HITSSSE
Höhere IT-Sicherheit durch
Sichere Software Entwicklung

IT-Sicherheit
IN DER WIRTSCHAFT

Risiken dauerhaft mit dem Quellcode verbinden

Mit Annotationen leistungsfähige Verknüpfungen für mehr Softwaresicherheit herstellen

Gefördert durch:

Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

1 Zusammenfassung

Dieser Artikel beschreibt, wie das Risiko-Management durch spezielle Markierungen im Quelltext, sogenannten *Security Annotationen* [1], unterstützt werden kann. Dabei wird zunächst auf Risikofaktoren und die Probleme bei der Identifikation eingegangen. Anschließend wird das Konzept der *Security Annotation* vorgestellt und gezeigt, wie diese das Management von *Assets* und *potentiellen Schwachstellen* unterstützen können.

2 Die Bedeutung von Risiken im Firmenkontext

Für Unternehmen jeder Größe sind Daten, wie zum Beispiel Prozess-, Personen- oder Systemdaten wichtige Werte und damit schützenswerte Güter (engl. *Asset*). Geraten solche Daten in falsche Hände oder werden manipuliert, kann dies die Wirtschaftlichkeit und Arbeitsfähigkeit des Unternehmens beeinträchtigen [1]. Dies stellt ein Risiko für Unternehmen dar.

Einen Überblick über beeinflussende Faktoren von Risiken liefert Abbildung 1. Dabei wird verdeutlicht, dass eine vorliegende Bedrohung mit einer bestimmten Wahrscheinlichkeit wirksam wird. Doch neben diesen beiden Faktoren spielt auch das Schadenspotential eine wichtige Rolle. Dieses ist abhängig vom Wert des konkreten *Asset* und dessen Quantifizierung.

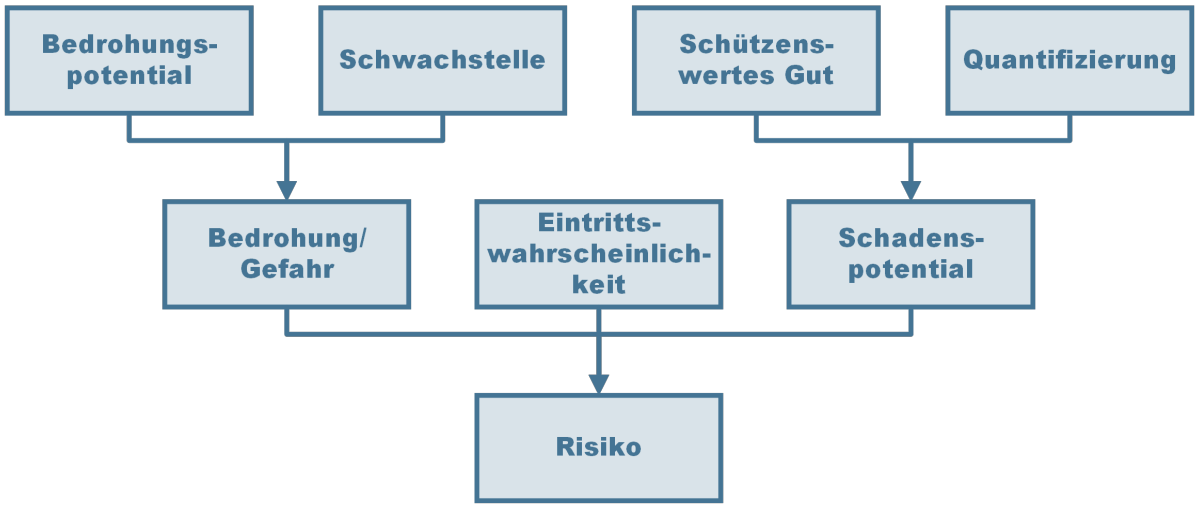

Abbildung 1: Risiko-Faktoren (nach [5])

Eine vorliegende Bedrohung wird durch eine vorhandene Schwachstelle (engl. vulnerability) und dessen Bedrohungspotential verursacht. Durch die Schwachstelle wird ein System angreifbar und für einen Angreifer ausnutzbar. Somit dient die Schwachstelle als Eintrittstor, um Sicherheitsdienste zu umgehen, zu täuschen oder unauthorisiert zu modifizieren [5].

[1] Für weitere Informationen über die Security Annotation, siehe mit Quellcode Annotationen Software sicherer machen

Schützenswertes Gut	englisch: Asset

Definition: Bestände von Objekten (auch Daten), die einen bestimmten Zweck zur Erreichung von Geschäftszielen haben [3].

Schwachstelle	englisch: Vulnerability

Definition: Sicherheitsrelevanter Fehler eines IT-Systems. Dieser führt in Kombination mit dem Bedrohungspotential dazu, dass eine Bedrohung für ein System wirksam wird [3].

3 Die Problematik mit Risiken

Das Wort Risiko hat für jedes Unternehmen eine unterschiedliche Bedeutung. Dies liegt unter anderem an der Vielzahl der Risiko-Faktoren (siehe Kapitel 2). Entscheidend für das Management von Risiken ist es, die jeweiligen Faktoren zu kennen. Dazu sollten insbesondere die Bedrohungen, sowie das daraus resultierende Schadenspotential für das jeweilige Unternehmen bekannt sein, damit Ressourcen gezielt zur Beseitigung eingesetzt werden können.

Bei der Betrachtung der Risikofaktoren wird deutlich, dass der Identifikation und Klassifizierung von *Assets* und Schwachstellen dabei eine Schlüsselrolle zukommt.

Doch wie genau werden *Assets* und Schwachstellen in einem Unternehmen identifiziert? Eine Umfrage im Rahmen des HITSSSE-Projekts durchgeführte Befragung mit fünf Unternehmen ergab, dass alle befragten Unternehmen in diesem Bereich Probleme haben. Zum einen fehlen Informationen, um *Assets* genauer zu klassifizieren, zum anderen fehlen Informationen, um *potentielle Schwachstellen* in Softwareprojekten zu kategorisieren und zu bewerten.

Aus diesem Grund wurde im Rahmen des HITSSSE-Projektes eine Lösung entwickelt, die das Risiko-Management in Softwareprojekten unterstützt. Diese Lösung wird im folgenden Kapitel beschrieben.

4 Die Verbindung von Risiken und Softwareentwicklung

Um den Schwierigkeiten bei der Identifikation und Klassifizierung von projektspezifischen *Assets* zu begegnen, hat das HITSSSE-Projekt das Konzept der ***Security Annotation*** in Quellcode entwickelt [6]. Dabei können bestimmte, sicherheitsrelevante Quellcodeabschnitte mit einer Markierung versehen und mit *Assets* verbunden werden. Diese Verbindung kann mit zusätzlichen Informationen, wie einer Beschreibung oder dem zuständigen Security-Experten angereichert werden, welche nicht direkt in den Quellcode geschrieben werden müssen, sondern gesondert gesammelt und verwaltet werden können.

Werden nun Schwachstellen bekannt, die das System bedrohen, können die markierten Quellcodebereiche auf die Bedrohungen hin überprüft und gegebenenfalls zugeordnet werden. Indirekt sind damit dann auch die bedrohten schützenswerten Güter bekannt.

Im Projekt wird der erweiterte Begriff „*potentielle Schwachstellen*“ (engl. potential vulnerability)

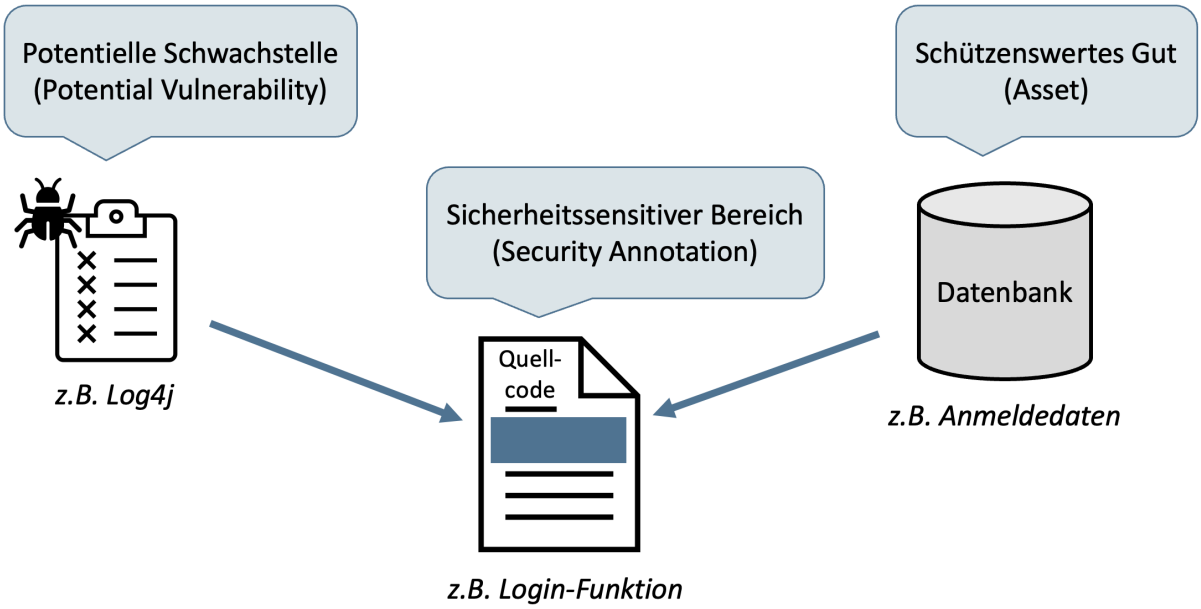

Abbildung 2: Zusammenhang zwischen Risiko-Faktoren und der Security Annotation

verwendet. Durch diese Begrifflichkeit ist es möglich, nicht nur Quellcodeabschnitte zu markieren, die eine tatsächliche Schwachstelle besitzen, sondern auch potentiell gefährdete Codeabschnitte. Dies ist vor allem dann hilfreich, wenn bei der ersten Markierung eines Codeabschnitts die davon ausgehende Gefahr noch nicht vollständig geklärt ist.

Durch die Verbindung zwischen *Assets* und *potentiellen Schwachstellen* (siehe Abbildung 2) wird der Quellcode mit zwei wichtigen Faktoren eines potentiellen Risikos verbunden (siehe Abbildung 1). Durch diese flexible und voneinander unabhängige Zuweisung lassen sich dem Quellcode zu jeder Zeit zusätzliche Informationen hinzufügen.

Definition:

Sicherheitssensitiver Bereich: Quellcode-Bereich innerhalb eines Softwareprojekts, welcher sicherheitsrelevante Elemente beinhaltet. Diese Bereiche können dabei *Assets* auf dem Quellcode repräsentieren oder die Bereiche markieren, in dem *potentielle Schwachstellen* auftreten können.

Security Annotation: Sprachunabhängige Quellcode Annotation, die es ermöglicht, *sicherheitssensitive Bereiche* dauerhaft zu markieren.

5 Integration in den Prozess des Risiko-Managements

Das Risiko-Management ist ein zyklischer und inhärenter Prozess und besteht aus unterschiedlichen Schritten [9]:

- Festlegung des zu behandelnden Gegenstands (*Asset*)
- Identifikation von Risiken des *Assets* durch eine Risiko-Analyse

- Bewertung der Risiken und der entsprechenden Behandlung
- Überwachung der Risiken, sowie Feedback und Start der nächsten Iteration (siehe Schritt 1)

Sowohl das Risiko-Management, als auch die Softwareentwicklung sind daher dynamische Prozesse, d.h. die Anforderungen und Prozesse können sich im Laufe der Zeit ändern. Dem kann durch den Einsatz der *Security Annotation* Rechnung getragen werden, da diese aus einem statischen Teil (der Annotation selbst) und einem dynamischen Teil (den sogenannten Meta-Informationen) besteht. Durch den statischen Anteil wird im Quellcode ein Codebereich dauerhaft markiert und somit der sicherheitssensitive Bereich festgelegt. Der betreffende Codebereich erhält ein für das Softwareprojekt einzigartige Identifikation. Durch die Berechnung eines Hashes des jeweiligen Bereichs, lassen sich Änderungen des Codebereichs feststellen und somit auch protokollieren. Die mit der *Security Annotation* verbundenen Meta-Informationen (*Assets und potentielle Schwachstellen*) lassen sich aktualisieren, ohne dass hierfür der Quellcode angepasst wird und stellen somit den dynamischen Anteil der beiden Prozesse dar.

6 Weiterführende Literatur

Die in diesem Artikel behandelten Aspekte haben sich stark auf den Mehrwert der *Security Annotation* für das Risiko-Management bezogen. Wie genau ein solches Risiko-Management abläuft und auf welche Aspekte hierbei geachtet werden sollte, wird unter anderem im IT-Grundschutzkatalog [4] des BSI beschrieben. Der Grundschutzkatalog stellt dabei eine zentrale Anlaufstelle für die IT-Sicherheit in Unternehmen dar und deckt technologische und organisatorische Aspekte des Risiko-Managements ab.

Für zusätzliche Informationen für das Bedrohungspotential eines Risikos wird auf das Angreifermodell von Intel TARA [13] verwiesen. In diesem Whitepaper werden Charakteristiken für einen Angreifer beschrieben, mit welchen Zielen, Methoden und Mitteln Angreifer vorgehen können. Auch wird beschrieben, welche Schutzziele dabei verletzt werden.

Eine Sammlung von aktuellen Schwachstellen für die Web-Entwicklung kann aus der OWASP TOP-10 [14] entnommen werden. In der Top-10 werden die häufigsten Schwachstellen in der Web-Entwicklung beschrieben. Auch wird dargestellt, wie sich die Schwachstellen im Laufe der letzten Jahre entwickelt haben.

Um den Quellcode nach Schwachstellen zu untersuchen, existiert unter anderem das Open-Source-Projekt DEFECTDOJO [2]. Dieses Projekt ist ebenfalls von OWASP und bietet eine Reihe von Schwachstellen-Scans. Auch wird eine Plattform geboten, um Schwachstellen im Quellcode zu verwalten.

Die *Security Annotation* dient dazu, Software inkrementell sicherer zu machen. Bei diesem Artikel wurde die *Security Annotation* verwendet, um den Mehrwert für das Risiko-Management zu verdeutlichen. Für weitere Informationen hierüber sind nachfolgende Artikel zu empfehlen. Das Konzept der *Security Annotation* wird in dem Artikel Mit Quellcode Annotationen Software sicherer machen [8] betrachtet. Hierbei wird auf die grundlegenden Aspekte eingegangen und beschrieben, dass es zwei verschiedene Perspektiven gibt. Zum einen gibt es die Sicht des Entwicklers. Dieser arbeitet direkt mit dem Quellcode und arbeitet aktiv mit der *Security Annotation*. Hierfür wurde von dem HITSSSE-Team ein Code Annotation Tool [7] entwickelt, um die Annotationen zu setzen und zu verwalten. Die zweite Perspektive ist die des Projekt-Managers. Der Fokus bei dieser Sicht liegt auf der Verwaltung und der Priorisierung der identifizierten *Assets* und *potentiellen Schwachstellen*. Hierfür wurde das Asset Management Tool [10] entwickelt,

um die Arbeit des Projekt-Managers zu erleichtern. Das Zusammenführen der Softwareentwicklung mit dem Risiko-Management ist ein komplexes Thema. Um dies zu erleichtern, wird von HITSSSE ein Demonstrator Projekt [12] bereitgestellt. Dabei handelt es sich um eine einfach gehaltene ToDo-Anwendung, an der das Konzept der *Security Annotation* veranschaulicht wird. Damit der Einstieg möglichst reibungslos gelingt, existiert eine Anleitung für den Einstieg in das Thema. Hierbei wird beschrieben, welche Programme für die Infrastruktur und für die Code Annotation notwendig sind. Auch wird aktuell an einer Sammlung von beispielhaften Assets und potentiellen Schwachstellen [11] gearbeitet. Dieser Katalog soll als Leitfaden dienen und Unternehmen den Einstieg in die *Security Annotation* erleichtern.

Impressum und Kontakt

Projekt HITSSSE – Höhere IT-Sicherheit durch Sichere Software Entwicklung

Immer mehr kleine und mittlere Unternehmen (KMUs) entwickeln Software für eigene Infrastrukturen oder Produkte. Hierbei herrscht meist ein hoher Zeitdruck und es stehen oft nur beschränkt personelle Ressourcen zur Verfügung. Oft werden inzwischen auch agile Softwareentwicklungsmethoden eingesetzt, die schnell einsetzbare Lösungen liefern sollen. Dadurch spielt die Sicherheit dieser Software oft eine untergeordnete Rolle, was sich letztendlich auch auf die IT-Sicherheit dieser Unternehmen und ihrer Kunden auswirkt. Im Fördervorhaben HITSSSE soll die IT-Sicherheit durch sichere Software Entwicklung für KMUs verbessert werden. Hierfür werden im Forschungsprojekt Handlungsempfehlungen sowie technische Hilfsmittel erstellt, die zuerst bei den assoziierten Partnern des Projekts konkret erprobt werden, um daraufhin generische Lösungsansätze für kleine und mittlere Unternehmen in Deutschland zu schaffen. Durch die Zusammenarbeit mit der Transferstelle „IT-Sicherheit in der Wirtschaft" soll die Breitenwirkung der entwickelten Angebote verstärkt werden. Gefördert wird das Projekt HITSSSE durch das Bundesministerium für Wirtschaft und Klimaschutz im Förderschwerpunkt Mittelstand-Digital. www.hitssse.de

Projektleitung

Prof. Dr.-Ing. Dominik Merli
Leiter HSA_innos
dominik.merli@hs-augsburg.de

Prof. Dr.-Ing. Alexandra Teynor
Leiterin HSA_ias
alexandra.teynor@hs-augsburg.de

Prof. Dr. Phillip Heidegger
HSA_ias
phillip.heidegger@hs-augsburg.de

Autoren

Raphael Mayr, M.Sc.
Wissenschaftlicher Mitarbeiter am HSA_ias
raphael.mayr@hs-augsburg.de

Daniel Haak, M.Sc.
Wissenschaftlicher Mitarbeiter am HSA_ias
daniel.haak@hs-augsburg.de

HSA_innos – Institut für innovative Sicherheit

HSA_innos hilft Unternehmen dabei, sich individuell zu schützen. Neben der Aus- und Weiterbildung von Sicherheitsexperten liegt der Schwerpunkt des Instituts auf der Entwicklung von Technologien und Prozessen für die IT-Sicherheit zur Anwendung in der Praxis. Zusammen mit HSA_innos schützen Unternehmen und andere Organisationen ihre Investitionen und Kunden vor digitalen Bedrohungen. Mehr Informationen zu HSA_innos finden Sie unter www.hsainnos.de.

HSA_innos
Institut für innovative Sicherheit

Hochschule Augsburg University of Applied Sciences

Institut für agile Softwareentwicklung

HSA_ias – Institut für agile Softwareentwicklung

Das Institut für agile Softwareentwicklung (HSA_ias) forscht in enger Zusammenarbeit mit Partnern aus Industrie und Wissenschaft zu den Schwerpunkten agile Softwareentwicklung, Programmiersprachen & Sicherheit, Prozessdigitalisierung sowie Anwendungen der KI. Die hierbei entstehenden Projekte decken eine breites Feld an Anwendungen ab, wie z.B. digitale Gesundheit, Produktionstechnik oder Digitalisierung der öffentlichen Verwaltung. Die Aus- und Weiterbildung von Software-IngenieurInnen für die Herausforderungen der Zukunft ist dabei ein zentrales Anliegen des Instituts.

Was ist Mittelstand Digital?

Das Mittelstand-Digital Netzwerk bietet mit den *Mittelstand-Digital Zentren*, der *Initiative IT-Sicherheit in der Wirtschaft* und *Digital Jetzt* umfassende Unterstützung bei der Digitalisierung. Kleine und mittlere Unternehmen profitieren von konkreten Praxisbeispielen und passgenauen, anbieterneutralen Angeboten zur Qualifikation und IT-Sicherheit. Das Bundesministerium für Wirtschaft und Energie ermöglicht die kostenfreie Nutzung und stellt finanzielle Zuschüsse bereit. Weitere Informationen finden Sie unter www.it-sicherheit-in-der-wirtschaft.de.

Mittelstand-Digital

Literatur

[1] DIN EN ISO/IEC 27000. Informationstechnik - Sicherheitsverfahren - Informationssicherheitsmanagementssysteme - Überblick und Technologie, 2020.

[2] Anderson, Greg and Weaver, Aaron and Tesauro, Matt and Scholten, Valentijn and Blaise, Fred. OpenSource Application Security Management, 2023. URL: https://www.defectdojo.com/.

[3] Bundesamt für Sicherheit in der Informationstechnik. IT-Grundschutz Kompendium, 2022. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/IT_Grundschutz_Kompendium_Edition2022.pdf?__blob=publicationFile&v=5#download=1.

[4] Bundesamt für Sicherheit in der Informationstechnik. IT-Grundschutz, 2023. URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/it-grundschutz_node.html.

[5] Eckert, Claudia. *IT-Sicherheit*. De Gruyter Studium. De Gruyter, Berlin and Boston, 10 edition, 2018. doi:10.1515/9783110563900.

[6] Haak, Daniel and Mayr, Raphael. Annotieren von Code mit Hilfe der Security Annotation, 2023. URL: https://cloud.hs-augsburg.de/s/SxB8QSa4tZ6KDex.

[7] Haak, Daniel and Mayr, Raphael. Beschreibung des Plugins, 2023. URL: https://www.hitssse.de.

[8] Haak, Daniel and Mayr, Raphael. Security Annotation, 2023. URL: https://cloud.hs-augsburg.de/s/6ENH2BQKqcqjJTf.

[9] Königs, Hans-Peter. *Risiko-Management-Prozesse im Unternehmen*. Vieweg+Teubner, Wiesbaden, 2009. doi:10.1007/978-3-8348-9993-4_12.

[10] Mayr, Raphael and Haak, Daniel. Asset Management Tool, 2023. URL: https://www.hitssse.de.

[11] Mayr, Raphael and Haak, Daniel. Beispielkatalog für Assets und PVs, 2023. URL: https://www.hitssse.de.

[12] Mayr, Raphael and Haak, Daniel. Demonstrator der Security Annotation, 2023. URL: https://cloud.hs-augsburg.de/s/3izjLHpBn763doF.

[13] Rosenquist, Matthew. Prioritizing information security risks with threat agent risk assessment (tara). 12 2009.

[14] van der Stock, Andrew and Glas, Brian and Smithline, Neil and Gigler, Torsten. OWASP Top Ten, 2021. URL: https://owasp.org/www-project-top-ten/.