Typical Secure Development Lifecycle (SDLC) Flow

Secure Software Requirements

It is important to establish the fact that “Without software requirements, software will fail and without secure software requirements, organizations will. Software development projects that lack security requirements suffer from the threats to confidentiality, integrity and availability, which include unauthorized disclosure, alteration and destruction. It is really not a question of ‘if’ but ‘when’, because it is only a matter of time before software built without security considerations will get hacked, provided the software is of some value to the attacker. The need to consider security and privacy “up front” is a fundamental aspect of secure system development. The optimal point to define trustworthiness requirements for a software project is during the initial planning stages. This early definition of requirements allows development teams to identify key milestones, deliverables and permits the integration of security and privacy in a way that minimizes any disruption to plans and schedules.

Security Architecture

Application architecture review can be defined as reviewing the current security controls in the application architecture. This helps a user to identify potential security flaws at an early stage and mitigate them before starting the development stage. Poor design of architecture may expose the application to many security loopholes. It is preferable to perform the architecture review at the design stage, as the cost and effort required for implementing security after development is high. The review should take into account the security policies and the target environment where the software will be deployed. Also, the review should be holistic in coverage, factoring in the network and host level protections that are to be in place so that safeguards don’t contradict each other and minimize the protection they provide. Special attention should be given to the security design principles and core security concepts of the application to assure confidentiality, integrity, and availability of the data and of the software itself. Additionally, layer-by-layer and tier-by-tier analysis of the architecture should be performed so that defence in depth controls is in place

Secure Software Design

One of the most important phases in the SDLC is the design phase. During this phase, software specifications are translated into architectural blueprints that can be coded during the development phase that follows. Software that is designed correctly improves software quality. Software can meet all quality requirements and still be insecure, warranting the need for explicitly designing the software with security in mind.. In addition to functional and quality aspects of software, there are other requirements that need to be factored into its design. Some of these other requirements include privacy, globalization, localization and security-requirements. Secure Design examines the implementation to help ensure that the product maintains the trust models laid out in the architecture and design phases. Investing the time upfront in the SDLC to design security into the software supports the security “built-in” opposed to “bolt-on” at a later stage. The bolt-on method of implementing security can become very costly, time consuming, and generate software of low quality characterized by being unreliable, inconsistent, unmaintainable, prone to errors, and susceptible to exploitation by hackers.

Secure Development (Secure Coding & Testing)

Reports in full disclosure and security mailing lists are evidence that software written today are rife with vulnerabilities that can be exploited. A majority of these weaknesses can be attributed to insecure software design and/or implementation and it is vitally important that software that is written is first and foremost reliable, and secondly less prone to attack and resilient when it is. Writing secure code is an important and critical component to ensuring the resiliency of software security controls.

Just because Software is designed with a security mind-set and developers implement security by writing secure code, it does not necessarily mean that the software is secure. It is imperative to validate and verify the functionality and security of software and this can be accomplished by quality assurance testing which should include testing for security functionality and security testing. Security testing is an integral process in the secure software development life cycle. The results of security testing have a direct bearing on the quality of the software. Software that has undergone and passed validation of its security through testing is said to be of relative higher quality than software that hasn't.

Secure Deployment, Operations, Maintenance & Disposal

One of the final stages in delivering secure software is ensuring the security and integrity of developed applications are not compromised during deployment; the Secure Deployment practice focuses on this. Just because software was designed and developed with security in mind, it does not necessarily mean that it will also be deployed with security controls in place. All of the software assurance efforts in designing and building the software can be rendered futile if the deployment process does not take into account security. In fact, it has been observed that software face hiccups when it is installed and decisions such as allowing the software to run with elevated privileges or turning off the monitoring and auditing functionality adversely impact the overall security of the software.

Once software is deployed, it needs to be monitored to guarantee that the software will continue to function in a reliable, resilient and recoverable manner. Ongoing operations and maintenance include addressing incidents impacting the software and patching the software to mitigate its chances of being exploited by hackers and malware threats

Finally there is a need to identify the software and conditions under which software needs to be disposed or replaced because insecure and improper disposal procedures can have serious security ramifications.