Verifysoft TECHNOLOGY

Security für vernetzte Medizingeräte

Viele Medizingeräte enthalten Mikrocontroller und werden durch Software gesteuert. Je nach Einsatzbereich und Kritikalität sind die Anforderungen an die Softwarequalität durchaus unterschiedlich. Man beurteilt die Programme hinsichtlich ihrer funktionalen Sicherheit (Safety) sowie ihrer Sicherheit gegenüber Angriffen von außen (Security).

Die Bedeutung der funktionalen Sicherheit für in der Medizin eingesetzte Geräte ist klar. Seit der Vernetzung dieser Geräte rückt jedoch der Bedarf an Sicherheit gegen Angriffe mehr in den Fokus. Heute kommunizieren medizinische Geräte über das Internet mit Ärzten, untereinander und auch mit anderen Institutionen. Dieses „Internet of Things“ (IoT) eröffnet große Chancen, birgt aber auch große Risiken, denen Softwarehersteller durch umfangreiche Test- und Analysemethoden begegnen müssen.

Was versteht man unter vernetzten Medizingeräten?

Bei vernetzten Medizingeräten kann man zwischen stationär, mobil und als Implantat eingesetzten Geräten unterscheiden. Zu den stationär eingesetzten Geräten zählen z.B. Ultraschall-, Röntgen- und MRT-Geräte, die – verbunden mit dem Internet – Daten direkt zur Auswertung weiterleiten und der elektronischen Patientenakte in einer zentralen Datenbank anfügen. Netzfähige mobile Geräte sind etwa Überwachungsgeräte für Herzfrequenz, Blutzucker- und Sauerstoffgehalt sowie Infusionspumpen. Zu den Implantaten zählen u.a. Herzschrittmacher, Blasen- und Zwerchfellstimulatoren.

Vorteile und Chancen vernetzter Medizingeräte

Netzfähige Medizingeräte können Kosten senken sowie Lebensqualität und Überlebenschancen von Patienten drastisch erhöhen. Sie ermöglichen es Ärzten etwa, die Vitalfunktionen ihrer Patienten remote zu überwachen und Medikamentendosierungen durch Fernsteuerung von Infusionspumpen anzupassen.

Die Möglichkeit der Überwachung von Patienten rund um die Uhr, verbunden mit im Notfall automatisch ausgelösten Alarm- und Standortmeldungen, verkürzt überlebenswichtige Reaktionszeiten signifikant.

Direkt miteinander kommunizierende Geräte können im medizinisch vertretbaren Umfang auch selbständig agieren. Wird z.B. durch eine kontinuierliche Blutzuckermessung eine drohende Unterzuckerung erkannt, so kann unverzüglich eine Meldung an eine Infusionspumpe erfolgen, die eine Glukosemenge injiziert.

Lebens- oder gesundheitsgefährdende Situationen durch Ausfall oder Fehlfunktion von medizinischen Geräten sind glücklicherweise selten, vereinzelt aber dennoch zu verzeichnen. Softwaregesteuerte Geräte sind heute vielfach in der Lage, Selbstdiagnosen durchzuführen und – sofern netzgebunden – Fehlfunktionen zu melden. Damit wird die Ausfallsicherheit erheblich verbessert.

Mit vernetzten Medizingeräten verbundene Risiken

Durch die stark gewachsene Vernetzung der Geräte wird jede individuelle Sicherheitslücke eines Gerätes gefährlicher. Netzwerkfähige Geräte besitzen Schnittstellen, welche immer als potenzielle Einfallstore für Angriffe von außen zu betrachten sind. So kann ein Angreifer hierüber, unter Ausnutzung von Sicherheitsschwachstellen, die Kontrolle über das Gerät erlangen. Solche Angriffe können weitreichende Folgen haben. Medizinische Geräte wurden mittelbar bereits genutzt, um in Infrastrukturen von Krankenhäusern und Gesundheitszentren einzubrechen. Bekannt geworden sind diese Einbrüche unter dem Namen „MEDJACK“ (Medical Device Hijack).

Ausgenutzt wurden unterschiedliche Schwachstellen wie veraltete Betriebssysteme oder unsichere Programmfunktionen. Durch das erfolgreiche Eindringen in medizinische Systeme lässt sich hoher Sach- und Personenschaden anrichten. So können Patienten- und Personaldaten abgegriffen, weitere Geräte infiziert oder wichtige Daten böswillig verändert oder verschlüsselt werden.

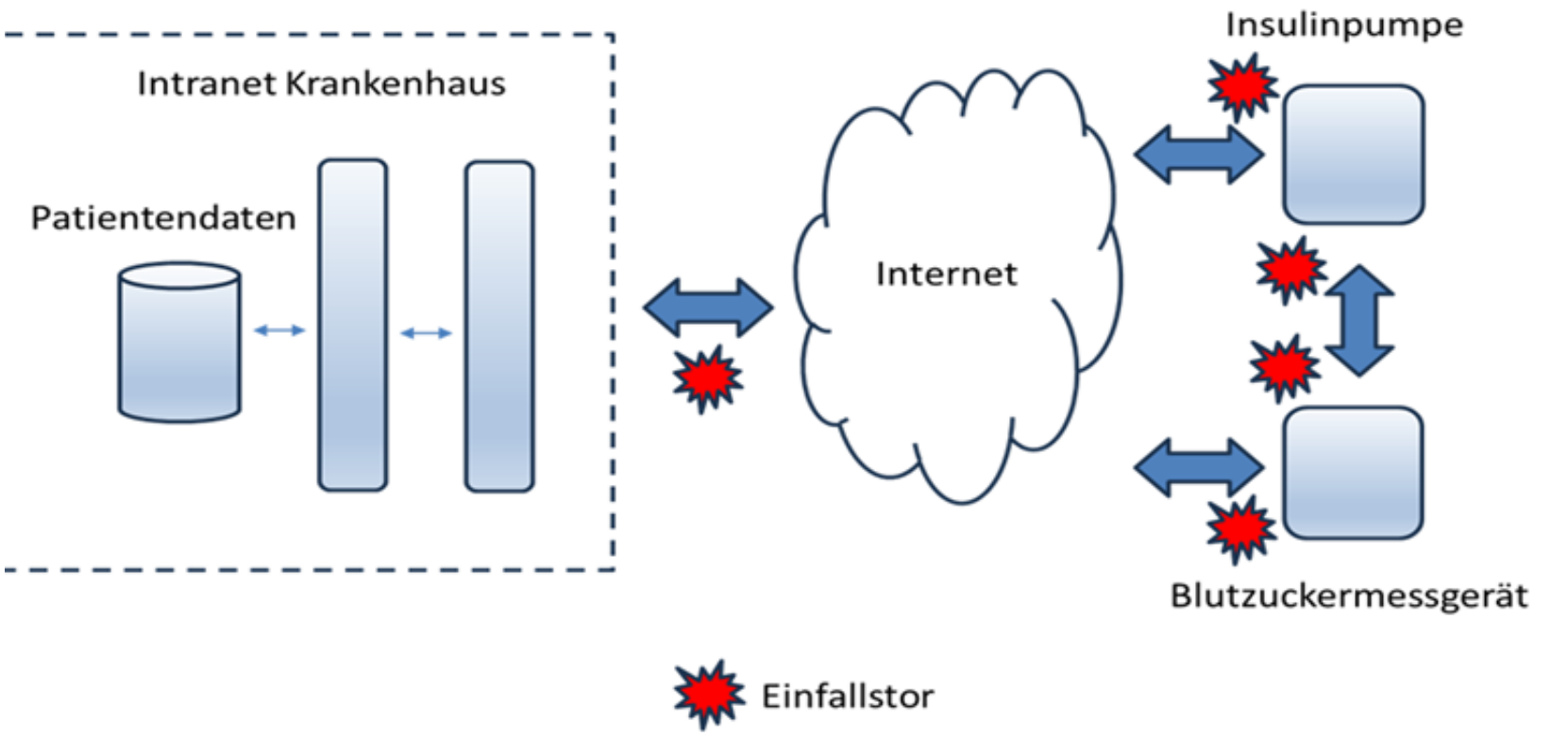

Abb. 1: Risiko vernetzte Medizingeräte

Die Hintergründe sind unterschiedlicher Natur und reichen von persönlichen kriminellen Motiven über typische Cyberverbrechen wie Ransomware-Attacken bis hin zu einem möglichen Terrorakt durch Manipulation möglichst vieler vernetzter, lebenserhaltender Geräte.

Wie werden solche Angriffe in der Regel durchgeführt?

Es gibt viele Einfallstore, die ein Angreifer auszunutzen kann. Oft werden mehrere Methoden kombiniert.

Ein Angreifer kann in einem Netzwerk als „Man-in-the-Middle“ agieren, um den Datenverkehr auszuspähen. Durch Extrahieren von Zugangsdaten und die Suche nach Schwachstellen ist es möglich, Schadcode zu injizieren. Dieser kann viele Ausprägungen haben. Dies reicht vom Zugriff aus der Distanz über Keylogger bis hin zum Sperren/Verschlüsseln von Daten oder ganzen Geräten, um diese gegen Zahlung hoher Geldbeträge wieder freizugeben – so geschehen bei zahlreichen Ransomware-Angriffen auf Einrichtungen des Gesundheitswesens.

Denial-of-service-Attacken hingegen setzen auf das Überfluten eines Datenempfängers mit einer sehr großen Menge an Datenpaketen. Dies führt zu einer eingeschränkten oder auch komplett verhinderten Funktion des Angegriffenen.

Aber auch die Software der einzelnen Geräte selbst (z.B. Firmware) kann als Sprungbrett für Angriffe genutzt werden. Diese Software besteht oft aus einem Eigenentwicklungsanteil und extern entwickelten Komponenten. Um Sicherheit gegenüber Angriffen von außen zu gewährleisten, müssen Entwickler sowohl selbst erstellten Code als auch externe Softwarekomponenten auf Sicherheitslücken hin überprüfen.

Anzuwendende Normen und Richtlinien für die Softwarehersteller

Zur Vermarktung von Medizinprodukten in Europa ist die Medizinprodukteverordnung (MDR) bindend. Diese schreibt eine Risikomanagement- und Risiko-Nutzen-Analyse vor. Einen Leitfaden zum Risikomanagementprozess hinsichtlich der funktionalen Sicherheit eines Produktes bietet die ISO14971. Nicht bindend, aber sehr hilfreich ist die AAMI TIR57 als Erweiterung der ISO14971, um den Risikomanagementprozess auf Cybersecurity auszuweiten.

Grundsätzlich sind für den Prozess von der Produktidee bis zur Marktherausnahme die von der EU-Kommission veröffentlichten Normen IEC 62304, IEC 82304-1 und IEC 60601-1 anzuwenden. Für Medizinprodukte, die in Netzwerken betrieben werden, ist zudem die Norm IEC 80001-1 relevant. Zu erwähnen ist auch auf die Richtlinie (EU) 2022/2555 des Europäischen Parlaments und des Rates vom 14. Dezember 2022 über Maßnahmen für ein hohes gemeinsames Cybersicherheitsniveau in der Union.

Analyseverfahren

Zur Gewährleistung der Softwarequalität kommen während des Softwareentwicklungsprozesses grundsätzlich zwei komplementäre Test- und Analyseverfahren zur Anwendung: Die statische Codeanalyse und das dynamische Testen.

Die statische Analyse untersucht Quellcode- und Binärdateien im Hinblick auf enthaltene kritische Fehler und Sicherheitsschwachstellen, ohne den Code ausführen zu müssen.Die dynamische Analyse beurteilt durch Ausführen von Tests dagegen das Laufzeitverhalten der Applikation oder des Moduls. Aufgedeckt wird damit überwiegend funktionales Fehlverhalten, das aber durchaus auch Sicherheitsschwachstellen bedingen kann. Zum Ausführen von Tests muss bereits funktionsfähiger Programmcode vorhanden sein. Die statische Analyse hingegen ist bereits früher – begleitend zur Implementierung – einsetzbar.

Im Entwicklungszyklus wird die statische Security Analyse (SAST) je nach Entwicklungstand auch bereits auf noch nicht lauffähige Applikationsteile (z.B. Funktionen) angewandt. In dieser Phase lassen sich nach Analyse mit Tools wie CodeSonar notwendige Korrekturen im Quellcode frühzeitig und damit noch kostengünstig umsetzen.

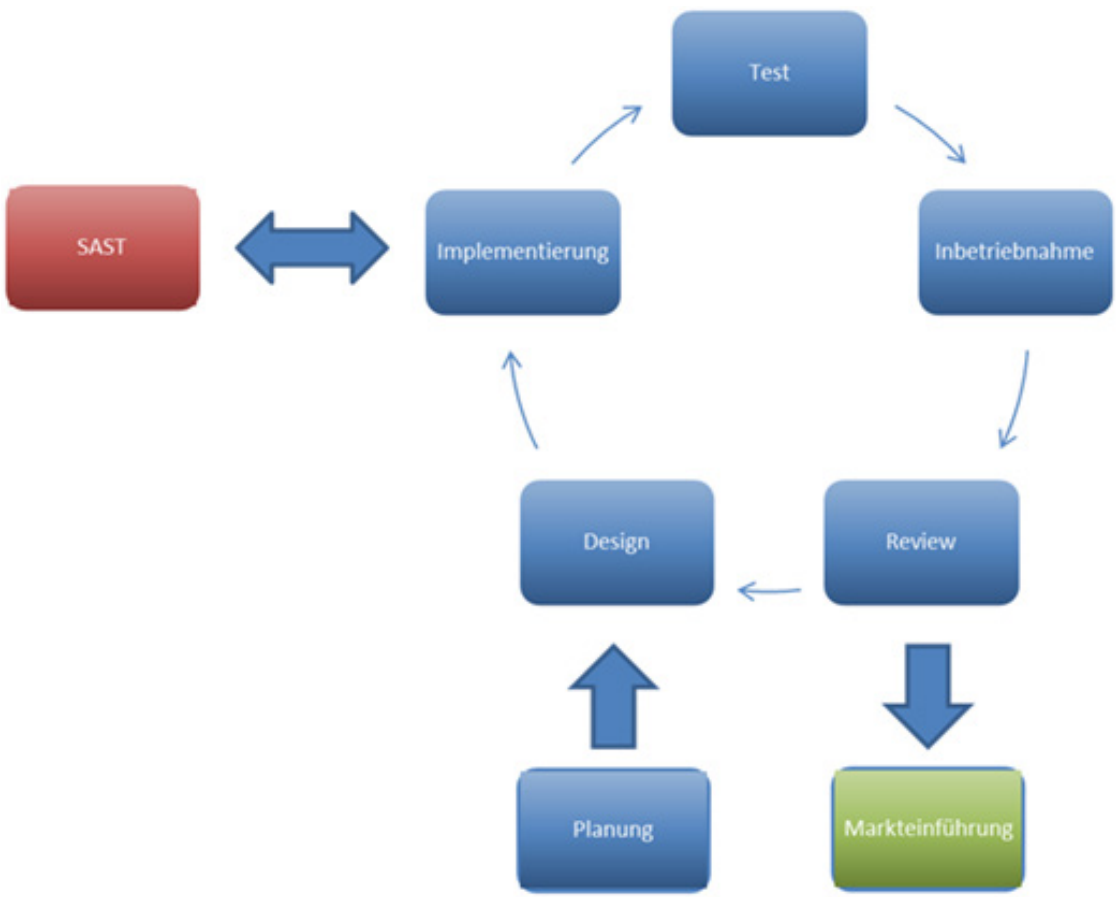

Abb. 2: SAST im Software-Lifecycle

Eingesetzt werden dabei in der Regel verschiedene Werkzeuge zur statischen Analyse. Einige Tools scannen den Quellcode und sind spezialisiert darauf, Schwachstellen wie Speicherüber- und Unterläufe, Format-String-Probleme, hart codierte Passworte, geringe Verschlüsselungstiefen und eingebaute Hintertüren (Backdoors) aufzudecken. Andere Werkzeuge konzentrieren sich auf die Analyse von Binärdateien wie Bibliotheken und deren Abhängigkeiten zu eventuell weiteren eingebundenen Komponenten, um potenzielle Sicherheitslücken zu identifizieren. Insbesondere wird dabei auch nach Code-Konstrukten gesucht, die aktuell zu bekannten Sicherheitsschwachstellen geführt haben. Durch die Kombination dieser Vorgehensweisen lassen sich viele potenziell kritische Sicherheitslücken im Quellcode beheben, bevor die Software in Betrieb genommen wird.

Bei eigenentwickeltem Code empfiehlt sich die oben beschriebene Quellcodeanalyse. Das gleiche gilt für eingebundene Open-Source-Komponenten, für die der Quellcode im Zugriff steht. Man spricht hier von „0-Day-Analyse“, da damit noch unbekannte Sicherheitsschwachstellen aufgedeckt werden können. Im Hinblick auf die Open Source-Komponenten bietet sich zudem eine Überprüfung auf bekannte Sicherheitsprobleme (N-Day-Ermittlung) mittels Recherche in einschlägigen Vulnerability-Datenbanken an. Entwickler erhalten so Informationen über die Kritikalität einer in einer Komponente enthaltenen Sicherheitsschwachstelle sowie eventuell bereits vorhandene Korrekturen.

Kommerzielle Komponenten liegen zumeist nur als Binärdateien vor. Hier empfiehlt sich zunächst eine statische 0-Day-Binärcodeanalyse. Die sicherlich notwendige N-Day-Ermittlung hinsichtlich in der Binärdatei eventuell enthaltener Open Source-Komponenten ist allerdings erst durchführbar, wenn diese identifiziert sind. Das leistet die Software Composition Analysis (SCA): Dabei werden beispielsweise Zeichenketten gesucht, die auf verwendete Open-Source-Komponenten hinweisen. Analysetools wie CodeSentry ermöglichen zudem auch ohne Zugriff auf den Quellcode skalierbare Analysen. Anschließend können die Datenbanken auf bereits bekannte Schwachstellen abgefragt werden.

Grundsätzlich sollte Software, die in vernetzten Geräten zum Einsatz kommt, einer statischen „Taint Data Analyse“ – einem virtuellen Penetrationstest – unterzogen werden. Man untersucht dabei, wie eingespeister Schadcode in der Applikation weitergeleitet würde und welche Funktionen davon betroffen wären. Dadurch lassen sich bereits im Vorfeld Gegenmaßnahmen treffen.

Zum Abschluss, wenn die vollständige Applikation gebaut worden ist, sollte noch eine Sicherheitsattributsanalyse durchgeführt werden. Hierbei wird überprüft, ob z.B. von der Möglichkeit, den Compiler zusätzliche Sicherheitsmechanismen in den Binärcode einbauen zu lassen, Gebrauch gemacht wurde.

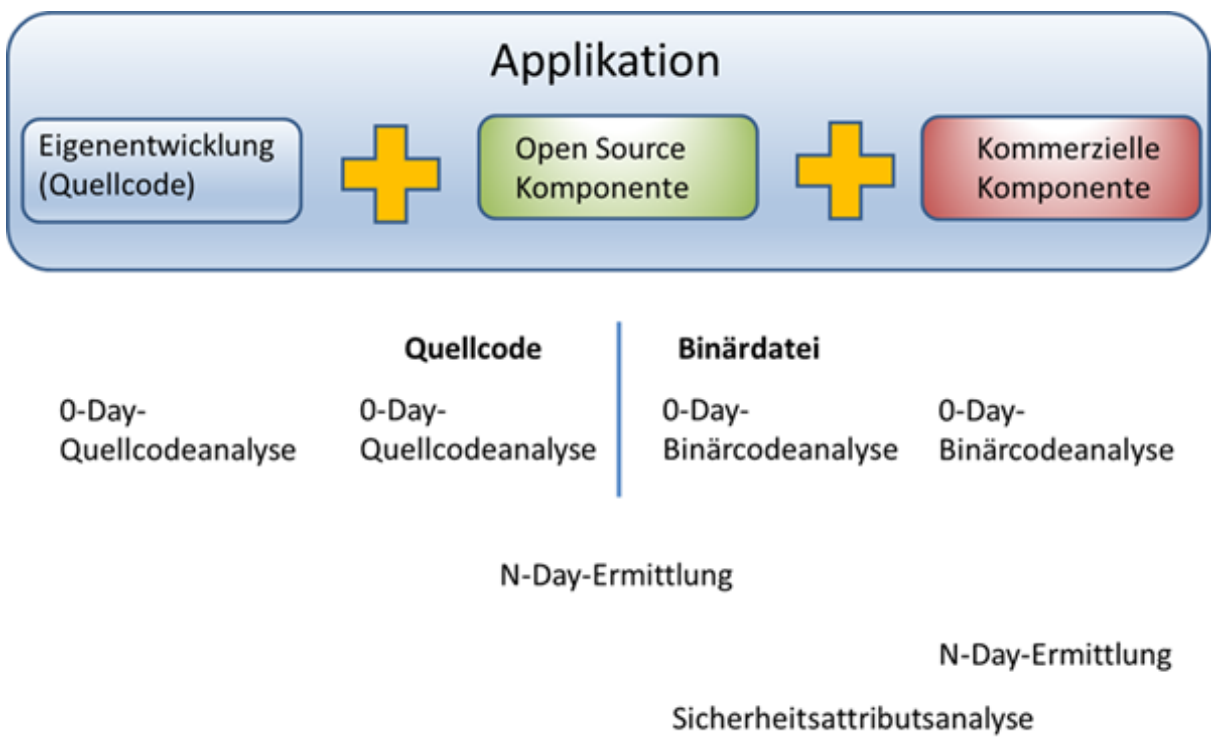

Abb. 3: Einsatz statischer Analysewerkzeuge (Security)

Verifysoft

TECHNOLOGY

Die dynamische Analyse ist ein zur statischen Analyse komplementäres und unverzichtbares Verfahren, um die Zuverlässigkeit von Software sicherzustellen. Das dynamische Testen kann allerdings erst dann erfolgen, wenn ablauffähige Teile der Applikation (z.B. Module) vorliegen.

Im Rahmen dieser Tests wird die einwandfreie Funktionalität inklusive der für die Security relevanten Komponenten überprüft. Ein Werkzeug zur Messung der Testabdeckung stellt sicher, dass keine Tests ausgelassen wurden.

Unter Einhaltung der beschriebenen Prozesse und Vorgaben zu Planung, Design, Entwicklung und Analyse von Software zum Einsatz in Medizingeräten kann ein hohes Maß an Sicherheit gewährleistet werden.

Autoren:

Dipl-Ing. Royd Lütke

Pre-Sales und Support statischer Analysetools

Verifysoft Technology GmbH

Artur Hirsch

Support statischer Analysetools

Verifysoft Technology GmbH

Veröffentlicht am 17. April 2025

Weitere Informationen finden Sie unter www.verifysoft.com

E-Mail: info@verifysoft.com