



KubeCon



CloudNativeCon

Europe 2020

Virtual

Handling Container Vulnerabilities with Open Policy Agent

Teppei Fukuda (@knqyf263)
Maintainer, Trivy
Open Source Team, Aqua Security

Software vulnerabilities



MELTDOWN



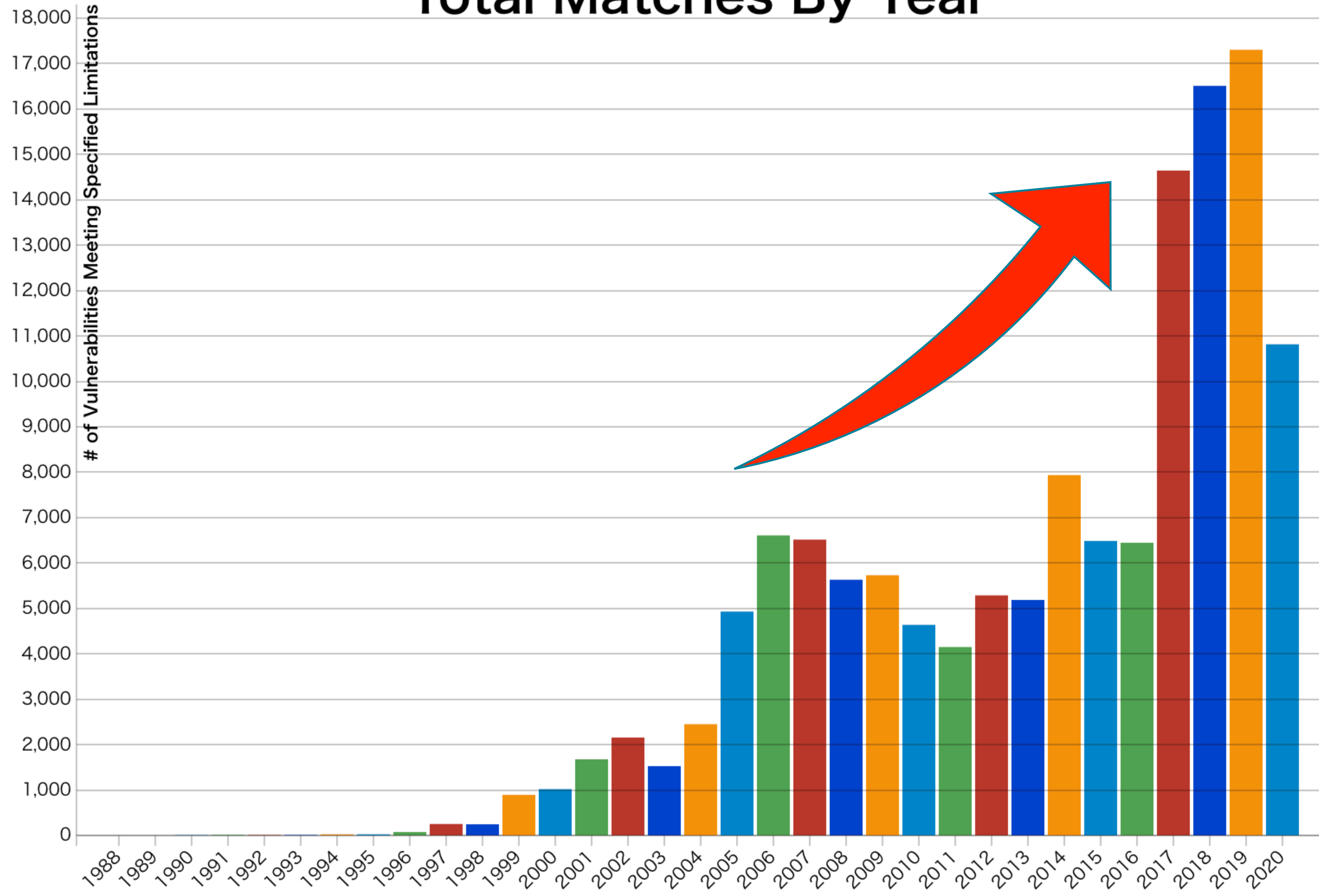
CVE-2014-0160



DIRTY COW

How many vulnerabilities are reported?

Total Matches By Year



<https://nvd.nist.gov/vuln/search/statistics>

The number of vulnerabilities (2019)

Per Year

17,306

Per Day

47.4

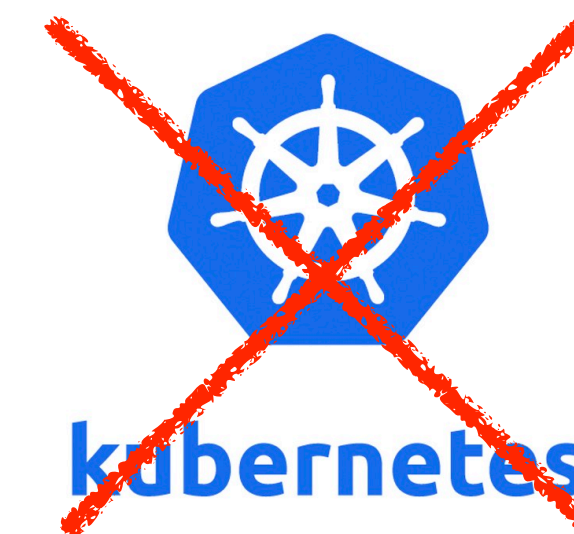
Not easy to understand how vulnerabilities work

Asset management

- Need to know
 - Which OS
 - What package
 - What programming language
 - What library
- are used in your system



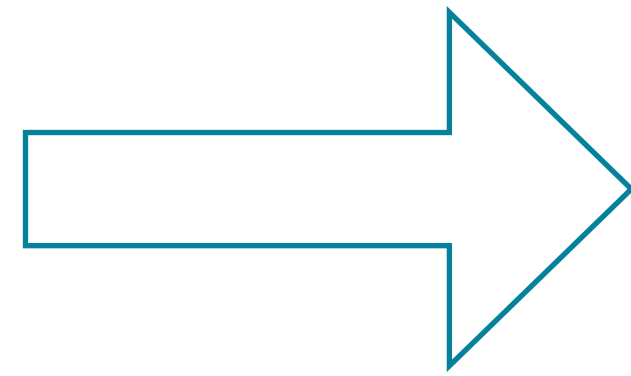
47.4 /day



Asset management

Remove vulnerabilities not related to your organization

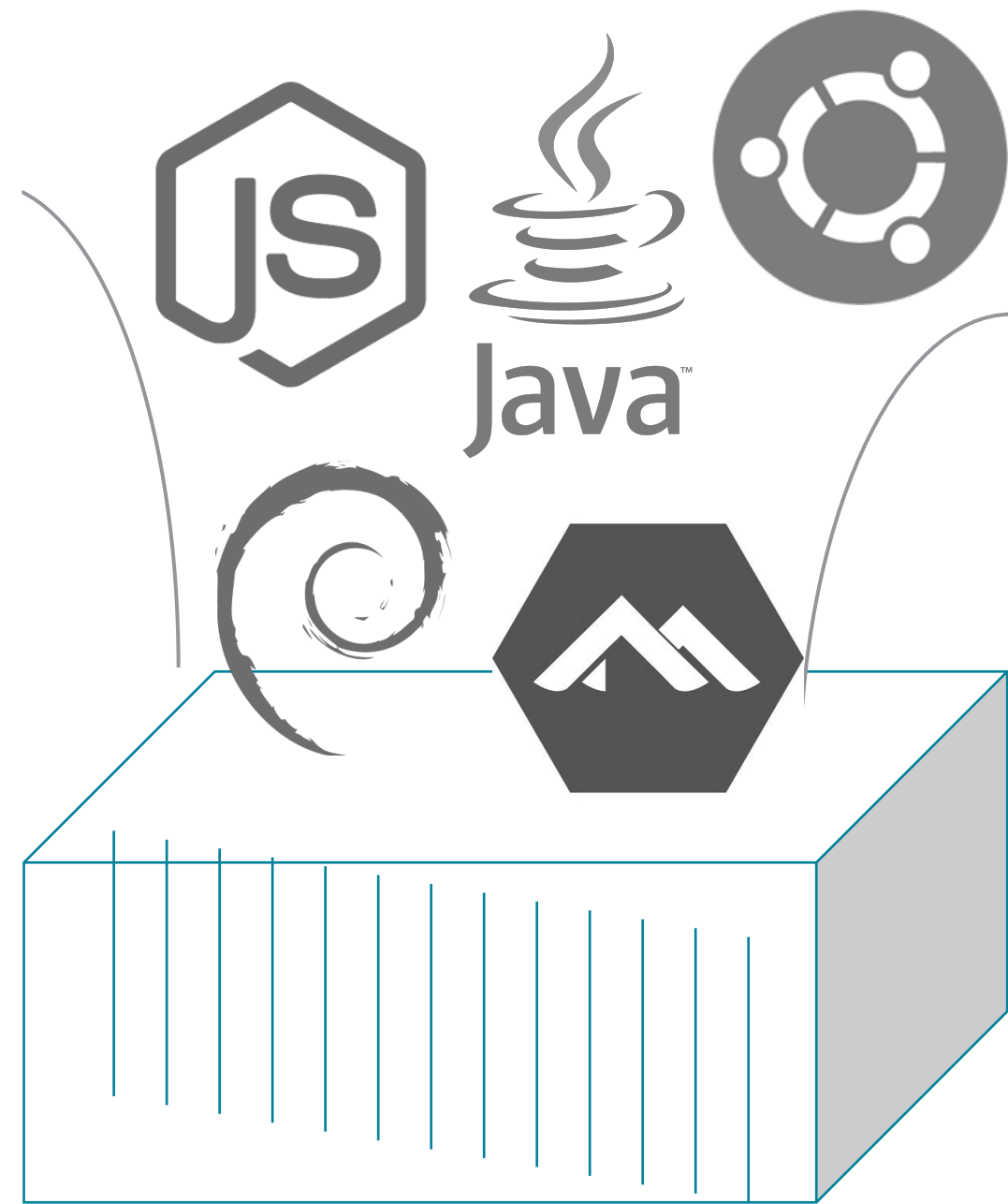
47.4 /day



5-10 /day

Manually?

Container



Asset

←→
Cross-reference



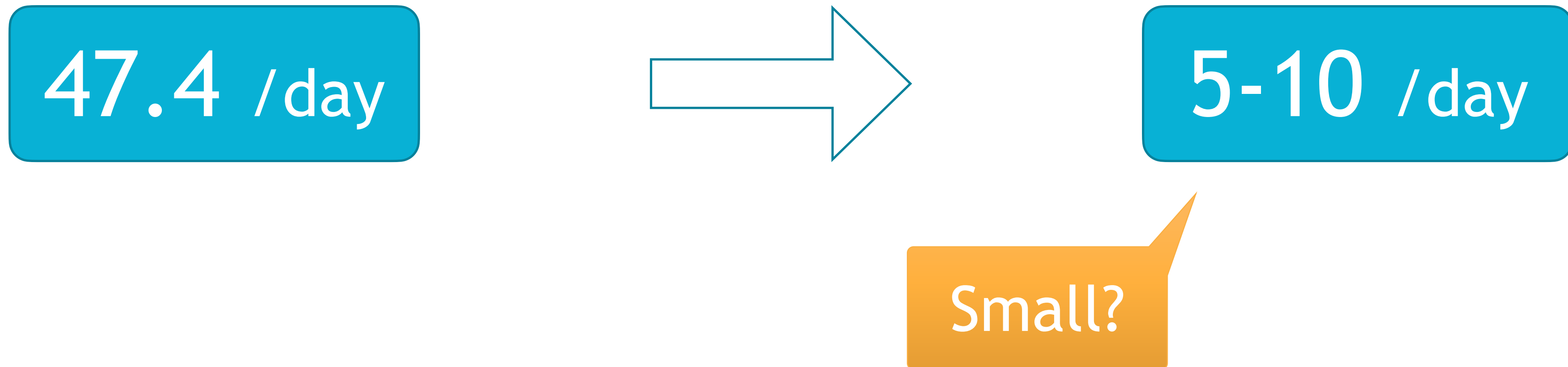
Vulnerability data

Vulnerability scanners in the cloud native area

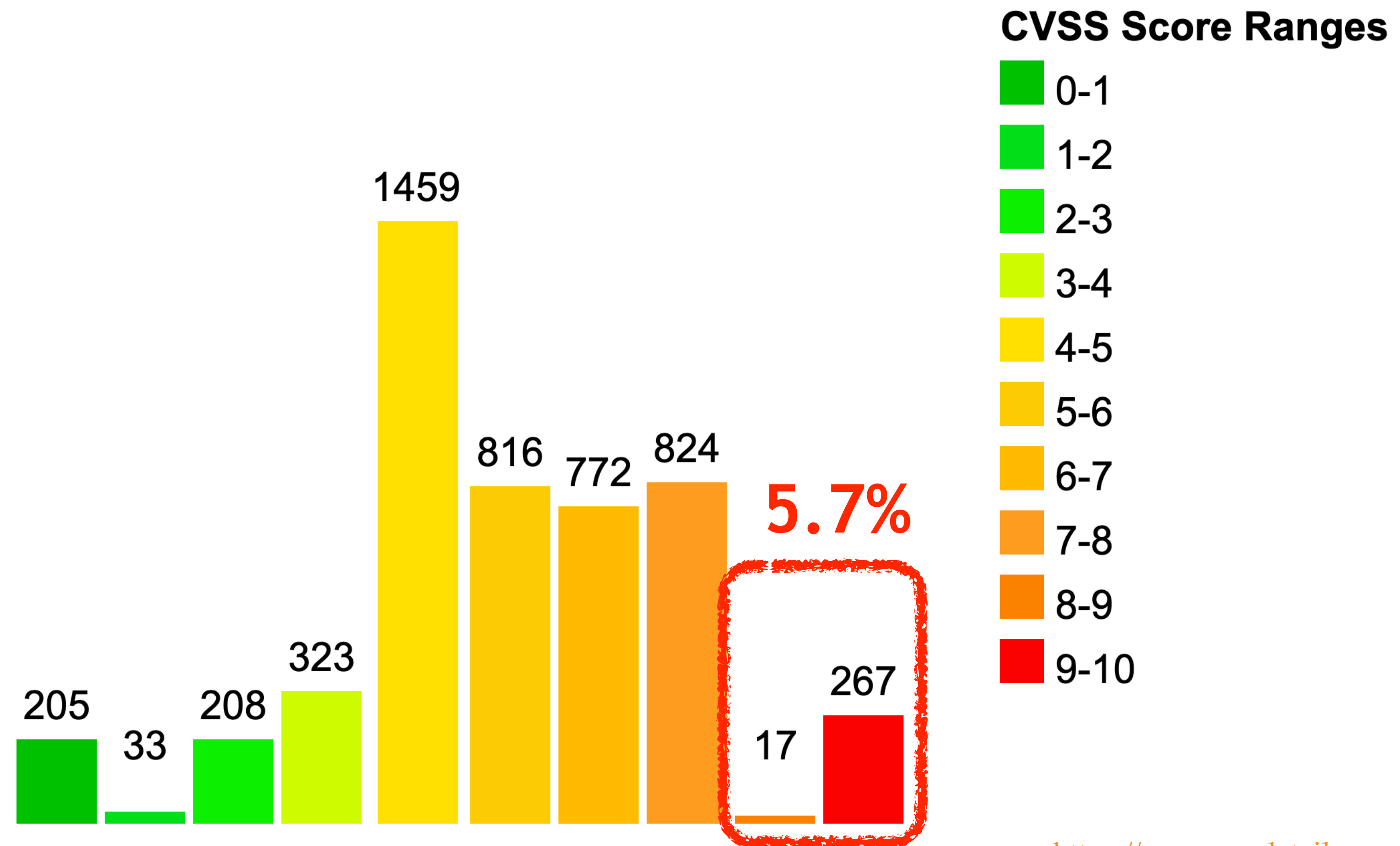


Automated vulnerability scanning

Vulnerability scanning



Vulnerability Distribution By CVSS Scores (2019)

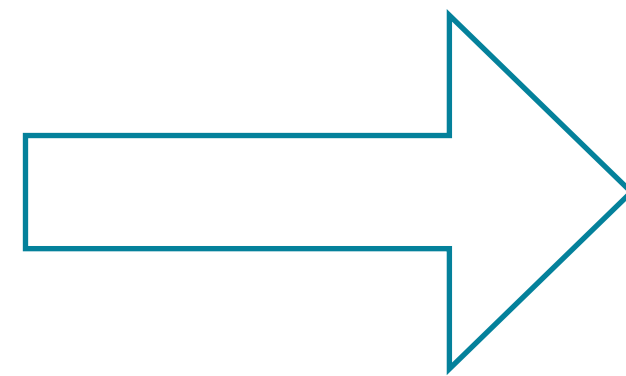


<https://www.cvedetails.com/cvss-score-charts.php>

Filter by CVSS score

Only critical vulnerabilities

5-10 /day



0-1 /day

Is the CVSS score reliable?

- CVE-2014-0160 (Heartbleed)



Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 2.0 Severity and Metrics:



NIST: NVD

Base Score: 5.0 MEDIUM

Vector: (AV:N/AC:L/Au:N/C:P/I:N/A:N)

Not critical?

Hackers exploit Heartbleed to swipe data of 4.5 million

FBI industry alert late to the game

By **Erin McCann** | December 12, 2014 | 02:34 PM



<https://www.healthcareitnews.com/news/hackers-exploit-heartbleed-swipe-data-45-million>

CVE-2017-15896

Severity CVSS Version 3.x CVSS Version 2.0


CVSS 3.x Severity and Metrics:

 **NIST: NVD**


Base Score:
9.1 CRITICAL

Vector:
CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

<https://nvd.nist.gov/vuln/detail/CVE-2017-15896>

 **Red Hat Customer Portal** Products & Services Tools Security

CVE-2017-15896
Public on 2017年12月7日

 Moderate Impact
[What does this mean?](#)

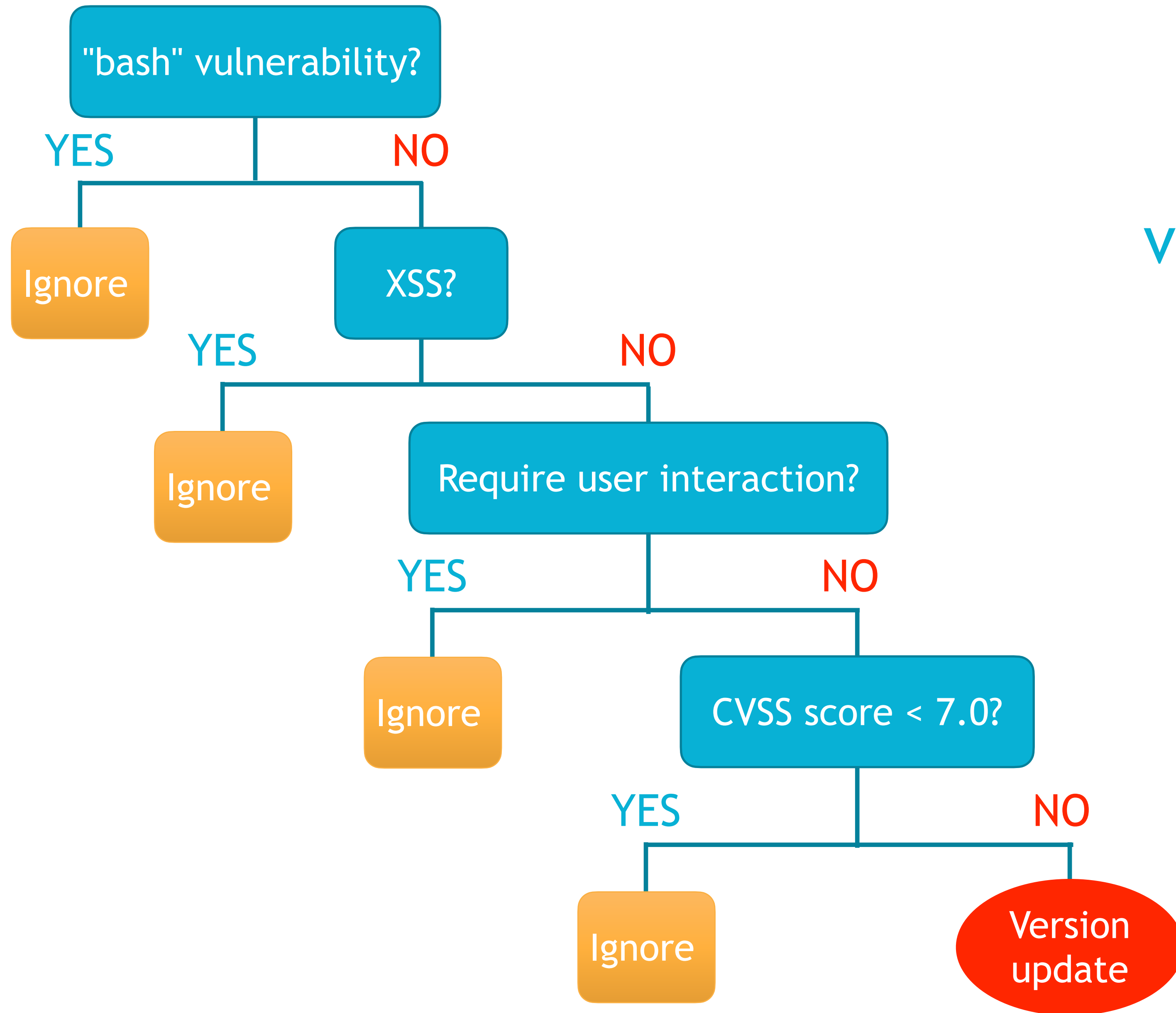
5.9 CVSS v3 Base Score
[CVSS Score Breakdown](#)

<https://access.redhat.com/security/cve/cve-2017-15896>

Vulnerability handling


- Define our own policy for vulnerability handling
 - It depends on your system, organization, etc.
- e.g.
 - The risk of "bash" vulnerabilities can be accepted
 - "bash" is not internet-facing
 - The risk of "XSS" can be accepted
 - the system is static
 - The vulnerability which requires user interaction can be ignored
 - e.g. a successful exploit may only be possible during the installation of an application by a system administrator.

Policy for vulnerability handling




Other useful information for vulnerability handling

- CVSS vector
- CWE-ID

 **NIST: NVD** **Base Score:** 7.5 HIGH **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	 NIST

CVSS vector

Attack Vector (AV)

Network (N) Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC)

Low (L) High (H)

Privileges Required (PR)

None (N) Low (L) High (H)

User Interaction (UI)

None (N) Required (R)

Scope (S)

Unchanged (U) Changed (C)

Confidentiality (C)

None (N) Low (L) High (H)

Integrity (I)

None (N) Low (L) High (H)

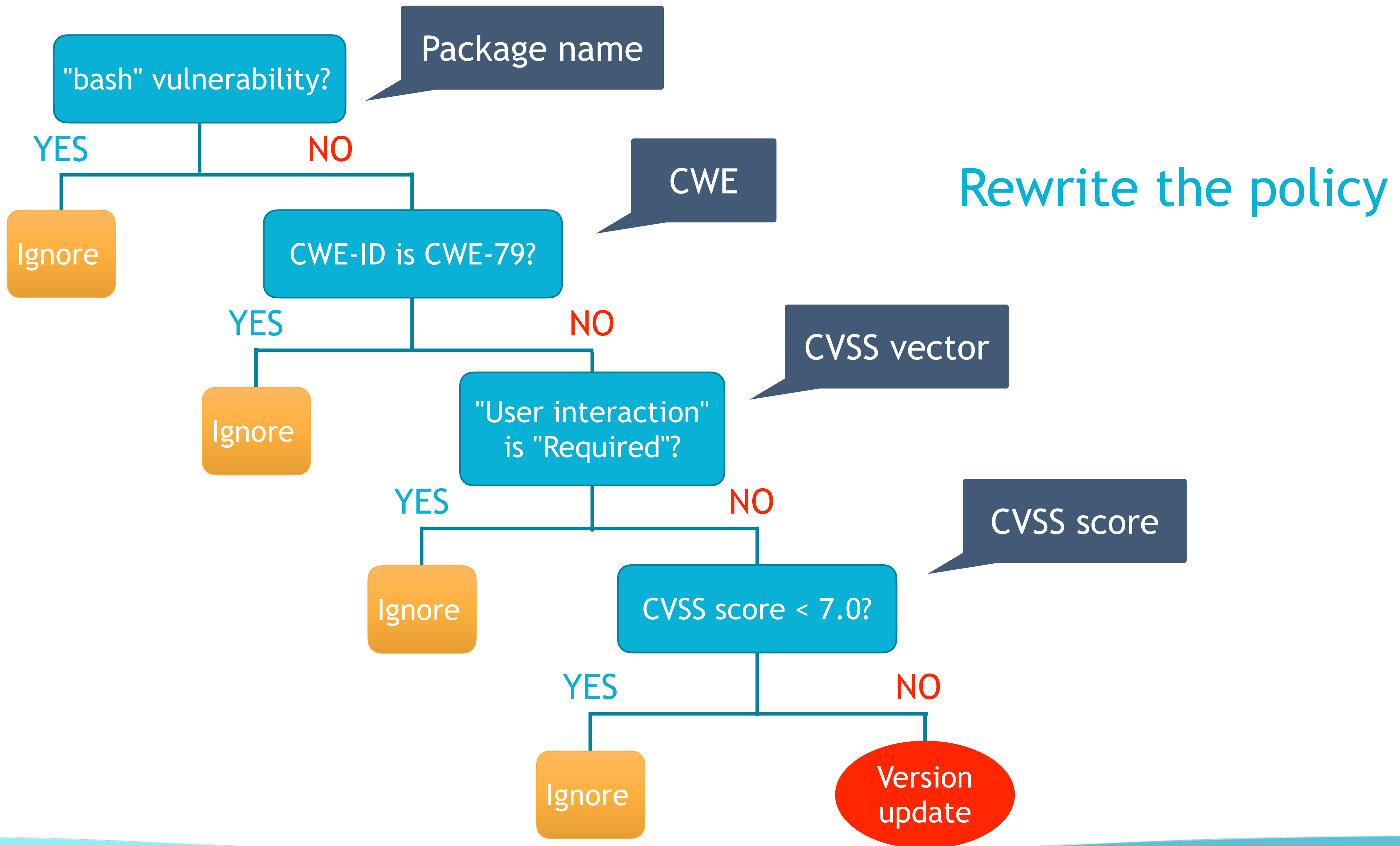
Availability (A)

None (N) Low (L) High (H)

<https://www.first.org/cvss/calculator/3.0>

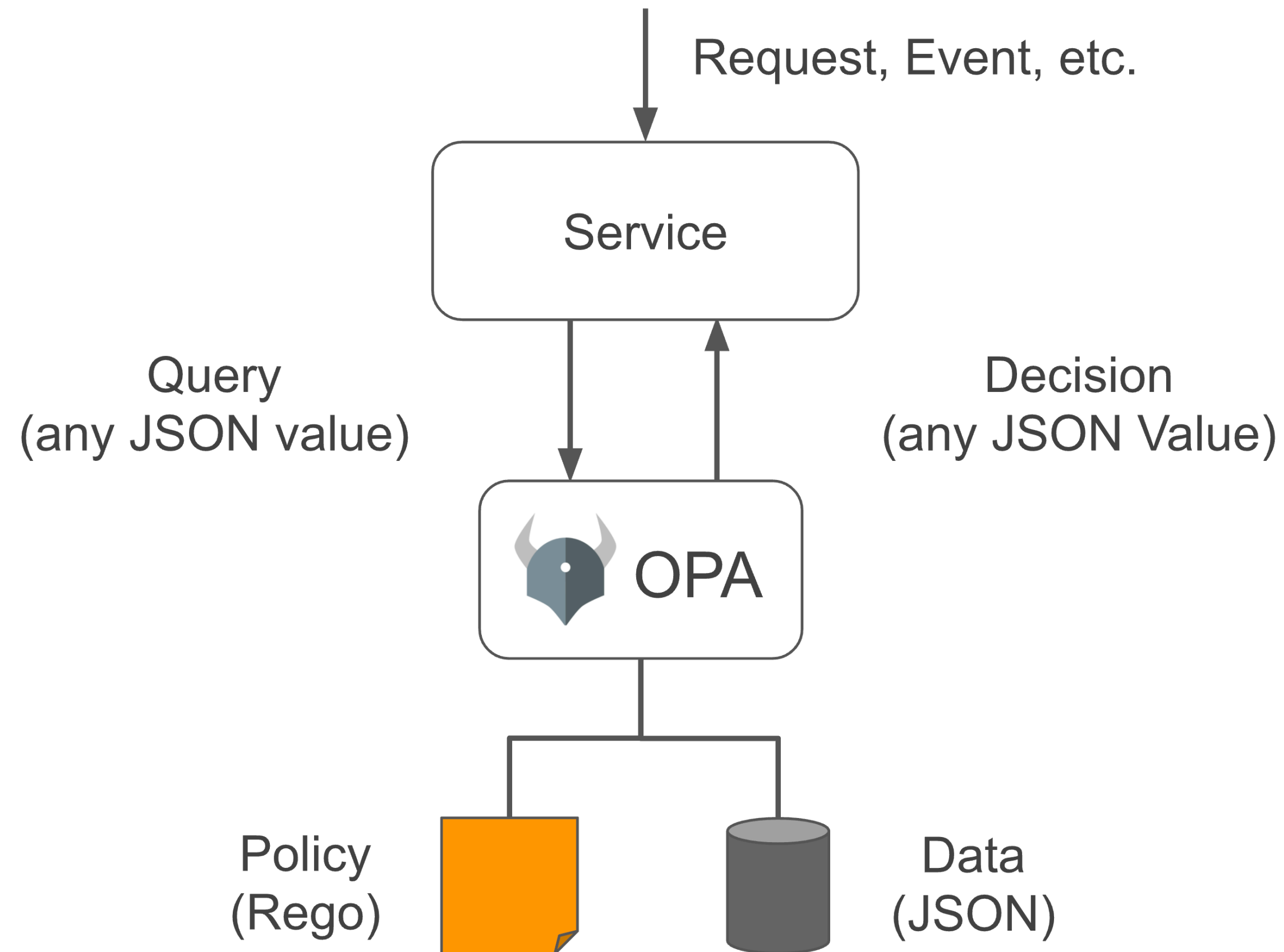
CWE

- CWE (Common Weakness Enumeration) aims to provide a common base to identify the type of software weakness (vulnerability).
- e.g.
 - CWE-78: OS Command Injection
 - CWE-79: Cross-site scripting (XSS)
 - CWE-89: SQL Injection

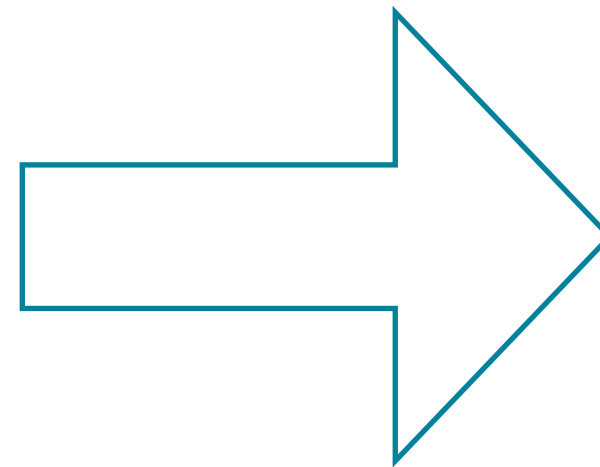
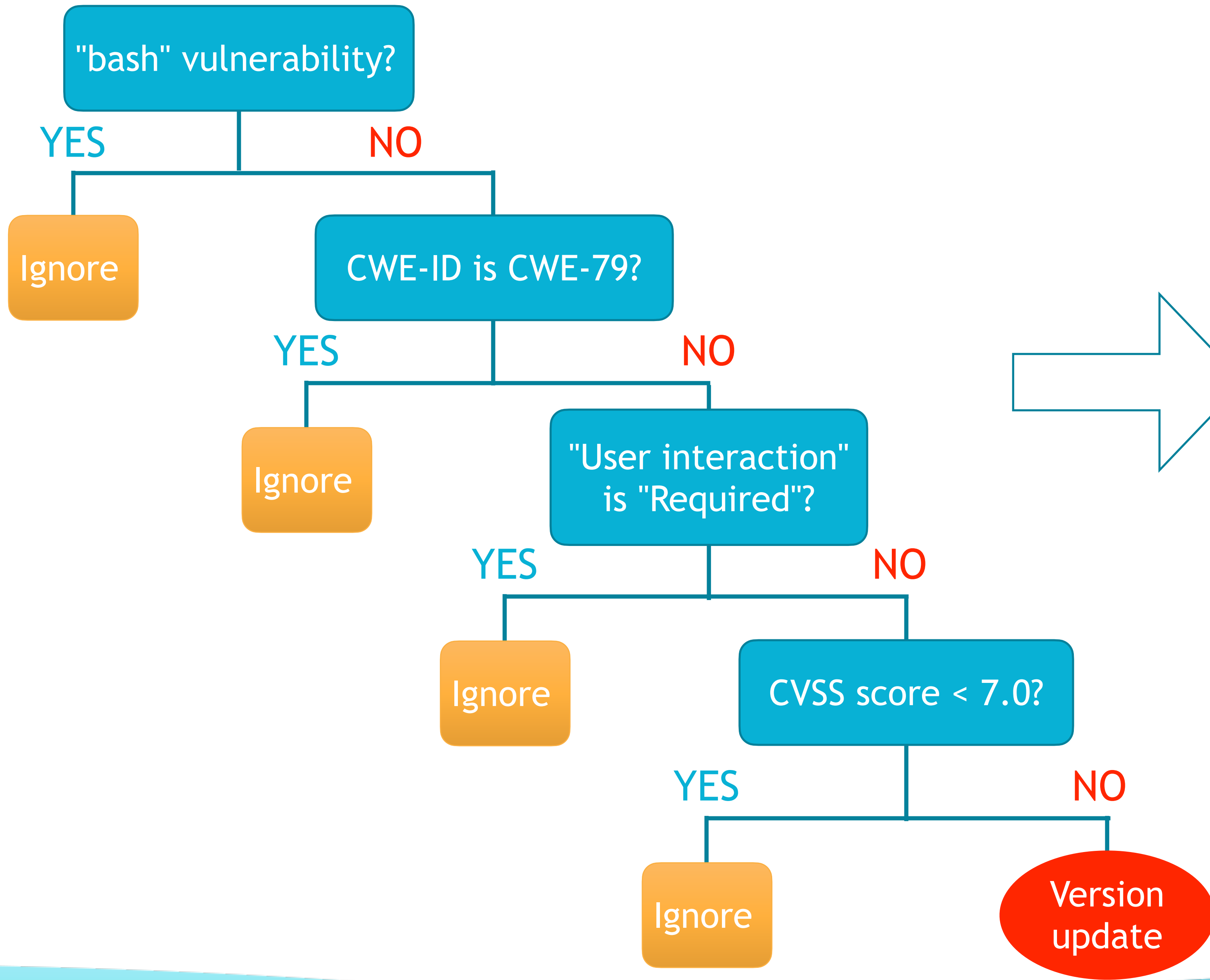


Vulnerability handling with Open Policy Agent

Open Policy Agent (OPA)



- Open source policy engine
- CNCF project
- Usable as a library and a service
- Provides a declarative DSL for writing policy called Rego



Rego




```
1 package vulnerability
2
3 default ignore = false
4
5 ignore {
6     input.pkg_name == "bash"
7 }
8
9 ignore {
10    input.cwe_id == "CVE-79" # XSS
11 }
12
13 ignore {
14    input.cvss_vector.user_interaction == "required"
15 }
16
17 ignore {
18    input.cvss_score < 7.0
19 }
```

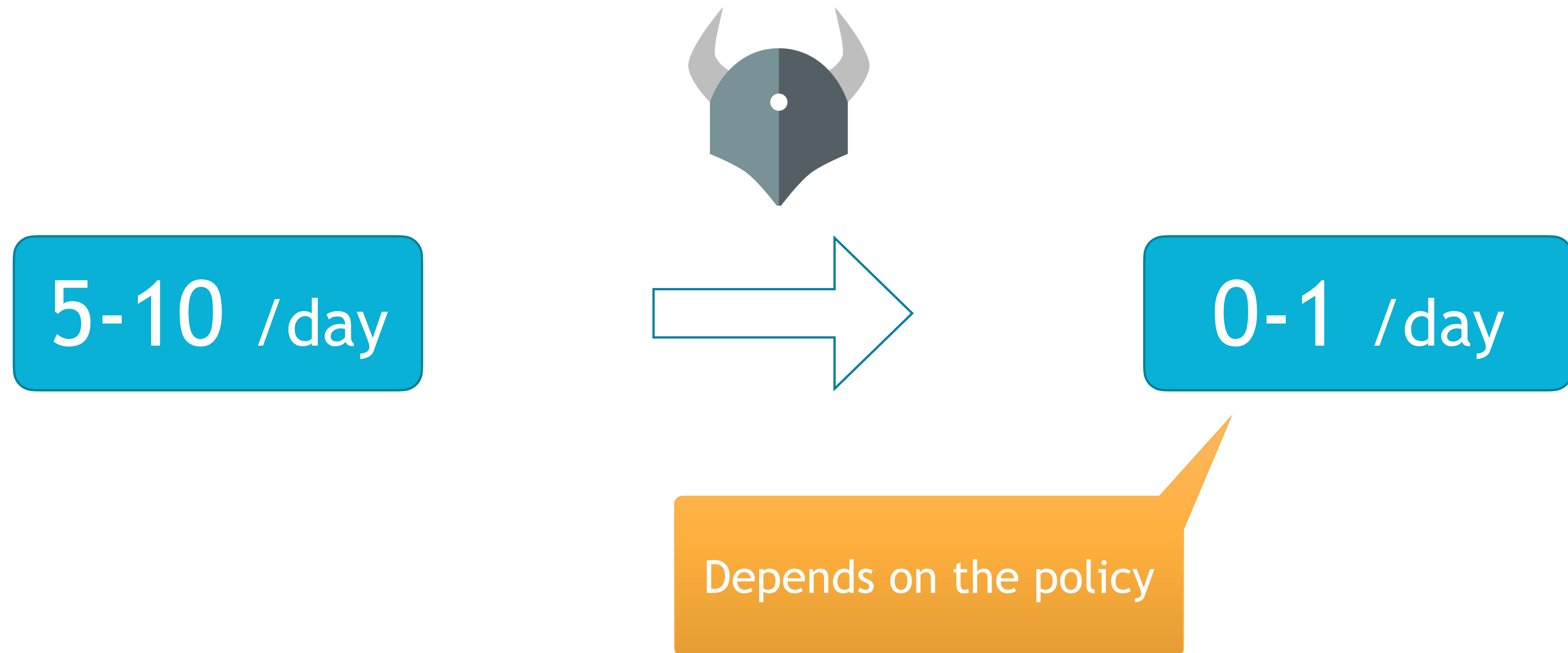
INPUT

```
1 {
2     "pkg_name": "openssl",
3     "cve_id": "CVE-2019-1547",
4     "cwe_id": "CWE-200",
5     "cvss_score": 5.5,
6     "cvss_vector": {
7         "attack_vector": "network",
8         "user_interaction": "required"
9     }
10 }
```

Vulnerability detail

<https://play.openpolicyagent.org/p/cBZA3qkslV>

Filter by composite rules



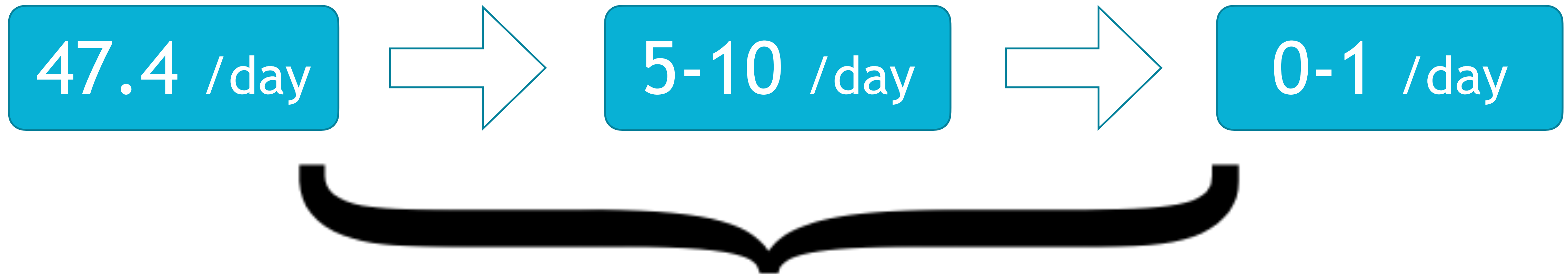
Vulnerability information is not always correct

- NVD says "User interaction" is "Required", but it might be wrong
- Don't trust the vulnerability information too much
- It's best to read the patch and primary source



Apply OPA to the result from vulnerability scanner

Vulnerability scanning



Automated process

OPA integration in Trivy

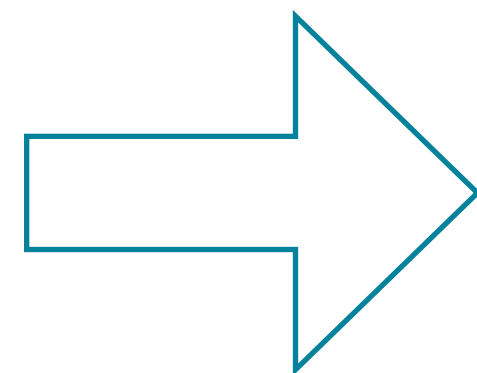
Case study: Trivy & OPA



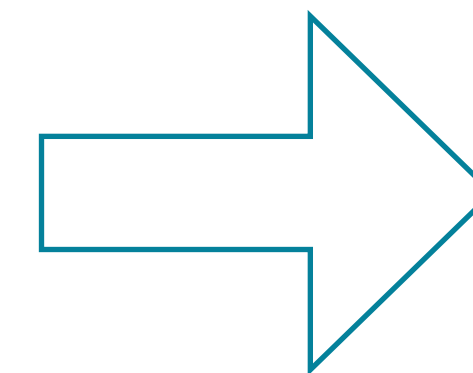
trivy



47.4 /day



5-10 /day



0-1 /day

Vulnerability
detection

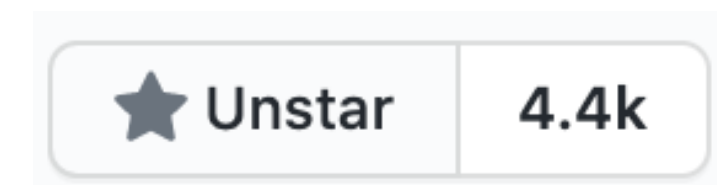
Evaluate
vulnerabilities

Trivy

- Open source scanner for container images
- Developed in 2019
- Features
 - Easy installation
 - Simple & Fast
 - **DevSecOps**

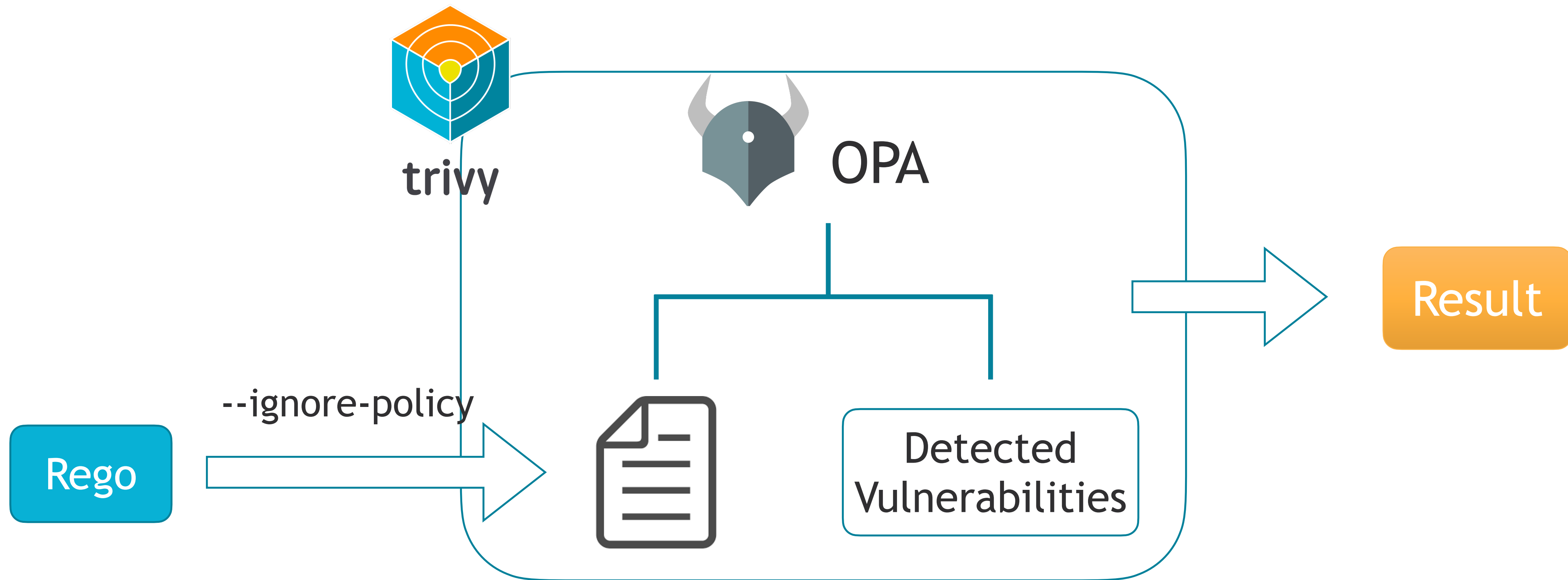


trivy



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

OPA Integration



* **EXPERIMENTAL** feature

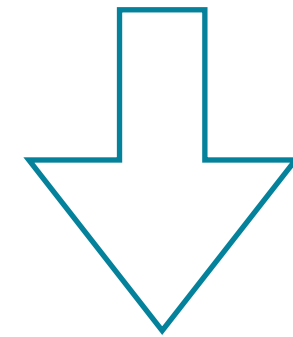
INPUT

```
1 {
2   "VulnerabilityID": "CVE-2019-1547",
3   "PkgName": "openssl",
4   "Title": "openssl: side-channel weak encryption vulnerability",
5   "Description": "Normally in OpenSSL EC groups always have a co-factor present and ...
6   "Severity": "LOW",
7   "InstalledVersion": "1.1.1c-r0",
8   "FixedVersion": "1.1.1d-r0",
9   "CweIDs": [
10    "CWE-311"
11  ],
12  "CVSS": {
13    "nvd": {
14      "V2Vector": "AV:L/AC:M/Au:N/C:P/T:N/A:N",
15      "V3Vector": "CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N",
16      "V2Score": 1.9,
17      "V3Score": 4.7
18    },
19  },
20  "References": [
21    "https://git.openssl.org/gitweb/?..."
22  ]
23 }
24
```

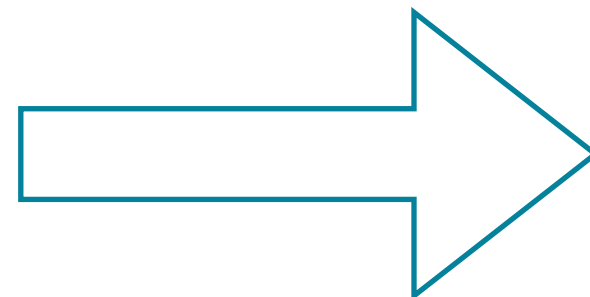
The structure of each vulnerability input is the same as for the Trivy JSON output.

Helper functions

CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N



parse_cvss_vector_v3



```
{  
  "AttackVector": "Local",  
  "AttackComplexity": "High",  
  "PrivilegesRequired": "Low",  
  "UserInteraction": "None",  
  "Scope": "Unchanged",  
  "Confidentiality": "High",  
  "Integrity": "None",  
  "Availability": "None"  
}
```

Policy example

```
1 package trivy
2
3 import data.lib.trivy
4
5 default ignore = false
6
7 ignore_pkgs := {"bash", "bind-license", "rpm", "vim", "vim-minimal"}
8
9 ignore_severities := {"LOW", "MEDIUM"}
10
11 nvd_v3_vector = v {
12     v := input.CVSS.nvd.v3
13 }
14
15 ignore {
16     input.PkgName == ignore_pkgs[_]
17 }
18
19 ignore {
20     input.Severity == ignore_severities[_]
21 }
```

```
23 # Ignore a vulnerability which is not remotely exploitable
24 ignore {
25     cvss_vector := trivy_parse_cvss_vector_v3(nvd_v3_vector)
26     cvss_vector.AttackVector != "Network"
27 }
28
29 # Ignore a vulnerability which requires high privilege
30 ignore {
31     cvss_vector := trivy_parse_cvss_vector_v3(nvd_v3_vector)
32     cvss_vector.PrivilegesRequired == "High"
33 }
34
35 # Ignore a vulnerability which requires user interaction
36 ignore {
37     cvss_vector := trivy_parse_cvss_vector_v3(nvd_v3_vector)
38     cvss_vector.UserInteraction == "Required"
39 }
40
41 # Ignore CSRF
42 ignore {
43     # https://cwe.mitre.org/data/definitions/352.html
44     input.CweIDs[_] == "CWE-352"
45 }
```

Demo

centos:7

Without policy

```
$ trivy image centos:7
centos:7 (centos 7.8.2003)
=====
Total: 622 (UNKNOWN: 0, LOW: 361, MEDIUM: 252, HIGH: 9, CRITICAL: 0)
```

With policy

```
$ trivy image --ignore-policy example.rego centos:7
centos:7 (centos 7.8.2003)
=====
Total: 7 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 7, CRITICAL: 0)
```

OPA integration in Kubernetes

Trivy Enforcer

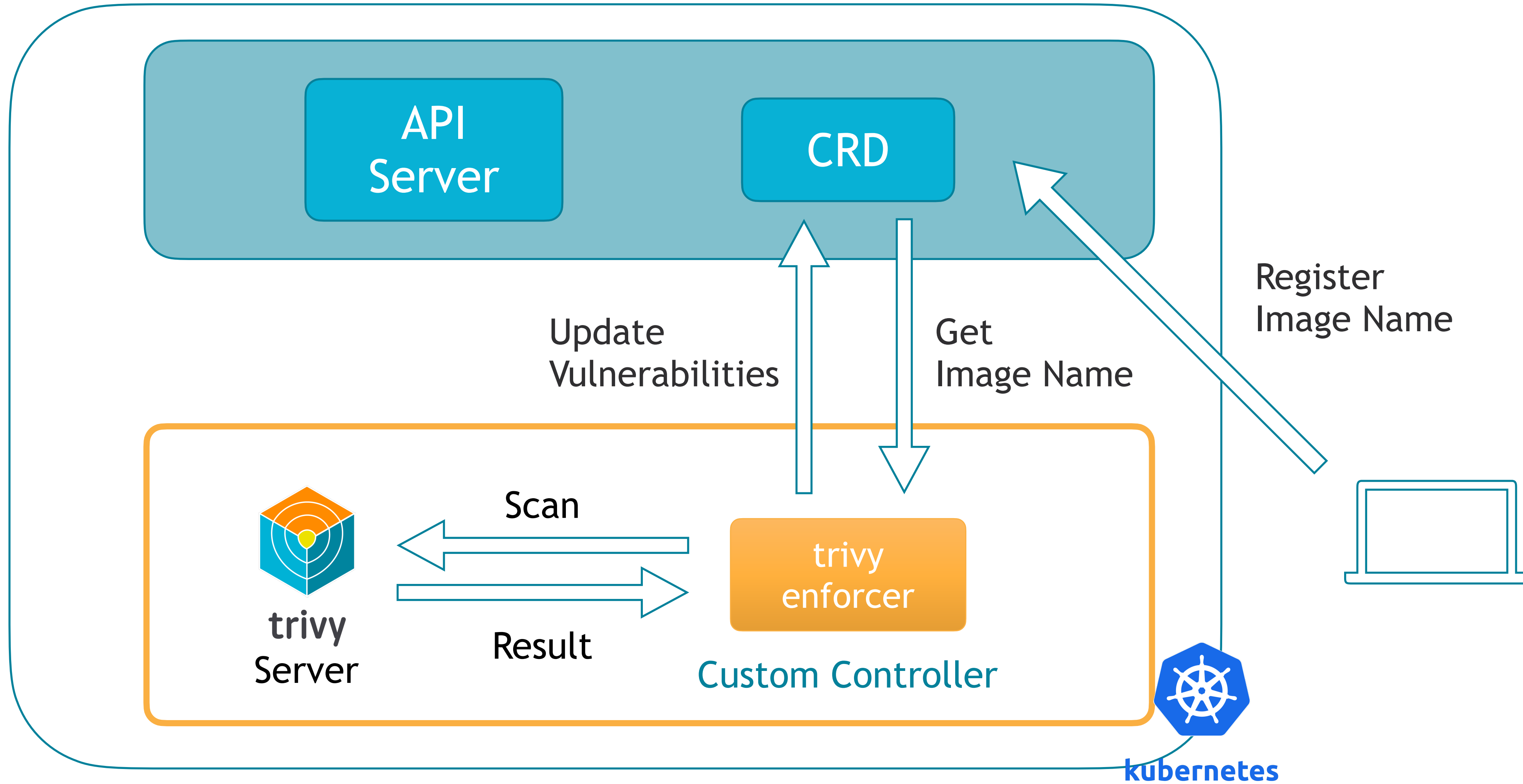
- Kubernetes Operator
 - Run as Custom Controller
 - Pre-Scan
 - Run as Admission Controller
 - Image Assurance
- **EXPERIMENTAL** project (PoC)



kubernetes

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

Pre-Scan



Load Policies

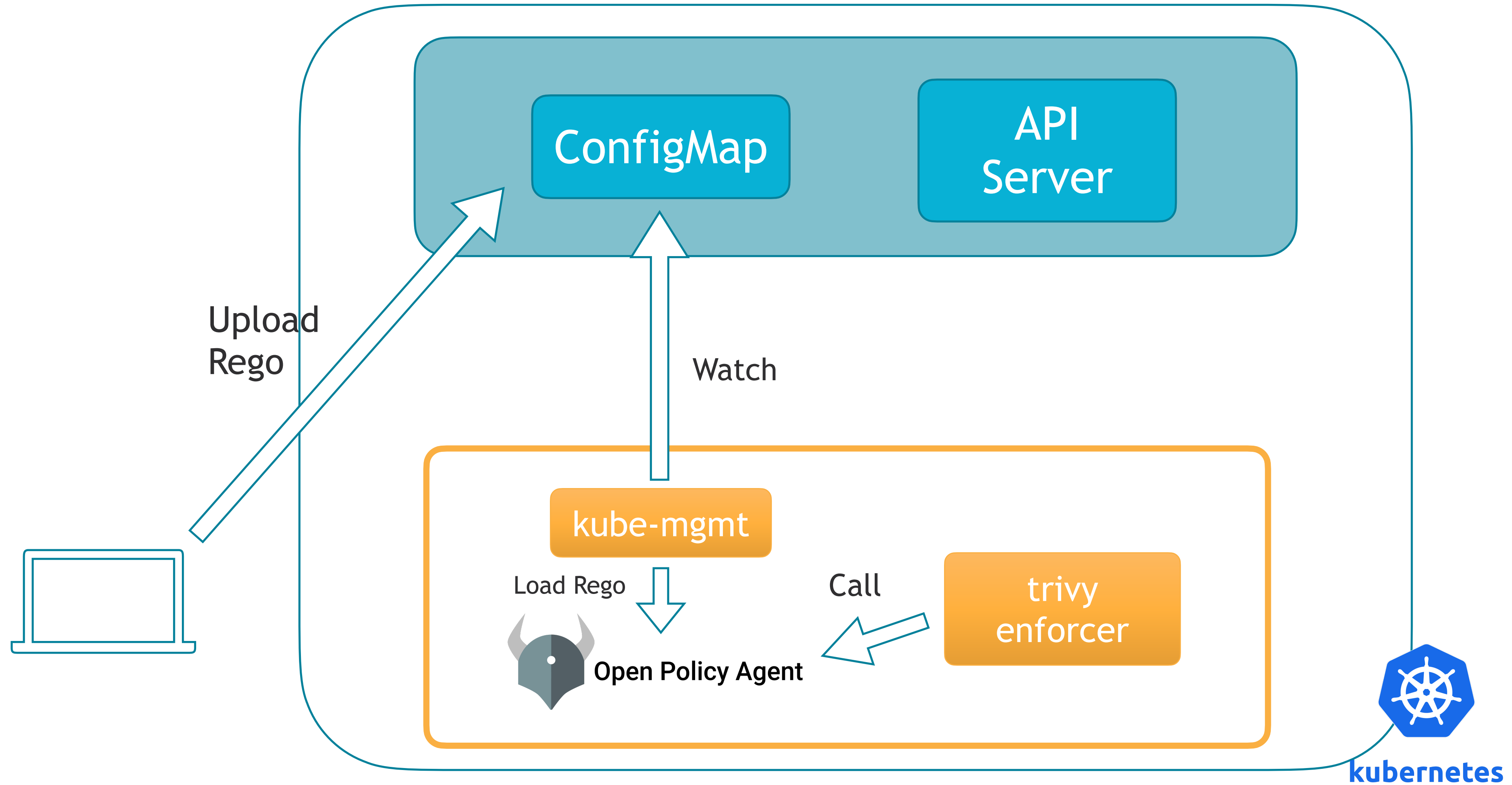
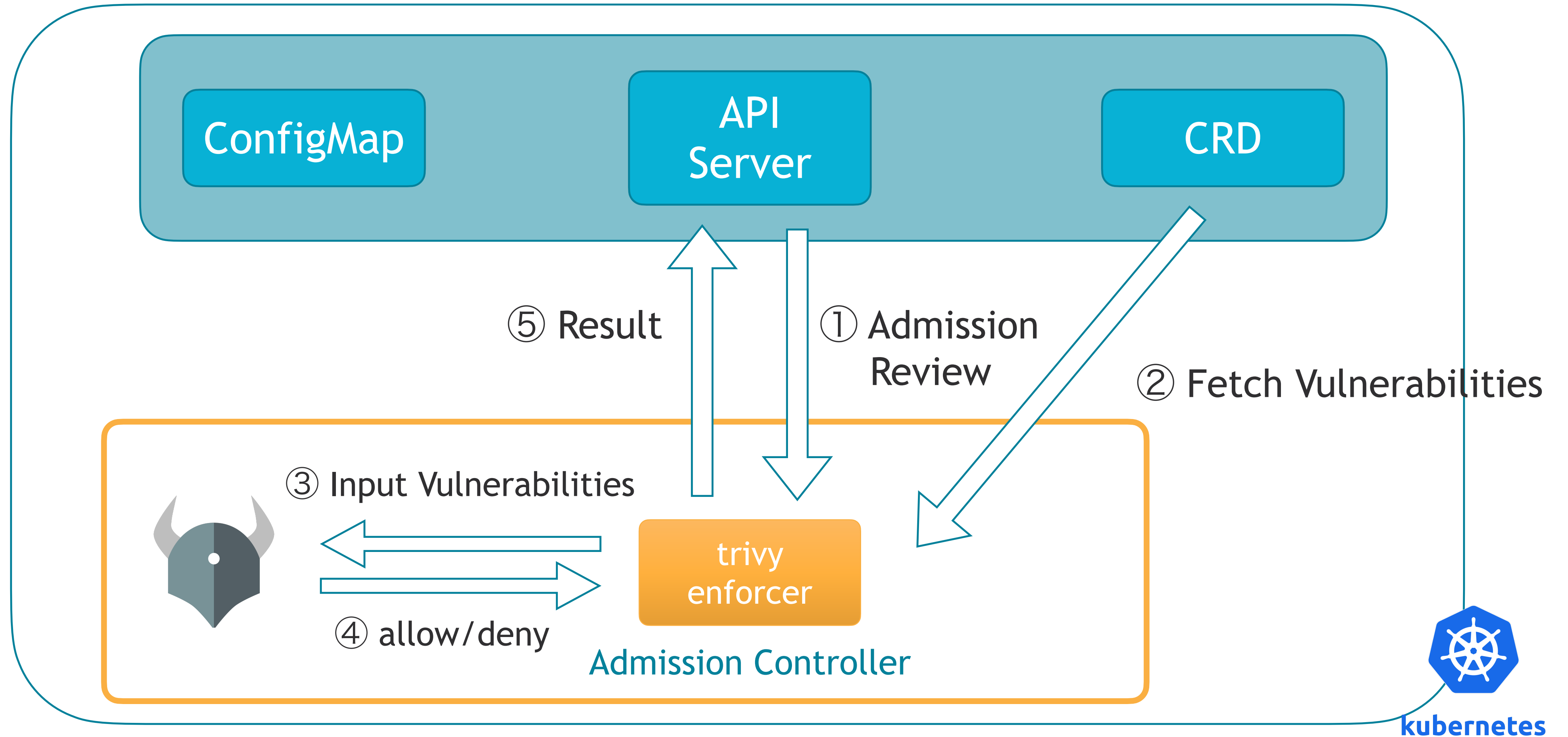
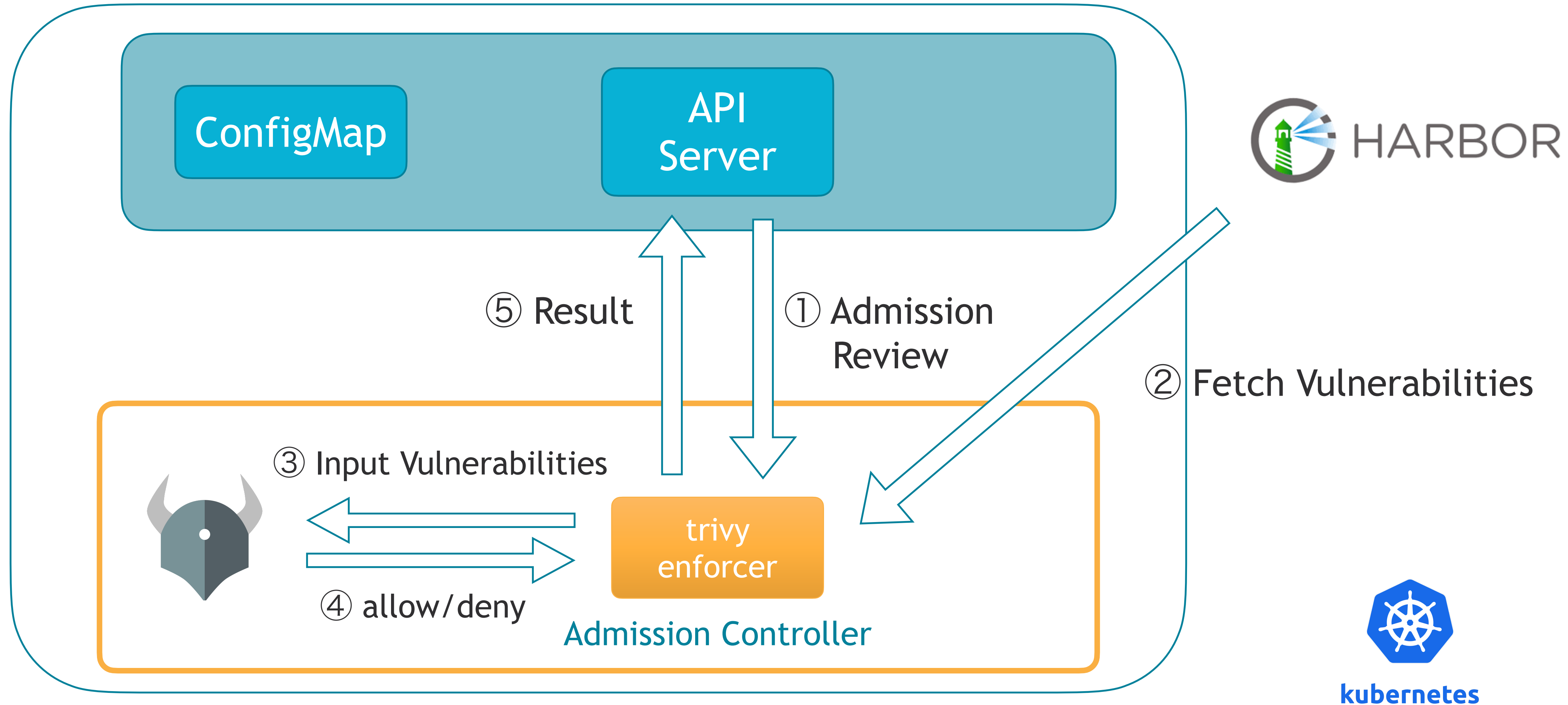


Image Assurance



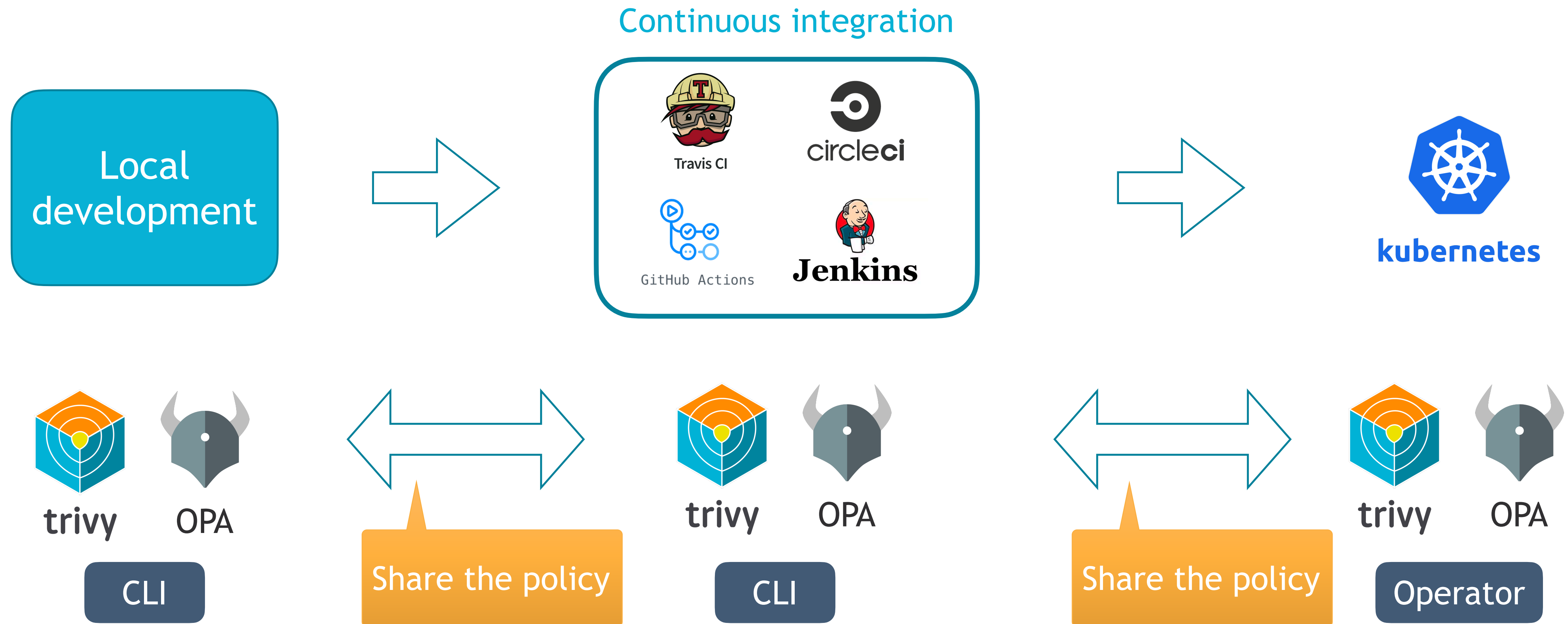
Demo

Image Assurance with Harbor



Demo

Image Assurance throughout the development lifecycle



Summary

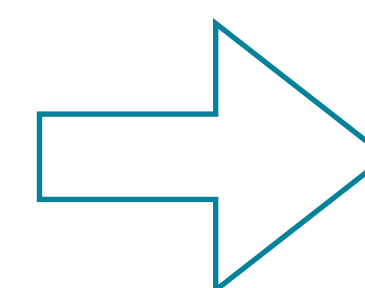
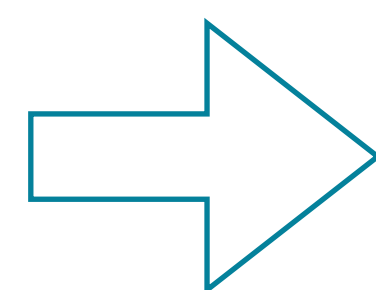
- Define your custom policy for vulnerability handling
- Open Policy Agent integration
 - Trivy CLI
 - Trivy Enforcer (Kubernetes Operator)
- Image Assurance throughout the development lifecycle



trivy



OPA



kubernetes

Share the policy

Share the policy

Thank you for your attention

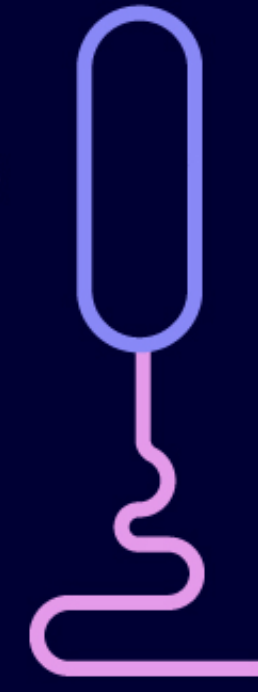


KubeCon



CloudNativeCon

Europe 2020



Virtual



KEEP CLOUD NATIVE

CONNECTED

