the information security provider

NEWS

01/2016

Standardisiertes Schlangenöl

Global, unabhängig, standardorientiert: So beschreiben wir uns bei atsec gerne. Global und unabhängig ist jedem klar, aber warum um Himmels Willen Standards? Weil wir überzeugt sind, dass wir durch international akzeptierte Standards wie ISO 27000 oder Common Criteria unseren Kunden eine neutrale und vergleichbare Messlatte unserer Arbeit und der Einordnung ihrer eigenen Informationssicherheit ermöglichen. International müssen die Standards sein, weil der Großteil unserer Kunden global orientiert ist und die eigenen Anstrengungen zur Informationssicherheit den eigenen Kunden nachweisen will.

Um keine Verwirrung aufkommen zu lassen: Standards alleine machen niemanden sicher, auch nicht, wenn sie international daherkommen. Ich werde meinen Winterspeck ja auch nicht dadurch los, dass ich nur auf die Waage steige.

Was mich besorgt, sind zwei Beobachtungen: Erstens, selbstgebastelte oder ausschließlich nationale Standards haben als Geschäftsmodelle zur „Kundenbindung“ nichts von ihrer Attraktivität eingebüßt. Beispielsweise wird gerade bei Unternehmen in kritischen Infrastrukturen viel Werbung für ISIS12 gemacht, mit dem sich die Komplexität von ISO 27001 für den Aufbau eines Informationssicherheits-Managementsystems (ISMS) angeblich auf einen Bruchteil reduzieren lässt. Das wird dadurch erreicht, dass gar kein ISMS aufgebaut wird, sondern nur versprochen wird, dass, wenn später einmal ein ISMS nach ISO27000 oder BSI-Grundschutz aufgebaut würde, nicht alles für die Katz gewesen ist. Dass das Verfahren laut Gutachten für kritische Infrastrukturen nicht geeignet ist, lässt mich vermuten, dass man die Stadtwerke hier mit Fördergeldern in eine falsche Richtung lockt. Wir bei atsec betreiben in unserem kleinen und überschaubaren Rahmen seit Jahren ein schlankes, nach ISO 27001 zertifiziertes ISMS. Statt Mittelständlern mit der Komplexität des Standards Angst zu machen, hätte man ihnen besser gesagt, wie seine generischen Bestandteile für diesen Bereich spezifisch auszulegen sind. Alle anderen schaffen das ja auch; wozu also nochmal ein eigener Standard?

Zum Zweiten sehe ich, dass auch internationale Standards nicht immun gegen nationale Interessen sind. Die Common Criteria (ISO 15408) zur Bewertung der Vertrauenswürdigkeit sicherer IT-Produkte wird aktuell von einigen beteiligten Nationen massiv umgedeutet, ausgehöhlt und auf ein nichts sagendes Einstiegsniveau reduziert. Wenn das wirklich gebraucht wird, sollte man dafür ehrlicherweise einen eigenen Standard definieren. Stattdessen bemächtigt man sich der Reputation der CC mit seiner unabhängigen Prüfung und ersetzt die Analysen der Evaluatoren durch Herstellererklärungen oder ein paar Compliance-Tests. So etwas nennt man in der IT-Branche Schlangenöl, in der Pharmazie Placebo.

Gerade deshalb bleibt für uns unser Engagement in der internationalen Standardisierung wichtig. Wir werden uns weiter dafür engagieren, dass es Sicherheitsstandards gibt, die aussagekräftig sind und niemanden in ein selbst gebasteltes Schema pressen, aus dem es kein oder nur ein teures Entrinnen gibt.

Herzlichst,
Ihr Gerald Krummeck
Leiter Prüfstelle

Events 2016

Die atsec information security wird in diesem Jahr auf folgenden Messen vertreten sein:

- **International Cryptographic Module Conference (ICMC) in Ottawa, Ontario, Canada:**
  18. bis 20.05.2016
  Die Experten-Konferenz für kryptografische Module findet auch 2016 wieder mit Unterstützung von atsec statt.
  Für Details siehe http://icmconference.org/.

MesseNürnberg

- **it-sa in Nürnberg:**
  18. bis 20.10.2016
  Auf der größten IT-Sicherheitsmesse im deutschsprachigen Raum wird sich atsec auch im Jahr 2016 wieder mit einem eigenen Stand präsentieren.

atsec it security blog
Verfolgen Sie brandaktuell Diskussionen und Erlebnisse unserer Mitarbeiter auf
http://atsec-information-security.blogspot.de

Risiko (Miss-)Management

Risikomanagement zählt heute zu den gängigen Schlagwörtern, wenn man über ein IT-Sicherheitskonzept spricht. Klar, man will seine Risiken kennen und sie – soweit möglich – eliminieren bzw. die Auswirkungen minimieren.

Im klassischen Safety-Bereich hat man gute Methoden zur Identifikation von Risiken und zu deren Abschätzung: Man analysiert, welche Ereignisse zu einem Schaden führen können, man berechnet die Eintrittswahrscheinlichkeit jedes zum Risiko führenden Ereignisses und den zu erwartenden Schaden pro Ereignis und multipliziert diese Werte. Man kann sich nun noch streiten, ob man den maximalen Schaden oder den durchschnittlichen Schaden in die Formel einfließen lässt, bzw. rechnet mit beiden Werten, wo-

bei der maximale Schaden meistens mit einer geringeren Wahrscheinlichkeit eintritt. Damit ist man bisher recht gut gefahren. Voraussetzung, dass man das Verfahren so anwenden kann, ist (was meist implizit angenommen wird), dass Eintrittswahrscheinlichkeit und Schadenshöhe nicht voneinander abhängig sind.

Auch im Bankenbereich ist Risikomanagement ein wichtiger Faktor. Hier wissen die Fachleute aber genau, dass Eintrittswahrscheinlichkeit und Schadenshöhe oft nicht unabhängig sind: Bei absichtlich herbeigeführten Schäden ist die Eintrittswahrscheinlichkeit umso höher, je höher der Schaden ist. Oder, wie mir einmal ein Freund im Spaß gesagt hat: Wer eine Bank um eine Million betrügen kann und es tut, der ist dumm. Wer eine Bank um 10 Millionen betrügen kann und es nicht tut, der ist auch dumm.

Kern der Aussage: Je höher der zu erwartende Gewinn für den Angreifer (= Schaden für die Bank) umso höher die Wahrscheinlichkeit, dass es geschieht! Folgerichtig konzentrieren sich Banken in vielen Fällen darauf, den zu erwartenden Schaden und nicht primär die Eintrittswahrscheinlichkeit zu reduzieren. Mit der Limitierung der Schadenshöhe sinkt automatisch die Eintrittswahrscheinlichkeit und sehr oft ist die Reduzierung der maximalen Schadenshöhe der wirtschaftlichere Weg.

Im Softwarebereich reden wir auch oft von „Risikomanagement“, häufig aber ohne zu wissen, welche Ereignisse zu einem Schaden führen können. Was wir wissen ist, dass oft Fehler in der Software Angriffe ermöglichen. Welche Fehler in welchen Produkten existieren, wissen wir dabei nicht bzw. erfahren wir erst, wenn es zu spät und der Fehler schon öffentlich bekannt ist und ausgenutzt wird. Die Zahl der Einträge in der CVE-Datenbank (cve.mitre.org) geben davon ein beredtes Zeugnis. Das einzige, was wir daraus lernen (sollten), ist: fehlerfreie Software ist seltener als das Einhorn und wer glaubt, dass wir in absehbarer Zeit in der Lage sein werden, fehlerfreie Software zu produzieren, der gibt sich einer Illusion hin.

Also sollten wir lernen, mit fehlerhafter Software zu leben. Das heißt nicht, dass wir in Zukunft auf die Korrektur von Fehlern verzichten sollten. Aber wir sollten Software so gestalten und nutzen, dass die Auswirkungen von Fehlern minimiert werden. Als Beispiel nehme ich gerne einen Fehler, der im Jahr 2014 für großes Aufsehen gesorgt hat: der „Heartbleed“ genannte Fehler in OpenSSL. Wer mehr über diesen Fehler wissen möchte (und wie man ihn mit größtmöglicher Publizität vermarktet hat), schaue sich die Webseite ‚heartbleed.com‘ an.

Also was ist Heartbleed? Ein Programmierfehler, wie er wohl täglich tausende Male gemacht wird: man hat vergessen zu prüfen, ob eine Anfrage von Unbekannt eventuell eine inkorrekte Länge enthält. Was ist die schlimmste Konsequenz im Falle von Heartbleed? Wie die Webseite so schön feststellt: der Verlust der „Kronjuwelen“, nämlich des privaten Schlüssels der eigenen Organisation!

Also um bildlich zu sprechen: Man „versteckt“ die Kronjuwelen hinter einer Tür, von der man weiß, dass sie so gut wie keinem Angriff standhält - wie etwa dem Versuch, sie mit Hilfe einer Kreditkarte zu öffnen - und wundert sich dann, dass die Kronjuwelen verschwunden sind!

Aber Abhilfe ist ja da: ein Patch, der diesen speziellen Fehler behebt! Das ist so gut wie ein spezieller Patch für die

Tür, der nur verhindert, dass die Tür weiterhin mit einer Kreditkarte geöffnet werden kann. Und dann wundert man sich, wenn wenig später jemand die Tür mit einem simplen Picking-Werkzeug öffnet - und die neuen Kronjuwelen sind dann schon wieder weg!

Tut mir leid, so drastisch zu sein, aber sein Risikomanagement im Wesentlichen auf das schnelle Einspielen von Patches und den Einsatz eines Virenscanners reduziert, betreibt Risiko-Missmanagement!

Wo liegt unser Problem: Wir entwickeln unsere Systeme nicht so, dass sie ‚fehlertolerant' sind!

Schauen wir uns einmal den Großrechnerbereich an. Dort gibt es Systeme, die seit Jahren kontinuierlich laufen ohne ein Re-Boot! Wow, wie machen die das? Hat diese Software keine Fehler?

Doch, sie hat - genau wie andere Software auch. Der Unterschied ist, dass diese Software in der Lage ist, Fehler möglichst frühzeitig zu erkennen und auftretende Fehler gezielt zu behandeln. In vielen Fällen merkt man im laufenden Betrieb nicht einmal, dass eine Komponente auf einen Fehler gelaufen ist und gezielt vorbereitete Maßnahmen ergriffen hat, um die Auswirkungen des Fehlers zu minimieren. Auch die Hardware ist dort entsprechend fehlertolerant konzipiert: Es kann einem bei einem Großrechner passieren, dass der Service-Techniker auftaucht, um eine Speicher- oder gar eine Prozessorplatine auszutauschen, ohne dass der Rechner angehalten werden muss.

Wie schützen nun Banken ihre „Kronjuwelen" in einem Großrechner? Nun, die privaten Schlüssel liegen dort natürlich nicht in einem Adressraum, der direkt eine Angriffsoberfläche nach Außen im Internet hat. Sie liegen stattdessen in einem Krypto-Coprozessor, also einem separaten Stück Hardware, welches im Gehäuse des Großrechners integriert ist. Diese Hardware selbst bietet einen extrem hohen Schutz, selbst wenn es jemandem gelänge, sich ihrer zu bemächtigen. Angesprochen werden können die Funktionen der Hardware nur über eine spezielle Komponente des Betriebssystems, welches in einem separaten Adressraum läuft.

Kurz gesagt, wer in diesem Fall über einen Fehler bei der Implementierung des TLS-Protokolls an die „Kronjuwelen" will, muss noch mehrere zusätzliche Schutzwälle überwinden, die sukzessive immer stärker werden.

Nun muss man ja nicht gleich an so weitgehende Schutzmaßnahmen denken wie sie im Bankenbereich verwendet werden. Aber es wäre schon ein großer Sicherheitsgewinn, seine Software so zu gestalten, dass wichtige Daten wie etwa private Schlüssel sich nicht in einem Adressraum befinden, der direkten Angriffen aus dem Internet ausgesetzt ist. Ein paar Gedanken zum Risikomanagement schon bei der Entwicklung bzw. Konfiguration von Software können helfen, Risiken durch Softwarefehler stark zu reduzieren.

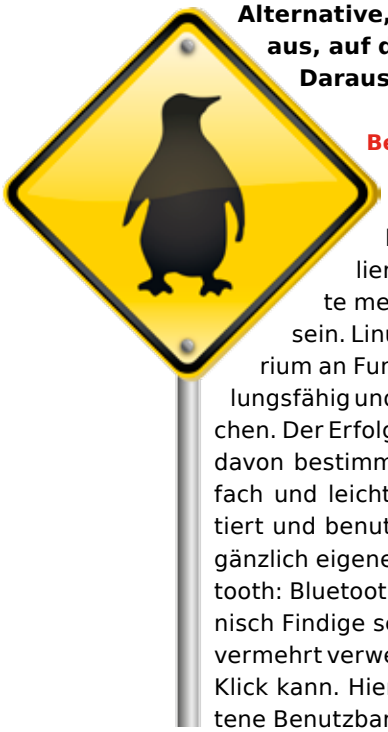
Ähnlich ist es übrigens auch bei der immer noch häufigsten Fehlerquelle: Fehler von Nutzern und Administratoren. Auch hier ist es sinnvoll, die Möglichkeit von Fehlern schon beim Design der Software zu berücksichtigen.Fehlertoleranz bedeutet hier, so weit wie möglich auszuschließen, dass ein einziger menschlicher Fehler zu einem katastrophalen Verlust führen kann. Speziell kritische administrative Funktionen sollten so gestaltet sein, dass man schon mehrere Dinge in geordneter Reihenfolge durchführen muss, um ein Ereignis mit einem katastrophalen Verlust herbeizuführen. Wenn dies dann doch passiert, ist das schon Vorsatz und den kann man nie ganz ausschließen.

Fazit

Risikomanagement sollte schon bei Software-Design und -Konfiguration eine wichtige Rolle spielen. Wer das Risikomanagement erst startet, wenn die Software schon im Einsatz ist, beraubt sich der Möglichkeit das Risiko durch Softwarefehler oder durch menschliches Fehlverhalten schon im Vorfeld deutlich zu reduzieren. Die Wahrscheinlichkeit, dass wir einmal fehlerfreie Software bekommen, ist wohl geringer als die Wahrscheinlichkeit, ein Einhorn in freier Wildbahn zu Gesicht zu bekommen. Ebenso wenig werden wir (unbeabsichtigtes) menschliches Fehlverhalten jemals ausschließen können. Was wir aber können, ist, unsere Softwaresysteme so zu bauen und zu konfigurieren, dass die Auswirkungen solcher Fehler minimiert werden. Dazu ist es aber notwendig, mit dem Risikomanagement schon bei der Konzeption solcher Systeme zu beginnen und sowohl die Möglichkeit von Softwarefehlern als auch von menschlichem Fehlverhalten in die Betrachtungen einzubeziehen. Wichtig ist, schon die Architektur unserer Systeme auf die Begrenzungen von Risiken auszurichten. Nur dann werden wir in Zukunft in der Lage sein, wichtige Daten angemessen zu schützen. Anderenfalls werden auch in Zukunft unsere „Kronjuwelen" durch simple, zu erwartende Fehler gefährdet sein.

Quo vadis, Linux?

Von der Waschmaschine über WLAN Access Points bis hin zu Entertainment-Systemen, von Desktops über Android Smartphones zu Rechenzentren mit Millionen Linux-basierten Server-Systemen: Linux ist als Betriebssystem nicht mehr aus unserem Leben wegzudenken. GNU/Linux hat nach den Jahren der Bekämpfung durch kommerzielle Hersteller das Underdog-Image des Frickler-Systems für Technik-Verliebte abgelegt und wird als vertrauenswürdige, sichere Lösung, nicht nur als Alternative, ernst genommen. Kaum ein Haushalt oder ein Unternehmen kommt ohne Systeme aus, auf denen nicht zumindest ein Linux-Kernel als Schnittstelle zur Hardware zu finden ist. Daraus erwachsen aber auch ganz neue Herausforderungen.



Benutzerfreundlichkeit als Zeichen des Wandels

Kostenersparnis sowie funktionale Aspekte dürften für die Etablierung von Linux als IT-Komponente mehrheitlich entscheidend gewesen sein. Linux-Systeme bieten ein Sammelsurium an Funktionalität, das sie extrem wandlungsfähig und damit fast überall einsetzbar machen. Der Erfolg eines Betriebssystems wird aber davon bestimmt, ob komplexe Technologie einfach und leicht zugänglich aufbereitet, präsentiert und benutzbar gemacht wird. Dies ist eine gänzlich eigene Disziplin wie beim Beispiel Bluetooth: Bluetooth-Tethering ist mit Linux für technisch Findige seit Jahren möglich, wird aber erst vermehrt verwendet, seit Android es bequem per Klick kann. Hier wird sichtbar, dass fortgeschrittene Benutzbarkeit durch einen direkten, monetär bezifferbaren Bedarf motiviert ist. Der Linux-Hobbyist hat Bluetooth-Tethering ja schon vor Android zufrieden verwendet und braucht die Checkbox nicht.

Vertrauenswürdigkeit: nachvollziehbare Annahmen

Auch für die Sicherheitsaspekte der Systeme hat der Bedarf zu Veränderungen beigetragen. Das Bewusstsein der Menschen dafür, dass der Eigentümer von Daten seine Werte schützen muss, ist spätestens seit den Snowden-Veröffentlichungen zum anhaltend brisanten Thema geworden. Nun fußen die Anstrengungen, die IT-Infrastruktur eines Unternehmens gegen Angriffe zu schützen, zwangsläufig auf der Annahme, dass gerade die Komponenten vertrauenswürdig seien, die die Daten tragen, verarbeiten oder weiterleiten: Betriebssysteme, Middleware, Software und Firmware für alle möglichen Hardware-Komponenten. Vertrauenswürdig bedeutet, dass die IT diejenigen Aufgaben verrichtet, die von ihr erwartet werden – und eben keine anderen. Überprüfen lässt sich diese Annahme über die Vertrauenswürdigkeit nur mit der Verfügbarkeit des Sourcecodes, denn nur damit kann die Software auf Fehlverhalten untersucht werden. Diese Transparenz und Nachvollziehbarkeit ist besonders dann unersetzlich, wenn hohe Werte auch großes Angriffspotential erwarten lassen.

Sicherheit als Prozess

Vor diesem Hintergrund sind sich die Linux-Distributoren der Qualität ihrer Produkte wohl bewusst und bauen die funktionierende Kultur dafür aus, wie sicherheitsrelevante Fehler in Software behandelt werden soll. Der jüngst im Februar 2016 veröffentlichte kritische Fehler CVE-2015-7547 im DNS-Code der glibc, einer nahezu überall verwendeten Linux-Bibliothek, zeigt jedoch, dass viele Hersteller von Routern, Appliances und auch dem „Internet-of-things“ noch nicht einmal ansatzweise verstanden haben, dass Sicherheit kein Zustand, sondern ein Vorgang ist: Updates für CVE-2015-7547 stehen oft nicht zur Verfügung, weil die Hersteller mit diesen Erwartungen teilweise überfordert sind. Wer Docker-Images vertreibt oder Android mit zusätzlichen Paketen veredelt, konzentriert sich eben auf seine eigenen Inhalte und nicht um das Machwerk darum herum. Dass potentiell viele tausende WLAN Access Points, Proxies und Appliances verwundbar sind und auch bleiben werden, offenbart einen Blick in tiefe Abgründe. Es mag beruhigend wirken, dass wenigstens einige gute Beispiele existieren. Für die Zukunft steht aber fest, dass Hersteller ihren Produkten und damit auch dem verwendeten Linux deutlich mehr Transparenz und Nachvollziehbarkeit angedeihen lassen müssen. Open Source als Konzept im Allgemeinen und Linux als System im Speziellen müssen sich auch weiterhin an den Werten messen lassen, mit denen sie ihre Vertrauenswürdigkeit gewonnen haben. Für die Anbieter von Lösungen auf Linux-Basis bedeutet das eine neuerliche Orientierung an den Bedürfnissen der Verbraucher – und das Kennenlernen dieser sich wandelnden Bedürfnisse. Zu allem Überfluss sind in dieser Hinsicht die neuerlichen regulatorischen Einschränkungen für den Software-Zugang zu WLAN- und WWAN-Geräten leider maximal kontraproduktiv und können schwerlich zu einer Verbesserung der Sicherheit beitragen.

IMPRESSUM

atsec information security GmbH
Steinstraße 70
81667 München
Deutschland
Telefon: +49-89-442-49-830
Telefax: +49-89-442-49-831
E-Mail: info@atsec.com

Vertretungsberechtigte Geschäftsführer:
Salvatore la Pietra
Staffan Persson
Registernummer: HRB 129439
Registergericht: Amtsgericht München
UST-ID-Nr. gemäß § 27a UStG:
DE205370914
Verantwortlich für den Inhalt:
Staffan Persson (Anschrift s.o.)