A Mechanism to Prevent Side Channel Attacks in Cloud Computing Environments

Tzong-An Su
Fenh Chia University
Taichung, Taiwan

Abstract - Cloud computing provides the benefits of scalability, agility, reliability, security, performance, and maintenance to enterprises and has emerged to become a reality. While cloud computing brings in many benefits it also introduces several security issues. One of the most serious security issues is the side channel attacks which are attacks based on information gained from the physical implementation of a machine. In this paper, we aim at providing a platform to prevent various types of side channel attacks, e.g., cache-based and timing attacks.

Keywords – Side Channel Attack, Cloud Computing, Hardware Virtualization, Virtual Machine

1 Introduction

1.1 Background

Cloud Computing has been a buzzword for the pass few years. The concept combined technologies in networking and computing to provide various services to individuals and organizations. Services like SaaS (Software as a Service), Uaas (Utility as a Service), Iaas (Information as a Service), PaaS (Platform as a Service) and others are among those people are talking about. From the IT's point of view, it offers an approach to expand capacity and capability without investing in new equipment, hiring new personnel, training staff, and licensing new software and adds up to the most important business goal, i.e., cost reduction. In fact, some services like SaaS and PaaS have been out for some time and new services are kept coming. For example, Amazon's Elastic Compute Cloud (EC2) [1], Microsoft's Azure Service Platform [2], Google's App Engine [3], and HP's HP Cloud Services [4].

The technology supports behind cloud computing is not new at all. Two major ones are hardware virtualization and grid computing. Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of distributed resources at runtime. Hardware virtualization creates a virtual machine (VM) acting like a real computer running some operating system on top of a physical computer to host a user's application. The software or firmware which performs the virtualization is call virtual machine monitor (VMM) or hypervisor, e.g., Xen [6] and VMware [7].

1.2 Side Channel Attacks in a Cloud

Original from cryptography, a side channel attack is any attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or the theoretical weakness of the algorithm [25]. The term has been extended to apply to any computing system now. Common side channel attacks can be divided into the following categories:

- Timing Attack
- Power Consumption Attack

- Electromagnetic Attack
- Acoustic Cryptanalysis
- Differential Fault Analysis

In a cloud environment, we believe that only timing attack and differential fault analysis need to be paid attention to. In fact, there have been some research works done based on these two types of attacks in the past few years. We describe two of such works below.

1.2.1 Co-Residence Related Attacks

In [9], authors used the Amazon EC2 as a case study to demonstrate possible side channel attacks in a cloud computing environment. In this study, the network probing technique which uses some popular networking tools like nmap, hping, and wget to gather networking information is employed to collect the host interconnect information of the cloud. With the collected data, the infrastructure of the cloud can be mapped. Given a target instance (VM), the map can offer lots of knowledge of how to select launching parameters to launch attacking instances (VMs) with co-residence property (both target SM and attacking SMs are assigned to the same physical machine).

If attacking SMs can be co-resided with the target on the same physical machine, cross-VM information leakage could happen. After establishing the co-residence relationship with a target SM, attacking SMs can use side channels to steal information from the target.

1.2.2 Thief of Service Attacks

In [26], authors proposed another type of attack which takes advantage of the scheduler vulnerabilities of a VMM and steal the service time of a cloud. They demonstrated the attack on the Amazon EC2 cloud which uses the Xen VMM.

1.3 Research Goal

Side channel attacks have been extensively worked under the traditional computing environment, for example, timing attacks [32, 33], power consumption behavior [34], instruction or data cache behavior [11, 12, 13, 16, 22, 23, 24, 29, 30, 31], branch predictor behavior [14, 15], CPU pipelines (e.g., floating point units) [19, 28], DRAM memory bus [18], scheduling of CPU cores and time-slices, disk access [20]. Due to factors such as core migration, scheduling algorithm, double indirection of memory addresses, and varied CPU configuration parameters in a cloud computing environment, it is somewhat more difficult to realize a cross-VM attacks. But, these attacks are still feasible in a cloud computing environment because the same hardware channels exist. Therefore, on the way to the cloud, we have to pay attention to all these blocks and try to remove them away. Thus, our goal of this research is to find a general, practical, and easy-to – implement approach to counter the attacks described in Section 1.3.

2 Related Research

2.1 Security Mechanism for Virtual Machine Monitor

[36] proposed an architecture that enable administrators to configure virtual machine to satisfy prescribed security goal. [37] described an architecture that gives security tools the ability to do active monitoring while still benefiting from the increased security of an isolated virtual machine. [38] described the work of how to move the domain builder of Xen into a

minimal trusted compartment to make it a trusted computing base (TCB). [39] evaluated a new type of malicious software called virtual-machine-based rootkit, installs a virtual machine monitor underneath an existing operating system and hoists the original operating system into a virtual machine. [40] proposed two techniques to protect the integrity of control flow of a VMM. [41] described, implemented, and evaluated a VMM-based hidden process detection and identification service called Lycosid that is based on cross-view validation principle.

2.2 Side Channel Attacks

The following side channel attacks are found in a traditional computing environment, i.e., a single machine. Resource sharing related attacks might be applied to cloud environments.

The idea of timing attacks first appeared in [42]. [32] presented a timing attack on the OpenSSL's RSA. It used the timing difference of the processing time to derive the key value. [33] presented an implementation with improvement of [42]'s idea and successfully cracked the key of a smart card. [43] investigated how the modern x86 processors can leak timing information through side-channels that related to control flow and data flow. It also implemented a compiler backend to convert/eliminated key-dependent control flow and timing behavior. [19, 28] described functional unit e.g., parallel floating-point multiplier, sharing attacks in a multi-thread environment. [18] described the memory performance attacks in which unfair memory sharing could cause the denial of memory service in multi-core systems. [20] discovered side channels in disk I/O optimization scheme. [34] examined specific methods for analyzing power consumption measurements to find secret key from tamper resistance devices. [14, 15] present a new software side channel attack caused by the branch prediction capability common to all modern high performance CPUs. The extra cycle for a mis-predicted branch can be used for cryptanalysis of cryptographic primitives that employ a data dependent program flow. [44] described several solutions to the problems found in [14, 15].

2.3 Cache-Based Side Channel Attacks

Side channel attacks happen mostly in the cache sharing environment. Several types of attacks have been identified and countermeasures proposed in the traditional computing environment. We survey some works below. [23] proposed several methods to guard against the threats of two types of cache-based attacks, namely, trace-down and time-driven. [24] proposed a hardware cache architecture by using page concept to prevent cache-based attacks. [11] demonstrated complete AES key recovery from knowing plaintext timings of a network server on another computer and proposed advice for AES implementers. [13] demonstrated that in a multi-thread environment, the shared access to memory caches provides not only an easily used high bandwidth covert channel between threads, but also permits a malicious thread to monitor the execution of another thread, allowing thief of cryptographic keys. [16] studied the cache attacks in another type of cache memory, i.e., instruction cache or I-cache. [31] proposed some software-based methods to mitigate cache attacks. Many more attacks and countermeasures can be found in [12, 21, 22, 29, 30].

3 The Prevention Scheme

Cloud computing represents a new computation model and needs new thoughts to uncover its hidden security problems. Considering the above drawbacks existing in those countermeasures proposed before in a traditional computing environment we conclude that they are not practical and are not easy to apply to the cloud. Also, from related works surveyed in Section 2, we know that the side channel attacks mostly arise from the sharing of resources among processes. In a cloud environment, this situation is similar to the co-residence of VMs in a physical host. Since one of the major goals of cloud computing is to share resources, thus, co-residence of VMs is a necessary condition of cloud computing and it will not go away. With this thought and those drawbacks of known solutions described above in mind, our approach should allow co-residence of VMs and should be a general, practical, and easy-to-implement one.

Since co-residence of the attacking VM and the target VM is inevitable, instead of avoiding co-residence of the attacking VM and the target VM, we propose an approach called VM policing which consists of the following components:

- Police VM
- Capability of Police VM
- Scheduling / Dispatching policies of Police VM

In the VM policing approach, special VMs created by the cloud are launched by a physical host at a randomized frequency based on a special police VM scheduling policy. The police VMs are used to "confuse" the attacking VM by executing some clean-up or resource sharing instructions, e.g., cache flush or disk access.

3.1 Police VM

A police VM is a virtual machine launched by the physical host. Its job is to prevent, and handle side channel attacks. Refer to Figure 3.1.

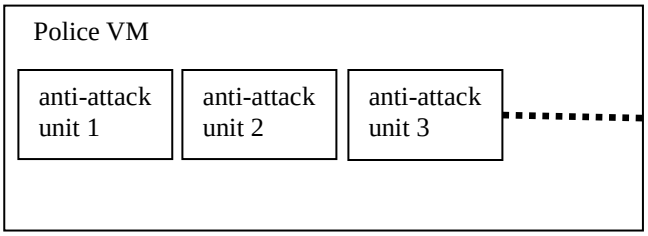

Figure 3.1 A Police VM

Inside a police VM, there could be none or more "anti-attack units". An anti-attack unit is a software component responsible for the prevention and handling of a specific type of side channel attacks, e.g., cache-based attacks. Anti-attack units are installed in a police VM dynamically depending on the needs of a situation. This structure has the following advantages:

- Scalability – The anti-attack units are installed only when necessary. When new types of attacks are found, new units can be developed and installed.

- Varied Execution Timing – The police VM could be used in timing need. Null police VMs (no anti-attack unit) could be created to stub or synchronize executions of VMs.

3.2 Capability of Police VMs

Other than the functionality of the installed anti-attack units, a police SM can also be launched by the physical host to mock the attacking VM by providing an

illusion picture of the cloud. Adding "makeup" information to the cloud would make the probing efforts from attacking VM not accurate and useless.

3.3 Scheduling / Dispatching Policy of Police VMs

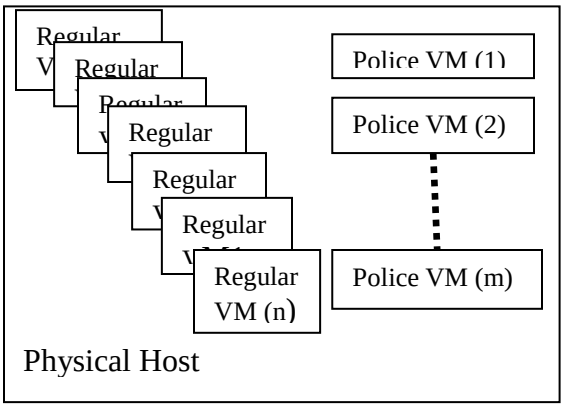

Figure 3.2 Running VMs in a VM Policing Physical

The number of police VMs running in a host (Figure 3.2) and how they are scheduled (Figure 3.3) could be guided by policies made according to the cloud environment. The factors should be considered are as following:

- the load of each host,
- special security request (e.g., running alone on a host),
- performance requirements of VMs,
- irregular VM launches (e.g., burst launches from same users), etc.

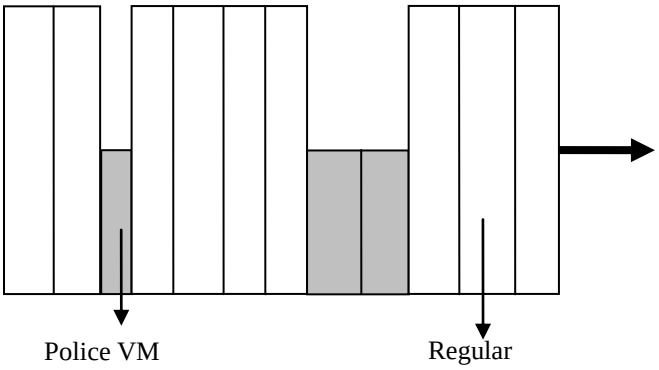

Figure 3.3 Hypervisor Scheduling

4. Conclusions

Cloud computing concept has evolved to become a reality. Not only enterprises begin to adopt this technology but also, some big companies have offer public cloud services. Cloud computing offers technology to consolidate servers, to utilize resources more efficiently, and to add or adjust computing capacity on demand faster. On the way to make cloud computing a more successful technology and let more people benefit from it, one of the major roadblocks is the security issues newly introduced in the cloud. Side channel attacks have been proved to be a type of serious and easy-to-implement attacks in a cloud. There is not much done in this area. We propose the VM policing technique to counter the attacks. It should be an effective countermeasure to the side channel attacks in the cloud and will cleanup some major obstacles in the promotion of cloud computing.

References

[1] http://aws.amazon.com/ec2/
[2] http://www.windowsazure.com/
[3] http://google.com/appengine/
[4] http://hpcloud.com/
[5] http ://gridcomputing.com/
[6] P.Barham, etc., "Xen and the Art of Virtualization", Proceedings of the ACM

Symposium on Operating Systems Principles, pp. 164-177, Oct. 2003.
[7] http://www.vmware.com/
[8] D. Gupta, etc., “Enforcing Performance Isolation Across Virtual Machine in Xen”, ACM/IFIP/USENIX Middleware, 2006
[9] T.Ristenpart, etc., “Hey, You, Get off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds”, ACM CCS, 2009
[10] J. Goguen and J. Meseguer, “Security Policies and Security Models”, IEEE Symposium on Security and Privacy, Apr. 1982.
[11] D. Bernstein, “Cache-Timing Attacks on AES”, http://cr.yp.to/antiforgery/cachetiming-20050414.pdf
[12] D. A. Osvik, A. Ahamir, and E. Tromer, “Cache Attacks and Countermeasures: The Case of AES”, RSA Conference Cryptographers Track (CT-RSA) 2006.
[13] C. Percival, “Cache Missing for Fun and Profit”, BSDCan, Ottawa, 2005.
[14] O. Aciicmez, C. Kaya Koc, and J. P. Seifert, “On the Power of Simple Branch Prediction Analysis”, Proc. 2nd ACM Symposium on Information, Computer and Communication Security, 2007
[15] O. Aciicmez, C. Kaya Koc, and J. P. Seifert, “Predicting Secret Keys via Branch Prediction”, RSA Conference Cryptographers Track CT-RSA 2007.
[16] O. Aciicmez, “ Yet Another Microarchitectural Attack: Exploiting I-Cache”, 14 th ACM Conference on Computer and Communications Security (ACM CCS’07) — Computer Security Architecture Workshop.
[17] D. Grunwald and S. Ghiasi, “Microarchitectural denial of service: insuring microarchitectural fairness”, Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture, 2002.
[18] T. Moscibroda and O. Mutlu, “Memoryy Performance Attacks: Denial of Memory Service in Multi-core Systems”, USENIX Security Symposium, 2007.
[19] O. Aciicmez, and J. P. Seifert, “Cheap Hardware Parallelism implies Cheap Security”, Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC, 2007.
[20] P. Karger and J. Wray, “Storage Channels in Disk Arm Optimization”. IEEE Symposium on Security and Privacy, 1991.
[21] D. Hyuk Woo, and H. H. Lee, “Analyzing Performance Vulnerability due to Resource Denial Service Attack on Chip Multi-processors”, Workshop on Chip Multiprocessor Memory Systems and Interconnects, 2007.
[22] E. Tromer, D. A. Osvik, and A. Shamir, “Efficient Cache Attacks on AES, and Countermeasures”, Journal of Cryptography, July, 2009.
[23] D. Page, “Defending Against Cache-Based Side-Channel Attacks”, Information Security Technical Report, Vol.8, Issue 8, 2003.
[24] D. Page, “Partitioning Cache Architecture as a Side-Channel Defense Mechanism”,IACR Cryptology ePrint Archive, Report 2005.
[25] http://en.wikipedia.org/
[26] F. Zhou, etc., “Scheduler Vulnerabilities and Coordinated Attacks in Cloud Computing”, IEEE International Symposium on Network Computing and Applications, Boston, MA USA, Aug., 2011.
[27] L. Cherkasova, D. Gupta, and A. Vahdat, “Comparison of the three CPU Schedulers in Xen”, SIGMETERICS Performance Evaluation Review, 2007.
[28] Z. Wang and R. B. Lee, “Covert and Side Channels Due to Processor Architecture,”ACSAC 2006
[29] Z. Wang and R. B. Lee, “New Cache Designs for Thwarting Software Cache-Based Side Channel Attacks,” SIGARCH Computer Architecture News, Vol 35, No. 2, pp.494-505, 2007
[30] J. Bonneau and I. Mironov, “Cache-Collision Timing Attacks Against AES,” Cryptographic Hardware and Embeded Systems CHES 2006, LNCS 4249, Springer, 2006.
[31] E. Brickell, etc., “Software Mitigations to Hedge AES Against Cache-Based Software Side Channel Vulnerabilities”, IACR ePrint Archive, Report 2006/052, 2006.
[32] D. Brumley, etc., “Remote Timing Attacks are Practical,” Proceedings of the 12th Conference on USENIX Security Symposium, 2003
[33] J. F. Dhem, etc., “A Practical Implementation of the Timing Attack,” CARDIS, 1998.

[34] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," Advances in Cryptology – CRYPTO 99, LNCS 1666, Springer-Verlag, 1999.
[35] D. Ongaro, A. Cox, and S. Rixner, "Scheduling I/O in Virtual Machine Monitors," ACM, VEE, 2008.
[36] S. Reuda, Y. Sreenivasan, and T. Jaeger, "Flexible Security Configuration for Virtual Machines," ACM Workshop on Computer Security Architectures, 2008
[37] B. Payne, M. Carbone, M. Sharif, and W. Lee, "Lares: An Architecture for Secure Active Monitoring Using Virtualization," 2008 IEEE Symposium on Security and Privacy.
[38] D. Murray, G. Milos, and S. Hand, "Improving Xen Security through Disaggregation," ACM, VEE, 2008
[39] S. King, P. Chen, etc., "SubVirt: Implementing Malware with Virtual Machines," IEEE Symposium on Security and Privacy, 2006.
[40] Z. Wang and X. Jiang, "HyperSafe: A Lightweight Approach to provide Lifetime Hpervisor Control-Flow Integrity," IEEE Symposium on Security and Privacy, 2010.
[41] S. Jones, etc., "VMM-based Hidden Process Detection and Identification using Lycosid," ACM, VEE, 2008.
[42] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," Advances in Cryptography – CRYPTO '96, Vol. 1109 of LNCS, Springer, 1996
[43] B. Coppen, etc., "Practical Mitigations for Timing-Based Side-Channel Attacks on Modern x86 Processors," IEEE Symposium on Security and Privacy, 2009
[44] G. Agosta, etc., "Countermeasures Against Branch Target Buffer Attacks," 4 th Workshop on Fault Diagnosis and Tolerence in Cryptography (FDTC'07)