© Jakub Krechowicz | Adobe.Stock

Zertifizierung von Compiler-Tools für funktionale Sicherheit und Cybersecurity

Umfassender Schutz

Die Automobilindustrie ist durch drei große Trends gekennzeichnet: Digitalisierung, Konnektivität und hochautomatisierte Fahrzeuge. Während die Prozesse für die funktionale Sicherheit etabliert und ausgereift sind, erfordert jeder dieser drei Trends umfassende Prozesse für die Cybersecurity. Wie gelingt das Management von Compiler-induzierten Cybersecurity-Schwachstellen?

Gerard Vink

UNECE veröffentlichte 2020 die Regelung über einheitliche Bedingungen für die Genehmigung von Fahrzeugen in Hinsicht auf Cybersecurity und des Cybersecurity-Management-Systems, auch bekannt als WP.29. In der Europäischen Union, Großbritannien und in mehreren außereuropäischen Ländern wie Korea und Japan wird diese Vorschrift in die Gesetzgebung für die Typgenehmigung von Fahrzeugen aufgenommen. Die Einhaltung von Cybersecurity ist für die Sicherung des Marktzugangs somit nicht verhandelbar. Zwar wird in WP.29 die ISO/SAE 21434:2021 Road Vehicles – Cybersecurity nicht erwähnt, aber es wird davon ausgegangen, dass ein Automobilhersteller und seine Zulieferkette, die die Einhaltung dieser Norm nachweisen können, auch die Einhaltung der WP.29-Verordnung belegen können. Zudem sollte der Nachweis, dass Automobilhersteller und ihre Zulieferer die Cybersecurity-Norm eingehalten haben, Schutz in Haftungsfragen bieten.

Compiler-Qualifizierung für funktionale Sicherheit

Die Qualifizierung von Compiler-Tools für die funktionale Sicherheit ist ein gelöstes Problem. Die Norm ISO 26262 widmet ein ganzes Kapitel den Kriterien zur Bestimmung des erforderlichen Vertrauensniveaus in ein Software-Tool und bietet Methoden zur Zertifizierung des Tools, um den Nachweis zu erbringen, dass es für die Entwicklung von Software im Bereich der funktionalen Sicherheit geeignet ist.

Es werden vier verschiedene Methoden angeboten, um ein Software-Tool zu zertifizieren:

- Erhöhtes Vertrauen aus der Anwendung,
- Bewertung des Entwicklungsprozesses,
- Validierung des Software-Tools und
- Entwicklung nach einem Sicherheitsstandard.

Für höhere ASILs sind nur die letzten beiden Methoden geeignet. In der Industrie ist die Methode der Werkzeugvalidierung weitverbreitet. Im Wesentlichen bedeutet das die Anwendung von Validierungsmaßnahmen, um nachzuweisen, dass das Software-Tool die für seinen Zweck festgelegten Anforderungen erfüllt.

Um die Anforderungen der ISO 26262 an die Tool-Validierung zu erfüllen, werden zwei unterschiedliche Ansätze verfolgt. Einige Compiler-Lieferanten führen die Werkzeugvalidierung intern durch und beauftragen eine Kon-

formitätsbewertungsstelle mit der Zertifizierung der Gebrauchstauglichkeit des Tools und seiner Sicherheitsdokumentation. In diesem Fall gibt es ein zertifiziertes Compiler-Toolset. So muss man nur noch die Richtlinien aus dem Sicherheitshandbuch anwenden, um nachzuweisen, dass sein Anwendungsfall mit einem zertifizierten Anwendungsfall kompatibel ist. Andere Anbieter bieten ein zertifizierbares Compiler-Toolset und eine zertifizierte Tool-Qualifizierungsmethodik mit unterstützenden Tools und Dokumentation an. Diese lässt sich sich wie folgt zusammenfassen:

- Spezifizierung des Anwendungsfalls, um die Anforderungen zu definieren, die das Werkzeug erfüllen muss
- Auswahl der geeigneten Tests zur Überprüfung dieser Anforderungen
- Durchführung der Tests
- Analyse der Testergebnisse
- Erstellung der Sicherheitsdokumentation
- Anwendung der Leitlinien aus den Sicherheitsdokumenten.

Der Nachteil dieses letzteren Ansatzes: Es gibt eine ganze Reihe versteckter Kosten, beispielsweise das Erlernen der Qualifizierungsmethodik und der zugehörigen Werkzeuge, die Lizenzierung der erforderlichen Testsuiten, die Durchführung des Tool-Validierungsprozesses, die Interaktion mit dem Zertifizierer und schließlich die Frage, was zu tun ist, wenn die Tests fehlschlagen.

Compiler-Qualifizierung für Cybersecurity

Die ISO 21434, Abschnitt 5.4.7 Tool Management, legt fest, dass „Tools, die die Cybersecurity eines Objekts oder einer Komponente beeinflussen können, geregelt werden müssen". Ein Compiler kann das Verhalten eines Programms in einer Weise verändern (optimieren), die der Programmierer nicht vorhergesehen hat. Die strukturelle Absicht des Software-Entwicklers wird in der endgültigen Darstellung des Quellprogramms möglicherweise nicht genau wiedergegeben, und daher beeinflusst der Compiler die Cybersecurity der Software.

Im Gegensatz zur ISO 26262 werden in der ISO-Cybersecurity-Norm keine Anforderungen an die Qualifikation der Tools gestellt. Es wird keine Anleitung dazu gegeben, was genau „geregelt werden soll" bedeutet, was viel Spielraum für Interpretationen lässt:

- Die Kriterien zur Bestimmung des erforderlichen Vertrauens in ein Compiler-Toolset für die Entwicklung von Software im Bereich Cybersecurity sind unbekannt, und
- es wird keine Methode angegeben, um nachzuweisen, dass die Cybersecurity-Kriterien erfüllt wurden.

Die ISO 21434 enthält jedoch viele Verweise auf die ISO-Norm für funktionale Sicherheit. Die bewährte Tool-Validierungsmethode der ISO 26262 eignet sich für die Qualifizierung eines Compiler-Toolsets, das bei der Entwicklung einer Software zur Beanstandung von Cybersecurity-Vorschriften zum Einsatz kommt. Um diese Methode anwenden zu können, müssen die Kriterien für die Werkzeugvalidierung bei der Entwicklung von Cybersecurity-bezogener Software festgelegt werden.

Kriterien für die Tool-Zertifizierung

Analog zu den Werkzeugvalidierungskriterien für die funktionale Sicherheit lassen sich die Kriterien für die Tool-Validierung für Cybersecurity wie folgt spezifizieren:

- Die Validierungsmaßnahmen müssen den Nachweis erbringen, dass das Entwicklungs-Tool die festgelegten Cybersecurity-Anforderungen erfüllt.
- Die Cybersecurity-Risiken, die von dem Software-Tool ausgehen können, und die entsprechenden Verhaltensweisen werden zusammen mit Informationen über ihre möglichen Folgen und mit Maßnahmen zu ihrer Vermeidung oder Erkennung analysiert.
- Die Reaktion des Werkzeugs auf anomale Betriebsbedingungen muss untersucht werden.

Um das erste Kriterium zu erfüllen,

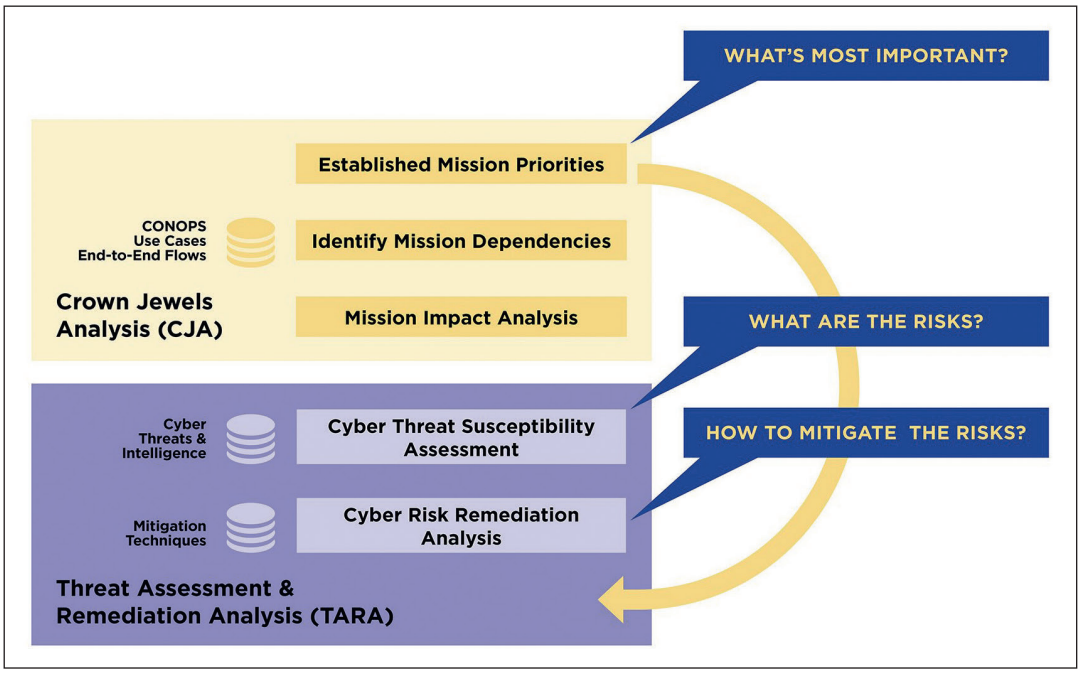

Bild 1: Der Prozessrahmen für Mission Assurance Engineering (MAE). © Tasking

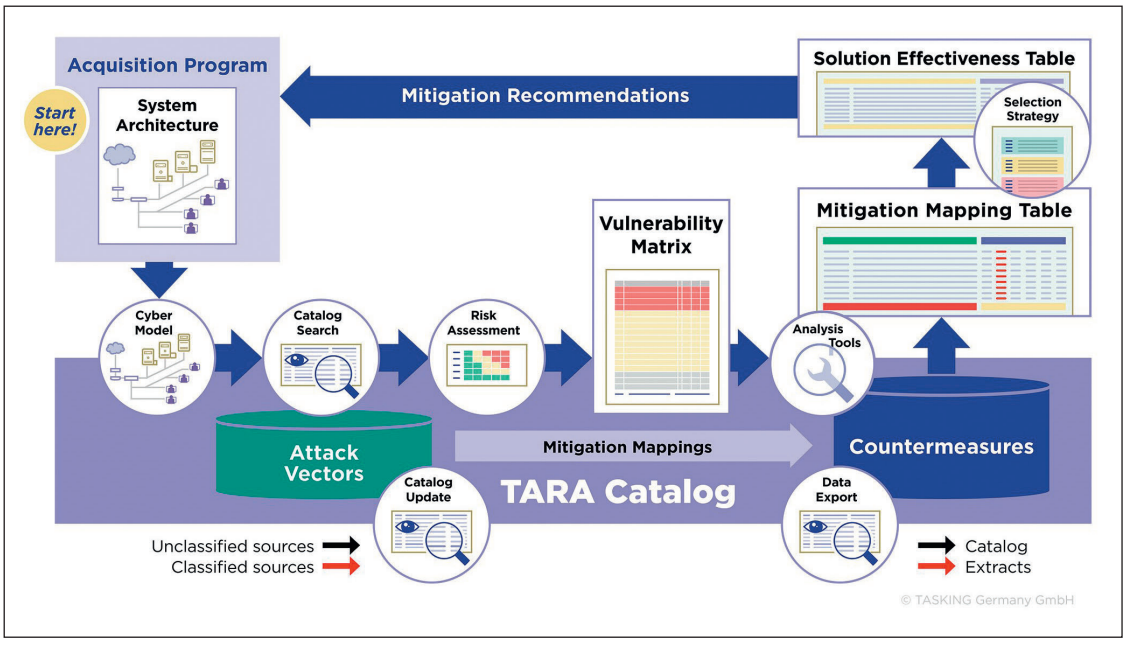

Bild 2: Arbeitsablauf der TARA-Beurteilung © Tasking

müssen die Cybersecurity-Anforderungen an das Compiler-Toolset spezifiziert werden. Zu diesem Zweck kann der vom National Cyber Security FFRDC (NCF) entwickelte Mission-Assurance-Engineering-Prozess (MAE-Prozess) zum Einsatz kommen. Dieser Prozess ist mit den Kriterien der ISO 21434, Kapitel 15, Threat Analysis and Risk Assessment Methods, kompatibel. Wenn der MAE-Prozess vom Werkzeuganbieter durchgeführt wird und die Ergebnisse in der Sicherheits- und Cybersecurity-Dokumentation des Tools beschrieben werden, muss der Anwender keine Tool-Validierung durchführen.

Das zweite Kriterium muss vom Werkzeuganwender auf Grundlage der Leitlinien des Handbuchs für Safety und Cybersecurity des Tools erfüllt werden. Schließlich muss er das mit seinem spezifischen Anwendungsfall verbundene Restrisiko bewerten, das nach Anwendung der Leitlinien verbleibt.

Um das dritte Kriterium zu erfüllen, muss der Anbieter die Reaktion des Werkzeugs analysieren. Die Ergebnisse werden in Richtlinien umgesetzt und in das Safety-&-Cybersecurity-Handbuch des Tools aufgenommen. Analog zur Erfüllung von Kriterium 2 muss der Anwender die bereitgestellten Richtlinien umsetzen und das Restrisiko für seinen Anwendungsfall abschätzen.

Identifizierung der Cybersecurity-Anforderungen

Der Prozess des Mission Assurance Engineering (MAE), der zur Ermittlung der Cybersecurity-Anforderungen im Zusammenhang mit dem Erstellungsprozess verwendet wird (**Bild 1**), bietet einen analytischen Ansatz:

- Identifizierung der für die Missionserfüllung wichtigsten Cyber-Ressourcen.
- Verständnis der Cyber-Bedrohungen und der damit verbundenen Risiken für diese Assets.
- Auswahl von Abhilfemaßnahmen, um Angriffe zu verhindern und/oder abzuwehren.

Tasking implementiert diesen MAE-Prozess mithilfe einer Fehler- und Folgenanalyse (Failure Mode and Effects Analysis, FMEA) und einer Bedrohungsanalyse (Threat Assessment and Remediation Analysis, TARA). Dabei ist zu beachten, dass der Zweck dieser Aktivitäten darin besteht, die Integrität der Software des Tool-Anwenders zu gewährleisten und nicht die Integrität des Compiler-Toolsets. Das steht im Einklang mit dem Ziel der ISO 21434, die sich mit der Cybersecurity-Perspektive bei der Entwicklung von E/E-Systemen in Straßenfahrzeugen befasst, wobei Systeme außerhalb des Fahrzeugs nicht in den Anwendungsbereich der ISO 21434 fallen. Die Integrität des Compiler-Toolsets und der Dateien, mit denen gearbeitet wird, wird durch andere Normen wie ISO/IEC 27001 – Information Security Management geregelt. Es ist wichtig, dass sowohl der Tool-Anbieter als auch der Anwender eine IT-Sicherheitsnorm einhalten.

Fehler- und Folgenanalyse

Die Fehler- und Folgenanalyse kommt zum Einsatz, um potenzielle Cybersecurity-Risiken zu ermitteln, die das Compiler-Toolset in die Software des Anwenders einbringen kann. Das Missionsziel des Compiler-Tools lässt sich wie folgt definieren: Das Verhalten der zu kompilierenden Software muss den Absichten des Anwenders sowohl unter normalen Bedingungen als auch unter den Bedingungen eines Cybersecurity-Angriffs entsprechen. Es ist zu beachten, dass die C- und C++-Sprachenspezifikation einem Compiler-Entwickler einen großen Spielraum für die Anwendung von Transformationen auf die Software bietet, die auf der Grundlage einer legalistischen Auslegung der ISO-C- und C++-Normen korrekt sind. Viele Software-Programmierer wären allerdings überrascht. Daher ist es von Vorteil, wenn die FMEA von Ingenieuren durchgeführt wird, die ein tiefes Verständnis der Anforderungen, der Architektur, des Designs und der Implementierung des Compilers haben. Für jeden identifizierten Fehlermodus müssen eine oder mehrere Maßnahmen zur Risikominderung vorgesehen werden.

Bedrohungsanalyse

Die Bedrohungsanalyse (Threat Assessment and Remediation Analysis, TARA) von MITRE ist eine Methodik zur Identifizierung und Bewertung von Cyber-Schwachstellen und zur Auswahl von Gegenmaßnahmen, die diese Schwachstellen wirksam abschwächen können. Diese Methode ist mit den Anforderungen der ISO 21434 kompatibel.
Der in **Bild 2** dargestellte TARA-Arbeits-

ablauf lässt sich wie folgt zusammenfassen: Die technischen Details des Systems kommen zum Einsatz, um ein Cyber-Modell der Systemarchitektur zu erstellen, das die Grundlage für die Katalogsuche nach plausiblen Angriffsvektoren bildet. Für diese Zwecke wird die Datenbank Common Attack Pattern Enumerations and Classifications (CAPEC) als Katalog verwendet. Anschließend wird die Liste der Angriffsvektoren gefiltert und auf der Grundlage des bewerteten Risikos eingestuft, wodurch eine Schwachstellenmatrix entsteht. Die Liste der Schwachstellen wird mit den Daten für die Zuordnung von Abhilfemaßnahmen aus dem Katalog kombiniert, um eine erste Liste von Gegenmaßnahmen zu erstellen. Diese wird anhand des bewerteten Nutzens und der Lebenszykluskosten gefiltert und eingestuft, wodurch eine Tabelle für die Zuordnung von Abhilfemaßnahmen entsteht. Anschließend werden Gegenmaßnahmen auf der Grundlage von Kosten und Risikotoleranz ausgewählt. Schließlich wird eine Tabelle zur Lösungseffektivität erstellt, in der die empfohlenen Gegenmaßnahmen aufgelistet sind und Details zur Effektivität jeder Maßnahme über den Bereich der bewerteten Schwachstellen enthält. Zusätzlich zu den Informationen aus dem CAPEC-Katalog können Informationen aus anderen Datenbanken abgerufen werden. Durch die hohe Dynamik des Bereichs Cybersecurity müssen die oben genannten Analysen regelmäßig wiederholt werden.

Ergebnis von FMEA und TARA

Die genannten Analysen zeigen, dass Compiler-induzierte Schwachstellen in Standard-Schwachstellenklassen, Seitenkanalangriffe, undefiniertes Verhalten und dauerhafte Zustandsverletzungen unterteilt werden können. Bei den zugehörigen Abhilfemaßnahmen handelt es sich entweder um Cybersecurity-Anforderungen, die vom Tool-Anbieter implementiert werden müssen, oder um Anforderungen, die vom Anwender des Werkzeugs erfüllt werden müssen.

Zu den vom Werkzeuganbieter zu implementierenden Anforderungen gehören unter anderem der Schutz vor Stack-Smashing-Angriffen durch vom Compiler platzierte Stack-Canaries, Maßnahmen zur Erkennung von Pufferüberläufen oder Vorkehrungen zur Unterstützung der Randomisierung des Speicherlayouts.

Anforderungen erfüllen

Die vom Tool-Anwender zu erfüllenden Anforderungen beziehen sich auf allgemeine Kodierungsrichtlinien und auf Richtlinien, die für das jeweils verwendete Compiler-Toolset gelten. Die Einhaltung der MISRA-Richtlinien gilt als Minimum, die Einhaltung der C/C++-Codierungsrichtlinien des SEI CERT bietet eine umfassendere Prävention gegen Risiken im Zusammenhang mit der Cybersecurity. Die Richtlinien für ein bestimmtes Compiler-Toolset hängen weitgehend von der Optimierungsphilosophie ab, die der Compiler-Anbieter anwendet. Einige Anbieter behaupten, dass Safety- und Cybersecurity-Anforderungen alle vom Compiler angewandten Optimierungen verhindern. Am anderen Ende des Spektrums wenden Compiler-Entwickler eine legalistische Interpretation des ISO-C-Standards an und betrachten Compiler-induzierte Cybersecurity-Risiken als Nebeneffekt des unzureichenden Verständnisses der Programmiersprache durch den Anwender. Optimierungen müssen allerdings keine Safety- und Cybersecurity-Risiken mit sich bringen, wenn der Compiler ausreichende Diagnoseinformationen über die angewandten Optimierungen bereitstellt, um den Anwender des Tools auf die möglichen Folgen aufmerksam zu machen. ■ (eck)

www.tasking.com

Gerard Vink studierte Maschinenbau und Informatik. Er arbeitet als Industry Specialist Product Definition bei Tasking. © Tasking