Michael Kristen

(E-Mail: michael.kristen@de.ibm.com)
ist IT Spezialist und seit 2006 bei IBM Rational im Bereich Qualitätsmanagement tätig mit dem Fokus auf Webanwendungssicherheit.

Statische Quellcode-Analyse zur Auffindung von Sicherheitslücken

Unternehmen, die Sicherheitslücken in Anwendungen vor deren Auslieferung oder Implementierung schließen, profitieren von enormen Kosteneinsparungen – intern und für Kunden, Geschäftspartner und andere Stakeholder, die mit ihrer Software arbeiten.

Integration von Quellcode-Schwachstellentests in den Softwareentwicklungslebenszyklus

Die möglichst frühzeitige Ermittlung und Beseitigung von Sicherheitsproblemen kann dazu beitragen, die Entwicklungskosten zu reduzieren und die Softwarequalität zu verbessern.

Unzählige Studien und Empfehlungen von Analysten weisen auf die Bedeutung hin, die Sicherheit bereits während der Softwareentwicklung zu verbessern, anstatt Sicherheitslücken in der Software zu schließen, die erst nach der Implementierung und umfassender Verwendung festgestellt werden. Die Gründe hierfür liegen auf der Hand.

Für die Softwareanbieter ergeben sich durch Sicherheitslücken, die in ihren Produkten gefunden werden, sowohl direkt als auch indirekt Kosten. Die erneute Bereitstellung von Mitarbeitern aus der Entwicklungsabteilung für die Erstellung und Verteilung von Patches kann Softwareanbieter in vielen Fällen Millionen kosten. Die Ausnutzung einer einzigen Sicherheitslücke hat in Unternehmen auf der ganzen Welt in einigen Fällen bereits zu Umsatzverlusten in Milliardenhöhe geführt. Anbieter, denen die Schuld an Sicherheitslücken im Quellcode ihres Produkts gegeben wird, müssen den Verlust von Glaubwürdigkeit, Imageschäden für ihr Markenprodukt und Wettbewerbsnachteile befürchten.

Es herrscht große Einigkeit darüber, dass Sicherheitslücken mit umso geringerem Kostenaufwand behoben werden können, je frühzeitiger sie im Lebenszyklus festgestellt werden. Untersuchungen haben ergeben, dass das Beheben von Softwarefehlern nach der Implementierung 100 mal teurer ist im Vergleich zu den Kosten, die für die Behebung dieser Fehler in den ersten Entwicklungsphasen anfallen. Bei Sicherheitsfehlern liegen die Kosten für deren Behebung zu einem späten Zeitpunkt häufig noch deutlich darüber, da hier nicht nur die Sicherheitslücken geschlossen werden müssen, sondern die Ausnutzung der Schwachstellen auch zu Datendiebstählen, Sabotage oder anderen Angriffen führen kann.

Automatisierte Quellcode-Analysen sind weithin als die effektivste Methode für Sicherheitstests zu einem frühen Zeitpunkt im Lebenszyklus anerkannt, da beliebige Teile des Codes analysiert werden können, ohne eine vollständige Anwendung zu benötigen. Die besten Lösungen auf der Grundlage dieser Technologien liefern die wertvollsten Ergebnisse, indem bei Schwachstellen auf die genaue Codezeile hingewiesen wird und Informationen über die Art der Schwachstelle, deren Bedeutung und die Vorgehensweise zur Fehlerbehebung mitgeliefert werden. Penetrationstests sind ebenfalls ein wichtiges Element der Softwaresicherheit, dessen Bedeutung allerdings erst zu einem späteren Zeitpunkt im Lebenszyklus zum Tragen kommt, wenn es für eine vollständige Anwendung über eine funktionale Schnittstelle eingesetzt werden kann.

Hindernisse auf dem Weg zur Sicherheit

Bei der Einführung von Sicherheitstests in den Entwicklungszyklus kann neben betrieblichen Hindernissen auch ein allgemeines Zögern im Hinblick auf Änderungen oder Korrekturen an bestehenden Prozessen in der Softwareentwicklung zu Verzögerungen bei der Umsetzung von Sicherheitstests beitragen. Ein einfaches Verständnis der zu erreichenden Vorteile auf Unternehmensebene sind jedoch häufig Motivation genug, um Dinge voranzutreiben. Ähnlich diesem konzeptionellen Hin-

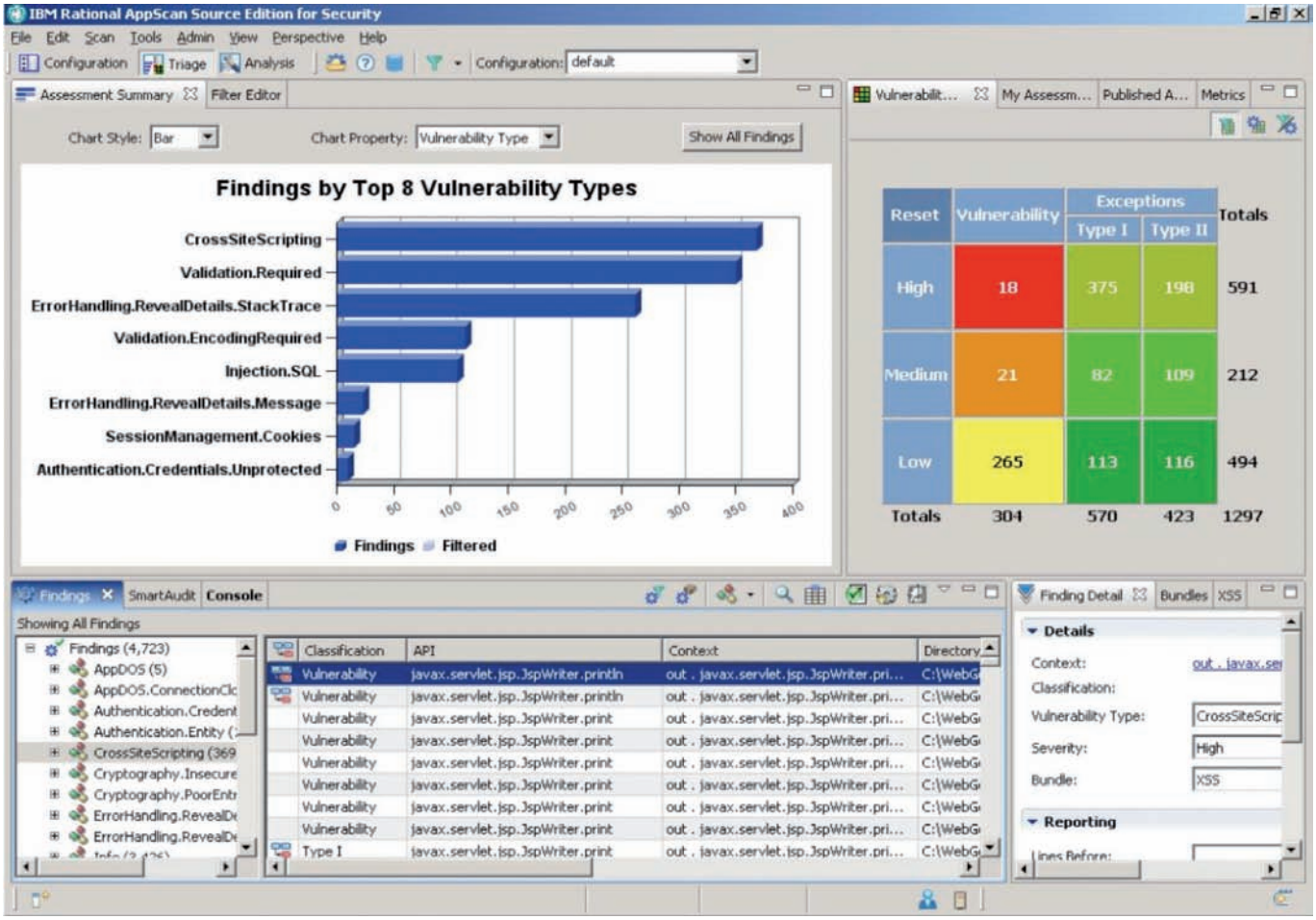

Abbildung 1: IBM Rational AppScan Source Edition

dernis gibt es viele falsche Vorstellungen über die Integration von Sicherheitsanalysen während der Entwicklung, die korrigiert werden müssen, bevor eine Initiative in Bewegung kommt. Folgend werden die gängigsten Hindernisse vorgestellt.

Annahme:
Der Zeitplan in der Entwicklung kann nicht weiter ausgedehnt werden, auch nicht, um Sicherheitsrisiken einzugrenzen.

Fakt:
Im Entwicklungszyklus kann es anfänglich zu Versäumnissen kommen, insbesondere, da sich jeder Einzelne mit dem neuen System vertraut machen muss. Sicherheitsanalyse von Quellcode ist dennoch die Methode mit dem geringsten Zeitaufwand zur Vermeidung von Software-basierten Risiken. Der Prozess trägt letztendlich zu einem kürzeren Entwicklungszeitraum bei, da die Entwickler bewährte und sichere Kodierungsverfahren kennenlernen. Die einzige schnellere Alternative besteht darin, überhaupt nichts zur Verbesserung der Softwaresicherheit zu unternehmen. Dies können sich die meisten Unternehmen langfristig sicherlich nicht leisten.

Annahme:
Wir führen bereits eine Prüfung durch Kollegen durch (Peer Review), zusätzliche Prüfungen des Sicherheitscodes sind daher nicht erforderlich.

Fakt:
Eine Prüfung durch Kollegen ist kein gleichwertiger Ersatz für eine Sicherheitsprüfung. Bei dieser Art der Prüfung wird die Software normalerweise nur auf funktionale Fehler hin überprüft. Viele der maßgeblicheren Sicherheitslücken und Schwachstellen im Design werden hierbei übersehen, es sei denn, die Prüfer sind umfassend mit dem Bereich Anwendungssicherheit vertraut. In vielen Fällen können gutgemeinte und ohne funktionale Fehler implementierte Benutzeranforderungen zu den größten Sicherheitsrisiken führen.

Softwarerisikoanalyse und IBM

Mittlerweile stehen erprobte Technologien für Sicherheitstests zur Verfügung, um zentrale Elemente dieses Prozesses zu automatisieren. Zu den im Handel erhältlichen Lösungen zählen Produkte, die Quellcode automatisch im Hinblick auf Sicherheitslücken analysieren können. Ein Beispiel für eine Lösung zur Analyse von Softwarerisiken ist IBM Rational AppScan Source Edition. Dieses Produkt bietet Funktionen, um Quellcode auf Sicherheitslücken zu überprüfen, und stellt präzise Detailinformationen und Korrekturempfehlungen zu Programmierungsfehlern, Mängeln im Programmentwurf und Richtlinienverletzungen bereit. Mithilfe dieser Informationen können Sicherheitsmanager, Analytiker und Entwickler die Sicherheitsprüfung von Softwareprodukten unterstützen, die Risiken gefährdeter Software kontrollieren und Sicherheitslücken in der Software bereits im Quellcode beseitigen. Die folgenden Tech-

nologien werden u.a. unterstützt: Java, JSP, C, C++, C#, VB.NET, ASP.NET, Classic ASP, PHP, ColdFusion, Perl, Client Side JavaScript.

Rational AppScan Source Edition bietet Organisationen zahlreiche Vorteile, wenn es um Fragen der Sicherheit bei der ausgelagerten Anwendungsentwicklung geht:

Schnelle Identifizierung der schwerwiegendsten Sicherheitsrisiken:

Fester Bestandteil jeder Analyse muss die Untersuchung grundlegender Fragestellungen wie Pufferüberlauf und Eingabe- oder Ausgabeüberprüfung sein. Die bloße Identifizierung dieser Bereiche macht eine Anwendung jedoch nicht sicher. Die unsachgemäße Implementierung anderer Sicherheitsmechanismen, was auch die Verwendung der Kryptografie, Methoden für sichere Netzwerkverbindungen und die Zugriffssteuerung umfasst, kann für die Organisation ein deutlich größeres Risiko darstellen.

Bei der ausgelagerten Anwendungsentwicklung ist die Identifizierung dieser eher unscheinbaren Schwachstellen eine noch anspruchsvollere Aufgabe. So ist das Know-how zum sachgerechten Umgang mit Kryptografie, um Compliance-Anforderungen (z. B. PCI) zu erfüllen, intern möglicherweise vorhanden. Es ist jedoch alles andere als leicht, diese Anforderungen präzise zu formulieren und dann sicherzustellen, dass sie tatsächlich erfüllt wurden.

Mit der patentierten Automatisierungslösung Rational AppScan Source Edition können die unterschiedlichsten Programmierungsfehler und Entwurfsmängel erkannt werden. Auf diese Weise lässt sich problemlos feststellen, ob der gelieferte Code die definierten Sicherheitsanforderungen erfüllt und ob – ausgehend von den Bedingungen des Service-Level-Agreements – ggf. Korrekturmaßnahmen erforderlich sind. Bei der Beurteilung von Zertifizierungs- und Zulassungsmethoden für die Abnahme ausgelagerten Codes werden diese Leistungsmerkmale sehr leicht außer Acht gelassen.

Effektivitätsmaximierung im Bereich der Sicherheitsverwaltung

Softwaresicherheit ist kein Thema, das nur innerhalb einer einzelnen Abteilung relevant ist, sondern im Gegenteil eine unternehmensweite Aufgabe, die Sicherheitsanalytiker, Entwickler, Führungskräfte und Prüfer betrifft. Code-Prüfer und Zertifizierungs- und Zulassungsexperten benötigen Ergebnisse innerhalb von Minuten und nicht erst nach mehreren Tagen. Berichte müssen anpassbar sein, um auf das aktuell geltende SLA abgestimmt zu sein und um die jeweils kritischen Bereiche hervorheben zu können. Auf diese Weise können Abweichungen vom SLA und entsprechende Lösungen schnell und eindeutig identifiziert und ausgehandelt werden. Dank der präzisen, handlungsrelevanten Ergebnisse, Berichte und Korrekturvorschläge von Rational AppScan Source Edition können notwendige Maßnahmen in kürzester Zeit ergriffen werden.

Rational AppScan Source Edition unterstützt Organisationen bei sicherheitsrelevanten Prüfungen und Kontrollen während des gesamten Softwareentwicklungszyklus und stellt eine Lösung bereit, die von unterschiedlichen Interessengruppen und Verantwortlichen genutzt werden kann, angefangen bei Sicherheitsanalytikern über Qualitätssicherungsanalytiker und -entwickler bis hin zu Führungskräften und Sicherheitsmanagern. Es liefert ausführliche Management- und Sicherheitsberichte, die Entwurfsmängel und Richtlinienverletzungen genau identifizieren, was auch Mängel in den Bereichen Zugriffssteuerung, Kryptografie, Eingabeüberprüfung und Protokollierung umfasst. Ein zentrales Verwaltungs-Dashboard stellt zusammengefasste Informationen für ein vollständiges Softwareportfolio zur Verfügung und trägt mit spezifischen Kennzahlen und Trendberichten zu Sicherheitsrisiken dazu bei, die Entscheidungsfindung zu optimieren.

Entwickler können Rational AppScan Source Edition in der integrierten Entwicklungsumgebung (Eclipse und Eclipse basierte Produkte, Visual Studio) ausführen, was die schnelle Identifizierung von Sicherheitslücken auf Code-Ebene und den problemlosen Zugriff auf ausführliche Handlungsanweisungen zu konkreten Problemstellungen ermöglicht. Durch den Einsatz dieser Lösung sind Entwickler bei der Entwicklung und Wartung sicheren Codes gleichberechtigte Akteure. Sie kann außerdem mit führenden Fehlererfassungssystemen verknüpft werden, um die Zeit zwischen Schwachstellenerkennung und -beseitigung zu verkürzen.

Mit Rational AppScan Source Edition stehen der gesamten Organisation die Tools und Informationen zur Verfügung, die notwendig sind, um Sicherheitslücken in jeder Phase des Entwicklungszyklus identifizieren und beheben zu können. Insbesondere bei engen Entwicklungszeitplänen bietet Rational AppScan Source Edition ein kostengünstiges, zweckmäßiges und konsi-

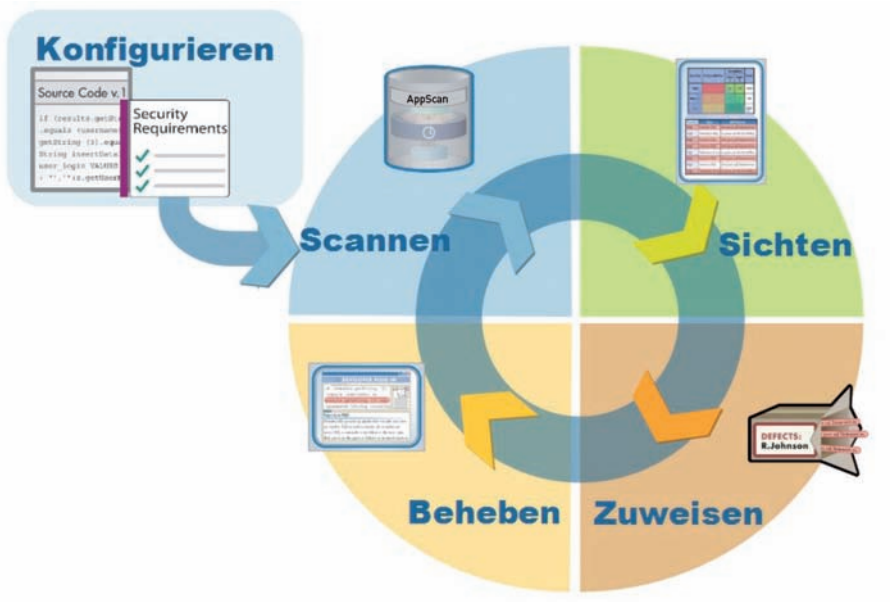

Abbildung 2: Einbindung von Sicherheitsprozessen in den Software Entwicklungszyklus

stente Ergebnisse produzierendes Verfahren an, um die Sicherheit ausgelagerter Anwendungen vor der endgültigen Abnahme zu überprüfen.

Risikomanagement für das gesamte Unternehmensportfolio:

Für die effektive Steuerung einer Anwendungssicherheitsstrategie ist ein Verfahren unverzichtbar, das es ermöglicht, relative Risiken für das gesamte Anwendungsportfolio zu messen, zu vergleichen und gegen die damit verbundenen geschäftlichen Risiken abzuwägen.

Dank einheitlicher Messungen und Kennzahlen, die Rational AppScan Source Edition unterstützt, können Benutzer die Softwarerisiken für das gesamte Softwareportfolio leichter messen und nachvollziehen und die Ergebnisse so aufbereiten, wie es den Anforderungen im Unternehmen entspricht. Die patentierte, compilerbasierte Analysetechnologie ermöglicht die schnelle Analyse auch bei einigen der komplexesten Anwendungen, die derzeit eingesetzt werden. Dank der flexiblen Bereitstellungsoptionen kann das Tool so genutzt werden, wie es sich für die jeweilige Organisation am besten eignet: lokal in der integrierten Entwicklungsumgebung, um überall im Netzwerk darauf zugreifen und Code analysieren zu können, oder als Remote-Installation, sodass mobile Benutzer von ihrem Laptop jederzeit darauf zugreifen können.

Integrierte Sicherheit

Die Frage der Sicherheit von Anwendungen, die das Fundament für wichtige Prozesse innerhalb einer Organisation bilden, darf nicht länger eine untergeordnete Rolle spielen. Natürlich wird niemand unterstellen, dass ein Softwarelieferant absichtlich und böswillig Sicherheitslücken in eine Anwendung einbaut. Es ist vielmehr so, dass die meisten Schwachstellen auf unzureichendes Know-how im Hinblick auf bewährte Verfahren für die sichere Softwareentwicklung oder – angesichts knapper Zeitpläne und immer neuer Anforderungen – auf mangelnde Sorgfalt bei der Programmierung zurückzuführen sind. Glücklicherweise stehen nun Verfahren zur Verfügung, um die Sicherheit unternehmenswichtiger Anwendungen zu untersuchen, Fehler ggfs. zu korrigieren und das Ergebnis erneut zu überprüfen, und zwar unabhängig davon, ob die Anwendungen intern oder von einem Outsourcing-Partner entwickelt werden.

Referenzen

BSIMM2
http://bsimm2.com/index.php
Microsoft's secure sdlc Empfehlungen.
http://msdn.microsoft.com/en-us/library/ms995349.aspx
University of Maryland
http://www.cs.umd.edu/projects/SoftEng/ESEG/papers/82.78.pdf