cloud-native software supply chain security: the hard truth

Daniel Drack

☝️ Disclaimer

- I‘m no (hardcore) security guy
- Observations from a cloud native consultant POV

🎯 My Goal

- Provide ideas about cloud native software supply chain bp
  - cheap + expensive examples

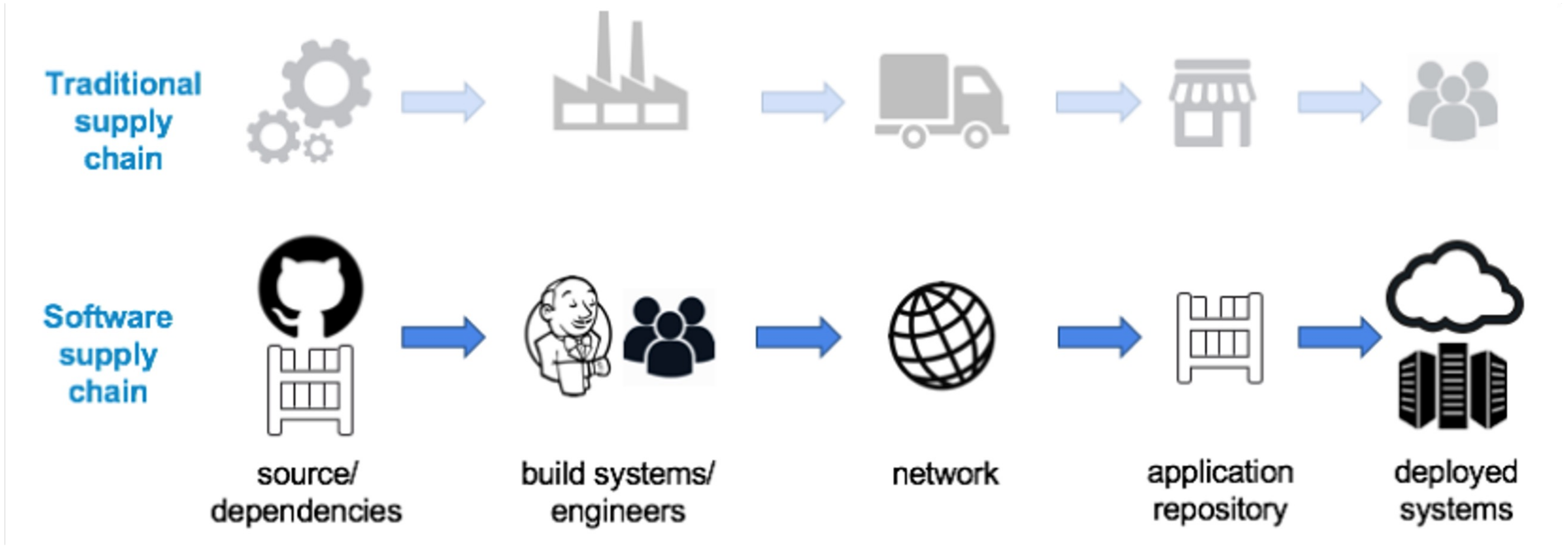
- Share my concerns 😅

Software Supply Chain

The software supply chain involves a multitude of tools and processes that enable software developers to write, build, and ship applications.

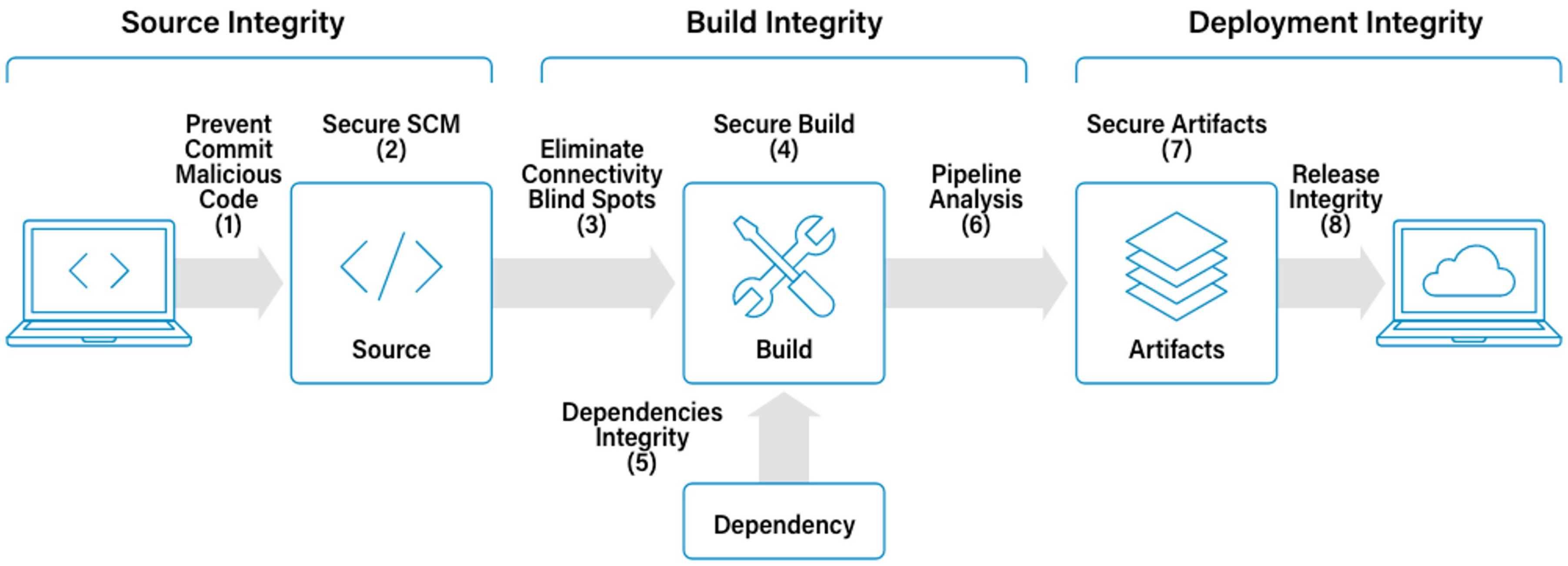
Melara & Bowman, 2022, Intel Labs

CNCF - SSC in a 🥥

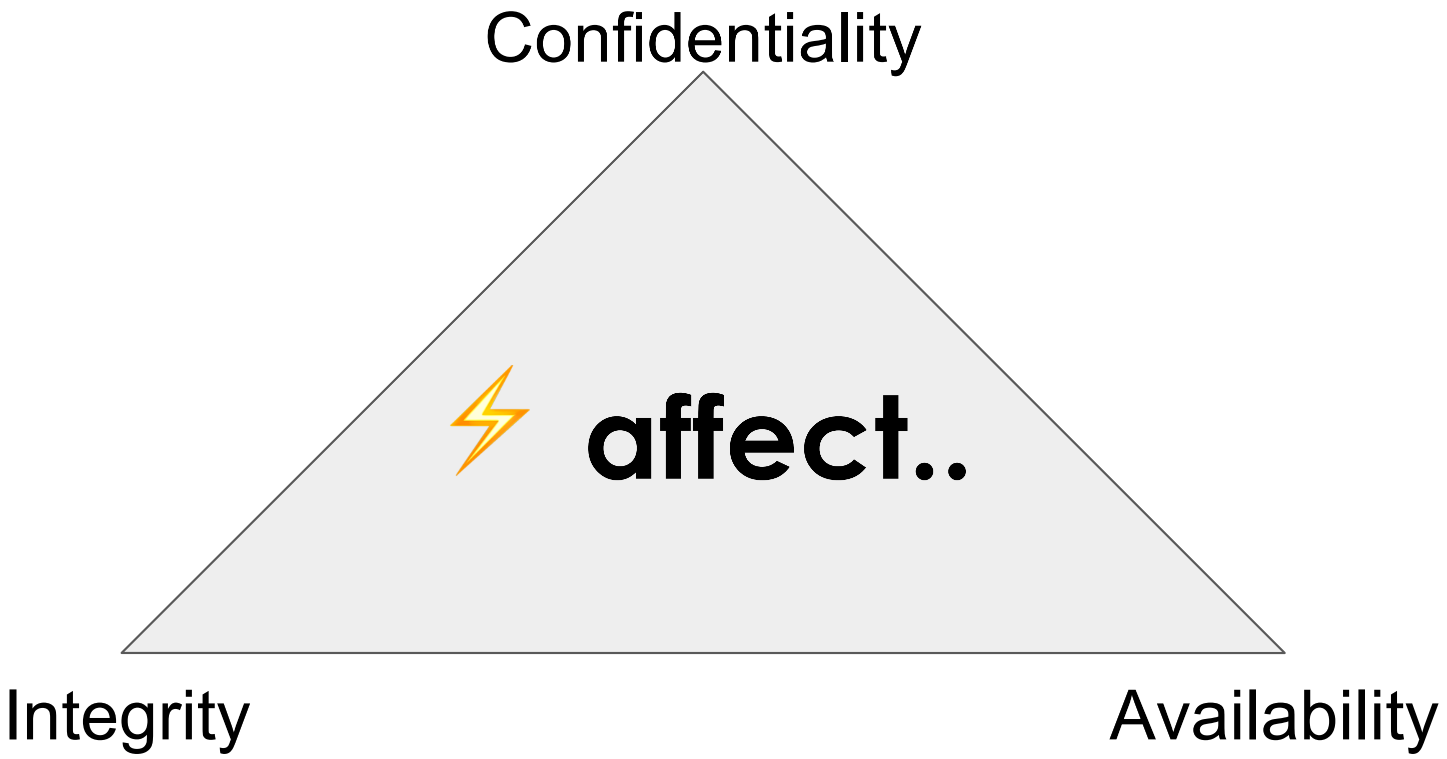


https://github.com/cncf/tag-security/blob/main/supply-chain-security/supply-chain-security-paper/CNCF_SSCP_v1.pdf

CIS - SSC ⚡ in a 🥥



https://www.cisecurity.org/insights/white-papers/cis-software-supply-chain-security-guide


Confidentiality
⚡ affect..
Integrity
Availability

Stages of the SSC

Stages/Elements of the SSC

- Self Written Code
- Dependencies
- Build
- Artifacts & Distribution/Deployment
- (Runtime)

Stage: Our Code

code content

code management

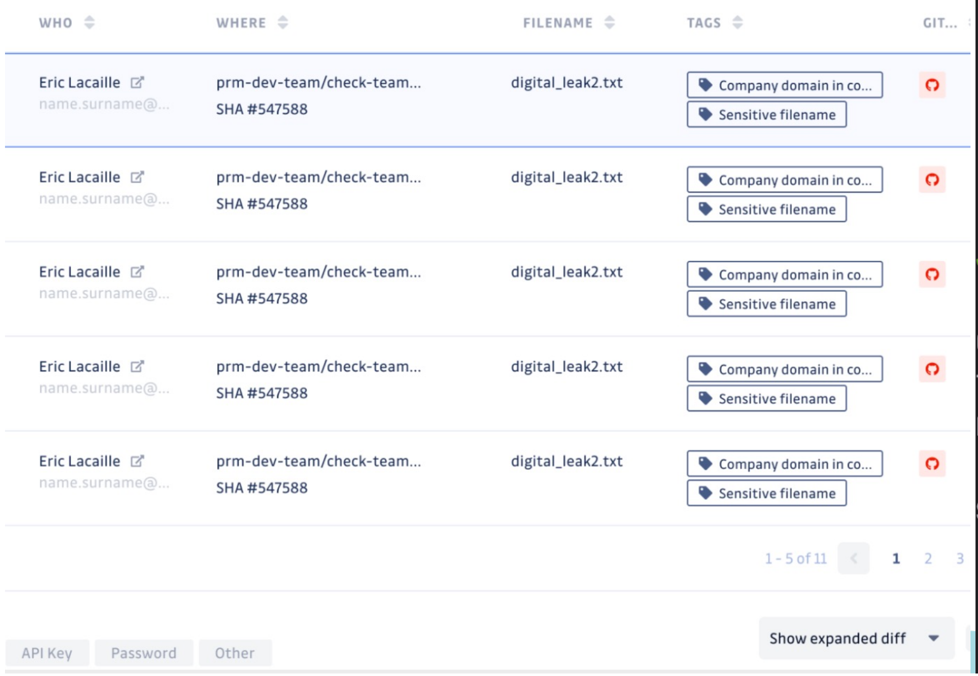
Stage: Our Code - code content

⚡ threats

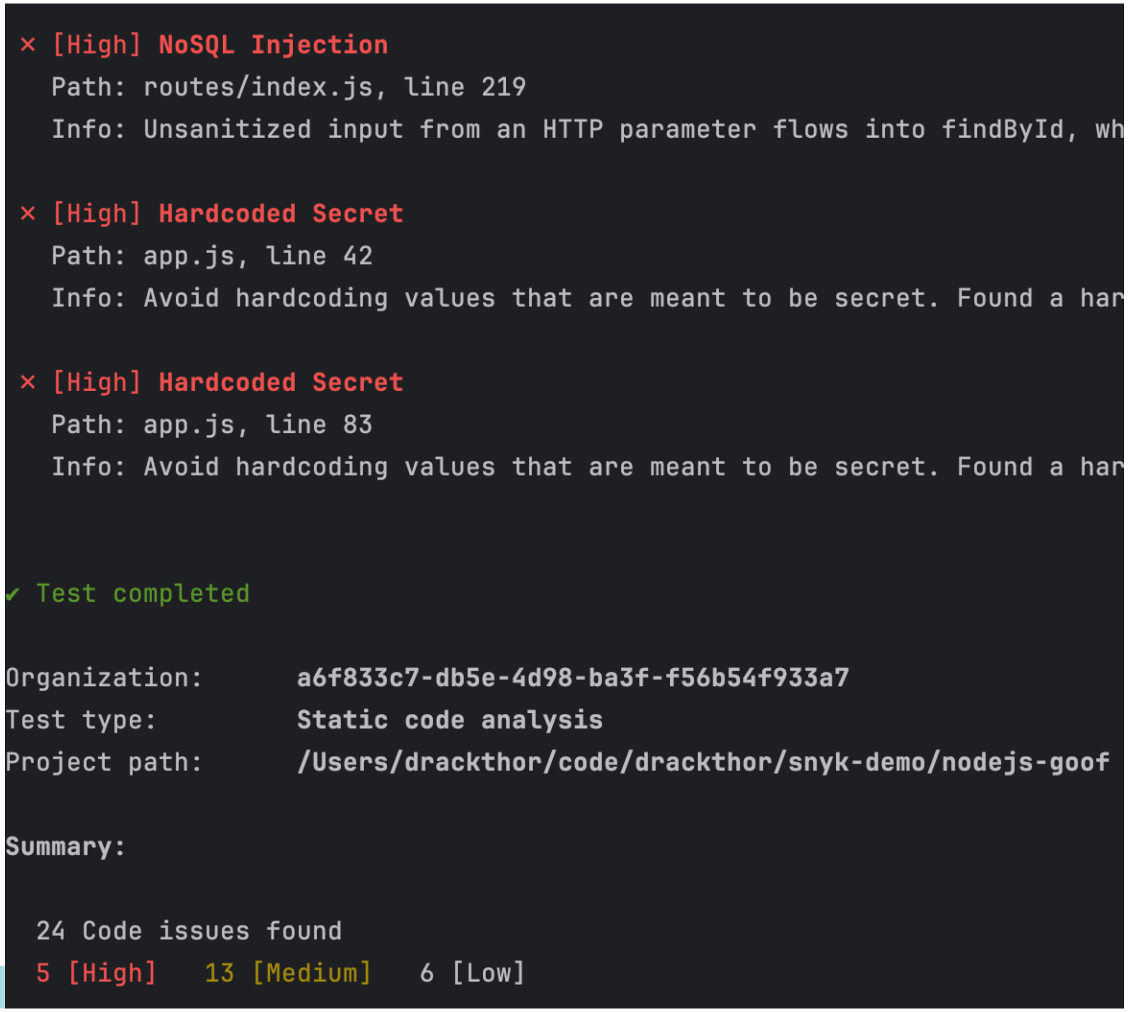
🪲 - bugs
☣️ - malicious code
🔑 - secrets

Our Code - [cheap] scanners

SAST code scanners
Secret detection scanners
IaC code scanners



WHO	WHERE	FILENAME	TAGS	GIT...
Eric Lacaille name.surname@...	prm-dev-team/check-team... SHA #547588	digital_leak2.txt	Company domain in co... Sensitive filename	
Eric Lacaille name.surname@...	prm-dev-team/check-team... SHA #547588	digital_leak2.txt	Company domain in co... Sensitive filename	
Eric Lacaille name.surname@...	prm-dev-team/check-team... SHA #547588	digital_leak2.txt	Company domain in co... Sensitive filename	
Eric Lacaille name.surname@...	prm-dev-team/check-team... SHA #547588	digital_leak2.txt	Company domain in co... Sensitive filename	
Eric Lacaille name.surname@...	prm-dev-team/check-team... SHA #547588	digital_leak2.txt	Company domain in co... Sensitive filename	

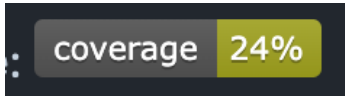


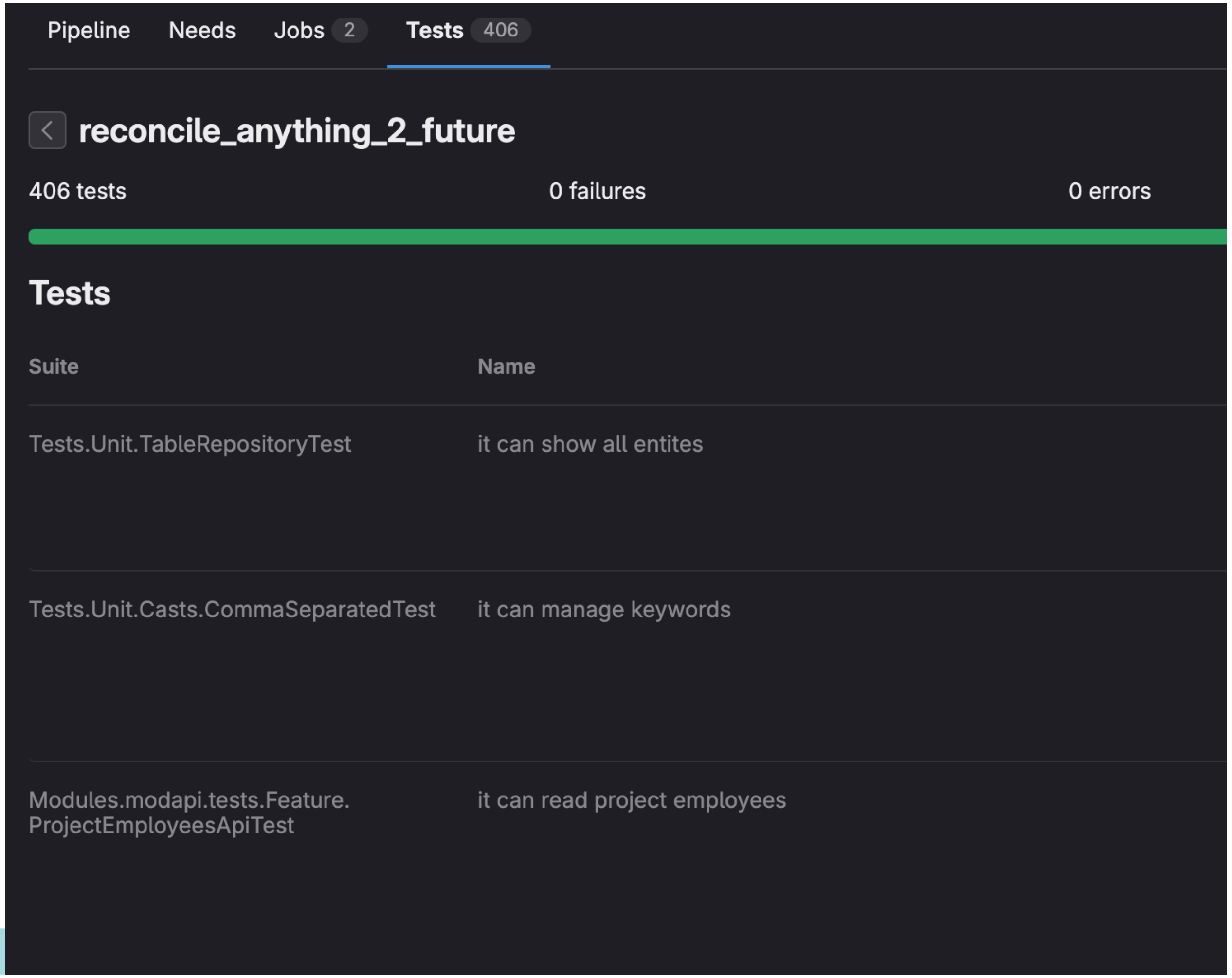
```
 ✗ [High] NoSQL Injection
   Path: routes/index.js, line 219
   Info: Unsanitized input from an HTTP parameter flows into findById, wh

 ✗ [High] Hardcoded Secret
   Path: app.js, line 42
   Info: Avoid hardcoding values that are meant to be secret. Found a har

 ✗ [High] Hardcoded Secret
   Path: app.js, line 83
   Info: Avoid hardcoding values that are meant to be secret. Found a har


✔ Test completed

Organization:      a6f833c7-db5e-4d98-ba3f-f56b54f933a7
Test type:         Static code analysis
Project path:      /Users/drackthor/code/drackthor/snyk-demo/nodejs-goof

Summary:

  24 Code issues found
  5 [High]   13 [Medium]   6 [Low]
```

Our Code - [expensive] tests

Unit Tests
System Tests
E2E Tests
Trace-Tests

Code Coverage





reconcile_anything_2_future

406 tests 0 failures 0 errors

Tests

Suite	Name
Tests.Unit.TableRepositoryTest	it can show all entites
Tests.Unit.Casts.CommaSeparatedTest	it can manage keywords
Modules.modapi.tests.Feature. ProjectEmployeesApiTest	it can read project employees

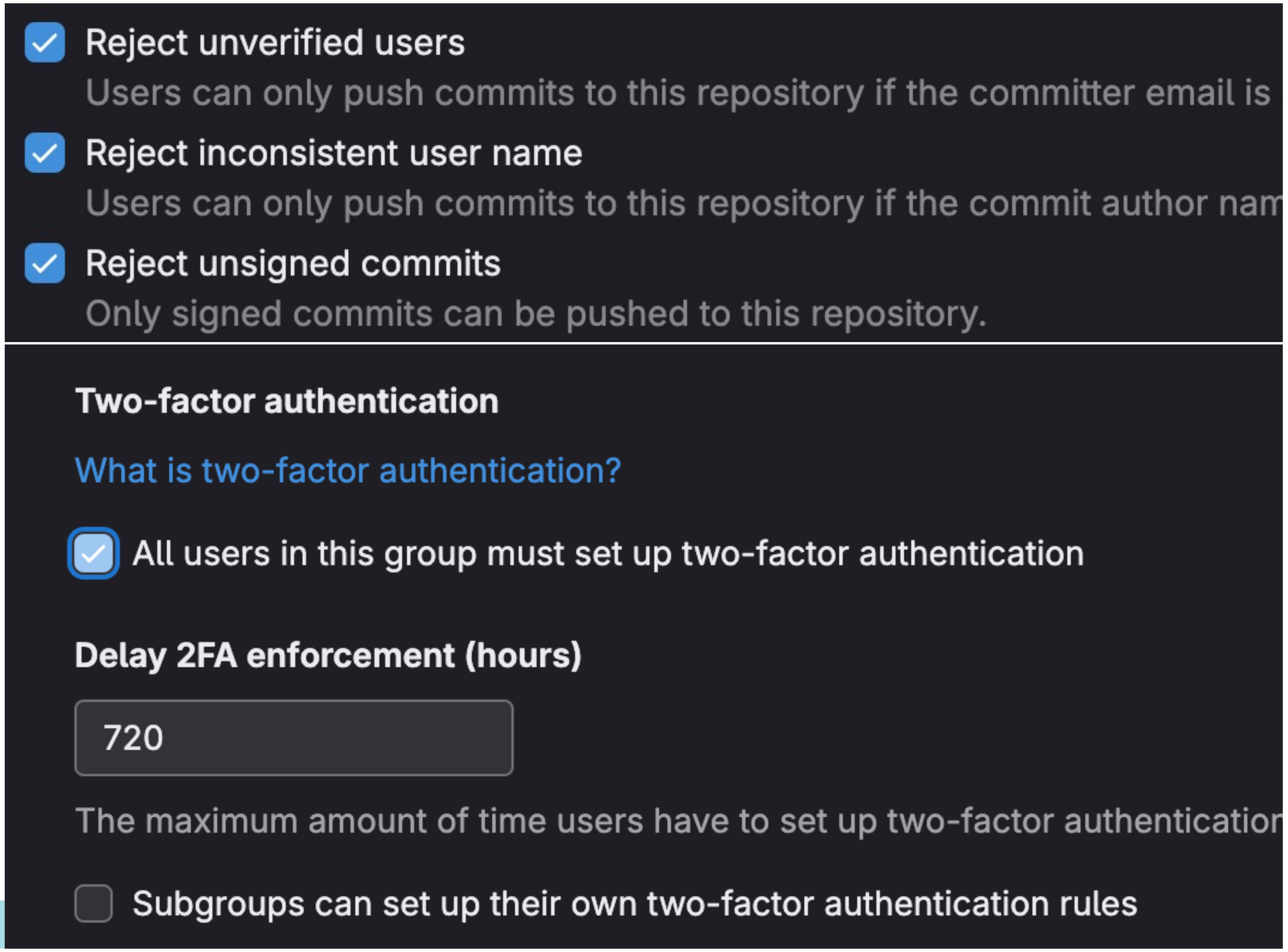
Stage: Code - code management

⚡ threats

⌨️ - manipulation
🥷 - theft
🗑️ - deletion

Our Code – [cheap]

Mandatory
Signed Commits
--
Mandatory MFA



[✓] Reject unverified users
Users can only push commits to this repository if the committer email is

[✓] Reject inconsistent user name
Users can only push commits to this repository if the commit author nam

[✓] Reject unsigned commits
Only signed commits can be pushed to this repository.

Two-factor authentication

What is two-factor authentication?

[✓] All users in this group must set up two-factor authentication

Delay 2FA enforcement (hours)

720

The maximum amount of time users have to set up two-factor authenticatio

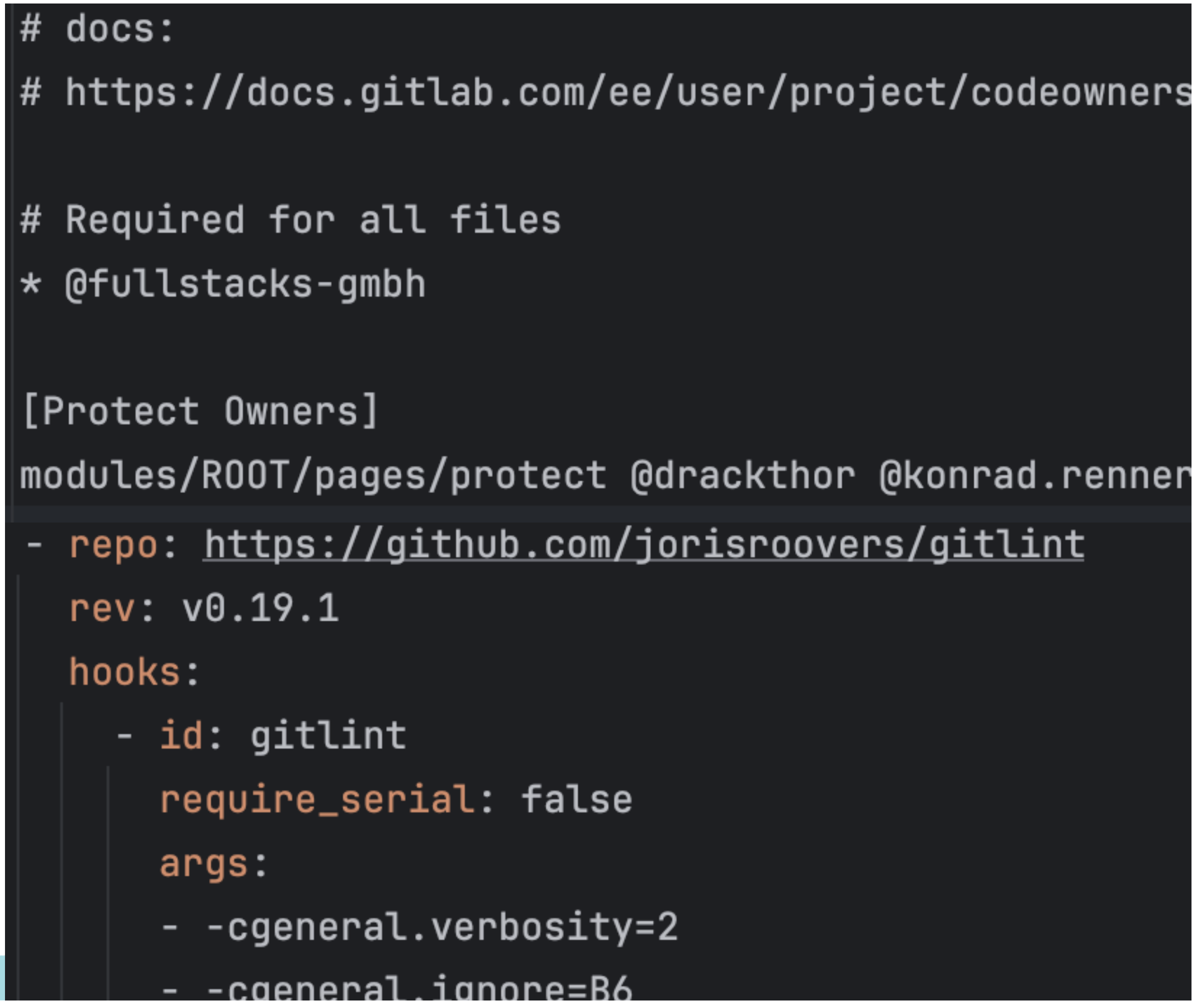
[ ] Subgroups can set up their own two-factor authentication rules

Our Code – [expensive]

CODEOWNERS

--

Pre-Commit



```
# docs:
# https://docs.gitlab.com/ee/user/project/codeowners

# Required for all files
* @fullstacks-gmbh

[Protect Owners]
modules/ROOT/pages/protect @drackthor @konrad.renner
```

```
- repo: https://github.com/jorisroovers/gitlint
  rev: v0.19.1
  hooks:
    - id: gitlint
      require_serial: false
      args:
        - -cgeneral.verbosity=2
        - -cgeneral.ignore=B6
```

Stage: Our Code - show of ✋

mandatory MFA for source code access

—

Pre-Commit or Push-Policy in place

Stage: Dependencies

packages, libraries, base-images,..

Please use a Package Manager

Stage: Dependencies

⚡ threats

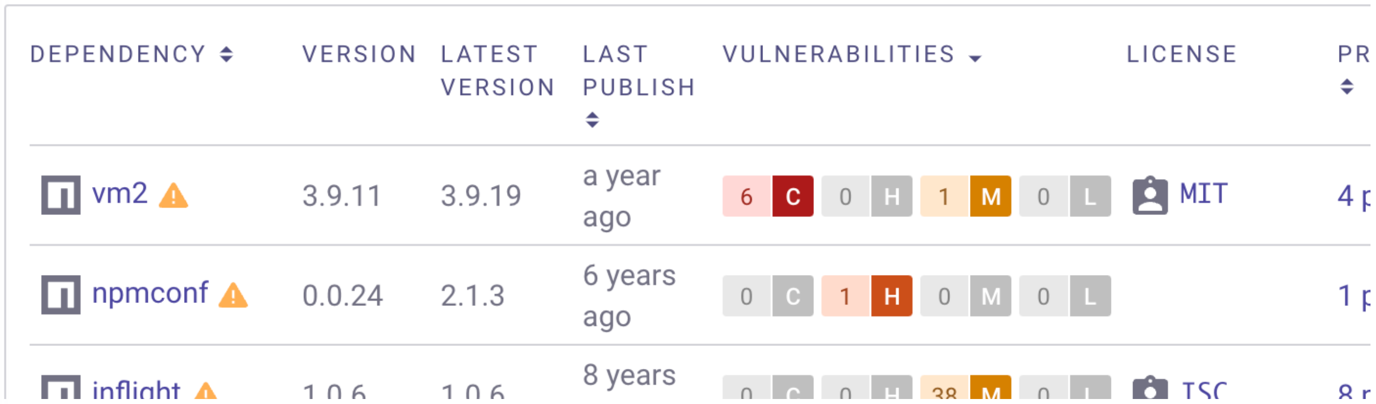
🪲 - bugs
☣️ - malicious code
⚖️ - license
🎭 - integrity

Dependencies – [cheap]

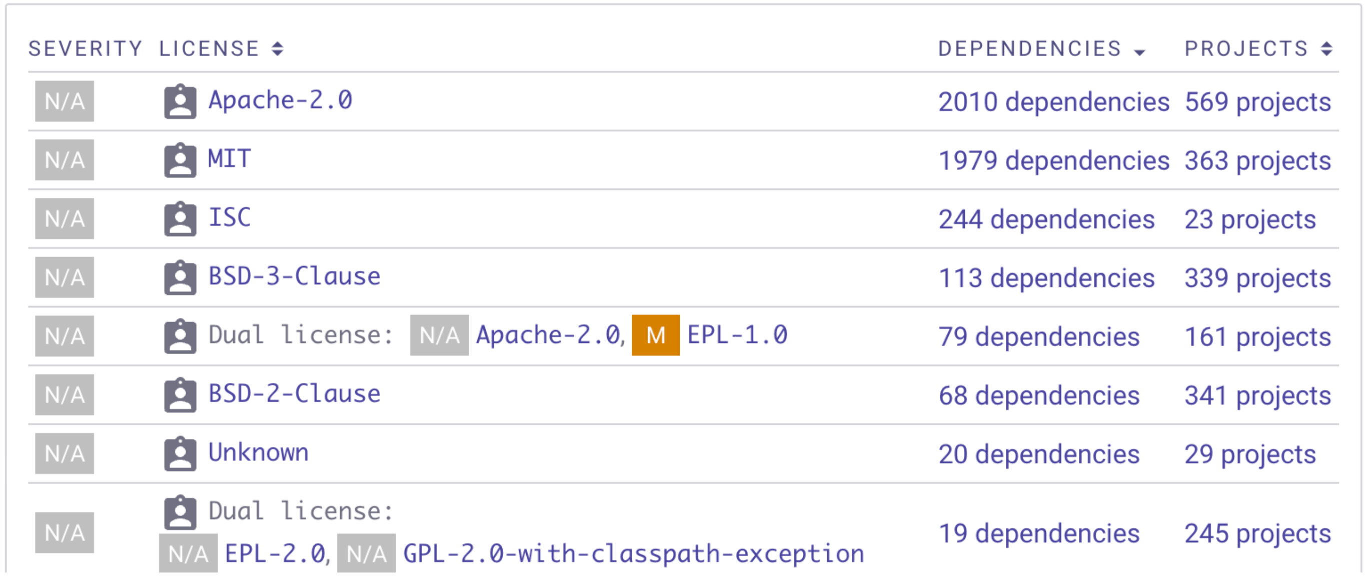
Inventory

--

License Checks



DEPENDENCY	VERSION	LATEST VERSION	LAST PUBLISH	VULNERABILITIES	LICENSE	PR
vm2 ⚠	3.9.11	3.9.19	a year ago	6 C 0 H 1 M 0 L	MIT	4 p
npmconf ⚠	0.0.24	2.1.3	6 years ago	0 C 1 H 0 M 0 L		1 p
inflight ⚠	1.0.6	1.0.6	8 years	0 C 0 H 38 M 0 L	ISC	8 r



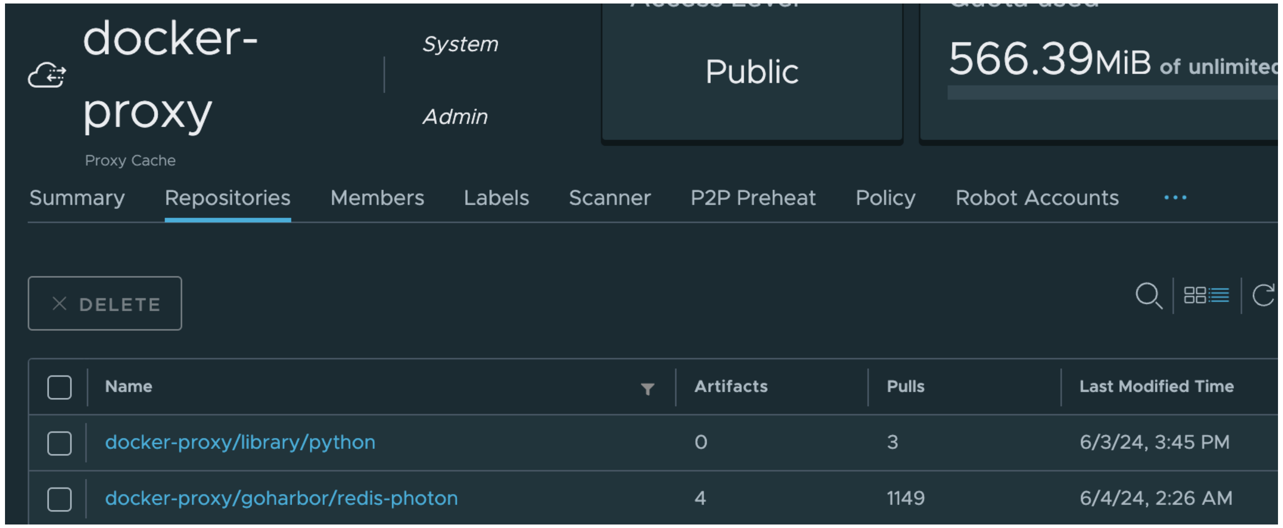
SEVERITY	LICENSE	DEPENDENCIES	PROJECTS
N/A	Apache-2.0	2010 dependencies	569 projects
N/A	MIT	1979 dependencies	363 projects
N/A	ISC	244 dependencies	23 projects
N/A	BSD-3-Clause	113 dependencies	339 projects
N/A	Dual license: N/A Apache-2.0, M EPL-1.0	79 dependencies	161 projects
N/A	BSD-2-Clause	68 dependencies	341 projects
N/A	Unknown	20 dependencies	29 projects
N/A	Dual license: N/A EPL-2.0, N/A GPL-2.0-with-classpath-exception	19 dependencies	245 projects

Dependencies – [expensive]

airgapping

--

require signed dependencies



docker-proxy

Proxy Cache

System | Admin

Public

566.39MiB of unlimite

Name	Artifacts	Pulls	Last Modified Time
docker-proxy/library/python	0	3	6/3/24, 3:45 PM
docker-proxy/goharbor/redis-photon	4	1149	6/4/24, 2:26 AM

```
<component group="com.github.javaparser" name="javaparser-core"
version="3.6.11">
   <artifact name="javaparser-core-3.6.11.jar">
      <pgp value="8756c4f765c9ac3cb6b85d62379ce192d401ab61"/>
   </artifact>
</component>
```

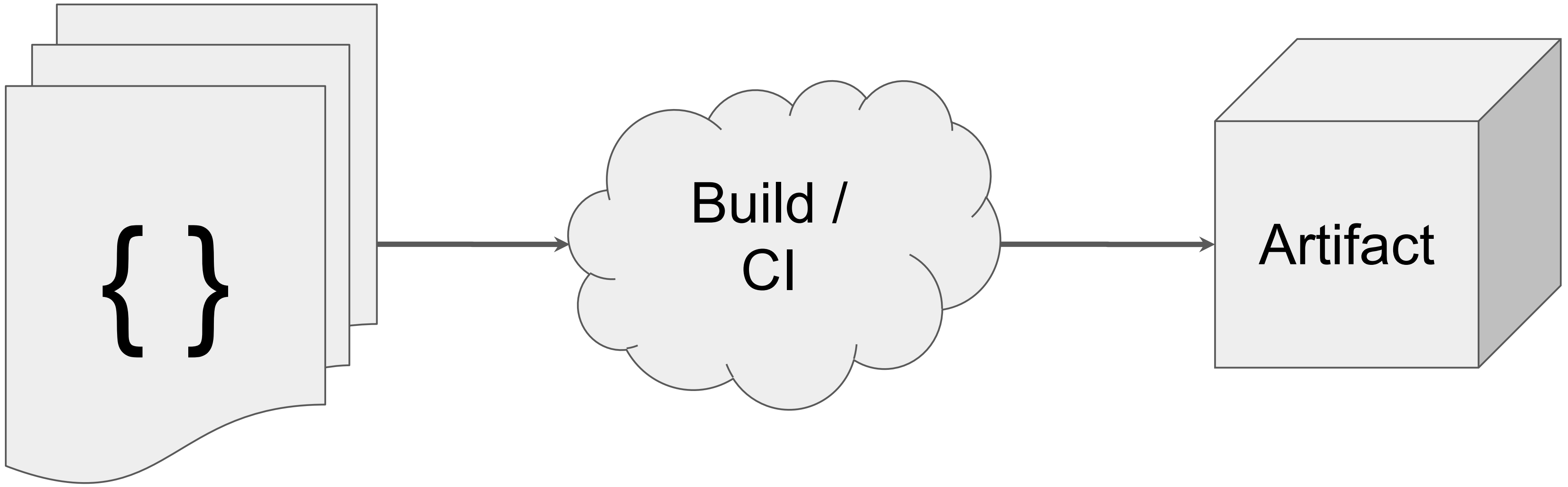
Stage: Dependencies - show of ✋

using a package manager

—

package usage policy in place

Stage: Build



Stage: Build

⚡ threats

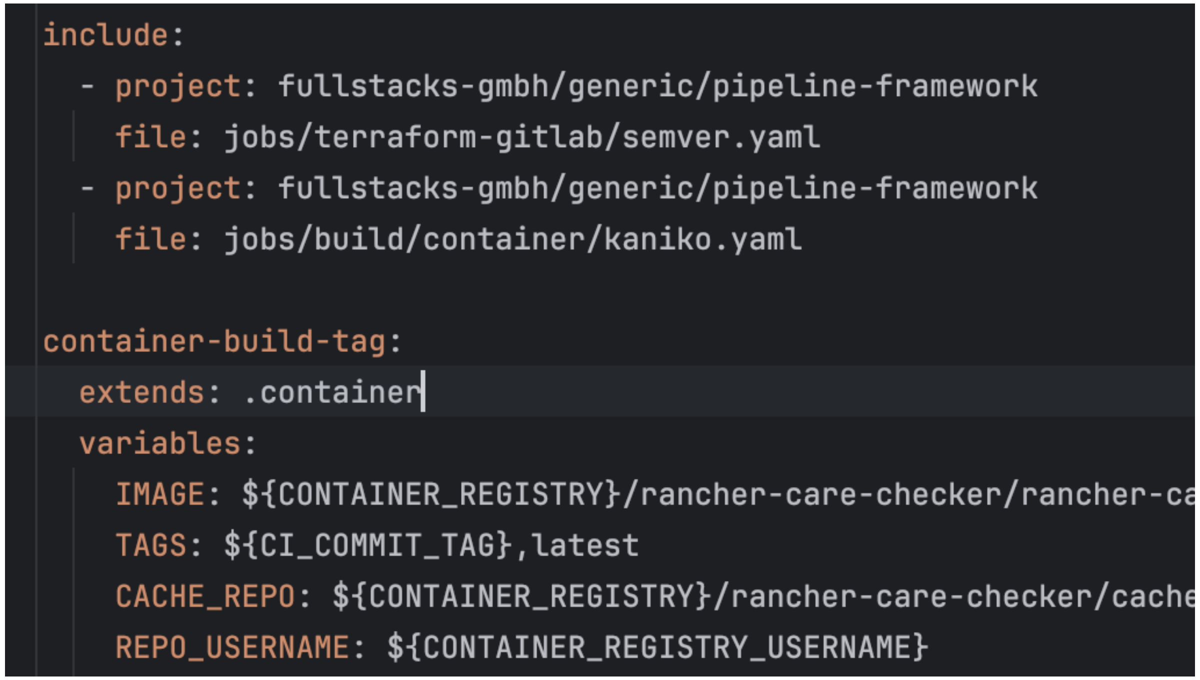
- 🪲 - build bugs
- ☣️ - malicious env

Build – [cheap]

Pipelines as Code

--

Dedicated Env



```yaml
include:
  - project: fullstacks-gmbh/generic/pipeline-framework
    file: jobs/terraform-gitlab/semver.yaml
  - project: fullstacks-gmbh/generic/pipeline-framework
    file: jobs/build/container/kaniko.yaml

container-build-tag:
  extends: .container
  variables:
    IMAGE: ${CONTAINER_REGISTRY}/rancher-care-checker/rancher-ca
    TAGS: ${CI_COMMIT_TAG},latest
    CACHE_REPO: ${CONTAINER_REGISTRY}/rancher-care-checker/cache
    REPO_USERNAME: ${CONTAINER_REGISTRY_USERNAME}
```

- Just don’t build on your 💻
- … and on the 💻 under your colleagues des.

Build – [expensive]

“[..] but, how do I troubleshoot my build now?”

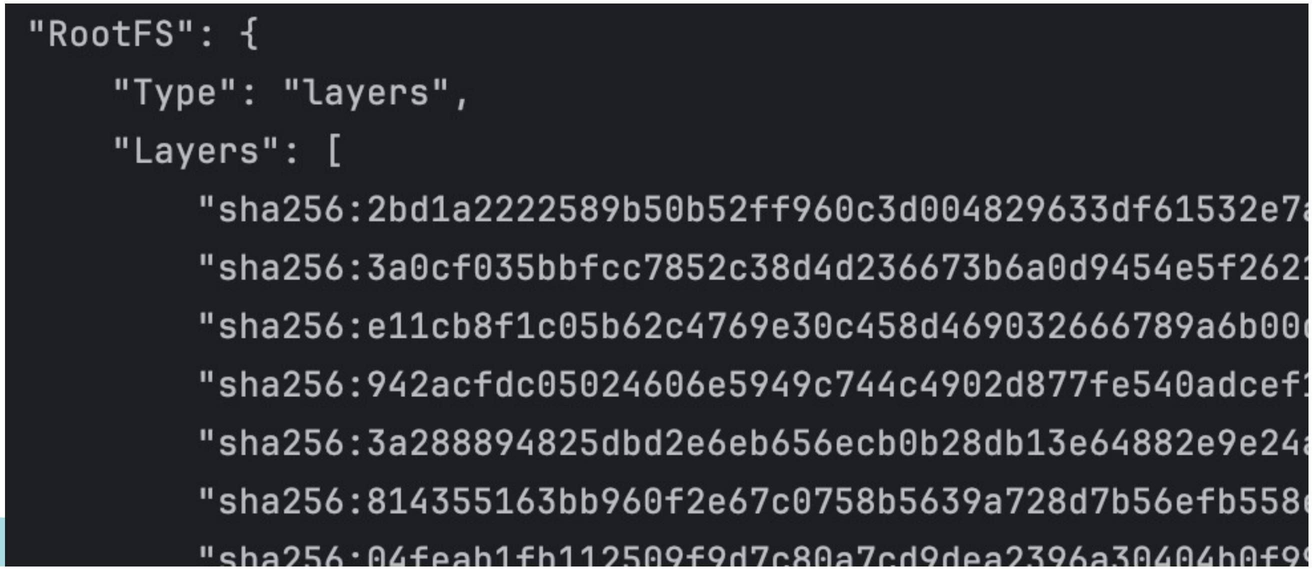
“[..] but, I’ve always had access
to the build machine”

“[..] wait what, no root access anymore 🤯”

Zero Trust

--

Reproducible Builds



```
"RootFS": {
    "Type": "layers",
    "Layers": [
        "sha256:2bd1a2222589b50b52ff960c3d004829633df61532e7
        "sha256:3a0cf035bbfcc7852c38d4d236673b6a0d9454e5f262
        "sha256:e11cb8f1c05b62c4769e30c458d469032666789a6b00
        "sha256:942acfdc05024606e5949c744c4902d877fe540adcef
        "sha256:3a288894825dbd2e6eb656ecb0b28db13e64882e9e24
        "sha256:814355163bb960f2e67c0758b5639a728d7b56efb558
        "sha256:04feab1fb112509f9d7c80a7cd9dea2396a30404b0f9
```

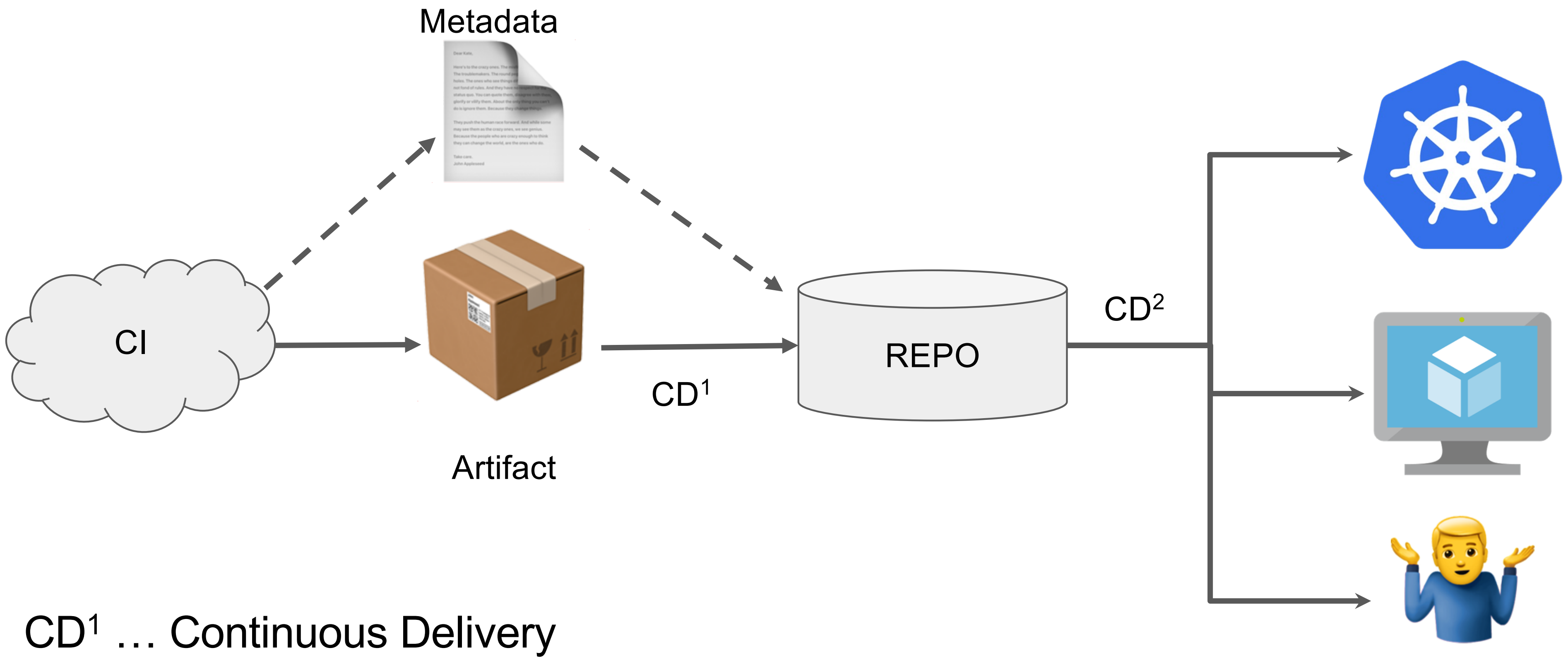
Stage: Build - show of ✋

fully automated build

—

truly bitwise reproducible builds

Stage: Artifacts & Distribution/Deployment



CD[1] … Continuous Delivery
CD[2] … Continuous Deployment

Stage: Artifacts & Distribution/Deployment

⚡ threats

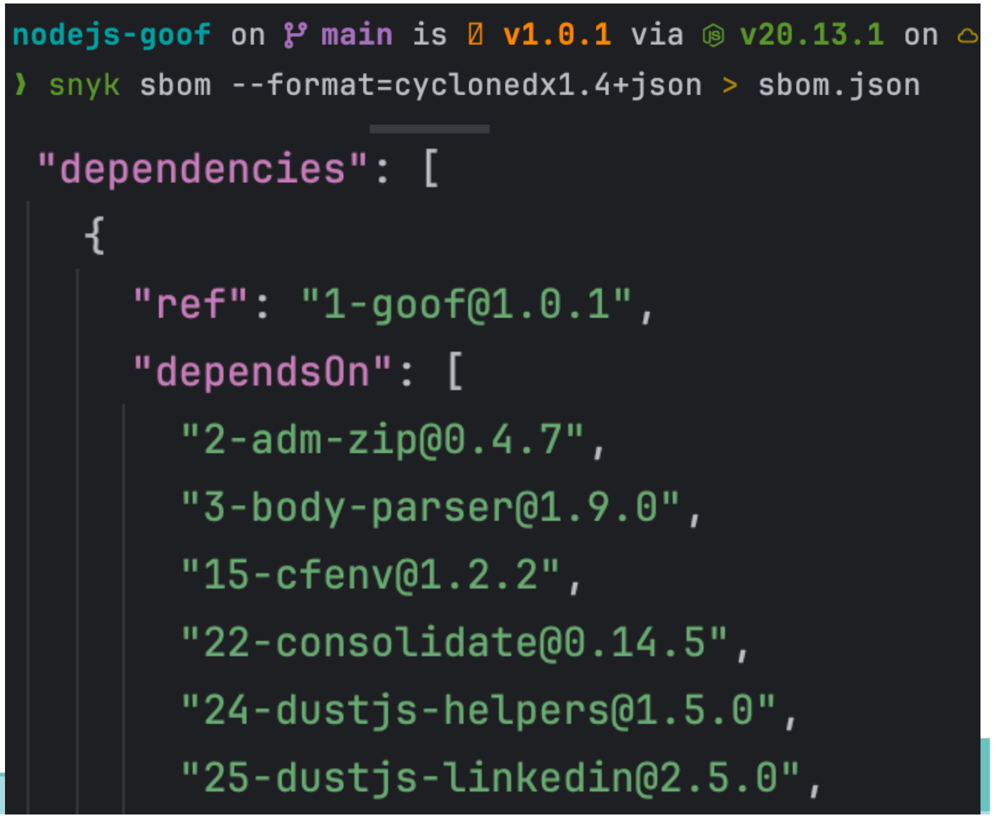
🥷 - theft / deletion
🔁 - replacement
🔍 - no transparency
📆 - updates

Artifact – [cheap]

Repo Security

👤 – RBAC
🤖 – Service Accounts
🔁 – cycle tokens/credentials
🔒 – MFA

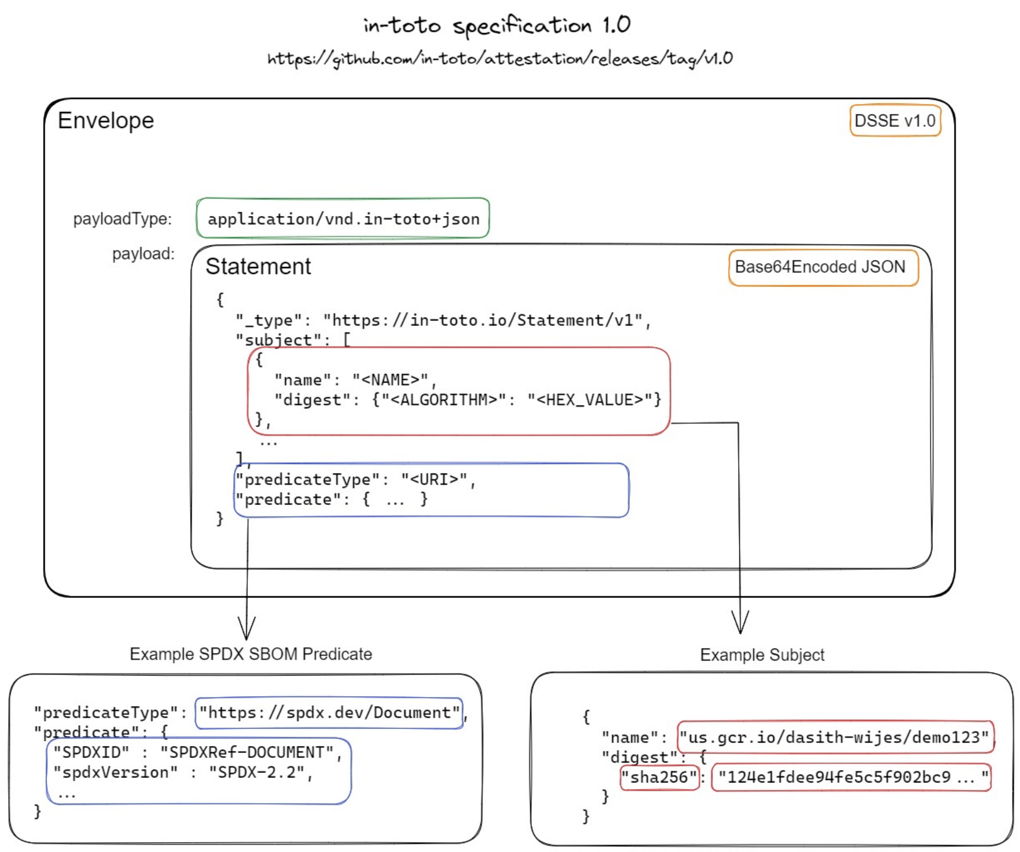
[cheap] SBOM



```
nodejs-goof on  main is  v1.0.1 via  v20.13.1 on 
❯ snyk sbom --format=cyclonedx1.4+json > sbom.json

"dependencies": [
  {
    "ref": "1-goof@1.0.1",
    "dependsOn": [
      "2-adm-zip@0.4.7",
      "3-body-parser@1.9.0",
      "15-cfenv@1.2.2",
      "22-consolidate@0.14.5",
      "24-dustjs-helpers@1.5.0",
      "25-dustjs-linkedin@2.5.0",
```

Artifact – [expensive]

Attestation



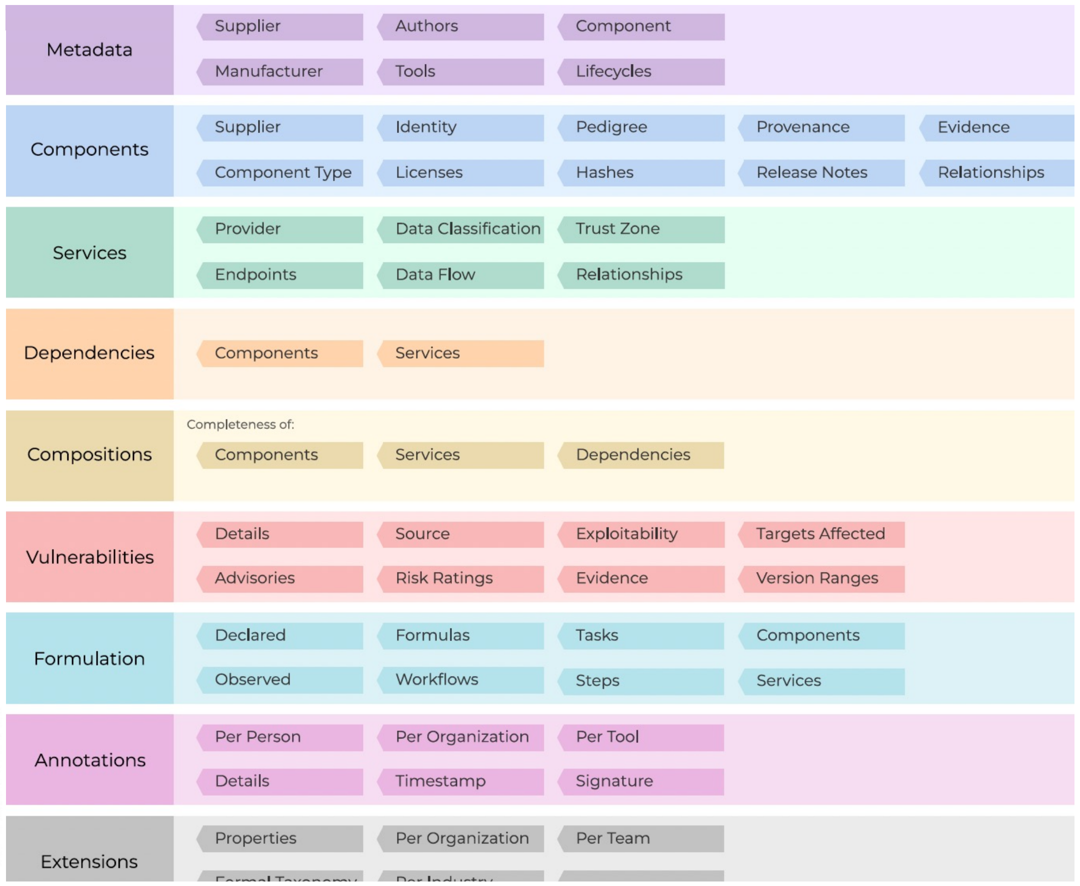
in-toto specification 1.0
https://github.com/in-toto/attestation/releases/tag/v1.0
Envelope
DSSE v1.0
payloadType:
application/vnd.in-toto+json
payload:
Statement
Base64Encoded JSON
Example SPDX SBOM Predicate
Example Subject

```
{
  "_type": "https://in-toto.io/Statement/v1",
  "subject": [
    {
      "name": "<NAME>",
      "digest": {"<ALGORITHM>": "<HEX_VALUE>"}
    },
    ...
  ],
  "predicateType": "<URI>",
  "predicate": { ... }
}
```

```
"predicateType": "https://spdx.dev/Document",
"predicate": {
  "SPDXID" : "SPDXRef-DOCUMENT",
  "spdxVersion" : "SPDX-2.2",
  ...
}
```

```
{
  "name": "us.gcr.io/dasith-wijes/demo123",
  "digest": {
    "sha256": "124e1fdee94fe5c5f902bc9 ..."
  }
}
```

[real] SBOM



Metadata	Supplier	Authors	Component		
	Manufacturer	Tools	Lifecycles		
Components	Supplier	Identity	Pedigree	Provenance	Evidence
	Component Type	Licenses	Hashes	Release Notes	Relationships
Services	Provider	Data Classification	Trust Zone		
	Endpoints	Data Flow	Relationships		
Dependencies	Components	Services			
Compositions	Completeness of:				
	Components	Services	Dependencies		
Vulnerabilities	Details	Source	Exploitability	Targets Affected	
	Advisories	Risk Ratings	Evidence	Version Ranges	
Formulation	Declared	Formulas	Tasks	Components	
	Observed	Workflows	Steps	Services	
Annotations	Per Person	Per Organization	Per Tool		
	Details	Timestamp	Signature		
Extensions	Properties	Per Organization	Per Team		

Stage: Artifacts - show of ✋

artifact repo basic security bp

—

create [real] SBOM

Bottom Line Message

Software Supply Chain has multiple levels → very different threats ⚡

Solutions / Mitigations on different levels of effort and complexity ☝️

in the real world

Context

consulting experience + master thesis research, looking for a “somewhat complete” set of SSCS controls

literature input from..

- CIS Software Supply Chain Security Guide
- CNCF Software Supply Chain Best Practices
- OWASP SCVS Software Component Verification Standard
- SLSA Supply-chain Levels for Software Artifacts
- Microsoft Secure Supply Chain Consumption Framework
- DoD Enterprise DevSecOps Reference Design

Context

3 Implementation Groups

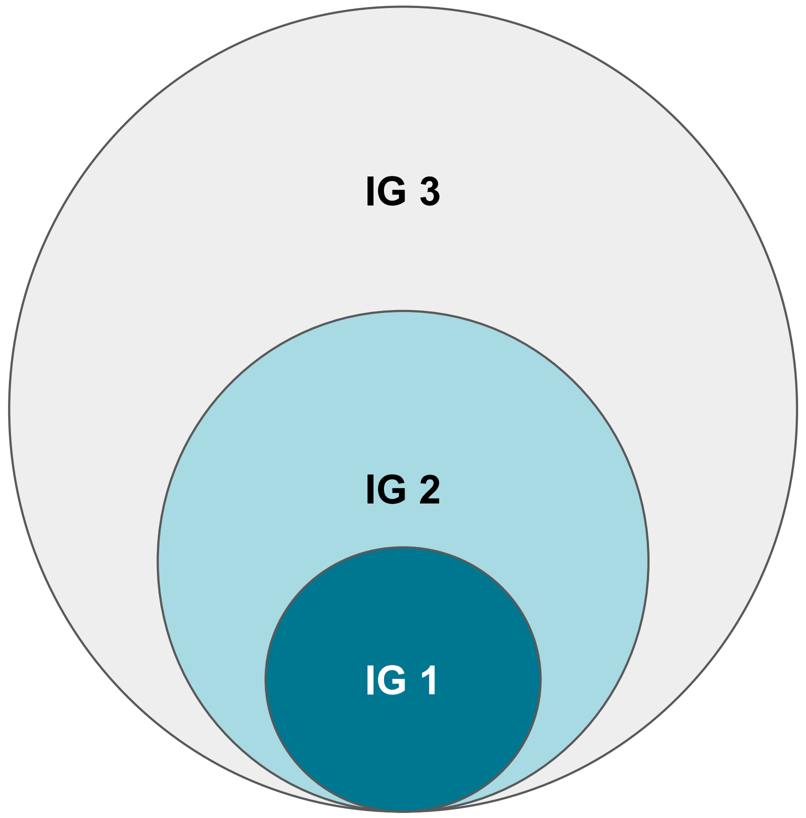
167 controls
6 categories

83 questions
4 possible answers

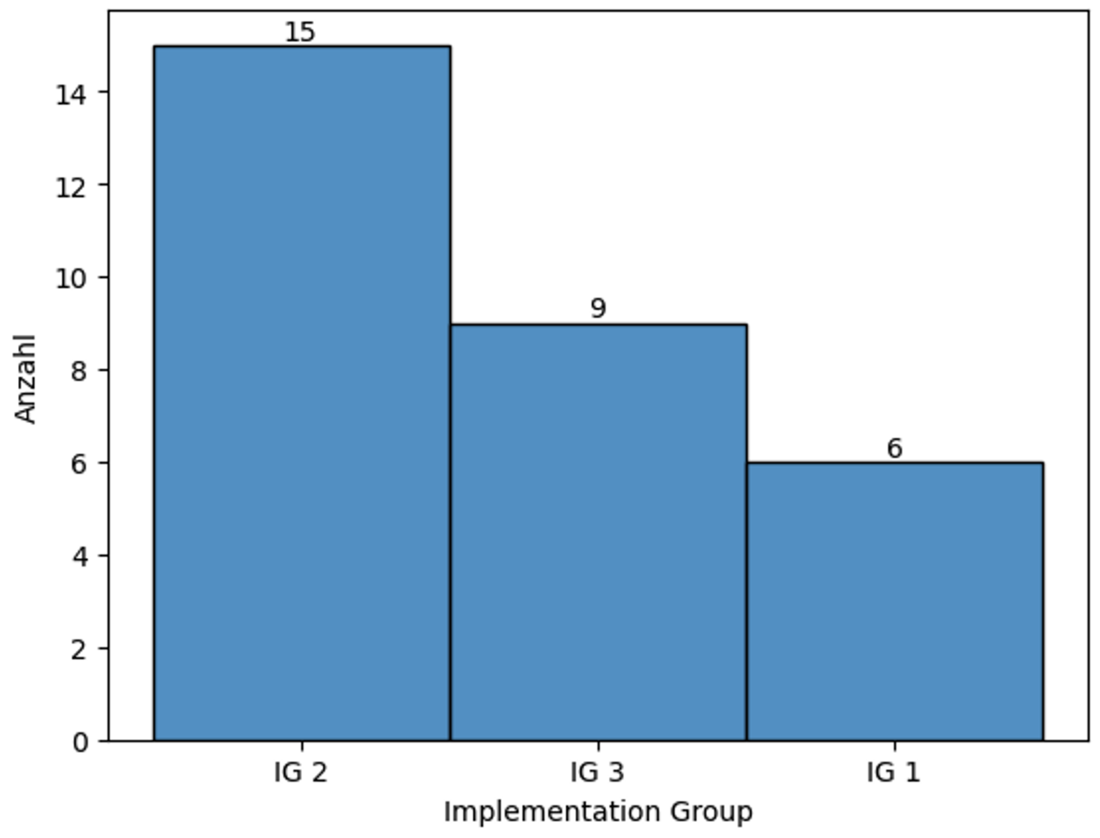
30 companies
(DACH)

Context

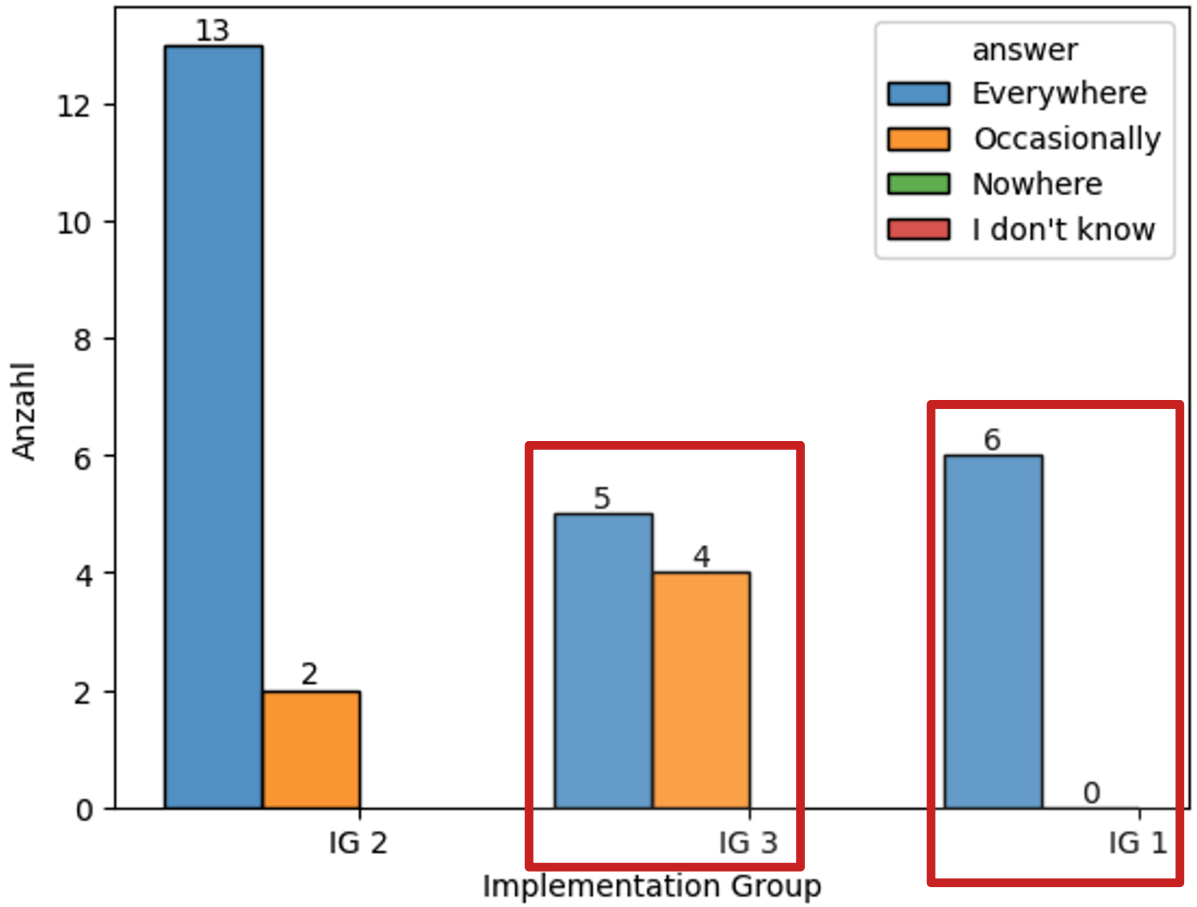
- IG 1
  - small company
  - no sensitive data
- IG 2
  - middle size company
  - some sensitive data
- IG 3
  - enterprise
  - highly sensitive data



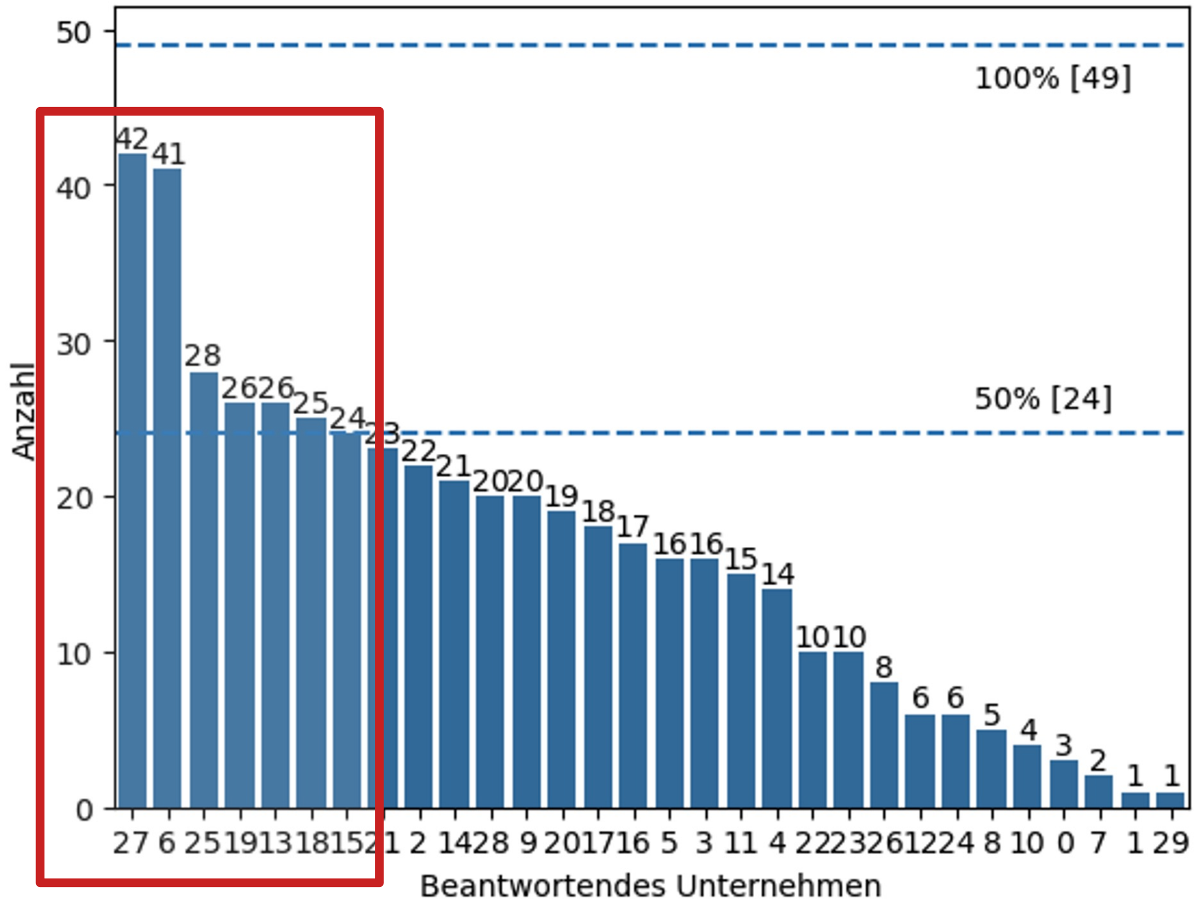
Findings - Companies per IP



Findings - Using VCS

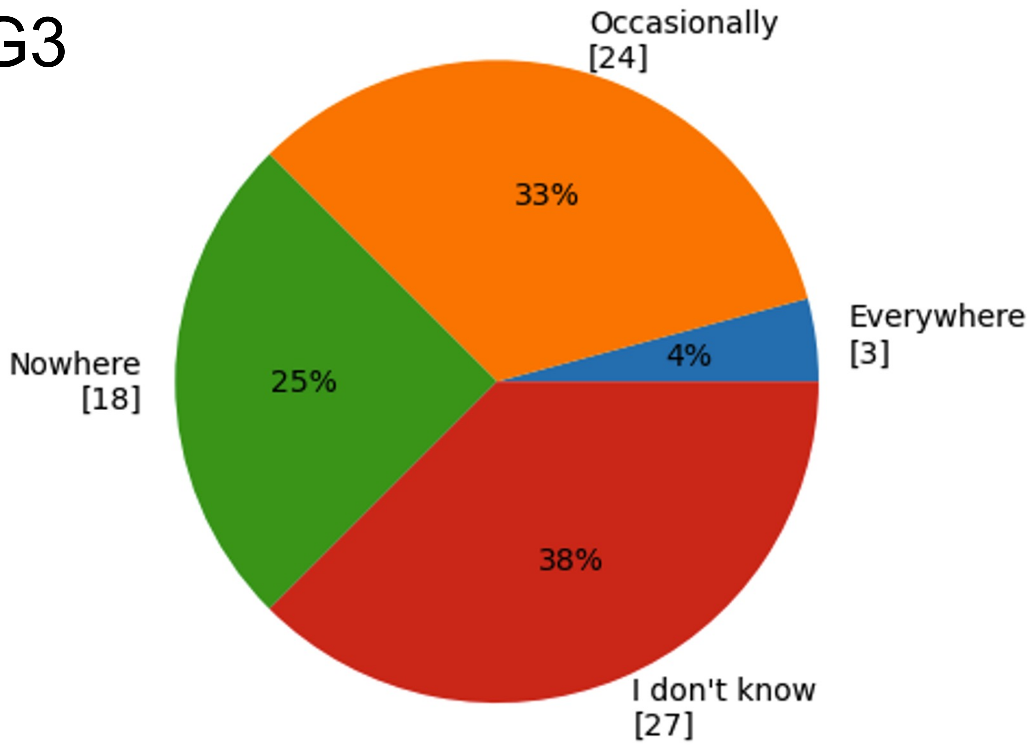


Findings - Implementing all IG1 controls

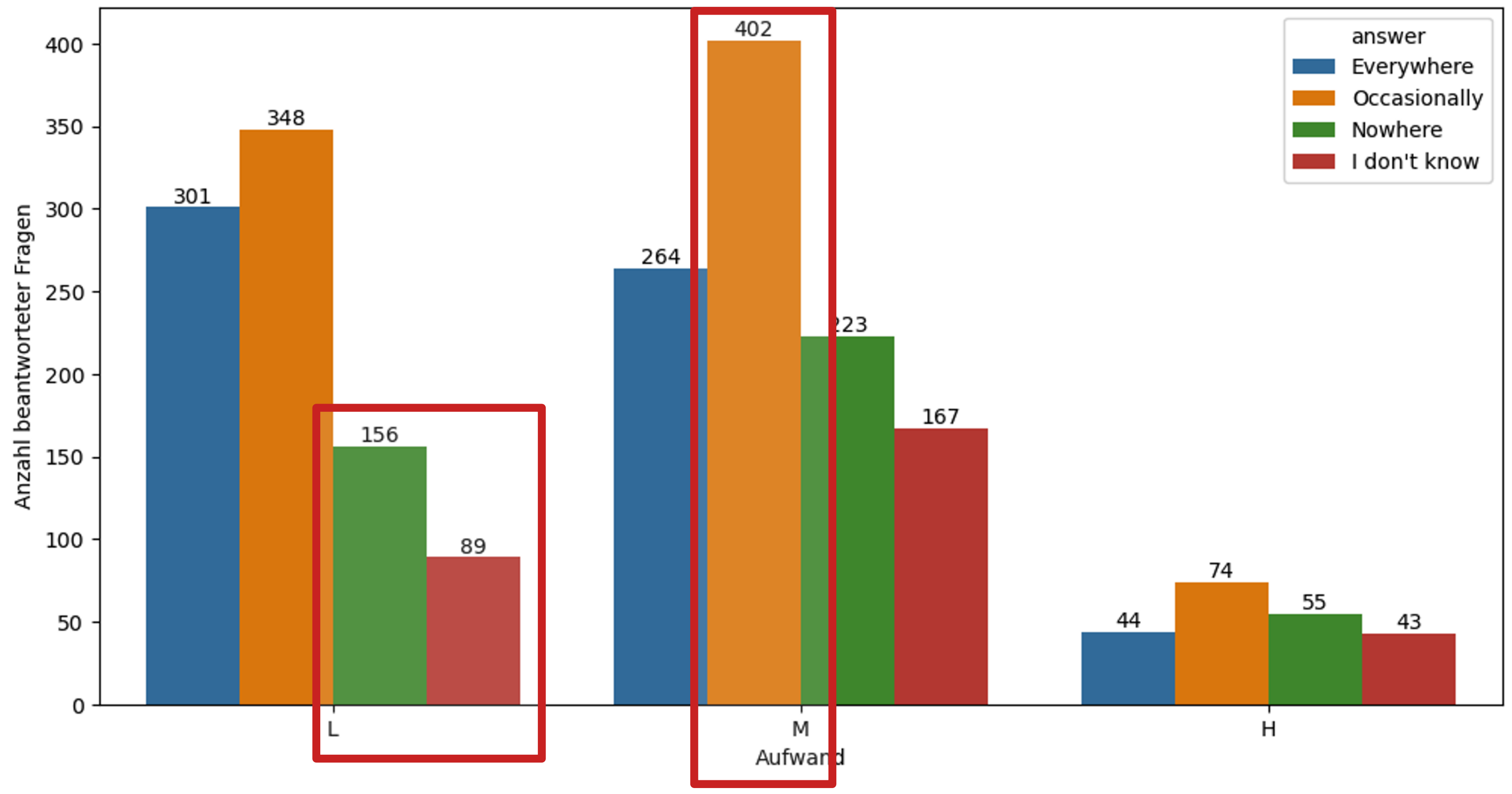


Findings - Implementing IG3 controls

- ☝️ only necessary for IG3 companies
- 😔 25% definitely not implemented
- 👍 ⅓ implemented somewhere
- 🤷‍♂️ > ⅓ unknown
    - no policy?
    - know how?



Findings - Controls vs Effort



Lessons Learned – from data

👆 IG / company size
👇 Transparency

Low hanging 🍇 not reaped

~25-50% of controls per group not implemented

🙃 build, SBOM, attestation

Lessons Learned – from experience

scans, tests & checks 🤝 policies

automation is 🔑
(IaC, pipelines, testing, PaC, ..)

The Hard Truth

👍 lots of information available

👎 many simple controls not implemented
👎 most complex controls not implemented

bigger company = less transparency/adaptation

Daniel Drack

Senior DevOps Engineer @ FullStackS

Organizer / Host
CNCG Graz + KCD Austria

- BSc MA MBA
- CK{A/AD}, TFA, VA, GitLab, PSM I, Snyk

📧 daniel.drack@fullstacks.eu
🌐 https://drackthor.me
@DrackThor

Further Reading

Code:
- SAST
- (GitLab) Push Rules
- Codeowners
- IaC Scanning Tools
- The Test Pyramid

Dependencies:
- SCA Tools
- SBOM Introduction
- Dependency Track

Build:
- Reproducible Builds
- Zero Trust Paradigm
- container based build

Artifacts, Distribution & Deployment:
- The Update Framework
- In-Toto Attestation
- Sigstore

used Literature (selection):
- CNCF Supply Chain Best Practices
- CIS Supply Chain Security Guide
- NIST SSDF
- SLSA
- OSSF S2C2F
- OWASP ASVS
- SSA Secure Software Controls