Master of Research:
Hardware based monitoring of chipset components

Chris Dalton (HP Labs), Guillaume Hiet (CentraleSupelec), David Plaquin (HP Labs), Ronny Chevalier (HP Labs/CentraleSupelec)

Context

Computing device hardware architectures are increasingly complex, integrating feature rich chipset components with a high level of connectivity. Furthermore, those connections often extend beyond the boundary of the devices themselves. For example, USB-C or thunderbolt now provide external plugged-in devices access to resources (e.g. main memory) traditionally only accessible by built-in chipset components.

This creates security issues where rogue devices can be used to either compromise the user's system or extract some confidential data[2][4]. Current counter measures[3] include CPU features (such as IO/MMU, PCIe ACS) to control access of the devices to certain resources. However, these mechanisms are limited to coarse grain access control policies allowing or denying complete communication between components (e.g. a PCIe device can be prevented from accessing the OS memory). For instance, such mechanisms cannot differentiate between device functions when applying the policies. Some countermeasures have been suggested to be implemented in the system software[5][1] but such approach rely on the assumption that the system software is not compromised. Furthermore, the system software running on the device is often unable to protect or even detect attacks occurring at the chipset level.

Objective

We want to explore the usage of a dedicated hardware component and architecture on the device to detect and prevent such attacks while giving the ability to apply more flexible security policies. We will consider solutions with various trust models, including cases where the system software is compromised. The student will be expected to:

- Survey the state of the art on attacks and counter measures in this field.

For more details, contact:

Guillaume Hiet
(guillaume.hiet@centralesupelec.fr)

- Identify a case study and focus on a connectivity protocol (e.g. USB, PCI-e, etc.) for the given case study.
- Study security relevant details about configuration and specification of such protocol.
- Design a hardware based approach to detect and prevent the chosen class of attacks.
- Evaluate the feasibility of the approach by implementing a proof of concept and validate its efficacy.

References

[1] Johnson, P.C. et al. 2017. Protecting against malicious bits on the wire: Automatically generating a usb protocol parser for a production kernel. *Proceedings of the 33rd annual computer security applications conference* (New York, NY, USA, 2017), 528–541.

[2] Markuze, A. et al. 2016. True iommu protection from dma attacks: When copy is faster than zero copy. *ACM SIGARCH Computer Architecture News*. 44, 2 (2016), 249–262.

[3] Sang, F.L. 2012. *Protection des systèmes informatiques contre les attaques par entrées-sorties*. INSA de Toulouse.

[4] Tian, J. (Dave) et al. 2018. SoK: "Plug & pray" today - understanding USB insecurity in versions 1 through C. *2018 IEEE symposium on security and privacy, SP 2018, proceedings, 21-23 may 2018, san francisco, california, USA* (2018), 1032–1047.

[5] Tian, D. (Jing) et al. 2016. Making USB great again with USBFILTER. *25th USENIX security symposium, USENIX security 16, austin, tx, usa, august 10-12, 2016.* (2016), 415–430.