System Security, Platform Security and Usability*

[Extended Abstract]

Paul C. van Oorschot
School of Computer Science, Carleton University, Canada

ABSTRACT

Scalable trusted computing seeks to apply and extend the fundamental technologies of trusted computing to large-scale systems. To provide the functionality demanded by users, bootstrapping a trusted platform is but the first of many steps in a complex, evolving mesh of components. The bigger picture involves building up many additional layers to allow computing and communication across large-scale systems, while delivering a system retaining some hint of the original trust goal. Not to be lost in the shuffle is the most important element: the system's human users. Unlike 40 years ago, they cannot all be assumed to be computer experts, under the employ of government agencies which provide rigorous and regular training, always on tightly controlled hardware and software platforms. It seems obvious that the design of scalable trusted computing systems necessarily must involve, as an immutable design constraint, realistic expectations of the actions and capabilities of normal human users. Experience shows otherwise. The security community does not have a strong track record of learning from user studies, nor of acknowledging that it is generally impossible to predict the actions of ordinary users other than by observing (e.g., through user experience studies) the actions such users actually take in the precise target conditions. We assert that because the design of scalable trusted computing systems spans the full spectrum from hardware to software to human users, experts in all these areas are essential to the end-goal of scalable trusted computing.

Categories and Subject Descriptors

D.4.6 [**Operating systems**]: Security and Protection; K.6.3 [**Management of computing and information systems**]: Software Management

General Terms

Security, Design, Human Factors, Experimentation

Keywords

trusted computing, software installation, usability, security and user experience

*Version: August 16, 2010. paulv@scs.carleton.ca

STC'10, October 4, 2010, Chicago, Illinois, USA.
ACM 978-1-4503-0095-7/10/10.

1. CONTEXT AND SETTING

We are interested in how scalable trusted computing (STC) relates to system security, platform security, and usability. Computer *system security* is an emergent property of a system. *Platform security* is more narrowly concerned with the properties of a particular node in a larger system, e.g., a specific hardware/ operating system platform; *trusted computing* is typically first defined relative to a particular platform. We interpret the *scalable* part of STC to include scalability across very large numbers of users, even a significant subset of the 7 billion people on earth; and across the web, meaning involving a mix of browser technologies, web server platforms, and the like. The complexity of today's browsers plus their incredible rate of technological evolution, adds significantly to the challenges for trusted computing. Scalability across the web also implies communications with end-users whose platforms may be under different security policy administrations (implying end-points of varying degrees of trust). By *usability*, in the context of computer security and usability, our ultimate interest is in how the design of a particular system influences the actions or behavior of a human user, and in particular whether these are desirable or undesirable from a system security viewpoint. To this end, we may prefer to use the term *security and user experience*.

A main focus of the trusted computing community has been hardware trusted platform modules (TPMs) [13] and security-enhanced operating systems like SELinux and AppArmor [7]. On the broader topic of bootstrapping trust in commodity computers, progress has been made well beyond trusted boot, to include dynamic trust measurements, remote attestation, and even usability [19, 16, 17].

A formidable challenge, however, is presented by today's typical client devices being subject to both very frequent software updates (some fully automated), and an unavoidable flow of executable content downloads triggered by visiting even benevolent web sites. The issue of software installation is, by itself, fundamental to trusted computing, usability, and system integrity. Worrisome issues have been raised related to package managers [6], software updates which are neither digitally signed nor otherwise authenticated [5], and the overall complexity and variety of software installation mechanisms that users are subjected to [1, 24]. Users become conditioned to entering root passwords, much as they naturally respond automatically when seeing stimuli matching those previously encountered, in so-called *click-whirr* reponses [15]. The challenges are amplified in today's smart phone ecosystems, where the emphasis is on making installation of applications simpler than ever before,

despite security and usability challenges arising from smart phone form factors. On Android phones, a novel permissions model is used which, while promising in some respects, is worrisome—users are prompted to approve of the permissions requested by newly loaded applications [12, 3], but given little information for informed choice.

Including cloud computing within the context of scalable trusted computing, as well as typical client-server communications across disjoint policy domains, introduces another formidable challenge: reliable authentication. Surprisingly, there remain many open problems on even the most basic aspects of authentication, including simple password authentication. Alas, end-user authentication involves humans, who are subject to phishing attacks [10], and are overwhelmed by increasing demands made in the name of better security [14]. Many basic usability issues related to passwords have only recently begun to receive research attention, such as password interference [8] and the usability of password managers [9]. Moving beyond passwords, to the hierarchy of certificates which now includes extended validation (EV) SSL certificates, raises far more complex issues related to users' mental models, and confidentiality vs. site identity.

A growing threat to security and trust, in a world reliant on browsers as the communications interface, is the ubiquity of browser extensions. Security mechanisms are now being proposed to partially address risks inherent in such extensions [4, 2, 11], but extensions are best considered as the installation of untrusted code, much as installing unvetted smart phone software. Among many other security issues related to browsers—which have become the operating systems for the web— are questions related to web page mashups and the interaction of code and data across different sites. Mechanisms for restricting the flow of information between sites have been proposed [18], for controlling ability of ads to overwrite arbitrary portions of web pages [22], and to discover client-side vulnerabilities related to rich web applications typically written in JavaScript [20]. Issues also remain regarding browser access control policies [21].

2. FUNDAMENTAL QUESTIONS

Many of the security concerns related to STC are captured by three basic questions, which can be related to visiting web sites, downloading software, how web page content is assembled, and related web activities:

1. Where am I? Who's on the other end?
2. What code is running in my box?
3. Where did it (really) come from?

Other questions relate specifically to software developers: Who are their target users? Do they punt decisions to end-users too often? Do they hurt or help usable security [23]?

Acknowledgments. Thanks are due to N. Asokan and Ahmad-Reza Sadeghi, co-chairs of the 5th Annual ACM Workshop on Scalable Trusted Computing, for the invitation to deliver this keynote. The author acknowledges funding from an NSERC Discovery Grant and NSERC ISSNet.

3. REFERENCES

[1] J. Anderson, J. Bonneau, F. Stajano. Inglorious installers: security in the application marketplace. WEIS 2010.

[2] S. Bandhakavi, S.T. King, P. Madhusudan, M. Winslett. VEX: Vetting Browser Extensions for Security Vulnerabilities. USENIX Security 2010.

[3] D. Barrera, H.G. Kayacik, P.C. van Oorschot, A. Somayaji. A metholodology for empirical analysis of permission-based security models and its application to Android. ACM CCS 2010.

[4] A.Barth, A.P.Felt, P.Saxena, A.Boodman. Protecting Browsers from Extension Vulnerabilities. NDSS 2010.

[5] A. Bellissimo, J. Burgess, K. Fu. Secure software updates: disappointments and new challenges. USENIX HotSec'06.

[6] J.Cappos, J.Samuel, S.Baker, J.Hartman. A look in the mirror: attacks on package managers. CCS 2008.

[7] H. Chen, N. Li, Z. Mao. Analyzing and Comparing the Protection Quality of Security Enhanced Operating Systems. NDSS 2009.

[8] S. Chiasson, A. Forget, E. Stobert, P.C. van Oorschot, R. Biddle. Multiple password interference in text and click-based graphical passwords. ACM CCS 2009.

[9] S. Chiasson, P.C. van Oorschot, R. Biddle. A Usability Study and Critique of Two Password Managers. USENIX Security 2006.

[10] R. Dhamija, J.D. Tygar, M.A. Hearst. Why phishing works. CHI 2006.

[11] V. Djeric, A. Goel. Securing Script-Based Extensibility in Web Browsers. USENIX Security 2010.

[12] W. Enck, M. Ongtang, P.D. McDaniel. On lightweight mobile phone application certification. CCS 2009.

[13] D. Grawrock. *Dynamics of a Trusted Platform.* Intel Press, 2008.

[14] C.Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. NSPW 2009.

[15] C. Karlof, J.D. Tygar, D. Wagner. Conditioned-safe Ceremonies and a User Study of an Application to Web Authentication. NDSS 2009.

[16] J.M. McCune, Y. Li, N. Qu, Z. Zhou, A. Datta, V.D. Gligor, A. Perrig. TrustVisor: Efficient TCB reduction and attestation. IEEE S&P, Oakland 2010.

[17] J.M. McCune, A. Perrig, A. Seshadri, L. van Doorn. Turtles all the way down: research challenges in user-based attestation. USENIX HotSec'07.

[18] T.Oda, G.Wurster, P.C.van Oorschot, A.Somayaji. SOMA: mutual approval for included content in web pages. ACM CCS 2008.

[19] B.Parno, J.M.McCune, A.Perrig. Bootstrapping trust in commodity computers. IEEE S&P, Oakland 2010.

[20] P. Saxena, S. Hanna, P. Poosankam, D. Song. FLAX: Systematic Discovery of Client-side Validation Vulnerabilities in Rich Web Applications. NDSS 2010.

[21] K. Singh, A. Moshchuk, H.J. Wang, W. Lee. On the Incoherencies in Web Browser Access Control Policies. IEEE S&P, Oakland 2010.

[22] M.Ter Louw, K.T.Ganesh, and V.N.Venkatakrishnan. AdJail: practical enforcement of confidentiality and integrity policies on web ads. USENIX Security 2010.

[23] G. Wurster, P.C. van Oorschot. The developer is the enemy. NSPW 2008.

[24] G. Wurster, P.C. van Oorschot. A control point for reducing root abuse of file-system privileges. ACM CCS 2010.