Beyond Firewalls: Application Security

The DevSecOps Approach to Elevating Application Security

Presented by:
Lakshman Kaveti
Ben Wilcox

Limitations of Conventional Security Approaches

Manual testing

Manual testing of code is slow, error-prone and unable to keep up with rapid releases.

Firewalls

Firewalls operate at network perimeter and can't see threats from within.

Vulnerability Scanning

Scanning identifies issues but doesn't provide guidance on fixes.

Traditional security methods like firewalls and manual testing have limitations in today's dynamic environments.

How DevOps Speed and Agility Impact Security

- **Faster software delivery**

  DevOps aims to deliver software faster through CI/CD pipelines and automation

- **Rapid deployment**

  Focus on speed can lead to cutting corners on security

- **Automation**

  Tools like IaC can enable automated security checks

- **Complex architectures**

  Microservices and distributed systems expand the attack surface

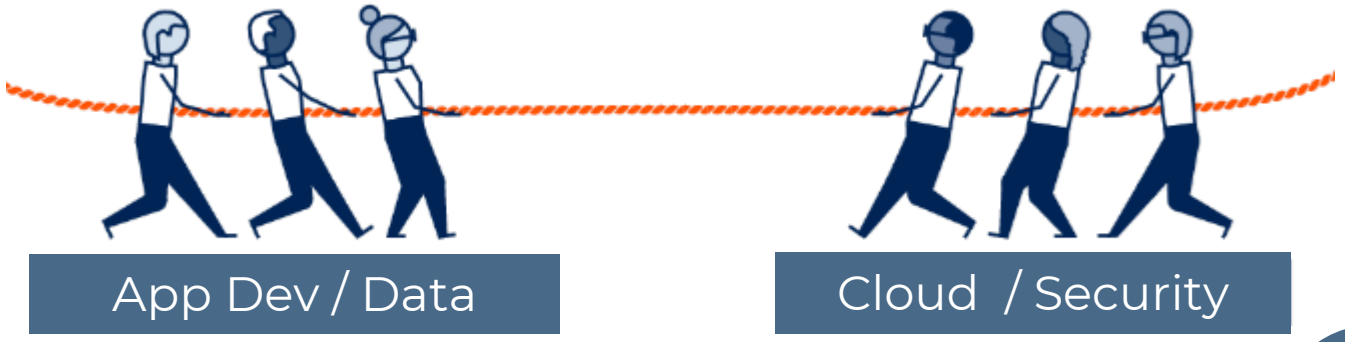
- **Shared responsibility**

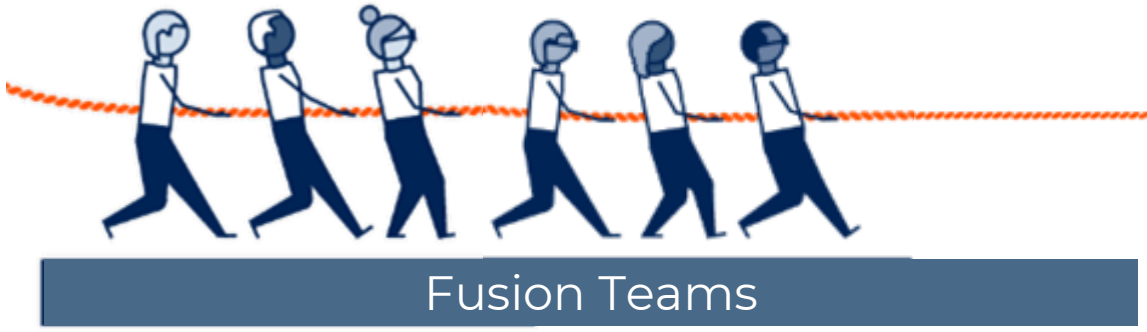
  Security shifts left and becomes everyone's job

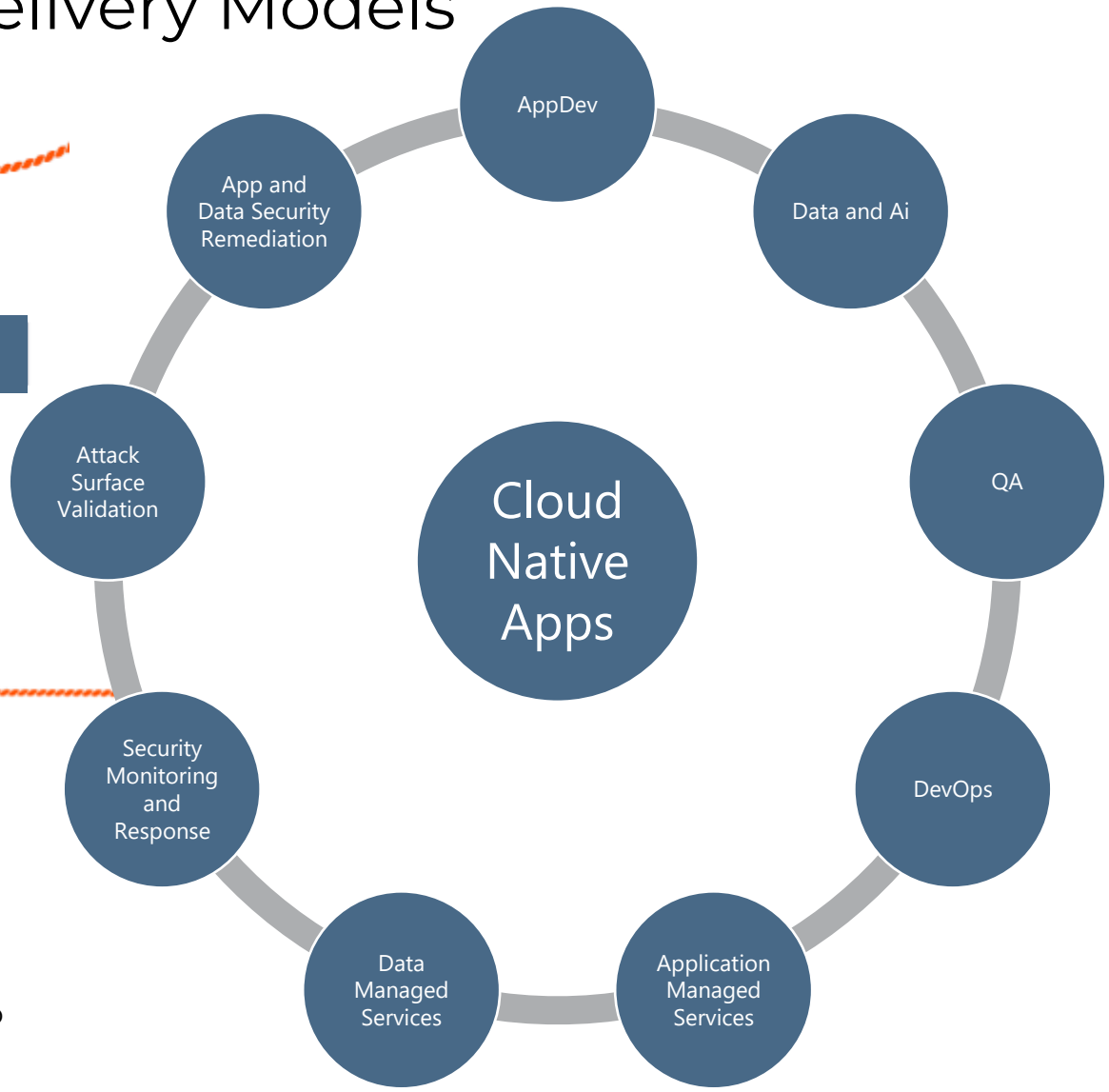
Threats to Data Security

- Software Supply Chain (SSC) attacks
- Public Cloud Infrastructure and Platform vulnerabilities
- Human misconfigurations and errors
- Commonly used libraries exploitable
- Commercially exploitable platforms
- Interconnectivity and dependencies on APIs
- Stolen credentials and code leaks
- Insider and 3rd party threats

Cloud Native Apps Requires New Delivery Models



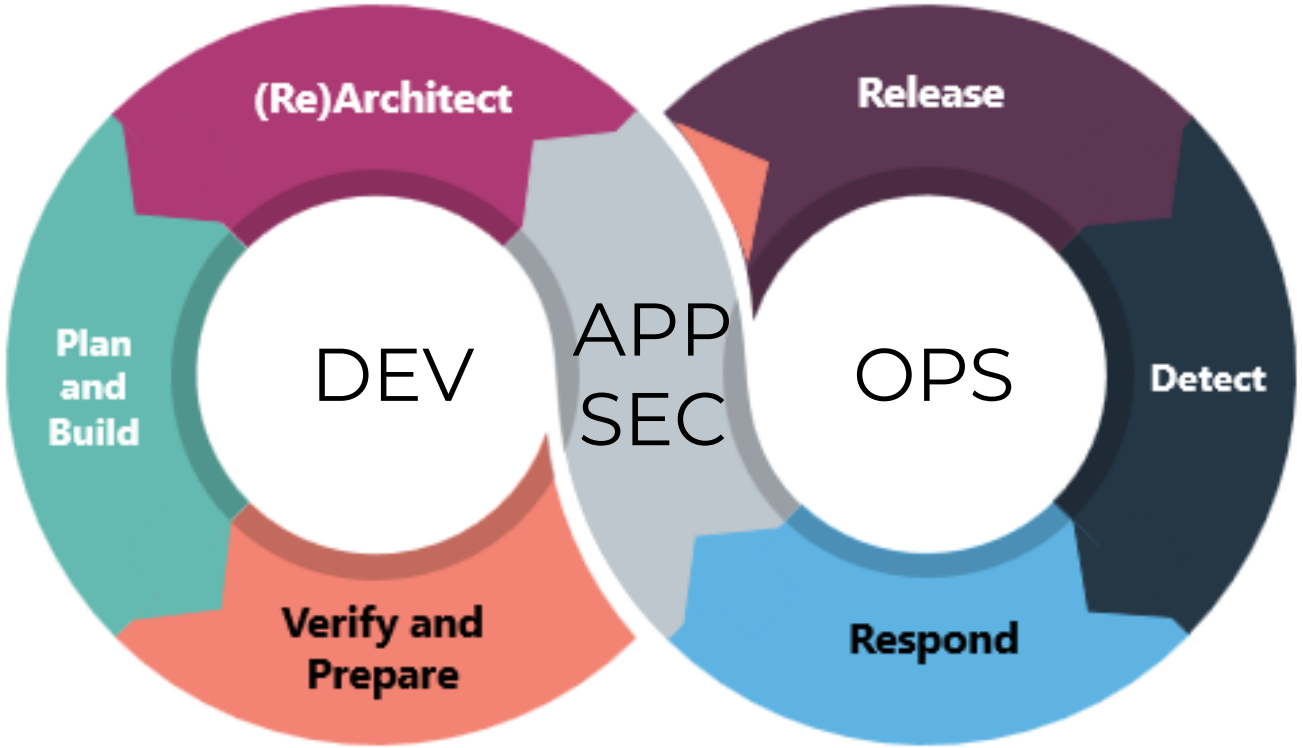




Shift to Cloud Native is driving a need for convergence of teams to work on the solution. Fusion Team members include Business, Cloud, Security, App Dev and Data stakeholders and operational members

DevSecOps

- Security should be integral to the Application Life Cycle, not just part of SDLC or Operations.
- In current Agile-style digital product development, new features and technical components are introduced as the application is built and maintained.
- The Dev Team, App Sec Team, and Operations teams need to work cohesively to ensure the application is safe and secure to use.
- Speed or Agility of DevOps is not reduced



DevSecOps Outcomes

Security Team

Quit stopping, participate and unblock

Operations Team

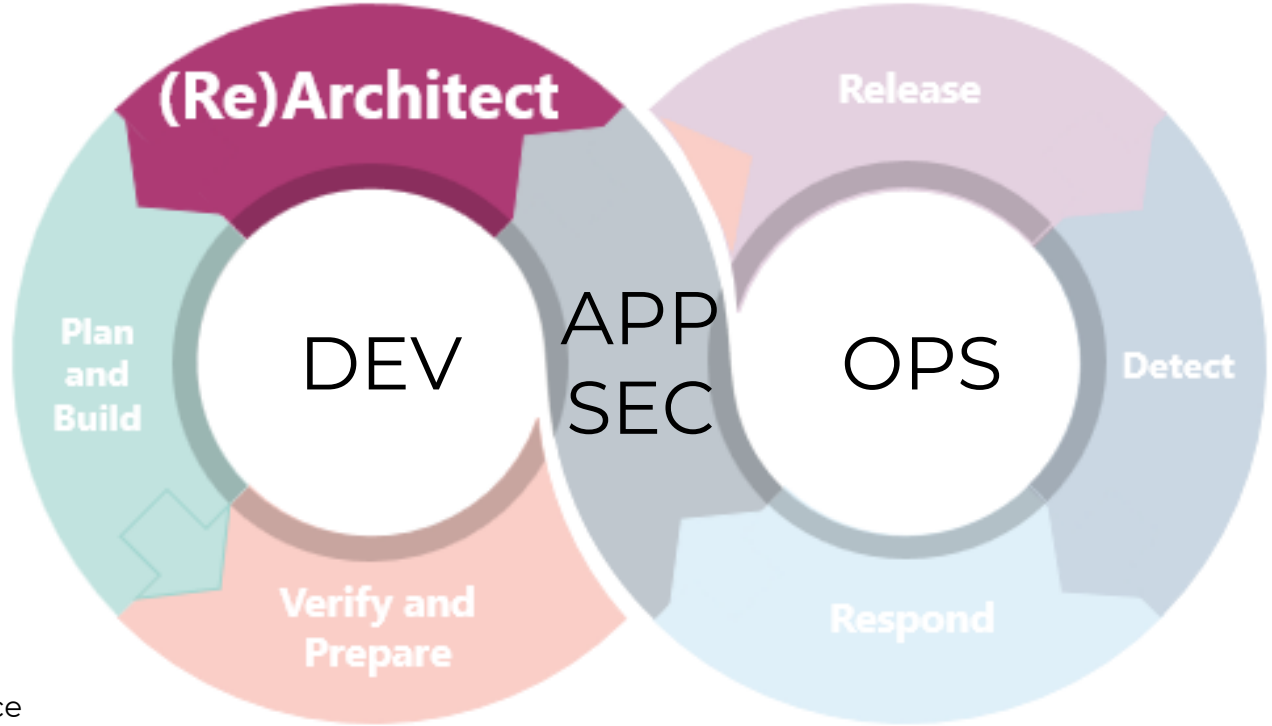
Enable success and shared responsibility in security.

Development Team

Enablement of a security mindset
Prioritize security requirements in backlog

DevSecOps - Architect

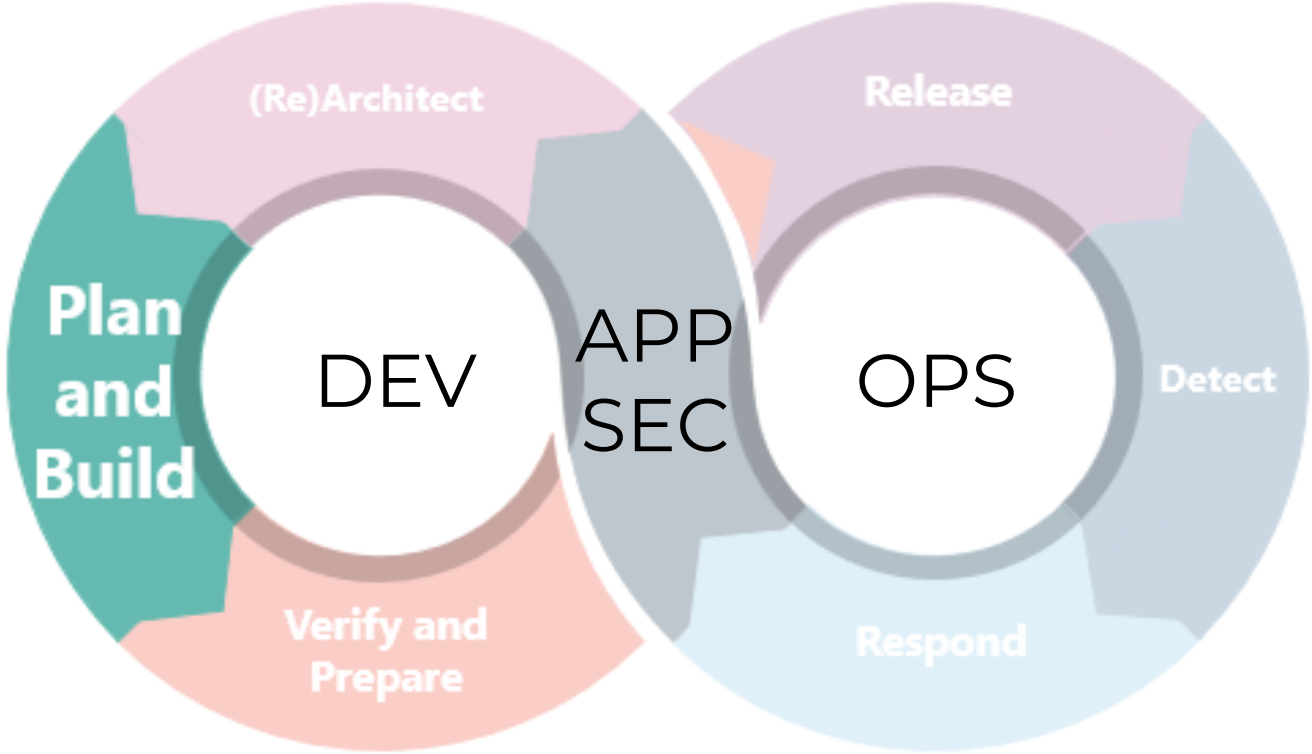
1. Application Security Components
   - IDAM and RBAC
   - Interoperability mechanisms
   - Secure communication
   - Input validation
2. Data Security
   - IDAM-backed data security
   - Data encryption at rest
   - Row-level and column-level security
3. Well-Architected
   - Business Requirements
   - Zero-Trust Principals
   - Cloud Security Posture Management - Defender for cloud security score hardening
   - Best practices: Key Vaults, Managed Identities, IAM, Service Principals, etc.



Although Architects are involved in the early stage of the application, as application matures and new features are being added, integrations and interoperability are setup, a review and refresh of Architecture is critical.

DevSecOps – Plan & Build

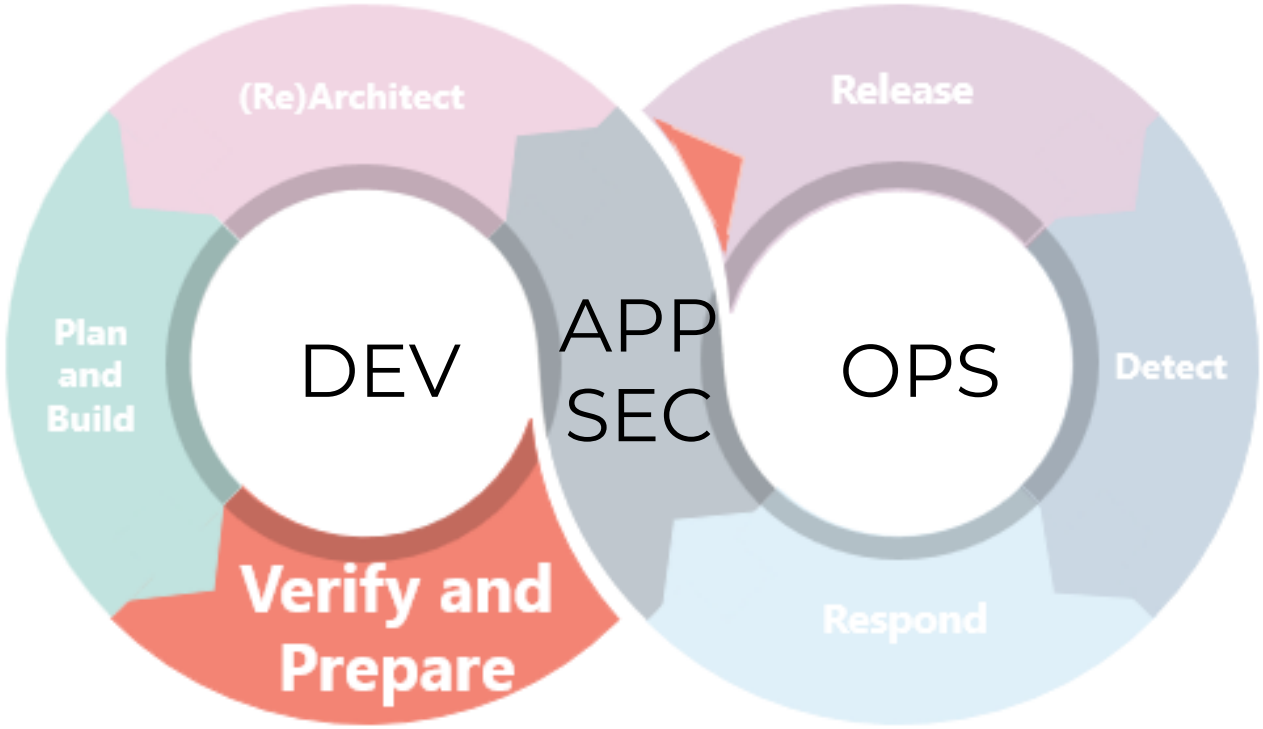
- Create and/or update Software Bill of Materials (SBOM)
    - Extend build to hashing and attestation to prevent tampering and prove policy adherence
    - Ensure considered SBOM components are still safe and has CISO's blessing
- Use IDE Security Plugins (resharper, etc.)
- Least Privilege planning and verification
- Ensure code reviews by the AppSec team and invit CISO and AppSec team to Show & Tells and Split Planning sessions
- Add SAST to the CI process



As you are building the product or making changes, ensure proper planning and leverage and consider security tools and best practices

DevSecOps – Verify & Prepare

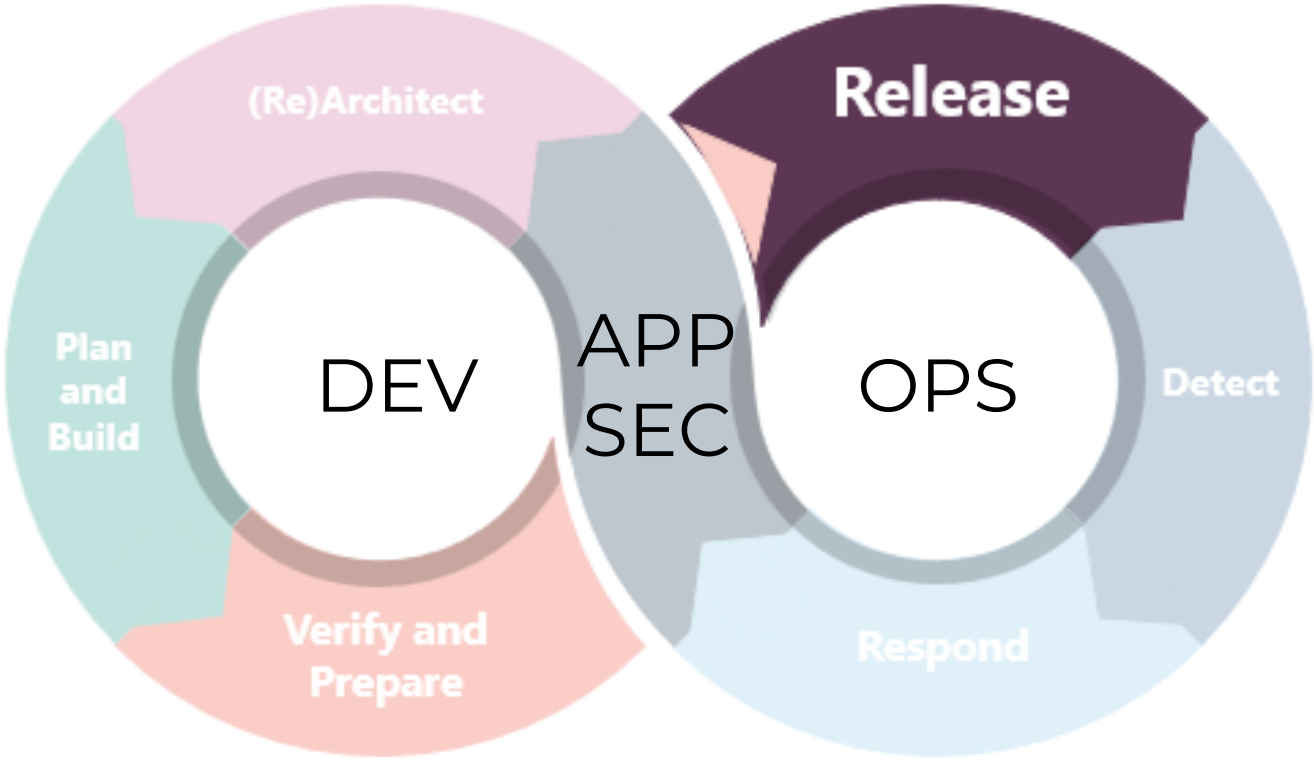
1. DAST – Dynamic Application Security Testing
2. IAST – Interactive Application Security Testing
3. SCA – Software Composition Analysis
4. Chaos Monkey, Input Fuzzing, Integration/Interoperability Testing
5. Software is obfuscated and signed
6. Penetration Testing



Once the application artifacts are created, verify build for security along with functionality in tandem

DevSecOps – Release

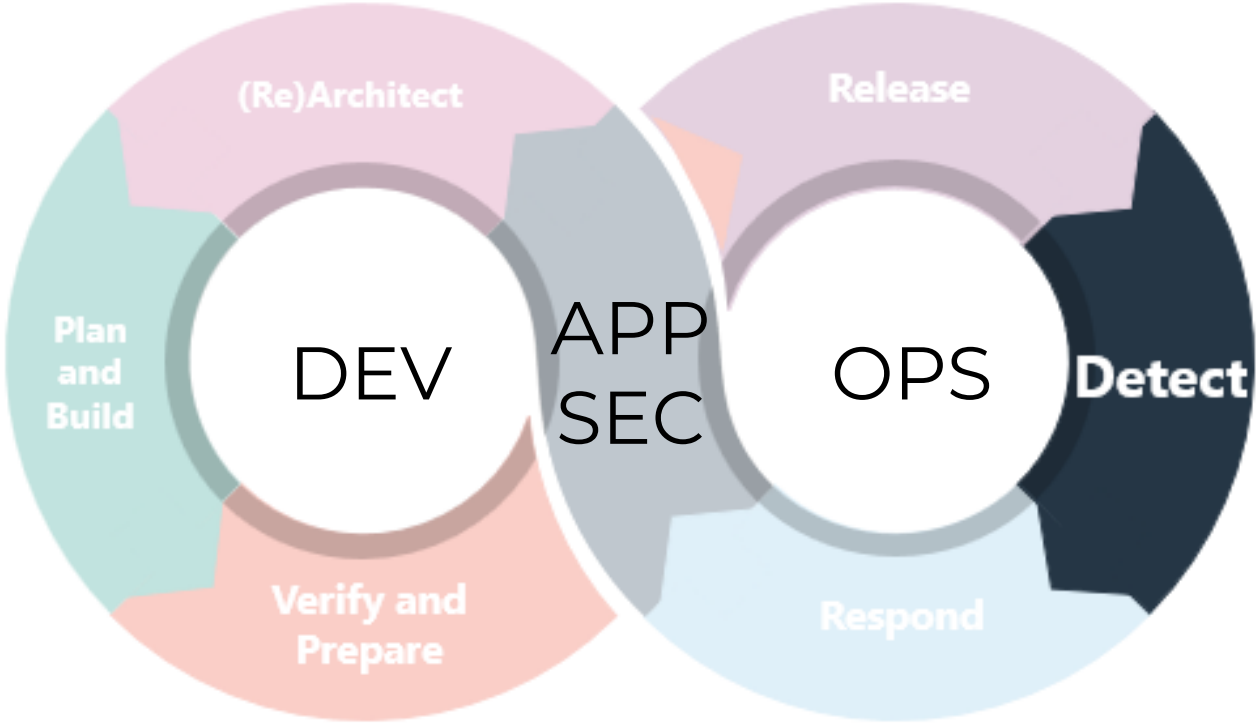
- Leverage the CD process to ensure proper tracking and security-gated releases.
- Create a version rolling process to ensure easy roll-back in case of bad artifacts are being shipped
- Very restricted access to production environments
- Use IaC to prevent human errors



Release Management plays a key role in ensuring security vulnerabilities are not created.

DevSecOps – Detect

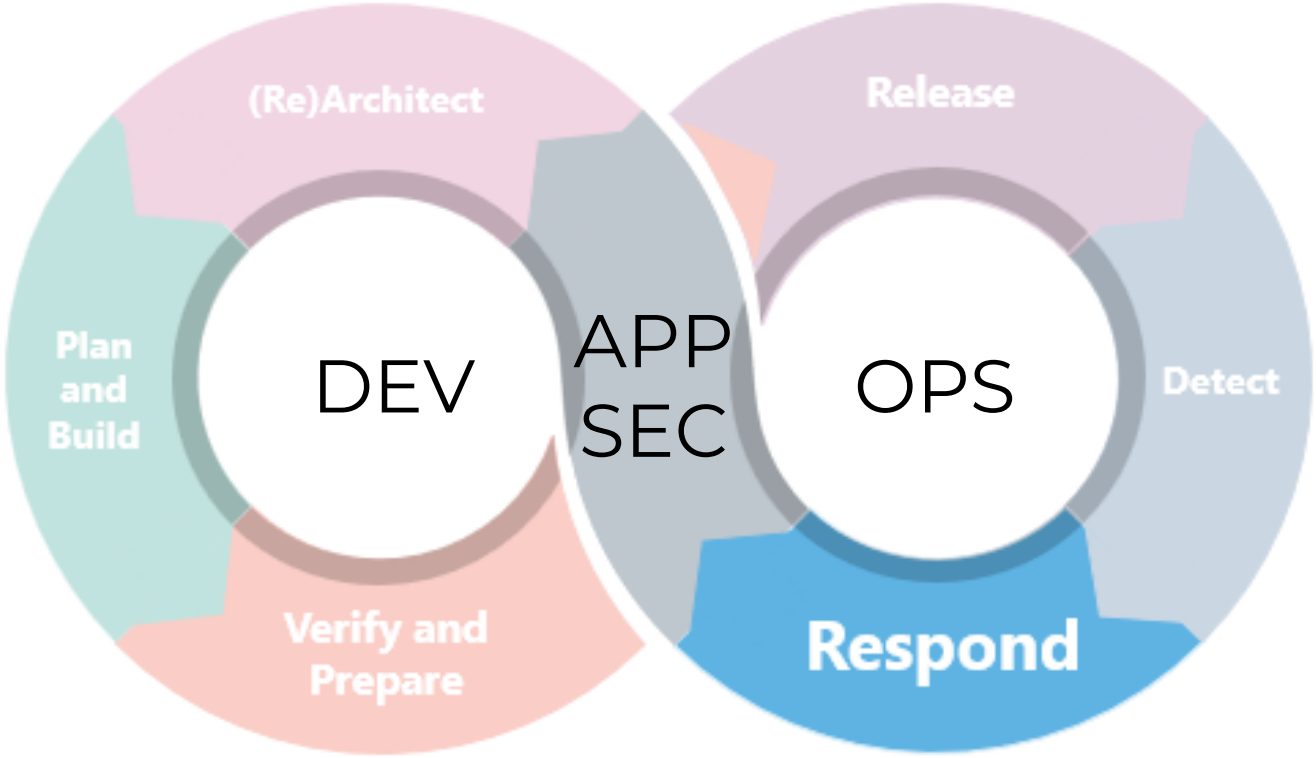
- Observability tools and telemetry
- IAST & RASP
- SEIM tool integration
- Alerts and Notifications
- Network Monitoring
- Penetration Testing



New vulnerabilities emerge daily, making applications, libraries, and technology that once were safe now vulnerable to attacks (log4j). Use proper tooling to make sure that when the application is in runtime, you are identifying vulnerabilities as they happen.

DevSecOps – Respond

1. Security Orchestration
2. WAF Shielding
3. Vulnerability Analysis
4. Security Technical Debt
5. Modify Incident Response
6. Tabletop exercises for business



Keeping application safe in a continuous process. As new vulnerabilities emerge and tools, like IAST, RASP, SEIM, and CSPM programs, notify the team of events or incidents, do a thorough analysis, and make necessary changes to the application and infrastructure.

Thank You

Contact for Ben Wilcox and Lakshman Kaveti

CTOpresentations@gmail.com