Stepping through Virtual Communication into Virtmon

Gwendolyn Tennell and Dr. Feng Li
Southern University Shreveport, Louisiana
Purdue School of Engineering and Technology
Indiana University-Purdue University Indianapolis

ABSTRACT
While isolation is an important property from a security perspective, virtual machines (VMs) often need to communicate and exchange a considerable amount of data. Research in virtualization technology has been focused mainly on increasing isolation of co-resident virtual machines. The isolation properties of virtualization demand that the shared resources are strictly separated. The machine registers are also restricted; therefore virtual machines are forced to fallback to inefficient network emulation for communication. This research is based upon a stealthy way to communicate between virtual machines and virtual machine managers (VMMs) running on the Arch Linux operating system, while eliminating malware. Virtmon is a para virtualized machine introspection (PVMI). It is a platform upon which users install and load a group of kernel modules. The Virtmon project utilizes the intra-to-exo channel to communicate stealthily between the virtual machine and its virtual machine manager, and the exo-to-intra channel to communicate stealthily between the virtual machine manager and the virtual machine, using a shadow region. The shadow region hides any activity between the monitors and monitor which keeps malware from detecting and hijacking the communication between the two.

Keywords – communication; channel; hypercalls; upcalls; vmexit; vmentry.

INTRODUCTION
- Virtual Machines offer isolation.
- Virtual Machines are restricted, holds a limited amount of data.
- Virtual Machines cannot share resources.
- Virtual Machine Managers cannot communicate to the Virtual Machine.

- Virtmon is a Platform upon which users install and load a group of kernel modules. (Not restricted, allows for unlimited data.)
- Virtmon allows the specification of VMexit conditions such as, accessing a specific file or acquiring the lock of a specific shared data structure in the guest operating system.
- Virtmon’s framework allows the Virtual Machine Manager to communicate to the Virtual Machine.
- The prototype Virtmon is an enhanced Virtual Machine Introspection (VMI); a paravirtualized VMI (PVMI).

METHODS
- Arch Linux Operating system
- Used KVM (Kernel based VM) for VMM
- Used Arch Linux for VM
- Intra-to-exo
- Hypercall induces the VM exit from the intra module. Followed up by massive data exchange between the intra-module and the exo-module at a memory region that is reserved by the intra-module inside the VM.
- Exo-to-intra
- Initiated by VMM injecting events which trigger a callback function that has been previously registered by the intra module.

RESULTS/VIRTMON’S ARCHITECTURE

SYSTEM PARAMETER
READ
UPDATE
KERNEL OBJECT
MORPHING
INDUCE
CALLBACK TABLE
VM

HYPERCALL
VIRTMON

VMM

CONCLUSION
The unrestricted PVMI framework shifts the challenges from bridging the semantic gap, to protecting and hiding the PVMI mechanism. Therefore, communication is secure, allowing undetected assistance from a privileged VMM to a VM. The Virtmon project has not only allowed the VMM to cross communication barriers undetected, but also allows for unrestricted registers, into which more data can be exchanged. Presently Virtmon is a stationary defense. The future goal for Virtmon is to eventually become a moving target defense.

ACKNOWLEDGEMENTS
The author/presenter would like to thank Dr. Feng Li, Wei Peng, Mohammed Mustafa Rangwala, and the Computer Information Technology department faculty and staff of IUPUI for their help and guidance. This work has been supported in part by the National Science Foundation and Department of Defense under NSF Award No. 1262984. The opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation or the Department of Defense.