

Готовим контейнеры полезно и вкусно

@rusdacent

@aleksey0xffff

whoami (Анатолий Карпенко)

- Автоматизатор автоматизации в [Luntry](#)
- Любитель митапошных форматов: [SPb Reliability Meetup](#), [ITGM](#), [TechTrain](#), [DevOops](#), [DEFCON'ы](#), [B4CKSP4CE](#), [DevOps40](#)
- Веду канал «[Технологический Болт Генона](#)»
- Рисую несмешные мемы



whoami (Алексей Федулаев)

- Руководитель направления Cloud Native Security в [МТС Web Services](#)
- В ИБ с 2011 года
- Специалист по безопасности
- Автор канала [«Ever Secure»](#)



С чего начинается готовка?

**С чего начинается
готовка?**

С выбора рецепта

**СОХРАНИ СЕБЕ, ЧТОБЫ
НЕ ПОТЕРЯТЬ РЕЦЕПТ**



В нашем случае Dockerfile

С чего начинается готовка?

Без какого ингридиента невозможен Dockerfile?

С чего начинается готовка?

FROM reci:pe

Base image

Откуда берём?

Base image

Откуда берём?

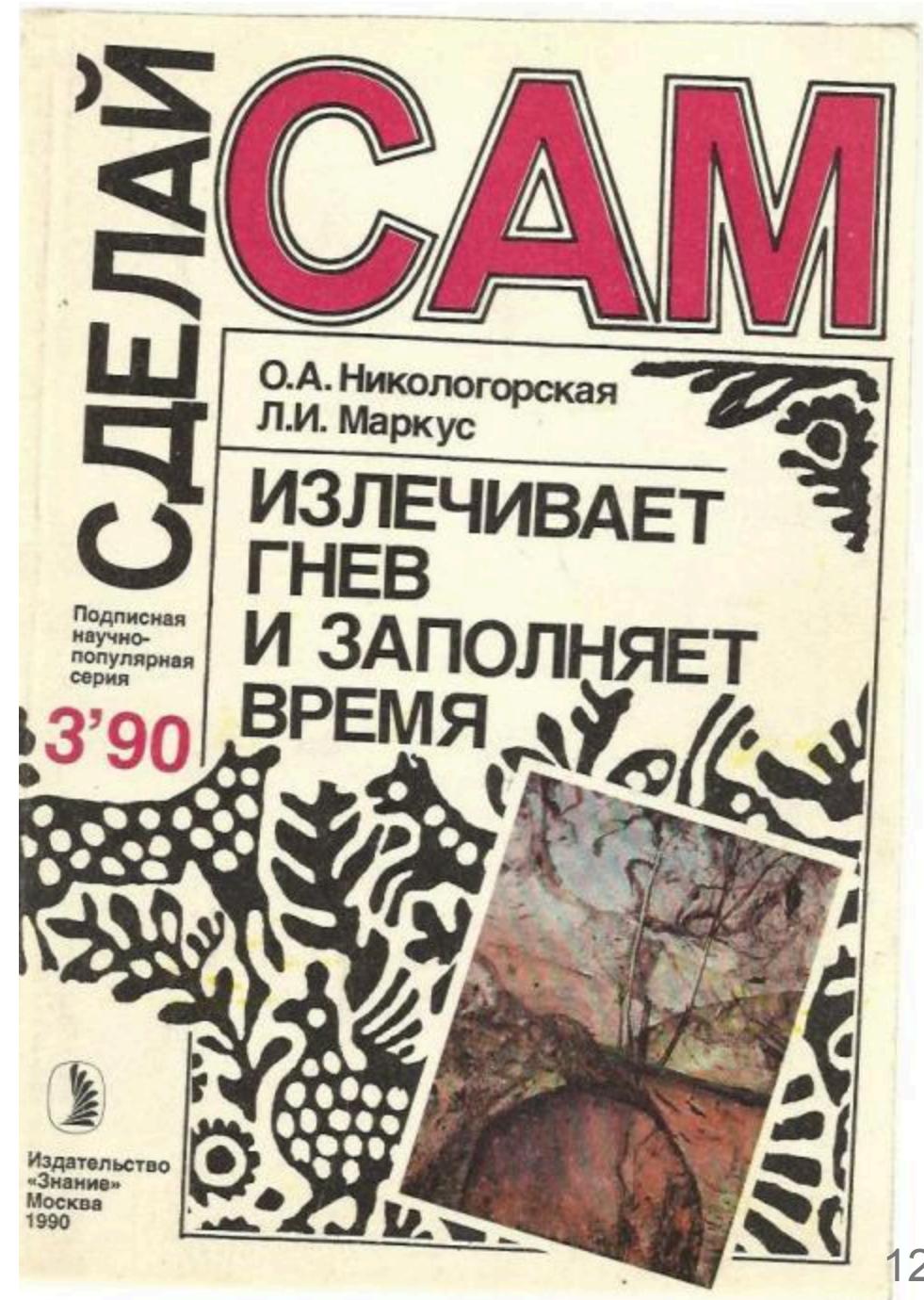
- DockerHub, ghcr.io
(GitHub), quay.io
(RedHat),...



Base image

Откуда берём?

- DockerHub, ghcr.io (GitHub), quay.io (RedHat),...
- Собираем сами



DockerHub

Каким доверяем?

Trusted Content



Docker Official Image



Verified Publisher



Sponsored OSS

DockerHub

Official

Docker Official Images are a curated set of Docker open source and "drop-in" solution repositories.

1 - 25 of 177 available results.

Docker Official Image ×

177



alpine

Updated 20 days ago

A minimal Docker image based on Alpine Linux with a complete package index.
5 MB in size!

Linux IBM Z riscv64 x86-64 ARM ARM 64 386 PowerPC 64 LE



nginx

Updated 6 hours ago

Official build of Nginx.

Linux unknown 386 mips64le PowerPC 64 LE IBM Z unknown x86-64 ARM
ARM 64



busybox

Updated a month ago

Busybox base image.

Linux unknown ARM ARM 64 mips64le unknown PowerPC 64 LE riscv64 |
x86-64 386

DockerHub

Verified Publisher

High-quality images from publishers verified by Docker. These products are published and maintained directly by a commercial entity. These images are not subject to rate limiting.

1 - 25 of 8 485 available results.

Verified Publisher ×

8485



grafana/grafana

±1B+ ·

By Grafana Labs · Updated 5 days ago

The official Grafana docker container

Linux arm64 x86-64 arm



bitnami/postgresql

±1B+ ·

By VMware · Updated 7 days ago

Bitnami PostgreSQL Docker Image

Linux x86-64 arm64



bitnami/kubectl

±1B+ ·

By VMware · Updated 5 days ago

Bitnami Docker Image for Kubectl

Linux arm64 x86-64

DockerHub

Sponsored OSS

Docker-Sponsored Open Source Software. These are images published and maintained by open-source projects that are sponsored by Docker through our open source program.

1 - 25 of 10 000 available images.
Sponsored OSS × images ×

Over 9000

 fluent/fluent-bit ⓘ 1B+

By Fluent organization: Fluentd project • Updated a month ago
Fluent Bit, lightweight logs and metrics collector and forwarder
unknown Linux Windows arm64 IBM Z unknown x86-64 arm

 istio/proxyv2 ⓘ 1B+

By istio • Updated 6 days ago
Istio proxy
Linux x86-64 arm64

 istio/pilot ⓘ 1B+

By istio • Updated 6 days ago
Istiod (formerly known as Pilot)
Linux x86-64 arm64

DockerHub

Other

to the 150 million images
created by the Docker
community ([2020](#))

1 - 25 of 10 000 available images.
[images](#) ×

Over Over 9000



[lachlantate/memcached-operator](#)

By [lachlantate](#) • Updated a few seconds ago

Linux arm64



[manuseligmann/sdypp-final-video-assembler-worker](#)

By [manuseligmann](#) • Updated a few seconds ago

Linux x86-64



[belasoft/senoide-backend](#)

By [belasoft](#) • Updated 2 minutes ago

Linux x86-64



[sarahepearce76/netrc](#)

By [sarahepearce76](#) • Updated 2 minutes ago

seadas sarah's attempts

DockerHub (Ищем образ)

Ищем на

<https://hub.docker.com/>

образ kafka



DockerHub (Ищем образ)

Ищем на

<https://hub.docker.com/>

образ kafka

Какой выберем?
Почему?

**КАКАЯ ТЫ КАФКА
СЕГОДНЯ?**



С чего начинается готовка?

С муки выбора

Base image

Кто знает что у него в базовом образе?

Base image

Кто собирал
базовый образ
сам?

Какие?

Зачем?



ИЛОН МАСК ВСПОТЕЛ,
КОГДА УЗНАЛ ПРО
ЭТО БИОТОПЛИВО
РЕЦЕПТ ПРИМЕРНО
УСРЕДНЕННЫЙ...

Base image (Как?)

- Из файловой системы
- Из Scratch
- Из iso

```
root@docker:~# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
root@docker:~# echo $(openssl rand -base64 32) > myimage/secret.txt
root@docker:~# tar -C myimage/ -c . -f myimage.tar
root@docker:~# docker import myimage.tar myimage:latest
sha256:64d777b0254f87ba5277b2e9c6e1f864c05b1740480b75c2d8d0e70fdfc3173f
root@docker:~# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
myimage        latest    64d777b0254f    6 seconds ago   1.07GB
root@docker:~#
```

Base image (Ссылки)

Ссылки

- [Base images \(via Docker\)](#)
- [Moby GitHub repo](#)
- [Creating your Own Base Image for Docker](#)

Конкретный пример

```
$ docker run -it --rm redos-7.3:latest cat /etc/redos-release
```

```
RED OS release MUROM (7.3.2)
```

SAST для Dockerfile



SAST для Dockerfile

Кто сканит Dockerfile?

Что используется?

Hadolint

- Опирается на [рекомендации Docker](#)

Hadolint

- Опирается на [рекомендации Docker](#)
- Использует ShellCheck для скриптов в RUN

Hadolint

- Опирается на [рекомендации Docker](#)
- Использует ShellCheck для скриптов в `RUN`
- OpenSource

Hadolint (Плюсы)

- Опирается на [рекомендации](#) Docker
- Использует ShellCheck для скриптов в `RUN`
- OpenSource
- Расширяемость



Hadolint (Тоже плюс)

- Haskell



Hadolint (DL3003)

```
module Hadolint.Rule.DL3003 (rule) where

import Hadolint.Rule
import Hadolint.Shell (ParsedShell, usingProgram)
import Language.Docker.Syntax (Instruction(..), RunArgs(..))

rule :: Rule ParsedShell
rule = simpleRule code severity message check
where
  code = "DL3003"
  severity = DLWarningC
  message = "Use WORKDIR to switch to a directory"
  check (Run (RunArgs args _)) = foldArguments (not . usingProgram "cd") args
  check _ = True
{-# INLINEABLE rule #-}
```

Hadolint (DL3049)

```
-- (67-85)

markGood :: Acc -> Acc
markGood Empty = Empty
markGood (Acc stageid good silent bad) =
    Acc stageid (good |> Set.insert stageid) (silent |> Set.delete stageid) (bad |> Set.delete stageid)

markSilentByAlias :: Text.Text -> Acc -> Acc
markSilentByAlias _ Empty = Empty
markSilentByAlias silentname (Acc stageid good silent bad) =
    Acc stageid good (silent |> Set.union stages) (bad |> remove stages)
where
    stages = Set.filter byName bad
    byName (StageID BaseImage {alias = Nothing} _) = False
    byName (StageID BaseImage {alias = Just als} _) = unImageAlias als == silentname
    remove set fromThis = Set.difference fromThis set

getCurrentStageName :: State Acc -> Text.Text
getCurrentStageName (State _ (Acc (StageID BaseImage {image, alias = Nothing} _) _ _ _)) = imageName image
getCurrentStageName (State _ (Acc (StageID BaseImage {alias = Just als} _) _ _ _)) = unImageAlias als
getCurrentStageName _ = ""
```

DL3013

Vlastimil Zeman edited this page on Nov 1, 2017 · 3 revisions

Pin versions in pip.

Problematic code:

```
FROM python:3.4
RUN pip install django
RUN pip install https://github.com/Banno/carbon/tarball/0.9.x-fix-events-callback
```



Correct code:

```
FROM python:3.4
RUN pip install django==1.9
RUN pip install git+https://github.com/Banno/carbon@0.9.15
```



Hadolint (Ищем проблемы)

Demo

Hadolint (Обсуждаем проблемы - DL3020)

| Use COPY instead of ADD for files and folders

```
ADD demo /home/myapp/demo
```

| ADD instruction introduces risks such as adding malicious files from URLs without scanning and unpacking procedure vulnerabilities.

Bonus

```
ADD --checksum=sha256:24454f830cdb571e2c4ad15481119c43b3cafd48dd869a9b2945d1036d1dc68d \
https://mirrors.edge.kernel.org/pub/linux/kernel/Historic/linux-0.01.tar.gz /
```

Hadolint (online)

```
1 FROM scratch as base
<string>:2:18:
|
2 | COPY --chmod=777 hadolint /bin/hadolint
| ^
invalid flag: --chmod
3
4 COPY --chmod=777 hadolint /bin/hadolint
5
6 LABEL org.opencontainers.image.source="https://github.com/hado
7 CMD ["/bin/hadolint", "-"]
8
9
```

Hadolint (COPY)

COPY

COPY has two forms:

```
COPY [ --chown=<user>:<group> ] [ --chmod=<perms> ] <src>... <dest>
COPY [ --chown=<user>:<group> ] [ --chmod=<perms> ] [ "<src>", ... "<dest>" ]
```

This latter form is required for paths containing whitespace

Semgrep

Semgrep

Кто использует Semgrep?

Semgrep (ruleset)

dockerfile.security.last-user-is-root.last-user-is-root

 + **5**

The last user in the container is 'root'. This is a security hazard because if an attacker gains control of the...



error

by semgrep 

dockerfile.security.missing-user-entrypoint.missing-user-entrypoint

 +

By not specifying a USER, a program in the container may run as 'root'. This is a security hazard. If an attack...



error

by semgrep 

dockerfile.security.missing-user.missing-user

 +

By not specifying a USER, a program in the container may run as 'root'. This is a security hazard. If an attack...



error

by semgrep 

Semgrep (GitHub rules)

```
$ tree -P '*.yaml' dockerfile/
dockerfile/
├── audit
│   └── dockerfile-source-not-pinned.yaml
├── best-practice
│   └── avoid-apk-upgrade.yaml
├── correctness
│   ├── invalid-port.yaml
│   ├── missing-assume-yes-switch.yaml
│   └── multiple-entrypoint-instructions.yaml
└── security
    ├── dockerd-socket-mount.yaml
    ├── last-user-is-root.yaml
    ├── missing-user-entrypoint.yaml
    ├── missing-user.yaml
    ├── no-sudo-in-dockerfile.yaml
    └── secret-in-build-arg.yaml
```

4 directories, 36 files

Semgrep (playground)

missing-user

structure [NEW](#) advanced

test code live code [NEW](#) metadata docs

Pro Turbo

```
1 rules:
2   - id: missing-user
3     patterns:
4       - pattern: |
5         | CMD $...VARS
6       - pattern-not-inside: |
7         | USER $USER
8       ...
9     fix: |
10    | USER non-root
11    | CMD $...VARS
12  message: By not specifying a USER, a program in the container may run as
13  | 'root'.
14  | This is a security hazard. If an attacker can control a process running as
15  | root, they may have control over the container. Ensure that the last USER
16  | in a Dockerfile is a USER other than 'root'.
17  severity: ERROR
18  languages:
19    - dockerfile
20  metadata:
21    ...

1 FROM busybox
2
3 # uncomment for ok
4 #USER notroot
5
6 RUN git clone https://github.com/returntocorp/semgrep
7 RUN pip3 install semgrep
8
9 # ruleid: missing-user
10 CMD semgrep -f p/xss
11
12 # ruleid: missing-user
13 CMD semgrep --config localfile targets
14
15 # TODO: metavar ellipses bug
16 # ok: missing-user
17 CMD ["semgrep", "--version"]
18
```

Semgrep (Ищем проблемы)

Demo

Checkov

Checkov

Scans Terraform, Terraform Plan, Terraform JSON, CloudFormation, AWS SAM, Kubernetes, Helm, Kustomize, Dockerfile, Serverless framework, Ansible, Bicep, ARM, and OpenTofu template files.

Scans Argo Workflows, Azure Pipelines, BitBucket Pipelines, Circle CI Pipelines, GitHub Actions and GitLab CI workflow files

Checkov (example)

```
FROM node:alpine
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000 22
HEALTHCHECK CMD curl --fail http://localhost:3000 || exit 1
CMD ["node", "app.js"]
```

```
$ checkov -d . --framework dockerfile
```

Checkov (Ищем проблемы)

Demo

Checkov (Rules, Python)

```
12  v  class RootUser(BaseDockerfileCheck):
13  v      def __init__(self) -> None:
14          name = "Ensure the last USER is not root"
15          id = "CKV_DOCKER_8"
16          supported_instructions = ("USER",)
17          categories = (CheckCategories.APPLICATION_SECURITY,)
18          super().__init__(name=name, id=id, categories=categories, supported_instructions=supported_instructions)
19
20  v      def scan_resource_conf(self, conf: list[_Instruction]) -> tuple[CheckResult, list[_Instruction] | None]:
21          last_user = conf[-1]
22          if last_user["value"] == "root":
23              return CheckResult.FAILED, [last_user]
24
25          return CheckResult.PASSED, [last_user]
26
```

Checkov (Rules, YAML easy)

```
metadata:
  id: "CKV2_DOCKER_2"
  name: "Ensure that certificate validation isn't disabled with curl"
  category: "APPLICATION_SECURITY"
definition:
  cond_type: attribute
  resource_types:
    - RUN
  attribute: value
  operator: not_regex_match
  value: ".*\n  curl[^\\|&]*\\s+((--insecure)|(-[^-\\s]*k))).*"
```

Checkov (Rules, YAML hard)

```
metadata:
  id: "CKV2_DOCKER_12"
  name: "Ensure that certificate validation isn't disabled for npm via the 'NPM_CONFIG_STRICT_SSL' environment variable"
  category: "APPLICATION_SECURITY"
definition:
  or:
    - cond_type: attribute
      resource_types:
        - ARG
        - ENV
      attribute: value
      operator: not_regex_match
      value: "^(.*\\s+)?(((NPM_CONFIG_STRICT_SSL)|(npm_config_strict_ssl))(=|\\s+)((false)|('false')|(\"false\"))).*"
    - cond_type: attribute
      resource_types:
        - RUN
      attribute: value
      operator: not_regex_match
      value: "^(.*[\\s;&|]+)?(((NPM_CONFIG_STRICT_SSL)|(npm_config_strict_ssl))=((false)|('false')|(\"false\"))).*"
```

SAST для Dockerfile

Что ещё может быть?

Что бы вы хотели ловить в своих Dockerfile?

SAST для Dockerfile

Всё что угодно

Leaky Vessels (Описание)

- CVE-2024-21626: уязвимость типа order-of-operations в команде WORKDIR в runc. Позволяет получить несанкционированный доступ к операционной системе хоста и потенциально скомпрометировать всю систему.
- CVE-2024-23651: состояние гонки в Buildkit, приводящее к непредсказуемому поведению
- CVE-2024-23652: проблема, позволяющая удалять произвольные файлы или каталоги на этапе удаления контейнера в Buildkit
- CVE-2024-23653: уязвимость возникает из-за недостаточной проверки привилегий в интерфейсе GRPC Buildkit

Leaky Vessels (Пример)

```
$ runc --version
runc version 1.1.7-0ubuntu1~22.04.2
...
cat /proc/11/cwd/../../../../etc/passwd
root:x:0:0:root:/root:/bin/bash
messagebus:x:498:498:User for D-Bus:/run/dbus:/usr/sbin/nologin
lp:x:497:497:Printing daemon:/var/spool/lpd:/usr/sbin/nologin
systemd-timesync:x:484:484:systemd Time
...
```

Leaky Vessels (Правила)

```
common.Rule{
    ID: 1,
    CVE: "CVE-2024-21626",
    Name: "runc process.cwd & Leaked fds Container Breakout",
    Blogpost: "https://snyk.io/blog/cve-2024-21626-runc-process-cwd-container-breakout",
    Inst:      "WORKDIR",
    ArgRegex:  regexp2.MustCompile("/proc/self/fd/"),
},
common.Rule{
    ID: 2,
    CVE: "CVE-2024-23651",
    Name: "Buildkit Mount Cache Race: Build-time Race Condition Container Breakout",
    Blogpost: "https://snyk.io/blog/cve-2024-23651-docker-buildkit-mount-cache-race",
    Inst:      "RUN",
    ArgRegex:  regexp2.MustCompile("--mount=type=cache"),
},
common.Rule{
    ...
}
```

Leaky Vessels (repo)

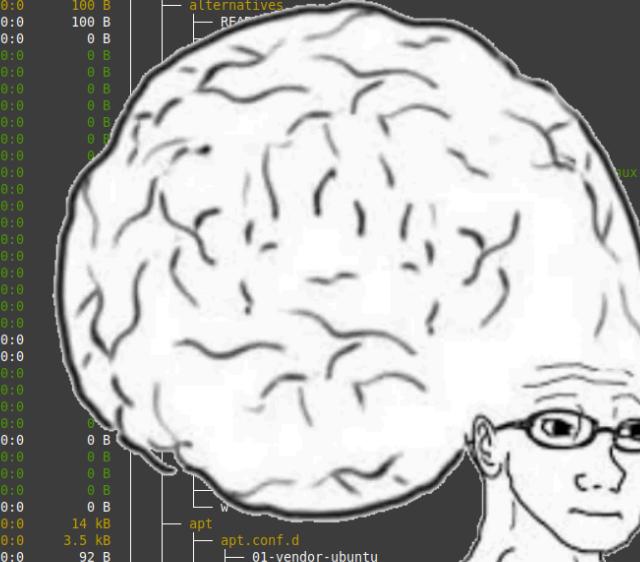
Умеет не только Dockerfile, но и образы

- Regex based detection rule matching.
- Dockerfile detection
- Image detection

Что на счёт
образов?



Что на счёт образов?



The image shows a terminal window with the following content:

```
| Layers |  
| Cmp Size Command  
73 MB FROM blobs  
1.2 MB RUN ||1 TARGETARCH=amd64 /bin/sh -c apt-get update && apt-get install -y --no-install-recommends tini && rm  
1.4 MB COPY ./dist/amd64/ttyd /usr/bin/ttyd # buildkit  
0 B WORKDIR /root  
0 B WORKDIR /var/run  
506 MB RUN /bin/sh -c apt-get update && DEBIAN_FRONTEND=noninteractive apt-get install -y curl iputils-ping nmap stow python  
4.7 kB COPY install.sh . # buildkit  
560 MB RUN /bin/sh -c chmod +x install.sh && ./install.sh && rm -f install.sh # buildkit  
  
| Layer Details |  
Tags: (unavailable)  
Id: blobs  
Digest: sha256:0074d82aa561e1486f5fc8d1625da94ecd10405bf6fd67c8502ac6d6a9ad1c6d  
Command:  
RUN /bin/sh -c apt-get update && DEBIAN_FRONTEND=noninteractive apt-get install -y curl iputils-ping nmap stow python  
git-core sudo util-linux p7zip-full jq ssh python  
upx && rm -rf /var/lib/apt/lists/* # buildkit  
  
| Image Details |  
Image name: mtkpi  
Total Image size: 1.1 GB  
Potential wasted space: 6.4 MB  
Image efficiency score: 99 %  
Count Total Space Path  
2 1.4 MB /usr/lib/x86_64-linux-gnu/libsystemd.so.0.28.0  
2 1.3 MB /var/cache/debconf/templates.dat  
4 933 kB /var/log/dpkg.log  
4 812 kB /var/lib/dpkg/status  
3 724 kB /var/lib/dpkg/status-old  
2 544 kB /usr/bin/wget  
2 240 kB /usr/bin/curl  
3 67 kB /var/log/apt/history.log  
2 66 kB /var/log/apt/term.log  
  
| Current Layer Contents |  
Permission UID:GID Size Filetree  
-rwxr-xrwx 0:0 0 B bin -> usr/bin  
drwxr-xr-x 0:0 0 B boot  
drwxr-xr-x 0:0 0 B dev  
drwxr-xr-x 0:0 1.0 MB etc  
-rw----- 0:0 0 B .pwd.lock  
drwxr-xr-x 0:0 880 B X11  
-rwxr--r-- 0:0 880 B Xsession.d  
-rwxr--r-- 0:0 3.0 kB 00pgp-agent  
drwxr-xr-x 0:0 100 B adduser.conf  
drwxr-xr-x 0:0 100 B alternatives  
drwxr-xr-x 0:0 0 B apt  
drwxr-xr-x 0:0 0 B apt.conf.d  
-rwxr-xr-x 0:0 0 B 01-vendor-ubuntu  
-rwxr-xr-x 0:0 630 B 01autoremove  
-rwxr--r-- 0:0 2.2 kB 50apt-file.conf  
-rwxr--r-- 0:0 182 B 70debconf  
-rwxr--r-- 0:0 44 B docker-autoremove-suggests  
-rwxr--r-- 0:0 318 B docker-clean  
-rwxr--r-- 0:0 70 B docker-gzip-indexes  
-rwxr--r-- 0:0 27 B docker-no-languages  
drwxr-xr-x 0:0 0 B auth.conf.d  
drwxr-xr-x 0:0 0 B preferences.d
```

At the bottom of the terminal window, there are several command-line options: ^C Quit, Tab Switch view, ^F Filter, ^L Show layer changes, ^A Show aggregated changes.

Dive

Demo

Что на счёт образов?

Есть что!

Dockle

Проверяет образ на

- CIS Docker Benchmark <- Центр интернет-безопасности (CIS)
- best-practice for Dockerfile

Dockle

Demo

Dockle (CIS-DI-0010 - Files)

```
func (a CredentialAssessor) RequiredFiles() []string {
    return []string{
        "credentials.json",
        "credential.json",
        // TODO: Only check .docker/config.json
        // "config.json",
        "credentials",
        "credential",
        "password.txt",
        "id_rsa",
        "id_dsa",
        "id_ecdsa",
        "id_ed25519",
        "secret_token.rb",
        "carrierwave.rb",
        "omniauth.rb",
        "settings.py",
        "database.yml",
        "credentials.xml",
    }
}
```

Dockle (CIS-DI-0010 - Extensions)

```
func (a CredentialAssessor) RequiredExtensions() []string {
    return []string{
        // reference: https://github.com/eth0izzle/shhgit/blob/master/config.yaml
        // TODO: potential sensitive data but they have many false-positives.
        // Dockle need to analyze each file.
        //".pem",
        //".key",
        //".p12",
        //".pkcs12",
        //".pfx",
        //".asc",
        ".secret",
        ".ovpn",
        ".private_key",
        ".cscfg",
        ".rdp",
        ".mdf",
        ".sdf",
        ".bek",
        ".tpm",
        ".fve",
        ".jks",
        ".psafe3",
        ".agilekeychain",
        ".keychain",
        ".pcap",
        ".gnucache",
    }
}
```

Dockle (MTKPI)

MTKPI – Multi Tool Kubernetes Pentest Image

```
| $ dockle r0binak/mtkpi:v1.4
```

```
| FATAL - DKL-DI-0001: Avoid sudo command
```

```
* Avoid sudo in container : RUN /bin/sh -c apt-get update &&
DEBIAN_FRONTEND=noninteractive apt-get install -y curl iputils-ping nano
python3-pip dnsutils apt-file net-tools nmap stow git-core sudo util-linux p7zip-full
jq ssh python python3 upx && rm -rf /var/lib/apt/lists/* # buildkit
```

```
| ...
```

Dockle (MTKPI, sudo)

```
RUN apt-get update \
&& DEBIAN_FRONTEND=noninteractive \
apt-get install -y \
    curl \
    ...
    nmap \
    stow \
    git-core \
    sudo \
    ...
    python3 \
    upx \
&& rm -rf /var/lib/apt/lists/*
```

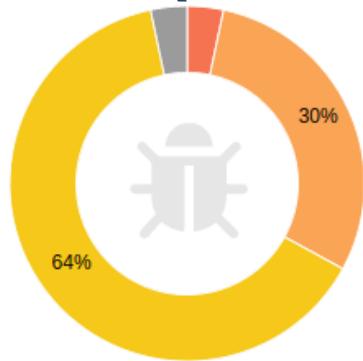
Clair

Vulnerability Static Analysis for
Containers

- RedHat
- Интеграция с [Quay](#)
- Использует отдельный пакет -
[Claircore](#)

- <https://secdb.alpinelinux.org/>
- http://repo.us-west-2.amazonaws.com/2018.03/updates/x86_64/mirror.list
- https://cdn.amazonlinux.com/2/core/latest/x86_64/mirror.list
- https://cdn.amazonlinux.com/al2023/core/mirrors/latest/x86_64/mirror.list
- <https://deb.debian.org/>
- <https://security-tracker.debian.org/tracker/data/json>
- <https://nvd.nist.gov/feeds/json/cve/1.1/>
- https://linux.oracle.com/security/oval/com.oracle.elsa-*.xml.bz2
- https://packages.vmware.com/photon/photon_oval_definitions/
- <https://access.redhat.com/security/data/metrics/cvemap.xml>
- <https://access.redhat.com/security/cve/>
- https://access.redhat.com/security/data/oval/v2/PULP_MANIFEST
- <https://support.novell.com/security/oval/>
- <https://api.launchpad.net/1.0/>
- https://security-metadata.canonical.com/oval/com.ubuntu.*.cve.oval.xml
- <https://osv-vulnerabilities.storage.googleapis.com/>

Clair (demo)



Quay Security Scanner has detected **91** vulnerabilities.

Patches are available for **9** vulnerabilities.

- ⚠️ 3 High-level vulnerabilities.
- ⚠️ 27 Medium-level vulnerabilities.
- ⚠️ 58 Low-level vulnerabilities.
- ⚠️ 3 Unknown-level vulnerabilities.

Advisories

ADVISORY	SEVERITY ↓	PACKAGE	CURRENT VERSION	FIXED IN VERSION	INTRODUCED IN LAYER
▶ CVE-2024-48957 %	⚠️ High	libarchive	3.5.3-4.el9	(None)	<button>RUN</button> mv -fZ /tmp/ubi.repo /etc/yum.repos.d/ub...
▶ CVE-2024-48958 %	⚠️ High	libarchive	3.5.3-4.el9	(None)	<button>RUN</button> mv -fZ /tmp/ubi.repo /etc/yum.repos.d/ub...
▶ GHSA-78wr-2p64-hpjwj %	⚠️ High	commons-io:commons-io	2.11.0	2.14.0	COPY tmp/cc/ /opt/cruise-control/libs/ # buildkit
▶ CVE-2023-36632 %	7.5 / 10	python3	3.9.18-3.el9_...	(None)	RUN 4 JAVA_VERSION=17 TARGETOS=linux TARGETARCH=amd64 strir

Trivy

Targets

- Container Image
- Filesystem
- Git Repository (remote)
- Virtual Machine Image
- Kubernetes
- AWS

Scanners

- OS packages and software dependencies in use (SBOM)
- Known vulnerabilities (CVEs)
- IaC issues and misconfigurations
- Sensitive information and secrets
- Software licenses

Trivy

Demo

Trivy (Allow rules)

```
{  
    ID:          "tests",  
    Description: "Avoid test files and paths",  
    Path:        MustCompile(`(^test|\/test|-test|_test|\.\test)`),  
},  
{  
    ID:          "examples",  
    Description: "Avoid example files and paths",  
    Path:        MustCompile(`example`),  
    Regex:       MustCompile("(?i)example"),  
},
```

Grype

Grype + Syft ≈ Trivy

Syft - CLI tool
and library for
generating a
Software Bill of
Materials from
container images
and filesystems

- Alpine Linux SecDB: <https://secdb.alpinelinux.org/>
- Amazon Linux ALAS: <https://alas.aws.amazon.com/AL2/alas.rss>
- RedHat RHSA: <https://www.redhat.com/security/data/oval/>
- Debian Linux CVE Tracker: <https://security-tracker.debian.org/tracker/data/json>
- Github GHSAs: <https://github.com/advisories>
- National Vulnerability Database (NVD): <https://nvd.nist.gov/vuln/data-feeds>
- Oracle Linux OVAL: <https://linux.oracle.com/security/oval/>
- RedHat Linux Security Data: <https://access.redhat.com/hydra/rest/securitydata/>
- Suse Linux OVAL: <https://ftp.suse.com/pub/projects/security/oval/>
- Ubuntu Linux Security: <https://people.canonical.com/~ubuntu-security/>

Как это выглядит в проде (Luntry)

docker.io/istio/pilot 1

docker.io/istio/pilot:1.10.0 1

Subject	SBOM (1)	Vulnerability (1)
Type	Vulnerability Report	
Name	docker.io/istio-pilot-294ca55b	
Report		
Update	6.02.2023/22:30:26	
Registry	docker.io	
Repository	docker.io/istio/pilot	
Scanner Name	Grype	
Scanner Vendor	Anchore	
Scanner Version	0.37.0	

Top Riskiest Components

Name	CVEs	Fixable
1. linux-tools-4.15.0-143	264	181
2. linux-tools-4.15.0-143-generic	264	181
3. linux-tools-common	264	181
4. libexpat1	15	15
5. libc6	13	10

Summary

Critical	0
High	60
Medium	694
Low	234
Negligible	56
Unknown	0

Report Vulnerabilities(1044)

Vulnerability ID ▲	Severity	Resource	Installed Version	Fixed Versions	Links
CVE-2022-26966	medium	linux-tools-4.15.0-143	4.15.0-143.147	4.15.0-177.186	<ul style="list-style-type: none">http://people.ubuntu.com/~ubuntu-security/cve/CVE-2022-26966

Что дальше?

НАСКАНИРОВАЛСЯ И СПИТ

Status	Pipeline
 failed	<u>#319049365</u> error
 failed	<u>#317510274</u> error
 failed	<u>#317485818</u> error

Наводим порядок!



Registry

"Чужие"

- Docker Hub
- GitHub
- GitLab
- Quay
- Chainguard
- . . .

"Свои"

- [registry](#)
- GitLab
- Quay
- Harbor
- Artifactory / JFrog Container Registry
- Nexus
- . . .

Docker Hub (demo)

COMPRESSED SIZE 483.24 MB LAST PUSHED 2 days ago by dojanky TYPE Image VULNERABILITIES 2 9 16 17 0 MANIFEST DIGEST sha256:4075a87f...

Image hierarchy

- FROM ubuntu:21cf807586307a44687b6adffffa6b982f44a05968541c163169e277f199... !
- FROM eclipse-temurin:1485be6650018888f3e42b8b70d794dfb0e48f0eb5dddbdb6731... !
- ALL sonarqube:its-enterprise** !

Layers (27)

- 0 ARG RELEASE 0 B ✓
- 1 ARG LAUNCHPAD_BUILD_ARCH 0 B ✓
- 2 LABEL org.opencontainers.image.ref.name=ubuntu 0 B ✓
- 3 LABEL org.opencontainers.image.version=22.04 0 B ✓
- 4 ADD file:ebe009f86035c175ba244badd298a2582914415cf6278... 29.54 MB !
- 5 CMD ["bin/bash"] 0 B ✓
- 6 ENV JAVA_HOME=/opt/java/openjdk 0 B ✓
- 7 ENV PATH=/opt/java/openjdk/bin:/usr/local/sbin:/usr/local/bin... 0 B ✓
- 8 ENV LANG=en_US.UTF-8 LANGUAGE=en_US:en LC_ALL=en_US.... 0 B ✓
- 9 RUN /bin/sh -c set -eux; apt-get update; DEBIAN_FRONTEND=no... 12.89 MB !
- 10 ENV JAVA_VERSION=jdk-17.0.12+7 0 B ✓

Vulnerabilities (28)

Package	Vulnerabilities
org.postgresql/postgresql 42.5.1	2 0 0 0 0
GHSA-xfg6-62px-cxc2	CWE-1035 CWE-89 CWE-937 10 C

Duplicate Advisory

This advisory has been withdrawn because it is a duplicate of GHSA-24rp-q3w6-vc56. This link is maintained to preserve external references.

Original Description

pgjdbc, the PostgreSQL JDBC Driver, allows attacker to inject SQL if using PreferQueryMode=SIMPLE. Note this is not the default. In the default mode there is no vulnerability. A placeholder for a numeric value must be immediately preceded by a minus. There must be a second placeholder for a string value after the first placeholder; both must be on the same line. By constructing a matching string payload, the attacker can inject SQL to alter the query,bypassing the protections that parameterized queries bring against SQL Injection attacks. Versions before 42.7.2, 42.6.1, 42.5.5, 42.4.4, 42.3.9, and 42.2.8 are affected.

CVSS Score: 10

<https://hub.docker.com/layers/library/sonarqube/its-enterprise/images/sha256-4075a87f5512b14b67f8a480aa8132131f705b2b999001e9f28a9b3f9ec950b4?context=explore> 79/110

Docker Hub (Log4Shell)

TAG

latest



Log4Shell CVE not detected

Docker Hub (Docker Scout)

- Умеет в интеграцию с
 - Artifactory
 - Amazon Elastic Container Registry
 - Azure Container Registry
- Требуется платный аккаунт
- Утверждается, что доступен для **sponsored** и **verified** аккаунтов, но я не нашёл ни одного живого примера

Harbor (Trivy, Clair)

Vulnerabilities Build History

 SCAN

Vulnerability	Severity	CVSS3	Package	Current version	Fixed in version	Listed in CVE Allowlist
CVE-2023-39325	High	bitnami: 7.5 ghsa: 7.5 nvd: 7.5 redhat: 7.5	golang.org/x/net	v0.8.0	 0.17.0	No
<p>Description: A malicious HTTP/2 client which rapidly creates requests and immediately resets them can cause excessive server resource consumption. While the total number of requests is bounded by the http2.Server.MaxConcurrentStreams setting, resetting an in-progress request allows the attacker to create a new request while the existing one is still executing. With the fix applied, HTTP/2 servers now bound the number of simultaneously executing handler goroutines to the stream concurrency limit (MaxConcurrentStreams). New requests arriving when at the limit (which can only happen after the client has reset an existing, in-flight request) will be queued until a handler exits. If the request queue grows too large, the server will terminate the connection. This issue is also fixed in golang.org/x/net/http2 for users manually configuring HTTP/2. The default stream concurrency limit is 250 streams (requests) per HTTP/2 connection. This value may be adjusted using the golang.org/x/net/http2 package; see the Server.MaxConcurrentStreams setting and the ConfigureServer function.</p>						
> GHSA-m425-mq94-257g	High	ghsa: 7.5	google.golang.org/grpc	v1.50.0	 1.56.3, 1.57.1, 1.58.3	No
> CVE-2023-48795	Medium	ghsa: 5.9 nvd: 5.9 redhat: 5.9	golang.org/x/crypto	v0.7.0	 0.17.0	No

Chainguard

Цель проекта

МИНИМЗАЦИЯ ПОВЕРХНОСТИ
атаки.

Minimal, hardened
images with SBOMs
and signatures

The screenshot shows the Chainguard interface for an nginx image. The top navigation bar includes links for Versions, Overview, Provenance, SBOM, and Vulnerabilities. The 'Vulnerabilities' tab is active, indicated by a blue underline. Below the navigation is a search bar with the placeholder 'Search...'. Underneath the search bar are filter buttons for 'latest' and 'Severity' (with a dropdown arrow), and columns for 'Package' and 'Version'. A large warning icon (an exclamation mark inside a triangle) is centered on the page. Below the icon, the text 'No known vulnerabilities' is displayed. At the bottom, there is a note: 'Visit [security advisories](#) to view the status of known vulnerabilities in a Chainguard Image.'

Chainguard (Wolfi)

Собственный дистрибутив с собственной экосистемой сборки

The key features of Wolfi are:

- Provides a high-quality, build-time SBOM as standard for all packages
- Packages are designed to be granular and independent, to support minimal images
- Uses the proven and reliable APK package format
- Fully declarative and reproducible build system
- Designed to support glibc and musl

Chainguard (Grype)

name	date	low	med	high	crit	unk	tot
cgr.dev/chainguard/nginx:latest	2024-02-17 14:34:01	0	0	0	0	0	0
nginx:latest	2024-02-17 14:34:22	4	30	12	2	2	121
cgr.dev/chainguard/nginx:latest	2024-01-21 14:33:57	0	0	0	0	0	0
nginx:latest	2024-01-21 14:34:18	4	25	11	2	9	123

Chainguard (Example)

```
learning-labs-java on ↵ main [!] via 🖥 desktop-linux is 📦 v1.0.0 via 🚧 on ☁ adrian@chainguard.dev
took 2s
> docker images java-maven-cg
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
java-maven-cg   latest   55e17d645b92  2 hours ago  360MB

learning-labs-java on ↵ main [!] via 🖥 desktop-linux is 📦 v1.0.0 via 🚧 on ☁ adrian@chainguard.dev
> docker scout quickview java-maven-cg
  i New version 1.6.4 available (installed version is 1.5.1) at https://github.com/docker/scout-cli
  ✓ SBOM of image already cached, 117 packages indexed

Target | java-maven-cg:latest | 0C 0H 0M 0L
       digest 55e17d645b92

What's Next?
  Include policy results in your quickview by supplying an organization → docker s
maven-cg --org <organization>
```



Chainguard

Минусы

- latest
- Своя экосистема

Плюсы

- Открытая экосистема
 - Build image - [apko](#)
 - Build APK - [melange](#)
 - [Package hero](#)
- Есть описание сборки через Dockerfile
- [SLSA](#) - Supply chain Levels for Software Artifacts
- [cosign](#)

Cosign

- "Keyless signing" with the Sigstore public good Fulcio certificate authority and Rekor transparency log (default)
- Hardware and KMS signing
- Signing with a cosign generated encrypted private/public keypair
- Container Signing, Verification and Storage in an OCI registry.
- Bring-your-own PKI

Cosign (example with keys: prepare)

```
$ cosign generate-key-pair --output-key-prefix='cosign'
```

```
$ cosign sign --key cosign.key $IMAGE
```

WARNING: Image reference rusdacent/hello-container:keys uses a tag,
not a digest, to identify the image to sign.

This can lead you to sign a different image than the intended one.

Please use a digest (example.com/ubuntu@sha256:abc123...) rather than tag

Cosign (example with keys: sign)

```
$ IMAGEDIGEST=$(docker manifest inspect --verbose $IMAGE | jq -r '.Descriptor.digest')  
$ export IMAGE=$IMAGE@$IMAGEDIGEST  
$ echo $IMAGE  
rusdacent/hello-container:keys@sha256:db5...452
```

```
$ cosign sign --key cosign.key $IMAGE  
.  
tlog entry created with index: 136317182  
Pushing signature to: index.docker.io/rusdacent/hello-container
```

Cosign (example with keys: verify)

```
$ cosign verify --key cosign.pub $IMAGE
Verification for index.docker.io/rusdacent/hello-container@sha256:. . . --
. . .
[
  {
    "critical": {
      "identity": {
        "docker-reference": "index.docker.io/rusdacent/hello-container"
      },
      "image": {
        "docker-manifest-digest": "sha256:db5. . .452"
      },
      "type": "cosign container image signature"
    },
    "optional": {
      "Bundle": {
        "SignedEntryTimestamp": "MEYC. . .tv+4F+2GHeP7KX",
        "Payload": {
          "body": "eyJhc. . .X0=",
          "integratedTime": 1727916426,
          "logIndex": 136317182,
          "logID": "c0d2. . .801d"
        }
      }
    }
]
```

Cosign (DockerHub)

TAG

[sha256-db54a230a821b4be2da51acda2489e192220bd163e64ca8d47122de5d3084452.sig](#)

Last pushed 10 minutes ago by [rusdacent](#)

Digest	OS/ARCH
<u>a62a665468b4</u>	---

TAG

[latest](#)

Last pushed 30 minutes ago by [rusdacent](#)

Digest	OS/ARCH
<u>db54a230a821</u>	linux/amd64

Cosign (Harbor)

Projects < Repositories

library/mariadb

Info Images

SCAN

COPY DIGEST

+ ADD LABELS

X DELETE

<input type="checkbox"/>	Tag	Size	Pull Command	Vulnerability	Signed	Author
<input type="checkbox"/>	10.3-signed	109.21MB		Not Scanned		
<input type="checkbox"/>	10.3	109.21MB		Not Scanned		

Cosign + Trivy = attestation

```
$ trivy image --format cosign-vuln --output vuln.json $IMG
```

```
$ cosign attest --key /path/to/cosign.key --type vuln --predicate vuln.json $IMG
```

```
$ cosign verify-attestation --key /path/to/cosign.pub --type vuln $IMG
```

Attestation (DockerHub)

TAG

[sha256-db54a230a821b4be2da51acda2489e192220bd163e64ca8d47122de5d3084452.sig](#)

Last pushed 10 minutes ago by [rusdacent](#)

Digest

OS/ARCH

[a62a665468b4](#)

TAG

[latest](#)

Last pushed 30 minutes ago by [rusdacent](#)

Digest

OS/ARCH

[db54a230a821](#)

linux/amd64

TAG

● [sha256-db54a230a821b4be2da51acda2489e192220bd163e64ca8d47122de5d3084452.att](#)

Last pushed 6 minutes ago by [rusdacent](#)

Digest

OS/ARCH

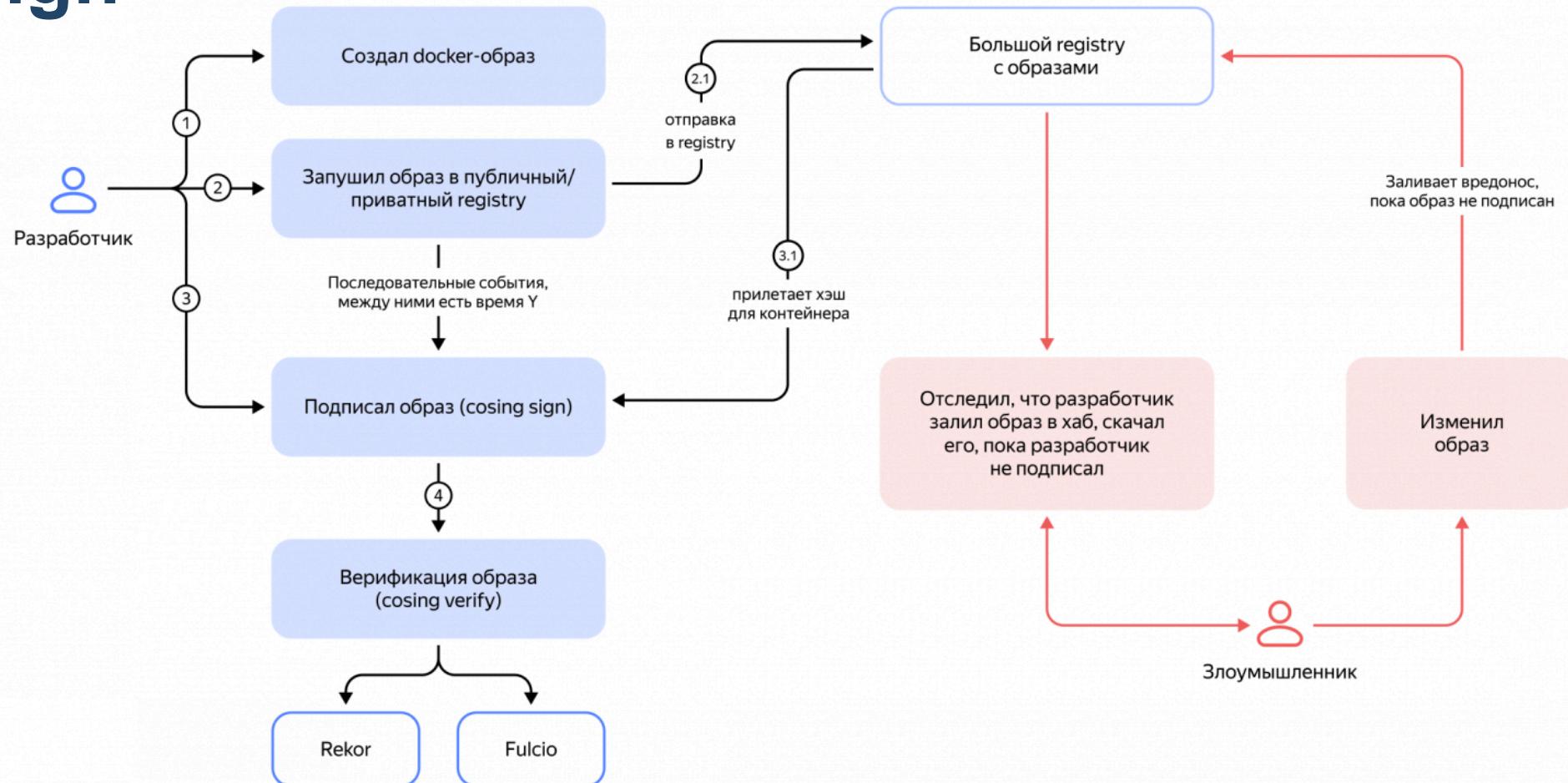
[54dc57011ef8](#)

DockerHub + cosign

Demo

<https://hub.docker.com/r/rusdacent/hello-container/tags>

Cosign



НА ЛЮБОЙ ВКУС

А ещё?

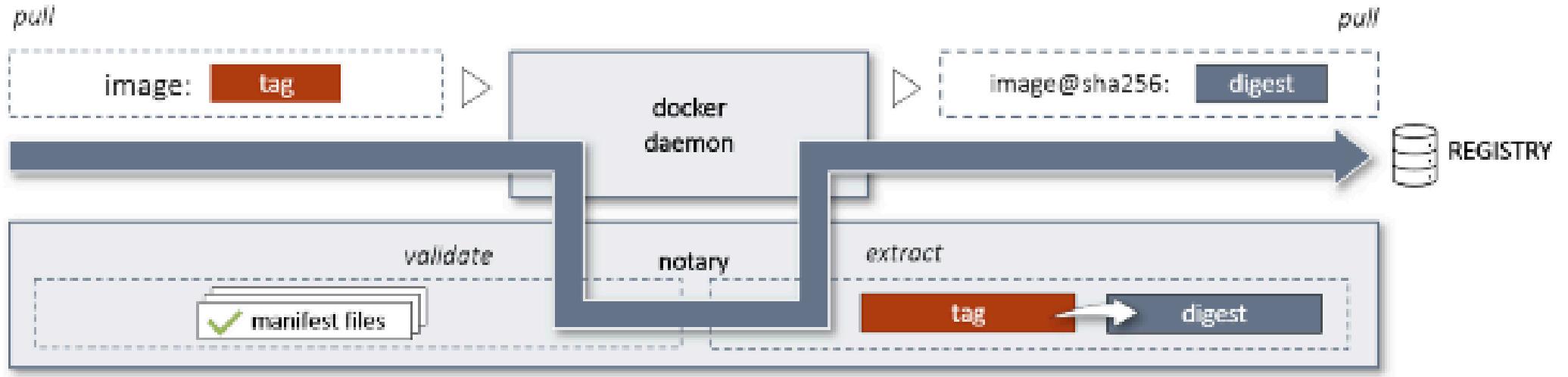


Яблоки у курином бульоне

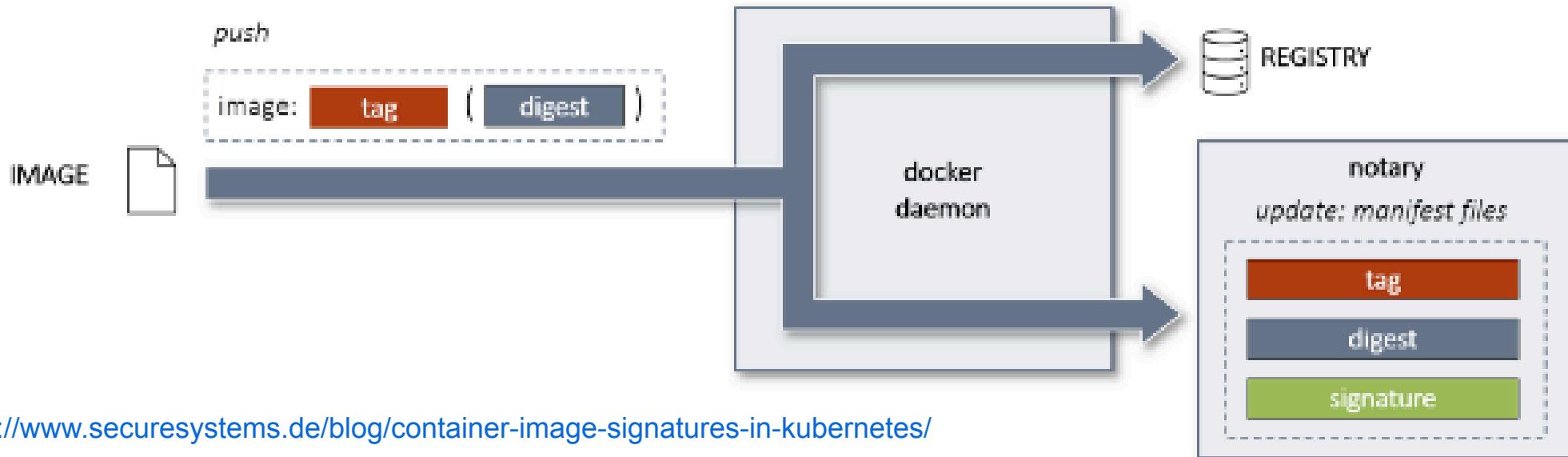
DCT + Notary

Docker Content Trust (DCT) - инструментарий, позволяющий подписывать части собираемых образов и публиковать в Notary

Notary - сервис, который хранит подписи для файлов



DCT + Notary (scheme)



DCT + Notary (example fail)

```
$ export DOCKER_CONTENT_TRUST=1
```

```
$ docker image pull nigelpoulton/tu-demo  
Using default tag: latest
```

```
Error: remote trust data does not exist for docker.io/nigelpoulton/tu-demo:  
- notary.docker.io does not have trust data for docker.io/nigelpoulton/tu-demo
```

```
$ docker image pull alpine:latest  
Pull (1 of 1): alpine:latest@sha256:c5b. . .d4f27761f8e1ad6b  
docker.io/library/alpine@sha256:c5b. . .d6b: Pulling from library/alpine  
4abcf2066143: Pull complete  
Digest: sha256:c5b. . .d6b  
Status: Downloaded newer image for alpine@sha256:c5b. . .7d6b  
Tagging alpine@sha256:c5b. . .d6b as alpine:latest  
docker.io/library/alpine:latest
```

DCT + Notary (example success)

```
$ docker trust key generate dctkey
```

```
Successfully generated and loaded private key.  
Corresponding public key available: /dirname/dctkey.pub
```

```
$ ls  
dctkey.pub
```

```
$ docker push rusdacent/hello-container:dct
```

```
$ docker trust sign rusdacent/hello-container:dct
```

DCT + Notary (Demo)

```
$ export DOCKER_CONTENT_TRUST=1

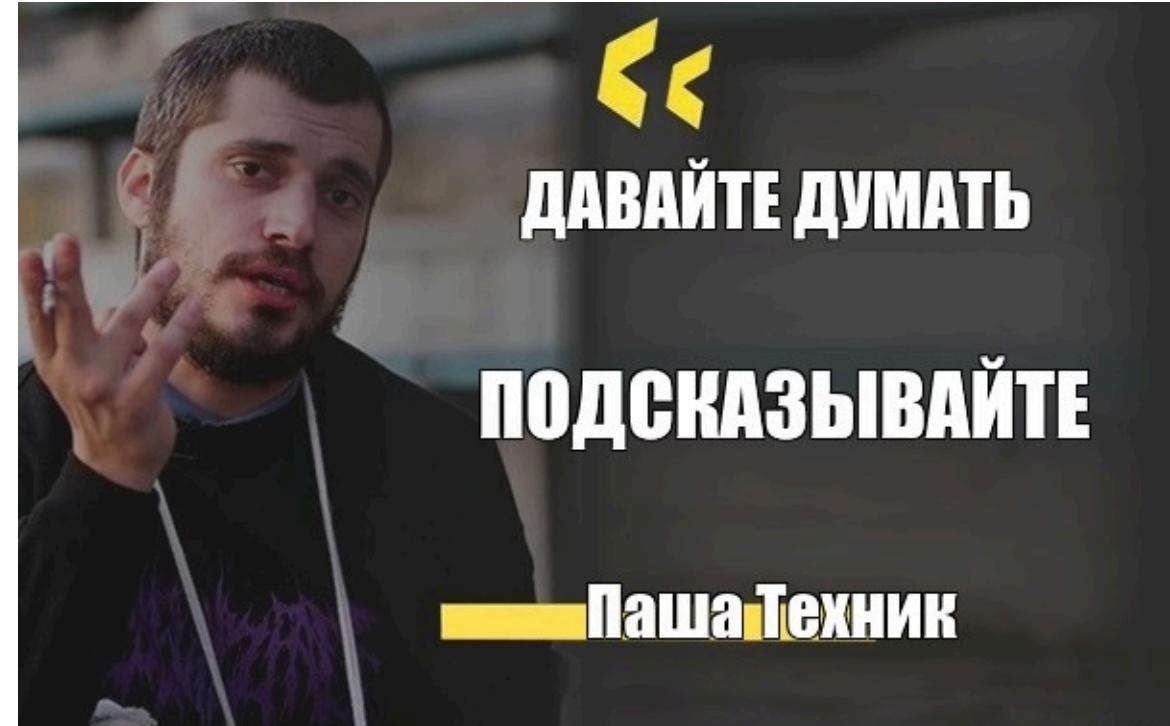
$ docker pull rusdacent/hello-container:latest
No valid trust data for latest

$ docker pull rusdacent/hello-container:dct
Pull (1 of 1): rusdacent/hello-container:dct@sha256:db5...452
docker.io/rusdacent/hello-container@sha256:db5...452: Pulling from rusdacent/hello-container
43c4264eed91: Already exists
Digest: sha256:db5...452
Status: Downloaded newer image for rusdacent/hello-container@sha256:db5...452
Tagging rusdacent/hello-container@sha256:db5...452 as rusdacent/hello-container:dct
docker.io/rusdacent/hello-container:dct
```

Practice (Public -> Private)

Какие правила можно
придумать для переноса
образов во внутренний
периметр?

Давайте подумаем

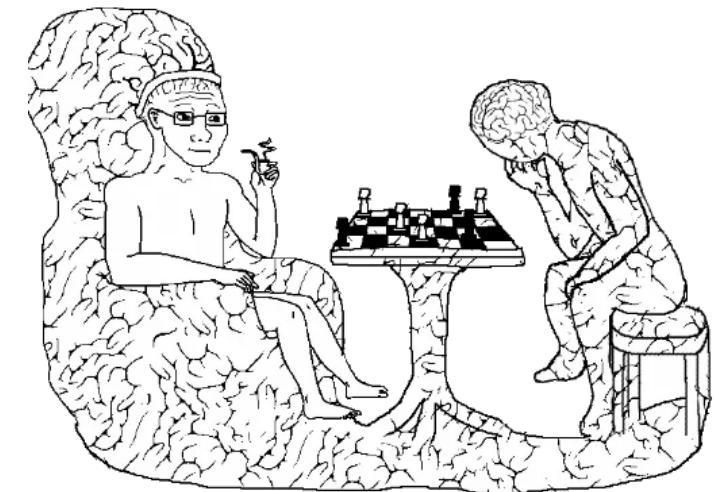


Practice (Public -> Private)



Какой концептуальный вывод?

Делая "SAST" не забывайте о "DAST"!



Демо пайплайн со всеми инструментами



Демо примеры



Всем спасибо!

Вопросы?