OWASP 2023 GLOBAL AppSec | DUBLIN IRELAND FEB 13-16

PRESENTED BY: CHRIS ROMEO

Ten DevSecOps Culture Failures

Chris Romeo

Background

- CEO @ Kerr Ventures
- 25 years in the security world, CISSP, CSSLP
- Previously Security Journey (CEO & Co-Founder)
- Previously Cisco (Chief Security Advocate)

Listen to me

The Application Security Podcast

The Security Table

Talk to me

@edgeroute

@AppSecPodcast

Agenda

- DevSecOps culture: definition and influence
- Ten Failures in DevSecOps
  - The failure itself
  - Guidance on building a better security culture with DevOps Security Successes
- Conclusion and Q+A

A time machine...

KERR

DevSecOps cultural drivers

KERR

GitLab asked developers to describe the most challenging parts of their role:

- “Security, security, security”...offered by more than one thousand respondents
- “To keep it secure and keep it updated”
- “Trying to build applications that are secure and stable”
- “Data security, data security, I repeat, data security”

Source: The GitLab 2022 Global DevSecOps Survey

A “Happy” DevOps culture

- Automation
- Lack of Friction
- Blameless retrospectives
- Transparency
- Collaboration
- Triumvirate of trust

The reality/ importance of your DevSecOps culture

What happens with security when people are left to make their own decisions.

KERR

DevSecOps culture stagnates or grows

KERR

* The Sec is silent.

DevSecOps – what's in a name?

The **10** DevSecOps failures

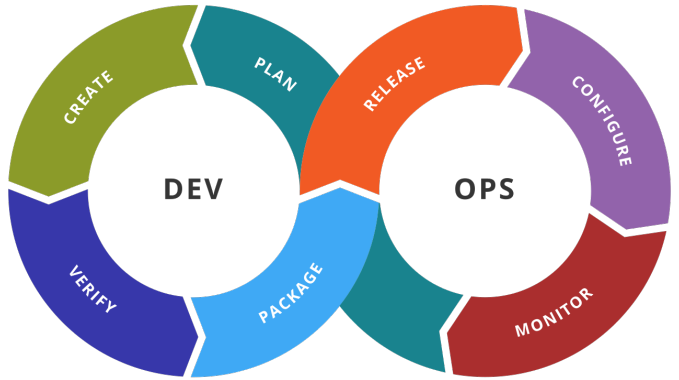
1. The infinity graph
2. Security as a special team
3. Vendor defined DevOps
4. Big company envy
5. Marketing term infatuation
6. Overcomplicated pipelines and doing everything now
7. Security as gatekeeper
8. Noisy security tools, and too many
9. Lack of threat modeling
10. Vulnerable code in the wild

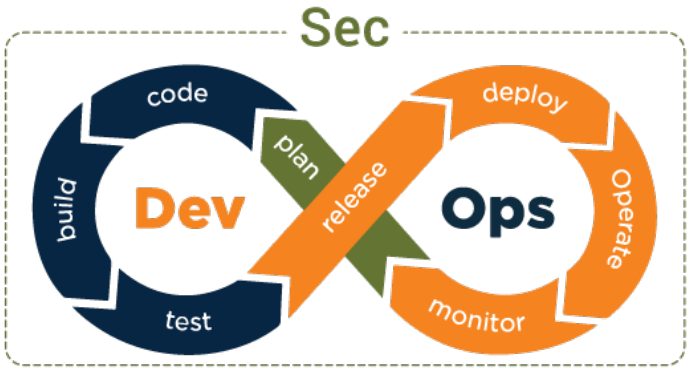
KERR

@edgeroute

Failure #1

The infinity graph



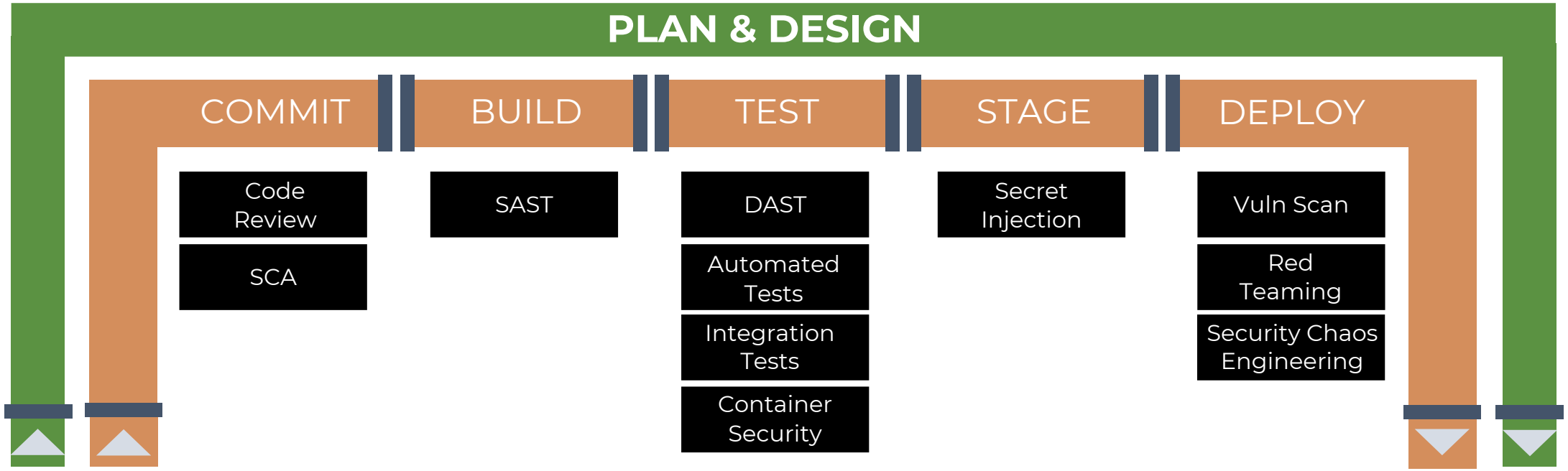


KERR

@edgeroute

Pipelines, not infinity graphs

Security Best Practices | Threat modeling



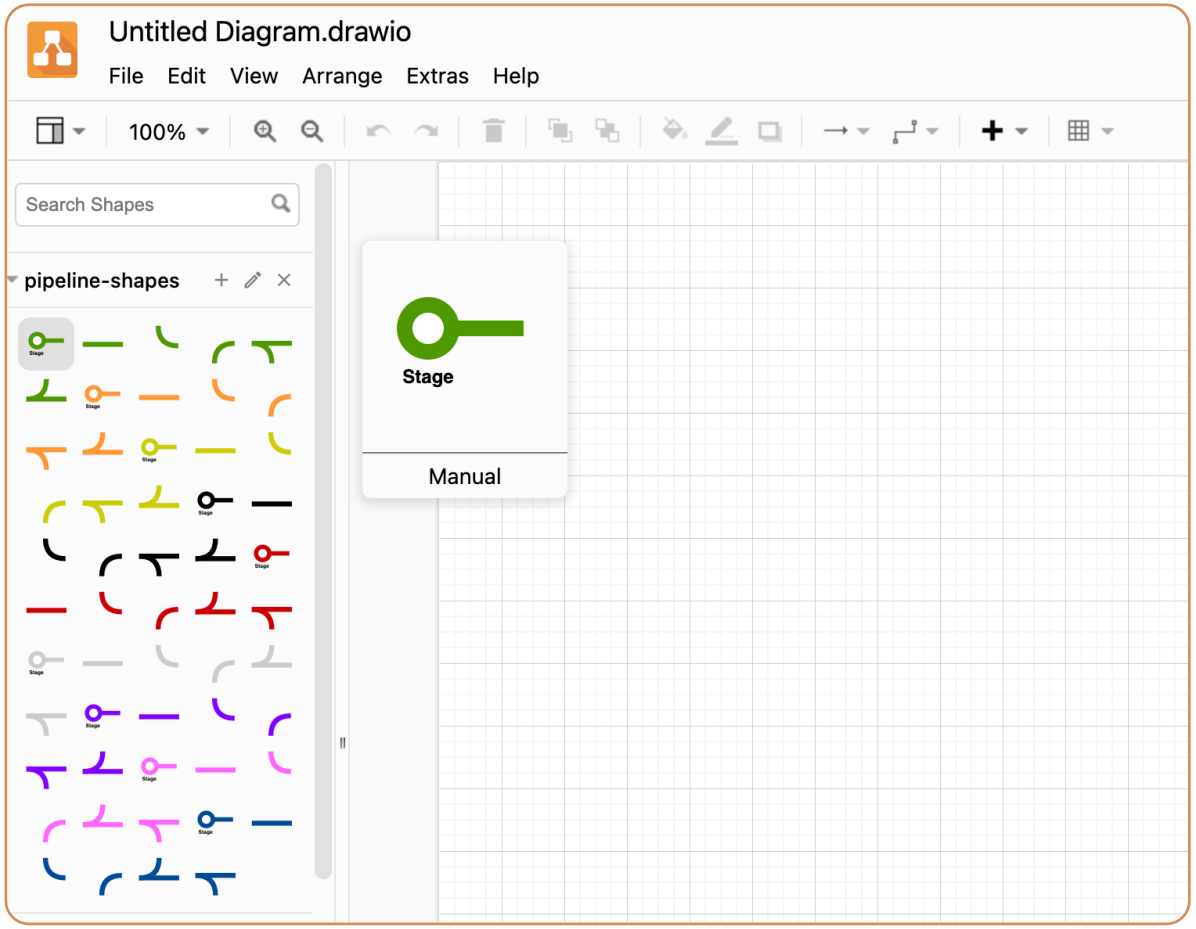
CULTURE CHANGE

Ban the infinity graph – DevOps and security is always a pipeline.

KERR

@edgeroute

devsecops-architecture-tools



https://github.com/djschleen/devsecops-architecture-tools

Failure #2

Security as a special team

KERR

@edgeroute

Security and coding is for everyone

Security answers...

Source: The GitLab 2022 Global DevSecOps Survey

Culture Change

- Lean into the fact that developers want to be responsible for security.
- Responsibility requires knowledge and experience.
- Teach everyone the foundational lessons of security.
- Teach security to code.

KERR

@edgeroute

Failure #3

Vendor defined DevOps

KERR

Embrace the reality of YOUR DevOps

Culture Change

- Realize that DevOps is not DevOps is not DevOps.
- Don’t let vendors define DevOps within the context of their solution.
- Study and teach the bigger, contextual reality of vendor-independent DevOps.

Failure #4

Big company envy

KERR

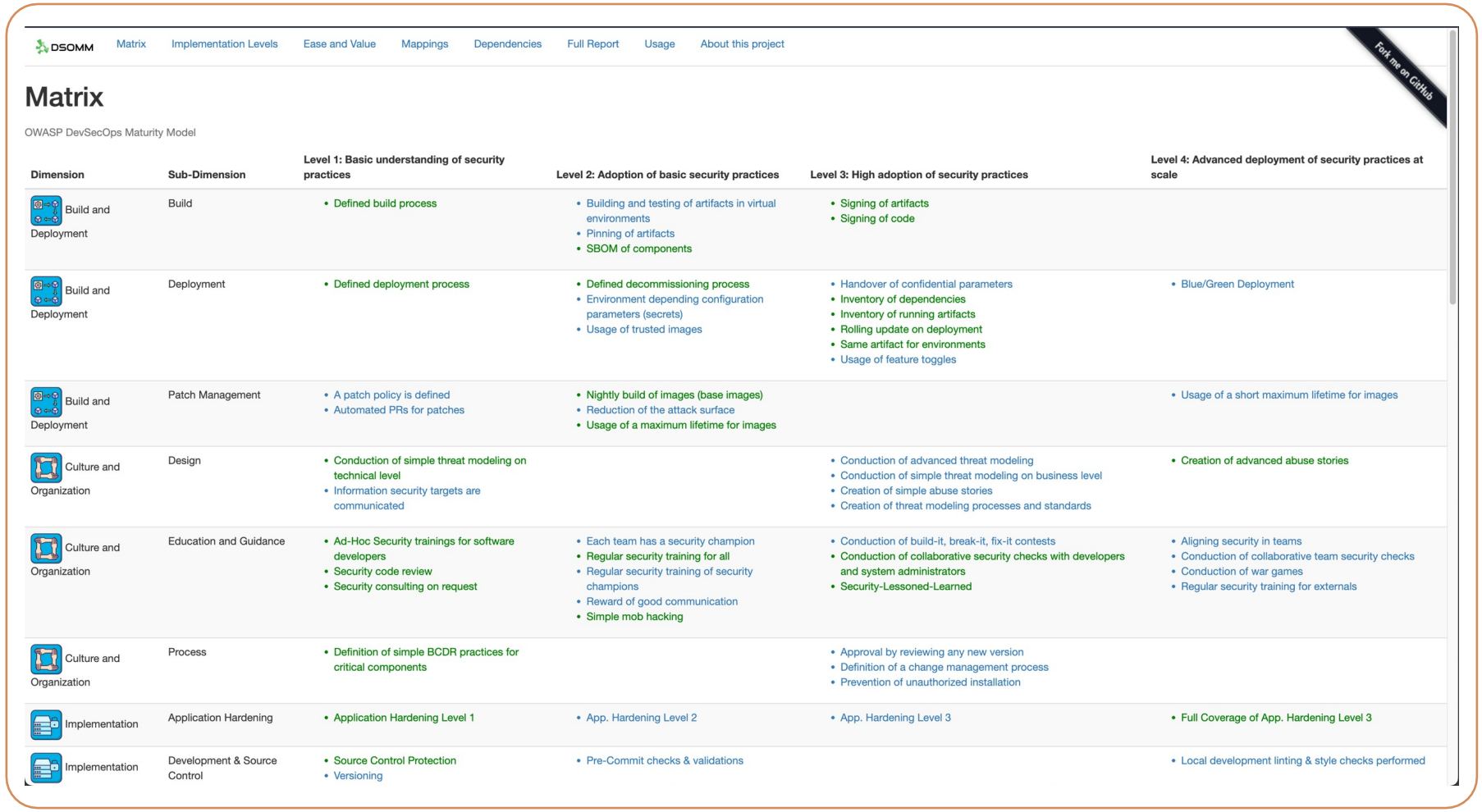
@edgeroute

Do the DevOps that you do

Culture Change

- Avoid distracting yourself by fixating on what those at the ”top” of the industry are doing with security.
- Make incremental progress; a tool here, a tool there.
- Cast a vision for where your DevOps with security needs to go.
- Use the OWASP DevSecOps Maturity Model to create a roadmap.

OWASP DevSecOps Maturity Model



DSOMM Matrix Implementation Levels Ease and Value Mappings Dependencies Full Report Usage About this project

Fork me on GitHub

Matrix

OWASP DevSecOps Maturity Model

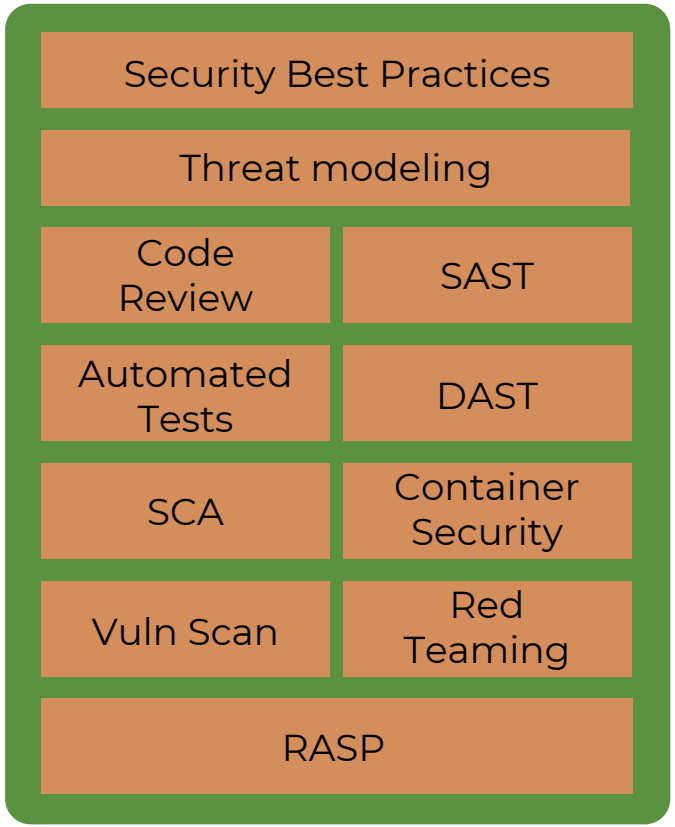
Dimension	Sub-Dimension	Level 1: Basic understanding of security practices	Level 2: Adoption of basic security practices	Level 3: High adoption of security practices	Level 4: Advanced deployment of security practices at scale
Build and Deployment	Build	• Defined build process	• Building and testing of artifacts in virtual environments • Pinning of artifacts • SBOM of components	• Signing of artifacts • Signing of code	
Build and Deployment	Deployment	• Defined deployment process	• Defined decommissioning process • Environment depending configuration parameters (secrets) • Usage of trusted images	• Handover of confidential parameters • Inventory of dependencies • Inventory of running artifacts • Rolling update on deployment • Same artifact for environments • Usage of feature toggles	• Blue/Green Deployment
Build and Deployment	Patch Management	• A patch policy is defined • Automated PRs for patches	• Nightly build of images (base images) • Reduction of the attack surface • Usage of a maximum lifetime for images		• Usage of a short maximum lifetime for images
Culture and Organization	Design	• Conduction of simple threat modeling on technical level • Information security targets are communicated		• Conduction of advanced threat modeling • Conduction of simple threat modeling on business level • Creation of simple abuse stories • Creation of threat modeling processes and standards	• Creation of advanced abuse stories
Culture and Organization	Education and Guidance	• Ad-Hoc Security trainings for software developers • Security code review • Security consulting on request	• Each team has a security champion • Regular security training for all • Regular security training of security champions • Reward of good communication • Simple mob hacking	• Conduction of build-it, break-it, fix-it contests • Conduction of collaborative security checks with developers and system administrators • Security-Lessoned-Learned	• Aligning security in teams • Conduction of collaborative team security checks • Conduction of war games • Regular security training for externals
Culture and Organization	Process	• Definition of simple BCDR practices for critical components		• Approval by reviewing any new version • Definition of a change management process • Prevention of unauthorized installation	
Implementation	Application Hardening	• Application Hardening Level 1	• App. Hardening Level 2	• App. Hardening Level 3	• Full Coverage of App. Hardening Level 3
Implementation	Development & Source Control	• Source Control Protection • Versioning	• Pre-Commit checks & validations		• Local development linting & style checks performed

https://owasp.org/www-project-devsecops-maturity-model/

Failure #5

Marketing term infatuation

Security front, right, left, and center



Culture Change

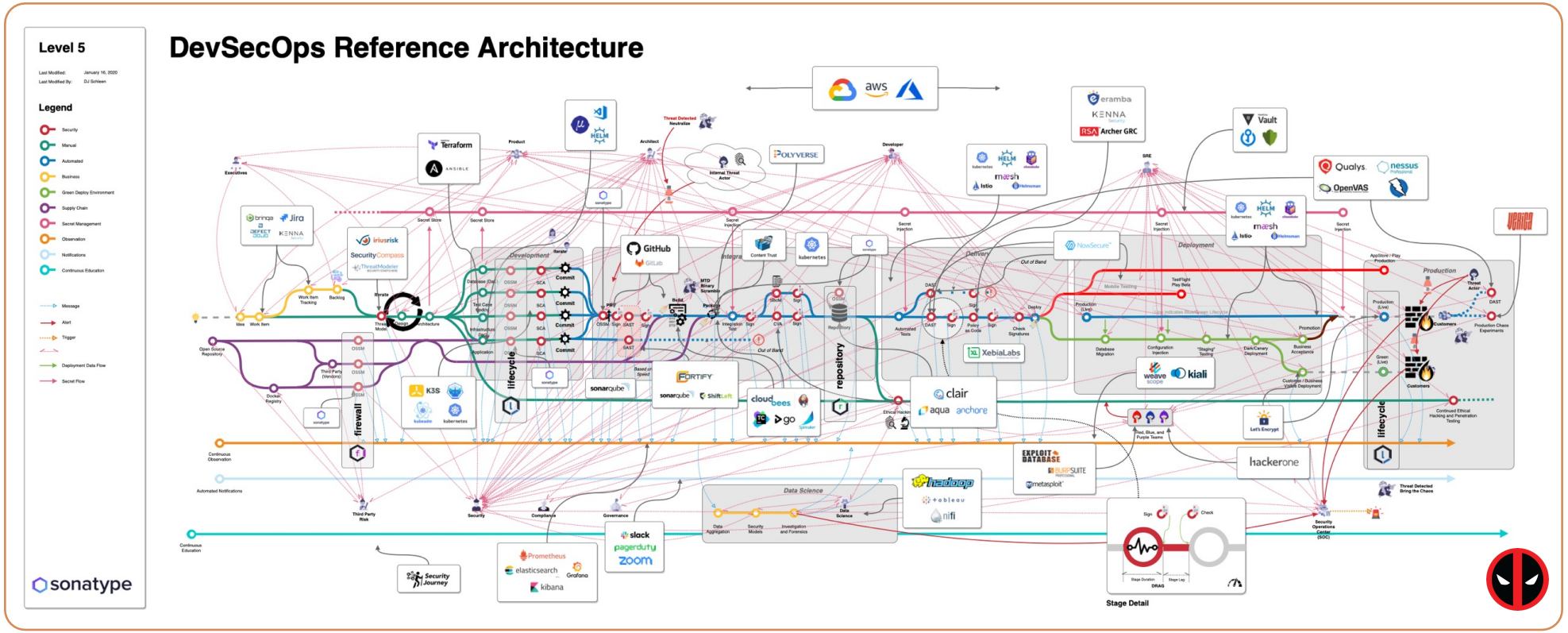
- Shift/start left by embedding security into every section of your DevOps pipeline.
- Shift/start right by integrating RASP into production.
- Please meet in the middle and call it a secure development lifecycle.

KERR

@edgeroute

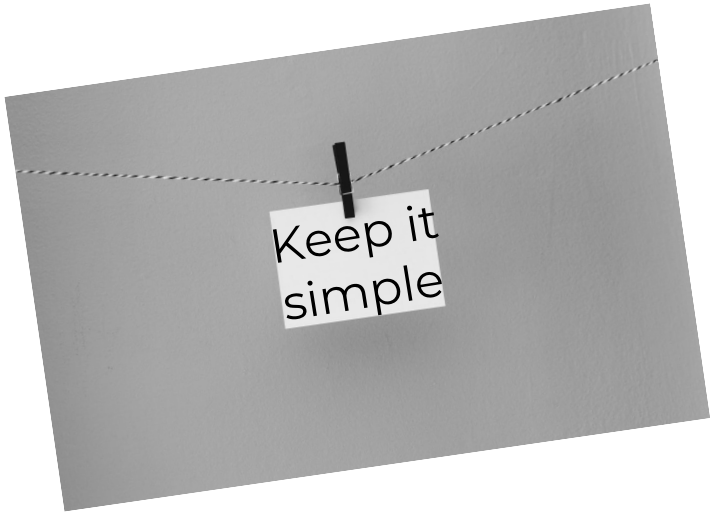
Failure #6

Overcomplicated pipelines and doing everything now



Source: https://blog.sonatype.com/2020-devsecops-reference-architecture

Start simple, live simply



Culture Change

- Realize that there are dozens of categories of security tools.
- Start with a small subset of security tools and grow the pipeline over time.
  - SCA or SAST
- Keep it simple – the pipeline must be simple enough that all product and application adjacent people can explain it.
- Avoid the pressure to solve all problems immediately. Take a phased approach.

Failure #7

Security as gatekeeper

KERR

Drop the no; try yes, if...

Culture Change

- Practice empathy.
  - Sec → Dev
  - Ops → Sec
  - Dev → Ops
  - * → *
- Work together to find solutions, not roadblocks.
- Provide Coaches from each discipline.

Failure #8

Noisy security tools

KERR

@edgeroute

Lower the noise

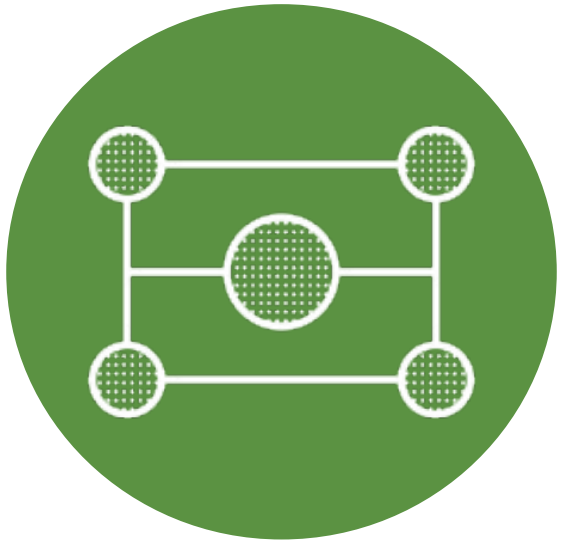
Culture Change

- Tune the existing security tools that exist in the pipeline.
- NEVER waste ANYONE's time with security findings that do not matter.
- When adding new tools, start with a minimal policy.

Failure #9

Lack of threat modeling

Threat model



Culture Change

- Perform threat modeling outside the scope of the pipeline.
- Connect threat modeling to your approach for doling out new feature assignments.
- Teach everyone how to threat model.

THREAT MODELING MANIFESTO

What is threat modeling?

Threat modeling is analyzing representations of a system to highlight concerns about security and privacy characteristics.

At the highest levels, when we threat model, we ask four key questions:

1. What are we working on?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good enough job?

Why threat model?

When you perform threat modeling, you begin to recognize what can go wrong in a system. It also allows you to pinpoint design and implementation issues that require mitigation, whether it is early in or throughout the lifetime of the system. The output of the threat model, which are known as threats, informs decisions that you might make in subsequent design, development, testing, and post-deployment phases.

Who should threat model?

You. Everyone. Anyone who is concerned about the privacy, safety, and security of their system.

How should I use the Threat Modeling Manifesto?

Use the Manifesto as a guide to develop or refine a methodology that best fits your needs. We believe that following the guidance in the Manifesto will result in more effective and more productive threat modeling. In turn, this will help you to successfully develop more secure applications, systems, and organizations and protect them from threats to your data and services. The Manifesto contains ideas, but is not a how-to, and is methodology-agnostic.

The Threat Modeling Manifesto follows a similar format to that of the Agile Manifesto by identifying the two following guidelines:

- **Values**: A value in threat modeling is something that has relative worth, merit, or importance. That is, while there is value in the items on the right, we value the items on the left more.
- **Principles**: A principle describes the fundamental truths of threat modeling. There are three types of principles: (i) fundamental, primary, or general truths that enable successful threat modeling, (ii) patterns that are highly recommended, and (iii) anti-patterns that should be avoided.

Failure #10

Vulnerable code in the wild

SCA as launch control

Culture Change

- Embed SCA into ALL your pipelines.
  - Commercial and open-source options are available.
- Set SCA policy to fail on vuln detection.
- Ensure you have a time-boxed filtering policy.

The **10** DevSecOps security successes

1. Pipelines
2. Security for everyone
3. Do the DevOps YOU do
4. Content with YOUR DevOps
5. Security front, right, left, and center
6. Simple and staged pipeline
7. Security as trusted partner
8. Tuned and valuable security tools
9. Threat modeling for everyone
10. Breaking the build for vulns

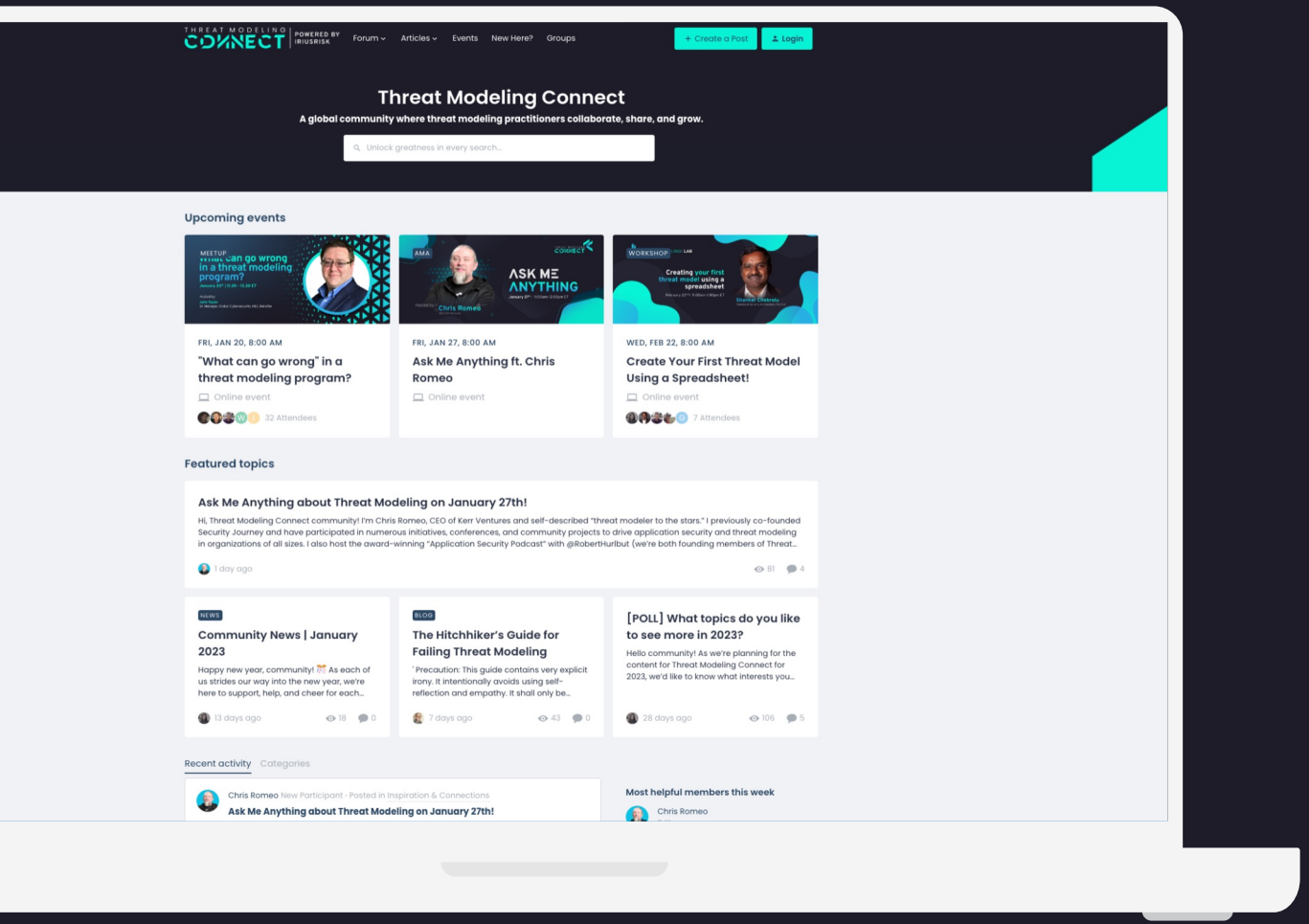
KERR

@edgeroute

Key Takeaways

- The impact of DevOps fifty years from now is on the security culture.
- Some of the failures were outside of your control, but you can change them now (ex. infinity graphs).
- Security and coding are for everyone. Embrace YOUR DevOps, keep it simple, lower the noise, be a partner, threat model, and break the build for vulns.
- Embrace and laugh at the failures.

THREAT MODELING CONNECT
POWERED BY IRIUSRISK



Let's continue the conversation at Threat Modeling Connect

An **open threat modeling community** where you can collaborate, share, and grow with practitioners worldwide through forum discussions, expert content, and events.

👉 Sign up now at threatmodelingconnect.com

Questions?

Chris Romeo
CEO, Kerr Ventures
chris@kerr.ventures

Listen to me
The Application Security Podcast
The Security Table

Talk to me
@edgeroute, @AppSecPocast

KERR

@edgeroute

OWASP 2023
GLOBAL
AppSec

DUBLIN
IRELAND
FEB 13-16

THANK YOU!