Verified Boot in IoT Devices with Low Power Consumption

Christos Profentzas
Chalmers University of Technology
Gothenburg, Sweden
chrpro@chalmers.se

ABSTRACT

In this paper, we describe our ongoing research regarding the security of operating systems for IoT devices. We try to highlight energy consumption issues posed by security measures. We start by securing the device boot-up process to provide the necessary dependency towards a trustful operating system. Lastly, our focus is a holistic view of the security model, which combines security measures and energy consumption in IoT devices.

CCS CONCEPTS

• **Security and privacy → Embedded systems security**; *Trusted computing*;

KEYWORDS

IoT, Verified Boot, Trusted Boot, Secure Boot, Power Consumption

1 INTRODUCTION

The modern field of *Internet of Things* (IoT) has emerged from the evolution of Wireless Sensor Networks (WSNs). The explosion of technologies and protocols has made applications beyond the simple data gathering available. For instance, an adaptive lighting in road tunnels [1] is possible using WSN as part of a closed-loop control system.

To extend and use such applications in the wider area of IoT, we need to add more capabilities to the embedded devices. Therefore, we have seen several Operating Systems (OS) emerge to manage the growing number of resources. In the survey of Padmini Gaur & Mohit P. Tahiliani [3], we find an extensive comparison among the most recent operating systems for IoT devices. Although the performance of the device is an essential factor of success in the field of IoT, the security of the device is another factor equivalent in importance.

Tock-OS [4], an embedded operating system, has already made a significant effort to address operating system security issues. The developers wrote the kernel of this particular OS in a new system programming language called RUST [8], where the compiler checks the safety of the memory during the compiling phase. Therefore, it avoids numerous security flaws. However, the user can only trust an IoT device if the device has booted an authenticated operating system in the first place. In this way, we must also secure the boot method of the device.

Even though, the security aspect of the boot process has already been highlighted in embedded devices for safe-critical applications such as Avionics Wireless Networks (AWN) [7], securing the boot process for IoT device has not received as much attention yet. The main reasons are extreme limitations regarding the power consumption and latency requirements for IoT devices.

Thus, we investigate the overall power consumption caused by security measures during the boot process. From our experience, we have seen a significant compromise between dealing with real-time constraints and implementing security measures in the boot process. Therefore, our hypothesis estimates a similar compromise between the power consumption and the security level we require to achieve.

Finally, it is within our research priorities to investigate a holistic security model that ensures all parts of the device from the boot process to operating system and finally the application level. Our research goals aim towards achieving a reliable platform for IoT devices that take energy consumption into consideration.

2 SECURITY THREATS

The main reason for a user to not trust the Operating System is the risk of malicious modification to the Kernel. The term Rootkit [6] covers these modifications, which is a set of tools designed to maintain privileged access to a compromised Operating System. How an attacker can gain privileged access to the system is outside of our scope, and we do not analyze it further. Therefore, we assume that the attacker has already found a way to compromise the system. Additionally, attackers try to hide their malicious software in deeper operating system structures, ultimately targeting the boot process and the startup code of the device. These advanced tools are covered by the term Bootkit [5].

Finally, the tools mentioned above are not the only methods for modifying the operating system for malicious purposes. For example, Cloaker [2] is a dynamic way to undermine the normal execution of the operating system running on ARM embedded systems. Therefore, it is within our research scope to construct security models to capture any limitations of security measures.

3 VERIFIED BOOT OVERVIEW

In general, we can say that a Verified Boot mainly provides a report (verification) about the authenticity of the boot-up code and the OS kernel regarding unauthorized modifications. However, we can find different terms used in various context.

Secure Boot [7] is another term used to describe such a verification method, which requires the device to verify before loading the operating system. However, we must not confuse the extent of the guarantees provided by the validation technique. The validation results only report modifications of the operating system; this states nothing about the trustworthiness of the verification process.

In this way, we need to introduce a trust between the verification process and the device. We can issue trust by measuring certain device configuration properties before and after the verification process. Also, the measurements could be validated by a third party upon a request.

The above architecture requires external hardware to provide the necessary trust. This entity already exists in the form of Trusted Platform Module (TPM) [7]. This module is a passive reporting device, and the enforcement policy is open to suggestions. Moreover, TPM can play the role of a root of trust, where we maintain the validation data. The aim is to create a chain of trust from the boot of device to the operating system and eventually the application layer. Figure 1 illustrates the boot process using TPM as a root of trust.

One of the challenges that our research is trying to address is the minimal hardware requirements. It is important to modify to a minimum the existing architecture of IoT platforms as cost and energy consumption should be kept low. Another challenge is the limitations of introducing a verification method into the low-level boot code, due to the small and compound memory space that such code occupies. Lastly, we have set the primary objective of minimizing power consumption that may conflict with the verification process. In this way, we should take into consideration the possibility of a compromise between security measures and power consumption.

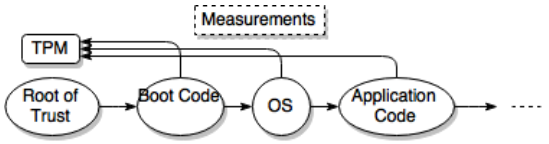

Figure 1: A Trusted Boot Process using TPM

4 CONCLUSIONS

In this paper, we introduce our ongoing research and interests regarding the security of IoT devices. Moreover, we argue that security is an essential factor in the success of IoT devices; as the ever-growing number of security threats make the importance of the security more urgent than ever. Any security measure should take into consideration the power consumption of the device, which is also another significant factor in the sustainability of IoT devices and their success. Our research focuses on addressing the challenges of a fully trusted IoT platform with low-power consumption.

REFERENCES

[1] M. Ceriotti, M. CorrÃă, L. D'Orazio, R. Doriguzzi, D. Facchin, S. T. GunÄČ, G. P. Jesi, R. L. Cigno, L. Mottola, A. L. Murphy, M. Pescalli, G. P. Picco, D. Pregnolato, and C. Torghele. 2011. Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. 187–198.

[2] Francis M. David, Ellick M. Chan, Jeffrey C. Carlyle, and Roy H. Campbell. 2008. Cloaker: Hardware Supported Rootkit Concealment. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy (SP '08)*. IEEE Computer Society, Washington, DC, USA, 296–310. https://doi.org/10.1109/SP.2008.8

[3] Padmini Gaur and Mohit P. Tahiliani. 2015. Operating Systems for IoT Devices: A Critical Survey. In *Proceedings of the 2015 IEEE Region 10 Symposium (TENSYMP '15)*. IEEE Computer Society, Washington, DC, USA, 33–36. https://doi.org/10.1109/TENSYMP.2015.17

[4] Amit Levy, Michael P. Andersen, Bradford Campbell, David Culler, Prabal Dutta, Branden Ghena, Philip Levis, and Pat Pannuto. 2015. Ownership is Theft: Experiences Building an Embedded OS in Rust. In *Proceedings of the 8th Workshop on Programming Languages and Operating Systems (PLOS '15)*. ACM, New York, NY, USA, 21–26. https://doi.org/10.1145/2818302.2818306

[5] X. Li, Y. Wen, M. H. Huang, and Q. Liu. 2011. An Overview of Bootkit Attacking Approaches. In *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks*. 428–431. https://doi.org/10.1109/MSN.2011.19

[6] XiangYu Li, Yi Zhang, and Yong Tang. 2015. Kernel Malware Core Implementation: A Survey. In *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2015, Xi'an, China, September 17-19, 2015*. 9–15. https://doi.org/10.1109/CyberC.2015.26

[7] K. Markantonakis, R. N. Akram, and R. Holloway. 2016. A secure and trusted boot process for Avionics Wireless Networks. In *2016 Integrated Communications Navigation and Surveillance (ICNS)*. 1C3–1–1C3–9. https://doi.org/10.1109/ICNSURV.2016.7486322

[8] Nicholas D. Matsakis and Felix S. Klock, II. 2014. The Rust Language. In *Proceedings of the 2014 ACM SIGAda Annual Conference on High Integrity Language Technology (HILT '14)*. ACM, New York, NY, USA, 103–104. https://doi.org/10.1145/2663171.2663188