Chapter 5

Managing Security in a DevOps Process

1

Outline

- SDLC: Iterative development and delivery in a DevOps Loop
- Agile Threat modeling
- Security Testing tools
- Continious Security testing
- Security Manual Reviews
- Monitoring Security by IT Operationals

2

SDLC: Iterative development and delivery in a DevOps Loop

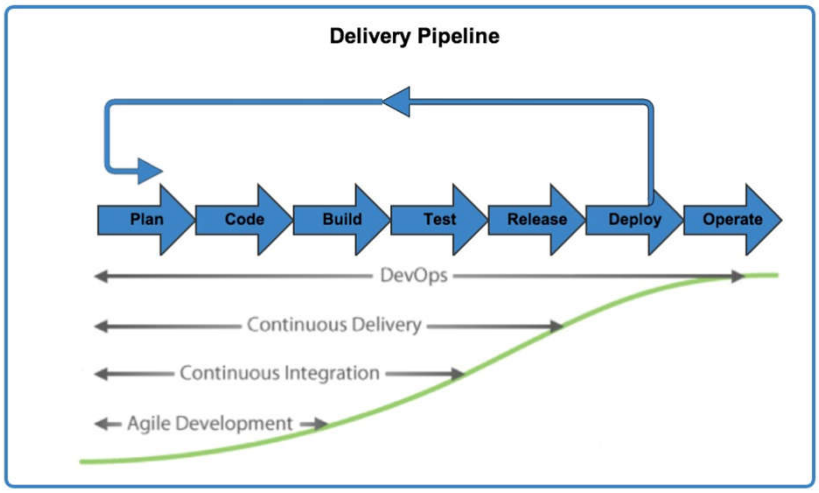
3

Principles of Information Security Management

- **Management is the process of achieving objectives given a set of resources.**
- **Principles of Information Security Mgmt**
  - Planning
  - Policy
  - Programs
  - Protection
  - People
  - Project Management

4

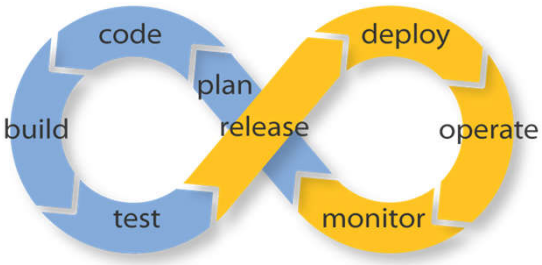
Delivery pipeline



5

DevOps

- **DevOps (a clipped compound of development and operations) is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes**

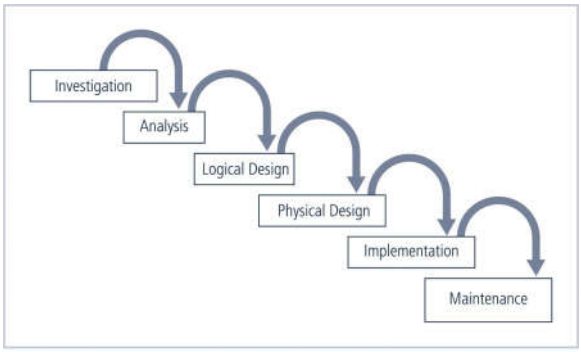


Endless Possibilities: DevOps can create an infinite loop of release and feedback for all your code and deployment targets.

6

The Systems Development Life Cycle (SDLC)

- **Methodology for the design and implementation of an information system**
- **SDLC-based projects may be initiated by events or planned**
- **Each phase concludes with a review or a feasibility analysis**



7

SDLC vs. SecSDLC: Investigation

Common steps

- Outline project scope/goals
- Estimate costs
- Evaluate existing resources
- Analyze feasibility

Steps unique to SecSDLC

- Define project process and goals and document them in the program security policy

8

SDLC vs. SecSDLC : Analysis

Common steps

- Assess current system against plan developed in phase 1
- Develop system requirements
- Study integration of new system
- Update feasibility analysis

Steps unique to SecSDLC

- **Analyze existing security policies and programs**
- **Analyze current threats and attacks**
- **Examine legal issues**
- **Risk analysis**

Agile Threat Modeling

Threadt modeling

- **Security threat modeling, or threat modeling, is a process of assessing and documenting a system's security risks.**
- **Security threat modeling enables you to understand a system's threat profile by examining it through the eyes of your potential foes.**
  - With techniques such as entry point identification, privilege boundaries and threat trees, you can identify strategies to mitigate potential threats to your system.
- **There are five aspects to security threat modeling:**
  - Identify threats
  - Understand the threat(s)
  - Categorize the threats
  - Identify mitigation strategies
  - Test

11

What is agile ?

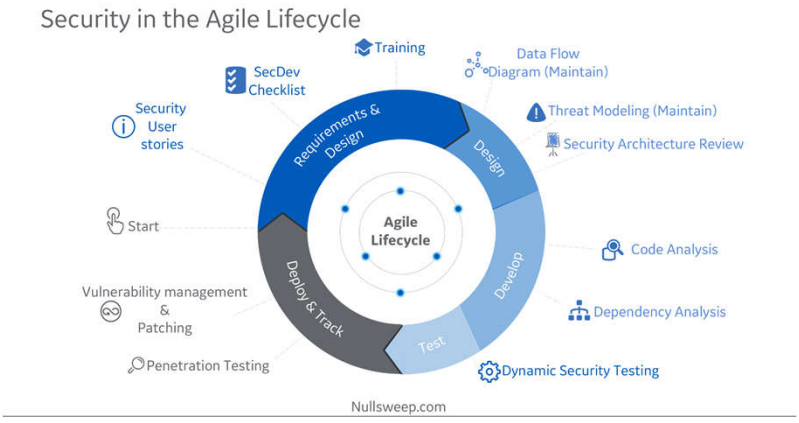
- **Deliver quickly, respond to emerging requirements**
- **Emphasis self-organization**
- **Empowers the team**
- **Intended to reduce or avoid “waste” and overburden**
- **Very popular in software development teams**
  - There are two main methods: Scrum and Kanban

12

The need for an agile approach

- **Traditionally, security has worked with project teams during two phases of execution: technical requirements design and right before go-live.**
  - After they have a working program and users have signed off, the team comes back to security for sign off. **At this point, security runs tools and penetration tests, and comes back with a stack of vulnerabilities**.
  - Unfortunately for everyone involved, this is often one of the last things before a production go-live is scheduled, and fixing security issues certainly jeopardizes the project schedule.
- **In an agile approach, security is integrated in user stories. Approved user stories are added to the backlog, which is simply a list of user stories that haven't yet been addressed.**
  - Security requirements are planned, designed and implemented altogether with the functional requirements of the project

An agile approach for Security



https://nullsweep.com/integrating-security-with-agile-development/

Security Testing Tools

15

Security Testing

- **Application security testing is critical to protecting your both your apps and your organization.**
  - Your web applications are likely to be the #1 attack vector for malicious individuals seeking to breach your security defenses. Available to users 24/7, web apps are the easiest target for hackers seeking access to confidential back-end data.
  - Web application security testing solutions are readily available, but most require a significant capital investment in hardware or software.

16

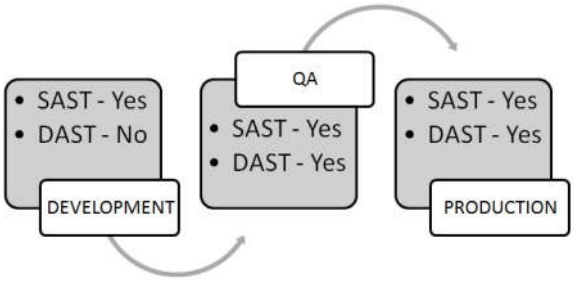
Web Security Testing Tools

- **Web Successful security testing protects web applications against severe malware and other malicious threats that might lead it to crash or give out unexpected behavior.**
- **Security testing helps in figuring out various loopholes and flaws of a web application in the initial stage. Furthermore, it also helps in testing whether an application has successfully encoded security code or not. Primary areas covered by security testing are:**
  - Authentication
  - Authorization
  - Availability
  - Confidentiality
  - Integrity
  - Non-repudiation

17

SAST vs DAST

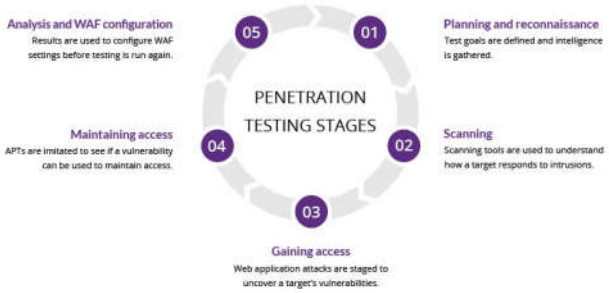
- **SAST solutions can be integrated directly into the development environment.**
  - This enables the developers to monitor their code constantly. Scrum Masters and Product Owners can also regulate security standards within their development teams and organizations. This leads to quick mitigation of vulnerabilities and enhanced code integrity.



18

Pen Testing

- **A penetration test, also known as a pen test, is a simulated cyber attack against your computer system to check for exploitable vulnerabilities.**
  - In the context of web application security, penetration testing is commonly used to augment a web application firewall (WAF)
- **Penetration testing stages**



https://www.imperva.com/learn/application-security/penetration-testing/

19

Pen Testing Tools

- **Here are some popular penetration testing tools**
  - Netsparker
  - Acunetix
  - Probely
  - Probely
  - Zap Proxy
  - Metasploit
  - Wireshark
  - W3af
  - Burpsuite
  - Nessus

20

Continious Security Testing

21

Continuous Security Testing for agile Development and DevOps

- **Continuous Security Testing ensures, that systems and applications are analyzed for vulnerabilities in a continuous cycle.**
  - With complete access to continuous security testing feedback, developers will be able to immediately identify and rectify any mistakes, which in turn, makes for a more effective and efficient development process.
- **As security practices are continually improved, organizations will be able to benefit both in the short and long run, while reducing future expenditure on remediation of security vulnerabilities and bug fixing.**

22

When is Continuous Security Testing useful?

- **Continuous Security Testing is useful for all applications that are developed in short iteration cycles.**
  - Modern development methods often do not allow the necessary time windows for manual security tests.
- **By integrating the security tests into the development process, vulnerabilities in the source code can be detected and remedied early on.**
  - In addition, the continuous inspection in production allows to continuously increase the security level as well as to prove a high degree of test coverage. The close integration drastically shortens the communication paths between tester and developer, thus increasing efficiency.

23

Security Manual Reviews

24

Security Manual Reviews

- **Manual secure code review is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.**
- **It is a tedious process that requires skill, experience, persistence, and patience.**
  - Vulnerabilities discovered, and subsequently addressed through the manual review process, can greatly improve an organization's security posture.
- **Security code review is a method of assuring secure application developers are following secure development techniques.**
  - A general rule of thumb is that a penetration test should not discover any additional application vulnerabilities relating to the developed code after the application has undergone a proper security code review.

25

Phases of Code Review

- **There are three primary phases of a manual secure code review**
  1. **Interview**: By beginning with an interview with the developers, the review team has a chance to understand the intent of the application before reviewing the code.
  2. **Code Review**: After the interview, the review team works individually to review the application as a whole. Rather than handing off individual code files to specific team members, each member reviews the entire application
  3. **Reporting Results**: After the individual code reviews are completed, the team meets to share results. Each reviewer has the chance to review the others' findings, providing an opportunity to discuss why certain findings may appear in one team member's list but not in another's.

26

when to perform a manual secure code review

- **Manual review is great at unwinding business logic and understanding the intentions of a developer.**
  - Automated tools struggle in these areas, resulting in false positives and, worse, missed issues.
- **Flaws related to authentication, authorization, cryptography, and overall data validation are often best identified by manual analysis.**
- **Manual review is also good for examining rarely traversed code paths.**
  - Evaluation techniques such as penetration testing or "fuzzing-only" examine paths for which inputs are provided.
  - Lesser exercised paths, or intentionally hidden paths, can be missed. Automated static analysis tools can follow these rare paths but often fail to understand the business logic associated with them.
  - A rigorous manual review is typically better able to identify, unravel, and examine these paths otherwise missed or misunderstood by automated tools.

27

Limitations of manual secure code review

- **Buffer overflows, dead code, and other subtle mistakes are tough for a human reviewer to find and are better suited to automated analysis. Also, manual review can easily miss vulnerabilities that stem from the integration of many small isolated problems spread out over large code bases.**
- **There is a cost associated with creating effective manual secure code review teams. Engineers need to be trained in the review process and must be available to perform reviews.**
- **A good software assurance program includes a variety of tools and methods. Manual review often finds flaws that no other technique can identify. When combined with penetration testing, automated analysis, design reviews, and developer education, a significant improvement can be seen in the security of an application.**

28

Monitoring Security by IT Operationals

29

Continuous security monitoring

- **Continuous security monitoring is a type of security solution that automates security controls across various sources of security information.**
  - Continuous security monitoring solutions provide real-time visibility into an organization's security posture, constantly monitoring for cyber threats, security misconfigurations, or other vulnerabilities.
- **Technology today has become an integral part of all business processes, but the ever-increasing threats to cybersecurity have given rise to the importance of a foolproof Continuous Monitoring Program.**

30

Building a continuous security monitoring strategy

- **At the outset of building a continuous security monitoring strategy for the purposes of cybersecurity, you first need to understand how data can be compromised. The three main ways are:**
  - **External attacks** (i.e., bad actors breaking into your network from the outside)
  - **Insider attacks** (i.e., trusted employees or company insiders either willingly or unknowingly becoming the source of data loss, theft, or compromise)
  - **Supply chain or third-party ecosystem attacks** (i.e., vendors that have access to your most critical data becoming the source of data loss, theft, or compromise)
- **Components you should consider while putting together your continuous monitoring plan**
  1. Identify the data you want to protect
  2. Create a process for patching security vulnerabilities regularly
  3. Ensure that you're continuously monitoring all of your endpoints
  4. Create a process for continuously identifying changes in standard user behavior from within your organization
  5. Put continuous security monitoring software in place to monitor your third parties

31

Finding the Right Tools for a Continuous Monitoring Program

- **It was a tough task to find the right tools for a CM program in the past, but things have improved these days. More and more vendors are now developing the tools to support the continuous monitoring strategy.**
  - This provides relief for the security teams who are looking to implement more secure methods for data collection and information sharing.
    - At a network configuration level, the management platforms serve with better centralization, policies and change management
    - In addition, there are scanning tools for the evaluation of vulnerability at the enterprise level
    - These scanning tools serve with both unauthenticated and authenticated scans. In addition, there are scanning tools to check database issues and the coding of the websites and database
    - Even some minor modifications to the already-installed antimalware tools support the continuous monitoring program

32