Kai Jendrian

Sicherheit als Qualitätsmerkmal mit OpenSAMM

Die mangelnde Sicherheit von Anwendungen gefährdet zunehmend die Sicherheit von Informationen und Geschäftsprozessen in Unternehmen. Sicherheit muss als wichtiges Qualitätsmerkmal im Software-Entwicklungsprozess etabliert werden. Mit dem Open Software Assurance Maturity Model (OpenSAMM) steht ein offener und freier Standard zur Verbesserung des Reifegrads der Sicherheit in der Software-Entwicklung bereit.

1 Sicherheit in der Software-Entwicklung

Die Sicherheit von Anwendungen gewinnt zunehmend an Bedeutung. Zum einen haben Unternehmen die Absicherung ihrer IT-Infrastruktur vermehrt im Griff, zum anderen öffnen viele Unternehmen ihre IT durch die Bereitstellung von Diensten durch Webanwendungen oder Service-orientierten Architekturen.

Sicherheitsprobleme in Anwendungen können weitreichende Auswirkungen auf Unternehmen haben. Besonders kritisch ist ein möglicher Zugriff auf sensible interne Daten durch Unbefugte. Prominente Beispiele für lang bekannte und immer noch stark verbreitete Schwachstellen sind *Cross-Site Scripting* (XSS) und *SQL-Injection*.[1] In öffentlich verfügbaren Schwachstellendatenbanken[2] lässt sich nachvollziehen, wie verbreitet kritische Schwachstellen heute immer noch sind.

In der Regel entstehen Sicherheitsschwachstellen nicht alleine durch Fehler in der Programmierung, sondern können ihre Ursache in jeder Phase der Software-Entwicklung, von der Spezifikation bis hin zur Inbetriebnahme oder im Betrieb haben. Aus diesem Grund ist es wichtig, die Sicherheit von Software als ein wesentliches Qualitätsmerkmal zu betrachten, das in allen Phasen der Software-Entwicklung explizit berücksichtigt wird.

1.1 Sicherheit mit OWASP

Eine wichtige Ressource, in der viele Ansätze zur Verbesserung der Sicherheit von Anwendungen gebündelt sind, ist das *Open Web Application Security Project* (OWASP)[3]. Zu den zahlreichen Projekten[4] unter dem Mantel von OWASP zählen

- die bekannten OWASP Top 10[5], in denen die zehn kritischsten Sicherheitsrisiken für Webanwendungen zusammengefasst werden,
- der *Application Security Verification Standard* (ASVS)[6], der einen Prüfkatalog zu Schutzmaßnahmen von Webanwendungen bereit stellt als auch
- das *Open Software Assurance Maturity Model* (OpenSAMM)[7][8], ein Ansatz zur durchgängigen Berücksichtigung von Sicherheit in allen Phasen der Entwicklung von Software.

Dieser Standard wird in diesem Beitrag vorgestellt.

1.2 OpenSAMM: Reifegrad der Software-Entwicklung

Für Unternehmen aller Arten und Größe stellt gerade die durchgängige Berücksichtigung von Sicherheit im Entwicklungsprozess eine große Herausforderung dar. Es bietet sich gerade für kleinere und mittlere Unternehmen mit überschaubaren Software-Entwicklungsabteilungen an, sich hier eines standardisierten Ansatzes zu bedienen.

OpenSAMM bietet einen solchen Ansatz, mit überschaubarem Aufwand eine Einschätzung zu gewinnen, welchen Reifegrad die Berücksichtigung von Sicherheit als Qualitätsmerkmal in der Software-Entwicklung hat. Auf der Basis des ermittelten Reifegrades können sinnvolle Folgeschritte zur Verbesserung der eigenen Entwicklungsprozesse festgelegt und umgesetzt werden.

1 Einen guten Einstieg in Sicherheitsrisiken von Web-Anwendungen geben die OWASP Top 10 – http://owasp.de/top10

2 Siehe: http://www.exploit-db.com

3 Siehe: http://www.owasp.org.

4 Weitere hilfreiche Projekte sind der „Testing Guide", der „Code Review Guide", die „Enterprise Secruity API (ESAPI)", die Testwerkzeuge „Zed Attack Proxy (ZAP)" und „WebScarab", die zahlreichen „Security Cheat Sheets" zu Themen wie XSS oder SQL-Injection, aber auch Werkzeuge zur Ausbildung wie „WebGoat" oder „OWASP Live CD".

5 Siehe Gateway, DuD 10/2006, S. 636.

6 Siehe DuD 03/2010 „Überprüfung von Webanwendungen mit dem ‚OWASP Application Security Verification Standard 2009'".

7 „A guide to building security into software development"

8 Siehe: http://www.opensamm.org

Kai Jendrian

Security Consultant bei der Secorvo Security Consulting GmbH, lizenzierter Auditor und OWASP-Mitglied. Beratungsschwerpunkte: Information Security Management und Anwendungssicherheit.
E-Mail: kai.jendrian@secorvo.de

Unternehmenskulturen und die damit verbundenen Prozesse, gerade auch in der Software-Entwicklung, ändern sich in der Regel nur sehr langsam. Aus diesem Grund, setzt OpenSAMM auf iterative Änderungen in kleineren Schritten hin zu langfristig gesetzten Zielen. Es gibt auch keinen klar vorgegebenen Königsweg zur Verbesserung von Sicherheit, der für alle Unternehmen gleich aussehen könnte. Jedes Unternehmen muss seine Risiken bewerten und angemessene Schutzmaßnahmen in der Software-Entwicklung etablieren.

Gleichzeitig müssen sich alle Vorgaben und Prozesse zur Verbesserung der Sicherheit in der Software-Entwicklung ganz besonders auch an Menschen wenden, die in der Regel keine Profis im Bereich IT-Sicherheit sind und häufig auch kein ausgeprägtes Interesse an diesem Thema mitbringen. Aus diesem Grunde sollten sich die entwickelten Maßnahmen durch Einfachheit und Messbarkeit auszeichnen.

2 OpenSAMM

Bei OpenSAMM handelt es sich um ein Projekt, das unter dem Mantel von OWASP unter einer freien Lizenz allen Interessierten zur Nutzung zur Verfügung steht. Obwohl es verschiedene weitere Ansätze zur Verbesserung der Sicherheit in der Software-Entwicklung gibt,[9] hat sich der pragmatische Ansatz von OpenSAMM zum Einstieg bewährt, auch wenn zur Zeit der Standard OpenSAMM, anders als z. B. die Top 10, noch nicht in deutscher Sprache zur Verfügung steht. Die Auseinandersetzung mit dem Standard erfordert daher englische Sprachkenntnisse.

2.1 Der Aufbau

Der Standard ist nach vier sogenannten *Business Functions* aufgebaut, die die Kernbereiche der Software-Entwicklung abbilden. Bei den „Business Functions" handelt sich um

- *Governance* (Steuerung)
- *Construction* (Entwicklung)
- *Verification* (Prüfung)
- *Deployment* (Inbetriebnahme und Betrieb)

Unterhalb dieser vier Funktionen sind jeweils drei Säulen, sogenannte *Security Practices* angeordnet, in denen thematisch gruppierte Aktivitäten mit Relevanz für die Sicherheit in der Software-Entwicklung die Grundlagen für die Bewertung des Reifegrads der Sicherheit in der Software-Entwicklung bilden.

Die Säulen des Bereichs *Governance* bilden die *Security Practices* zu

- *Strategy & Metrics*,
- *Policy & Compliance* sowie
- *Education & Guidance*.

Der Themenbereich *Construction* stützt sich auf die Säulen

- *Threat Assessement*,
- *Security Requirements* und
- *Security Architecture*.

Die sicherheitsrelevanten Aktivitäten zu *Verification* sind zusammengefasst in den *Security Practices*

- *Design Review*
- *Code Review* und
- *Security Testing*.

Zu guter letzt bilden

- *Vulnerability Management*,
- *Environment Hardening* und
- *Operational Enablement*

die Säulen der Business Function *Deployment*.

Der stark strukturierte Aufbau zieht sich durch den gesamten Standard und erleichtert die Übersicht sowie das Verständnis der Methodik.

2.2 Das Bewertungsmodell

Nach OpenSAMM kann jede Säule im Entwicklungsprozess durch einen Reifegrad von 0 bis 3 bewertet werden. Jedem Reifegrad ist die Erfüllung von Mindestanforderungen zur jeweiligen *Security Practice* zugeordnet. Dabei wird den Reifegraden im Einzelnen die folgende Bedeutung zugemessen:

- 0: Keine relevanten Aktivitäten zur Sicherheit bei der jeweiligen *Security Practice*
- 1: Initiales Verständnis und unkoordinierte Maßnahmen zur Sicherheit
- 2: Kontrollierte Verbesserung von Wirksamkeit und Effizienz von Maßnahmen
- 3: Umfassende Beherrschung von Sicherheit in der jeweiligen *Security Practice*

Für jede Säule sind im Standard konkrete Ziele und damit verbundene Mindestmaßnahmen als Grundlage für die Bewertung des Reifegrads vorgegeben. Als Beispiel sei die Bewertung der Security Practice *Security Requirements* herausgegriffen. Zur Erreichung des Reifegrads 1 ist das Ziel *Consider security explicitly during the software requirements process* zu erreichen. Dazu sind die beiden Maßnahmen

- A. *Derive security requirements from business functionality* und
- B. *Evaluate security and compliance guidance for requirement*

im Entwicklungsprozess umzusetzen. Zusätzlich ist für den Reifegrad 2 das Ziel *Increase granularity of security requirements derived from business logic and known risks* zu erfüllen. Hierzu ist die Umsetzung der Maßnahmen

- A. *Build an acess control matrix for resources and capabilities* und
- B. *Specify security requirements based on known risks*

umzusetzen. Als vollständige Beherrschung der *Security Practice* mit dem Reifegrad 3 wird die Erreichung des Ziels *Mandate security requirements process for all software projects and third-party dependencies* angesehen. Hierunter fallen die Maßnahmen

- A. *Build security requirements into supplier agreements* und
- B. *Expand audit program for security requirements*.

Anhand dieses Beispiels wird deutlich, wie die Anforderungen mit steigendem Reifegrad deutlich schärfer werden. Diese Verschärfung geht mit einer Steigerung der Verlässlichkeit der Sicherheit einher.

Im Verlauf des Standards werden alle *Objectives*, *Security Practices* und *Activities* ausführlich erläutert. Zusätzlich werden als Grundlage für detaillierte Audits erwartete Ergebnisse und Vorschläge für Metriken zur Überprüfung der Wirksamkeit von Maßnahmen dargestellt.

9 U.a. „Building Security In Maturity Model (BSIMM)" oder „Microsoft Security Development Lifecyle (SDL)"

Abbildung 1 | OpenSAMM Selbsteinschätzung mit Hilfe von Excel-Fragebogen (http://www.opensamm.org)

OpenSAMM — Assessment Worksheet

Business Functions	Security Practices	Activities	Answer	Ratings
Governance	Strategy & Metrics	Is there a software security assurance program already in place?		0
		Do most of the business stakeholders understand your organization's risk profile?		
		Is most of your development staff aware of future plans for the assurance program?		
		Are most of your applications and resources categorized by risk?		
		Are risk ratings used to tailor the required assurance activities?		
		Does most of the organization know about what's required based on risk ratings?		
		Is per-project data for cost of assurance activities collected?		
		Does your organization regularly compare your security spend with other organizations?		
	Policy & Compliance	Do most project stakeholders know their project's compliance status?		0
		Are compliance requirements specifically considered by project teams?		
		Does the organization utilize a set of policies and standards to control software development?		
		Are project teams able to request an audit for compliance with policies and standards?		
		Are projects periodically audited to ensure a baseline of compliance with policies and standards?		
		Does the organization systematically use audits to collect and control compliance evidence?		
	Education & Guidance	Have most developers been given highlevel security awareness training?		0
		Does each project team have access to secure development best practices and guidance?		
		Are most roles in the development process given role-specific training and guidance?		
		Are most stakeholders able to pull in security coaches for use on projects?		
		Is security-related guidance centrally controlled and consistently distributed throughout the organization?		
		Are most people tested to ensure a baseline skillset for secure development practices?		
Construction	Threat Assessment	Do most projects in your organization consider and document likely threats?		0
		Does your organization understand and document the types of attackers it faces?		
		Do project teams regularly analyze functional requirements for likely abuses?		
		Do project teams use a method of rating threats for relative comparison?		
		Are stakeholders aware of relevant threats and ratings?		
		Do project teams specifically consider risk from external software?		
		Are all protection mechanisms and controls captured and mapped back to threats?		
	Security Requirements	Do most project teams specify some security requirements during development?		0
		Do project teams pull requirements from bestpractices and compliance guidance?		
		Are most stakeholders reviewing access control matrices for relevant projects?		
		Are project teams specifying requirements based on feedback from other security activities?		
		Are most stakeholders reviewing vendor agreements for security requirements?		
		Are the security requirements specified by project teams being audited?		
	Security Architecture	Are project teams provided with a list of recommended third-party components?		0
		Are most project teams aware of secure design principles and applying them?		
		Do you advertise shared security services with guidance for project teams?		
		Are project teams provided with prescriptive design patterns based on their application architecture?		
		Are project teams building software from centrally controlled platforms and frameworks?		
		Are project teams being audited for usage of secure architecture components?		
Verification	Design Review	Do project teams document the attack perimeter of software designs?		0
		Do project teams check software designs against known security risks?		
		Do most project teams specifically analyze design elements for security mechanisms?		
		Are most project stakeholders aware of how to obtain a formal design review?		
		Does the design review process incorporate detailed data-level analysis?		
		Does routine project audit require a baseline for design review results?		
	Code Review	Do most project teams have review checklists based on common problems?		0
		Are project teams generally performing review of selected high-risk code?		
		Can most project teams access automated code analysis tools to find security problems?		
		Do most stakeholders consistently require and review results from code reviews?		
		Do project teams utilize automation to check code against application-specific coding standards?		
		Does routine project audit require a baseline for code review results prior to release?		
	Security Testing	Are projects specifying some security tests based on requirements?		0
		Do most projects perform penetration tests prior to release?		
		Are most stakeholders aware of the security test status prior to release?		
		Are projects using automation to evaluate security test cases?		
		Do most projects follow a consistent process to evaluate and report on security tests to stakeholders?		
		Are security test cases comprehensively generated for application-specific logic?		
		Do routine project audits demand minimum standard results from security testing?		
Deployment	Vulnerability Management	Do most projects have a point of contact for security issues?		0
		Does your organization have an assigned security response team?		
		Are most project teams aware of their security point(s) of contact and response team(s)?		
		Does the organization utilize a consistent process for incident reporting and handling?		
		Are most project stakeholders aware of relevant security disclosures related to their software projects?		
		Are most incidents inspected for root causes to generate further recommendations?		
		Do most projects consistently collect and report data and metrics related to incidents?		
	Environment Hardening	Do the majority of projects document some requirements for the operational environment?		0
		Do most projects check for security updates to third-party software components?		
		Is a consistent process used to apply upgrades and patches to critical dependencies?		
		Do most project leverage automation to check application and environment health?		
		Are stakeholders aware of options for additional tools to protect software while running in operations?		
		Does routine audit check most projects for baseline environment health?		
	Operational Enablement	Do you deliver security notes with the majority of software releases?		0
		Are security-related alerts and error conditions documented for most projects?		
		Are most project utilizing a change management process that's well understood?		
		Do project teams deliver an operational security guide with each product release?		
		Are most projects being audited to check each release for appropriate operational security information?		
		Is code signing routinely performed on software components using a consistent process?		

2.3 Das Scoring

Ein schneller Einstieg mit OpenSAMM erfolgt in der Regel durch eine Evaluierung des Reifegrads auf der Grundlage eines Fragebogens mit konkreten Fragen zur Umsetzung der erforderlichen Maßnahmen zu den *Security Practices*. Hierzu wird im Projekt auch ein auf Microsoft-Excel basierendes Arbeitsblatt zur Verfügung gestellt, mit welchem auf Basis der Antworten der jeweilige Reifegrad ermittelt wird. Eine solche einfache Bewertung kann zur Ermittlung detaillierter Ergebnisse noch um zusätzliche Audits bzw. Befragungen ergänzt werden, in denen die jeweiligen *Security Practices* im Detail untersucht und gegen die erwarteten Ergebnisse sowie die im Standard vorgegebenen Erfolgsmetriken geprüft werden.

Die Zusammenstellung der Einzelbewertungen zu einem übersichtlichen Gesamtergebnis kann mit Hilfe einer *Scorecard* erfolgen, um entweder aufzuzeigen, welcher Handlungsbedarf besteht oder aber im Verlauf einer kontinuierlichen Verbesserung über die Zeit Fortschritte zu dokumentieren.

2.4 Die nächsten Schritte

Auf der Grundlage der Bestimmung des Reifegrads der Software-Entwicklung in einem Unternehmen sollten angemessene und sinnvolle Schritte festgelegt werden, um die Sicherheit zu verbessern. Die Erkenntnis, dass Rom nicht an einem Tag erbaut wurde, findet sich auch im OpenSAMM-Standard wieder. Ein Unternehmen sollte sich realistische kurz-, mittel- und langfristige Ziele für die Verbesserung des Reifegrads in den *Security Practices* setzen, die durch die spezifischen Risiken eines jeden Unternehmens bestimmt werden.

Im Rahmen verschiedener Phasen, die möglichst genau festgelegt werden sollten, werden dann schrittweise Verbesserungen des Reifegrads in ausgewählten *Security Practices* vorgenommen. Vielleicht konzentriert man sich zunächst auf die Säulen *Strategy & Metrics* und *Security Testing*, während man *Code Review* auf eine spätere Umsetzungsphase verschiebt.

Für verschiedene Unternehmenstypen macht OpenSAMM an dieser Stelle sogar konkrete Vorschläge, wie entsprechende Roadmaps aussehen könnten. So werden Profile für *Independent Software Vendor*, *Online Service Provider*, *Financial Service Organization* und *Government Organization* vorgestellt. Diese Profile eignen sich, um einen Eindruck davon zu gewinnen, wie die Verbesserung der Sicherheit in der Praxis aussehen könnte.

Bei der Gestaltung einer *Roadmap* für das eigene Unternehmen sollte aber keines der Profile blind übernommen werden. Die Gestaltung von Programmen zur Verbesserung in der Software-Entwicklung muss stark auf die spezifischen Bedürfnisse eines Unternehmens sowie dessen Prozesse und Kultur zugeschnitten sein, um erfolgreich zu sein.

Die Erklärungen zu den einzelnen Profilen können die Anpassung für das eigene Unternehmen erleichtern.

3 Die Praxis

In der Praxis hat sich gezeigt, dass die niedrige Hürde der schnellen Bewertung des Reifegrads mit plausiblen Ergebnissen häufig mit geringem Aufwand aussagekräftige Bewertungen und die Ermittlung konkreten Handlungsbedarfs erlaubt. Mit Hilfe dieser strukturierten, belastbaren Ergebnisse ist dann nachvollziehbar zu begründen, dass ein entsprechendes Programm zur Verbesserung der Sicherheit im Entwicklungsprozess erforderlich ist. Gleichzeitig lässt sich gut darstellen, wie dieses Programm aussehen kann und welche Aufwände damit verbunden sind.

Ein weiterer Vorteil eines Vorgehens nach OpenSAMM ist, dass Anforderungen anderer Standards bei Erreichen von Reifegrad 2 oder 3 erfüllt werden. So werden zahlreiche Anforderungen aus dem Standard ISO/IEC 27001:2005, insbesondere die Forderungen aus dem Anhang A.12 (*Information systems acquisition, development and maintenance*), bei Umsetzung eines Programms nach OpenSAMM erfüllt.

Fazit

OpenSAMM ist ein nützliches Werkzeug, um Sicherheit in der Software-Entwicklung als wichtiges Qualitätsmerkmal zu etablieren. Neben der Bestimmung des Reifegrads zur Sicherheit in der Software-Entwicklung stellt der Standard konkrete Ansätze zur Verbesserung für alle Entwicklungs-Phasen vor.

Die Vorgehensweise nach OpenSAMM ermöglicht einen schnellen und pragmatischen Einstieg in die dauerhafte Verbesserung der Sicherheit bei der Entwicklung von Software.