VM researchers post rude awakening about virtualization security

November 9 2012, by Nancy Owano

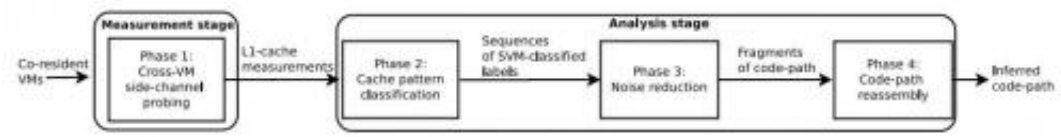

Diagram of the main steps in the side-channel attack. Credit: Yinqian Zhang et al.

(Phys.org)—A virtual machine stealing information from another virtual machine running on the same piece of hardware? That's not supposed to happen. Virtual machines run various tasks on a single computer rather than relying on a separate machine to run each one. The assumption is that one can't eavesdrop or tamper with the other. But now a technique reported in a paper, "Cross-VM Side Channels and Their Use to Extract Private Keys," by Yinqian Zhang of the University of North Carolina, Chapel Hill, and computer scientist colleagues from the University of North Carolina, University of Wisconsin, and RSA Laboratories, suggests a different story.

The researchers said they have completed the first demonstration of a successful side-channel attack on a virtualized, symmetric multiprocessing system, using a virtual machine manager (VMM).

They said it is possible for one VM to steal the cryptographic keys that

are in place to keep data secure from another running on the same physical hardware. This does not paint a happy blue-skies picture for computing facilities that leverage virtualization.

In hours, they recovered the private key for a 4096-bit ElGamal-generated public key using the libgcrypt v.1.5.0 cryptographic library. They extracted the ElGamal decryption key stored on a VM running the GNU Privacy Guard. How it works: Both VMs share the same hardware cache, which stores data for use by the computer processor. The attacking VM fills the cache in a way that the target VM, which is processing a cryptographic key, may overwrite some of the attacker's data. By looking at which parts of the cache are changed, the attacking VM learns about the key in use.

"VM side channels" are likely to become familiar words to those who track security in cloud environments. The authors' technique boiled down to "side-channel analysis," in which a private key is cracked by studying the targeted cryptographic system's behaviors. "In this paper," said the authors, "we present the development and application of a cross-VM side-channel attack," which they further described as an access-driven attack in which the attacker VM alternates execution with the victim VM and leverages processor caches to observe the behavior of the victim. The attack worked only when both attacker and target VMs were running on the same physical hardware or, in virtual computing language, as "co-residents" on a single machine. Co-author Ari Juels of RSA Laboratories said that one of the lessons to be learned is that virtualized machines running highly sensitive workloads should not be placed on the same hosts as potentially untrustworthy virtual machines.

Ways to avoid such exploit headaches in the real world consist of countermeasures that administrators may take to avoid the leakage. One is to use a separate, "air-gapped" computer for high-security tasks.

"In high-security environments, a longstanding practice is to simply not use the same computer to execute tasks that must be isolated from each other, i.e., to maintain an 'air gap' between the tasks. This remains the most high-assurance defense against side-channel (and many other) attacks," the authors wrote.

Other countermeasures may call upon side-channel resistant algorithms; the authors also mentioned "core scheduling." The paper said, "Another defense might seek to modify scheduling to at least limit the granularity of interrupt-based side-channels."

More information: Research paper: www.cs.unc.edu/~reiter/papers/2012/CCS.pdf