Application Security for QA Managers – Pain or Gain

Dr. Ravi Kiran Raju Yerra
Arsin Corporation
ryerra@arsin.com

Abstract

It is often the case that developers and software vendors are not fully aware of application security vulnerabilities such as cross site scripting, injection flaws, cross site request forgery and etc. In many cases, these vulnerabilities can be prevented with training, more consistent and standardized software development practices, software acquisition protocols, and appropriate use of manual and/or automated security vulnerability test, manual and/or automated security code reviews.

Biography

Dr. Ravi Kiran Raju Yerra is an internationally known speaker with his doctorates degree in internet security management. Dr. Ravi Yerra has over fifteen years of real-world experience in delivering information security solutions around applications, cloud, virtualization, products, and database along with risk assessments and software testing across the globe in numerous industries and verticals. Since 1995, Dr. Yerra has been involved in multiple security projects and played a vital role in establishing various private and government information security initiatives.

1. Introduction

Historically, application developers and Quality Assurance (QA) teams have not focused extensively on security. IT management typically asks developers to achieve two goals - build innovative features and see that the project is completed on time. On the other hand, QA teams ensure that the application functions as intended and that it can scale effectively and perform under load (functional and performance testing). Unfortunately, during the development and QA phases management rarely ask that any real form of security testing take place. In fact, security testing is often viewed as an initiative that works in opposition to the aforementioned goals, as it can extend the already lengthy development and testing phases. Far too many organizations treat security as an afterthought as opposed to being integrated throughout the development process. In addition, most developers, QA professionals and QA managers do not consider themselves responsible for application security - assuming that security will be managed while the application is live.

1.1 Here are several myths of QA Managers:

Below are the most common security myths behind why QA managers don't always include security testing in the QA cycle in most modern and mature enterprises.

- Security Testing is already being done
- Security Testing is too hard
- Security Testing is not viewed as QA's job
- QA can't effectively perform security testing [they're not experts]
- QA analysts don't understand Security Testing

In this whitepaper we will discuss how QA teams can bridge the testing and application security gap in the software development lifecycle by debunking the several myths, actual reality and how a QA manager will benefit implementing Application Security.

2. The Problem - Application Security is a Quality issue

The majority of vulnerabilities in web applications reside in the custom business logic of the application itself. Compensating controls provided by external products are temporary solutions which seek to hide the vulnerability. It is typically only a matter of time before an attacker identifies an alternate entry point or is able to encode an attack in such a manner that a signature-based technology is unable to detect the attack packet. Only by implementing security during the Software Development Lifecycle (SDLC) is it possible to fully protect the application. This is the reason that developers, QA teams, and the QA management must share in the responsibility of implementing security at SDLC. Auditing a web application, either prior to or following release into production, simply is not sufficient to identify all vulnerabilities adequately. Application security is most effective as an iterative process that is applied consistently throughout the development lifecycle. The better way is providing security from the initial stage of software development i.e. security from the requirement stage to the deployment of any software product. This mainly involves the adaptations or augmentations of existing SDLC activities, practices, and checkpoints, and in a few instances it may also entail the addition of new activities and practices. Secure software is not easily achieved and the actual scenario is that investments in software development process improvement do not assure software that is resistant from attacks or modern security vulnerabilities. It can be demonstrated that changes to the software development process can help to minimize the number of vulnerabilities in new or developing software. Some of the risks posed by an insecure application are financial in nature and the cost of a single security breach can be significant. It is important to remember that the

total outcome can be difficult to fully measure due to the intangible nature of many costs. While the cost of labor to remediate the damage would be an obvious factor, damage to a corporate reputation caused by a defaced website or an unavailable application due to a distributed denial of service attack can be much more difficult to measure. Regulatory risk is another substantial and growing concern. Failure to adhere to a growing list of government and industry regulations can lead to fines, discontinuation of services, and even civil and criminal penalties. The common compliances all emphasize the need for security, especially at the application level.

2.1 Application Security - The Unfamiliar Limit for QA

Unfortunately, the availability of QA managers who have application-security testing knowledge is extremely limited. Existing tools such as static code analyzers or black box testing tools are complex and require security and vulnerability expertise that is rarely available within QA organizations. Businesses need a simplified, cost-effective means to incorporate security expertise into QA processes without impacting production schedules or resources.

In order to meet these rigorous requirements, QA managers must believe that security is important, security ensures quality, security is testable, and security makes a difference. It's crucial for QA managers to adopt security testing in a step by step process, including application security as a primary focus and eliminating any common excuses such as:

- We are behind a firewall.
- We use Secure Socket Layer (SSL) for authentication.
- We don't generate session ID.
- We don't use cookies.
- We use secure file transfer protocol (SFTP) to transfer data to a third party.

3. The Solution – Connecting Your QA Department

QA teams have some interesting ideas when it comes to answering this question: "Are you doing any application security testing currently? and depending on who you ask it's possible you will receive a variety of different answers. "Testing for security" can mean many things to many different people, and we wanted to take a moment to debunk some of the myths that spread over the last 3-5 years.

There are three ways that your Quality Assurance department may become involved with Web application security testing:

- Your company's Web security experts request that application security testing be completed by the QA group to ensure that all fixes have been implemented and no security holes exist prior to releasing the product to production.
- Your compliance officer, facing concerns about SOX, HIPAA, PCI, and so on, requests that further application security testing be performed during the QA process.
- Your QA department themselves request involvement with testing for Web application security, because an application that has security holes in it is not going to be perceived as high-quality by users.

No matter how the department gets involved, certain steps will need to be taken to establish the application security testing process. It will need to be determined whether there will be specific, dedicated staff members who will be performing Web application security testing, or whether the task will be dispersed throughout your entire QA group. In addition, the timing of Web application security testing during the Quality Assurance process will need to be managed. Ideally, application security testing will be performed as early as possible, so that developers can fix any security issues in a timely manner, without compromising the project's schedule. Finally, the right software for application security testing will need to be selected and implemented. Let us further explore the myths in detail

vs. how security is taking shape in the real world and learn the techniques how to integrate Application Security and enable QA teams to handle security issues. The below table debunks the previously talked about myths and guides QA managers about how to plan and integrate application security and benefit from it.

Myth	Fact	Tasks for the QA Teams
Myth 1 - Security Testing is already being done		
We are checking for the existence of the password requirement, or making sure pages aren't accessible without authentication doesn't actually mean you're doing security testing.	▪ Most security testing that I've observed QA organizations performing is centered on session management, and account authentication (and in some rare cases authorization). ▪ Very few QA organizations actually check for the types of issues that come from manipulating application logic, sending in garbage or malicious inputs, or attempting to break in to the application with malicious intent. ▪ Over the past five years we've seen an exceptional growth in the sophistication of application security measures. Unfortunately, that has been paralleled by an increase in the ability to penetrate applications. ▪ With recent attacks more targeted at Web applications, it has become increasingly important to address application vulnerabilities overlooked in the past. Though some of the companies are maturing and addressing application security issues, a vast majority either underemphasize the importance of those measures or labor under misconceptions.	▪ Identify a core QA Analysts and conduct a workshop on importance of application security. o Why Security & different trends? o Why Application Security? o What is vulnerability, threat, risk? o What is business risk? o Tools and techniques to identify and fix Security bugs. ▪ Setup readily available free to use IBM's Testfire and WebGoat, test environments for practicing application security skills ▪ Adhere to very well known web application security standards such as OWASP, WASC Threat Classification V2.0 in-order Increase security testing coverage. ▪ Apply a risk based approach ▪ Test with malicious intent.
Myth 2 - Security Testing is too hard [complicated]		
Security testing can be quite difficult, if there is no expressive and well-established framework or process utilizing tools and documented, repeatable methodology. *QA teams complain about security testing is that some security teams try to force a whole new suite of testing tools and methodologies on the QA*	▪ Any testing would be hard unless there is a pre-defined process and methodologies are available and practiced. ▪ In order to be successful it's critical to first understand the existing processes, tools and methodologies the QA teams use today and then adapt security testing protocols, tools and methods to fit within those existing processes and be	▪ Yes, security testing is too hard and complicated when there is no o Defined test procedures, guidelines, checklists, test cases & etc. o Refer OWASP Testing Guide for detailed methodology. o Train team on new set of tools that aid automating majority of application security testing ▪ Security needs to be tightly

Myth	Fact	Tasks for the QA Teams
analysts – that's a sure-fire way to meet opposition.	minimally invasive or disruptive. ▪ This issue is further intensified by the lack of proper process or guidelines. As a result, a vast majority of development managers rely heavily on pre-programmed tools such as IBM AppScan or WebInspect from HP to identify the vulnerabilities in their applications. Based on my experience in this field, I have found that the problem is two-fold: people rely on out-of-date information for their decision-making and they lack initiative for staying current in this rapidly developing field. ▪ With recent attacks targeting applications, firewalls and intrusion detection systems (IDS) are not sufficient.	integrated into the application, and there should be a proper secure design methodology. Managers, architects and developers need to increase their awareness and receive differential training. For example, developers should concentrate more on secure coding of their applications, whereas architects ought to focus on designing secure applications. ▪ The architect should look beyond using just SSL. Some points to consider while designing the module: o Is SSL being used only for login/authentication or for the entire site? o If the SSL is being used only for authentication, is it also used for the change password or forgot password page? o Does a session get created? If yes, does it use a secure cookie to store the session ID? o The document should also contain encryption detail such as key length, encryption algorithm, hashing algorithm, etc. Not to mention the fact that it should comply with the company's standards. o Where are the user ID and password stored? Are they stored in the database? Are they stored in clear text or encrypted? If encrypted, then is it one-way encryption? o If a user forgets the password, does a system send his original password in an e-mail or does the system generate a new unique one-time password and send it via e-mail? Is the user forced to change the password when he logs in for the first time? Is the e-mail that contains the password secure?

Myth	Fact	Tasks for the QA Teams
Myth 3 - Security Testing is not QA's job [that's security's job!]		
QA gets the application much earlier than security teams, for testing and finding defects earlier is always a bonus.	▪ Security is everyone's job, more explicitly – the QA organization is responsible for software quality – of which security is a component. Security is the overall pillar or quality to see does it function? Does it perform? Is it secure? QA teams own almost exclusively.	▪ Engage QA teams to provide training o Firstly a change in mindset o A bit of background on Web Applications o Basic things to integrate into your QA Process (Vulnerability Discovery) o A bit of validation ▪ Further Reading for moving away from basic to deeper understanding (Open Web Application Security Project (OWASP) & Web Application Security Consortium (WASC)) ▪ Confidently your team will be now prepared with some basic information on how to identify basic vulnerabilities within your web application as well as places you can search for more information on web application security while gaining more peace of mind. ▪
Myth 4 - QA can't be effective at security testing [they're not experts]		
A valid argument… if you completely disregard the fact that the application security testing tools market is maturing faster than our national debt. *QA testers don't instinctively think like hackers and work from functional specifications*	▪ Hacker look for landmarks a little different - but with the same principles as common is applicable to functional or manual testing. ▪ Shift your mind to a hacker touring your site or application for application security issues. ▪ Whereas if you look at they have quite opposite mentality difference: ⇒ QA asks - How does it perform? ⇒ Hacker asks - How can I break it? o To summarize one need his change his/her mindset - Think like a Hacker. ▪ Attackers look for "exploitable functionality whereas functional testers understand "use cases".	▪ The secret here is that QA analysts don't have to be security "hacking" experts if they have the right tools in place. QA analysts don't have to be expert hackers; in fact, organizations prefer them not to be. o Basic things to integrate into your QA Process (Vulnerability Discovery) o QA analysts just need to be able to use tools and follow process effectively to identify basic application security flaws. o Gain training from consulting teams just need to make sure that the tools and processes they set up for their QA analyst counterparts are templated and uncomplicated ▪ Quick hints for QA teams for finding hacker landmarks: o Look for changes in privilege or trust o Look for application interaction

Myth	Fact	Tasks for the QA Teams
		points o Look for opportunistic data validation o Follow the money (Commerce)
Myth 5 - QA analysts don't understand Security Testing		
	▪ QA analysts can provide needed support and enable security testing much earlier in the development lifecycle, but not without the proper training. ▪ It's the lack of exposure that's currently hindering adoption and fueling this final myth. o In reality, many QA analysts start to become very interested in hacking and security testing once they've been given the opportunity to learn and experience.	o Provide a foundation for understanding web application security testing by identifying potentially vulnerable targets to basic attack strategies and advanced tools based hacking techniques. o Guide them to examine the most common web application defects o Understand the concept of "negative testing" o Organize application security workshops on OWASP, WASC threats and vulnerabilities.

4. The Benefit - GAIN for QA Managers

Within the application development lifecycle, the QA Manager is responsible for both the resources and tools used to ensure that defects are identified early and do not materialize once the applications go into production be it Functional, Performance and Security issue. The cost of resolving a defect in production can be more expensive than capturing the defect early in the development lifecycle. The QA Manager will lead a team that will address the three pillars of quality:

- Does the application provide the functionality needed to meet business requirements?
- Does the application function with sufficient performance to meet business requirements?
- Finally and most importantly does the application deliver adequate security to meet business requirements?

Hopefully QA teams are now armed with a new approach to design Application Security Testing Practices within the QA group by integrating security across all stages of the software delivery lifecycle. You can search for more information on web application security while gaining more peace of mind. Here are few lists of direct benefits for QA Management:

- ✓ Equip QA teams to handle advanced application security testing skills
- ✓ Operate a modernize QA team that address the three pillars of quality i.e. Functionality, Performance and most importantly Security
- ✓ Lay down a roadmap to advance QA tester skills on application security
- ✓ Providing opportunity to enhance employee skills
- ✓ Increase test coverage by focusing on most important pillar of quality i.e. Security
- ✓ Bridge the testing and application security gap in SDLC by involving QA to identify and developers to fix security problems early in SDLC.
- ✓ Delivering adequate application security to meet business requirements
- ✓ Show value to customer by saving cost for resolving a defect in production which can be more expensive than capturing the defect early in the development lifecycle
- ✓ Introduce security from early stages of development and achieve software security

- ✓ Deliver high quality secure software and gain customer mileage

5. Conclusion:

Changing the paradigm

In order to break this cycle, we must change the way that we fundamentally approach application security. Gone are the days when anyone involved in application development can say “Security is not my responsibility.” Security is everyone’s responsibility as it has severe impact on the business if not taken seriously. We must integrate security throughout the SDLC, not just hastily add it to the end. This integration will only occur if we involve developers, QA teams, and the QA management in security. Making such a fundamental shift will not happen overnight, but it is essential if we are to stem the tide of applications riddled with security vulnerabilities which offer multiple attack vectors and leave enterprises wide open to attack. However, if security is not an integral part of your company’s development process for custom Web-based applications, your Web interfaces may be vulnerable. Ideally, security concerns should be addressed during development phase. Finding and fixing security defects early in the SDLC is up to 100x less expensive than doing so in production according to estimates by the IBM Systems Sciences Institute 2009 Cost per defect metrics.

Many organizations today are realizing the importance of bringing Security Testing under their QA umbrella and are doing so in a phased approach that includes enablement, test plan creation, and automation.

References:

http://www.testfire.com/ - Initially Watchfire Inc., published and hosted Testfire website having functionality of vulnerable demo banking services to detect web application vulnerabilities and website defects. Later Watchfire Inc. was acquired by IBM and all of its product suites operate under IBM Rational portfolio.

www.owasp.org/index.php/Category:OWASP_WebGoat_Project - WebGoat is a deliberately insecure J2EE web application maintained by OWASP designed to teach web application security lessons.

http://www.owasp.org - The Open Web Application Security Project (OWASP) is a 501c3 not-for-profit worldwide charitable organization focused on improving the security of application software.

http://projects.webappsec.org/w/page/13246978/Threat-Classification - The members of the Web Application Security Consortium have created this project to develop and promote industry standard terminology for describing these issues. Application developers, security professionals, software vendors, and compliance auditors will have the ability to access a consistent language and definitions for web security related issues.

http://h71028.www7.hp.com/ERC/cache/568162-0-0-0-121.html - This article is from Hewlett Packard that throws discusses how to incorporate web application security in to Quality Assurance process.

http://searchsoftwarequality.techtarget.com/tip/Application-security-Past-myths-present-excuses - Anurag Agarwal is the author of “Application Security: Past myths, present excuses” where he discusses the application security myths, excuses and solutions.

http://h30499.www3.hp.com/t5/Following-the-White-Rabbit-A/5-QA-Myths-Debunked-Why-QA-Doesn-t-Do-Security-Testing/ba-p/2407792 - Rafal Los is the author of this beautiful article that’s sheds light and clarified most common and important QA myths in very much detail.

https://www.owasp.org/index.php/OWASP_Testing_Guide_v3_Table_of_Contents - "OWASP Testing Guide", Version 3.0 - Released at the OWASP Summit 08.