Rote Karte für IoT-Devices?

von Jonas Schuster

Bild: greenbutterfly/stock.adobe.com

Vor gut einem Jahr setzte die EU eine neue Fassung der „Radio Equipment Directive" in Kraft – und brachte damit schärfere Vorgaben für die Cybersicherheit auch von Embedded Systemen auf den Weg. Obwohl verbindliche Regelungen erst 2024 wirksam werden, sollten OEMs und Embedded Entwickler schon heute auf die Anforderungen von morgen reagieren.

Ab August 2024 verpflichtet die EU-Verordnung der „Radio Equipment Directive", kurz RED, einen Großteil aller Gerätschaften mit Datenfunkverbindung zu höherer Cybersicherheit. Zur Funk-Kommunikation-fähige Geräte dürfen dann kein Netzwerk gefährden oder betrieblich einschränken; sie müssen stattdessen aktiv personenbezogene Daten schützen und Betrug entgegenwirken. Defacto wird Updatefähigkeit ein Muss sein. Allerdings ist die Methode nicht vorgegeben, sodass auch manuelle Lösungen etwa über USB-Sticks denkbar wären. Aus praktischen Gründen dürften sich jedoch Over-the-Air (OTA) Updates über vorhandene oder neu zu schaffende Cloud-Portale durchsetzen, zumindest, wenn die eingesetzte Funktechnologie genügend Bandbreite bietet. So oder so wird Updatefähigkeit zu einer Basisanforderung für eine CE-Kennzeichnung – mit allen Konsequenzen.

Zeit zum Handeln

Auch wenn Einzelheiten wohl erst gegen Jahresende bekannt gegeben werden, sollten Anbieter von Geräten, die unter die RED fallen, deshalb schnell reagieren. Ein sinnvoller erster Schritt ist dabei, eigene Produkte auf Konformität zu bestehenden Normen wie ETSI EN 303 645 (Verbraucherprodukte) oder IEC 62443 (Industrieprodukte) zu prüfen und gegebenenfalls anzupassen. Die RED bezieht sich auf diese Normen, die nach Meinung vieler Fachleute die wahrscheinliche Grundlage für künftige harmonisierte Vorgaben darstellen. Zudem spricht vieles dafür, dass die Endkunden als Gerätebetreiber ihrerseits deren Einhaltung verlangen werden: Schließlich steigt das Interesse an Cybersecurity aufgrund einer steigenden Anzahl von Hackerangriffen.

Für Verantwortliche von Embedded-Entwicklungen – sei es in der Entwicklung oder im Produktmanagement – lohnt sich ein Blick auf beide Normen, da sich viele der Regelungen für Industrie- und Verbraucherprodukte auch auf die Embedded Security übertragen lassen. Besondere Aufmerksamkeit verdient jedoch der risikobasierte Ansatz der IE 6244. Die Sicherheitsmechanismen dieser Norm bauen wie Schichten aufeinander auf: Wird eine Schicht kompromittiert, so greift die darauffolgende. Ziel dieses Ansatzes ist die

schrittweise Verringerung der Angriffsfläche. Weil sich dies in der Praxis aber nur eingeschränkt umsetzen lässt, ist eine systematische Herangehensweise an das Thema Embedded Security ratsam.

Systematisch zur Sicherheit

Embedded Security schließt alle Tools, Prozesse und Best Practices zum Schutz der Software und Hardware von Embedded-Geräten ein. Ein sicheres Embedded-System zeichnet sich demnach durch folgende Merkmale aus:

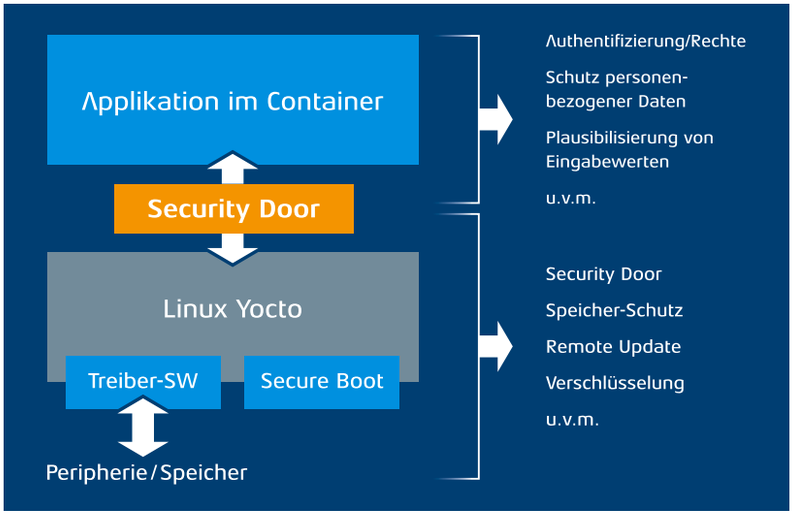
• **Vertrauenswürdige Laufzeitumgebung** (Trusted Execution Environment oder kurz TEE), im ARM-Umfeld auch TrustZone genannt: In einer TEE können nur speziell dafür freigeschaltete Applikationen ausgeführt werden, jedoch z.B. keine Hackerprogramme, denen naturgemäß die Freischaltung fehlt.
• **Partitionierte Hardware-Ressourcen:** Trennung von CPU, Cache, Speicher und Schnittstellen, um ihre Funktion möglichst unabhängig und getrennt voneinander bereitzustellen. Dies ermöglicht gegenseitigen Schutz bei Fehlern.
• **Speichersperrung:** Mit der Executable Space Protection werden explizit bestimmte Speicherbereiche als nicht rechtmäßig nutzbar gekennzeichnet. Der Missbrauch löst einen Alarm oder Gegenmaßnahmen aus, wodurch ungewollter Hacker-Code keinen Platz findet.
• **Secure Boot:** Jedes Embedded Device bootet ähnlich einem PC und sucht dazu sein kleinstes Startprogramm (Boot), in welchem der korrekte Startvorgang inklusive der zu startenden Applikationen sowie deren Reihenfolge festgelegt ist. Eine Manipulation an dieser Stelle ließe den Missbrauch des gesamten Geräts zu. Mit dem Secure Boot Feature wird das Boot Programm mittels kryptografischer Algorithmen bei jedem Start geprüft.
• **Schutz der gespeicherten Daten (Protect Data at Rest):** Dazu gehören Anwendungsdaten, Konfigurationsdaten, Sicherheitsschlüssel, aber auch Usernamen, Userrechte und Kennwörter. Diese sind explizit zu verschlüsseln und sollten in spezieller, im Controller vorhandenen, Sicherheitshardware geschützt, verschlüsselt und gespeichert werden.

Schwachstellen erkennen und beseitigen

Hilfreich bei der Entwicklung sicherer Embedded Systems ist zudem eine Analyse häufig auftretender Schwachstellen in der Software – hier drängen sich fünf Punkte auf, deren erster besonders kritisch ist:

• **Buffer Overflow:** Pufferüberlaufangriffe treten auf, wenn ein Angreifer Daten oder Code in einen Speicherpuffer schreibt, die Grenzen des Puffers über-

Bilder: Grossenbacher Systeme



Auch Container-Software ist auf Zugang zum System und Input-Daten der Hardware angewiesen. Applikations- und Systementwickler müssen deshalb für ein ganzheitliches Security-Konzept an einem Strang ziehen.

Grossenbacher Systeme hat für den Universal Controller Cybersecurity gemäß IEC 62443 umgesetzt – inklusive sicherer Kommunikationsmechanismen zwischen Applikation und System.

schreitet und benachbarte Speicheradressen zu überschreiben beginnt. Um Schäden oder eine Übernahme des Systems zu verhindern, darf die Anwendung neue Daten oder neuen ausführbaren Code keinesfalls verarbeiten.

• **Improper Input Validation:** Wenn ein Embedded-System Benutzereingaben erfordert, kann ein böswilliger Benutzer oder Prozess unerwartete Eingaben liefern, die zum Absturz einer Anwendung führen, zu viele Ressourcen verbrauchen, vertrauliche Daten preisgeben oder schädliche Befehle ausführen. Textuelle Eingaben oder Inputwerte dürfen also nur im gültigen oder plausiblen Wertebereich verarbeitet werden.

• **Improper Authentication:** Die Authentifizierung beweist, dass Benutzer oder Prozesse auch jene sind, für die sie sich ausgeben. Eine unsachgemäße Authentifizierung kann es einem Angreifer ermöglichen, die Authentifizierung zu umgehen, wiederholt zu versuchen, ein Passwort zu erraten, gestohlene Zugangsdaten zu verwenden oder ein Passwort mit einem schwachen Passwort-Wiederherstellungsmechanismus zu ändern.

• **Improper Restriction of Operations within the Bounds of a Memory Buffer:** Wenn Programmiersprache oder Betriebssystem es einem Programm ermöglichen, auf unerlaubte Speicherorte zuzugreifen, kann ein Bedrohungsakteur möglicherweise die Kontrolle über das System übernehmen oder es zum Absturz bringen. Jedes Programm darf also nur auf erlaubte Speicherbereiche zugreifen, keines Root-Rechte besitzen.

• **Information Exposure:** Sensible Informationen dürfen keinem Bedrohungsakteur zugänglich sein, Kommunikation und Speicherung müssen verschlüsselt erfolgen.

Doch was bringt diese Bestandsaufnahme OEMs und Embedded Entwicklern? Zunächst hoffentlich ein besseres Bewusstsein für Cybersicherheit. OEMs sollten drauf achten, dass ihre Entwicklungspartner in der Thematik zu Hause sind und ihrerseits entsprechende Hinweise nicht als Hindernisse oder lästige Kostenfaktoren abtun. Entwickler wiederum dürfen trotz der erforderlichen Fokussierung auf Sicherheit die Wirtschaftlichkeit nicht aus den Augen verlieren.

Sicherheit entscheidet – Kompetenz erst recht

Wer mittel- und langfristig auf Nummer Sicher gehen will, sollte darauf achten, dass seine Embedded Systeme mindestens über eine vertrauenswürdige Ausführungsumgebung (TEE), eine Secure Boot-Funktionalität sowie geschützte Speicherbereiche verfügen. Vorhandene Defizite lassen sich dabei unter Umständen durch Updates der Firm- und Software ausgleichen. In vielen Fällen sollte die RED 2024 jedoch Anlass sein, alte Embedded Systeme kritisch zu hinterfragen und gegebenenfalls durch neue, sichere und marktgängige Systeme zu ersetzen – was im industriellen Umfeld eine Vielzahl von Controllern betreffen kann. ag

Jonas Schuster
ist Mitglied der Geschäftsleitung der Grossenbacher Systeme AG.

Das Einmaleins für sichere Embedded-Software

Konsequenz bei den Updates: Jedes Updates sollte selbst verschlüsselt übertragen werden, mit einer Double Copy des OTA Signed Image, damit im Notfall ein sicheres Rollback möglich wird. Dabei sollte unbedingt der geschützten Speicherbereich der Running Code Section beachtet werden.

Container-Technologie verwenden: Wenn immer möglich Container-Technologie zur Abgrenzung zwischen der Applikations- und der Systemsoftware (Betriebssystem & Co.) verwenden. Nur gezielte Transaktionen zwischen Containern und System zulassen und die entsprechenden Schnittstellen („Türen") besonders sichern.

Ports überprüfen: Alle unnötigen Ports entfernen oder schließen und das System generell so weit wie möglich verschlanken.

Penetrations-Tests durchführen: Bei erhöhtem Sicherheitsbedürfnis Pen(etrations)- Tests durchführen beziehungsweise durchführen lassen.

Passwörter: Und zu guter Letzt: Niemals Standard-Passwörter verwenden!