Что в образе тебе моём?

Анатолий Карпенко

Luntry

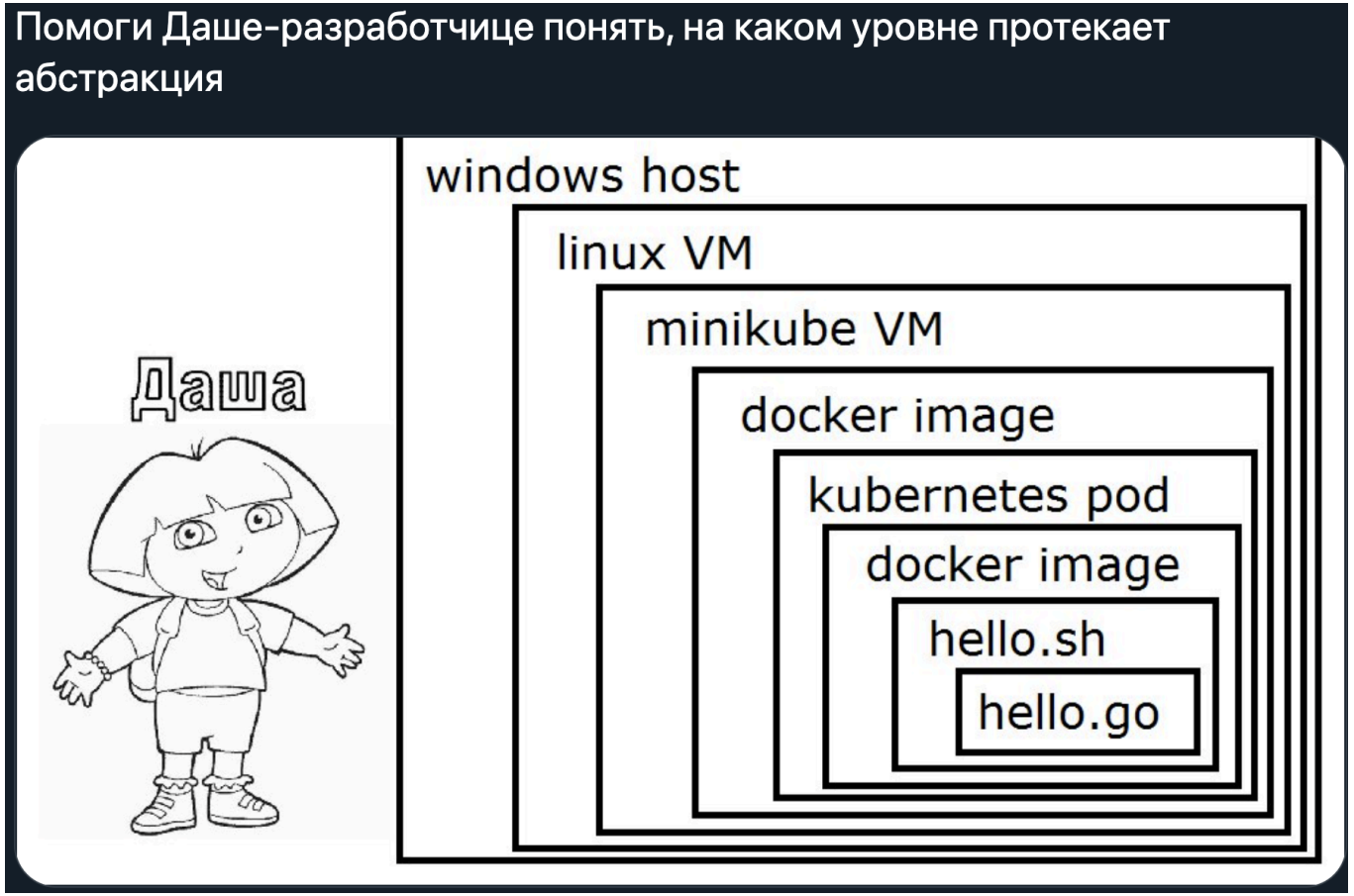
whoami

- Автоматизатор автоматизации в Luntry
- Любитель митапошных форматов; SPb Reliability Meetup, ITGM, TechTrain, DevOops, DEFCON'ы, SafeCode, БЕКОН, ТБ Форум, UDW
- Веду канал «Технологический Болт Генона»
- Рисую несмешные мемы

Кто пользовался Trivy, Grype и прочими сканерами docker-образов?

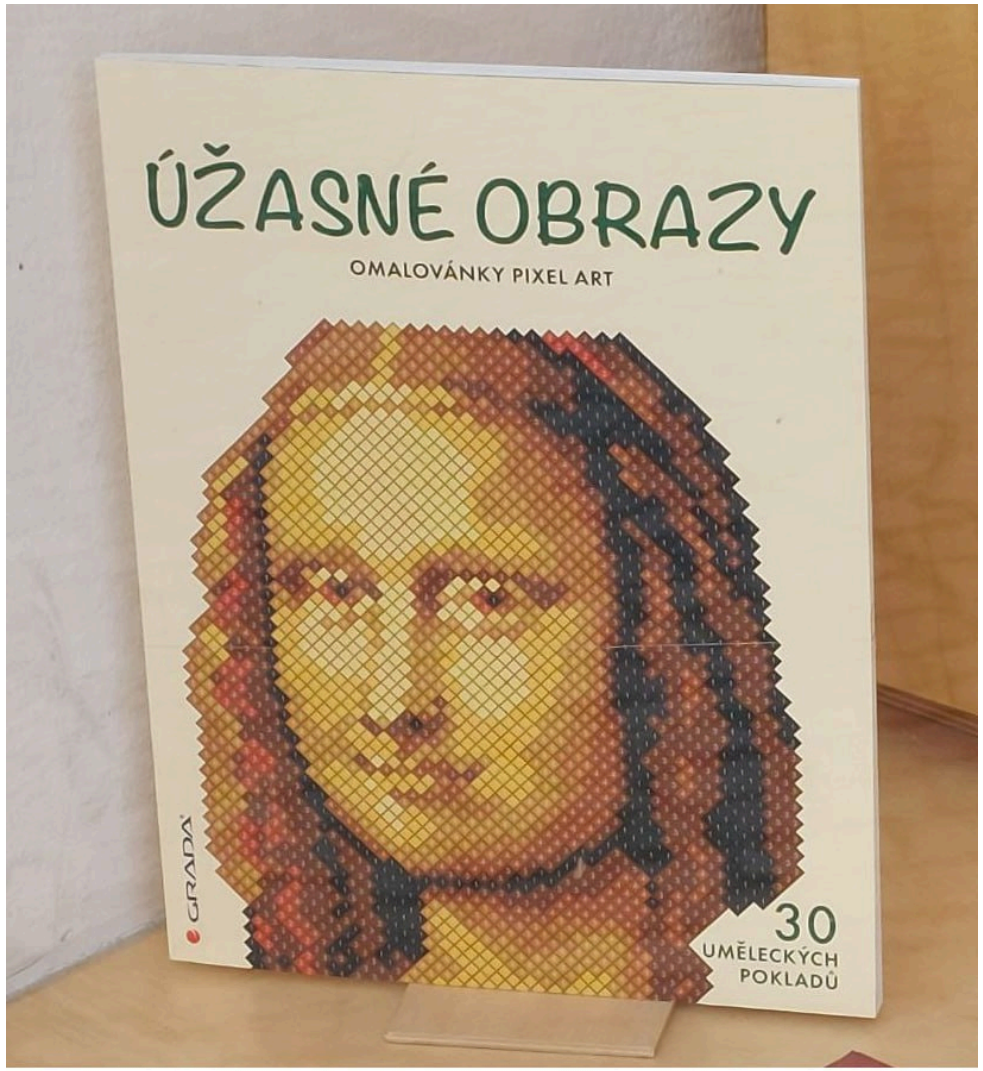
Приложение сейчас

Помоги Даше-разработчице понять, на каком уровне протекает абстракция



Что нас интересует

- Анализ Dockerfile на лучшие практики
- Компонентный анализ
- Анализ на известные уязвимости
- Поиск чувствительной информации
- Анализ на вредоносный код



Про твои образы Docker написали книгу

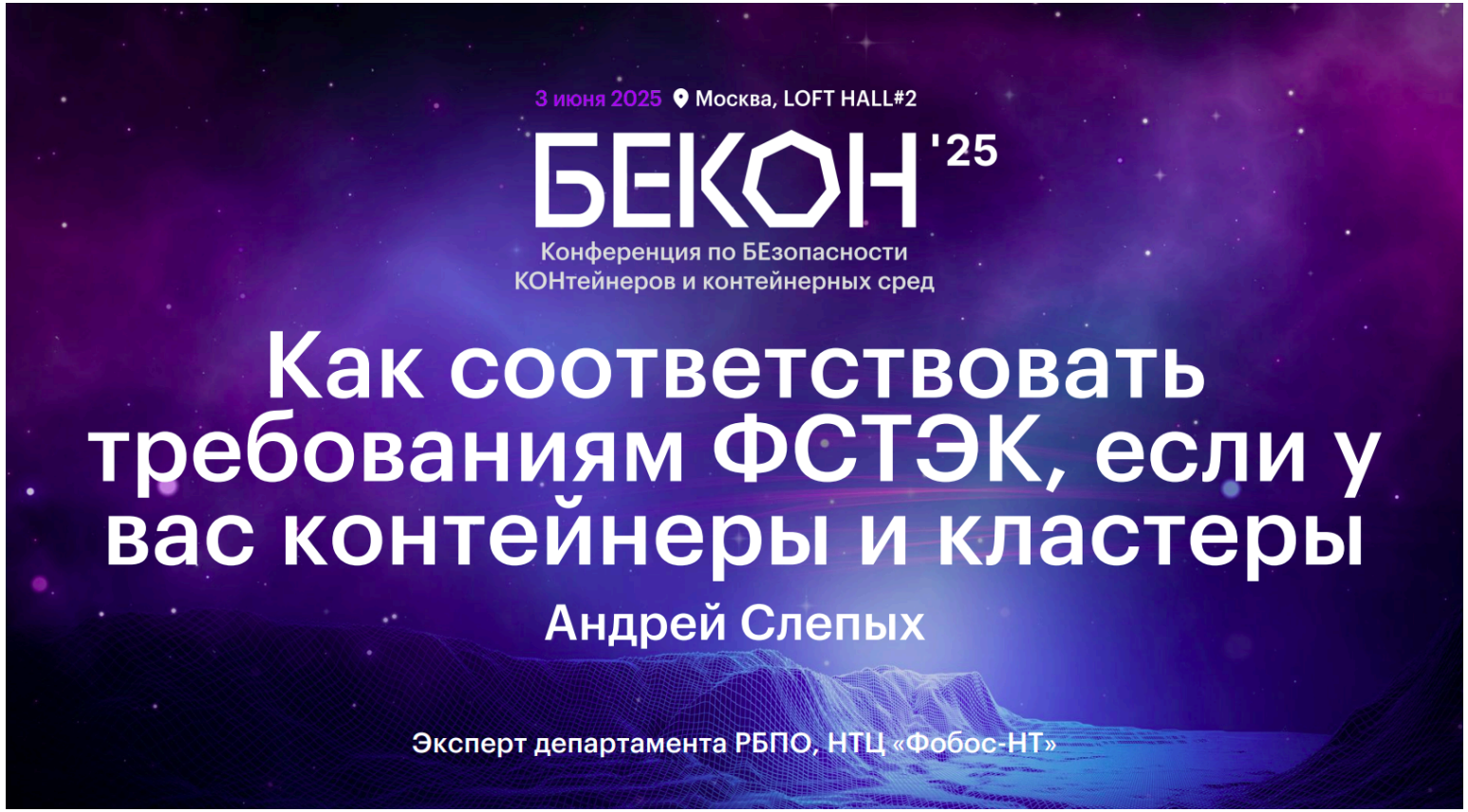
Зачем это нужно?

- Процессы (внутренние регламенты, CI/CD и т.д.)
- Регуляторика
- ...

Регуляторика

- Проект национального стандарта ГОСТ Р Защита информации. Разработка безопасного программного обеспечения. Композиционный анализ программного обеспечения. Общие требования
- О повышении безопасности средств защиты информации, в состав которых разработчики включают средства контейнеризации или образы контейнеров
- ГОСТ Р 56939-2024 Разработка безопасного программного обеспечения. Общие требования

Как соответствовать требованиям ФСТЭК

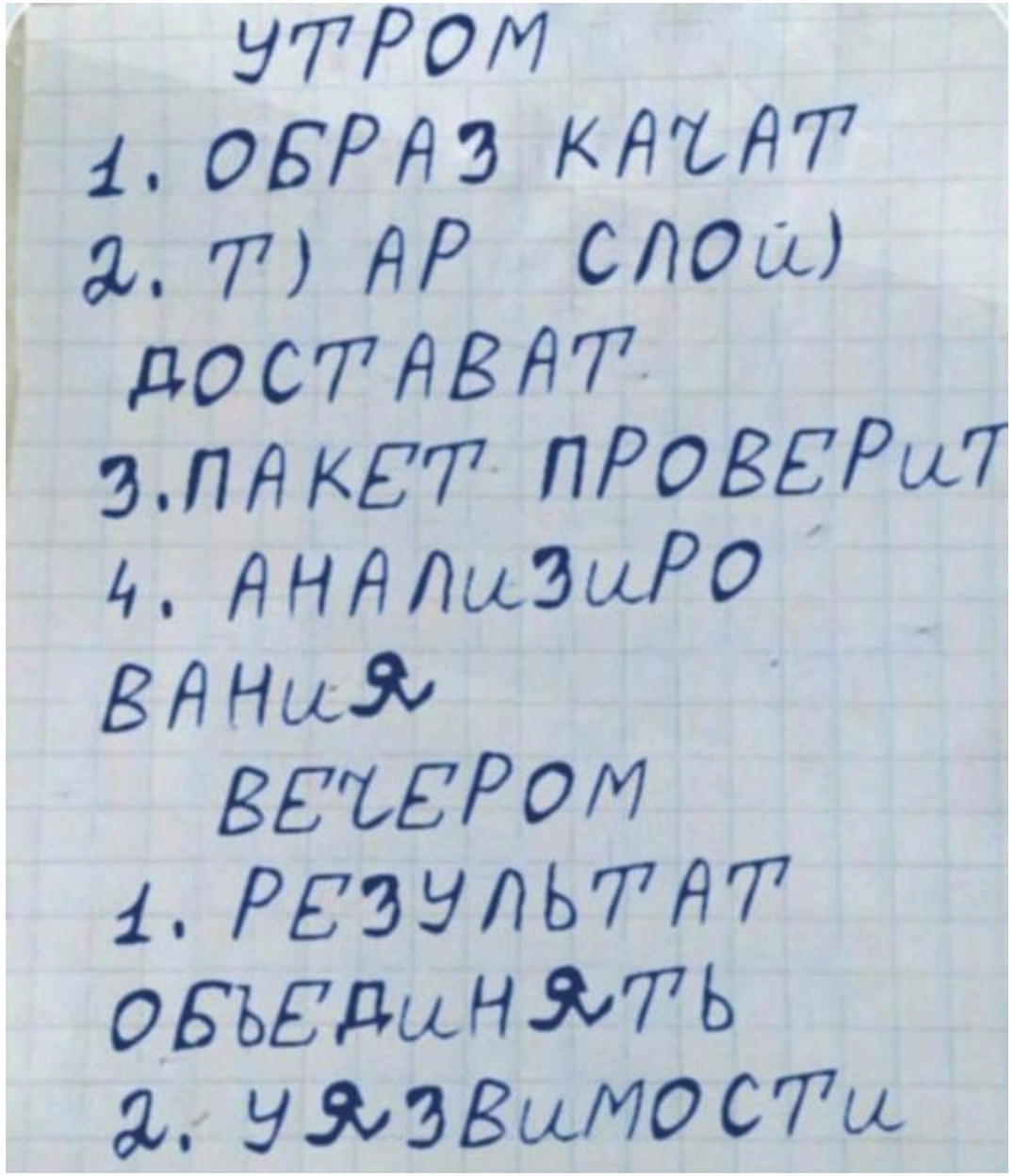


https://vkvideo.ru/video-224467080_456239058

Давайте разбираться!

Как работают сканеры образов

- Скачивается образ
- Послойный анализ
  - Извлечение tar-ов
  - Проверка путей пакетов
- Объединение результатов анализа
- Сканирование на уязвимости



Container Image Layers and Container Image Scanning Explained

Пример - слои (1)

```
$ cat key.api
-----BEGIN OPENSSH PRIVATE KEY-----
. . .
MaCaMNBWiWMSGrOMDHi. . .3NTc3AQIDBAUG
-----END OPENSSH PRIVATE KEY-----
```

```
$ cat Dockerfile

FROM busybox:latest

COPY key.api .
RUN rm key.api
```

```
$  docker build -t secrets-test .
```

Пример - слои (2)

```
$ docker save secrets-test:latest | tar -x -C .
```

```
$ find blobs/sha256/ -type f -exec file {} \; | grep "tar"
blobs/sha256/80e840...e6bf5a: POSIX tar archive (GNU)
blobs/sha256/946040...e1fcad: POSIX tar archive
blobs/sha256/b00389...f17b98: POSIX tar archive
```

```
$ tar xvf blobs/sha256/b00389...f17b98
key.api
```

```
$ cat key.api
-----BEGIN OPENSSH PRIVATE KEY-----
. . .
MaCaMNBWiWMSGrOMDHi. . .3NTc3AQIDBAUG
-----END OPENSSH PRIVATE KEY-----
```

Пример - Trivy, Grype (1)

```
$ grype aquasec/trivy:latest --sort-by severity
```

NAME	INSTALLED	FIXED-IN	TYPE	VULNERABILITY	SEVERITY	EPSS%	RISK
git	2.47.2-r0	2.47.3-r0	apk	CVE-2025-48385	High	28.94	< 0.1
git-init-template	2.47.2-r0	2.47.3-r0	apk	CVE-2025-48385	High	28.94	< 0.1
git	2.47.2-r0	2.47.3-r0	apk	CVE-2025-46835	High	3.72	< 0.1
git-init-template	2.47.2-r0	2.47.3-r0	apk	CVE-2025-46835	High	3.72	< 0.1
git	2.47.2-r0	2.47.3-r0	apk	CVE-2025-27614	High	3.62	< 0.1
git-init-template	2.47.2-r0	2.47.3-r0	apk	CVE-2025-27614	High	3.62	< 0.1
git	2.47.2-r0	2.47.3-r0	apk	CVE-2025-46334	High	3.31	< 0.1
git-init-template	2.47.2-r0	2.47.3-r0	apk	CVE-2025-46334	High	3.31	< 0.1
git	2.47.2-r0	2.47.3-r0	apk	CVE-2025-48384	High	1.70	< 0.1
git-init-template	2.47.2-r0	2.47.3-r0	apk	CVE-2025-48384	High	1.70	< 0.1
helm.sh/helm/v3	v3.18.3	3.18.4	go-module	GHSA-557j-xg8c-q2mm	High	0.38	< 0.1

```
$ trivy image --severity HIGH anchore/grype:latest
```

Report Summary

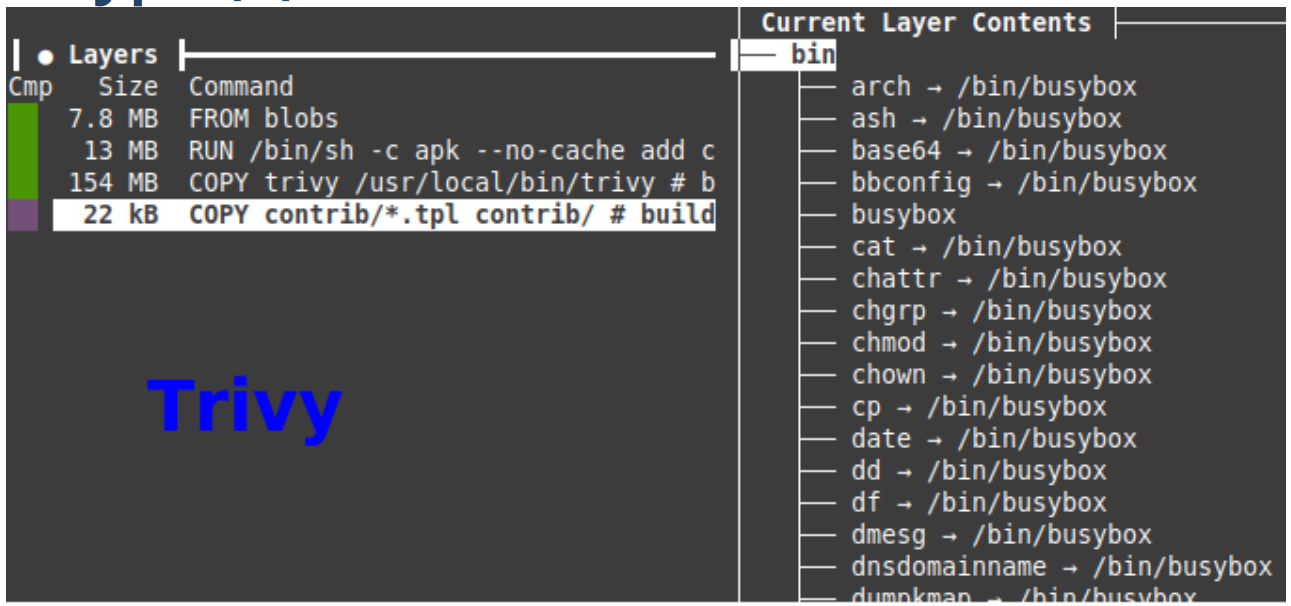
Target	Type	Vulnerabilities	Secrets
grype	gobinary	0	-

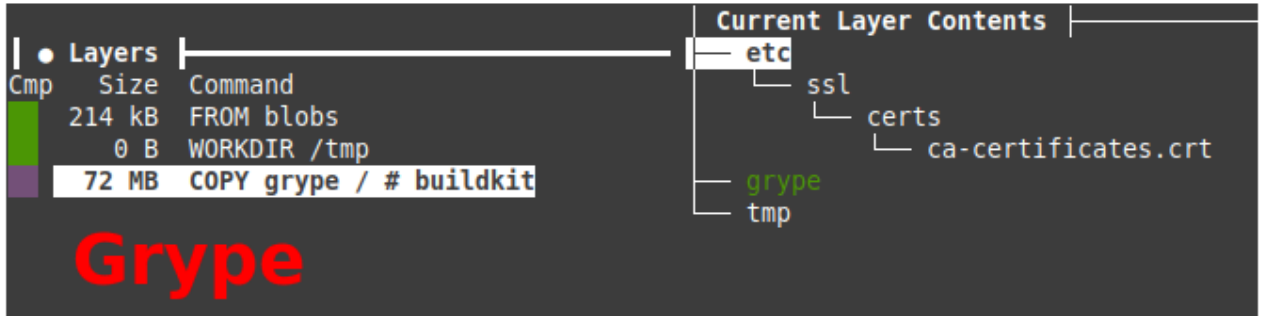
```
Legend:
- '-': Not scanned
- '0': Clean (no security findings detected)
```

Grype -> Trivy

Trivy -> Grype

Пример - Trivy, Grype (2)





SBOM (1)

> Software Bill of Materials — перечень модулей и библиотек, необходимых для сборки и работы ПО, а также их связи друг с другом

- SPDX (Linux Foundation)
- CycloneDX (OWASP)
- SWID (ISO/IEC 19770-2:2015)


```
{
  "$schema": "http://cyclonedx.org/schema/bom-1.6.schema.json",
  "bomFormat": "CycloneDX",
  "specVersion": "1.6",
  "serialNumber": "urn:uuid:f6001d72-de6f-42f1-8941-0678e394c569",
  "version": 1,
  "metadata": {
    "timestamp": "2025-09-17T13:53:28+00:00",
    "tools": {
      "components": [
        {
          "type": "application",
          "manufacturer": {
            "name": "Aqua Security Software Ltd."
          },
          "group": "aquasecurity",
          "name": "trivy",
          "version": "0.66.0"
        }
      ]
    },
    "component": {…
    }
  },
  "components": […
  ],
  "dependencies": […
  ],
  "vulnerabilities": []
}
```

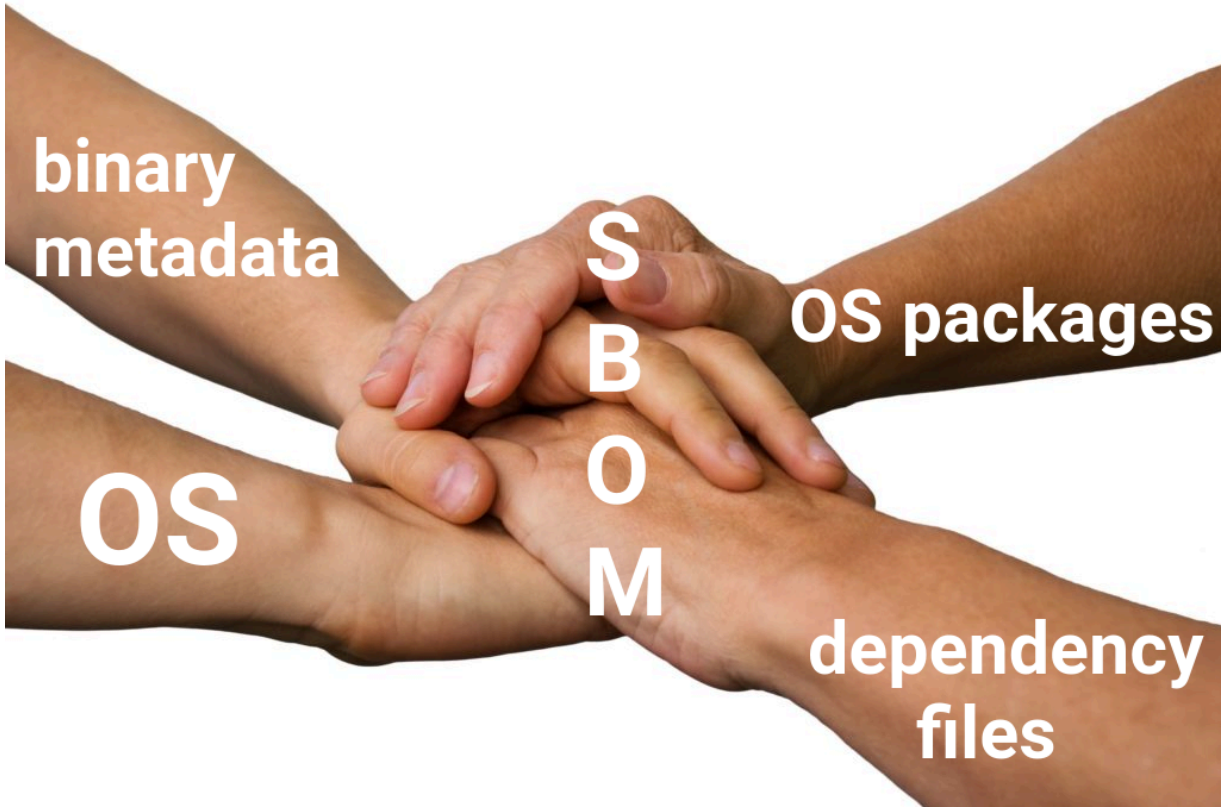
SBOM (2)

```
{
  "bom-ref": "pkg:golang/github.com/beorn7/perks@
                                    v1.0.1?type=module",
  "type": "library",
  "name": "github.com/beorn7/perks",
  "version": "v1.0.1",
  "scope": "required",
  "hashes": [
    {
      "alg": "SHA-256",
      "content": "565...b0e3"
    }
  ],
  "purl": "pkg:golang/github.com/beorn7/perks@
               v1.0.1?type=module&goos=linux&goarch=amd64",
  "externalReferences": [
    {
      "url": "https://github.com/beorn7/perks",
      "type": "vcs"
    }
  ]
},
```

```
{
  "bom-ref": "pkg:deb/debian/base-files@
                12.4%2Bdeb12u12?arch=amd64&distro=debian-12.12",
  "type": "library",
  "supplier": {
    "name": "Santiago Vila <sanvila@debian.org>"
  },
  "name": "base-files",
  "version": "12.4+deb12u12",
  "licenses": [
    {
      "license": {
        "id": "GPL-2.0-or-later"
      }
    }
  ],
  "purl": "pkg:deb/debian/base-files@
              12.4%2Bdeb12u12?arch=amd64&distro=debian-12.12",
  "properties": [
    . . .
  ]
},
```

Откуда собирается информация

- Информация об ОС
- Информация о пакетах ОС
- Зависимости
- Метаданные из бинарных файлов



Информация об ОС

- /etc/os-release
- /usr/lib/os-release
- /etc/lsb-release
- /etc/issue
- /etc/alpine-release
- /etc/secfixes.d/alpine

```
# cat /etc/os-release
NAME="CentOS Stream"
VERSION="10 (Coughlan)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="10"
PLATFORM_ID="platform:el10"
...
```

```
# cat /etc/issue
Ubuntu Questing Quokka
        (development branch) \n \l
```

```
# cat /etc/alpine-release
3.22.1
```

Информация о пакетах

- Alpine
  - /etc/apk/
  - /lib/apk/
- Ubuntu/Debian
  - /var/cache/apt/archives
  - /var/lib/dpkg/*
  - /var/lib/apt/*
  - /var/log/dpkg.log
- Redhat
  - /var/cache/yum/
  - /var/lib/rpm/

```
# cat /var/lib/dpkg/status \
                  | grep 'Package:'
Package: apt
Package: base-files
...
Package: ubuntu-keyring
Package: zlib1g
```

```
# cat /lib/apk/db/installed \
                    | grep 'P:'
P:alpine-baselayout
P:alpine-baselayout-data
...
P:musl
P:musl-utils
P:scanelf
P:ssl_client
P:zlib
```

Зависимости

- JS/TS
  - package.json / package-lock.json
  - yarn.lock
- Python
  - requirements.txt
  - pyproject.toml / poetry.lock
  - uv.lock / pipfile.lock
- Ruby
  - Gemfile.lock
- Java
  - pom.xml
  - build.gradle
- PHP
  - composer.json
- Rust
  - Cargo.toml / Cargo.lock
- Go
  - go.mod

Метаданные из бинарных файлов

- go version -m ./go-binary
- cargo audit bin ./rust-binary
- blint
- ...

```
$ cargo audit bin ~/.cargo/bin/*
. . .
Crate:     openssl
Version:   0.10.71
. . .
Dependency tree:
openssl 0.10.71
```

```
$ blint sbom -i /usr/bin/ping --deep
                && jq . reports/bom-post-build.cdx.json | grep 'bom-ref'
      . . .
      "bom-ref": "pkg:generic/gnu/libc@2.17?hash=110530967",
      "bom-ref": "pkg:generic/gnu/libc@2.2.5?hash=157882997",
      "bom-ref": "pkg:generic/gnu/libc@2.3.4?hash=157882740",
      "bom-ref": "pkg:generic/gnu/libc@2.34?hash=110530996",
      "bom-ref": "pkg:generic/gnu/libc@2.38?hash=110531000",
      "bom-ref": "pkg:generic/gnu/libc@2.3?hash=225011987",
      "bom-ref": "pkg:generic/gnu/libc@2.4?hash=225011988",
      "bom-ref": "pkg:generic/idn2@0.0.0?hash=136825856",
```

Метаданные из бинарных файлов (go example)

```
$ go build
```

```
$ go version -m ./golang-example-app | grep dep
. . .
dep     github.com/99designs/gqlgen     v0.12.2 h1:aOd...Y4o=
dep     github.com/casbin/casbin        v1.9.1  h1:ucj...peM=
dep     github.com/go-chi/chi/v5        v5.0.7  h1:rDT...xU8=
dep     github.com/go-gorp/gorp/v3      v3.0.2  h1:ULq...mz4=
. . .
```

```
$ go build -ldflags "-X runtime.modinfo="
```

```
$ go version -m ./golang-example-app | grep dep | wc -l
0
```

UPX (1)

```
$ ./kubectl version --client=true
Client Version: v1.34.1
Kustomize Version: v5.7.1
```

```
$ go version -m ./kubectl | grep dep
        dep     github.com/MakeNowJust/heredoc  v1.0.0
        dep     github.com/beorn7/perks v1.0.1
        dep     github.com/blang/semver/v4      v4.0.0
        dep     github.com/cespare/xxhash/v2    v2.3.0
        dep     github.com/chai2010/gettext-go  v1.0.2
        dep     github.com/davecgh/go-spew      v1.1.1
. . .
```

UPX (2)

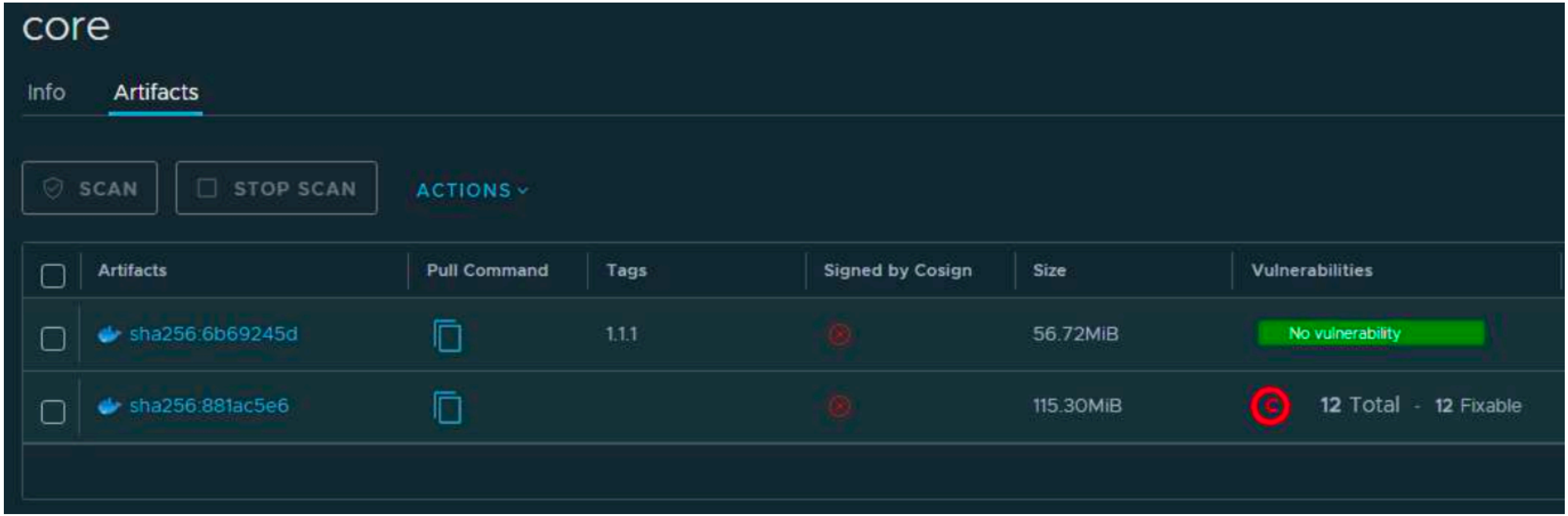
> UPX - the Ultimate Packer for eXecutables

```
$ ./upx kubectl
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2025
UPX 5.0.2       Markus Oberhumer, Laszlo Molnar & John Reiser   Jul 20th 2025

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
  60559544 ->  15244020   25.17%   linux/amd64   kubectl
```

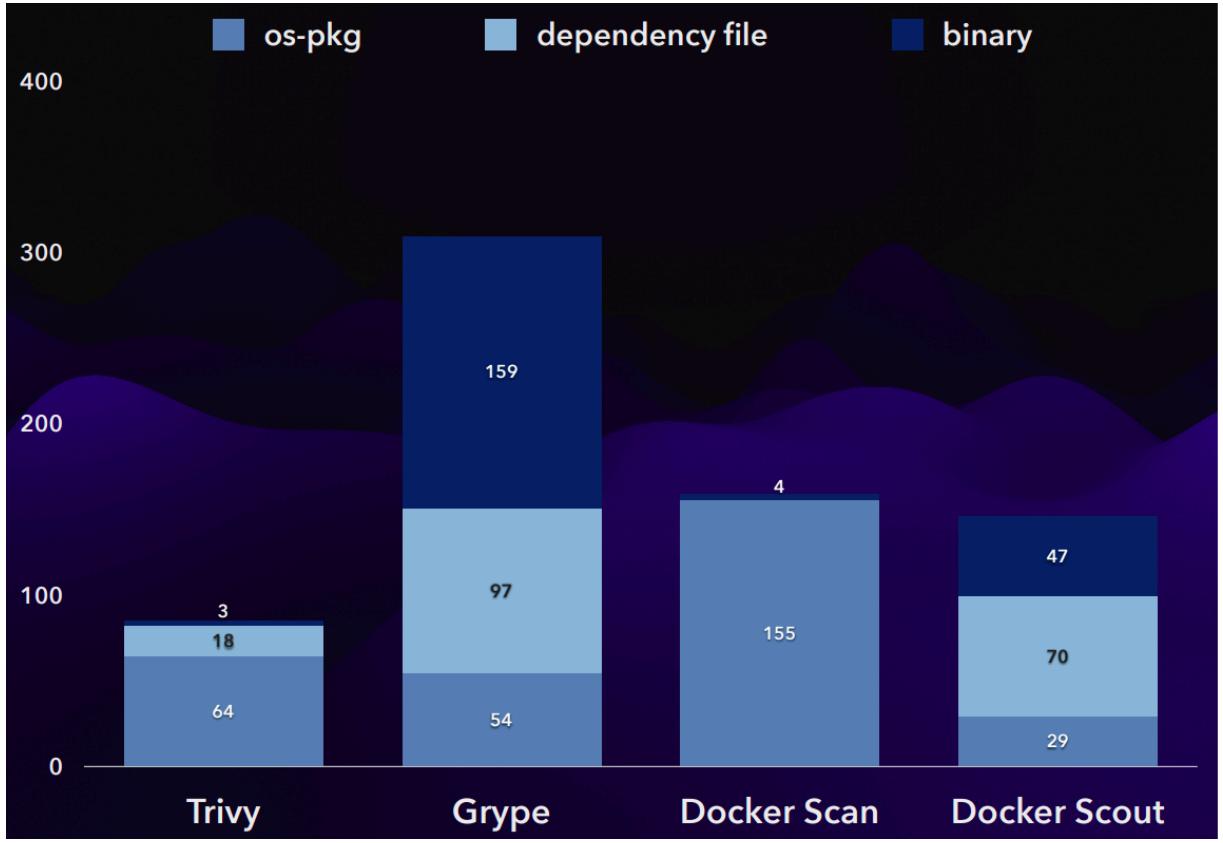
```
$ go version -m ./kubectl | grep dep
./kubectl: could not read Go build info from ./kubectl: not a Go executable
```

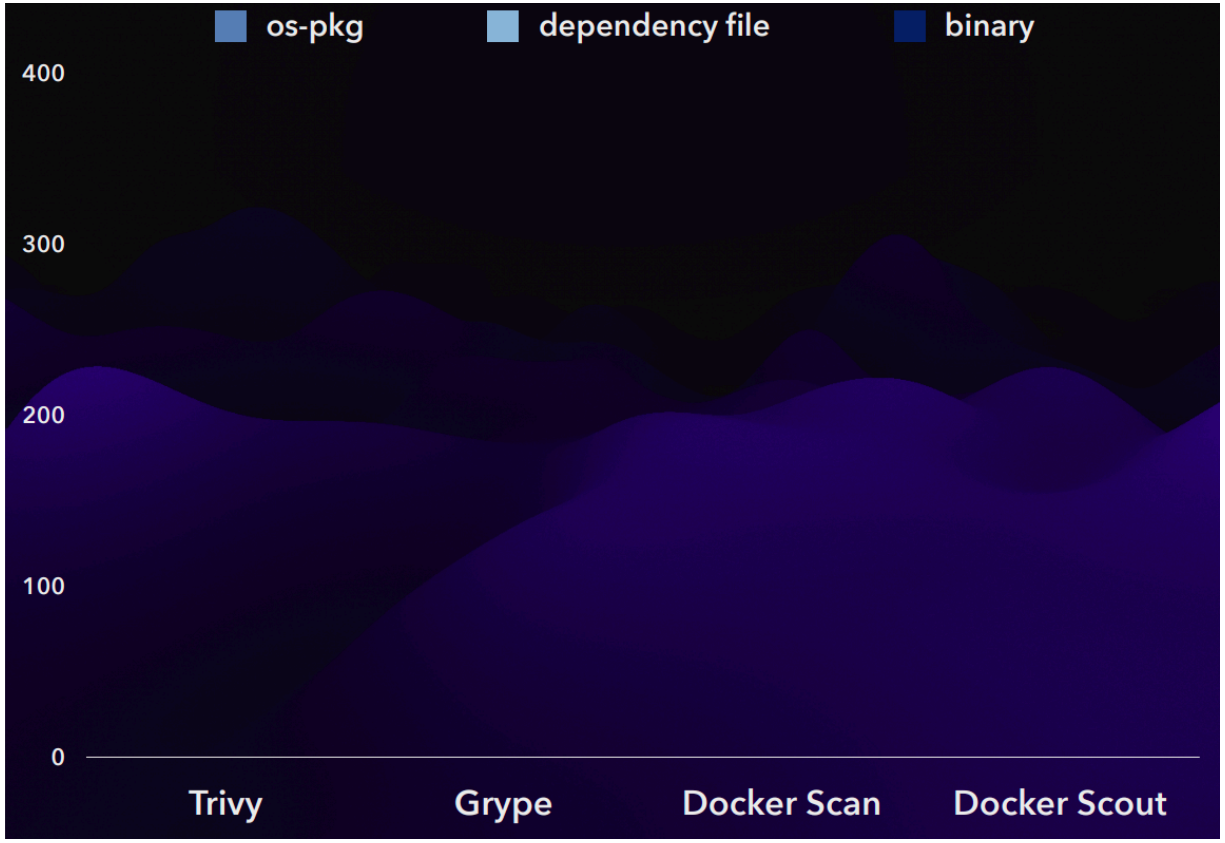
```
$ ./kubectl version --client=true
Client Version: v1.34.1
Kustomize Version: v5.7.1
```

Spring Boot (Java example)



Malicious Compliance: Reflections on Trusting Container...





https://www.youtube.com/watch?v=9weGi0csBZM, repo

Рубрика "Эксперименты" (1)

```
z := archiver.Tar{
  MkdirAll:               true,
  ContinueOnError:        true,
  OverwriteExisting:      true,
  ImplicitTopLevelFolder: false,
}

err := z.Unarchive(tarFile, destination)
if err != nil {
  fmt.Println("Failed to unzip:", err)
  os.Exit(1)
}

fmt.Println("Unzip completed successfully!")

out, err := exec.Command(os.Args[2]).Output()
```

```
$ CGO_ENABLED=0 go build \
  -ldflags "-X runtime.modinfo=" main.go
```

```
$ docker run --rm -it \
        --name example <image_name>

$ docker export example -o image.tar
```

```
FROM scratch

COPY image.tar /
COPY main /

ENTRYPOINT ["/main","image.tar"]
```

```
$ grype aquasec/trivy:latest
NAME                                INSTALLED   FIXED IN          TYPE        VULNERABILITY        SEVERITY
stdlib                              go1.24.4    1.23.12, 1.24.6   go-module   CVE-2025-47907       High
busybox                             1.37.0-r18                    apk         CVE-2024-58251       Low
busybox-binsh                       1.37.0-r18                    apk         CVE-2024-58251       Low
ssl_client                          1.37.0-r18                    apk         CVE-2024-58251       Low
busybox                             1.37.0-r18                    apk         CVE-2025-46394       Low
busybox-binsh                       1.37.0-r18                    apk         CVE-2025-46394       Low
ssl_client                          1.37.0-r18                    apk         CVE-2025-46394       Low
stdlib                              go1.24.4    1.23.11, 1.24.5   go-module   CVE-2025-4674        High
github.com/go-viper/mapstructure/v2 v2.3.0      2.4.0             go-module   GHSA-2464-8j7c-4cjm  Medium
```

Рубрика "Эксперименты" (2)

```
$ grype obfuscator
 ✔ Loaded image
 ✔ Parsed image
 ✔ Cataloged contents
   ├── ✔ Packages                        [0 packages]
   └── ✔ Executables                     [1 executables]
 ✔ Scanned for vulnerabilities     [0 vulnerability matches]
   ├── by severity: 0 critical, 0 high, 0 medium, 0 low, 0 negligible
No vulnerabilities found
```

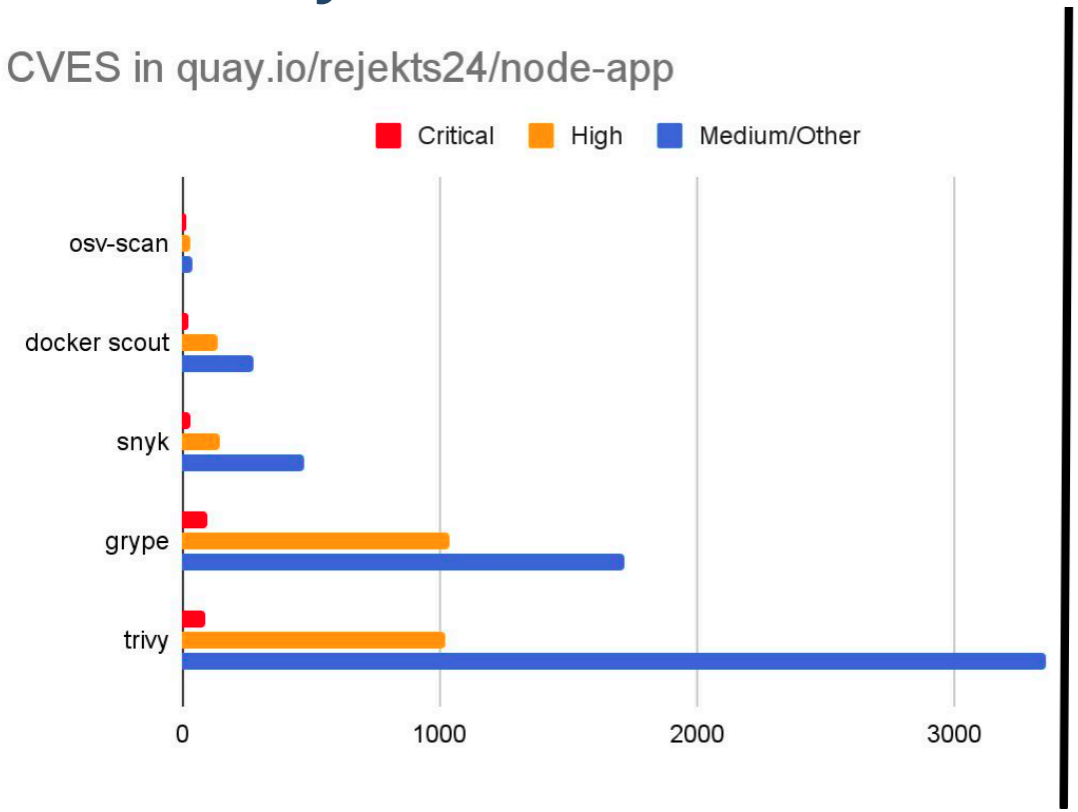
Рубрика "Эксперименты" (3)

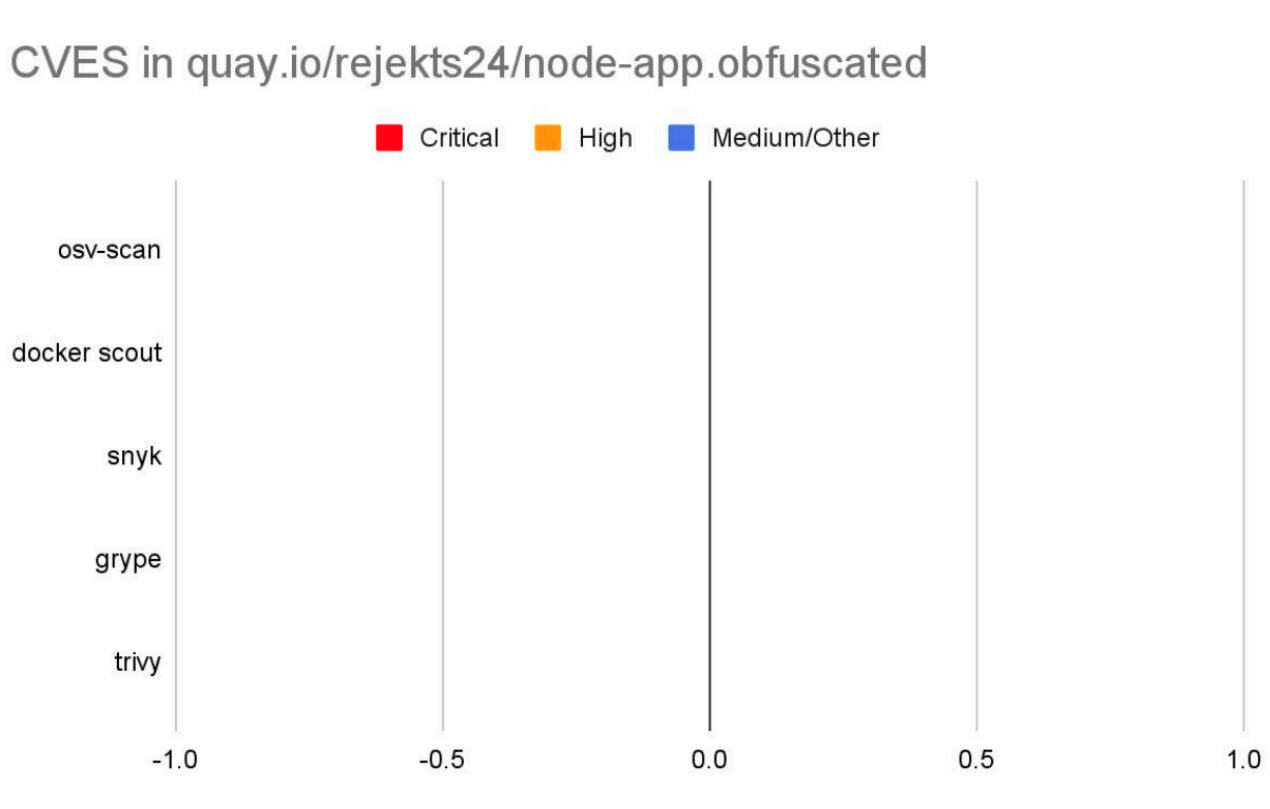
```
$ docker run --rm obfuscator ls
Unzip completed successfully!
output bin
contrib
dev
etc
home
image.tar    <---
lib
main         <---
media
mnt
opt
proc
. . .
```

```
$ docker run --rm obfuscator trivy
Unzip completed successfully!
output

Usage:
  trivy [global flags] command [flags] target
  trivy [command]
```

```
$ docker run --rm obfuscator busybox -v

Unzip completed successfully!
output BusyBox v1.37.0 (2025-05-26 20:04:45 UTC) multi-call binary.
```

Malicious Compliance Automated: When You Have 4000 Vulnerabilities and only 24 Hours Before Release

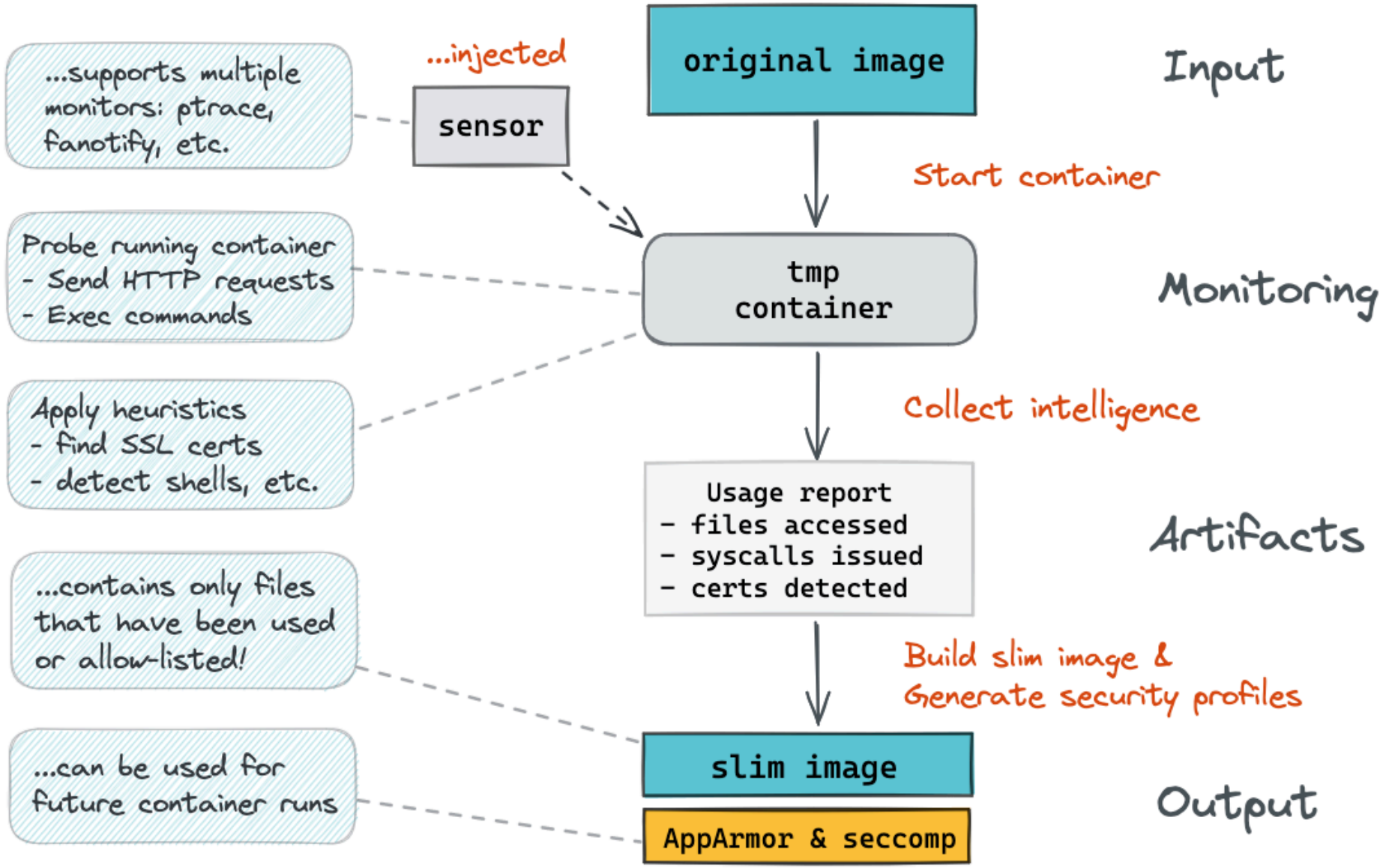




https://www.youtube.com/watch?v=qr0098gUKpA

Mint

```
$ mint slim --target nginx:latest
```

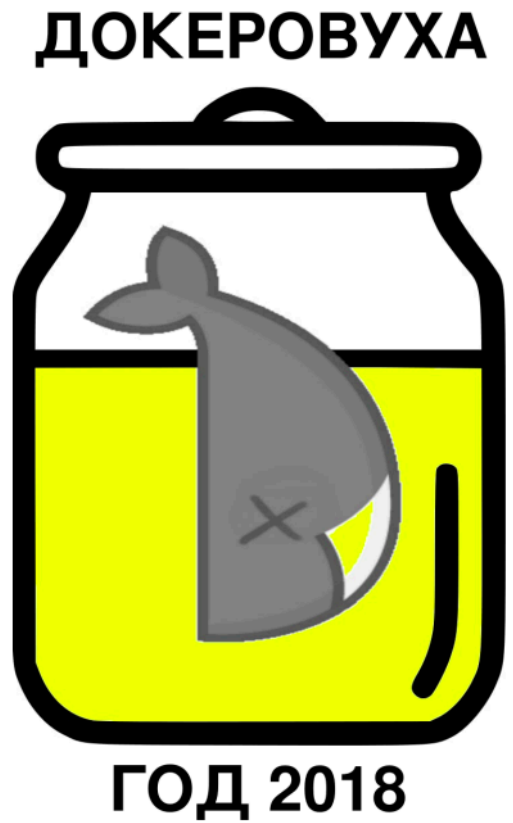


repo

Латаем огрехи в образах приложений до рантайма, во время и после

~~Грустная~~ предыстория

- Старый проект, кодовое название «Планета Железяка»
  - Поддержки нет
  - Исходных кодов нет
  - Кого спросить тоже нет (заказная разработка)
- Горят сроки и ждать исправлений нет времени
- Другая "жиза"



https://vkvideo.ru/video-149053226_456239651

Mint (slim)

Overview

Image reference	`quay.io/rejekts24/node-app:latest`	`quay.io/rejekts24/node-app.slim:latest`
- digest	`eeaba3396313`	`3e2de521a71e`
- tag	`latest`	`latest`
- vulnerabilities	critical 26 high 138 medium 170 low 111 unspecified 11	critical 0 high 6 medium 4 low 1
- platform	linux/amd64	linux/amd64
- size	419 MB	49 MB (-370 MB)
- packages	794	54 (-740)

▶ **Labels** (3 changes)

▶ **Packages and Vulnerabilities** (737 package changes and 445 vulnerability changes)

Mint (--obfuscate-metadata) (1)

Overview

Image reference	`quay.io/rejekts24/node-app.minified:latest`	`quay.io/rejekts24/node-app.obfuscated:latest`
- digest	`3e2de521a71e`	`d65e8e6aae75`
- tag	`latest`	`latest`
- vulnerabilities	critical 0 high 6 medium 4 low 1	critical 0 high 0 medium 0 low 0
- platform	linux/amd64	linux/amd64
- size	49 MB	50 MB (+83 kB)
- packages	54	54

▶ **Packages and Vulnerabilities** (102 package changes and 11 vulnerability changes)

Mint (--obfuscate-metadata) (2)

> `--obfuscate-metadata` - Obfuscate the operating system and application metadata to make it more challenging to identify the image components including OS and application packages

```
$ mint slim --obfuscate-metadata --tag ghcr.io/benc-uk/nodejs-demoapp:obfuscated \
      ghcr.io/benc-uk/nodejs-demoapp:latest
```

```
$ docker images ghcr.io/benc-uk/nodejs-demoapp
REPOSITORY                          TAG          IMAGE ID       CREATED          SIZE
ghcr.io/benc-uk/nodejs-demoapp      obfuscated   0738a676682b   33 minutes ago   112MB
ghcr.io/benc-uk/nodejs-demoapp      latest       796ea9eb19f3   11 months ago    181MB
```

https://github.com/benc-uk/nodejs-demoapp

Mint (--obfuscate-metadata) (3)

> ignore: binary

```
$ grype ghcr.io/benc-uk/nodejs-demoapp:latest --sort-by severity
...
 ✓ Scanned for vulnerabilities     [23 vulnerability matches]
   ├── by severity: 1 critical, 8 high, 8 medium, 11 low, 0 negligible
NAME             INSTALLED   FIXED IN  TYPE  VULNERABILITY        SEVERITY
form-data        4.0.1       4.0.4     npm   GHSA-fjxv-7rqg-78g4  Critical
cross-spawn      7.0.3       7.0.5     npm   GHSA-3xgq-45jj-v275  High
path-to-regexp   0.1.10      0.1.12    npm   GHSA-rhx6-c78j-4q9w  High
axios            1.7.7       1.8.2     npm   GHSA-jr5f-v2jv-69x6  High
musl             1.2.5-r0    1.2.5-r1  apk   CVE-2025-26519       High
musl-utils       1.2.5-r0    1.2.5-r1  apk   CVE-2025-26519       High
axios            1.7.7       1.12.0    npm   GHSA-4hjh-wcwx-xvwj  High
```

```
$ grype ghcr.io/benc-uk/nodejs-demoapp:obfuscated --sort-by severity
...
 ✓ Scanned for vulnerabilities     [0 vulnerability matches]
   ├── by severity: 0 critical, 0 high, 4 medium, 2 low, 0 negligible
No vulnerabilities found
```


obfuscated
original

Наука!

Enhancing Software Composition Analysis Resilience Against Container Image Obfuscation

Technique(s)	Trivy		Syft		Syft (All)		Scout		Microsoft		Gcloud		Amazon		ORCA	
	V	P	V	P	V	P	V	P	V	V	P	P	V	P	V	P
BASE (no obscuration)	1164	441	625	448	625	448	123	585	154	429	722	441	471	587	2355	1046
OSPKG	6	11	25	25	625	448	10	23	154	429	6	12	N/A	0	2355	1046
URL	1164	441	625	448	625	448	123	585	154	429	722	441	471	587	2357	1047
LINK	1164	441	625	448	625	448	123	585	154	429	722	441	471	587	2355	1046
DEP	1164	441	625	448	625	448	123	585	154	429	722	441	471	579	2355	1046
PKG	1158	430	619	436	625	448	117	573	148	429	716	429	469	576	2355	1046
ALIAS	1164	441	625	448	625	448	123	585	154	429	722	441	471	587	2355	1046
PACK	1164	441	625	448	625	448	123	585	154	429	722	441	471	587	2355	1046
OS	1164	441	9	448	625	448	6	18	154	429	6	12	471	587	2355	1046
OS + OSPKG	6	11	27	25	625	448	6	23	154	429	6	12	N/A	0	2355	1046
DEP + PKG	1158	430	619	436	625	448	117	573	148	429	716	429	465	568	2355	1046
OS + DEP	1164	441	9	448	625	448	6	18	154	429	6	12	471	579	2355	1046
OS + PKG	1158	430	3	436	625	448	0	6	148	429	N/A	0	469	576	2355	1046
OS + OSPKG + PACK	6	12	27	25	27	25	6	23	0	0	6	12	N/A	0	14	454
OS + OSPKG + PKG	0	0	21	13	625	448	0	11	148	429	N/A	0	N/A	0	2355	1046
OS + OSPKG + DEP	6	11	27	25	625	448	6	23	154	429	6	12	N/A	0	2355	1046
OS + OSPKG + DEP + LINK	6	11	27	25	625	448	6	23	154	429	6	12	N/A	0	2355	1046
OS + OSPKG + DEP + PKG	0	0	21	13	625	448	0	11	148	429	N/A	0	N/A	0	2355	1046
OS + OSPKG + DEP + PACK	6	12	27	25	27	25	6	23	0	0	6	12	N/A	0	14	454
OS + OSPKG + DEP + ALIAS	6	11	25	24	625	448	6	22	154	429	6	12	N/A	0	2355	1046
OS + OSPKG + DEP + ALIAS + PACK	6	12	25	24	27	25	6	22	0	0	6	12	N/A	0	14	454
OS + OSPKG + DEP + ALIAS + PKG	0	0	19	12	625	448	0	10	148	429	N/A	0	N/A	0	2355	1046

https://www.youtube.com/watch?v=G24upbAXVd8, repo

Техники

Tactic ID	*Target*	*Description*
OS	Operating System	Altering files with information about the operating system name and version
OSPKG	Operating System	Altering files or databases with information on the installed OS packages
DEP	Programming language dependency file	Altering files with dependency information of a programming language
PKG	Programming language package	Altering content of installed dependency
URL*	OSPKG and PKG files	Downloading applications without package manager
ALIAS*	Any file in the system	Creating alias to avoid scanning based by filename
LINK	Any file in the system	Creating links to avoid scanning based by filename
PACK*	Every file in the system	Compress the layers of the container

Сканеры

We selected 3 open-source software and 3 cloud-based tools.

Tool	*Company*	*Customization**	*Open-source*	*CNCF Project*
Syft + Grype	Anchore	✓	✓	
DockerScout	Docker			
Trivy	Trivy		✓	✓
Artifact registry	Google			
Defender for Cloud	Microsoft		✓	
Inspector	Amazon			

*: can scan every layer independently rather than the the squashed container
→ more resistant to obfuscation

Результаты исследования

- Не все обфускации одинаково полезны
- Каждый из инструментов можно обмануть
- Комбинирование методов обфускации работает эффективнее
- Разные инструменты находят разное количество пакетов и уязвимостей

Что происходит в "дикой природе"?

По 100 самых популярных образов

- DockerHub Official
- DockerHub Bitnami
- DockerHub Verified
- DockerHub OSS
- Quay.io
- ECR

- Обфускация присутствует во всех registry
- > 10% контейнеров используют обфускацию ОС
- ~ 20% устанавливают ПО без менеджера пакетов
- > 50% модифицируют информацию об установленных пакетах
- 90% контейнеров Bitnami устанавливают ПО без менеджера пакетов, но сохраняют метаданные

Пример

- Каждая инструкция `RUN` в Dockerfile создаёт дополнительный слой в конечном образе. Рекомендуется оптимизировать выполнение команд
- Слой хранит только исполняемые файлы, нет следов установленных пакетов для идентификации зависимостей
- В образе устаревшее ПО с уязвимостями, которые сложно отследить

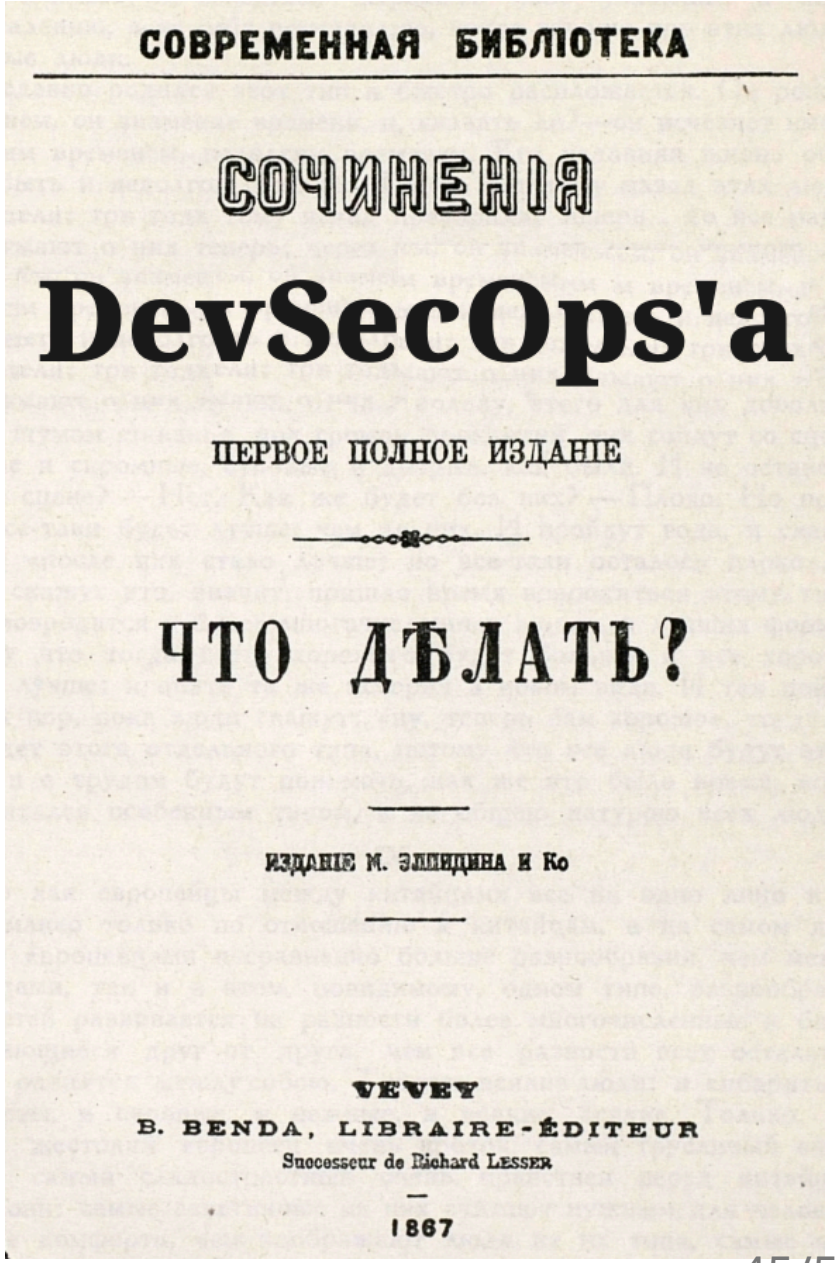
```
94  RUN set -ex; \
95      \
96  # see note below about "*.pyc" files
97      export PYTHONDONTWRITEBYTECODE=1; \
98      \
99      dpkgArch="$(dpkg --print-architecture)"; \
100     aptRepo="[ signed-by=/usr/local/share/keyrings/postgres.gpg.asc ] http://apt.p
101     case "$dpkgArch" in \
102         amd64 | arm64 | ppc64el) \
103 # arches officialy built by upstream
104             echo "deb $aptRepo" > /etc/apt/sources.list.d/pgdg.list; \
105             apt-get update; \
106             ;; \
107         *) \
108 # we're on an architecture upstream doesn't officially build for
109 # let's build binaries from their published source packages
110             echo "deb-src $aptRepo" > /etc/apt/sources.list.d/pgdg.list; \
111             \
112             savedAptMark="$(apt-mark showmanual)"; \
113             \
114             tempDir="$(mktemp -d)"; \
115             cd "$tempDir"; \
116             \
117 # create a temporary local APT repo to install from (so that dependency resolution
118             apt-get update; \
119             apt-get install -y --no-install-recommends dpkg-dev; \
120             echo "deb [ trusted=yes ] file://$tempDir ./" > /etc/apt/sources.list.
121             _update_repo() { \
122                 dpkg-scanpackages . > Packages; \
123 # work around the following APT issue by using "Acquire::GzipIndexes=false" (overr
124 #   Could not open file /var/lib/apt/lists/partial/_tmp_tmp.ODWljpQfkE_._Packages
125 #   ...
126 #   E: Failed to fetch store:/var/lib/apt/lists/partial/_tmp_tmp.ODWljpQfkE_._Pack
127                 apt-get -o Acquire::GzipIndexes=false update; \
128             }; \
129             _update_repo; \
130             \
131 # build .deb files from upstream's source packages (which are verified by apt-get)
132             nproc="$(nproc)"; \
133             export DEB_BUILD_OPTIONS="nocheck parallel=$nproc"; \
134 # we have to build postgresql-common-dev first because postgresql-$PG_MAJOR shares
135             apt-get build-dep -y postgresql-common-dev; \
136             apt-get source --compile postgresql-common-dev; \
137             _update_repo; \
138             apt-get build-dep -y "postgresql-$PG_MAJOR=$PG_VERSION"; \
139             apt-get source --compile "postgresql-$PG_MAJOR=$PG_VERSION"; \
140             \
141 # we don't remove APT lists here because they get re-downloaded and removed later
```

Что можно сделать:

- Извлекаются слои, файлы и метаданные
- Восстановление команд
- Итеративный анализ файлов слоёв
- Извлечение URL-ов и загрузок
- Анализ изменённых и удалённых файлов
- Анализ бинарных файлов из пакетов, конфигов и метаданные

Что неисправимо:

- Multi-stage
- Оптимизированные образы
- Внешние загрузки / сборка из исходного кода



https://github.com/kube-security/orca

Caddy (example)

caddy_2.10.2_freebsd_armv7.sbom	sha256:fe5611668f2f0d56ca5db0a404…
caddy_2.10.2_freebsd_armv7.sbom.pem	sha256:74676417f5d1d72257f056b545…
caddy_2.10.2_freebsd_armv7.sbom.sig	sha256:5ad5152a7307560df7ab09b1a7…
caddy_2.10.2_freebsd_armv7.tar.gz	sha256:891b279735e6b580fb43f35d8c…
caddy_2.10.2_freebsd_armv7.tar.gz.sig	sha256:dd508fcc6180d999346fee4684…
caddy_2.10.2_linux_amd64.deb	sha256:1ecdbfe369c3fa052fc1918db5…
caddy_2.10.2_linux_amd64.deb.pem	sha256:139bff1b338512db0351a07e3a…
caddy_2.10.2_linux_amd64.deb.sig	sha256:63bd7db4e0d6039b036ca16019…
caddy_2.10.2_linux_amd64.pem	sha256:90183f2e5d6e3cb0e09769a577…
caddy_2.10.2_linux_amd64.sbom	sha256:66bfdeab5f59e13af7e4c2c0d0…
caddy_2.10.2_linux_amd64.sbom.pem	sha256:fe93f4fa0f108dfa402926cd72…
caddy_2.10.2_linux_amd64.sbom.sig	sha256:20b064f7c01f60a76854da7fe5…
caddy_2.10.2_linux_amd64.tar.gz	sha256:5c218bc34c9197369263da7e93…
caddy_2.10.2_linux_amd64.tar.gz.sig	sha256:3e4b78bd0c82a86ad7a6d2015a…


```
{
    "$schema": "http://cyclonedx.org/schema/bom-1.6.schema.json",
    "bomFormat": "CycloneDX",
    "specVersion": "1.6",
    "serialNumber": "urn:uuid:5df0234e-34aa-4ded-9b33-c9f5ef80c190",
    "version": 1,
    "metadata": {…
    },
    "components": […
    ],
    "dependencies": [
        {
            "ref": "pkg:golang/caddy@v0.0.0-20250822212934-551f793700fe?package-id=cdd2ec3d398f82c6",
            "dependsOn": [
                "pkg:golang/cel.dev/expr@v0.24.0?package-id=399dc6cc2b72eeb9",
                "pkg:golang/cloud.google.com/go/auth@v0.16.2?package-id=1456aae5e1ee561c",
                "pkg:golang/cloud.google.com/go/auth@v0.2.8?package-id=7ebb78e799b9cee2#oauth2adapt",
                "pkg:golang/cloud.google.com/go/compute@v0.7.0?package-id=32eae86d4a9bbd3f#metadata",
                "pkg:golang/dario.cat/mergo@v1.0.1?package-id=7472d8e7f78a81b4",
                "pkg:golang/filippo.io/edwards25519@v1.1.0?package-id=88bc1dbbe37fba4f",
                "pkg:golang/github.com/alecthomas/chroma@v2.20.0?package-id=a3e1e244ff6cb9b5#v2",
                "pkg:golang/github.com/antlr4-go/antlr@v4.13.0?package-id=f70a9dd28aa41dc6#v4",
                "pkg:golang/github.com/aryann/difflib@v0.0.0-20210328193216-ff5ff6dc229b?package-id=6
                "pkg:golang/github.com/beorn7/perks@v1.0.1?package-id=db0114c045e8bfd1",
                "pkg:golang/github.com/burntsushi/toml@v1.5.0?package-id=b5ec7e2ef272bb0f",
                "pkg:golang/github.com/caddyserver/caddy@v2.10.2?package-id=dfdb60ed9499f46f#v2",
                "pkg:golang/github.com/caddyserver/certmagic@v0.24.0?package-id=365d2922212c0063",
                "pkg:golang/github.com/caddyserver/zerossl@v0.1.3?package-id=2e4132e663123b07",
                "pkg:golang/github.com/ccoveille/go-safecast@v1.6.1?package-id=a9adf6573c187b1d",
                "pkg:golang/github.com/cenkalti/backoff@v5.0.2?package-id=d236512f8ebb352c#v5",
                "pkg:golang/github.com/cespare/xxhash@v2.3.0?package-id=a751bc616b70401e#v2",
                "pkg:golang/github.com/chzyer/readline@v1.5.1?package-id=6eb909a3925d99d9",
                "pkg:golang/github.com/cloudflare/circl@v1.6.1?package-id=ba3e9059d0e4bcc3",
                "pkg:golang/github.com/coreos/go-oidc@v3.14.1?package-id=1958899417a8db1f#v3",
                "pkg:golang/github.com/cpuguy83/go-md2man@v2.0.7?package-id=29125c33fc044e55#v2",
                "pkg:golang/github.com/dlclark/regexp2@v1.11.5?package-id=ea7ce6205937d1c3",
                "pkg:golang/github.com/dustin/go-humanize@v1.0.1?package-id=d2b74d2118cb919e",
                "pkg:golang/github.com/felixge/httpsnoop@v1.0.4?package-id=87bd9e3d2c5af3b3",
                "pkg:golang/github.com/francoispqt/gojay@v1.2.13?package-id=f37ad7b9506deb03",
                "pkg:golang/github.com/fxamacker/cbor@v2.8.0?package-id=655f71068fe07e33#v2",
                "pkg:golang/github.com/go-chi/chi@v5.2.2?package-id=64cb047ec0bafb6f#v5",
                "pkg:golang/github.com/go-jose/go-jose@v3.0.4?package-id=06d3b7161debc228#v3",
                "pkg:golang/github.com/go-jose/go-jose@v4.0.5?package-id=8a2057af81eeff66#v4",
                "pkg:golang/github.com/go-logr/logr@v1.4.3?package-id=65b99f9f57d4b904",
                "pkg:golang/github.com/go-logr/stdr@v1.2.2?package-id=4931c5d8a486aca9",
                "pkg:golang/github.com/google/cel-go@v0.26.0?package-id=b2a53f0d3df22f76",
                "pkg:golang/github.com/google/certificate-transparency-go@v1.1.8-0.20240110162603-74a
```


https://github.com/caddyserver/caddy

Что и как ещё можно сделать?

Подходы

- Инструментировать утилиты для получения SBOM'ов под каждый ЯП/окружение/и т.д.
- Собирать всю информацию во время сборки

```
[unstable]
sbom = true

[build]
sbom = true
```

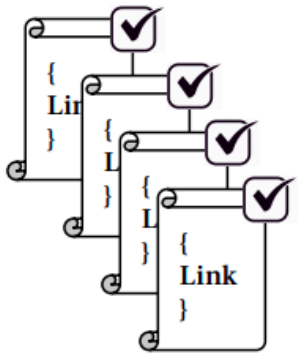
```
$ cargo sbom
```

```
--output-format <OUTPUT_FORMAT>
          The SBOM output format.
          [default: spdx_json_2_3]
          [possible values: spdx_json_2_3,
          cyclone_dx_json_1_4,
          cyclone_dx_json_1_6]
```

https://crates.io/crates/cargo-sbom

SBOMit (1)

SBOMit = SBOM + in-toto (A kitchen camera 📸 records every step of baking the pie.)



in-toto Attestations

Collect .link Files
Gather all in-toto attestations from your build pipeline

Extract File Artifacts
Parse materials and products with SHA256

Classify Crate & Build Output
Use path patterns to infer source → build relationships

Create SPDX Package and File Objects
Annotate components and artifacts with SPDX metadata

Link Them with SPDX Relationships
Define how files relate to pakages and the build doc

Assemble SPDX SBOM
Build JSON object following SPDX 2.3 format

Verifiable SBOM

SBOMit - .link (2)

```
{
 "signatures": [],
 "signed": {
  "_type": "link",
  . . .
  "command": [
   "tar",
   "xzf",
   "demo-project.tar.gz"
  ],
  "environment": {},
  "materials": {
   ".keep": {
    "sha256": "e3b0...b855"
   },
   "alice.pub": {
    "sha256": "54d...695"
   },
   "demo-project.tar.gz": {
    "sha256": "c5f...463"
   },
   "root.layout": {
    "sha256": "413...2f3"
   }
. . .
```

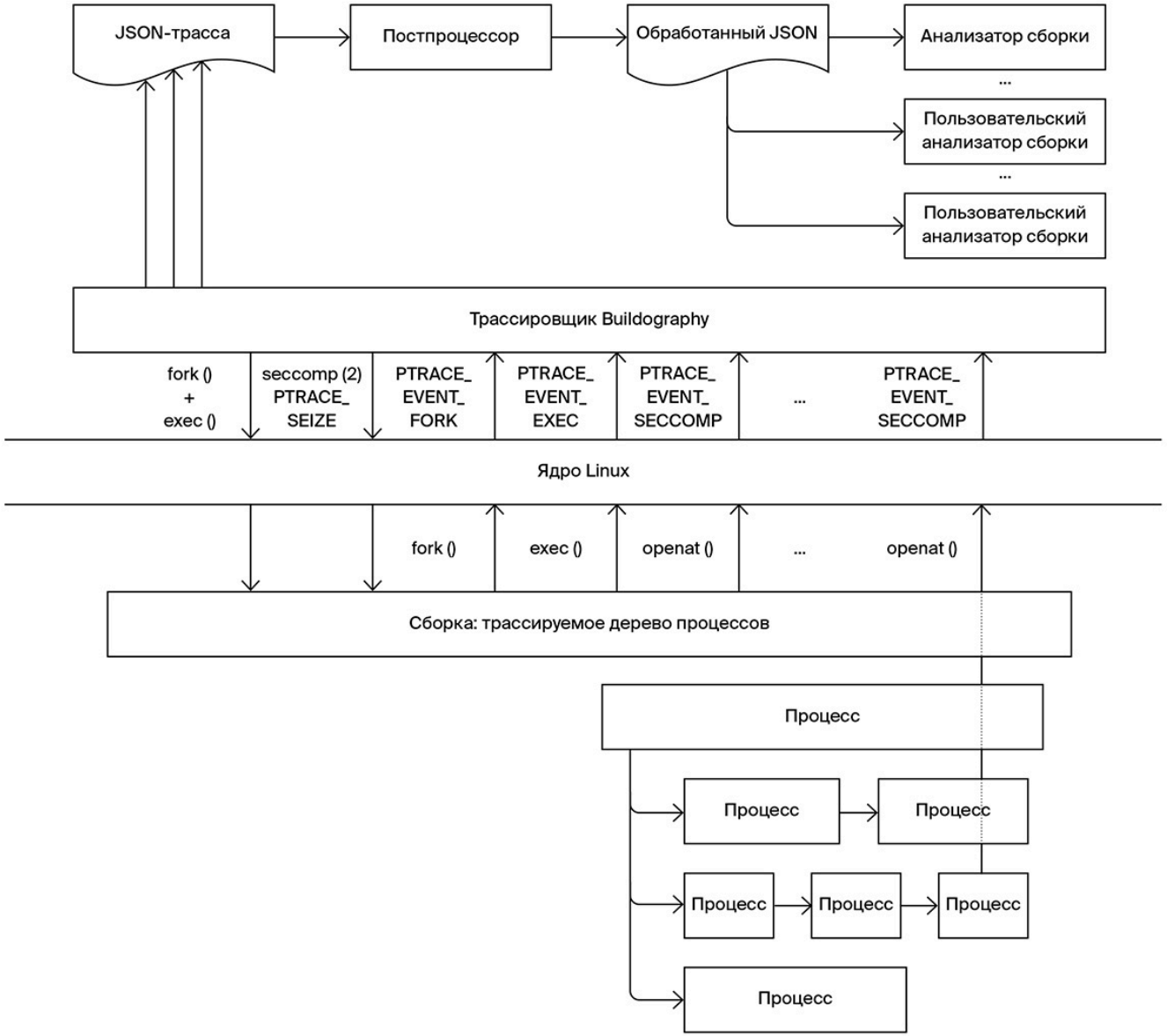
```
  },
  "name": "untar",
  "products": {
   ".keep": {
    "sha256": "e3b...855"
   },
   "alice.pub": {
    "sha256": "54d...695"
   },
   "demo-project.tar.gz": {
    "sha256": "c5f...463"
   },
   "demo-project/foo.py": {
    "sha256": "c2c...31b"
   },
   "root.layout": {
    "sha256": "413...2f3"
   }
  }
 }
}
```

SBOMit (3)

	A Attacker compromises VCS and inserts malicious code	B Testing failed or testing process is missing	C Attacker hacks the compiler	D SBOMs miss information	E SBOMs are forged
SBOMs	❌	❌	❌	❌	❌
Signed SBOMs	❌	❌	❌	❌	✅
SBOMit Phase 1	✅* *Depending on in-toto setup	✅* *Depending on in-toto setup	❌	❌	✅
SBOMit Phase 2	✅	✅	❌	✅	✅

Buildography

- Рабочая директория
- Аргументы командной строки порождаемых процессов
- Данные о файлах, которые открывались этими процессами
- Информация, идентифицирующая вызываемые программы



sbom-checker (регуляторика in action)

- sbom-updater
  - добавление полей ПА и ФБ
  - указание источника зависимости
- sbom-checker
  - проверка репозитория зависимости
  - проверка корректности SBOM'а зависимостей
  - проверка корректности SBOM'а образов контейнеров


```
{
  "bom-ref": "pkg:deb/debian/netbase@6.4?arch=all&distro=debian-12.11",
  "type": "library",
  "supplier": {
    "name": "Marco d'Itri <md@linux.it>"
  },
  "name": "netbase",
  "version": "6.4",
  "licenses": [
    {
      "license": {
        "id": "GPL-2.0-only"
      }
    }
  ],
  "purl": "pkg:deb/debian/netbase@6.4?arch=all&distro=debian-12.11",
  "properties": [
    {…
    },
    {…
    },
    {…
    },
    {…
    },
    {…
    },
    {…
    },
    {
      "name": "GOST:attack_surface",
      "value": "no"
    },
    {
      "name": "GOST:security_function",
      "value": "no"
    }
  ]
}
```

Вместо заключения

- SBOM это непросто
- Избавляемся от неизвестности всеми доступными способами
- Доверяй, но проверяй
- Подходы на базе "статики" позволяют защититься только от легитимного пользователя, но не являются никакой помехой для вредоносного
- Делая статику не забывайте о динамике

LUNTRY

Спасибо за внимание!

site: luntry.ru

tg: @rusdacent

channel: @tech_b0lt_Genona

Вопросы?