Postvertriebsstück Entgelt bezahlt D 81 50

07/2017

www.dud.de

DuD

Datenschutz und Datensicherheit

Recht und Sicherheit in Informationsverarbeitung und Kommunikation

Springer Gabler

Eberhard von Faber, Walter Sedlacek

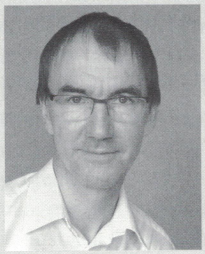
Spieltheorie im Dienst der IT-Sicherheit im Internet-der-Dinge

Mindesthaltbarkeit oder: Was passiert, wenn sich niemand kümmert?

Die Spieltheorie ist ein Zweig der Mathematik und der Ökonomie. Sie versucht, rationale und daher vermutlich real zu erwartende Entscheidungen vorherzusagen bzw. die Gründe dafür zu verstehen, warum eine bestimmte Alternative einer anderen vorgezogen wird. Entscheidungen von Marktteilnehmern beeinflussen auch ganz wesentlich die erreichte Informationssicherheit, geht es doch immer darum, ob sich jemand um die Sicherheit kümmert und ob Marktteilnehmer dies einfordern. Im Internet-der-Dinge scheint vieles im Argen zu liegen. Schwachstellen in Kameras und andere Geräte werden ausgenutzt, um sie zur Waffe umzufunktionieren. Kann die Spieltheorie dem etwas entgegensetzen?

Einleitung

Computersysteme und andere elektronische, mit IT ausgestattete Geräte oder Systeme besitzen oft Schwachstellen (Sicherheitslücken), die von Angreifern ausgenutzt werden können. Eine gängige Methode, solche Schwachstellen zu beseitigen, besteht in der Aktualisierung der Software, meist als Patchen bezeichnet. Insbesondere bei Alltagsgegenständen, die Software enthalten und mit dem Internet verbunden sind, **besteht die Gefahr, dass nicht erkannt wird, ob die Aktualisierung der Software erstens vorgesehen ist und zweitens durchgeführt wird. Drittens kommt erschwerend hinzu**, dass Geräte- und Softwarehersteller die Software Dritter verwenden, für die sie sich eventuell nicht verantwortlich fühlen. Dieser Artikel beschäftigt sich ausschließlich mit diesem Problem und skizziert eine mögliche Lösung dafür. Ergänzend wird beschrieben, wie die Aktualisierung der Software in anderen Computersystemen bzw. in industriell betriebenen Rechenzentren sichergestellt wird. Die hier vorgestellte Lösung ist als Idee und Denkanstoß zu verstehen. Vor- und Nachteile der Umsetzung werden diskutiert. **Für spieltheoretische Arbeiten wurde bislang achtmal der Nobelpreis für Wirtschaftswissenschaften vergeben. Das zeigt die Bedeutung dieser manchmal unterschätzten Disziplin, die in diesem Beitrag exemplarisch auf die IT-Sicherheit angewendet wird.**



Prof. Dr. Eberhard von Faber

T-Systems, Chief Security Advisor, IT Division; Arbeitsgebiete: Sicherheitsarchitektur, Entwickler von ESARIS, sichere IT-Produktion, sicheres IT-Outsourcing, Prozess- und ITIL-Integration, Standardisierung, Cloud, IAM

E-Mail: E.von-Faber@t-systems.com

Walter Sedlacek

Mag., MSc MBA PMP; T-Systems; versch. Rollen im intern. Management und Projektmanagement; Leitung der intern. Einführung von „ESARIS"; Leitung des regionalen Rechenzentrums in Singapur

E-Mail: Walter.Sedlacek@t-systems.com

1 Hintergrund und Problembeschreibung

1.1 Gegenstand

Das moderne Internet vereint zwei Eigenschaften bzw. Trends: die dezentrale, verteilte Nutzung und Erfassung von Informationen einerseits und die zentrale Bereitstellung von IT-Diensten und das zentralisierte Computing andererseits. Beide Bereiche existieren nicht getrennt voneinander. Vielmehr nutzen und verarbeiten zentrale Anwendungen mehr und mehr die von dezentral verteilten Komponenten und Geräten erzeugten Daten; und sie stellen diesen Komponenten und Geräten ihrerseits Daten zur Verfügung. Sensoren, die im Internet verteilt sind, erfassen Informationen, die zentral verarbeitet werden. Aktoren erhalten ihre

Steuersignale von zentralen IT-Anwendungen. Seit einigen Jahren nimmt die Zahl elektronischer, mit IT ausgestatteter Geräte, die als Sensoren und Aktoren agieren und Informationen auch selbst verarbeiten, massiv zu. Da solche Geräte in Alltagsgegenständen und Industrieanlagen verbaut sind oder solche darstellen, spricht man auch vom Internet of Things (IoT).

Wir bezeichnen die dezentral verteilten Komponenten und Geräte vereinfacht als Geräte im Internet der Dinge (IoT) bzw. kurz IoT-Geräte. (Dass dies eine Vereinfachung ist, weil z.B. PCs in privater Hand eigentlich keine IoT-Geräte sind, ist für die folgende Diskussion unerheblich.) Demgegenüber stehen die zentralen Komponenten und Geräte.

Allerdings besteht der Unterschied dieses Gerätetypen nicht wirklich darin, wo sie sich befinden. Er besteht darin, ob sie sich eindeutig im Besitz und in der Obhut eines IT-Dienstleisters befinden, weil sie z.B. in dessen Rechenzentrum installiert sind, oder ob Besitz und Obhut nicht so klar geregelt bzw. erkennbar sind. Warum ist die Unterscheidung wichtig? Man kann davon ausgehen, dass ein IT-Dienstleister die Aktualisierung der Software seiner Systeme als seine Aufgabe begreift, weil er selbst direkt oder indirekt betroffen ist, wenn mögliche Schwachstellen nicht beseitigt werden. (Dieser Fall wird in Kapitel 2 näher beleuchtet.) Auf der anderen Seite gibt es aber IoT-Geräte, bei denen für den Anwender oder Betreiber nicht ersichtlich ist, ob die Aktualisierung der Software vorgesehen ist und durchgeführt wird. (Eine mögliche Lösung wird in Kapitel 3 skizziert.) Doch zunächst soll das Problem noch eingehender erläutert und anhand von Beispielen verdeutlicht werden.

1.2 Problembeschreibung anhand von Beispielen

Betrachten wir ein paar Beispiele solcher IoT-Geräte, um das Problem besser zu verstehen. Erstens: Um IT-Dienste bzw. Server direkt anzugreifen oder aber z.B. Schadsoftware in Umlauf zu bringen, bauen Angreifer sogenannte Botnetze auf. Sie bestehen aus einer Vielzahl gekaperter Computersysteme, die ohne das Einverständnis des Besitzers ferngesteuert werden und zweckentfremdet vom Betreiber des Botnetzes benutzt werden. Schon seit langem werden unzureichend geschützte PCs übernommen und Teil solcher Botnetze. Seit einiger Zeit gibt es eine neue Qualität durch das Internet-der-Dinge.

Erstens: 2016 wurde berichtet, dass Botnetze wie „Mirai" billige Internet-Kameras zu hunderttausenden umfunktionierte, um diese für Angriffe zu nutzen. Dies wurde durch Schwachstellen in der Kamerasoftware möglich. *Zweitens:* Es ist nicht lange her, dass ein unbekannter IT-Sicherheitsspezialist mit dem Pseudonym „Kenzo2017" eine Warnung veröffentlichte, dass bestimmte Router (sie sind im Wortsinn keine IoT-Geräte[1]), mit denen Haushalte und Firmen sich mit dem Internet verbinden, anfällig dafür sind, diese fernzusteuern und für einen Angriff zu verwenden. Der Hersteller veröffentlichte eine aktualisierte Software; sonst geschah nichts. *Drittens:* Industrieanlagen wie z.B. Windkraftanlagen, Wasserwerke oder andere Maschinen von Kleinunternehmen übermitteln Mess- und Diagnosedaten über das Internet an zentrale Anwendungen in Rechenzentren und erhalten Steuersignale auf dem gleichen Wege. Für den Austausch dieser Daten werden in der Industrieanlage Standardkomponenten verbaut. Der Besitzer und Betreiber der Industrieanlage, aber auch der von Videorecordern und Fernsehern ist sich oft gar nicht bewusst, diese IoT-Geräte zu betreiben und entsprechend der Sicherheitsvorschriften instand halten zu müssen. Sie sind Teil einer Funktion der Anlage, die einem branchen- und geschäftsspezifischen Zweck dient, für den sich der Besitzer und Betreiber der Industrieanlage primär verantwortlich fühlt. Der (bekannte) Hersteller verbaut ein Teil eines anderen (vielleicht sogar unbekannten) Herstellers, für dessen Pflege in Bezug auf die IT-Sicherheit sich am Ende niemand verantwortlich fühlt und kümmert.

Allen drei Beispielen ist gemeinsam, dass sich der Käufer, Besitzer und Betreiber des IoT-Gerätes oft nicht über die Bedeutung der IT-Sicherheit im Klaren ist. Warum? Käufer, Besitzer bzw. Betreiber werden oft nicht so informiert, dass sie unterschiedliche Sicherheitsniveaus sofort erkennen können und eventuelle Implikationen wirklich ausreichend bewusst werden. Es gibt kein Logo und kein Label, dass es gestattet, Geräte und Systeme mit besserer Sicherheit von denen mit schlechterer IT-Sicherheit zu unterscheiden. Selbst Fachleute haben hier oft Schwierigkeiten.

Im zweiten Beispiel kommt es darauf an, das bereitgestellte Patches (Softwareaktualisierungen) auch wirklich eingespielt werden. Dazu ist es zunächst erforderlich erkennen zu können, wenn die notwendige Softwareaktualisierung ausbleibt.

Das dritte Beispiel zeigt, dass es Konsequenzen haben muss, wenn z.B. verborgene IoT-Geräte nicht aktualisiert werden. Nur dann würde der Hersteller der Anlage den Besitzer und Betreiber darauf hinweisen müssen. Und nur dann kann erwartet werden, dass der Besitzer und Betreiber verlangen würde, die Aktualisierung der Software zum Bestandteil des Vertrages zu machen, um sie durchzusetzen.

Die in diesem Artikel skizzierte Lösung für eine bestimmte Klasse von IoT-Geräten wird sich an diesen drei Beobachtungen orientieren. Allerdings soll schon jetzt angemerkt werden, dass die Beseitigung von Schwachstellen durch die Aktualisierung der Software kein Allheilmittel ist: Sie ist kein Garant für eine ausreichende Absicherung von Systemen, aber eine wichtige Voraussetzung dafür, da IT-Systeme in der Regel nicht perfekt abgesichert, also frei von Schwachstellen sind.

2 Ansatz im Bereich zentrales Computing

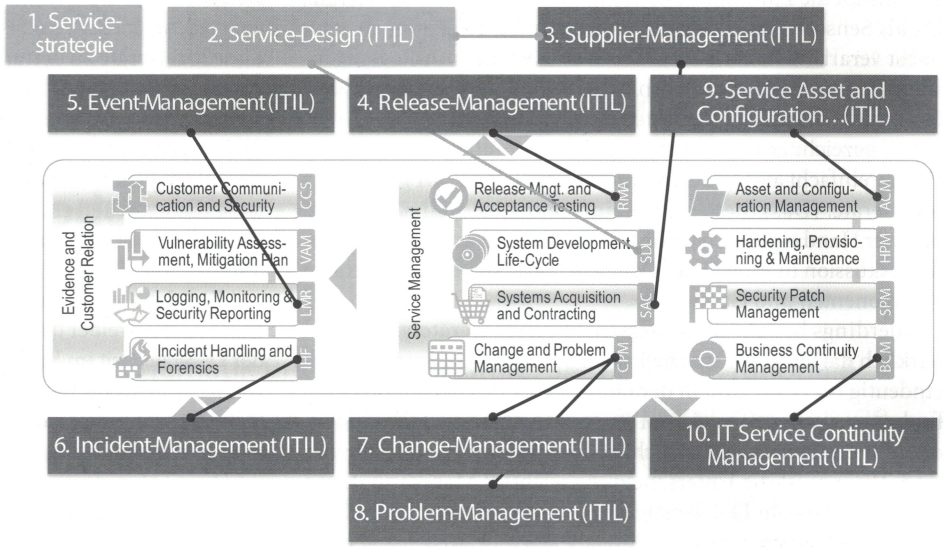
Bevor eine Lösung für IoT-Geräte entwickelt wird, sollte man einen Blick auf „traditionelle" IT-Systeme werfen und untersuchen, wie mit den entsprechenden Herausforderungen dort umgegangen wird. Genauer gesagt wird in diesem Kapitel erläutert werden, wie dies in einer großtechnischen, industrialisierten IT-Produktion zu erfolgen hat, da diese die größte Reife zu haben verspricht. Leser, die nicht an diesem Ausflug interessiert sind, können das Kapitel auch überspringen.

Zunächst definieren wir, was unter einer großtechnischen, industrialisierten IT-Produktion zu verstehen ist. Auf eine Rechenzentrumsfläche eines typischen, großen Fußballfeldes passen vielleicht etwa 2800 Racks mit insgesamt knapp 40.000 physischen Servern (Computersystemen). Klima, Stromversorgung usw. sind nicht eingerechnet. Eine solche Anzahl von Systemen erfordert große Rechenzentren. Die IT-Systeme werden von einem IT-Dienstleister betrieben, der seinen Kunden damit IT-Services zur Verfügung stellt. Insbesondere wenn große Firmen

1 Wir verwenden Router in diesem Beispiel, weil sie allgemein bekannte und verwendete Geräte im Internet sind, an denen sich Problem und Lösung sehr gut studieren lassen, auch wenn die Definition für IoT-Geräte nicht auf sie zutrifft.

zu seinen Kunden zählen, wird die IT eine besondere Komplexität besitzen, die über ihre schiere Größe hinausgeht. Große Firmen haben sehr spezielle Geschäftsprozesse. Die Unterstützung solcher Geschäftsprozesse mit moderner IT zieht spezielle Anforderungen und Lösungen nach sich, die die Komplexität der IT und TC des IT-Dienstleisters erhöht. Um Qualität sicherzustellen und Kosten zu beherrschen, ist die Bereitstellung der IT-Services prozessorientiert und sehr arbeitsteilig organisiert: Ähnlich wie ein Auto heutzutage nicht mehr von einem Team von Spezialisten hergestellt wird, sondern von vielen, die entlang eines Fließbands nur bestimmte, einfache Tätigkeiten ausführen, wird auch IT heute industriell produziert.

Abbildung 1 | Ausschnitt aus der ESARIS Security Taxonomy (12 von 31 Bereiche)



ESARIS und die *ESARIS Security Taxonomy* [9] helfen dabei, die IT-Sicherheit in einer solchen prozessorientierten und sehr arbeitsteiligen IT-Produktion zu beherrschen. Dabei werden bekannte Maßnahmen integriert. Es geht also weniger darum, Maßnahmen wie Zugriffsschutz, Verschlüsselung, Überwachung usw. zusammenzustellen. (Dafür gibt es genügend andere Quellen wie [6] und [7].) Vielmehr besteht die Herausforderung darin, eine Methode zu besitzen, die sicherstellt, dass hunderte und tausende solcher Sicherheitsmaßnahmen definiert, kommuniziert und korrekt angewendet werden in einer großtechnischen, industriellen Produktionsumgebung mit tausenden und manchmal zehntausenden von hochspezialisierten Angestellten in vielen Ländern rund um den Globus [9].

Da die IT-Produktionsumgebung prozessorientiert organisiert ist, bezieht sich die *ESARIS Security Taxonomy* etwa zur Hälfte auf Aktivitäten bei der Entwicklung, Implementierung und dem Betrieb von IT-Diensten einschließlich ihrer Pflege bzw. Instandhaltung und Weiterentwicklung. Da die IT-Produktionsumgebung eine Vielzahl von Technologien verwendet, bezieht sich die andere Hälfte auf typische Technologiebereiche und die damit verbundene Arbeitsteilung innerhalb des IT-Dienstleisters und mit all seinen Partnern und Zulieferern. An dieser Stelle interessieren nur die Aktivitäten bei der Entwicklung, Implementierung und dem Betrieb von IT-Diensten, da es ja um die Durchsetzung von IT-Sicherheit im weitesten Sinne geht. Ihre Darstellung orientiert sich ganz wesentlich an etablierten Verfahren, wie sie in ITIL[2] definiert werden. Allerdings berücksichtigen weder ITIL noch die entsprechende internationale Norm ISO/IEC 20000 und andere derartige Standards die IT-Sicherheit ausreichend. Sicherheitsstandards wie die der ISO/IEC 27000er Serie berücksichtigen wiederum die Spezifika einer großtechnischen IT-Produktion nicht oder nur sehr unzureichend. Dies sind Gründe [10] für die Erweiterung bestehender Best-Practices durch den Verein *Zero Outage Industry Standard* auf der Basis von ESARIS [11].

Abbildung 1 zeigt einen Ausschnitt aus der ESARIS Security Taxonomy mit Bezügen zu Prozessen (aus ITIL), wie sie in einer großtechnischen IT-Produktion etabliert sind. Es fällt auf, dass vier Bereiche aus der ESARIS Security Taxonomy keine Entsprechung in ITIL haben. Das unterstreicht, warum eine direkte Übernahme der ITIL-Prozesse nicht nur zu unübersichtlich, sondern eben auch nicht ausreichend gewesen wäre. Drei dieser hinzugekommenen Bereiche berühren das Thema dieses Artikels direkt: *Vulnerability Management and Mitigation Planning (VAM)*, *Patch Management (SPM)* und der Bereich *Hardening, Provisioning and Maintenance (HPM)*. Das Schwachstellen- oder Vulnerability-Management ist für das Thema „Sicherheit" so fundamental, dass ein zusätzlicher Bereich (VAM) geschaffen werden musste. Ein weiterer zusätzlicher Bereich ist das *Patch-Management (SPM)*. Überraschenderweise findet man diese wichtige Aktivität nur sehr schwer in ISO/IEC 20000 bzw. ITIL. Sie ist aber zu wichtig und muss dediziert ausgearbeitet werden, um sicherzustellen, dass Schwachstellen systematisch beseitigt und Sicherheitslücken geschlossen werden. Der Bereich *Hardening, Provisioning and Maintenance (HPM)* umfasst wichtige praktische Richtlinien für die Umsetzung von Vorgaben im allgemeinen Lebenszyklus, wie er durch ITIL-Prozesse beschrieben ist.

In der Ausformung und Ergänzung der ITIL-Prozesse durch ESARIS entsteht ein vollständiges Bild eines Lebenszyklus, der die IT-Sicherheit vollständig abbildet und berücksichtigt. Ablauf und Zusammenhänge werden nun kurz skizziert. Für Details sei auf die zitierte Literatur verwiesen.

Für das Thema dieses Artikels besonders relevant sind nun folgende Vorgänge (siehe Abbildung 1). Das Life-Cycle-Management sieht die regelmäßige Aktualisierung von Software (Patch) vor. Die Grundlage dafür sind vorab definierte Abläufe und Verfahrensweisen, was die Definition von Patch-Klassen einschließt *(Security Patch Management (SPM))*. Der Patch-Prozess beginnt mit Aktivitäten wie dem Sammeln, der Qualitätsüberprüfung, der Paketierung und der Erstellung eines Zeitplans für die Implementierung. Dem schließen sich Verträglichkeitstests an. Die

2 IT Infrastructure Library; ITIL ist eine Sammlung von bewährter Verfahren zur Umsetzung eines IT-Service-Managements (ITSM), die inzwischen einen De-facto-Standard darstellt.

Implementierung wird durch das Change-Management gesteuert *(Change and Problem Management (CPM))*. Dieser Prozess sorgt für die Notfallvorsorge und Vorbereitung eines Roll-back, für die Risikobewertung und für Prüfungen und Freigaben. Am Ende werden die Patches implementiert und der Change wird als durchgeführt gemeldet. Der Patch-Status wird im Rahmen des Asset-Managements gepflegt und ist bekannt *(Asset and Configuration Management (ACM))*.

Viele Sicherheitspatches gehen diesen Weg, da der Übergang zwischen Funktion und Sicherheit oft fließend ist. Allerdings gibt es nicht umsonst den erwähnten zweiten Bereich des Schwachstellenmanagements *(Vulnerability Management and Mitigation Planning (VAM))*. Es werden beispielsweise Scanner betrieben, die Systeme gezielt daraufhin untersuchen, ob Schwachstellen vorliegen. Es gibt viele und manchmal besondere Quellen, die über Schwachstellen informieren, und Penetrationstests und forensische Untersuchungen liefern weitere Hinweise auf Schwachstellen gleich welcher Art.

Das Schwachstellenmanagement nimmt nur eine Analyse und Bewertung vor und gibt Vorschläge für das weitere Vorgehen (hier „mitigation planning" genannt). Was kommt danach? Liegt eine unmittelbare Gefahr vor, wird der Vorgang dem *Incident-Management (IHF)* übergeben, auch wenn Ursache und Maßnahme zur Behebung schon bekannt ist, weil zum Beispiel bereits ein Patch existiert. Die Implementierung erfolgt wieder über den *Change-Prozess*[3] mit dem *Patch-Management-Prozess* als letztlich ausführendem Organ. Liegt keine unmittelbare Gefahr vor, so wird die Frage gestellt, ob die Lösung zur Behebung der Schwachstelle bekannt ist. Liegt ein Softwareaktualisierung vor, ist das natürlich der Fall. Doch das Schwachstellen-Management betrachtet auch andere Fälle. Im ersten Fall werden alle Informationen an das Patch-Management übergeben, im zweiten Fall ist die Lösung vielleicht gar nicht bekannt, sodass ins *Problem-Management* verzweigt wird.

Auf diese Weise wird in einer großtechnischen, industrialisierten IT-Produktion sichergestellt, dass Schwachstellen systematisch erkannt und beseitigt werden, indem Software aktualisiert wird. Der IT-Dienstleister wird diese Leistung nur dann in der erforderlichen Qualität erbringen, wenn dies Bestandteil seines IT-Services ist. Lässt der Kunde z.B. eigene Anwendungen auf der Infrastruktur des IT-Dienstleisters betreiben, so kann der Kunde selbst für die Aktualisierung der Anwendungssoftware verantwortlich sein, während sich der IT-Dienstleister um die Aktualität der Infrastrukturkomponenten kümmert. Umgekehrt ist es aber häufig auch so, dass der IT-Dienstleister, evtl. auch im Auftrag von Dritten, die Aktualisierung von Geräten und Systemen übernimmt, die sich nicht in seinem Besitz befinden und auch gar nicht in einem Rechenzentrum installiert sind. Das kann auch IoT-Geräte umfassen, die dann als sogenannte „managed Services" angeboten werden. IT-Dienstleister verfügen also sehr wohl über das Instrumentarium, kontinuierlich Schwachstellen durch die Aktualisierung von Software zu beseitigen.[4]

Firmenkunden können ihren Verträgen mit dem IT-Dienstleister entnehmen, ob diese Aktualisierungen Teil der angebotenen und vertraglich vereinbarten Dienstleistung sind. Wenn sie IT-Services direkt beziehen, verfügen sie auch über die notwendige Expertise, entsprechende Nachweise in Form von Sicherheitsberichten auszuwerten, die der IT-Dienstleister ihnen zur Verfügung stellt.

Anders sieht es bei Konsumenten aus und bei Firmenkunden, die die IT-Dienstleistung gar nicht explizit beziehen, sondern implizit Komponenten und Systeme einsetzen, die die IT-Dienstleistung der Aktualisierung eigentlich erfordern (würden). D.h. es besteht ein Problem bei Konsumenten, die diese Zusammenhänge und ihre Tragweite nicht erfassen, und es besteht ein Problem bei Firmenkunden, die IT-Geräte und Systeme betreiben, von denen sie gar nichts oder nicht genug wissen. Für diese beiden Fälle ist die nachfolgend skizzierte Lösung gedacht. Sie ist als Idee und Anstoß zu verstehen.

3 Idee für sichere Geräte im Internet-of-Things

3.1 Blickfang und Lösungsskizze

Die folgende Skizze enthält schon fast die gesamte Lösungsbeschreibung. Die Lösung wird weiter unten diskutiert, wofür wir die Spieltheorie bemühen werden.

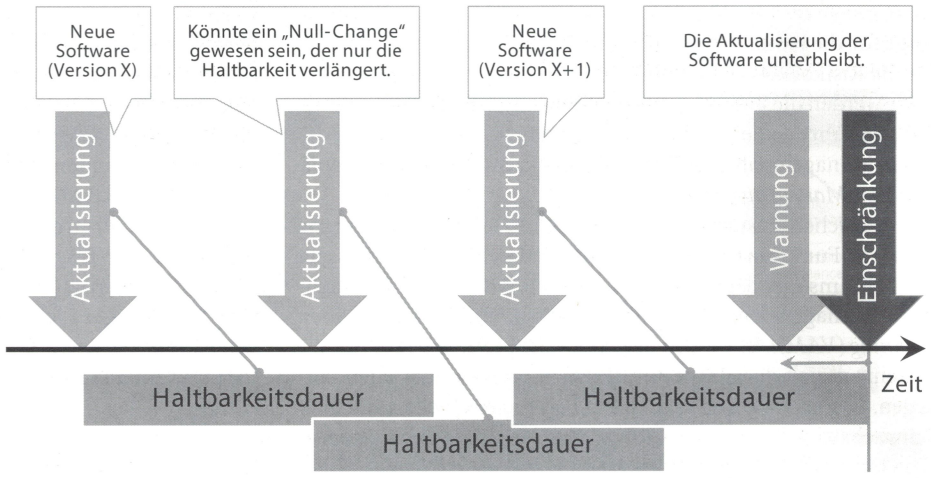
- **Ein dahin rasender Zug kann für Mitreisende und für die Umgebung eine ernsthafte Gefahr darstellen, wenn sein Lokomotivführer einschläft oder aus anderen Gründen seinen Pflichten nicht nachkommt. Deshalb hat man in Züge eine sogenannte Totmannschaltung eingebaut: Alle 30 Sekunden muss der Lokführer diese betätigen. Tut er das nicht, wird er erst akustisch gewarnt. Reagiert er nicht, wird eine Zwangsbremsung ausgelöst, um Schlimmes zu verhindern.**
- **Brauchen wir das nicht auch für die Geräte im „Internet of things"? IoT-Geräte besitzen Schwachstellen. Werden diese durch die Aktualisierung der Software nicht beseitigt, können die Geräte großen Schaden anrichten, ja sogar zur Waffe werden. Wie wäre es, wenn diese Geräte eine Mindesthaltbarkeit hätten? Wird die Software in dieser Zeit aktualisiert, wird die Haltbarkeit verlängert. Wenn nicht, erhält der Nutzer erst eine Warnung. Dann, falls keine Reaktion erfolgt, verweigert das Gerät seinen Dienst oder reduziert seine Funktionalität dergestalt, dass es nicht zur Gefahr werden kann. Die regelmäßige Aktualisierung des Herstellers führt dagegen zur Beseitigung von Schwachstellen und verlängert die Haltbarkeit jeweils um eine weitere Periode.**
- **Wir nennen dies „Life sign control". Der Anwender erkennt das Mehr an Sicherheit an einem Zeichen/Logo! Es signalisiert: Dieses Gerät ist immer „frisch", es ist noch haltbar und auf dem neuesten Stand der Technik des Herstellers. Das kann jeder erkennen: ganz ohne technisches Know-how, ganz ohne die Kontrolle von Softwareversionen usw. Der Anwender sieht nur das Logo mit der Haltbarkeitsperiode und weiß, der Hersteller kümmert sich. Das Resultat? Nur regelmäßig gepflegte, „lebende" Geräte verbleiben im Netz. Für mehr Sicherheit. Für mehr Qualität.**[5]

3 Normale "Changes" werden oft in Form von "Release" gesammelt und geplant. Alle „Changes" (normale und Notfall-Änderungen) werden über den gleichnamigen Prozess implementiert.

4 Da wir immer von einem Rollenmodel ausgehen, schließt dies natürlich Hersteller ein, die die Aktualisierung als IT-Dienstleistung anbieten.

5 Der Begriff "Mindesthaltbarkeit" geht auf eine Pressekonferenz des Chaos-Computer-Clubs zurück, der diese jedoch anders definiert: *33C3: Hacker rufen nach Mindesthaltbarkeitsdatum für vernetzte Geräte*; Heise online, 31.12.2016, 14:40 Uhr.

Abbildung 2 | Schematische Darstellung der Funktion der Lösung



Es ist darauf hinzuweisen, dass diese Lösung nur die in Abschnitt 1.2 identifizierten drei Probleme löst. Das bedeutet folgendes. *Erstens:* Auch Konsumenten können Geräte und Systeme mit besserer Sicherheit von denen mit schlechterer IT-Sicherheit unterscheiden. Ein entsprechend angebrachtes Zeichen/Logo signalisiert dies. *Zweitens:* Ähnlich wie bei Öko-Siegeln garantiert das Logo selbst nicht die Einhaltung des damit verbundenen Versprechens. Allerdings ist der Hersteller oder Dienstleister in der Pflicht und Kontrollen sind möglich. Da es sich bei den in Rede stehenden IoT-Geräten um Massenware handelt, werden die Softwareaktualisierungen auf eine große Masse von Geräten ausgebracht. Doch auch das lässt sich kontrollieren. In jedem Fall erfolgt die Aktualisierung flächendeckend und in der Regel aktiv, ohne dass der Anwender die Initiative ergreifen muss. Bevorzugen Anwender solche Geräte mit Zeichen/Logo, so wird die Verbreitung der Lösung weiter erhöht.

Drittens: Während der Fall 2 bei vielen Computersystemen auch für Konsumenten Standard ist, führt das Nichteinspielen der aktualisierten Software in der Regel nicht zu Konsequenzen für den Anwender oder den Hersteller/Dienstleister. Da die „Totmannschaltung" jedoch nach einer Warnung zwingend zur Einschränkung der Funktionalität bzw. zur vollständigen Abschaltung führt, sind nun die Konsequenzen klar. Dabei geht es eigentlich primär nicht darum, dass Geräte ihren Dienst versagen. Vielmehr sorgt die Lösung dafür, dass alles dafür getan werden wird, die Aktualisierung regelmäßig durchzuführen. Spieltheoretisch handelt es sich um den Mechanismus der *Selbstbindung*, die dafür sorgt, dass sich die Beteiligten nicht durch andere, scheinbar wichtigere Dinge von der Einhaltung ihrer Zusage abbringen lassen.

Selbstbindung ist ein "strategischer Zug" in "sequentiellen Spielen". Strategische Züge verfolgen die Absicht, Erwartungen und Verhalten anderer zu beeinflussen. Im vorliegenden Fall der Selbstbindung geht es darum, dass sich die verpflichtende Partei sich selbst beeinflusst ("Einhaltung des Versprechens") und damit auch andere Marktteilnehmer zu beeinflussen beabsichtigt ("dem Versprechen glauben"). Mehr Details enthält z.B. [1].

Natürlich ist die skizzierte Lösung nicht einsetzbar bei Geräten, bei denen Verfügbarkeit das oberste Gebot ist, wie z.B. bei lebenserhaltenen Systemen. Doch mehr zu den Einschränkungen in Abschnitt 3.3. Im folgenden Abschnitt 3.2 geht es um Details bei der Umsetzung.

3.2 Implementierungshinweise

Die Lösung soll nur geringe technische Anforderungen an die IoT-Gerät stellen. Die unmittelbar folgenden Anforderungen sind unabhängig von der Implementierung der Lösung und sind nur erforderlich, um die Funktion der Softwareaktualisierung unterstützen zu können. Für Massenware ist das keine Selbstverständlichkeit, auch wenn selbst Prozessorchipkarten über solche Funktionen verfügen. Weiter unten werden die Anforderungen beschrieben, die sich auf die Lösung selbst beziehen.

- Das IoT-Gerät muss über die Fähigkeit verfügen, Softwarepatches zu laden und die Software zu aktualisieren. Dazu wird neben IT-Funktionalität auch ein entsprechend großer, schreibbarer, nicht-flüchtiger Speicher benötigt, der zur Programmausführung befähigt ist. Unterbrechungen z.B. in der Stromversorgung dürfen dabei nicht zum Defekt des IoT-Gerätes führen. Die eigentlichen Funktionen zur Umsetzung sind jedoch bekannt und Stand der Technik.
- Das IoT-Gerät muss über eine Möglichkeit verfügen, die Authentizität des Patches prüfen zu können, bevor die neue Softwareversion aktiv wird. Die Prüfung der Authentizität schließt die Prüfung der Echtheit (des Datenursprungs) und der Unverfälschtheit des Patches ein. Üblicherweise werden dafür Signaturen oder MACs eingesetzt, was jedoch die Fähigkeit des IoT-Gerätes erfordert, kryptografische Operationen mit Algorithmen ausreichender Stärke durchzuführen und die verwendeten kryptografischen Schlüssel sicher speichern zu können.

Die folgenden Funktionen sind zusätzlich nötig, um die vorgeschlagene Lösung zu unterstützen:

- Mit jeder Softwareausstattung (Patch) erhält das IoT-Gerät dessen Datum und speichert diese „Zeit der letzten Aktualisierung" persistent. Das IoT-Gerät muss in der Lage sein, die „aktuelle Zeit" zu ermitteln. Dafür verfügt es entweder über eine interne Echtzeituhr oder es nimmt regelmäßig Verbindung zu einem Zeitserver auf.
- Das IoT-Gerät bestimmt regelmäßig die Differenz aus „aktueller Zeit" und der „Zeit der letzten Aktualisierung". Verwendet man die Unixzeit, so ist es zur Ermittlung der „verstrichenen Zeit" lediglich notwendig, zwei 32 Bit lange Zahlen voneinander zu subtrahieren. Eine zweite solche Subtraktion liefert den Vergleich der „verstrichenen Zeit" mit der im Gerät gespeicherten „Haltbarkeitsperiode".[6]
- Ist die „verstrichene Zeit" größer als die gespeicherte „Haltbarkeitsperiode", so muss das IoT-Gerät in der Lage sein, seine Funktionalität soweit zu reduzieren, dass von ihm keine Gefahr

6 Auch wenn die Differenz sekundengenau ermittelt wird, geht es doch hier wahrscheinlich nur um Tage. Die „Haltbarkeitsperiode" kann Wochen oder sogar Monate betragen.

ausgehen kann. Kommt die verstrichene Zeit der gespeicherten „Haltbarkeitsperiode" nahe, so wäre es wünschenswert, wenn das IoT-Gerät eine Warnung ausgeben könnte.

Ein solcher Ablauf ist schematisch in Abbildung 2 dargestellt.

Die nicht-technischen Anforderungen umfassen unter anderem folgende:

- Die Regeln für die Implementierung der Lösung und die damit verbundenen Implikationen müssen genau ausgearbeitet und öffentlich einsehbar sein.
- Es muss eine Version dieser Regeln geben, die auch für Laien vollständig verständlich ist. Das Zeichen/Logo besitzt ausreichenden Wiedererkennungs- und Informationswert.
- Das Zeichen/Logo wird ebenfalls bekannt gemacht. Es signalisiert, dass die Partei, die dieses Zeichen/Logo für Produkte oder Dienstleistungen verwendet, sich zur Einhaltung der Regeln verpflichtet. (Das bedeutet auch, dass Anwender und Hersteller den Defekt von IoT-Geräten in Kauf nehmen, sollte sich der Hersteller oder IT-Dienstleister dazu entscheiden, die Aktualisierung nicht mehr durchzuführen. Das gleiche gilt, wenn er dazu z.B. aufgrund der Geschäftsaufgabe nicht mehr in der Lage ist.)
- Das Zeichen/Logo könnte zusätzlich folgende Informationen vermitteln: Art der Warnung (z.B. Email, Anzeige, keine) und Länge der „Haltbarkeitsperiode" (z.B. Monate, ein Monat, Wochen).

Die technischen Anforderungen machen die *Selbstbindung* glaubhaft und setzen sie durch. Zusätzlich gibt es nicht-technische Anforderungen. Sie werden häufig unterschätzt. Die Spieltheorie belehrt uns auch hier eines besseren. Die Lösung des Spiels hängt entscheidend davon ab, welche Informationen den einzelnen Spielern zur Verfügung stehen. Die Unkenntnis der Strategie des Mitspielers kann dazu führen, selbst unberechenbar zu reagieren (d.h. eine sogenannte *Mischstrategie* anzuwenden [1]). Der Versuch, die Nicht-Beobachtbarkeit der eigenen Strategie auszunutzen, führt in der Regel zu einem für beide Seiten schlechteren Ergebnis [3]. Deshalb muss versucht werden, das Spiel in Richtung eines *Spiels mit vollständiger Information* zu entwickeln, weil sich dann gewünschte Ergebnisse leichter herbeiführen lassen bzw. wahrscheinlicher werden.

Juristische Fragen sind nicht Gegenstand dieser Skizze.

3.3 Einschränkungen

Es muss darauf hingewiesen werden, dass es bei der skizzierten Lösung einen Zielkonflikt zwischen Sicherheit und Verfügbarkeit gibt. Dieses rührt daher, dass Funktionseinschränkungen erfolgen, falls der Hersteller/Dienstleister seiner Zusage nicht nachkommt oder nachkommen kann, für eine Aktualisierung der Software zu sorgen. Das ist die Achillesverse der Lösung! Sie ist aber unbedingt notwendig, was weiter unten noch einmal spiel-

theoretisch untermauert wird. Deshalb wird es eine Klasse von IoT-Geräten geben, für die diese Lösung nicht einsetzbar ist! Man könnte überlegen, ob es sinnvoll ist, für diese Fälle eine etwas abweichende Lösung anzubieten, deren Implementierung wiederum über das Zeichen/Logo signalisiert werden müsste.

Unter Softwareaktualisierung ist nicht zu verstehen, dass diese regelmäßig verändert wird. Gibt es keine Schwachstelle, die der Hersteller beseitigen muss/kann, so gibt es einen „Null-Patch", der dem IoT-Gerät signalisiert, dass es auf dem aktuellen Stand ist und die „Zeit der letzten Aktualisierung" aktualisieren muss. Das bedeutet aber auch, dass die Hersteller/Dienstleister innerhalb der Spanne einer „Haltbarkeitsperiode" mindestens einen Patch erzeugen und verteilen müssen.

Die Lösung bedeutet nicht, dass die Aktualisierung der Software vollständig automatisch und ohne eine Mitwirkung des Anwenders erfolgen muss. Hier kann es durchaus unterschiedliche Implementierungen geben. Für IoT-Geräte, die nicht durch den Anwender betreut oder gewartet werden, wird es einen automatischen „Push-Service" geben. Bei Geräten, bei denen der Anwender direkt mit der IT-Funktionalität arbeitet, kann es dagegen sinnvoll sein, es dem Anwender zu erlauben, die Aktualisierung der Software zunächst aufzuschieben, z.B. wenn die Aktualisierung mit einer temporären Einschränkung von Funktionalität oder Verfügbarkeit verbunden ist. Dies ist z.B. von Windows-Rechnern in Privatbesitz bekannt.

4 Diskussion und Ausblick

Es sind weitaus kompliziertere und auch wirksamere Lösungen denkbar, um die Sicherheit von IoT-Geräten zu erhöhen. Die hier vorgestellte zeichnet sich dadurch aus, dass auf eine eigene, zentrale Infrastruktur verzichtet wird und dass sich die Implementierung an den beschränkten Möglichkeiten vieler IoT-Geräte orientiert, wie sie z.B. für eingebettete Systeme[7] kennzeichnend sind.

Deshalb sei an dieser Stelle daran erinnert, dass die Lösung nur das zu Anfang beschriebene Problem lösen soll: Bei einer bestimmten Klasse von Geräten besteht die Gefahr, dass nicht erkannt wird, ob die Aktualisierung der Software erstens vorgesehen ist und zweitens durchgeführt wird.

Dahinter steht die Annahme, dass kompliziertere Systeme sich nicht durchsetzen lassen oder nur in einem sehr eingeschränkten Bereich Anwendung finden. Im Gegensatz etwa zu Zahlungsverkehrssystemen, bei denen Evaluierungen der IT-Sicherheit vorgeschrieben und Zulassungen Voraussetzung für den Einsatz sind, handelt es sich bei den Geräten im Internet nicht um ein geschlossenes sondern ein offenes System und nicht um einen kontrollierten sondern einen freien Markt. Deshalb war es das Bestreben, die Anforderungen möglichst gering anzusetzen. Das bedeutet natürlich auch, dass der Effekt begrenzt ist. Es besteht die Hoffnung, dass die Wirkung dennoch ausstrahlt und erheblich ist, sollte die Lösung jemals zum Einsatz kommen.

Eine weitere Annahme besteht darin, dass es nicht klug ist darauf zu bauen, dass gute Lösungen schließlich weniger gute verdrängen. Die IT ist eher ein *„Markt für Zitronen"*, bei dem Produkte höherer Qualität eher verdrängt werden. Dies gilt insbesondere im Segment der Konsumentenprodukte, aber nicht nur dort. Der *Zitronenmarkt* ist ein von George Akerlof (Wirtschaftsnobelpreis 2001) beschriebenes Phänomen, dass seinen Ausgangspunkt darin nimmt, dass Konsumenten weniger gut über die Qualitäten eines Produktes informiert sind (als der Hersteller) und daher eher billigere Produkte wählen, auch wenn sie dabei das Risiko eingehen, eine „Zitrone" zu erwischen statt einer „Orange". Siehe z.B. [2].

Der spieltheoretische Ausweg aus dem Dilemma (weniger Zitronen und höhere Erlöse) besteht klassisch in der Auflösung der Informationsasymmetrie: Der Kunde wird besser informiert, was aber in der IT regelmäßig scheitert, da Konsumenten oft nicht über das nötige Know-how verfügen, solche Informationen und ihre Konsequenzen zu verstehen und die richtigen Schlussfolgerungen daraus zu ziehen. Das Zeichen/Logo informiert zunächst auch. Aber es leistet mehr! Es signalisiert (1) ein Garantieversprechen und bringt (2) den Hersteller/Dienstleister über die damit verbundene *Selbstbindung* dazu, für die Aktualisierung wirklich zu sorgen. Die Selbstbindung im Sinne der Spieltheorie wird dadurch erreicht bzw. verstärkt, dass die Nichteinhaltung des Versprechens zu ernsten Konsequenzen führt: Das Gerät ist nicht mehr so funktionsfähig wie erwartet und bezahlt. Das dürfte den Druck auf den Hersteller/Dienstleister enorm erhöhen.[8]

Doch was hat der Hersteller/Dienstleister davon? Was könnte seine Motivation sein, sich diesen Regeln der *Selbstbindung* zu unterwerfen? Es gibt zum einen eine große Anzahl von Herstellern/Dienstleistern, für die die Aktualisierung der Software ohnehin schon zum Service gehört. Sie werden die Kennzeichnung gerne aufgreifen, weil für sie damit keinerlei zusätzlicher Aufwand verbunden ist, sondern nur ihre potenziell bessere Leistung sichtbar wird, weil sie ja süße „Orangen" verkaufen. Diese Sichtbarkeit setzt die Hersteller der sauren „Zitronen" unter Druck. Die Frage ist, wie sich die Anwender entscheiden. Werden sie das Risiko eingehen, „Zitronen" zu kaufen, wo sie doch „Orangen" hätten haben können? Die Frage kann nicht abschließend beantwortet werden. Zu viele Faktoren beeinflussen den Kauf von IoT-Geräten und die Dynamik der Installationen.

Welches Risiko gehen die Anwender ein und sind sie motiviert, die „neuen" Lösungen zu bevorzugen? Vordergründig haben die so gekennzeichneten Geräte ja einen Makel. Sie versagen den Dienst, wenn der Hersteller/Dienstleister nicht mehr mitmacht. Aber wie wahrscheinlich ist es, dass die Marktführer, die Kaufentscheidungen dominieren, einmal eingegangene Versprechen ohne weiteres und für jeden sichtbar brechen? Die Autoren schätzen diese Wahrscheinlichkeit als eher gering ein. Wahrscheinlicher ist der Fall einer Insolvenz oder einer Änderung des Geschäftsmodells, die unausweichlich dazu führt, die Versprechungen nicht mehr zu erfüllen. Ein Imageschaden mit Auswirkungen für das Folgegeschäft wäre in diesen Fällen eher nicht zu erwarten. Allenfalls könnte es Auswirkungen auf andere Geschäftsbereiche geben. Bei alledem ist zu bedenken, dass die Lebensdauer von IT in der Regel begrenzt ist und die IT einem schnellen

7 *Embedded Systems* sind Systeme (a) mit Computer- bzw. Rechnerfunktionalität, die jedoch (b) nicht frei programmierbar, sondern für einen sehr spezifischen Einsatzzweck gebaut sind und nur diesen unterstützen, die (c) in der Regel mit ihrer Umwelt interagieren, also Informationen austauschen, und (d) in besonderer Weise bautechnischen Beschränkungen unterworfen sind. Die zuletzt genannte Eigenschaft ist hier besonders wichtig. Bei IoT-Geräten handelt es sich oft um Massenware, bei denen ein geringer Preis, geringe Abmessungen und geringe Unterhaltskosten eine entscheidende Rolle spielen.

8 Eine weitere Überlegung wäre, dass dieses Zeichen/Logo von staatlicher Stelle unterstützt wird, ähnlich wie z.B. der „Blauer Engel" (Umweltzeichen) von staatlichen Stellen unterstützt wird. Im weiteren Sinne geht es bei beidem um den Schutz des öffentlichen Raums.

Wandel mit hohen Erneuerungsraten unterworfen ist. In diesem Kontext ist es aus Sicherheitsgründen auch durchaus beabsichtigt, dass veraltete, „vergessene" und nicht mehr gepflegte Geräte durch die Lösung aussortiert werden.[9]

Auf der anderen Seite gehen Anwender eventuell ein Risiko ein, wenn sie IoT-Geräte kaufen, die über diese Zeichen/Logo nicht verfügen. Diese Geräte haben schon beim Kauf erkennbar ein Makel; sie werden nicht verbessert. Gegebenenfalls riskiert der Besitzer oder Betreiber Schadensersatzansprüche für den Fall, dass die Geräte zweckentfremdet verwendet werden.

Dennoch ist die vorgeschlagene Lösung kein Allheilmittel. Das wurde mehrfach betont. Insbesondere ist sie nicht für alle Klassen von Geräten anwendbar. Grob könnte man die mit dem Internet agierenden Geräte oder Systeme in drei Klassen teilen und diesen unterschiedliche Lösungen wie folgt zuordnen:

- Für Geräte bzw. Systeme mit einer hohen Rechnerkapazität wird die in Kapitel 2 beschriebene Lösung einer zentralen, vertraglich verabredeten Aktualisierung der Software bevorzugt und als adäquat angesehen.
- Für Geräte bzw. Systeme mit einer mittleren Rechnerkapazität ist die neue Lösung (siehe Kapitel 3) genau die richtige. Diese machen die Masse und die schnell wachsende Zahl der Dinge im Internet-der-Dinge aus. Hier wird momentan das Hauptproblem gesehen.
- Für Geräte bzw. Systeme mit sehr geringe Rechnerkapazität oder mit sehr hohen Verfügbarkeitsanforderungen fällt die erste Variante aufgrund einem Mangel an technischen Voraussetzungen aus. Die zweite mag technologisch realisierbar sein, aber die hohen Anforderungen an die Verfügbarkeit verbieten es, die hier neu vorgestellte Lösung zu implementieren.

IT ist komplex. Es ist nicht schlimm, wenn eine Lösung nicht die Welt heilt. Es ist ausreichend, wenn sie ihren klar definierten Zweck erfüllt und ihr Versprechen wirklich einlöst. Genau in diesem Sinne ist dieses Diskussionspapier zu verstehen. Die vorgeschlagene Lösung einer *„Mindesthaltbarkeit"* soll die Diskussion über mehr Sicherheit im Internet-der-Dinge befördern.

In der IT-Sicherheit spielen Entscheidungen eine wesentliche Rolle. Die Anwendung spieltheoretischer Konzepte kann hierbei Hilfestellungen geben, die Entscheidungen einzelner Akteure zu verstehen und die Gegebenheiten so zu gestalten, dass das gewünschte Verhalten wahrscheinlicher wird oder sogar erzwungen wird. Die Verwendung der Konzepte *Selbstbindung* (als strategischer Zug, [1]) und der Schlussfolgerungen aus der Analyse der *Zitronenmärkte* [2] sind dabei nur Beispiele. *Information* und *Kommunikation* spielen eine entscheidende Rolle. Die Spieltheorie hilft zu verstehen, dass diese Aspekte nicht einfach nur Voraussetzungen darstellen, sondern dass *Information* und *Kommunikation* helfen können, das Spiel in Richtung eines *Spiels mit vollständiger Information* zu entwickeln und die Auszahlungen so zu steuern, dass das gewünschte Verhalten auftritt bzw. wahrscheinlicher wird [3]. Damit ist das Potenzial der Spieltheorie nicht erschöpft. Mechanismen der *Koalitionsspiele* [3] können helfen zu verstehen, wie sich Abmachungen zwischen Spielern auswirken. *Das Studium wiederholter Spiele* mit problematischer Ausgangssituation wie dem *Gefangenendilemma* können zeigen, wie Auszahlungen und Zukunftsperspektiven in Form gesteuerter Wiederholungen oder diskontierter Auszahlungen rationale Entscheidungen in Richtung Zusammenarbeit lenken können [4]. Und schließlich erinnern die Grundlagen der Spieltheorie daran, dass es vor allem auf die richtige Analyse der Alternativen und die Einschätzung des jeweiligen Nutzen [5] ankommt. Obwohl die Nutzenfunktionen oft auf Annahmen und fehlerhaften Einschätzungen beruhen, liefert die Analyse des richtig identifizierten Spiels doch oftmals erstaunlich eindeutige und hilfreiche Strategieempfehlungen.

9 Die Aktualisierung könnte zusätzlich an eine Garantiezeit gebunden sein, um das finanzielle Risiko für den Hersteller zu verringern, dass mit der Pflege von „Altgeräten" verbunden wäre.

Literatur

[1] Avinash K. Dixit and Barry J. Nalebuff: Spieltheorie für Einsteiger, Strategisches Know-how für Gewinner; Schäffer-Poeschel Verlag für Wirtschaft, Ulm, 1993; englische Ausgabe: Thinking Strategically; W. W. Norton & Company, 1993

[2] Ken Binmore: Fun and Games, A Text a Game Theory; D. C. Heath and Company, 1992

[3] Manfred J. Holler und Gerhard Illing: Einführung in die Spieltheorie; Springer, 7.Auflage, 2009

[4] Robert Axelrod: Die Evolution der Kooperation; R. Oldenburg Verlag, München, 1987

[5] Morton D. Davis: Spieltheorie für Nichtmathematiker; R. Oldenburg Verlag, München, 1993

[6] ISO/IEC 27002 – Information technology – Security techniques – Code of practice for information security management

[7] Information Security Forum (ISF) – The Standard of Good Practice for Information Security 2016

[8] ISO/IEC 20000 – Information technology – Service management – Part 1: Service management system requirements, Part 2: Guidance on the application of service management systems

[9] Eberhard von Faber and Wolfgang Behnsen: Secure ICT Service Provisioning for Cloud, Mobile and Beyond, ESARIS: The Answer to the Demands of Industrialized IT Production Balancing Between Buyers and Providers, 2017, ISBN- 978-3-658-16481-2, XIV + 368 pages – 2. ERWEITERTE und AKTUALISIERTE Auflage

[10] Eberhard von Faber: Organisation der Absicherung einer industriellen IT-Produktion, Drei Handlungsfelder jenseits von Protection, Detection, Reaction; Datenschutz und Datensicherheit (DuD), Heft 10, 2016, Seiten 647-654

[11] ESARIS Security Taxonomy – Synopsis, Scope and Content; Zero Outage Industry Standard, Release 1 about Security, February 2017, https://www.zero-outage.com/security