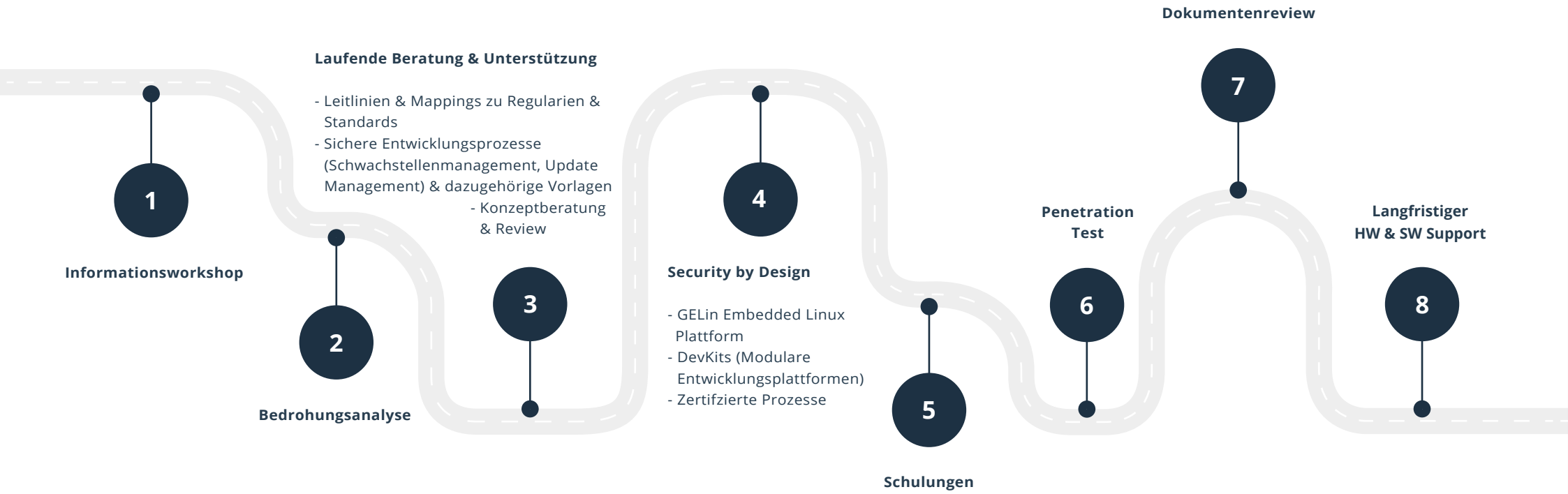
LIMES SECURITY

GINZINGER electronic systems

BEREIT FÜR DEN CYBER RESILIENCE ACT?

WHITEPAPER - TEIL 2 UMSETZUNG

SECURITY JOURNEY



LEISTUNGSBESCHREIBUNG

LIMES SECURITY

Als führender OT-Security-Experte im deutschsprachigen Raum unterstützt Limes Security Unternehmen bei der sicheren Entwicklung industrieller Produkte und Lösungen.

Zur strukturierten und praxisnahen Umsetzung des Cyber Resilience Acts bietet Limes Security ein breites Spektrum an unterstützenden Leistungen – von der ersten Orientierung bis hin zur technischen Umsetzung und Schulung:

- Informationsworkshops
- Leitlinien & Mappings
- Bedrohungsanalysen
- Konzepterstellung & Review
- Templates für Prozesse, Analysen & Best Practices
- Einführung von sicheren Produktentwicklungsprozessen
- Workshops & Trainings
- Security Tests
- Laufende Beratung & Unterstützung

Die angebotenen Leistungen greifen ineinander und ermöglichen eine strukturierte, nachvollziehbare und praxisgerechte Umsetzung der CRA-Anforderungen – abgestimmt auf individuelle Produkt- und Unternehmensbedürfnisse.

Ziel ist es, als Partner bestmöglich bei der effizienten Entwicklung eines sicheren, CRA-konformen Produkts zu unterstützen.

Im Rahmen der Limes Academy werden darüber hinaus praxisnahe OT-Security Ausbildungen und Zertifizierungen angeboten – für Fachkräfte, die Security nicht nur verstehen, sondern aktiv gestalten wollen. Zwei besonders relevante Trainings im Kontext des Cyber Resilience Acts sind:

- SEC.311 Sichere Entwicklungsprozesse für OT und (I)IoT
- SEC.331 Sichere Embedded & (I)IoT-Produkte

GINZINGER electronic systems

Ginzinger electronic systems unterstützt Hersteller bei der CRA-konformen Entwicklung sicherer Embedded-Systeme mit einer praxisbewährten, integrierten Lösungslandschaft.

– GELin (Ginzinger Embedded Linux) - Plattform

Robuste Embedded-Linux als Basis maßgeschneiderter Produkte mit Secure Boot, Partitionierung, Sicherheitsfunktionen, Software-Stücklisten und Update-Funktionen.

– Security-by-Design

Sicherheitsmechanismen wie Verschlüsselung, Zugriffskontrolle und Integritätsprüfung werden von Beginn an in Hard- und Software integriert, für eine konforme Umsetzung des Cyber Resilience Act.

– Hardware & Plattformdesign

Entwicklung und Produktion sicherer, industrietauglicher Hardware mit Fokus auf Langzeitverfügbarkeit, Manipulationsschutz und geprüften Komponenten.

– Langfristige Wartung & Pflege

Langfristig gewartete Soft- und Hardware Plattform, Sicherheits-Updates (lokal/OTA) für den gesamten Produktlebenszyklus, zuverlässig, reproduzierbar und auditierbar.

– Protokollierung & Compliance

Robustes Logging, ISO-konforme Prozesse und Unterstützung bei Audits, Risikoanalysen und Incident Response.

INHALT TEIL 2

CYBER RESILIENCE ACT
WAS HERSTELLER JETZT WISSEN MÜSSEN

Der Cyber Resilience Act (CRA) wurde seitens des Europäischen Rates im Oktober 2024 verabschiedet. Hiermit definieren sich zwei Stichtage, welche für Hersteller, Importeure und Händler von Produkten mit digitalen Elementen von Relevanz sind:

Ab 11. September 2026

gilt eine Meldepflicht. Unternehmen müssen schwerwiegende Sicherheitsvorfälle sowie aktiv ausgenutzte Schwachstellen, die die Cybersicherheit eines Produkts betreffen, an nationale oder europäische Behörden melden.

Ab 11. Dezember 2027

müssen alle technischen und organisatorischen Anforderungen des CRA erfüllt sein. Dazu zählen u. a. risikobasierte Sicherheitsmaßnahmen und die Bereitstellung von Sicherheitsupdates über den gesamten, zu definierenden Unterstützungszeitraum hinweg.

Damit gilt: Ein Produkt mit digitalen Elementen darf ab 11.12.2027 nur dann mit dem CE-Kennzeichen versehen und im EU-Binnenmarkt vertrieben werden, wenn es den Anforderungen des CRA entspricht.

Bei Verstößen drohen empfindliche Bußgelder: Bis zu 15 Millionen Euro oder 2,5 % des weltweiten Jahresumsatzes des Unternehmens – je nachdem, welcher Betrag höher ist. Weiters kann die Marktaufsichtsbehörde auch Produkte vom Markt nehmen oder Nachbesserungen fordern.

Bei Verstößen drohen Bußgelder:

15 Mio € oder 2,5%	Artikel 13 & 14
10 Mio € oder 2%	andere Artikel
15 Mio € oder 1%	Falschaussagen

Was sind Produkte mit digitalen Elementen?

Als Produkt mit „digitalen Elementen" gelten alle Hardware-, sowie Softwareprodukte und deren Datenfernverarbeitungslösungen, welche einen physischen oder logischen Datenaustausch mit anderen Produkten oder Netzwerken temporär oder auch permanent ermöglichen.

Vereinfacht gesagt: Es kann angenommen werden, der CRA gilt für reine Softwareprodukte sowie für Hardwareprodukte, wenn diese über eine Kommunikationsschnittstelle (bei Hardware z.B.: Ethernet, USB, SD-Card, PCIe, ...) verfügen. Und das sowohl für Produkte für Endkunden, als auch für Produkte, die in andere integriert werden. Nur wenige Ausnahmen sind vorgesehen.

In den folgenden Abschnitten wird detaillierter auf zentrale Elemente des CRAs eingegangen, was man als Hersteller von betroffenen Produkten nun tun muss und wie man diese Herausforderungen am Besten angeht.

WIE DIE ANFORDERUNGEN DES CRA UMSETZEN?

Um eine CE-Kennzeichnung für Produkte mit digitalen Elementen zu erhalten, müssen die im Anhang I des CRA definierten Sicherheitsanforderungen erfüllt werden. Das betrifft sowohl technische Maßnahmen im Produkt, als auch organisatorische Abläufe im Unternehmen z.B. Schwachstellenmanagement, Sicherheitsupdates, Dokumentation und Meldeprozesse.

Aktuell gibt es noch keine harmonisierten Normen, die offiziell auf den CRA angewendet werden können. Bis dahin können bewährte Standards als Orientierung dienen.

– **IEC 62443-4-1:** Anforderungen an sichere Entwicklungsprozesse
– **IEC 62443-4-2:** Sicherheitsanforderungen an Industriekomponenten

Diese helfen, zentrale CRA-Anforderungen strukturiert in Produktentwicklung und Unternehmenspraxis zu integrieren.

Wichtig ist: Es gibt keine Einheitslösung, um „sicher" im Sinne des CRA zu sein. Jede Organisation muss maßgeschneiderte Maßnahmen treffen – angepasst an ihre Produkte, Kunden und Prozesse. Bestehende Strukturen sollten gezielt um Sicherheitsaspekte erweitert werden.

Eine Zertifizierung nach IEC 62443 ist nicht zwingend erforderlich, zumal sie auch keine Rechtssicherheit bietet, konform zum CRA zu sein. Vielmehr ist sie als Wegweiser zu verstehen, um die ersten Schritte in Richtung Konformität zu machen.

Erste zentrale Schritte zur Umsetzung könnten sein:

– Initiale Risikoanalyse und Dokumentation gemäß Anhang I
– Erstellung einer Software Bill of Materials (SBoM)
– Einführung eines strukturieren Schwachstellenmanagements
– Festlegung eines Sicherheits-Update-Prozesses
– Aufbau einer Meldestelle für Sicherheitsvorfälle

Dabei kann externe Unterstützung helfen, den Umstellungsprozess fachlich fundiert und effizient nach Stand der Technik zu gestalten. Die **Experten von Limes Security** haben hier jahrelange Erfahrung und begleiteten unzählige Hersteller von Medizingeräten, Embedded bzw. IoT Lösungen, Industriekomponenten und Maschinenbauern dabei, Bedrohungsanalysen durchzuführen und sichere Entwicklungsprozesse & Strukturen im Unternehmen zu etablieren.

Reifegrad zur CRA-Konformität:

Eine der zentralen Anforderungen des Cyber Resilience Act (CRA) ist ein risikobasierter Ansatz, mit dem die Gefährdungslage eines Produkts auf ein vertretbares und akzeptables Restrisiko reduziert werden soll. Entscheidend dabei ist die systematische Erstellung und Dokumentation einer **Bedrohungsanalyse auch bekannt als Threat and Risk Assessment** (TRA).

Diese Analyse bildet die Grundlage, um gezielte Schutzmaßnahmen abzuleiten und richtig zu priorisieren:

- In der IT-Sicherheit gilt: 100 % Schutz gibt es nicht. Selbst gut gesicherte Systeme können durch bislang unbekannte Schwachstellen kompromittiert werden.
- Ziel ist daher nicht absolute Sicherheit, sondern ein nachvollziehbares Maß an Schutz, das technisch und wirtschaftlich vertretbar ist.
- Dieses Ziel wird durch das Prinzip der Defense in Depth unterstützt: Mehrere aufeinander abgestimmte Schutzmechanismen erschweren Angriffe deutlich und sorgen dafür, dass im Ernstfall zumindest frühzeitig reagiert werden kann.

Wie in der funktionalen Sicherheit (Safety) oder im Risikomanagement von Geschäftsprozessen, so folgt auch eine Bedrohungsanalyse einem strukturierten Ablauf:

- Identifikation relevanter Bedrohungsszenarien: Was kann passieren – technisch, organisatorisch, physisch?
- Bewertung nach Eintrittswahrscheinlichkeit und Schadensausmaß: Wie wahrscheinlich ist ein Szenario und welche Folgen hätte es?
- Abschätzung des Risikos: Kombination beider Werte ergibt eine priorisierte Risikoliste.
- Berücksichtigung bestehender Schutzmaßnahmen: Was ist bereits umgesetzt?
- Definition neuer Maßnahmen, falls bestehende Maßnahmen nicht ausreichen, um das Risiko auf ein akzeptables Niveau zu senken.

Diese strukturierte Risikobeurteilung schafft Transparenz im Unternehmen, unterstützt technische Entscheidungen und liefert belastbare Argumente gegenüber Auditoren oder Behörden. Sie ersetzt das bloße Bauchgefühl durch ein methodisch nachvollziehbares Vorgehen.

Für Hersteller von Embedded-Systemen bedeutet das:

- Bereits in der Architekturphase sollte durch „Security by Design“ das Risiko reduziert werden – etwa durch segmentierte Software, gesicherte Bootprozesse oder gehärtete Linux-Systeme.

- Optimal ist der Einsatz von Plattformen mit integrierten, anpassbaren Sicherheitsfunktionen – wie etwa der Ginzinger Embedded Linux Plattform (GELin).

- Limes Security unterstützt bei Bedrohungsanalysen, Sicherheitskonzepten und deren Integration in Entwicklungsprozesse, besonders in kritischen Bereichen wie Medizintechnik oder Industrie 4.0.

Die große Herausforderung bei Bedrohungsanalysen im Security-Bereich liegt in der Einschätzung der Eintrittswahrscheinlichkeit von Angriffen. Im Gegensatz zur funktionalen Sicherheit, wo Ausfallraten oder Fehlerwahrscheinlichkeiten technisch quantifiziert werden können, fehlt bei Cyber-Security eine belastbare Metrik.

> Angriffe sind alltäglich. Wer ein System mit Netzwerkzugang betreibt, sieht regelmäßig Portscans, Login-Versuche oder automatisierte Schwachstellensuche. Die Frage ist daher **nicht ob, sondern wann** ein Angriff stattfindet – und ob das System dann widerstandsfähig genug ist.

Ein weiterer wesentlicher Unterschied zur funktionalen Sicherheit ist die Dynamik von Security-Risiken:

- Neue Schwachstellen und Angriffsmethoden entstehen laufend.
- Deshalb muss eine Bedrohungsanalyse regelmäßig aktualisiert werden – es handelt sich nicht um eine Einmalaktivität.
- Annahmen und Bewertungen aus einer früheren Analyse können in wenigen Wochen überholt sein.

Auch in fortgeschrittenen Entwicklungsphasen ist eine Bedrohungsanalyse keineswegs zu spät. Zwar lassen sich Sicherheitsmaßnahmen in der frühen Designphase einfacher und kostengünstiger umsetzen, doch auch nachträgliche Schutzmaßnahmen oder gezielte Nutzungsempfehlungen sind möglich und sinnvoll. Häufig sind bereits unbewusst gewisse Schutzmechanismen vorhanden – eine strukturierte Analyse hilft dabei, diese sichtbar zu machen und gezielt zu bewerten.

Security ist zudem Teamarbeit: Hersteller sind dafür verantwortlich, geeignete Sicherheitsmaßnahmen direkt in ihre Produkte zu integrieren und deren korrekte Anwendung verständlich zu dokumentieren. Gleichzeitig tragen Betreiber, Systemintegratoren und Endnutzer eine Mitverantwortung für die Einhaltung dieser Maßnahmen, sowie für einen sachgerechten Umgang mit den Systemen. Dabei sind realistische Erwartungshaltungen entscheidend: Es muss abgewogen werden, was dem Nutzer zugemutet werden kann – und welche Schutzfunktionen das Produkt von sich aus gewährleisten sollte.

Der CRA macht zur Methodik keine Vorgaben:

- Es gibt keine vorgeschriebene Form oder Skala für Bedrohungsanalysen.
- Normen wie IEC 62443-4-1, IEC 62443-3-2 oder AAMI TIR57 können als Leitfaden dienen, sind aber nicht verpflichtend.
- Sie lassen Raum für Interpretation und müssen oft an spezifische Kontexte angepasst werden z.B. Embedded-Systeme in der Industrie oder Medizintechnik.

Zur praktischen Umsetzung gibt es eine Vielzahl an Werkzeugen:

Kostenfreie Tools
wie das Microsoft™ Threat Modelling Tool oder OWASP™ Threat Dragon bieten gute Visualisierungsmöglichkeiten.

Kommerzielle Tools
bieten oft erweiterte Funktionen, sind aber meist methodisch festgelegt.

Excel
hat sich als flexibles und praxistaugliches Werkzeug bewährt: Es lässt sich individuell anpassen, ist teamübergreifend einsetzbar und ermöglicht eine vollständige Dokumentation – inklusive Risikomatrix, Bedrohungsübersicht und Maßnahmenplanung.

Für Embedded-Systeme bedeutet das konkret:
Eine Bedrohungsanalyse gehört fest zum Entwicklungsprozess. Sie wird dabei auf die Produktarchitektur, die Anforderungen der Kunden und das jeweilige Einsatzumfeld abgestimmt.

Durch die Kombination aus technischer Plattform, Branchenwissen und einem strukturierten Analyseansatz lässt sich der CRA-Anforderungen entsprechend arbeiten – und das auf eine gleichzeitig praxisnahe und umsetzbare Weise.

GENERISCHES IOT GERÄT MIT CLOUD-ANBINDUNG

6.1 Praxisbeispiel – Security Use Case: Smart Meter Adapter

Der Smart Meter Adapter ist ein lokales Kommunikationsmodul, das zwischen einem digitalen Stromzähler (Smart Meter) und dem Heimnetzwerk installiert wird. Er ermöglicht es Anwendern, Stromverbrauchsdaten lokal und in Echtzeit auszulesen z.B. über JSON API, MQTT oder Modbus TCP. Eine Anbindung an Cloud-Dienste ist dabei bewusst ausgeschlossen, um maximale Datensouveränität und minimale Angriffsflächen zu gewährleisten.

Dabei stellen sich verschiedene, sicherheitsrelevante Herausforderungen:

– Datenschutz im Heimnetz sicherstellen
  Energiedaten gelten als sensibel. Sie dürfen nur lokal verfügbar und im Standardbetrieb nicht von außen erreichbar sein.

– Cloud-Unabhängigkeit wahren
  Eine autarke Systemarchitektur ohne Cloud-Abhängigkeiten reduziert Angriffsrisiken und erhöht die Ausfallsicherheit.

– Sichere Schnittstellenkontrolle
  Protokolle wie Modbus TCP oder JSON API müssen durch authentifizierte Zugriffe geschützt und gegen Missbrauch abgesichert sein.

– Hardware-Flexibilität ohne Sicherheitskompromisse
  Unterschiedliche Zählerprotokolle erfordern modulare Hardware. Diese muss dennoch einheitlich absicherbar und wartbar bleiben.

Daher benötigen wir für dieses Projekt gezielt Lösungen, **die Sicherheit von Anfang an mitdenken**:

– Lokal orientiertes Systemdesign
  Der Adapter ist komplett cloudfrei konzipiert und speichert keine Daten. Damit entfällt eine große Angriffsfläche bereits im Grundkonzept.

– Lokale Authentifizierung
  Zugriff auf Daten erfolgt über lokal vergebene Zugangstokens, verwaltbar über eine einfache, gesicherte Benutzeroberfläche.

– Standardmäßige WLAN-Verschlüsselung
  Die drahtlose Datenübertragung ist ab Werk abgesichert, inkl. Verschlüsselung und kontrollierbaren Zugangspunkten.

– Modulare, steckbare Hardware-Schnittstellen
  Eine klare Variantenstrategie ermöglicht den sicheren Einsatz unterschiedlicher Hardwaremodule ohne individuelle Re-Validierung jeder Kombination.

– Bewusster Verzicht auf OTA-Updatefunktionalität
  Statt aufwendiger und potenziell angreifbarer Remote-Updates nutzt der Adapter eine statische Firmwarestruktur, die Angriffsflächen minimiert.

Im nächsten Schritt wird dieses Produktbeispiel genutzt, um den Ablauf einer **praxisnahen Bedrohungsanalyse** (Threat and Risk Assessment) nach der bewährten Limes Security Methodik exemplarisch zu skizzieren. Dieser Ansatz wird typischerweise in einem fokussierten Workshopformat durchgeführt, häufig eintägig, und liefert greifbare Ergebnisse für Entwicklungs- und Managementteams.

Die Methodik umfasst **neun strukturierte Schritte**, die im Dialog **flexibel** durchlaufen werden. Die Reihenfolge ist nicht starr – vielmehr wird iterativ zwischen den Analysepunkten gewechselt, je nachdem, wo in der Diskussion gerade relevante Erkenntnisse entstehen.

ÜBER DEN SMART METER ADAPTER

Der Smart-Meter-Adapter ist eine von Österreichs E-Wirtschaft beauftragte, einheitliche Schnittstelle, die als Brücke zwischen dem Smart Meter des Netzbetreibers und dem privaten Heimnetzwerk dient. Er ermöglicht es Endkunden, ihre eigenen Zählerdaten nahezu in Echtzeit lokal auszulesen, zu visualisieren und für Energiemanagement- oder Smart-Home-Anwendungen zu nutzen. Dies erfolgt ohne direkte Verbindung zum Internet, wodurch die Datensicherheit gewährleistet bleibt.

Technisch übersetzt der Adapter die unterschiedlichen Hard- und Software-Schnittstellen der verschiedenen in Österreich eingesetzten Smart-Meter-Modelle in eine standardisierte, kundenseitige Schnittstelle. Die Daten können anschließend über WLAN im Browser oder via REST-API, Modbus TCP oder MQTT-Protokoll weiterverarbeitet werden. Damit unterstützt der Smart-Meter-Adapter nicht nur die Analyse von Verbrauch und Einspeisung, sondern auch die Regelung von Photovoltaikanlagen oder anderen Energiemanagementsystemen und schafft eine sichere Grundlage für innovative Anwendungen im Eigenheim.

DIE 9 SCHRITTE ZUR UMSETZUNG

1. SYSTEMABGRENZUNG
2. ANNAHMEN (SECURITY CONTEXT)
3. SCHNITTSTELLEN & ASSETS
4. PRODUKTSPEZIFISCHE AUSWIRKUNGEN
5. PRODUKTSPEZIFISCHE EINTRITTSWAHRSCHEINLICHKEIT
6. BEDROHUNGEN IDENTIFIZIEREN
7. RISIKOBEURTEILUNG
8. MASSNAHMEN ABLEITEN
9. WIEDERHOLEN & UPDATEN

7.1 Setup der Bedrohungsanalyse

Bevor eine Bedrohungsanalyse sinnvoll durchgeführt werden kann, ist es entscheidend, den passenden Rahmen zu definieren und alle relevanten Informationen vorab bereitzustellen. Ziel ist es, ein klares und gemeinsames Systemverständnis zu schaffen – fachlich wie organisatorisch.

Zunächst stellt sich die Frage: **Wer sollte an der Analyse beteiligt sein?**

– **Projektleitung & Produktverantwortliche**
bringen das Verständnis für Scope, Ziele und technische Abhängigkeiten mit.

– **Systemarchitekten & Security-Experten**
sind für die Bewertung der Angriffsflächen und Ableitung von Maßnahmen unerlässlich.

– **Entwickler**
kennen die konkreten technischen Umsetzungen – ideal für Detailfragen zu Schnittstellen, Protokollen und Firmware.

– **Lieferanten**
können wertvolle Informationen zu verbauten Komponenten, Modulen oder Softwarepaketen liefern.

– **Service-Techniker & Support**
wissen, wie Produkte im Feld tatsächlich verwendet und gewartet werden.

– **Personen mit Kundenkontakt** (z.B. Vertrieb, Customer Success, Support)
liefern Einblicke in Nutzungserwartungen und Akzeptanzgrenzen von Sicherheitsmaßnahmen.

Diese Rollen müssen nicht alle permanent im Workshop präsent sein – sie können auch situativ eingebunden werden, um konkrete Fragen zu klären, etwa:

- **?** Wie wird das Produkt im Alltag tatsächlich eingesetzt?
- **?** Welche Sicherheitsmaßnahmen sind Kunden zumutbar?
- **?** Was sind typische Fehlbedienungen oder Support-Fälle?

Gerade bei Embedded-Produkten – wie etwa Gateways, Sensoren oder Steuerungssystemen – ist die Einbindung von Personen mit Blick auf das reale Einsatzumfeld besonders wertvoll.

Neben dem passenden Personenkreis ist es wichtig, dass technische Informationen zur Vorbereitung vorliegen, um effizient in die Analyse einsteigen zu können (siehe Checkliste).

Diese Unterlagen helfen, ein gemeinsames Ausgangsbild zu schaffen – besonders wichtig bei interdisziplinären Teams. Je besser die Vorbereitung, desto strukturierter und aussagekräftiger wird die Bedrohungsanalyse.

- ☐ **Produktbeschreibung**
  skizziert Funktionalität und den Einsatzbereich

- ☐ **klar formulierter Product Security Context**
  Wo liegt das Risiko, welche Akteure sind relevant?
  Was wird von Endkunden erwartet?
  (Netzwerksegmentieurng, Physischer Zugangsschutz,...)

- ☐ **Systemübersicht, inklusive:**
  Beteiligte Komponenten & Subsysteme
  Physikalische Schnittstellen (z.B. UART, CAN, USB)
  Netzwerkdienste & verwendete Ports (z.B. TCP/UDP)
  API-Dokumentationen (z.B. REST, MQTT, Modbus)

- ☐ **Darstellung der Datenflüsse zwischen den Systemkomponenten**

- ☐ **Aufteilung in Netzwerkzonen oder Sicherheitsbereiche**

- ☐ **Dokumentierte Security-Anforderungen (falls vorhanden)**

- ☐ **Ergebnisse aus früheren Bedrohungsanalysen oder Audits**

Notizen:

Checkliste

7.2 Schritt 1: Systemabgrenzung

Der erste Schritt einer Bedrohungsanalyse ist die klare Definition des betrachteten Systems, auch bekannt als **„System under Consideration (SuC)“**. Ziel ist es, ein gemeinsames Verständnis darüber zu schaffen, welches Produkt, in welcher Version, mit welchen Funktionen und Schnittstellen betrachtet wird.

Warum ist eine Definition des Systems essenziell?
- Damit alle Beteiligten im Workshop dieselbe Sicht auf das System haben
- Um die Analyse nachvollziehbar zu dokumentieren – insbesondere für spätere Reviews, Nachbesserungen oder Audits
- Um etwaige Änderungen in zukünftigen Analysen klar vergleichen und bewerten zu können

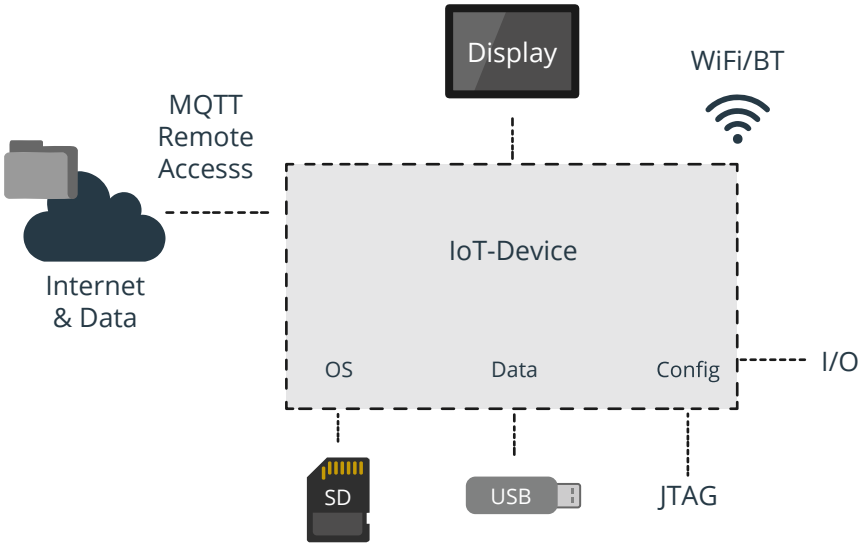
Als Beispiel dient ein generisches, fiktives Embedded IoT-Gerät, das viele realistische Sicherheitsherausforderungen abbildet.
Die nachfolgende Beschreibung visualisiert die Systemgrenzen:

- Das Gerät verfügt über mehrere I/O-Schnittstellen, an die Sensoren angeschlossen werden können – potenzielle Angriffspunkte durch manipulierte Peripherie.
- Eine JTAG-Schnittstelle wird in der Produktion zur Programmierung genutzt klassisches Risiko für Debugging-Angriffe bei nicht deaktivierter Nutzung im Feld.
- SD-Karte oder USB-Schnittstelle dienen zur lokalen Datenspeicherung und zur Bereitstellung von Updates oder Konfigurationsdaten.
- Ein Display informiert über Gerätestatus und Sensordaten – wichtig für Privacy und Physical-Security-Aspekte.
- Über WiFi wird das Gerät ins Heim- oder Firmennetz eingebunden:
    - Fernwartung erfolgt via SSH
    - Datenübermittlung erfolgt via MQTT an einen externen Server
- Eine Bluetooth-Schnittstelle ist physisch vorhanden, derzeit aber deaktiviert Künftig ist der Einsatz einer App vorgesehen

Ziel dieser Abgrenzung ist es, zu klären:

- **?** Welche Komponenten und Schnittstellen sind Teil des SuC?
- **?** Welche Funktionen sind aktiv unde welche sind vorgesehen?
- **?** Welche Kommunikationspfade und Datenflüsse sind relevant für die Security?

Es hat sich bewährt, diese Systemübersicht visuell zu dokumentieren (z.B. als Blockdiagramm oder Netzplan), mit klaren Markierungen von Ein- und Ausgängen, Kommunikationswegen und Zonen. Dadurch wird bereits in Schritt 1 die Basis für eine strukturierte Analyse gelegt.



Netzwerkdiagramm fiktives IoT Gerät

7.3 Schritt 2: Annahmen (Security Context)

Sobald das System abgegrenzt ist, folgt im zweiten Schritt die Definition des Security Contexts – also der Annahmen über die vorgesehene Nutzung, das Betriebsumfeld und das Sicherheitsverhalten des Produkts. Diese Rahmenbedingungen sind entscheidend für die spätere Bewertung der Risiken und den Umgang mit Verantwortlichkeiten.

Im Sinne des CRA bedeutet das:
- **Zweckbestimmung definieren:**
  Wie ist das Produkt vorgesehen zu funktionieren?
- **Fehlanwendungen berücksichtigen:**
  Welche unerwünschten, aber realistischen Nutzungen könnten vorkommen?
- **Betriebsanforderungen benennen:**
  Was muss der Betreiber/Kunde tun, um die sichere Nutzung zu gewährleisten?

Diese Annahmen bilden die Verbindung zwischen Herstellerverantwortung und Betreiberpflicht. Sie helfen, Risiken realistisch einzuordnen – und dienen gleichzeitig als Argumentationshilfe im Haftungsfall, wenn sie dokumentiert und kommuniziert wurden.

Solche Annahmen betreffen typischerweise:

– **Physische Einsatzumgebung**
  Ist das Gerät frei zugänglich oder in einem verschlossenen Schaltschrank verbaut?
  Gibt es Zugangskontrollen, ist der Raum alarmgesichert oder öffentlich begehbar?
– **Netzwerkanforderungen**
  Darf das Gerät direkt mit dem Internet verbunden werden?
  Wird ein Router mit Firewall erwartet?
  Ist ein VPN-Tunnel verpflichtend oder ein Air-Gap-Netz erforderlich?
– **Sicherheitsverhalten im Betrieb**
  Ist die Nutzung nur durch geschultes Personal vorgesehen?
  Wie müssen WLAN, Passwörter oder APIs konfiguriert sein?
  Gibt es Einschränkungen bei der Datenverwendung, -speicherung oder -weitergabe?

Für das Beispiel eines eingebetteten IoT-Geräts mit lokalen Schnittstellen und WLAN-Anbindung könnten folgende Annahmen gelten:

– Das Gerät darf nicht direkt mit dem Internet verbunden werden. Eine Firewall oder ein Router mit Portfilterung ist vorzuschalten.
– Eine Verbindung zu öffentlichen WLAN-Netzen ist untersagt.
– Das lokale WLAN muss mindestens WPA2-PSK verwenden – mit einem Passwort, das gängigen Komplexitätsanforderungen entspricht.
– Das Display dient rein zur Anzeige und erlaubt keine Interaktion.
– Das Produkt ist nur in physisch gesicherten Bereichen zu betreiben z.B. in Technikräumen mit Zugangsbeschränkung.
– Auf der SD-Karte gespeicherte Daten sind unverschlüsselt, weshalb dem physischen Zugriff auf das Gerät besondere Bedeutung zukommt.

> Diese Annahmen sind **rechtlich nicht bindend**, ermöglichen aber eine realistische Risikobewertung und helfen, Maßnahmen zwischen Produktdesign und Betriebspflicht abzugrenzen. Dies ist besonders wichtig bei Embedded-Produkten.

7.3 Schritt 3: Schnittstellen & Assests

Nachdem Systemgrenzen und Security-Kontext definiert sind, folgt die detaillierte Erhebung der Schnittstellen und Assets. Beide bilden die Grundlage für die spätere Bewertung möglicher Angriffsvektoren und Schutzbedarfe.
Zuerst zu den Schnittstellen:

Diese sind die Verbindungspunkte zur Außenwelt – und damit potenzielle Einfallstore für Angreifer. Deshalb müssen folgende Informationen erfasst werden:

- **?** Welche Ports und Services sind erreichbar?
- **?** Welche Protokolle werden verwendet? (z.B. MQTT, Modbus TCP, HTTPS, SSH)
- **?** Was sind die Use Cases dieser Schnittstellen? Wozu dienen sie konkret?
- **?** Auf welche Daten kann über diese Schnittstellen zugegriffen werden? Welche werden übertragen?

Für ein Embedded-Gerät wie z.B. ein lokales IoT-Gateway mit SD-Karte und WLAN bedeutet das:

– WiFi: Nutzung zur Fernwartung (SSH), Datenübertragung (MQTT)
– USB/SD-Karte: Zugriff auf Backups, Firmware-Updates, Sensordaten
– JTAG/UART: intern für Produktion und Debugging, aber sicherheitsrelevant bei physischem Zugriff
– API-Zugänge (REST, JSON, etc.): lokale Weboberfläche zur Konfiguration und Visualisierung

Sämtliche Schnittstellen sollten zusätzlich mit ihrem Sicherheitsstatus dokumentiert werden: offen, geschützt, deaktiviert, nur intern erreichbar etc.

Im zweiten Teil werden die sogenannten Assets betrachtet, also die Elemente, die besonders schutzwürdig sind oder die Angreifer gezielt kompromittieren könnten.

Typische Software- oder Daten-Assets:

– Personenbezogene Daten
– Zugangsdaten (Passwörter, Tokens, Zertifikate)
– Firmware & Bootloader
– Konfigurationsdateien
– Mess- und Prozessdaten
– Kundenspezifische Logik oder Algorithmen (geistiges Eigentum)

Doch auch **Hardwarekomponenten** können Assets sein:

– Eine integrierte COTS-Messeinheit mit hohem Wiederbeschaffungswert
– Ein HMI-Modul, dessen Ausfall die gesamte Nutzerinteraktion lahmlegt
– Sicherheitsrelevante Sensoren oder Aktoren, die für den Produktzweck essenziell sind

Wichtig ist die Frage: **Welche Folgen hätte ein Verlust, eine Manipulation oder Offenlegung dieses Assets?**

Besonders schützenswert sind Daten mit regulatorischer Relevanz, kritische Konfigurationen und ausfallgefährdete Hardware. Diese sollten strukturiert, z. B. in einer Asset- und Schnittstellenliste, erfasst werden. Diese Übersicht bildet die Grundlage für die Risikoanalyse, denn nur wer weiß, was und wie etwas geschützt werden muss, kann passende Maßnahmen definieren.

Bei unserem Beispiel sehen die Auflistungen wie folgt aus:

ZWECK	SCHNITT-STELLE	PORT/ADRES-SE/PROTOKOL	ZWECK	NOTIZ
1	USB	USB-OTG	Wechseldatenträger: BackUp, Speicherung Sensordate, Up-date-Datei	
2	JTAG		Programmierung/Debugging	deakt.
3	SD-Card	SPI	Wechseldatenträger: - BackUp - Speicherung Sensordaten - Update-Datei	
4	I/O	I2C, SIP	Anbindung Sensoren	
5	WiFi	1883/MQTT 22/ssh	Anbindung an lokales Netz mit potenziellem Internetzugriff	WPA2-PSK
6	Bluetooth	BLE	Aktuell nicht in Verwendung	
7	Display		Ausgabe Status/Daten	

ASSET	TYPE
Device	Hardware
Firmware	Software
Secrets	Daten
Konfiguration	Daten
Sensordaten	Daten
Update File	Daten
Back	Daten
Zertifikate	Daten

7.5 SCHRITT 4: PRODUKTSPEZIFISCHE AUSWIRKUNGEN

Nachdem im vorherigen Schritt die schutzwürdigen Assets identifiziert wurden, geht es nun darum, zu analysieren, welche Auswirkungen deren Verlust, Veränderung oder Offenlegung haben könnten. In diesem Schritt wird nicht die Wahrscheinlichkeit eines Angriffs betrachtet, sondern ausschließlich, wie gravierend potenzielle Schäden wären, wenn ein Szenario eintritt.

Die zentrale Frage lautet:
Welche unerwünschten Folgen kann ein Angriff oder eine Schwachstelle für das Produkt, den Betreiber oder das Unternehmen haben und wie kritisch sind diese Folgen?

Zur Bewertung bietet es sich an, die Auswirkungen in mehreren Kategorien zu betrachten:

– **Verfügbarkeit**
  Fällt das Produkt komplett oder teilweise aus?
  Welche Betriebsfolgen hat ein Ausfall z. B. bei Steuerungssystemen, Medizingeräten, Prozessüberwachung?
– **Integrität**
  Können Daten oder Konfigurationen manipuliert werden?
  Könnten dadurch Fehlverhalten oder gefährliche Zustände entstehen?

– **Vertraulichkeit**
  Werden sensible Daten offengelegt, etwa Nutzerdaten, Passwörter oder interne Algorithmen?
  Welche Datenschutz- oder Geschäftsfolgen drohen?
– **Sicherheit von Personen (Safety)**
  Ist das Produkt sicherheitskritisch (z. B. in der Medizintechnik, Landtechnik oder Heizungssteuerung)?
  Gibt es potenzielle Gefährdungen von Leib und Leben?
– **Vertragsverletzungen / regulatorische Folgen**
  Wird durch den Vorfall gegen gesetzliche Vorgaben (z. B. DSGVO, MDR) oder vertragliche Verpflichtungen verstoßen?
– **Reputation & Imageverlust**
  Wie stark würde ein öffentlich bekannter Vorfall das Vertrauen in Produkt oder Marke beeinträchtigen?

Die Auswirkungsbewertung ist dabei nicht objektiv normiert, sondern hängt stark ab von:
– Produktart (z. B. sicherheitskritisches Steuerungssystem vs. einfacher Sensor)
– Zielbranche
– „Risikokultur" des Unternehmens (welche Art und welches Maß an Schaden wird als noch akzeptabel angesehen)

Es empfiehlt sich, eine **mehrstufige Skala (z. B. 1–5) zur Bewertung der Auswirkungsschwere** zu verwenden, abgestimmt auf die spezifischen Produkt- und Branchenanforderungen. Für Kunden aus der Medizintechnik (z. B. gemäß ISO 14971 oder IEC 62304) kann hier bereits auf vorhandene Safety-Analysen zurückgegriffen werden – diese liefern oft wertvolle Vorarbeit für Sicherheits- und Verfügbarkeitskategorien.

Die Aufbereitung der Auswirkungen, unabhängig von Szenarien und Maßnahmen, ist wichtig, weil sie einen Überblick schafft und Auswirkungen vergleichbar macht. Sie zeigt, welche Bedrohungen tatsächlich kritisch sind und wo sich Investitionen in Schutzmaßnahmen lohnen. Damit entsteht auch eine akzeptierte Risikogrenze als **Grundlage für eine CRA-konforme und wirtschaftlich sinnvolle Sicherheitsstrategie**.

Beispiel: Wenn keine schweren Folgen drohen sind hohe Restrisiken unwahrscheinlich. Die Einstufung muss jedoch nachvollziehbar bleiben – ein möglicher Schaden mit Todesfolge lässt sich etwa nicht als „vernachlässigbar" bewerten.

In unserem Beispiel kann eine Einordnung wie folgt aussehen:

		Gefahr für Leib & Leben	Verfügbarkeit	Vertraulichkeit	Integrität
4	desaströs			WiFi Passwort Cloud Creden-tials	Manipulation BackUps / Firmware
3	kritisch		Sensordaten für > 5h nicht aufgezeichnet	Konfiguration	Manipulation aktueller Sensordaten
2	moderat		Sensordaten für > 1h und < 5h nicht aufgezeichnet	Sensordaten	Manipulation vergangener Sensordaten (z.B. BackUp)
1	vernach-lässigbar		Sensordaten < 1h nicht aufgezeichnet		

7.6 Schritt 5: Produktspezifische Eintrittswahrscheinlichkeit

Nachdem im vorherigen Schritt die möglichen Auswirkungen bewertet wurden, folgt nun die Einschätzung der Eintrittswahrscheinlichkeit: **Wie wahrscheinlich ist es, dass eine identifizierte Bedrohung tatsächlich real wird?**

Dieser Schritt gilt in der Praxis als besonders herausfordernd, da es in der IT -Security keine festen Wahrscheinlichkeitswerte gibt, wie etwa in der funktionalen Sicherheit (Safety), wo man mit statistisch belastbaren Ausfallraten arbeiten kann. Ein bewährter Ansatz aus der Praxis ist es, die Eintrittswahrscheinlichkeit in zwei Teilparameter zu untergliedern:

Exponiertheit:

Wie leicht ist die jeweilige Schnittstelle oder Komponente für Angreifer zugänglich?
Hier spielen die zuvor definierten Betriebsannahmen eine zentrale Rolle.

Ausnutzbarkeit:

Wie komplex oder aufwendig wäre es, die Bedrohung technisch umzusetzen?
Bewertet wird z. B., ob der Angriff spezielles Know-how, physikalischen Zugang oder gezielte Vorbereitung erfordert.

Beispiele für die Bewertung der Exponiertheit:
- Ein Gerät im versperrten Schaltschrank mit deaktivierter Debug-Schnittstelle hat eine niedrige Exponiertheit.
- Ein Produkt mit offenem WLAN-Zugangspunkt oder frei zugänglicher USB-Schnittstelle im Feld ist hoch exponiert.
- OTA-fähige Systeme oder APIs, die über das Internet erreichbar sind, gelten oft als besonders kritisch, wenn sie nicht zusätzlich abgesichert sind.

Dieser zweistufige Ansatz hilft, transparente, wiederholbare und diskutierbare Bewertungen zu erzielen – eine wichtige Grundlage für alle weiteren Risikoentscheidungen. Insbesondere bei Embedded-Produkten, deren physikalische Einsatzbedingungen stark variieren können, ist diese Differenzierung von zentraler Bedeutung.

4-stufige Skalierung der Exponiertheit

4	hoch	Generell: sehr große Gruppe Logischer Zugriff: Internet; Großes Intranet Physischer Zugriff: Öffentlichkeit
3	mittel-hoch	Generell: Limitierte aber große Gruppe z.B. alle registrierten Kunden Logischer Zugriff: Extranet, medium Intranet Physischer Zugriff: Besucher Bereich mit eingeschränkter Überwachung (z.B. Konferenzräume) ; WiFi / BT
2	mittel-niedrig	Generell: kleine limitierte Gruppe, ohne strikte Kontrollen; z.B. Projekt-Team Logischer Zugriff Access: Feldbus Physischer Zugriff: Gehäuse des Produktes (USB, Display, SD-Karte,...); Physische Zugangsbeschränkung (z.B. Werkshalle)

1	niedrig	Generell: sehr kleine, gut kontrollierte Gruppe z.B. 3 Administratoren welche nur unter Kontrolle Zugriff zum System erhalten Physischer Zugriff: Versperrte Schaltschränke; Platine (JTAG, SPI, I2C, UART)

Die Ausnutzbarkeit beschreibt, wie aufwendig und anspruchsvoll ein Angriff auf das jeweilige Produkt wäre – also den technischen Schwierigkeitsgrad eines erfolgreichen Angreifers. Dieser Aspekt lässt sich besonders gut von Entwicklern und sicherheitsaffinen Fachkräften im Unternehmen bewerten, da sie die eingesetzten Technologien, Schutzmechanismen und internen Abläufe kennen.

Folgende Fragen helfen bei der Einschätzung:

– Kann ein Angriff mit frei verfügbaren Tools und ohne technisches Wissen durchgeführt werden?
– Erfordert er Reverse Engineering, tiefes Protokollverständnis, oder das Überwinden mehrerer Schutzschichten?

Die Ausnutzbarkeit von Embedded-Systemen variiert stark. Systeme mit sicherem Boot, kontrollierten Updates und ohne Debug-Schnittstellen sind schwerer angreifbar. Offene Systeme oder Entwicklungsboards können hingegen schon durch physischen Zugang, etwa per USB-Exploit, stark gefährdet sein.

4-stufige Skalierung der Ausnutzbarkeit

4	einfach	Praktisch kein Wissen notwendig z.B. Benutzung von Standardpasswörtern Gewöhnliche Benutzung des Systems IEC 62443 SL-1 Angreifer
3	mittelmäßig	Kann mit verfügbaren Werkzeugen durchgeführt werden Erfordert grundsätzliches IT oder Domänen Wissen z.B. Brute-Force von schwachen Passwörtern Ausnutzung bekannter Schwachstellen IEC 62443 SL-2 Angreifer
2	schwierig	Erfordert die Entwicklung von oder tiefe Kenntnis über Werkzeuge Erfordert tiefes IT oder Domänen Wissen z.B. Kompilieren von eigenen Tools IEC 62443 SL-3 Angreifer
1	extrem schwierig	Erfordert außergewöhnliche technische Expertise Erfordert interne Informationen mit geringem Verbreitungsgrad z.B. Reverse Engineering einer geschützten Anwendungsdatei IEC 62443 SL-4 Angreifer

Auf Basis dieser Parameter können eigene Zusammenhänge definiert werden. Folgende Matrizen haben sich bewährt:

Auswirkung \ Wahrscheinlichkeit	1	2	3	4
1	1	1	1	2
2	2	2	2	3
3	2	2	3	4
4	2	3	4	4

Exponiertheit \ Ausnutzbarkeit	1	2	3	4
1	1	1	2	2
2	1	2	2	3
3	2	2	3	3
4	2	3	3	4

Auswirkung und Wahrscheinlichkeit haben eine leichte Asymmetrie. Diese wurde absichtlich gewählt, um Auswirkungen gegenüber der ohnehin schwierigen Eintrittswahrscheinlichkeit etwas stärker zu gewichten. Ausnutzbarkeit und Exponiertheit ist relativ selbsterklärend. Beurteilungen hinsichtlich unseres Beispiels finden sich in den Tabellen.

7.7 Schritt 6: Bedrohungen identifizieren

Der nächste Schritt der Bedrohungsanalyse besteht darin, konkrete Bedrohungen (Threats) zu identifizieren. Also Szenarien, in denen Angreifer versuchen, Schwachstellen im Produkt auszunutzen. Dies gilt als einer der komplexesten Schritte der Analyse, da es keine fixen Listen gibt, sondern Kreativität, Systemverständnis und Security-Erfahrung gefragt sind.

Hilfreiche Herangehensweisen zur strukturierten Findung von Bedrohungen:

– **Attack Trees**
Alternativ oder ergänzend kann ein sogenannter Angriffsbaum erstellt werden:
Man beginnt mit einer potenziellen Auswirkung (z. B. Datenverlust, Systemmanipulation) und arbeitet sich rückwärts durch: „Wie könnte es dazu kommen? Welche Schritte müsste ein Angreifer durchlaufen?"

– **STRIDE-Methode**
Ein bewährter Ansatz zur systematischen Analyse von Bedrohungstypen.

S **Spoofing** – Identitätsfälschung

T **Tampering** – unbefugte Veränderung von Daten

R **Repudiation** – fehlende Nachvollziehbarkeit von Aktionen

I **Information Disclosure** – ungewollte Datenoffenlegung

D **Denial of Service** – Verfügbarkeitsstörung

E **Elevation of Privileges** – unberechtigter Rechtezuwachs

Für jedes Asset oder jede Schnittstelle kann STRIDE als Checkliste verwendet werden: „Könnte hier jemand etwas manipulieren? Auslesen? Sperren?"

– **Expertenbasierter Ansatz**
Gerade bei Embedded-Systemen mit spezifischer Hardware, Legacy-Protokollen oder herstellereigenem Know-how empfiehlt sich ein Workshop mit erfahrenen Security-Spezialisten, um typische, aber nicht offensichtliche Bedrohungsszenarien zu erfassen.

Beispielhafte Bedrohungen für unser IoT-Gerät:

⚠ **Firmware-Manipulation via USB oder SD-Karte**
Ein Angreifer bringt ein präpariertes Update-Image ein, das die bestehende Firmware ersetzt und Schadfunktionen aktiviert (STRIDE: Tampering, Elevation of Privileges).

⚠ **Datenabgriff über physische Speichermedien**
Auf dem USB-Stick oder der SD-Karte befinden sich unverschlüsselte Backups oder Logs, die sensible Sensorinformationen preisgeben (STRIDE: Information Disclosure).

⚠ **Datenmanipulation historischer Messwerte**
Ein Angreifer verändert gespeicherte Konfigurations- oder Sensordaten und untergräbt die Integrität des Systems (STRIDE: Tampering).

⚠ **Remote-Zugriff über ungeschützte Netzwerkdienste**
Über SSH oder Web-UI wird Zugriff erlangt, z. B. durch schwache Zugangsdaten oder offene Ports (STRIDE: Spoofing, Tampering, Elevation of Privileges).

⚠ **Root-Rechte durch Exploit in privilegierten Diensten**
Angreifer nutzt Schwachstelle in einem mit Root-Rechten laufenden Service aus, um an WLAN-Zugangsdaten oder Cloud-Credentials zu gelangen (STRIDE: Elevation of Privileges, Information Disclosure).

⚠ **USB-basierter Angriff mit Malware (z. B. Ransomware)**
Das System wird durch eine präparierte Datei oder ein kompromittiertes Speichermedium blockiert oder verschlüsselt (STRIDE: Denial of Service).

Diese Bedrohungen werden in der Praxis tabellarisch dokumentiert und idealerweise ergänzt, um betroffene Schnittstellen, Assets, potenzielle Auswirkungen und bereits bestehende Gegenmaßnahmen zu erkennen. So entsteht eine strukturierte Übersicht, die als Basis für die Risiko- und Maßnahmenbewertung dient.

> Gerade bei CRA-relevanten Embedded Systemen ist eine fundierte, nachvollziehbare Bedrohungsanalyse ein zentraler Bestandteil der Konformitätsprüfung – nicht zuletzt, weil sie regulatorisch gefordert ist und als Entscheidungsgrundlage für technische Schutzmaßnahmen dient.

BEDROHUNG	BEWERTUNG	EXPONIERTHEIT	BEWERTUNG	AUSNUTZBARKEIT	BEWERTUNG	WAHRSCHEINLICKEIT	RISIKO
Angreifer könnten über die SD Karte/USB ein manipuliertes Firmware Update Image einspielen, welches installiert wird und dadurch die Firmware manipulieren.	4	Am Gehäuse des Geräts zugänglich	2	Analyse Firmware Image & Bau einer lauffähigen Firmware, welche gezielt Daten manipuliert	2	2	3
Angreifer könnten über die SD Karte/USB BackUps abgreifen und sich Zugriff zu Sensordaten verschaffen.	1	Am Gehäuse des Geräts zugänglich	2	Unverschlüsseltes CSV/JSON Format	4	3	1
Angreifer könnten über die SD Karte/USB BackUps manipulieren, wodurch historische Sensordaten korrumpiert werden.	2	Am Gehäuse des Geräts zugänglich	2	Unverschlüsseltes CSV/JSON Format	4	3	2
Angreifer könnten sich über die Fernzugriffsschnitt-stelle Zugriff auf das System verschaffen und Konfigurationen verändern.	3	Unmittelbare Nähe	3	Zugang zu Wifi - SSH Authentifizierung muss ge-brute-forced werden oder anderweitig Zugang zu Credentials erlangt werden	2	2	2
Angreifer könnten sich über die Fernzugriffsschnittstelle Zugriff auf das System verschaffen und über eine Schwachstelle in einem mit Root-Rechten laufenden Service erweiterte Rechte erlangen. Dadurch könnten sie Zugriff auf WiFi- oder Cloud-Credentials erhalten.	3	Unmittelbare Nähe	3	- Zugang zu Wifi - SSH Authentifizie-rung ge-brute-forcen oder anderweitig Zugang zu Credentials erlangt werden - Schwachstelle in Root-Service	1	2	2
Angreifer könnten einen USB Stick mit Ransom-ware anschließen und das IoT Gerät unbrauchbar machen.	3	Am Gehäuse des Geräts zugänglich	2	Autorun/Mount aktiv	4	3	3

7.8 Schritt 7: Risikobeurteilung

Sobald Bedrohungen identifiziert sind, werden diese nun anhand der zuvor definierten Parameter bewertet:
– Auswirkung
– Exponiertheit der betroffenen Schnittstelle
– Ausnutzbarkeit des jeweiligen Angriffsszenarios

> **Hierbei gilt:** Bereits existierende Schutzmaßnahmen (z. B. Verschlüsselung, Authentifizierung, physische Barrieren) werden mitberücksichtigt. Sie senken die tatsächliche Ausnutzbarkeit und somit das Gesamtrisiko.

Auf Basis der kombinierten Bewertungen ergibt sich dann eine **Risikobeurteilung**, meist mithilfe einer einfachen Matrix. Es hat sich folgendes, 4-stufiges Modell bewährt:

RISIKO LEVEL	BEDEUTUNG	HANDLUNGSBEDARF
1	akzeptabel	keine Maßnahme notwendig
2	grundsätzlich akzeptabel	beobachten, dokumentieren
3	eingeschränkt akzeptabel	Maßnahme empfohlen
4	nicht akzeptabel	Maßnahmen zwingend erforderlich

Besonders bei Embedded-Produkten ist diese Skalierung hilfreich, da einzelne Bedrohungen je nach Einsatzszenario stark unterschiedlich gewichtet werden. Ein Auszug aus der Bewertung unseres Beispiel-IoT-Geräts könnte so aussehen:

BEDROHUNG	AUSWIRKUNG	EXPONIERTHEIT	AUSNUTZBARKEIT	RISIKO
Firmware-Manipulation über SD-Karte	hoch	mittel	Gering (keine Signaturprüfung)	4
Abgriff von Sensordaten via USB	mittel	hoch	mittel	3
Fernzugriff auf Systemkonfiguration	hoch	hoch	mittel	4
Rechteausweitung via Root-Service	hoch	mittel	hoch	3
Malware über USB-Stick	mittel	hoch	gering	4

7.9 Schritt 8: Maßnahmen ableiten

Risiken mit Bewertung 3 oder 4 erfordern in jedem Fall weitere Maßnahmen, um das Risiko zu senken.

Dabei ist es sinnvoll, bereits im Zuge der Analyse **konkrete Gegenmaßnahmen zu diskutieren**, um Aufwand und Wirksamkeit direkt einschätzen zu können. Neue Schutzmaßnahmen können:

– die Ausnutzbarkeit reduzieren (z. B. durch Authentifizierung oder Signaturprüfung)
– die Exponiertheit senken (z. B. durch Deaktivierung oder physische Abschirmung)
– die potenzielle Auswirkung begrenzen (z. B. durch Logging, Fail-Safe Design, beschränkten Zugriff)

Die Wirksamkeit dieser Maßnahmen kann durch eine erneute Risikobewertung dokumentiert werden z. B. als zusätzliche Spalte oder neue Zeile in der Risikotabelle.

BEDROHUNG	RISIKO	GEGENMASSNAHME
Angreifer könnten über die SD Karte/USB ein manipuliertes Firmware Update Image einspielen, welches installiert wird und dadurch die Firmware manipulieren.	3	Secure Update Mechanismus
Angreifer könnten über die SD Karte/USB Back-Ups abgreifen und sich Zugriff zu Sensordaten verschaffen.	1	Verschlüsseltes BackUp Image
Angreifer könnten über die SD Karte/USB BackUps manipulieren, wodurch historische Sensordaten korrumpiert werden.	2	Verschlüsseltes BackUp Image

Angreifer könnten sich über die Fernzugriffs-schnittstelle Zugriff auf das System verschaffen und Konfigurationen verändern.	2	MFA
Angreifer könnten sich über die Fernzugriffs-schnittstelle Zugriff auf das System verschaffen und sich über eine Schwachstelle in einem mit Root-Rechten laufenden Service, erweiterte Rechte erlangen und so Zugriff auf WiFi Credentials oder Cloud Credentials erlangen.	2	MFA
Angreifer könnten einen USB Stick mit Ransomware anschließen und das IoT Gerät unbrauchbar machen.	3	Einschränkung valide USB Devices und Deaktivierung Autorun

Diese Maßnahmen helfen, Risiken systematisch zu senken, genauso, wie es der CRA im Sinne eines risikobasierten Ansatzes fordert. Gesetzte Maßnahmen sind dann als Anforderung in das verwendete Requirements-Engineering Tool zu übernehmen und zu referenzieren.

7.10 Schritt 9: Wiederholen und Updaten

Die Bedrohungsanalyse ist **kein einmaliges Ereignis**, sondern ein kontinuierlicher Prozess, der den gesamten Lebenszyklus eines Produkts begleiten muss. Der Cyber Resilience Act (CRA) fordert explizit, dass **Risiken regelmäßig überprüft und bei Bedarf neu bewertet werden**, um auf veränderte Rahmenbedingungen reagieren zu können.

Eine Aktualisierung der Analyse ist insbesondere in folgenden Fällen erforderlich:

- Neue Schnittstellen oder Funktionen werden hinzugefügt (z. B. zusätzliche Netzwerkprotokolle, neue Kommunikationsmodule, OTA-Funktionen)
- Funktionalitäten entfallen oder ändern sich (z. B. ein Dienst wird deaktiviert, Authentifizierungsmechanismus geändert)
- Bekannte Vorfälle oder neue Schwachstellen betreffen eigene oder vergleichbare Produkte
- Software-Komponenten oder externe Bibliotheken werden aktualisiert, was Auswirkungen auf die Bedrohungslage haben kann
- Einsatzbedingungen verändern sich, z. B. durch Migration von Offline- zu Online Betrieb

Die regelmäßige Überprüfung dient dabei zwei Zielen:

- Sicherstellung der Aktualität der Risikoeinschätzung – insbesondere im Hinblick auf dynamische Bedrohungslagen im Bereich Embedded Security.
- Nachweis der Sorgfaltspflicht – auch gegenüber Behörden und im Rahmen von Audits.

Diese wiederholte Analyse ermöglicht es, frühzeitig auf Veränderungen zu reagieren und die CRA-Anforderungen konsequent umzusetzen. Mit einem Sicherheitskonzept, das nicht nur dokumentiert, sondern gelebt wird.

ANHÄNGE

SOFTWARE BILLS OF MATERIALS (SBOM) & SCHWACHSTELLENMONITORING

Ein weiterer zentral geforderter Punkt im CRA ist die Erstellung einer Software-Stückliste, auch Software Bill of Materials (SBoM) genannt. Hierbei handelt es sich **um eine Liste, in der alle Bestandteile einer Software**, wie bspw. Bibliotheken, Module und andere Abhängigkeiten aufgeführt sind. Sie hilft dabei, einen klaren Überblick darüber zu bekommen, wie die Software aufgebaut ist und welche Komponenten verwendet werden. Moderne Softwareprojekte haben oft viele Abhängigkeiten, welche wiederum selbst Abhängigkeiten an weitere Pakete definieren, aber teils unterschiedliche Versionen fordern. Package-Manager helfen dabei, diese Abhängigkeiten aufzulösen und passende Versionen zu laden. Mithilfe einer SBoM kann anschließend festgehalten werden, welche Komponenten in welcher Version nun tatsächlich verwendet wurden.

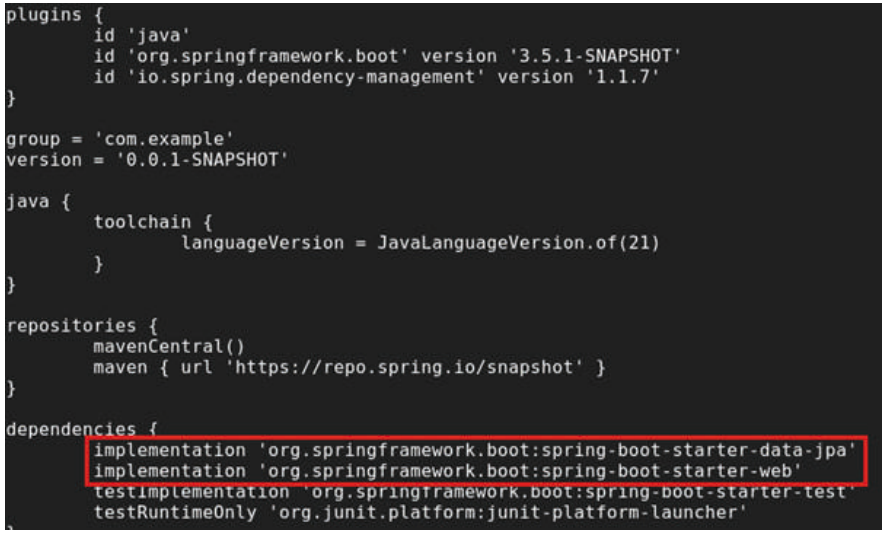
Solche Stücklisten können automatisch mit Tools wie Syft erstellt werden. Diese Tools nutzen oftmals die Konfigurations- bzw. Lockfiles der Package Manager und sind somit ohne großen Aufwand in bestehende Build-Prozesse zu integrieren. Je nach Format können auch die Application-Packages selbst analysiert werden, wenn diese die notwendigen Metadaten enthalten, wie das zum Beispiel bei .jar-Files der Fall ist. Wenn man eingebettete Systeme entwickelt, kann man dafür auch Build-Werkzeuge wie Yocto nutzen und die damit erfassten Abhängigkeiten anschließend mit Programmen wie Dependency-Track oder Trivy überprüfen und verwalten.
Es bietet sich an, die Erstellung der SBoM in eine bestehende Build-Pipeline zu integrieren, somit kann diese ohne zusätzlichen manuellen Aufwand für jeden Build bzw. Release automatisch erstellt werden.

Die SBoM dient nicht nur zur Erfüllung der CRA-Anforderungen, sondern bietet auch einen praktischen Mehrwert im Softwareentwicklungsprozess. So ermöglicht es eine SBoM die Lizenzen der verwendeten Komponenten zu überprüfen, aber auch in Schwachstellendatenbanken nach bekannten Schwachstellen zu suchen. Diese Information kann genutzt werden, um die Auswirkungen auf das eigene Produkt zu analysieren und bei Bedarf Maßnahmen ergreifen zu können. Idealerweise ist dieses Schwachstellenmonitoring im Softwareentwicklungsprozess fest verankert.

Die Experten der Limes Security können bei der Definition eines solchen Prozesses beratend zur Seite stehen, bieten aber auch Trainings an, damit die handelnden Personen das notwendige Wissen erwerben können, um mit den vorhandenen Daten bestmöglich arbeiten zu können. Sollten Annahmen getroffen werden, dass z.B. bekannte Schwachstellen in verwendeten Komponenten keine Auswirkungen auf das eigene Produkt haben können, können diese in einem Penetration-Test durch erfahrene Tester der Limes Security unabhängig überprüft werden.

GELIN SBOM UND MONITORING AUS DER PRAXIS; SCREENSHOTS VON GELIN

In Abbildung 1 ist die Liste der Abhängigkeiten im build.gradle-File eines beispielhaften Java-Programms zu sehen. Diese Liste beinhaltet nur die vom Entwickler geforderten Abhängigkeiten, nicht aber deren Komponenten. Daher bietet diese Auflistung keinen vollständigen Überblick. Im Gegensatz dazu zeigt Abbildung 2 die gesamte Liste der verwendeten Komponenten und deren exakte Version, wie von einer SBoM zu erwarten ist. Diese Liste wurde mit Syft erstellt und kann zur weiteren maschinellen Verarbeitung auch in anderen Formaten wie SPDX oder CycloneDX generiert werden.



```
plugins {
        id 'java'
        id 'org.springframework.boot' version '3.5.1-SNAPSHOT'
        id 'io.spring.dependency-management' version '1.1.7'
}

group = 'com.example'
version = '0.0.1-SNAPSHOT'

java {
        toolchain {
                languageVersion = JavaLanguageVersion.of(21)
        }
}

repositories {
        mavenCentral()
        maven { url 'https://repo.spring.io/snapshot' }
}

dependencies {
        implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
        implementation 'org.springframework.boot:spring-boot-starter-web'
        testImplementation 'org.springframework.boot:spring-boot-starter-test'
        testRuntimeOnly 'org.junit.platform:junit-platform-launcher'
```

```
}

tasks.named('test') {
        useJUnitPlatform()
}
```

Abbildung 1: Defintion der Abhängigkeiten eines Java-Programms

NAME	VERSION	TYPE
HikariCP	6.3.0	java-archive
angus-activation	2.0.2	java-archive
antlr4-runtime	4.13.0	java-archive
aspectjweaver	1.9.24	java-archive
byte-buddy	1.17.5	java-archive
classmate	1.7.0	java-archive
demo	0.0.1-SNAPSHOT	java-archive
hibernate-commons-annotations	7.0.3.Final	java-archive
hibernate-core	6.6.15.Final	java-archive
istack-commons-runtime	4.1.2	java-archive
jackson-annotations	2.19.0	java-archive
jackson-core	2.19.0	java-archive
jackson-databind	2.19.0	java-archive
jackson-datatype-jdk8	2.19.0	java-archive
jackson-datatype-jsr310	2.19.0	java-archive
jackson-module-parameter-names	2.19.0	java-archive
jakarta.activation-api	2.1.3	java-archive
jakarta.annotation-api	2.1.1	java-archive
jakarta.inject-api	2.0.1	java-archive
jakarta.persistence-api	3.1.0	java-archive
jakarta.transaction-api	2.0.1	java-archive
jakarta.xml.bind-api	4.0.2	java-archive
jandex	3.2.0	java-archive
jaxb-core	4.0.5	java-archive
jaxb-runtime	4.0.5	java-archive
jboss-logging	3.6.1.Final	java-archive
jul-to-slf4j	2.0.17	java-archive
log4j-api	2.24.3	java-archive
log4j-to-slf4j	2.24.3	java-archive
logback-classic	1.5.18	java-archive
logback-core	1.5.18	java-archive
micrometer-commons	1.15.0	java-archive
micrometer-observation	1.15.0	java-archive
slf4j-api	2.0.17	java-archive
snakeyaml	2.4	java-archive
spring-aop	6.2.7	java-archive
spring-aspects	6.2.7	java-archive
spring-beans	6.2.7	java-archive
spring-boot	3.5.1-SNAPSHOT	java-archive
spring-boot-autoconfigure	3.5.1-SNAPSHOT	java-archive
spring-boot-jarmode-tools	3.5.1-SNAPSHOT	java-archive
spring-context	6.2.7	java-archive
spring-core	6.2.7	java-archive
spring-data-commons	3.5.0	java-archive
spring-data-jpa	3.5.0	java-archive
spring-expression	6.2.7	java-archive
spring-jcl	6.2.7	java-archive
spring-jdbc	6.2.7	java-archive
spring-orm	6.2.7	java-archive
spring-tx	6.2.7	java-archive
spring-web	6.2.7	java-archive
spring-webmvc	6.2.7	java-archive
tomcat-embed-core	10.1.41	java-archive
tomcat-embed-el	10.1.41	java-archive
tomcat-embed-websocket	10.1.41	java-archive
txw2	4.0.5	java-archive

Abbildung 2: Liste der tatsächlich verwendeten Komponenten des Java Programms

In Abbildung 3 ist ein das Monitoring Dashboard von OWASP Dependency Track zu sehen. Damit können die Inhalte der SBoMs dargestellt und deren Änderungen getrackt werden, um vulnerable und/oder veraltete Komponenten zu erkennen und deren Upgrades zu tracken.

Abbildung 3: Monitoring Dashboard von Dependency-Track

Mit der Verwendung der oben genannten Tools gelingt es einfach und schnell, eine SBoM ohne großen Aufwand zu generieren. Außerdem tätigt man damit erste Schritte in Richtung Konformität mit dem CRA. Aus genau diesem Grund ist es jedoch wichtig, Tools zu kennen und zu verwenden, auf die man sich verlassen kann.

VERWENDUNG VON BEREITS EXISTIERENDEN STANDARDS

Der CRA ist vorsätzlich sehr generisch formuliert worden, sodass das Inverkehrbringen von Produkten mit digitalen Elementen so einfach wie möglich ist. Dies erschwert es allerdings, konkrete Anweisungen und Richtlinien für die Konformität mit dem CRA zu definieren. Trotz dessen hat die ENISA für bereits bestehende Standards erörtert, wie gut sie die Anforderungen des CRA erfüllen und welche Lücken bei diesen noch bestehen.

Als sinnvolle Stütze zur Implementierung und Konformität mit dem Anhang 1 hat sie die Serie der IEC 62443 genannt, da die Standards 4-1 und 4-2 große Teile der Anforderungen des CRA abdecken.

HARMONISIERTE NORMEN

Harmonisierte Normen sind technische Spezifikationen, welchen von anerkannten europäischen Normungsorganisationen (z.B. CENELEC) im Auftrag der Europäischen Union erarbeitet und veröffentlicht werden mit dem Ziel, Vorgaben zur Erfüllung von Rechtsvorschiften zu definieren. Wenn die erarbeiteten Normen den Vorstellungen der EU-Kommission entsprechen, werden diese aufgenommen und als harmonisierte Norm geführt. Dadurch bieten sie Rechtssicherheit. Wenn ein Hersteller seine Konformitätsvermutung auf Basis der Norm(en) darlegen kann und die Anforderungen wie spezifiziert erfüllt sind, ist auch genug zur Erfüllung der Regularie umgesetzt worden. Für den CRA gibt es noch keine harmonisierten Normen. Arbeitsgruppen haben hier ihre Tätigkeit aufgenommen. Mit ersten Entwürfen ist im Jahr 2026 zu rechnen.

Internationale Normen wie die IEC 62443, speziell die für Hersteller relevanten Teile 4-1 (sichere Entwicklungsprozesse), bzw. 4-2 (technischen Security Anforderungen an Komponenten), sind keine harmonisierten Normen und können es auch künftig nicht werden. Eine Zertifizierung nach den Normen bietet also auch keine Rechtssicherheit, ausreichend für den CRA getan zu haben. Trotzdem ist in Abwesenheit von harmonisierten Normen eine international anerkannte und bewährte Standardreihe wie die IEC 62443 definitiv ein guter Startpunkt. Viele der Anforderungen werden in ähnlicher Form zu erwarten bzw. umzusetzen sein. Um einen Eindruck zu bekommen, wie eine harmonisierte Norm aufgebaut sein kann, empfiehlt sich ein Blick in die EN 18031 Reihe. Diese definiert Vorgaben zur Erfüllung der Cybersicherheitsanforderungen aus der Funkanlagenrichtlinie. Die EN 18031 ist nicht als harmonisierte Norm für den CRA gelistet und bietet daher auch keine Rechtsicherheit.

Als erste harmonisierte Norm, die die Cybersecurity Anforderungen definiert, ist sie ein guter Richtwert dafür, was erwartet werden kann.

Einige der Anforderungen werden sich wahrscheinlich in leicht abgeänderter Version in einer Norm für den CRA wiederfinden. Im Folgenden findet sich eine Gegenüberstellung, in der inhaltlich die Anforderungen des CRA mit den technischen Anforderungen der EN 18031 verknüpft werden.

CRA Anforderung	Passende EN18031 Anforderungen	Kommentar
2a	GEC-1	
2b		
2c	SUM-1, SUM-2, SUM-3	
2d	ACM-1, ACM-2, AUM-1, AUM-2	
2e	SSM-1, SSM-3, SCM-1, SCM-3	
2f	SSM-1, SSM-2, SCM-1, SCM-2	Meldung von Beschädigung wird nicht erwähnt.
2g		
2h	NMM-1, TCM-1	
2i	GEC-2	Wahrscheinlich nicht ausreichend, aber hilfreich.
2j	GEC-2, GEC-3, GEC-5	
2k		
2l	LGM-1 (18031-2)	Nur Aufzeichnung wird erwähnt, Opt-Out fehlt ebenso.
2m	DLM-1 (18031-2)	Nur anwendbar auf Daten und Einstellungen.

IEC 62443:

In der Einleitung zu diesem Kapitel wurde bereits erwähnt, dass die ENISA die Normen IEC 62443-4-1 und -4-2 analysiert hat und dabei feststellte, dass diese Standards eine solide Grundlage für den Cyber Resilience Act bilden. Wir geben in folgender Tabelle unsere Einschätzung zu diesem Statement ab. Hier ist jedoch Vorsicht geboten: Weder die Aussage der ENISA noch die Einschätzung der Experten bei Limes Security bieten rechtliche Sicherheit im Falle einer Konformitätsprüfung!

Da den CRA keine harmonisierten Standards betreffen und Implementierungs-richtlinien von der EU noch folgen sollten, gibt es noch keine Sicherheit auf Konformität in Bezug auf existierende Standards wie der IEC 62443. Dennoch bietet sie eine gute Grundlage, um sich mit der Cybersicherheit eines Produkts zu befassen und erste Meilensteine auf dem Weg zur Konformität zu legen. Die folgende Tabelle soll zeigen, wie Limes Security die IEC 62443 hinsichtlich des CRA bewertet:

ANFORDERUNG	PASSENDE 62443 ANFORDERUNGEN	KOMMENTAR
Anhang | Teil |		
2a	SM-11, SM-12	
2b	SD-4, FR-7: CR7.7	
2c	SUM-4, FR-3: CR3.10	
2d	FR-1, FR-2	
2e	FR-4	
2f	SM-6, FR-3	Relevante erhaltene Daten, Meldung wird nicht angesprochen
2g		
2h	FR-7: CR7.1, CR7.2	
2i	FR-5: CR5.1	
2j	SD-4, FR-3: CR3.11, FR-7: CR7.6, CR7.7	
2k	SD-4	
2l	FR-6, FR-2: CR2.8, CR2.11, CR2.12	Opt-Out wird nicht erwähnt
2m	SR-3	

ANFORDERUNG	PASSENDE 62443 ANFORDERUNGEN	KOMMENTAR
Anhang | Teil ||		
Abs. 1	SVV-3, DM-1	
Abs. 2	SM-11, DM-4, SUM-1, SUM-5	
Abs. 3	SVV	
Abs. 4	DM-3, DM-5, SUM-2	DM-3 für Dokumentation, DM-5 für Offenlegung
Abs. 5	DM-5	
Abs. 6	DM-1	
Abs. 7	SUM-4	Automatische Updates werden nicht erwähnt.
Abs. 8	SUM-5, DM-5	
Anhang ||		
Abs. 1		
Abs. 2	DM-1	
Abs. 3		
Abs. 4	SR-1	
Abs. 5	SR-1, SR-2	
Abs. 6		
Abs. 7		
Abs. 8		
8a	SG-1, SG-3, SG-5	
8b	SG-3	
8c	SG-3, DM-5	
8d	SG-4	
8e	DM5	
8f	SG-1, SG-2, SG-3, SG-4, SG_5, SG-6	
Abs. 9		

GLOSSAR

„Safety. bzw. Funktionale Sicherheit“
Schutz von Mensch & Umwelt vor dem System

„Security“
Schutz des Systems vor dem Menschen

„Cybersicherheit“
Schutz des Systems, im speziellen vor Angriffen aus dem Internet.

„Schwachstelle“
Fehler im Design oder in der Umsetzung, die ein Angreifer ausnutzen könnte.

„Vorfall“
Von einem Angreifer ausgenutzte Schwachstelle in einem System mit negativen Auswirkungen.

„Bedrohung“
Potenzielles Szenario eines Vorfalls.

„Risiko“
Gemäß Eintrittswahrscheinlichkeit & Auswirkungen beurteilte Bedrohung mit einer daraus resultierenden Risikoeinschätzung.

„Schnittstelle“
Interface (logisch oder physisch), das für einen Angriff verwendet werden kann.

„Asset“
Schützenswerter Aspekt eines Systems (Programm, Konfiguration, Daten,...).

„Embedded System”
Ein spezialisiertes Computersystem, das fest in ein technisches Gerät integriert ist und dort eine klar definierte Funktion übernimmt.

“Embedded Linux”
Eine angepasste Version des Linux-Betriebssystems, die in Embedded Systemen eingesetzt wird, um komplexe Aufgaben wie Vernetzung, GUI oder Update-Management zu ermöglichen.

“Embedded Hardware”
Die für Embedded Systeme entwickelte oder angepasste Hardware, bestehend aus Prozessor, Speicher und Schnittstellen, optimiert für Stabilität, Energieeffizienz und Langlebigkeit.

“Microcontroller”
Eine kompakte Recheneinheit mit begrenzter Leistung, die Steuerungs- und Überwachungsaufgaben in Embedded Systemen zuverlässig übernimmt.

“Applikationsprozessor”
Ein leistungsfähiger Prozessor mit Unterstützung für Betriebssysteme wie Linux, der für grafikintensive, vernetzte oder rechenintensive Anwendungen in Embedded Devices verwendet wird.

„SBOM (Software Bill of Materials)“
Eine vollständige Auflistung aller enthaltenen Software-Komponenten eines Produkts, die Transparenz schafft und Sicherheits- sowie Compliance-Prüfungen ermöglicht.

ÜBER DIESES WHITEPAPER

Ginzinger electronic systems und Limes Security verbinden technologische Kompetenz mit fundierter Security-Expertise.

Ginzinger electronic systems entwickelt maßgeschneiderte Embedded-Hard- und Software, Limes Security unterstützt mit praxisnaher Beratung und zielorientierter Umsetzung von Sicherheitsanforderungen. Gemeinsam entstehen Lösungen, die von Anfang an sicher und zuverlässig konzipiert sind.

Neben erfolgreich realisierten Projekten veröffentlichen wir Whitepaper und veranstalten regelmäßig Seminare und Webinare zur sicheren Geräteentwicklung. Unternehmen profitieren so nicht nur von konkreten Lösungen, sondern auch von praxisnahem Wissenstransfer.

Limes Security ist international anerkannt: Experten sind Mitglieder in Gremien zur IEC 62443 und gefragte Speaker auf Konferenzen wie Black Hat, SANS SCADA oder IFIP WG11.10. Zusammen mit der Entwicklungs- und Fertigungskompetenz von Ginzinger entsteht eine Partnerschaft, die Unternehmen ganzheitlich auf dem Weg zu sicheren, zukunftsfähigen Produkten begleitet.

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für beiderlei Geschlecht.

LIMES SECURITY

GINZINGER electronic systems

START YOUR SECURITY JOURNEY.

Limes Security GmbH – Softwarepark 49 – 4232 Hagenberg im Mühlkreis – **www.limessecurity.com**

Ginzinger electronic systems GmbH – Gewerbegebiet Pirath 16 – 4952 Weng im Innkreis – **www.ginzinger.com**