Bitkom e.V. |

Warum die Sicherheit in Software so entscheidend ist

Berlin, 8. November 2020 - Die Digitalisierungsprozesse in Wirtschaft und Gesellschaft führen dazu, dass Software und softwarebasierte Produkte zunehmend allgengewärtig sind. Neben den damit verbundenen Möglichkeiten und Chancen gilt es aber auch Gefahren und Bedrohungsszenarien zu berücksichtigen. Darauf weist der Digitalverband Bitkom in seinem neuen Leitfaden „Zur Sicherheit softwarebasierter Produkte“ hin. Über grundlegende Fragen beleuchtet der Leitfaden ausführlich das Thema Sicherheit in der Herstellung und Verwendung von Software. „Gut funktionierende, sichere Software ist die entscheidende Komponente für die künftige Wertschöpfung in Unternehmen“, sagt Bitkom-Experte Dr. Frank Termer. Umso wichtiger sei es, ein allgemeines Verständnis für Softwaresicherheit zu etablieren, das über die Fach-Community hinausgehe. „Dazu soll der Leitfaden einen Beitrag leisten“, so Termer.

Vor allem die folgenden sieben Leitfragen gilt es aus Bitkom-Sicht allgemeinverständlich zu beantworten:

1. Wie wird Software hergestellt?

An der Entwicklung von Software sind in der Regel viele Personen beteiligt. Zunächst müssen Anforderungen an eine digitale Lösung oder ein zu entwickelndes IT-System eingeholt werden. Anschließend wird die Architektur der Software geplant. Aber nur in wenigen Fällen müssen Entwickler einen Code komplett neu schreiben. Oft können sie auf Code-Bibliotheken zurückgreifen oder sich an Open-Source-Software bedienen. Software enthält daher oft Versatzstücke aus anderer Software und wird für den speziellen Gebrauch angepasst und weiterentwickelt. Vor der Veröffentlichung wird Software idealerweise einem Stresstest unterzogen und mit Kunden unter realen Bedingungen getestet. Auch nach der Veröffentlichung muss die Software regelmäßig überarbeitet werden.

2. 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.

3. Warum ist Software nie ganz fehlerfrei?

Moderne Methoden und Werkzeuge zur Softwareentwicklung reduzieren die möglichen Fehlerquellen. Solange Software jedoch von Menschen erstellt wird, lassen sich Fehler 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.

4. Wie gewährleisten Hersteller trotzdem eine hohe Software-Qualität?

Softwareentwickelnde Unternehmen erreichen eine möglichst umfassende Qualität im Entwicklungsprozess durch „Security by Default“ und „Security by Design.“ Dabei sind drei Aspekte essenziell: Erstens, entsprechende Sicherheitswerkzeuge sollten in die eigentlichen Software-Entwicklung integrieret werden. Zweitens, Sicherheit sollte als eine allgemeingültige Code-Kultur in den beteiligten Bereichen der Softwareentwicklung verankert werden. Drittens, die Teamorganisation. Statt einzelne Teams in Silos nebeneinander zu stellen, sollte ein gesamtheitliches cross-funktionales Team geschaffen werden, das die drei Komponenten Entwicklung, Betrieb und Sicherheit gemeinsam vorantreiben kann und einen offenen Umgang mit Wissen pflegt.

5. Was passiert, wenn Software-Fehler zu Schäden führen?

Wenn Software-Fehler auftreten, können Hersteller auch bei fehlenden Vertragsbeziehungen nach den Grundsätzen der Produkt- und der Produzentenhaftung haftbar gemacht werden. Allerdings haftet der Hersteller in diesen Fällen nur, wenn ein Software-Fehler Schäden an jenen Rechtsgütern verursacht hat, die laut Rechtsordnung einen besonderen Wert aufweisen, etwa Gesundheit oder Eigentum. Um die Haftung zu vermeiden, muss der Hersteller die aus dem Software-Fehler resultierende Gefahr beseitigen. Allgemeingültige Grundsätze lassen sich hierfür aber kaum aufstellen.

6. Woran können Nutzer „sichere“ Software erkennen?

Ein eindeutiger Beweis für sichere Software lässt sich nie liefern. Anerkannte Zertifikate können jedoch ein Indikator für qualitativ hochwertige Software sein. Es ist aber nicht auszuschließen, dass Software trotz Zertifikaten nach wie vor Fehler und Sicherheitsmängel aufweist. Ein weiterer Indikator für qualitativ hochwertige Software ist, wenn in den einschlägigen Datenbanken für Sicherheitsschwachstellen keine Einträge zur jeweiligen Software vorhanden sind. Dazu zählen die Datenbanken OWASP, CWE, NVD, CAPEC, CVE, VDBs. Zudem sollte Software immer über vertrauenswürdige Lieferanten eingekauft werden.

7. Was können Nutzer 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. Unerlaubte Zugriffe oder Missbrauch können besser vermieden werden, wenn Nutzer ihre Programme an die eigene Arbeitsweise anpassen.

Der vollständige Bitkom-Leitfaden „Zur Sicherheit softwarebasierter Produkte“ ist zum kostenlosen Download verfügbar: https://www.bitkom.org/Bitkom/Publikationen/Bitkom-Leitfaden-zur-Sicherheit-softwarebasierter-Produkte

Kontakt

Andreas Streim
Pressesprecher
Telefon: +49 30 27576-112
E-Mail: a.streim@bitkom.org
Download Pressefoto

Felix Ansmann
Referent - Software & IT-Services
Nachricht senden

Felix Kuhlenkamp
Leiter Sicherheit
Download Pressefoto
Nachricht senden

Link zur Presseinformation auf der Webseite:
https://www.bitkom.org/Presse/Presseinformation/Warum-die-Sicherheit-in-Software-so-entscheidend-ist