ACI Worldwide®

ACI's development security management practices

October 2026



Table of contents

Introduction 3

Static code analysis 4

Application penetration testing 5

Third-party component management 6

Processing scan results 7

Threat modeling 7

Governance & exceptions 7

Continuous improvement 8

Confidential & proprietary information 8



Introduction

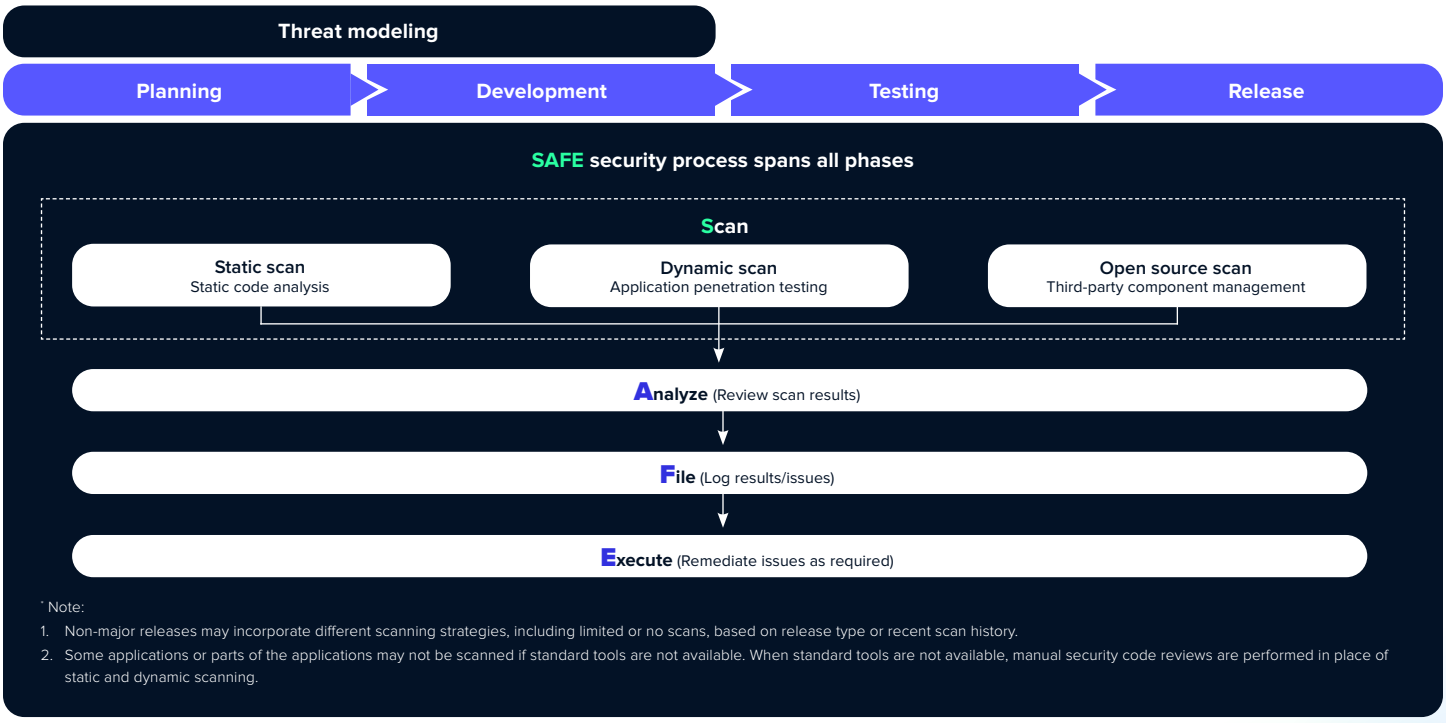
ACI is committed to industry best practices for managing security within its products and solutions and thus follows a structured secure software development lifecycle (SSDLC) approach that requires security-related artifacts at every stage (concept generation, project definition, design, development, validation, and distribution and maintenance). SSDLC principles are embodied within ACI's product development methodology (PDM), which accommodates waterfall, agile, and iterative development methodologies. Regardless of development methodology, robust security-related development steps and artifacts are required, including:

1. Static code analysis (also known as static scanning
2. Dynamic code analysis (also known as dynamic scanning or penetration testing)
3. Third-party component scanning and vulnerability management (also known as software composition analysis or SCA)

In addition, ACI engineers undergo mandatory annual secure coding training. This process is designed to address sensitive personally identifiable information (PII) and payments data used within ACI's solutions while satisfying industry standard best-practice behaviors as called out by PCI and PA-DSS expectations to mitigate common threats like the OWASP top 10 for sensitive cardholder data.

This document provides an overview of the capabilities of these tools within the context of ACI's SSDLC and PDM applied to our Active products. Mature products follow similar practices, but at a slower cadence than the Active products.

Release development-related security activities



* Note:
1. Non-major releases may incorporate different scanning strategies, including limited or no scans, based on release type or recent scan history.
2. Some applications or parts of the applications may not be scanned if standard tools are not available. When standard tools are not available, manual security code reviews are performed in place of static and dynamic scanning.

Static code analysis

Security-related static analysis (aka static scanning) inspects code for patterns that may adversely affect the ability of the application to maintain confidentiality, integrity, and security of all or part of the application. Static code analysis may also find defects that are not security related but are quality related. A sampling of security- and quality-related code defects that static code analysis can discover are available upon request.

For all Major or Enhancement releases, the PDM requires that the engineering teams ensure that static code analysis has been performed in a timely manner within the project lifecycle, and that mandated remediations be completed before development is complete. Additionally, developers are encouraged to scan code changes in near-real time from their development environments to identify potential security defects early in the development lifecycle.

Static code analysis can be accomplished through manual or automated code inspection.

Automated static code analysis

Many automated static scanning tools are available that detect and classify security- and/or quality-related anti-patterns. The PDM specifies that only tools that are pre-approved by security engineering may be utilized for this purpose. Presently, approved tools include: Checkmarx and Klocwork Insight. Checkmarx is ACI's preferred tool because it supports nearly all of ACI's diverse development environments and integrated development environments (IDEs), and has a strong repertoire of customizable rules (also called "queries").

Checkmarx includes strong IDE that facilitates detection and resolution of vulnerabilities. Lines of code are highlighted with issues found via the Checkmarx scans. Developers can quickly navigate through the IDE to all files related to the defect, and thus are able to fully understand the issue. The tool helps identify areas where a change in behavior may result in the resolution of multiple issues, thus improving the quality of the changes and efficiency of the developer.

Checkmarx supports many languages and frameworks, including Java, C#/.NET, PHP, Visual Basic 6.0, VB.NET, APEX, Ruby, Javascript, ASP, Perl, Android, Objective C, PL/SQL, HTML5, Python, and Groovy. For C and C++ applications, Klocwork is our tool of choice.

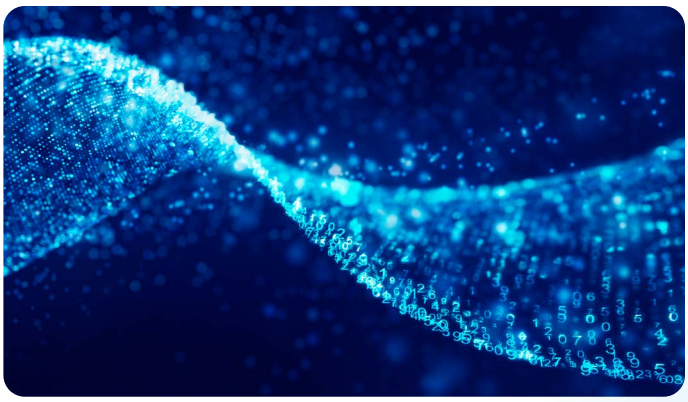
When using automated tools, each automated static code analysis tool must be executed on the entire code base using the list of queries or rules prescribed by security engineering.

The results are uploaded to a central server to facilitate review and tracking across multiple builds and projects. Each defect is assigned a unique defect ID and a severity level. The engineer works with developers to triage the results and resolve issues within the project. Analysis of false positive issues requires review by security engineering.

Manual code analysis

Manual code analysis (aka code reviews) may be performed instead of or in addition to automated static code analysis. The PDM site includes a set of best practices, including documented coding standards, code review checklists, and code review templates. Manual code reviews must follow PDM standards for documenting the reviews, the participants, and any issues identified.

When conducting code reviews, many development teams use workflow tools that support the effort, such as Atlassian's Crucible, to enforce peer code reviews prior to checking code into the source code repository.



Application penetration testing

Application penetration testing (aka dynamic scanning or pen testing) detects security flaws in run-time application behavior (rather than in code). The method employed simulates normal and malicious user behavior against a deployed version of the product under test to expose symptoms of underlying security flaws in the application code. Dynamic code analysis is typically accomplished through a combination of tools and manual modeling of user-application interaction.

The PDM requires dynamic scanning to be completed for all Major and Enhancement releases and to test security fixes. The goal of dynamic scanning is to identify remaining security vulnerabilities after static code analysis and associated remediation have been performed. Dynamic scanning may be performed using both automated testing tools and manual testing techniques.

Automated dynamic scanning

ACI development teams may use satellite security members to perform automated dynamic scanning as part of their internal milestones using preferred tools including HCL AppScan Standard and HCL AppScan Enterprise, and secondary tools like Burp Suite Pro and other developer-centric tools to support this level of scanning. The goal of this activity is to identify and resolve common web application and application programming interface (API) vulnerabilities early in the development cycle. The scope of these scans may be limited to testing only the portion of the application affected by the changes implemented within the development period, or it may include testing of the entire application.

Dynamic scanning tools generally search for three types of vulnerabilities: application, infrastructure, and third-party web components. ACI dynamic scanning focuses primarily on application-level vulnerabilities, but also includes scanning for the other types of vulnerabilities present in the development environment.

Manual dynamic scanning

Automated tools performing application penetration testing test many variations of common web application vulnerabilities, but they typically are unable to detect a class of vulnerabilities involving data entered in one part of the application but consumed in a different part. The goal of manual security testing is to probe the application for vulnerabilities not detected by the automated testing tools and to test workflows that are not easily tested using automated tools.





Third-party component management

Many ACI products leverage the use of third-party components, such as open-source software. These components may carry specific licensing requirements as well as have the potential for having vulnerabilities that could impact the security of the product. To address the potential issues that could arise, ACI policy requires that all products be scanned to identify all third-party components in use and that ongoing monitoring for vulnerabilities is used on supported release threads.

The ACI security policy requires that third-party component scanning be completed for all projects that are Major, Minor, or Enhancement releases and when there are changes to the third-party component versions. All third-party components delivered with the application that are identified as the result of scanning are required to be submitted for review for legal (licensing), security, and export control purposes.

Third-party component scanning

ACI utilizes Black Duck Hub for third-party component scanning within the applications. This tool provides the ability to scan the ACI product to identify all third-party components that are in use. Once components have been identified, the tool provides detailed information on the component(s) including licensing, security, and operational issues.

The Black Duck tool provides access to the Black Duck knowledge base which actively monitors the National Vulnerability Database (NVD) and other sources for common vulnerability and exposures (CVE) data. In addition to identifying components that are in use by a product, the Black Duck tool provides the ability to scan ACI source code to identify sections of code that may have been copied from a third-party source. This functionality, known as snippet matching, allows ACI to identify embedded third-party source code that may carry additional licensing requirements.

Applications that bundle the ACI components with additional third-party content such as middleware into a self-contained image for distribution are required to undergo additional scanning with Anchore. The Anchore tool then monitors the bundled content for publicly disclosed vulnerabilities. Images with vulnerabilities are tracked for vulnerability remediation.

Technical review of intellectual property

When third-party components are identified as a result of the scanning process, the PDM and ACI policy require that all such components be reviewed using the Technical Review of Intellectual Property (TRIP) process. This process provides for the review of the third-party components from a legal (licensing), security (vulnerability), and ECCN (export control) perspective.

The TRIP process helps to ensure that:

- Use of third-party components complies with the licensing requirements specified by the component supplier
- Security issues are identified and remediated. Any component with outstanding vulnerabilities or that is no longer supported is not approved for use in ACI products
- Third-party components that contain cryptographic functionality that could affect the export classification of an ACI product are identified

Third-party component vulnerability management

When an ACI product that contains third-party components is released, the tooling provides the ability to perform ongoing monitoring and management of newly discovered vulnerabilities. The tooling periodically (on a daily basis) retrieves new and updated information from the NVD to identify any new vulnerabilities that may apply to third-party components in use or bundled with an ACI product.

ACI has developed additional automated processes to generate internal cases for managing issues and vulnerabilities identified by our tools and processes.



Processing scan results

Vulnerabilities are classified as consistently as possible according to an industry-accepted common vulnerability scoring system (CVSS). Security engineering reviews the vendor-assigned severities and tightens them up where appropriate to reflect the stringent security requirements of individual payment applications. The tenets of ACI's scoring system are as follows:

- **Critical** – High probability of a vulnerability being exploited which could result in the loss of confidentiality, integrity, or availability of sensitive data, or the loss of integrity and availability of the application. Usually associated with a CVSS score of 9.0 or higher.
- **High** – High probability of a vulnerability being exploited which could result in the loss of confidentiality, integrity, or availability of sensitive data, or the loss of integrity and availability of the application.
- **Medium** – Medium probability of a vulnerability being exploited which could result in the loss of confidentiality, integrity, or availability of sensitive data, or the loss of integrity and availability of the application. This classification is also used to identify possible coding errors which may affect business logic.
- **Low** – Low probability of a vulnerability being exploited which could result in the loss of confidentiality, integrity, or availability of sensitive data, or the loss of integrity and availability of the application.
- **Informational** – A variant of low probability vulnerabilities related to the leakage of information or identification of sub-optimal coding practices.

A summary of outstanding security issues is created during the validation phase of a project, identifying the number, severity, and type of security issues detected.

Threat modeling

ACI uses the IriusRisk threat modeling tool to identify and track threats, weaknesses, and countermeasures related to our Active products' architectures and designs. IriusRisk provides architects with a well-defined user interface to model their designs and identify threats, weaknesses, and countermeasures both manually and using libraries of compliance and security rules and artifacts. It integrates with issue tracking systems and supports report generation demonstrating compliance with various standards and regulations.

Governance & exceptions

The PDM requires static scans and dynamic and third-party component scans be performed for all Major release projects and for all Enhancement type projects. Other types of projects, including patches, service packs, and mandate projects may be exempt from the scanning requirements, but scanning is required if those projects include product enhancements. Projects that do not alter the web UI or application interfaces may be exempt from the dynamic scan analysis requirement. Any customer-specific modifications (CSMs) should refer to their statement of work for details of the types of scans and remediation to be performed on each modification.

Within the PDM a Scan, Analyze, File, and Execute (SAFE) process includes requirements to classify and track any detected vulnerabilities through remediation. These actions are tracked in release checklists and reviewed by business process analysts to ensure projects have complied with the PDM.

Projects that have outstanding critical- or high-severity security issues at the end of the development phase cannot be released for general availability unless the issues are being tracked for remediation and the product manager approves.

Continuous improvement

ACI prides itself on following a continuous improvement philosophy to bring best-in-class products and solutions to our customers. This philosophy extends to security scanning tool selection. We proactively assess scanning tools we feel will: expand the suite of detectable vulnerabilities and coding issues, support the languages and frameworks used at ACI, provide the ability to detect and resolve issues earlier in the secure software development lifecycle with fewer false positives, and use flexible architectures that permit the development of custom rules and extensions. We look for tools that integrate seamlessly with developer IDEs, version control systems, bug tracking systems, and other tools in our toolkit.

Tool assessment is an ongoing activity. We assess new tools and technologies as they become available and reassess existing technologies whenever there appears to be sufficient improvements to warrant our re-evaluation. When change is warranted, we manage it quickly and efficiently through our tools strategy and application development tools teams dedicated to providing world-class service to our development community.

Confidential & proprietary information

ACI policy mandates that the security scanning reports produced internally by ACI are considered ACI confidential and proprietary information and are not to be shared externally. Although ACI works to resolve the indicated types of issues prior to a release becoming generally available, earlier releases may exhibit unresolved vulnerabilities due to progressively improving defect detection processes within ACI. Security scan results are kept confidential to mitigate and minimize security risks for all ACI customers.

ACI Worldwide, an original innovator in global payments technology, delivers transformative software solutions that power intelligent payments orchestration in real time so banks, billers, and merchants can drive growth, while continuously modernizing their payment infrastructures, simply and securely. With more than 50 years of trusted payments expertise, we combine our global footprint with a local presence to offer enhanced payment experiences to stay ahead of constantly changing payment challenges and opportunities.

LEARN MORE

www.aciworldwide.com
@ACI_Worldwide
contact@aciworldwide.com

Corporate headquarters +1 402 390 7600

© Copyright ACI Worldwide, Inc. 2026
ACI, ACI Worldwide, ACI Payments, Inc., ACI Pay, Speedpay, and all ACI product/solution names are trademarks or registered trademarks of ACI Worldwide, Inc., or one of its subsidiaries, in the United States, other countries, or both. Other parties' trademarks referenced are the property of their respective owners.

00163 11-26