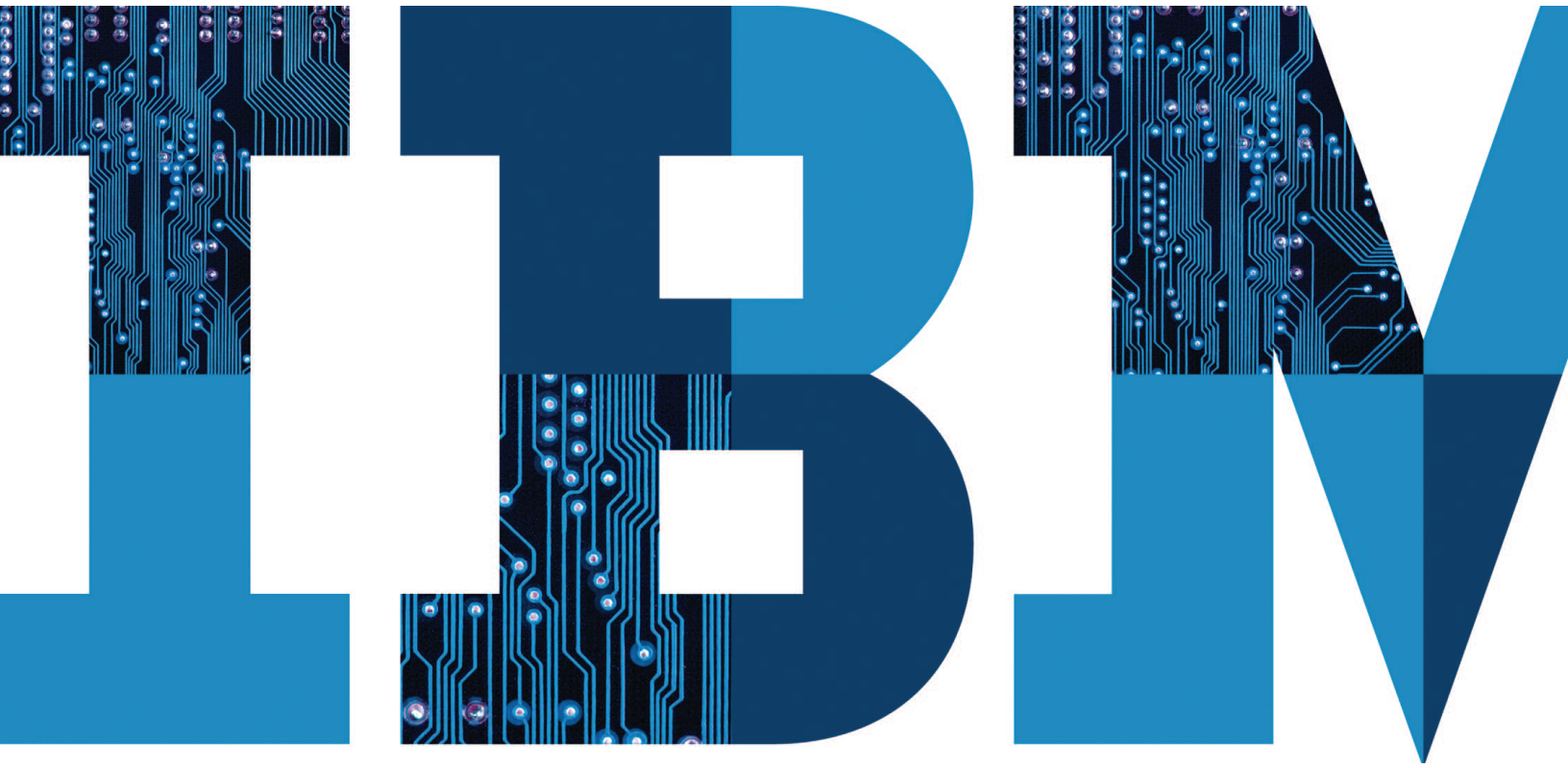
IBM Security
Thought Leadership White Paper

February 2018

Five Steps to Achieve Risk-Based Application Security Management

A best practices guide to make application security a strategically managed discipline in your organization.



IBM

Table of Contents

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

Software applications play a critical role in business. They support strategic business processes, facilitate interaction with customers and business partners, house sensitive customer and employee data, and store an organization's mission-critical intellectual property. Application security is paramount. Development teams must implement authentication, access control, identity management, data protection, session control, data encryption and mobile security. In doing so, development teams face a number of daunting challenges.

Technical Challenges

Software developers are seldom 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 4: SQL Injection Attacks and the OWASP Top 10 Security Flaws.[1])

Aggressive Adversaries

A growing number of cybercriminals, "hacktivists" and state-sponsored hackers understand that capitalizing on application vulnerabilities is often the fastest and easiest way to launch multimillion-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.

Organizational Factors

In most organizations, incentive systems work against a strong emphasis on security. Developers and quality assurance (QA) engineers are 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 armed only with spreadsheets and bug tracking packages) to assess security levels, manage testing for vulnerabilities, and track remediation efforts.

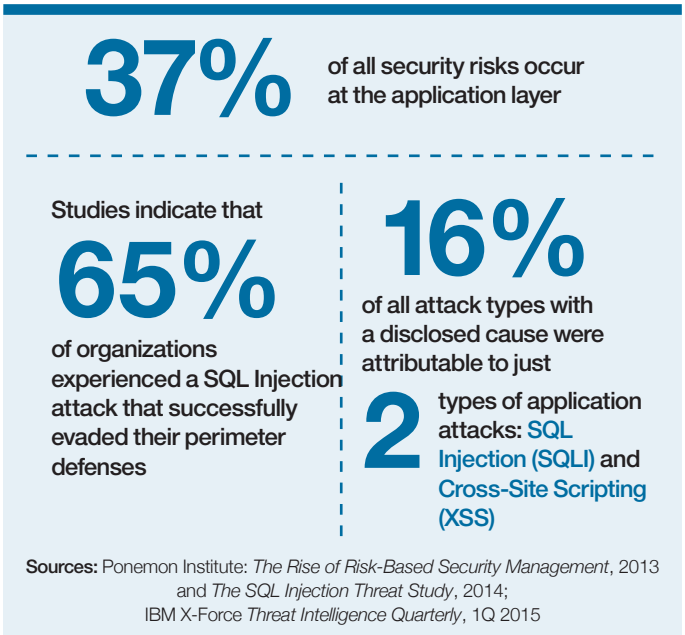


Figure 1: Application-layer attacks are commonplace, and often successful

▶ SQL Injection Attacks and 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.

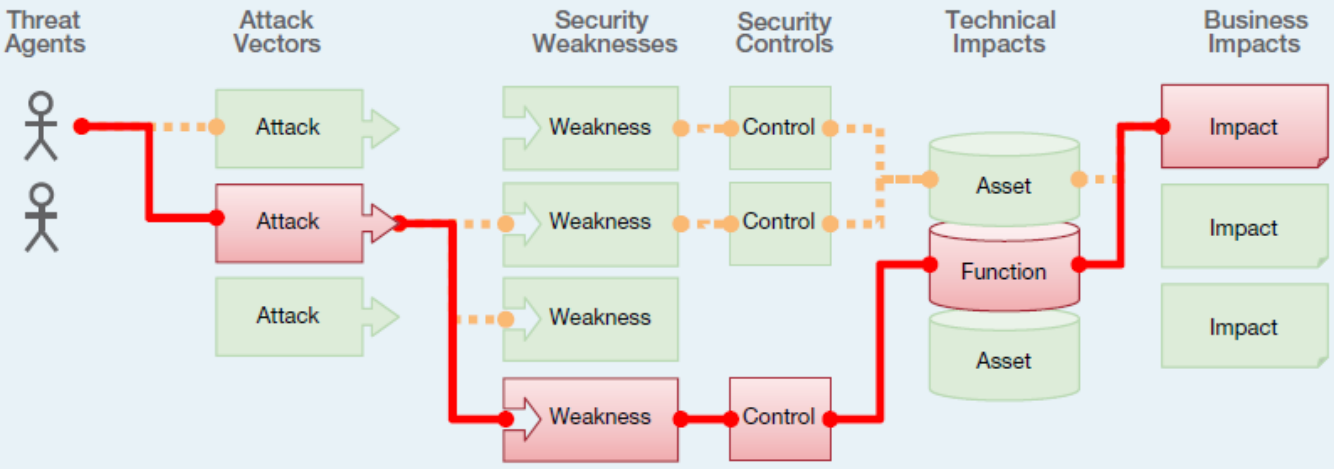
For example, injecting the query SELECT UserId, Name, Password FROM Users WHERE UserId = 105 or 1=1 could cause the application to reveal the username and password of every authorized user. Adding the phrase DROP TABLE Customers to the end of the query would cause the application to delete the entire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

Several techniques 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, and SMTP header injection attacks. 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.

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 warrant 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)

As a result of this ad hoc approach to application security management:

- Organizations don't realize the potential consequences of their "rush-to-release" behavior.
- No one 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 organizational silos and cannot be reallocated to 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

It is important to begin with a consolidated view of all applications across the enterprise. IBM's AppScan Enterprise provides a consolidated view and 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 exposure.

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 to focus on the greatest risk. 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 before the applications are released.
- 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 internal "bullying" 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.

Here are the five steps to achieve risk-based application security management:

1. Use an application security risk management product to create and manage an inventory of application assets and assess their business impact.
2. Test the applications for vulnerabilities using a broad set of techniques (static, dynamic, open source, etc.). Favor testing tools that use cognitive/AI approaches to reduce resources and time.
3. Determine the risks and prioritize vulnerabilities according to their potential impact on your business.
4. Remediate the vulnerabilities by seamlessly integrating into the DevOps process.
5. Measure progress, demonstrate compliance and monitor performance.

Step 1: Create an Inventory of Application Assets and Assess Impact

Create an Application Profile Template

To capture the critical attributes of every application in the enterprise, create an application profile template. The attributes vary by company but 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 data.

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/open source)

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). IBM AppScan allows you to customize the out-of-the-box attributes to your specific needs.

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 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 DSS and HIPAA compliance requirements and contain either customer or employee PII.

▶ Assessing Business Impact

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

- Does the application need to comply with regulations and industry standards such as PCI DSS, HIPAA, FISMA, the EU GDPR, and ISO/IEC security control standards?
- Would a data 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 that is 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 positioning, such as engineering designs, software code or business plans?
- Could a breach result in contractual violations or legal liability toward customers or business partners?

Step 2: Test the Applications for Vulnerabilities with Cognitive/AI Application Security Testing Technology

Next, test the applications for vulnerabilities. At this step, the dilemma of speed versus accuracy is most apparent.

Conventional application security testing involves scanning the data flows of an application in search of gaps or lack of data sanitization methods, then having a security expert sift through thousands of potential vulnerabilities, the vast majority of which are likely to be false positives. However, this tedious and time-consuming process works against the agile development goals of DevOps methods.

Organizations have attempted to overcome this bottleneck by scanning more rapidly but less thoroughly, or by hiring more security experts. But experts are scarce and expensive—a skills shortage that has led many organizations to outsource the process, which may increase their costs without reducing their turnaround times.

To solve this problem, IBM has applied its groundbreaking work in machine learning computing to the realm of application security testing. The result is two key capabilities: Intelligent Finding Analytics (IFA) and Intelligent Code Analytics (ICA).

To simulate the knowledge and skills of a security expert, IFA utilizes machine learning capabilities that reduce the number of false positives, enabling IFA to deliver the same degree of accuracy (up to 98%) as human security experts. However, unlike a human, IFA delivers results in minutes, if not seconds, not in hours or days. This quick turnaround

▶ The Vital Importance of DevOps and Open Source Testing

The goal of DevOps is to increase the speed of application development by combining software engineering with software operations and quality assurance, in a continuous process. In this context, the use of conventional security testing practices, typically performed manually by security experts, would traditionally create delays and defeat the purpose of DevOps. However, by automating and integrating security testing so that it is performed frequently in the DevOps cycle—a process sometimes known as SecDevOps—the goal of rapid, yet secure application development can be achieved.

Open source software is a mainstay of applications across many enterprises. But the popularity of open source components makes them attractive targets for malicious actors. Some of the most damaging attacks, including Heartbleed, POODLE and Shellshock, have been launched against open source code. Remediating open source vulnerabilities has quickly become the highest priority to reduce risk in many corporate application security programs.

It is important to integrate open source application security testing into DevOps. This can be done by utilizing IBM Application Security Open Source Analyzer, included in IBM's Application Security on Cloud offering, which identifies open source components and matches them against a list of known vulnerabilities. With open source software, security reviews must be performed early and often in the development cycle, since threats are constantly evolving.

enables application security testing to be performed frequently so developers can maintain the process of continuous development. IFA shows developers precisely where security issues are located in the code and gathers them into fix groups, a practice that enables multiple problems to be remediated simultaneously.

ICA applies machine learning to the identification and markup of APIs. Developers are using more and more APIs through a variety of components, including frameworks. New APIs need to be analyzed on how they process data and new security rules written to help automate the use by the developers. If new APIs are discovered and not analyzed properly, potential false negatives can be left in the code. New APIs often take AppSec teams time to analyze and create new security rules or markup. ICA can accomplish this in seconds, improving scan coverage without delaying the scan for manual review and markup.

There are five types of testing techniques available as part of the IBM solution:

Static Analysis

Static analysis (sometimes called "white-box analysis") examines application code for potential vulnerabilities using trace analysis or pattern matching. 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 volume of security flaws.

Mobile Application Analysis

Mobile application analysis is a form of interactive analysis that tests how the mobile app interacts with the back-end application. It helps detect client-side vulnerabilities and exploits.

Open Source Analysis

Open Source Analysis examines the use of open source packages in an application and identifies which ones have known vulnerabilities that can be exploited. Using these techniques in combination provides the most complete coverage of potential security defects.

▶ 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, IBM AppScan calculates:

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 calculated based on vulnerability severity and volume found in the application (Critical/High/Medium/Low/None).

Step 3: 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 the applications, which give them a complete picture of their overall application risk (Figure 2).

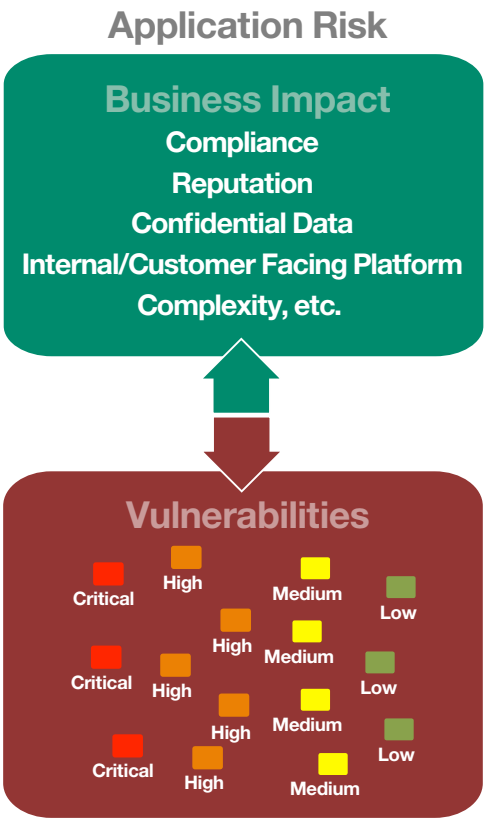


Figure 2: Application risk is a function of the potential impact of a successful attack on the application, and the severity of vulnerabilities within it.

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).

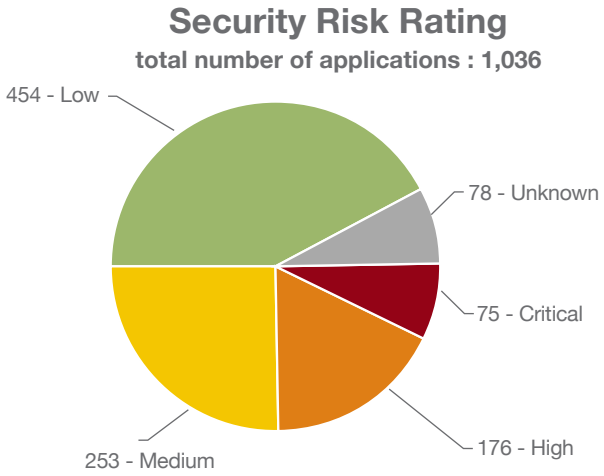


Figure 3: Managers get enterprise-wide visibility into the number of applications with high risk ratings

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

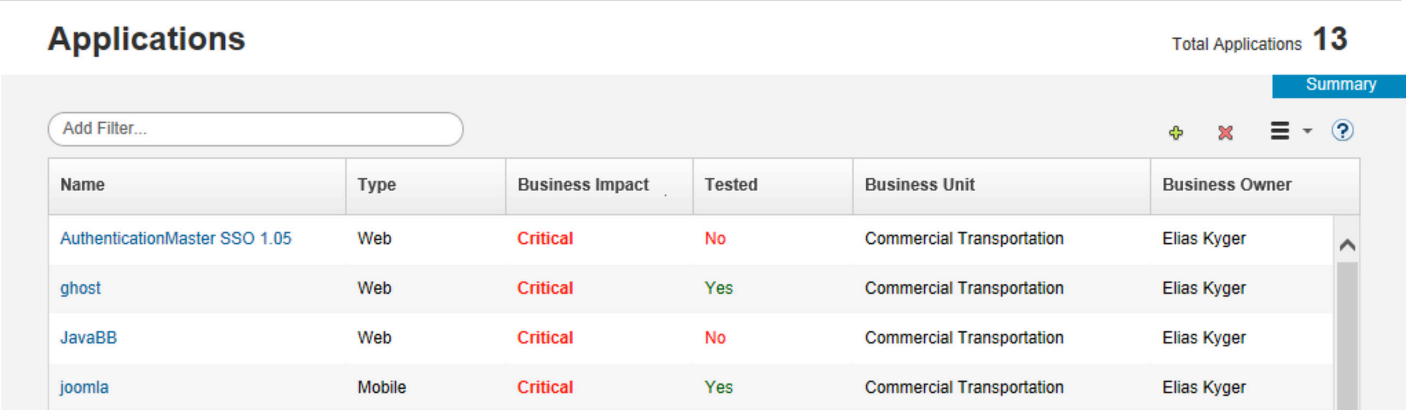


Figure 4: Reports show why each application represents a potential risk.

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)

Within teams or business units across the enterprise, this information can be used to set priorities for remediation. For example, a team that has been generating a large number of cross-site scripting vulnerabilities can be told to put those defects at the top of its task backlog.

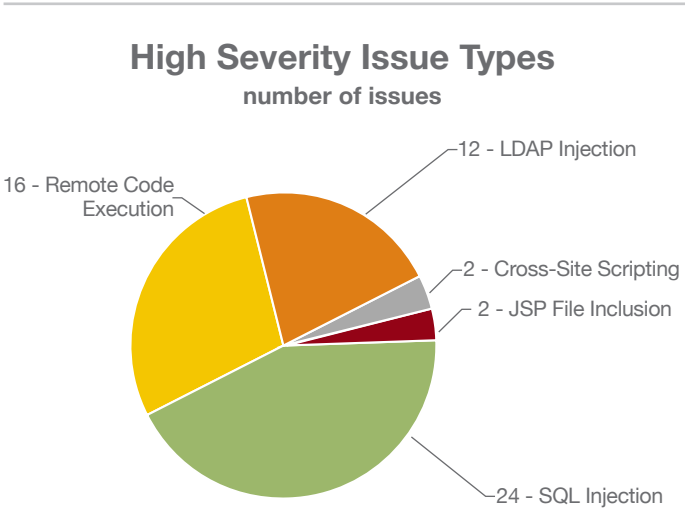


Figure 5: Data can show the most prevalent vulnerabilities in specific applications, or across the enterprise.

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 maximum return.

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..

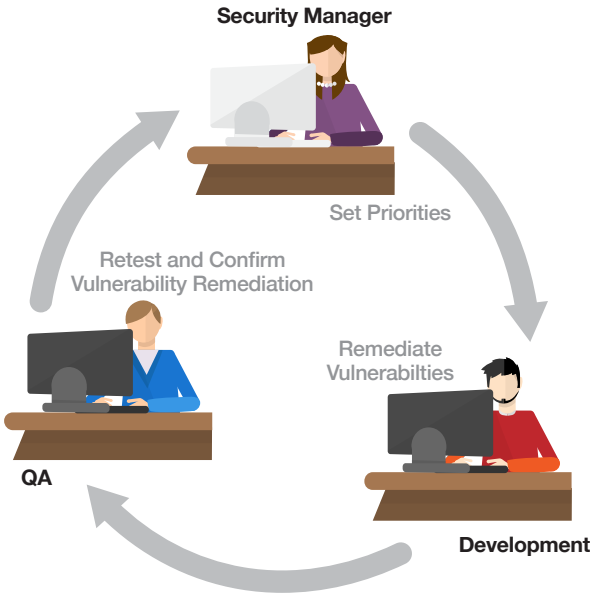


Figure 7: A workflow cycle for remediating vulnerabilities.

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 be exploited only in conjunction with specific software or servers that are not present in their environment.

Step 4: 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 in Figure 7 to the left.

A security manager sets priorities for remediation and assigns ownership of remediation tasks to the development team. A best practice is to provide this information directly into the tools developers already use: e.g., their IDE or defect tracking system. 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.

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 that a large number of authentication-related vulnerabilities are present in applications, 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.
- As mentioned earlier, IFA aggregates vulnerable traces into fix groups, which identifies a single code fix that can resolve multiple vulnerabilities, making developers more efficient.

Step 5: Measure Progress, Demonstrate Compliance, Demonstrate Focus and Progress, and Monitor Performance

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

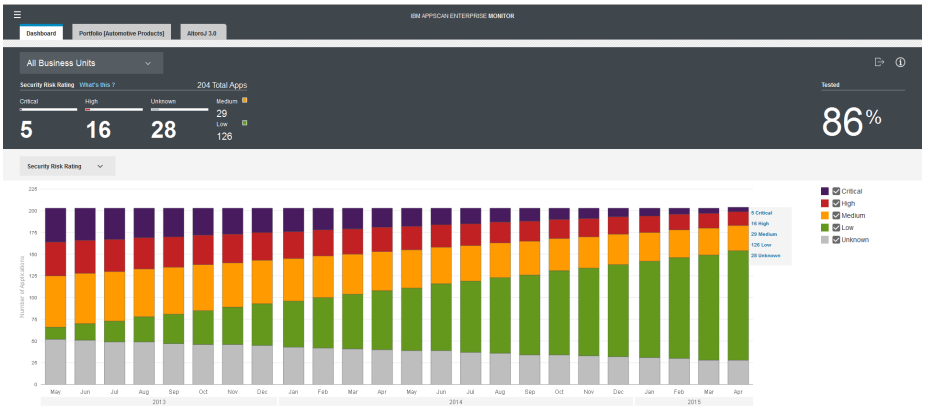


Figure 8: Graphs and dashboards can give CISOs and security officers an at-a-glance summary of the organization's overall security posture.

teams with the most (and least) pressing challenges, and the proportion of applications that have been tested.

Demonstrating Compliance

Compliance with government regulations and industry standards is one of the main drivers of today's application security activities. The emergence of the General Data Protection Regulation (GDPR), a European Union mandate for protection of personal data, significantly raises the stakes for compliance. GDPR carries severe penalties for noncompliance—the greater of €4 million or 20% of annual revenue—with a May 2018 go-live date. A risk-based approach to managing application security makes it easier to focus resources on activities that will improve compliance and demonstrate progress to compliance officers and auditors.

▶ 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 on 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?

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 testing 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 mandates, 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 processes described in this guide objectively demonstrates the enterprise's commitment to improve 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 demonstrate progress to compliance officers and auditors.

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.

Security Information and Event Management

Vulnerability data and application risk scores can be shared with a security information and event management (SIEM) solution such as the IBM QRadar® Security Analytics Platform. With that information, the SIEM can issue alerts based on application risk levels, so analysts in the security operations center (SOC) can prioritize their 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. The IBM QRadar platform integrates log management, anomaly detection, and configuration and vulnerability management. The result is superior threat detection, greater ease of use and lower cost of ownership.

Database Activity Monitoring

Application information shared with a database monitoring tool enables these tools to more quickly pinpoint database vulnerabilities and configuration flaws. IBM Security Guardium® is a comprehensive data security platform that provides database monitoring among its full range of capabilities, from discovery and classification of sensitive data, to vulnerability assessment of data and file activity, to monitoring, masking, encryption, blocking, alerting and quarantining to protect sensitive data.

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 Application Security on Cloud

IBM Application Security on Cloud helps secure your organization's web and mobile applications, by detecting dozens of today's most pervasive published security vulnerabilities with its machine learning capabilities. IBM Application Security on Cloud helps eliminate vulnerabilities from applications before they are placed into production and deployed. Convenient, detailed reporting permits you to effectively address application security risk, enabling application users to benefit from a more secure experience. IBM Open Source Analyzer helps to identify vulnerabilities in open source components, by automating security testing and configuring scanning for open source. For a complimentary trial of IBM Application Security on Cloud, please visit: https://www.ibm.com/us-en/marketplace/application-security-on-cloud

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 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.

For a complimentary trial of IBM Security AppScan Standard, please visit: https://www.ibm.com/developerworks/downloads/r/appscan/index.html

IBM Security AppScan Source

IBM® Security AppScan® Source leverages the machine learning technology to help organizations lower costs and reduce risk exposure by identifying web-based and mobile application 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 Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
February 2018

References

1 OWASP Top 10 Application Security Risks – 2017

Please Recycle