Secure by Design

Automating Security in Agile Development
About the presenter

Brendan Lafond

- Director of Agile Transformation at Wesco with more than 16 years of experience in the IT industry
- Background in support, development, product management, and Agile transformation
- Current role is driving DevSecOps evolution at Wesco
- Leads a platform engineering and coaching team
- Responsible for leading the enhancement, adoption, and strategy for Wesco's CICD platform and integration of secure by design best practices
About Wesco

Wesco is a world leader in electrical, communications and utility distribution and supply chain services, we’re ready and able to help you navigate business complexities.

$21+ Billion
2022 net sales

Operations in 50+ Countries
to serve you wherever you are

3 Strategic Business Units
with dedicated focus & expertise

Environmental, Social & Governance
corporate focus and commitment

Digital Investments
to build for the future

Fortune 500 Corporation
To lead through challenging supply chain and economic conditions (#181)
Background

- Required code vulnerability scans occurred too close to a production release
- If vulnerabilities were discovered, developers would not know until the code was frozen for production release
- Developers would need to unfreeze the code to apply the necessary fixes and scan again
- This was viewed as ‘unplanned’ work that could potentially delay production releases by several weeks

We needed a better way to meet business and security requirements.
Objective

• Shift security scans left to identify and resolve vulnerabilities quickly
• Perform static application and software code analysis scans on every code commit
• Perform dynamic application scans in test environments on a repeating schedule
• When vulnerabilities are identified, break build or prevent deployment
• This process needs to be scalable and code agnostic to support our diverse business needs

Align expectations with key stakeholders to improve the process
Approach

- Added security scans and rules into the CICD workflow to break builds and prevent deployments
- Developed GitHub Actions and User Interface to abstract underlying functionality from Developers
- Integrated best in class vulnerability scanning for code and container deployment
- Integrated Jfrog Artifactory for clean code artifact storage
- Developed new tools to expedite onboarding and implementation of the CICD process to existing platforms across the company

We are working with GitHub to introduce additional rulesets in a future release
Measuring Success

• Developers are receiving security feedback within 15 minutes of every code commit, which could be several times per day
• This continuous feedback is training Developers to produce better code from the start
• Significant reduction of vulnerabilities found in all environments
• Project delays due to security findings are now a rare occurrence
• Production release is down from over 4 hours to less than 15 minutes

Better designs, faster cycles, and significantly fewer security concerns!!!
Challenges

• Scaling the platform to accommodate the 80+ major initiatives as well as other smaller projects
• Supporting multiple programming languages that enable our complex environment
• Managing the ongoing lifecycle of the third-party dependencies in our applications

There was not a commercially available solution to meet our needs
Lessons Learned

• Integration of scans into the CICD workflows requires:
  – Developer education on scan types, when to use each scan, how to mitigate identified vulnerabilities
  – Creation or modification of existing policies to promote timely resolution

• Open-source software, while viable, has stability challenges requiring routine maintenance, contributions, and governance
  – Leverage commercially available solutions where possible

• Build strong vendor partnerships to further enhance their products to meet your needs and grow the community

• Make sure you have clear business requirements and policy alignment to resolve the most significant concerns

Veracode partnership has led to open-source contributions and enhancements
Special Thanks!

- Dave Rask
- Boris Nakhlis
- Jason Kratz
- Prathap Motupalli
- Rahul Tongase
- Scott Ballard
- Dharmayu Purohit
- Vipul Pant
- Manjusha Saju
- Anantha Ranganathan
- Hansraj Sao

They made this project possible with their dedication and hard work!