ALL DAY DEVOPS

NOVEMBER 6, 2019

Jeroen Willemsen

Why Manual Verification Still Matters

About me

Jeroen Willemsen
@commjoenie
jeroen.willemsen@owasp.org

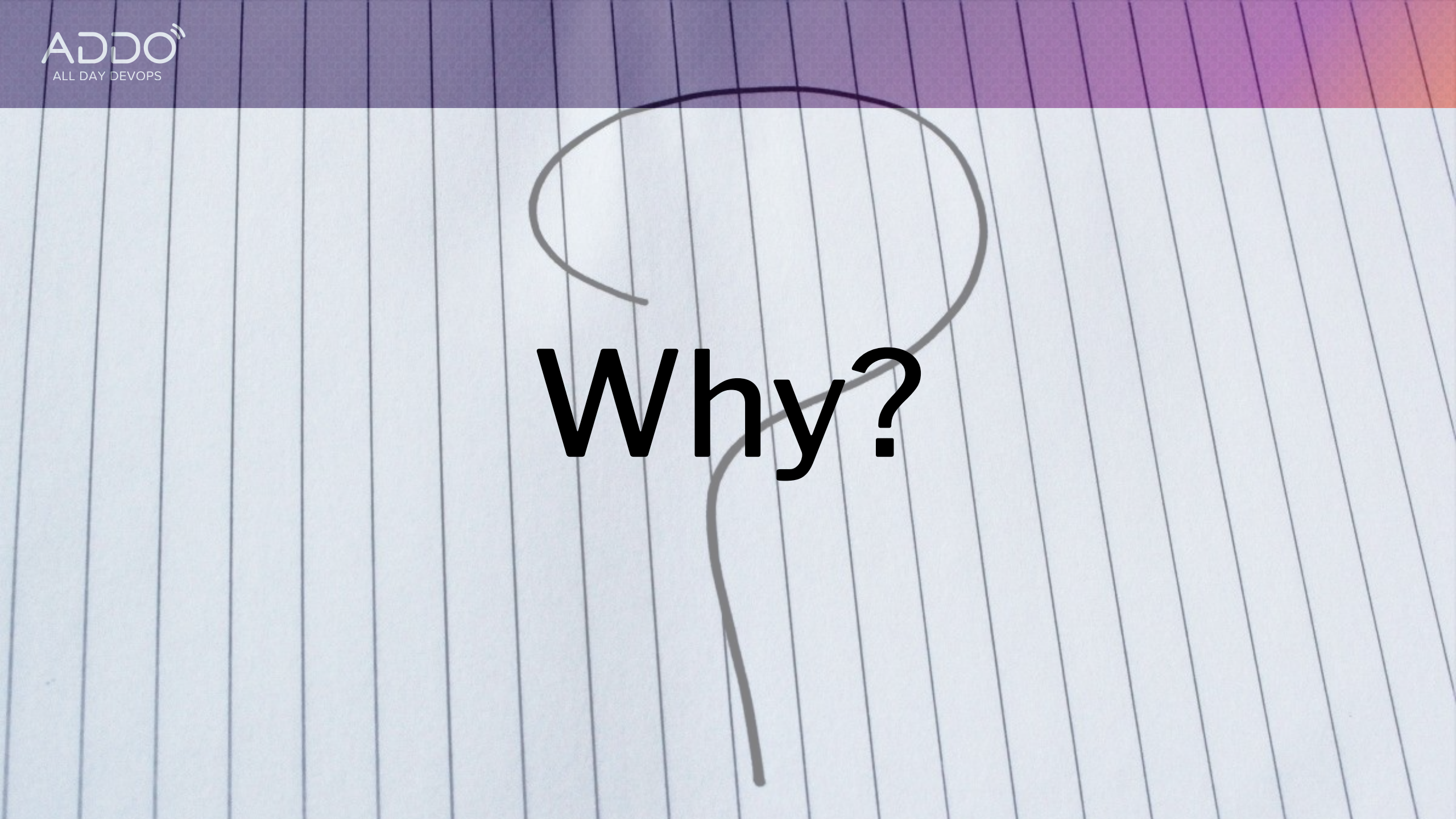
"Security architect"
"Full-stack developer"

Xebia Security


GOAL of Today:
Help you on the next step of your security journey

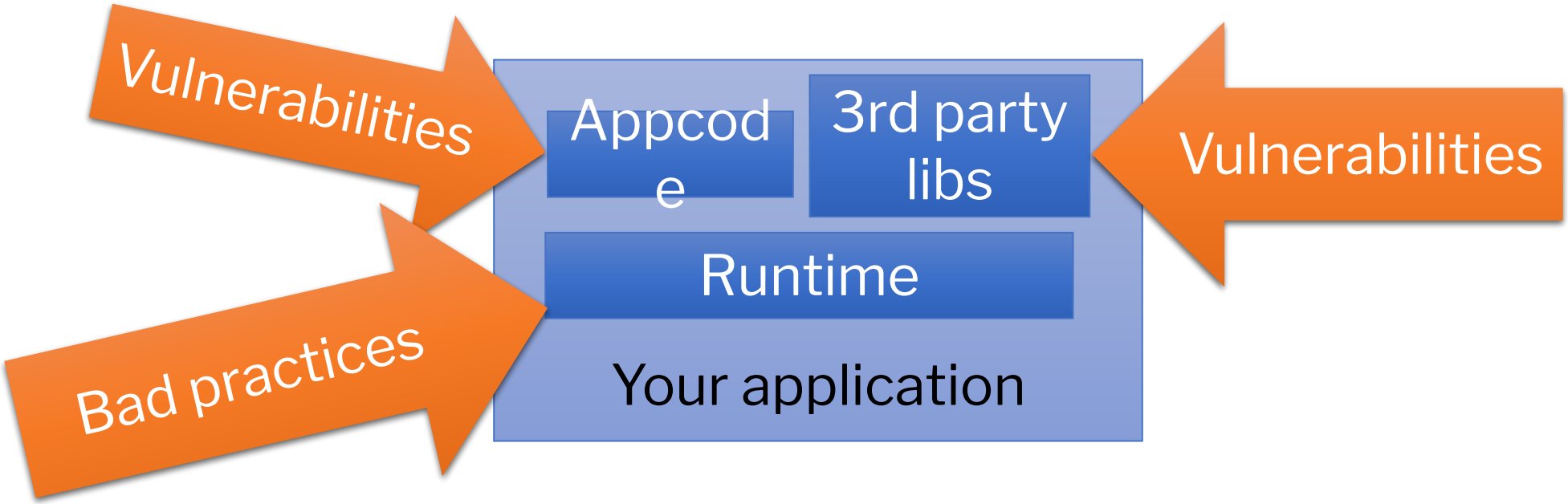
Index

- The application in its runtime
- The container up close
- The container on its platform
- The way forward


Why?

Index

- **The application in its runtime**
- The container up close
- The container on its platform
- The way forward


Vulnerabilities
Appcod
e
3rd party
libs
Vulnerabilities
Runtime
Bad practices
Your application

3rd party libs

- Start with a dependency checker!
- Start manual.
- Learn how to verify and how to “fix” it
  - Suppression, replacement, harden/add guards.
- Automate and keep rechecking.

3rd party libs

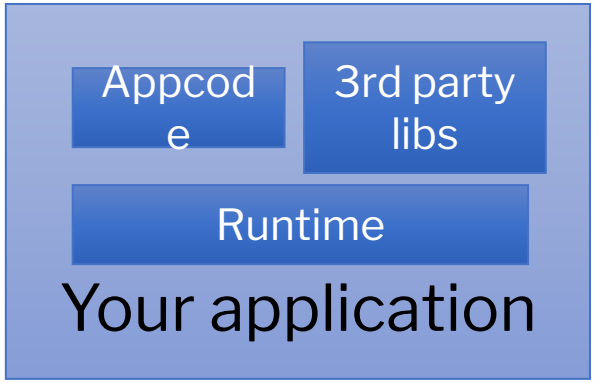
- Not all ecosystems have proper checkers.
- Not finding a CVE does not mean it is not there.
- ***Libraries still need vetting.***
- **Especially: Internal "libraries" need your own vetting!**
- What if it only takes 1 Class?

Appcode

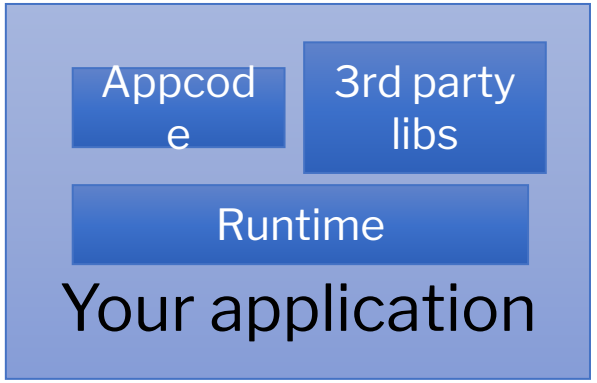
- SAST tooling mileage will vary depending on the language, framework & tooling
  - Always have linters on board!
  - Start manual before automation
  - Have explainable suppression in place
  - Test before buy

Appcode

- Create your own tool / extend existing tools if necessary
- Use GREP!
- ***Manual reviews are king!***
- **Missing authorization? Disclosing information in an error?**



- Want to know how your backends behave in general?
- DAST will
  - help finding missing headers;
  - detect some of the obvious missed XSS/SQLI;
  - create a lot of false positives;
  - create a lot of false negatives due to reach and recoginition.

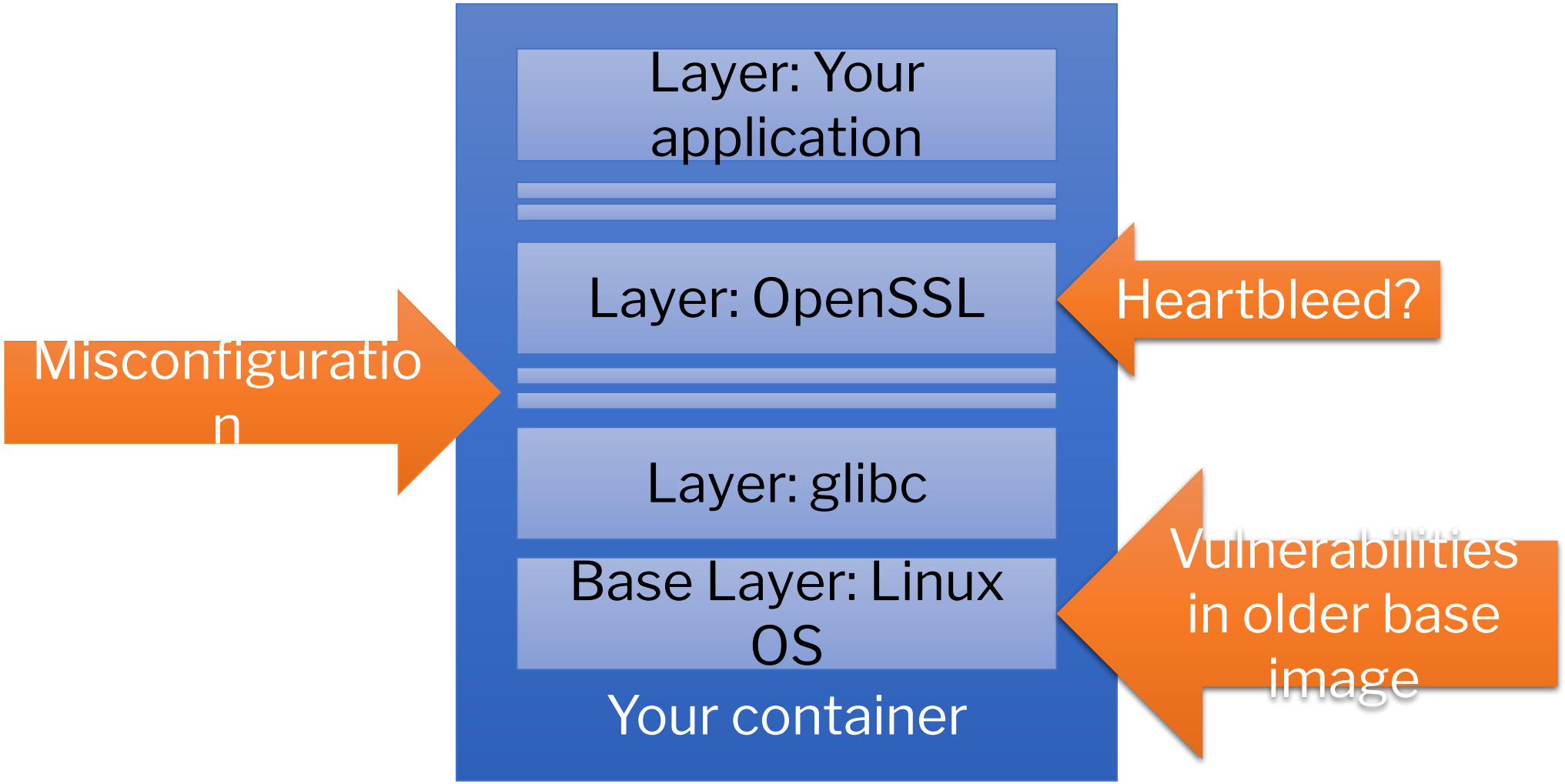


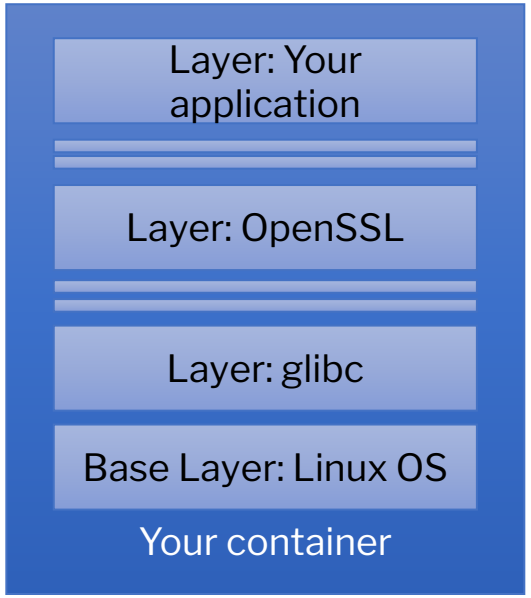
- Don’t want to burn yourself against an earlier finding?
  - Write a security test

- Write security tests based on evil user stories.

Index

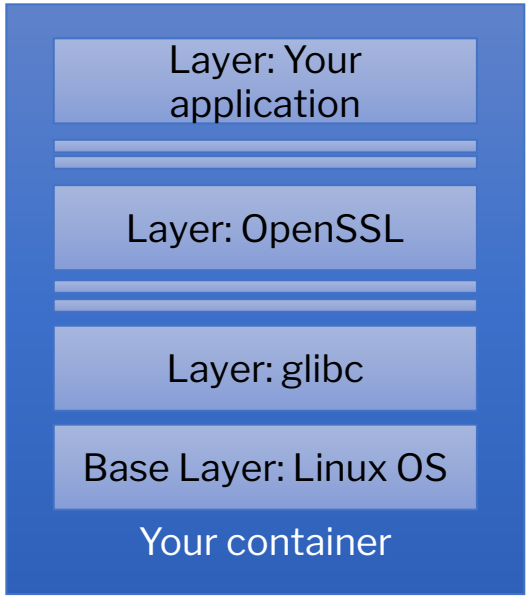
- The application in its runtime
- **The container up close**
- The container on its platform
- The way forward

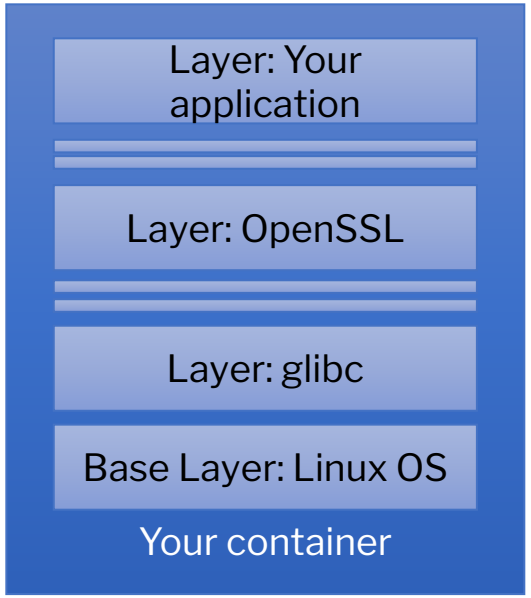

Layer: Your application
Layer: OpenSSL
Layer: glibc
Base Layer: Linux OS
Your container
Misconfiguration
Heartbleed?
Vulnerabilities in older base image



- Analyzing layers for vulnerabilities with Clair/Anchore gives great first insight
  - Start manual, then automate
  - Keep retesting: a CVE might pop up in a later stage.
  - Focus on base-images!
  - **But what about unknown binaries?**

- What about configuration mistakes?
  - Lynis/inspec to the rescue.
  - Create your own target profile!
  - Focus on base-images and verify on change!
  - Profiles require maintenance.



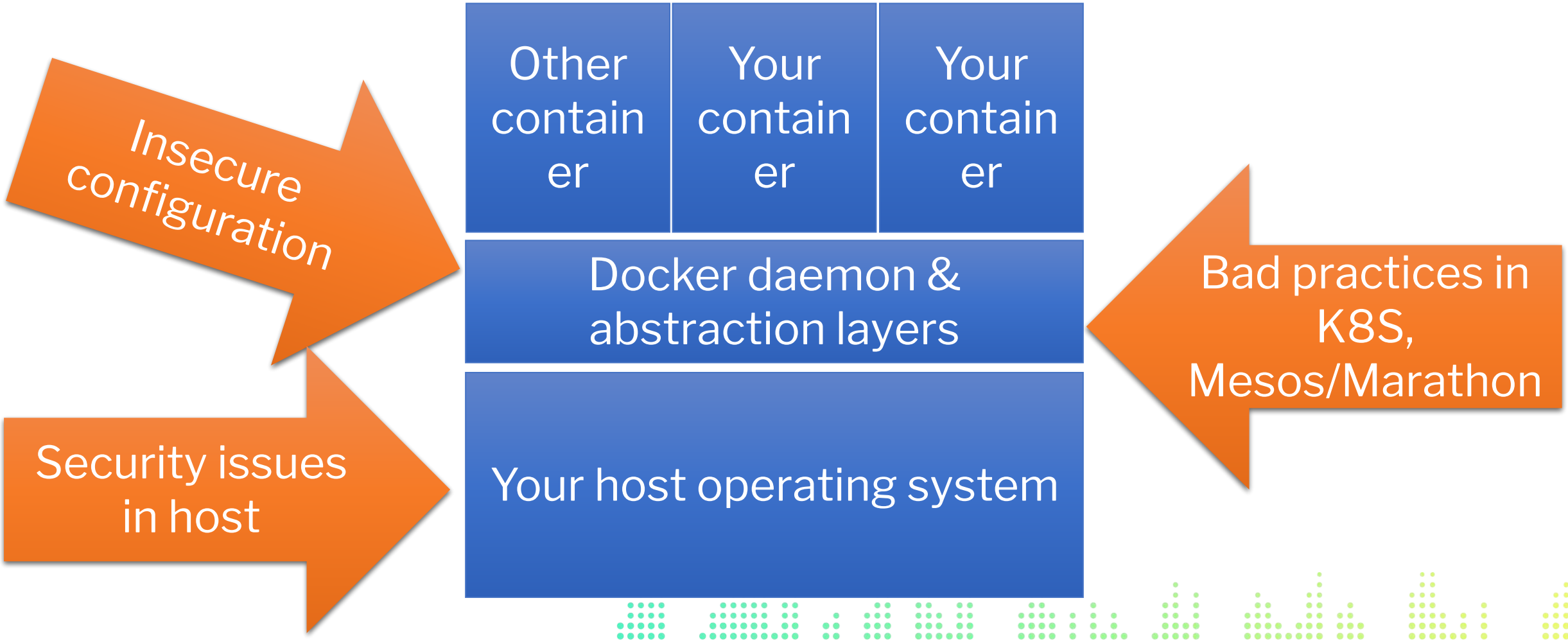


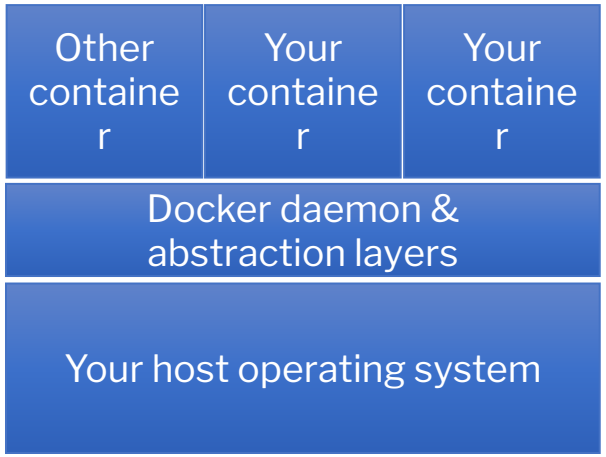
- What about configuration mistakes?
  - Dockle/hadolint on a syntax level.
  - Focus on base-images and verify on change!

- It might not find your private key in the container.

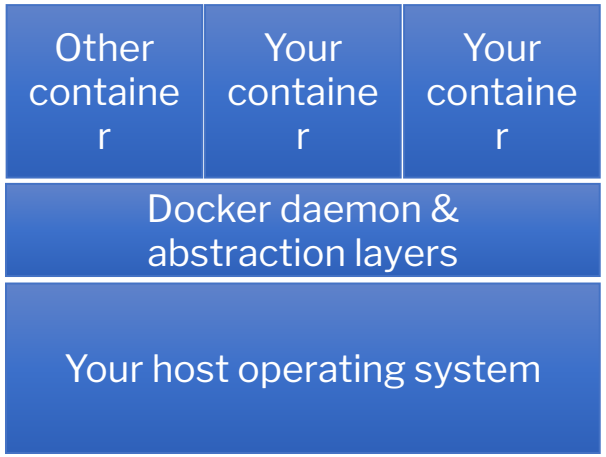
Index

- The application in its runtime
- The container up close
- **The container on its platform**
- The way forward


Insecure configuration
Security issues in host
Other container
Your container
Your container
Docker daemon & abstraction layers
Your host operating system
Bad practices in K8S, Mesos/Marathon



- Take the hygiene steps first:
  - CIS Benchmarks through Inspec/ciscat
  - Lynis/Openscap on your system
  - Check for vulnerabilities: OpenVAS or Nessus
  - Other commercial tools



- Immutability is king
- Fat hosts often have too much attack surface out of sight
- ***The richer the host, the more checking is required***

General issues

- Secrets leaked
- Network issues
- IAM issues
- “Cloud” hosting configuration

Secrets leaked

- Start with the sources (trufflehog)
- Check the containers (grep, tooling)
- Check the virtual machine images
- Check the runtimes
- Check the CI/CD pipeline
- VALIDATE the process, then the tooling

Network issues

- Keep it simple: start by “deny by default”
- Start as managed as possible: cloud providers offer tooling
- Not in the cloud? Use tools like nmap

- TLS and such? Use checkssl.sh and alike
- **Allways check for “unexpected” infiltration/exfiltration paths.**

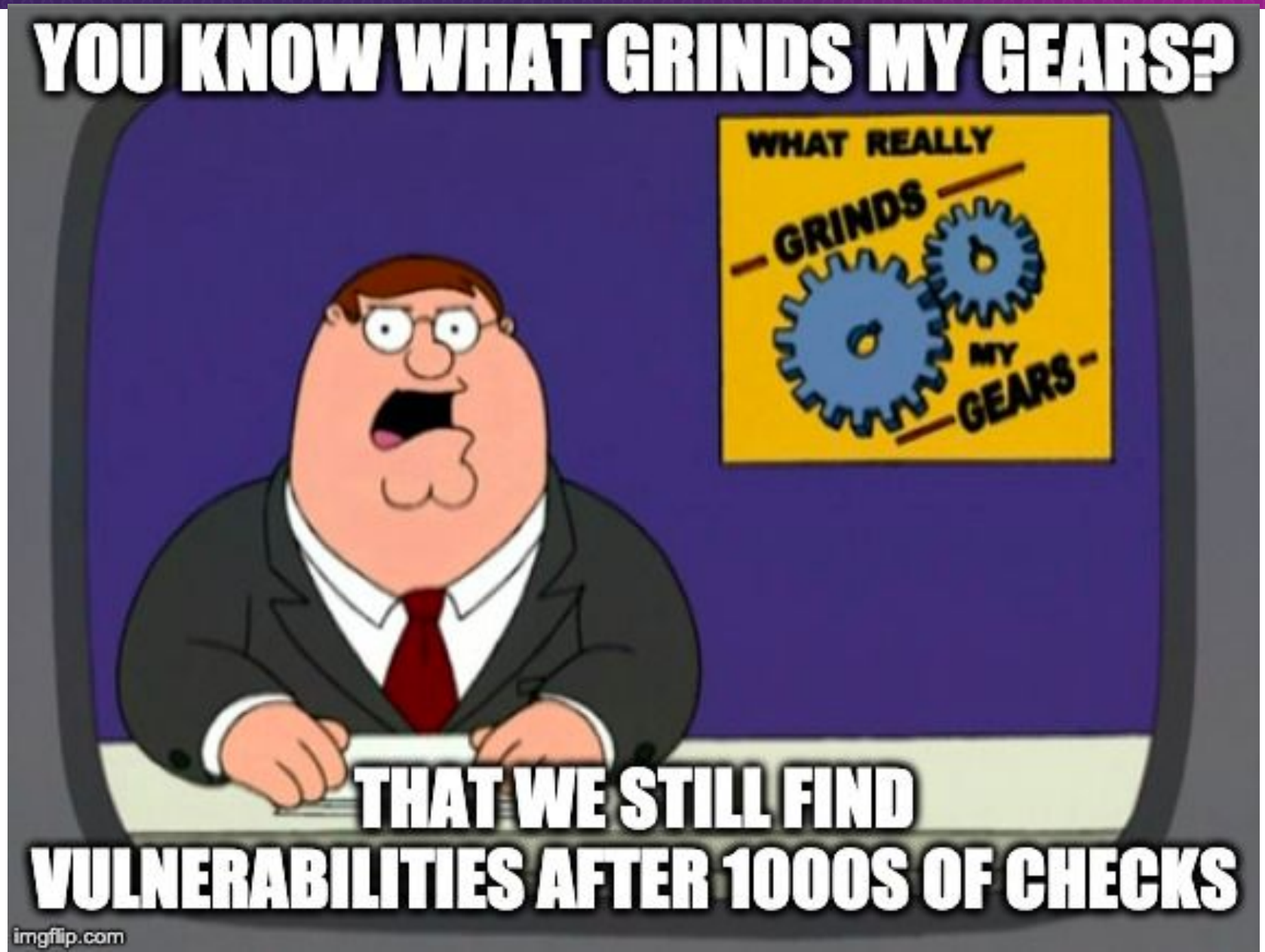
IAM issues

- Different environments require different tools
  - Application level
  - Host level
  - Cloud provider level
- When is something too broad? When is something too fine grained?
- ***Still you will need manual reviews for compliance and toxicisity.***
- ***User having access to all environments***

"Cloud" hosting configuration

- Various cloud providers have their own tooling ecosystem
- Cloud custodian is "cross-cloud"
- Using terraform? Check tools like terraform-compliance
- Run them on every check
- ***Manual verification of effective cloud configration is always necesseary***
- ***The too open network design?***


I have to automate more
CHANGE MY MIND
imgflip.com


YOU KNOW WHAT GRINDS MY GEARS?
WHAT REALLY
GRINDS
MY
GEARS
THAT WE STILL FIND
VULNERABILITIES AFTER 1000S OF CHECKS
imgflip.com

Index

- The application in its runtime
- The container up close
- The container on its platform
- **The way forward**

The Way Forward

- Automate as much as you can!

- ***Manual verification matters:***
  - We cannot catch every mistake
  - We learn about “new mistakes”

- Take risk based actions: it’s easy to drown in the pool of todo’s.

Questions?

- @commjoenie
- jeroen.willemsen@owasp.org

Thank you!

@commjoenie

jeroen.willemsen@owasp.org

ADDO
ALL DAY DEVOPS

SPONSORS

Sponsorship packages for All Day DevOps are available. If your organization is interested, please contact us for details.

DIAMOND SPONSORS

sonatype

StateFarm

GOLD SPONSORS

vmware

Fannie Mae

NORTHROP GRUMMAN

COMMUNITY ADVOCATES AND VIEWING PARTY SPONSORS

XebiaLabs

cloudbees

Carnegie Mellon University Software Engineering Institute

TIGERA

CONTINO

NowSecure

MEDIA SPONSORS

DZone

TechBeacon

DevOps.com

MediaOps

ITSP MAGAZINE

the cyberwire