A Trusted Mobile Phone Prototype

Onur Acıiçmez Afshin Latifi Jean-Pierre Seifert Xinwen Zhang
onur.aciicmez@gmail.com a.latifi@samsung.com
jeanpierreseifert@yahoo.com xinwen.z@samsung.com
Samsung Information Systems America, Samsung Electronics R&D Center
95 West Plumeria Drive, San Jose, CA 95134, USA

***Abstract*—Due to the increasing security demands in mobile devices, the Trusted Computing Group (TCG) formed a dedicated Mobile Phone Working Group (MPWG) to address these security needs. MPWG recently released a Trusted Mobile Phone Reference Architecture (TCG-MPRA) specification that integrates well-known security concepts (TPM, isolation, Integrity Measurement and Verification (IMV), etc.) from the trusted PC universe, tailored for mobile phones. The business needs of the mobile phone industry mandate 4 different stakeholders (platform owners): device manufacturer, cellular service provider, general service provider, and the end-user. The specification requires separate trusted and isolated operational domains (Trusted Engines) for each stakeholder. Although the TCG MPWG does not explicitly prescribe a specific technical realization of these trusted engines, a general consensus is use of established (Trusted) Virtualization concepts from corresponding PC architectures. However, we will demo another isolation technique specifically crafted for mobile platforms that respects their resource limitations. We achieve this goal by realizing the MPWG specification by leveraging SELinux which provides a generic domain isolation concept at the kernel level. In addition to utilizing SELinux to realize mobile phone specific (isolated) operational domains, we are also able to seamlessly integrate the important IMV concept into our SELinux-based Trusted Mobile Phone architecture. In our demo we will present a hardware prototype, representing a generic mobile phone, implementing the TCG MPWG specification. First, we will "Securely Boot" our TC-aware SELinux kernel out of a hardware Mobile Trusted Module (MTM). Next, we will show how easy and efficient we can realize the 4 isolated Trusted Engines. The value of the Trusted Engines and the fundamental IMV principle will be demonstrated through successful mitigation of two automatic Linux cell-phone worms. The prototype in this demo is in effect, the world's first novel, efficient and inherently secure implementation of MPWG specification.**

I. Background Information

In the last decade, the significant increase in processing power of the average mobile phone, coupled with pervasive network connectivity, has resulted in large number of applications and services to be developed and deployed on these devices. Some example of these new usage models are mobile payment and ticking, software as a service (SaaS), pervasive information and content (audio, video, and text) sharing, seamless collaborations, voice-over-IP (VoIP), and many more. Unavoidably, these advances present new security challenges, which cannot be satisfied with present security facilities available to current limited purpose phones. As a result, we have seen an increase in number of security attacks in mobile computing.

Several organizations have proposed security requirements and specifications for mobile phone security. For example, the Open Mobile Terminal Platform (OMTP), an operator-sponsored forum, has proposed an application security framework to specify security requirements for applications running on a mobile phone [14]. LiMo (Linux Mobile) Foundation, which is formed by major mobile device manufacturers, is designing a generic Linux-based open software platform for mobile devices. Security is claimed to be an important component in these framework. However, the formal specification has not been released yet [2]. The Open Mobile Alliance (OMA) [3] is an organization founded by nearly 200 companies including mobile operators, device and network suppliers, and application/content/service providers. The security working group of OMA mainly specifies protocols for secure communication between mobile clients and servers at both the transport and the application layers.

The TCG recently published specifications for MTM [7], which is a modified version of the TPM — the counterpart for PC platforms. Typically, a mobile phone is "owned" by multiple stakeholders, including device manufacturer, network operator, 3rd party service provider, and the user (customer). One owner cannot turn off or damage the services of another owner by compromising the protection and security mechanisms. Since it is a unique feature of MTMs, each MTM can be owned by local and remote stakeholders. Each stakeholder is required to own basic trusted infrastructure such as secure boot and storage, IMV, remote attestation, etc.

The TCG further developed the TCG-MPRA specification [6], which specifies a general architecture for mobile devices relying on the trusted services of MTMs. Besides IMV, trusted storage and reporting mechanisms, the TCG-MPRA requires the resources of different stakeholders to be strongly isolated and the communications between different agents and services — representing corresponding stakeholders — to be controlled according to pre-agreed security policies. Particularly, the TCG-MPRA "generalizes" the concept of a platform to mean a set of conventional TCG-enabled platforms, and calls them "engines" to differentiate them from the ensemble platform" [6].

The TCG-MPWG does not "mandate a specific form or strength of isolation since those are dictated by the purpose of the trusted mobile platform". Typically, on PC and server-based platforms, Virtualization approaches [16], [4], [12], [15] are used to fulfill the isolation requirement of TCG-MPRA. Specifically, the operating system (OS), runtime environment, resources and applications of a domain are located in a dedicated virtual machine (VM) so that a virtual machine monitor (VMM) or hypervisor assures the isolation of individual domains. The communication between the domains is controlled by the VMM according to pre-defined policies. Although this approach successfully fulfills the isolation requirement, it has a main drawback. Typically, virtualization is realized with an additional software layer that abstracts the underlying hardware resources (e.g., CPU, memory, and storage) and it introduces significant overhead to the device. Hardware based approaches to virtualization can have even worse overhead. [8]. Virtualization is evidently not a practical isolation solution for mobile phones, due to limited computational capabilities and low power consumption requirement. The practicality of such a heavy-weight solution also recently questioned in [9]. Also, controlled communication and resource sharing between VMM layers is not fine-grained and would require additional system resources.

Strong isolation often contradicts with resource sharing flexibility. For example, many malware on mobile phones spread via Short Message Service (SMS) and Multimedia Messaging Service (MMS). If these services are isolated from the rest of the software and resources on the platform, these malware cannot easily infect the entire platform and its associated network.

However, in order for a mobile device to be interactive one must provide efficient methods to support dynamic interactions between different processes. These methods should be efficient while upholding the security of the system. Absence of such security will leave running processes vulnerable. Some

examples of common attacks are direct interference from other processes, fake/malicious input, unexpected data from sources with lower integrity/sensitivity levels. These examples point out the necessity of isolation and secure resource sharing and interprocess interactions.

Towards an affordable but effective solution for mobile devices, we propose an efficient isolation and flexible communication mechanism between different engines in multi-stakeholder environments by leveraging matured mechanisms of the secure kernel research area. Traditional Unix-like operating systems only provide discretionary access control (DAC), which makes it impractical to enforce strong isolation between system services and application processes. Fortunately, several kernel level but general-purpose mandatory access control (MAC) mechanisms have been developed for main stream operating systems such as Security Enhanced Linux (SELinux) [13], AppArmor [1], LOMAC [11], and LIDS [5]. Typically, a MAC system strictly controls — according to some predefined sets of rules — the interactions between subjects (e.g., services or processes) and objects (e.g., files, sockets, etc.), which are differentiated based on the labels assigned to them. Different policies can be implemented by defining different rules to enforce (i.e. resource separation, data confidentiality and integrity, and a general information flow control).

The main advantages of our approach on mobile devices are the following.

1.) Kernel-level mechanisms are intrinsically trusted, unlike ALTM such as those of OMTP. Simply because the kernel is a part of the trusted computing base (TCB) [10]. Therefore, the security enforcement in the kernel space is much more trustworthy when compared to ALTM.

2.) The advantage of a MAC-based isolation compared to virtualization techniques is pure performance. Since mobile phones have limited computational capabilities and low power consumption requirements, virtualization becomes an impractical solution. However, one can argue that current MAC mechanisms, (i.e. SELinux) are also resource hungry and would result in poor performance on mobile devices. Although MAC mechanisms consume substantial computing power on PC platforms (due to vast number of subjects and objects), mobile devices in contrast are still limited and cannot be compared to classical PC environments in this regard. Most of the services/applications used in a PC are not present on mobile platforms (e.g., web server, etc.). This significantly simplifies the security policies (and policy development) and improves the potential performance of MAC mechanisms on mobile devices.

The TCG specifies also that all software executed on a secure-boot engine must be authenticated, thus requires that each software has to be measured when loaded. Also, the integrity should be verified by a security enforcement component before it communicates with existing processes in the engine, as otherwise it may compromise the overall security of the stakeholder. The normal resource of an engine can be measured by its trusted resource, and the integrity can be verified by other engines or remote entities through attestation.

Similar to resource isolation and controlled communication, TCG does not specify the implementation of IMV by a specific engine. We propose an architecture where IMV are integrated into the security policy enforcement mechanism. Our architecture is extended from the SELinux security module.

The architecture used in our demo is based on SELinux, which is built on the Linux Security Module (LSM). With LSM enabled, a set of hooks inside the kernel monitor sensitive system calls by high level processes. The current SELinux security context is limited to Type Enforcement only. To enable IMV before a software is executed, our advanced SELinux-based TCG MPRA, we extend the SELinux policy model to also include integrity-related attributes, cf. [17], [18]. This enhanced SELinux contains two more contextual attributes called `profile` and `system`. The extended security context is of the form

`user:profile:role:type:system`.

The `profile` contextual attribute allow the specification of a general set of attributes that can be associated with a particular subject or object. The extended policy not only specifies the domain or type of a subject, but also its integrity status in order to obtain the access permission. Therefore, SELinux policy enforcement mechanism is extended to consider also the integrity requirement.

II. Demonstration Content

Our demonstration will first "Securely Boot" our TC-aware SELinux kernel out of a hardware MTM. Secondly, it is demonstrated the ease and efficiency at which one can realize the required 4 isolated Trusted Engines. The value of the trusted engines and the IMV principle will be demonstrated by successful mitigation of two automatic Linux cell-phone worms. The first attack is blocked by SELinux's access control to the respective engine's Trusted Resources. However, the second worm escapes the security net. In the final stage, the demo will point out how addition of the IMV capability to our platform, blocks the second cell-phone worm. Our demo and architecture is supported by the following publications describing various pieces of our Trusted Mobile Phone architecture and vision, cf. [17], [18], [19], [20], [21].

References

[1] Apparmor. http://en.opensuse.org/AppArmor.
[2] Limo foundation. https://www.limofoundation.org.
[3] Open Mobile Alliance. http://www.openmobilealliance.org.
[4] Open trusted computing (opentc) consortium. http://www.opentc.net/.
[5] The Linux Intrusion Defence System (LIDS). http://www.lids.org/.
[6] TCG mobile reference architecture specification version 1.0. https://www.trustedcomputinggroup.org/specs/mobilephone/ tcg-mobile-reference-architecture-1.0.pdf, June 2007.
[7] TCG Mobile Trusted Module Specification Version 1.0. https://www.trustedcomputinggroup.org/specs/mobilephone/ tcg-mobile-trusted-module-1.0.pdf, June 2007.
[8] K. Adams and O. Agesen. A comparison of software and hardware techniques for x86 virtualization. In *Proceedings of the Twelfth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 2–13, San Jose, CA, USA, October 21-25 2006.
[9] L. P. Cox and P. Chen. Pocket Hypervisors: Opportunities and Challenges. *HotMobile 2007*, Tucson, AZ, February 2007
[10] Department of Defense National Computer Security Center. *Department of Defense Trusted Computer Systems Evaluation Criteria*, December 1985. DoD 5200.28-STD.
[11] T. Fraser. LOMAC: MAC you can live with. In *Proc. of the 2001 Usenix Annual Technical Conference*, June 2001.
[12] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh. Terra: A virtual machine-based platform for trusted computing. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, pages 193–206, Bolton Landing, New York, USA, 2003.
[13] P. Loscocco and S. Smalley. Integrating flexible support for security policies into the linux operating system. In *Proceedings of USENIX Annual Technical Conference*, pages 29–42, June 25-30 2001.
[14] OMTP. Application security framework. http://www.omtp.org/docs/OMTP_Application_Security_Framework _v2_0.pdf, 2007.
[15] R.Sailer, T. Jaeger, E. Valdez, R. Perez, S. Berger, J. L. Griffin, and L. van Doorn. Building a mac-based security architecture for the xen opensource hypervisor. Technical report, IBM Research Report RC23629, 2005.
[16] A. Sadeghi and C. Stüble. Taming trusted platforms by operating system design. In *Proceedings of the 4th International Workshop for Information Security Applications, LNCS 2908*, pages 286–302, Berlin, Germany, August 2003.
[17] M. Alam, M. Hafner, J.-P. Seifert, and X. Zhang. Extending SELinux Policy Model and Enforcement Architecture for Trusted Platforms Paradigms. In *Annual SELinux Symposium 2007*.
[18] Onur Acıiçmez, J.-P. Seifert, and X. Zhang. A Trusted Mobile Phone Reference Architecture via Secure Kernel. In *2nd ACM Workshop on Scalable Trusted Computing 2007 (STC 2007)*.
[19] M. Alam, R. Breu, M. Hafner, J.-P. Seifert, and X. Zhang. Trusted SECTET: A Model-Driven Framework for Trusted Computing based Systems. In *11th IEEE International EDOC Conference (EDOC 2007)*.
[20] B. Agreiter, M. Alam, R. Breu, M. Hafner, A. Pretschner, J.-P. Seifert, X. Zhang. A Technical Architecture for Enforcing Usage Control Requirements in Service-Oriented Architectures. In *2007 ACM Workshop on Secure Web Services (SWS 2007)*.
[21] B. Agreiter, M. Alam, R. Breu, M. Hafner, J.-P. Seifert, X. Zhang. Model Driven Configuration of Secure Operating Systems for Mobile Applications in Healthcare. In *1st ACM/IEEE International Workshop on Model-Based Design of Trustworthy Health Information Systems (MOTHIS 2007)*