Understanding Exploitability with VEX, EPSS, and Other Standard Frameworks

The Problem

Too Many Vulnerabilities, Impossible to Keep Up

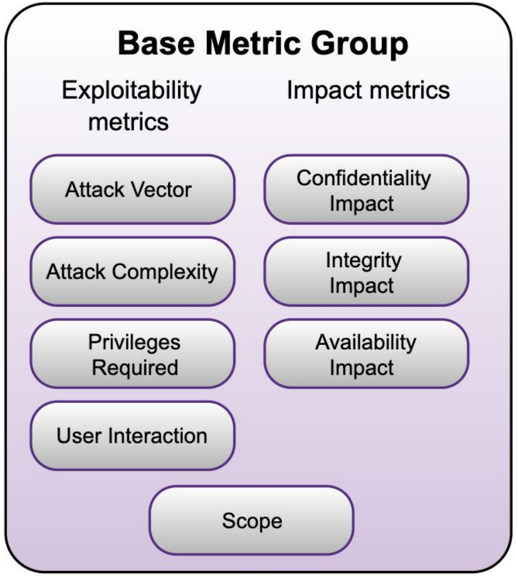
The Solution

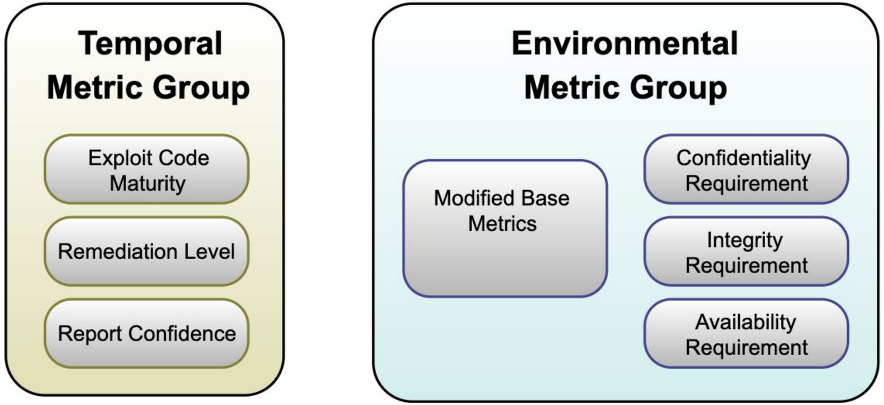
Exploitability, Reachability, Environment Context

Vulnerability Severity (CVSS)

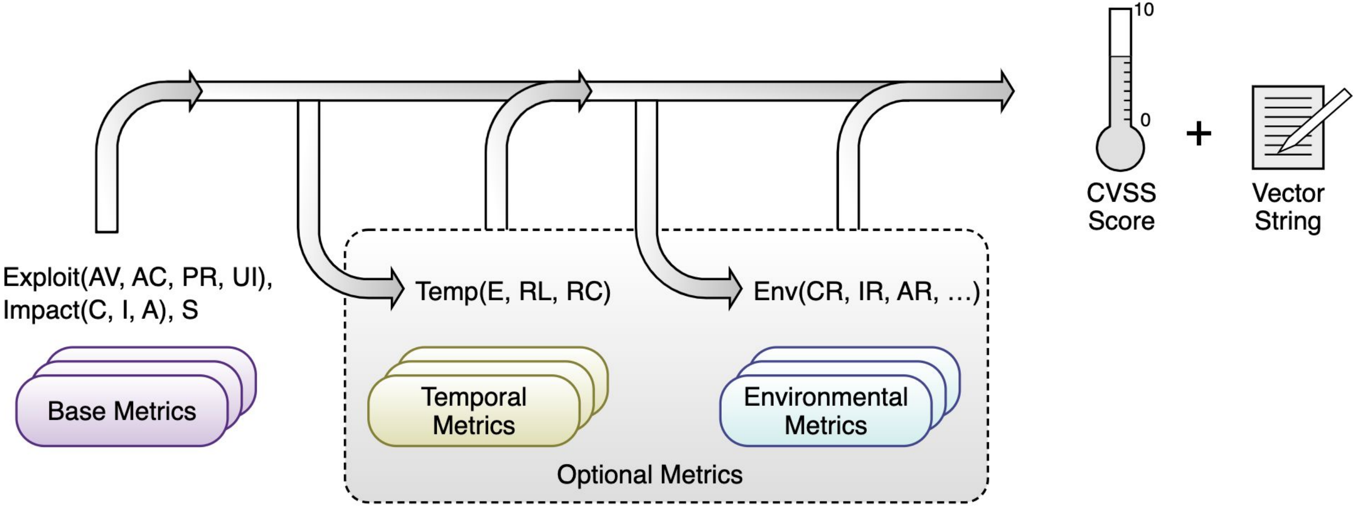
- **Common Vulnerability Scoring System** (CVSS)
- **Not a Risk Score!**
- **(Technical) severity of vulnerability**
- **Need temporal and environment context:**

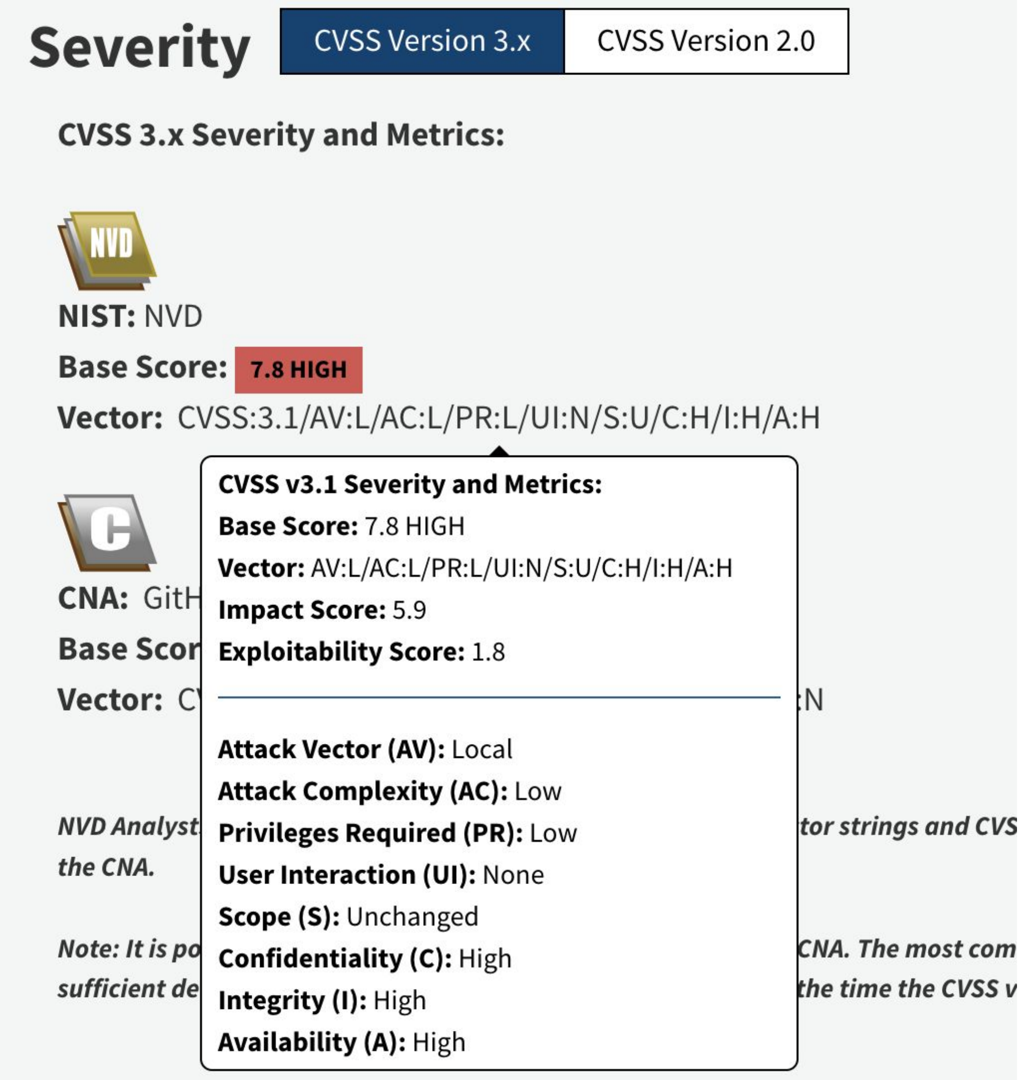
> “**Consumers of CVSS should supplement the Base Score with Temporal and Environmental Scores specific to their use of the vulnerable product to produce a severity more accurate for their organizational environment**”





Rating	CVSS Score
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0





Severity

CVSS 3.x Severity and Metrics:

NIST: NVD

Base Score: 7.8 HIGH

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

CVSS v3.1 Severity and Metrics:

Base Score: 7.8 HIGH

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

Impact Score: 5.9

Exploitability Score: 1.8

Attack Vector (AV): Local

Attack Complexity (AC): Low

Privileges Required (PR): Low

User Interaction (UI): None

Scope (S): Unchanged

Confidentiality (C): High

Integrity (I): High

Availability (A): High

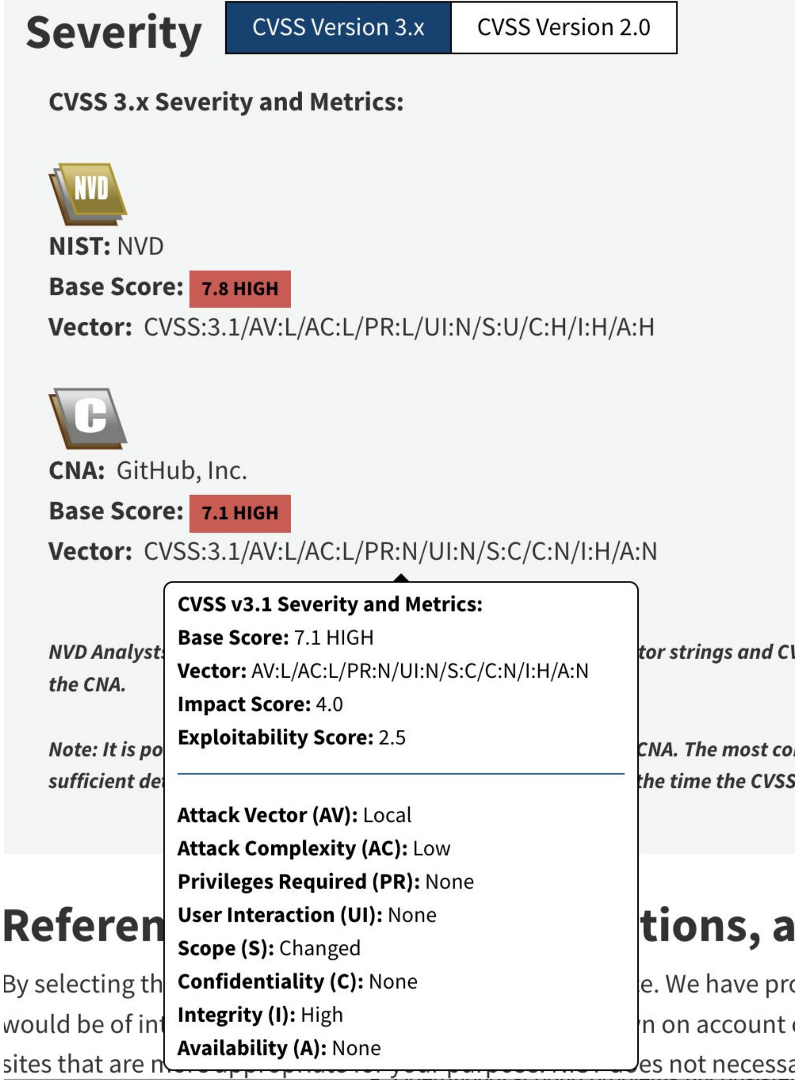
CNA: GitH

Base Scor

Vector: C :N

NVD Analyst tor strings and CVS
the CNA.

Note: It is po CNA. The most com
sufficient de the time the CVSS v



Severity

CVSS 3.x Severity and Metrics:

NIST: NVD

Base Score: 7.8 HIGH

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

CNA: GitHub, Inc.

Base Score: 7.1 HIGH

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

CVSS v3.1 Severity and Metrics:

Base Score: 7.1 HIGH

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

Impact Score: 4.0

Exploitability Score: 2.5

Attack Vector (AV): Local

Attack Complexity (AC): Low

Privileges Required (PR): None

User Interaction (UI): None

Scope (S): Changed

Confidentiality (C): None

Integrity (I): High

Availability (A): None

NVD Analysts tor strings and C
the CNA.

Note: It is po CNA. The most co
sufficient de the time the CVSS

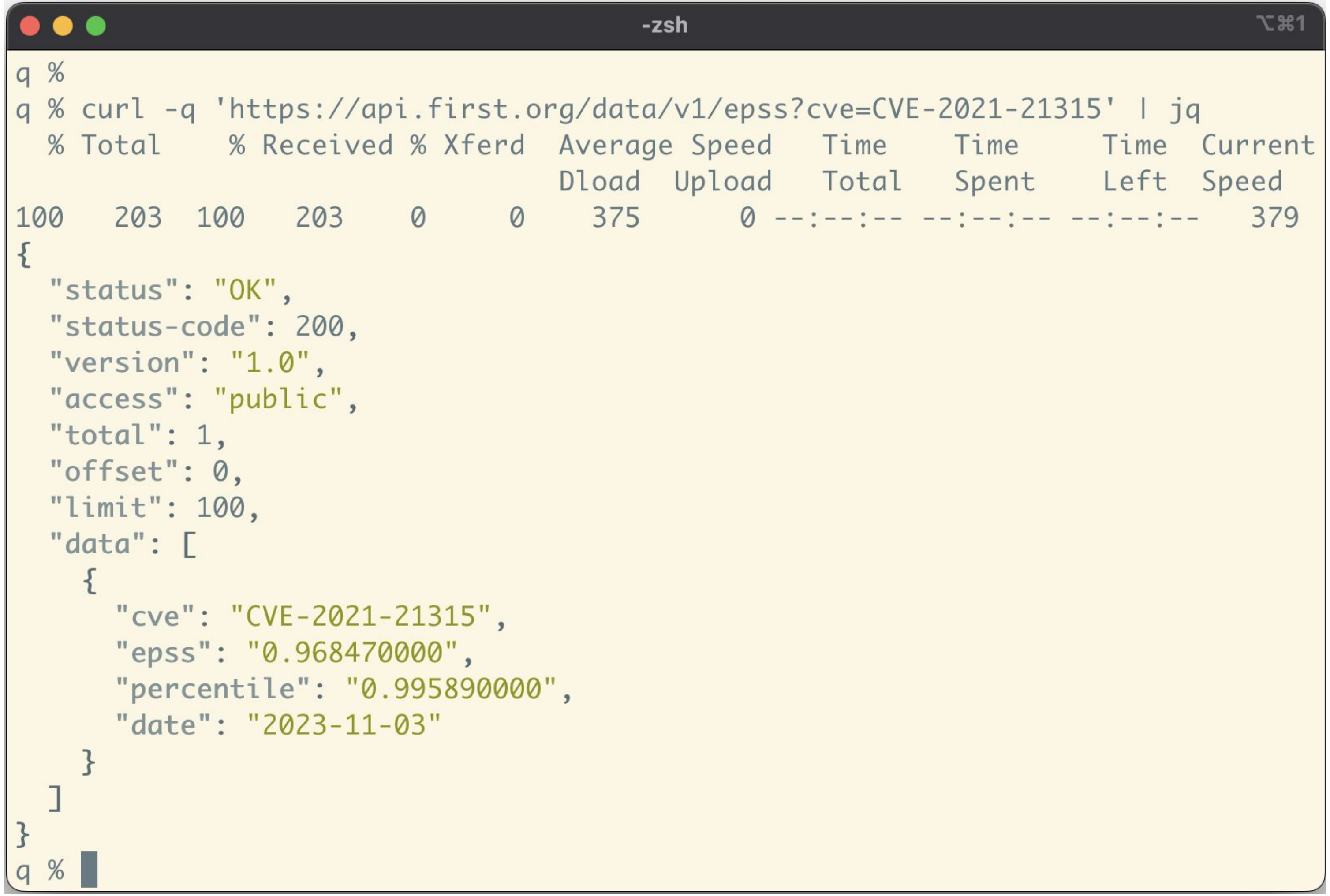
Referen tions, a

By selecting th e. We have pro
would be of int n on account o
sites that are m es not necessa

Exploitability
(CVSS Temporal/EPSS/KEV)

EPSS

- **Exploit Prediction Scoring System** (EPSS)
- **Probability vulnerability will be exploited in the next 30 days (0 -> 1)**
- **Key input data - exploit attempt logs from various vendors**
- **Not a Risk Score! (still need environment context)**
- **Pre-threat intel**
- **Has flaws/gaps (require IDS signatures, better for certain types of infra)**
- **Must have a CVE ID**
- **Not using known exploited vulnerability data intentionally**



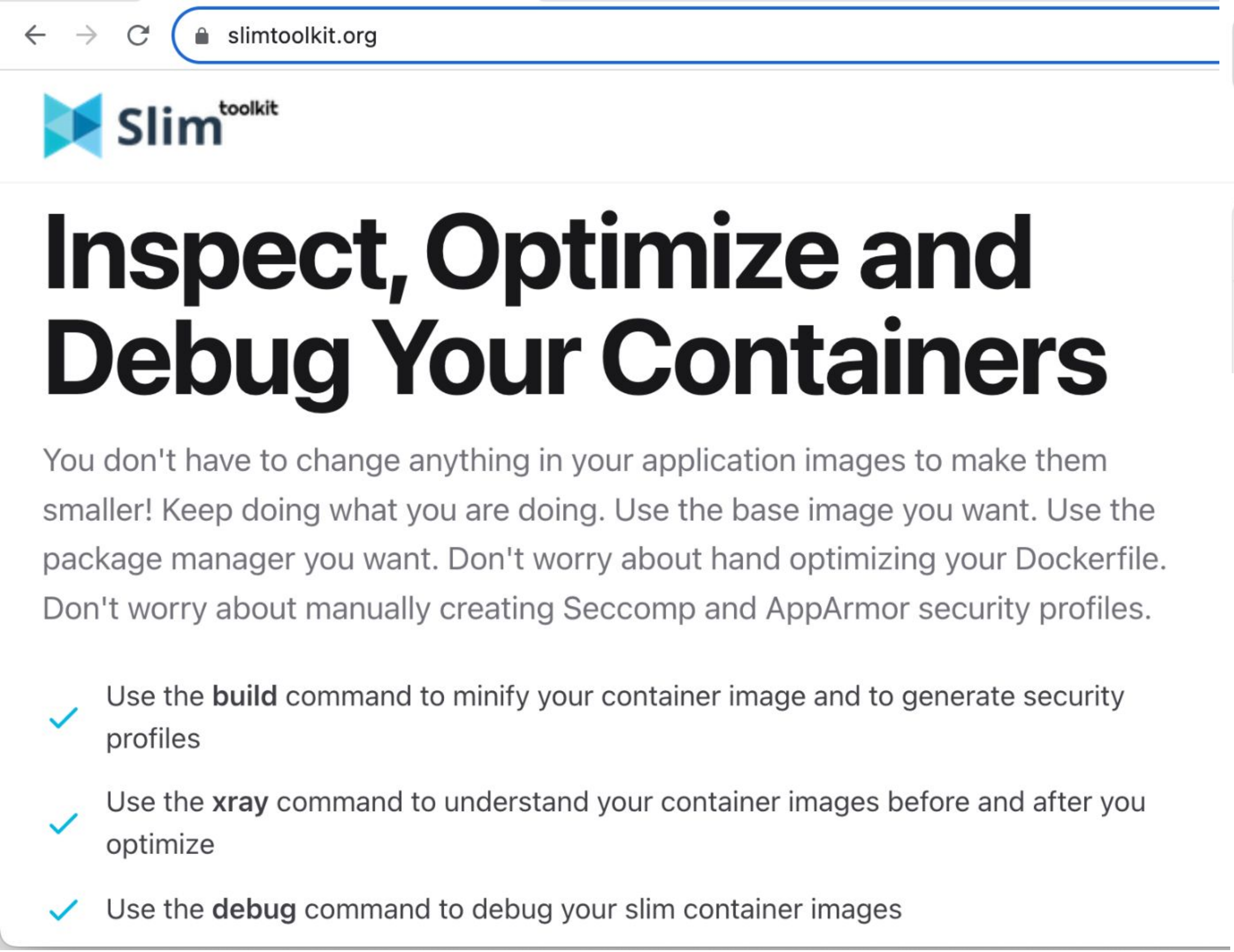
```
q %
q % curl -q 'https://api.first.org/data/v1/epss?cve=CVE-2021-21315' | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   203  100   203    0     0    375      0 --:--:-- --:--:-- --:--:--   379
{
  "status": "OK",
  "status-code": 200,
  "version": "1.0",
  "access": "public",
  "total": 1,
  "offset": 0,
  "limit": 100,
  "data": [
    {
      "cve": "CVE-2021-21315",
      "epss": "0.968470000",
      "percentile": "0.995890000",
      "date": "2023-11-03"
    }
  ]
}
q % 
```

KEV

- **Known Exploited Vulnerabilities** (KEV)
- **CISA**
- https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- **Useful, but incomplete (vs commercial threat intel)**
- **Also includes attempted exploits (not just successful)**
- **Must have a CVE ID (and need to have a fix/mitigation)**

Reachability (Static/Runtime/Network)

SlimToolkit



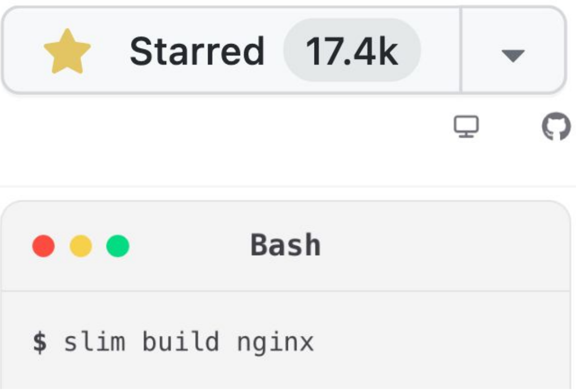
slimtoolkit.org

Slim toolkit

Inspect, Optimize and Debug Your Containers

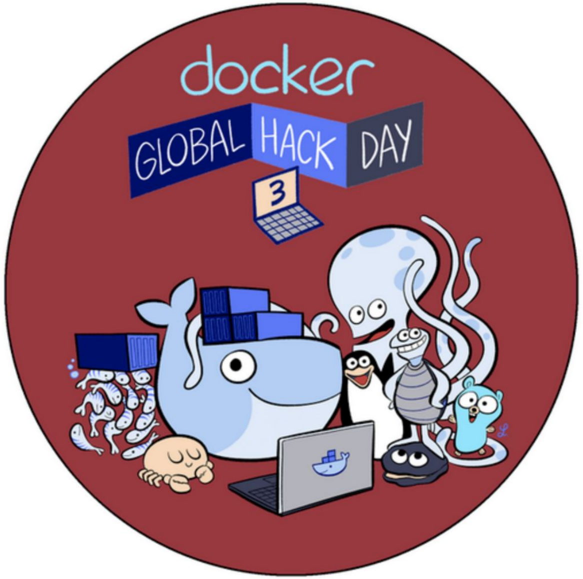
You don't have to change anything in your application images to make them smaller! Keep doing what you are doing. Use the base image you want. Use the package manager you want. Don't worry about hand optimizing your Dockerfile. Don't worry about manually creating Seccomp and AppArmor security profiles.

- ✓ Use the **build** command to minify your container image and to generate security profiles
- ✓ Use the **xray** command to understand your container images before and after you optimize
- ✓ Use the **debug** command to debug your slim container images

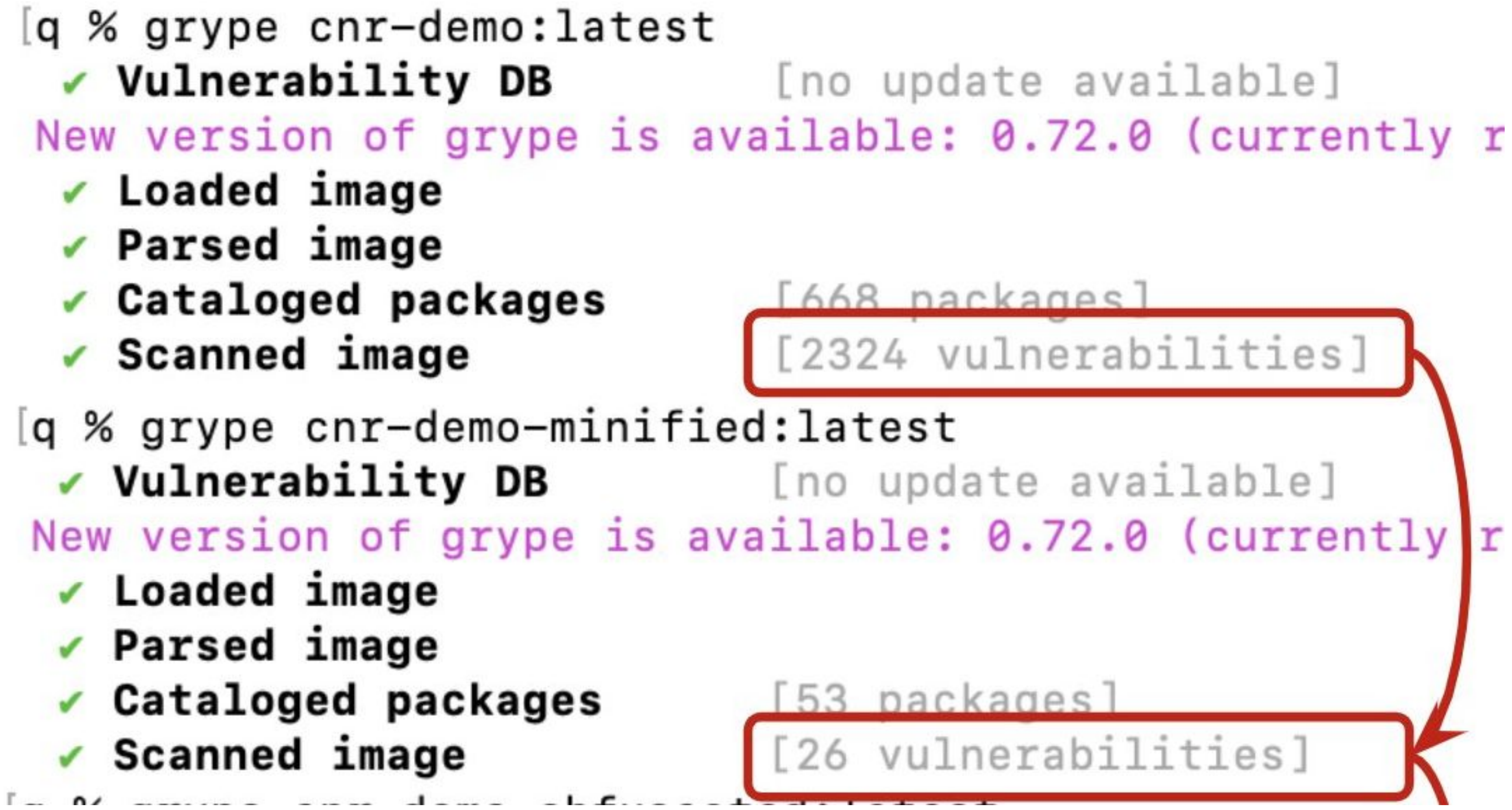


Bash

```
$ slim build nginx
```



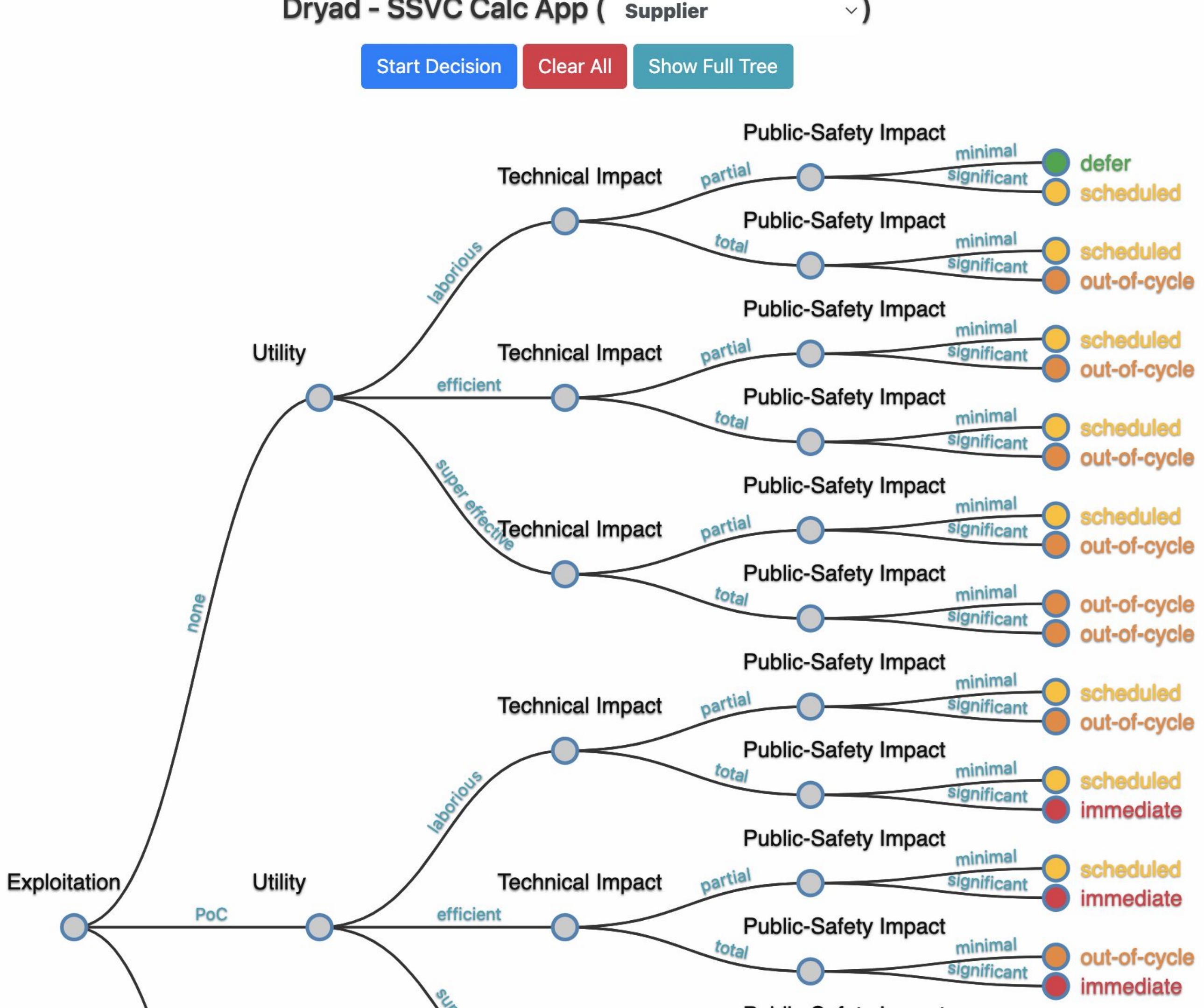
- **Runtime reachability with SlimToolkit**
  - **Minify container with “slim build” and then rescan** (simple option)
  - **Demo with Grype** (2324 vulns (before) -> 26 vulns (after))
  - **Alternative: “slim profile”** (requires additional post-processing on the container report file, creport.json)



```
[q % grype cnr-demo:latest
  ✔ Vulnerability DB          [no update available]
 New version of grype is available: 0.72.0 (currently r
  ✔ Loaded image
  ✔ Parsed image
  ✔ Cataloged packages        [668 packages]
  ✔ Scanned image             [2324 vulnerabilities]
[q % grype cnr-demo-minified:latest
  ✔ Vulnerability DB          [no update available]
 New version of grype is available: 0.72.0 (currently r
  ✔ Loaded image
  ✔ Parsed image
  ✔ Cataloged packages        [53 packages]
  ✔ Scanned image             [26 vulnerabilities]
[q % grype cnr-demo-obfuscated:latest
```

Connect All with Decision Trees (SSVC)

- **Stakeholder-Specific Vulnerability Categorization** (SSVC)
- **CMU & CISA**
- **Stakeholder-Specific Decision Trees**
- **Exploitability-centric**
- **Custom Decision Trees (for best results)**


Dryad - SSVC Calc App (Supplier)
Start Decision
Clear All
Show Full Tree
Exploitation
none
PoC
Utility
laborious
efficient
super effective
Technical Impact
partial
total
Public-Safety Impact
minimal
significant
defer
scheduled
scheduled
out-of-cycle
scheduled
out-of-cycle
scheduled
out-of-cycle
scheduled
out-of-cycle
out-of-cycle
out-of-cycle
Utility
laborious
efficient
Technical Impact
partial
total
Public-Safety Impact
minimal
significant
scheduled
out-of-cycle
scheduled
immediate
scheduled
immediate
out-of-cycle
immediate

SBOM and VEX

Demo

Key Takeaways

Thank You

https://twitter.com/kcqon

https://github.com/kcq

Tools

- **EPSS**
  - https://holisticinfosec.shinyapps.io/epsscall
  - curl 'https://api.first.org/data/v1/epss?cve=CVE_ID'
- **CVSS**
  - https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator
- **SSVC**
  - https://certcc.github.io/SSVC/ssvc-calc/
- **KEV**
  - https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- **Enhanced CVE Info**
  - https://www.cvedetails.com/cve/CVE_ID
- **CVE Prioritizer**
  - https://github.com/TURROKS/CVE_Prioritizer