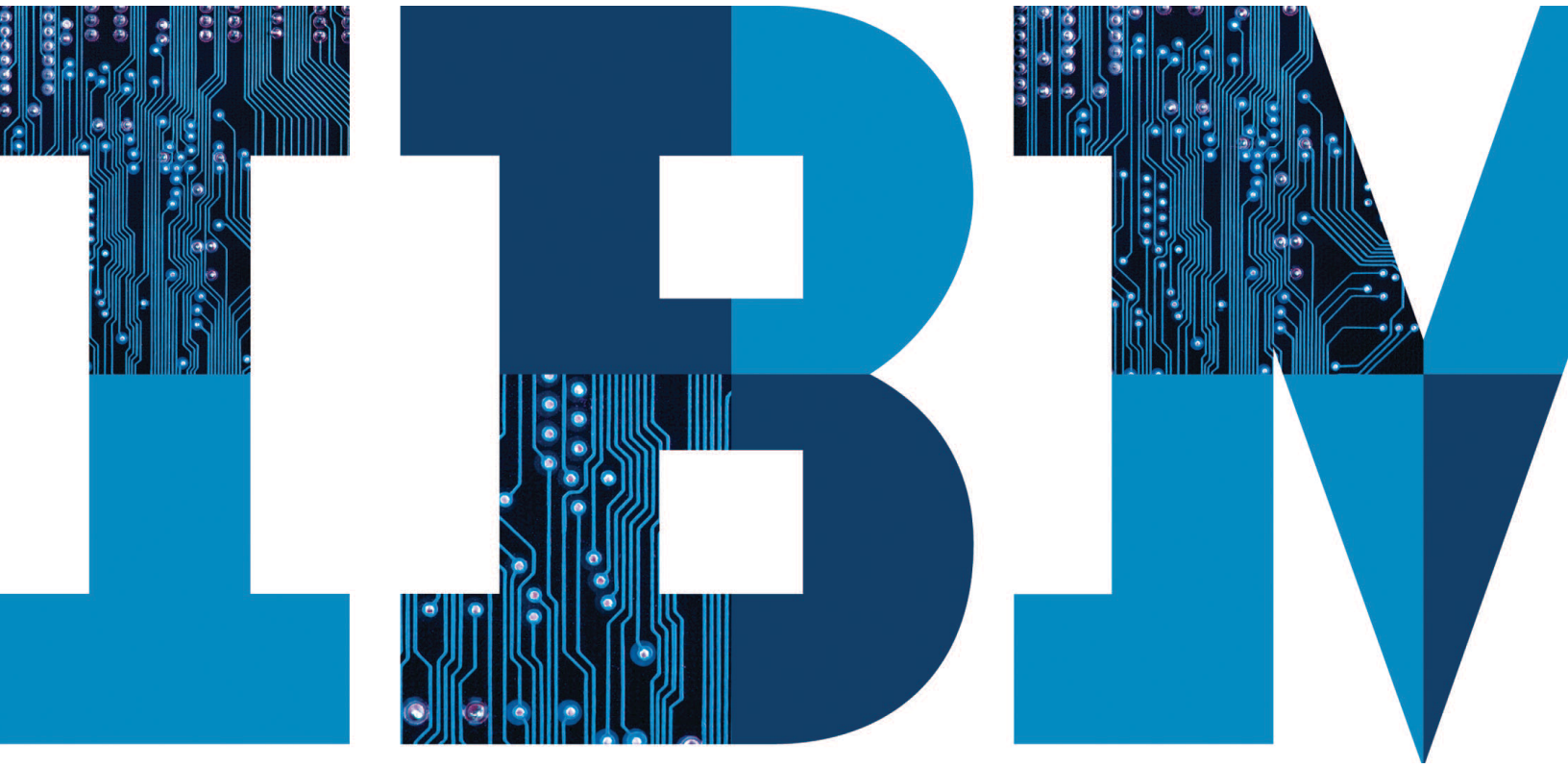
IBM Security
Thought Leadership White Paper

July 2015

Five Steps to Achieve Risk-Based Application Security Management

Make application security a strategically managed discipline



IBM

Table of Contents

Software applications support the most sensitive and strategically important business processes of most enterprises. Yet application security is one of the most neglected fields of cybersecurity.

IT and business management typically have no visibility into the overall state of application security. Activities for assessing, prioritizing and remediating application vulnerabilities are ad hoc, fragmented and carried out at low levels in the IT security organization. Quality assurance and software development groups lack the knowledge and incentives to address critical vulnerabilities early in application development lifecycles, where testing and fixing vulnerabilities are most cost-effective.

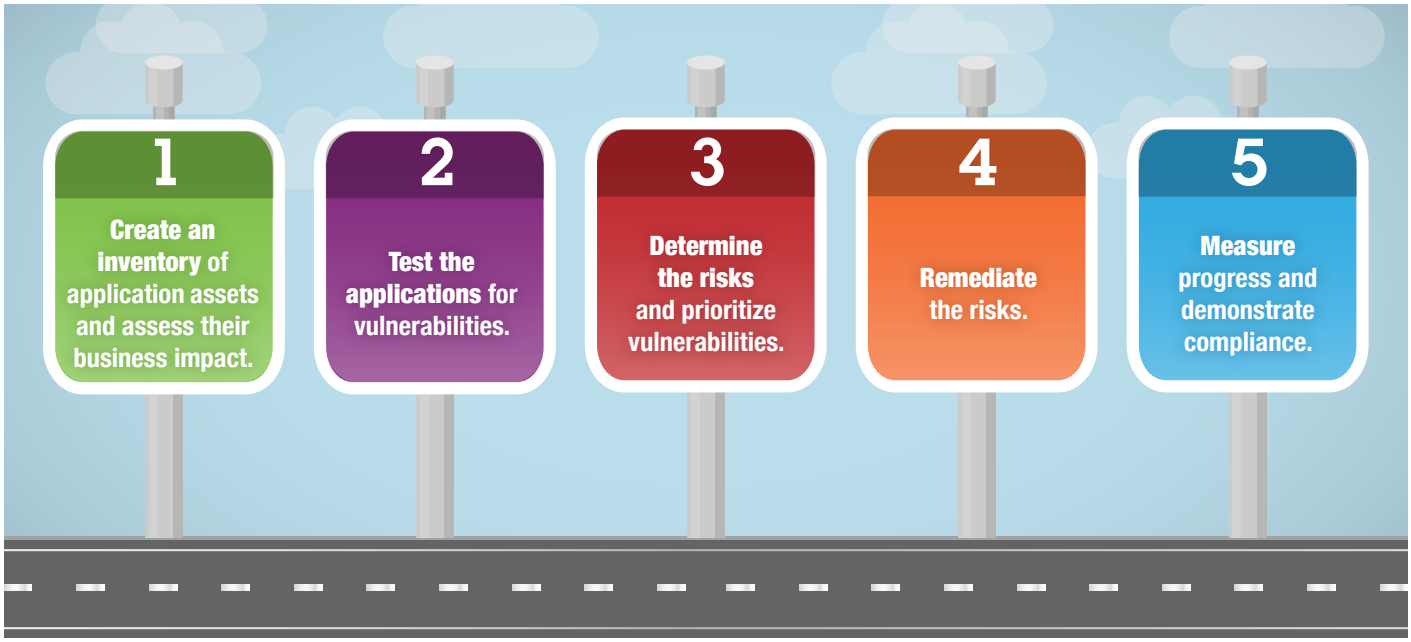
But you can make application security a strategically managed discipline by following a five-step process:

This process enables managers to:

- Obtain visibility into the state of application security across the enterprise.
- Set priorities for testing and remediation that align with business risks and strategies.
- Allocate resources to help prevent the most likely and most harmful data breaches.
- Measure progress toward application security goals.
- Continuously monitor the organization's overall risk posture.

This e-guide discusses potential obstacles to managing application security effectively and describes five steps for implementing risk-based application security management.

Whether you manage a handful of applications or thousands, this guide will help you take a disciplined, strategic approach to finding and mitigating risks.



Chapter 1: Why You Need a Risk-Based Approach to Application Security Management

Why is Application Security So Difficult?

Software applications are part of the critical infrastructure of most enterprises. They support the most strategic business processes, manage interaction with the most important customers and business partners, manage the most sensitive customer and employee data, and store most of the organization's intellectual property.

But with great power comes great responsibility. Internal development teams must address all aspects of IT security, including authentication, access control, identity management, data protection, session control, data encryption, and mobile security. And there are many reasons why application security is extremely difficult to achieve.

Technical Challenges

Software developers are not all security experts. Just one mistake or omission can result in a vulnerability that could be exploited by an attacker, and even the best-trained developer can focus on only a few security issues at a time. (See the callout box on page 6: SQL Injection Attacks and the OWASP Top 10 Security Flaws.)

Aggressive Adversaries

A growing number of cybercriminals, "hacktivists" and state-sponsored hackers have realized that attacking application vulnerabilities is often the fastest and easiest way to launch multi-million dollar data breaches and steal intellectual property. Today, attacks on applications represent a very significant portion of overall security risk, as illustrated by the statistics in Figure 1.

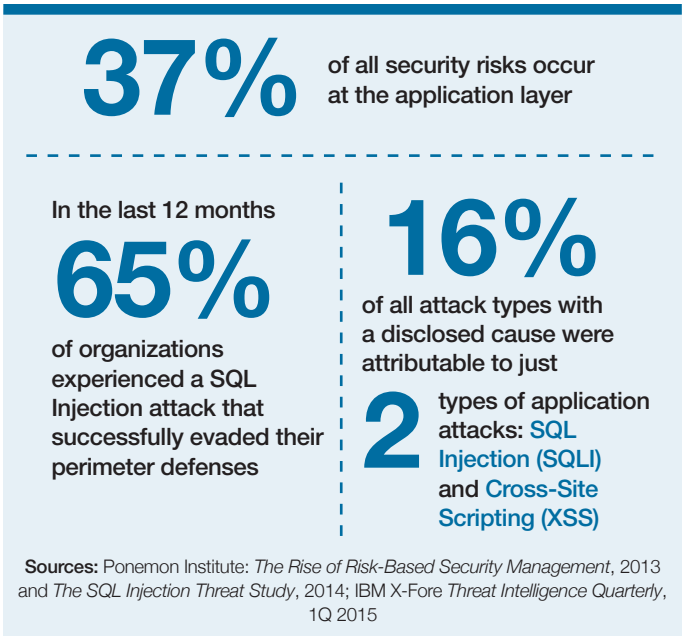


Figure 1: Application-layer attacks are commonplace, and often successful

Organizational Factors

In most organizations, incentive systems work against a strong emphasis on security. Developers and quality assurance (QA) engineers are usually rewarded based on their ability to deliver features quickly, not for discovering and eliminating security flaws.

In addition, enterprises typically rely on a handful of cybersecurity experts to drive application security across dozens of development teams and hundreds or thousands of applications. Often the security team members work in divisional silos, with few tools (sometimes only spreadsheets and bug tracking packages) to assess security levels, manage

testing for vulnerabilities, and track remediation efforts. As a result of this ad hoc approach to application security management:

- Nobody has visibility into the state of application security across the enterprise.
- There is no mechanism to set enterprise-wide priorities, or to align security activities with business risks and strategies.
- Resources are frozen within silos, and cannot be reallocated to the areas that need them the most.
- It is impossible to measure overall progress toward application security goals.

A Risk-Based Approach to Application Security Management

Leading-edge software development teams have succeeded in implementing a strategic, risk-based approach to application security management.

The starting point for this approach is to create a consolidated view of all applications across the enterprise. The consolidated view creates an inventory of all applications used in the enterprise. For each application in the inventory it also captures:

- A descriptive profile.
- An assessment of the criticality of the application and the potential impact on the business in the event of a breach.
- A list of vulnerabilities within the application, rated by severity.
- An overall "risk score" based on latent and potential risk.

When this consolidated view is available, management can turn application security from a collection of ad hoc processes carried out at the local level into a strategically managed discipline that optimizes resources and learning across the enterprise. Supported by the right processes and tools, management can:

- Obtain visibility into the state of application security in each business unit and across the enterprise.
- Set priorities for testing and remediation that align with enterprise-wide business risks and strategies.
- Allocate resources to protect the most important information assets and help prevent the most likely and most harmful data breaches.
- Measure global progress toward application security goals.

Better information can also improve the collaboration between development, QA and security personnel. Developers and test engineers are typically immune to exhortations and vague requests to improve security for the good of the organization. On the other hand, they are usually very receptive to facts, such as the number of vulnerabilities in their applications, and the performance of their team relative to other teams. Objective criteria and metrics make it much easier for development, QA and security teams to work collaboratively to develop test plans, prioritize backlogs and fix security flaws.

SQL Injection Attacks and the OWASP Top 10 Security Flaws

The SQL injection (SQLI) attack is the poster child of web application threats. To execute one, the attacker finds a web form and enters a SQL language query into one of its fields. If the developer who created the form did not take the proper precautions, the database will execute the query. The query might return confidential information to the attacker, or destroy data in the database.

For example, injecting the query: *SELECT UserId, Name, Password FROM Users WHERE UserId = 105 or 1=1* could cause the application to reveal the user name and password of every authorized user. Adding the phrase *DROP TABLE Customers* to the end of the query would cause the application to delete the entire Customers table in the database.

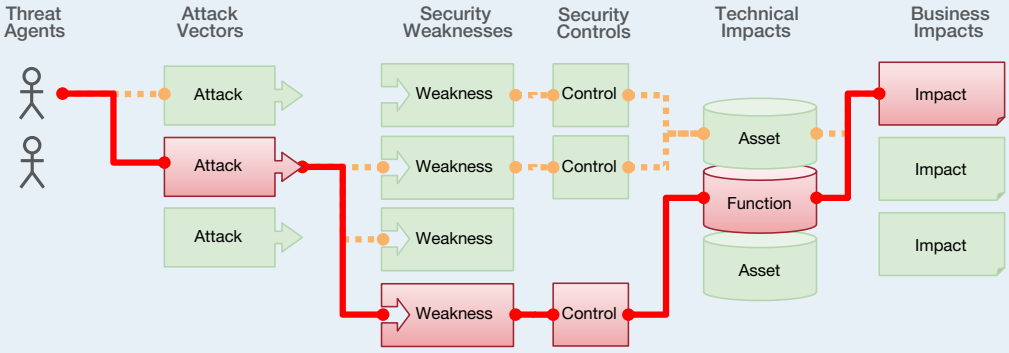
There are several techniques that can be used to prevent such queries from executing. However, developers need to protect against not only SQL Injection attacks, but also many other types of injection attacks, including LDAP, NoSQL, OS, SMTP Header injection attacks and dozens of other threat types. No developer, and for that matter no QA engineer or security analyst, can be expected to master all of these attacks and the countermeasures to prevent them.

The Open Web Application Security Project (OWASP) provides a very useful list of the Top 10 web application security flaws, including an excellent summary of the nature, severity and impact of each.

Videos illustrating each of the OWASP Top 10 are available from the IBM Security Intelligence blog post: The 10 Most Common Application Attacks in Action.

What Are Application Security Risks?

Attackers can potentially use many different paths through your application to do harm to your business or organization. Each of these paths represents a risk that may, or may not, be serious enough to merit attention.



Sometimes, these paths are trivial to find and exploit and sometimes they are extremely difficult. Similarly, the harm that is caused may be of no consequence, or it may put you out of business. To determine the risk to your organization, you can evaluate the likelihood associated with each threat agent, attack vector, and security weakness and combine it with an estimate of the technical and business impact to your organization. Together, these factors determine the overall risk.

Sources: The Open Web Application Security Project (OWASP)

Chapter 2: Create an Inventory of Application Assets and Assess Impact

This guide will now present a five-step process for implementing risk-based security management.

Create an Application Profile Template

The first step for implementing risk-based security management is to create an application profile template that can be used to capture critical attributes of every application in the enterprise. The relevant attributes to collect will vary by enterprise, but they typically fall into three categories.

Vital Statistics

The template should include the name of the application, the development team and business unit responsible for maintaining it, a "business owner" or main contact, and details such as the creation date.

Attributes That Reflect Inherent Risk

The template should list attributes that reflect the inherent risk of the application, such as:

- Is the application customer facing, partner facing or internal?
- Functional complexity
- Infrastructure complexity
- Maturity (length of time in production)
- Platform (web/client-server/desktop/mobile)

Attributes That Reflect Criticality and Impact

The template should have room for assessments of each application's criticality to the business and the potential impact of a data breach or an interruption in operations. Factors to consider include:

- Compliance requirements
- Potential damage to the reputation of the organization
- Personally identifiable information (PII)
- Intellectual property
- Legal and contractual obligations

The values for these attributes can be discrete options ("customer facing," "internal facing," or "unknown"), or may represent measurements on a continuous scale (e.g., 1 to 10).

Collect the Data and Make Assessments

Once the application profile template is complete, the security team can drive the process of collecting application data and making assessments about the attributes of each application. It is important to involve developers, QA engineers and IT administrators who understand how the applications are used and managed. It is also desirable to include as participants in the process business managers, compliance officers, members of the legal staff, and others who are in a position to judge the consequences of data breaches and other threats.

It is unlikely that complete information on all applications can be obtained immediately. However, when a critical mass of inventory information is reached, decision-making for application security will improve, which should inspire more groups to participate in the process. Also, the

▶ Assessing Business Impact

When assessing the criticality to the business of applications and the potential impact of breaches or interruptions, security officers should ask questions like:

- Does the application need to comply with regulations and industry standards such as PCI DSS, HIPAA, FISMA, the EU Data Protection Directive, and ISO/IEC security control standards?
- Would a breach cause long-term damage to reputation or customer trust?
- What are the costs if the application is unavailable for an hour? For a day? Even longer?
- Does the application handle confidential data subject to breach notification laws, such as personally identifiable information (PII) about customers or employees?
- Does the application handle intellectual property affecting the enterprise's competitive position, such as engineering designs, software code or business plans?
- Could a breach result in contractual violations or legal liability toward customers or business partners?

process of information gathering can jump-start improved communication and collaboration between security, QA and development.

It is theoretically possible to maintain the inventory information in spreadsheets or homegrown database applications. However, usually these do not scale well or provide good reporting and information-sharing capabilities. Purpose-built application security management solutions like IBM Security AppScan Enterprise include many useful capabilities, such as data import tools, dashboards, reporting and information sharing. They have standard reports and query interfaces that make it easy, for example, to list all applications in the enterprise that are affected by HIPAA standards, or to identify all applications in a specific business unit that are subject to breach notification laws, or to pinpoint exactly which applications are subject to both PCI and HIPAA compliance and contain either customer or employee PII.

Chapter 3: Test the Applications for Vulnerabilities

The second step in the process is to test the applications for vulnerabilities. A spectrum of assessment techniques is available.

Static Analysis

Static analysis (sometimes called "white-box analysis") examines application source code for potential vulnerabilities. It facilitates the detection of security flaws early in the development lifecycle.

Dynamic Analysis

Dynamic (or "black-box") analysis tests running applications from "outside." It simulates many of the techniques of cybercriminals and hackers who would attack the applications from remote systems on the Web.

Interactive Analysis

Interactive (or "glass-box") analysis analyzes applications by placing runtime agents on the servers where they are running and examining results from within the application environment as well as outside of it. It combines aspects of static and dynamic analysis to detect a greater variety of security flaws.

Mobile Application Analysis

Mobile application analysis examines code that is downloaded to clients using mobile apps. It helps detect client-side vulnerabilities and exploits.

Using these techniques in combination provides the most complete coverage of potential security defects.

▶ Testing for Vulnerabilities

Security managers should ask if their application security testing solution:

- Manages vulnerability testing throughout the software development lifecycle?
- Scales efficiently and seamlessly from a few applications to thousands?
- Is accurate in identifying valid defects and eliminating false positives?
- Can be used to scan and manage multiple applications by multiple users across development, QA and production?
- Covers modern and complex sites and major code languages?
- Is equipped to handle mobile application vulnerabilities?

Chapter 4: Determine Risks and Prioritize Vulnerabilities

Managers and security teams are now equipped with a comprehensive view of applications across the enterprise. They have detailed assessments of the business criticality of each application, and the vulnerabilities within it, which give them a complete picture of application risk (Figure 2).

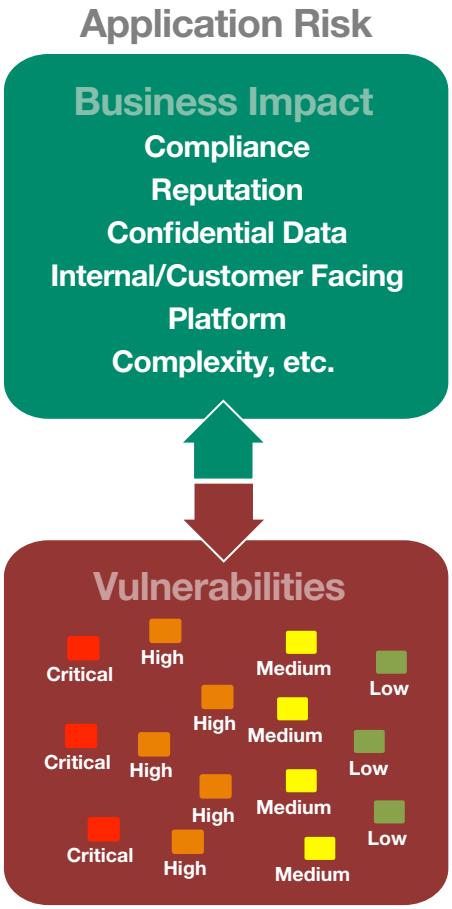


Figure 2: Application risk is a function of the potential impact of a successful attack on the application, and the severity of vulnerabilities within it.

▶ Creating an Application Security Risk Rating

A best practice at this point in the process is to compute an overall security risk rating for each application. Although the risk rating can be based on a complex formula, it is usually preferable to take a conceptually simple approach. For example:

Security risk rating =
Business Impact x
Maximum Vulnerability Severity

where **Business Impact** is calculated based on risk attributes and business impact assessments, and **Maximum Vulnerability Severity** is the level of the highest-rated vulnerability found in the application (Critical/High/Medium/Low/None).

Setting Priorities: By Application

The security team is now ready to begin setting priorities for remediation. A typical first step is to focus on the applications with "critical" risk ratings (Figure 3). The data in the application asset inventory can also provide an explanation of why each of these represents a significant risk to the enterprise (Figure 4).

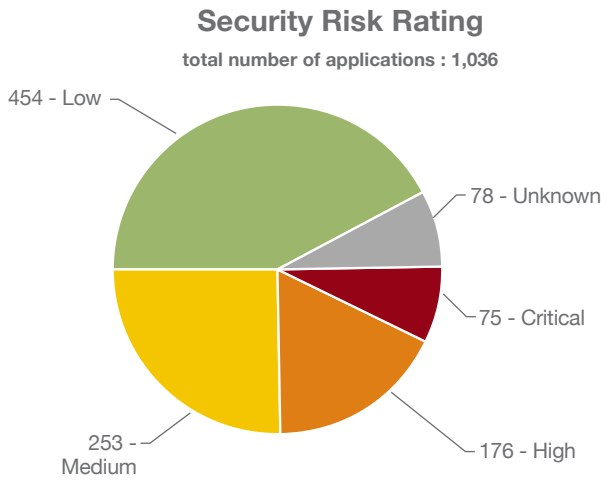


Figure 3: Managers get enterprise-wide visibility into the number of applications with high risk ratings

This information allows managers to create remediation plans based on priorities that align with business risks and strategies. They can address the highest-priority vulnerabilities first, both across applications and within applications. Security teams can focus on preventing the breaches that might have the greatest negative impact on the enterprise.

Setting Priorities: By Vulnerability Type

The same data set can be used to show the prevalence of specific vulnerabilities at several different levels:

- Within individual applications (Figure 5 on the next page)
- Within teams or business units
- Across the enterprise

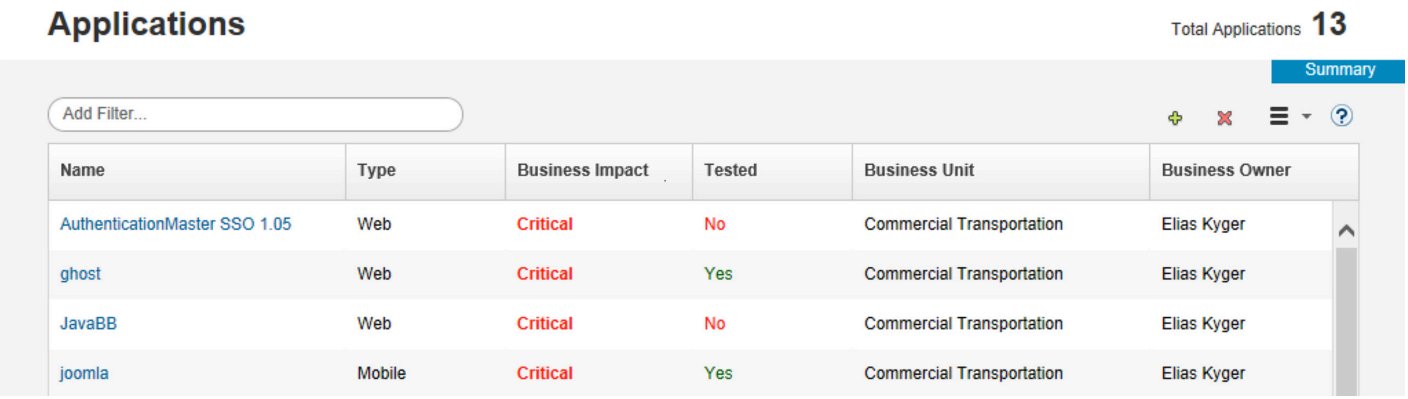


Figure 4: Reports show why each application represents a potential risk.

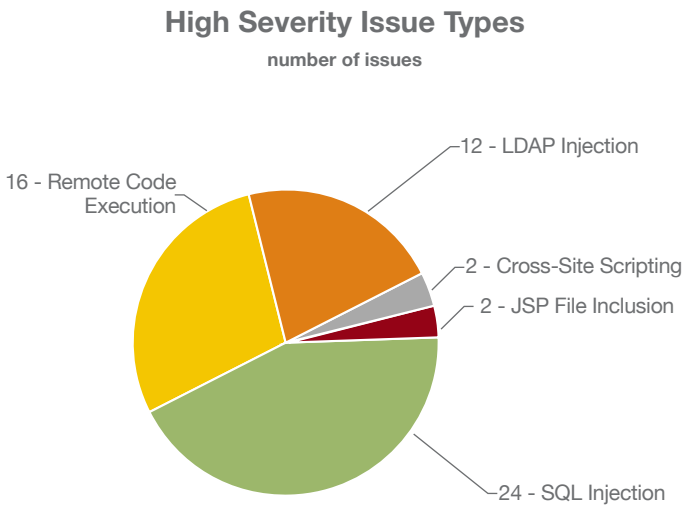


Figure 5: Data can show the most prevalent vulnerabilities in specific applications, or across the enterprise.

This information can be used to set priorities for remediation. For example, a team that has been experiencing a large number of cross-site scripting vulnerabilities can be told to put those defects at the top of its task backlog.

Setting Priorities: By Development Team or Business Unit

The same information can also provide comparative data across development teams, business units, and other organizational entities (Figure 6). This enables organizations to share security and testing resources, from silos with low risk profiles or low workloads to areas that represent higher risks to the business or have larger backlogs of critical vulnerabilities. It also provides guidance on where investments in training, tools and management can provide the biggest return.

De-Prioritizing Vulnerabilities and Accepting Risks

Another benefit of a strategic process for prioritizing vulnerabilities is that it provides a systematic way to lower the priorities of issues that are not critical to the specific enterprise. These might include vulnerabilities associated with attacks on other industries, or vulnerabilities that can only be exploited in conjunction with specific software or servers that are not present in the environment.

Figure 6: Data can show which teams or business units can benefit the most from additional security and testing resources, such as the Commercial Transportation team in the fictional example above..

Chapter 5: Remediate the Risks

After priorities have been established, the security, development and QA teams can work together to remediate vulnerabilities within applications. A basic workflow is shown in Figure 7:

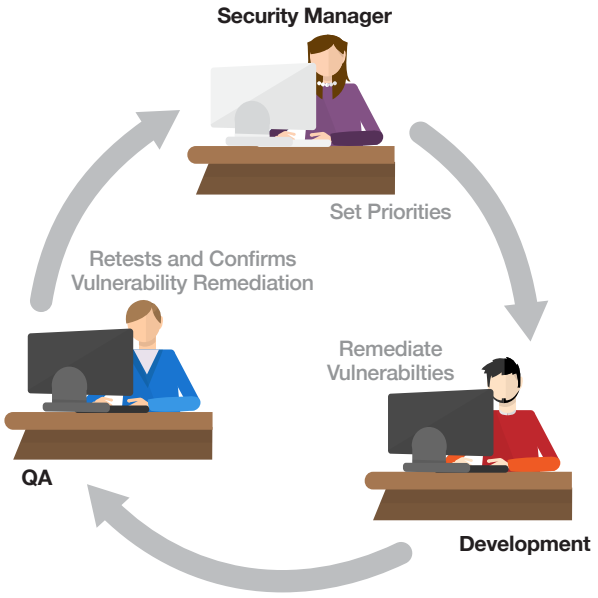


Figure 7: A workflow cycle for remediating vulnerabilities.

- A security manager sets priorities for remediation and assigns ownership of remediation tasks to the development team.
- Developers remediate the highest-priority vulnerabilities.
- A QA engineer runs the appropriate tests against the new version of the application, confirms that remediation steps have been effective, and forwards the data to the security manager.

However, not all vulnerabilities need to be addressed by changing application code. One alternative is to create virtual patches and deploy them to a Web application firewall (WAF) or an intrusion prevention system (IPS). Virtual patches enable the WAF or IPS to detect attempts to exploit application vulnerabilities and block them before they reach the application. While this is generally not a good long-term fix, it can provide protection until resources are available to fix the application.

Another alternative is to accept some risks and simply monitor for indicators that the vulnerabilities are being exploited. This approach is acceptable when the potential impact of the threats is minimal.

▶ Improve Processes

Remediation is not just about fixing individual defects. Security managers should look for opportunities to improve underlying processes. For instance, if data indicates a large number of authentication-related vulnerabilities, the security organization could:

- Give developers more training on how to ensure secure authentication and protect session tokens.
- Provide code libraries or templates that address the issues.
- Create test plans and test scripts to detect authentication defects early in the development cycle.
- Require best practices for secure authentication in application specifications, so the issues are visible to developers and QA engineers.

Chapter 6: Measure Progress and Demonstrate Compliance

Measuring Progress

As organizations follow the remediation workflow described in the previous chapter, trend data becomes available showing progress (or lack of it) for teams and business units, in terms of high-priority vulnerabilities, total vulnerabilities, and vulnerabilities of specific types. This data provides guidance for the next round of remediation plans, enabling security and development teams to continually focus on the highest-priority vulnerabilities.

But application security information can also give CISOs and risk officers a view of the organization's overall risk posture. For example, a graph might show a long-term trend of reducing critical and high vulnerabilities across all business units (Figure 8). A dashboard can depict at a glance information such as the total number of applications with critical vulnerabilities, the business units and development teams with the most (and least) pressing challenges, and the proportion of applications that have been tested.

▶ Taking a Risk Perspective

With the right application security data, security managers can answer questions such as:

- Is the overall risk posture of the organization improving?
- Are we allocating resources where they will have the greatest impact reducing business risk?
- Can we show our CEO and board positive results, and do we have the facts to show them where additional investments in security could further reduce risk?

Demonstrate Compliance

Compliance with government regulations and industry standards is one of the main drivers of today's application security activities. A risk-based approach to managing

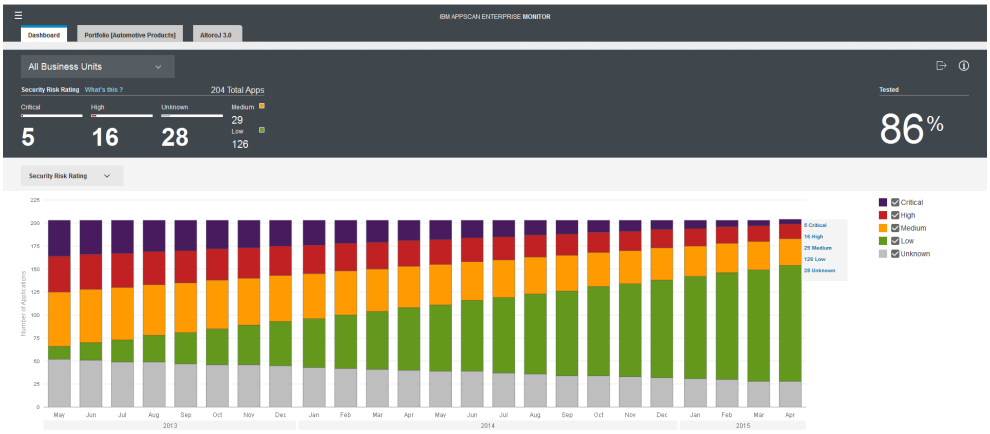


Figure 8: Graphs and dashboards can give CISOs and security officers an at-a-glance summary of the organization's overall security posture.

application security makes it easier to focus resources on activities that will improve compliance, and also to demonstrate progress to compliance officers and auditors.

Security managers can use the application inventory to identify applications with compliance requirements, and to target compliance-related vulnerabilities within individual applications, so those can be given priority by the development teams.

Some application security management solutions (such as IBM Security AppScan Enterprise) include reporting templates that map application security data to key government regulations and industry standards. By quantifying a reduction in the number of high-priority vulnerabilities associated with compliance issues, enterprises can document progress toward compliance goals. And because the risk-based approach to managing application security represents a best practice in information security, implementing the process described in this guide objectively demonstrates the enterprise's commitment to improving compliance.

▶ Demonstrating Focus and Progress

A risk-based approach to managing application security makes it easier to focus resources on activities that will improve compliance, and also to demonstrate progress to compliance officers and auditors.

Chapter 7: Monitoring Applications in Production

A risk-based approach to managing application security, when supported by the right tools, can also improve the ability of other security tools to monitor running applications, identify attempts to exploit vulnerabilities, and protect against attacks.

Intrusion Prevention

Detected vulnerabilities can be shared with an intrusion prevention system (IPS). That allows the IPS to correlate web application attack events with vulnerabilities in real time to determine if the vulnerabilities are being actively exploited.

Security Information and Event Management

Vulnerability data and application risk scores can be shared with a security information and event management (SIEM) solution. That allows the SIEM to issue alerts based on application risk levels, so analysts in the security operations center (SOC) can prioritize alerts more effectively. The alerts also contain more context about threats, so that incident response (IR) teams can understand more about the attacks and address them before they impact the business.

Database Activity Monitoring

Application information shared with database monitoring tools enables these tools to more quickly pinpoint database vulnerabilities and configuration flaws.

Web Application Protection

Vulnerability data shared with a Web application firewall (WAF) can be used to create "virtual patches" that block threats at the network perimeter before they can reach and exploit security flaws in the applications.

Mobile Application Protection

Integrating application security solutions with mobile app protection tools allows the latter to more quickly identify and eliminate client-side vulnerabilities in mobile applications. An integrated application hardening and runtime protection mechanism helps shield individual applications from risks, including hacking attacks and malware exploits.

▶ What You Have Learned

A risk-based approach to managing application security can help enterprises:

- Obtain visibility into the state of application security across the enterprise.
- Set priorities for testing and remediation that align with business risks and strategies.
- Allocate resources to help prevent the most likely and most harmful data breaches.
- Measure progress toward application security goals.
- Strengthen collaboration between security, QA and development.
- Improve the monitoring of applications in production.
- Continuously monitor the organization's overall risk posture.

Appendix: IBM Products and Services for Application Security

IBM Security AppScan Standard

IBM® Security AppScan® Standard automates application security vulnerability testing to help organizations decrease the likelihood of web application attacks. It supports:

- Broad coverage to scan and test for a wide range of application security vulnerabilities.
- Precise scanning and advanced testing that deliver high levels of accuracy.
- Quick remediation with prioritized results and fix recommendations.
- Enhanced insight that helps manage compliance and provides awareness of key issues.

IBM Security AppScan Source

IBM® Security AppScan® Source helps organizations lower costs and reduce risk exposure by identifying web-based and mobile application source code vulnerabilities early in the software development lifecycle, so they can be fixed before deployment.

IBM Security AppScan Enterprise

IBM® Security AppScan® Enterprise enables organizations to mitigate application security risk, strengthen application security program management initiatives and achieve regulatory compliance. It delivers:

- Scalable application security testing using a variety of testing techniques.
- Test policies, scan templates and vulnerability remediation advisories to help implement application security programs.
- Detailed security reports and enterprise level dashboards to provide visibility of risk and compliance.

IBM Application Security Analyzer

IBM Application Security Analyzer is designed to help secure mobile and dynamic applications by detecting dozens of today's most pervasive published security vulnerabilities, and to secure web applications deployed on IBM Bluemix™.

Arxan Application Protection

Arxan Application Protection for IBM® Solutions extends IBM Security AppScan® vulnerability analysis capabilities to mobile application hardening and runtime protection.

Cigital Application Security Testing Managed Services

Cigital Application Security Testing Managed Services from IBM® deliver scalable scanning, testing and remediation services. An annual subscription provides effective web application penetration testing on demand, up-to-date vulnerability insight, and live expert remediation guidance.

For more information:

Test-drive an application scanning solution for your applications with our free IBM AppScan Trial.

Share a 3-minute video with your team that highlights the importance of application security management.

Read an industry analyst report about the top application security software vendors.

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
July 2015

TechTarget Custom Media

WGM12350-USEN-00