Mohit Rampal

mrampal@yahoo.com

+91-98715-83777

Open Source Summit 2018

Open Source-DevOps and Security

Transitioning

DevOps to DevSecOps

Open Source- DevOps and Security

- – Open Source celebrated 20 years recently
- – Contributors growing, Organizations dedicating contributory teams
- – OSS consumption increasing *- 80-90% of all commercial software developers use open source components
- – 87.3 Billion node packages & 6.3 billion python packages downloaded (Jan-Sep 17)**

Open Source- DevOps and Security

- Open Source - 20 years old
- OSS consumption increasing *- 80-90% of all commercial software developers use open source components
- 87.3 Billion node packages & 6.3 billion python packages downloaded (Jan-Sep 17)**
- Public applications available on Docker Hub doubles
- Decrease in Red Hat vulnerabilities since 2012 **

Increasing

- OS Packages indexed
  - Rubygems- 10%
  - Python libraries- 32%
  - Maven - 28%
  - NPM - 57%
  - open source application library security vulnerabilities – 40% **
- Contributors & Organizations dedicating contributory teams

* Forrester and Gartner Reports

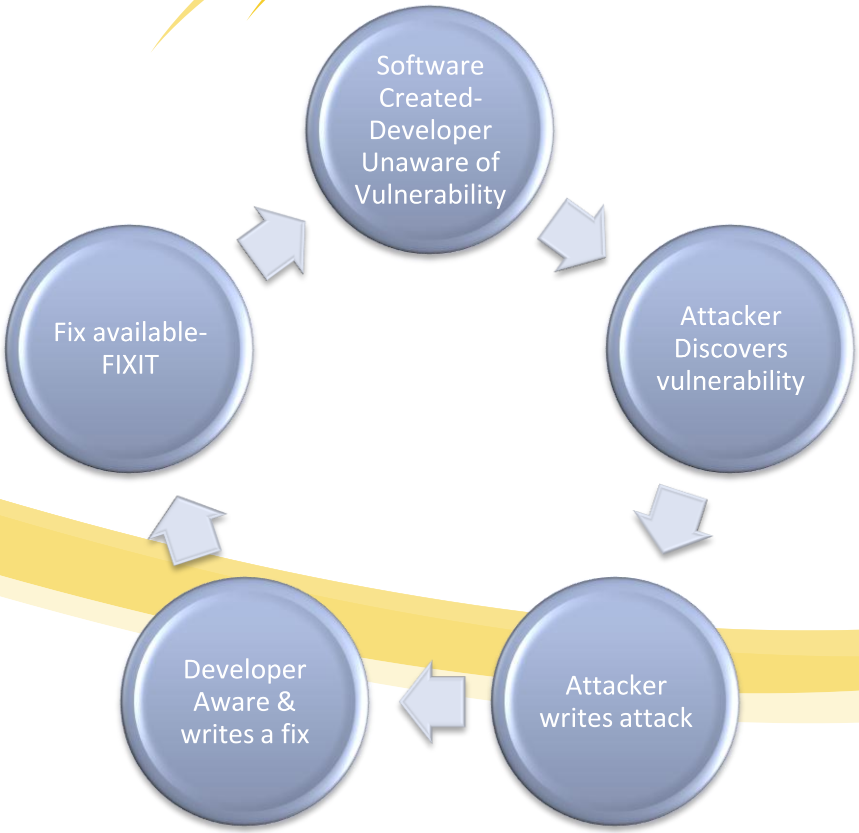
** Snyk report

Open Source- DevOps and Security

Security & Devops

- In Github's recent Open Source Survey, 86% of users said security was extremely or very important.**
- Recent exploits-Equifax etc gave security & DevOps the required push
- Organizations move to DevOps and DevSecops adoption increasing
- Managing OSS Lifecycle is a focus for organizations

Vulnerability Window


Software Created- Developer Unaware of Vulnerability
Attacker Discovers vulnerability
Attacker writes attack
Developer Aware & writes a fix
Fix available- FIXIT

- 2.89 years- median time from vulnerability introduced - publicly disclosed
- 75% of vulnerabilities - not discovered by the maintainer
- 79.5% of maintainers – have no public-facing disclosure policy
- 21% of maintainers **who do not** have & 73% **who do have** a public disclosure policy have been notified privately about a vulnerability

Report from Snyk.io

What About Open Source

OPEN SOURCE SECURITY RISKS

Hidden Entry

Rich Target

Varying Quality

Moving Target

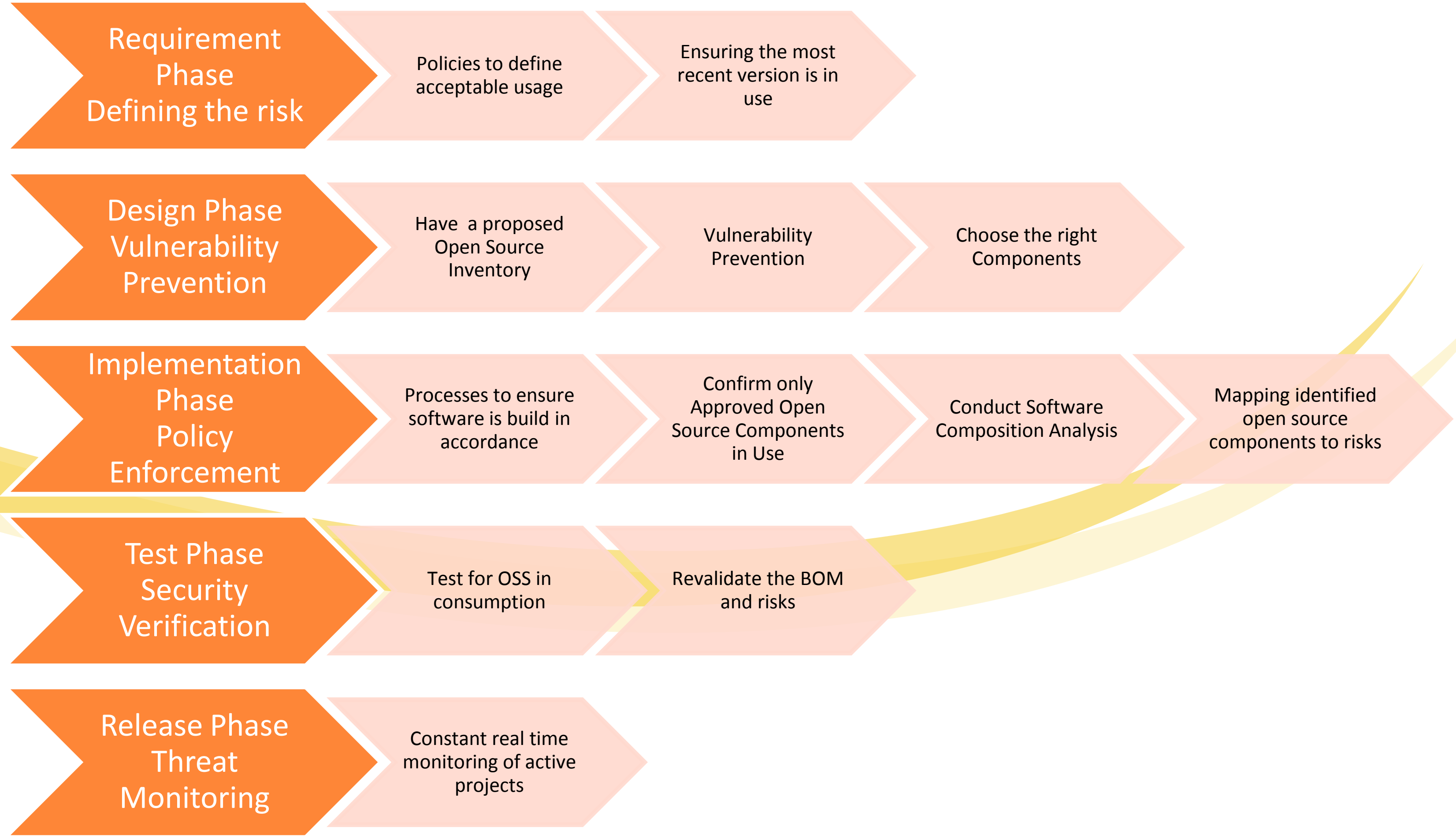
Results show organisations unaware of Open Source in their applications

Use of OSS Projects is broad, so a single vulnerability has a high rate of return

Most projects have no specific team tasked for quality, results vary from project to project and in versions

No regular monitoring or scanning result in vulnerabilities remaining undiscovered

Open Source Use in Different Phases



Requirement Phase
Defining the risk

- Policies to define acceptable usage
- Ensuring the most recent version is in use

Design Phase
Vulnerability Prevention

- Have a proposed Open Source Inventory
- Vulnerability Prevention
- Choose the right Components

Implementation Phase
Policy Enforcement

- Processes to ensure software is build in accordance
- Confirm only Approved Open Source Components in Use
- Conduct Software Composition Analysis
- Mapping identified open source components to risks

Test Phase
Security Verification

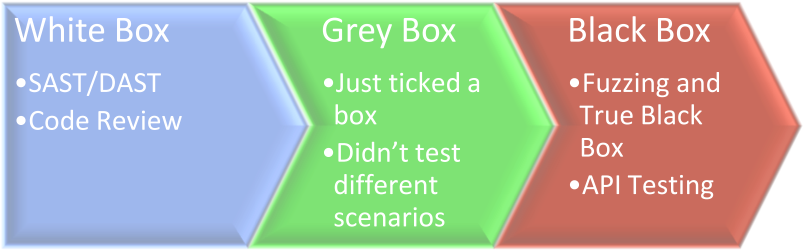
- Test for OSS in consumption
- Revalidate the BOM and risks

Release Phase
Threat Monitoring

- Constant real time monitoring of active projects

Security Testing

Traditional

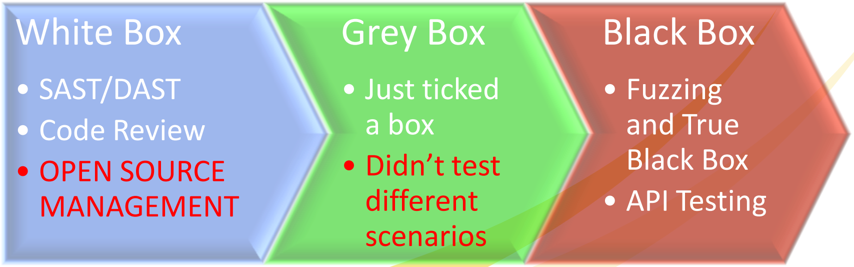


White Box
- SAST/DAST
- Code Review

Grey Box
- Just ticked a box
- Didn't test different scenarios

Black Box
- Fuzzing and True Black Box
- API Testing

New Approach



White Box
- SAST/DAST
- Code Review
- OPEN SOURCE MANAGEMENT

Grey Box
- Just ticked a box
- Didn't test different scenarios

Black Box
- Fuzzing and True Black Box
- API Testing

DEVOPS-SECURITY-DEVSECOPS

DevOps & Security-Shift to DevSecOps

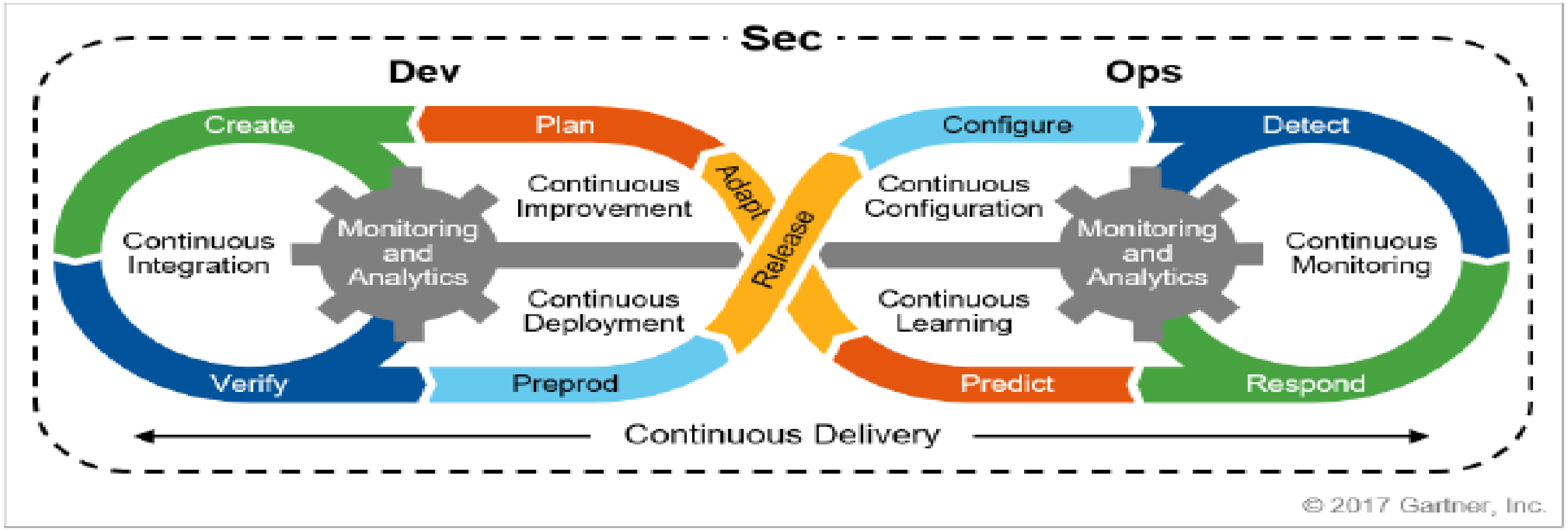
- DevOps – Increased adoption
- Security and compliance - typically afterthoughts.
- DevOps – stress on automation- security adoption slow.
- DevOps developers – traditionally no knowledge of security.
- Market shifts are fundamentally changing the way businesses approach software driven innovation

DevOps & Security-Shift to DevSecOps

- Focus on automation, with loosely coupled architectures and teams facilitating continuous delivery
- High-performing IT teams -deploying more frequently & recovering faster
- Key drivers for high performing teams- Transformational leadership & and lean product management practices
- Increase from 16 to 27% on respondents working on DevOps teams
- Leaders
  - Technology companies - 34%
  - Financial services - 14%
  - Education, retail, telecom and government agencies - 6-8% range.

•2017 State of DevOps Report by PuppetLabs

DevOps & Security-Shift to DevSecOps



Source: Gartner (November 2017)

DevSecOps: Integrating Security in DevOps

DevSecOps

- Purpose- Security is everyone’s responsibility
- Goal- safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required.

Integrating Security Into the DevSecOps Toolchain Gartner 2017

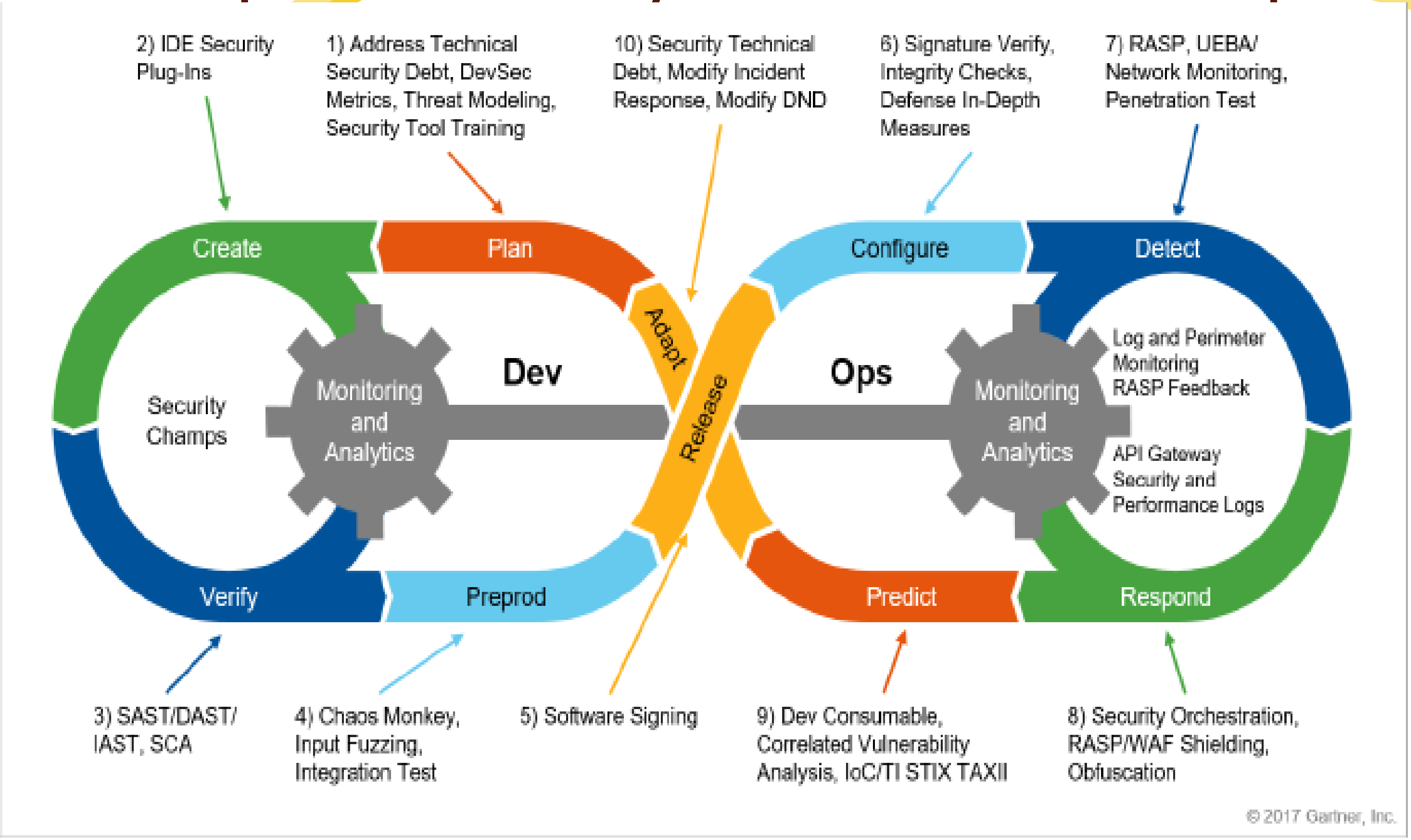
DevOps & Security-Shift to DevSecOps

- Traditional Security
  - Heavyweight
  - one-time gating inspections
  - Typically performed during testing & taking days if not weeks
  - requiring security professionals to perform them

- DevSecOps
  - Emphasizes continuous feedback
  - Improved automation
  - Security needs- adopting to mindset that security starts at the very beginning of the service creation
  - Security is continuous, automated and improves with each subsequent iteration.

Integrating Security Into the DevSecOps Toolchain- Gartner 2017

DevOps & Security-Shift to DevSecOps



2) IDE Security Plug-Ins

1) Address Technical Security Debt, DevSec Metrics, Threat Modeling, Security Tool Training

10) Security Technical Debt, Modify Incident Response, Modify DND

6) Signature Verify, Integrity Checks, Defense In-Depth Measures

7) RASP, UEBA/ Network Monitoring, Penetration Test

3) SAST/DAST/ IAST, SCA

4) Chaos Monkey, Input Fuzzing, Integration Test

5) Software Signing

9) Dev Consumable, Correlated Vulnerability Analysis, IoC/TI STIX TAXII

8) Security Orchestration, RASP/WAF Shielding, Obfuscation

Integrating Security Into the DevSecOps Toolchain Gartner 2017

DevOps & Security-Shift to DevSecOps

- Shift Left- Start testing early in the product's life cycle.
- Plan- address security and technical debt. What's priority and be prepared to handle it
- Create- Emphasize- the Shift left policy, use tools that integrate within the IDE
- Verify- Ensure you do both known and unknown vulnerabilities
- Preproduction - Favor solutions like IAST that allow you to instrument code and see how it reacts to both known attacks and to fuzzing and Chaos-Monkey-style testing
- Release – Deploy into CI/CD
- Prevent - Focus on configuration assurance at instantiation — that the code is what we expect it to be and that it indeed meets all of the requirements to be released into production.
- Detect- Use a broad range of technologies for detecting application, network known and unknown vulnerabilities. Go beyond the normal
- Respond – Understand the attack surface and monitor and feed information to plan phase to ensure quick turnaround and hardening
- Predict- Using the visibility and telemetry received from the detect and respond phases ability to predict and anticipate what new countermeasures the system will need and build them.
- Adapt – Quick adaptability and turnaround is important to ensure learning are incorporated

BEGINNING OF DEVSECOPS

....