Web Application Security

Das Pferd am vorderen Ende aufzäumen mit automatischer Source Code Analyse

Dass man zur Herstellung von sicheren Webanwendungen möglichst weit vorn im Software Develoment Lifecycle (SDLC) ansetzen sollte, hat sich mittlerweile herumgesprochen. All zu deutlich sind die Unterschiede in den Auswirkungen auf die Gesamtkosten eines Softwareprojektes, je nach Phase in welcher sicherheitsgebende Maßnahmen ergriffen werden. Eine Vielzahl von Untersuchungen hat dies in der Vergangenheit gezeigt (In [1] heißt es: „Einen Softwarefehler nach dem Deployment zu beheben kostet mehr als das Hundertfache gegenüber der Behebung in der frühen Phase im SDLC").

Doch wo fängt man an, und vor allem, wie? Abbindung 1 zeigt die unterschiedlichen Ansatzpunkte im Entwicklungszyklus eines Stücks Software.

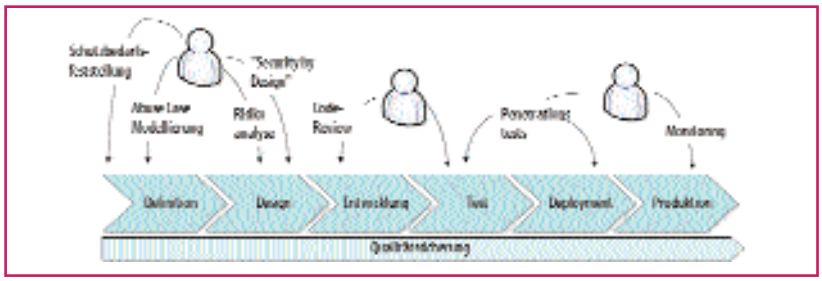

Abbildung 1: Der Software Develoment Lifecycle (SDLC) mit den Aktionspunkten für Sicherheit

Wir wollen uns in dieser Ausgabe mit der Entwicklungsphase beschäftigen. Das ist, wie das Bild zeigt, zwar nicht ganz vorne im SDLC, wohl aber der momentan vielleicht effektivste Ansatzpunkt. Und das liegt daran, dass das Thema Sourcecodeanalyse (SCA) – also die Inspektion des Quellcodes auf Sicherheitslücken – in letzter Zeit große Fortschritte bei der Toolunterstützung gemacht hat. War es bisher immer eine große Hürde, sich auf dieses Feld zu begeben – aufwändig, langwierig und nur von Experten wirksam zu behandeln – so eröffnen spezielle SCA Tools nun die Möglichkeit, die Source Code Analyse zu einem weitgehend automatisierten Schritt im Softwareherstellungsprozess zu machen.

Wo setzt man an?

Automatische Sourcecodeanalyse kann grundsätzlich zu drei Zeitpunkten stattfinden: (1) Während der Entwicklung, (2) beim Erreichen größerer Meilensteine und (3) während der abschließenden Qualitätssicherung. Bei (1) befindet sich das Tool unter Kontrolle des Entwicklers, bei (2) und (3) sinnvollerweise in der Hand des zentralen Sicherheits- oder Qualitätssicherungsteams. In der Hand des Entwicklers besteht die Rolle des Tools nicht nur darin, den Code auf Sicherheitsmängel zu untersucht, es trägt auch nachhaltig dazu bei, dass beim Entwickler ein Bewusstsein für die Sicherheit entsteht, so dass manch sonst auftretender Mangel gar nicht erst bis in den Code gelangt. Die Analyse in (2) und (3) durch ein geschultes und geübtes Team bringt zudem den Vorteil, dass Aspekte, die über das Gesichtsfeld des einzelnen Entwicklers hinausgehen, hier Berücksichtigung finden. Wichtig ist es, dass das Tool eine Komponente mitbringt, die das verteilte Arbeiten mit den typischen Workflows zwischen Entwicklern, Projektleitern und Qualitätssicherung unterstützt.

Wie arbeitet ein SCA Tool?

SCA Tools untersuchen den Code statisch, d.h. sie verarbeiten den Quellcode in ähnlicher Weise wie ein Compiler. Durch einen entsprechenden Schalter im Buildprozess wird das SCA Tool aufgerufen. Es liest die Codedateien ein und konvertiert sie in ein für die Sourcecodeanalyse optimiertes Zwischenformat. Auf diesem Code erfolgt nun die Analyse auf Sicherheitsmängel. Dabei kommen neben den Standardregeln auch benutzerdefinierte, auf die jeweilige Anwendung, Architektur und spezifischen Konventionen zugeschnittene Regeln zur Anwendung. Das Ergebnis der Analyse kann auf verschiedene Weise aufbereitet werden. Für den Entwickler ist die Integration in die IDE sicher das nützlichste Feature. Ein Klick auf den Fehlereintrag führt ihn direkt an die identifizierte Codestelle. Die mitgelieferte Erklärung des Fehlers hilft beim Verständnis und liefert Hinweise für die Behebung – etwas, das bei herkömmlichen Blackboxpentests von Webanwendung immer wieder Schwierigkeiten hervorruft, nämlich aus der extern gemachten Beobachtung einer Schwachstelle auf die Ursache im Code zu schließen und die richtige Maßnahme zur Behebung zu ergreifen. Schließlich ermöglicht eine Managementconsole allen am Projekt Beteiligten die Einsichtnahme in die Ergebnisse und den Bearbeitungsstand der einzelnen Probleme.

Wer sich bereits mit Application Scannern, also Tools, die zum Blackboxtest von Webanwendungen eingesetzt werden, beschäftigt hat, sollte beachten: Im Unterschied zu solchen Scannern, die das Ziel haben, möglichst einen fertigen und um

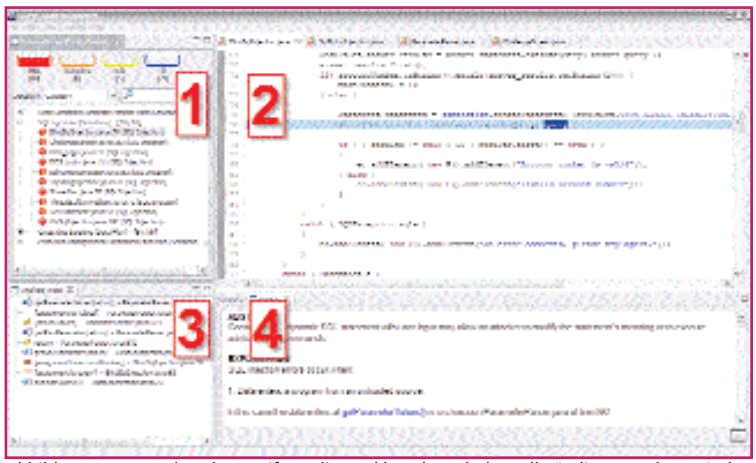

Abbildung 2: Screenshot der Fortify Audit Workbench nach der vollständigen Analyse mit den Bereichen: (1) Liste der gefundenen Schwachstellen, (2) die Fehlerstelle im Sourcecode, (3) Trace des Fehlers bis zur Fehlerquelle, (4) Beschreibung der Schwachstelle und Anleitung zur Behebung.

Falschmeldungen bereinigten Bericht abzuliefern, haben SCA-Tools etwas geringere Ansprüche. Sie teilen die Arbeit in drei Schritte ein und weisen dem menschlichen Experten darin eine zentrale Rolle zu: Die Analyse ist die Phase des statischen Scans durch das Tool. Hierbei treten in der Regel auch viele Fundstellen auf, die entweder nicht als Schwachstelle zu bewerten sind oder die schlichtweg False Positivs darstellen. In der Phase zwei, dem Audit, ist daher der Experte gefragt, der den Bericht analysiert und manuell bewertet und bereinigt. Er stellt mithilfe des Tools den fertigen Bericht in Phase 3 zusammen und macht ihn schließlich über das Managementsystem den Projektbeteiligten zugänglich.

Welche Schwachstellen deckt eine automatische Sourcecodeanalyse auf?

Die Bandbreite an Schwachstellen, die ein SCA-Tool aufdecken kann, ist beachtlich: Die berühmtesten wie Buffer Overflows, Command Injection, SQL Injection, Cross-Site Scripting, Session Fixation gehören dazu. Und die mit einem Blackboxtest nur sehr schwer erkennbare Second Order Codeinjection ist für das SCA-Tool eine Leichtigkeit. Naturgemäß liegen die Stärken auf der Implementierungsebene. Die Möglichkeiten eines solchen Tools auf der logischen und der semantischen Ebene [2] der Web Application Security dagegen sind begrenzt. Immerhin aber geht deutlich mehr als bei einem automatischen Application Scanner. So kann ein SCA Tool etwa die Passwortstärke oder den sicheren Umgang mit Passworten bewerten.

Hersteller

Marktbestimmend sind zwei Firmen, nämlich der Marktführer Fortify Software [3] mit **Fortify SCA** und die **Ounce Labs Product Suite 4.0** [4]. Die Preise führen bei typischen Umfängen in Großunternehmen schnell in den sechsstelligen Bereich hinein. Fortify geht dabei weit über die Sourcecodeanalyse hinaus, indem es das Analyseergebnis zur Instrumentalisierung der Anwendung nutzt: die Produktkomponente **Tracer** verwendet dies, um die Zugriffe eines Blackboxtests bis zur Codestelle hin zu verfolgen, die Anwendung also sozusagen durchsichtig zu machen, und der **Defender** stellt eine (nachträgliche) Härtung der Anwendung her, stattet sie also gewissermaßen mit einem eigenen Web Application Firewall aus.

Zusammengefasst:

Automatische Sourcecodeanalyse ist eine leistungsfähige und ausgereifte Technik. Sie setzt dort an, wo Kosten- und Nutzenvorteile am schnellsten und nachhaltigsten zu erreichen sind, nämlich in der Entwicklungsphase. Unternehmen sollten deren Einbindung in den SDLC unbedingt in Erwägung ziehen.

Thomas Schreiber ist Geschäftsführer der SecureNet GmbH in München und Berater für Web Application Security.

Quellen

[1] "Software Defect Reduction Top 10 List", Boehm/Basili, IEEE 01/2001
[2] „Sicher auf allen Ebenen", SecureNet/BSI, http://www.securenet.de/papers/BSI-Sechs-Ebenen-Modell.pdf
[3] http://www.fortify.com
[4] http://www.ouncelabs.com

Compliance

Compliance-Anforderungen durch **internationale Standards**

Anforderungen der Compliance sind abhängig vom Schutzgrad der zu schützenden Daten und IT-Systeme. Dabei sind international etablierte Standards maßgeblich für den zu erreichenden Stand der Technik. In diesem Artikel werden daher aufbauend auf einem IT-Risikomanagement grundlegende Maßnahmen zur Informationssicherheit und Netzwerksicherheit aufgezählt, die ein Unternehmen bzw. eine Behörde unbedingt berücksichtigen sollte.

Stand der Technik

Bei der Gewährleistung der Compliance ist neben gesetzlichen Erfordernissen und zwischen etwaigen Vertragspartnern vereinbarten Bestimmungen auch der Stand der Technik wichtig. Dieser wird gemeinhin als Entwicklungsstand technischer Systeme verstanden, der zur (vorsorgenden!) Abwehr der betrachtenden (und branchenspezifischen) Gefahren geeignet und der verantwortlichen Stelle zumutbar ist.

Eine gute Referenz für den zu beachtenden Stand der Technik liefern international etablierte Standards, zumal sich bei diesen im Zuge von Normierungsprozessen eine zunehmende Konvergenz feststellen lässt. So kann das internationale Normenwerk zum IT-Risikomanagement (noch ISO/IEC TR 13335-3), zur Informationssicherheit (ISO/IEC 17799 und ISO/IEC 27001) und zur Netzwerksicherheit (ISO/IEC 18028-1) als maßgeblich angesehen werden. Gleichwohl ist der im Einzelfall relevante Stand der Technik insbesondere abhängig vom Schutzgrad (hinsichtlich Sensibilität und Kritikalität) der zu schützenden Daten und IT-Systeme. An den aufgeführten Standards orientieren sich auch die in Deutschland verwandten Kriterien der Wirtschaftsprüfer (etwa nach IDW PS 330). Je größer ein Unternehmen oder eine Behörde ist und je umfangreicher ihre Tätigkeit ausfällt, desto ausschlaggebender werden jedoch die internationalen Standards.

IT-Risikomanagement

Ein Unternehmen bzw. eine Behörde benötigt zunächst Klarheit über ihre zu schützende Vermögenswerte. Zu diesen zählen neben Einrichtungen und IT-Systemen eben auch etwaige