Security-Aspekte

certitudo GmbH
Quality for safety critical systems

Security-Aspekte beim Einsatz von COTS- und Open Source-Softwarekomponenten für kritische Systeme

Der Einsatz von COTS und Open Source nimmt über alle Branchen hinweg stetig zu. Grund dafür ist der zunehmende Wettbewerbsdruck, Entwicklungskosten und Entwicklungszeit zu reduzieren und gleichzeitig einen Innovationsvorsprung gegenüber Mitbewerbern zu erzielen. Dieser Artikel gibt einen Überblick über die Security-Aspekte, insbesondere die Gefährdung durch Hackerangriffe, bei der Verwendung von COTS- und Open Source-Softwarekomponenten in kritischen Systemen. Hierbei ist auch die immer stärkere Vernetzung von IT-Systemen von Bedeutung. Daher werden beide Themen im Zusammenhang erörtert.

01 Kritische Systeme – ein Überblick

Kritische Systeme können Teil einer kritischen Infrastruktur (KRITIS) sein. Systeme aus diesem Bereich stehen nach dem IT-Sicherheitsgesetz [1] unter besonderer Kontrolle des BSI.

Hier sind beispielsweise Kraftwerke, Verkehrsleitsysteme, Krankenhäuser und Raffinerien zu nennen, bei deren Ausfall oder deren Beeinträchtigung mit erheblichen volkswirtschaftlichen Schäden bis hin zu Personenschäden zu rechnen ist.

Ein Fallbeispiel mit möglicher Personengefährdung ist die Verschlüsselung von Patientendaten im Lukas-Krankenhaus im Februar 2016 in Neuss durch einen Krypto-Trojaner. Dieser wurde durch Öffnen eines Email Anhanges aktiviert. In dessen Folge mussten Behandlungen von Patienten und Operationen verschoben werden [2].

Kritische Systeme mit direkter Personengefährdung sind Safety-Critical Systems. Diese Systeme müssen im Sinne der funktionalen Sicherheit entwickelt und betrieben werden. Dies gilt zum Beispiel für medizinische Geräte, Airbag Steuergeräte, Stellwerk Signalisierungen oder Triebwerke in einem Flugzeug. Safety-Critical Systems können wiederum Bestandteil einer kritischen Infrastruktur sein.

Beispielsweise kann es beim Ausfall eines Atomkraftwerks nicht nur zu großen Engpässen in der Stromversorgung kommen, sondern es steigt auch das Risiko für den unkontrollierten Austritt von radioaktiver Strahlung und damit die Gefahr von Personenschäden.

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de

02 Trends in der IT

2.1 Vernetzung von Systemen

Aus Kosten-, Qualitäts- und Innovationsgründen werden industrielle Steuerungsanlagen zunehmend in andere Geschäftssysteme integriert, um ein besseres Monitoring, eine vereinfachte Wartbarkeit und eine erhöhte Produktivität zu erreichen. Aber auch durch Innovationen in anderen Branchen setzt sich dieser Trend fort. Neue Fahrzeugkonzepte wie Autonomous Drive und Connected Car und intelligente Verkehrsleittechniken erfordern eine umfassende Vernetzung und intensiven Datenaustausch zwischen den Fahrzeugen (C2C Kommunikation) und dem Fahrzeug mit der Verkehrsinfrastruktur (C2I Kommunikation).

Es zeichnet sich allgemein eine immer stärkere Vernetzung von zuvor proprietären Anwendungen hin zu vernetzten, offenen und fernsteuerbaren und -wartbaren Systemen ab. Dieser Trend spiegelt sich in innovativ klingenden Wortschöpfungen wie Internet of Things (IOT) und IT 4.0 wieder.

Laut Marktforschungsstudie „The Top Tech Trends To Watch: 2016 To 2018“ [3] von Forrester Research steht Konnektivität bei den IT Trends an erster Stelle. Interessant und nicht überraschend ist in diesem Zusammenhang, dass in der Studie bereits an fünfter Stelle neue Anforderungen an Security und Risiko erwartet werden.

2.2 Verwendung von COTS und Open Source

Die zunehmenden Sparmaßnahmen bei Bund und Ländern im Gesundheitswesen, im öffentlichen Dienst und beim Militär sowie der Kosten- und Innovationsdruck und dem damit verbundenen Know-how Einkauf in der Industrie hat zu einem anhaltend hohen Einsatz von COTS- und Open Source-Produkten geführt.

Ein großer Anteil von COTS- und Open Source-Software wird in Betriebssystemen (Windows, Linux), im Office-Bereich (MS Office, OpenOffice), im Entwicklungsumfeld (z.B. GNU-Compiler Toolchain) sowie in der Qualitätssicherung (Open Source Produkte wie CUnit, Jenkins) eingesetzt.

Open Source-Software wird auch in kritische Anwendungen integriert, da sie spezielle Funktionalitäten (z.B. Kommunikationsprotokolle, Bilderkennungsverfahren) bieten.
Das Ergebnis von mehr als 1.000 durchgeführten On-Demand-Audits durch Black Duck zeigt, dass Open Source-Software je nach Branche ca. 23 bis 46 % des Gesamtcodes ausmachen [4].

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de

%	Branche
46	Gesundheitswesen, Health Technology, Biowissenschaften
41	Einzel- und Onlinehandel
41	Cybersecurity
38	Bigdata, AI, BI, Maschinelles Lernen
37	Unternehmenssoftware/Software as a Service (SaaS)
37	Internet und Mobile Apps
36	Andere (Educational Technology, Marketing Technology, IoT)
35	Internet und Software Infrastruktur
35	Fertigung, Industrie, Robotik
34	Computer Hardware und Halbleiter
31	Telekommunikation und Funk
28	Virtual Reality, Gaming, Unterhaltung, Medien
28	Financial Services und Technology (FinTech)
27	Energie und Umwelt
23	Luft- und Raumfahrttechnik, Transport und Logistik

Tabelle 1:
Open Source Einsatz nach Branchen [4]

Im Folgenden werden die Security-Aspekte beim Einsatz von COTS-Software, die nicht im Quellcode vorliegt, und Open Source-Software in kritischen Infrastrukturen betrachtet. Die Software, sowohl COTS als auch Open Source, kann ein eigenständiges Produkt (z.B. Bürosoftware, Betriebssystem, Datenbank) oder Teil eines Gerätes (z.B. Software im Netzwerkrouter) sein.

03 Betrachtung von ausgewählten Security-Risiken

3.1 Vernetzung von Systemen

Eine höhere Vernetzung führt zu einem größeren Angriffsvektor und bietet damit ein größeres Risiko hinsichtlich der funktionalen Sicherheit. Ein Angreifer muss nicht mehr vor Ort sein, um ein kritisches System zu kompromittieren, sondern kann nun – durch die Abkehr von gekapselten hin zu offenen an das Internet gekoppelten Systemen – aus der Ferne mögliche Schwachstellen identifizieren und Folgeangriffe durchführen.

Als Fallbeispiel sei hier der Jeep Cherokee Hack in 2015 durch Sicherheitsexperten genannt. Die Fahrzeugsteuerung konnte mit einem Angriff auf den CAN-C Bus manipuliert werden [5]. Durch die Anbindung des nicht sicheren Entertainment Systems UConnect wurde das bestehende Sicherheitskonzept ausgehebelt. Das ursprünglich sichere Design sah es nicht vor, die Identität einer Nachricht zu prüfen.

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de

Eine komfortable und sehr effiziente Möglichkeit, gezielt unzureichend geschützte Systeme aufzuspüren und bekannte Schwachstellen auszunutzen, bietet die Suchmaschine Shodan [6].

Es ist keine klassische Suchmaschine wie z.B. Google, die nach Webinhalten crawlt und Suchergebnisse nach Relevanz sortiert liefert. Shodan ist eine spezielle Suchmaschine für alle Arten von Netzwerk-Geräten (Webcams, Drucker, Router, Server, IoT), die bestimmte Dienste im Netz zur Verfügung stellen. Hierbei führt der Shodan Crawler systematische Portscans auf zufällig ausgewählte IP Adressen aus und indiziert die Antworten (sogenannte Servicebanner) der Netzwerkgeräte. Über eine Suchanfrage werden die indizierten Inhalte der Servicebanner entsprechend der Suchkriterien ausgewertet und die zutreffenden Service Banner in Listenform mit den entsprechenden Portnummern und IP-Adressen ausgegeben.

Beispiele für Shodan Suchanfragen

- vuln:CVE-2014-0160
  liefert alle Server mit OpenSSL Heartbleed Schwachstelle
- category:malware
  spürt Botnet Server auf
- product:conpot
  findet Honeypots

Informationen zu Heartbleed

Heartbleed ist eine schwerwiegende Sicherheitslücke in OpenSSL v1.0.1 bis v1.0.1f. Dieser Fehler erlaubt das Auslesen des kompletten Arbeitsspeichers (Benutzernamen, Passwörter, Private Keys) des angegriffenen Systems. Mit Kenntnis der Passwörter sind weitergehende Angriffe auf die Infrastruktur möglich.

Eine aktuelle Heartbleed Abfrage mit der Suchmaschine Shodan (https://www.shodan.io/search?query=vuln%3ACVE-2014-0160) vom 30.08.2017 lieferte 136.262 Treffer zurück. Diese hohe Zahl ist insofern erstaunlich, da die Sicherheitslücke mit Release v1.0.1.g am 07.04.2014 behoben wurde und durch die Presse einer größeren Öffentlichkeit bekannt gemacht wurde. Nach einer Schwachstellenanalyse durch den BSI gilt OpenSSL v1.0.1g hinsichtlich der Verschlüsselungstechnik als weitgehend sicher [7].

Dieses Ergebnis verdeutlicht, wie wichtig ein funktionierendes, kontinuierliches Security Management ist.

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de

3.2 COTS- und Open Source-Software

COTS- und Open Source-Software wird unter Berücksichtigung der jeweiligen Sicherheitsanforderungen in allen Bereichen, vom „nicht sicheren Schreibtisch“ bis zum sicheren ESTW eingesetzt.

Durch ihre große Verbreitung sind sie ein besonders lukratives Ziel für Angreifer. Häufig werden diese Komponenten in unkritischen Anwendungen verwendet, da sie für diesen Einsatz nicht qualifiziert werden müssen.

Der weitaus größte Anteil an Malware-Attacken (Viren, Trojaner, Würmer, Skripte, Ransomware) entfällt auf Windows Systeme [8].

Die Ausnutzung einer Sicherheitslücke in einem nicht kritischen Umfeld ist besonders problematisch, wenn es einem Angreifer durch Folgeangriffe gelingt, weiter in das interne Netz vorzudringen und die funktionale Sicherheit von kritischen Systemkomponenten gefährdet.

Daher sind bei der Verwendung von COTS- und Open Source-Software in einer kritischen Infrastruktur geeignete Maßnahmen zu definieren, um die Security-Eigenschaften über die Laufzeit des Systems gewährleisten zu können.

Auflistung von bewerten Security-Maßnahmen [9] zur Risikominimierung beim Einsatz von COTS und Open Source:

- Identifikation der COTS-/Open Source -Komponenten
- Abwägung Business Value gegenüber tolerierbares Risiko
- Identifikation der Schnittstellen zu anderen Komponenten
- Schaffung einer sicheren Infrastruktur zur Einbettung von COTS-/ Open Source -Komponenten
- Zugriffskontrollmechanismen schaffen und kontinuierlich evaluieren
- Hersteller bezüglich Patches und Security Issues kontaktieren
- User und Security Foren konsultieren
- Expertise von Security Experten einholen
- Software Tests durchführen
  - White Box (Open Source)
  - Black Box (COTS), z.B. Fault Injection Test
- Kontrolle der Ein- und Ausgänge durch Software Wrapper
- Einsatz zertifizierter COTS-/Open Source -Komponenten
- Zeitnahe Installation von Security Patches
- Genaues Verständnis erlangen, WIE die COTS-/Open Source -Komponente arbeitet (z.B. Speichermanagment)
- Datalogging und automatische Analyse von Anomalien
- Regelmäßige, externe Audits
- Vorbereitet sein auf Security Vorfälle durch Etablierung von Notfallkonzepten

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de

3.3 Bewertung

Die steigende Komplexität stellt eine erhebliche Herausforderung für Hersteller und Betreiber von kritischen Infrastrukturen dar. Dies ist als Auswirkung der zunehmenden Vernetzung von nicht kritischen und kritischen Bereichen zu verstehen. Hinzu kommt die Verzahnung von Softwarekomponenten mit ganz unterschiedlichen Security- und Safety-Anforderungen aus Eigenentwicklung, Custom Software, COTS und Open Source und unterschiedlichen Releasezyklen. Um eine kritische Infrastruktur zu überwachen und Angriffe abzuwehren, steht eine Vielzahl mächtiger, größtenteils nicht kommerzieller Security Tools beispielsweise Portscanner (NMap), Disassembler, Exploit Tools (Metasploit Framework), Sniffer Tools (Wireshark), IDS (Intrusion Detection Systems) zur Verfügung. Leider bieten einige dieser Werkzeuge ebenfalls effiziente Möglichkeiten an, um Sicherheitslücken auszuspähen und Angriffe durchzuführen.

Vor diesem Hintergrund ist die stetige Zunahme von Hacker Angriffen ein ernst zunehmendes Risiko für die Security und Safety von kritischen Infrastrukturen.

Die nachfolgende **Tabelle 2**: BSI - Top 10 Bedrohungen von ICS in 2016 [10] basiert auf einer Analyse von Vorfällen durch den BSI. Ein Teil der Ergebnisse entstammt der Meldepflicht von Sicherheitsvorfällen gemäß IT Sicherheitsgesetz [1].

Industrial Control Systems (ICS) sind kritische System der Fabrikautomation und Prozesssteuerung, die teilweise zur kritischen Infrastruktur (KRITIS) gehören. Die blau hervorgehobenen Bedrohungen stehen tendenziell im Zusammenhang mit der Vernetzung und der Verwendung von COTS- und Open Source-Software.

Nr.	Art der Bedrohung
1	Social Engeneering und Phishing
2	Einschleusen von Schadsoftware über Wechseldatenträger und externe Hardware
3	Infektion mit Schadsoftware über Internet und Intranet
4	Einbruch über Fernwartunszugänge
5	Menschliches Fehlverhalten und Sabotage
6	Internetverbundene Steuerungskomponente
7	Technisches Fehlverhalten und höhere Gewalt
8	Kompromittierung von Extranet und Cloud-Komponenten
9	DDoS Angriffe
10	Kompromittierung von Smartphones im Produktionsumfeld

Tabelle 2:
BSI - Top 10 Bedrohungen von ICS in 2016 [9]

Die wachsenden Herausforderungen erfordern besondere Security-Maßnahmen, um einen sicheren Betrieb kritischer Infrastrukturen zu gewährleisten. Da diese Maßnahmen sehr vielschichtig ausfallen – siehe hierzu die BSI ICS-Security-Kompendien für Betreiber [11] und

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de

Hersteller von Komponenten [12] – ist ein ganzheitlicher Ansatz gemäß eines Information Security Management Systems (ISMS) unerlässlich.

In den kritischen Branchen wie z.B. Avionik, Automotive, Eisenbahnsignaltechnik, Medizintechnik und Kernenergie sind branchenspezifische Normen und Prozesse für die Sicherstellung der funktionalen Sicherheit etabliert, aber spezifische Anforderungen an Security fehlen bisher in der Regel. Der branchenübergreifende Security Standard der Normenreihe IEC 62443 Industrial Automation and Control Systems Security richtet sich vor allem an Betreiber und Hersteller von Safety-Critical Systems, die bisher keine eigenen Security Standards etabliert haben. Für elektrische Bahn-Signalanlagen wurde ein Leitfaden für die IT-Sicherheit auf Grundlage der IEC 62443 erstellt [13].

04 COTS- und Open Source-Software in Safety-Critical Systems

In diesem Abschnitt wird die Eignung von COTS- vs. Open Source-Software für Safety-Critical Systems genauer untersucht. Es stehen sich zwei Lager gegenüber, die ganz unterschiedliche Paradigmen (siehe Abschnitt 4.2 Security-Aspekte) verfolgen und miteinander im Wettbewerb stehen.

Da COTS-Software und Open Source-Software keine Individuallösungen darstellen, wird in aller Regel mehr Funktionalität bereitgestellt als im kritischen System benötigt wird. Um einen möglichen Safety Impact auszuschließen, ist es vorgeschrieben, nicht verwendete Funktionen zu deaktivieren. Dies ist mit Open Source-Software einfacher zu realisieren als bei COTS-Software, da der Code direkt modifiziert werden kann. Hierbei sind jedoch die Lizenzrechte zu beachten.

Wesentliche Gründe, COTS- oder Open Source-Software gegenüber Eigenentwicklungen in einem kritischen Produkt vorzuziehen, sind Kostenersparnis, signifikante Leistungsmerkmale und die Vermeidung von Vendor-Lock-In-Effekten. Die gewünschte Kostenersparnis ist von Fall zu Fall kritisch zu hinterfragen, da der Nachweis von Safety und Security von COTS- und Open Source-Software unter Umständen schwierig zu erbringen ist:

- Fehlende Security Expertise bezüglich der zugekauften Softwarekomponente, da diese nicht unbedingt zum Kern Know-How des Herstellers gehört (im Gegensatz zu Eigenentwicklungen).
- COTS und Open Source-Software sind generisch. Die Betrachtung des Gesamtkontextes (d.h. Einsatzzweck, Einsatzort, Konfiguration) und die Interaktionen mit dem Gesamtsystem (Schnittstellen, Vernetzung, Kritikalität) können sehr komplex sein.

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de

Die vorangegangenen Punkte sind für die Verifikation von COTS und Open Source-Software problematisch. Dies gilt insbesondere für COTS-Software da diese i.d.R. nur durch Black-Box Methoden getestet werden kann. Das Problem vergrößert sich, wenn die Dokumentation der zugekauften Software dünn ist.

Aus Zulassungssicht wird sowohl COTS- als auch Open Source-Software als Third-Party-Komponente betrachtet. Im Folgenden werden zentrale zulassungsrelevante Safety- und Security-Aspekte näher untersucht.

4.1. Safety-Aspekte

Die Integration von nicht qualifizierter COTS- und Open Source-Software in Safety-Critical Systems ist sehr aufwendig. Hierfür ist ein entsprechender Qualifikationsprozess gemäß branchenspezifischer Safety Standards zu beachten. Eine Gefahren- und Risikoanalyse ist in der Regel vorgeschrieben.

Für eine erfolgreiche Validierung von kritischen Komponenten muss zudem der Nachweis erbracht werden, dass die geforderte Code-Abdeckung für den angestrebten Safety Level erreicht wurde, d.h. reine Black-Box-Tests sind nicht hinreichend. Da bei COTS der Source Code für die Anwendungsentwicklung nicht zur Verfügung steht, kann dieser Nachweis nicht erzielt werden. Aus dem gleichen Grund entfallen Code Reviews. Als kompensatorische und vertrauensbildende Maßnahme für die unzureichenden Verifikationsaktivitäten ist für COTS-Software ein Proven-in-Use zwingend erforderlich. Eine zusätzliche, technische Schutzmaßnahme besteht in der Verwendung von Software Wrappern.

Da bei Open Source-Software die Sourcen frei zugänglich sind, entfällt diese Problematik hier gänzlich. Der Nachweis über die geforderte Codeabdeckung kann durch die Ausführung von Modultests belegt werden. Ebenso ist es möglich, den Source Code mit Hilfe von statischen Analyse Tools und per Code Review zu inspizieren und unter Einhaltung der lizenzrechtlichen Bestimmungen zu modifizieren.

Ein weiteres Problem ist in der Regel die Traceability zu den Anforderungen der Anwendung. Dies gilt gleichermaßen für COTS- und Open Source-Software. Die Beschreibung der Funktionalität und die entwicklungsbegleitende Dokumentation sind häufig nur rudimentär vorhanden, da COTS- und Open Source-Software typischerweise für den Massenmarkt und nicht für den spezifischen Einsatz in einem Safety-Critical System erstellt wurde. An dieser Stelle müssen gegebenenfalls Reverse Engineering Methoden eingesetzt werden, um die Software und den Entwicklungsprozess genauer zu determinieren.

Ein kritisches System ist auf Grund der extrem hohen Entwicklungskosten auf eine sehr hohe Laufzeit ausgelegt – je nach Branche 25 Jahre und darüber. In Folge der besonderen Marktanforderungen existieren im Gegensatz hierzu kurze Releasezyklen für

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de

COTS- und Open Source-Software. Daher kann es schnell zu einer Überalterung dieser Softwarekomponenten im kritischen System kommen. Dies kann ein großes Problem darstellen, wenn nach der Inbetriebnahme des kritischen Systems schwerwiegende Sicherheitslücken entdeckt werden, da diese auf Grund der geltenden Regularien für ein bereits zugelassenes System nicht einfach gepatcht werden können.
Im Folgenden werden zentrale zulassungsrelevante Security-Aspekte näher untersucht.

4.2. Security-Aspekte

Bei der Betrachtung von Security von COTS und Open Source stößt man auf zwei unterschiedliche Paradigmen:

- „Security Through Obscurity“ (COTS),
- „Security Through Disclosure“ (Open Source)

Die Geheimhaltung des Source Codes bei COTS kann zu mehr Sicherheit beitragen, da hierdurch Sicherheitslücken schwerer aufzuspüren sind. Umgekehrt können, durch die freie Verfügbarkeit des Source Codes bei Open Source, einfacher Sicherheitslücken entdeckt, aber auch schneller gefixt werden – was am Ende mehr Sicherheit bedeutet. Dies ist umso effektiver, je größer die Verbreitung des Codes ist.

Vor allem in Bereichen, die eine direkte Security-Funktion (z.B. Kryptographie) erfüllen, schafft Open Source Vertrauen und wird bevorzugt eingesetzt: Es ist in Open Source-Software weitaus schwieriger, unerkannt Spyware oder Backdoors zu platzieren als dies bei COTS-Software der Fall ist.

Eine gängige Methode um die Sicherheit von COTS- und Open Source-Software zu evaluieren, ist die Zahl der bekannten Sicherheitslücken (Vulnerabilities) zu ermitteln und zu bewerten.

Der Industriestandard für Common Vulnerabilities and Exposures (CVE) gewährleistet anhand eines Nummernsystems eine eindeutige Identifizierung von Schwachstellen. Beispielsweise trägt Heartbleed die ID CVE-2014-0160. Über eine CVE Datenbankabfrage [14] können für ein bestimmtes Softwareprodukt die bestehenden Sicherheitslücken ermittelt werden. Für eine genauere Beurteilung der Kritikalität wird jedem CVE Eintrag über ein standardisiertes Scoring System ein Schweregrad (kein, niedrig, mittel, hoch, kritisch) zugeordnet.

Die Security eines Softwareproduktes lässt sich allerdings nicht alleine aufgrund dieser Metriken (Anzahl der Sicherheitslücken, Schweregrad) beurteilen. Für die Evaluierung einer geeigneten Softwarekomponente sind weitere Faktoren wie Marktverbreitung, Patch- und Updatezyklen, Funktions- und Codeumfang, sowie die Compliance bezüglich Secure Development Life-Cycle (siehe nachfolgender Abschnitt) zu berücksichtigen.

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de

4.3 Security Standard IEC 62443

Auf Grund des Gefährdungspotenzials von Security-Vorfällen in Safety-Critical Systems erfolgt eine abschließende Betrachtung der Anforderungen gemäß Normenreihe IEC 62443. In diesem Security Standard werden die Prozessanforderungen gemäß eines Secure Development Life-Cycle (SDL) definiert. Dieser beinhaltet Security-Anforderungen (Authentifikation, Autorisierung, Verschlüsselung, Schutz durch Firewall), Secure Design, Secure Implementation (Methoden, Codierrichtlinien), Verifikation und Validierung, Defect Management, Patch Management und End-of-Life Management.

Das Modul IEC 62443-3-3 System Security Requirements and Security Levels [15] definiert die Security Level SL 1 bis SL 4, wobei SL 4 den höchsten Schutzbedarf erfordert. Die SL Einstufung erfolgt gemäß Risikobewertung. Hierfür ist der Angriffstyp (Skills, Ressourcen, Motivation) maßgeblich, da die Risikowahrscheinlichkeit eines Angriffs nicht quantifizierbar ist.

Das Schutzniveau SL 1 bietet Schutz gegen unbeabsichtigten oder zufälligen Missbrauch. Dieses Schutzniveau ist in der Regel durch die Safety-Anforderungen des kritischen Systems erreicht:

- Die Gefährdung durch zufällige Fehler und unbeabsichtigte Störungen muss minimiert werden.
- Nur autorisiertes und geschultes Personal darf kritische Funktionen ausführen. Dies setzt besondere betriebliche Vorkehrungen voraus.

Dort wo der unberechtigte Zugriff nicht ausgeschlossen werden kann, sind weitere Security-Maßnahmen erforderlich.

Die Security-Anforderungen für COTS- und Open Source-Software werden im Modul IEC 62443-4-1 Product Development Requirements [16] beschrieben:

- **SM-8** IT Sicherheit eingebetteter Fremdkomponenten,
- **SM-9** Fremdkomponenten für besondere Anwendungen

Die Anforderung SM-8 gilt für COTS- und Open Source-Komponenten. Demgegenüber bezieht sich SM-9 auf die Verwendung von Custom Software.

Die Anforderung SM-8 umfasst die Evaluierung der Softwarekomponente hinsichtlich:

- Abdeckung der Security-Anforderungen und Einhaltung eines gestaffelten Sicherheitskonzeptes (Schichtenmodel),
- Gewährleistung einer sicheren Implementierung bezüglich Security Coding Rules durch statische Analyse und Codereview,
- Umfang der durchgeführten Security-Testaktivitäten,
- Gewährleistung und Effektivität von Defect- und Patchmanagement,
- Umfang und Qualität der Security-Dokumentation (Konfiguration, Härtung, Sicherheitsrichtlinien).

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de

Je nach Ergebnis der Evaluierung der Softwarekomponente sind ergänzende Maßnahmen erforderlich, z.B. zusätzliche Tests, Dokumentation oder Code Inspection. Da für COTS-Komponenten der Source Code nicht vorhanden ist, sind abweichende kompensatorische Maßnahmen notwendig, vergleiche Abschnitt 4.1 Safety-Aspekte.

05 Zusammenfassung

Safety- und Security-Prozesse betrachten die Qualität der Software aus verschiedenen Blickwinkeln. Obwohl ähnliche Anforderungen gestellt werden (sichere Implementierung, Test, Dokumentation) ergeben sich Konflikte, z.B.

- Langlebigkeit und Stabilität der Software aus Safety-Sicht,
- schnelle Softwareupdates als Reaktion auf Angriffe aus Security-Sicht.

In diesem Spannungsfeld effizient Software zu entwickeln, erfordert ein gut strukturiertes Management sowie motivierte Entwickler und Tester, die sowohl die Safety- als auch die Security-Aspekte im Focus behalten.

Autoren
Nils Dittmar (Dipl.-Ing.)
Software Validierung von Safety-Critical Systems /
Software Validation of Safety-Critical Systems
certitudo GmbH
Anschrift: Große Kreuzstr. 13, 23909 Ratzeburg
E-Mail: nils.dittmar@certitudo-gmbh.de

Thorsten Lörch
Geschäftsführer / Managing Director
certitudo GmbH
Anschrift: Große Kreuzstr. 13, 23909 Ratzeburg
E-Mail: thorsten.loerch@certitudo-gmbh.de

Literaturverzeichnis

[1] IT-Sicherheitsgesetz (Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme), 17.07.2015
[2] https://www.heise.de/tp/features/Hackerangriff-auf-kalifornisches-Krankenhaus-3378410.html, 28.08.2017 um 09:59
[3] http://www.nctechnology.org/events/overview/outlook_for_it/outlook-forrester-slides.pdf, 06.09.2017 um 11:55
[4] Black Duck Report: 2017 Open Source Security & Risk Analysis, 2017
[5] c't Magazin: Kontrollverlust am Steuer, 2015 Heft 24
[6] https://www.shodan.io, 29.08.2017 um 10:15
[7] BSI: Quellcode-basierte Untersuchung von kryptographisch relevanten Aspekten der OpenSSL-Bibliothek, 2015
[8] AV-TEST: Security Report 2015/16
[9] https://www.us-cert.gov/bsi/articles/best-practices/legacy-systems/security-considerations-in-managing-cots-software, 16.11.2017 um 16:30
[10] BSI: Top 10 Bedrohungen und Gegenmaßnahmen 2016
[11] BSI: ICS-Security-Kompendium (Betreiber), 2013
[12] BSI: ICS-Security-Kompendium (Hersteller), 2014
[13] DIN VDE V 0831-104: Elektrische Bahn-Signalanlagen – Teil 104: Leitfaden für die IT-Sicherheit auf Grundlage IEC 62443
[14] https://www.cvedetails.com/, 06.09.2017 um 15:47
[15] IEC 62443-3-3 System Security Requirements and Security Levels, Edition 1.0, 2013
[16] IEC 62443-4-1 Secure Product Development Life-Cycle Requirements, Draft 3, Edition 11, 2016

certitudo GmbH | Gr. Kreuzstr. 13 | 23909 Ratzeburg | Tel: +49-4541-859804 | Mail: vertrieb@certitudo-gmbh.de
www.certitudo-gmbh.de