Arnd Weber, Dirk Weber

Verified Virtualisation for more Security and Convenience

Paper published in German as:

Verifizierte Virtualisierung für mehr Sicherheit und Komfort. In: Datenschutz und Datensicherheit 1/ 2012, 43-47

Dr. Arnd Weber is an economist with a PhD in sociology. He is senior researcher with the Institute for Technology Assessment and Systems Analysis of Karlsruhe Institute of Technology. E-Mail: arnd.weber@kit.edu

Dirk Weber is a certified SAP Technology Associate, Microsoft Certified Systems Engineer and Certified Novell Engineer. He worked for the Institute for Technology Assessment and Systems Analysis of Karlsruhe Institute of Technology. E-Mail: daw@dawitcon.de

Private and business PC users will continue to experience attacks from viruses and Trojan horses. The latter might, e.g. eavesdrop on banking passwords or send confidential business data to a criminal. It is very difficult to provide protection against such attacks on private information within the current operating systems. We present a vision for securing such data outside the user's main operating system by using virtualisation techniques. Considerable efforts will, however, need to be made to achieve truly trustworthy solutions. In order to protect users, the development of such approaches could be monitored, observed and pushed by the interested public, and influenced at the political level, e.g. by governments procuring such systems.

Introduction

This paper addresses the problem of malicious code from the Internet infecting computers. Examples of malicious code include viruses, worms, and Trojan horses. Trojan horses typically appear to be something useful, e.g. an update of a program or an email attachment worth reading, while in reality they also perform an attack, such as by collecting passwords or other user data and reporting these back to the attacker [4, 13]. We are also thinking of malware as sophisticated as Stuxnet, attacking not only industrial production processes [3], but also targeting other business processes and secrets. Such attacks can have severe consequences for the individual victim, whether this is a private person, a company or a government institution. However, other types of malicious code, such as viruses, also lead to significant costs, e.g. in terms of the labour needed by the victim to restore systems, the expenses necessary for purchasing the usual means of protection, or for the administrators who are continuously necessary to update systems and clean up the systems after an attack has taken place, e.g. from a new virus not yet known to the existing protection tools. The future use of digital signatures, for instance in eGovernment, will have to confront faked signatures due to weaknesses in the signing environment. Fake signatures will form a potential threat to organisations, but more so to the individual who signs or relies on the signature being valid. An example would be an attacked PC sending fake data to a so-called Secure Signature-Creation Device, such as a smart card.

Approaches

There appear to be various ways to address these problems. Let us quickly review their feasibility:

- Future updates of Microsoft Windows might address this problem. However, completely securing Windows “isn’t going to work”, as Paul England of Microsoft put it, due to the complexity and desired extensibility of the system [6]. Also, attacks often exploit weaknesses of applications.
- Other operating systems, such as the Apple Macintosh system or Linux, are similar to Windows and therefore would probably also be attacked in a similar manner as soon as the user base is large enough for criminals to consider attacks worthwhile.
- Yet another approach would be to totally redesign computers from scratch. However, in practice such a system would not be very useful as existing user applications and data could not be used on such a system.
- Another approach would be to use physically separate machines, and only use them for certain security-critical actions. While this works for small devices such as smartcard readers with a display and may also work for certain applications such as military ones, these solutions are costly and inconvenient, just like rebooting a different operating system. Also, if the smart card reader can be hacked [19] or if the vendor demands that

the user only operate it in an environment free of malware, the risk is only marginally reduced.

- Existing hypervisors have not been designed to provide bullet-proof isolation to laypersons.

We believe that only one solution is viable, namely to develop a system which allows work with existing operating systems to continue while isolating other applications, be they security-critical or risky ones. In other words, all applications would run in sandboxes which are designed in such a way that malicious code cannot spread from one to another. Such sandboxes, or compartments, could isolate particularly sensitive data as well as new, secure applications. Secure applications would then run with their own, special purpose operating system. However, the sandboxes could also be used to isolate potentially malicious applications. For example, dubious emails could be isolated, arbitrary websites could be visited, or new software, operating systems, or drivers could be tested. If necessary, an infected compartment could be deleted and reinstalled from scratch. In order to have isolated compartments and manage them, a new layer would be needed, running "beneath" the operating systems. Such a layer is called a hypervisor or a virtual machine monitor. In such a system, an operating system would no longer communicate with the hardware, but only with a virtual hardware layer, provided by the hypervisor, hence one speaks of virtualisation. The authors participated in a research project which aimed at building a prototype of such a system by improving existing hypervisors. This was the Open Trusted Computing project which ran from 2005 to 2009.[1]

Such new systems, to be accepted by the average user, have to be designed to be used in a convenient way. A user does not want to take care of or be bothered with security features to protect himself against malware. Instead, the user wants a reliable system to work with.

Our Vision

Our vision is that users and administrators can freely create compartments containing any operating system they choose. Policies and rights to compartments can be set as appropriate, e.g. an employer can bind rights to some compartments while users get full rights to others. The basic idea is that a private user getting a hypervisor uses it for any purpose, from isolating sensitive data to viewing risky data.

Example: A company owns a laptop computer. The employee has full rights regarding the hypervisor but not regarding every compartment. The company itself has full rights for one compartment relying on the hypervisor. The employee can delete the whole company compartment, but not change its rights or policies. The same mechanisms could be used for

[1] The OpenTC-project was supported by the European Commission (project IST-027635).

digital rights management (DRM) but also, if useful, for banking and for keeping private data confidential.

The operating systems handled by the hypervisor in its compartments have to be designed to be easy to manage, e.g. regarding new versions, restore to previous state, hardware migration and usage of several operating systems in parallel.

Our thesis is that a secure hypervisor is efficient for everybody, be it because of flexibility, speed or convenience, or be it because of the security, i.e. the isolation property of its compartments. Hence a hypervisor is needed which supports (1) traditional features of hypervisors and (2) isolation-related features and which has (3) a user interface usable by laypersons. To really provide security features needed to protect, e.g. business secrets or private passwords, a malware-protected, open source (backdoor-free) hypervisor is required which needs to rely on appropriate hardware.

The following use of compartments could then become daily routine:

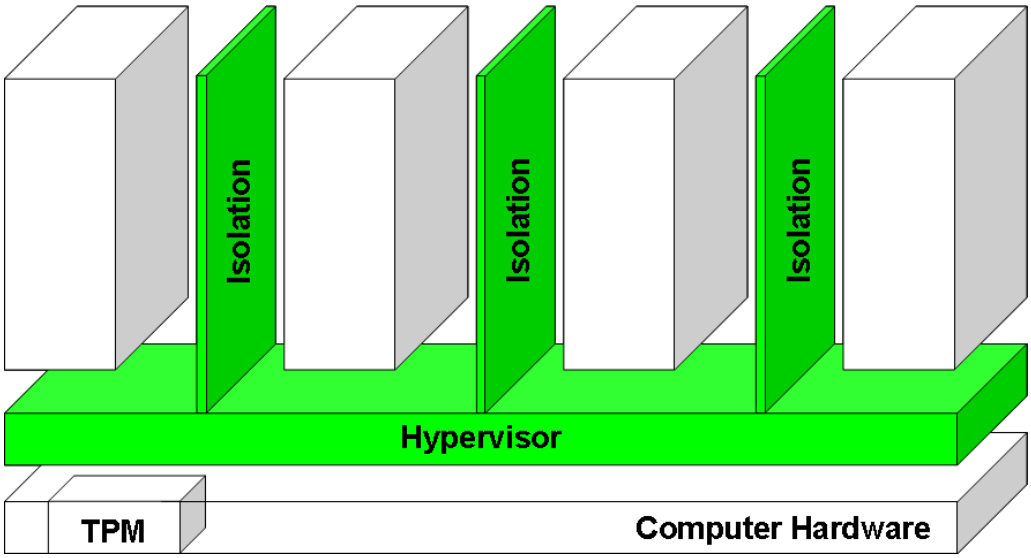
- Today's system
- DRM systems for protecting IPR
- Encrypted data
- Auctioning, banking and other eCommerce transactions (possibly with a provider-specific browser)
- Private data
- Unrestricted Internet surfing
- Games
- Digital signatures: signing and verifying documents
- Testing and sandboxing, including any malware

A small number of channels between the compartments may have to be pre-specified for selection by the user. Migration of compartments is possible, such as from a laptop via a USB stick to a home PC or to a cloud provider. One can use the approach of trusted virtual domains (TVDs) [2] on such as system in order to manage several applications across several machines.

Architecture and Trusted Computing

To reach this goal most likely both an evaluated open source hypervisor and tamper-resistant hardware are needed to protect the hypervisor against manipulation from outside. An example of such hardware is a Trusted Computing chip (TPM), which can be used to measure components when booted (chain of trust, cf. [9]) in order to block fake updates or to alert any behaviour straying from the secure path. This requires a PKI-based secure update path. Figure 1 shows the architecture of such a system at the highest level.

Figure 1: High level architecture of a secure hypervisor.



Computer with a hypervisor (green), providing isolation to the operating system compartments (white).

A key characteristic of the system would be that the owner of the hypervisor keeps all rights regarding the hardware and the hypervisor itself, and can, for example, delete a misbehaving copy of an operating system (e.g. Windows) or of an unpleasant DRM system unconditionally and without any remainders in the hypervisor. Also the owner of the compartment keeps all rights to the compartment itself. Both owners do not have to be the same. So if a compartment is not altered by its owner but by the owner of the hypervisor, it might simply cease to function.

We think that Trusted Computing used with virtualization is useful for individuals or corporations wishing to check whether their computers are in a known good state. Also, corporations might wish to check whether a computer requesting access to their Intranet has been properly set up (using remote attestation). It has often been assumed, for example, that, with Trusted Computing, the hardware vendors need to provide some guarantee about their computers, that it would be necessary to keep track of long lists of good values for ever-changing hardware components, or that a global public key infrastructure is needed for using Trusted Computing. None of these is necessary if the purchaser of a computer trusts his vendor and sets up a private system. That buyer can verify all of his or her own machines. Trusted Computing could then evolve. Maintenance of these good values could be outsourced to a computer vendor, and values and keys could be exchanged with business partners, so that ultimately a global system might emerge.

User Interface

Several prototypes were built by the Open Trusted Computing project. It was shown that an open-source hypervisor works with Microsoft Windows and can use the Trusted Platform Module as a hardware security anchor to monitor the system.

In the framework of the project, the authors developed key aspects of a user interface. Our starting questions were: Is it possible to design a user interface for this kind of system so that it can be used by laypersons? The challenge is based, for instance, on the fact that users need to understand that there are programs outside their usual operating system. Furthermore, the task is challenging as part of the screen space would be needed to inform the user about the new layer and the new programs, which might reduce the usability of legacy applications.

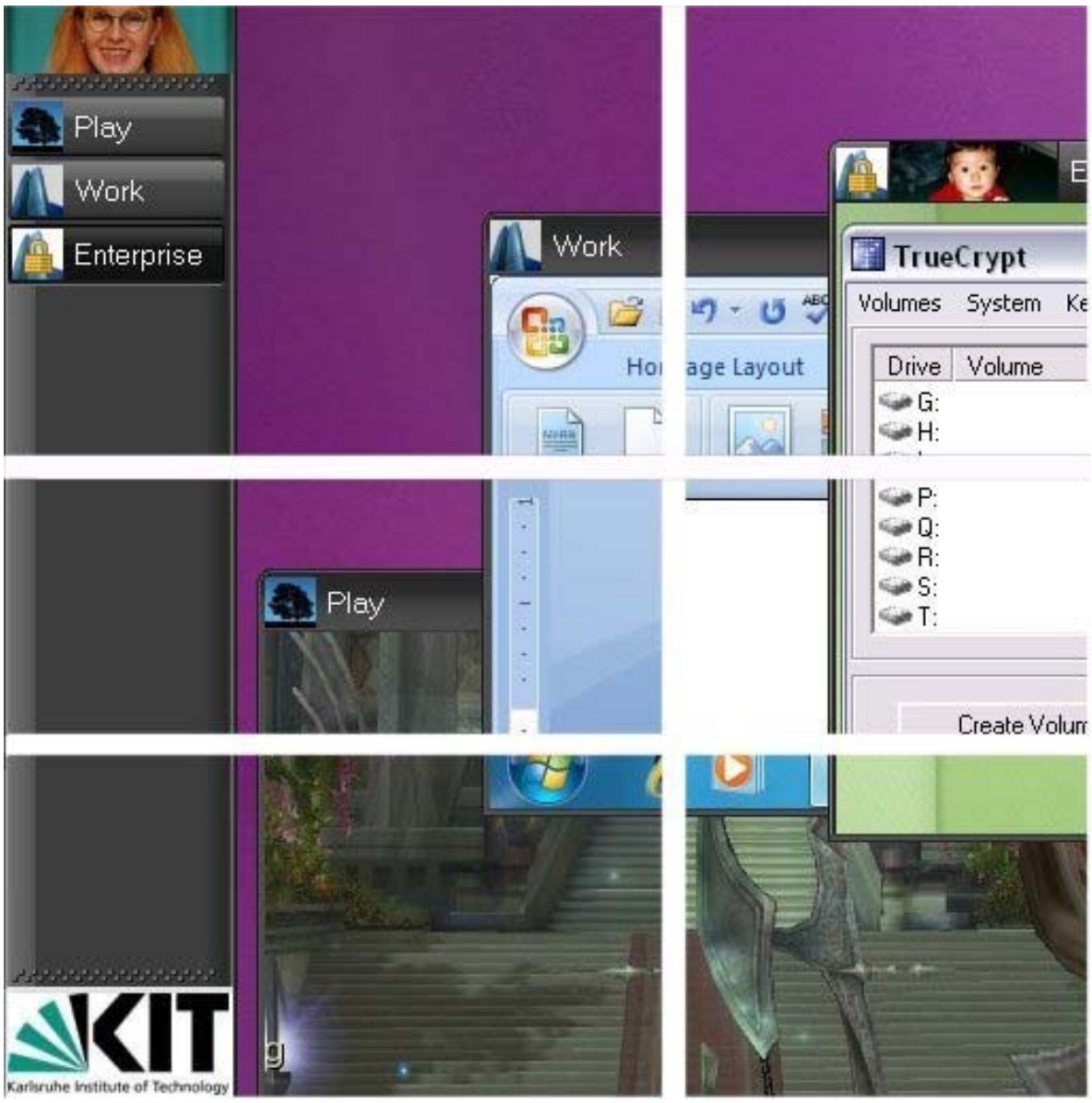


Figure 2: Sections from a possible future user interface.

Furthermore, questions emerged for the design of such a user interface such as: How can the owner be protected against Trojan horses in the form of, e.g. pop-up windows claiming to represent a software update and asking for passwords or linking into confidential areas? Should a physically separate display provide information about the state of a compartment? Should a new hardware key on the keyboard be used to switch between compartments, which would also allow each compartment to obtain access to the full screen? We discussed such questions in a small in-depth survey of CTOs and leading administrators in Germany in 2006 [23]. The answers were, in short:

- The hypervisor should have a simple graphic user interface (GUI), e.g., with buttons using left and right mouse clicks. Neither administrators nor users want to spend time learning how to use new computer interfaces.
- Switching between compartments should be as simple as switching between applications today, e.g. using a mouse click.
- Our respondents thus discarded separate displays and buttons as previously discussed in the virtualisation literature.

Figure 2 shows that a taskbar similar to today's could be used to manage the various sandboxes. This novel taskbar would use a secure part of the screen in order to unambiguously display whether the hypervisor or certain sandboxes are trustworthy. A sealed image is shown there as a means of protecting the hypervisor against mimicry by malicious code. The image is only displayed if the hypervisor was measured correctly. The taskbar also shows buttons for a corporate or business compartment (running with Linux), and buttons for switching to other compartments for editing documents or gaming. The lock symbol indicates that one of the compartments is measured upon start. It uses a different image than the hypervisor, this one unsealed and which indicates its good state. Using images from a family, such as of a parent and child, symbolises that the latter relies on the former and eases identifying groups. Note that the proposed amendment to the user interface requires very little screen space, so that the user interface available for normal work is hardly changed. As shown in Figure 2, for use on wide screens, the novel taskbar could be placed on a narrow side; it could even be hidden if not in use.

The usability of our user interface is indicated by a video of real code, available at www.open-hypervisor.org.

Feedback

Some quotations from 15 expert interviews, conducted in 2009, illustrate the benefits of our approach:

- With virtualisation, corporations "could allow their employees more surfing and private use."
- "Isolation of highly confidential data is an interesting idea."
- "The ability to trash an OS instance is relevant, in case of suspicious behaviour, for getting back to the latest state."
- "Encapsulating the office network from user-administrated test environments and experimental software" (is beneficial)
- "Quarantining could reduce inactive periods."
- The hypervisor "could be used against zero-day attacks."
- "Security officers would give an arm and a leg for such a solution."

This feedback encourages us to strive for ways to make such a solution available in product quality.

Progress

Parts of our approach have in principle been known for some time [1, 9, 17]. In the OpenTC project, several prototypes with legacy operating systems running on a few hardware platforms were developed. Prototypically, solutions for DRM and "What you see is what you sign" have been developed as well. Several key modules were evaluated, and any errors detected were corrected[2] [12, 15, 22]. However, a research prototype of a hypervisor is neither a final product nor an entire platform. There are various other research prototypes and even products. Some of them have Trusted Boot (TVE [7], and others have evaluated kernels but are proprietary (Integrity, cf. [10]). Yet none has a hypervisor that provenly provides isolation or that provenly supports hardware isolation ([20] was critical about the quality of the Intel TXT isolation), or a user interface protecting a user against mimicry by Trojan horses. New approaches to secure mobile phones [11] or netbooks using Chrome [8] are encouraging, but do not support legacy PC applications.

The approach of secure hardware-based virtualisation can be applied to servers, mobile devices or embedded systems. Similar to using remote attestation for DRM purposes, the approach can in principle also be used to protect data on cloud computing servers.

As of writing, it is uncertain whether an entire system can be built that is protected against any type of malware while also being open for legacy code. However, we think that this

[2] For technical details, see the information available at www.opentc.net. Use for example the project's final report or the issues given in the newsletter for guidance.

paper describes a useful path for the hardware and software industry to follow. Note that the objective is not only to create reliable isolation for users interested in security, but also to create easy to manage compartments for all users. To push this approach from idea into reality, we currently see the following paths towards the development and large-scale deployment of such a solution:

(1) Privately financed marginal improvements.

(2) A large government-funded project, starting with a proper, threats-based specification of hardware and software and followed by subsequent implementation. This would take the commons-nature of the problem into account, i.e. the distribution of costs across many users.

(3) Governments create incentives via regulations or procurement. Requirements might push this approach, similar to the incentives provided by PCI-DSS (Payment Card Industry Data Security Standard), the Sarbanes-Oxley Act on auditing requirements, requirements for Trusted Computing, or military requirements.

(4) Global discussion of this path will create demand. Think of the changes in the car industry, which was challenged by events such as a book titled “unsafe at any speed” [14]. Also imagine big customers demanding updates towards the goal. Newsletters and our website http://open-hypervisor.org/ can also be regarded as steps towards the creation of demand.

Acknowledgements

We wish to express our thanks to Dirk Kuhlmann, Armand Puccetti, Matthias Schunter and Michael Wilson.

Bibliography

[1] ARBAUGH, W., FARBER, D., and 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. 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 by FURGER, F. and 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., 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". Presentation given at: 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. and LEONID, R. The Road to Trustworthy Systems. Communications of the ACM, 53(6), 107–115, June, 2010.

[12] KUHLMANN, D. and WEBER, A. OpenTC Final Report. The Evolution of the OpenTC Architecture Illustrated via its Proof-of-Concept-Prototypes. 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. Project website. http://www.opentc.net/.

[16] OPENTC. Project newsletter, available at www.opentc.net.

[17] PFITZMANN, B., JAMES, R., STÜBLE, C., WAIDNER, M. and 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., 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, June 2010. http://www.secorvo.de/security-news/secorvo-ssn1006.pdf.

[20] SEIFERT, J.-P. Keynote presentation at: Computers, Privacy and Data Protection, Brussels 2010.

[21] SERDAR, C., DALTON, C.I., ERIKSSON, K., KUHLMANN, D., RAMASAMY, H. G. V., RAMUNNO, G., SADEGHI, A.-R., SCHUNTER, M., and 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. (eds): Proceedings of CPDP 2010, Brussels. Springer, 2011. 201-207. http://www.springerlink.com/content/t067038412352321/.

[23] WEBER, A., WEBER, D., PRESTI, S.L. Requirements and Design Guidelines for a Trusted Hypervisor User Interface. Paper presented at: Future of Trust in Computing. Berlin, Germany, 30 June – 2 July, 2008. Proceedings published by Vieweg & Teubner, Wiesbaden 2009.