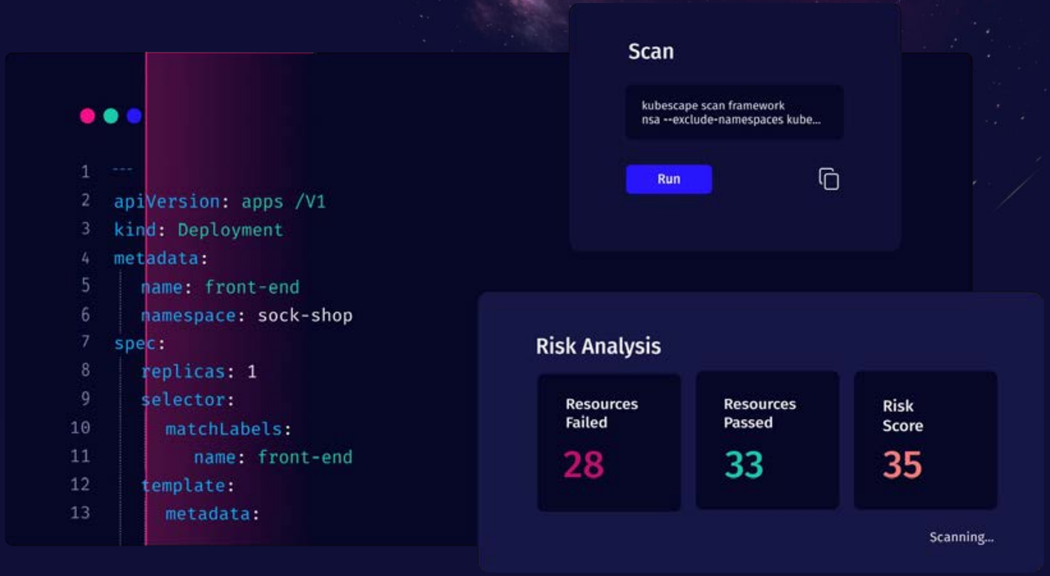
ARMO

Unpacking Open Source Security in Public Repos & Registries



```
---
apiVersion: apps /V1
kind: Deployment
metadata:
  name: front-end
  namespace: sock-shop
spec:
  replicas: 1
  selector:
    matchLabels:
      name: front-end
  template:
    metadata:
```

```
kubescape scan framework
nsa --exclude-namespaces kube...
```

Risk Analysis

Resources Failed 28

Resources Passed 33

Risk Score 35

Yiscah Levy Silas | SW Engineer

/Starting point_

Kubescape is here to tell you what’s wrong

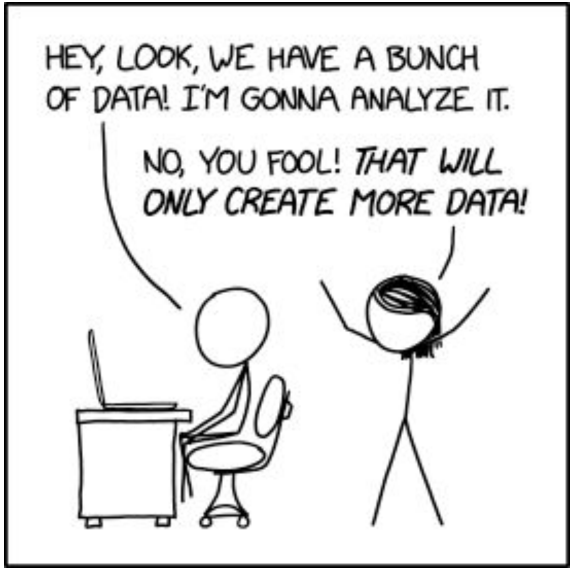
- In your clusters
- In your container registries
- With YAML/Helm charts in your Git repositories and CI processes

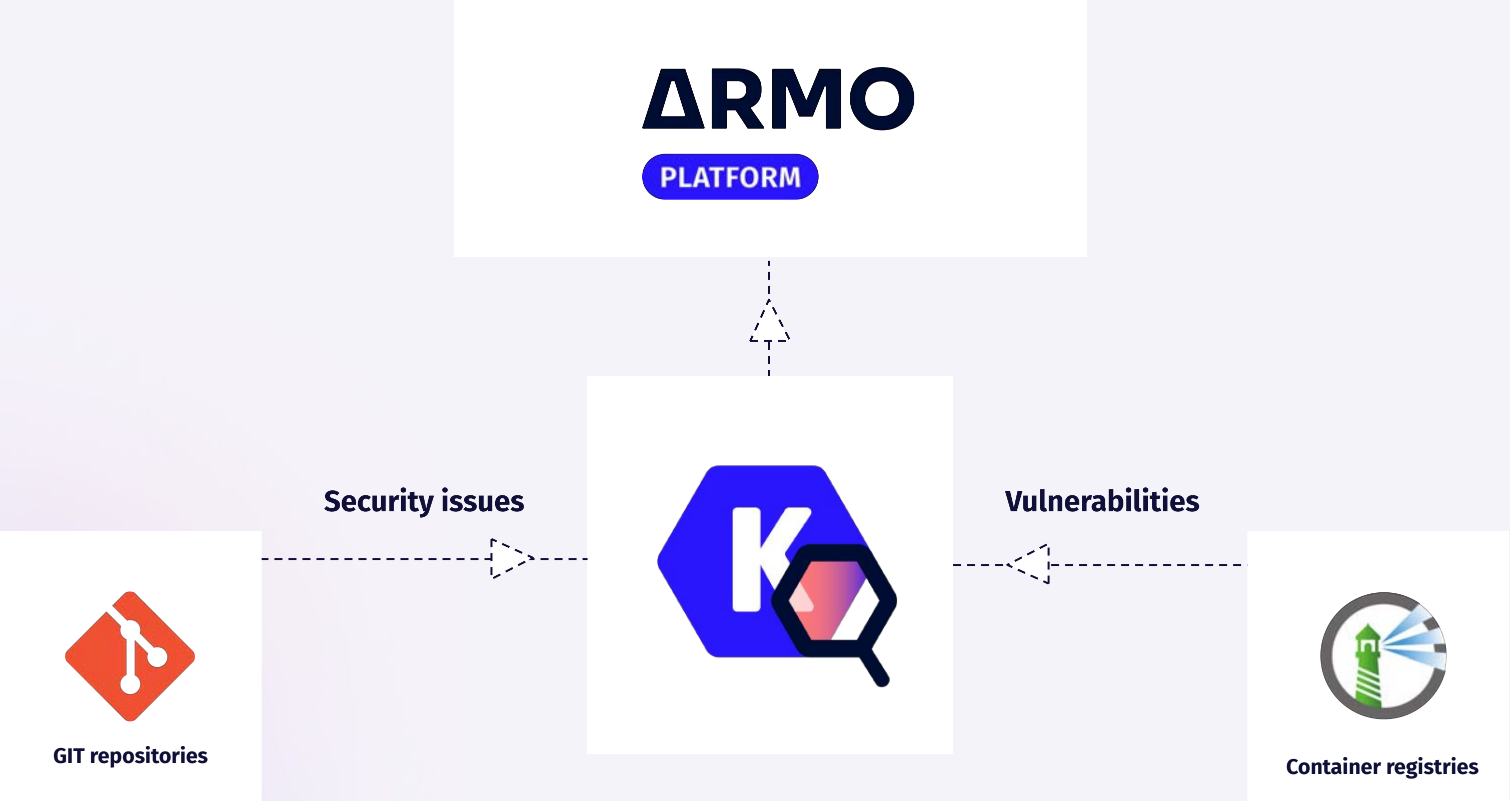
More important to tell how to fix and prioritization of the issues

ARMO Platform is a cloud service (beyond other things) storing KS results

Kubescape

CLOUD NATIVE SANDBOX


HEY, LOOK, WE HAVE A BUNCH OF DATA! I'M GONNA ANALYZE IT.
NO, YOU FOOL! *THAT WILL ONLY CREATE MORE DATA!*



ARMO

PLATFORM

Security issues

Vulnerabilities

GIT repositories

Container registries

179

Registries

43,539

Images

1,914

Repositories

164,887

Files scanned

/Container image scans

Comparing the whole sample to the sub-sample of graduated projects

Reviewing the distribution of severities

Reviewing top CVEs in both

Relevancy

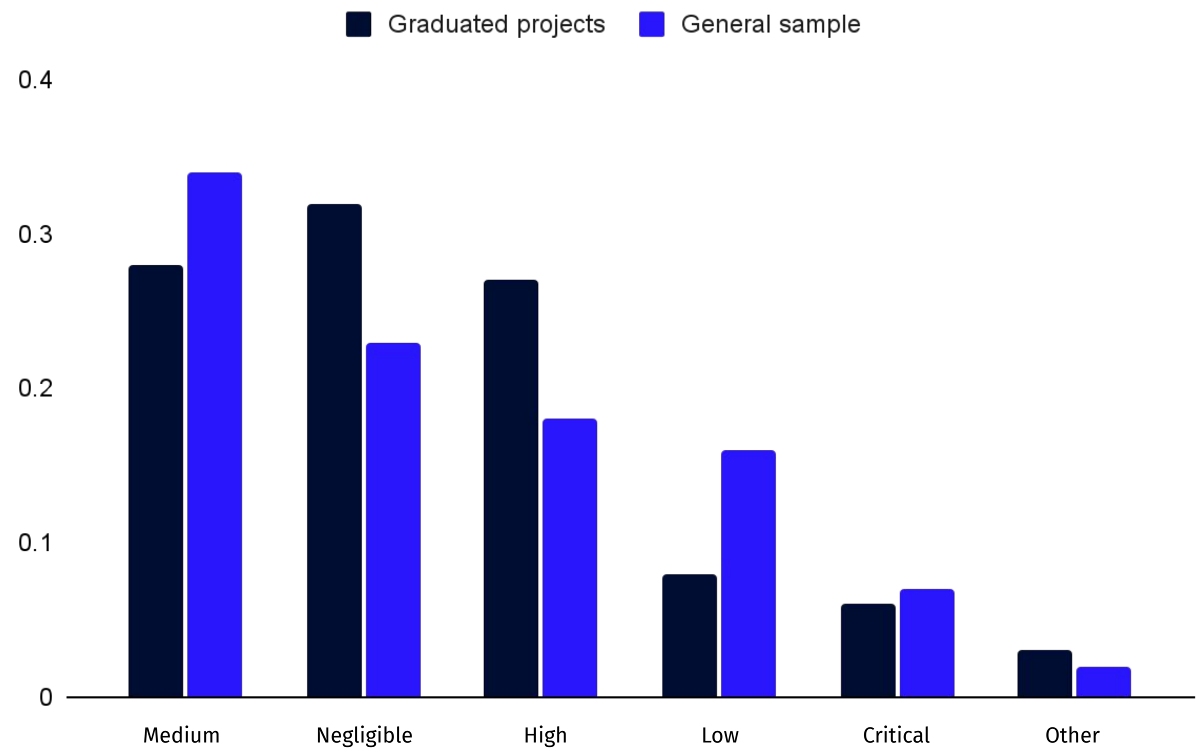
/Image repos with most scans in the general sample

Top count of repo	# workload image scans
quay.io/argoproj/argocd	19,426
docker.io/bitnami/redis	13,308
quay.io/argoproj/argoexec	11,427
quay.io/prometheus-operator/prometheus-config-reloader	11,275
quay.io/kiwigrid/k8s-sidecar	6,581
quay.io/prometheus/prometheus	6,390
docker.io/bitnami/mongodb	6,312
quay.io/prometheus/node-exporter	5,569
gcr.io/datadoghq/agent	5,404

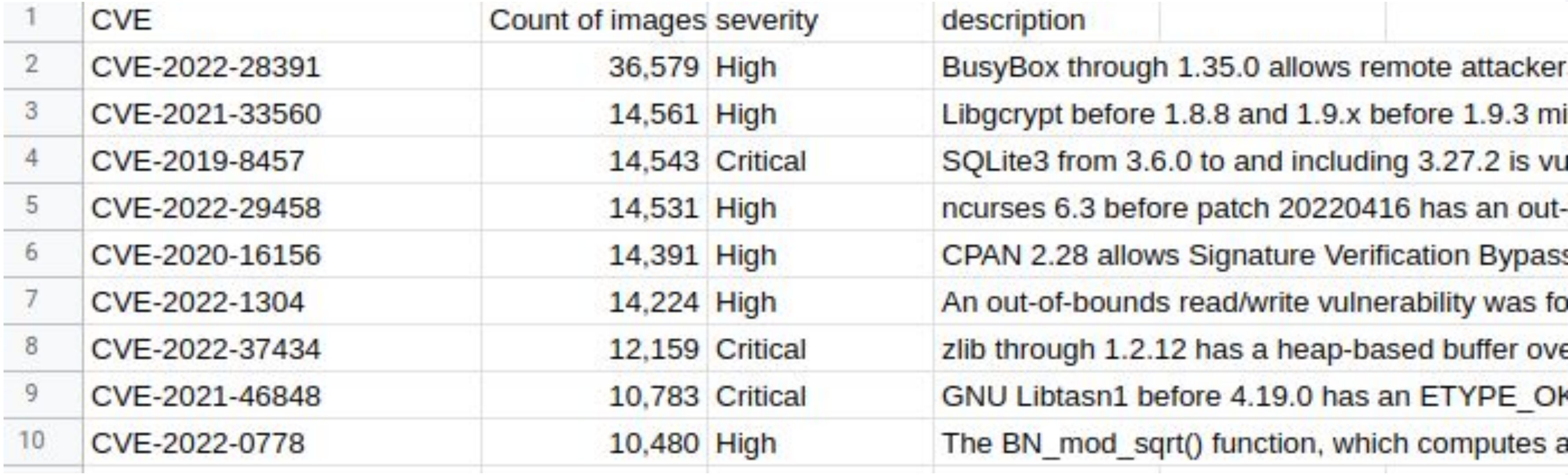
/Image tags with most scans in the graduated sample

Top count of repo	# workload image scans
quay.io/argoproj/argocd	19,426
quay.io/argoproj/argoexec	11,427
quay.io/prometheus-operator/prometheus-config-reloader	11,275
quay.io/prometheus/prometheus	6,390
quay.io/prometheus/node-exporter	5,569
quay.io/prometheus/alertmanager	4,172
quay.io/prometheus-operator/prometheus-operator	4,088
registry.k8s.io/kube-proxy	3,530
registry.k8s.io/kube-state-metrics/kube-state-metrics	3,039

/Comparison_


■ Graduated projects
■ General sample
0.4
0.3
0.2
0.1
0
Medium
Negligible
High
Low
Critical
Other

/TOP vulnerabilities in general population_



1	CVE	Count of images	severity	description
2	CVE-2022-28391	36,579	High	BusyBox through 1.35.0 allows remote attacker
3	CVE-2021-33560	14,561	High	Libgcrypt before 1.8.8 and 1.9.x before 1.9.3 mi
4	CVE-2019-8457	14,543	Critical	SQLite3 from 3.6.0 to and including 3.27.2 is vu
5	CVE-2022-29458	14,531	High	ncurses 6.3 before patch 20220416 has an out-
6	CVE-2020-16156	14,391	High	CPAN 2.28 allows Signature Verification Bypass
7	CVE-2022-1304	14,224	High	An out-of-bounds read/write vulnerability was fo
8	CVE-2022-37434	12,159	Critical	zlib through 1.2.12 has a heap-based buffer ove
9	CVE-2021-46848	10,783	Critical	GNU Libtasn1 before 4.19.0 has an ETYPE_OK
10	CVE-2022-0778	10,480	High	The BN_mod_sqrt() function, which computes a

/CVE-2022-28391

CVSS vector: `AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H`

Description:
BusyBox through 1.35.0 allows remote attackers to execute arbitrary code if netstat is used to print a DNS PTR record's value to a VT compatible terminal. Alternatively, the attacker could choose to change the terminal's colors.

Cloud native environment:
If someone is running netstat in a Pod from a terminal while the attack controls the DNS entry the terminal is prone to the attack. Not a common scenario.

/CVE-2021-33560

CVSS vector: `AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N`

Description:
Libgcrypt before 1.8.8 and 1.9.x before 1.9.3 mishandles ElGamal encryption because it lacks exponent blinding to address a side-channel attack against mpi_powm, and the window size is not chosen appropriately. This, for example, affects use of ElGamal in OpenPGP.

Cloud native environment:
Libgcrypt is around in many images for GPG signature verification of APT/YUM packages. It is mostly not in use during deployment + uo private key in the image

/CVE-2019-8457

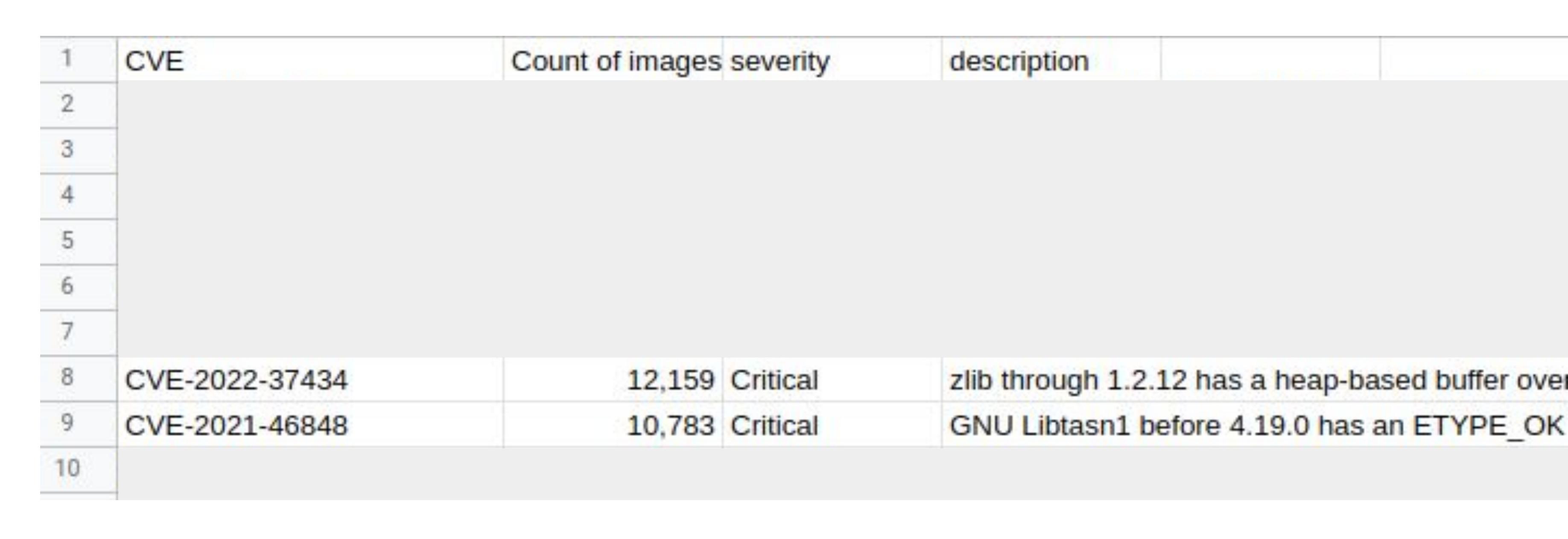
CVSS vector: `AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H`

Description:
SQLite3 from 3.6.0 to and including 3.27.2 is vulnerable to heap out-of-bound read in the rtreenode() function when handling invalid rtree tables.

Cloud native environment:
If the attacker can inject arbitrary SQL statements then the attacker can get arbitrary code execution. SQLite is part of Centos/RH base images.

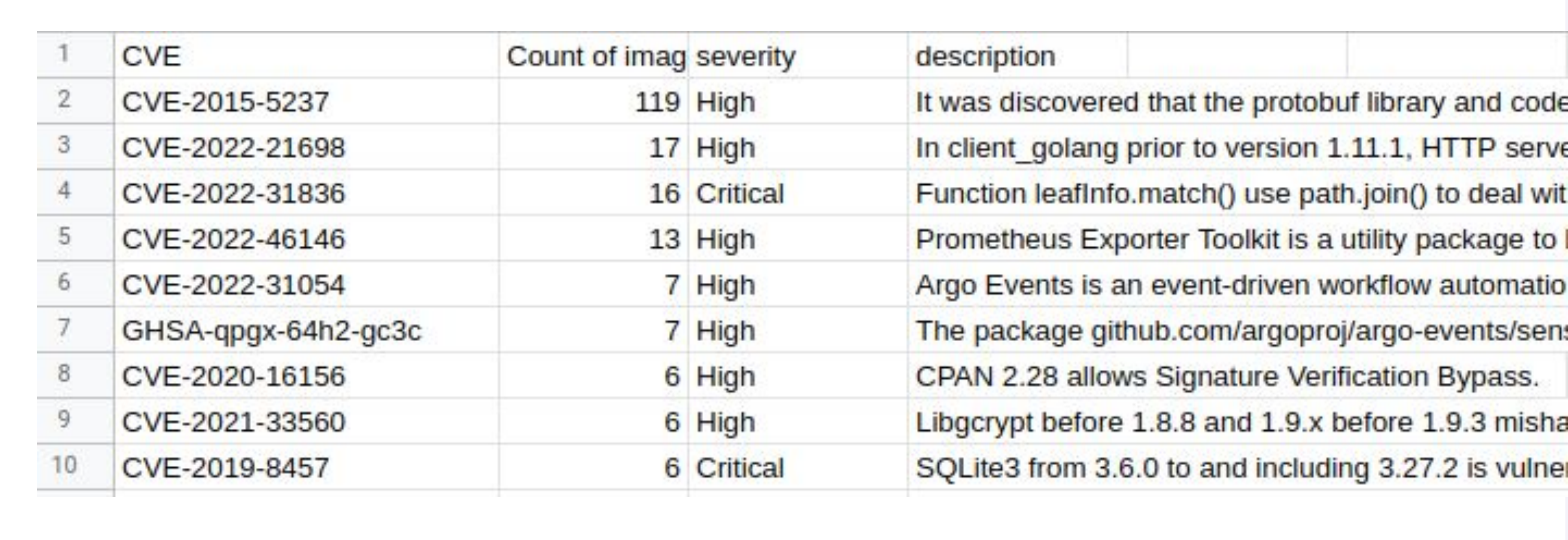
/Opinion: these are the vulnerabilities has some probability above 0.1* to be exploited

*gut feeling :-/



1	CVE	Count of images	severity	description		
2						
3						
4						
5						
6						
7						
8	CVE-2022-37434	12,159	Critical	zlib through 1.2.12 has a heap-based buffer over		
9	CVE-2021-46848	10,783	Critical	GNU Libtasn1 before 4.19.0 has an ETYPE_OK		
10						

/TOP vulnerabilities in graduated projects



1	CVE	Count of imag	severity	description
2	CVE-2015-5237	119	High	It was discovered that the protobuf library and code
3	CVE-2022-21698	17	High	In client_golang prior to version 1.11.1, HTTP serve
4	CVE-2022-31836	16	Critical	Function leafInfo.match() use path.join() to deal wit
5	CVE-2022-46146	13	High	Prometheus Exporter Toolkit is a utility package to
6	CVE-2022-31054	7	High	Argo Events is an event-driven workflow automatio
7	GHSA-qpgx-64h2-gc3c	7	High	The package github.com/argoproj/argo-events/sens
8	CVE-2020-16156	6	High	CPAN 2.28 allows Signature Verification Bypass.
9	CVE-2021-33560	6	High	Libgcrypt before 1.8.8 and 1.9.x before 1.9.3 misha
10	CVE-2019-8457	6	Critical	SQLite3 from 3.6.0 to and including 3.27.2 is vulne

/CVE-2015-5237

CVSS vector: `AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H`

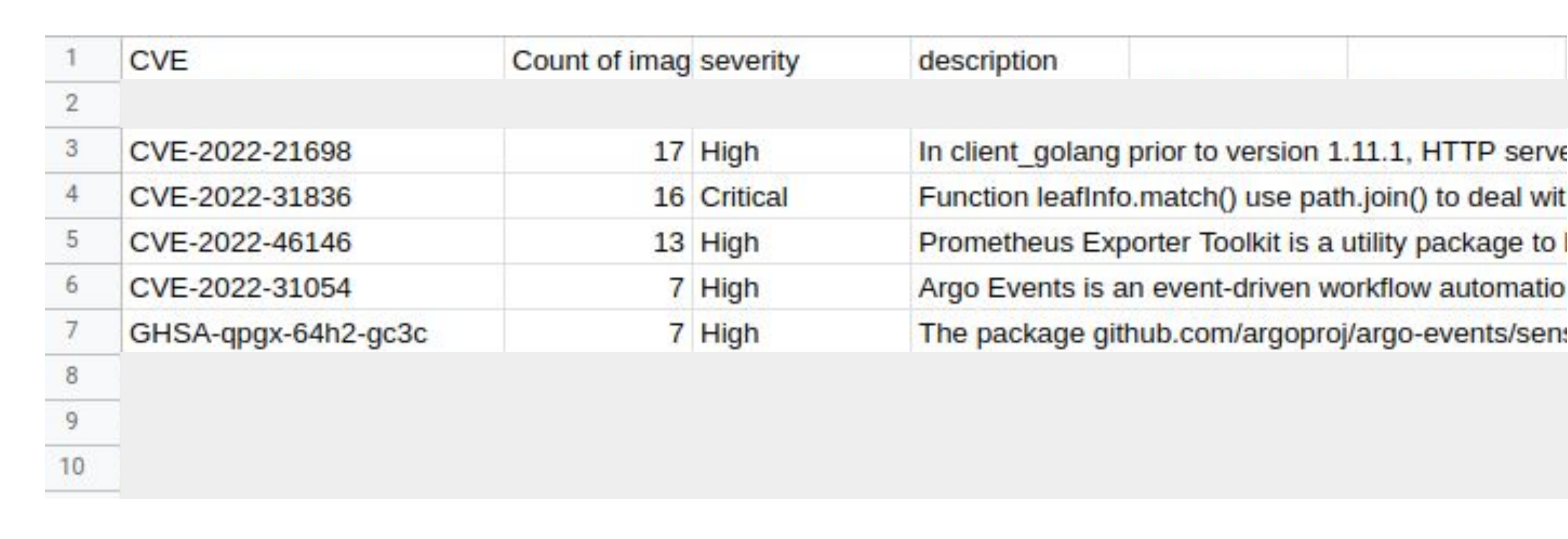
Description:
protobuf allows remote authenticated attackers to cause a heap-based buffer overflow

Cloud native environment:
It is indeed a vulnerability in protobuf C/C++ package. But not in the Golang package!

https://github.com/anchore/grype/issues/558

/Opinion: **these are the vulnerabilities has some probability above 0.1* to be exploited**

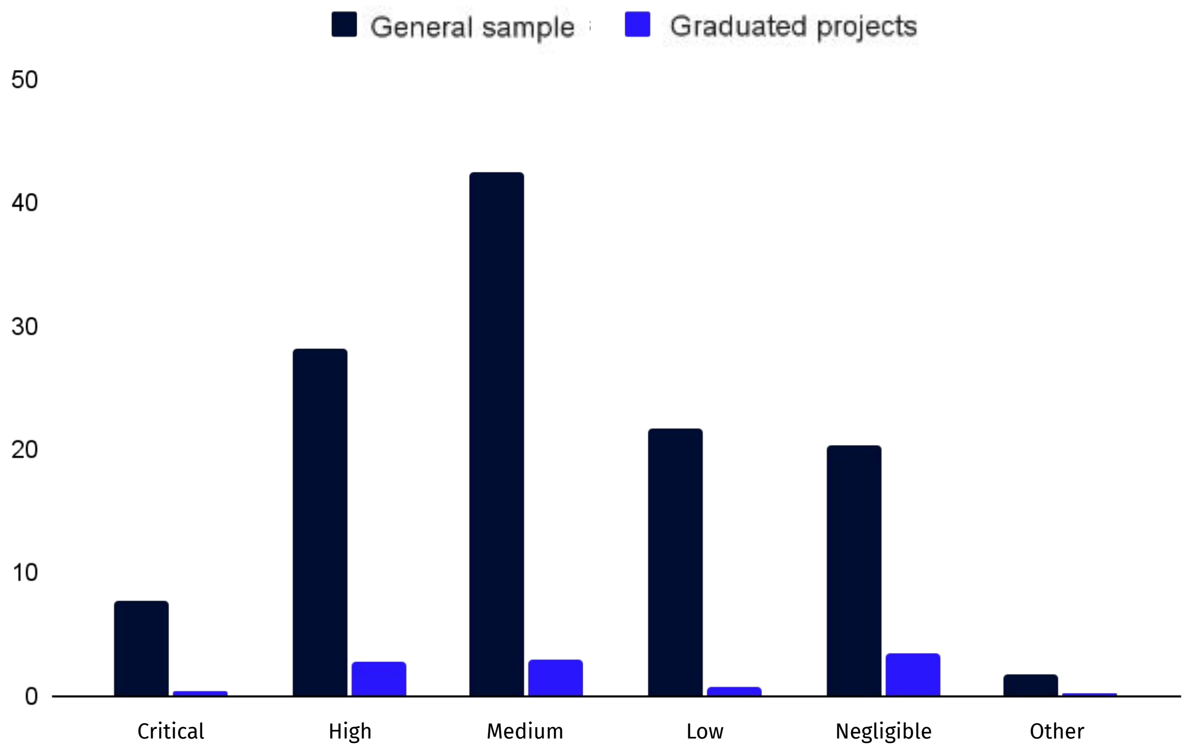
*gut feeling :-/



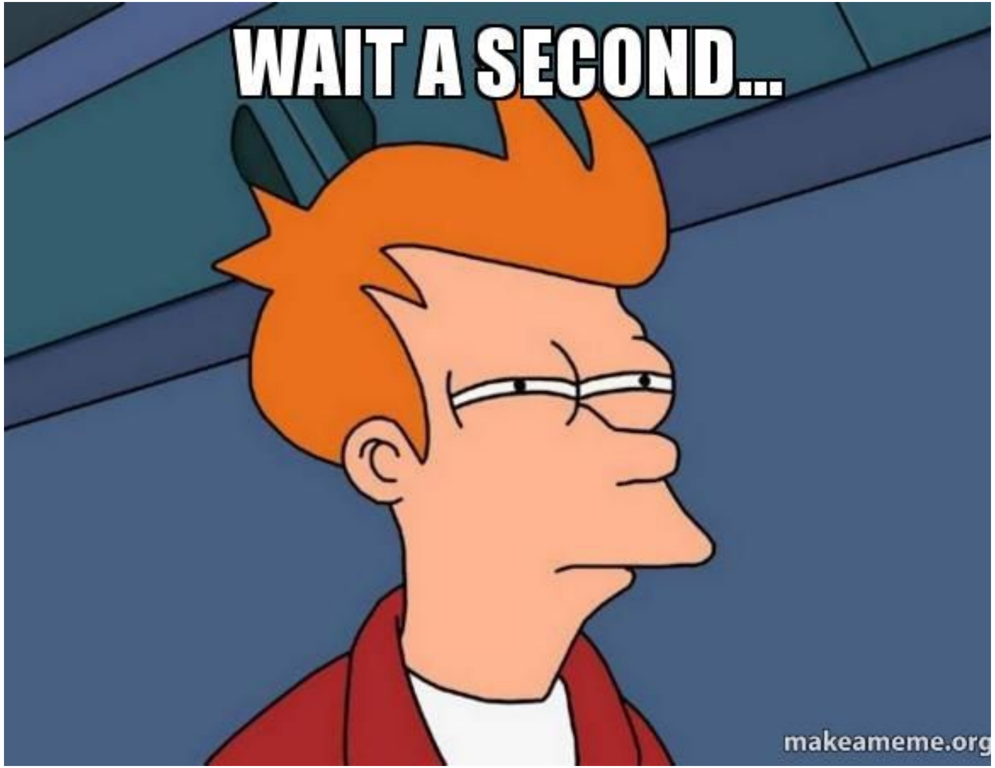
1	CVE	Count of imag	severity	description		
2						
3	CVE-2022-21698	17	High	In client_golang prior to version 1.11.1, HTTP serve		
4	CVE-2022-31836	16	Critical	Function leafInfo.match() use path.join() to deal wit		
5	CVE-2022-46146	13	High	Prometheus Exporter Toolkit is a utility package to		
6	CVE-2022-31054	7	High	Argo Events is an event-driven workflow automatio		
7	GHSA-qpgx-64h2-gc3c	7	High	The package github.com/argoproj/argo-events/sen		
8						
9						
10						

/Looking only at filtered results_

Average vulnerability count per severity



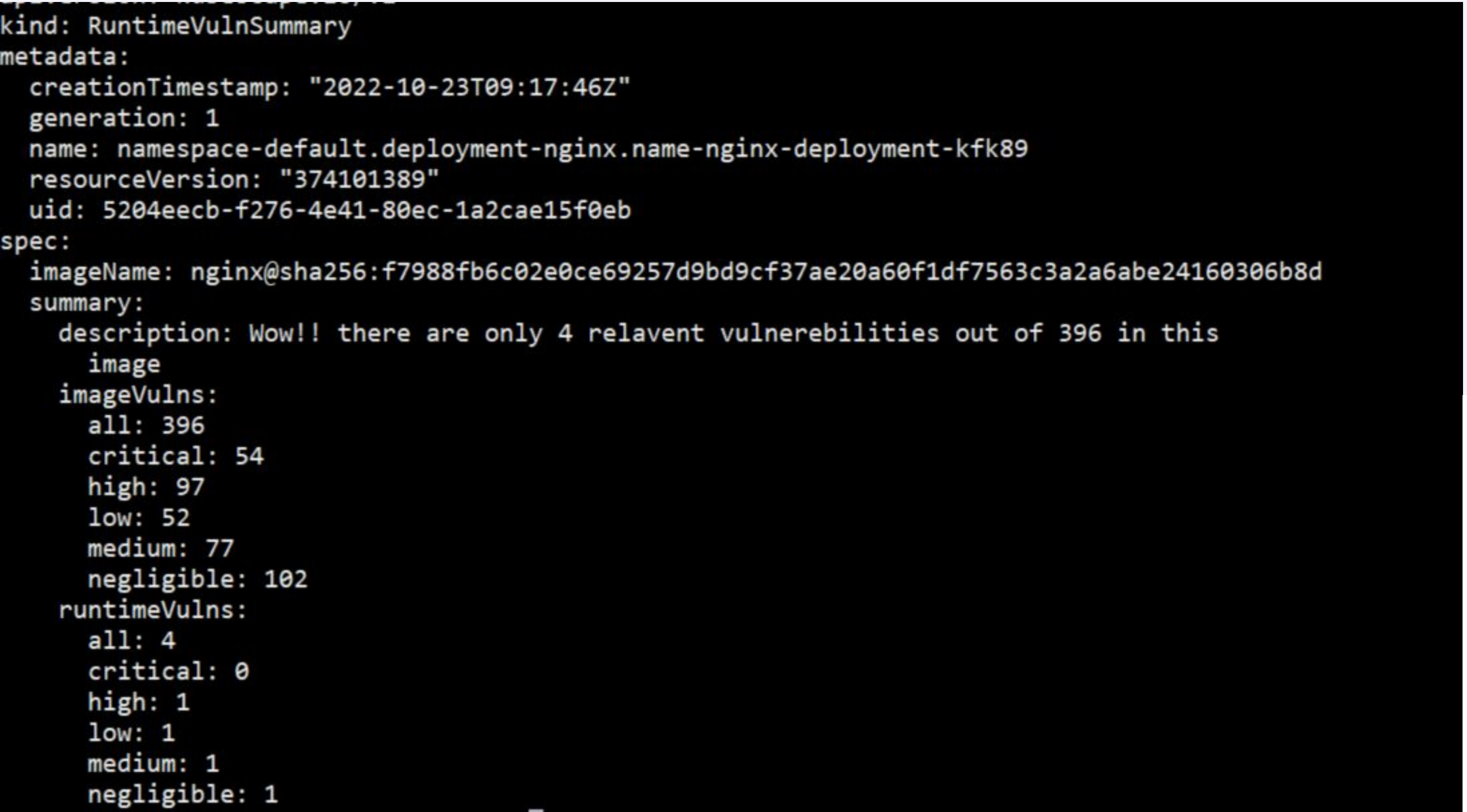
CNCF IS AWESOME
imgflip.com


WAIT A SECOND...
makeameme.org

/Image vulnerability relevancy

Vulnerability in image ≠ Workload exploit

/Kubescape sneeffer results

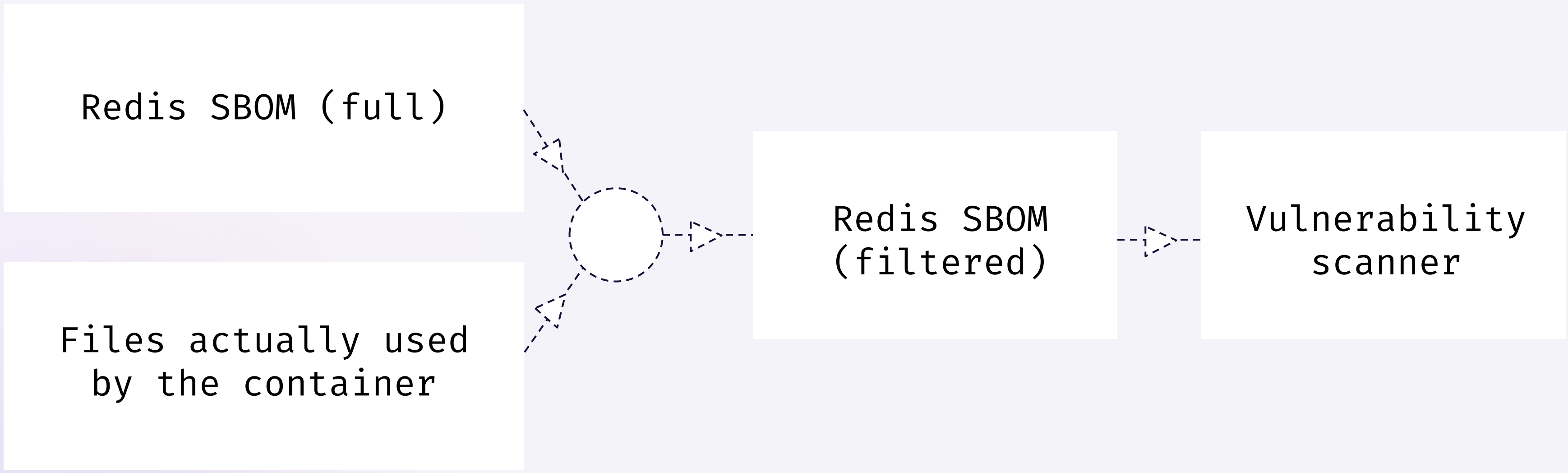


```
kind: RuntimeVulnSummary
metadata:
  creationTimestamp: "2022-10-23T09:17:46Z"
  generation: 1
  name: namespace-default.deployment-nginx.name-nginx-deployment-kfk89
  resourceVersion: "374101389"
  uid: 5204eecb-f276-4e41-80ec-1a2cae15f0eb
spec:
  imageName: nginx@sha256:f7988fb6c02e0ce69257d9bd9cf37ae20a60f1df7563c3a2a6abe24160306b8d
  summary:
    description: Wow!! there are only 4 relavent vulnerebilities out of 396 in this
      image
    imageVulns:
      all: 396
      critical: 54
      high: 97
      low: 52
      medium: 77
      negligible: 102
    runtimeVulns:
      all: 4
      critical: 0
      high: 1
      low: 1
      medium: 1
      negligible: 1
```

Kubescape

/Kubescape Sneeffer

https://github.com/kubescape/sneeffer



Redis SBOM (full)

Files actually used by the container

Redis SBOM (filtered)

Vulnerability scanner

Scan image

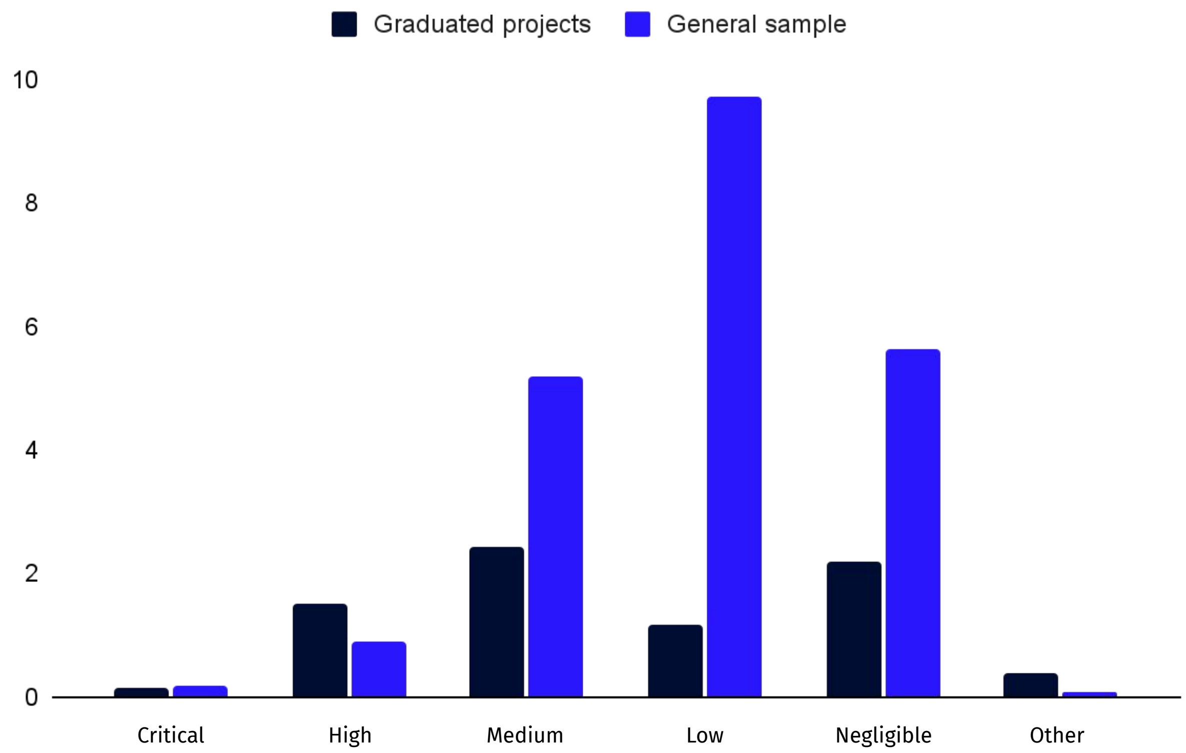
eBPF

Compare against SBOM

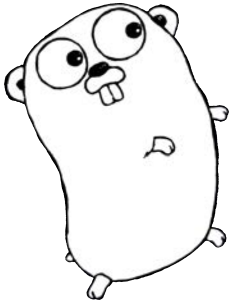
Feed to Vulnerability scanner

/Looking only at filtered results_

Average relevant vulnerability count per severity



/Explaining the numbers



docker

AICPA SOC

aicpa.org/soc4so

SOC for Service Organizations | Service Organizations

HIPAA

Health Insurance Portability and Accountability Act

CLOUD NATIVE COMPUTING FOUNDATION

ISO

International Organization for Standardization

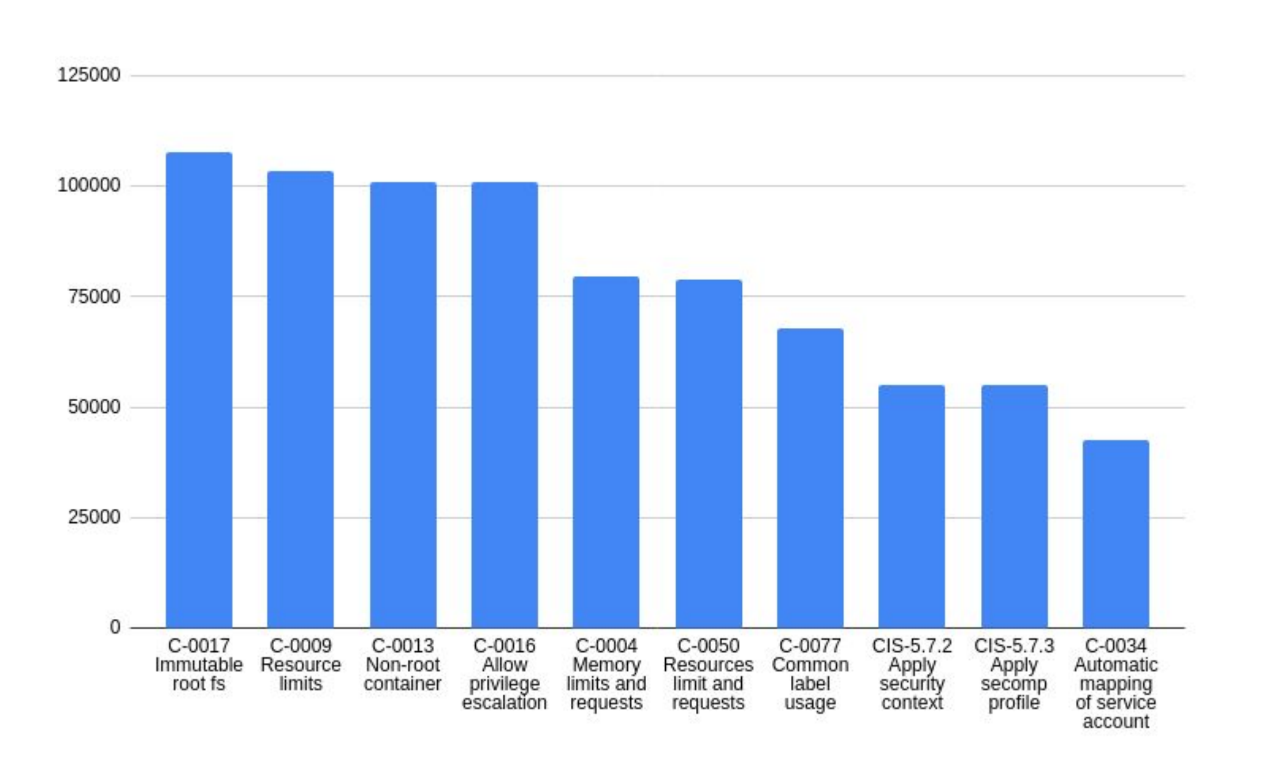
/Git repository scans_

Comparing the whole sample to the sub-sample of graduated projects

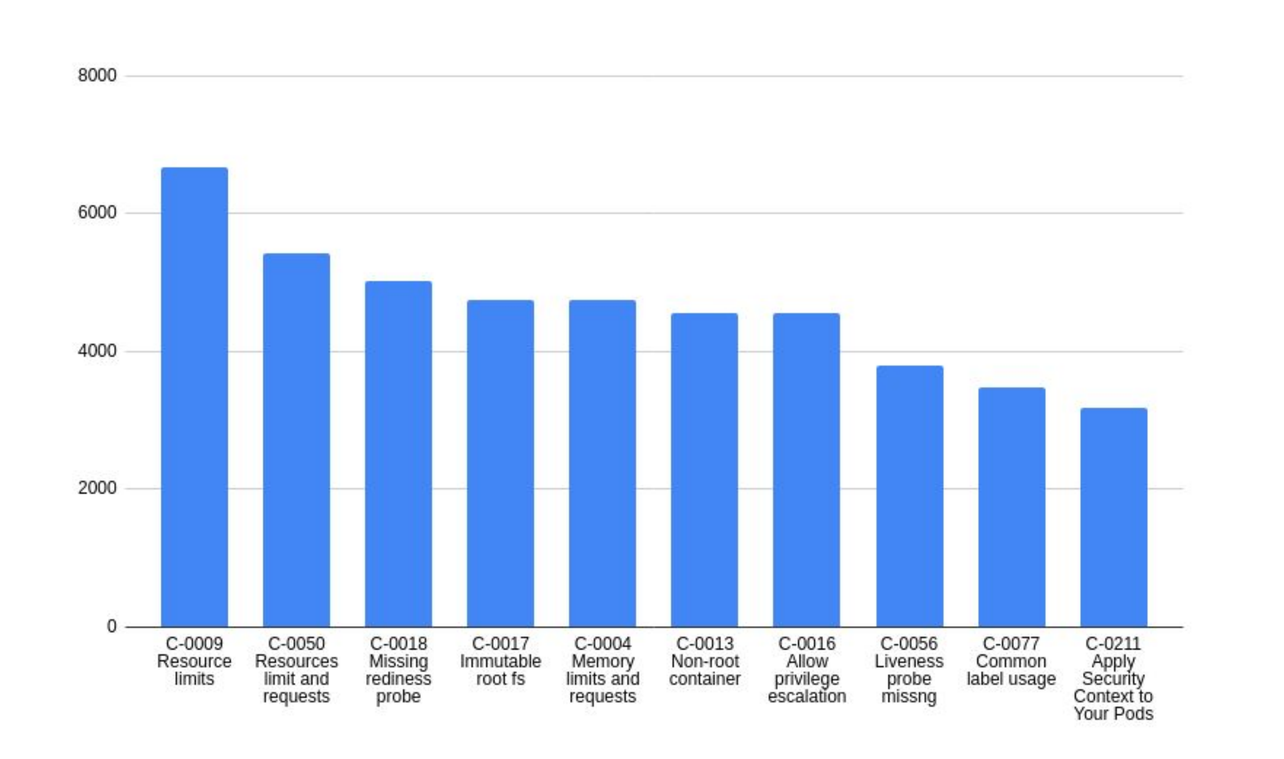
Reviewing the distribution of controls

Evaluating the the number of failed controls ratio

/Most failed in general population_



/Most failed among graduated projects_



/Percent of controls failing_

Control failure ratio = Failed controls : all relevant controls (per resource)

35%

Graduated projects sample

38%

General sample

/Closing thoughts_

Vulnerabilities

Hard to clearly say that CNCF Graduated projects are less vulnerable

Vulnerability scan results are like have million problems

Generally, newer technologies and languages covering low some hanging security fruits

Misconfigurations

Graduated projects has a slightly better security posture

Many still prone to simple issues

Thank you_

ARMO