VIEW POINT

WHY SECURE BY DESIGN

Infosys®
Navigate your next

Table of Contents

Introduction: Why Secure by design

Significant advancement in Technology and Software Development is invariably accompanied by a heightened exposure to risk which in turn gives rise to newer risk management strategies. The Pandemic has thrown traditional defense infrastructure into disarray; huge Digital Transformations and movement of Business Processes to Cloud have challenged security systems like never before. According to Gartner - *By 2023, 75% of organizations will restructure risk and security governance to address new cyber-physical systems (CPS) and converged IT, OT, Internet of Things (IoT) and physical security needs, an increase from fewer than 15% today. This is due to Digital business transformation and emerging cyber-physical systems that create unprecedented security risk.* **Secure By Design(SbD)** is one of the most significant risk mitigation strategies. Wikipedia defines : "Secure by design, in software engineering, means that software products and capabilities have been designed to be foundationally secure." It is a developmental approach that focuses on making the software as secure as possible, as early in the Software Development Life Cycle (SDLC) as possible. Security defects and vulnerabilities can creep into an application at all stages of the SDLC. But the more we delay the detection of such defects, the higher the cost that an organization pays in fixing them; it can be as high as 100 times in Deployment Phase as compared to the Design Phase. Other than cost and legal implications, there are huge losses in form of reputational loss and Business loss. Hence it is imperative that security implementation shifts left in SDLC. Secure SDLC is a manifestation of this strategy.

Cyber Security: Enhanced Significance for Financial Services industry

Sensitive data of its customers is at the heart of Financial Services Industry. The very foundation of this industry lies in nurturing trust and credibility. There is an added complexity due to myriad government regulations around use and storage of financial and personal data with financial industry being one of the most highly regulated ones. For Banks, data breaches can mean not just loss of business but can cause havoc in unscrupulous hands. Cybercrime is so widespread in this industry that it is said "There are two types of financial services companies. Those that have experienced a cyberattack and those that will in the future."

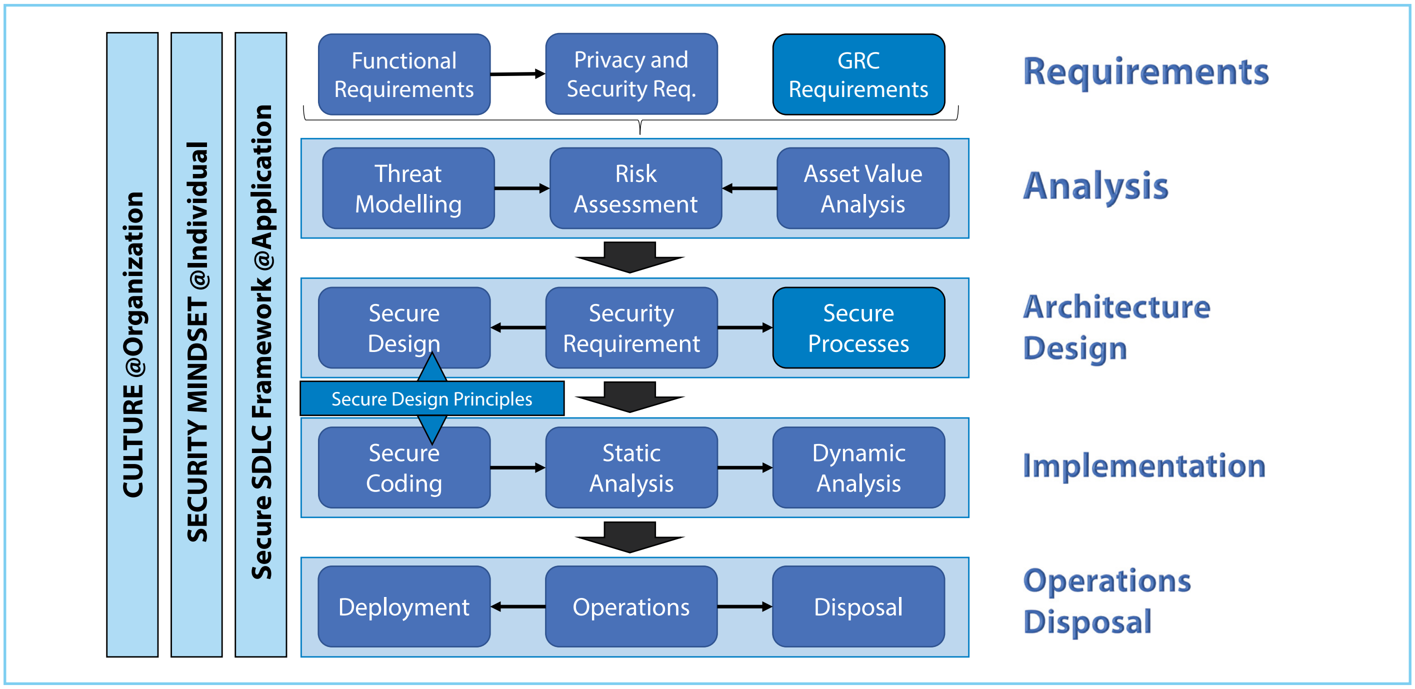
SbD principles in various SDLC Phases

Security culture is not limited to project teams. It should be imbibed in an organization at large. Security mindset must be inculcated in every member of the organization. A proper security framework should be defined for a project during initiation. Gartner suggests – *"Setting objectives and building business case to align with security strategy is essential . So also is creating a risk prioritization framework and developing an action plan for implementation of security strategy."*

Let us look at various SSDLC phases and security aspects relevant to FSADM teams at each of these phases.

Requirements and Analysis

During this phase Privacy & Security Requirements should be derived from Functional requirements. Team should consider GRC requirements (Governance Risk and Compliance) along with other NFRs. Using Asset Value Analysis and Threat Modelling, a comprehensive Risk Assessment should be presented for the Architecture and Development teams to work upon. Threat Modelling provides a view of the residual risk for the application.



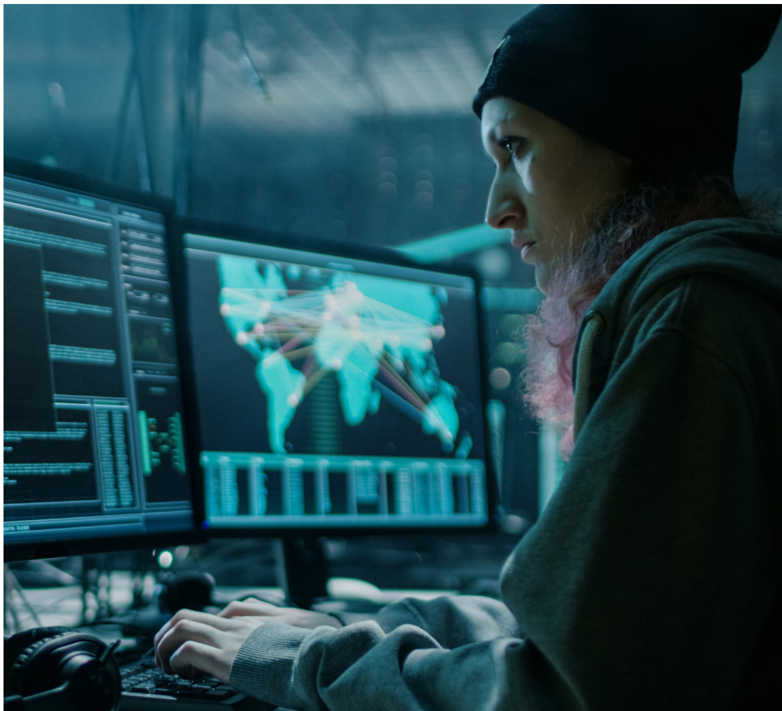
Primary security objective is known as CIA Triad (Confidentiality, Integrity, and Availability).This is supported by AAA (Authentication, Authorization, and Accounting/ Non-repudiation).

Architecture and Design

Architecting for security is the first step to its implementation in the development phase. A well-defined architecture provides consistency and coherence to security design. The Security Architecture of the OSI Reference Model (ISO 7498-2) considers five main classes of security services: authentication, access control, confidentiality, integrity, and non-repudiation. These form the basis of security during definition of System Architecture.

For building secure web applications, OWASP Security Design Principles have been created. They are :-

- Minimize attack surface area
- Establish secure defaults
- Use Least privilege principle
- Use the Defence in depth principle
- Failure should also be secured
- Services should not be trusted
- Separation of duties
- security by obscurity should be avoided
- security should not be complex to maintain
- security issues should be fixed thoroughly



Implementation Phase - Coding

Code and design flaws can alone lead to a wide range of vulnerabilities such as Brute Force Attacks, SQL Injections, Session Management, failure to Restrict URL Access and Cross-site scripting (XSS). Certain types of applications are more prone to security attacks. According to the 2021 Verizon Data Breach report , 85% of cyber-attacks materialized on web and mobile applications. These applications expose a larger attack surface and hence need close monitoring of the security measures during development. There are some general secure coding practices identified by OWASP that are language agnostic-https://owasp.org/www-pdf-archive/OWASP_SCP_Quick_Reference_Guide_v2.pdf

Language specific guidelines are also prepared by Infosys and available at this location:

https://infosystechnologies.sharepoint.com/sites/ISG/SitePages/Secure-SDLC.aspx

Writing secure code is just a step towards application security.

Implementation Phase - Secure Code Review

Once the code is developed using secure coding practices, it is important to perform a round of code review with security in mind. There are automated tools available that can help with Static as well as Dynamic secure code review. In addition to this, a manual code review must be performed . This is more strategic and can help identify logic flaws. Combining manual review with feedback from automated tools improves the overall security of the code being committed and helps reduce the number of defects that slip into production.

Implementation Phase - Security Testing

Functional and performance testing will ensure that the code meets functional and non-functional requirements. Static and Dynamic security testing (SAST & DAST) must be performed so that the application also meets security requirements . Open-Source Security Testing techniques define different types of security testing which should be used as appropriate for the application under consideration e.g., Security Scanning, Vulnerability Scanning, Penetration Testing, Ethical Hacking, etc.

Data Security

Data breaches are perhaps the largest risk to security in the Financial Industry. There are stringent Data Compliance laws in various parts of the world like Global Data Protection Rights(GDPR) in the European Union that the Banks must comply with. A comprehensive Data Security strategy is defined by such institutes which must be followed at every stage of SSDLC. Data must be secured at all touchpoints - Data at rest, Data in transit and Data in usage. First line of defense is the controlled access to the data by implementing fine grained element level security. Encryption acts as the second line of defense. Advanced Encryption techniques should be implemented for both – Data at rest and in transit. Advanced Encryption Standard (AES) and Rivest–Shamir–Adleman (RSA) are some of the typical standards used. Data Loss Prevention (DLP) tools employ advanced technologies such as Blockchain & ML to protect data on-premises on endpoints (when in use), during transit(network) or at rest (on storage).

DevSecOps

Gartner had predicted - By 2021, DevSecOps practices will be embedded in 60% of rapid development teams, as opposed to 20% in 2019. It suggests - Perfect security and zero risk are impossible. We must bring a continuous, adaptive risk- and trust-based assessment strategic and prioritization of application vulnerabilities to DevSecOps .

DevOps is a SDLC process model. At every phase of DevOps software development- from integration, testing, releasing, to deployment and infrastructure management, there is a relentless pursuit of velocity, automation, and monitoring. In the short time frames of scrum, security is many-a-times neglected. DevSecOps ensures that sufficient focus is provided to security considerations in all processes. It entails embedding governance and cybersecurity functions throughout the DevOps workflow. Identity and access management (IAM), privilege management, unified vulnerability management, and configuration management are all considered in implementation of DevSecOps.Use of automation tools help to match the speed of DevOps process. This reduces the tendency to resist the embedding of security practices in the workflow. DevSecOps is a process model which implements Secure SDLC practices.

RACI for SSDLC

During the start of a project, it is a recommended practice to create a Responsibility, Accountability, Consulted, Informed (RACI) matrix for the development team clearly showing the responsibilities for SbD. A sample RACI created by Infosys ETA is shown below. This can be used as a guide by Scrum Teams to define their own.

RACI Matrix

Responsible (Doer)
Accountable (Decision maker)

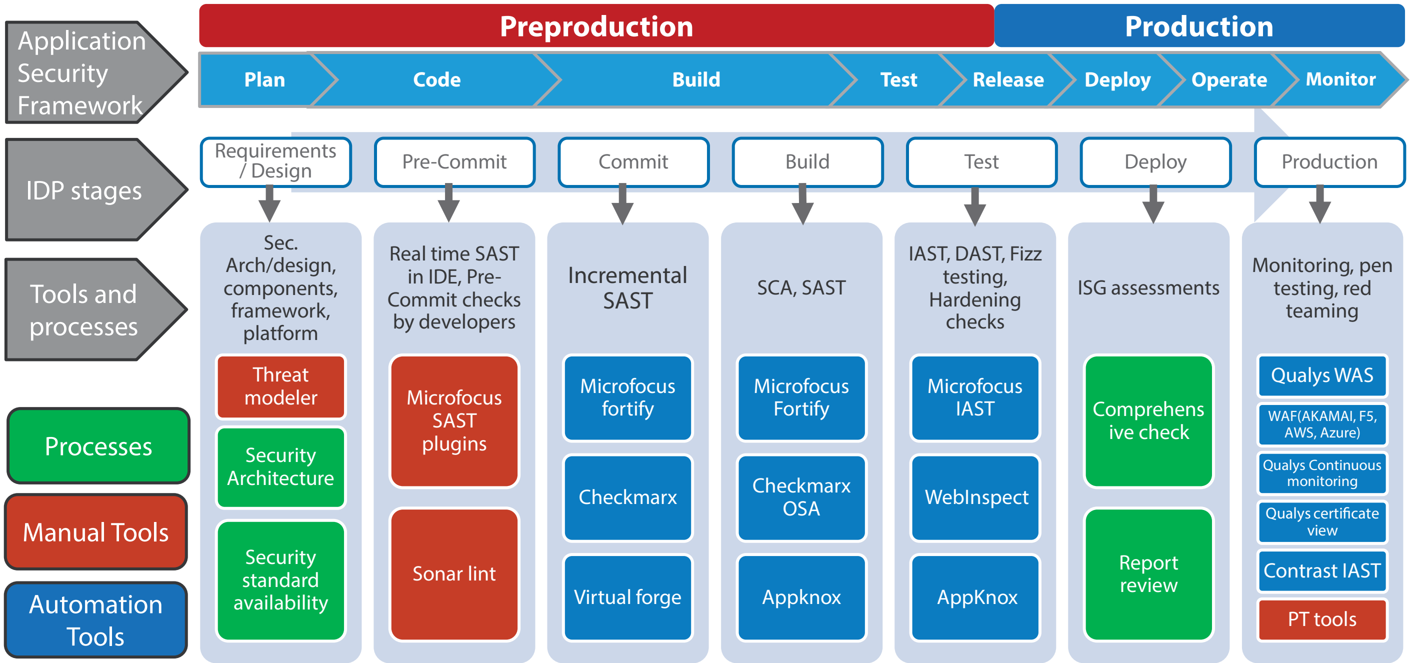
Consulted (for decision making)
Informed (reg decision made)

	Architect	PO / BA	Dev Team / Programmer	Dev Team / Tester	Deployment & Operations
Initiation	R	A	C	C	C / I
Development	A / R	C	R / C	R / C	C / I
Implementation	A	C	A / R	A / R	C / I
Operations	C	C	C	C	A / R
Disposal	A / R	C	R	R	R

PO = Product Owner BA = Business Analyst

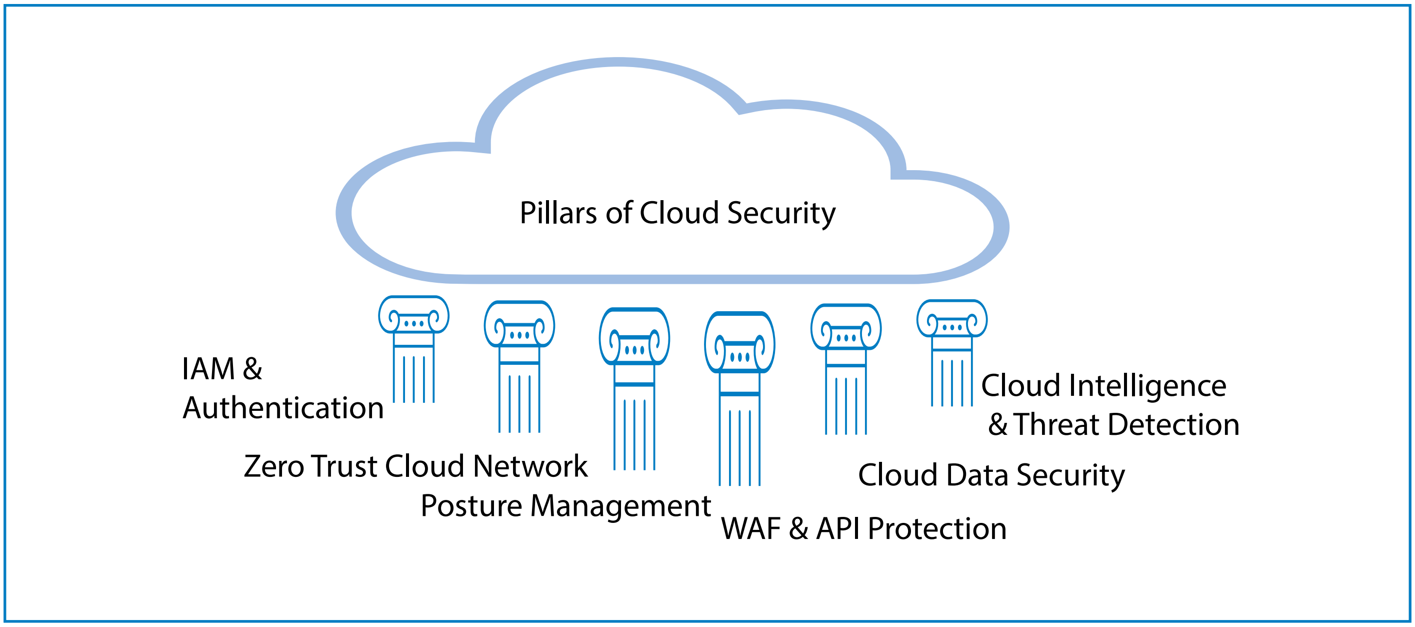
Popular Tools & Solutions

There are many layers to cyber security. We have considered the Application Development layer in this document. There are a good number of tools available for managing application security at every stage of SSDLC. An overview of some of the popular tools and solutions is shown below:



Cloud security principles

Cloud radically shifts how companies view technology design and versatility. In the FSADM world, more and more applications are now operating on Cloud. No discussion on Application Security is complete without a mention of Cloud Security. It is important to understand that Cloud Security is a shared responsibility between the Cloud Service Provider (CSP) and the Customer. Major principles of Cloud Security are shown below:



Depending on which Service Model is used (IaaS, PaaS, SaaS) , the type of secure responsibility shifts between the CSP and the Customer. In most of the cases responsibility of ensuring security of infrastructure, patching, and configuration of hosts & physical network is with the Service Provider.

Customer responsibilities generally include IAM (Identity and Access Management) for users, Authorization , Authentication and ensuring compliance of cloud-based data assets. Zero Trust Principle should be applied to all aspects of Cloud Security i.e., to automatically NOT trust anything or anyone within or outside the network. Remember to adopt a least privilege and granular approach of providing access to cloud resources for any set of users. Granular network security can be provided using micro-segmentation or zero-trust networks. Domain driven or Cloud native design is also one way of reducing the risk of vulnerability.

Conclusion

Security can no longer be an afterthought for FSADM teams. Shifting left, security has now become an integral part of the SDLC. Gartner sees the usage of more and more tools being adopted for security. *"Growing risks and ubiquitous use of open-source software in development make software composition analysis(SCA) essential to application security. Security and risk management leaders must expand the scope of tools to include detection of malicious code, operational and supply chain risks."*

Most Application Security Testing (AST) suite vendors as well as application development tooling vendors have already begun to include SCA capabilities as features in their offerings. Development Teams will get more and more sophisticated tools that utilize automation and ML for secure development. Their adoption should be accelerated due to the ubiquitous nature of Cloud Deployments.

Key Takeaways

- Consider security in all phases of SDLC
- Adopt appropriate tools to fix vulnerabilities early

About the Author

Rasana Karanjikar
Senior Technology Architect

About the Mentors

Viral Thakkar
AVP - Senior Principal Technology Architect

Anuj Jajoo
Senior Principal Consultant - Learning

References

1. Wikipedia : https://en.wikipedia.org/wiki/Secure_by_design
2. Secure Coding : https://wiki.sei.cmu.edu/confluence/display/seccode/Top+10+Secure+Coding+Practices
3. Gartner analysis reports about cyber security : Improve Your Security with Security Architecture and other research papers e.g. https://www.gartner.com/en/information-technology/insights/cybersecurity
4. Lex Learning Path : Security by Design and Cyber Security - Safeguarding your Digital Journey
5. Cloud Security : https://www.checkpoint.com/cyber-hub/cloud-security/what-is-cloud-security/
6. Security Testing : https://www.netsolutions.com/insights/software-security-testing/
7. DevOps Security : https://www.beyondtrust.com/blog/entry/devops-security-best-practices
8. Verizon : https://www.verizon.com/business/resources/reports/2021/2021-data-breach-investigations-report.pdf
9. https://www.copado.com/devops-hub/blog/data-security-for-banks-standards-for-success
10. https://www.datacenterknowledge.com/industry-perspectives/three-must-implement-data-security-steps-reduce-vulnerabilities

For more information, contact askus@infosys.com

Infosys®
Navigate your next

Infosys.com | NYSE: INFY

Stay Connected