



Application Security in the Software Development Life Cycle (SDLC)



Table of Contents

Executive Summary	3
The Rush to Get Applications to Web, Cloud and Mobile.....	3
Issues in Software Development	3
Challenges in Application Security Testing.....	4
Who Benefits from Application Security Testing?.....	4
Requirements for Securing Application Testing.....	5
Cenzic’s Approach to Application Security Quality Assurance.....	6
The Experts Take	6
About Cenzic	7



Executive Summary

New security vulnerabilities are found almost daily. When they are, affected software must be retrofitted with an appropriate patch while companies fend off the wrath of customers. Applications have become the most fertile ground for attackers to ply their trade – seeking out the seemingly innocuous features and utilities in today’s complex systems that can give them unauthorized access. Each newly discovered vulnerability results in a frantic patch, halfway deployed, that potentially opens up another hole – all taking days, weeks or months to implement throughout an installed base. Meanwhile, new vulnerabilities continue to be found, making the game of catch-up never-ending. The critical place to address security vulnerabilities is in the software development process. This white paper discusses the challenges of enabling security in the software development process and introduces Cenzic’s solution to automate security quality assurance with its five-step methodology.

The Rush to Get Applications to Web, Cloud and Mobile

A rush to get applications on website, into clouds and onto mobile devices has left the application security of the connected world in a state of disrepair. Organizations of all sizes rushed to get new apps online without much thought about their longevity or reliability. The applications, in most instances, had been slapped together in a rush to get to market, with minimal thought given to security at the application code level. As long as the application worked, the thought went, the firewall could perform the security “heavy lifting.”

But the firewall can’t do all the work. As systems evolve, the nature of the threat matures and morphs in new ways. Attacks that focused at the network level were largely unsuccessful against properly configured firewalls, so attackers looked for other ways.

With the discovery that web traffic (via port 80) was allowed in through virtually all firewalls, attackers hit upon a treasure trove of vulnerabilities – the millions upon millions of lines of code that make up the complex web applications on the Internet.

Hackers seek out weaknesses in the many modules and components of complex systems, looking for hidden fields, embedded passwords, and available parameters to manipulate. They attack the applications, seeking out ways to manipulate input strings to steal data or create buffer overflows for super-user access. They seek out weaknesses in the many modules and components of complex systems, looking for hidden fields, embedded passwords, and available parameters to manipulate.

Security breaches continue to occur in Cloud, Mobile and Web applications because we have not addressed a core problem with those applications: insecure software development and lack of security testing. Human error resulting from lack of education or skill can cause these issues to surface, as can the increasing complexity associated with integration of disparate applications and systems that have not had to communicate in the past. To compound matters, most third-party software has the same issues as software developed internally. Attackers now see the application as fertile ground for exploitation and intend to stay for a while as businesses move more and more critical functions to networks, neglecting the need for a methodical, proactive, and comprehensive security testing process.

Issues in Software Development

Methods of software coding are being used that are inherently insecure. Logic modules are written that create security issues when they are combined, and implementations are rolled out that create vulnerabilities rather than eliminate them.

There is no question that development is a complex undertaking. When 100 programmers collaborate on a software project, they work on individual software functions or architectural components. These components are often developed separately and periodically tested to ensure



that they work together. But when it comes time to tie it all together, it takes a conductor to orchestrate the many different components and developer groups into a cohesive whole for testing purposes. This complexity of development makes integration testing difficult to accomplish in a comprehensive manner.

These issues and complexities in today's software development world lead to problems, with security vulnerability being one of the more significant byproducts. With development forces working furiously just to make an application work within all of its components (amid the sometimes unspoken pressure to be done on time and under budget), testing for success appears to be the path of least resistance. Testing for points of failure and looking for ways to "break" the system are often ignored. Even when they aren't ignored, it is a sophisticated process that needs to be managed to accomplish the breadth and depth of testing necessary to ensure security.

Challenges in Application Security Testing

Software development groups are faced with a key decision – should they attempt to identify and fix security vulnerabilities during development and testing, when they have control over the code, or should they risk having to fix a security vulnerability in software that has been packaged, distributed and implemented in thousands of organizations, with a help desk providing support along the way? The answer, however painful, is obvious. When developers build security into the development and quality assurance process, they reduce the total cost of ownership of the application from both the vendor's perspective and the user's perspective.

But, when developers figure out a way to accomplish a particular function, they test it to verify that it works and then move on to the next function. What they often neglect is testing to see under what conditions the function won't work – testing for failure. For example, manufacturing lines have perfected the ability to ensure that the appropriate amount of torque is put into twisting the cap on a soda bottle. Too much torque creates stress on the bottle (and a new market for bottle openers), and not enough torque could lead to leaks at the seams. Manufacturers figured this out by testing for failure.

In the computer hardware world, testing for failure is common, and a metric measuring the mean time between failures characterizes the life of the hardware. It is easy to understand the need to test for failure in hardware because the costs involved in rebuilding parts are enormous. Likewise, rewriting software can be very costly.

Every organization has its own architecture when it puts together the components in a software solution. Particularly on the Web, components like web servers, application servers, backend databases, messaging backbones, load balancers, peripheral capabilities and other systems are often being combined in a way that leads to ultimate value to a company. The location of these components relative to each other and within the overall process flow can lead to weaknesses and vulnerabilities. Testing these components together can be daunting, but it is a critical part of having a strong security posture.

Who Benefits from Application Security Testing?

Maintaining a strong reputation for security not only builds trust with prospects and customers, but also propagates the benefits of security testing across an entire organization – from the "corner office" executive through operations and development groups. In any situation, addressing security during the development process, prior to the deployment of an application, ensures reliability of the application and a high level of uptime.



A recent study highlights the concern that users have about software development. When asked their opinion about the current process for disclosing software vulnerabilities to the public, 44% of those surveyed said that disclosure was essential to forcing software vendors to write better code.

While developers must consider security within their development process, security operations staff members are responsible for the entire computing environment. The introduction of a new application can create significant risk to this environment. Security testing ensures that applications are developed in a way to minimize this risk so that they can be implemented with a higher degree of confidence. Security operations groups can act as a catalyst for the secure development process, which leads to cost savings and lower risk in the end.

Quality assurance engineers must ensure that an application functions as needed to the organization. If the application is mission critical, the time to market (or deployment time) must be counterbalanced by the risk that an application might need to be overhauled immediately after the discovery of a security vulnerability. These engineers are at the gateway between development and production and therefore must be fast yet complete (under plenty of pressure) in their final testing.

While other types of bugs can lead to the faulty functioning of an application, a security vulnerability can lead to a data breach. The value at risk to an organization is significantly higher for security holes. Developers must test and retest for security to learn new coding techniques and quickly gain the benefits associated with strong security development procedures.

Requirements for Securing Testing

It is clear that strategic organizations will opt to conduct extensive security testing prior to an application's deployment. Not only is it more cost-effective, but it also results in greater customer satisfaction and has the residual benefit of reducing the possibility that other functions will break during the patching process. Testing for security is defined as testing for failure of an application in conjunction with its operating system, network, and hardware. Security testing procedures must be applied to the entire environment – testing of individual components or a single layer is insufficient. These procedures must be consistent, repeatable and reusable. They must encompass the whole of security quality assurance. This is the only way software can be considered reliably secure.

The pursuit of reliability, however, is fleeting based on today's targets and tomorrow's expectations. That is, software testing is evolving from a process whereby reliability once meant the application worked and then that it performed with the many other applications and services with which it was integrated. Now reliability also means security – that the application will not only function as planned with all of its peer applications and perform as promised, but that it will do its share to ensure security of the application, the data, and the computing environment.

The complexity of development coupled with the realities of multiple developers and testers working within a particular environment creates the requirement for tests that are repeatable and reusable, regardless of who is conducting them. This is important when developers are resolving particular bugs as well as supporting the ongoing software revision and maintenance process.

Ultimately, security quality assurance means a full understanding of how an application will be implemented and comprehensive testing based on that understanding. It means not just focusing on individual components or the interaction with the outside environment, but also on providing a holistic security quality assurance testing process in a real environment, from routers and networks to applications and operating systems.



Cenzic's Approach to Security Quality Assurance

This need for a secure development process is more apparent every day, when the latest software vulnerability is publicized worldwide and hackers wait with bated breath. The problem is so complex that some security professionals are skeptical about the possibility that it can ever be solved.

Cenzic uses a unique methodology, rooted in software fault injection, to test an application as well as the whole network environment and observe the interactions among components to identify vulnerabilities. These tests include buffer overflow attacks, SQL injection attacks, cross-site scripting attacks, IDS evasion attacks and a host of others. This methodology automates today's manual (or nonexistent) security testing process, establishing a process that achieves the auditability and repeatability that is necessary for ongoing security. What is perhaps even more important is that it dramatically reduces the costs associated with security testing.

Cenzic employs a five-step process of security quality assurance that aims for consistent, repeatable, and reusable results:

1. Discovery and Inventory creates the application "universe" that must be tested by dynamically monitoring and fingerprinting the networks and systems that make up the application environment.
2. Fault Injection creates a set of aggressive tests tailored to examine the particular application.
3. Fault Detection monitors injectors as the application is tested to determine the attack results and provides them to the reporting engine.
4. Reporting provides the auditable, repeatable results necessary to institutionalize the process within an organization.
5. Automation allows the testing to be repeated and modified as necessary to meet the needs of new versions, new software, dynamic architectures and changing network environments.

In addition, Cenzic offers developers a software development kit (SDK) to create customized security tests based on their own specific environmental needs.

The Experts Take

Cenzic has created an evolutionary approach to security quality assurance. Whereas traditional approaches protected against incorrect results of software functions, it is now clear that enterprises must identify and resolve security vulnerabilities before a hacker does. Two challenges arise in this type of testing:

- How to address the full magnitude and scope of security testing
- How to institutionalize the process so that each application receives comprehensive testing within a real environment that can be consistent, repeatable and reusable

Cenzic's automation of fault-injection testing techniques satisfies these two requirements, resulting in an efficient and effective process that brings real security quality assurance to the enterprise.



About Cenzic

Cenzic provides an application security intelligence platform to continuously assess Cloud, Mobile and Web vulnerabilities. This helps brands of all sizes protect their reputation and manage security risk in the face of malicious attacks. Today, Cenzic secures more than half a million online applications and trillions of dollars of commerce for Fortune 1000 companies, all major security companies, government agencies, universities and SMBs.

Cenzic Products

Cenzic Enterprise	Cenzic Enterprise is a software solution that assesses the security of Cloud and Web applications and supports security risk management throughout the software development lifecycle. Cenzic Enterprise provides a company-wide view of security vulnerabilities and status to executives as well as customized views for other users from a Web-based dashboard.
Cenzic Desktop	Cenzic Desktop is a single-user version of Cenzic Enterprise. It is designed for the power user that wants to run their security assessments on Cloud and Web applications from a single system.
Cenzic Managed Cloud	With Cenzic Managed Cloud, Cenzic's security experts remotely perform full vulnerability testing on Cloud, Mobile and Web applications. From a Web-based dashboard, users can request assessments, view results, run reports, analyze trends and re-test applications to verify remediation efforts.
Cenzic Cloud	Cenzic Cloud allows users to test their own Cloud and Web applications for basic attacks and receive actionable results all within their own Web portal – no security experts needed.
Cenzic Hybrid – Software + Cloud	Cenzic Hybrid provides access to both Cenzic Enterprise software and Cenzic Managed Cloud services. Vulnerability testing can be done either by using the software on premise or by leveraging Cenzic's expert security services team.
Cenzic Mobile	Cenzic Mobile service is delivered as a managed service. Cenzic's security experts remotely perform full vulnerability testing on mobile applications. All testing is managed from a Web-based dashboard where users can request assessments, view results, run reports, analyze trends and re-test applications to verify remediation efforts.
Cenzic Services	Cenzic services and training help those responsible for Cloud, Mobile and Web application security to implement best practices and procedures to protect data from hacker attacks.