How to Choose an Application Security Testing Solution

ACCURACY

CLARITY

SIMPLICITY

1 Introduction

Many organizations are currently in the process of adopting, changing or enhancing their suite of application security testing solutions. It could be the first time application security is introduced into the SDLC, or there is a need to improve on existing solutions. It is at the point where all agree that application security should be part of the SDLC and ultimately used by developers, testers or DevOps.

To assist with this process, we have compiled a list of the most important factors to consider. The list is based on years of experience of consulting for organizations on how to build secure development programs, the processes we have gone through with our customers, and the lessons we learned together.

Three important principles lead the way – **Accuracy, Clarity, Simplicity.** These three principles need to coincide with your **Business Goals**.

We know that each organization is different, and with the listed concerns being relevant to all , the weight of each concern often varies between individual organizations.

Consider the following factors to find the solution that best fits your needs.

2 Factors for Choosing an Application Security Testing Solution

2.1 Accuracy

MINIMIZES FALSE NEGATIVES WHILE FINDING REAL AND RELEVANT VULNERABILITIES

False negatives are those vulnerabilities which actually exist in the application but were missed during testing.

There could be multiple reasons leading to false negatives: application coverage – meaning not testing the entire application, improper testing – actually testing but not being able to identify a vulnerability, or lack of testing – meaning there are vulnerabilities which were not even tested. Regardless of the cause, false negatives pose a risk.

DOESN'T CRY 'WOLF', ELIMINATES FALSE POSITIVES

Results that contain false positives require security-knowledgeable staff to separate real vulnerabilities from noise. This process is expensive and extremely time consuming, thus being detrimental especially in Agile environments where speed of delivery plays an important role.

Then there is the human factor - a strong advantage of automatic tools is relying less on the human factor. False positives reintroduce dependence on this factor as human analysis is needed for result triage.

On top of the above, false positives cause the loss of trust between security and development. Developers who are required to analyze and fix nonexistent issues rightfully lose trust in security people and tools that deliver these results. Trust between developers and security is crucial for delivering secure applications.

DELIVERS MAXIMAL CODE COVERAGE, NO EXCUSES

Modern applications feature many components and libraries from different sources – previously developed in-house components, open source libraries, common development frameworks and more. When attackers find a vulnerability they will go ahead and exploit it no matter where it is found. For this reason it is important that all code on all tiers is properly tested, including automatically generated code, on-the-fly generated code, components for which the source code is not available and more. Another important part of the discussion about

application coverage is the ability to test vulnerabilities that span multiple locations in the application as part of a single transaction.

WORKS QUICKLY, TO FIT AGILE AND/OR CONTINUOUS DEVELOPMENT

One of the goals behind using automatic application security testing solutions is to allow ongoing testing of applications as part of the development process. In Agile development and continuous integration environments, code changes are frequent. Under these conditions a security testing tool must be very efficient to meet tight delivery schedules. Application security testing cannot be a bottleneck, meaning application delivery cannot be delayed due to the need to perform security checks. On the other hand, with the modern cyber threat landscape, organizations cannot afford rolling out software which has not been fully tested.

2.2 Clarity

PROVIDES CLEAR RESULTS - WHAT YOU SEE IS WHAT YOU NEED TO FIX

Viewing and understanding the results by those who are not security experts are imperative to prioritize and remedy identified vulnerabilities. In order to truly integrate into the SDLC, non-security personnel should be able to operate the software routinely. These are most often developers, testers, or DevOps engineers. Results must be clear and easy to understand, while still providing all relevant technical and risk information in a simple yet informative manner.

HELPS FORM A VULNERABILITY MANAGEMENT AND REMEDIATION PLAN

Any security tool, after analyzing the web application, will be required to report on the vulnerabilities it detects, pinpoint the place in the source code where the vulnerability originated if possible, and suggest remediation steps to be taken for the particular vulnerability. In addition, provide full reports on the vulnerabilities found.

This is required since application security testing is part of risk management, and identified vulnerabilities are prioritized in terms of remediation urgency. A good application security testing solution will enable and assist this process by delivering results from which it is easy to see both the risk associated with the vulnerability as well as the remediation effort.

ALLOWS DIFFERENT STAKEHOLDERS TO UNDERSTAND RESULTS AND ASSOCIATED RISK

The chosen solution should enable visibility for different stakeholder levels and interests. Business owners, product managers, project managers are concerned with the big picture of risk level and action items, and the chosen solution should be able to provide this big picture, along with justification for the suggested remediation plan.

Security managers need to be able to see the overall security status in order to provide more resources to applications with more serious security issues, prioritize a global remediation plan, evaluate the current exposure of the organization, and more.

Team leaders and team members should be able to assess their security status independently, know their expectations, understand the associated risks and prioritize a remediation plan.

ENABLES A LEARNING PROCESS FOR MORE SECURE CODE

To minimize costs and deliver more secure applications, it is important that developers constantly learn and improve in order to deliver more secure code in the future. A good application security testing solution will be one that enables this learning in the least intrusive and time consuming way possible.

2.3 Simplicity

DOESN'T REENGINEER THE WHEEL – FITS INTO EXISTING DEVELOPMENT AND TESTING PROCESSES

Integrating into existing procedures of an organization without upsetting the SDLC process makes life easier and allows the SDLC to proceed smoothly. When a solution requires complex new procedures to be put into place it leads to two outcomes – either it does not really become part of the SDLC and is used as an extra layer in the post-development phases, or there is a lengthy integration process which adds many resources to the organization and takes years to implement. In both these two cases, the application is at risk due to security flaws.

DOESN'T REQUIRE EXTENSIVE TRAINING

If the tools are highly technical, and require a certain expertise in order to understand and implement the results of a test, the need for knowledgeable staff experienced in security to operate them becomes unavoidable

Regardless of whether those analyzing the results in this case are in-house experts or external consultants, the process will not be fast enough, especially not for Agile or continuous deployment

environments. If external security experts are hired, they of course cannot be part of the SDLC, meaning application security will be done post-development, again causing the same problem which the solution was chosen to fix.

EASY TO DEPLOY, CONFIGURE, MAINTAIN AND SCALE

The chosen application security testing solution should be one that is easy to deploy, configure and scale without requiring too much time and effort.

2.4 Business Goals

COMES WITHOUT HIDDEN COSTS

The cost of ownership of an application security testing solution consists of the cost of the product itself together with associated costs of integration and ongoing operation and maintenance.

The calculation of Total Cost of Ownership should take into account time spent identifying and fixing issues that are not real (invalid bugs are considered to take up to four times longer to address than real ones), result triage, configuring and optimizing the solution, time spent on running tests, delays and repetitions in existing processes and more.

DOESN'T TAKE A LIFETIME TO DEPLOY

In complex organizations solutions that require lengthy integration, configuration, and extensive changes to existing workflows can take years to implement properly. During this time applications may not go through the proper security process and the organization is exposed to threats, while effort and money are invested in securing applications at the post-development stages by using external experts or in-house security teams that work independently of R&D.

The chosen solution should require as little as possible change to existing processes, little configuration and integration, in order to allow short time to market.

3 Summary

In this paper we have outlined several factors that are important to consider when choosing an application security testing solution for your SDLC.

These factors fall into four major categories – **Accuracy, Clarity, Simplicity** and **Business Goals**. They are mentioned to allow each organization to consider individually which factors they consider as most important.

When approaching the task of choosing an application security testing solution go with the one which best addresses your individual needs, and best addresses the issues that are most important to you.

We believe that the best application security testing solution is the one that answers these factors as best as possible, without making compromises.

4 About Us

4.1 About Seeker

Seeker is the run-time code & data analysis application security testing solution for the software development life-cycle. By analyzing application behavior in response to simulated attacks, Seeker detects code vulnerabilities that pose a real threat. It assists in vulnerability management by generating exploits that demonstrate the risk to business critical data. Seeker is the perfect application security testing solution for the SDLC, it can be fully automated and works great in Agile and continuous integration environments.

4.2 About Quotium

Quotium Technologies (NYSE Euronext: QTE) was founded in 2004. Quotium envisioned the future of the internet as the largest business platform for all organizations and set on a mission to make the on-line business secure and robust. Today, Web-driven applications and on-line transactions are in the heart of the activities of all enterprises. Security, availability and performance of the web environments are vital factors for global economy.

We bring innovation and quality in software solutions that secure, monitor and improve performance of web applications.

In 2011 Quotium pioneered the run time analysis technology to detect and mitigate the technical and logical vulnerabilities in code. The

application security software solution Seeker has revolutionized the way in which web applications are being secured.

Founded and managed by experienced business people and recognized industry leaders Quotium is dedicated to providing quality in software, customer, service and support. Quotium considers that the best return of investment in the acquisition of a software solution is directly linked to the quality of the solution.

For more information www.quotium.com or info@quotium.com