Paths to Secure Information Technology in Open Networks

Abstract for TA13 – Sicherheit als Technik, Vienna 2013

http://www.oeaw.ac.at/ita/veranstaltungen/konferenzen/ta13-konferenz-362013/ueberblick

Version of 4 Sept 2013

Arnd Weber[x], Dirk Weber[y]

x) Karlsruhe Institute for Technology, ITAS. Email: arnd.weber@kit.edu

y) DAWITCON. Email: daw@dawitcon.de

Problem

More and more sensitive applications are being linked to the Internet. Anxieties arise when plans show that the future smart electricity grid will be managed via the Internet, e.g., when a smartphone will be used to manage the settings for charging one's car battery or for transferring its charge to the grid when there is not enough renewable energy available. Could an attacker hack one's phone and discharge one's battery? Or even discharge all car batteries and turn off all house lights, too? Anxieties also arise when malware such as Stuxnet is created – with several man-years of effort – that can influence industrial plants (Falliere et al. 2010). Indeed, large, professionally protected firms such as Coca Cola or even the security company RSA have not succeeded in protecting themselves against attacks (Bloomberg 2012; Weber, Weber 2011).

Technical Solutions

Clean Slate

It would be possible to start all over again – wipe a clean slate – and build a new network with new computers. This system would only accept digitally signed programs, perhaps even only those whose correctness were proven. Yet this does not solve all the problems because we would have to give hundreds of thousands of developers the opportunity to develop applications. This would make it possible for malware to be produced, whether inadvertently or intentionally. Furthermore, it would be economically desirable to let existing applications continue to run. This software, just like the respective operating systems, is so large that it would take years for it to be examined and improved until its safe functioning were demonstrated, which would stop its development and be very expensive.

Isolation

The only remaining alternative is to isolate sensitive applications from potentially harmful ones. The basic idea is to separate applications from others by putting them into a type of container, e.g. into a compartment of virtualization software. Such a container might, for example, communicate with a selected server. Several problems are tied to this basic approach.

- Decisive is that the entire computer is secure. It must be impossible for any novel Trojan horses to be hidden in the hardware. This possibility cannot be ruled out if so many components are manufactured in countries such as China (see CPNI 2012). For this reason, the production would have to be very transparent and supervised, such as in Germany or another country in the European Union.

- We have little experience in building error-free or proven, complex, open computer systems. It is fundamentally possible for a novel attack to be conducted. For example, there are side-channel attacks in which malware could attempt to listen in on information in an isolated container (Heiser et al. 2012). Or malware could succeed in identifying access data (such as passwords) by listening in on sensors. In this case, the difficulty consists in the fact that new attacks can arise that no one had thought of before. Fundamentally, it is even possible for an arbitrary number of developers of a high-security system to succumb to a mistake, be bribed, or something similar. The usual statement by security experts that there is no 100% security is based on such possibilities of being attacked.

- The solution has to be practicable. For example, in principle there would be no clipboard to be used for exchanging information between the containers. Yet if we do not provide an intelligent clipboard, the users could attempt to link the secure and the insecure parts using a tunnel, e.g., via e-mail (Weber et al. 2009).

- The channels between the computers have to be made secure. On the one hand, this is easily possible using current encryption technology. If other weaknesses were eliminated, attackers could theoretically, on the other hand, concentrate on breaking this technology. For example, intelligence services occasionally boast about having confidential mathematical know-how (cf. Simmons 1986).

Now there are several possibilities for creating security using isolation.

- First, it would be possible to isolate only applications and their data from one another (cf. http://www.bizztrust.de/en/product.html).

- It would also be possible to isolate operating systems, i.e. use virtualization.

- It is also conceivable for the different applications to use separate chip sets and for them to be operated from a common user interface (I/O).

Finally, security can be improved to different degrees. In principle, there are three levels:

1. Business as usual: Computers are being gradually improved. Consider, for example, the use of Trusted Computing modules, the increasing use of virtualization software, and the support for virtualization by means of memory curtaining in the hardware. These ideas have been

implemented in part, but nowhere completely (see Grawrock 2006), and no implementation has been completely checked for errors.

2. A major improvement would come from the utilization of tested or certified systems. An example is the military's Security-Enhanced version of Linux.

3. Finally, another way is to prove the security of components and systems (Heiser et al. 2010, 2012), which is also being examined by the US military (DARPA 2013).

Role of Technology Assessment

For technology assessment, this means evaluating the different approaches to reaching a solution with regard to their security, economics, and, if appropriate, usability.

Also important for technology assessment is the appraisal of the residual risks associated with highly secure solutions. The objective should be to inform citizens of these risks comprehensively. For example, if an application could be attacked through a covert channel attack, it would be possible to demand separation in the hardware. If concerns were expressed that the communication channel between two computers were decrypted, technology assessment researchers and technicians could point out that thousands of mathematicians around the world had vainly attempted to find any weakness allowing this. Each residual risk could correspondingly be analyzed and assessed.

In cooperation with specialists from technical fields, technology assessment could design IT systems whose theoretical residual risks are, according to our best knowledge, irrelevant for certain applications.

Technology assessment should inform the public about the options and the residual risks. Legislative bodies and other actors in the political sphere could use the resulting public awareness to build political pressure for the introduction of secure systems. Public action in the form of legal stipulations and liability regulations would have the advantage that manufacturers offering more secure solutions would not be subject to economic disadvantages, as is the case today.

In doing this, technology assessment would point out technical and socioeconomic paths leading to IT systems that could practically not be attacked from the Internet.

Acknowledgments

We would like to thank Michael Decker, Reinhard Heil, Maggie Jaglo, Ulrich Riehm and Michael Wilson for their comments on earlier versions of this abstract.

References

Bloomberg: Coke Gets Hacked And Doesn't Tell Anyone. 5.11.2012. http://www.bloomberg.com/news/2012-11-04/coke-hacked-and-doesn-t-tell.html

CPNI: National Infrastructure Protection. Emerging Technologies. April 2012. http://www.cpni.gov.uk/documents/publications/2012/2012014-national_infrastructure_protection_emerging_technologies.pdf?epslanguage=en-gb

Darpa: Information Innovation Office. Access 17.3.2012. http://www.darpa.mil/Our_Work/I2O/Programs/Clean-slate_design_of_Resilient_Adaptive_Secure_Hosts_%28CRASH%29.aspx

Falliere, N.; O Murchu, L.; Chien, E.: W32.Stuxnet Dossier. 2010. http://www.symantec.com/connect/blogs/w32stuxnet-dossier

Grawrock, D.: The Intel Safer Computing Initiative. Intel Press, 2006.

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.

Heiser, G.; Murray, T.; Klein, G.: It's Time for Trustworthy Systems. IEEE Security & Privacy 10(2): 67-70, 2012.

Simmons, G.: Cryptology. In: Encyclopædia Britannica 1986, S. 913-924B.

Weber, A., Weber, D. und Lo Presti, S.: Requirements and Design Guidelines for a Trusted Hypervisor User Interface. Presentation at: Future of Trust in Computing. Berlin, Germany, 30 June – 2 July, 2008. Proceedings. Vieweg & Teubner, Wiesbaden 2009.

Weber, A.; Weber, D.: Isolating Spears. Karlsruhe, July 28, 2011. http://www.open-hypervisor.org/index.php/HPvisor/news/33/