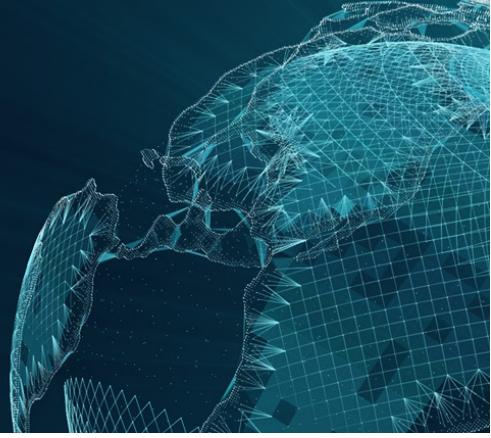
Why QA Should be Doing Security Testing



INTRODUCTION

Historically QA was focused on identifying logical failures in the code written by engineers. In today's world, where security vulnerabilities are a far more significant threat than product "bugs", it is imperative that the role of QA expands to cover security testing in addition to its classical role.

The purpose of security testing as a part of the product QA strategy is to address security concerns before the application is packaged and released into production while leveraging established testing practices and relationships with development. Security testing both ensures the stability of an application and minimizes the likelihood of negative outcomes that malicious adversaries could cause by exploiting vulnerabilities.

Integrating security testing with QA in a modern development environment significantly reduces risks and helps streamline the SDLC by shortening the application release cycles and reducing failed application releases. Historically QA specialists did not have the ability to conduct vulnerability tests due to lack of training and knowledge. This is no longer the case and QA specialists no longer need to spend time analyzing vulnerabilities, but instead focus on collaborating with developers to remediate the identified vulnerabilities. All the while enabling developers to become more knowledgeable about application security, thus preventing the reappearance of vulnerabilities after each cycle of development.

The advantages of enabling security testing as part of the QA process

For QA

With the help of NexDAST, security testing can be easily carried out as part of the quality assurance process without the need of security expertise. QA teams can extend functionality-based testing by empowering their tools and techniques to address security concerns in any stage of the development lifecycle. As a result, they can step-up their contributions to the development lifecycle by reducing security risks that could impact a running application. Moreover, by engaging in application security testing, the QA personnel can advance their professional competency with a security skillset as they perform each scan.

For development

Security vulnerabilities are reported to developers much faster, making their job of remediation, much more efficient and easy. Features such as a reproducible proof-of-concept, issue severity, and a suggested remediation plan, significantly reduce the time needed to analyze and remediate risks, thus streamlining the SDLC with faster application releases and reduced time to market.

Developers become more knowledgeable about security and correct coding practices, by getting the relevant information exactly when they need it. Thus, improving the culture of security risk awareness in the organization.

How to utilize AIAST in the QA process

Vulnerability detection with NeuraLegion's AIAST is initiated directly from a web browser by recording interactions within the application that are performed manually (manual QA), or as part of automated unit testing (automated QA). The recorded interactions create an HTTP Archive (HAR) file, which the engine analyzes to extract the attack surface to be scanned for vulnerabilities.

In a software development life cycle, quality assurance testers initiate a scan of the target endpoint (or feature) as soon as the latest version of the application is deployed to a testing environment. The engine then explores the targeted application's environment, inspects the runtime components, and reports the identified vulnerabilities in real-time over the web dashboard; via integration to SDLC tools; or via a customized API.

Identified vulnerabilities are ranked, and a suggested remediation plan is generated for each vulnerability. This allows QA testers to easily communicate security issues to developers who are assigned to remediate them.

As vulnerabilities and exact proof is provided to developers, they can focus their efforts on remediation without wasting their time analyzing each vulnerability.

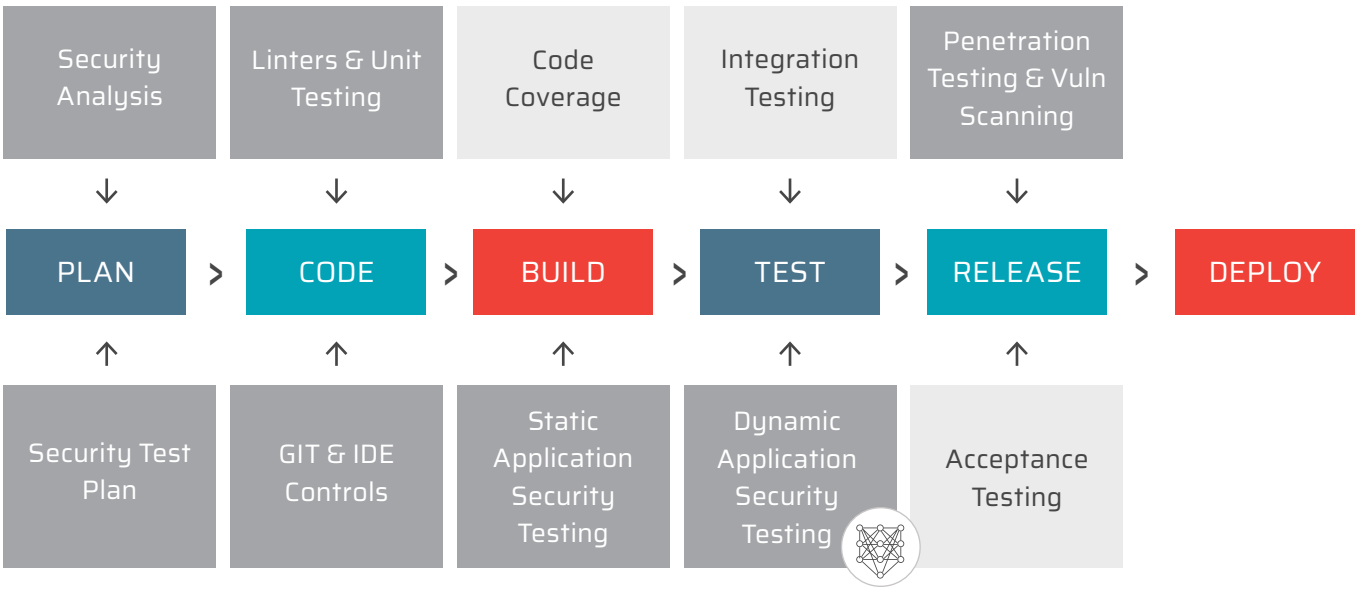
The application is built and packaged in the SDLC

The application is deployed to a testing environment

A HAR file is recorded to track interactions with the target application

The HAR file is uploaded to the scanning engine

The engine scans the application to detect and expose vulnerabilities



Integration of QA and security testing in a development pipeline

SUMMARY

In today's fast-paced development environment, organisations run a significant risk if not incorporating security testing into their processes. Security testing is not a core skill of most QA organisations and thus is not considered one of QA's responsibility, however they are already performing many of the tasks required for security testing.

Accepting a certain amount of exposure to risks without extensively testing the application exposes businesses to malicious adversaries who exploit both applications and the resources used by these applications, causing tremendous damage to business goals and reputation.

Automated AST solutions enable Quality Assurance effortlessly identify and address security vulnerabilities after each step in the development process.

NeuraLegion's NexDAST enables organizations to seamlessly integrate security testing into their QA process with no investment in training the QA teams to manually identify, analyze, and report security vulnerabilities. Instead, QA teams can conveniently engage in application security testing to produce actionable results, that are delivered to development for remediation, as efficiently as functionality bugs are reported. Moreover, QA specialists that use NexDAST will improve their security skills, building a better security culture in the organization's SDLC, while at the same time streamlining the QA process and making sure that the application is published to production without any security risks.