(Bild: Tashatuvango – Shutterstock)

Das IoT und die Sache mit der Sicherheit:

Cybersecurity by Design

Vernetzte Produkte schaffen neue Risiken, die es zu beherrschen gilt. Connectivity bedeutet, dass auch Angreifer einen Weg finden können, in ein Gerät einzudringen. Es gibt viele Empfehlungen, wie man sichere Software entwickelt. Auf welche sollte man sich konzentrieren?

Von Tobias Kästner und Michael Lerch

Unsere heutige Zeit – das Informationszeitalter – ist geprägt vom zunehmend schneller wachsenden Vernetzungsgrad von Daten und der Geräte, die sie erzeugen oder durch sie gesteuert werden. Zweifelsohne sind damit vormals unmöglich erscheinende Anwendungen verknüpft, die unser Leben auf vielfältige Weise verbessern können. Andererseits gibt es auch schon genug Beispiele, wie die neuen Technologien gegen das Wohl von Menschen gerichtet werden. Eines haben dabei fast alle Vorfälle wie „WannaCry“ oder „Mirai“ gemein: Dem Thema Sicherheit wurde im Vorfeld weder bei den Herstellern der betroffenen Produkte noch bei ihren späteren Nutzern angemessen Beachtung geschenkt. Schlimmer noch: Das Gleiche trifft auch auf eine Vielzahl von Geräten zu, die – missbräuchlich benutzt – unmittelbar Gesundheit und Menschenleben gefährden können. In diese Kategorie fallen u.a. vernetzte Medizingeräte.

Cybersecurity in der Medizintechnik

So sah sich zum Beispiel unlängst ein großes Medizintechnik-Unternehmen gezwungen, für seine mit Funktechnologie ausgestatteten Herzschrittmacher einen Rückruf zu veranlassen. Mittels eines Software-Updates müssen nun elementare Sicherheitsmerkmale wie etwa ein autorisierter und verschlüsselter Datenaustausch nachgereicht werden. Und das lange, nachdem die Geräte Patienten bereits implantiert worden sind. Natürlich ist die Betrachtung von Gefährdungen in der Medizintechnik keineswegs neu. Allerdings bezog sich die Sprechweise eines „sicheren“ Produktes in der Vergangenheit zumeist auf die funktionale Sicherheit. Sicherheit im informationstechnischen Sinne wurde oft nicht ausreichend betrachtet.

Im Englischen lassen sich die unterschiedlichen Begriffe leichter unterscheiden: Safety bezeichnet die funktionale Sicherheit, also den Schutz des Menschen vor dem System. Security meint Sicherheit im Sinne eines Schutzes des Systems vor dem Menschen und Cybersecurity demzufolge die Sicherheit von vernetzten Informationssystemen. Folgerichtig fordert die US-Zulassungsbehörde für Medizinprodukte FDA seit Anfang dieses Jahres eine Betrachtung der Produktrisiken unter dem Gesichtspunkt der Cybersecurity.

Wie für die FDA üblich legt die Behörde ihre Erwartungen dazu in Form sogenannter Guidance-Dokumente dar [1], [2]. Darin taucht eine Reihe zentraler Forderungen auf, die sich so oder ähnlich auch in anderen Guidance-Dokumenten finden. Zum einen muss Cybersecurity stets während des gesamten Produktlebenszyklus betrachtet werden. Konkret heißt das einerseits, dass bereits während der Entwicklung Gefährdungen identifiziert und wirksame Gegenmaßnahmen implementiert werden sollen. Andererseits müssen Hersteller und Betreiber für in Umlauf befindliche Produkte bekannt gewordene Sicherheitslücken schnell erkennen und zeitnah schließen.

Umfang und Aufwand der Maßnahmen sollen sich dabei am zu erwartenden Schadensfall orientieren – Cybersecurity wird also als ein risikobasierter Prozess verstanden. So verlangt die FDA daher in ihrer Guidance für die Premarket Submission, also in Bezug

auf den Entwicklungsprozess, die explizite Dokumentation von Cybersecurity-Risiken mitsamt ihrer Bewertung und den getroffenen Gegenmaßnahmen. In der ergänzenden Guidance für das Postmarket Management fordert die FDA von den Herstellern für das Entdecken bisher unbekannter Security-Risiken eine Beobachtung verfügbarer Quellen und eine Kultur der Offenlegung. Mithin brauchen Hersteller in Zukunft einen entsprechenden Incidence-and-Response-Prozess. Die FDA verweist dabei mehrmals auf ein anderes Dokument, das Hersteller fast jeder Branche mit Blick auf den US-amerikanischen Markt in Zukunft kennen und beachten müssen. Die Rede ist vom „Framework for Improving Critical Infrastructure Cybersecurity" des National Institutes of Standards and Technology (NIST) [3].

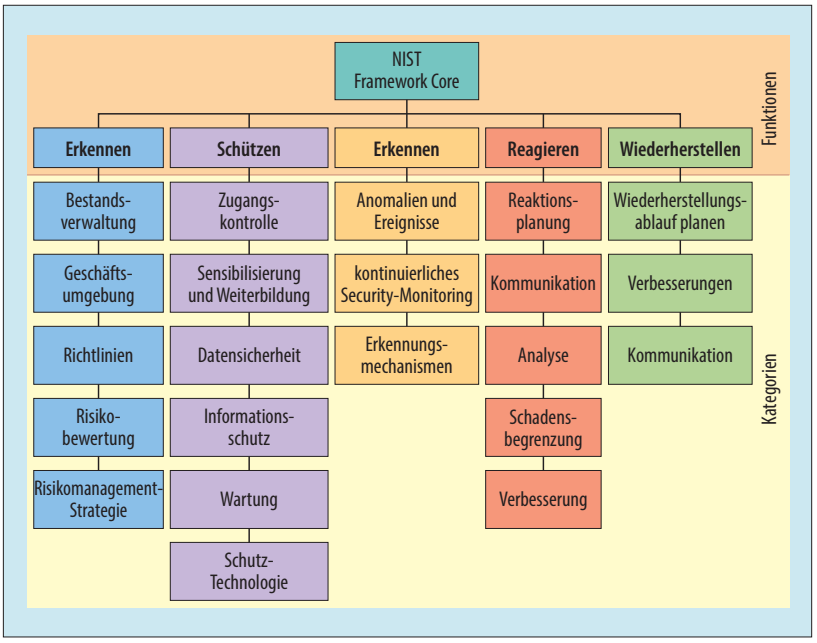

Bild 1. Aufbau des NIST Cybersecurity Frameworks. Nicht gezeigt sind die Sub Categories, die die einzelnen Categories noch einmal aufschlüsseln. (Quelle aller Bilder: Method Park)

Richtlinien für Funktionsbereiche

Wie schon die Guidance-Dokumente der FDA erfindet auch das Cybersecurity Framework das Rad an vielen Stellen nicht neu. Vielmehr bereitet es bestehende Best Practices in strukturierter Form auf, um am Ende stets auf weitere existierende Standards zu verweisen. Im Framework werden dazu zunächst fünf sogenannte Funktionen definiert, von denen jede wiederum in Kategorien und anschließend in Unterkategorien gegliedert ist. Die Funktionen

- ➜ Identifizieren,
- ➜ Schützen,
- ➜ Erkennen,
- ➜ Reagieren und
- ➜ Wiederherstellen

decken dabei jeweils einen Themenschwerpunkt im Umgang mit den Gefährdungen im Cyberspace ab (**Bild 1**). Damit scheinen alle Lebenszyklusphasen eines Produktes bzw. der damit verbundenen digitalen Dienstleistungen abgedeckt zu sein. Tatsächlich stellt sich jedoch bei näherer Betrachtung des Frameworks heraus, dass es sich nur an die Betreiber solcher Systeme richtet. Im konkreten Fall könnte das also beispielsweise ein Krankenhaus sein, das zentral steuerbare Infusionspumpen einsetzt. Deutlich wird das u.a. daran, dass den Fragen „Was passiert, wenn es passiert ist? Und durch wen? Und in welcher Weise?" entsprechend viel Raum gegeben wird.

Für IoT-Geräte nicht ausreichend

Für Medizintechnik-Hersteller ist es natürlich auch nützlich zu wissen, welchen Kriterien spätere Nutzer bei der Produktauswahl besondere Bedeutung beimessen. Dies trifft umso mehr zu, weil viele Hersteller inzwischen selbst zu Anbietern von digitalen Dienstleistungen rund um ihre Produkte werden. Für die Entwicklung von sicheren IoT-Produkten sind die Informationen im NIST Framework jedoch zu allgemein. Hersteller scheinen gut beraten, weitere Guidance-Dokumente zu konsultieren.

An Möglichkeiten dafür mangelt es längst nicht mehr. Der Security-Experte Bruce Schneier zählte Anfang 2017 in seinem Blog [4] an die zwanzig verschiedene Dokumente, die sich einzig und allein mit Cybersecurity im Zusammenhang mit dem Internet der Dinge beschäftigen. Doch gibt es wirklich Anlass genug, derart viele Guidances zu betrachten? Wie sich herausstellt, gibt es eine Grundmenge immer wiederkehrender Empfehlungen, die sich wie ein roter Faden durch fast alle Dokumente ziehen. Auf der organisatorischen Seite gehört dazu zweifelsfrei, dass Hersteller einen geeigneten Entwicklungsprozess etablieren und die notwendigen Kompetenzen in ihren Entwicklungsteams aufbauen müssen. Auf technischer Seite findet sich fast immer der Einsatz von Zugangskontrollen und kryptographischen Verfahren zum Verschlüsseln und Signieren von Daten oder Software-Updates. Auf Letztere wird ebenfalls stets eingegangen und darauf hingewiesen, dass hierzu neben den technischen Möglichkeiten im Gerät auch die notwendige Infrastruktur zum Verteilen von Updates gehört. Das ist erwähnenswert, weil viele Hersteller gerade hier noch keinerlei Erfahrung haben.

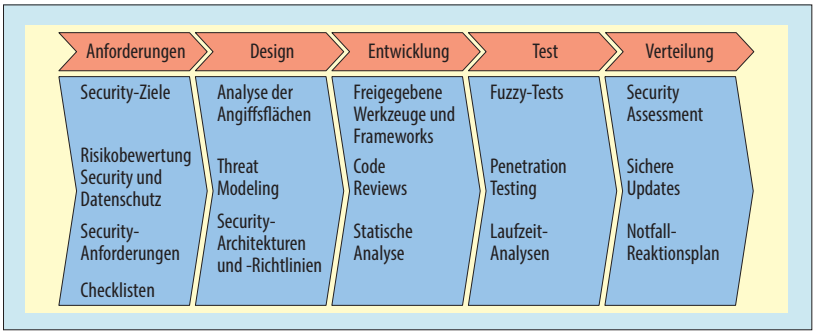

Bild 2. Beispiel für einen Security-Engineering-Prozess basierend auf dem Microsoft Security Development Lifecycle.

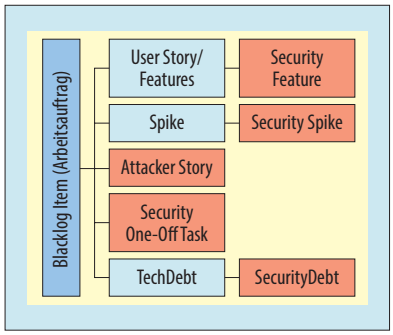

Bild 3. Verschiedene Möglichkeiten, das Produkt-Backlog um securitybezogene Items zu erweitern.

Last but not least wird regelmäßig auch auf die notwendige Durchführung entsprechender Tests hingewiesen. Darin eingeschlossen z.B. das Fuzzy Testing, um die Stabilität eines Systems gegenüber willkürlichen Eingaben zu bewerten, und das Penetration Testing, um den Nachweis zu erbringen, dass ein System vor bekannten Angriffsvektoren hinreichend geschützt ist. Interessierten Lesern sei das Guidance-Dokument der Cloud Security Alliance „Future Proofing the Connected World" und die „IoT Security Guidance" des Open Web Application Security Projects (OWASP) für eine weitere Beschäftigung empfohlen.

Entwicklungsprozess und Security

Wie sollte nun ein Entwicklungsprozess, der einhellig empfohlen wird, gestaltet werden? Lassen sich die zusätzlichen Prozessanforderungen in einem bestehenden und oftmals agilen Prozess überhaupt umsetzen? Um diese Fragen für sich beantworten zu können, gibt es einen guten Ausgangspunkt: den Microsoft Security Development Lifecycle [5]. Bereits seit 2008 verfügbar, wird dieser in regelmäßigen Abständen aktualisiert und ist inzwischen bei Version 5 angekommen. Der Lebenszyklus einer Software wird darin in die üblichen fünf Phasen und eine vorangestellte Trainings- sowie eine nachgestellte Response-Phase gegliedert (**Bild 2**). Jeder dieser Phasen sind für die Cybersecurity relevante Praktiken – 17 an der Zahl – zugeordnet. Wichtig ist, dass der durch den Prozess erzielbare Schutz nur so gut ist wie die Phase mit der schwächsten Umsetzung. Es nützt also wenig, wenn zwar während der Anforderungsanalyse und dem Design alle Praktiken befolgt, dann aber z.B. auf Grund von Zeitdruck die vorgegebenen Praktiken der Verifikations- und Release-Phase nicht angewandt werden.

Ein wesentliches Beispiel für eine der erwähnten Praktiken ist das Threat Modeling. Für die Designphase gefordert, findet es bereits zu einem recht frühen Zeitpunkt statt. Dabei wird ein Modell der zu einem Software-System gehörenden Bedrohungen erstellt. Im Anschluss wird dieses Modell um geeignete Gegenmaßnahmen erweitert, um deren Wirksamkeit bestimmen zu können. Die wesentliche Stärke liegt im systematischen Vorgehen, so dass Aussagen über die Vollständigkeit der erfolgten Analyse möglich sind. Bekannte Angriffsvektoren lassen sich außerdem toolgestützt gegen ein solches Modell recht einfach prüfen. Das gesammelte Wissen und die Erfahrungen aus bisher erfolgten Angriffen können so für die eigenen Produkte genutzt werden.

Microsoft hat die Zeichen der Zeit erkannt und einen Vorschlag unterbreitet, wie sich die Praktiken mit einem agilen Prozess kombinieren lassen. Bekannterweise werden bei einem agilen Vorgehen die einzelnen Entwicklungsphasen stark verkürzt und jeweils nur im Kontext einer einzelnen User Story durchgeführt. Das Produkt entsteht somit iterativ – die Phasen werden immer wieder durchlaufen – und inkrementell – das Produkt entsteht Feature für Feature. Innerhalb dieser sehr kurzen Zeitspannen – eine Iteration dauert meist zwischen zwei und vier Wochen – ist schlicht nicht genug Zeit, stets alle der vorgeschriebenen Praktiken durchzuführen. Als Ausweg werden die Praktiken daher nochmals unterschieden: in solche, die pro Projekt einmalig ausgeführt werden müssen, und in diejenigen, die tatsächlich in jeder Iteration zur Anwendung kommen müssen.

Das bereits erwähnte Threat Modeling ist dafür genauso ein Beispiel wie das Durchführen statischer Code-Analysen. Letztere vorzugsweise automatisiert mithilfe eines Continuous Integration Servers.

Dazwischen führt Microsoft eine dritte Kategorie von Praktiken ein – die sogenannten „Bucket Practices". In wechselnder Folge soll pro Iteration jeweils eine Aktivität aus den Bereichen Design Review, Verifikation und Response Planning durchgeführt werden. Mit diesen Anpassungen wird der Microsoft SDL für agil arbeitende Teams relevant und enthält sinnvolle Aktivitäten und Techniken, die helfen, das Thema Cybersecurity angemessen zu berücksichtigen.

Cybersecurity und Scrum

Scrum ist das wohl am häufigsten eingesetzte agile Vorgehensmodells und zeichnet sich vor allem durch seine Einfachheit aus. Es gibt im Grunde nur drei Rollen, wenige Artefakte und nur eine Handvoll vorgegebener Events, wie beispielsweise das Sprint Planning oder die Retrospektive. Alles, was das Team in einem Sprint zu tun beabsichtigt, wird im zugehörigen Sprint Backlog festgehalten. Dies stellt sicher, dass alle Aktivitäten zu Beginn des Sprints bekannt und bei der Planung berücksichtigt werden können.

Wie hält es Scrum nun mit der Cybersecurity? Es liegt nahe, alles mit Bezug zur Security ebenfalls im Backlog festzuhalten. Wie anderen Backlog Items auch, wird einzelnen Security-Aktivitäten auf diese Weise ein Geschäftswert und eine Priorität zugewiesen.

Durch die damit einhergehende Transparenz lassen sich Missverständnisse zwischen Product Owner und Team über die Erledigung securityrelevanter Aktivitäten („Ich dachte, das sei selbstverständlich …") vermeiden. Für die konkrete Gestalt der securitybezogenen Backlog Items gibt es verschiedene Möglichkeiten. Am unmittelbarsten ist sicherlich die Aufnahme eines Security Features in das Backlog, das als Gegenmaßnahme im Rahmen des Threat Modeling identifiziert wurde. Die Einführung von https an Stelle des bisher genutzten http-Protokolls wäre ein Beispiel dafür. Finden sich im Backlog bereits Spikes, mit denen sich das Team hin und wieder einen Überblick über technologische Alternativen verschafft, kann dieser Typus auch verwendet werden, um z.B. die Implementierung verschiedener Krypto-Algorithmen miteinander zu vergleichen. Sind bestimmte Schwachstellen bekannt, die sich aber nicht sofort umsetzen lassen, kann das in Form eines Security Debt analog zur Technical Debt ins Backlog aufgenommen werden.

Schließlich sind auch Attacker Stories eine effektive Möglichkeit sicherheitsrelevante Aspekte zu beleuchten. In Anlehnung an die User Stories wird formuliert, was ein Angreifer zu tun beabsichtigen könnte, um eines seiner

unheilvollen Ziele zu erreichen. Z.B. könnte ein Botnetz-Betreiber versuchen wollen, unautorisiert ein Software-Update einzuspielen, um das so angegriffene System in einer Denial-of-Service-Attacke zu benutzen. Seiner Natur nach ist das Formulieren einer Attacker Story also weniger formal als das bereits beschriebene Threat Modeling. Die Anwendung dieser Methode zielt vielmehr darauf ab, die Kreativität und Fantasie der Entwickler in das Thema einzubeziehen. Es wird klar, dass keine zusätzlichen Planungsinstrumente eingeführt werden müssen. Entsprechend genutzt, kann das Product Backlog alle securityrelevanten Aktivitäten aufnehmen und lenken. Ergänzend dazu ist es sinnvoll, auch die „Definition of Done" des Teams entsprechend zu ergänzen. Teams halten darin fest, welche Qualitätsmerkmale jede User Story für ihre Akzeptanz erfüllen muss.

Gerade für immer wiederkehrende Aufgaben wie die statische Code-Analyse, das Fuzzy Testing von Eingabemasken oder die Überprüfung von Drittanbieter-Code auf bekannte Schwachstellen bietet sich eine Aufnahme in die „Definition of Done" an.

Aller Interdisziplinarität und Cross-Funktionalität zum Trotz sind agile Teams gut beraten, sich auch immer wieder einmal externe Hilfe zu holen. Cybersecurity ist ein weites Feld und kaum einem Team wird es gelingen, für alle Fragestellungen genügend eigene Expertise aufzubauen. In größeren Organisationen gibt es oftmals bereits ein System- oder Architekturteam, das den Entwicklungsteams beratend zur Seite steht. Es ist im ureigenen Interesse der Organisation, in diesem Team auch das Thema Security entsprechend zu repräsentieren. Teams kleinerer Unternehmen sollten an dieser Stelle nicht davor zurückschrecken, sich entsprechende Hilfe bei externen Dienstleistern einzukaufen. Die Angebote reichen von der initialen Unterstützung bei der Erstellung eines ersten Threat Models über Design- und Code-Reviews bis hin zum Penetration Testing, also dem simulierten Angriff auf ein Live-System.

Nur abgesichert zum Erfolg

Zahlreiche und zum Teil haarsträubende Berichte zeigen, dass das Internet der Dinge nur für diejenigen Hersteller Erfolg verspricht, die es verstehen, Cybersecurity maßgeblich in ihre Produktlebenszyklen zu integrieren. Cybersecurity kann nicht einfach als Add-On zugekauft werden. Vielmehr müssen Entwickler und Betreiber gleichermaßen die relevanten Punkte verstehen – zumal das Vertrauen der staatlichen Behörden in die selbstregulierenden Kräfte des Marktes schwindet und in den kommenden Jahren mit einer ganzen Reihe weiterer Vorschriften zu rechnen ist, die die bereits existierenden ergänzen werden. Das Cybersecurity Framework des NIST wird vor allem für den US-amerikanischen Markt maßgeblich sein. Auch abseits (noch) nicht regulierter Märkte sind Hersteller gut beraten, sich eingehender mit den verfügbaren Guidance-Dokumenten auseinanderzusetzen. Agilität und das erforderliche Bewusstsein, um sichere Produkte entwickeln zu können, schließen sich dabei keineswegs aus. Entsprechende Anpassungen vorausgesetzt, haben agile Vorgehen sogar einige Vorteile, da sie von Anfang an Kontinuität und Feedback einen hohen Stellenwert einräumen. *jk*

Literatur & Links

[1] https://www.fda.gov/downloads/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm482022.pdf
[2] https://www.fda.gov/downloads/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm482022.pdf
[3] https://www.nist.gov/cyberframework
[4] https://www.schneier.com/blog/archives/2017/02/security_and_pr.html
[5] http://www.microsoft.com/en-us/sdl/

Dr. Tobias Kästner
ist promovierter Physiker und hat mehrjährige Erfahrung in der Entwicklung vernetzter Medizingeräte. Seit 2016 ist er bei Method Park als Expert Engineer für das Internet der Dinge tätig.
Tobias.Kaestner@methodpark.de

Michael Lerch
arbeitet seit 2000 bei Method Park als Software Engineer. Mehr als zehn Jahre leitete er das Team „Medical Devices", bevor er 2016 die Leitung des Teams „Internet of Things" übernahm.
Michael.Lerch@methodpark.de