Lexmark™

Lexmark Secure Software Development Lifecycle

Index—Lexmark Secure Software Development Lifecycle

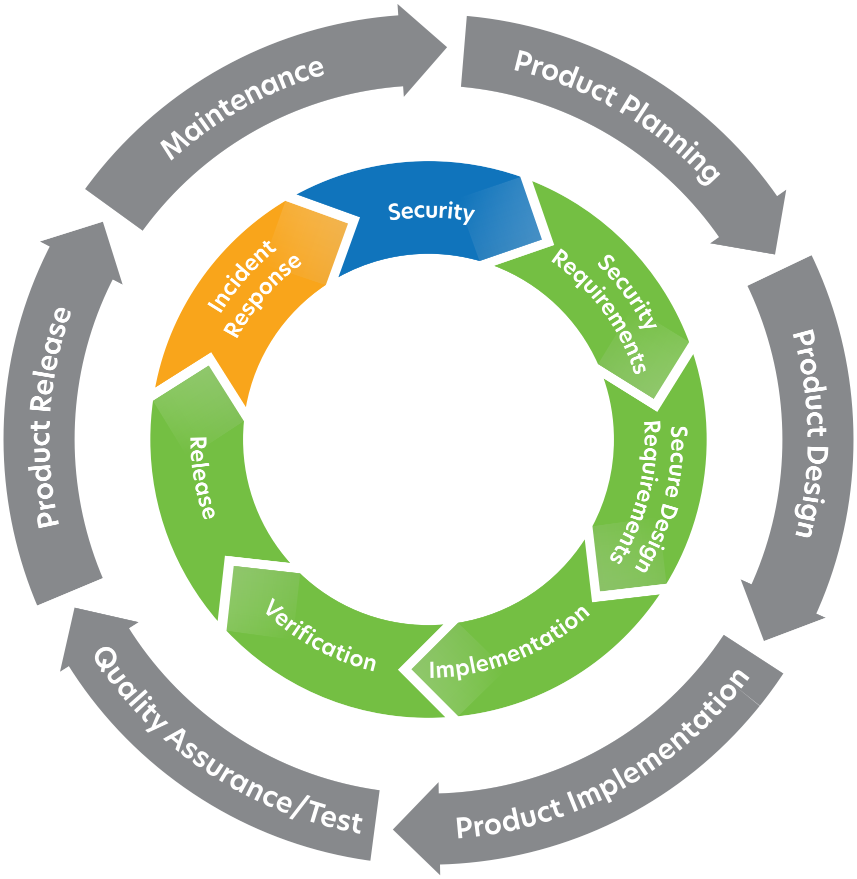
Lexmark Secure Software Development Lifecycle

Introduction

Lexmark is a global technology company that creates enterprise software, hardware and services to help organizations draw deeper value from their business information and serves customers in 170 countries. This document describes Lexmark's process for developing products, both software and hardware, that are more secure and better able to meet the security requirements of our customers. It provides a description of a generally applicable security assurance process for the improvement of software security that has been modeled after current industry best practices.

Overview

Lexmark's Secure Software Development Lifecycle process consists of a series of product activities designed to address various aspects of security related to the software development process from Planning through the Design and Implementation phases and finally through Quality Assurance, Release and Maintenance. These activities are illustrated in the diagram and are described in detail in the following sections.



While most of the security practices are generally applicable to all Lexmark software and hardware, Lexmark evaluates each software/hardware product with respect to the most applicable and appropriate security practices for that product or product class based on actors including but not limited to target market, product maturity, and target user environment.

Lexmark Security Training

Lexmark provides numerous avenues for security-related training that is available to all employees of the company, regardless of role. Furthermore, security training is mandatory for all employees involved in the planning, design, engineering and testing of any Lexmark product. This includes program managers, software architects, code developers, and quality assurance staff. It is also required at least biennially for current employees and is part of the on-boarding process for new employees in these roles.

Lexmark's security training curriculum includes courses with associated assessment tests covering many aspects of secure software development. The training system tracks completion status and test scores, ensuring Lexmark's developers are aware of and in compliance with current security best practices.

Security Awareness

Basic security awareness training is mandatory for all Lexmark product research and development staff. Topics covered by this training include but are not limited to:

- Fundamentals of Application Security
- Software Security Awareness

Security Engineering and Design

Training in the area of security engineering design is required for domain leads, developers, quality assurance, and project/product managers as applicable to their specific roles. Topics covered by this training include but are not limited to:

- Creating application security design requirements
- Performing a security code review

Secure Coding Practices

Training in the area of secure coding is required for all Lexmark software developers as it applies to the particular language or product component area of expertise. Topics covered by this training include but are not limited to:

- Fundamentals of secure development
- Input and output validation

Threat Modeling

Threat modeling training is required for developers, quality assurance, and project/product managers, as applicable to their specific roles. Topics covered by this training include but are not limited to:

- Creating an Application Security Threat Model
- Attack Surface Analysis and Reduction

In addition to the available eLearning training related to threat modeling, Lexmark has a series of recorded on-site education sessions from a third-party software security expert that demonstrates the threat modeling process as it is applied to specific Lexmark software products.

Security Testing and Assurance

Training in the area of security testing is required for all Lexmark software developers and quality assurance staff as it applies to their particular product component area of expertise. Topics covered by this training include but are not limited to:

- Fundamentals of Security Testing
- Application Penetration Testing

Product Planning and Design Practices

While the consideration of security and privacy-related issues and tasks is important during all phases of the development cycle, it is generally most effective when security requirements are defined at the start of the process during the initial planning and design phases. Waiting to address security during the later stages of the development cycle can potentially be expensive in both development time and cost. Irrespective of the type of project development cycle being initiated (e.g., new products vs. a new underlying feature for a current product), it is important to thoroughly analyze, define, and document the security requirements before starting the implementation phase of the development cycle.

Security Requirements Definition

During the planning phase of each Lexmark software development project, the project is analyzed and the security requirements for the project are determined. Specific security requirements can differ for a particular project (product or feature) depending on the type of software development project being initiated and the target market for the product. For example, a product being designed to process personally identifiable health information can have a vastly different set of security and security feature requirements (HIPAA and HITRUST) than a product being designed to process other types of information.

Lexmark's development architects and security staff determine these criteria by leveraging applicable industry and government standards such as:

- OWASP Application Security Verification Standard (ASVS)
- SANS Securing Web Application [SWAT] Checklist
- CWE/SANS Top 25 (non-web applications)
- Application Security and Development Security Technical Implementation Guide (STIG)

Lexmark software teams use the set of industry standards and best practices that is most applicable to the type of software/firmware as well as the type of product/feature being developed to determine the security requirements for the project.

Quality Gates/Bug Bars

Another planning phase security practice that every Lexmark software development project performs is the definition and documentation of the minimum acceptable level of security for the project from a "quality" perspective, establishing a threshold for the types of security issues that must be addressed and at what stage of the development cycle. This includes how security issues found during the implementation phase of the development cycle (either by developers or various testing tools) are ranked and classified in the bug tracking system.

Generally, the quality gates and release criteria with respect to security issues are consistent across all Lexmark software development teams. However, there may be slight variations in the ranking and classification based on the type of development project. An example of this may be that a server-side software development project may have a higher-ranking process for issues affecting stability than a client-side software project.

Additional Security Practices

The security requirements definition stage of the software development project is also when the Lexmark development architects and security staff define the "level" of security verification and documentation required for the particular project. This can include additional design reviews, additional security testing, or external expert review or testing as deemed necessary.

Design Requirements Definition

During the design phase of the development cycle, Lexmark's development architects and security staff leverage the security requirements defined in the planning phase of the development cycle and distill specific security design specifications. These specifications detail security feature requirements such as authentication or encryption and determine potential security risks and vulnerabilities that need to be addressed in the design. This may include the creation of a formal threat model to identify potential threats that could lead to a compromise of the product.

Threat Modeling

A formal threat modeling process is used for all Lexmark software products whose security requirements definition indicates an operational environment where there may be a meaningful security risk. The process identifies potential threats that could compromise the product being developed, or potential threats that might be used as a launch point to compromise other components or systems that interact with the product. The threat modeling exercise includes:

- System decomposition that identifies the actors, processes, data flows, data stores, and trust boundaries
- Identification of the critical assets
- Identification of potential threats to those assets
- Determining the impact for each threat identified
- Determining the probability (risk) of compromise by an attacker
- Ranking of the risks
- Determining mitigating countermeasures as needed

A result of the formal threat modeling exercise is that it helps Lexmark's architects and developers define each product's attack surface (the areas of the product where it can be compromised).

Attack Surface Analysis/Reduction

Closely aligned with the threat modeling exercise is attack surface analysis and reduction. This practice is a means of reducing the risk of compromise by blocking, or reducing, an attacker's opportunity to exploit a weakness via mechanisms that aren't necessarily related to the design, but may leverage other system or environment capabilities. For example: Data being transmitted over a network is at risk of being exposed to an attacker with a network sniffer. That attack surface can be reduced/eliminated by using an encrypted channel to transmit network traffic as opposed to encrypting the data at the sender and having to design a key management system to allow the receiver to decrypt the data.

Implementation Practices

During the implementation phase of each Lexmark software development project, care is taken that the implementation practices and processes leverage the appropriate security-related configurations and analysis tools during the software build pipeline process.

Use of approved tools and code functions

For new-product development activities, Lexmark's development/build pipeline uses the most current versions of various compilers and tools for the development environment best suited to each product's requirements. The build process includes, where available, checksto ensure that appropriate compiler/linker options and warnings are enabled and reports are reviewed for security-related issues. For forward-compatibility reasons, Lexmark also maintains earlier versions of various compilers and tools for use during the maintenance phase of the development process.

Lexmark's development/build process also contains checks for deprecated code functions and routines and issues warnings when they are used.

Static Application Security Testing (SAST)

Lexmark's software product development process includes static analysis of the source code, where appropriate tools for that language are available. Alternatively, compiled "binary" code scanning tools can be used to scan for security-related defects. This scanning is usually done as part of the Lexmark development build pipeline, but the scanning tools can also be used separately as part of an auditing or final security review process.

The most widely used static analysis tools throughout Lexmark's software development teams are the Coverity® source code scanning tool and the Veracode binary code scanning tool, though other task-specific static analysis tools may be used when appropriate.

Lexmark has also integrated Black Duck OpsSight into our build procedures to validate open source and license usage of components within our firmware and security. This tool helps us inventory our software usage and be aware of any updates required by updates.

Security-related issues found from these scans are evaluated by the architects and developers and are ranked and classified per the quality requirements defined during the planning phase of the development cycle.

Any security-related issues that map to the OWASP Top 10 are fixed and verified before the release phase per the project's quality and bug bar requirements.

Software QA/Test Practices (verification)

While quality assurance and system testing are a continuous part of Lexmark's normal software development process and are performed throughout the design and implementation phases of the development project, some types of products can require additional security-related verification nearer to the end of the development cycle. Such products are those that contain independent subsystems that are independently developed and tested and then integrated into a final complete system (e.g., printers and multifunction products). For these types of systems, security related verification testing practices such as dynamic, run-time analysis tools and targeted fuzz testing are completed after system integration. These testing practices are part of the normal continuous testing philosophy otherwise.

Dynamic Application Security Test (DAST)

Lexmark's software product development process includes verification testing using multiple dynamic runtime analysis tools. Because of the wide variance in the types of software developed by Lexmark's software teams (web software, server-based software, embedded firmware); DAST tools used by Lexmark's development teams throughout the organization include but are not limited to Acunetix, BurpSuite, Arachni, Nextpose, Nessus, and IBM Appscan.

Security-related issues found from these DAST tools are treated much the same as issues found by either software developers or by the SAST tools. Issues are evaluated by the architects and developers and are ranked and classified per the quality and bug bar requirements defined during the planning phase of the development cycle.

Any security-related issues that map to the OWASP Top 10 are fixed and verified before the release phase per the project's quality and bug bar requirements.

Fuzz Testing

Where appropriate, Lexmark's software QA/test process also includes a specialized form of dynamic analysis to try to induce program failure by introducing deliberately malformed or random data into the input channels of an application or program (Fuzz Testing). Software where this is appropriate includes network protocol handlers, data stream interpreters, or in cases where human input may be acted upon and not merely recorded.

Lexmark uses a combination of commercial, in-house developed, and manual fuzz testing to improve the robustness of the software.

Automated Testing

Where appropriate, Lexmark uses a combination of both manual and automated testing in the development and QA process. The use of automated testing allows teams to increase overall testing efficiencies and to find problems earlier in the development cycle.

Automated testing is used at various phases in the test process, including:

- Unit Testing
- Integration Testing
- System Testing

Source Code Controls

Lexmark has always focused on secure access to source code and release procedures. This can include day-to-day activities as well as the onboarding and offboarding of developers. All code is maintained in a Code Management System that logs and tracks all code modifications.

Procedures for Source Code controls include validation for the following activities:

- Developer Onboarding
- Developer Offboarding
- Request Code Access

- Remove Code Access
- Firmware Release Process
- Software Release Process

Each of these procedures includes built-in monthly validation based on employee status, access level required, and usage activity.

While we maintain internal controls, a large amount of focus is on the release process. In order for Firmware to be generally available the following must occur:

1. Firmware is digitally signed and encrypted by the build tools.
2. Customer-ready code is declared from all involved groups in operational review.
3. Code is reviewed and signed off by United States (US) management.
4. Code is uploaded to a secure location along with a secure hash for validation.
5. Code is promoted through internal procedures and file validation performed.
6. Controller card is built with the new firmware and all file comparisons are validated.
7. IT Security runs ongoing hash comparison scripts once a quarter and archive results.

Software releases follow a similar path. This software includes all drivers, device management software, print management software, and embedded solutions. The path for Software includes:

1. Production code is declared from all groups involved in operational review.
2. Release checklist is reviewed by United States (US) management.
3. Request form is completed for posting.
4. Product is released through build procedures.
5. Secure hashes are captured for audit purposes.
6. Notification is sent to Release Team, IT Security, and Lexmark personnel responsible for publishing and/or distributing to end users.
7. IT Security runs ongoing hash comparison scripts once a quarter and archives results.

Release Practices

Security-related practices that are performed during the release phase of the Lexmark software development process include activities intended for final or near-final (function-complete) versions of the software or firmware under development. These practices include a Release Security Review and may include Application Penetration Testing if this testing is indicated in the security requirements definition.

Application Vulnerability Assessment

Penetration testing is simply an attempt to evaluate the security of an application or system by safely trying to find and exploit unknown vulnerabilities in the system (i.e., ethical hacking). Penetration testers use a combination of manual and automated hacking methods to systematically compromise an application, server, network device or endpoint that is part of the software or firmware system, attempting to gain access to the critical assets of the system to impact their confidentiality, integrity, or availability. Once a vulnerability has been found and exploited, the penetration tester may attempt to use that compromise to launch subsequent attacks against other parts of the system under test, or other systems on the network. Because of the “ethical hacking” nature of the testing, the penetration tester does not actually corrupt or otherwise compromise the assets of the system; they merely provide detailed documentation about the vulnerabilities found and the exact methods used to exploit the vulnerability.

This type of testing is performed for software development projects whose security requirements definition indicates the need for penetration testing. This penetration testing is performed by an internal test team with penetration testing expertise and occasionally by an independent firm that specializes in software penetration testing; or both in some cases.

Ideally, penetration testing is completed and the results are addressed prior to the release of the software product or feature. However, when penetration testing is performed by an independent third party it is often prudent to perform the test on a released version of the software due to the following factors:

1. Independently testing a pre-release version may miss vulnerabilities induced in the final stages of implementation and testing.
2. The time it takes to initiate (bid, contract, develop a statement of work); configure the test setup; perform the testing; and receive the test report for a full code-assisted third-party penetration test can significantly impact the release timeline for software products with shorter release cadences (e.g., quarterly releases).

Release Security Review

Prior to exiting the release phase of the Lexmark software development process, a Release Security Review is performed to examine the overall product or feature’s security posture. The goal of this review is to ensure that the appropriate practices in the SSDL have been properly completed and to evaluate any remaining security-related issues that are above the minimum acceptable level of security defined in the planning phase of the project and determine if they can be eliminated, reduced, mitigated or accepted.

This review is performed by Lexmark’s security staff and the development teams involved in the software development project.

Incident Response Process

Lexmark’s ultimate goal is to produce software and hardware that is free from security-related vulnerabilities. However, the sheer complexity of the code in the products results in the need to be able to address security-related issues in released products. Lexmark’s software and hardware products contain hundreds of files, thousands of objects, and tens of millions of lines of code and a product that was released with no known vulnerabilities may indeed have new vulnerabilities identified over time. This can be due to a previously unidentified vulnerability found in custom code written by Lexmark, in a common shared system library, or in a third-party library integrated into the Lexmark software or firmware.

Lexmark’s security staff and experts monitor multiple channels for the identification of new security vulnerabilities including internal review, customer service, security-focused press, security-related academic research, and technical alerts from organizations like NIST-National Vulnerability Database and US-Computer Emergency Readiness Team (US-CERT). Additionally, Lexmark uses scanning tools during the implementation phase that scan source code for out-of date or vulnerable shared libraries.

When new vulnerabilities are identified which might affect Lexmark’s products, they are addressed via the following process:

1. The vulnerability is analyzed to determine if it affects the product. (Vulnerabilities found in shared system or third-party code libraries may not apply, depending on the way the code is used in the system).
2. Lexmark's security staff determines if the exploit mechanism for the vulnerability is possible in Lexmark's implementation.
3. If yes to 2, above, the security bug is scored using industry standard Common Vulnerability Scoring Systems (CVSS). Note: The severity score published in a technical alert may score differently in specific implementations.
4. Internal processes are initiated to log, track, patch, and test the bug fix, and updated code is provided via a patch process.
5. If the CVSS score warrants, Lexmark issues a security advisory for the products affected.

When issued, Lexmark security advisories are posted on http://support.lexmark.com/alerts. Lexmark also offers a subscription service to notify subscribed customers when a security advisory is posted. Also available is the ability for someone to submit a potential vulnerability or concern to our team. This submission form allows for direct communication with our subject matter experts. We then follow our standard vulnerability process to assign severity and timelines for resolution.

Conclusion

Lexmark's Secure Software Development Lifecycle process is a series of product activities designed to address various aspects of security as related to the Lexmark software development process.

This process provides a framework for designing enterprise software and hardware products that are more secure and resilient in the changing security landscape and is essential to meet the security requirements of our customers.