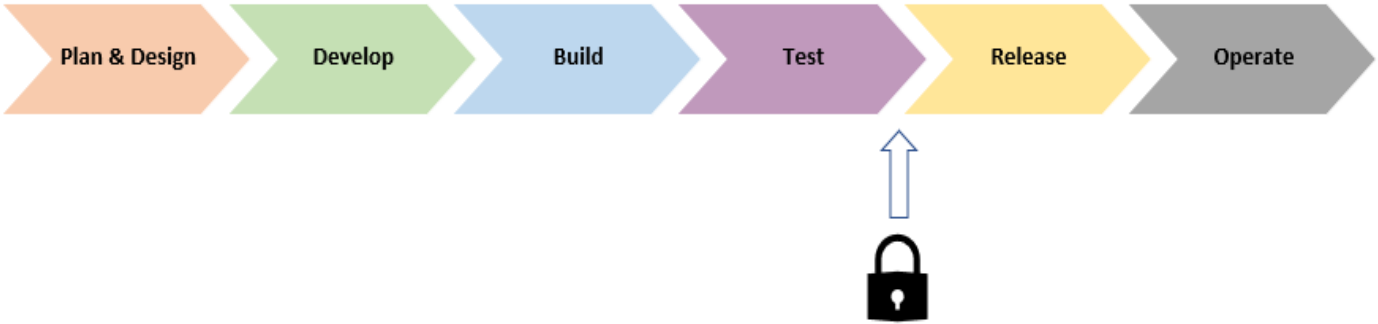
Welcome

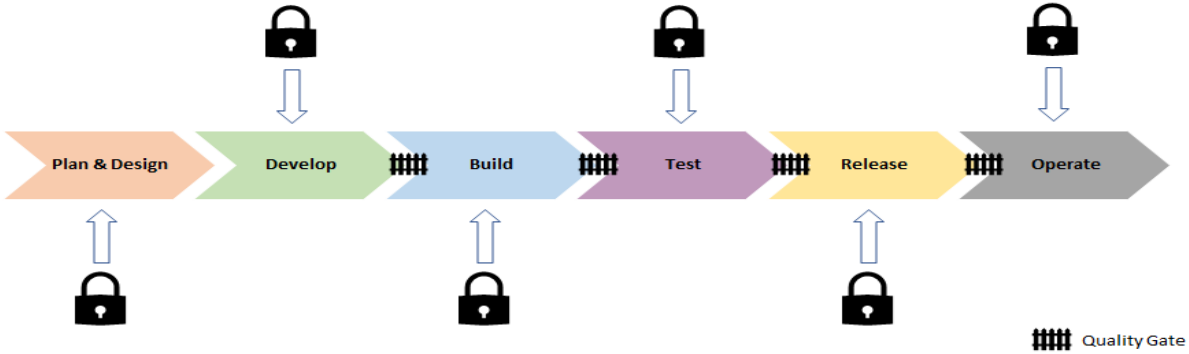
Python – TDD & Cybersecurity

DevSecOps

Remember This?



Better Approach



DevSecOps & Automation

- Plan & Design – Threat modeling, data protection, and risk assessment
- Develop – SAST (Static Application Security Testing) tools
- Build – SAST and SCA (Software Composition Analysis) tooling
- Test – DAST (Dynamic Application Security Testing) tools, passive/active scans and “fuzzing”
- Release – Additional security-specific scanning, port scans, and log validation
- Operate – Monitoring & alerting, RCA (Root Cause Analysis)

DevSecOps & Automation

- Quality gates guard against moving security defects forward
- In true DevOps fashion:
  - Information gathered from early phases feeds into later phases
  - Lessons learned feed continuous improvement of overall process

Shifting Security Left

Shifting Security Left – Plan & Design

- As discussed previously, threat modeling is a great tool for this stage

Shifting Security Left – Develop

- Ideally, security flaws in code would be caught on a developer’s machine
- Can use plugins for IDEs to provide automated static analysis
- In some cases, tools can be configured with custom rules and priorities

Shifting Security Left – Develop

- Rulesets and priorities should be driven by results of threat modeling
- Scanning tools if used should be a standard part of all developer's code tooling, to ensure consistently applied (standard dev build)
- Goal is to catch issues VERY early on in the SDLC

Shifting Security Left – Develop (Demo)

GitLab CI/CD Pipeline (Demo)

Shifting Security Left – Build

- Changes from a developer are ready to be integrated with others
- Typically driven through submission of Pull Request in GitHub
- Notifies a peer that changes are pending and need review

Shifting Security Left – Build

- In addition to logic/syntax errors, peer reviewer can focus on areas of high risk/high impact identified in threat modeling
- In some cases, a specialist peer reviewer (e.g., one from InfoSec) can be engaged to validate security of code

Shifting Security Left – Build

- Coupled with manual review, SAST (Static Application Security Testing) tools can be integrated into CI/CD to automate security checks
- Tools like SonarCloud/SonarQube can be integrated into pipeline via plugins (https://sonarcloud.io/)
- Some even include ability to capture scan results in resulting build report and use results of scan as quality gate to prevent move forward on failure

Shifting Security Left – Build

- SCA (Software Composition Analysis) tools also provide benefit
- Tools like WhiteSource (https://www.whitesourcesoftware.com/) can be integrated into CI/CD for automated security scans of Open-Source components
- Ideally, governance would be in place to monitor and manage “accepted” set and versions of Open-Source tools approved for usage

Shifting Security Left – Build

- SCA tools can provide a couple of benefits
- Primarily, security scans can be executed, and Open-Source vulnerabilities identified
- Secondarily, can help an organization catalog the Open-Source components (and versions) “in play” already

Shifting Security Left – Build

- Resulting report can be used to identify components that have not been properly vetted through governance
- Results can be merged with overall build output
- Quality gates (manual or automated) can be built around results to prevent move forward if insecure

GitLab CI/CD Pipeline + SAST (Demo)

GitLab CI/CD Pipeline + Dependency Analysis (Demo)

Shifting Security Left – Test

- Includes changes ready for move to QA for additional testing
- Dynamic Application Security Testing (DAST) tools provide more sophisticated vulnerability checks
- Dynamically exercise application’s runtime interfaces using injected data

Shifting Security Left – Test

- DAST tools can use complex combinations of invalid input via fuzzing
- Provides multiple layers of protection
- A tool like OWASP Zed Proxy or OWASP ZAP is an example (https://www.zaproxy.org/)

Shifting Security Left – Test

- Two types – passive scan and active scan
- Passive scan less aggressive and minimizes use of fuzzing
- As a result, not as robust in its ability to find exposure but also takes less time to run

Shifting Security Left – Test

- Because of that, passive scan can be good fit for CI/CD pipelines, especially those that look to push updates at a greater frequency
- For the C (continuous) part, need flows to complete quickly

Shifting Security Left – Test

- Active scans are more aggressive in their attack approach
- Make extensive use of fuzzing to elevate level of attack sophistication
- Send multiple requests, continually altering request data to simulate attack payloads

Shifting Security Left – Test

- Can also make use of a technique called “spidering”
- Allows the scan to crawl a site or service to simulate sequenced or multi-level attacks (stateful attacks)
- Can be very effective in securing a site or service but also takes much longer to execute (due to added complexity and sophistication)

Shifting Security Left – Test

- Because of its more aggressive nature, should only be executed against owned sites and in a sandboxed environment (risky)
- Often active scans are scheduled to occur out-of-band on a regular basis (weekly, nightly, etc.)

Shifting Security Left – Test

- In addition to the automated scans, traditional functional test scripts and manual testing may be used
- Should include security-focused testing as well (or in separate PEN testing)
- Tests should be informed by prioritized set of vulnerabilities identified during threat modeling

GitLab CI/CD Pipeline + DAST (Demo)

Shifting Security Left – Release

- Software confirmed ready for release
- Often deployed to a pre-prod environment (like staging)
- Can include automated “smoke” testing

Shifting Security Left – Release

- This state may also include security-specific scans as a “last mile” protection
- SAST, SCA, and DAST scans may be repeated in this environment as a double-check
- Port scans can also be used (depending on how closely this env matches prod) to help validate exposure at network communication layer

Shifting Security Left – Release

- Additionally, this stage can utilize application logging as a type of validation
- Provides a unique opportunity to confirm correct sanitization of log detail
- Also, can include verification of specific AuthN/AuthZ controls prior to a prod release

Shifting Security Left – Operate

- In this stage, monitoring and alerting become paramount for capturing and notifying support of any security exposure “in the wild”
- System logs, telemetry, and event detail become useful
- Likely not everything will be found prior – ongoing vigilance is a must

Shifting Security Left – Operate

- When (not if) something happens, Root Cause Analysis (RCA) are important
- Helps with understanding the why
- Can also help with identifying short-term workarounds (to stop the bleeding) and long-term fixes (for enhancement prioritization)

Shifting Security Left – Operate

- Preceding detail can feed dashboards or reports to help stakeholders visualize the security posture of the system and org
- System logs can also be used to satisfy auditing requirements (depending on the industry)
- Key Performance Indicators (KPIs) can be used to measure

Shifting Security Left – Operate

- Number of security-related tickets
- Build or release delays caused by security alerts
- Mean time to compliance
- These can all help support the Continuous Improvement function in the area of security focus

Shifting Security Left – Operate

- One additional layer of protection that a company can employ involves security bug “bounty hunters”
- Ethical hackers that can be paid to independently attempt to attack a site or system
- Can result in a set of reports on uncovered vulnerabilities and recommended remediations

Other Key Considerations - IaC

- As with application code, artifacts used to automate infrastructure should also be considered in scope for security considerations
- Infrastructure-as-Code (IaC) workflows can also leverage secure pipeline (see GitLab for an example)
- Helps ensure automatic and ephemeral infrastructure creation and teardown occurs in a secure manner

Other Key Considerations - SAST

- Be aware that SAST tools have the propensity for false positives (depending on tooling employed)
- Tools may err on the side of caution with security scans
- Might require that an organization customize the tool to more accurately report on environment and application
- Otherwise, teams run the risk of wasting time on non-value add activities, impeding “frictionless security”

Other Key Considerations - Containers

- If application profile includes containers or container orchestration (K8S), additional consideration required relative to security of images
- Tools like Aqua (https://www.aquasec.com/) and Prisma Cloud (https://docs.paloaltonetworks.com/prisma/prisma-cloud/) can help with scanning of container images

Other Key Considerations – Sensitive Config

- Sensitive configuration data and app secrets (e.g., connection strings, passwords, etc.) should never be hardcoded in source
- Run the risk of checking into and exposing via source control
- Tools like truffleHog (https://github.com/dxa4481/truffleHog/) can help with GitHub repo scanning

Other Key Considerations – Sensitive Config

- Other platforms include built-in support
- Services like AWS KMS or Azure Key Vault can provide dynamic linking of secure config into the CI/CD pipeline

Dev(Sec)Ops and “The Three Ways”

First Way – Systems Thinking

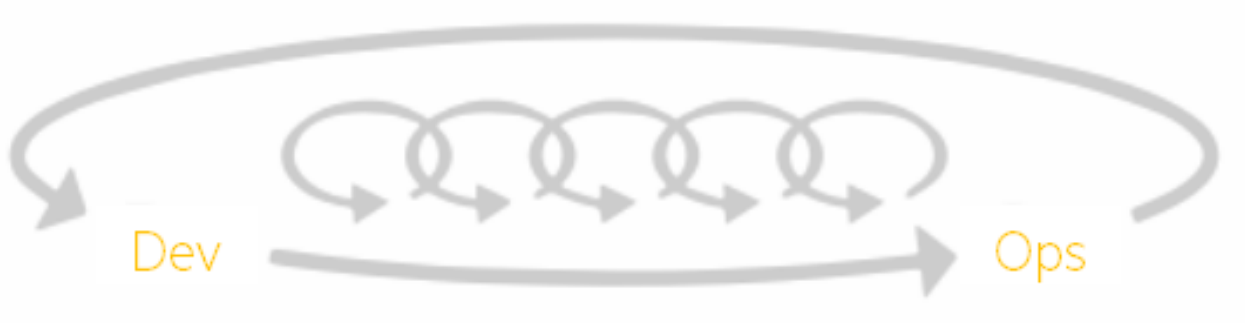
- Think holistically about the system
- Attacker-centric, asset-centric, and application-centric

Second Way – Amplify Feedback Loops

- Improve communication across the org
- About getting critical info into the right hands at the right time
- Informs continuous improvement

Third Way – Culture of Continual Experimentation and Learning

- Continual learning & growth
- Agile execution, innovation, & “fail fast”



Where Do We Go From Here?

Thank you!

If you have additional questions, please reach out to me at: (asanders@gamuttechnologysvcs.com)

DevelopIntelligence
A PLURALSIGHT COMPANY