HITSSSE
Höhere IT-Sicherheit durch
Sichere Software Entwicklung

IT-Sicherheit
IN DER WIRTSCHAFT

Security Annotation

Mit Quellcode Annotationen Software sicherer machen

Gefördert durch:

Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

1 Motivation

Laut dem Bundesamt für Sicherheit in der Informationstechnik (BSI) wurden knapp 70% der deutschen Unternehmen im Jahr 2016 und 2017 Opfer eines Cyberangriffs [3]. Eine weitere aktuelle Studie des BSI [2] besagt, dass die Sicherheit von Produkten zunehmend ein Verkaufsargument ist und eine zentrale Anforderung der Digitalisierung darstellt.
In 2020 gab es in Deutschland 89.502 Unternehmen die in der Software und IT-Service Branche tätig waren [1], davon sind ein Großteil kleine und mittelständische Unternehmen (KMU) [14]. Für diese stellt der zeitliche Aufwand, die hohen Kosten und das notwendige Wissen grundlegende Probleme in der Entwicklung sicherer Software dar [6].

Im Rahmen der Initiative “IT-Sicherheit in der Wirtschaft” wurde vom Wissenschaftlichen Institut für Infrastruktur und Kommunikationsdienste (WIK) die Studie “Aktuelle Lage der IT-Sicherheit in KMUs” [6] erstellt. Bei dieser Studie wurden KMUs befragt was aus ihrer Sicht die wichtigsten Hindernisse zur Erhöhung der IT-Sicherheit in ihrem Unternehmen sind. Die beiden prozentual am häufigsten genannt sind:

1. 76% scheuen den zeitlichen Aufwand.
2. 68% beklagen zu hohe Kosten.

Obwohl sich die Studie hauptsächlich auf die Infrastruktur und Organisation der IT-Sicherheit in KMU bezieht, sind die identifizierten Probleme allgemeiner Natur. Die beiden aufgeführten Punkte zeigen, dass KMU in der Regel wenig Zeit und Geld in die Sicherheit ihrer Produkte investieren können oder wollen. Daher ist es wichtig, dass KMU ihre Ressourcen so gezielt und effizient wie möglich einsetzen.

Der Prozess der Entwicklung sicherer Software umfasst viele Teilschritte von der Anforderungsdefinition bis zur Wartung von Geräten im Feld [11]. Der zunehmenden Komplexität in der Softwareentwicklung kann mit spezialisierten Werkzeugen begegnet werden [15]. Umso mehr muss die Frage gestellt werden, warum trotz verfügbarer Assessment-Werkzeuge (z.B. OWASP SAMM [13]) und Prozesse (z.B. Microsoft Security Development Lifecycle (SDL) [12]), der Aufwand und die Kosten von KMU als zu hoch empfunden werden?

Die beschriebenen Probleme erfordern klare und effektive Lösungen für die praktische Umsetzung eines sicheren Softwareentwicklungsprozesses und eine entsprechende Unterstützung durch effiziente und einfach zu bedienende Werkzeuge mit niedrigen Einstiegshürden.

2 Problemstellung

In einem Softwareprojekt ist es besonders wichtig, den allgemeinen Sicherheitsstatus feststellen und verfolgen zu können. Gerade in wachsenden Softwareprojekten ist es jedoch schwierig und aufwändig, den Überblick über alle Bedrohungen und die davon betroffenen Codeabschnitte zu gewinnen und über die Projektlaufzeit zu behalten. Bei fehlendem Überblick werden notwendige Maßnahmen nicht zielgerichtet eingesetzt. Beispielsweise werden Code Reviews entweder unregelmäßig durchgeführt, was Angriffen und Fehlern den Weg ebnet, oder sie werden pauschal durchgeführt, was unnötig Ressourcen kostet.

Die Entwicklung sicherer Software hängt nicht nur von der Erfahrung der Entwickler ab, sondern auch von den Werkzeugen, die sie verwenden. Die Dokumentation von Softwareprojekten erfolgt in der Regel durch Kommentare im Code sowie durch begleitende Dokumente wie z.B. eine Architekturdokumentation. Hierbei liegt der Fokus auf der Dokumentation von Methoden- und Funktionssignaturen sowie auf den architekturrelevanten Strukturen des Projekts. Die Dokumentation der Sicherheitsarchitektur der Software wird dabei jedoch häufig vernachlässigt.

In der Praxis ist es schwierig, mit Hilfe von Kommentaren den Überblick über die Schwachstellen von Codestellen und deren Zuordnung zu bestimmten Prozessen und Daten zu behalten. Außerdem sind Änderungen und deren Auswirkungen auf den Sicherheitsstatus von Codestellen über die gesamte Projektlaufzeit mit Kommentaren schwer nachvollziehbar und sehr zeitaufwendig, insbesondere weil Kommentare nicht standardisiert und automatisiert weiterverarbeitet werden. Darüber hinaus bieten Kommentare dem menschlichen Anwender keine Hilfestellung bei typischen Anpassungen des Codes.

Um diese Probleme zu lösen, hat HITSSSE das Konzept der Quellcode-Annotation für sichere Software entwickelt: die *Security Annotation*. Diese stellt eine leichtgewichtige, inkrementell einsetzbare Annotationsmöglichkeit dar, mit der eine auf It-Sicherheit abgestimmte Dokumentation erstellt werden kann.

3 Security Annotation

3.1 Konzept der Security Annotation

Die *Security Annotation* ist eine sprachunabhängige Quellcode Annotation, die es ermöglicht, sicherheitskritische Codestellen dauerhaft zu markieren. Damit können diese Codestellen überwacht und Änderungen an ihnen im Projektverlauf nachvollzogen werden. Außerdem können, den markierten Codestellen, über diese Annotationen schützenswerte Güter (*Assets*) und *potentielle Schwachstellen* zugeordnet werden. Weitere Informationen zur Rolle von *Assets* und *potentiellen Schwachstellen* finden sich in [9].

Das Unternehmen kann klein anfangen und nur wenige Annotationen setzen die mit einzelnen *Assets* und *potentiellen Schwachstellen* verknüpft sind. Im Laufe der Zeit kann das Unternehmen darauf aufbauen, indem es kontinuierlich weitere Annotationen, *Assets* und *potentielle Schwachstellen* hinzufügt. Beginnend mit den wichtigsten Codestellen kann so über die Projektlaufzeit ein vollständiges Bild des Sicherheitsstatus erzeugt werden. [5]. Dementsprechend ist der initiale Aufwand minimal, da er sich auf wenige Annotationen beschränkt.

HITSSSE stellt außerdem einen Katalog mit häufig auftretenden *Assets* und *potentielle Schwachstellen* zur Verfügung [7]. Dieser Katalog soll es KMUs ermöglichen, die Vorteile der Verknüpfung von annotiertem Quellcode mit *Assets* und *potentiellen Schwachstellen* auch ohne vorherige Risikoanalyse zu nutzen. Die vordefinierten *Assets* und *potentiellen Schwachstellen* können jederzeit durch unternehmensspezifische Einträge ergänzt werden.

Sobald der Quellcode annotiert ist, können die Entwickler und das Projektmanagement diese Annotationen nutzen. Zu diesem Zweck wurde im Rahmen des HITSSSE-Projekts ein IDE-Plugin entwickelt [4]. Durch die direkte Integration in die IDE erspart den Entwicklern ein weiteres Tool, Entwickler können somit in ihrer gewohnten Umgebung weiterarbeiten. Für das Projektmanagement wird von HITSSSE ebenfalls ein Werkzeug zur Verwaltung der *Security Annotation* [10] bereitgestellt.

3.2 Nutzen für Unternehmen

Dokumentation Durch die *Security Annotation* können die verschiedenen *Assets* und der Fortschritt ihrer Absicherung im Quellcode permanent überwacht werden. Dank der Annotationen können zu jedem Zeitpunkt im Projekt genaue Aussagen über die aktuelle Bedrohungslage durch die identifizierten *potentiellen Schwachstellen* für jedes *Asset* und die dafür verantwortlichen Codestellen getroffen werden. Dies ermöglicht gezielte Maßnahmen und damit eine ressourcenschonende Erhöhung der Produktsicherheit.

Entscheidungshilfe Die *Security Annotation* vereinfacht die automatisierte Überwachung von Änderungen an sicherheitskritischen Codestellen. Wird eine annotierte kritische Stelle geändert, lohnt sich der Aufwand eines Code-Reviews. Wird eine nicht annotierte Stelle verändert, kann auf ein Review verzichtet werden, ohne das Risiko einzugehen, eine sicherheitskritische Eigenschaft der Software zerstört zu haben.

Kommunikationshilfe Die *Security Annotation* umfasst zwei verschiedene Sichtweisen. Zum einen gibt es die Sicht des Projektmanagements und zum anderen die Sicht der Entwickler. Ersterer hat eine sogenannte Top-Down-Sicht auf das Projekt. Damit ist gemeint, dass vor allem die *Assets* und Risiken relevant sind. Bei den Entwicklern spricht man dagegen von einer Bottom-Up-Sicht. Bei ihnen liegt der Fokus auf dem Quellcode des Projektes. Die *Security Annotation* bietet nun eine Plattform, die es ermöglicht, diese beiden Aspekte zu verknüpfen und diese Verknüpfung für beide Seiten aufbereitet darzustellen.
Als Beispiel möchte das Projektmanagement basierend auf den limitierten Ressourcen ihrer Firma ein Projekt so gut wie möglich absichern. Die identifizierten *Assets* werden durch das Projektmanagement priorisiert. Die Priorisierung ergibt, dass es am wichtigsten ist, zuerst die Zugangsdaten der Kunden zu sichern. Durch die Annotationen sieht das Projektmanagement direkt, dass die Zugangsdaten an vier Codestellen von *potentiellen Schwachstellen* bedroht sind. Die Entwickler erhalten vom Projektmanagement den Auftrag die Zugangsdaten der Kunden gegen alle *potentiellen Schwachstellen* abzusichern. Durch die Annotation wissen die Entwickler direkt, welche Codestellen angepasst werden müssen und auf welche Schwachstellen sie achten müssen. Sobald die Entwickler fertig sind sieht das Projektmanagement, durch die Annotation, dass die Schwachstellen des *Assets* abgearbeitet wurden.

3.3 Technische Umsetzung

Die *Security Annotation* ist so konzipiert, dass sie unabhängig der Programmiersprache funktioniert. Sie ist als Hybrid aus Kommentaren und Zusatzinformationen (*Meta-Informationen*), die in einer Datenbank gespeichert werden, umgesetzt. Die Kommentare verankern die *Annotationen* direkt im Code und stellen eine kontinuierliche, fehlerfreie Verknüpfung der Annotation mit der betreffenden Codestelle sicher.

Es existieren drei verschiedene Varianten der Annotation, die für unterschiedlich große Codeabschnitte bestimmt sind. Damit können Codeabschnitte individuell gekennzeichnet werden. Wie in der Abbildung 1 dargestellt, besteht die Annotation aus der Variante und einer UUID zusammen. Die UUID verankert die Annotation fest im Code und stellt die Zuordnung zu den in der Datenbank gespeicherten Zusatzinformationen sicher. Dadurch wird garantiert, dass unabhängig von Änderungen am Code die Annotation immer dem korrekten Codeabschnitt zugeordnet ist.
In der Datenbank sind die eigentlichen Nutzdaten der Annotationen gespeichert. Gespeichert

Abbildung 1: Beispiel einer Annotation

werden unter anderem die Positionsdaten der Annotation und des annotierten Codebereichs. Außerdem wird ein Hash gespeichert, der aus dem Inhalt des annotierten Codes gebildet wird. Im Zusammenspiel mit den ebenfalls gespeicherten Commit-Ids der Versionsverwaltung, können so Änderungen an den als kritisch eingestuften Quellcodeabschnitten nachvollzogen werden. Zusätzlich werden die bereits erwähnten zugeordneten *Assets* und *potentielle Schwachstellen* in der Datenbank gespeichert.

4 Zusammenfassung

Zusammenfassend ist die *Security Annotation* ein neues Konzept für Entwickler und Projektmanagement, das vor allem die kleinen Unternehmen helfen soll, ihre bergrenzten Ressourcen so effizient wie möglich einzusetzen.
Die Annotationen können inkrementell eingesetzt werden und bieten bereits bei minimalem Einsatz einen Nutzen. Deswegen ist die Einstiegshürde für Unternehmen deutlich geringer als bei anderen Werkzeugen und Prozessen, die vollumfänglich eingesetzt werden müssen. Für Unternehmen, die die *Security Annotations* testen möchten, stellt HITSSSE ein Demonstratorprojekt [8] zum freien und unverbindlichen Ausprobieren der Annotationen bereit.

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

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

Raphael Mayr, M.Sc.
Wissenschaftlicher Mitarbeiter am HSA_ias
raphael.mayr@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] Bitkom. *Anzahl der Unternehmen in der IT-Branche in Deutschland von 2008 bis 2020*. Statista GmbH, 2022. URL: https://de.statista.com/statistik/daten/studie/189870/umfrage/anzahl-der-unternehmen-in-der-it-branche-in-deutschland/.

[2] Bundesamt für Sicherheit in der Inforamtionstechnik. *Die Lage der IT-Sicherhiet in Deutschland 2018*. Bundesamt für Sicherheit in der Inforamtionstechnik, 2018. URL: http://docs.dpaq.de/14069-bsi_lagebericht_2018.pdf.

[3] Bundesamt für Sicherheit in der Informationstechnik. *Cyber-Angriffe haben erhebliche Konsequenzen für die Wirtschaft*. Besucht am: 2023-02-01. URL: https://www.bsi.bund.de/DE/Presse/Pressemitteilungen/Presse2018/Cyber-Angriffe_haben_erhebliche_Konsequenzen_fuer_die_Wirtschaft_31012018.html.

[4] Haak, Daniel and Mayr, Raphael. *Code Annotation Tool*. 2023. URL: https://www.hitssse.de/.

[5] Haak, Daniel and Mayr, Raphael. *Vorgehen beim Setzen von Security Annotation*. 2023. URL: https://www.hitssse.de/.

[6] Hillebrand, Annette and Niederprüm, Antonia and Schäfer, Saskja and Thiele, Sonja and Henseler-Unger, Iris. *Aktuelle Lage der IT-Sicherheit in KMU*. WIK Wissenschaftliches Institut für Infrastruktur und Kommunikationsdienste GmbH, 2017. URL: https://epflicht.ulb.uni-bonn.de/urn/urn:nbn:de:hbz:5:2-883360.

[7] Mayr, Raphael and Haak, Daniel. *Asset/ potentielle Schwachstellen Katalog*. 2023. URL: https://www.hitssse.de/.

[8] Mayr, Raphael and Haak, Daniel. *Demonstrator der Security Annotation*. 2023. URL: https://www.hitssse.de/.

[9] Mayr, Raphael and Haak, Daniel. *Risiken dauerhaft mit Quellcode verbinden*. 2023. URL: https://www.hitssse.de/.

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

[11] McGraw, Gary. “Software Security”. In: *IEEE Security and Privacy* 2.2 (März 2004), S. 80–83. ISSN: 1540-7993. DOI: 10.1109/MSECP.2004.1281254. URL: https://doi.org/10.1109/MSECP.2004.1281254.

[12] Microsoft Corporation. *Microsoft Security Development Lifecycle.* Besucht am: 2023-02-01. URL: https://www.microsoft.com/en-us/securityengineering/sdl.

[13] OWASP Project. *Your organization’s dynamic software security strategy.* Besucht am: 2023-02-01. URL: https://owaspsamm.org/.

[14] Statistisches Bundesamt. *Kleine und mittlere Unternehmen*. Besucht am: 2023-01-25. URL: https://www.destatis.de/DE/Themen/Branchen-Unternehmen/Unternehmen/Kleine-Unternehmen-Mittlere-Unternehmen/_inhalt.html.

[15] Viega, John and McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way (paperback)(Addison-Wesley Professional Computing Series)*. 2011.