
SBOM to VEX - Discovering What's in the Box and How Badly it Can Hurt You
</ From the creators of the fastest growing open source kubernetes security platform
Kubescape
ARMO

/whoami

Ben Hirschberg

Co-founder & CTO @ARMO

Kubescape maintainer

Whitehat in the past (unofficially still ;-)

Fluent in Hebrew, Hungarian, C, ASM and Go

Contributor in CNCF + organizer of CNCF Jerusalem

Father of 4 <3

@Ben Hirschberg

Ben-hirschberg

@slashben81

github.com/slashben

/man_kubescape

CLOUD NATIVE SANDBOX

Kubernetes security

Scanning and monitoring

From dev to production

Misconfigurations and vulnerability finding

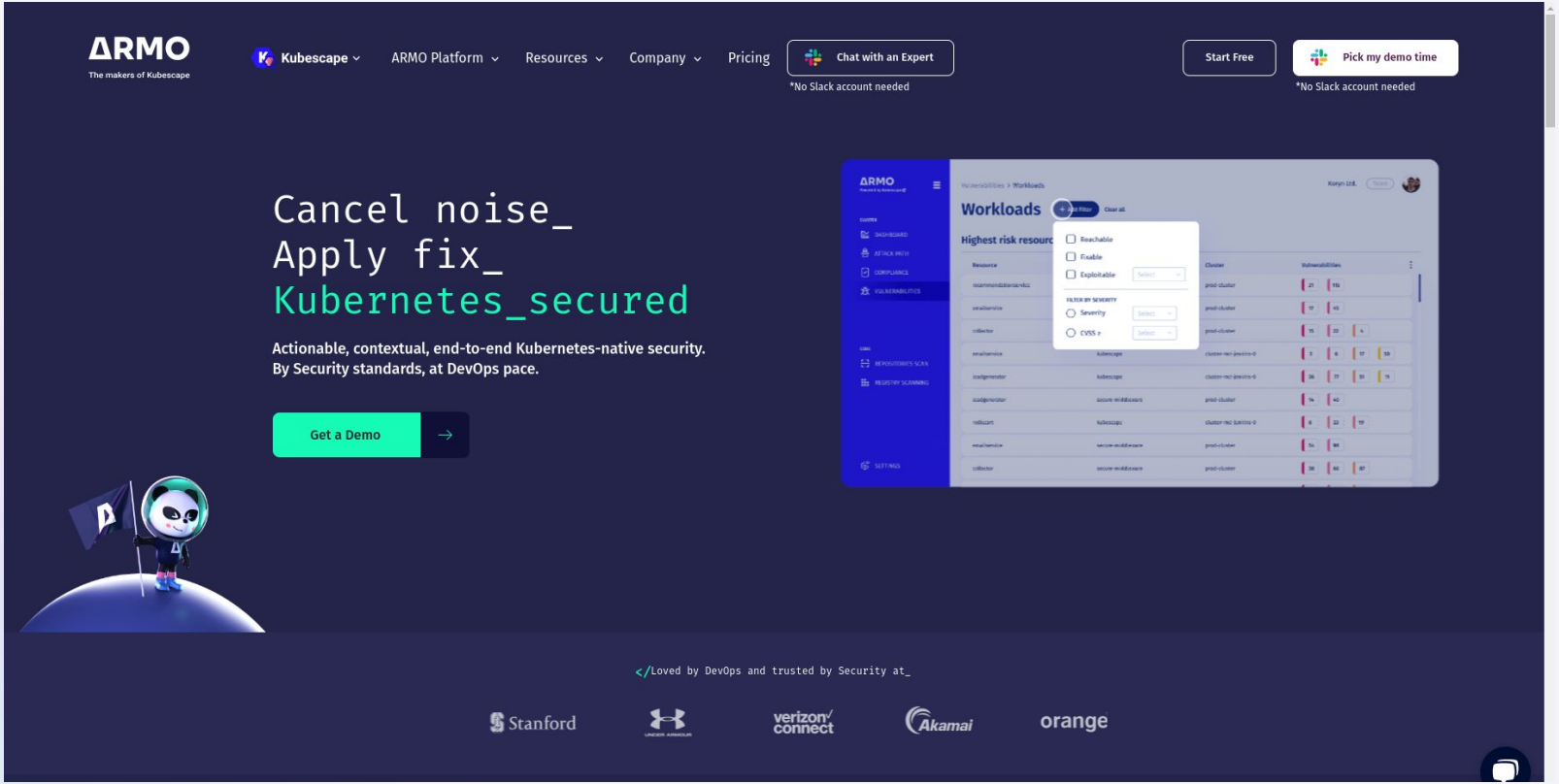
Operator and CLI

Kubescape

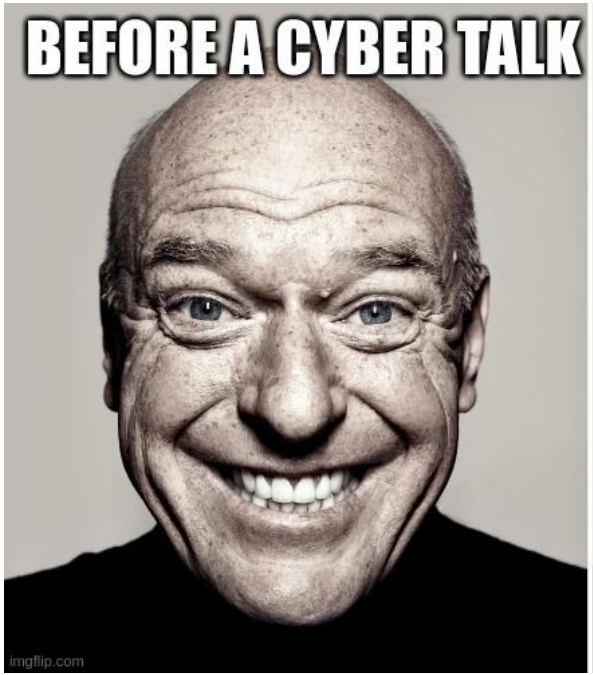
@Kubescape

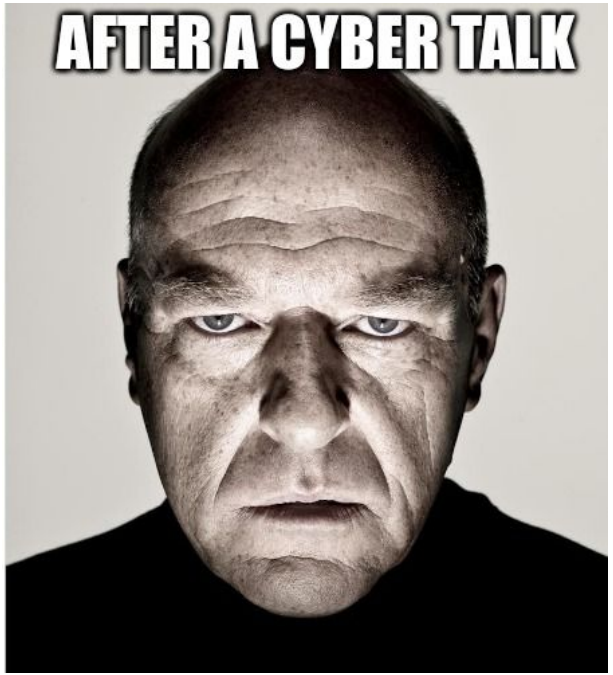
github.com/kubescape/kubescape

/man_armo


ARMO
The makers of Kubescape
Kubescape
ARMO Platform
Resources
Company
Pricing
Chat with an Expert
*No Slack account needed
Start Free
Pick my demo time
*No Slack account needed
Cancel noise_
Apply fix_
Kubernetes_secured
Actionable, contextual, end-to-end Kubernetes-native security.
By Security standards, at DevOps pace.
Get a Demo
Workloads
Reachable
Fixable
Exploitable
Severity
CVSS
</Loved by DevOps and trusted by Security at_
Stanford
verizon connect
Akamai
orange

/usually...





/this time

/cat agenda

- **SBOMs and VEXs**
- **Exploitability evaluation with eBPF**
- **Automation of VEX generation with Kubescape**

/sausage-as-a-service

- Modern software contains **80-90%** open source software
- At least **70%** of the containerized workloads are coming from external sources
- **90%** of the first level dependencies have devepncies themselves

/sbom

If the sausage is your software, the SBOM is the list of the ingredients

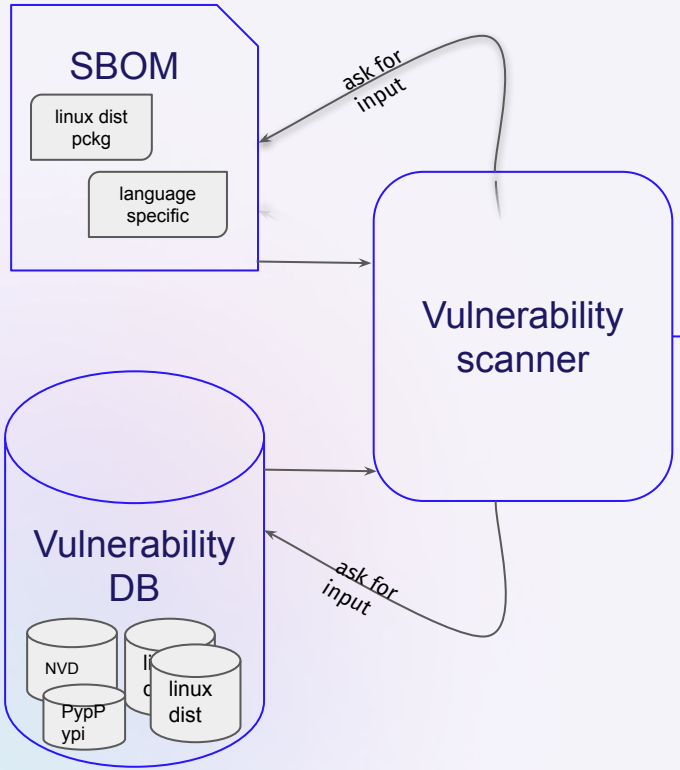
/sbom-use

- Licensing issues in an organization (software composition analysis)
- Security posture/exposure (software posture management)
- Strategic exposures in organizational software

/sbom-generators

- Can only find what they are looking for
- Near 100% true positives

/vulnerabilities-and-sbom



libgssapi-krb5-2	1.18.3-6+deb11u1		deb	CVE-2018-5709	Negligible
libgssapi-krb5-2	1.18.3-6+deb11u1		deb	CVE-2004-0971	Negligible
libk5crypto3	1.18.3-6+deb11u1		deb	CVE-2018-5709	Negligible
libk5crypto3	1.18.3-6+deb11u1		deb	CVE-2004-0971	Negligible
libkrb5-3	1.18.3-6+deb11u1		deb	CVE-2004-0971	Negligible
libkrb5-3	1.18.3-6+deb11u1		deb	CVE-2018-5709	Negligible
libkrb5support0	1.18.3-6+deb11u1		deb	CVE-2004-0971	Negligible
libkrb5support0	1.18.3-6+deb11u1		deb	CVE-2018-5709	Negligible
libmount1	2.36.1-8+deb11u1		deb	CVE-2022-0563	Negligible
libpcre2-8-0	10.36-2	(won't fix)	deb	CVE-2022-1586	Critical
libpcre2-8-0	10.36-2	(won't fix)	deb	CVE-2022-1587	Critical
libpcre3	2:8.39-13		deb	CVE-2017-11164	Negligible
libpcre3	2:8.39-13		deb	CVE-2017-7246	Negligible
libpcre3	2:8.39-13		deb	CVE-2017-7245	Negligible
libpcre3	2:8.39-13		deb	CVE-2017-16231	Negligible
libpcre3	2:8.39-13		deb	CVE-2019-20838	Negligible
libsepol1	3.1-1	(won't fix)	deb	CVE-2021-36085	Low
libsepol1	3.1-1	(won't fix)	deb	CVE-2021-36086	Low
libsepol1	3.1-1	(won't fix)	deb	CVE-2021-36084	Low
libsepol1	3.1-1	(won't fix)	deb	CVE-2021-36087	Low
libsmartcols1	2.36.1-8+deb11u1		deb	CVE-2022-0563	Negligible
libss2	1.46.2-2	(won't fix)	deb	CVE-2022-1304	High
libssl1.1	1.1.1n-0+deb11u3	(won't fix)	deb	CVE-2022-2097	Unknown
libssl1.1	1.1.1n-0+deb11u3		deb	CVE-2010-0928	Negligible
libssl1.1	1.1.1n-0+deb11u3		deb	CVE-2007-6755	Negligible
libsystemd0	247.3-7		deb	CVE-2013-4392	Negligible
libsystemd0	247.3-7		deb	CVE-2020-13529	Negligible
libtinfo6	6.2+20201114-2		deb	CVE-2021-39537	Negligible
libtinfo6	6.2+20201114-2	(won't fix)	deb	CVE-2022-29458	High
libudev1	247.3-7		deb	CVE-2013-4392	Negligible
libudev1	247.3-7		deb	CVE-2020-13529	Negligible
libuuid1	2.36.1-8+deb11u1		deb	CVE-2022-0563	Negligible
login	1:4.8.1-1		deb	CVE-2013-4235	Negligible
login	1:4.8.1-1		deb	CVE-2019-19882	Negligible
login	1:4.8.1-1		deb	CVE-2007-5686	Negligible
logsave	1.46.2-2	(won't fix)	deb	CVE-2022-1304	High
mount	2.36.1-8+deb11u1		deb	CVE-2022-0563	Negligible
ncurses-base	6.2+20201114-2	(won't fix)	deb	CVE-2022-29458	High
ncurses-base	6.2+20201114-2		deb	CVE-2021-39537	Negligible
ncurses-bin	6.2+20201114-2	(won't fix)	deb	CVE-2022-29458	High
ncurses-bin	6.2+20201114-2		deb	CVE-2021-39537	Negligible
passwd	1:4.8.1-1		deb	CVE-2007-5686	Negligible

/vulnerabilities-and-sbom

/State of vulnerabilities

ARMO

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

Reviewing the distribution of severities

Reviewing top CVEs in both

Reachability

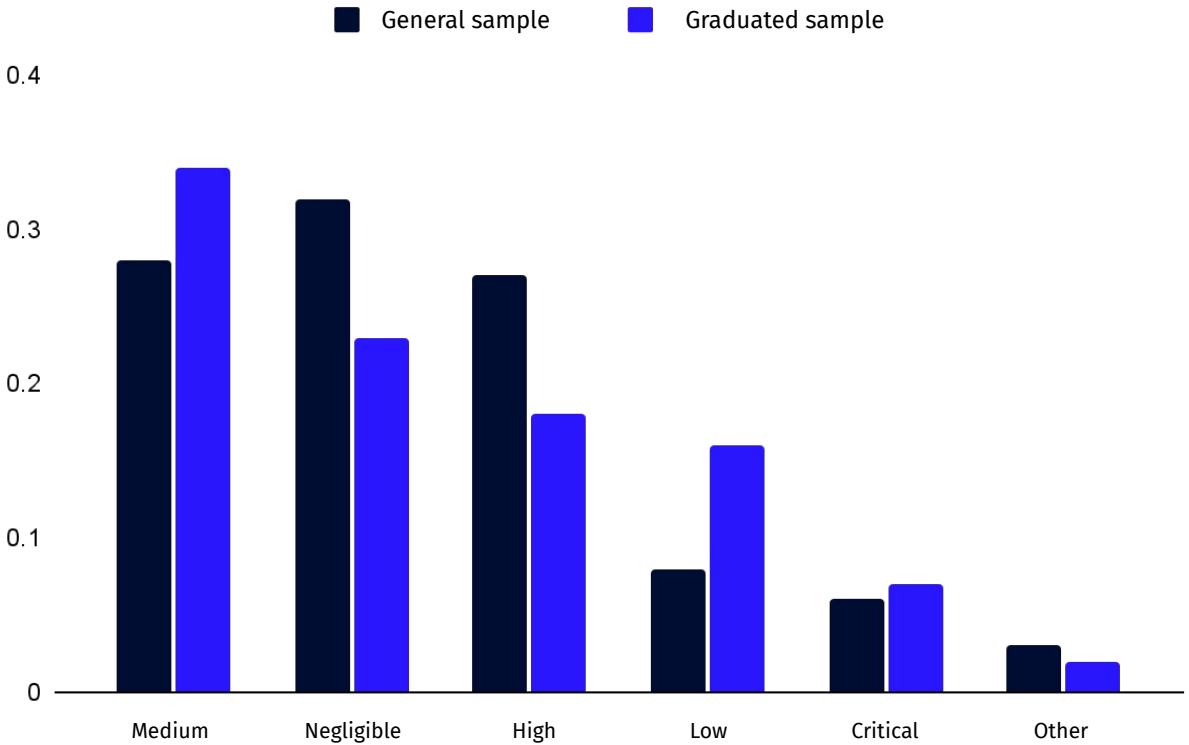
/Image repos with most scans in the general sample

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_


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

/TOP vulnerabilities in general population_

	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 to be exploited**

*gut feeling :-/

	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

	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

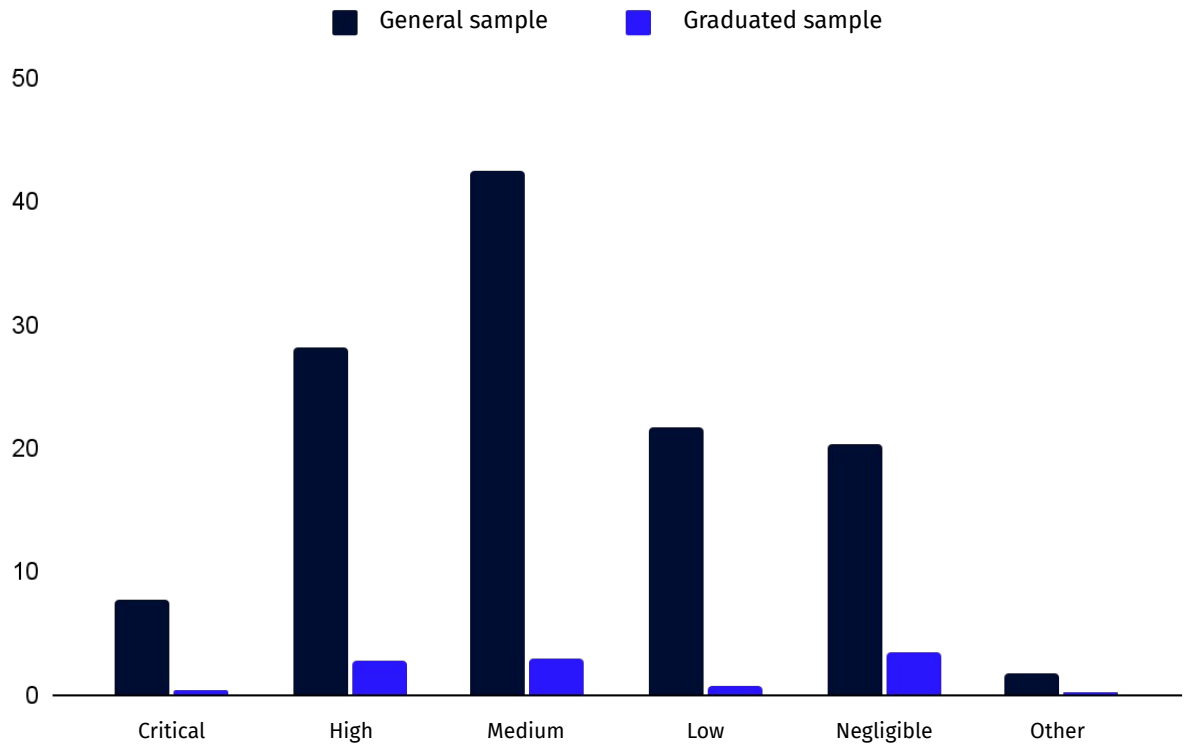
/Opinion: **these are the vulnerabilities has some probability to be exploited**

*gut feeling :-/

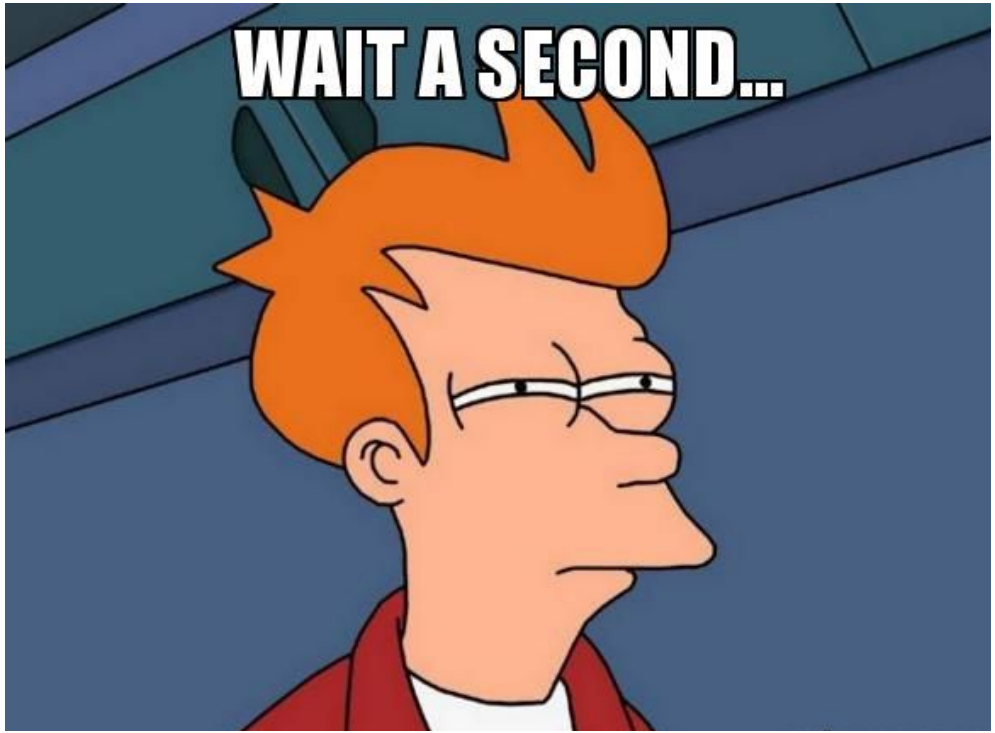
	CVE	Count of imag	severity	description
1				
2				
3	CVE-2022-21698	17	High	In client_golang prior to version 1.11.1, HTTP serve
4	CVE-2022-31836	16	Critical	Function leafInfo.match() use path.join() to deal wit
5	CVE-2022-46146	13	High	Prometheus Exporter Toolkit is a utility package to
6	CVE-2022-31054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

/Looking at general results_

Average vulnerability count per severity




CNCF IS AWESOME


WAIT A SECOND...

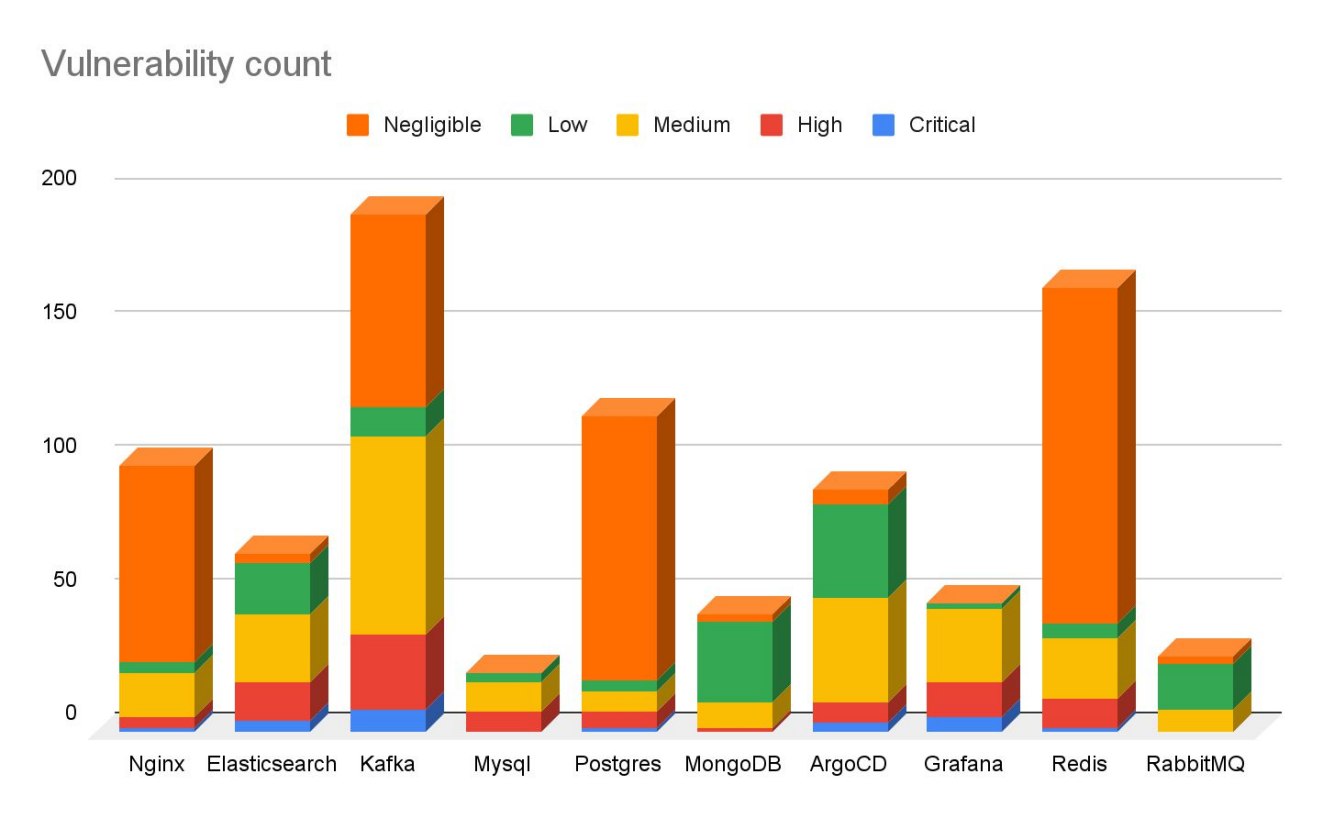
/rewind

/is_it?

Vulnerability in image

Application exploit

/Vulnerabilities in common images_



/Kubescape reachability

SBOM (full)

Files actually used by the container

SBOM (filtered)

Vulnerability scanner

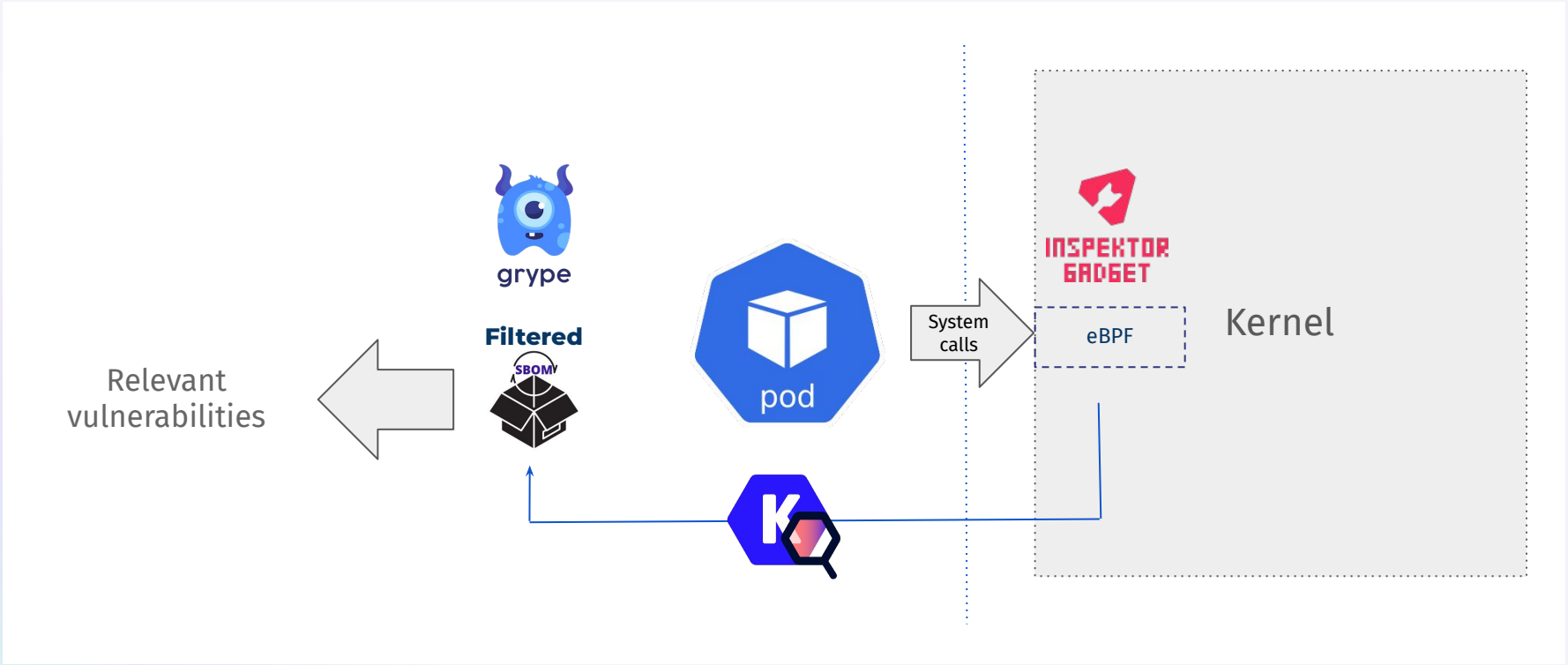
Scan image

eBPF

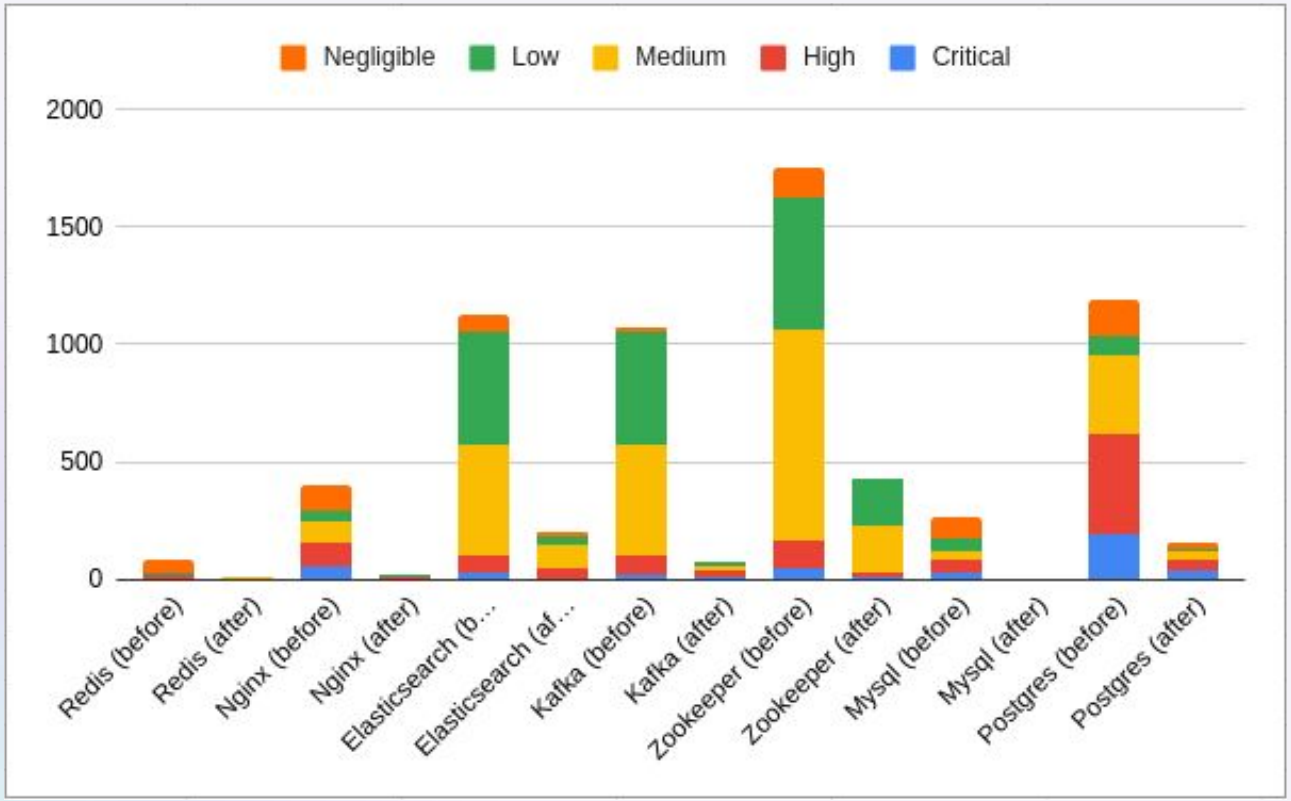
Compare against SBOM

Feed to Vulnerability scanner

/Kubescape reachability



/Kubescape reachability results



Kubescape

/what-is-kubescape

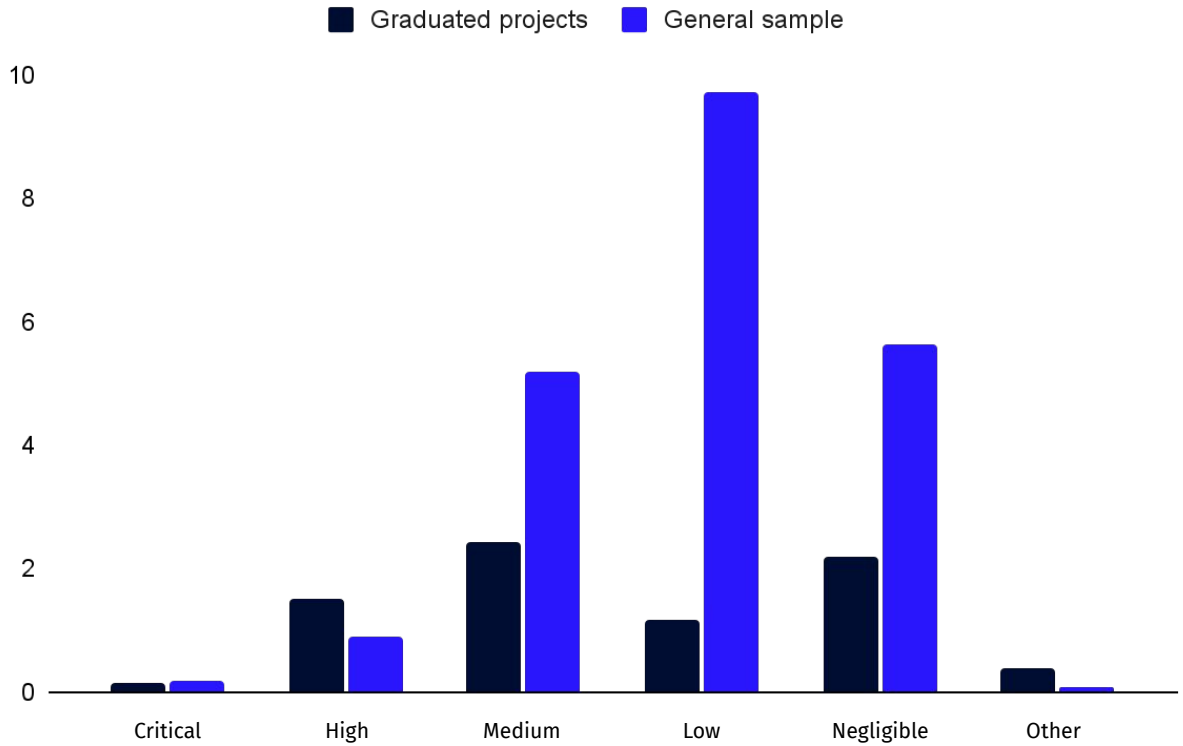
Kubescape

- CNCF Project
- Kubernetes security side-kick
- Configuration & Vulnerability analysis
- Runtime detection
- 10k GitHub starts
- Widely adopted tool (both CLI and service)

SCAN ME

/Looking only at filtered results_

Average relevant vulnerability count per severity



/vex

If SBOM is like the ingredient list of a sausage, then VEX is like the list of allergens

/vex


```
{
  "@context": "https://openvex.dev/ns/v0.2.0",
  "@id": "https://openvex.dev/docs/example/vex-9fb3463de1b57",
  "author": "Wolfi J Inkinson",
  "role": "Document Creator",
  "timestamp": "2023-01-08T18:02:03.647787998-06:00",
  "version": "1",
  "statements": [
    {
      "vulnerability": {
        "name": "CVE-2014-123456"
      },
      "products": [
        {"@id": "pkg:apk/distro/git@2.39.0-r1?arch=armv7"},
        {"@id": "pkg:apk/distro/git@2.39.0-r1?arch=x86_64"}
      ],
      "status": "fixed"
    }
  ]
}
```

/vex-promise

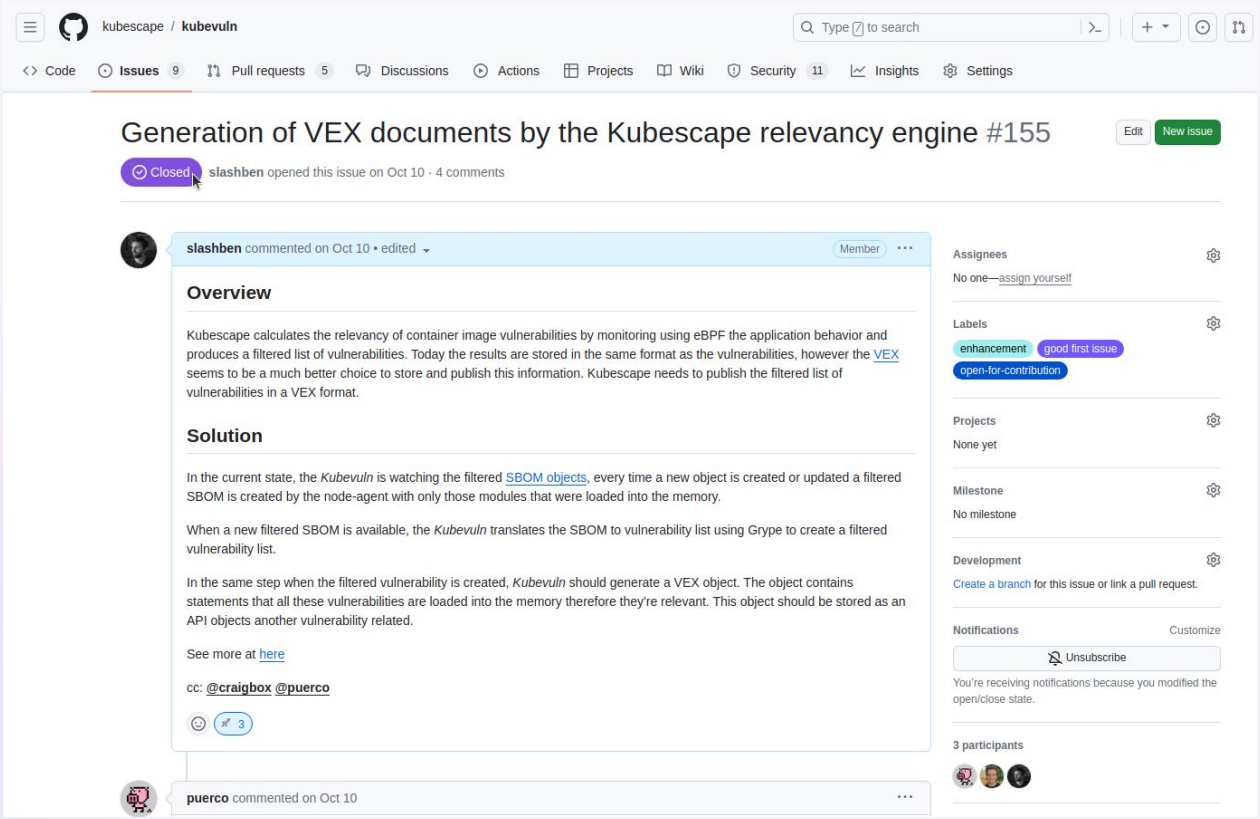

I AM GOING TO TELL YOU
LIGHTYEAR
WHICH VULNERABILITIES REALLY MATTER
imgflip.com

/vex-reality

- Preparing and maintaining reliable VEX is time consuming
- “Not good” if not reliable

/vex-solution-1

/vex-support


kubescape / kubevuln
Type / to search
Code
Issues 9
Pull requests 5
Discussions
Actions
Projects
Wiki
Security 11
Insights
Settings
Generation of VEX documents by the Kubescape relevancy engine #155
Edit
New issue
Closed
slashben opened this issue on Oct 10 · 4 comments
slashben commented on Oct 10 • edited
Member
Overview
Kubescape calculates the relevancy of container image vulnerabilities by monitoring using eBPF the application behavior and produces a filtered list of vulnerabilities. Today the results are stored in the same format as the vulnerabilities, however the VEX seems to be a much better choice to store and publish this information. Kubescape needs to publish the filtered list of vulnerabilities in a VEX format.
Solution
In the current state, the Kubevuln is watching the filtered SBOM objects, every time a new object is created or updated a filtered SBOM is created by the node-agent with only those modules that were loaded into the memory.
When a new filtered SBOM is available, the Kubevuln translates the SBOM to vulnerability list using Grype to create a filtered vulnerability list.
In the same step when the filtered vulnerability is created, Kubevuln should generate a VEX object. The object contains statements that all these vulnerabilities are loaded into the memory therefore they're relevant. This object should be stored as an API objects another vulnerability related.
See more at here
cc: @craigbox @puerco
3
puerco commented on Oct 10
Assignees
No one—assign yourself
Labels
enhancement
good first issue
open-for-contribution
Projects
None yet
Milestone
No milestone
Development
Create a branch for this issue or link a pull request.
Notifications
Customize
Unsubscribe
You're receiving notifications because you modified the open/close state.
3 participants

/ks_installation


```
$ helm repo add kubescape https://kubescape.github.io/helm-charts/
$ helm repo update
$ helm upgrade --install kubescape kubescape/kubescape-operator -n kubescape
--create-namespace --set clusterName=`kubectl config current-context` --set
capabilities.vexGeneration=enable
$ kubectl -n kubescape get pods
NAME                           READY   STATUS    RESTARTS   AGE
kubescape-6bd764869d-nmk5k     1/1     Running   0          99s
kubevuln-76bbbdfcd4-8fxcq      1/1     Running   0          99s
node-agent-dnf6l               1/1     Running   0          99s
operator-75c999bfc6-dlfj8      1/1     Running   0          99s
storage-5898d46fd-rmv4x        1/1     Running   0          99s
```

/generating VEX


```
$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml

$ kubectl -n kubescape get openvulnerabilityexchangecontainer $(kubectl -n kubescape
get openvulnerabilityexchangecontainer -o jsonpath='{.items[0].metadata.name}') -o
jsonpath='{.spec}' > nginx.json

$ jq "." nginx.json | grep -c "\"affected\""
58
$ jq "." nginx.json | grep -c "\"not_affected\""
338
```

/using_with_grype


```
$ grype nginx:1.14.2 --vex nginx.json
 ✔ Vulnerability DB                [no update available]
 ✔ Loaded image
nginx:1.14.2
 ✔ Parsed image
sha256:295c7be079025306c4f1d65997fcf7adb411c88f139ad1d34b537164aa060369
 ✔ Cataloged packages              [111 packages]
 ✔ Scanned for vulnerabilities     [58 vulnerability matches]
   ├── by severity: 55 critical, 102 high, 85 medium, 52 low, 102 negligible
   └── by status:   126 fixed, 270 not-fixed, 338 ignored
```

/takeaways

- **Vulnerabilities by scanners are mostly wrong**
- **Good VEX can mitigate this**
- **VEX can be enhanced automatically**

/contribute_to_the_effort

As a user

As a developer

As a security expert

Thank you

www.armosec.io