RSAConference2020
San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: HUM-W01

10 Things I Wish Every Developer Knew About Security

Christopher J. Romeo

CEO
Security Journey
@edgeroute

About Chris Romeo

SECURITY BACKGROUND

- CEO / Co-Founder @ Security Journey
- 22 years in the security world, CISSP, CSSLP
  - *10 years at Cisco, leading security education.*
- Co-Lead of the OWASP Triangle Chapter

LISTEN TO ME

The Application Security Podcast

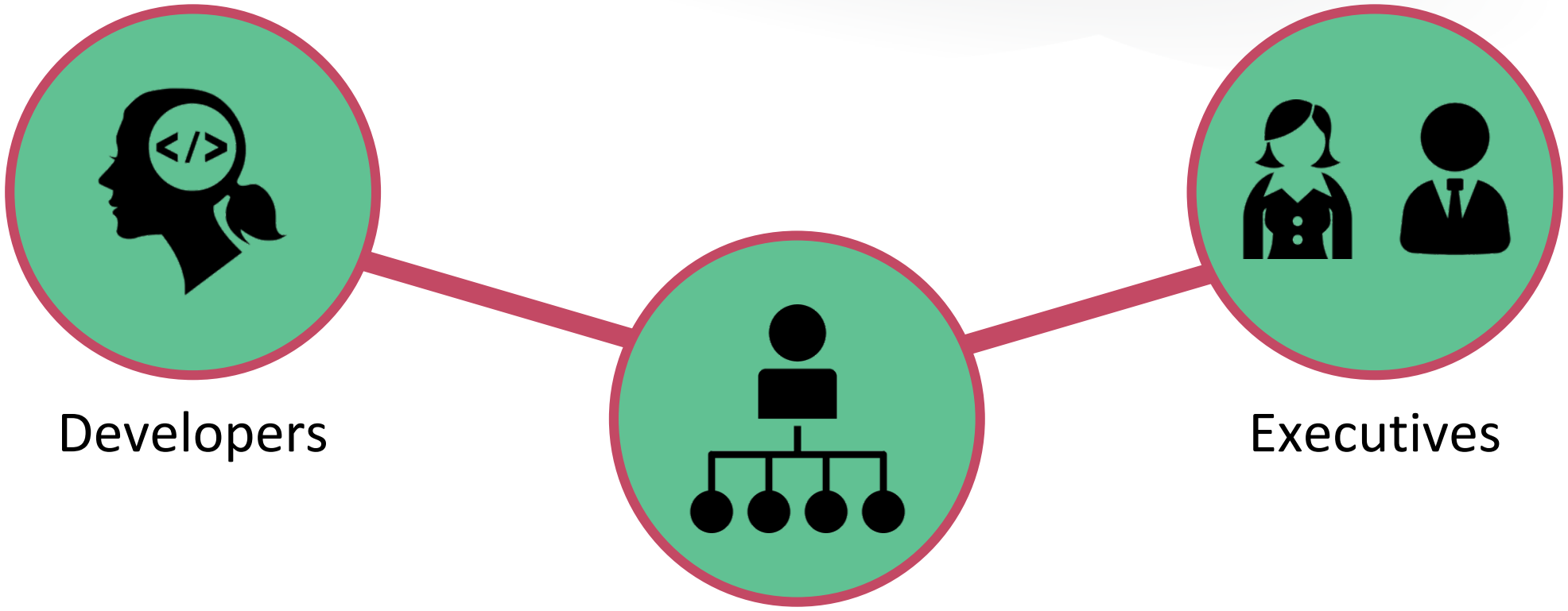
TALK TO ME

@edgeroute

@AppSecPodcast

A future where all developers are security enlightened

Applicability of the ”Ten Things”



Developers

Application Security Program Managers and Directors

Executives

Agenda

- The security state of the developer
- Ten things I wish every developer knew about security
  - Description
  - Assess
  - Build
- Conclusion
- Q+A

The goal of an application security program

IT IS *NOT*

To reach zero bugs because it's not possible!

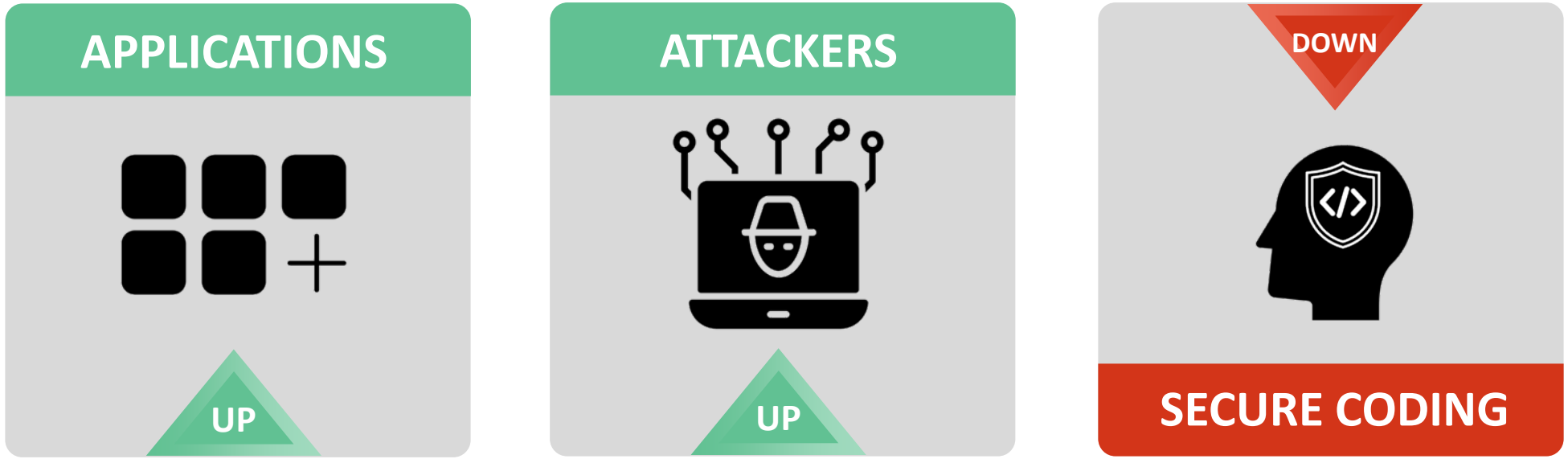
To reflect blame upon developers for security concerns.

IT *IS*

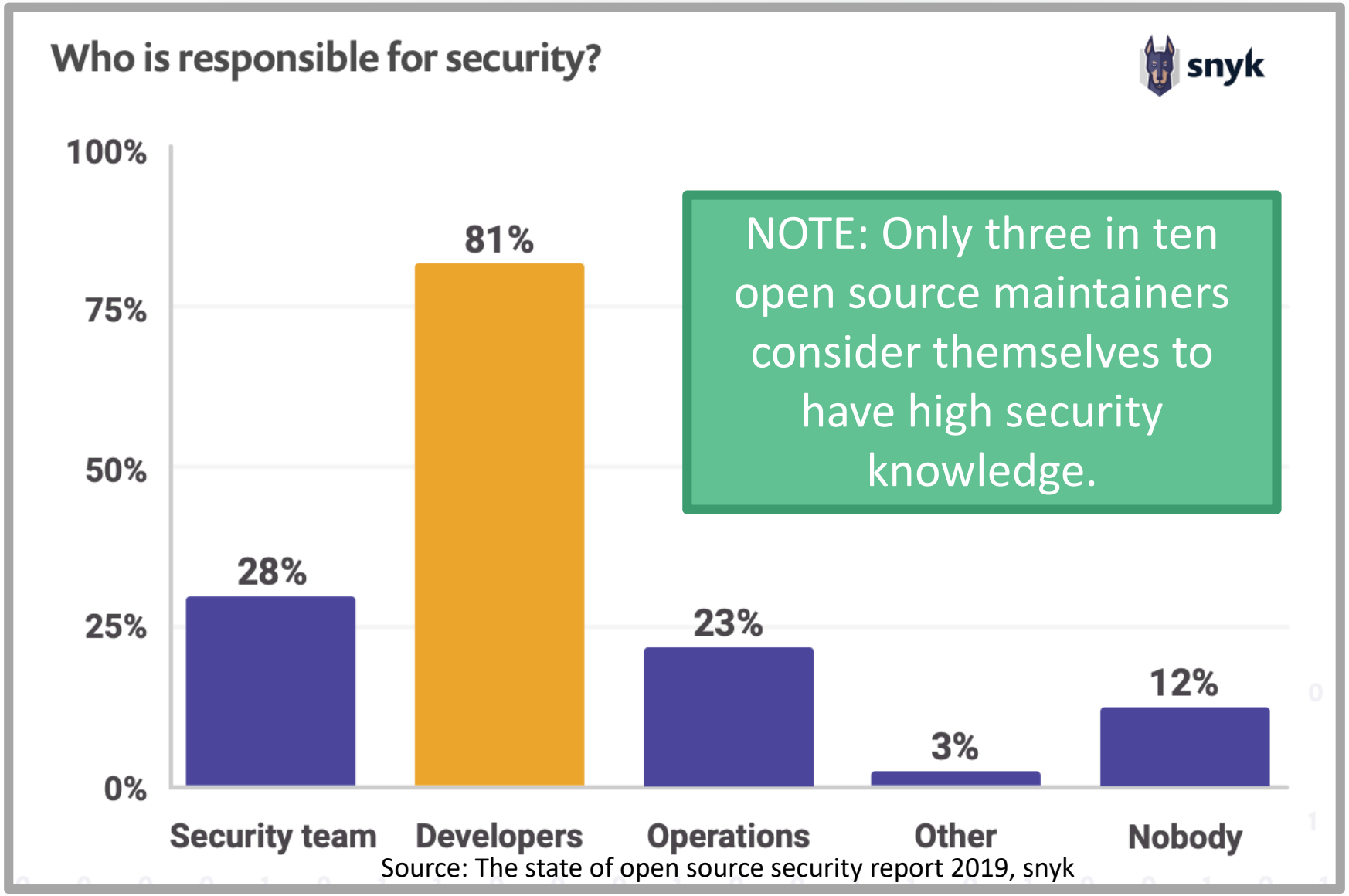
To fix bugs faster…and…to illuminate as many vulns as possible.

To measurably improve the security posture of the organization.

The security state of the developer



The responsibility for security

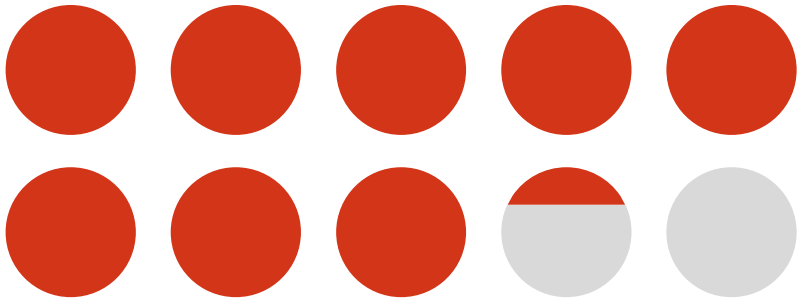


Source: The state of open source security report 2019, snyk

Concerned, but no time

Concern about open source security

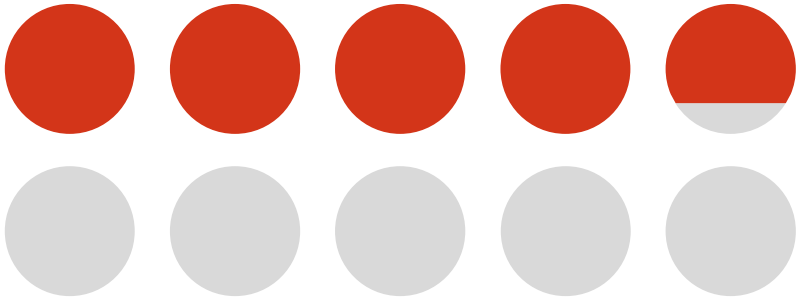
83% of developers are concerned about whether the open source code they use is secure.



Source: Enterprise JavaScript in 2019, npm

Security is important, but time is scarce

48% of developers say they believe security is important but don't have enough time to spend on it.



Source: DevSecOps Community Survey 2019, Sonatype

Stack Overflow: Dev's current source of knowledge

> … in some cases an insecure suggestion by a user with a high reputation score was selected as the accepted answer, as opposed to the correct fix by a user with a lower reputation score.

Secure Coding Practices in Java: Challenges and Vulnerabilities

> … in 1,305,820 Android applications available at Google Play. We show that 196,403 (15%) … contain vulnerable code snippets that were very likely copied from Stack Overflow.

Stack Overflow Considered Harmful? The Impact of Copy&Paste on Android Application Security

First set of conclusions

CONCLUSION 1

Security knowledge is low.

CONCLUSION 2

Security concern is serious.

CONCLUSION 3

Security resource allocation is lacking.

CONCLUSION 4

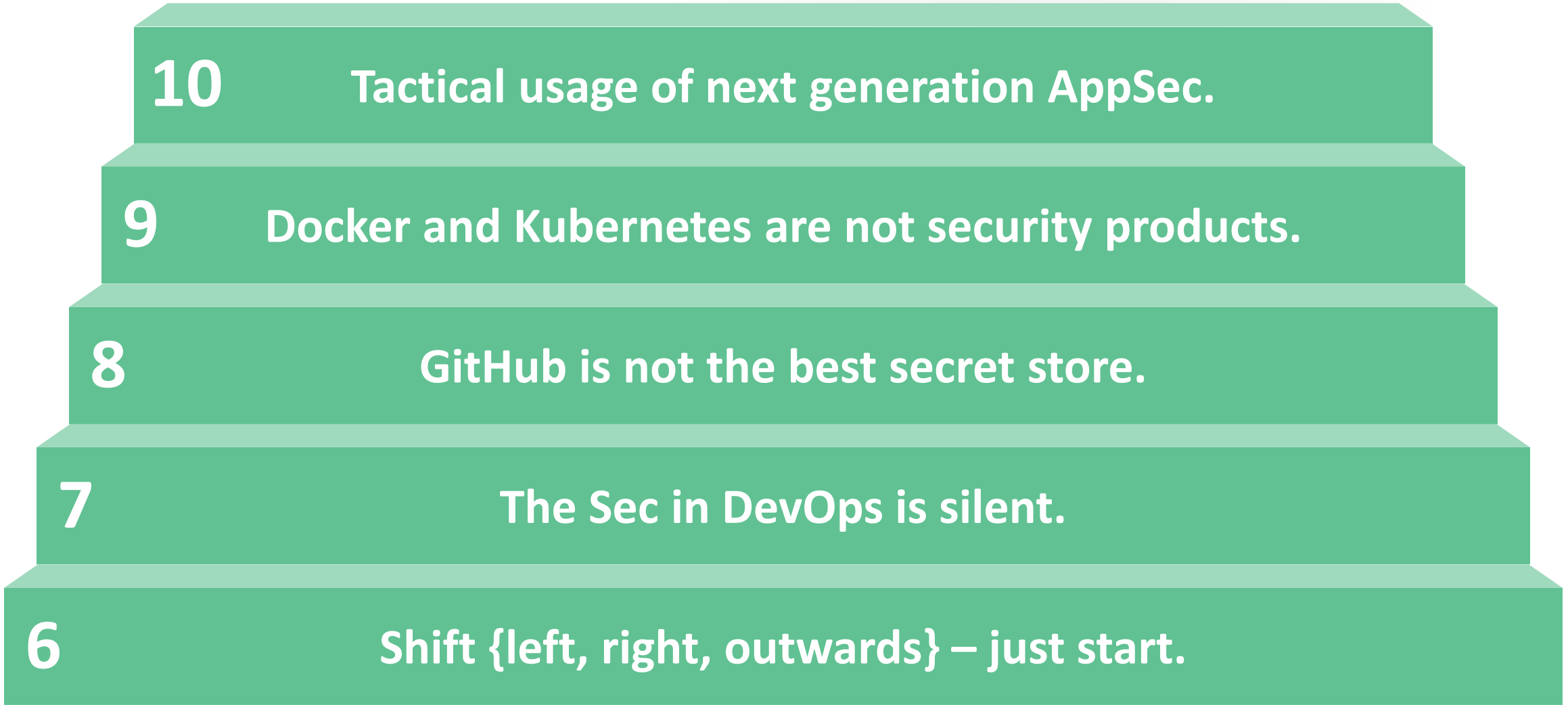
Stack Overflow is a source of security vulnerability.

The easy way out

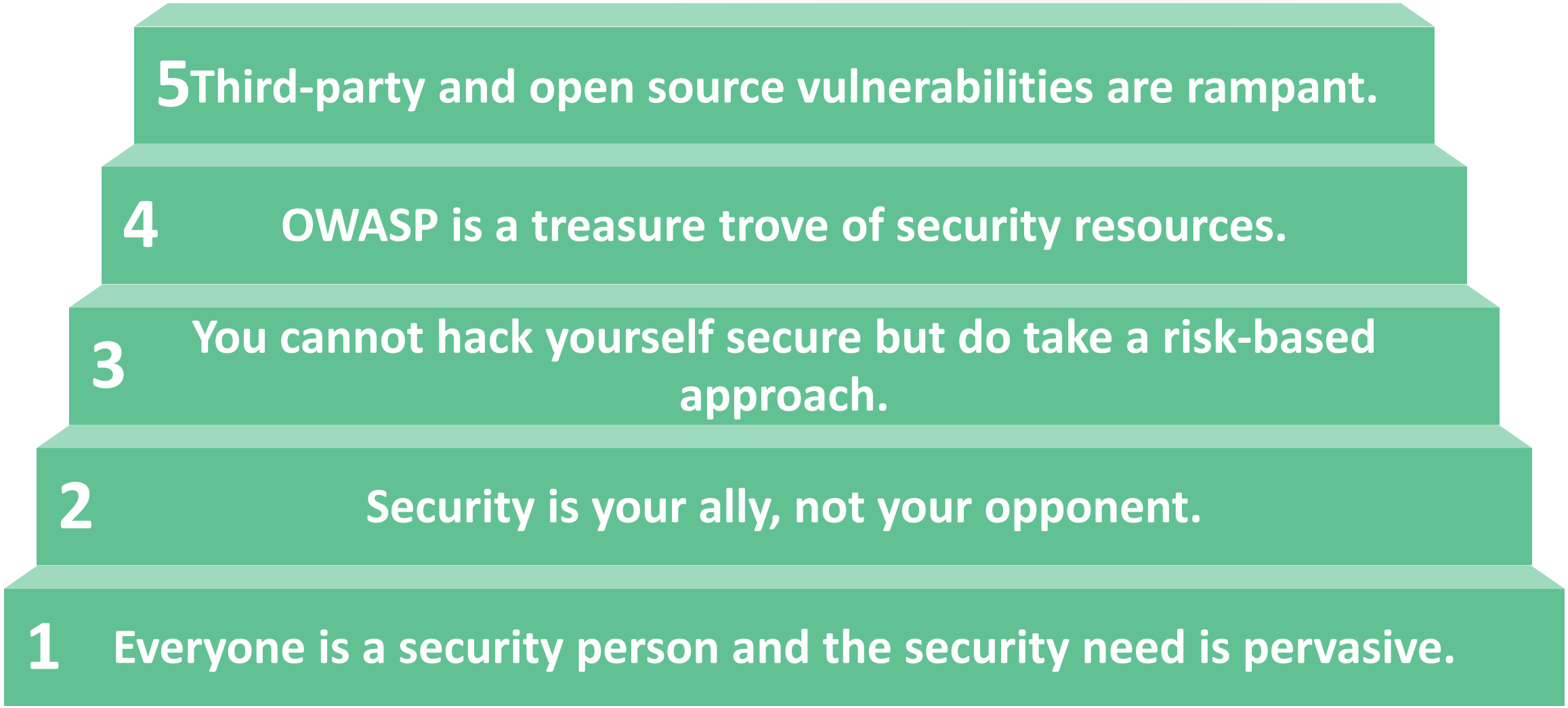
- C1 Define Security Requirements
- C2 Leverage Security Frameworks and Libraries
- C3 Secure Database Access
- C4 Encode and Escape Data
- C5 Validate All Inputs
- C6 Implement Digital Identity
- C7 Enforce Access Control
- C8 Protect Data Everywhere
- C9 Implement Security Logging and Monitoring
- C10 Handle All Errors and Exceptions

Think at a higher level

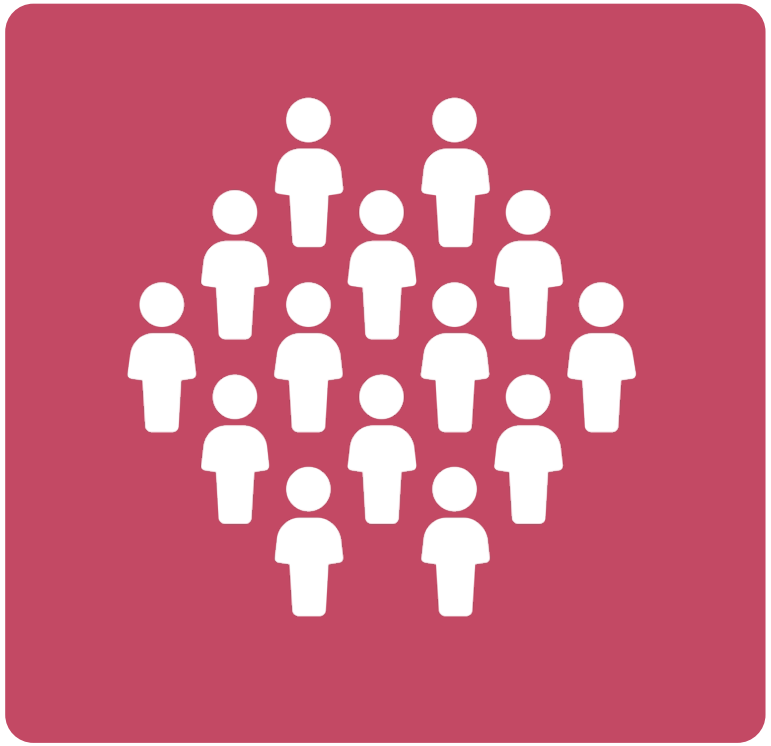
Ten things I wish every developer knew about security



Ten things I wish every developer knew about security



#1: Everyone is a security person and the security need is pervasive.



#1: Everyone is a security person and the security need is pervasive.

ASSESS

- Take the organization's security culture pulse.
- Gather the perspectives of Security Champions.
- Review technology areas & identify weak spots.
- Examine SDL and consider uniformity of artifacts.

BUILD

- Gain Executive buy-in.
- Communicate publicly about investment in security.
- Simplify methodology, language, and framework choices and provide adequate security guidance.

#2: Security is your ally, not your opponent.

The security team succeeds when the developer succeeds, in a valued partnership towards decreasing vulnerabilities.

#3: Security is your ally, not your opponent.

ASSESS

- Do developers hide things or intentionally avoid security?
- Does a security champions program exist?
- Are security coaches available?

BUILD

- Practice developer empathy.
- Deploy champions as building inspectors and issuers of security building permits.
- Provide Security Coaches to work w/your developers.
- Recognize security achievements.

#3. You cannot hack yourself secure but do take a risk-based approach.

Regardless of how much effort you place into breaking, the security properties of the unit under test will be no better after the engagement.

10 THINGS

Security Journey

#3. You cannot hack yourself secure but do take a risk-based approach.

ASSESS

- Consider how your organization portrays security.
  - Red team, penetration test, breaker approach?
  - Defenders/builders of secure code?
- Determine if a risk-based approach exists.

BUILD

- Use a well-rounded approach to educate your developers, with a risk-based focus:
  - **Builder**: secure coding
  - **Defender**: red team / blue team / purple team
  - **Breaker**: JuiceShop

#4: OWASP is a treasure trove of security resources.

Top Tens

OWASP Top 10 - 2017
The Ten Most Critical Web Application Security Risks

OWASP Pro Active CONTROLS

OWASP API Security Top 10 2019
The Ten Most Critical API Security Risks

Cheat Sheets

OWASP CHEAT SHEET SERIES PROJECT

Tools

DEPENDENCY-CHECK

DEPENDENCY-TRACK

#4: OWASP is a treasure trove of security resources.

ASSESS

- Poll how many developers know of the existence of OWASP.
- Verify the inclusion of OWASP resources into your application security program.

BUILD

- Host an OWASP Top 10 Lunch and Learn.
- Expose Developers to Cheat Sheets.
- Integrate ZAP and Dependency Check.

#5: Third-party and open source vulns are rampant.

Downloaded Java components with vulns: 6.1% (2015), 5.5% (2016), 12.1% (2017), and 10.3% (2018).

Source: 2019 State Software Supply Chain by Sonatype

11% of downloaded npm modules in 2018 had CRITICAL vulns (51% had vulns).

Source: NPM Future of JavaScript

#5: Third-party and open source vulns are rampant.

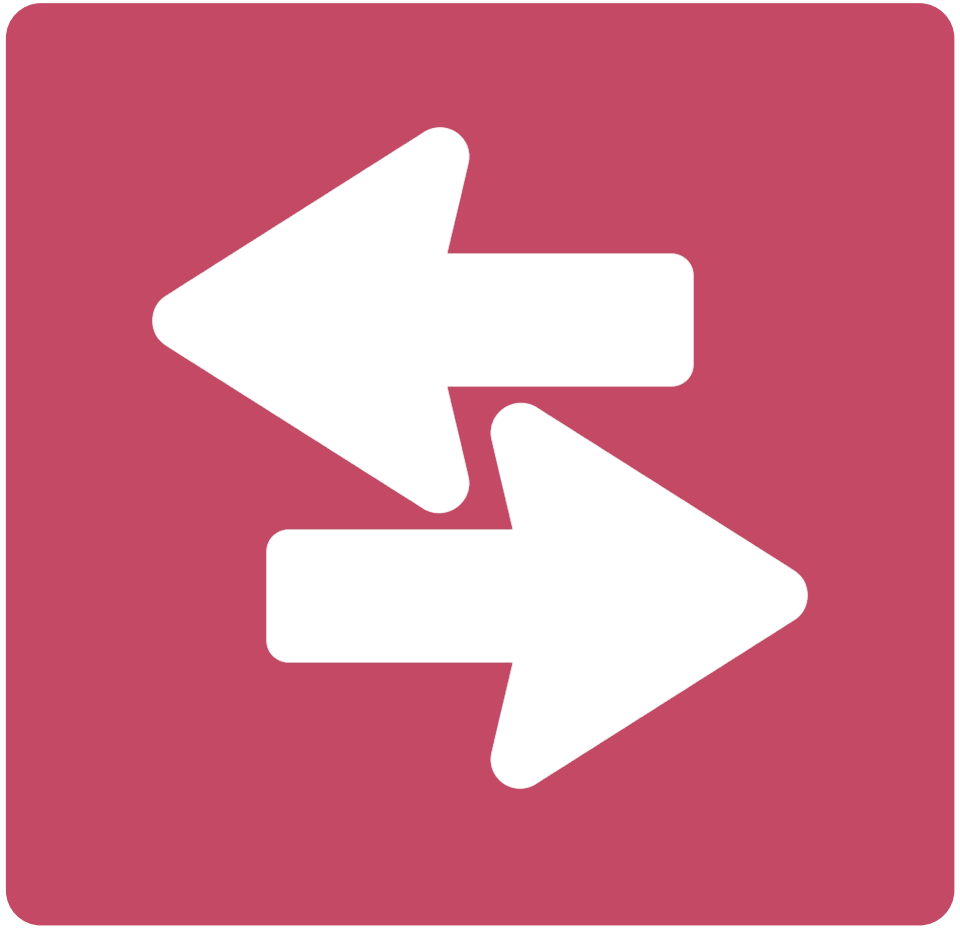
ASSESS

- Analyze the process of dealing with third-party and open-source software.
- Confirm toolsets in use by developers both while they code and in build pipelines.

BUILD

- Consider threat scenarios to ensure proper response.
  - Vuln in known component
  - Backdoor
- Deploy SCA tools sets and break the build on any finding.

#6: Shift {left, right, outwards} – just start.



Start left and start right = secure development lifecycle (what's old is new again).

#6: Shift {left, right, outwards} – just start.

ASSESS

- Analyze the secure development life cycle.
- Collaborate with Dev + Ops to determine SDL utilization.

BUILD

- Sell the concept of starting left to engineering and release management.
- Start right -- investigate & implement tools that provide security visibility, and defense in production.

#7: The Sec in DevOps is silent.

You cannot have a true DevOps without security integrated.

10 THINGS

Security Journey

#7: The Sec in DevOps is silent.

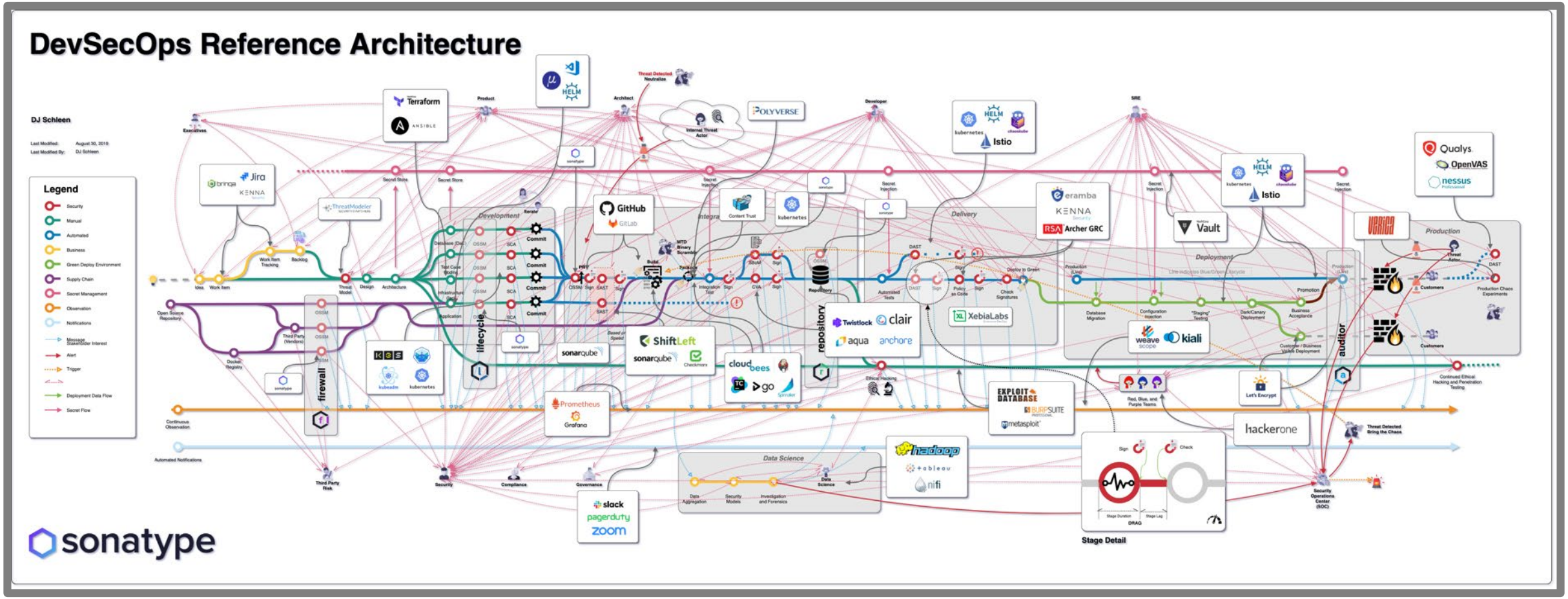
ASSESS

- Evaluate your dev methodology.
  - WaterFail, AgileFall, Agile, DevOps
- Consider the different phases of your DevOps pipeline and analyze the integration of security in each step.

BUILD

- Cross pollinate and develop DevSecOps Unicorns -- specialists that cross all three barriers.
- Add additional tools and automation to bolster security.

#7: The Sec in DevOps is silent.



Source: https://www.sonatype.com/devsecops-reference-architecture-2019

#8: GitHub is not the best secret store.

Secrets should never live in your source code control system.

#8: GitHub is not the best secret store.

ASSESS

- Code review for usage of secrets / API keys in source and configuration files.
- Utilize tools such as gitrob, truffleHog, git-secrets, and OWASP Sedated to find secrets and keys.

BUILD

- Educate on the risk of keeping secrets and keys in source code control.
- Automate the execution of secret sniffing tools on code commits.

#9. Docker and Kubernetes are not security products.

Will Sargent
@will_sargent

Replying to @edgeroute

They aren't. Even the docker security lead says if you want secure docker you should run it in isolated hypervisor

#9. Docker and Kubernetes are not security products.

ASSESS

- Determine the developer's reliance on Docker and Kubernetes in your architecture.

BUILD

- Educate on the vast security adjacent features of both Docker and Kubernetes.
  - Docker: Namespaces, Capabilities, Seccomp, AppArmor, SELinux
  - Kubernetes: Security Context. Pod Security Policies, RBAC, Namespaces
- Guide on how to use Docker and Kubernetes securely.

#10: Tactical usage of next generation AppSec.

RASP
Runtime
Application
Self Protection

IAST
Interactive
Application
Security
Testing

SCA
Software
Composition
Analysis

CWPP
Cloud
Workload
Protection
Platform

Security Journey

@edgeroute

#10: Tactical usage of next generation AppSec.

ASSESS

- Determine the next-generation application security tool consideration of your organization.

BUILD

- Add IAST to development and build to discover vulns.
- Add RASP in production to block and alert on attacks.
- Add SCA to detect and stop builds with vulnerable third-party components.
- Add CWPP to secure and audit your containers and serverless workloads.

Second set of conclusions

CONCLUSION 1

Developers are not the problem; they are the solution.

CONCLUSION 2

Security partnership unifies everyone and focuses us on the important things.

CONCLUSION 3

"Hacking" is not the answer; take a risk-based and OWASP approach.

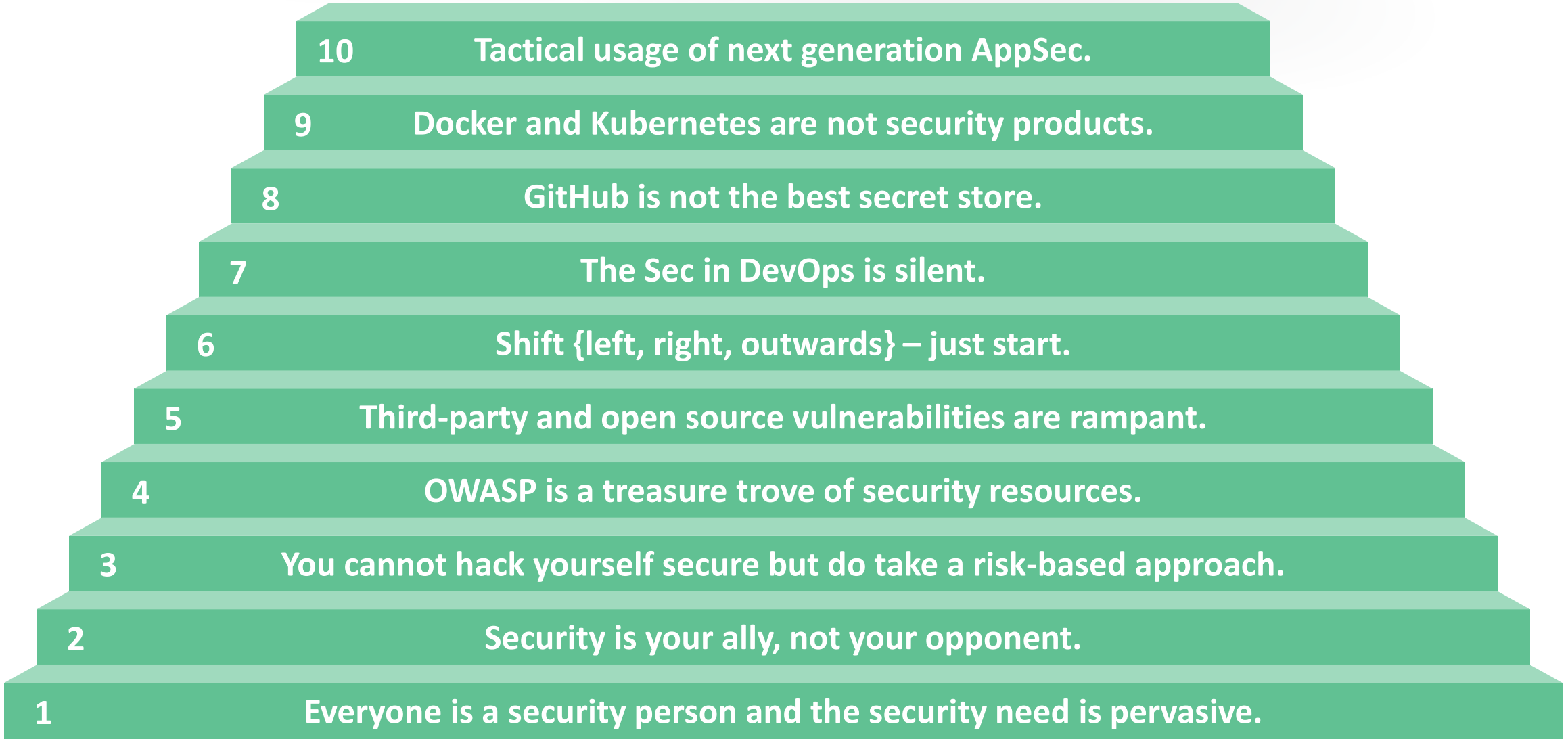
CONCLUSION 4

Monitor the dependencies, adjust the process, and select the proper tools.

"Apply" Slide

- Next week you should:
  - Review the ten things and begin to think through how many of these your organization has embraced.
- In the first three months:
  - Build a plan to educate your developers on the topics that are not a part of your security culture.
- Within six months you should:
  - Continue to implement the plan and communicate with your developers regularly.

Summary of "Ten things I wish every developer knew about security"



Q&A

LISTEN 

The Application Security Podcast

EMAIL 
chris_romeo@securityjourney.com

SOCIALS 
@edgeroute @AppSecPodcast

References

- https://res.cloudinary.com/snyk/image/upload/v1551172581/The-State-Of-Open-Source-Security-Report-2019-Snyk.pdf
- https://cdn2.hubspot.net/hubfs/5326678/Resources/JavaScript%20Surveys/2019_npm_survey_FINAL.pdf
- https://www.sonatype.com/2019survey