ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Exploiting Virtual Machine Vulnerability in Third-Party Clouds With Competition For I/O Resources

Buddha Tejaswi
M.Tech Student,
Department of CSE,
PRAGATI Engineering College,
Surampalem, E.G District, AP-533437.

Dr. M. Radhika Mani
Associate Professor,
Department of CSE,
PRAGATI Engineering College,
Surampalem, E.G.District, AP-533437.

ABSTRACT:

Multiple virtual machines (VM) share the same physical resources (e.g., CPUs, caches, DRAM, and I/O devices). All the application allocated to individual VM separate from another. At the time performance weakness will occur. Performance weakness caused by struggle between virtual I/O workloads i.e., by increase the competition for shared resources and another could purposely slow down the execution of a targeted application in a VM. For that the increase model of cloud computing, e.g., Amazon Elastic Compute Cloud (EC2), provide a stretchy strong environment for large-scale applications. The focus on I/O resources such as hard-drive throughput and/or network bandwidth - which are important for data-intensive applications.Swiper: the framework which uses a carefully planned workload to incur significant delays on the embattled application and VM with lowest cost (i.e., resource consumption).

KEYWORDS:

Co-location, Synchronization, Exploiting, VMClient and Server.

1.INTRODUCTION:

A cloud computing system offers to its users the illusion of “infinite” computing and storage capacities on an on-demand basis. Examples of commercial cloud computing platforms include Amazon Elastic Compute Cloud (EC2) and Simple Storage Service (S3), Google App Engine, Microsoft Azure, etc. Virtualization plays a vital role in cloud computing. In particular, for the purpose of scalability and flexibility of resource delivery, a cloud computing system does not provide each user with a different physical machine - instead, it allocates each user to an independently managed virtual machine (VM) which can be dynamically created, modified, and migrated.

Examples of such a platform include Xen VM for Amazon EC2 and the .NET-based run-time environment for Microsoft Azure.The essence of virtualization is that multiple VMs may multiplex and share the same physical resources (e.g., CPU, cache, DRAM, and I/O devices). Nonetheless, each VM is supposed to enjoy isolation (in terms of security and performance) from the other VMs. That is, different VMs should not be able to interfere with the executions of each other. Unfortunately, the lack of physical isolation can indeed pose new security threats to co-located VMs. In this paper, we consider a new type of VM vulnerability which enables a malicious user (i.e., VM) to exploit the resource contention between co-located VMs and obstruct the execution of a targeted application running in a separate VM that is located on the same physical machine as the malicious one. In particular, we focus on exploiting contentions on shared I/O resources that are critical to data-intensive applications - e.g., hard disks and networks. In practice, service providers often exclude such threats from their service level agreement (SLA). That is, customers are solely responsible for their loss caused by resource contention from co-located VMs. Most service providers do not enable dynamic migration for user control.

2.EXISTING SYSTEM:

In this work, we also evaluate our framework on KVM (Kernel-based Virtual Machine) that utilizes hardware assisted full virtualization instead of Xen’s par virtualization. Although Xen and KVM are used to demonstrate this threat in our work, our test and previous work indicate that other virtualization framework like VMware also exhibits similar interference problem.An I /O-based co-location detection technique and verified its effectiveness on public clouds. A discrete Fourier transformation (DFT) based algorithm which recovers the victim’s original I/O pattern from the observed (distorted) time-series of I/O throughout, and then determines if the victim application has reached a pre-determined point when it is

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

most vulnerable to an exploitation. Discover patterns which cause maximum interference. A theoretical framework to observe and synchronize with predefined I/O patterns. A comprehensive set of experiments on Amazon EC2 - with the results clearly showing that Swiper is capable of degrading various server applications by 22.54% on average (and up to 31%) for different instance types and benchmarks, while keeping the resource consumption to a minimum. The remainder of this paper is organized as follows. Introduces the system and threat models. Presents our IO-based co-location detection method. Describes our approach, and explains the synchronization and exploiting stages. Sec. 5 discusses issues in practicing Swiper. Presents the experiments and results, including VM co-location in Amazon EC2. We conclude in Sec. 7. Interested readers may also access the Appendices for additional details on the design, implementation and evaluation.

DISADVANTAGE:

1.A straightforward way to delay a victim process is to launch an attacking process which constantly requests a large amount of resources shared with the victim.
2.A critical problem for the adversary is the cost because now the attacker needs to launch a number of probing VMs to search for the target after the VM migration.
3.Attack can be easily detected and countered (e.g., a dynamic resource allocation algorithm can restrict the amount of resources.

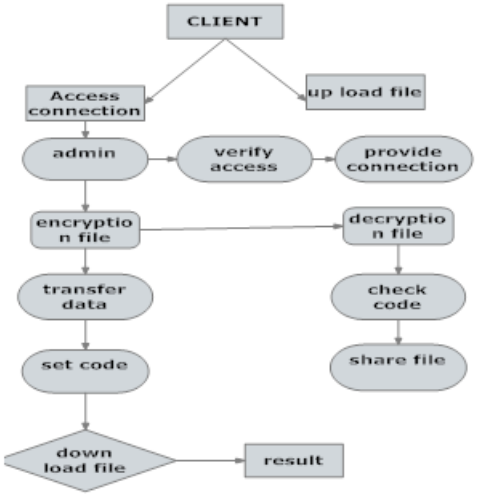
4.PROPOSED SYSTEM:

A relevant prior work that proposed to exploit the contention on hard disks required access to the hard-disk queue in order to analyze the requests from both the adversary and the victim. The proposed lock-on approach is feasible on public clouds. For example, our experiments on Amazon EC2 us-east-1c zone show the success rates of about 8% and 2% for the probing and the locking-on stages respectively. An important feature for virtualized systems to manage resources. Co-locating the target and attacker is critical in the proposed method. Since the target VM could be migrated. The effectiveness metrics of three selected applications are shown In Table 1. The attacker's data usage is limited at 500 MB. The proposed peak attack clearly captures I/O request patterns and achieves additional performance degradation on both Xen and KVM. We have proposed a number of possible solutions to these types of attacks as future work.

ADVANTAGE:

1.Historical traces could help in predicting I/O behaviors, self-learning and adaptively to new I/O patterns are still good to have in a fast-changing world.
2.The distribution now is skewed to higher values.
3.The higher demand the victim has at a given time, the larger request the adversary should submit to the shared resource.

4.SYSTEM ARCHITECTURE:



PROBLEM STATEMENT:

Given a workload fingerprint of a victim process, determine an adversarial workload of I/O request which incurs the maximum delay on the victim process with-out exceeding the pre-determined threshold on the adversary's own resource consumption.

5.MODULES:

1.Hosting
2.Representational state transfer
3.Splitting
4.Download File
5.Multi-Access

5.1. MODULES DESCRIPTION:

5.1.1. Hosting:

In this Module our client application will be request to the cloud connection. Then the Server will provide the access to the client machine.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The hosting service must include system administration since it is shared by many users; this is a benefit for users who do not want to deal with it, but a hindrance to power users who want more control. In general shared hosting will be inappropriate for users who require extensive software development outside what the hosting provider supports. Almost all applications intended to be on a standard web server work fine with a shared web hosting service. But on the other hand, shared hosting is cheaper than other types of hosting such as dedicated server hosting. Shared hosting usually has usage limits and hosting providers should have extensive reliability features in place.

5.1.2.Representational state transfer:

First the client get permission from the Cloud Server Access to upload the file then the keyword will be implemented for the file and it will be encrypted. In order for the adversary to incur the maximum delay under a resource constraint, it has to be able to determine whether the victim process is running, and predict the resource request from the victim process at a given time.

5.1.3.Split File:

The Encrypted File will be spited into Multiple Files and then it will be save in diffident location. It is used to Squire the File which couldn't Be Access by unauthorized user itself. Files come in a wide variety of materials, sizes, shapes, cuts, and tooth configurations. The cross-section of a file can be flat, round, half-round, triangular, square, knife edge or of a more specialized shape. There is no unitary international standard for file nomenclature; however, there are many generally accepted names for certain kinds of files.

5.1.4.Download File:

The Client should first get access from the server then only he can use the option Download. Next the Client machine can access the file by selecting and give the password. If the password is wrong we can't access the file the password will be save as Encrypted Format. Then he can access the file by download or view it. While downloading the file the password key should be send entered and in the same way the each and every time the client should request to the server after the acknowledgement of the server then only the user can access the download option.

The client can choose the file name and he can view the Encrypted spitted file. If the user type the correct password then only he can access the file. He can copy the content and paste in his own software or he can also download the document file.

5.1.5.Multi-Access :

While multiple user access one content file normally the Delay will accrue to overcome this we can implement multiple file access concept. Using this one file can send to multiple user fast and frequently. The file will be in spited format so we can send one to first and the next file to another user in random method process. This reduce the time access and deadlock of the networking process.

6.PRACTICAL ISSUES IN RUNNING SWIPER :

We have established a framework to locate and in-terfere with target VMs, including a theory for syn-chronizing I/O patterns. There are critical issues that need to be addressed when deploying Swiper in real-world. We explicitly discuss two important factors in this section. First, some applications' activities depend on user inputs. Second, migration is an important feature for virtualized systems to manage resources. Co-locating the target and attacker is critical in the proposed method.

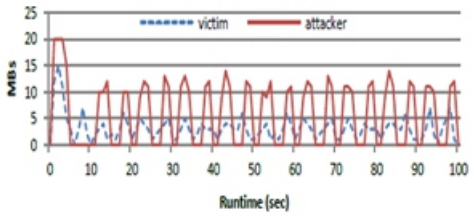


Fig. 1: Overlapping I/O of an attacker and a victim

7.EXPERIMENT RESULTS:

Because a substantial portion of Amazon EC2's address space hosts publicly accessible web servers, we test Swiper with the following popular cloud applications or benchmarks: YCSB (Yahoo! Cloud Serving Benchmark) is a performance measurement framework for cloud data serving. YCSB's core workload C is used to emulate read-intensive applications; Wiki-1 and Wiki-2 are running Wiki bench with real Wikipedia re-quest traces on the first day of September and October 2007 respectively;

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Darwin is an open source version of Apple's QuickTime media streaming server; File Server mimics a typical workload on a file system, which consists of a variety of operations (e.g., create, read, write, delete) on a directory tree; Video Server emulates a video server, which actively serves videos to a number of client threads and uses one thread to write new videos to replace obsolete videos. Web Server mostly performs read operations on a number of web pages, and appends to a log file.

The File Server, Video Server and Web Server belong to the File Bench suite. Micro and small Amazon EC2 instances and a local machine are used as the test platforms in this work. We use technique described in Sec. 3 to locate Amazon EC2. Instances, which dwell in the same storage device. The tests are repeated for 50 times and the means are reported.

8.CONCLUSION:

In this paper, we presented a novel I/O workload based performance attack which uses a carefully designed workload to incur significant delay on a targeted application running in a separate VM but on the same physical system. Such a performance attack poses an especially serious threat to data-intensive applications which require a large number of I/O requests. Performance degradation directly increases the cost of per workload completed in cloud-computing systems.

Our experiment results demonstrated the effectiveness of our attack on different types of victim workloads in real-world systems with various number of VMs. Interested readers may refer to Appendix I for the literature review and more discussions, where we have proposed a number of possible solutions to these types of attacks as future work. Also, it would interested to study the effects of system parameters, e.g., I/O schedulers and buffer sizes, on defending such attacks

.9. REFERENCES:

[1] M. Armbrust et al., "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.

[2] M. Rosenblum et al., "Virtual machine monitors: Current technology and future trends," IEEE Computer, vol. 38, pp. 39–47, 2005.

[3] Amazon, "Ec2 sla." [Online]. Available: http://aws.amazon.com/ec2-sla/

[4] V. Varadarajan et al., "Resource-freeing attacks: improve your cloud performance (at your neighbor's expense)," in CCS, 2012, pp. 281–292.

[5] R. Kohavi et al., "Online experiments: Lessons learned," Computer, vol. 40, no. 9, pp. 103–105, 2007.

[6] T. Ristenpart et al., "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in CCS, 2009.

[7] A. Greenberg et al., "The cost of a cloud: research problems in data center networks," SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 68–73, Dec. 2008.

[8] E. Deelman et al., "Clouds: An opportunity for scientific applications?" in High Performance Computing Workshop, 2008, pp. 192– 215.

[9] S. K. Barker et al., "Empirical evaluation of latency-sensitive application performance in the cloud," in Proceedings of the first annual ACM SIGMM conference on Multimedia systems. ACM, 2010, pp. 35 46.

[10] K. Ye et al., "Virtual machine based energy-efficient data center architecture for cloud computing: a performance perspective," in Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing. IEEE Computer Society, 2010, pp. 171–178.