2023

MORPHEUS

Secure Software Development Lifecycle

MORPHEUS DATA LLC.

Overview

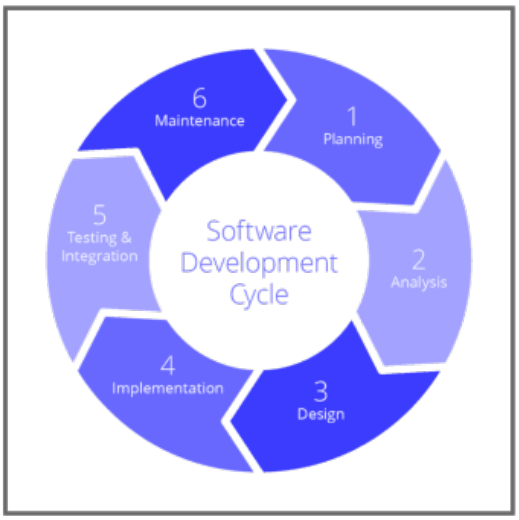
The overarching goal of this document within Morpheus Data LLC is to fortify product security and uphold exceptional quality standards throughout the software development lifecycle. It serves as a comprehensive guide, providing strategic insights and tactical methodologies to all stakeholders involved in software creation. By emphasizing OWASP best practices, it aims to elevate security measures, mitigate risks, and augment the overall quality of the products developed at Morpheus Data LLC.

Purpose

This document assumes a pivotal role as a compass directing all stakeholders towards integrating robust security practices and ensuring top-tier quality in every phase of software development. It aims to instill a culture of proactive security awareness and meticulous quality standards. By aligning best practices, the document acts as a definitive guide, fostering collaboration and coherence among teams, ultimately enhancing the security posture and product quality of our software offerings.

Scope

The scope of the guidelines within this document spans across the entirety of software development projects. This encompasses an exhaustive range of phases, including planning, design, implementation, testing, deployment, maintenance, and incident response. By encompassing all these phases, the document ensures that every aspect of software creation adheres to stringent security measures and uncompromising quality standards to fortify the resilience of the final products.



A typical SDLC representation

Planning

Overview

The Planning phase is the foundational stage where the Product Planning Team collaborates to define the product's feature set and release cycle scope. This phase aligns strategic objectives with the technical roadmap, incorporating robust security considerations.

Process Details

Feature Assessment and Risk Refinement: The Product Planning Team meticulously assesses and refines the features based on size, complexity, and associated risks. This includes a comprehensive risk assessment to identify high-risk elements that demand specialized attention.

Security Requirement Evaluation

Each feature undergoes a thorough evaluation to ensure alignment with security prerequisites. Questions addressed at this stage include:

- Do proposed features align with security requirements?
- What specific security considerations are imperative for the development of these features?
- Features identified with heightened risk are explicitly labeled for dedicated security scrutiny and verification by the Security team in collaboration with QA.

Security requirements can differ based on the type of project. Morpheus' Security team determines these criteria based on relevant industry and government standards such as:

- OWASP Application Security Verification Standard
- NIST-800-53 Requirements
- Application Security and Development Security Technical Implementation Guide (STIG)

Key Objectives

- **Strategic Alignment with Security:** Align product planning with security objectives, ensuring a proactive stance towards risk mitigation and adherence to security standards.
- **Early Risk Identification and Mitigation:** Detect potential security vulnerabilities at the planning stage to preemptively address them, avoiding costly implications later in the development lifecycle.
- **Collaborative Security Ownership:** Foster a culture where security responsibility is shared across teams, notably empowering the Security and QA teams to collaborate effectively on high-risk features.

Stakeholder Involvement

The CTO plays a pivotal role in reviewing and approving release plans, ensuring alignment with security objectives. Additionally, the involvement of the Security team, along with QA, in assessing high-risk features signifies a proactive approach towards security assurance.

Documentation and Review

All planning decisions, risk assessments, and security considerations are meticulously documented and reviewed, serving as crucial references throughout the subsequent SDLC phases. The clarity and comprehensiveness of this phase's documentation are imperative for informed decision-making.

Testing

Overview

The Testing phase in the SDLC is an integral part of ensuring robust security measures within our software. It involves comprehensive evaluation, validation, and verification of the developed features and functionalities to identify and address potential security vulnerabilities.

Testing Procedures

Continuous Quality Assurance (QA) Testing

Embracing an agile format, QA conducts iterative tests multiple times a day during the development process. This encompasses not only functional validation but also automated security vulnerability assessments.

Agile Security Testing via Geb

QA Automation testing, facilitated by Geb, performs exhaustive integration testing of the web application. This includes rigorous security permission testing across the application's functionalities.

Vulnerability Management

Identification and Labeling

Any identified security concerns are flagged with a 'security' label within the project management tool for immediate attention. These issues are mandated to be resolved before authorizing any release.

Collaborative Remediation

Collaboration between developers, QA, and the Security team is paramount in addressing identified vulnerabilities. This collaborative effort ensures a comprehensive approach to remediation, emphasizing both immediate fixes and preventive measures.

Security Compliance Checks

Prior to deployment, rigorous security checks are performed to verify adherence to predefined security protocols. Post-deployment, another round of security checks ensures the continued integrity and resilience of the software.

Reporting and Resolution

Comprehensive Reporting

QA generates detailed reports on security-related testing, encompassing identified vulnerabilities, their severity, and suggested remediation steps. These reports are distributed to relevant stakeholders for transparent communication and decision-making.

Mandatory Resolution

No release is authorized until all identified security vulnerabilities are addressed satisfactorily. This stringent criterion ensures the deployment of a more secure and resilient software product.

Continuous Improvement

Weekly engineering discussions involve a thorough analysis of any security-related incidents. The aim is to disseminate knowledge, understand causative factors, and proactively integrate preventive measures into the application framework to reduce the risk of recurrence.

Roles and Responsibilities

Product Planning Team

The Product Planning Team assumes a critical role in orchestrating the features and scope of product releases. They meticulously assess the size, complexity, and associated risks of each feature during the planning phase. Moreover, this team actively collaborates with the Chief Technology Officer (CTO) for review and approval of release plans. Their responsibilities encompass assessing the security requirements of proposed features, ensuring alignment with the overarching security objectives. High-risk features are scrutinized further, and if necessary, adjustments are made in collaboration with the Security team. This proactive engagement by the Product Planning Team ensures that security considerations are embedded into the initial stages of product ideation, facilitating a robust and secure software development process.

Development Team

The Development team plays a pivotal role in implementing and upholding stringent security practices throughout the software development lifecycle. They are responsible for embedding secure coding practices within their codebase, meticulously following security guidelines and best practices. This team conducts regular and thorough code reviews, focusing not only on functionality but also on identifying and remedying potential security vulnerabilities. Collaborating seamlessly with both the Security and Testing teams, developers ensure a holistic approach towards creating secure and high-quality software products. Their commitment extends beyond writing robust code; it encompasses a proactive stance in advocating and fostering a culture of security consciousness among team members.

Security Team

The Security Team operates at the forefront of ensuring robust security measures across all phases of software development. They actively engage in threat modeling, risk assessments, and assist in devising security strategies aligned with our security principles. Beyond these foundational responsibilities, the Security Team leads by example in conducting regular security scans and comprehensive vulnerability reviews. They leverage sophisticated tools and methodologies to proactively identify and address potential weaknesses or threats within the software ecosystem. Their role extends beyond mere identification, encompassing guidance and collaboration with other teams, fostering a proactive and vigilant security culture.

Testing Team

The Testing Team operates as a cornerstone in fortifying the security measures of our software products. They conduct rigorous security tests in tandem with functional testing throughout the development lifecycle, ensuring the identification and reporting of vulnerabilities. Moreover, the Testing Team plays a crucial role in validating vulnerability patches post-remediation, ensuring the effectiveness

of the implemented fixes in addressing identified security concerns. This comprehensive approach underscores their commitment to delivering robust and secure software solutions to our customers.

Tools and Technologies

Static Code Analysis Tools

Mend (Static Analysis)

Nightly static code analysis is conducted using Mend, ensuring early detection of potential vulnerabilities within the codebase. The tool provides comprehensive insights into code quality and security issues, empowering developers to address them proactively.

Guidelines and Integration

Well-defined guidelines govern the usage and integration of Mend within our SDLC. This ensures consistent and effective utilization across development teams, streamlining the process of identifying and mitigating security risks.

Penetration Testing Tools

Invicti (Application Penetration Testing)

Rigorous penetration testing, performed before every release cycle using Invicti, aims to simulate real-world attacks and identify exploitable vulnerabilities within the application. This proactive approach ensures the robustness of our security posture.

Horizon 3 (Network Internal Penetration Testing)

Monthly internal network penetration testing via Horizon 3 scrutinizes internal network structures for potential vulnerabilities, guaranteeing a resilient network infrastructure.

Dynamic Analysis Library Scanning

Nightly dynamic analysis library scanning by Mend checks for vulnerabilities within external libraries and components used in the software. This ensures that any vulnerabilities introduced by third-party libraries are promptly identified and addressed.

Training and Awareness

Developers within our organization actively utilize online training platforms dedicated to continuous development security training. These platforms offer a wide array of modules focusing on secure coding practices, threat identification, and mitigation strategies. Additionally, stakeholders across various departments partake in security awareness sessions aimed at fostering a culture of shared responsibility and heightened vigilance against potential security threats. This combination of specialized online resources and interactive sessions empowers our teams to stay updated with evolving security trends, equipping them with the necessary skills to fortify our software products against potential vulnerabilities.

Incident Response Plan

Overview

The Incident Response Plan serves as a structured approach to promptly detect, assess, and mitigate security incidents that may arise during the software development lifecycle. It ensures a coordinated, swift, and effective response to maintain the integrity and security of our software products.

Incident Identification

Diverse Source Identification

Security-related incidents are reported from multiple sources including support channels, ongoing security scanning, and internal audits. This multi-channel approach enables comprehensive incident detection.

Risk Assessment and Escalation

Upon incident detection, the Security team assesses the risk and severity of the incident against predefined Service Level Agreements (SLAs). Escalation into the project management software with priority security labels and SLA notifications is immediate.

Incident Resolution Workflow

Timely Resolution Protocol

A stringent framework mandates the resolution of security incidents within stipulated time frames. If an incident persists unresolved beyond defined time frames, automated escalations are triggered to relevant Security team members for swift resolution.

Weekly Engineering Discussions

All security-related incidents become focal points of discussion during weekly engineering calls. These discussions aim to educate team members about the incident, its root cause, and preventive measures to mitigate future occurrences.

Proactive Measures and Remediation

Native Prevention Integration

Whenever feasible, measures to prevent recurrence of incidents are integrated into the application framework during the remediation process. This proactive approach minimizes the risk landscape and enhances the inherent security of our software.

Continuous Improvement Loop

Incidents prompt a reflective analysis within the engineering team, leading to actionable insights. These insights drive continual improvements in security practices, aiming to proactively address vulnerabilities and potential threats.

Communication and Documentation

Transparent Reporting

All incidents, their assessments, and resolutions are meticulously documented using standardized templates. This comprehensive documentation aids in understanding incident trends and shaping future preventive strategies.

Stakeholder Communication

Relevant stakeholders receive incident reports, ensuring transparency and informed decision-making. Regular reporting on incidents contributes to a robust understanding of the security posture and encourages collective ownership.

Post-Incident Analysis

Root Cause Analysis

Post-incident, detailed root cause analysis is conducted, aiming to understand the fundamental reasons behind the incident. This analysis drives corrective actions to prevent similar incidents in the future.

Lessons Learned Integration

Insights garnered from incident analysis are integrated into ongoing security awareness programs and development processes. This knowledge sharing enhances the collective understanding and proactive mitigation of potential risks.

Conclusion

This document represents our unified commitment across Morpheus Data LLC's software development stages. It epitomizes our dedication to continuous improvement, proactive security measures, and a culture of shared responsibility. Through collaboration, transparent practices, and a relentless pursuit of quality, we ensure the creation of resilient software solutions. Our steadfast commitment to innovation while upholding stringent security practices underscores our pledge to maintain the trust our customers place in our products.