Analyse de logiciels malveillants et détection d'intrusion depuis un environnement d'exécution de confiance

Mots-clés : firmware, TEE, sécurité système, UEFI

Contexte

Les plateformes récentes (ordinateurs portables ou téléphones) possèdent des environnements d'exécution de confiance, ou Trusted Execution Environment (TEE). Par exemple, Intel dispose du System Management Mode (SMM) et du Management Engine (ME), AMD dispose du Platform Security Processor (PSP), et ARM dispose du mode Secure World de la TrustZone. Un TEE a pour but d'exécuter du code de confiance dans un environnement isolé du reste du système en lui fournissant des garanties d'intégrité et de confidentialité. Ainsi, même si des attaquants compromettent le système d'exploitation, le code et les données du TEE ne doivent pas pouvoir être compromis à leur tour.

Des travaux ont été réalisés afin d'utiliser un TEE pour analyser le comportement de logiciels malveillants [9, 5] ou pour analyser l'état d'un système pour déterminer s'il a été compromis [8, 9, 10, 1, 6, 7]. Ces approches répondent principalement à la problématique du fossé sémantique [2, 4], qui existe entre le TEE et le système qui est inspecté (p. ex., le TEE n'a pas la connaissance des structures utilisées par le système d'exploitation pour gérer les processus ou les fichiers), grâce à un mécanisme d'introspection. Cependant, différentes limitations peuvent être constatées selon les approches qui rendent difficile leur adoption ou l'utilisation de méthodes classiques de détection d'intrusions par un TEE :

— Un environnement d'exécution avec des contraintes liées aux ressources à disposition (p. ex., un espace de stockage limité et des faibles ressources CPU).
— Certains TEE, tel que Intel ME, nécessitent des étapes de rétro-ingénierie afin d'ajouter du code [10].
— L'utilisation du TEEs, tel que le SMM, peut avoir un impact négatif en performance sur le reste du système [3].

Une solution pourrait être de déporter une partie des calculs ou de stocker de l'information sur un serveur tiers. Cependant, l'utilisation du réseau depuis un environnement comme le SMM peut se révéler difficile, comme le montre certaines approches [7, 8] qui n'ont pu mettre en place que des communications réseau rudimentaires (envoie de frames Ethernet).

Dans ce stage, on s'intéressera plus particulièrement au SMM, en raison de

sa plus grande accessibilité aux plateformes Intel et AMD, et car il n'apporte pas contraintes de développement comme l'Intel ME.

Objectifs

Dans un premier temps, l'objectif des travaux sera de rédiger un état de l'art de l'utilisation des TEE à des fins de détection d'intrusion et d'analyse de logiciels malveillants. Dans un second temps, des solutions devront être proposées, implémentées et évaluées afin de répondre aux limites qui freinent le déploiement de solutions. Afin de limiter l'impact des contraintes matérielles sur le déroulement du stage, il sera possible de se baser sur de la virtualisation (e.g., QEMU [1]) et d'un firmware open source existant (e.g., EDK2 [2]) pour l'évaluation.

Une première approche pourrait se concentrer sur la mise en place d'un mécanisme de communication réseau sécurisé depuis le SMM via introspection ou coopération. Cela permettrait de démontrer la faisabilité d'approches reposant sur un tiers pour résoudre certains problèmes dues aux contraintes liées aux ressources.

Une analyse pourrait aussi être effectuée pour comprendre les différents moyens qu'un logiciel malveillant évasif aurait à sa disposition pour détecter qu'il est surveillé par un programme au sein du TEE.

Références

[1] Ahmed M Azab et al. "Hypervision Across Worlds: Real-Time Kernel Protection from the ARM TrustZone Secure World". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 90–102.

[2] Erick Bauman, Gbadebo Ayoade, and Zhiqiang Lin. "A Survey on Hypervisor-Based Monitoring: Approaches, Applications, and Evolutions". In: *ACM Computing Surveys (CSUR)* 48.1 (2015), pp. 1–33.

[3] Brian Delgado and Karen L Karavanic. "Performance Implications of System Management Mode". In: *2013 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE. 2013, pp. 163–173.

[4] Bhushan Jain et al. "Sok: Introspections on Trust and the Semantic Gap". In: *IEEE Symposium on Security and Privacy*. IEEE. 2014, pp. 605–620.

[5] Kevin Leach et al. "Towards Transparent Introspection". In: *Proceedings of the 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. Vol. 1. IEEE. 2016, pp. 248–259.

[6] Jiang Wang et al. "Firmware-assisted Memory Acquisition and Analysis Tools for Digital Forensics". In: *Proceedings of the sixth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*. IEEE. 2011, pp. 1–5.

1. `https://www.qemu.org`
2. `https://github.com/tianocore/edk2`

[7] Fengwei Zhang et al. "HyperCheck: A Hardware-Assisted Integrity Monitor". In: *Transactions on Dependable and Secure Computing* 11.4 (2013), pp. 332–344.

[8] Fengwei Zhang et al. "Spectre: A Dependable Introspection Framework via System Management Mode". In: *Proceedings of the 43rd Annual IEEE/IFIP international conference on dependable systems and networks (DSN)*. IEEE. 2013, pp. 1–12.

[9] Fengwei Zhang et al. "Towards Transparent Debugging". In: *IEEE Transactions on Dependable and Secure Computing* 15.2 (2016), pp. 321–335.

[10] Lei Zhou et al. "Nighthawk: Transparent System Introspection from Ring -3". In: *European Symposium on Research in Computer Security*. Springer. 2019, pp. 217–238.