SearchSecurity.com

E-Guide

Application Security Assessment Options

Learn what to do when you have a huge portfolio of potentially insecure applications, limited resources and an overwhelming sense of urgency.

SearchSecurity.com

E-Guide

Application Security Assessment Options

Table of Contents

Vulnerability test methods for application security assessments

By Rohit Sethi and Nish Bhalla, Contributors

This scenario may be familiar: Your organization has hundreds or even thousands of applications, but few have received an adequate security assessment, despite a mandate to protect the enterprise against application threats.

Many security managers are thrust into the uncomfortable position of dealing with a huge portfolio of potentially insecure applications, limited resources and an overwhelming sense of urgency. Security managers should ensure applications undergo security assessments, as applications have quickly become a favorite vector of malicious attackers seeking to disrupt day-to-day business activities or infiltrate corporate defenses to steal sensitive data.

In this tip, we'll add some clarity to the enterprise application security assessment process by outlining the techniques used to review applications and comparing and contrasting strategic paradigms for application assessments.

Technical application security assessment options

Those who are new to application security may be overwhelmed by the sheer number of assessment options, each with its own aficionados touting the merits of their favoured assessment type:

- Runtime vulnerability assessment -- Runtime assessments come in three varieties: automated, manual and combined. Automated assessments are generally faster and broader than manual assessments, but often miss obscure vulnerabilities and cannot discover business logic flaws. Most mature application security shops lean toward a combined approach.

- Source-code review -- Source-code review allows assessors to find various vulnerabilities, but requires deep language and security expertise, and often takes longer than runtime assessments. Like runtime vulnerability assessments, source-code reviews can be automated, manual or combined: all with pros and cons analogous to their runtime assessment counterparts.
- Threat-modeling techniques -- These assess pertinent, theoretical application threats from a design perspective. Often threat modeling precedes source-code review and/or runtime vulnerability assessments.

Selecting the right mix of assessment types can be difficult. Many companies face this problem, and there are a number of approaches to finding the right application assessment process:

1. **The Big Bang Approach:** Perhaps the most traditional method is to focus testing resources on the applications with the most public exposure, such as the most widely used Internet-facing Web applications. Once those apps are identified, comprehensive automated and manual runtime vulnerability assessments can be performed. Unfortunately, this approach ignores other critical albeit lower-profile applications, such as extranet apps, internal accounting applications and critical Intranet sites.

2. It's important to remember that all Internet-facing applications are subject to external attack, regardless of how popular they are. Moreover, the rising danger posed by insider threats and client-side vulnerabilities makes ignoring internal applications a significant risk. In addition, many experts in the application security community believe that blackbox testing alone is not as effective as combining source-code review with black/gray box assessments.

3. **The Steam Roller Approach:** Often when organizations realize the risks posed by The Big Bang Approach, they decide to broaden their comprehensive testing initiative to more applications during a longer period of time. We've seen companies hire teams of penetration testers to test every Web application in the enterprise. As you can imagine, only a handful of organizations can afford this approach. More

importantly, the applications that aren't tested right away may be exposed to attacks until both testing and remediation are completed; this can often take a year or more!

4. **The Application Triaging Approach:** A preferred approach is to rank application risk using several factors, including a variety of assessment techniques based on an application's risk profile. To start, look at the following dimensions of each application:
    - Purpose of the application: What is the application used for? How many people use it? A telephone directory application doesn't have the same risk profile as an accounting application.
    - Data risk: Are confidentiality or integrity requirements tied to the application? Does the application or its servers need 99.999% availability? Is the application affected by any compliance drivers, such as PCI DSS, HIPAA, etc.?
    - Architecture and design: Is the application a Web application, Web service, client/server, mainframe, mid-tier, desktop or something else? Is it Internet or intranet facing? What programming language and framework was it developed in? Does the application use any known high-risk components such as Ajax or PHP? Approximately how large is the application (in lines of source code)?
    - Existing security features: What security features are already known to exist in the application? For example, how does the application perform authentication, authorization, input validation, etc.?

    With this method, it's important to build guidelines that assign numeric risk values for each of these factors. For example, "Add 25 points for Internet-facing applications," "Subtract 5 points for applications that don't share data or interfaces with any other applications," etc. The end result should be a number that allows you to rank applications against one another. Remember that profiling applications is often time consuming and hard to perfect, so rather than forcing yourself to get all data for all applications, try to stick to a limit for how much time to spend gathering info on each app. Your scoring methodology should be tolerant of imperfect information and should be able to rank applications against each other even if you have a deeper understanding of one versus another. Don't be too rigid about the scoring system -- if a security expert sees an application as particularly high risk, but the scoring system does not backup his intuition, side with the security expert.

Applications in the high-risk bucket should undergo threat modeling, followed by manual and automated runtime vulnerability testing and source-code review. Moderate-risk applications should be subject to automated runtime vulnerability testing and source-code review with manual verification. Low-risk applications may simply need to undergo runtime vulnerability testing and, time permitting, manual verification. If the results of testing an application from the lower buckets are particularly negative, then the application should undergo more comprehensive testing.

5. **The Health Check Approach:** An alternative to normal triaging is to perform short, one-day combined manual and automated runtime assessments on all applications. In this scenario, assessors limit automated scanning to a small number of test cases, substantially reducing scanning time (to close to an hour typically). To do this, it's important to reduce the total number of variants performed for each attack type, such as 10 cross-site scripting, 10 SQL injection, etc. The manual component entails reviewing and validating the scan results and spending additional time to perform a limited set of manual tests. Based on the results, an experienced assessor can decide whether to prioritize an application for additional assessment time or to defer additional testing until after reviewing higher-risk applications.

6. **The Unauthenticated Health Check Approach:** An alternative to the Health Check Approach is to perform short 1-2 day automated runtime vulnerability assessment on all applications in a short period of time without authentication credentials. This approach mirrors the attack methods of script kiddies and bots, such as the infamous ASP SQL injection bot that continues to plague Web applications. Consider this method in cases when it would be too difficult or time-consuming to get authentication credentials. However, be cognizant that in many applications authenticated users pose the most significant risks. Unauthenticated scans miss all these attacks.

So what's the best approach? Aligning the assessment with business risk allows for meaningful prioritization of time and money. A hybrid of approaches is ideal: Immediately identify and comprehensively test a small set of the highest risk applications (e.g. your company.com website). In parallel, start the application triaging process to determine what

gets tested next. If the resources are available, begin the unauthenticated health check assessments while you're triaging. This process allows you to benefit from the broad analysis of profiling along with the objective results of a quick scan. Follow up the rest of the process like a normal Risk Triaging Approach: Start with the highest risk apps and work toward the lowest.

Assessments, of course, are only one part of the entire application security equation. The next important step involves remediation. Luckily, the triaging approach lays the ground work for prioritizing remediation: Start with the highest risk vulnerabilities in the highest risk apps, and move down from there. A good application security team will also be able to identify root causes to system findings and suggest remediation steps in the software development lifecycle to make its applications more secure from the ground up.

Regardless of the application security assessment approach you choose, remember that any of these approaches are better than turning a blind eye to the many risks posed by insecure enterprise applications.

About the authors:
Security Compass is an information security consulting and training company that specializes in secure software development. With its in-depth knowledge of information security and software engineering along with unmatched commitment to professionalism and training quality, Security Compass is repeatedly engaged by several of the world's most security conscious organizations to help build software trust from the ground up.

Nish Bhalla
Nish Bhalla, the Founder of Security Compass, is a specialist in Application and Network Assessments.

Mr. Bhalla has coauthored and contributed to many books including "Buffer Overflow Attacks: Detect, Exploit & Prevent" "HackNotes: Network Security", "Writing Security Tools and Exploits" and "Hacking Exposed: Web Applications, 2nd Edition".

Nish is a frequent speaker on emerging security issues and has spoken at many reputed Security Conferences including RSA, Blackhat, RECon, HackInTheBox, ShmooCon and many others.

Rohit Sethi

Rohit Sethi, Director of Professional Services, Security Compass, is a specialist in threat modeling, application security reviews, and building security controls into the software development life cycle (SDLC). Mr. Sethi is a frequent guest speaker and instructor at several national conferences. He has written articles for Security Focus and the Web Application Security Consortium (WASC), and has been quoted as an expert in application security for ITWorldCanada and Computer World.

At Security Compass, Rohit teaches hundreds of students various topics on Web application security in cities across North America. He has also managed and performed extensive threat analysis, source code reviews, and penetration testing for clients in financial services, utilities, telecommunications and healthcare. He is often consulted for his dual expertise in information security and software engineering.

Security Innovation®
THE SOFTWARE SECURITY COMPANY

Reducing IT Risk
By Securing Your Applications

Application Security TRAINING | CONSULTING | E-LEARNING

Security Innovation focuses on the most difficult problems of IT security, and the root cause of most data breaches – those at the application layer.

The company's world-class application security eLearning and consulting solutions provide IT and Development Teams with the knowledge and programs needed to build internal expertise, create secure software applications and reduce overall information risk.

TRY our application security eLearning FREE at elearning.securityinnovation.com

getsecure@securityinnovation.com

www.securityinnovation.com

+1.978.694.1008

Resources from Security Innovation

Security Innovation®
THE SOFTWARE SECURITY COMPANY

Ensure your applications are in compliance. eLearning for secure application design, coding, & testing

Out-of-the-Box Secure Coding Standards for PCI-DSS, NIST, OWASP, ISO, etc.

High-Performance, Standards-Based Crypto - ideal for constrained devices & high-transaction volume

About Security Innovation

Security Innovation (SI) focuses on the most difficult problems of IT security, and the root cause of most data breaches – those at the application layer. Application security is of increasing importance in compliance regulations and SI helps organizations not only meet these standards, but reduce overall information risk. The company's world-class eLearning system, coupled with expert consulting, enables organizations to roll out applications with confidence.

SI equips GRC teams with the knowledge and programs needed to risk-rank applications, determine priority assessments, and execute assessments efficiently – allowing clients to mitigate risk for a single application or across an entire enterprise portfolio.

The company's flagship products include TeamProfessor, the industry's largest library of application security eLearning titles, and TeamMentor, web-based secure coding standards