ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

A Survey on Security Improvements in VM Allocation of Cloud Computing to Defend Attacks

Revati Vijay Rajane, Prof.Pradnya Kasture

M.E Student, Dept. of Computer Engg., RMD Sinhgad School of Engineering, SavitribaiPhule Pune University, Pune, Maharashtra, India

M.E Student, Dept. of Computer Engg., RMD Sinhgad School of Engineering, SavitribaiPhule Pune University, Pune, Maharashtra, India

ABSTRACT: In recent years, cloud computing has become a popular paradigm for hosting and delivering services over the internet. The key technology that makes cloud computing possible is server virtualization, which enables dynamic sharing of physical resources. Cloud computing provides opportunity to dynamically scale the computing resources for applications. However, customers can face new security risks when they use cloud computing platforms like co-resident attacks. The co-resident attack, where malicious users aim to co-locate their virtual machines (VMs) on the same server and extract confidential information from the virtual machine. This report specially introduces security scheme policies of VM allocation policies. Our analysis shows the deploying three policies, the cloud provider decreases the attacker's possibility of achieving co-location by having a policy, where each policy is selected with a certain probability. These solutions do not require any changes to underlying infrastructure. However, most of these methods are not suitable for immediate deployment due to the required modifications to current cloud platforms. Aim is to solve the problem from a unique perspective, by studying how to improve the virtual machine allocation policy, so that it is difficult for attackers to co-locate with their targets**.**

KEYWORDS: Cloud computing, resource allocation, Co-resident attack, virtual machine allocation policy, security metrics

I. INTRODUCTION

Cloud is a group of computers or servers which are interconnected together to provide resources to the clients. It emerges as a brand-new computing paradigm that aims to supply reliable, custom-made and QoS (Quality of Service) warranted computing dynamic environments for the end customers. There are numerous advantages of cloud computing, the most basic ones being lower costs, re-provisioning of resources and remote accessibility. Cloud computing lowers cost by avoiding the capital expenditure by the company in renting the physical infrastructure from a third-party provider. Due to the flexible nature of cloud computing, one can quickly access more resources from cloud providers when one needs to expand our business.

Security is one of the major concerns that arise from the use of cloud computing systems. From the customers perspective, while migrating to the cloud brings a number of advantages, including higher availability, better scalability and lower maintenance overhead, it also means customers are exposed to additional risks brought about by the other tenants with whom they share the resources. In cloud environments, it is almost unavoidable that any user shares one or multiple types of resources with other users computing, networking and storage.

This paper presents a survey on VM allocation policy. A security policy that can handle the co-resident attack and also satisfies requirements for workload balance and low power consumption. The main purpose is to design a secure VM allocation policy, in order to mitigate the threat of co-resident attacks. Specifically, to determine whether an allocation policy is secure based on the following three metrics 1) Efficiency 2) Coverage 3) Vmin.

DOI: 10.15680/IJIRCCE.2017. 0505289

Virtualization

Virtualization is a technique, which allows to share a single physical instance of a resource or an application among multiple customers and organizations. It does by assigning a logical name to a physical storage and providing a pointer to that physical resource when demanded. By using virtualization, all severs and the software application which are required by other cloud providers are maintained by the third-party people, and the cloud providers has to pay the money on monthly or annual basis.

Co-resident Attacks

The co-resident attacks discussed here comprise the following two steps. First, the attacker has a clear set of target VMs, and their goal is to co-locate their VMs with these targets on the same physical servers. Second, after co-residence is achieved, the attacker will construct different types of side channels to obtain sensitive information from the victim. The co-resident attack that is considered here, before the attacker is able to extract any private information from the victim, they first need to co-locate their own VMs with the target VMs.

However, the existence of co- resident attacks severely breaches confidentiality. Hence, it is important to effectively prevent side channels from being built, and mitigate the impact of co-resident attacks.

A. *Coarse Grained Side Channels*

Prime-Probe is a commonly used technique for constructing cache-based channels. It mainly consists of three steps: (1) prime the attacker fills one or more cache sets by reading from a specific memory region; (2) idle the attacker waits for a predefined period of time, while the cache is used by other tenants, including the target being monitored; (3) probe the attacker refills the cache sets by reading from the same memory region. During the second stage, if there is much cache activity from the target, it is likely that the attacker's data will be evicted from the cache. As a result, compared with the situation where the target does not use the cache, the read time in the last step will be significantly higher in this case.

B. *Fine Grained Side Channels*

This includes cache observations to particular operations of the victim. for many types of victim operations, the fine granularity achieved by the attack VM's IPI Inter Process Interrupts based spying can yield multiple observations per individual operation.

II. RELATED WORK

In [1] This paper presents a design of new balanced policy Previously-selected-servers-first (PSSF). The key idea is to give a higher priority to servers that already host or once hosted VMs from the same user, when a new VM request is being processed. In addition to security, another two practical issues workload balance and power consumption are also taken into consideration, for the proposed policy to be more applicable to existing cloud computing. Security is the focus of this paper, and ultimate goal is to substantially decrease the efficiency and coverage rates for the attacker. In order for the proposed policy to be applicable to existing commercial cloud platforms, other issues such as workload balance and power consumption should also be taken into consideration.

In [2] In this paper states how the attacker is likely to behave under the various VM allocation policies, identify the potential differences between the behaviours of attackers and legal users, and integrate the findings into a defence mechanism that substantially increases the overall cost for attackers, and hence further mitigates the impact of co-resident attacks. Note that here explicitly add the concept of the financial cost for launching an attack, which is a necessary supplement when purely technical solutions do not suffice. This gives an analysis of the attacker's behaviours in various circumstances. The attacker's best strategy is to keep creating new accounts, each of which starts one VM. In order to prevent the attacker from achieving co-residence by behaving in this manner. The basic idea is that it explicitly models the risk associated with an account. When a new account is created, it is considered to be a medium risk. By monitoring its behaviour over time, the system will mark it as low risk or high risk, or keep considering it as medium risk. VMs of any account will only co-locate with VMs created by accounts with the same label. In this way, if it is assuming that all target accounts are marked as low risk, then the attacker cannot use newly created accounts to start VMs immediately, as these accounts will be labelled" medium risk" and their VMs will never co-locate with their targets. Instead, the new account must meet various criteria in order to be considered as low risk.

In [3] This paper focuses on the initial VM allocation within a data centre. Although dozens of algorithms have been proposed based on various criteria and goals, judging from the final allocation process they can be generally classified into two types: stacking and spreading. In other words, the VMs are either concentrated to a number of physical servers, in order to decrease the power consumption and maximize the utilization rate, or distributed across the whole data centre, for the purpose of workload balance and higher reliability.

Table 2.1 summarizes some commonly used or widely cited allocation.

Name	Descriptions	Type
First Fit	All servers are ordered by their identifier, and a new VM is allocated to the legitimate server (i.e., the server with enough remaining resources and satisfying all other requirements if there are any) with the smallest identifier	Stacking
Workload Stacking	One example is to allocate a new VM to the legitimate server with the most number of VMs (started by any user) This is what we call Most VM policy.	Stacking
Random	The simplest policy that selects at random from those legitimate servers	Spreading
Next Fit	Similar to First Fit, except that the search begins from the server that was last selected	Spreading

In [4] This paper introduces a game theoretic approach that mixes multiple VM allocation policies, so that it will be difficult for attackers to identify any pattern in the allocation process. Therefore, they will not be able to find one single most effective way to spread their VMs, and hence the probability for them to achieve co-residence will be reduced.
In addition, in this chapter it is taken into consideration another two important practical issues other than security: workload balance and power consumption. Workload balance: Workload here refers to the VM requests. From the cloud provider's point of view, spreading VMs among the servers that have already been switched on can help reduce the probability of servers being over-utilised, which may cause SLA (service level agreement) breaches. From the customer's perspective, it is also preferable if their VMs are distributed across the system, rather than being allocated together on the same server. Otherwise, the failure of one server will impact all the VMs of a user. Power Consumption: Managing the servers in an energy efficient way is crucial for cloud providers in order to reduce the power consumption and hence the overall cost.

II. Existing Countermeasures against Co-Resident Attacks

Previous studies have proposed the following five types of possible defence methods against co-resident attacks:
1. **Eliminating side channels, and preventing sensitive information from being transferred between co-resident VMs:** Side channel attacks are not unique to cloud systems. Prior to the popularization of cloud platforms, different methods [3], [6] had already been proposed to mitigate the threat of side channels. However, these methods are at the hardware layer, and hence are normally costly to adopt. In cloud environments, many side channels rely on high resolution clocks, therefore, [2] propose to remove such clocks; and [3] choose to add latency to potentially malicious operations; while the approach of stated in [4] is to eliminate all internal reference clocks. An alternative solution is to enforce isolation by preventing the sharing of sensitive resources [5] use page-colouring to limit cache-based side channels.Nevertheless, the problem with these methods is that they often require substantial changes to be made to existing cloud platforms, and hence are unlikely to be adopted by cloud providers any time soon. More recently, [7] propose to perform periodic time-shared cache cleansing, in order to make the side channel noisy.

2. Increasing the difficulty of verifying co-residence: As it is introduced earlier, the easiest way of detecting the co-residence of two VMs is to perform a TCP traceroute operation from one VM to the other, and check whether the two Dom0 IP addresses are the same. Cloud providers can prevent Dom0s IP address from being exposed to customers [1], so that attackers will be forced to resort to other options that do not rely on network measurements, and often require greater effort. In fact, according to [8] modern public clouds have already adopted new techniques to thwart certain co-residence detection techniques. However, they are still not sufficient to solve the problem of co-residence attack, as more and more different methods of detecting co-residence have been proposed [9], [5], [4].
3. Detecting the features of co-resident attacks: It is observed that when attackers use the Prime-Probe technique to extract information from the victim, there are abnormalities in the CPU and RAM utilization, system calls and cache miss behaviours. They propose different methods to detect these abnormalities, and design the defence mechanisms accordingly.
4. Migrating VMs periodically: This problem is tackle by applying a Vickrey-Clarke-Groves (VCG) [6] mechanism to migrate VMs periodically. Specifically, they discuss the number of VMs to be migrated as well as the destination hosts. In addition, they propose a method to generate a VM placement plan, in order to decrease the overall security risk. However, frequently migrating VMs can cause extra power consumption, and may lead to performance degradation, which increases the probability of cloud providers breaking their SLA (service level agreement).
5. Using VM allocation policy to make it difficult to achieve co-residence: This is the approach is taken in this paper. also consider how to use the VM allocation policy to defend against co-resident attacks. Their co-location resistant (CLR) algorithm labels all servers as either open or closed, where open (closed) means the server can (cannot) receive more VMs. At any time, CLR keeps a fixed number (Nopen) of server's open, and allocates a new VM to one of these servers randomly. If the selected server cannot take more VMs due to this allocation, it will be marked closed, and a new server will be opened.

III. PROPOSED ALGORITHM

A. *Definition security metrics:*

In order to quantitatively analyse different VM allocation policies, in terms of their abilities in defending against the co-resident attack, the following three security metrics are defined.
1. *Efficiency*
For attackers, clearly it is desirable to co-locate with as many targets as possible by starting the minimum number of VMs. Hence, Efficiency is defined as the gains divided by the costs. More precisely speaking, it equals the number of servers on which malicious VMs are collocated with at least one of the T targets, divided by the total number of VMs launched by the attacker, i.e.,

$$\text{Efficiency}(|VM(A,t)|) = \text{Servers}(\text{SuccVM}(A,t))|/\ |VM(A,t)|$$

The reason why —Servers(SuccVM(A,t))— instead of just —SuccVM(A,t)— is used that when two malicious VMs co-locate with the same target, the second VM should not be counted. Note that the focus is to prevent attackers from co-locating with their targets, and consider that once co-residence is achieved, attackers are able to construct side channels. Although a second co-resident VM can make it easier for attackers to extract sensitive information from the victim.
2. *Coverage*
Another criterion to measure the success of an attack is the percentage of the conquered targets, i.e., Coverage, which equals the number of target VMs co-located with malicious VMs started in the attack, divided by the number of targets T, i.e.,

$$\text{Coverage}(|VM(A,t)|) = |(\text{SuccTarget}(A,t))|/\ T$$

3. VMmin

This is defined as the minimum number of VMs that the attacker needs to start so that at least one of them co-locates with at least one target. It is an estimate of the minimum effort an attacker has to take in order to achieve co-residence.

Illustration

The following example illustrates the definitions of attack efficiency and coverage.

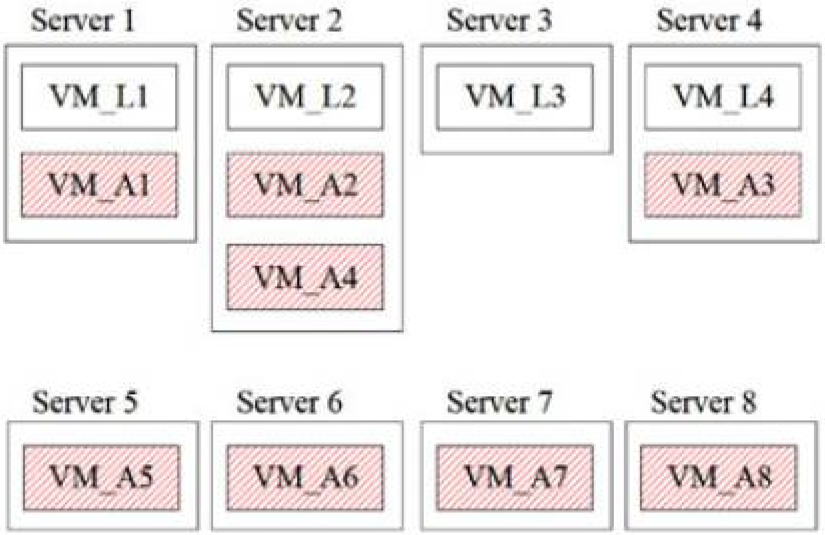

Fig. 1. An example to explain attack efficiency and coverage.

a legal user L starts four VMs (VM_L1, VM_L2, VM_L3 and VM_L4), and they are running on four different servers (Server 1, Server 2, Server 3 and Server 4). Then attacker A starts eight VMs, (VM_A1, VM_A2, . . ., VM_L8), four of which co-locate with three VMs of L. In this case, the attack efficiency is 3/8 (instead of 4/8, as VM_A2 and VM_A4 co-locate with the same target VM_L2), and the coverage is 3/4.

B. *Description of Proposed Policy*

PSSF (Previously Selected Server First)

A new policy named PSSF (Previously Selected Server First) is proposed that takes into consideration all three aspects of security, workload balance and power consumption it mitigates the threat of co-resident attacks, but also satisfies the constraints in workload balance and low power consumption. In addition, large scale experiments are performed to demonstrate the effectiveness of the policy. A defence mechanism is designed that further improves the effectiveness of the PSSF policy, by including a machine learning approach based on clustering analysis and semi-supervised learning. Our experimental results suggest that once the mechanism is deployed, the overall costs for the attacker will be substantially increased.

This paper introduces step by step how PSSF satisfies the objectives of security, workload balance and power consumption.

1.Security In order to minimise the average number of users per server, when a user ui creates new VMs, they will first be assigned to those servers that already host or once hosted VMs started by ui (i.e., previously selected servers).

2.Workload balance Recall that the workload balance is maximised if the VMs are distributed evenly across a large number of servers, which conflicts with the security objective. As a compromise, previously selected servers can at most host N* VMs of any user, where N* is less than the capacity of the server. In more detail, the new VMs will not be assigned to previously selected servers in the following three circumstances: (i) every previously selected server already hosts N* VMs of ui, (ii) none of the previously selected servers has enough resources left, and (iii) the user has never started VMs before. In these three cases, PSSF will spread the workload instead, e.g., choose the servers with the least number of VMs.

3.Power consumption One main reason why the Least VM policy and the Random policy perform poorly in power consumption is that an excessive number of servers are switched on. The most straightforward way to minimize the number of running servers is stacking, or in other words, allocating new VMs to the same server until there is not enough remaining resources. However, clearly this breaks the rule of workload balance. Therefore, a compromise solution is proposed: logically divide all servers into groups of NG; within each group, the workload is spread; the next group of servers will not be started until servers in all the former groups are fully utilized. If less power consumption is desired, then more groups are formed (i.e. Smaller NG) but at the cost of decreased workload balance. Conversely, a smaller number of groups (i.e., larger NG) will result in greater spread of VMs, and hence more power consumption but better workload balance. For simplicity, the group index equals to the servers index, i.e., 0, 1, , K-1, divided by the group size NG.

C. *Description of the Proposed Algorithm:*

1. Security

In order to minimise the average number of users per server, when a user ui creates new VMs, they will first be assigned to those servers that already host or once hosted VMs started by ui (i.e., previously selected servers).

Workload balance

In the following three circumstances, the new VMs will not be assigned to previously selected servers:

(1) every previously selected server already hosts N VMs of ui,

(2) none of the previously selected servers has enough resources left

(3) the user has never started VMs before. In these three cases, PSSF will spread the workload instead, e.g., choose the servers with the least number of VMs.

```
PSSList = {}, NPSSList = {}
foreach server si in S
    if (si has enough remaining resources)
        if (si already hosts or once hosted u’s VMs)
            if (si hosts less than N*of u’s VMs)
                PSSList.add(si)
        else
            NPSSList.add(si)
if (!PSSList.isEmpty())
    return PSSList.get(random (PSSList.size()))
else
    Sort (NPSSList, group index, resources left)
    i = the number of servers with the same group index and remaining resources as the first server in NPSSList (NPSSList.get(0))
    Mark NPSSList.get(random(i)) as “previously selected” for u, and return it
```

IV. Pseudo Code

```
Step 1: Assign previously selected server's List as PSSList
Step 2: Assign not previously selected server's List as NPSSList
Step 3:  loop through PSSList
Step 4:   if Server has enough remaining resources
Step 5: if Server already hosts less than Vmin.
Step 6: Then add server to PSSList.
Step 7: else
Step 8: add server to NPSSList.
Step 9:        End
Step 10: PSSList. Is not Empty
Step 11:  Then return PSSList
Step 12:   Else
Step 13:  Sort NPSSList and find resources left.
Step 14:  Mark that resource as previously selected
Step 15: End.
```

V. Conclusion and Future Work

This paper proposes a defense mechanism against the co-resident attack in cloud computing environments. The mechanism exploits the behavioral differences between attackers and normal users, and classifies all users into three categories low/medium/ high risk by applying clustering and semi-supervised learning techniques. In this way, attackers are forced to behave similarly to legal users (their targets), as it requires that only VMs belonging to the same type of users can co-locate with each other. In addition, the defense mechanism builds on the earlier work of a secure VM allocation policy PSSF. The integration of PSSF makes sure that it is difficult for malicious users to co-locate with their targets, even if they are classified as low risk. increases the complexity will increase.

VM live migration further complicates the problem of defending against the co-resident attack, as the attacker may exploit a loophole in these algorithms. So, future scope contains consideration of this possibility. Moreover, PSSF policy can be further improved to handle the concerns like the possibility of servers being overloaded and considering the fact that not all running servers consume the same amount of power.

References

1. ” Using Virtual Machine Allocation Policies to Defend against Co-Resident Attacks in Cloud Computing”,Yi Han, Jeffrey Chan, Tansu Alpcan, and Christopher Leckie,”IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 14, NO. 1, JANUARY/FEBRUARY 2017”

2. Y. Han, J. Chan, T. Alpcan, C. Leckie, and B. I. P. Rubinstein, A Game Theoretical Approach to Defend against Co-resident Attacks in Cloud Computing: Preventing Coresidence using Semi-supervised Learning, IEEE Transactions on Information Forensics and Security December 2015

3. Virtual Machine Allocation Policies against Co-resident Attacks in Cloud Computing Proc. IEEE International Conference on Communications (ICC 2014), pp. 786-792,2014.

4. Security Games for Virtual Machine Allocation in Cloud Computing Proc. Conference on Decision and Game Theory for Security (GameSec 2013), pp.99-118, 2013

5. T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, ”Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds,” Proc. 16th ACM Conference on Computer and Communications Security (CCS 2009), pp.

6. J. Wu, L. Ding, Y. Lin, N. Min Allah, and Y. Wang, ”XenPump: A New Method to Mitigate Timing Channel in Cloud Computing,” Proc. Fifth IEEE International

7. A. Aviram, S. Hu, B. Ford, and R. Gummadi, ”Determinating Timing Channels in Compute Clouds,” Proc. ACM Workshop on Cloud Computing Security Workshop

8. M. Li, Y. Zhang, K. Bai, W. Zhang, M. Yu, and X. He, ”Improving Cloud Survivability through Dependency based Virtual Machine Placement,” Proc. International

9. V. Varadarajan, Y. Zhang, T. Ristenpart, and M. Swift, ”A Placement Vulnerability Study in Multi-Tenant Public Clouds,” Proc. 24th USENIX Security Symposium, pp.

10. P. Graubner, "Energy-efficient Management of Virtual Machines in Eucalyptus," Proc. Fourth IEEE International Conference on Cloud Computing (CLOUD 2011), pp. 243-250, 2011.

BIOGRAPHY

Prof. Pradnya Kasture is a Professor in the Computer Engineering Department, RMD Sinhgad School of Engineering, Warje, Pune .Her research interests are Computer Networks (wireless Networks), Cloud Computing, web 2.0 etc.

Ms. Revati Vijay Rajane is a Student in the Computer Engineering Department, RMD Sinhgad School of Engineering, Warje, Pune University. She is pursuing Master of Computer engineering degree in. Her research interests are Information Retrieval, Web Data mining, Cloud Computing, etc.