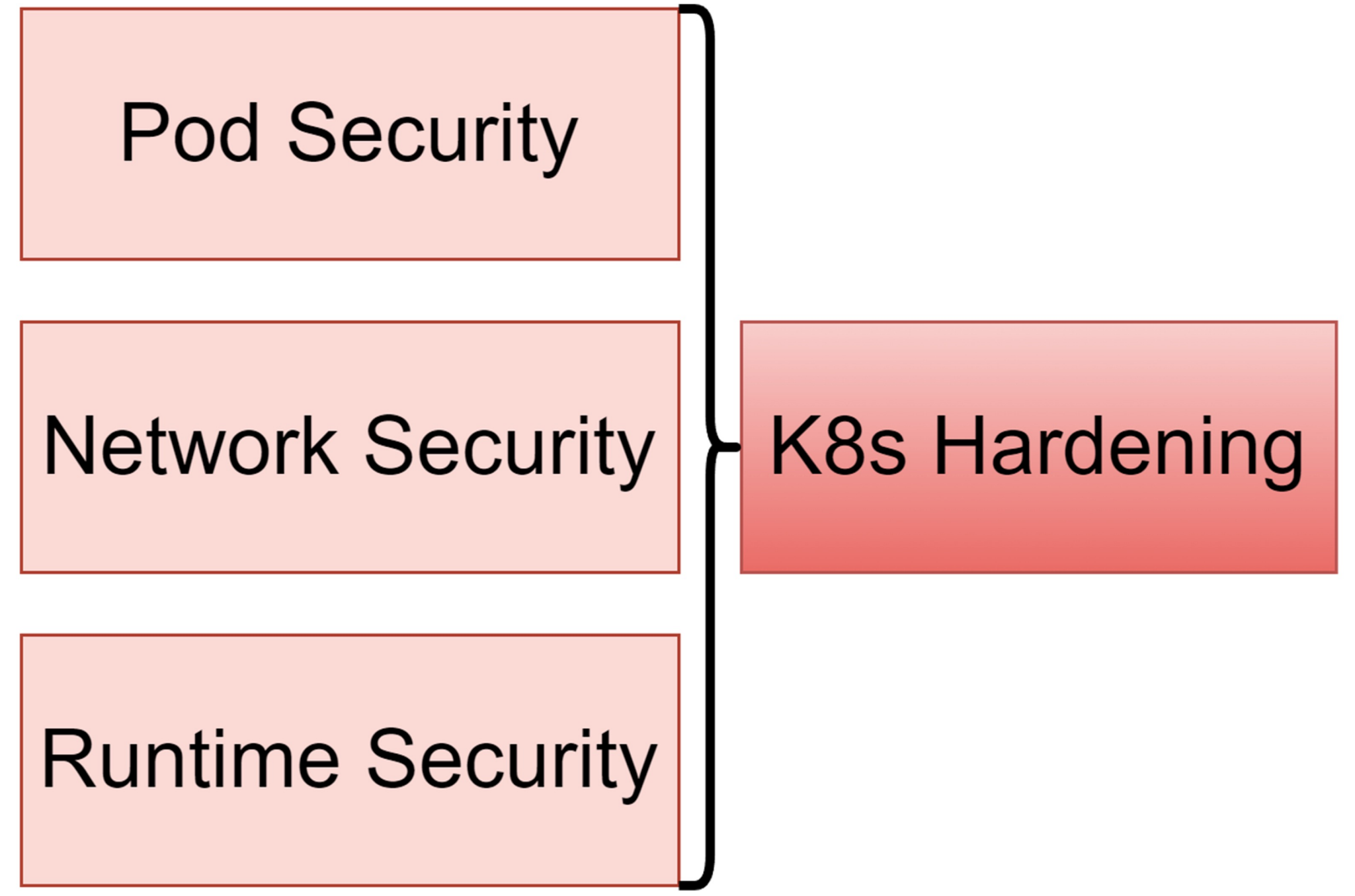
Secure by default

Hardening

01 Операционной системы

02 Контейнеров

03 Кластеров k8s



Secure by default

Hardening

01 Операционной системы

02 Контейнеров

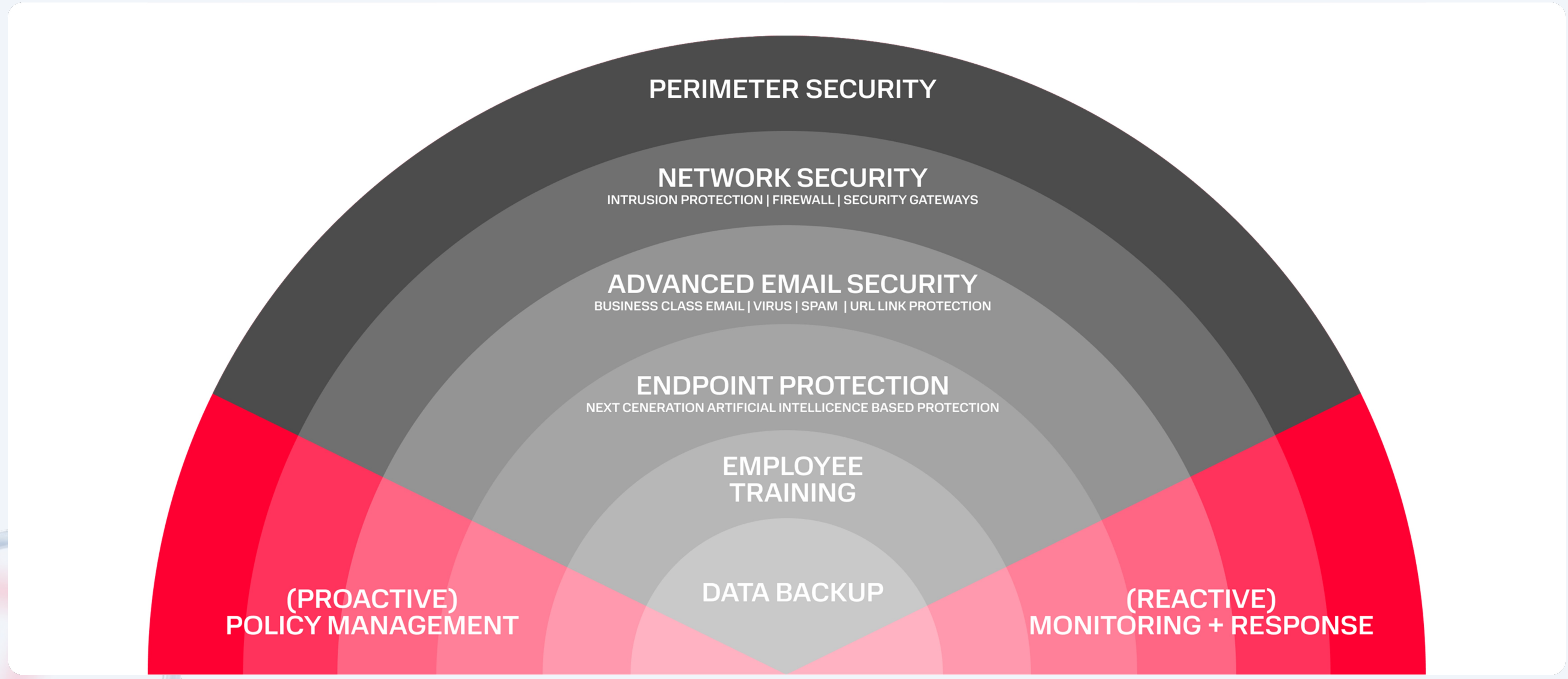
03 Кластеров k8s



<https://github.com/kubescape/kubescape>

Defense in Depth

Эшелонированная защита



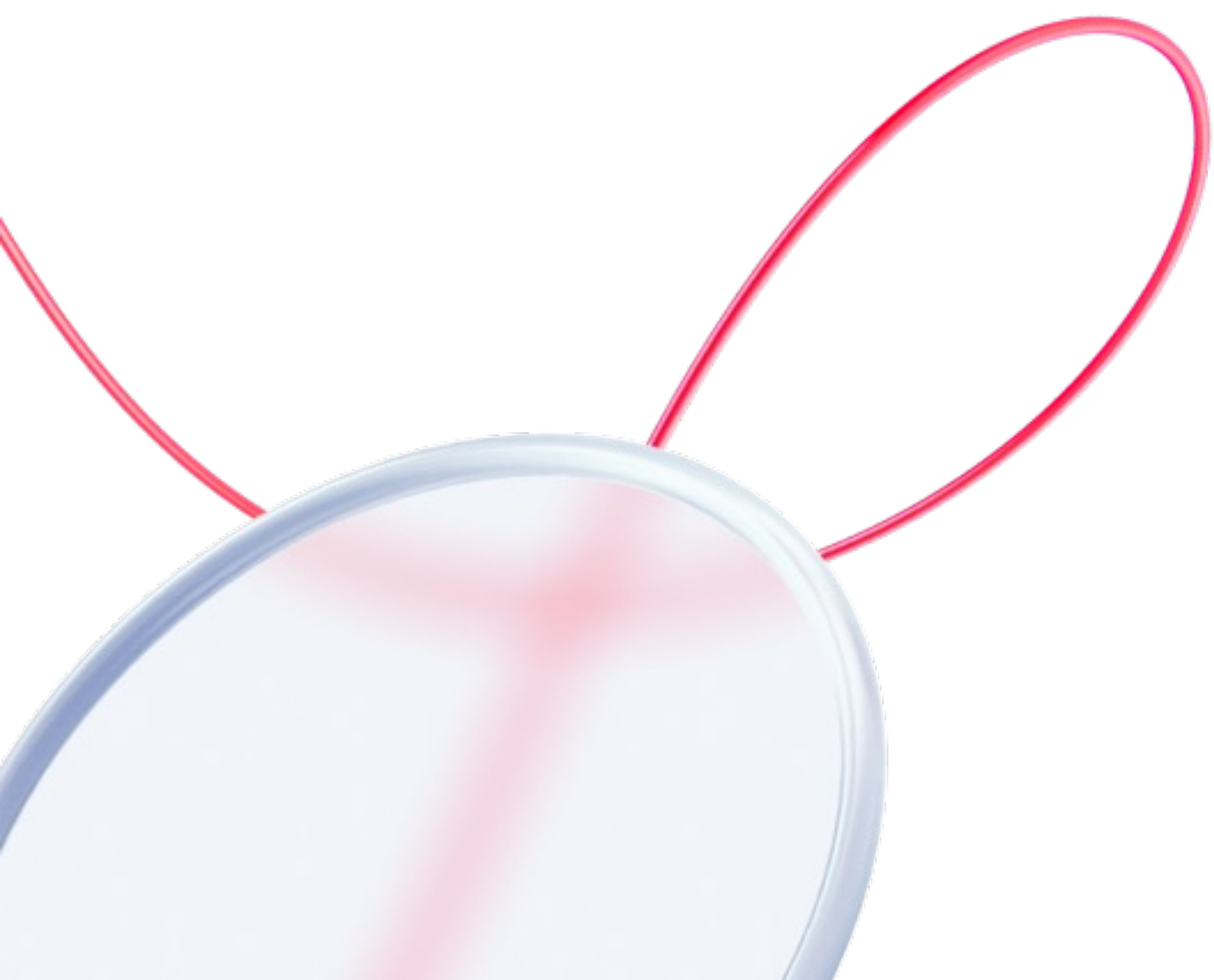
S И остальные...

01 Fail Safe

03 Secure by Design

02 Least Privilege

04 Economy of Mechanism



M W

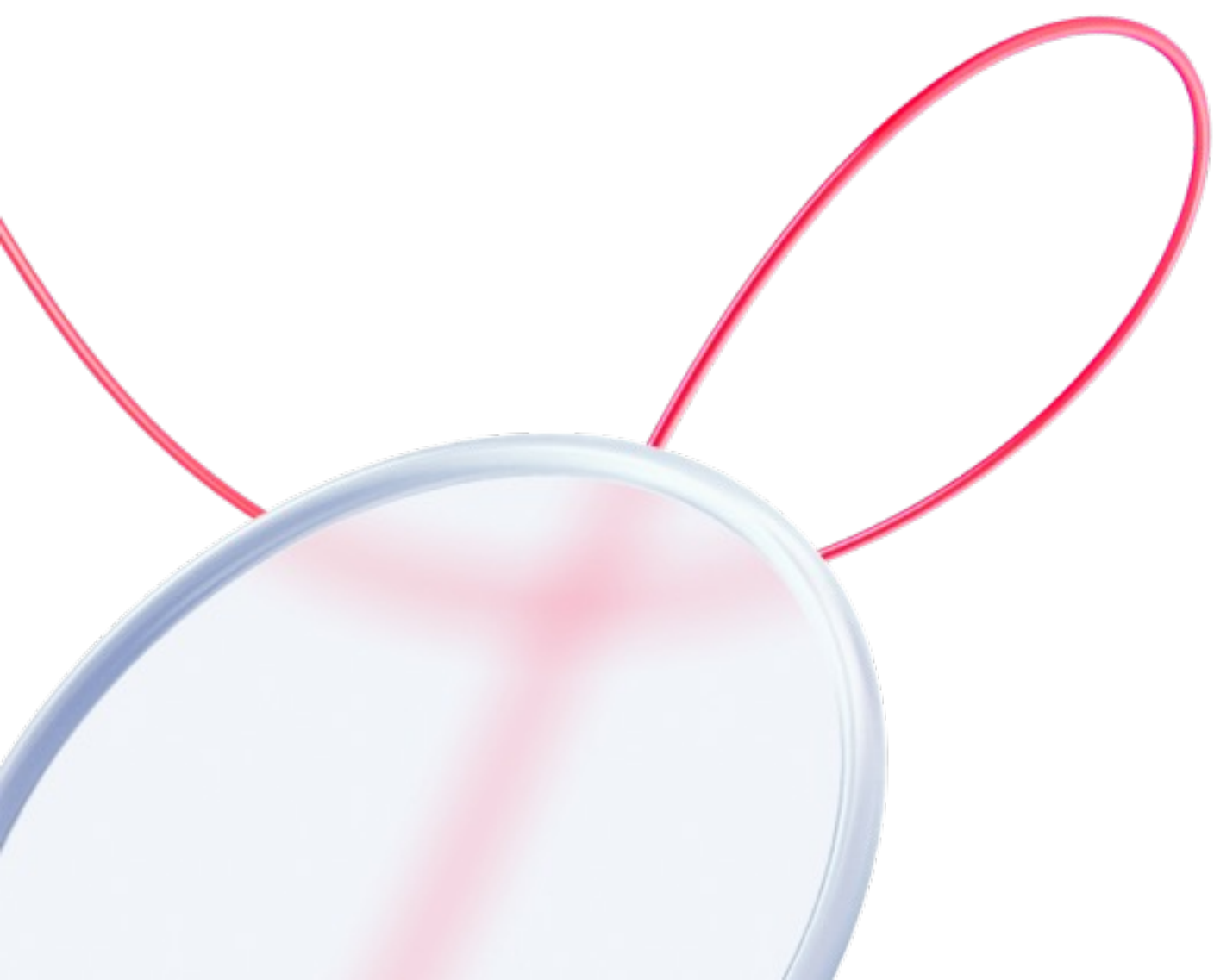
S И остальные...

01 Fail Safe

03 Secure by Design

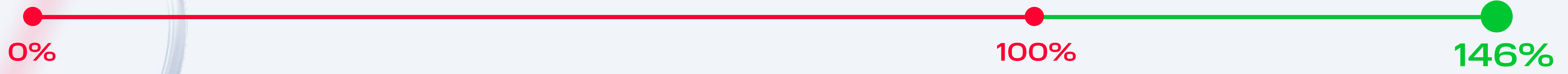
02 Least Privilege

04 Economy of Mechanism



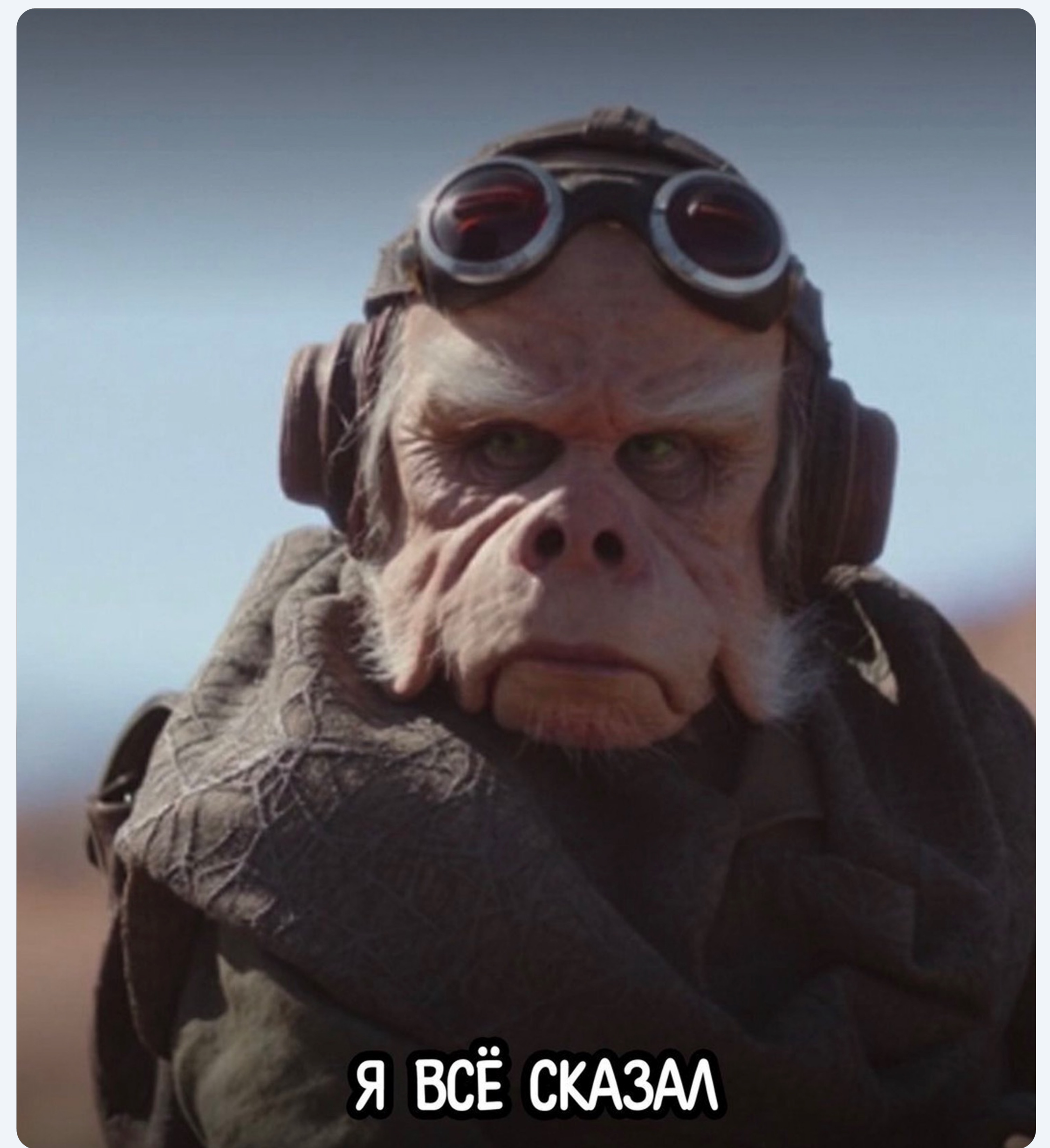
Чего мы добились

- 01 Внедрили SAST
- 02 Внедрили Gitlab security templates
- 03 Собрали свой security template
- 04 Поняли что пора нанять DevSecOps



M W
S

ТЕПЕРЬ
ВСЁ!



Я ВСЁ СКАЗАЛ