Security-Zertifizierung im IoT-Kontext

Effiziente Evaluierung durch komponentenbasiertes Software-Design

Sergey Tverdyshev, SYSGO AG

Im Internet der Dinge wird die klassische IT-Sicherheit immer mehr auf eingebettete Komponenten ausgeweitet. Charakteristisch für die Sicherheitsanforderungen vieler IoT-Systeme ist, dass Integrität und Verfügbarkeit in stärkerem Fokus stehen. Dies schlägt sich auch in Zertifizierungsstandards nieder: die klassischen Common Criteria for Information Technology Security (ISO 14508) werden durch domänenspezifische Sicherheitsstandards ergänzt, wie z.B. IEC 62443 für Industrial Control Systems, EDSA (Embedded Device Security Analysis) oder J3061 im Automobilbereich, die von einem starken Fokus von „Security for Safety“ geprägt sind.

Der Erfolg von IoT-Ansätzen liegt darin, dass mit Hilfe von offenen Standards und offenen Protokollen eingebettete COTS Komponenten vergleichsweise lose verknüpft werden. Diese Modularität spiegelt sich auch in den Zertifizierungsstandards wider: wir zeigen auf, wo und wie ein Komponenten-basiertes Softwaredesign es signifikant erleichtert, Zertifizierungsanforderungen zu genügen.

1 Inhärente Modularität von IoT

Das Internet der Dinge (Internet of Things, IoT, [1]) ist extrem heterogen, sehr dynamisch, immer verfügbar, damit jederzeit angreifbar. Das IoT besteht aus IoT-Komponenten als modulare Bausteinen.

2 Sicherheit ist in der Regel modular

Abbildung 1 zeigt das Sicherheitsmodell nach Teil 1 der CC (Abschnitt 7.1), das ebenfalls von IEC 62443-1-1 (Abschnitt .5.1) explizit aufgegriffen wird:

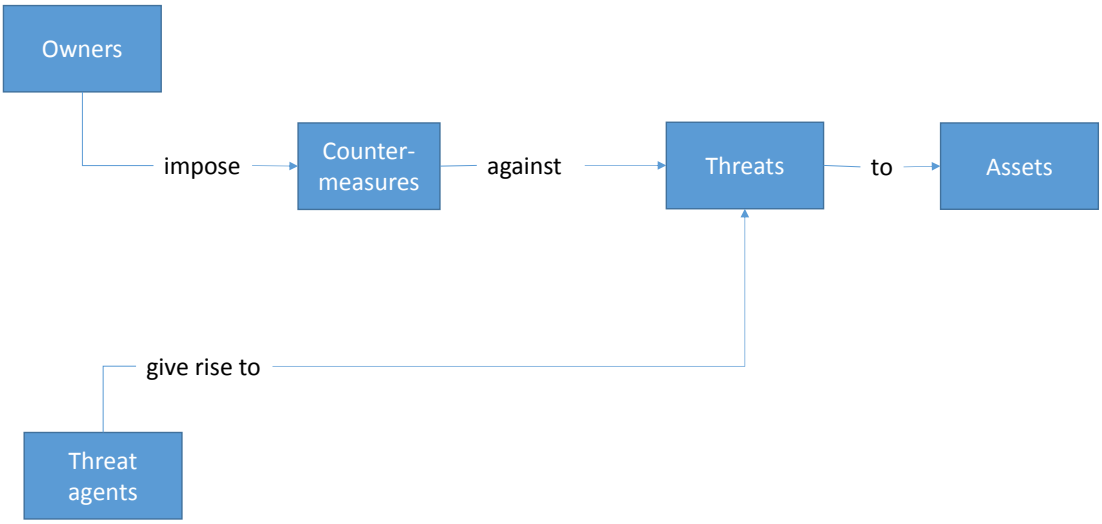

Abbildung 1: Sicherheitsmodell von CC und IEC 62443-1-1 (vereinfacht)

Abbildung 1 stellt *alle* Assets, Bedrohungen und Gegenmaßnahmen als jeweils eine Box dar. Bei *genauerer* Betrachtung sieht unserer Erfahrung nach das Bild hingegen oft eher aus wie in Abbildung 2.

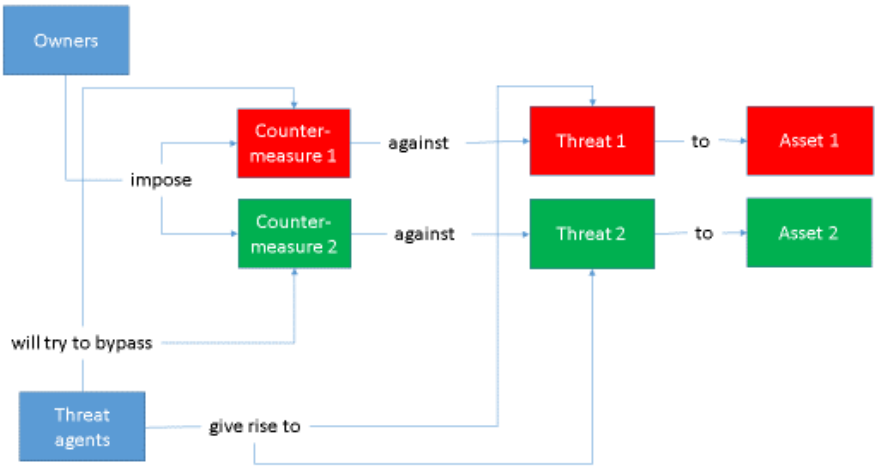

Abbildung 2: Sicherheitsmodell CC und IEC 62443-1-1 mit zwei Assets

Tabelle 1 gibt ein Beispiel für eine denkbare Belegung:

Nummer/Farbe	Asset	Bedrohung	Gegenmaßnahme
1/rot	Motorsteuerung	Verzögerung in Hinderniserkennung → Personenschaden	Echtzeitgarantien, Separierung
2/grün	Performancelogger	Verlust von Operational History	Passwort in der Webschnittstelle

Tabelle 1: Belegung von Abbildung 2 mit zwei konkreten Assets

Im Beispiel von
Tabelle 1 sind die Assets und ihre Bedrohungen von sehr unterschiedlicher Kritikalität. Anhand der konkreten Belegung von Abbildung 2 wird auch klar, dass eine *weitere* Bedrohung darin besteht, dass der Angreifer versucht, die Gegenmaßnahmen zu umgehen („will try to bypass“), z.B. könnte ein Ethernet-Zugriff auf den Performancelogger dazu missbraucht werden, auch die Motorsteuerung anzugreifen. Ein Mittel, um komplexe IT-Systeme und insbesondere IoT-Komponenten unterschiedlicher Kritikalität auf einer angemessenen Abstraktionsebene zu beherrschen, ist die Aufteilung in Sicherheitdomänen („teile und herrsche“). Eine Sicherheitsdomäne ist eine Zone, in der alle Objekte derselben

Sicherheitspolitik unterliegen [8]. Die Grenzen von Sicherheitsdomänen werden auch „trust boundaries“ genannt.

3 „Teile und Herrsche“ (TuH) in der Common Criteria Zertifizierung (z.B. security domains, domain separation)

Bei der Common Criteria for Information Technology Security Evaluation (CC) arbeitet ein Hersteller zusammen mit einer Prüfstelle an der Evaluierung eines vom Hersteller vorgeschlagenen IT-Produktes. Das Produkt kann aus Software oder Software und Hardware bestehen, IoT-Komponenten sind also ausdrücklich enthalten. Ist die Evaluierung erfolgreich, so erteilt die Zertifizierungsstelle, in Deutschland das BSI (Bundesamt für Informationstechnik), ein Zertifikat.

Die Common Criteria fordern vom Entwickler als zentralen Bestandteil der der Prüfstelle vorzulegenden Dokumentation eine Designdokumentation, in der, je nach angestrebter Evaluierungsstufe, eine ein- oder zweistufige Unterteilung in Subsysteme (einstufig) oder Subsysteme und Module (zweistufig) vorgenommen werden muss. Die Eigenschaften von Subsystemen und Module und ihre Interaktionen müssen beschrieben werden. Die Designdokumentation beschreibt auch inwiefern die Schnittstellen von Subsystemen und Modulen für den Angreifer direkt oder indirekt zugänglich sind.

Eine Sicherheitsarchitektur ist ein Mittel, die Sicherheitseigenschaften eines Systems in Hinblick auf seine Sicherheitsdomänen zu analysieren und zu dokumentieren.

Ein Sicherheitsarchitektur (ADV_ARC) nach Common Criteria hat die folgenden Punkte zu erklären:

- welche Sicherheitsdomänen hat das System? Inwieweit sind diese Sicherheitsdomäne vollständig getrennt oder dürfen sie miteinander (kontrolliert) kommunizieren? In unserem Beispiel wären der Performancelogger und die Motorsteuerung verschiedene Domänen.
- wie wird das System initialisiert?
- wie schützt sich das System selber dagegen, dass ein Angreifer versucht, es anzugreifen?
- wie schützt das System dagegen, dass es umgangen wird (in unserem Beispiel: der Webzugriff auf den Perfomancelogger kann die Motorsteuerung nicht umgehen)?

4 TuH in IEC 62443

IEC 62443 ist ein Standard für die Sicherheit von industriellen Steuerungsanlagen als Ganze (insbesondere Teile 3-1 bis 3-3) und deren Komponenten (insbesondere Teile 4-1 und 4-2). IEC 62443 wird vom Safety-Standard IEC 61508 für Security referenziert (IEC 61508 Teil 1-1 Abschnitt 7.5.2.2: „If security threats have been identified, then a vulnerability analysis shall be undertaken in order to identify security requirements. Note: Guidance is given in the 62443 series”). IEC 62443 ist zum großen Teil noch in (fortgeschrittener) Entwicklung unter dem Dach von IsaSecure.

In IEC 62443 heißen die Sicherheitsdomänen „Zonen“ und das System soll die Partitionierung in Zonen unterstützen (IEC 62443 Teil 3-3 Abschnitt SR 5.4), sowie ein Ressourcenmanagement betreiben welches robust gegenüber Angreifern ist (IEC 62443 Teil 3-3 Abschnitte SR 7.1 und SR 7.2 und IEC Teil 4-2 Abschnitte CR 7.1 und CR 7.2), also z.B. gegenüber Denial-of-Service.

In Hinblick auf Entwicklungsprozess fordert IEC 62443 Teil 4-1 SR-2 die Erstellung eines Bedrohungsmodells mit Trust boundaries, welches auch behandelt, wie Informationen über diese Trust boundaries fließen. Es wird ein Defense-in-depth-Design empfohlen (IEC 62443 Teil 4-2 SD-2). IEC 62443 Teil 4-1 SD-6 fordert darüber hinaus, dass es ein Designziel sein muss, die Angriffsfläche zu minimieren.

5 TuH in IsaSecure EDSA / SDLA / SSA

In den Standards SDLA (Prozesse), EDSA (funktionale Anlagen für Komponenten) und SSA (funktionale Anforderungen für ganze Anlagen) hat IsaSecure ein konkretes Zertifizierungsschema für die IEC 62443-Reihe entwickelt, in das auch Anregungen aus NIST 800-53 eingeflossen sind. So fordert z.B. EDSA-311 (
Tabelle 2) auf funktioneller Ebene:

FSA-RDF-1	The IACS embedded device shall provide means to enforce assigned authorizations for controlling the flow of information outside the embedded controller zone and between interconnected systems in accordance with user specific policy
FSA-RDF-2	The IACS embedded device shall separate data acquisition services, from management functionality
FSA-RDF-3	The IACS embedded device shall isolate security functions from non-security functions by means of partitions, domains, etc., including control of access to and integrity of, the hardware, software, and firmware that perform those security functions
FSA-RDF-4	The IACS embedded device shall prevent unauthorized and unintended information transfer via shared system resources where it supports connection sessions from users with different levels of access

Tabelle 2: FSA-RDF-Anforderungen von EDSA-311.

Ebenso wie in IEC 62443 Teil 4-1 fordert SDLA-312 auf der Prozessebene ein modulares Design (SDLA-DSD-1.*) und die klare Identifizierung von Trust Boundaries und Angriffsflächen (SDLA-SAD-*).

6 TuH in J3061

J3061 von SAE ist der jüngste der Standards, die wir betrachten und wir beziehen uns auf den veröffentlichten Entwurf von 2016. Das Erstellen einer Softwarearchitektur beginnt hier mit „Refine Functional Cybersecurity Concept into Technical Cybersecurity Concept“ (Abschnitt 8.4.3), und wiederum ist dabei die Isolation spezifischer Funktionen wichtig. Als Beispiel wird genannt: „Isolation/partitioning of systems that have external access (e.g. Wi-Fi, Bluetooth, OBD) from safety-critical systems and systems that can have important impacts on the operation of the vehicle.” Die Softwarearchitektur wird dann einer

Bedrohungsanlayse in Hinblick auf Vertraulichkeit, Integrität und Verfügbarkeit (Abschnitt 8.6.3) untezogen und auf Verwundbarkeit und Bedrohungen untersucht Abschnitt 8.6.4 nennt STRIDE, ASF, und DREAD als mögliche Hilfsmittel zur Bedrohungskategorisierung.

7 Gemeinsamkeiten und Ausblick

Mit der Betrachtung von Anforderungen zur Softwarearchitektur haben wir uns hier hauptsächlich auf das TuH im linken Ast des V-Modells beschränkt. Es ist klar, dass die Erfüllung dieser Anforderungen nicht nur das Design, sondern auch das Testen / Vulnerability Analyse vereinfacht.

Unter dem Aspekts des Materialverbrauchs (z.B. je Security-Domäne ein Steuergerät) erscheint eine Architektur mit klarer und vor allem nicht umgehbarer Aufgabentrennung auf den ersten Blick aufwändiger. Der Materialverbrauch kann allerdings durch Virtualisierung wie folgt kontrolliert werden:

- In Hinblick auf Netzwerkvirtualisierung sind hier vor allem neuere Entwicklungen lastbalancierender Echtzeit-Netzwerk-Standards interessant (z.B. TSN, IEEE 802.1 Qbu/Qbv), mit der Verkabelung sicher geteilt werden kann.
- In Hinblick auf CPU-Virtualisierung findet seit einigen Jahren das ursprünglich aus der sowohl Sicherheits- wie auch Material-sensitiven Avionik stammende MILS-Konzept zunehmend Anklang [9] [10].

Auf einem MILS-System sähe das Beispiel aus Abschnitt 2 wie folgt aus:

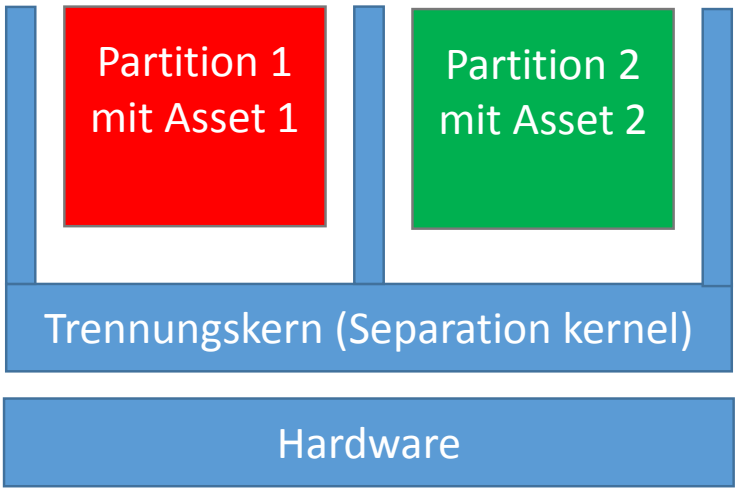

Abbildung 3: Die beiden Assets aus Abschnitt 2 auf einem MILS-System

Bei der Realisierung durch ein MILS-System wie in Abbildung 3 wird von Common Criteria, IEC 62443, EDSA, und J3061 geforderte Isolierung, Ressourcenmanagement und Informationsflusskontrolle von einem Trennungskern gestellt. Die Partitionen eines Trennungskerns sind dabei die Vorlagen für Sicherheitsdomains in den mit Systemen, die eine MILS-Architektur verwenden.

8 Danksagung

Diese Publikation beruht auf Arbeiten, die vom Bundesministerium für Bildung und Forschung, Förderprojekt BaSys, Förderkennzeichen 01 IS 16 022 J, teilweise unterstützt wurden.

9 Literatur

[1] I. Yaqoob, E. Ahmed, I. Hashem, A. Ahmed, A. Gani, M. Imran und M. Guizani, „Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges," [Online]. Available: http://dx.doi.org/10.1109/MWC.2017.1600421.

[2] R. Davidson, „Keeping hackers at bay," [Online]. Available: https://www.windpowermonthly.com/article/1442824.

[3] R. Ernst, „Automotive Ethernet – Opportunities and Pitfalls," [Online]. Available: http://www.etfa2016.org/images/keynote/keynote-ernst.pdf and https://www.youtube.com/watch?v=AEhb2CCam6o.

[4] Silicon Labs, „Silicon Labs Advances Bluetooth Smart Connectivity with Energy-Friendly SoC and Software Solution (NASDAQ:SLAB)," [Online]. Available: http://investor.silabs.com/releasedetail.cfm?releaseid=956446.

[5] A. Greenberg, „'Crash Override': The Malware That Took Down a Power Grid," [Online]. Available: https://www.wired.com/story/crash-override-malware/.

[6] Miller, *Remote Exploitation of an Unaltered Passenger Vehicle*, 2015.

[7] Forbes, *Internet of Things (IoT) Predictions From Forrester, Machina Reserach, WEF, Garntner, IDC*, 2016.

[8] B. W. Lampson, „Protection," 1971. [Online].

[9] S. Tverdyshev, H. Blasum, B. Langenstein, J. Maebe, B. De Sutter, B. Leconte, B. Triquet, K. Müller, M. Paulitsch, A. Söding-Freiherr von Blomberg und A. Tillequin, MILS Architecture, EURO-MILS, 2013.

[10] S. Tverdyshev und 3rd International Workshop on MILS, *Security by Design: Introduction to MILS*, 3rd International Workshop on MILS, 2017.

Autor
Dr.-Inf. Sergey Tverdyshev ist Head of R&D bei SYSGO AG.
Aufgabengebiet:

- Leiter der Forschungsabteilung bei SYSGO und technische Leitung nationaler und EU-weiter Projekte
- Leitung von Projekten im Security Bereich
- Verfasser von diversen wissenschaftlichen Publikationen und Präsentationen im Bereich RTOS, Hypervisor und Security
- Leitung und Durchführung technischer Workshops
- Beitrag in Standardisierungsgruppen

Profil:

- Doktor in der Ingenieurwissenschaften auf dem Gebiet „Microprocessor and SoC Verifikation“
- 10 Jahre Erfahrung in Betriebssystementwicklung/Hypervisors
- 6 Jahre Erfahrung in Safety Zertifizierung
- 5 Jahre Erfahrung in Security Zertifizierung

Kontakt
Email: sergey.tverdyshev@sysgo.com
LinkedIn: https://www.linkedin.com/in/sergey-tverdyshev-6bb75713/
Youtube (Bsp.): https://www.youtube.com/watch?v=pLfuY4SYg0c

SYSGO AG, Am Pfaffenstein 14, 55270 Klein-Winternheim
Tel.: +49-6136-9948-0