RSAConference2022
San Francisco & Digital | June 6 – 9

TRANSFORM

SESSION ID: **TECH-M03**

Building an Enterprise-scale DevSecOps Infrastructure: Lessons Learned

Prateek Mishra

Senior Director - Security Architecture
Developer Experience, ADP

Gaurav Bhargava

Director of Product Management
Developer Experience, ADP

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

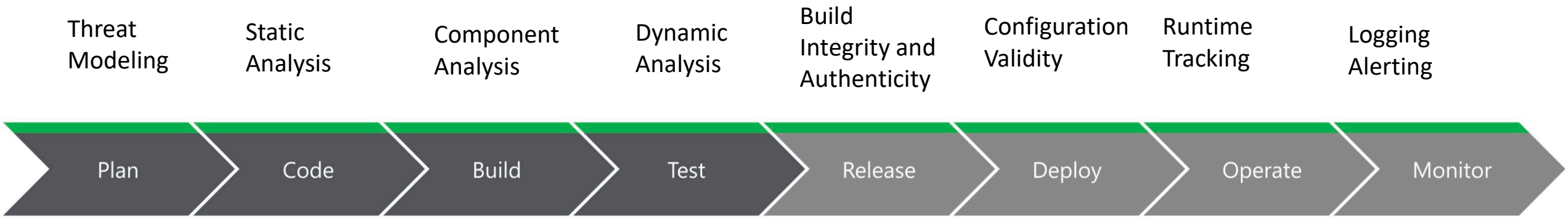
Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

Agenda

- What is DevSecOps?
- Enterprise DevSecOps Problem Statement
- Solution Proposal and its Characteristics
- Demo!
- Learnings
- Conclusion

What is DevSecOps?

- DevSecOps is the integration of security into emerging agile IT and DevOps development (and deployment) as seamlessly and as transparently as possible (Gartner)

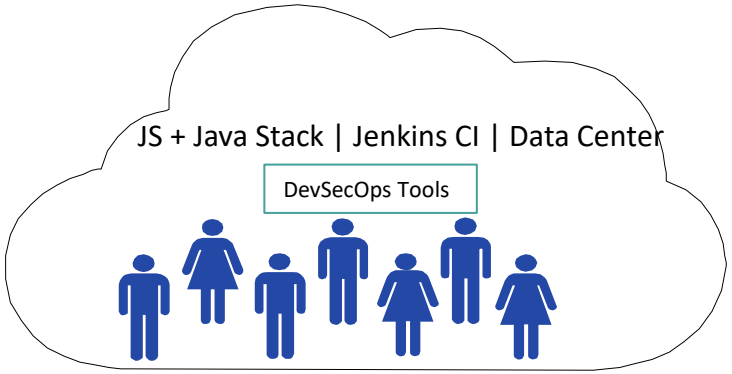


- Seems straightforward enough, so why do we need this session?

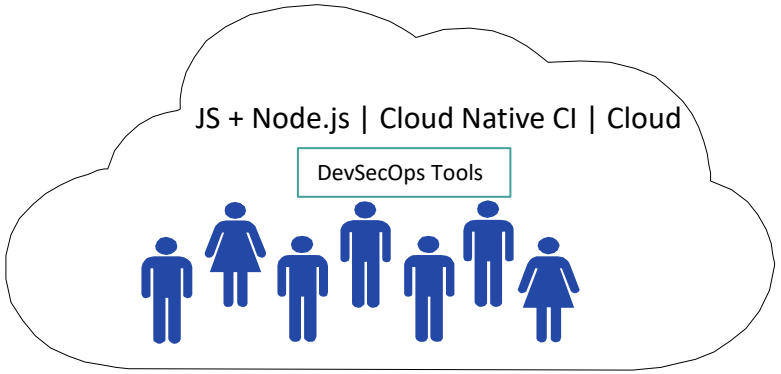
RSAConference2022

Problem Statement

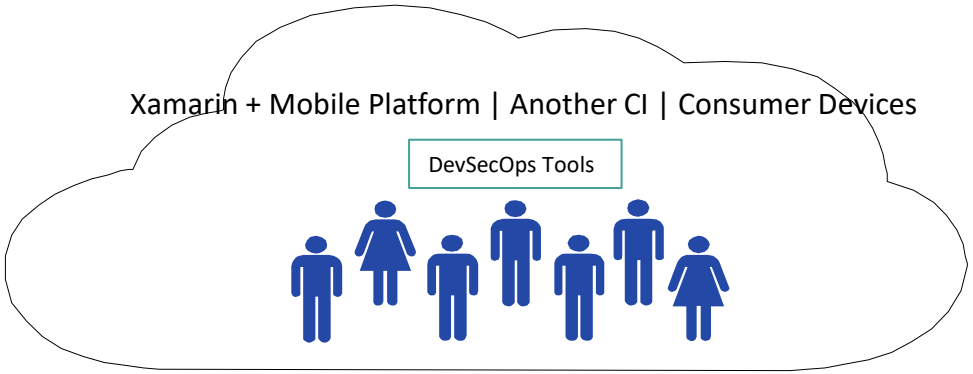
Problem Statement



Development Group A



Development Group B



Development Group C

- Many autonomous development groups (10s -100s), developing apps using several different technology stacks
- Some groups are working on well-established ”legacy” apps, others on next generation applications with modern tools
- Different approaches to detecting and closing security vulnerabilities: with every commit or build or at sprint boundaries or all of the above

Always Designing for People™

Challenge: Lack of Uniform View across Departments

- Lack of a uniform view and understanding of security state of applications
  - Difficult to assess and compare security posture of various products across departments
  - Difficult to enforce a uniform level of compliance with enterprise guidelines

- Cost of maintaining separate departmental infrastructures
  - Headcount/License/Training/Maintenance

Challenge: Managing Diversity of Scanners and Security Information Sources

- Numerous scanners and scanner types available
  - Open Source, Existing Enterprise Licenses for on-prem and SaaS models
  - Lightweight (take a few minutes) vs Exhaustive (may take hours to run)
- Selection of appropriate scanners for comprehensive coverage
  - SAST [Static Code Security] Scanner
  - SCA [Software Composition Analysis] Scanner
  - Embedded Secrets Scanner
  - DAST [Dynamic App Security] Scanner
  - Infrastructure mis-configurations (cloudformation, kube deploy, etc.)
- Ability to process vulnerability feeds from various input sources
- Coping with scanner noisiness/chattiness

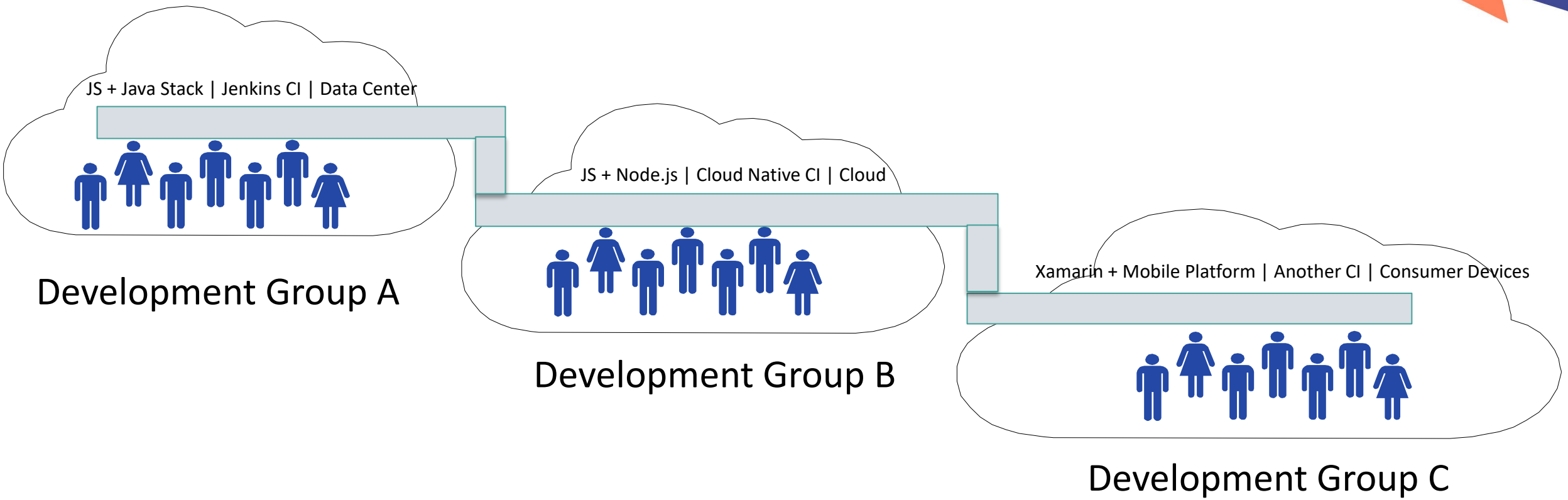
Challenge: Developer Enablement

- How to ensure that developers and development teams are tasked only with security vulnerabilities relevant to them?
- Rich set of application artifacts - git branches, repositories, build processes, docker containers, application assemblies, cloud accounts - create difficulties in linkage to devs or dev teams
- Lack of information sharing between development teams
- Actionability of remediation guidance by app developers
  - App developers are not security experts!!
- Process inefficiencies in consultation between development teams and CSO (Security SMEs)

RSAConference2022

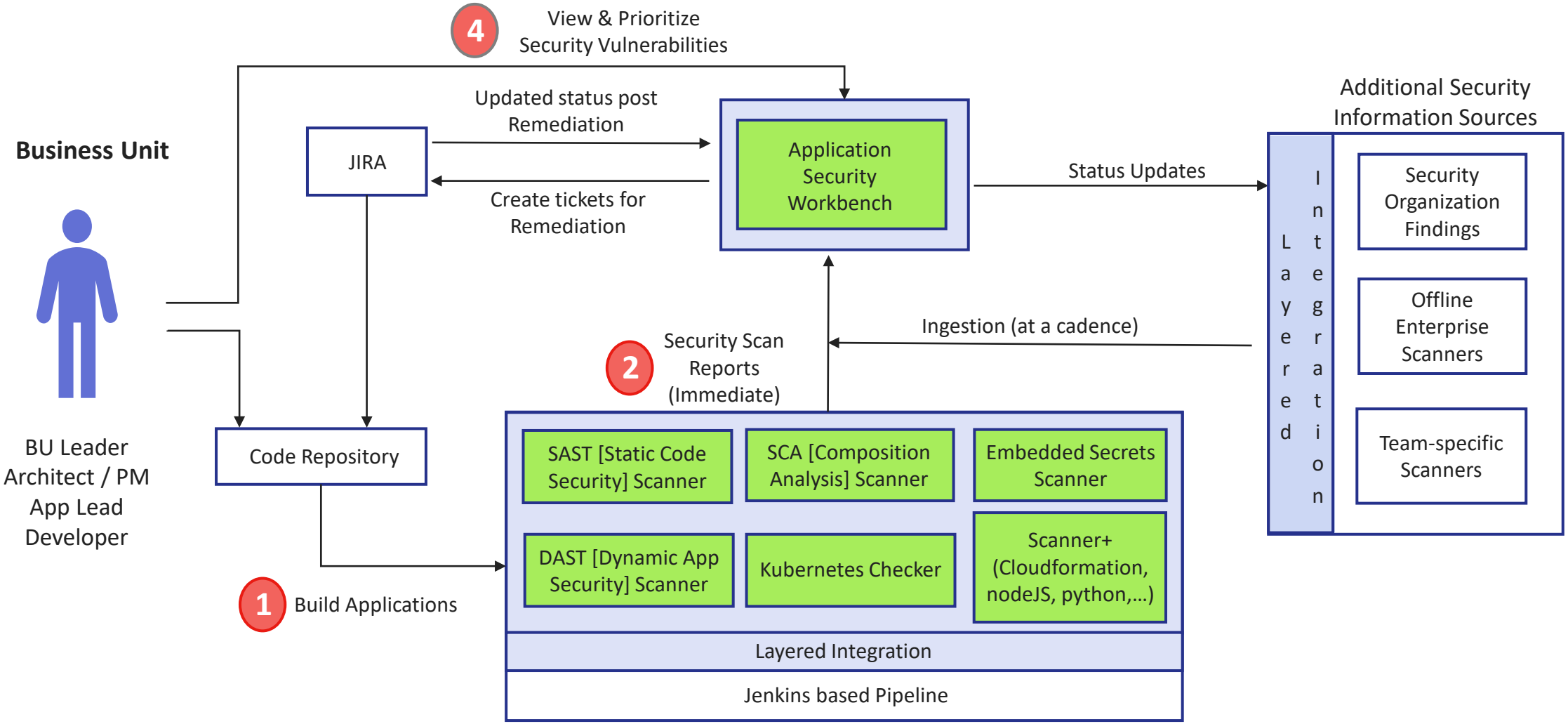
Solution

Solution: Enterprise-scale DevSecOps



- Shared infrastructure “plugs-in” to the specific development tech stacks and CI+CD frameworks used by different groups
- Layered architecture accommodates new tech stacks, languages and target platforms
- Ability to process and manage security vulnerabilities from diverse scanners and security information sources

High Level Logical Architecture



Solution

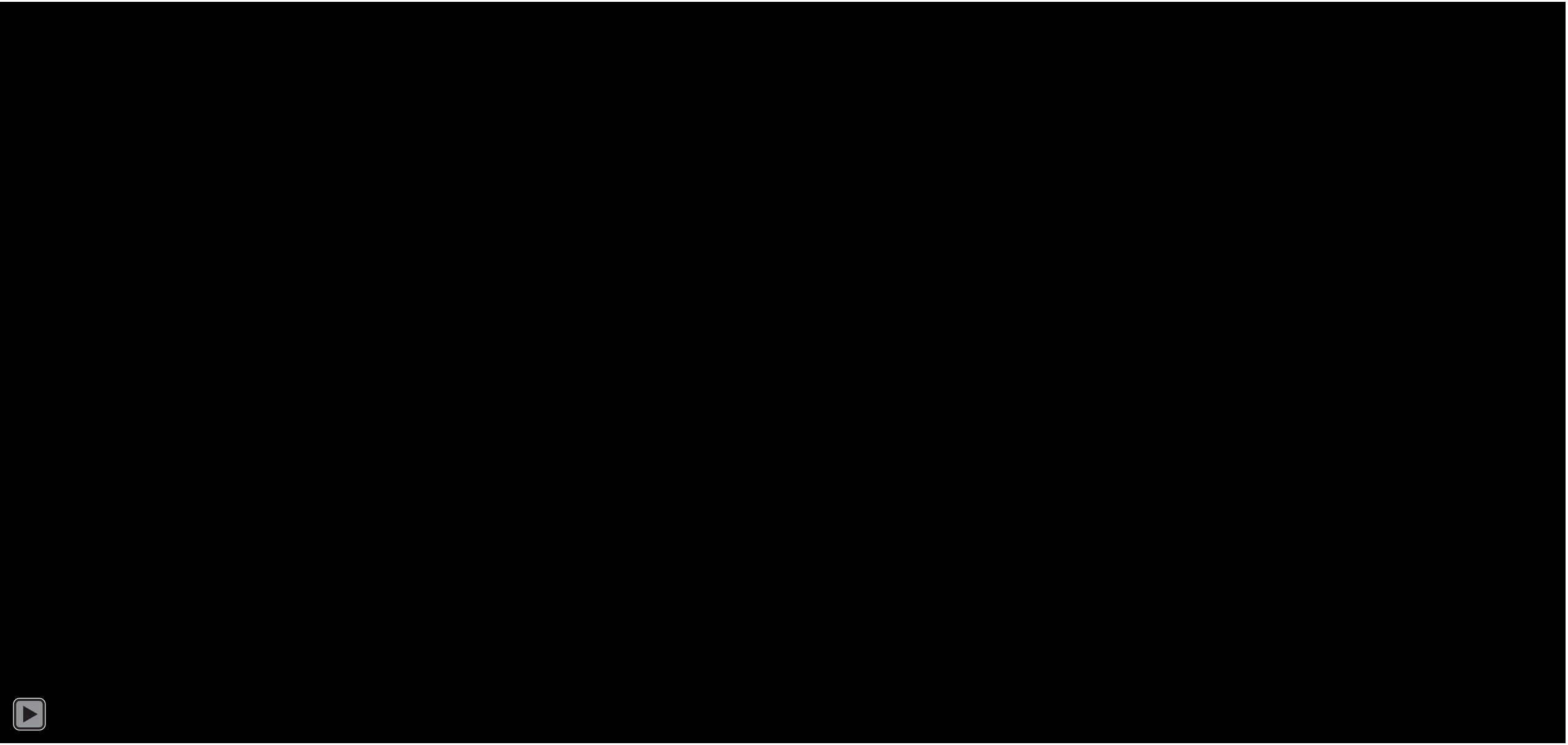
- Software framework built out of stock components and open source
  - Scanner layer provides a uniform way to package scanners into docker images
    - We selected a standard set of open source scanners familiar to us
    - Can be replaced by alternatives or licensed versions as needed
  - Dockerized scanners can be plugged into many different CI pipelines
    - Current focus is on Jenkins-based pipelines, but others are in our backlog

- Linkage between repositories/projects to products and teams
  - Based on machine readable meta-data added by teams to repositories/projects
  - Ensures that code/artifacts/assemblies/docker images/cloud accounts can be linked to products and teams

Solution (contd.)

- Workbench provides a generic data model and uniform GUI for all security vulnerabilities
  - Allows new scanners or security information sources to be added as needed
    - Including off-line scanners and security information sources that are available asynchronously
  - Standardized display of vulnerabilities
    - Severity, Remediation Guidance, False Positives, Acceptable Risk
    - Main focus is helping application developers to act on the information
      - Including ways of sharing information between teams acting on similar vulnerabilities
- Workbench provides division/product level rollups
  - Fine grained vulnerability reports at repositories/artifact level
  - Aggregate vulnerabilities at department/product level
  - Exposes extent of security maturity across division/products

Video Demo



RSAConference2022

Learnings

Apply #1 [Immediate]

- Agree on an enterprise-wide approach with all stakeholders (Dev teams, leadership, CSO office)
  - Identify DevSecOps efforts that are on-going in different teams and capture their requirements
- Identify if there are any existing enterprise license agreements with security vendors
  - Supplement with use of well-respected open source systems
  - Security scanners are a commodity, examine vendor claims of superiority with caution!
- Agree on a base level set of security scanners and information sources
  - Getting off the ground is key; don’t try to achieve nirvana!!

Apply #2 [30-60 days]

- Ensure that the selected tools support varied team development methodologies and different tech stacks
  - This is why a pluggable framework is important
  - This will also guide your BUY vs BUILD decision
- Agree on how security vulnerabilities should be surfaced
  - E.g., Webapp, webex, email, CI/CD links, bitbucket annotations
- Identify and engage with early adopters / champions
  - Establish regular feedback mechanism

Apply #3 [180 days]

- Develop a process for remediation timelines and priority
  - Who determines impact and risk and how will it be manifested by your tools?
  - Not every security issue can be fixed easily or quickly, important to have a tracking process
- Culture shift through office hours/trainings
  - Developers to become familiar with security vulnerability patterns common within the enterprise
  - Train developers on becoming proficient at remediating these vulnerabilities

RSAConference2022

Conclusions

Closing Thoughts, Credits and Demerits

- Enterprise-scale DevSecOps requires going beyond selecting or creating a toolset
  - Culture change for both Development Teams and Security SMEs
  - Requires process changes in Development Teams and movement away from a model of “security as punishment/shaming”
- Credits
  - OWASP organization and community esp. open source tools
  - Vendors supporting open source tools (SonarQube, Anchore - Grype/Syft)
- Demerits
  - Vendors with unrealistic and unreasonable claims pushing proprietary solutions