»Der Angriff weiß, was er will. Die Verteidigung befindet sich in dem Zustand der Ungewissheit.«

Helmuth Graf von Moltke

Zur Sicherheit softwarebasierter Produkte

Status Quo, Ausblick und FAQ zu Entwicklung und Betrieb softwarebasierter Produkte

Version 2.0

www.bitkom.org

bitkom

Herausgeber

Bitkom e. V.
Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V.
Albrechtstraße 10 | 10117 Berlin

Ansprechpartner

Dr. Frank Termer | Bereichsleiter Software
T 030 27576-232 | f.termer@bitkom.org

Verantwortliches Bitkom-Gremium

AK Quality Management

Projektleitung

Dr. Frank Termer | Bitkom e. V.

Inhaltsverzeichnis

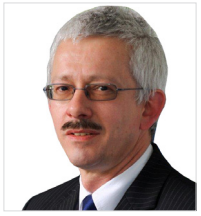
Abbildungsverzeichnis

Über die Autor:innen

Stefan Hessel ist Rechtsanwalt und Associate im Team Cybersecurity & Datenschutz bei ↗ reuschlaw Legal Consultants in Saarbrücken. Zuvor studierte er Rechtswissenschaften mit dem Schwerpunkt IT-Recht und Rechtsinformatik an der Universität des Saarlandes. Im Rahmen seines Referendariats am OLG Saarbrücken absolvierte er unter anderem Stationen beim Unabhängigen Datenschutzzentrum Saarland und beim Sonderdezernat Cybercrime der Staatsanwaltschaft Saarbrücken. Während Studium und Referendariat war er Mitarbeiter am Lehrstuhl für Rechtsinformatik der Universität des Saarlandes und am CISPA – Helmholtz-Zentrum für Informationssicherheit in Saarbrücken. Im Rahmen seiner dortigen Forschungstätigkeit stieß er 2017 das Verbot der Spielzeugpuppe »My friend Cayla« als illegales Spionagegerät durch die Bundesnetzagentur an.

Thomas Kriesel verantwortet den Bereich Steuern und Unternehmensrecht beim Bitkom mit besonderen Schwerpunkten im Vertragsrecht und in der Unternehmenscompliance. In dieser Funktion betreut er die Arbeitskreise Steuern, Vertrags- und Rechtsgestaltung und Recht im Unternehmen und Compliance sowie das Forum Recht. Außerdem ist er Ansprechpartner für bereichsübergreifende Rechtsthemen im Bitkom. Bevor Thomas 2002 beim Bitkom tätig wurde, war er bei einer Bank und einer Wirtschaftsprüfung beschäftigt.

Jacob Loring ist für die ↗ Code Intelligence GmbH im Business Development tätig. Bei dem Unternehmen handelt es sich um eine Ausgründung der Universität Bonn, das auf die Automatisierung von Software Security Tests spezialisiert ist, mit Hilfe von Feedback-based-Fuzzing. Im Rahmen seiner Tätigkeit vertritt er das Unternehmen in Gremien- und Verbandsangelegenheiten.



Stefan Luckhaus ist Informatiker und seit 1981 in der Softwareentwicklung tätig. Bei der ↗ PASS Consulting Group ist er verantwortlich für das Qualitätsmanagement und für Fragen der Compliance in der Softwareentwicklung wie auch beim Betrieb von Kundensystemen in eigenen Rechenzentren der PASS-Gruppe. Er ist Buchautor und publiziert in verschiedenen Blogs. Im BITKOM steht er dem Arbeitskreis Qualitätsmanagement vor.

Dipl.-Inform. Ramon Mörl ist seit Beendigung seines Informatik-Studiums an der TU München 1987 als Berater in Fragen der IT-Sicherheit tätig. Für Firmen wie HP, IBM, Siemens, ICL und Bull, hat er leitende Tätigkeiten in Projekten in Belgien, Deutschland, Frankreich, Italien, Österreich, Schweiz und USA übernommen. Als unabhängiger Evaluator und Berater der Europäischen Union war er vor allem im Bereich der ECMA und ISO-Standards für die IT-Sicherheit tätig. Seit 2002 bringt Herr Mörl die Erfahrung eines kosteneffizienten Einsatzes sicherer IT-Systeme aus internationalen Großprojekten in die ↗ itWatch GmbH als Geschäftsführer ein.

Profn. Dr.-Ing. Sabine Radomski studierte von 1984–1988 Informationsverarbeitung an der Ingenieurhochschule Dresden und promovierte 1995 an der TU Dresden im Informatik Zentrum. Sie war als Systemadministrator in verschiedenen Firmen, u. a. dem Regierungspräsidium Dresden tätig. Seit 2000 ist sie Professorin im Fachbereich Nachrichtentechnik der ↗ Hochschule für Telekommunikation Leipzig. Ihr Lehrangebot umfasst die Veranstaltungen in Bachelor und Masterstudiengängen: Verteilte Systeme, Software Engineering, Software Management sowie Netzwerkmanagement. Profn. Radomski besitzt eine Reihe von internationalen Kontakten im europäischen Raum. Sie ist Bitkom-Expertin in verschiedenen Arbeitskreisen und wurde als MINT Botschafterin 2012 und als Professor des Jahres 2015 von der Zeitschrift Unicum ausgezeichnet. Ihre Forschungsthemen sind Software Qualität, IT Sicherheit und Cloud Computing.

Dr. Roland Spengler studierte und promovierte in Physik an der Friedrich-Alexander-Universität Erlangen-Nürnberg. Er ist seit 2016 für die ↗ Atos Information Technology GmbH als Account & Industry Lead im Bereich Public Services tätig und Senior Experte in der Atos Expert Community. Seine Erfahrungen umfassen Audits, Analyse von Geschäftsprozessen und Durchführung von Prozessoptimierungen, ebenso wie Konzeption und Auswahl von IT-Lösungen, Entwicklung von IT-Strategien und Applikationsarchitekturen und Data Center Consolidations. Seit 2011 besitzt er das CISA-Zertifikat und hält immer wieder mal Fachvorträge zum Thema organisatorische Sicherheit.

Ines Theilen studierte Wirtschaftsingenieurswesen mit dem Schwerpunkt Softwareengineering und war nach ihrem Abschluss 1996 viele Jahre als IT-Consultant im Bereich Public und Defense tätig. Seit 2018 ist sie Leiterin des Qualitätsmanagements bei der VRG IT (↗ VRG-Gruppe). Sie führte dort das Qualitätsmanagement ein, vereinheitlichte die Prozesse und führte Anforderungsmanagement und Testdokumentation ein. Im Jahr 2020 übernahm sie außerdem die Bereiche Projektmanagement und Customer Support. Frau Theilen ist Mitglied im Arbeitskreis Qualitätsmanagement des Bitkom.

Dr. Frank Termer ist Bereichsleiter Software beim Bitkom e. V. Nach seinem Studium der Wirtschaftsinformatik an der Otto-von-Guericke Universität Magdeburg war er ab 2006 als IT-Consultant tätig und führte Projekte im Geschäftsprozess- und IT-Servicemanagement durch. Hierbei betreute er vor allem Unternehmen aus den Bereichen Energiewirtschaft, Finanzdienstleistungen sowie der öffentlichen Verwaltung. Ab 2010 arbeitete er als wissenschaftlicher Mitarbeiter an der Technischen Universität Ilmenau im Fachgebiet Wirtschaftsinformatik für Dienstleistungen. Dort legte er seinen Forschungsschwerpunkt auf Fragen des strategischen IT-Managements. Während dieser Tätigkeit entstanden zahlreiche Publikationen zu den Themen IT-Agilität, Enterprise Architecture Management, Geschäftsprozessmanagement und Business-IT-Alignment. Er promovierte zum Thema »Determinanten der IT-Agilität«. Seit 2015 betreut er im Bitkom e. V. die Gremien des Kompetenzbereichs Software. Er konzipiert, organisiert und moderiert Gremienveranstaltungen, ist verantwortlich für die thematische Weiterentwicklung dieser Gremien sowie deren inhaltliche Positionierung innerhalb und außerhalb des Bitkom.

Einführung und Motivation

Mit zunehmender Digitalisierung im Berufs- und Privatleben und der damit entstehenden Omnipräsenz softwarebasierter Produkte rückt das Thema Sicherheit in das allgemeine Bewusstsein. Ob der heimische Fernseher, das genutzte Smartphone oder das moderne Auto: Software ist allgegenwärtig! Sie ist ein zentrales Element der Digitalen Transformation und ermöglicht in vielen Fällen erst die Digitalisierung ganzer Branchen. Um das volle Potenzial digitaler Technologien zu nutzen, gilt es Fehler der Vergangenheit nicht zu wiederholen. Digitalisierung bedeutet nicht nur eine weitere Technologie, es sind vollkommen neue Möglichkeiten! So fordert **↗ der Lenkungsausschuss Software im Bitkom eine neue Haltung zur Digitalisierung zu entwickeln**, bei der insbesondere in das Verständnis des Digitalen investiert wird.

Durch den steigenden Softwaredurchdringungsgrad in Produkten ergeben sich allerdings nicht nur Möglichkeiten und Chancen, sondern auch Gefahren und Bedrohungsszenarien, die auf die entsprechende Software abzielen. Damit erhält Software eine zunehmende Bedeutung bei der Betrachtung und Bewertung der Sicherheit von Produkten insgesamt, und eine erhöhte Wahrnehmung bei den Anwendern softwarebasierter Produkte. Die Akzeptanz von Produkten und Dienstleistungen, die durch einen nicht unerheblichen Teil durch Software definiert werden, hängt in besonderem Maße von deren Sicherheit ab. Damit wird Sicherheit zu einem zunehmenden Erfolgsfaktor für Unternehmen, um langfristig am Markt agieren zu können.

Software wird durch eine Reihe von besonderen Eigenschaften charakterisiert: Sie ist **immateriell**: dadurch verschleißt sie (theoretisch) nicht durch Benutzung, sondern durch äußere Einflüsse (Veränderung der Anforderungen, der Betriebssysteme, Hardware, etc.). Software ist **inhomogen**: sowohl bezüglich der Komplexität, der Zusammensetzung (unterschiedliche Komponenten werden kombiniert), der Anforderungen, als auch der Fehlerverteilung. Deshalb kann die Qualität von Software nicht einfach gemessen werden. Um die Qualität (Erfüllungsgrad der gestellten Anforderungen) von Software zu messen, werden viele verschiedene Verfahren, Metriken und Testmethoden angewendet und entwickelt. Es existieren viele verschiedene Werkzeuge, um Softwarequalität zu messen, und in verschiedenen Schwachstellendatenbanken (u. a. OWASP, CWE, NVD, CAPEC, CVE, VDBs) werden Sicherheitslücken bekanntgegeben. Je nach Aufgabenstellung, Einsatzort und Nutzer unterscheiden sich die Anforderungen an die Software.

Bitkom, und die im Bitkom organisierten softwarebezogenen Gremien, erhielten bereits in der Vergangenheit zahlreiche Fragen, warum das Thema Sicherheit von softwarebasierten Produkten scheinbar nicht in den Griff zu bekommen ist. Dies führte zur Erstellung der ersten Auflage des vorliegenden Dokuments im Jahr 2016. Mittlerweile sind einige Fragen hinzugekommen und der Stand der Technik[1] hat sich weiterentwickelt. In dieser zweiten Auflage des Leitfadens sollen daher neben den bisherigen Fragen auch neue Fragen aufgegriffen und möglichst allgemein verständlich erläutert werden. Die Fragen haben dabei keinerlei Anspruch auf Vollständigkeit, werden aber so oder in leicht modifizierter Form immer wieder gestellt. Gleichzeitig ist es dem Bitkom ein wichtiges Anliegen, in diesem Dokument darzustellen,

1 Zum Stand der Technik zählen wir nicht nur aktuelle technische und technologische Möglichkeiten, sondern auch Organisation und Prozesse in (softwareentwickelnden) Unternehmen sowie die regulatorischen Rahmenbedingungen.

auf welche vielschichtige Art und Weise die Anbieter softwarebasierter Produkte und Dienstleistungen sich um die Sicherheit ihrer Produkte kümmern, und welchen Aufwand sie betreiben, um den diesbezüglich berechtigten Erwartungen der Anwender gerecht zu werden. Ebenso werden aber auch die Grenzen des derzeit Machbaren dargestellt und an einigen Stellen erläutert, warum manches von Anbietern softwarebasierter Produkte eben nicht gemacht wird.

Im vorliegenden Dokument werden nun aufgeworfene Fragen einzeln adressiert und in jeweils einem separaten Kapitel betrachtet. Dabei werden auch unterschiedliche Sichtweisen und Aspekte berücksichtigt, die bei der Produktauswahl für den Anwender von Bedeutung sein könnten (z. B. Preis, die Zeit, Leistungsfähigkeit oder die Erwartungshaltung des Konsumenten). Am Ende des Dokuments fasst zudem eine FAQ-Sammlung diese Fragen und kurze Antworten dazu zusammen.

Wir danken an dieser Stelle den Autor:innen der ersten Auflage des Dokuments aus dem Jahr 2016: Susanne Dehmel, Marc Fliehe, Stefan Luckhaus, Dr. Frank Simon und Dr. Frank Termer. Durch diese erste Version des Leitfadens konnten wir schnell und zügig in eine Überarbeitung starten und hatten gleichzeitig eine gute Orientierung bzgl. Inhalten und Format.

Wir hoffen, mit diesem Dokument dem Thema Sicherheit von softwarebasierten Produkten allgemeinverständlich und mit hohem fachlichem Anspruch gerecht zu werden. Es ist explizites Ziel, dieses Dokument laufend fortzuschreiben, zu aktualisieren und um weitere Fragen zu ergänzen. Wir laden daher herzlich ein, sich an der Weiterentwicklung des Dokuments zu beteiligen. Ihr Feedback nehmen wir gern entgegen! Für Fragen, aber auch für kritische Anmerkungen zu unseren Antworten, stehen wir ebenfalls gerne jederzeit zur Verfügung.

- Stefan Hessel, Reusch Rechtsanwaltsgesellschaft mbH
- Thomas Kriesel, Bitkom e. V.
- Jacob Loring, Code Intelligence GmbH
- Stefan Luckhaus, PASS Consulting Group
- Ramon Mörl, itWatch GmbH
- Profn. Dr.-Ing. Sabine Radomski, Deutsche Telekom AG, Hochschule für Telekommunikation Leipzig
- Dr. Roland Spengler, Atos Information Technology GmbH
- Ines Theilen, VRG IT GmbH
- Dr. Frank Termer, Bitkom e. V.

1 Der Prozess der Softwareentwicklung

Im Kino und in der Literatur entsteht gelegentlich der Eindruck, dass Software nur von genialen Hackern hergestellt wird, die alleine vor ihren Rechnern sitzen. Mit der Realität hat das jedoch wenig zu tun. Diese Vorstellung ist sogar irreführend, da sie die Fähigkeiten einzelner Softwareentwickler überschätzt und die Relevanz von Verfahren und Prozessen für die Entwicklung und die Qualitätssicherung der Software ignoriert. Daher stellt sich zunächst die Frage, **wie wird Software eigentlich hergestellt?**

An der Entwicklung von Software sind in der Regel viele Personen mit unterschiedlichen Fähigkeiten und aus verschiedensten Fachbereichen beteiligt. Zunächst müssen Anforderungen eingeholt werden. Was soll die Software können? Welche Gesetze gelten, zum Beispiel für Datenschutz und IT-Sicherheit? Wie bei einem Haus muss die Architektur der Software geplant werden. Danach können Aufgaben an einzelne Mitarbeitende oder Teams weitergeleitet werden.

In der Softwareentwicklung gibt es Spezialisten für verschiedene Aufgaben. Einige Entwickler sind zum Beispiel besonders gut in der Programmierung von Datenbanken. Andere sind hingegen Spezialisten für die Entwicklung von Benutzeroberflächen. Immer wieder werden Qualitätskontrollen durchgeführt. Funktioniert der Code? Passen die verschiedenen Bausteine der Software zusammen? Bei großen Konzernen arbeiten zum Teil hunderte Entwickler an einem Produkt. Die Arbeitsabläufe müssen dabei natürlich genau aufeinander abgestimmt sein. Dafür braucht es viel Planung und klare Prozesse.

Nur in wenigen Fällen müssen Entwickler einen Code komplett neu schreiben. Oft können Sie auf Code-Bibliotheken zurückgreifen, oder sich an Vorlagen aus Open-Source-Software bedienen. Software enthält daher oft Komponenten und Teile aus anderer Software und wird für den speziellen Gebrauch angepasst und weiterentwickelt. Hier übernimmt der Software-Hersteller automatisch auch die Haftung für die Open-Source-Komponenten und muss für deren Qualität sorgen. Das bedeutet, dass jeder Hersteller für die Open-Source-Komponenten in seinen Produkten ähnliche Qualitätsprüfungen sicherstellen muss, wie für die selbst entwickelte Software. Vor dem Release wird Software idealerweise einem Stresstest unterzogen und mit Kunden unter realen Bedingungen getestet. Auch nach der Veröffentlichung muss die Software regelmäßig überarbeitet werden. Zum Beispiel, um neue Sicherheitslücken zu schließen, oder die Software auf spezifische Wünsche der Nutzer anzupassen. Der Lebenszyklus einer Software endet meist, wenn der Support für das jeweilige Produkt eingestellt wird.

Wie die Software im Detail entwickelt wird, kann sich jedoch nach verschiedenen Rahmenbedingungen unterscheiden. Zum Beispiel wird der Entwicklungsprozess beeinflusst von der Größe und der Dynamik des Entwicklerteams, von seiner Zusammensetzung (Soft Skills, Spezialisierungen) oder auch der Ausbildung der Entwickler. Die vorhandene Infrastruktur, die Hardware und die Entwicklungsumgebungen, die Testwerkzeuge, oder auch die Dynamik des Projekts wirken sich ebenfalls auf den Entwicklungsprozess aus. Je nachdem wie viel Zeit für die Entwicklung bereitsteht (time to market), welche Ressourcen/Budgets vorhanden sind, welche Aufgaben die Software erfüllen soll (Einsatzgebiet) oder auch je nachdem, welche rechtlichen Rahmenbedingungen gelten, kann der Prozess variieren. Die Rahmenbedingungen wirken sich auch auf die Qualität der Software aus. Je mehr Ressourcen in die Qualitätssicherung investiert werden, desto besser ist dies für die Qualität der Software.

2 Die Notwendigkeit von Patches und Updates

Softwareprodukte werden entwickelt, damit sie eine bestimmte Aufgabe über einen längeren Zeitraum hinweg erfüllen. **Doch warum werden regelmäßig Patches und Updates für Softwareprodukte ausgeliefert?** Je länger der Zeitraum der Softwarenutzung ist, desto wahrscheinlicher sind Veränderungen, die Auswirkungen auf das Softwareprodukt haben:

- Die Aufgabe, die das Produkt erfüllen soll, verändert sich. Funktionen der Software oder Interaktionen beispielsweise mit anderen Diensten, Systemen oder Akteuren müssen angepasst werden, fallen weg oder es werden neue benötigt.

- Die Rahmenbedingungen, unter denen das Produkt seine Aufgabe erfüllen muss, verändern sich. Ein Beispiel dafür sind geänderte gesetzliche Anforderungen.

- Unzulänglichkeiten des Produkts gewinnen an Bedeutung. Die Ursache können bereits bekannte und bisher als vernachlässigbar angesehene Unzulänglichkeiten sein oder auch neu festgestellte Fehler, durch die sich die seitens der Nutzer wahrgenommene Qualität des Produkts vermindert. Möglicherweise sind auch schon länger bestehende Sicherheitslücken erst spät von potentiellen Angreifern erkannt worden.

Veränderungen dieser Art erfordern eine Weiterentwicklung der Softwareprodukte. Findet keine Weiterentwicklung mehr statt, sinkt nach einer gewissen Zeit der Nutzen, das Produkt kann verbindliche Anforderungen nicht mehr erfüllen oder es besteht Bedrohungspotential für Angriffe. Jedes Softwareprodukt muss beim Hersteller oder, im Fall von Open Source Software, seitens einer Community über funktionierende Verfahren zum Fehler- und Änderungsmanagement verfügen. Ein Prozess zum Release-Management muss zuverlässig sicherstellen, dass in angemessener Zeit neue Versionen des Produkts erstellt und seinen Nutzern zur Verfügung gestellt werden. Angemessen ist es, wenn beispielsweise Patches zum Schließen neuer Sicherheitsschwachstellen sofort zur Verfügung stehen, weniger kritische funktionale Erweiterungen dagegen erst im nächsten jährlich ausgelieferten Major Release. Patches und Updates über den gesamten Produkt-Lebenszyklus hinweg sind ebenso wichtig wie die anfängliche Qualitätssicherung nach der Neuentwicklung.

3 Zur Bedeutung regelmäßiger Aktualisierungen

»Ein neues Software Update ist verfügbar. Bitte aktualisieren Sie die Anwendung und starten Sie das Gerät neu.« Ähnliche Aufforderungen erhalten Nutzer regelmäßig. Doch was zunächst lästig erscheint, geschieht immer im Interesse der Nutzer. **Doch warum muss Software so häufig aktualisiert werden?**

Software wird permanent weiterentwickelt und verbessert. Sie muss regelmäßig angepasst werden, um den Nutzern ein attraktives Produkt anzubieten. Daher braucht es häufige Updates. In der Regel handelt es sich dabei um funktionale Updates. Diese werden zum Beispiel notwendig, wenn Service-Wartungen durchgeführt werden, oder wenn die Hersteller neue Funktionen der Software veröffentlichen. Gelegentlich werden auch Sicherheits-Updates durchgeführt, um auf neue Sicherheitslücken in der Software zu reagieren. Da Software heute in immer kürzeren Intervallen entwickelt und verändert wird, kann der Eindruck entstehen, dass die Häufigkeit von Updates zunimmt. Das liegt meist daran, dass Anpassungen in der Software den Anwendern fast immer augenblicklich bereitgestellt werden. Dies ist jedoch auch im Interesse der Nutzer.

Funktionale Updates sind wichtig, da Software zunehmend in verschiedenen Kontexten eingesetzt wird. Diese Bereiche sind im Vorfeld des Einsatzes nicht vollständig bekannt und Software wird zudem auch in Bereichen eingesetzt, für die diese ursprünglich nicht konzipiert war. Ein Beispiel für diese Zweckentfremdung könnte sein, dass Nutzer versuchen eine Anwendung auf dem Display ihres vernetzten Autos zu starten und Ihnen die Anwendungsoberfläche verzerrt dargestellt wird. Funktionale Updates sind notwendig, um auf Kundenwünsche und veränderte Anforderungen im Alltag der Nutzer einzugehen. Das bedeutet, funktionale Updates erhöhen somit für die Nutzer auch den Mehrwert der Software, da sie Ihnen neue Funktionen bereitstellen, die ihnen die Nutzung der Software erleichtern.

Während die Erfüllung funktionaler Anforderungen an eine Software mit Hilfe konstruktiver und analytischer Qualitätssicherung nahezu vollständig verifiziert werden kann, ist die Überprüfung des Qualitätsmerkmals Sicherheit nur bezogen auf einen bestimmten Zeitpunkt und einen bestimmten Anwendungskontext möglich. Angriffspunkte und Angriffsmethoden bei Software entwickeln sich permanent weiter. Somit steigen auch die Anzahl und der Umfang möglicher Bedrohungen weiter an. Hinsichtlich des Merkmals der Sicherheit kann Software daher nicht »reifen«, im Sinne irgendwann das höchste Maß an Sicherheit erreicht zu haben. Sie kann nur versuchen, Schritt zu halten mit der Weiterentwicklung seitens der Angreifer und proaktive Maßnahmen zu ergreifen, bspw. durch eine Risikoreduktion durch Security-by-Design, um zukünftige Bedrohungsszenarien zu adressieren.

Nutzer sollten verfügbare Updates schnellstmöglich installieren. Dies trägt deutlich zur Sicherheit der Software bei und macht neue Funktionen für die Nutzer schneller verfügbar.

4 Wenn Software Fehler aufweist

Wenn also ein hoher Aufwand betrieben wird, um grundsätzlich sichere Software herzustellen und durch Updates und Patches Anpassungen an veränderte Gegebenheiten vorgenommen werden, stellt sich die Frage, **warum Software denn trotzdem nie fehlerfrei ist?** Grundsätzlich sind bei der Betrachtung von Softwarefehlern folgende Fälle zu unterscheiden:

- Softwarefehler Typ 1: Die Software weicht von einer spezifizierten Eigenschaft ab, liefert beispielsweise ein anderes Ergebnis oder verhält sich anders als explizit beschrieben. Mit anderen Worten liegt die Abweichung gegenüber einer **expliziten Anforderung** vor.

- Softwarefehler Typ 2: Ein Ergebnis oder das Verhalten der Software entspricht nicht den Erwartungen eines Nutzers. In diesem Fall liegt die Abweichung gegenüber einer **impliziten Anforderung** vor.

Fehler des ersten Typs lassen sich mit Hilfe einer geeigneten Entwicklungsmethodik deutlich reduzieren. Dabei werden zunächst detaillierte Konzepte erstellt und noch vor ihrer Implementierung gründlich überprüft. Die Konzepte dienen den Entwicklern als zuverlässige Vorlage und ermöglichen ihnen erste Tests. Nachgelagert der Entwicklung kann die Software vor ihrer Nutzung daraufhin getestet werden, dass Verhalten und Ergebnisse mit den Konzepten übereinstimmen. Die Anzahl von Fehlern des ersten Typs (Abweichungen von expliziten Anforderungen) kann deutlich reduziert werden, was jedoch zusätzlichen Aufwand und Zeit erfordert.

Während Abweichungen von expliziten, dokumentierten Anforderungen zweifelsfrei als Fehler festgestellt werden können ist dies schwieriger, wenn Erwartungen der Nutzer als Maßstab dienen. Dabei gibt es viele Anforderungen an eine Software, bei denen sich alle Nutzer einig sind: falsche Eingaben sollen nicht zu einem Absturz führen, Berechnungen sollen kein falsches Ergebnis ergeben, personenbezogene Daten sollen nur den im Rahmen der vorgesehenen Anwendungsfälle autorisierten Nutzern zugänglich sein usw. Demgegenüber gibt es jedoch auch Erwartungen, die von Nutzer zu Nutzer unterschiedlich sein können: die Zeit bis zur wahrgenommenen Reaktion auf eigene Aktionen, die Anordnung von Eingabe-Elementen auf einer Maske, das Verhalten und mögliche Hilfestellungen bei falschen Eingaben, die Dauer bis zu einem erzwungenen Passwortwechsel usw. Fehler dieses Typs sind nutzerspezifisch, d. h. was für einen Nutzer ein schwerer Fehler ist, sieht ein anderer Nutzer als noch akzeptabel an. Außerdem kann sich die Wahrnehmung mit der Zeit ändern, d. h. was gestern noch akzeptabel erschien, wird vielleicht heute als kritischer Fehler empfunden.

Verhindern lassen sich Fehler des zweiten Typs dadurch, dass alle Eigenschaften einer Software und ihres Verhaltens möglichst genau spezifiziert werden, dass exakt nach der Spezifikation entwickelt, die Einhaltung der Spezifikation durch Tests überprüft wird und die (frühzeitig einbezogenen) Nutzer nicht mehr erwarten dürfen als in der Spezifikation des Produkts beschrieben ist. Die Anzahl von Fehlern könnte mit Methoden der analytischen Qualitätssicherung (z. B. Tests), einem entsprechenden Budget und vor Allem Zeit-Kontingent signifikant gemindert werden. Oft stellt sich dabei Zeit als größter Engpass dar.

Softwareentwicklung ist ein vergleichsweise junger Industriezweig. Qualitativ hochwertige, fehlerarme Software erfordert (noch) einen hohen Entwicklungsaufwand. Ein noch größeres Problem ist dabei jedoch die Zeit, die zur Entwicklung benötigt wird. In vielen Fällen wird Software entwickelt, weil man damit eine innovative Idee umsetzen und schneller als eventuelle Mitbewerber an den Markt bringen möchte. Manchmal geht es auch darum, auf Veränderungen des Marktes schneller zu reagieren als die Konkurrenz. Eventuell müssen auch gesetzliche Änderungen fristgemäß umgesetzt werden. Alle diese Fälle haben eines gemeinsam: Zeitdruck. Die Hersteller von Softwareprodukten werden also meist abwägen, ob sie für einen kurzfristigen Wettbewerbsvorteil und die schnelle Verfügbarkeit einer Software, mögliche Fehler und Sicherheitslücken hinnehmen, oder ob sie in die Qualität ihrer Softwareprodukte investieren – mit anderen Worten, in welchem Maße überhaupt Konzepte erstellt und Tests durchgeführt werden sollen.

Selbstverständlich ist es eine vereinfachte Darstellung, dass bei der Softwareentwicklung entweder Fehlerfreiheit oder eine schnelle Markteinführung angestrebt werden kann. Es gibt vielversprechende Entwicklungen in Richtung neuer Programmiersprachen und -paradigmen, Standardisierung, Automatisierung und selbstverständlich auch einer optimalen Einbindung des Menschen in den Entwicklungsprozess, die sowohl die Produktivität des Entwicklungsprozesses als auch die Qualität der Softwareprodukte stetig verbessern werden.

5 Wie »die Sicherheit« in die Software kommt

Software weist, wie andere Produkte auch, bestimmte Qualitätseigenschaften auf. Art und Umfang der Qualität bestimmt sich dabei hauptsächlich in der Ausgestaltung der Herstellungsprozesse. In der Literatur und in Industriestandards werden unterschiedliche Qualitätsmerkmale für Software herangeführt, z. B. Zuverlässigkeit, Sicherheit, Benutzerfreundlichkeit, Robustheit, Effizienz und verschiedene mehr. Dabei kommt es zu einem klassischen Zielkonflikt. Es ist eine Herausforderung alle diese Qualitätsmerkmale zu berücksichtigen und Software gleichzeitig schnell und kostengünstig zu entwickeln. Daraus ergibt sich zwangsläufig die Frage, **was machen Hersteller um die Qualität ihrer Software sicher zu stellen?**

Als Schlussfolgerung aus dem geschilderten Zielkonflikt und den gewünschten Merkmalen der Softwarequalität entwickeln sich unterschiedliche Prinzipien, Methoden und Werkzeuge. Im Bereich der Softwaresicherheit lassen sich signifikante Weiterentwicklungen in den letzten Jahren der Digitalisierung beobachten, die direkt auf eine bessere Softwarequalität einzahlen.

Das Prinzip »Security by Design« oder auch »Security by Default« bedeutet, dass Sicherheitsanforderungen an Software schon während der Entwicklungsphase eines Produktes berücksichtigt werden, um spätere Sicherheitslücken zu verhindern. Denn mit dem laufenden Projektfortschritt steigen auch die Kosten für die Beseitigung von Sicherheitslücken.

Neue (vor allem agile) Methoden lösen dabei zunehmend klassische Entwicklungsmethoden ab. Dies führt u. a. zu prozessualen Anpassungen der Softwareentwicklung, die unter den Begrifflichkeiten »DevOps« und »Shift-Left« bekannt sind. Der »DevOps«-Ansatz hat zum Ziel, in den Bereichen Development (Entwicklung), IT Operations (IT-Betrieb) und Qualitätssicherung für eine effizientere und effektivere Zusammenarbeit zu sorgen. Durch die optimierte Zusammenarbeit der verschiedenen Teilbereiche wird die Geschwindigkeit des gesamten, bereichsübergreifenden Entwicklungsprozesses bis zur Produktivsetzung, und die Kooperation zwischen den einzelnen Teams verbessert und somit eine optimierte Balance zwischen Qualität und Effizienz gefunden. Mit »Shift-Left« geht es darum, im Entwicklungszyklus möglichst frühzeitig mit dem Testen zu beginnen bzw. kontinuierliches und automatisiertes Testen über den gesamten Lebenszyklus der Softwareentwicklung für eine verbesserte Qualität sicherzustellen. Um die qualitätsorientierten Prinzipien und Methoden umzusetzen und effizient zu gestalten, braucht es die richtigen Werkzeuge. Moderne Werkzeuge testen die entwickelte Software hochgradig automatisiert und erreichen damit verkürzte Testzeiten sowie gute Testabdeckungen, insbesondere in den Bereichen Funktion, Leistung und Sicherheit.

Abschließend ein Beispiel zur Verdeutlichung: Klassische Ansätze des Software Testings brauchen viel Zeit, denn die Entwicklung findet in verschiedenen Phasen statt. Die traditionelle Qualitätssicherung funktioniert nach dem Prinzip: »Erst wird entwickelt. Dann wird getestet«. Das manuelle Penetrations-Testing (Pentesting) für eine Software-Applikation dauert in der Regel mehrere Tage. Wenn Fehler gefunden werden, müssen Entwickler den Code überarbeiten. Der ganze Prozess beginnt wieder von vorne. In modernen DevOps-Ansätzen und Agilen Methoden gibt

es hingegen eine permanente Qualitätssicherung. Jedes Mal, wenn Entwickler etwas am Code verändern, findet sofort eine Qualitätskontrolle statt. Dies beschleunigt den Entwicklungsprozess und kann hochgradig automatisiert werden. Verschiedene Arbeitsschritte werden, wie auf einem Förderband, hintereinander angeordnet. Man spricht von sogenannten Continuous Integration / Continuous Deployment (CI/CD)-Pipelines. In diese Arbeitsschritte kann zum Beispiel feedback-basiertes Fuzzing (FAST) integriert werden, in Kombination mit statischen Codeanalysen (SAST). Dabei wird automatisiert geprüft, ob der Code unerwünschte Unregelmäßigkeiten aufweist und wie der Code unter realen Bedingungen auf simulierte Test-Eingaben reagiert. Gefundene Fehler werden dokumentiert und können von den Entwicklern sofort bearbeitet werden.

6 Maßnahmen zur Herstellung von Qualität bei der Softwareentwicklung

Das Bereitstellen von Patches und Updates hat zum einen das Ziel, neue Funktionalitäten bereitzustellen. Zum anderen werden mit der Zeit des Einsatzes aber auch Schwachstellen und Sicherheitslücken bekannt, die ebenfalls durch Aktualisierungen beseitigt oder zumindest abgemildert werden können. Auch gesetzliche Änderungen machen die Auslieferungen von Patchen und Updates erforderlich. Das Bereitstellen von Patches und Updates ist für Unternehmen jedoch auch mit Kosten verbunden. Daher stellt sich die Frage, **weshalb Unternehmen fortlaufend in die Qualitätssicherung ihrer Software Produkte investieren?**

Viele Hersteller können es sich nicht leisten, die Sicherheit ihrer Software zu vernachlässigen. Wenn Fehler erst nach Veröffentlichung einer Software erkannt werden, kann dies ein Unternehmen finanziell maßgeblich schädigen. Zum Beispiel durch Reputationsverlust, oder durch die hohen Kosten notwendiger Nachbesserungen. In einigen Fällen können Softwarefehler sogar Menschenleben gefährden. Zum Beispiel im Fall der Boeing 737 Max. Durch einen Softwarefehler ist im Jahr 2018 ein Flugzeug der Firma Boeing abgestürzt, bei denen insgesamt 189 Menschen zu Tode kamen.[2] Wenn ein Softwarefehler hingegen bereits in der Entwicklung erkannt wird, lässt sich dieser Fehler oft innerhalb weniger Stunden kostengünstig beheben. Für Unternehmen ist es daher wirtschaftlich in die Qualität ihrer Software zu investieren. Sie versuchen deshalb zu verschiedenen Phasen der Softwareentwicklung Maßnahmen zur Qualitätssicherung durchzuführen, um Fehler so früh wie möglich zu entdecken.

Die Maßnahmen der Qualitätssicherung in den einzelnen Phasen unterscheiden sich insoweit, dass Teststufen mit jeweils verschiedenen Testzielen zugrunde liegen. So wird oft zwischen Komponenten-, Integrations-, Schnittstellen und schließlich dem Abnahmetest unterschieden. Für die einzelnen Phasen sollten Testumgebungen zur Verfügung stehen, die denen der Produktivumgebung entsprechen. Generell sollten die Maßnahmen und insbesondere Testfälle nachvollziehbar dokumentiert werden, um eine Wiederverwendbarkeit in nachfolgenden Testphasen zu ermöglichen. Standardtestfälle können damit nach einmaliger Erstellung immer wieder zur Ausführung gebracht werden. So wird eine hohe Testfallabdeckung in bereits vorhandenen Funktionalitäten sichergestellt.

Bereits seit einiger Zeit lassen sich gerade Security-Schwächen durch ein sicheres Systemdesign konzeptuell reduzieren. Diesen Ansatz nennt man Security-by-Design. Es geht dabei darum die Sicherheit der Software bereits während des Entwicklungsprozesses sicherzustellen. Dies betrifft die Planung, die Architektur und die Qualität des Codes und kann sowohl das Management, als auch operative Maßnahmen umfassen. So muss zum Beispiel zu verschiedenen Zeitpunkten getestet werden, ob die Software den gewünschten Funktions- und Sicherheitsanforderungen

2 ↗ https://www.tagesschau.de/wirtschaft/boeing-235.html

standhält. In Funktions- und Integrationstests wird zudem getestet, wie verschiedene Komponenten des Codes zusammen wirken. Die Software muss auf Sicherheitslücken, sowie auf Zuverlässigkeits- und Performance-Probleme untersucht werden. Bevor eine Software veröffentlicht wird, ist es auch üblich in Penetrationstests zu prüfen, wie das System unter Belastung reagiert. Zu verschiedenen Zeiten können mit Kunden auch Akzeptanztests durchgeführt werden, um zu testen, ob die Software den Ansprüchen der Nutzer genügt. Und auch nach der Veröffentlichung muss die Software in regelmäßigen Abständen gewartet werden. Security-by-Default, beschreibt einen weiteren Ansatz, um die Sicherheit von Software zu erhöhen. Dabei handelt es sich um die Grundeinstellungen, die bei der Installation der Software empfohlen werden und gleichzeitig die beste Sicherheit für die Software ermöglichen.

7 Softwareerstellung ist unabhängig von einem Herstellerland

Softwareentwicklung findet häufig über geographische und politische Grenzen hinweg statt. Dabei sind Formen des Outsourcings, wie Near- und Offshoring üblich. In der gesellschaftlichen Wahrnehmung hat sich allerdings das Gütesiegel »Made in Germany« als besondere Auszeichnung qualitativ hochwertiger Produkte etabliert. Im Umkehrschluss kann dadurch der Eindruck entstehen, dass Software umso unsicherer ist, je mehr davon in Offshore-Ländern umgesetzt wird. Damit geht auch die Frage einher: **Wenn die Hersteller alles in Deutschland machen würden, wäre dann nicht die Sicherheit von softwarebasierten Produkten besser?**

Die einfache Antwort lautet: Nein! Es gibt keine belastbaren Studien, die besagen, dass Software, die in Deutschland entwickelt wird, »besser« (im Sinne von besserer Qualität oder Sicherheit) ist als Software, die Offshore entwickelt wurde. Softwareunternehmen bedienen sich aus unterschiedlichen Gründen global agierender Ressourcen. Dies mag vordergründig aus Kostengründen heraus geschehen. Speziell in Deutschland kommt heute in jedem Fall auch der Fachkräftemangel hinzu. Zudem werden in vielen komplexen Softwareprodukten Komponenten aus Open-Source-Projekten, oder aus anderen Softwareprodukten verwendet. An der Wertschöpfungskette von Software sind daher meist viele Personen beteiligt, die in verschiedenen Ländern arbeiten.

Grundsätzlich gilt, je komplexer eine Software ist, desto wahrscheinlicher wird es, dass sie Fehler und Sicherheitslücken enthält. Dies gilt für jede Software, unabhängig von ihrer Herkunft. Ausschlaggebend für die Sicherheit ist vielmehr, wie viel Aufwand in die Qualitätssicherung einer Software investiert wird. In Ländern und in Branchen, in denen die Anforderungen an die IT-Sicherheit besonders hoch sind, hat die Qualitätssicherung von Software daher eine besonders hohe Priorität. Rechtsverstöße oder Vertragsverletzungen können für Unternehmen große finanzielle und rechtliche Konsequenzen haben. Insofern steigt für die Unternehmen auch das ökonomische Interesse, in die Qualität ihrer Software zu investieren. Es kann für Unternehmen jedoch auch sinnvoll sein, Teile ihrer Softwareentwicklung an andere Unternehmen auszulagern, die auf diese Dienstleistungen spezialisiert sind. In vielen Fällen können diese Dienstleister die Qualitätssicherung sehr viel routinierter und kostengünstiger erbringen, als das Unternehmen dies selbst gekonnt hätte.

8 Warum sich die Sicherheitsanforderungen für Software zwischen verschiedenen Branchen unterscheiden

Bei komplexen Software-Produkten, die z. B. in Fahrzeugen und Flugzeugen eingesetzt werden, scheint die Sicherheit von Software beherrschbar zu sein. Bei anderen Softwareprodukten, werden hingegen gefühlt ständig Sicherheitsmängel bekannt, wie z. B. aktuell in 2020 bei Tools, die wir für Videokonferenzen nutzen. Daher stellt sich die Frage, **warum Softwaresicherheit in einigen Branchen und Industriezweigen scheinbar besser funktioniert als in anderen?**

In Branchen, die stark reguliert sind, sind die Sicherheitsanforderungen für Software besonders hoch. Dies führt dazu, dass ein überdurchschnittlicher Anteil von Projektbudgets in die Qualitätssicherung von Software investiert wird. Dies trifft zum Beispiel auf den Bereich Automotive und Aviation zu. Auch im Bereich der Medizintechnik, oder des Bankenwesens gelten besondere Qualitätsanforderungen für bestimmte Produkte.

Für Produkte wie Flugzeuge oder Roboter, die physischen Schaden bewirken können, existieren heute bereits starke Regularien, die ein Vielfaches an Qualitätssicherungsmaßnahmen (und -kosten) erzwingen. So muss Software, die zum Beispiel dazu beiträgt, dass ein Flugzeug in der Luft bleibt, vielen Regularien erfüllen (z. B. STANAG 4671, DO-178-B, DO-178C oder DO-254). Diese erzwingen teilweise sehr klare Qualitätssicherungsmaßnahmen wie z. B. Peer-Review, wonach jede Zeile Code von mindestens einem weiteren Programmierer freigegeben werden muss, Testcode-Abdeckungen, wonach so intensiv getestet werden muss, dass wenigstens jede Anweisung und jede Bedingung einmal nachweislich durchlaufen wurde, oder auch das Verbieten besonders moderner Programmierung (z. B. Polymorphie in objektorientierten Sprachen), da diese nicht transparent genug sind. Dies führt dazu, dass üblicherweise in solchen Systemen mehr als 50–90 Prozent[3] des gesamten Projektbudgets in Qualitätssicherung investiert wird, wodurch potentiell mehr Fehler früher erkannt werden können.

In vielen Fällen steht auch das ökonomische Interesse der Unternehmen im Vordergrund. Sicherheitslücken, sowie Funktionalitäts- und Performance-Probleme in der Software können sich auf die Umsätze, sowie die Reputation eines Unternehmens auswirken. Dies ist vor allem in Branchen relevant, die auf das Vertrauen der Nutzer angewiesen sind. Vereinfacht ausgedrückt: »Bei einer Bank auf der mein Geld nicht sicher ist, möchte ich nicht anlegen.« Deshalb stufen diese Unternehmen das Risiko, das von Softwarefehlern ausgeht, in ihrer internen Risikobewertung besonders hoch ein und treffen entsprechende Gegenmaßnahmen. In diesen Branchen wird ebenfalls viel in die Qualitätssicherung von Software investiert.

3 ↗ https://www.verifysoft.com/de_softwaretest.html

9 Sicherheitserwartungen bei der Softwarenutzung erfüllen

Es zeigt sich, dass es möglicherweise also Unterschiede zwischen der Spezifikation einer Software und den Erwartungen an eine Software gibt. Es bleibt daher die Frage, **wie die Sicherheitserwartungen auch bei der Softwarenutzung adressiert werden können**, wenn dies bei der Erstellung ja auch der Fall ist.

Das Notebook eines Mitarbeiters ist per LAN mit dem Unternehmensnetzwerk verbunden. Wird der Arbeitsplatz mobilisiert oder ins Home Office verlegt, so gleicht die Erwartungshaltung hinsichtlich der Sicherheit jener des stationären Betriebes, da schließlich der selbige Anwender auf dieselben Daten zugreift. Die Umgebung unterscheidet sich jedoch und es kann leichter zur Übertragung von/Infizierung mit Schadcode kommen. Daher sind zusätzliche Komponenten, wie beispielsweise eine Personal Firewall, aus Sicherheitsaspekten in Echtzeit durch eine differenzierende Sicherheitsstruktur hinzuzuschalten, um die Erwartung zu erfüllen. Zur Verbesserung der Verfügbarkeit sind diese Komponenten im Betrieb im Unternehmensnetzwerk ausgeschalten und nicht zwingend erforderlich. Sicherheit bedarf verschiedener Komponenten, die wie bereits erwähnt auch das Betriebssystem, auf welchem die Anwendung läuft, sowie die Einbindung der Klassifikationsinformationen umfassen. Um der Erwartung gerecht zu werden, muss die Sicherheitsarchitektur folglich die Nutzungsszenarien differenzieren können.

Problematisch ist hierbei nicht zuletzt die fehlende Metrik zur Bewertung der erreichten Sicherheit. Ein Zertifikat nach Common Criteria scheint eine gewisse Sicherheit der Software zu garantieren. Dies nimmt jedoch Bezug zu definierten Gefahren in idealem Betrieb und sagt somit lediglich aus, dass die Software ihr Soll erfüllt, jedoch nicht, dass Sie sicher ist. Folglich lässt erst eine Prüfung der Robustheit des Gesamtsystems im Betrieb eine realitätsnahe Beurteilung der Sicherheit zu.

Grundsätzlich ist immer zu hinterfragen, welcher Sicherheitsanker auf welcher Schutzannahme beruht, um nachvollziehen zu können, ob der Schutzanker in der eigenen Struktur ebenfalls Wirkung hat und gegebenenfalls ungewollte Effekte im Betrieb hervorrufen kann. Bereits fehlerhafte Installation und administrative Pannen können die Schutzwirkung negativ beeinflussen, ebenso wie fehlende Aufklärung des Anwenders über sicheres Handeln in der digitalen Umgebung. Organisation zur Generierung von User Awareness ist ein ebenso bedeutender Faktor, wie die technische Komposition zu einer sicheren, durchgehenden Vertrauenskette. Auch Intransparenz der Lieferketten, vorangetrieben durch Multisourcing sowohl in Soft- als auch Hardware, kann zu ungeahnten Sicherheitsdefiziten führen. Mit dem Heartbleed-Exploit wurde bspw. eine Schwachstelle in einer gängigen SSL-Implementierung ausgenutzt. Softwarehersteller integrieren solch weit verbreitete De-facto-Standards wie auch Drittprodukte, was dem Endkunden unbekannt bleibt, sodass dieser bei Bekanntwerden von Sicherheitslücken nicht reagiert, da er glaubt, nicht betroffen zu sein. Es ist somit für die schnelle Reaktion auf Angriffe sinnvoll/erforderlich, zu wissen, welche Open-Source-Software und Bibliotheken von Dritten implementiert sind, und dies auch bereits in die Kaufentscheidung einzubeziehen. Somit ist der gesamte Software-Lebenszyklus von der Bedarfsermittlung, über Ausschreibung, Herstellung, Marketing, Vertrieb, Kauf, Instal-

lation und Betrieb in die Betrachtung einzubeziehen. Hinzu kommt die Handlungskette im Sinne des Umgangs mit Daten, welche konkret Datenursprung und -transport, Zugriff über Services und Netzwerke, Darstellung der Daten für den User über Anwendungen, Weiterverarbeitung, Speichern, Drucken, Senden, Löschen u. a. über meist wiederum andere Anwendungen beinhaltet und somit die Integrität bedrohen kann. Die gesamte Handlungskette ist daher so zu organisieren, dass ein adäquater Schutz in jeglichen Gegebenheiten besteht, und im Sinne von Integrität und Vertraulichkeit gehandelt wird. Die Unterbringung von sensiblen Daten in geschützten Datenräumen, das Agieren von Anwendern und Services nur unter Kontrolle in diesem Datenraum zu gewähren, ist in Anbetracht des vollständigen Software-Lebenszyklus nicht ausreichend, wie der Angriff auf das interne Netz des Bundestages von 2015 zeigte. Die Identität eines Anwenders kann auch durch andere Angriffe, wie Mimikatz und PassTheHash, gestohlen werden und zum unbefugten Zugriff auf sensible Daten führen. Daher sind insbesondere solche IT-Umgebungen als nicht-vertrauenswürdig zu kategorisieren, welche über viele Services mit dem Internet verbunden sind und Nutzdaten austauschen. Dann ist in der Sicherheitsarchitektur, bei gleichzeitiger Nutzung einer Kennung für diese Umgebungen, ein Zusatzschutz für sichere Datenräume notwendig. Die Vertrauensstellung der einzelnen Systeme untereinander ist dabei von Bedeutung. Ein System mit hoher Vertrauensstellung kann ein automatisches Login in ein niedrigeres System durchführen, jedoch nicht umgekehrt.

10 Welche Rolle der Staat spielt, wenn es um die Sicherheit von Software geht

Wenn also kein 100%-ige Sicherheit von Software durch den Hersteller und auch nicht durch den Nutzer garantiert werden kann, wer könnte dies dann? Könnte es seitens des Staats entsprechende Maßnahmen geben? **Welche Rolle spielt der Staat, wenn es um die Sicherheit von Software geht?** Es ist eine grundsätzliche Aufgabe jedes Staates, seine Bürger zu schützen und ihre Sicherheit zu gewährleisten. Dies schließt auch die Abwehr von Gefahren ein, denen Bürger durch die zunehmende Digitalisierung ausgesetzt sind. Zur Erfüllung dieser Aufgabe muss ein Staat Mindestanforderungen an die Sicherheit von Software über die gesamte Wertschöpfungskette von der Entwicklung über den Betrieb bis zur Nutzung von Software vorgeben, idealerweise unterschieden nach Branchen und nach Kritikalität hinsichtlich der Daten bzw. der Systemrelevanz. Die Einhaltung der Sicherheitsanforderungen muss überprüft und Verstöße müssen angemessen sanktioniert werden.

In Deutschland gibt es für verschiedene Branchen eigene Regulierungsbehörden zur Weiterentwicklung der Regelwerke und zur Kontrolle ihrer Einhaltung. Es gibt Regeln zur Einstufung von kritischen Infrastrukturen mit verbindlichen Vorgaben, die ihrem Schutzbedarf angemessen sind. Meldungen über Sicherheitslücken oder potenzielle Bedrohungen werden durch staatliche Stellen wie das Bundesamt für Sicherheit in der Informationstechnik (BSI) bewertet und zeitnah an beispielsweise Betreiber oder Entwickler kritischer Infrastrukturen weitergegeben. Auch personenbezogene Daten, geistiges Eigentum und Geschäftsgeheimnisse sind hierzulande durch Gesetze geschützt, deren Einhaltung vom Staat kontrolliert und deren Nichteinhaltung entsprechend bestraft wird.

Angesichts eines hohen nationalen Sicherheitsstandards, wie wir ihn beispielsweise für kritische Infrastrukturen in Deutschland haben, darf nicht übersehen werden, dass für Software länderübergreifende Wertschöpfungsketten längst zum Normalfall geworden sind. So werden in der Entwicklung häufig Komponenten wiederverwendet, die aus unterschiedlichsten Ländern stammen, oder Betriebsstandorte werden je nach Auslastung automatisch zwischen in verschiedenen Ländern lokalisierten Rechenzentren umgeschaltet.

Solche länderübergreifenden Wertschöpfungsketten schränken die Wirksamkeit guter nationaler Sicherheitsstandards ein, sofern nicht alle Staaten ein vergleichbares Sicherheitsniveau konsequent fordern, deren Einhaltung kontrollieren und gegen Verstöße vorgehen. Aus der Aufgabe des Staates, Gefahren der Digitalisierung von seinen Bürgern abzuwehren, ergibt sich aufgrund der Globalisierung implizit auch seine Pflicht zur Kooperation mit anderen Staaten, um das globale Sicherheitsniveau zu harmonisieren. Jegliche Unterschiede in den Sicherheitsanforderungen, Gesetzen, insbesondere auch der Kontrolle ihrer Einhaltung, werden zwangsläufig dazu führen, dass sich Unternehmen, die Software entwickeln, betreiben oder die in einer anderen Art und Weise an der Wertschöpfungskette beteiligt sind, sich in Ländern mit geringeren

Verpflichtungen niederlassen, weil dies ihre Wirtschaftlichkeit und somit ihre Wettbewerbsfähigkeit erhöht.

Neben seiner Rolle als Regulierer und Moderator einer internationalen Harmonisierung von Sicherheitsanforderungen kommt dem Staat noch eine weitere Rolle zu. Er muss neue technische Entwicklungen zeitnah erkennen, fördern, sie selbstverständlich zur Erhöhung der eigenen Wettbewerbsfähigkeit nutzen, jedoch gleichzeitig Risiken erkennen und ebenso zeitnah notwendige Maßnahmen ergreifen. Dies erfordert eine enge Vernetzung der staatlichen Regulierungs- und Aufsichtsbehörden mit der Forschung. Als Beispiel mögen Entwicklungen im Umfeld der Künstlichen Intelligenz dienen, die uns einerseits viele Chancen eröffnen, andererseits aber auch Risiken mit sich bringen. In einer globalisierten Welt ist es notwendig, dass neue technische Entwicklungen umgehend auch eine Weiterentwicklung der Sicherheitsanforderungen an Software einleiten – in einem globalen Rahmen und mit gleicher Intensität und Konsequenz für alle Staaten.

11 Wie die vertragliche Haftung für Softwarequalität geregelt ist

Softwareanbieter haben nach den Vorgaben des Vertragsrechts für die Qualität ihrer Leistung einzustehen. Daher stehen dem Softwarekäufer nach Kaufvertragsrecht und dem Besteller einer Software nach Werkvertragsrecht Mangelansprüche zu, wenn die gelieferte Software einen Sach- oder Rechtsmangel aufweist (§ 437 BGB bzw. § 634 BGB). Nach Mietvertragsrecht muss der Vermieter dafür Sorge tragen, dass die Mietsache während der Vertragslaufzeit zum vertragsgemäßen Gebrauch geeignet ist (§ 535 BGB). Weist eine gemietete Software einen Sach- oder Rechtsmangel auf, der den vertragsgemäßen Gebrauch nicht nur unerheblich beeinträchtigt, steht auch dem Mieter ein Mangelanspruch zu.

Die genannten gesetzlichen Regelungen gelten ihrem Wortlaut nach nur für Sachen. Inzwischen hat jedoch die Rechtsprechung geklärt, dass diese Vorschriften ohne Einschränkung auch auf Software Anwendung finden. Die Mängelansprüche des Softwarenutzers bei Sach- und Rechtsmängeln umfassen je nach zugrundeliegendem Vertragstyp z. B. Rücktritt vom Vertrag, Schadensersatz oder Mietminderung.

Ein Rechtsmangel bei Software besteht insbesondere dann, wenn der Anbieter dem Softwarenutzer die zum vertragsgemäßen Gebrauch der Software notwendigen Nutzungsrechte (Lizenzen) nicht oder nicht in ausreichendem Umfang überträgt.

Ein Sachmangel liegt vor, wenn der Vertragsgegenstand nicht der vertraglichen Vereinbarung entspricht, sich nicht für die vertraglich vorausgesetzte Verwendung eignet oder eine üblicherweise zu erwartende Beschaffenheit nicht aufweist (§ 434 BGB bzw. die vergleichbare Formulierung in § 633 BGB). Ein Sachmangel von Software ist darüber hinaus anzunehmen bei einer fehlenden oder nicht ausreichenden Dokumentation (Benutzerhandbuch oder Programmbeschreibung). Die üblicherweise zu erwartende Beschaffenheit von Software ergibt sich aus dem aktuellen Stand der Technik. Insoweit verweist das Recht für die Bestimmung von Sachmängeln auf außerrechtliche, technische Vorgaben für Design, Herstellungsprozess und Qualitätskontrolle von Software. Dennoch sind rechtlicher und technischer Mangel- bzw. Fehlerbegriff nicht deckungsgleich. Denn das Vertragsrecht orientiert sich bei der Bestimmung eines Mangels in erster Linie an der vertraglichen Vereinbarung. So kann eine Software technisch einwandfrei programmiert und dennoch mangelhaft sein, wenn sie vertraglich zugesicherte Funktionen nicht besitzt. Andererseits führt ein Softwarefehler dann nicht zu einem Sachmangel im rechtlichen Sinn, wenn der Softwarefehler trotz Einhaltung aller technischen Standards und Qualitätsanforderungen aufgetreten ist und sich daher nicht vermeiden ließ.

Die vertragliche Verantwortlichkeit des Anbieters für die Mangelfreiheit des Vertragsgegenstands reicht bei Kauf- und Werkverträgen bis zum Zeitpunkt des Gefahrübergangs (Einräumung einer endgültigen Nutzungsmöglichkeit am Vertragsgegenstand für den Kunden). Der Anbieter muss also nicht für Gebrauchsbeeinträchtigungen haften, die ihre Ursache in der Zeit nach Gefahrübergang haben. Wird z. B. ein neuartiges Computervirus erst nach Auslieferung der Software bekannt, trifft den Softwareanbieter keine vertragliche Haftung für Softwarelücken, die von

dem neuartigen Virus ausgenutzt werden. Allerdings ist insoweit die Rechtslage im Fluss. Die **↗ Richtlinie EU/2019/770 über bestimmte vertragsrechtliche Aspekte der Bereitstellung digitaler Inhalte und digitaler Dienstleistungen** sieht in Art. 8 Abs. 2 vor, dass der Anbieter beim Verkauf digitaler Inhalte (zu denen auch Software zählt) an einen Verbraucher für eine gewisse Zeit Aktualisierungen (Updates) bereitstellen muss. Die Richtlinie ist aber derzeit (Juli 2020) noch nicht in deutsches Recht umgesetzt.

Versuche von Software-Anbietern, ihre vertragliche Haftung für Sach- und Rechtsmängel auszuschließen oder zu begrenzen, sind weitgehend unwirksam. So verbietet z. B. § 476 BGB für Verträge mit Verbrauchern jede vertragliche Regelung, die Mangelrechte des Verbrauchers einschränken würde. Auch kann die Haftung für vorsätzliches Fehlverhalten nicht ausgeschlossen werden (§ 276 Abs. 3 BGB). Darüber hinaus sind die gesetzlichen Vorgaben für Allgemeine Geschäftsbedingungen (AGB) zu beachten Unter AGB versteht man Vertragsklauseln, die durch eine Vertragspartei einseitig vorgegeben werden. Der Ausschluss sämtlicher Mangelansprüche in AGB verstößt gegen § 309 Nr. 8 b) BGB und ist damit unwirksam. Da die Mangelansprüche zu den wesentlichen gesetzlichen Vertragsvorgaben gehören, gilt dieses Verbot nicht nur für Vertragsbeziehungen mit Verbrauchern (B2C), sondern gemäß § 307 BGB auch im Geschäftsverkehr zwischen Unternehmen (B2B).

12 Die Verantwortung der Hersteller

Aus den bisherigen Darstellungen wird deutlich, dass es im Regelfall keine Garantien für das einwandfreie Funktionieren einer Software geben kann. Das gilt vor allem dann, wenn der Kontext der zukünftigen Verwendung nicht klar definiert werden kann und wenn der Einsatz der Software mit nicht vorhersehbaren Bedrohungen verbunden ist. Dennoch stellt sich die Frage nach den Konsequenzen, wenn Fehlfunktionen oder Sicherheitslücken in softwarebasierten Produkten Schäden verursachen. **Wer kann in diesen Fällen zur Verantwortung gezogen werden?** Ein Softwarenutzer kann Sicherheitslücken oder Qualitätsmängel an einer Software regelmäßig nicht selbst beheben und ist dazu wegen urheberrechtlicher Restriktionen auch nur selten befugt.

Soweit zwischen dem Nutzer und dem Anbieter der Software ein Vertragsverhältnis besteht, bietet das Vertragsrecht Schutz vor Schäden. Verursacht eine Software allerdings Schäden außerhalb eines bestehenden Vertragsverhältnisses, kommt nur eine Haftung nach den Grundsätzen der Produkt- oder der Produzentenhaftung in Betracht.

a) Produkthaftung

Im Rahmen der Produkthaftung hat ein Hersteller Schadensersatz zu leisten, wenn ein Fehler seines Produkts dazu führt, dass ein Mensch getötet, sein Körper oder seine Gesundheit verletzt oder Privateigentum geschädigt wird. In gleicher Weise haftet auch der Hersteller eines Teilprodukts, wenn dieses Teilprodukt den Schaden verursacht hat (§ 4 Abs. 1 ProdHaftG). Von der Produkthaftung nicht erfasst werden Schäden am fehlerhaften Produkt selbst und reine Vermögensschäden (z. B. entgangener Gewinn oder Datenverlust).

Nach dem Wortlaut des Produkthaftungsgesetzes gilt als Produkt jede »bewegliche«, d. h. körperliche Sache. Ob die Produkthaftung auf den unkörperlichen Gegenstand Software angewendet werden kann, ist in der Rechtswissenschaft umstritten. Nach der Rechtsprechung lösen aber jedenfalls Fehler in der Steuerungssoftware von Geräten oder Gerätekomponenten eine Produkthaftung des Geräte- bzw. des Komponentenherstellers aus.

Ein Produkt weist einen Fehler auf, wenn es nicht die Sicherheit bietet, die berechtigterweise zu erwarten ist (§ 3 Abs. 1 ProdHaftG). Dabei richtet sich das einzuhaltende Sicherheitsniveau nach dem Stand von Wissenschaft und Technik zu dem Zeitpunkt, zu dem das Produkt in Verkehr gebracht wird. Der Stand der Wissenschaft und Technik wird vor allem durch technische Standards, z. B. DIN-, CEN- und ISO-Normen indiziert. Darüber hinaus können technische Anforderungen an besonders gefahrgeneigte Produkte auch in speziellen Gesetzen festgelegt sein, z. B. für Medizingeräte oder Pharmaprodukte.

Ein Verschulden des Herstellers, also ein vorwerfbarer Verstoß gegen eine Sorgfaltspflicht, ist keine Haftungsvoraussetzung für die Produkthaftung. Der Hersteller kann sich jedoch von einer Produkthaftung befreien, wenn er nachweist, dass in seiner Verantwortungssphäre kein haftungsrelevanter Fehler begangen wurde. Der Hersteller ist auch nicht für die Weiterentwicklung von bereits im Markt befindlichen Produkten verantwortlich. Ändern sich nach Inverkehrbringen eines Produkts die Sicherheitsstandards für dieses Produkt, so muss dies nur für neu in Verkehr

gebrachte Produkte beachtet werden. Desgleichen ist ein Hersteller nach Produkthaftungsrecht nicht verantwortlich, wenn die Fehlfunktion durch Veränderungen beim Einsatz des Systems verursacht wird. So kann ein Softwarehersteller nicht zur Verantwortung gezogen werden, wenn der Nutzer die Software umprogrammiert, sie außerhalb des vorgesehenen Anwendungsbereichs einsetzt oder sie in einer unsicheren Umgebung ablaufen lässt.

b) Produzentenhaftung

Das Recht verpflichtet jeden Produzenten, das ihm Mögliche und Zumutbare zu tun, um Gefahren seiner Produkte abzuwenden. Tut er dies nicht, droht ihm die von der Rechtsprechung aus § 823 Abs. 1 BGB abgeleitete Produzentenhaftung. Die Produzentenhaftung kommt zur Anwendung, wenn die Verletzung eines Menschen, die Beschädigung einer Sache oder ein Eingriff in ein eigentumsgleiches Recht kausal auf die Missachtung einer Verkehrssicherungspflicht durch den Hersteller zurückzuführen ist. So hat der Hersteller u. a. die Pflicht, sein Produkt und dessen Nutzung am Markt zu beobachten und erkannte Gefahrenquellen des Produkts durch angemessene Maßnahmen zu beseitigen.

Anders als nach dem Produkthaftungsgesetz ist die Haftung des Unternehmers für fehlerhafte Produkte nach § 823 Abs. 1 BGB tatbestandlich nicht auf Sachen beschränkt. Daher findet die Produzentenhaftung ohne weiteres auch Anwendung auf Schäden, die durch fehlerhafte Software verursacht wurden.

Damit rückt die Frage in den Fokus, welche Maßnahmen von Softwareherstellern zu ergreifen sind, um eine Gefahr ihres Produkts und damit eine Produzentenhaftung abzuwenden. Die bisherige Rechtsprechung gibt hierfür keine ganz klaren Grundsätze vor. Eine absolute Gefahrlosigkeit von Produkten erwartet die Rechtsprechung nicht. Jedoch erhöhen sich die Sorgfaltsanforderungen an den Hersteller, wenn schwerwiegende Schäden an wertvollen Rechtsgütern mit nicht zu vernachlässigender Wahrscheinlichkeit drohen.

Zunächst darf ein Software-Hersteller Software nicht in den Markt bringen, wenn ihm Fehler oder Sicherheitslücken dieser Software bekannt sind. Zeigt sich – z. B. im Rahmen der Produktbeobachtung – ein Fehler oder eine Sicherheitslücke erst bei einer bereits im Einsatz befindlichen Software, kann es genügen, dass der Hersteller vor dem Einsatz der Software warnt. Eine Warnung kann insbesondere bei individueller oder wenig verbreiteter Unternehmenssoftware ausreichen. Wenn jedoch davon auszugehen ist, dass diese Maßnahme nicht ausreicht, um das Sicherheitsrisiko auf ein zulässiges Maß zu reduzieren, kann der Hersteller zu Reparatur oder Nachrüstung bzw. im Fall von Software zur Bereitstellung von Updates verpflichtet sein.

Zu welchen Maßnahmen der Hersteller im Einzelfall verpflichtet ist, richtet sich nach einer umfassenden Interessenabwägung im Einzelfall, wobei insbesondere das Ausmaß der Gefahr, das gefährdete Rechtsgut und die wirtschaftliche Zumutbarkeit eine Rolle spielen. Solange der Hersteller einer Standard-Software die Software noch vertreibt, muss er bekannte Fehler und Sicherheitslücken in neuen Versionen der Software ausmerzen. Denn er darf keine Software

mit bekannten Sicherheitslücken ausliefern. Es stellt für ihn daher in der Regel keinen unzumutbaren Aufwand dar, entsprechende Patches auch für bereits ausgelieferte Kopien der Software bereit zu stellen. Beendet ein Softwarehersteller den Vertrieb einer Software, gehen die Anforderungen an die Produktbeobachtung und an den Produktsupport sukzessive zurück, sodass Warnungen vor neuen Gefahren im Zusammenhang mit der Software im Regelfall ausreichen.

13 Wodurch sich sichere Software auszeichnet

Damit stellt sich im Weiteren die Frage, **wie bei der Anschaffung einer Software diese als sicher erkannt werden kann**. Die Antwort fällt derzeit entsprechend kurz aus: Eine sichere Software kann man nicht erkennen, da es keine 100%ige Sicherheit gibt und weil Software immateriell ist. Dennoch unternehmen Softwarehersteller entsprechende Anstrengungen, um den Prozess der Softwareherstellung sicher zu durchlaufen und bestmöglich auf Gefährdungen und Gefahren, die im späteren Gebrauch auftreten können, einzugehen.

Sichere Softwareentwicklung zeichnet sich durch eine Entwicklung entlang von Prozessschritten aus, die schon von Beginn an die Sicherheitsaspekte in der Entwicklung berücksichtigen. Microsoft hat hierfür mit dem Security Development Lifecycle (SDLC) eine übersichtliche Darstellung der notwendigen Maßnahmen aufgezeigt. Software Entwickler arbeiten schon während der Entwicklung mit Security Code Scannern, die statische, dynamische und Fuzzing Scans durchführen und schlechten Code schon in der Entstehung der Software identifizieren. Durch diese Security Driven Development genannte Entwicklung wird bereits in den frühesten Phasen versucht die Software auf Zero-Trust Workplaces (unsichere Arbeitsplätze) zu gestalten. In der agilen Softwareentwicklung geschieht dies z. B. durch Erweiterungen des Scrum-Modells. Neben den User Stories (Anwendungsfällen) werden auch Misuse Stories (Missbrauchsszenarien) verwendet und ein Security Feature Backlog angelegt, also eine Liste mit umzusetzenden Sicherheitsaspekten. In der Qualitätssicherung ergeben sich dadurch zusätzliche Maßnahmen, die in der Folge skizziert werden.

Code Analyse Tools untersuchen die Software und nutzen dazu verschiedene Metriken (Messmethoden und Berechnungen), die Auskunft über die Qualität der Software geben. Dabei werden Kriterien wie Komplexität, Duplikate, Code-Smell (zum Beispiel toter Code oder Gottklassen), mangelnde Testabdeckung, Architekturfehler und Sicherheitsmängel überprüft und in Grafiken und Kennzahlen dargestellt.

Die Abbildung 1 zeigt ein Analyseergebnis einer Software. Die Bewertungsskala geht von –5 (sehr schlecht) bis +5 (sehr gut). In diesem Beispiel ist zu erkennen, dass eine sehr hohe Komplexität von 215 erkannt wurde (der Schwellwert liegt bei 50). Darüber hinaus konnten auch 12 kritische Sicherheitslücken identifiziert werden. OpenSSL hat bereits 2014 durch die Heartbleed-Sicherheitslücke auf sich aufmerksam gemacht. Da OpenSSL für die Verschlüsselung von personenbezogenen Daten im Internet verwendet wird, sind hier Sicherheitslücken besonders fatal.

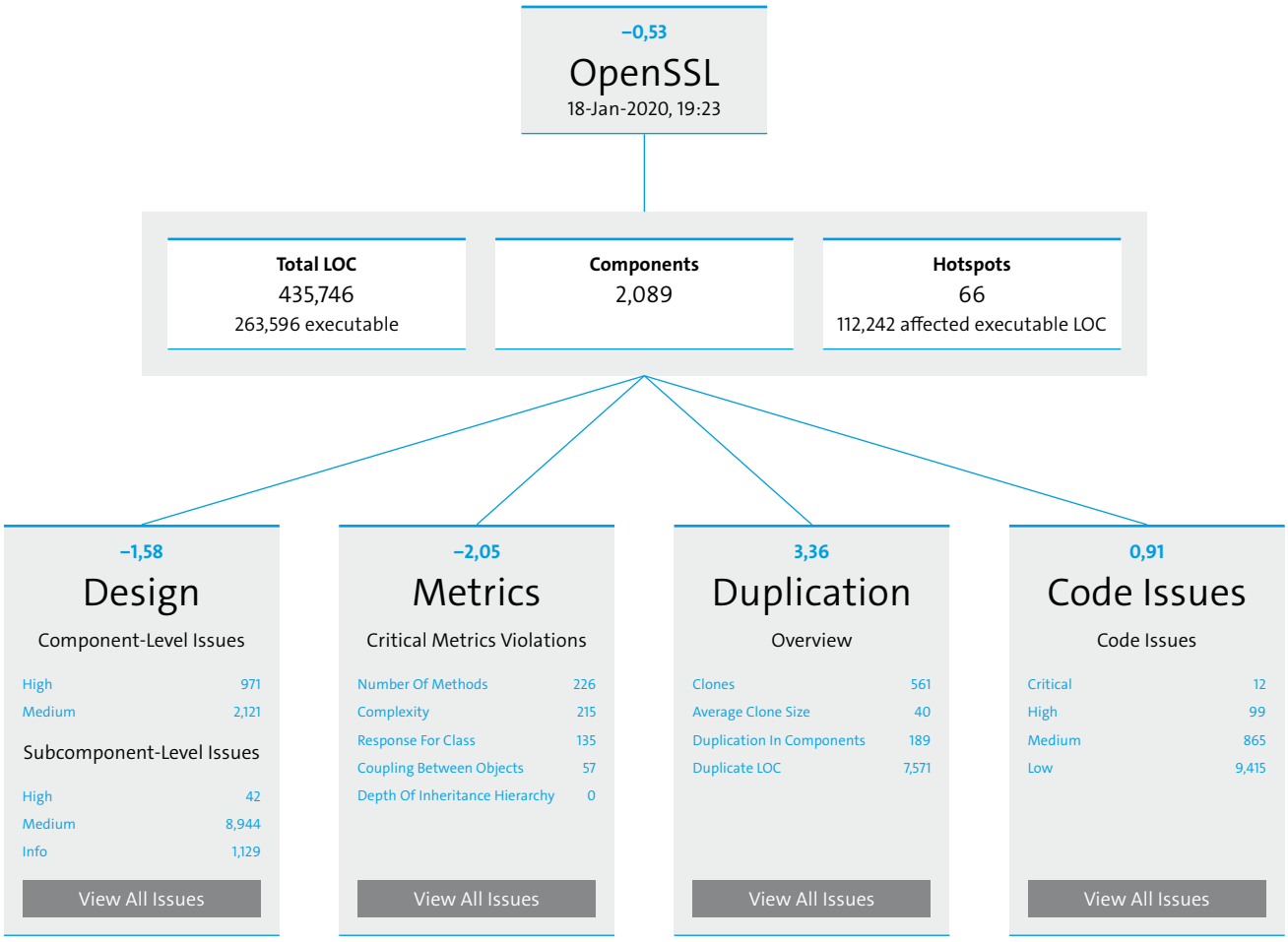


Abbildung 1: Code Analyse OpenSSL mit embold, durchgeführt am 17.1.2020

Je komplexer eine Software-Komponente desto höher ist die Fehlerwahrscheinlichkeit. Deshalb ist bei vielen statischen Code-Analyse-Werkzeugen die Komplexität ein wichtiges Kriterium. Für die Programmierung von guter Softwarequalität wurden Entwicklungsregeln aufgestellt, die als Clean Code bezeichnet werden. Wird von diesen Regeln abgewichen, entsteht Code-Smell – übel riechender Code. Als Code-Smell werden public-Attribute, fehlende Kommentare, lange Parameterlisten etc. bezeichnet. Da es sich hier um formale Syntaxregeln handelt, werden diese Fehler leicht mit Code-Analyse-Werkzeugen gefunden.

Diesen Analyse-Tools liegen meist die Software-Qualitätsstandards zugrunde und nutzen Schwachstellendatenbanken, um Sicherheitslücken in der Software zu lokalisieren.

Diese Analysen setzen voraus, dass der Quellcode zur Verfügung steht.

Zertifikate können ein Indikator für qualitativ hochwertige Software sein (siehe Kapitel 14). Es ist jedoch nicht auszuschließen, dass die Software trotz Zertifikaten nach wie vor Fehler und Sicherheitsmängel aufweist. Ein weiterer Indikator für qualitativ hochwertige Software ist, wenn Einträge in den einschlägigen Sicherheitsschwachstellendatenbanken fehlen: OWASP, CWE, NVD, CAPEC, CVE, VDBs.

Neben statischen Code-Analysen (SAST) werden auch dynamische Code-Analysen (DAST), Feedback-based Application Security Testing (FAST) und Angriffsflächentests durchgeführt, um vor der Auslieferung von Software diese auf Schwachstellen zu analysieren. Wichtig sind die Fuzzing-Tests, da durch diese der Horizont vorher identifizierter Risiken erweitert wird.

14 Zertifikate und Prüfstellen für sichere Software

Im Einzelhandel gibt es verschiedene Zertifikate und Kennzeichnen, an denen sich Verbraucher orientieren können, ob ein Produkt sicher ist. Zum Beispiel die CE-Kennzeichnung. Diese Zertifikate bescheinigen, ob ein Produkt nach bestimmten Qualitätskriterien erstellt und getestet wurde. Ähnliche Zertifikate gibt es auch im Bereich der Softwareentwicklung. Für Verbraucher stellt sich jedoch die Frage, **welche dieser Zertifikate und Prüfstellen sind vertrauenswürdig?**

Hersteller können sich zum Beispiel zertifizieren lassen, dass sie die Empfehlungen des IT-Grundschutz einhalten. Dieses Zertifikat orientiert sich an den internationalen Standards ISO 27001 und ISO 27002. Der IT-Grundschutz setzt technische Sicherheitsmaßnahmen voraus, sowie Maßnahmen, die sich auf die IT-Infrastruktur des Unternehmens beziehen. Es berücksichtigt dabei auch organisatorische und personelle Schutzmaßnahmen. Das Zertifikat sagt im Grunde aus: »Dieses Unternehmen nimmt IT-Sicherheit ernst«.

Für den deutschen Markt erfolgt die Zertifizierung durch das Bundesamt für Sicherheit in der Informationstechnik (BSI). Auch weitere Prüfstellen dürfen Zertifizierungen ausstellen, wenn sie vom BSI akkreditiert sind (z. B. TÜV). Welche Zertifikate diese Prüfstellen ausstellen dürfen, ist auf der Website des BSI einsehbar.

Hersteller können sich auch zertifizieren lassen, dass sie bei der Softwareentwicklung bestimmte Prozesse und Qualitätsstandards einhalten. In diesem Kontext sind vor allem internationale Standards, wie die Common Criteria (CC), TCSEC für den amerikanischen Markt, und ITSEC innerhalb der EU relevant. Diese Zertifikate geben einen Einblick, wie die Software entwickelt wurde und welche Maßnahmen zur Qualitätsprüfung durchgeführt worden sind.

Zertifikate können bislang jedoch wenig über die tatsächliche Sicherheit eines Produktes aussagen. Und selbst wenn der IT-Grundschutz und die Common Criteria eingehalten werden, kann eine Software im Code nach wie vor Qualitäts- und Sicherheitsmängel aufweisen.

Viele mittelständische Unternehmen verzichten daher aus Kostengründen auf eine formelle Zertifizierung, selbst wenn sie sich bei der Entwicklung der Software an den internationalen Standards orientieren. Die Reputation eines Unternehmens und die Zufriedenheit seiner Kunden ist neben Zertifikaten daher auch ein wichtiger Indikator, um die Vertrauenswürdigkeit eines Herstellers zu bewerten.

Das BSI arbeitet zurzeit (Juli 2020) an einem Mindeststandard für die Qualität von Softwareprodukten. Dieser Mindeststandard soll vorerst einen empfehlenden Charakter haben. Es wird jedoch kontrovers diskutiert, ob dieser Mindeststandard für Software Qualität insbesondere für KRITIS-Unternehmen verpflichtend sein sollte.[4]

4 Bundesministerium des Inneren (2016): Cyber-Sicherheitsstrategie für Deutschland 2016, S. 17. Online abrufbar unter: ↗ https://www.bmi.bund.de/cybersicherheitsstrategie/BMI_CyberSicherheitsStrategie.pdf

Die üblichen Bewertungen von Produkten, die auch bei Software von Nutzern angegeben werden, beschreiben nur die durch Nutzung erkennbaren Eigenschaften, wie Bedienbarkeit und Performance. Bereits Stabilität wird selten in diesen Bewertungen beschrieben, und wenn, dann nur ungenau, weil aussagekräftige Messungen und Vergleiche spezialisierte Kenntnisse erfordern. Noch weniger wird die Sicherheit in den Bewertungen berücksichtigt, da diese ebenfalls nur mit spezialisierten Kenntnissen erkennbar ist.

In der Abschlussarbeit von Eric Skowronski: »Klassifizierung und Bewertung der Sicherheitssiegel in Deutschland und Europa«, eingereicht am 28.08.2017 an der HfT Leipzig wurden verschiedene Zertifikate untersucht, die in Deutschland und der EU verbreitet sind. Dabei wurden die Informationen, die zu den Zertifikaten von den zertifizierenden Stellen im Internet bereitgestellt werden, ausgewertet. Alle Zertifizierer wurden von Eric Skowronski kontaktiert und um Bereitstellung zusätzlicher Informationen (wie z. B. die Verbreitung/Reichweite der Zertifikate) gebeten. Einige Zertifizierer haben dazu Angaben gemacht, andere nicht.

Bei den nachfolgenden Grafiken gilt für jedes Kriterium: je größer die blaue Fläche, desto besser das Zertifikat. Die Grafiken sind Bestandteil der Bachelorarbeit von Eric Skowronski und mit seinem Einverständnis von dort entnommen:

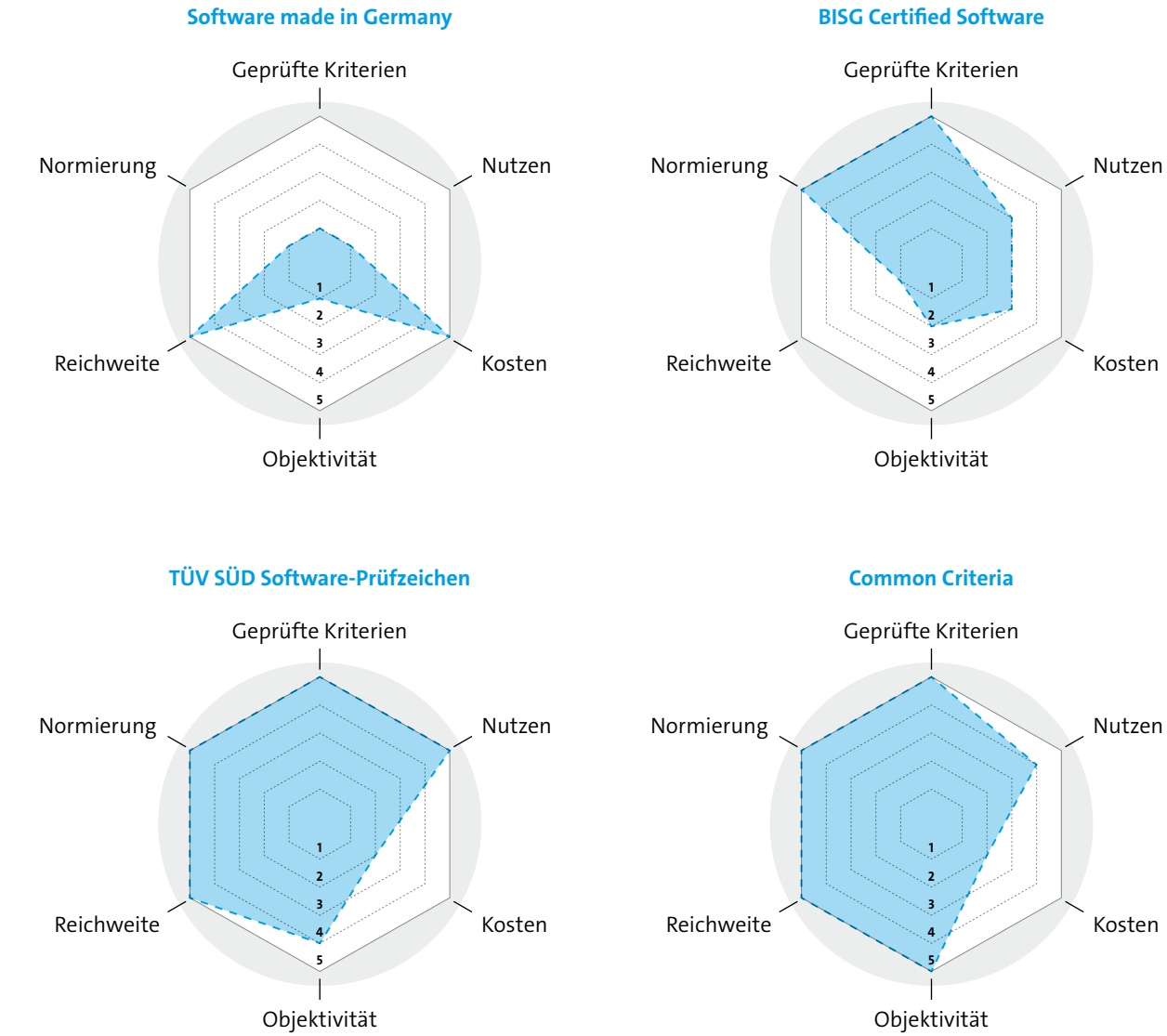
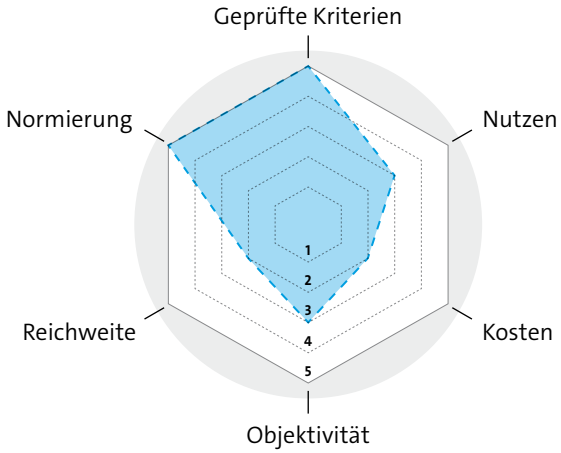


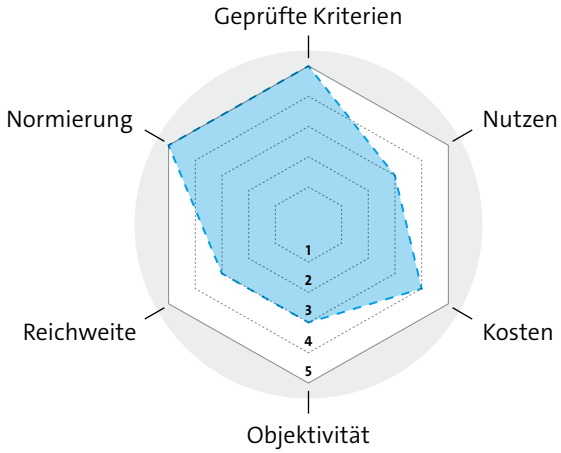
Abbildung 2: Klassifizierung und Bewertung der Sicherheitssiegel in Deutschland und Europa

Ein Zertifikat erhält volle Punktzahl für Normierung, wenn der Prüfprozess einem Standard folgt, also nach ISO 9100 oder ISO 25.000 durchgeführt wird. Ein Siegel wird als objektiv eingestuft, wenn die an der Überprüfung beteiligten Parteien unabhängig voneinander sind. Daher wird mit diesem Kriterium untersucht, ob die beteiligten Parteien in Beziehung zueinander stehen. Unter anderem wird berücksichtigt, ob das Siegel durch die DAkkS akkreditiert ist und ob das Siegel intern überprüft wird (einstufiges Prüfungsverfahren) oder ein externes Gutachten erstellt wird (zweistufiges Prüfungsverfahren). Mit den anderen Kriterien wird ähnlich verfahren.

EuroPrise



Gütesiegel für IT-Produkte (ULD)



IT-Security made in Germany

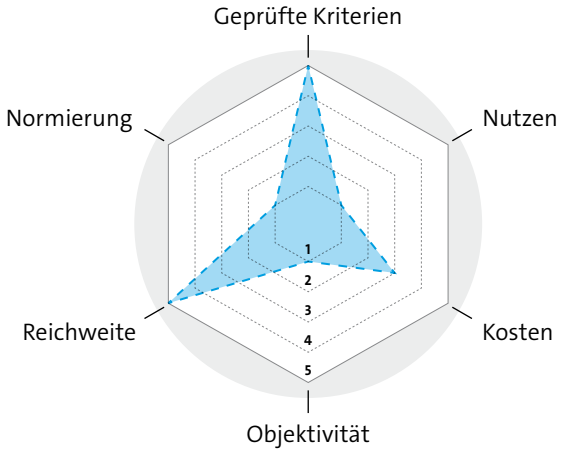


Abbildung 3: Klassifizierung und Bewertung der Sicherheitssiegel in Deutschland und Europa – Fortsetzung

In den Grafiken werden teilweise deutliche Unterschiede zwischen den Zertifikaten deutlich. Dennoch lässt sich daraus nicht eindeutig ableiten, dass einige Zertifikate anderen tatsächlich überlegen sind, oder nicht. Damit lässt sich lediglich konstatieren, dass es eine Reihe von Zertifikaten für Software gibt, deren Vorhandensein für eine Software auf einen achtsamen Umgang des Herstellers mit dem Thema Sicherheit und Softwarequalität schließen lässt.

15 Die Vertrauenswürdigkeit von Software und Herstellern erkennen

Zertifikate sind also ein Indikator für vertrauenswürdige Hersteller und qualitativ hochwertige Produkte. Sie deuten darauf hin, dass ein Unternehmen viel Aufwand in die Qualitätssicherung seiner Produkte investiert. Hersteller können sich zum Beispiel zertifizieren lassen, dass sie die Empfehlungen des IT-Grundschutz einhalten. Dieses Zertifikat orientiert sich an den internationalen Standards ISO 27001 und ISO 27002. Der IT-Grundschutz setzt technische Sicherheitsmaßnahmen voraus, sowie Maßnahmen, die sich auf die IT-Infrastruktur des Unternehmens beziehen. Es berücksichtigt dabei auch organisatorische und personelle Schutzmaßnahmen. Das Zertifikat sagt im Grunde aus: »Dieses Unternehmen nimmt IT-Sicherheit ernst«.

Im Softwareentwicklungsprozess sind Qualitätssicherungsmaßnahmen von Anfang an wichtig. In jeder Phase oder bei jedem Entwicklungsschritt sind Qualitätssicherungsmaßnahmen notwendig. Diese können durch Sicherheitsgates sichtbar gemacht werden. Wie bei dem Aufstellen der Anforderungen werden die Qualitätsanforderungen, meist als »nicht funktionale Anforderungen – NFA« bezeichnet, mit berücksichtigt. Nur wenn die Qualitätsanforderungen formuliert wurden, kann das Sicherheitsgate erfüllt werden. Diese NFA können nicht erst am Ende der Entwicklung berücksichtigt werden, da ein nachträgliches Einbauen sehr aufwändig ist und zusätzliche Fehlerquellen ermöglicht.

COBIT, ITIL und CMMI sind weitere Standards, die anzeigen, dass Unternehmen Qualität und Sicherheit ernst nehmen. Hier sind verschiedene Reifegrade definiert, die zeigen, ob ein Unternehmen ein Qualitätsmanagement betreibt. Es werden beim CMMI fünf Reifegrade beschrieben,

- initial (das Unternehmen startet mit dem Qualitätsmanagement),
- managed (ein Projektmanagement mit systematischer Planung und Umsetzung wurde eingeführt,
- defined (die Projekte werden nach einem Standardprozess durchgeführt),
- quantitatively managed (Kennzahlen zur Steuerung der Prozesse sind vorhanden) und
- optimizing (die Prozesse werden kontinuierlich verbessert).

ITIL beschreibt verschiedene best practices im Managementprozess, um die Prozess- und damit auch die Produktqualität stetig zu verbessern. COBIT ist der internationale Standard der Wirtschaftsprüfer und IT-Auditoren, nach dem Unternehmen bezüglich der durch die IT entstehenden Risiken geprüft und bewertet werden. Hier werden Prozesse, Struktur und speziell die IT-Sicherheit des Unternehmens und der Softwareentwicklung auditiert (geprüft). Der Prüfstandard berücksichtigt auch rechtliche und normative (Zusammenwirken der verschiedenen ISO-Standards) Aspekte.

Auch ein ehrlicher Umgang mit Fehlern kann die Vertrauenswürdigkeit eines Unternehmens verbessern, denn nur erkannte Mängel können abgestellt werden. Eine offene Kommunikation mit Kunden und Vertragspartnern über die eigene Leistungsfähigkeit erzeugt Vertrauen. Die Nutzung von AGB, die den geltenden Gesetzen wie dem Produkthaftungsgesetz entsprechen,

erzeugen ebenfalls Vertrauen. Leider existieren insbesondere bei vielen Softwareherstellern AGBs, die dem Produkthaftungsgesetz widersprechen.

Die Vertrauenswürdigkeit von Herstellern erkennt man auch an Indikatoren wie:

- Wie geht der Hersteller mit Kundenanfragen um?
- Hat der Hersteller ein Versionsmanagement, erkennbar an der Nummerierung und Häufigkeit von Patches?
- Werden Fehler und Sicherheitsmängel am Produkt zügig beseitigt bzw. wie sind die Reaktionszeiten auf Sicherheitsvorfälle (z. B. auch Presseberichte, wie schnell ein Hersteller auf aktuelle Bedrohungen reagiert hat)?
- Wird vom Hersteller versucht mittels AGB eine Haftung auszuschließen?
- Der Umgang von Herstellern mit der Softwarezulieferung (im Vergleich zum Beispiel Apple App Store versus Google Play Store, bei dem ersterer nur durch ihn auf Sicherheit und verdächtigen Code analysierte Software hochlädt, während dies beim Google Play Store zu Beginn gar nicht der Fall war und dann nur rudimentär nachgebessert wurde).
- Wie kommuniziert der Hersteller mit den Vertragspartnern und Kunden?
- Ein sehr guter Indikator ist auch eine Internet-Recherche nach Sicherheitsvorfällen mit der Software, die durch die Trefferlisten sehr schnell zeigt, wie oft der Hersteller mit Sicherheits- und Datenschutzproblemen öffentlich Probleme hatte.

16 Möglichkeiten Sicherheitsvorfälle und Angriffe zu erkennen

Es wird deutlich, dass jeder Einsatz von Software grundsätzlich mit einem Risiko behaftet sein kann. Somit sollte jeder Nutzer mit einer entsprechenden Sensibilität Software nutzen. Gleichzeitig sollten Nutzer auch immer die Möglichkeit einer Gefährdung in Betracht ziehen und daher sehr aufmerksam beim Einsatz von Software sein. Es gilt dann mögliche Bedrohungen und Gefährdungen rechtzeitig zu erkennen. Es stellt sich also die Frage, **wie man erkennen kann, dass man Opfer eines Cyberangriffs geworden ist**.

Die Möglichkeiten zur Erkennung von Cyberangriffen ist abhängig davon, ob man selbst angegriffen wurde oder eigene Daten bei einem Dritten, beispielsweise einem Online-Shop abhandengekommen sind. Generell ist die Analyse von Angriffen jedoch eine Herausforderung.

Ob man selbst angegriffen wurde, kann man in einigen Fällen daran erkennen, dass die eigene Antiviren-Software Alarm schlägt. In diesen Fällen ist die Antiviren-Software meist in der Lage, den Angriff zu unterbinden und die Schadsoftware zu isolieren. Schwieriger sind Fälle, in denen keine Antiviren-Software installiert ist oder diese von der Schadsoftware ausgehebelt wird. In diesen Fällen kann man einen Angriff daran bemerken, dass sich der Computer oder das Smartphone merkwürdig verhält und ein Eigenleben zu entwickeln scheint. Auch die plötzliche Überlastung des eigenen Computers oder Smartphones kann ein Hinweis darauf sein, dass eine unerwünschte Software läuft, die einen Teil der Rechenleistung verbraucht. Unseriöse Pop-Ups auf dem Desktop oder im Browser können ebenfalls ein Indiz für eine Infektion mit Schadsoftware sein. Beteiligt sich der eigene Rechner durch eine Schadsoftware an weiteren Angriffen, verschicken viele Provider Warnhinweise und man kann auf diese Weise von der Infektion des eigenen Systems erfahren. Einen unberechtigten Zugriff auf eigene Online-Konten kann man – unabhängig davon, ob es um ein Online-Banking-Konto, den Account bei einem sozialen Netzwerk oder ein E-Mail-Konto geht – ebenfalls anhand von auffälligem Verhalten erkennen. Sind beispielsweise Zahlungen ohne eigenes Zutun getätigt worden oder Posts bzw. E-Mails versendet worden, sind dies ebenfalls gute Indizien dafür, dass man Betroffener eines Angriffs geworden ist.

Schwieriger wird die Situation, wenn man nicht selbst Opfer eines Angriffs geworden ist, sondern ein Unternehmen, bei dem in irgendeiner Art und Weise persönliche Daten gespeichert sind. Dies können z. B. Name, Adresse, Kontakt- oder auch Bankdaten sein, die bei einem Onlinehändler gespeichert sind. Wird dieser Händler nun Opfer eines Angriffs, kann es sein, dass man selbst davon zunächst nichts bemerkt. Doch auch hier ist man nicht schutzlos. Nach Art. 34 DSGVO ist der Verantwortliche für die Datenverarbeitung, also z. B. ein gehackter Onlinehändler, dazu verpflichtet, die Betroffenen zu informieren, wenn ein hohes Risiko durch den Angriff entsteht. Entsteht kein hohes Risiko ist der Verantwortliche immerhin zur Meldung an die Datenschutzaufsicht verpflichtet, die den Datenschutzvorfall genauer prüft und den Verantwortlichen, wenn dieser eine Benachrichtigung zu Unrecht unterlassen hat, anweisen kann diese nachzuholen. Gerade bei Anbietern außerhalb der EU ist die DSGVO jedoch nicht immer anwendbar. Gleichzeitig kann auch die Durchsetzung schwierig sein. Eine sinnvolle Ergänzung sind daher Angebote, die

auf einschlägigen Handelsplätzen von Cyberkriminellen nach erbeuteten Login-Informationen suchen und diese dann in einem sicheren Verfahren für Betroffene zur Überprüfung bereitstellen. Zwei zu empfehlende Angebote in diesem Zusammenhang sind der HPI Identity Leak Checker des Hasso-Plattner-Instituts (↗ https://sec.hpi.de/ilc) und das Projekt »Have I Been Pwned« des IT-Sicherheitsexperten Troy Hunt (↗ https://haveibeenpwned.com/).

17 Zur generellen Bedrohungslage im Internet und in der IT

Über die Bedrohungslage und die Sicherheit von IT-Systemen und des Internets wird immer wieder diskutiert und durch eine umfangreichere Berichterstattung scheint es so zu sein, dass die Gefährdungslage im IT-Umfeld steigt. **Nehmen also Cyberangriffe bzw. Hacks in jüngster Zeit zu und wird das Internet bzw. die IT generell unsicherer?**

Grundsätzlich ist dazu festhalten, dass sich Cyberkriminelle und IT-Sicherheitsexperten ein fortwährendes Katz-und-Maus-Spiel liefern. Während auf der einen Seite fortlaufend neue Angriffe entwickelt und neue Sicherheitslücken ausgenutzt werden, werden auf der anderen Seite permanent Angriffe vereitelt und Software kontinuierlich verbessert. Dass es dabei zu Schwankungen kommen kann und mal die Angreifer, mal die Verteidiger die Nase vorn haben, liegt in der Natur der Sache. Ein weiterer Grund warum absolute Aussagen dazu schwierig sind, liegt darin begründet, dass längst nicht alle Angriffe publik werden. Dies kann beispielsweise daran liegen, dass Angriffe gar nicht bemerkt werden oder sogar bewusst verschwiegen werden. Schließlich ist IT-Sicherheit auch stark branchenabhängig. Gerade im Onlinebanking sind Angriffe in den vergangenen Jahren zurückgedrängt worden. Statistiken, wie das Lagebild Cybercrime des BKA, das für das Jahr 2018 mit 87.106 Fällen von Cybercrime einen Zuwachs von 1,3% in Vorjahr verzeichnet,[5] müssen daher stets im Kontext betrachtet werden.
So können steigende Fallzahlen beispielsweise schon auf ein geändertes Anzeigeverhalten zurückgeführt werden. Betrachtet man statt konkreter Fallzahlen jedoch Tendenzen, lässt sich feststellen, dass im Bereich der IT-Sicherheit erhebliche Fortschritte erzielt worden sind. So nimmt beispielsweise die Zahl der Webseiten, die mit Transportverschlüsselung (https) ausgestattet sind seit Jahren kontinuierlich zu und auch in vielen anderen Bereichen hat sich die technische Sicherheit in den vergangenen Jahren verbessert. Mit der steigenden technischen Sicherheit ergibt sich auch die Notwendigkeit stärkere Maßnahmen zur Abwehr von Angriffen auf den Sicherheitsfaktor Mensch, z. B. durch Social Engineering, zu ergreifen. Angreifer, die aufgrund der höheren technischen Sicherheit auf technologischem Wege nicht erfolgreich sind, verlegen sich nämlich verstärkt darauf menschliche Verhaltensweisen, z. B. durch Phishing, ausnutzen.

5 ↗ https://www.bka.de/DE/Presse/Listenseite_Pressemitteilungen/2019/Presse2019/191111_PMBLBCybercrime.html

18 Schutzmaßnahmen zur Vorbeugung von Unsicherheit und zum Erschweren von Angriffen

Was kann also jeder Nutzer selbst tun, um sich bestmöglich vor Angriffen und Bedrohungen zu schützen? Ein wesentlicher Aspekt der Informationssicherheit ist das Bewusstsein von Gefahren und vernünftiges Verhalten aller Beteiligten. Ein Großteil der erfolgreichen Angriffe auf Personen, Unternehmen und Institutionen zeigt immer wieder auf, dass organisatorische Maßnahmen und gesunder Menschenverstand die wichtigsten Waffen im Kampf gegen Angriffe sind.

Um ein Bewusstsein für die Gefahren und möglichen Einfallstore von Angriffen zu schaffen, ist zunächst die Reduktion des Unwissens über die eigenen Bestände von IT-Systemen ein Ziel. Hierzu müssen zunächst sämtliche Gegenstände und Produkte, die einer Risikobetrachtung zu unterziehen sind, in Listen (Inventaren) erfasst werden.

Hierzu werden Inventarlisten erstellt:

- IT-Device-Listen: Dazu gehören alle Gegenstände mit IT-Funktionen (auch Smart-Devices wie Babyphones, die Smart-Zahnbürste, die Voice-Teddybären, Fernseher, Router, Switches, etc.
- Software-Listen: Welche Software und Firmware ist auf den Devices installiert?

Dann müssen Risiken und Gefahren identifiziert werden. Hierzu ist es zunächst notwendig sich Gedanken über die Sicherheitsanforderungen zu machen und diese zu identifizieren. Im zweiten Schritt müssen mögliche Gefahren und Angriffsflächen analysiert und am besten ausformuliert werden, um die notwendigen Maßnahmen ableiten zu können. In der Folge werden in einer Risikoanalyse die Wahrscheinlichkeit von und die Maßnahmen im Fall von Angriffen definiert und Risikostrategien entwickelt. Dabei kann auch ein externes Audit durch Spezialisten helfen.

Wichtig sind Absicherungsthemen, wie Versicherung gegen Schäden, speziell auch eine Anpassung der eigenen Verträge der Haftpflichtversicherung. Die Vertragsgestaltung muss wohl überlegt werden mit Internet-Providern, Zulieferern, etc. und es sollte bei der Auswahl der Lieferanten vorab Kriterien festgelegt werden, um eine gezielte Auswahl zu treffen.

Zur organisatorischen Sicherheit gehören auch Rollen- und Rechtekonzepte für alle Anwender und Sicherheitsrichtlinien, die von allen Personen zu beachten sind. Hierzu müssen die Personen im Haushalt/der Firma auch gezielt geschult werden und ein Bewusstsein für die Sicherheit und damit verbundene Aufmerksamkeit bei typischen Angriffsszenarien geschaffen werden.

Die physikalische Sicherheit fängt mit Standards an, also Virenscannern, Firewall, Systeme zum Erkennen bzw. zur Vermeidung von Angriffen. Hierzu sollte man ausschließlich akkreditierte Werkzeuge verwenden. Die physikalische Vermeidung von Angriffen erfordert aber auch eine Härtung von IT-Systemen, in dem nicht benötigte Funktionen auf den Geräten deaktiviert und

auf den Geräten Sicherheitsmechanismen auch wirklich spezifisch konfiguriert werden. Prinzipiell sollten Log-Files aktiviert sein und regelmäßig, am besten täglich geprüft werden, hierfür gibt es auch spezielle Security-Monitoring-Werkzeuge.

Es sollte ein Schwachstellenanalyse-Zyklus etabliert werden, in dem die Geräte und die Software der Inventarlisten in möglichst kurzen Zyklen einem Security-Testing unterzogen werden. Hierzu sollten statische, dynamische und Fuzzing-Analysewerkzeuge verwendet werden.

Trotz aller Maßnahmen sind für Schadensereignisse auch Maßnahmen wie externe Backups, die nicht dauerhaft an den Geräten hängen, notwendig, da moderne Angriffe diese sonst mit in die Angriffe integrieren, um Backups wertlos zu machen. Wenn es die Möglichkeit gibt, so sollte neue Software zunächst in einer Sandbox installiert und der ein- und ausgehenden Traffic erst einige Tage mit Scannern analysiert werden, um zu entdecken, ob die Software nach außen Kontakte herstellen will, um Daten abzugreifen, oder Angreifern von innen die Türen zu den Systemen zu öffnen.

Die wichtigste Schutzmaßnahme ist aber immer noch die Anwendung des gesunden Menschenverstandes, um die Vertrauenswürdigkeit von Angeboten einzuschätzen und nicht in die typischen Fallen zu laufen. Ein gesundes Misstrauen gegenüber allen Angeboten ist immer geboten.

19 Vorgehensweisen und Empfehlungen nachdem ein Sicherheitsvorfall festgestellt wurde

Ist man Opfer eines Hackerangriffs geworden oder hat sich ein anderer Sicherheitsvorfall ereignet, stellt sich die Frage, **wie man sich verhalten sollte?** Auf jeden Fall ist es erforderlich schnell zu reagieren. Wie in anderen Notsituationen auch, sollte man beim Handeln aber ruhig und besonnen bleiben. So können beispielsweise weitere Schäden durch vorschnelle Reaktionen vermieden werden. Im Unternehmenskontext werden zu diesem Zweck üblicherweise Notfallpläne erstellt, die bei einem Vorfall routiniert abgearbeitet werden können. Teilweise führen Unternehmen hierzu auch Übungen und Trainings durch. Im Privatbereich ist dies eher unüblich, wobei es gleichwohl nicht schadet sich auch hier einige Gedanken zum Umgang mit akuten Sicherheitsvorfällen zu machen. Der Ablauf beim Incident-Response, wie die Bearbeitung von IT-Sicherheitsvorfällen in der Fachsprache heißt, ist immer ähnlich.

An erster Stelle bedarf es einer Erkennung des Angriffs. Dies kann auf ganz unterschiedlichem Wege stattfinden. In manchen Fällen werden Mitarbeiter auf Angriffe aufmerksam. In anderen Fällen sind es IT-Sicherheitsexperten oder Provider, die sich melden und auf einen Vorfall aufmerksam machen. Nicht selten fallen Sicherheitsvorfälle auch auf, weil sich Kunden oder sonstige Dritte bei einem Unternehmen melden und verdächtige Aktivitäten anzeigen. Um eine möglichst schnelle Erkennung von Angriffen zu erreichen, schulen viele Unternehmen ihre Mitarbeiter sogar in der Erkennung von Angriffen. In der Praxis besonders wichtig sind auch die für solche Fälle definierten Informationsketten. Die Erkennung eines Sicherheitsvorfalls nutzt nämlich wenig, wenn er zwar bemerkt wird, aber niemand Alarm schlägt oder der Alarm nicht die richtigen Stellen erreicht.

Erreicht die Meldung über einen Vorfall den Richtigen, in der Regel die IT-Abteilung oder IT-Sicherheitsabteilung, wird dort geprüft, ob der gemeldete Vorgang tatsächlich einen Sicherheitsvorfall darstellt und näher untersucht was vorgefallen ist. Anschließend werden mögliche Reaktionsmöglichkeiten erarbeitet und, oftmals unter Einbeziehung der Geschäftsleitung, entschieden, welches weitere Vorgehen sinnvoll erscheint. Dabei kann, sofern der Angriff noch andauert, nicht nur seine schnelle Beendigung, sondern auch die Sicherung von Beweisen eine Rolle spielen. Die ausgewählten Maßnahmen werden dann im Rahmen der Eindämmung des Sicherheitsvorfalles umgesetzt. In der Regel umfasst dies eine Begrenzung des verursachten Schadens, z. B. durch eine Abschaltung von Systemen, sowie eine Isolation betroffener Systeme. Letztere dient dazu einer Verbreitung des Sicherheitsvorfalles innerhalb des Unternehmensnetzwerks zu verhindern. Nach einer erfolgreichen Eindämmung ist es erforderlich, die Ursache oder den Auslöser des Vorfalls zu finden. Ist dies gelungen, können Systeme, die nicht betroffen sind aber im Rahmen der Eindämmung möglicherweise ausgeschaltet wurden, wieder gestartet werden. Dann beginnt die Wiederherstellung der betroffenen Systeme. Das bedeutet, es wird sichergestellt, dass keine weitere Bedrohung von ihnen ausgeht. Geschehen kann dies unter anderem durch das Entfernen einer installierten Schadsoftware, durch das Einspielen von Sicherheitsupdates oder auch durch einen Austausch von Systemkomponenten. In einer abschließenden

Phase geht es darum, Erkenntnisse, die aus dem Vorfall gewonnen werden können, so zu dokumentieren und zu analysieren, dass sie in der Zukunft für die Abwehr von anderen Sicherheitsvorfällen genutzt werden können. Auch bei der Aufarbeitung von Sicherheitsvorfällen gilt also »Nach dem Spiel ist vor Spiel!«.

Für die Erstellung eines Notfallplanes unterstützen kompetente Stellen, das BSI, der Datenschutzbeauftragte und auch die Cyber Security Alliance, siehe ↗ **https://www.allianz-fuer-cybersicherheit.de/**. Diese Stellen unterstützen auch, wenn ein Angriff bereits stattgefunden hat. Die Cyber Security Alliance nimmt die Sicherheitsvorfälle auch anonym auf. Sie kann durch sehr umfangreiches und kompetentes Wissen zielgerichtet unterstützen. Zum Beispiel können wichtige Informationen schnell zur Verfügung gestellt werden (z. B. welche Stellen bei welchen Cyber-Angriffen zu informieren sind) und der Schaden begrenzt werden.

Wesentliche Voraussetzung, um großen Schaden zu vermeiden, ist ein regelmäßiges Backup. Denn ohne Backup, können auch die kompetentesten Stellen nicht wirkungsvoll helfen. Deshalb entstand der Slogan: »Ohne Backup kein Mitleid!«.

Nicht vergessen werden darf auch, dass Sicherheitsvorfälle rechtliche Folgen haben können. So sieht etwa § 33 DSGVO eine Meldepflicht von Datenschutzverletzungen an die zuständige Datenschutzaufsicht vor. Bei besonders gravierenden Verletzungen des Datenschutzes kann es nach § 34 DSGVO darüber hinaus sogar erforderlich sein den Betroffenen, z. B. Kunden oder Mitarbeiter, zu informieren.

20 Die Verantwortung der Nutzer, um den Einsatz von Software sicherer zu machen

Nach eingehender Betrachtung der Themen Softwaresicherheit und Softwarequalität bleibt am Ende die Frage, **was kann jede und jeder selbst als Nutzer und Nutzerin eigentlich tun, um den Einsatz von Software sicherer zu machen?**

Die erste Entscheidung des Nutzers einer Software ist die Wahl der Software selbst. Sie sollte aus einer vertrauenswürdigen Quelle stammen und ihre Entwicklung sollte unter Einhaltung anerkannter Sicherheitsstandards erfolgen. Als Orientierung sind gültige Zertifikate zugelassener Prüfstellen, Selbstverpflichtungen des Herstellers bezüglich der Einhaltung solcher Standards oder auch Indikatoren seiner Vertrauenswürdigkeit wie Bewertungen anderer Nutzer oder Meldungen bzw. Artikel über das Unternehmen hilfreich (siehe auch Kapitel 14 »Zertifikate und Prüfstellen für sichere Software«). Neben der Softwareauswahl haben der Betreiber bzw. das Betriebsmodell wesentlichen Einfluss auf die Sicherheit. Hier erfordert eine optimale Auswahl Transparenz bezüglich der Standorte, an denen die eigenen Daten bei der Nutzung der Software verarbeitet bzw. gespeichert werden sollen. Wichtige Kriterien sind die an diesen Standorten geltenden gesetzlichen Regelungen, beispielsweise zum Datenschutz. Dabei sollten alle möglichen Standorte eines Betreibers in die Beurteilung einbezogen werden, auch solche, die nur als Ausweich- oder Notfall-Standorte dienen. Grundsätzlich ist dies eine Entscheidung, bei welcher der Schutzbedarf der zu verarbeitenden Daten mit dem persönlichen Vertrauen in die Sicherheit eines Betreibers bzw. der nationalen Regelungen des Betriebsstandortes abgewogen werden muss.

Neben der Entwicklung und dem Betrieb stellt die eigene IT-Infrastruktur, d. h. die technische Ausstattung des Nutzers, den nächsten Bereich dar, der entscheidenden Einfluss auf die Sicherheit hat. Alle selbst betriebenen Komponenten, beispielsweise Firewall, Betriebssystem, Browser, Add-ons und Hilfsprogramme, sollten dem aktuellen Stand entsprechen. Das erfordert mindestens, dass es sich um eine Version handelt, die vom Hersteller gepflegt wird, in der bekannt gewordene Fehler und damit auch Sicherheitslücken zeitnah korrigiert und in Form eines neuen Releases bzw. Patches an die Nutzer weitergegeben werden. Dabei ist durch die reine Verfügbarkeit einer neuen Version noch keine Sicherheitslücke geschlossen. Sie muss auch installiert werden. Dem Nutzer sollte bewusst sein, dass die Aktualisierung der einzelnen Komponenten seiner IT-Infrastruktur kein einmaliger Vorgang ist sondern ein fortwährender Prozess, dessen Auslöser jeweils die Erkennung eines neuen Fehlers, einer neuen Schwachstelle oder einer bisher unterschätzten Bedrohung und die daraus resultierende Bereitstellung eines Updates oder Patches durch den Hersteller ist (siehe auch Kapitel 2 »Die Notwendigkeit von Patches und Updates«).

Über die Aktualität aller an der Nutzung einer Software beteiligten eigenen technischen Komponenten des Nutzers hinaus ist auch deren Konfiguration entscheidend für die Sicherheit. In einigen Fällen wird der Nutzer für ihn wichtige Schutzfunktionen erst explizit aktivieren müssen,

damit sie wirksam sind. Hinzu kommen Berechtigungen einzelner Anwendungen oder Komponenten des Nutzers auf Funktionen oder Daten, mit denen man so sparsam wie möglich umgehen sollte. Orientierung gibt das Need-to-know-Prinzip, nach dem der Zugriff nur auf solche Daten erlaubt sein soll, die für die Erledigung der vorgesehenen Aufgabe unbedingt benötigt werden. Dieses Prinzip gilt für das Smartphone, wo der Nutzer entscheiden muss, welche Apps wirklich Zugriff auf seine Kontakte benötigen, ebenso wie für Geschäftsanwendungen, bei denen einzelne Benutzer mit Zugriffsrechten ausgestattet werden können, die deutlich über ihre betriebsbedingte Notwendigkeit hinausgehen.

Nicht zu unterschätzen ist ferner der Einfluss, den das Verhalten des Nutzers selbst auf die Sicherheit hat. Im Normalfall besteht sein System aus mehreren Bestandteilen, die an unterschiedlichen Orten laufen, z. B. Browser im Client-Rechner, Server-Applikation, Datenbank-Managementsystem, Rechenzentrums-Infrastruktur, Smartphone-App, Backend, Service Provider, Smart-Watch, Fitness-Tracker, Kraftfahrzeug usw. Diese Bestandteile des Gesamtsystems kommunizieren in der Regel über das Internet, ein öffentlich zugängliches Netzwerk, zu dem mehrere Milliarden Menschen Zugang haben. Aus diesem Grund sollte es die Regel sein, dass jeder Datentransfer verschlüsselt ist. Gerade wenn das in der heimischen, behüteten Infrastruktur betriebene System auf dem neusten Stand ist, im Sinne der Datensparsamkeit konfiguriert und regelmäßig auf Schadsoftware überprüft wird, ist es umso wichtiger, dass alles, was von außerhalb an Daten oder Mitteilungen an die eigene Software oder deren Nutzer adressiert wird einer zuverlässigen Überprüfung sowohl der Freiheit von Schadsoftware als auch der Authentizität unterzogen wird. Technisch unterstützen bei dieser Aufgabe Werkzeuge wie Firewalls oder Intrusion Prevention Systeme, die ebenso aktuell gehalten und vorteilhaft konfiguriert werden müssen wie die sonstige eigene Software.

Der Schutz der eigenen Software gegen Bedrohungen von außen bzw. durch Fremde ist die Regel und seine Notwendigkeit ist jedem einleuchtend. Weniger naheliegend ist jedoch der Schutz vor dem eigenen User, der eigenen Identität. Sind einem Fremden die eigenen Zugangsdaten bekannt, kann er sich in der entsprechenden Software, vielleicht dem Online Banking, als der Eigentümer dieser Zugangsdaten ausgeben und möglicherweise in seinem Namen eine Finanztransaktion durchführen. Nutzer besonders kritischer Software sollten heute darauf achten, dass die Überprüfung ihrer eigenen Authentizität nicht nur von einem Passwort abhängt, sondern zusätzlich auch den Besitz eines bestimmten, registrierten Smartphones, eines Zertifikats, eventuell auch das Wissen um persönliche Fakten erfordert oder gar biometrische Merkmale analysiert werden. Mit Einführung der Europäischen Zahlungsdienstleister-Richtlinie PSD II ist die Verwendung einer Zwei-Faktor-Authentifizierung inzwischen zur Pflicht für alle Anbieter elektronischer Zahlungsdienste geworden. Das Prinzip ist jedoch darüber hinaus in allen Bereichen empfehlenswert, in denen die Identität eines Nutzers zuverlässig festgestellt werden muss. Selbst beim einfachen E-Mail-Versand einer verschlüsselten Datei steigt die Sicherheit signifikant, wenn das Passwort nicht in der unmittelbar nachfolgenden Mail verschickt wird, sondern auf einem völlig anderen Medium, beispielsweise telefonisch oder per SMS, übermittelt wird.

Zusammenfassung – Q & A

Was ist denn eigentlich Software?

Unter dem Begriff Software versteht man strukturierte Anweisungen zur Lösung gestellter Aufgaben mittels Computersystemen, die von mehr als einem Entwickler erstellt und von mehr als einem Nutzer verwendet werden, während gleichzeitig die Weiterverwendung und Erweiterung durch Dritte problemlos möglich ist. Software stellt die notwendigen Informationen zur richtigen Zeit, am richtigen Ort, dem richtigen Nutzer/System, in der richtigen Form und mit akzeptablem Aufwand zur Verfügung.

1 Wie wird Software eigentlich hergestellt?

An der Entwicklung von Software sind in der Regel viele Personen beteiligt. Zunächst müssen Anforderungen eingeholt werden. Was soll die Software können? Welche Gesetze gelten, zum Beispiel für Datenschutz und IT-Sicherheit? Die Architektur der Software muss geplant werden. Danach können Aufgaben an einzelnen Mitarbeitenden, oder Teams weitergeleitet werden. Es gibt Spezialisten für verschiedene Aufgaben. Einige Entwickler sind zum Beispiel besonders gut in der Programmierung von Datenbanken. Andere Entwickler sind hingegen Spezialisten für die Entwicklung von Benutzeroberflächen. Immer wieder werden Qualitätskontrollen durchgeführt. Funktioniert der Code? Passen die verschiedenen Bausteine der Software zusammen? Bei großen Konzernen arbeiten zum Teil hunderte Entwickler an einem Produkt. Die Arbeitsabläufe müssen dabei natürlich genau aufeinander abgestimmt sein. Dafür braucht es viel Planung und klare Prozesse.

Nur in wenigen Fällen müssen Entwickler einen Code komplett neu schreiben. Oft können Sie auf Code-Bibliotheken zurückgreifen, oder sich an Vorlagen aus Open-Source-Software bedienen. Software enthält daher oft Versatzstücke, aus anderer Software und wird für den speziellen Gebrauch angepasst und weiterentwickelt. Vor dem Release wird Software idealerweise einem Stresstest unterzogen und mit Kunden unter realen Bedingungen getestet. Auch nach der Veröffentlichung muss die Software regelmäßig überarbeitet werden. Zum Beispiel, um neue Sicherheitslücken zu schließen, oder die Software auf spezifische Wünsche der Nutzer anzupassen. Der Lebenszyklus einer Software endet meist, wenn der Support für das jeweilige Produkt eingestellt wird.

2 Warum erfährt fast jedes softwarebasierte Produkt dauernd Patches und Updates?

Patches und Updates sind notwendig, um die Funktionen einer Software zu erweitern, sie an geänderte Rahmenbedingungen wie gesetzliche Anforderungen anzupassen oder auch Unzulänglichkeiten wie Sicherheitslücken zu beseitigen oder zu mindern. Aufgrund der zunehmenden Häufigkeit solcher Anlässe entsteht der Eindruck »dauernd« notwendiger Patches und Updates.

3 Warum müssen so häufig Software-Updates durchgeführt werden?

Software muss regelmäßig angepasst werden, um den Nutzern ein bestmögliches Produkt anzubieten. Daher braucht es regelmäßige Updates. In der Regel handelt es sich dabei um funktionale Updates. Diese werden zum Beispiel notwendig, wenn Service-Wartungen durchgeführt werden, oder wenn die Hersteller neue Funktionen der Software veröffentlichen. Gelegentlich werden auch Sicherheits-Updates durchgeführt, um auf neue Sicherheitslücken in der Software zu reagieren.

4 Warum ist Software denn nie fehlerfrei?

Moderne Methoden und Werkzeuge zur Softwareentwicklung reduzieren die möglichen Fehlerquellen. Solange Software jedoch von Menschen erstellt wird, lassen sich Fehler bei ihrer Erstellung nicht grundsätzlich ausschließen. Umfang und Komplexität moderner Software verhindern, dass solche Fehler vollständig durch analytische Verfahren, beispielsweise Tests, mit einem vertretbaren Aufwand gefunden und vor Nutzung der Software beseitigt werden können. Zu berücksichtigen ist bei dieser Fragestellung ebenfalls, dass die Bewertung eines beobachteten Verhaltens als Fehler bei verschiedenen Nutzern abweichen kann. Schließlich spielt auch der Faktor Zeit eine Rolle, denn bei der Änderung von Rahmenbedingungen (z. B. gesetzliche Vorgaben) kann ein gestern noch korrektes Verhalten heute ein Fehler sein.

In den meisten Bereichen ist »fehlerfreie« Software eine Illusion. Angestrebt werden sollte »fehlerarme« Software mit einem zuverlässig funktionierenden Prozess zur Fehlermeldung, Analyse, Korrektur und zeitnahen Bereitstellung der korrigierten Software.

5 Was machen Hersteller, um die Qualität ihrer Software in der Produktion sicher zu stellen?

Softwareentwickelnde Unternehmen erreichen eine möglichst umfassende Integration von Qualität in Form von Sicherheit im Entwicklungsprozess durch »Security by Default« und »Security by Design«. Dabei sind drei Aspekte essenziell: Erstens, die Integration von Sicherheit durch Tools (z. B. SAST, FAST uvm.). Zweitens, die Verankerung von Sicherheit als eine allgemeingültige Code-Kultur in den beteiligten Bereichen der Softwareentwicklung (das bedeutet auch: z. B. kein Blame Game zwischen den Bereichen, Fokus auf Kollaboration und eine inklusive Kultur, Aufbau von Sicherheitsbeauftragten und vieles mehr). Drittens, und statt die einzelnen Teams (Entwicklung, Betrieb, Sicherheit) »silofiziert« nebeneinander zu stellen, die Schaffung eines gesamtheitlichen cross-funktionalen Teams, welches die drei Komponenten Entwicklung, Betrieb und Sicherheit gemeinsam vorantreiben kann und einen offenen Umgang mit Wissen pflegt. Dies ist notwendig, da durch die rapide Beschleunigung des Entwicklungsprozesses durch DevOps und Agile Methoden die klassischen Security-Ansätze nicht mehr greifen. So dauert das Pentesting für eine App in der Regel 3–5 Tage, aber es gibt heute Deployment-Szenarien, die mehrfach am Tag stattfinden. Diesen Gap zu adressieren wird wesentlich

einfacher, wenn Security integraler Bestandteil des Gesamtprozesses und der Entwicklungs-Pipeline ist (z. B. im Rahmen von Shift-Left/DevSecOps-Initiativen in Form automatisierter CI/CD-Integration, feedback-basierter Fuzzing-Tests in Kombination mit statischer Code Analyse). Immer begleitet durch passgenaue kulturelle Change-Management-Maßnahmen.

6 Warum wird generell in Qualitätssicherung von Software investiert?

Je komplexer eine Software ist, desto wahrscheinlicher wird es, dass sie Fehler und Sicherheitslücken enthält. Viele Fehler und Sicherheitslücken können jedoch vermieden werden, wenn die Software vor ihrer Veröffentlichung auf Ihre Qualität geprüft wird. Wenn Fehler erst nach Veröffentlichung der Software erkannt werden, kann die Behebung eines Fehlers hohe Kosten verursachen. In einigen Fällen können Softwarefehler sogar Menschenleben gefährden. Wenn Fehler bereits in der Entwicklung auffallen, lassen sie sich hingegen meist schnell beheben. Für Unternehmen ist es daher wirtschaftlich, in die Qualität ihrer Software zu investieren. Deshalb versuchen Unternehmen in verschiedenen Phasen ihres Entwicklungsprozesses Maßnahmen zur Qualitätssicherung durchzuführen.

7 Spielt die Herkunft einer Software eine Rolle für die IT-Sicherheit?

Je komplexer eine Software ist, desto wahrscheinlicher wird es, dass sie Fehler und Sicherheitslücken enthält. Dies gilt für jede Software, unabhängig ihrer Herkunft. Ausschlaggebend für die Sicherheit ist, wie viel Aufwand in die Qualitätssicherung einer Software investiert wird. In Ländern und in Branchen, in denen die Anforderungen an die IT-Sicherheit besonders hoch sind, hat die Qualitätssicherung von Software daher eine besondere Priorität. Rechtsverstöße oder Vertragsverletzungen können für Unternehmen große finanzielle und rechtliche Konsequenzen haben. Insofern steigt für die Unternehmen auch das ökonomische Interesse, in die Qualität ihrer Software zu investieren. Es kann für Unternehmen jedoch auch sinnvoll sein, Teile ihrer Softwareentwicklung an andere Unternehmen auszulagern, die auf diese Dienstleistungen spezialisiert sind. In vielen Fällen können diese Dienstleister die Qualitätssicherung sehr viel routinierter und kostengünstiger erbringen, als das Unternehmen dies selbst gekonnt hätte.

8 Warum unterscheiden sich die Sicherheitsanforderungen für Software zwischen verschiedenen Branchen?

In Branchen, die stark reguliert sind, sind die Sicherheitsanforderungen für Software besonders hoch. Dies führt dazu, dass ein überdurchschnittlicher Anteil von Projektbudgets in die Qualitätssicherung von Software investiert wird. Dies trifft zum Beispiel auf den Bereich Finance und Aviation zu. Auch im Bereich der Medizintechnik, oder des Versicherungswesens gelten besondere Qualitätsanforderungen für bestimmte Softwareprodukte.

9 Wie wird die Erwartungshaltung an die Sicherheit von Software bei der Herstellung adressiert?

Die Herstellung der Software kann nicht allein zu Sicherheit führen, sondern es ist der gesamte Lebenszyklus der Software beziehungsweise die vollständige Handlungskette sowie der Nutzungskontext zu betrachten. Hinsichtlich der Herstellung ist die Transparenz der Lieferkette, beispielsweise, welcher Code zugekauft worden ist, relevant, um dann (schnelles) Handeln zu ermöglichen, wenn eine Sicherheitslücke in einer bestimmten Implementierung bekannt wird. Zertifikate nach Common Criteria sind kritisch zu betrachten, da ein Test auf Robustheit erst dann aussagekräftig ist, sofern dieser im realen Betrieb im eigenen Gesamtsystem durchgeführt wird. Generell ist Eigenverantwortung in der Beschaffung unter Betrachtung der eigenen Struktur, und ob die Lösung im Betrieb auch wirklich Sicherheit und die Schutzziele gewährleisten kann, unumgänglich.

10 Welche Rolle spielt der Staat, wenn es um die Sicherheit von Software geht?

Es ist heute eine Rarität geworden, wenn Entwickler, Betreiber und Nutzer einer Software im selben Land niedergelassen sind. Meist befinden sie sich nicht einmal im selben Staatenbund. Das schränkt die Wirksamkeit nationaler, staatlicher Regelungen erheblich ein.

Staaten haben die Rolle eines Regulierers, der beispielsweise Mindestanforderungen an die Sicherheit in Entwicklung, Betrieb und Nutzung von Software festlegt. Dies kann effektiv jedoch nur innerhalb eines Staatenbundes (z. B. EU-Verordnungen), besser noch global, funktionieren. Die Sicherheit von Software wird künftig in hohem Maße davon abhängen, wie es den regulierenden Staaten gelingt zusammenzuarbeiten, sich auf gemeinsame Anforderungen und Regelwerke zu einigen, diese wirksam umzusetzen, zu kontrollieren bzw. Verstöße zu sanktionieren sowie zeitnah auf Veränderungen zu reagieren (z. B. Entwicklungen im KI-Umfeld).

11 Wie haftet der Hersteller vertraglich für Softwarequalität?

Ungeachtet der Tatsache, dass Software nicht zu 100 Prozent fehlerfrei programmiert werden kann, muss der Softwareanbieter für Mängel der von ihm bereit gestellten Software einstehen. Rechtlich relevante Software-Mängel sind z. B. gegeben, wenn die Software vertraglich vereinbarte Funktionen nicht besitzt oder allgemein akzeptierte technische Qualitätsstandards unterschreitet. Die Ansprüche des Softwarenutzers wegen Mängeln der Software richten sich danach, welche Vertragsform der Softwareüberlassung jeweils zugrunde liegt. Denkbar sind z. B. Rücktritt vom Vertrag oder Schadensersatz. Der Softwareanbieter kann sich von dieser Verantwortung für Softwaremängel durch Vertragsgestaltung nicht komplett befreien.

12 Wer kann zur Verantwortung gezogen werden, wenn Softwarefehler zu Schäden führen?

Softwarefehler können auch außerhalb von Vertragsbeziehungen zu einer Haftung des Softwareherstellers nach den Grundsätzen der Produkt- und der Produzentenhaftung führen. Allerdings haftet der Hersteller in diesen Fällen nur, wenn ein Softwarefehler Schäden an Rechtsgütern verursacht hat, denen die Rechtsordnung einen besonderen Wert zuweist (z. B. Leben, Gesundheit, Eigentum). Um die Haftung zu vermeiden, muss der Hersteller die aus dem Softwarefehler resultierende Gefahr beseitigen. Allgemeingültige Grundsätze lassen sich hierfür aber kaum aufstellen.

13 Woran erkenne ich eigentlich »sichere« Software?

Eine sichere Software kann man nicht erkennen, da es keine 100%ige Sicherheit gibt und weil Software immateriell ist. Zertifikate können ein Indikator für qualitativ hochwertige Software sein. Es ist jedoch nicht auszuschließen, dass die Software trotz Zertifikaten nach wie vor Fehler und Sicherheitsmängel aufweist. Ein weiterer Indikator für qualitativ hochwertige Software ist, wenn Einträge in den einschlägigen Sicherheitsschwachstellendatenbanken fehlen: OWASP, CWE, NVD, CAPEC, CVE, VDBs. Um sicher zu gehen empfiehlt es sich die Software über vertrauenswürdige Lieferanten zu beziehen.

14 Welchen Zertifikaten und Prüfstellen kann man als Verbraucher vertrauen?

Im Einzelhandel gibt es verschiedene Zertifikate und Kennzeichnen, an denen sich Verbraucher orientieren können, ob ein Produkt sicher ist. Zum Beispiel die CE-Kennzeichnung. Ähnliche Zertifikate gibt es auch im Bereich der Softwareentwicklung.

Für den deutschen Markt erfolgt die Zertifizierung durch das Bundesamt für Sicherheit in der Informationstechnik (BSI) oder durch private Prüfstellen, die vom BSI akkreditiert sind (z. B. TÜV). Es gibt Zertifikate für zuverlässige Hersteller und Zertifikate für qualitativ hochwertige Produkte. In Deutschland lassen sich zum Beispiel viele Unternehmen zertifizieren, dass sie die Empfehlungen des IT-Grundschutz einhalten, die sich an den internationalen Standards ISO 27001 und ISO 27002 orientieren. Dies sagt aus, dass sich die Unternehmen in ihrer Organisation an bestimmte Sicherheitsauflagen halten, die sich zum Beispiel auf die IT-Infrastruktur beziehen. Für die Qualität von Softwareprodukten, ist hingegen ausschlaggebend, ob sie nach den Common Criteria (CC) entwickelt wurden. Für bestimmte Regionen und Branchen können zudem zusätzliche Kriterien gelten, zum Beispiel TCSEC für den amerikanischen Markt und ITSEC innerhalb der EU. Diese Zertifikate sagen aus, dass bei der Herstellung der Software gewisse Prozesse eingehalten wurden, die sich zum Beispiel auf Maßnahmen zur Qualitätssicherung beziehen. Über die Qualität des Codes sagen diese Zertifikate jedoch wenig aus. So kann eine Software trotz eines Zertifikates nach wie vor Qualitäts- und Sicherheitsmängel aufweisen.

15 Woran kann die Vertrauenswürdigkeit angebotener Dienste bzw. vertrauenswürdige Hersteller erkannt werden?

Zertifikate sind ein Indikator für vertrauenswürdige Hersteller und qualitativ hochwertige Produkte, da der Hersteller viel Aufwand in Qualitätssicherung und Sicherheit investiert.

Ein Service kann jedoch trotzdem weiterhin Fehler und Sicherheitslücken aufweisen. Hier sind Reaktionsgeschwindigkeit, Kommunikation und Umgang des Herstellers mit Kundenanfragen und Sicherheitsvorfällen wichtige Indikatoren.

Nicht alle Hersteller leisten sich die Kosten eines Zertifizierungsverfahrens. Wenn sie sich aber bei der Entwicklung an den internationalen Standards orientieren, dann sollte man sich darüber informieren können und daher sind Reputation und Kundenzufriedenheit eines Unternehmens neben Zertifikaten gute Indikatoren, um die Vertrauenswürdigkeit eines Herstellers zu bewerten. Internetrecherchen zeigen relativ schnell über die Trefferlisten, wie Ernst ein Serviceanbieter oder Hersteller den Datenschutz und die Sicherheit des Kunden nimmt.

16 Wie erkenne ich eigentlich, dass ich Opfer eines Angriffs geworden bin?

Zu erkennen, dass man Opfer eines Cyberangriffes geworden ist, kann schwierig sein. Zunächst kommt es darauf an, ob man selbst angegriffen wurde oder Daten bei einem Dritten erbeutet wurden. Ist man selbst, etwa durch eine Schadsoftware auf dem eigenen Rechner, Opfer eines Angriffs geworden, können technische Analysen helfen einen Angriff aufzudecken. Kommen eigene Daten bei Dritten abhanden, bestehen bei schweren Angriffen Benachrichtigungspflichten nach dem Datenschutzrecht. Darüber hinaus können auch Dienste, die bei Hacks gestohlene Daten sammeln und den Betroffenen eine Überprüfung ermöglichen, helfen.

17 Nehmen die Cyberangriffe bzw. Hacks in jüngster Zeit zu? Wird das Internet bzw. IT generell zunehmend unsicherer?

Eine Aussage zur Sicherheit des Internets und zum aktuellen Stand der Bedrohungen ist schwierig und von vielen Faktoren abhängig. Grundsätzlich befinden sich IT-Sicherheitsexperten in einem ständigen Wettstreit mit den kriminellen Tätern hinter den Angriffen. Das Lagebild Cybercrime des BKA stellt für das Jahr 2018 mit 87.106 Fällen von Cybercrime einen Zuwachs von 1,3% fest. Dies kann aber auch an einem geänderten Anzeigeverhalten liegen. Insgesamt kann – losgelöst von den Fallzahlen – festgehalten werden, dass bei der IT-Sicherheit in den vergangenen Jahren große Fortschritte gemacht wurden. Nicht zuletzt deshalb versuchen die Täter zunehmend statt der Technik den Menschen hinter dem Computer anzugreifen.

18 Was kann ich tun, um mich bestmöglich vor Angriffen zu schützen?

Mit gesundem Misstrauen und gesundem Menschenverstand die Vertrauenswürdigkeit von Angeboten zu bewerten, ist immer der richtige Ansatz. Wichtig ist, sich Angriffsszenarien und Gefahren bewusst zu machen und dann gezielt Maßnahmen zu überlegen. Daten sollten immer noch intelligent in Backups und nicht mehr überschreibbaren Archiven gesichert werden, falls doch etwas passiert. Man sollte sich immer bewusst machen, wo überall IT und Software installiert ist und bei allen Geräten dann die notwendigen Absicherungen ausschöpfen. Netze sollten abgesichert und Systeme gehärtet werden. IT-Sicherheit erfordert heute nicht nur in Unternehmen das Monitoring der Sicherheit der Geräte und dies konstant und durchgehend.

19 Wie sollte ich mich verhalten, wenn tatsächlich einmal ein Sicherheitsvorfall stattgefunden hat?

Zunächst sollte man prüfen, welches Ausmaß der Angriff hat und in welcher Phase er sich befindet. Dauert der Angriff noch an, sollte man versuchen den Schaden schnellstmöglich zu minimieren und den Angriff zu unterbrechen. Wenn der Angriff beendet ist, kann die Aufarbeitung beginnen. Aus dieser können, etwa wenn ein Anbieter Schutzmaßnahmen vernachlässigt hat, auch Schadensersatzansprüche entstehen. Darüber hinaus sind Cyberangriffe in der Regel strafbar, sodass auch die Möglichkeit einer Strafanzeige besteht. Wichtiger noch ist aber, dass zukünftige Angriffe verhindert werden. Sind bei dem Angriff Zugangsdaten entwendet worden, sollten diese beispielsweise dringend geändert werden. Gerade Unternehmen sollten sich bei der Aufarbeitung von Angriffen durch Experten beraten lassen. Neben technischen Fragen sind hier auch rechtliche Aspekte, etwa eine Pflicht zur Meldung an die Datenschutzaufsicht von Bedeutung.

20 Was kann ich selbst als Nutzer eigentlich tun, um den Einsatz von Software sicherer zu machen?

Eine elementare Voraussetzung für den sicheren Einsatz ist, dass die gesamte Technik dem aktuellen Stand der Entwicklung entspricht. Jedes nicht installierte Update stellt ein Risiko dar, wenn dadurch eine bekannte Sicherheitslücke offen bleibt. Dabei ist die Aktualisierung der einzelnen Komponenten kein einmaliger Vorgang sondern ein fortwährender Prozess, dessen Auslöser die Erkennung neuer Schwachstellen und die resultierende Bereitstellung neuer Updates ist.

Neben der Aktualität aller am Softwareeinsatz beteiligten Komponenten ist deren Konfiguration von Bedeutung. Häufig kann das Risiko hinsichtlich missbräuchlicher Nutzung oder unautorisierter Zugriffe durch eine angemessene und an die eigene Arbeitsweise angepasste Konfiguration gemindert werden.

Nicht unterschätzt werden darf außerdem das Bewusstsein der Nutzer, dass es sich beim Internet um einen öffentlichen Raum handelt, zu dem mehrere Milliarden Menschen Zugang haben. Alle Daten, die zwischen den Komponenten eines Systems transportiert oder an andere Systeme bzw. Nutzer übermittelt werden und dabei diesen öffentlichen Raum »durchqueren«, müssen verschlüsselt sein. Alles, was von außerhalb eines Systems an Daten oder Mitteilungen an die eigene Software oder deren Nutzer adressiert wird muss einer zuverlässigen Überprüfung sowohl der Freiheit von Schadsoftware als auch der Authentizität unterzogen werden.

Die Prüfung der Authentizität wird sicherer, wenn sie neben einem Passwort auch den Besitz bestimmter Endgeräte oder Zertifikate, das Wissen um persönliche Fakten oder biometrische Merkmale einbezieht. Auch hilft die Aufteilung auf verschiedene Kommunikationskanäle, beispielsweise der SMS-Versand eines Passworts zum Öffnen der per E-Mail verschickten Datei, oder der Telefonanruf beim Bankberater, um die Korrektheit einer verdächtigen E-Mail zu überprüfen.

Bitkom vertritt mehr als 2.700 Unternehmen der digitalen Wirtschaft, davon gut 1.900 Direktmitglieder. Sie erzielen allein mit IT- und Telekommunikationsleistungen jährlich Umsätze von 190 Milliarden Euro, darunter Exporte in Höhe von 50 Milliarden Euro. Die Bitkom-Mitglieder beschäftigen in Deutschland mehr als 2 Millionen Mitarbeiterinnen und Mitarbeiter. Zu den Mitgliedern zählen mehr als 1.000 Mittelständler, über 500 Startups und nahezu alle Global Player. Sie bieten Software, IT-Services, Telekommunikations- oder Internetdienste an, stellen Geräte und Bauteile her, sind im Bereich der digitalen Medien tätig oder in anderer Weise Teil der digitalen Wirtschaft. 80 Prozent der Unternehmen haben ihren Hauptsitz in Deutschland, jeweils 8 Prozent kommen aus Europa und den USA, 4 Prozent aus anderen Regionen. Bitkom fördert und treibt die digitale Transformation der deutschen Wirtschaft und setzt sich für eine breite gesellschaftliche Teilhabe an den digitalen Entwicklungen ein. Ziel ist es, Deutschland zu einem weltweit führenden Digitalstandort zu machen.

Bundesverband Informationswirtschaft,
Telekommunikation und neue Medien e.V.

Albrechtstraße 10
10117 Berlin
T 030 27576-0
F 030 27576-400
bitkom@bitkom.org
www.bitkom.org

bitkom