Dear Wenqing Liu,

We are sorry to inform you that your submission

"HyperMI: A Privilege-level Secure Execution Environment for VM Isolation against Compromised Hypervisor"

was not accepted to Inscrypt 2018.

We received 93 submissions this year and only 36 submissions were considered as acceptance (including full papers, conditionally accepted papers and short papers).

Thank you very much for submitting your work to Inscrypt 2018.

Sincerely,

Inscrypt 2018 program chairs

----------------------- REVIEW 1 ---------------------

PAPER: 21

TITLE: HyperMI: A Privilege-level Secure Execution Environment for VM Isolation against Compromised Hypervisor

AUTHORS: Wenqing Liu, Kunli Lin, Kun Zhang and Bibo Tu

Overall evaluation: -1 (weak reject)

----------- Overall evaluation -----------

This paper proposes HyperMI which is a VM isolation framework that is

both "same-privilege level" and defends against compromised

hypervisors. The body of the paper walks the reader through the

design, and a later chapter then reports on the implementation,

including giving timing information (the overhead seems acceptable).

It is not easy for me to review this paper as, despite being

interested in modern CPU design principles, I am not really from that

field. I hence attach a low confidence level to this review. That

said, I think the paper does not do a good job in explaining the

details (or, to say it with the words of the call for papers: the

"level is not appropriate for a non-specialist reader", the paper is

not "self-contained"). Here are examples for this:

- intuitively is seems impossible that one can defend against a

corrupt hypervisor if one sits on the "same privilege level". (On

p.3., occurrences of the words "same privilege level" appear

routinely in double-quotes; not sure what this shall express). The

article does not attempt to clarify why this intuition is wrong. In

the end it remains unclear to me what the challenges were, and

whether they were achieved on a satisfactory level.

- many acronyms are used that are never explained. Examples include:

"VMX operation", "CR3-like register", "ROP attack", "CR0 register",

"WP bit", "DEP mechanism", "SMEP mechanism", "IOMMU". As a reader

I cannot make sense of these notions.

Beyond that, the article is neither clear on the threat model nor on

the targeted security guarantees: The threat model is discussed in

Section 3, but rather than clarifying on against what attacks the

HyperMI framework shall defend, the section rather says against which

attack it does not defend. In Section 4.2 we then read that "accessing

data illegally and falsifying data" are things to be prevented, but

there is not mention of this in Section 3. Concerning the security

guarantees, Section 4.3 doesn't formulate any but rather admits that

"it makes no doubt [sense to] focus on DMA attacks"; it is left open

whether HyperMI protects against these or not.

----------------------- REVIEW 2 ---------------------

PAPER: 21

TITLE: HyperMI: A Privilege-level Secure Execution Environment for VM Isolation against Compromised Hypervisor

AUTHORS: Wenqing Liu, Kunli Lin, Kun Zhang and Bibo Tu

Overall evaluation: -1 (weak reject)

----------- Overall evaluation -----------

The paper proposed HyperMI to provide protection for VMs in a cloud environment against compromised hypervisor. The idea is to remove VM management and EPT operations from the hypervisor and place them into a new world called HyperMI, which is trusted, so that memory management, EPT switching, and VM monitoring can be performed in a secure way. The paper performs experiments to show that HyperMI introduces small overhead in execution of VMs.

My main concern is on the idea of splitting the hypervisor and relocate some of its functionality into HyperMI. Since hypervisor is not trusted due to its complexity and bigger codebase, I'm not sure why splitting it up into two pieces and simply trust one of them (the HyperMI) solves the problem. In other words, I'm not sure why the hypervisor cannot be trusted, while HyperMI is trusted.

Speaking about the assumption and threat model, I'm also not sure what the following means: "And we don’t consider ROP attack for hypervisor and the attack incurred by interaction between VM and software with vulnerabilities".

The paper needs a lot of improvement in its writing. Typos and grammar errors are everywhere, to the extent that I fail to understand certain details and discussions.

There is security analysis but not security evaluation in the paper. Evaluation is only on the performance of the system.

----------------------- REVIEW 3 ---------------------

PAPER: 21

TITLE: HyperMI: A Privilege-level Secure Execution Environment for VM Isolation against Compromised Hypervisor

AUTHORS: Wenqing Liu, Kunli Lin, Kun Zhang and Bibo Tu

Overall evaluation: 0 (borderline paper)

----------- Overall evaluation -----------

In this paper, author presented HyperMI, a software based approach for providing run-time protection to other VMs from compromised Hypervisor. A comparative performance analysis based on prototype implementation on KVM and x86 architecture is also presented.

Author introduced a secure isolation execution environment (HyperMI-World) and implemented VM isolation module and event-driven VM monitoring module. HyperMI-World runs in same privilege level as KVM hypervisor. Implementation introduces lesser overhead compared to few other software based protection strategies.

In the threat model, the author has considered only software based attacks and stated that DoS, hardware based attacks are not feasible under certain time restriction.

Author's approach is to separate sensitive operations between HyperMI-World and Normal-world.

VM isolation module separates VMX root and non-root switch operation (VM entries and VM exits). It protects Extended page table (EPT) data structure, EPT pointer and Virtual Machine Control structure (VMCS) by trapping access to them and moving access code to these structures from Normal-World to HyperMI-World. Address translation capability is moved from Normal-World to HyperMI-World and memory management function is modified to use page marking technique to ensure one physical memory page is owned and accessed by only one guest VM.

Event-driven VM monitoring module captures access to data structure and memory mapping operation.

For ensuring integrity of HyperMI-World and prevent attacks against HyperMI, all mapping entries related to HyperMI-World are removed from Normal-World addresses. Kernel is deprived of access to CR3 register to avoid loading illegal page table and bypassing HyperMI-World. When HyperMI code is running, Kernel code is set as non-executable.

Overall paper is well written with few typos as in "4.1 Overview".