Erschienen in: Datenschutz und Datensicherheit 1/ 2012, 43-47

Arnd Weber, Dirk Weber

Verifizierte Virtualisierung für mehr Sicherheit und Komfort

Dr. Arnd Weber ist Volkswirt und hat in Soziologie promoviert. Er ist Senior Researcher beim Institut für Technikfolgenabschätzung und Systemanalyse des Karlsruher Instituts für Technologie. E-Mail: arnd.weber@kit.edu

Dirk Weber ist zertifizierter SAP Technology Associate, Microsoft Certified Systems Engineer und Certified Novell Engineer. Er arbeitete für das Institut für Technikfolgenabschätzung und Systemanalyse des Karlsruher Instituts für Technologie. E-Mail: daw@dawitcon.de

Viren und trojanische Pferde werden auch weiterhin private und geschäftliche PC-Nutzer angreifen. Trojaner können z.B. ein Homebanking-Passwort ausspähen oder vertrauliche Geschäftsdaten an einen Betrüger schicken. Existierende Betriebssysteme bieten keinen ausreichenden Schutz gegen solche Angriffe. Wir präsentieren hier einen Vorschlag, solche Daten durch den Einsatz von Virtualisierungstechniken außerhalb des hauptsächlichen Betriebssystems des Nutzers zu sichern. Dies ermöglicht gleichzeitig allen Nutzern ein einfacheres Management mehrerer Betriebssysteme. Es werden jedoch beträchtliche Anstrengungen nötig sein, wirklich robuste Virtualisierungslösungen herzustellen. Um die Nutzer zu schützen, sollten entsprechende Prototypen und Produkte analysiert und beobachtet werden. Die interessierte Öffentlichkeit könnte diese Produkte fordern. Auf politischer Ebene ist ihre Entwicklung beeinflussbar, z.B. indem Behörden derartige Systeme nachfragen.

Einführung

In diesem Papier geht es um Probleme durch Schadprogramme aus dem Internet, die Computer infizieren. Beispiele hierfür sind Viren, Würmer und trojanische Pferde. Trojanische Pferde geben vor, etwas nützliches zu sein, wie ein Update eines Programms oder eine lesenwerte Email-Anlage, während sie in Wahrheit einen Angriff ausführen, also

z.B. Passwörter oder andere Nutzerdaten sammeln, die sie an den Angreifer zurücksenden [4, 13].

Wir befürchten auch weitere komplexe Angriffe ähnlich dem Stuxnet-Programm, bei denen nicht nur industrielle Produktionsprozesse [3], sondern auch andere Geschäftsprozesse und -geheimnisse angegriffen werden könnten. Derartige Attacken können ernsthafte Schäden für das Opfer haben, sei dies eine Privatperson, eine Firma oder eine öffentliche Verwaltung. Andere Arten von Schadprogrammen, wie Viren, führen jedoch auch zu erheblichen Kosten, etwa für die Arbeitszeit, die die Opfer brauchen, ihre Systeme wiederherzustellen, für den Kauf der üblichen Schutzprodukte und für die Administratoren, die permanent nötig sind, die Systeme zu aktualisieren und nach Angriffen zu säubern, etwa wenn ein neuer Virus zugeschlagen hat, der den existierenden Scannern noch nicht bekannt war. Außerdem muss man in der zukünftigen Nutzung digitaler Signaturen, etwa im eGovernment, damit rechnen, dass Signaturen gefälscht werden, wenn die Signaturumgebung Schwächen hat. Gefälschte Signaturen sind nicht nur eine potentielle Bedrohung für Organisationen, sondern noch viel stärker eine für das Individuum, das signiert oder sich auf eine Signatur verlässt. Ein Beispiel hierfür wäre ein gehackter PC, der falsche Daten an ein sogenanntes Secure Signature Creation Device schickt, wie eine Smartcard.

Ansätze

Es gibt zunächst mehrere Ansätze, solche Probleme anzugehen. Schauen wir kurz ihre Machbarkeit an:

- Zukünftige Versionen von Microsoft Windows könnten die Probleme lösen. Allerdings wird eine perfekte Sicherung von Windows „nicht funktionieren“, wie Paul England von Microsoft es ausdrückte, und zwar aufgrund der Komplexität und der Ausbaubarkeit des Systems [6]. Außerdem werden bei Angriffen oft Schwächen von Anwendungen ausgenutzt.
- Andere Betriebssysteme, wie das Apple Macintosh System oder Linux, sind ähnlich aufgebaut wie Windows und würden wahrscheinlich genauso angegriffen, sobald die Nutzerzahl groß genug ist, dass sich für Kriminelle der Aufwand lohnt.
- Wieder ein anderer Ansatz ist, Computer komplett neu zu designen. In der Praxis wäre so ein System jedoch wenig nützlich, weil existierende Anwendungen und Daten nicht genutzt werden könnten.
- Ein weiterer Ansatz besteht darin, für sicherheitskritische Aktivitäten eine separate Maschine zu benutzen. Das mag mit kleinen Geräten, wie Smartcard-Lesern mit Display funktionieren, und bei militärischen Anwendungen auch mit größeren. Der Ansatz ist jedoch teuer und umständlich (genauso wie das Booten eines anderen Betriebssystems). Im Übrigen können auch Smartcard-Leser gehackt werden [19]. Und wenn der Hersteller verlangt, dass er nur in einer sicheren Umgebung benutzt werden darf, ist das Risiko nicht beseitigt, sondern nur verringert.

- Schließlich gibt es Hypervisor. Diese stellen eine weitere Softwareschicht unterhalb der Betriebssysteme dar. In einem derartigen System kommuniziert ein Betriebssystem nur scheinbar mit der Hardware, d.h. es kommuniziert nur mit der virtuellen Hardware, die vom Hypervisor zur Verfügung gestellt wird. Damit kann man grundsätzlich potentiell riskante Aktionen, wie das beliebige Surfen oder das Lesen von Email-Anlagen außerhalb des normalen Betriebssystems erledigen, d.h. in einem anderen Compartment mit einem weiteren Betriebssystem. Heutige Produkte sind jedoch nicht so gebaut, dass sie Laien eine robuste Isolierung bieten, die einen Angriff eines Schadprogrammes auf ein anderes Compartment zuverlässig verhindert.

Damit bieten alle diese Lösungsansätze nicht den Schutz, der nötig ist. Um eine wirklich sichere Lösungen zu erreichen, die für existierende Anwendungen geeignet ist, sollten Hypervisor so weiterentwickelt werden, dass sie einen zuverlässigen Schutz von Compartments bieten, selbst wenn sich in einem Compartment ein Schadprogramm befindet. Damit wird es möglich, dubiose Emails zu isolieren, beliebige Websites zu besuchen oder neue Programme, Betriebssysteme oder Treiber zu testen. Ggf. kann ein infiziertes Compartment gelöscht und re-installiert werden. Damit derartige Systeme von durchschnittlichen Nutzern akzeptiert werden, müssen sie so designed werden, dass man sie leicht und zuverlässig benutzen kann.
Die Autoren nahmen an einem Forschungsprojekt teil, in dem Prototypen eines solchen Systems gebaut wurden, aufbauend auf existierenden Hypervisorn. Es handelt sich um das Open Trusted Computing-Projekt, das zwischen 2005 und 2009 durchgeführt wurde.[1]

Unser Ziel

Unser Ziel ist, dass Nutzer und Administratoren frei Compartments kreieren können, die jedes Betriebssystem beinhalten, das sie haben möchten. Policies und Rechte können gesetzt werden, wie es angemessen ist, z.B. kann eine Firma Rechte an einigen Compartments erhalten, während Nutzer die vollen Rechte an anderen bekommen. Die Idee ist, dass ein Privatnutzer einen Hypervisor für jeden Zweck verwenden kann, vom Isolieren sensibler Daten bis zum Betrachten riskanter Programme.

Ein Beispiel: Ein Arbeitgeber ist Eigentümer eines Laptops. Der Angestellte erhält volle Rechte in Bezug auf den Hypervisor, aber nicht in Bezug auf jedes Compartment. Die Firma erhält volle Rechte in Bezug auf ihr Compartment und verlässt sich insofern auf den Hypervisor. Der Angestellte kann das ganze Firmen-Compartment löschen, aber die Rechte und Policies dieses Compartments nicht ändern. Derselbe Mechanismus könnte für Digital

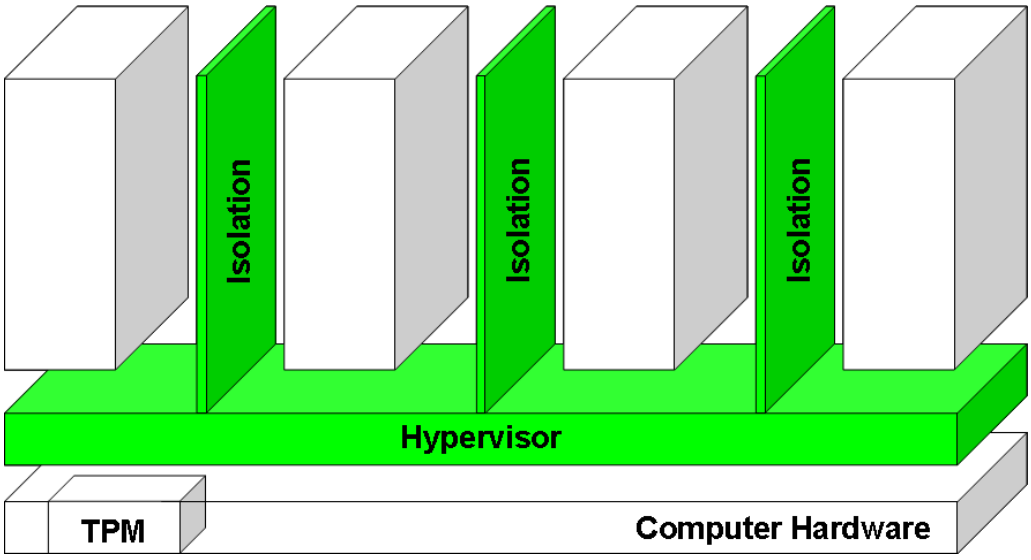
[1] Das Open Trusted Computing-Projekt wurde von der Europäischen Kommission unterstützt (Projekt IST-027635).

Rights Management (DRM) genutzt werden und auch für Homebanking oder um private Daten vertraulich zu halten.

Die Handhabung der Betriebssysteme sollte so gestaltet sein, dass z.B. die Installation neuer Versionen, die Wiederherstellung eines vorherigen Standes oder die Migration eines Compartments von einem Rechner auf einen anderen einfach durchzuführende Aktionen werden.

Unsere These ist, dass ein sicherer Hypervisor für jeden nützlich ist, sei es wegen der Flexibilität, der Geschwindigkeit, der Bequemlichkeit oder der Sicherheit. Also wäre ein Hypervisor nötig, der (1) die herkömmlichen Eigenschaften von Hypervisorn hat, (2) die Fähigkeit, robust zu isolieren und (3) eine Nutzerschnittstelle, die von Laien genutzt werden kann. Um wirklich diese Ziele zu erreichen, muss ein Hypervisor also Schutz gegen Schadprogramme bieten, in Open Source angeboten werden (damit er keine Hintertüren enthält) und auf geeigneter Hardware laufen.

Abb 1: Die Architektur eines sicheren Hypervisors im Überblick.



Computer mit einem Hypervisor (grau), der Isolation zwischen den Betriebssystem-Compartments bietet (weiß). TPM = Trusted Platform Module

Mit einem solchen System kann es dann zur Routine werden, Compartments für Folgendes zu benutzen:

- für Betriebssysteme wie heutzutage
- für DRM
- zur Verschlüsselung von Daten
- für Banking und eCommerce, evtl. mit einem Browser des Anbieters

- für private Daten
- zum beliebigen Surfen im Internet
- für Spiele
- zur Erzeugung und Überprüfung digitaler Signaturen
- als “Sandkiste” zum Testen von nicht bekannten oder nicht vertrauenswürdigen Programmen

Es wäre dann sinnvoll, wenn eine gewisse Anzahl von Kanälen zur Kommunikation zwischen den Compartments im Voraus spezifiziert würde, unter denen der Nutzer wählen kann. Darüber hinaus können Compartments migriert werden, etwa von einem Laptop auf einen USB-Stick, auf einen PC oder zu einem Cloud Computing-Anbieter. Der Ansatz der Trusted Virtual Domains (TVDs) [2] kann in einem solchen System genutzt werden, um mehrere Anwendungen auf mehreren Computern zu handhaben.

Architektur und Trusted Computing

Um das Ziel zu erreichen, sind ein evaluierter Open Source-Hypervisor nötig, sowie gegen Manipulationen geschützte Hardware, um den Hypervisor vor Veränderungen von außen zu bewahren.

Ein Beispiel für solche Hardware ist der Trusted Computing Chip (TPM), der dazu benutzt werden kann, Komponenten beim Booten zu überprüfen, d.h. zu messen („chain of trust“, vgl. [9]). Damit kann vor allen Abweichungen vom sicheren Pfad gewarnt werden, z.B. können gefälschte Updates blockiert werden. Hierzu ist ein mit einer Public Key Infrastruktur gesicherter Update-Pfad nötig. Abbildung 1 zeigt die Architektur eines solchen Systems im Überblick.

Eine wesentliche Charakteristik eines solchen Systems ist, dass der Eigentümer des Hypervisors alle Rechte in Bezug auf die Hardware und den Hypervisor behält und z.B. eine nicht richtig funktionierende Fassung eines Betriebssystems oder ein ungewolltes DRM-System löschen kann, ohne dass Reste im Hypervisor bleiben. Auch der Eigentümer eines Compartments hat alle Rechte an diesem. Die beiden Eigentümer müssen aber nicht identisch sein. Wenn ein Compartment nicht von seinem Eigentümer, sondern vom Eigentümer des Hypervisors geändert wird, funktioniert es evtl. nicht mehr.

Wir denken, dass Trusted Computing zusammen mit Virtualisierung nützlich für Individuen oder Firmen ist, die überprüfen möchten, ob ihre Computer im ihnen bekannten richtigen Zustand sind. Darüber hinaus möchten Firmen evtl. überprüfen, ob ein Computer, der Zugang zu ihrem Netzwerk erhalten möchte, richtig eingerichtet ist. Das können sie mit der Funktion der „remote attestation“ erreichen. Oft wurde angenommen, (1) dass beim Trusted Computing die Hersteller der Hardware Garantien aussprechen müssen, (2) dass es notwendig wäre, eine lange Liste an guten Messwerten sich dauernd ändernder

Komponenten zu haben und (3) dass eine weltweite Public Key Infrastruktur zur Benutzung von Trusted Computing nötig wäre. Auf diese umfangreiche Infrastruktur könnte allerdings verzichtet werden, wenn der Käufer des Computers dem Verkäufer vertraut und ein privates System aufsetzt und so alle seine Maschinen überprüft. Auf diese Art könnte sich Trusted Computing allmählich entwickeln. Die Überprüfung der Messwerte kann in einem späteren Stadium gemeinsam mit dem Hersteller erfolgen, Messwerte und Schlüssel könnten mit Geschäftspartnern ausgetauscht werden und so würde allmählich ein globales System entstehen.

Nutzerschnittstelle

Im Open Trusted Computing-Projekt wurden mehrere Prototypen gebaut. Es wurde gezeigt, dass ein Open Source-Hypervisor mit Microsoft Windows arbeitet und das Trusted Platform Module als Sicherheits-Anker verwendet werden kann, um das System zu überprüfen.

Im Rahmen des Projektes haben die Autoren zentrale Teile einer Nutzerschnittstelle entwickelt. Unsere Ausgangsfrage war: Ist es möglich, eine Nutzerschnittstelle so zu gestalten, dass sie durch Laien genutzt werden kann? Eine Herausforderung ist, dass die Nutzer verstehen müssen, dass es Programme außerhalb ihres Betriebssystems gibt. Eine weitere Herausforderung bestand darin, dass ein Teil des Bildschirms benutzt werden muss, um die Nutzer über die neue Schicht und die neuen Programme zu informieren, was die Nutzbarkeit vorhandener Anwendungen reduzieren könnte.

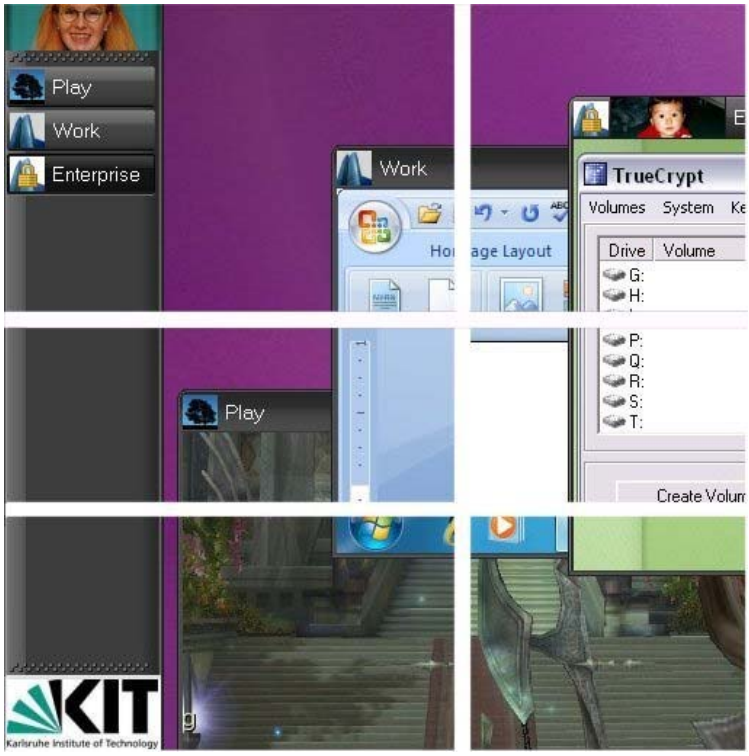


Abbildung 2: Ausschnitte aus einer möglichen zukünftigen Nutzerschnittstelle.

Darüber hinaus entstanden weitere Fragen für das Design einer solchen Schnittstelle: Wie kann man die Eigentümer gegen trojanische Pferde schützen, die z.B. in der Form eines Pop-up Fensters vorgeben, ein Update zu sein und nach einem Passwort fragen oder in vertrauliche Bereiche gehen wollen? Sollte eine physisch getrennte Anzeige, etwa auf der Tastatur, über den Status eines Compartments informieren? Sollte eine neue Taste eingeführt werden, um zwischen Compartments zu wechseln, was den Vorteil hätte, dass jedes Compartment den ganzen Bildschirm zur Verfügung hätte? Wir diskutierten diese Themen in einer kleinen Befragung von CTOs und Chef-Administratoren im Jahr 2006 in Deutschland [23]. Die Antworten waren, in Kürze:

- Der Hypervisor sollte eine einfache graphische Nutzerschnittstelle haben und z.B. Buttons verwenden, die man mit der rechten oder linken Maustaste anklicken kann. Weder Administratoren noch Nutzer wollen Zeit aufwenden, neue Schnittstellen zu lernen.
- Das Umschalten zwischen Compartments sollte so einfach sein wie das Umschalten zwischen Anwendungen, also z.B. mit Mausklick.

Das waren die Argumente, mit denen unsere Interviewpartner separate Displays und Tasten abgelehnt haben, wie sie vorher in der Literatur zur sicheren Virtualisierung vorgeschlagen worden waren.

Abbildung 2 zeigt eine Taskbar ähnlich einer heutigen, mit der man die verschiedenen Compartments handhaben kann. Diese neuartige Taskbar würde einen sicheren Teil des Bildschirms nutzen, um eindeutig anzuzeigen, ob ein Compartment vertrauenswürdig ist. Ein „versiegeltes“ Bild wird als Schutz des Hypervisors gegen Mimikry verwendet. Ein solches Bild wird nur angezeigt (entschlüsselt), wenn der Hypervisor korrekt gemessen wurde. Die Taskbar zeigt auch Tasten für ein Firmencompartment (hier mit Linux) sowie Tasten für andere Compartments, z.B. ein Spiel. Das Schloss zeigt an, dass ein Compartment beim Start erfolgreich gemessen wurde. Weitere versiegelte Bilder zeigen an, dass weitere Compartments korrekt gestartet wurden. Die Bilder können einer Familie entstammen, wie Eltern und Kinder, wobei ein weiteres Bild nur gezeigt wird, wenn das vorangegangene korrekt war. Dadurch werden Gruppen von Compartments möglich. Die hier vorgeschlagenen Elemente benötigen sehr wenig Platz, so dass die Anzeige der normalen Programme kaum verändert wird. Wie in Abbildung 2 gezeigt, könnte die neue Taskbar bei breiten Bildschirmen an einer Schmalseite angezeigt werden. Wenn sie nicht benutzt wird, könnte sie sogar verschwinden. Die Nutzbarkeit unserer Schnittstelle wird in einer Videoaufnahme des Prototypen demonstriert, die wir auf www.open-hypervisor.org zeigen.

Feedback

Einige Zitate aus 15 Experteninterviews, die wir 2009 in Deutschland, Großbritannien und den USA gemacht haben, illustrieren die Nützlichkeit unseres Ansatzes:

- „Man müsste nicht alles Abblocken, die Angst würde reduziert."
- „Die Kapselung von normaler Bürosoftware, getrennt von Software für Experimente und den mobilen Einsatz" wäre gut.
- „Man könnte eine Quarantäne einrichten."
- „Es könnte infrage kommen, besonders wichtige Daten besonders zu schützen."
- "Sicherheitsbeauftragte lecken sich die Finger nach so einer Lösung."
- "The ability to trash an operating system instance is relevant, in case of suspicious behaviour, for getting back to the latest state."
- "Isolation of highly confidential data is an interesting idea."
- "It could be used against zero-day attacks."

Dieses Feedback ermutigt uns, nach Wegen zu suchen, eine solche Lösung in Produktqualität zu erreichen.

Fortschritt

Teile unseres Ansatzes sind seit längerem bekannt [1, 9, 17]. Im Open Trusted Computing-Projekt wurden mehrere Prototypen mit existierenden Betriebssystemen für einige wenige PC-Typen gebaut. Prototypen für DRM und digitale Signaturen ("What you see is what you sign") wurden ebenfalls entwickelt. Mehrere wichtige Komponenten wurden evaluiert und Fehler behoben[2] [12, 15, 22]. Ein Prototyp eines Hypervisors aus der Forschung ist jedoch weder eine komplette Plattform, noch ein fertiges Produkt. Es gibt inzwischen auch noch andere Prototypen und sogar Produkte. Einige Produkte messen beim Booten (TVE [7]), andere haben evaluierte, aber proprietäre Kerne (Integrity [10]). Kein Produkt hat einen Hypervisor, der erwiesenermaßen Schadprogramme isoliert, keines verwendet bewiesene Isolation in der Hardware (kritische Anmerkungen über die Qualität der Isolierung in Intel TXT fanden sich in [20]). Kein Produkt verwendet eine Nutzerschnittstelle, die gegen Mimikry schützt. Neue Ansätze, Handys [11] oder Netbooks zu sichern (mit Chrome [8]) zeigen in die richtige Richtung, unterstützen aber keine vorhandenen PC-Anwendungen.

Der Ansatz der Virtualisierung unter Nutzung sicherer Hardware kann auch für Server, mobile Geräte oder Embedded Systems verwendet werden. Ähnlich wie die „remote attestation" für DRM genutzt werden könnte, ließe sie sich benutzen, um Daten auf Cloud Computing Servern zu schützen.

Zum Zeitpunkt der Manuskripterstellung ist nicht sicher, dass ein Gesamtsystem gebaut werden kann, das sowohl gegen jede Art von Schadprogrammen schützt und auch für vorhandene Anwendungen offen ist. Wir glauben jedoch, dass dieses Papier einen nützlichen Pfad zur Entwicklung von Hard- und Software beschreibt, dem die Industrie folgen kann. Wir

[2] Technische Details sind auf www.opentc.net verfügbar. Als Einstieg können der Schlussbericht oder die Ausgaben des Newsletters dienen.

möchten noch einmal betonen, dass das Ziel nicht allein darin besteht, an Sicherheit interessierten Nutzern zuverlässige Isolation zu bieten, sondern auch darin, allen Nutzern ein einfaches Management von Betriebssystemen zu ermöglichen.

Um diesen Ansatz von der Idee zur Realität zu bringen, sehen wir gegenwärtig die folgenden Wege zur Entwicklung und Verbreitung einer solchen Lösung:

(1) Allmähliche, privat finanzierte Verbesserungen.

(2) Ein großes Projekt, das von Regierungen finanziert wird und, basierend auf einer Analyse von Bedrohungen, eine Spezifikation von Hard- und Software entwickelt und implementiert. Dies würde die Gemeingut-Eigenschaft der Lösung berücksichtigen, d.h. die Kosten über viele Nutzer verteilen.

(3) Anreize durch Regierungen, wie Regulierungen oder entsprechende Beschaffungen. Dies geschah ähnlich bei der Schaffung von „Trusted Computing“, in Folge des Sarbanes-Oxley Act in den USA oder durch die Einführung des PCI-DSS Standards (Payment Card Industry Data Security Standard).

(4) Eine globale Diskussion dieses Pfades wird Nachfrage schaffen. Wir denken an Entwicklungen wie in der Automobilindustrie, die durch Bücher wie “Unsafe at Any Speed” angespornt wurde, die Sicherheit ihrer Fahrzeuge zu erhöhen [14]. Wir stellen uns auch vor, dass große Industriekunden Updates in Richtung auf unser Ziel nachfragen. Newsletter oder unsere Website http://open-hypervisor.org/ können als Schritte gesehen werden, Nachfrage zu erzeugen.

Danksagungen

Wir danken Dirk Kuhlmann, Armand Puccetti und Matthias Schunter.

Literatur

[1] Arbaugh, W., Farber, D. und Smith, J. *A Secure and Reliable Bootstrap Architecture.* Proceedings of the 1997 IEEE Symposium on Security and Privacy: 65-71.

[2] Catuogno, L., Löhr, H., Manulis, M., Sadeghi, A.-R., Stüble, C. und Winandy, M. *Trusted Virtual Domains: Color Your Network.* Datenschutz und Datensicherheit: 5/2010. 289-294. http://www.springerlink.com/content/r8g9u60847w5g72r/fulltext.pdf.

[3] CNET news November 17, 2010: *Symantec to Congress: Stuxnet is 'wake-up call'.* http://news.cnet.com/8301-27080_3-20023124-245.html.

[4] Dalton, C. *A Hypervisor Against Ferrying Away Data.* Interview von Furger, F. und Weber, A. OpenTC Newsletter, April 2009. http://www.opentc.net/publications/OpenTC_Newsletter_07.pdf. http://www.itas.fzk.de/deu/lit/2009/webe09b.htm.

[5] Dalton, C., Plaquin, D., Weidner, W., Kuhlmann, D., Balacheff, B. und Brown, R.: *Trusted Virtual Platforms: A Key Enabler for Converged Client Devices.* In: Newsletter ACM SIGOPS Operating Systems Review Volume 43 Issue 1, January 2009, 36-43.

[6] England, P. *Practical Techniques for Operating System Attestation.* Vortrag auf: Trusted Computing - Challenges and Applications, First International Conference on Trusted Computing and Trust in Information Technologies, Trust 2008, Villach, Austria, March 11-12, 2008.

[7] GENERAL DYNAMICS. *TVE for Desktops and Laptops.* 2011. http://www.gdc4s.com/content/detail.cfm?item=35a995b0-b3b7-4097-9324-2c50008b3a75.

[8] GOOGLE: *Security Overview.* http://www.chromium.org/chromium-os/chromiumos-design-docs/security-overview

[9] Grawrock, D. *The Intel Safer Computing Initiative.* Intel Press, 2006.

[10] GREEN HILLS. *Integrity Real-time Operating System.* 2011. http://www.ghs.com/products/rtos/integrity.html.

[11] Heiser, G., Andronick, J., Elphinstone, K., Klein, G., Kuz, I. und Leonid, R. *The Road to Trustworthy Systems.* Communications of the ACM, 53(6), 107–115, June, 2010.

[12] Kuhlmann, D., Weber, A. *The Evolution of the OpenTC Architecture Illustrated via its Proof-of-Concept-Prototypes.* OpenTC Final Report. Bristol, Karlsruhe 2009, http://www.opentc.net/.

[13] MI5: *Espionage*. http://www.mi5.gov.uk/output/espionage.html.

[14] Nader, R. *Unsafe at Any Speed.* Grossman Publishers, New York 1965.

[15] OPENTC. Projekt Website. http://www.opentc.net/.

[16] OPENTC. Projekt Newsletter, verfügbar auf www.opentc.net.

[17] Pfitzmann, B., James, R., Stüble, C., Waidner, M. und Weber, A. *The PERSEUS System Architecture.* IBM Research Report RZ 3335, IBM Research – Zurich, April 2001. http://www.zurich.ibm.com/security/publications/2001.html.

[18] Ristenpart, T., Tromer, E., Shacham, H. und Savage, S. *Hey, You, Get off of My Cloud! Exploring Information Leakage in Third-Party Compute Clouds.* Proc. ACM Conference on

Computer and Communications Security 2009, 199-212, ACM, 2009. http://people.csail.mit.edu/tromer/Ristenpart_cloudsec.pdf.

[19] SECORVO: *Security News,* Juni 2010. http://www.secorvo.de/security-news/secorvo-ssn1006.pdf.

[20] Seifert, J.-P. Keynote Presentation auf: Computers, Privacy and Data Protection, Brussels 2010.

[21] Serdar, C., Dalton, C., Eriksson, K., Kuhlmann, D., Ramasamy, H., Ramunno, G., Sadeghi, A.-R., Schunter, M. und Stüble, C. *Towards Automated Security Policy Enforcement in Multi-Tenant Virtual Data Centers.* Journal of Computer Security, IOS Press, Vol. 18, Number 1, pp. 89-121, 2010.

[22] Weber, A., Weber, D.: *Options for securing PCs against phishing and espionage.* A report from the EU-project "Open Trusted Computing". In: Gutwirth, Serge et al. (Hrsg.): Proceedings of CPDP 2010, Brussels. Springer, 2011. 201-207. http://www.springerlink.com/content/t067038412352321/.

[23] Weber, A., Weber, D. und Lo Presti, S. *Requirements and Design Guidelines for a Trusted Hypervisor User Interface.* Vortrag auf: Future of Trust in Computing. Berlin, Germany, 30 June – 2 July, 2008. Proceedings veröffentlicht von Vieweg & Teubner, Wiesbaden 2009.