I. J. Computer Network and Information Security, **2024, 2, 55-64**
Published Online on April 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2024.02.05

A Secure VM Placement Strategy to Defend against Co-residence Attack in Cloud Datacentres

Ankita Srivastava
Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow-226010, India
E-mail: ankita31srivastava@gmail.com
ORCID iD: https://orcid.org/0000-0003-3842-600X

Narander Kumar*
Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow-226010, India
E-mail: nk_iet@yahoo.com
ORCID iD: https://orcid.org/0000-0003-4633-1879
*Corresponding Author

Received: 09 September 2022; Revised: 01 November 2022; Accepted: 29 December 2022; Published: 08 April 2024

Abstract: With the increasing number of co-residence attacks, the security of the multi-tenant public IaaS cloud environment has become a growing concern. The co-residence attacker creates a side channel to retrieve the secured data. These attacks help the adversary to leak out the sensitive information of the user with whom it is co-located. This paper discusses a secured VM placement technique, Previous Server and Co-resident users First (PSCF), which focuses on facilitating security against the co-residence attack by minimizing the probability of co-locating the malicious user with the authentic user. Co-location resistance and core utilization metrics are utilized to evaluate the algorithm's performance. The proposed method is simulated, and the result is analysed and compared with existing approaches like Best Fit, Worst Fit, PSSF, and SC-PSSF. It is observed that the proposed approach furnished maximum co-location resistance of 74.32% and a core utilization of 82.63%. Further, the algorithm has shown significant performance in balancing the load and energy consumption. The result has reduced the probability that malicious users co-located with the authentic one, thus reducing the security breach of confidential information.

Index Terms: Co-residency, Co-location Attacks, Cloud Security, Virtual Machine Placement, Energy Consumption.

1. Introduction

Cloud has become the most popular computing paradigm in all the sectors of industries, businesses, and self-usage. Infrastructure as a Service has facilitated users to access multiple resources in shared and dynamic modes. This sharing has given the advantage to the cloud system to reach its maximum efficiency and fully utilize the resources. Additionally, it has allowed cloud users to reduce their expenses by paying only for the resources that are used and leaving the resources that are not in use. These benefits of cloud computing have attracted many businesses that want to do a lot of computational operations economically.

The resource sharing at the infrastructure level is executed through multi-tenancy. The practice of having multiple tenants placed on the same physical hardware is known as multi-tenancy [1]. Cloud users and service providers have got benefitted from such types of practice. It has eliminated the requirement of buying and maintaining hardware resources for cloud users. While the providers can rent out the resources efficiently to the users and generate revenue [2]. Unfortunately, this practice has raised one of the security concerns, a co-residence attack. The threat due to co-residency is being recognized as a co-residence attack, a potent attack in cloud infrastructures that enables the attackers to spy the sensitive and confidential data or affect the efficacity of another tenant co-existing on the same host [3]. Attackers can take advantage of co-residence, enact diverse attacks against the co-tenant, and threaten the security of the cloud and the integrity of the users' confidence. The vulnerabilities in the cloud have caused several types of data breaches [4]. For instance, information leakage can occur due to the misconfigured hypervisor hosting various virtual machines (VMs) from different tenants [5]. The [6] perform a fascinating work by modeling the attacks and defense issue as a game. It utilizes the belief model to segregate the benign and malicious VM. It provides a set of security mechanisms for the defender, which facilitates it to execute a strategy against the attackers. An SC attack to extract confidential information and the technique to secure the public key is demonstrated in [7]. In this paper, server and PM are used irreversibly, representing the machine on which VM is deployed.

These co-resident attacks possess two distinguishing characteristics. Firstly, it is associated directly with resource sharing among users and continues to prevail until the users are isolated on the different physical machines. Secondly, it mainly exploits legitimate resource requests from the users. Hence, it becomes strenuous for conventional security practices like authentication and authorization to identify co-resident attacks and block them without restraining the normal access to the resources in shared mode. A cloud provider is solely in control of the placement of VM to PM through various placement strategies. The provider can implement a placement algorithm that not only optimizes the usage of cloud resources but also furnishes security against co-resident attacks. Recent research [8-10] has provided enough support that a cloud provider can detect such attacks in some situations. The cloud provider can utilize this detecting capability to collect prior information about the types of cloud users (authentic or malicious) based on their past behaviour and use that knowledge to reinforce the security offered by placement algorithms automatically.

Moreover, the increment in the VM allocation factorially makes it tedious to be allocated to the PMs and it has been designated as an NP-hard problem for optimal allocation [11,12]. Nevertheless, this research area is recently explored and has a broad scope of work, as a substantial amount of work is required to rectify and control this issue. Fig.1 demonstrates the co-residence attack. The malicious user (in red) is co-located with the VM of user 1(blue) and user 2(green), and hence user 1 and user 2 are not safe. While users 3 (light pink) and 4 (pink) are safe as they are not co-located with the malicious user.

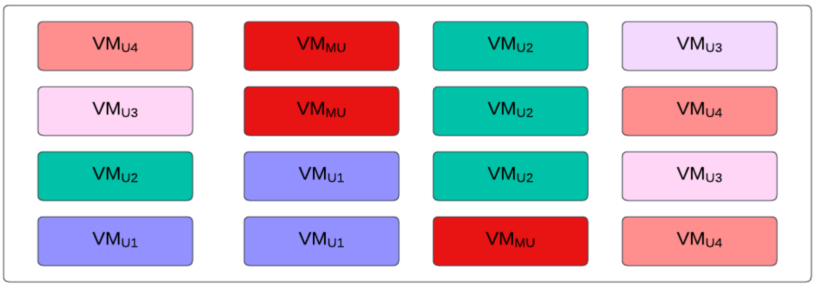

Fig.1. Co-residence attack

Therefore, a secure VM allocation strategy to defend against the co-resident attack has been discussed in the paper. The paper proposes an allocation scheme, Previous Server and Co-resident users First (PSCF) which has a dual objective for optimizing the co-location resistance and resource utilization, which can be enforced by cloud service provider (CSP) to address the security concerns in muti-tenant public clouds. The algorithm tries to place the VMs on the host while maximizing the CR and core utilization. The energy consumption metrics and the system's load balance are also analysed while implementing the algorithm. Further, the algorithm is simulated, and the result obtained is compared and analysed with a few existing techniques.

Further, the paper is organized as follows: Section 2 gives a detailed description of the techniques existing in this field. The work is proposed under Section 3; further, the simulation and the result analysis are performed in Section 4. Lastly, the work is concluded along with the future scope in section 5.

2. Background Work

Recent research has focussed on minimizing power usage through efficient and maximum resource utilization, thus reducing the system's operational cost. However, increasing resource usage and maintaining co-resident security is still a significant concern. The most obvious way to mitigate the co-resident-based attack is to furnish a committed PM to each user. Although this will reduce the co-resident attack to a large extent but will affect resource utilization. Allocating a devoted PM to each user will result in many idle cores in the active PMs, which will tremendously increase the energy consumption of the data centre and hence the cost. Therefore, an allocation strategy is required to stand against co-resident-based attacks and efficiently utilize the resources, thus decreasing the system's execution cost. This section reviews some work on providing the defines against co-resident-based attacks in the past few years. The work can be broadly categorized as a) Defence against information leakage via SC. b) Mitigating the allocations of co-resident VM.

2.1. Defence against Information Leakage via SC

These works primarily aim the devising methods that either remove the SC or reduce the information leakage through SC. A multidimensional analysis for SC attack is given in [13], which studies the security at the hypervisor and resource level. It also analyses the threats and security with various attack levels to virtualization. It discussed the various types of attacks and their detection techniques. It performs the analysis of SC threats on RSA and AES-based procedures. SC attack aims to leak co-residency information over the cache, which is eliminated through time window-based methods. [14] provided a technique to reduce the SC attacks, which works in three phases. The capturing phase, the notification phase, and the anomaly phase. These phases help in minimizing cache-based SC attacks. An embedding scheme to secure the virtual network to shield against information leakage via covert channels in a virtual network [15]. A detection and prevention technique were suggested in [16], which employs the statistical procedure for detecting the SC attack, and a random permutation method was proposed for the prevention. The detection was performed through hypothesis testing while the permutation function mapped the operation's physical address to some other cache location than the one designated. An approach is discussed in [17] to detect some SC attacks based on cache like prime and probe, flush and reload, and flush and flush attack. The proposed work performs its operation in two phases: training and monitoring phase. In the first phase, the author created a contention model of the process under the monitor. It classifies the workload and trains the workload classifiers, produces a classification tree, and uses k-means clustering. The monitoring phase contention model is represented through four characteristic numbers, three indicating the sphere's center and one storing the sphere's radius. If a window falls outside the sphere's perimeter, it is declared to have a spy process, and the system is intimated. An embedding scheme to secure the virtual network to shield against information leakage via covert channels in the virtual network is discussed in [18]

2.2. Mitigating the Allocation of Attackers with an Authentic User

This works to prevent and reduce the probability of co-location of VMs of attackers with the VMs of the authentic user. The main prerequisite of an SC attack is to establish co-location. The VM placement technique, which can minimize the probability of co-location of malicious VMs, would reduce the scope of SC attacks. This can be mitigated by designing a secure VM placement technique or a secure VM migration technique. A preventive measure is discussed in [19], which has used probabilistic methods to reduce the occurrence of these attacks by minimizing the number of PMs shared among the resources. It used the statistical methodology for maximizing the likelihood estimation based on the particular data set. It also increases core utilization and reduces the makespan time. A threat defense framework against co-resident attacks for the VM deployment, SVMDF, is introduced in [20], including secured VM allocation, procedure-based threat score process, and attack-aware VM reallocation. It utilizes the weight mechanism to identify the optimal PM, and then the attack monitor looks out for the performance data, evaluates the threat score, and migrates the hazardous VM. The probability that a VM is a culprit or victim is evaluated. The technique helps in reducing the workload effectively.[21] discussed a technique PSSF that mitigated the threat of attack and addressed the workload balance issue and power consumption. [22] further enhances the PSSF technique and introduces SC-PSSF to perform the allocation providing resistance against the co-location threat. The abovesaid technique performed better than the PSSF in load balancing and power consumption. A sampling-based technique is discussed in [23], which proposed a VM placement technique providing a significant level of resistance against the co-location attack. Additionally, it tries to minimize the number of PMs. A polynomial time technique utilizing the technique of confidence interval estimation is utilized. These techniques have opened various dimensions for research and motivated the researcher to take up this field. Most of the work discussed above provides a prevention technique that mainly concentrates on minimizing the co-residence attack. These works do not consider energy consumption, core utilization, and load balancing, which are some of the major aspects one needs to consider while performing the placement procedure. The work discussed here includes all these factors and furnished a major hands-on the technique in the literature discussed above.

3. Proposed Work

This section will describe the proposed approach to performing the secure VM placement procedure in detail. Firstly, the assumption and threat model is discussed. The mathematical formulation has been done along with the problem statement description. Further, the algorithm is discussed.

3.1. Problem Statement

The CSP uses some placement strategies for VM placement and fulfils the demand of the users. Some of the assumptions which should be undertaken concerning CSP are as follows:

- The CSP may or may not have prior information regarding the authenticity of the user.
- The CSP does not have any future information associated with the VM’s request and takes the placement decision as per their arrival.
- The placement of VMs to PMs is solely under the control of CSP.
- Once a VM is allocated to a PM, it will remain allocated to that particular PM until the request is executed completely or the user terminates it.

The malicious user’s main goal is to co-locate its VM with the authentic user’s VM, conduct an SC (co-residence-based) attack, and compromise it. Some assumptions associated with the malicious user are as follows:

- The malicious users have full control over their workload and can configure their VM requirements like processor, memory, and timing requirements.
- Various malicious users coordinate together to compromise the VMs of the authentic user.
- Once the malicious user compromises the VMs of an authentic user by co-locating its VMs with the authentic

one, it starts leaking its information via SC.

A multi-tenant IaaS cloud system has total U users. These users include both malicious and authentic. The authentic users are those who request VM instances to perform the computation through the cloud resources, while the malicious user has a clear intention to compromise the authentic users through co-location-based attacks. In this environment, the VM's request may vary with the time demanding more resources. Each VM request V is associated with a cloud user which is represented as V^c specifies cores' quantity and V^m specifies memory requirements. The secure VM placement problem deals with the placement of incoming VMs to PMs in such a manner that maximizes the core usage of PM and reduces the co-located VMs. A user is regarded as SAFE if not a single VM of this user is co-located with the malicious user all through the placement procedure. A good placement technique has a high number of SAFE users.

3.2. Mathematical Formulation

It is assumed that there are p number of PMs represented as $P = \{P_1, P_2, P_3, P_4 \dots \dots . P_p\}$ having a core and memory represented as P_i^c and P_i^m and set of n VMs represented as $V = \{V_1, V_2, V_3, V_4 \dots \dots . V_n\}$ having core and memory represented as V_i^c and V_i^m. Let x_{ij} and y_j be the binary variables which are being defined as:

$$x_{ij} = \begin{cases} 1 & if\ VM\ i\ \forall\ i\ \in V\ is\ allocated\ to\ PM\ j\ \forall j \in P \\ 0 & otherwise \end{cases} \tag{1}$$

$$y_j = \begin{cases} 1 & if\ the\ PM\ j\ \forall j\ \in P\ is\ in\ use \\ 0 & otherwise \end{cases} \tag{2}$$

The core utilization of the system can be represented as a ratio of cores utilized by all the VMs till the time they are active to the total available live PMs core that is not idle given as

$$CU = \frac{\sum_{j=1}^{n} V_i^c \times t_j^v}{\sum_{i=1}^{p} P_i^c \times t_i^p} \tag{3}$$

Where, t_j^v represent the lifetime of a VM v and t_i^p represent the life of a PM, that is, the duration it is hosting at least a VM.

A Co-location Resistance (CR) is defined as the ratio of SAFE users to the total authentic users, which is represented as

$$CR = \frac{Total\ number\ of\ SAFE\ users}{Total\ number\ of\ authentic\ users} \tag{4}$$

The SAFE users may vary depending upon the sequence of the incoming request. Suppose there U_m malicious users and U_l authentic users. There is a sequence σ which denotes the sequence of occurrence of a user. E.g., if $\sigma = U_a U_b$ this shows that the U_a has requested the first VM before the request was made by U_b. There are two cases in which a user U_ϑ gets co-located with another user U_μ:

Case 1: If $> \mu$, it means the user U_μ has placed its first request before U_ϑ then U_ϑ can become co-resident with U_μ when it gives its first request. If the user U_ϑ does not get co-located with U_μ in the first request, then PSCF will ensure that it will never get co-located with U_μ with any further requests.

Case 2: If $\vartheta < \mu$, means the user U_μ has placed its first request after U_ϑ then U_ϑ can become co-resident with U_μ when U_μ gives it the first request. If the user U_μ does not get co-located with U_ϑ in the first request, then PSCF will ensure that it will never get co-located with U_ϑ with any further requests.

The probability that a user is authentic and does not get co-residence with the malicious user is given as:

$$P = P(U_\vartheta\ is\ authentic\ user) \times P\big(U_\vartheta\ is\ not\ coresident\ with\ any\ malicious\ user\ U_\mu \big|\ \vartheta > \mu\ and\ U_\vartheta\ is\ authentic) \times P(No\ malicious\ user\ is\ co - resident\ with\ U_\vartheta\ |\ \vartheta < \mu\ and\ U_\vartheta\ is\ authentic) \tag{5}$$

The probability that the user is authentic is evaluated as:

$$P(U_\vartheta\ is\ authentic\ user) = \frac{U_l}{U} \tag{6}$$

The probability to be evaluated in case 1 needs some consideration. Whenever a U_μ demands for a new VM either it is allocated randomly or the PM having enough resources is allocated. It is supposed that a novel user is co-located with some random α users on average. The probability that the user is not co-resident with the malicious user is equivalent to the probability of α users co-locating with the $(\vartheta - 1)$ users coming before in the sequence are authentic.

The probability is evaluated as:

$$P\left(U_{\vartheta}\ is\ not\ co-resident\ with\ any\ malicious\ user\ U_{\mu}\middle|\ \vartheta > \mu\ and\ U_{\vartheta}\ is\ authentic\right) = \frac{(U_l-1)^{min(\vartheta-1,\alpha)}}{U-1} \qquad (7)$$

The probability for case 2 needs first the evaluation of the probability for the user U_{μ} who are malicious and can be co-resident with U_{ϑ}. This can be estimated as:

$$P\left(\ U_{\mu} is\ malicious\ and\ can\ get\ co-located\ with\ U_{\vartheta}\ \right) = \frac{U_m}{U-1} \times \frac{\min\ (\mu-1,\alpha)}{\mu-1} \qquad (8)$$

The probability that no malicious user is co-resident with authentic is calculated as:

$$P(No\ malicious\ user\ is\ co-resident\ with\ U_{\vartheta}\ |\vartheta < \mu\ and\ U_{\vartheta}\ is\ authentic) = \prod_{\mu=\theta+1}^{U}(1 - \frac{U_m}{U-1} \times \frac{\min\ (\mu-1,\alpha)}{\mu-1}) \ (9)$$

For an efficient placement, CR and CU needs to be maximized subject to the given constraint:

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad \forall\ i\ \in V \qquad (10)$$

$$\sum_{i=1}^{n} V_i^c\ \times x_{ij}\ \leq\ P_j^c\ \times\ y_j\ \forall j \in P \qquad (11)$$

$$\sum_{i=1}^{n} V_i^c\ \times x_{ij}\ \leq\ P_j^c\ \times y_j \forall j \in P \qquad (12)$$

Where the summation of all the cores allocated to the VMs should be less than the cores

Equation (10) restricts VM i to be deployed to one server only. The restriction in (11) & (12) ensures that the allocated VM does not drive beyond the extent of the PM.

3.3. Proposed Algorithm

The proposed algorithm, “Previous Server and Co-tenant First” (PSCF), is a placement algorithm to maximize CU and CR. As a new request for a VM arrives, the algorithm is called. The algorithm is displayed as Algorithm 1.

Algorithm 1: PSCF

```
Input: $U_i$, $V_i$ , $P_{active}$ , $P_{idle}$
Output: $V_i$ is allocated to a PM $P$
Start
  if ($U_i$ is an old user)
    for each $P_k$ where $P_k \in P_{active}$
      if ($P_k$ has enough resources)
        if ($P_k$ already host or hosted once $U_i$’s VM && $Users(P_k)\backslash U_i \subseteq Colocated(U_i)$)
        then $P_{list} \leftarrow P_k$
        endif
      endif
    end for
    if (!isempty($P_{list}$))
      Sort ($P_{list}$ , available_cores, ascending)
      $P_{assign} \leftarrow P_{list}[0]$
    else
      Sort ($P_{idle}$ , available_cores, ascending)
      $P_{assign} \leftarrow P_{idle}$ with enough resources
      remove $P_{idle}$ from $P_{idle}$ and insert in $P_{active}$
    endif
else
    $P_{list} \leftarrow$ all $P_k$ with enough resources to host $V_i$ such that $P_k \in P_{active}$
    $P_{assign} \leftarrow P_k$ any random such that $P_k \in P_{list}$
    if (isempty($P_{assign}$))
      Sort ($P_{idle}$ , available_cores, ascending)
      $P_{assign} \leftarrow P_{idle}$ with enough resources to host $V_i$
      remove $P_{idle}$ from $P_{idle}$ and insert in $P_{active}$
    endif
endif
  assign $V_i$ to $P_{assign}$
End
```

The algorithm is executed in some steps, which are defined as follows:

Step 1: It starts with the identification of the user, whether it is a new or old user.

Step 2 handles the old user, i.e., if the user has already placed a VM request, then a list of PMs, P_{list} is created, which includes all those PMs subjected to the following two constraints:

- P_k has enough resources to host V_i and
- those users' VMs which are currently placed on PM P_k are the subset of those users with whom the U_i was co-located once.

Step 3: It checks if this P_{list} is empty or not, if the list is not empty, then the list is sorted according to the available cores in increasing order, and the first in the list is assigned to VM. Otherwise, the P_{idle} is sorted, and the one with enough resources and the least available cores is assigned to V_i.

Step 4: It is executed if the user U_i, is a new user and has not placed any request in the past, PM from the P_{active} the list is randomly chosen which has enough resources to host V_i. This random assignment makes the situation hard for the malicious user to be co-located with the destined or the target user.

Step 5: It will work when no PM is available for allocation then, the PM is assigned from the P_{idle} list which has enough resources to host V_i and the PM is removed from the P_{idle} list and inserted in the P_{active} list. This assignment helps reduce the co-location of VMs of malicious users with the authentic user; further, it decreases the probability of co-location of a particular user with an adversary. Lastly, the VM is assigned to the PM.

4. Simulation

The effectiveness and efficiency of the PSCF are analysed through a series of simulation experiments. The suggested algorithm is simulated on Cloudsim, open-source software for cloud computation [24,25]. It facilitates the simulation of large data centres, placement policies, and many other functionalities. As shown in table 1, a datacentre with 150 servers and more than 1500 VMs is considered for the simulation. There are three variations in the VMs with different CPU and RAM requirements. Table 1 represents the configuration of VMs undertaken. Type 1 VM consists of 613 MB RAM and 500 MIPS, type 2 VM has RAM of 870 MB and single core CPU of 1000 MIPS, while type 3 VM is of 1740 MB RAM and 2500MIPS processor speed. The PM considered for the simulation consists of 49152 MB RAM with 12 cores CPU. As the VM request arrives, it is allocated randomly to the PMs as per the allocation policy discussed above. In the simulation, a large authentic user can initiate with 20 VMs. The malicious tenant starts respectively 5,10, 15, …,100 VMs. The experiment is repeated 100 times to obtain the result. The result has been compared with PSSF[20], SC-PSSF[21], Best Fit(BF), and Worst Fit (WF). The simulation environment used is same as given in the techniques from which the proposed work has been compared for the better analysis of the suggested work. Moreover, the parameters considered highly resembles the configuration of the real systems.

Table 1. VMs and PM configuration

	PM1	VM1	VM2	VM3
CPU cores	12	1	1	1
MIPS	2600	500	1000	2500
RAM(MB)	49152	613	870	1740

4.1. Co-location Resistance

Co-location Resistance evaluation may vary with the malicious user's occurrence in the allocation of the VM. Fig. 2 shows the graph between the co-location resistance and the percentage of the malicious user in the system. It can be analyzed that as the malicious user increases the co-location resistance decreases. When a malicious user's percentage increases to more than 30%, the CR of BW and WF almost becomes negligible, which shows that these algorithm does not provide any resistance against this attack. In the case of PSSF, SC-PSSF, and PSCF as the percentage of malicious users increase, the co-location resistance decreases rapidly, showing the likelihood of co-locating malicious users with authentic users increases. The proposed algorithm has shown a better performance as compared to others. The maximum resistance shown by the PSCF is 74.32% when the percentage of the malicious user is 10%, and the least resistance is 4.16% with 90% malicious users, which is best among all the other algorithms that are being compared.

4.2. Core Utilization

Core utilization is an important aspect of all placement algorithms. The more utilization, the more efficient the placement techniques. An effective placement algorithm with maximum core utilization reduces the system's energy consumption and the data center's expenditure. Fig. 3 depicts the core utilization of all the algorithms considered for the analysis. It can be observed from the figure that the WF algorithm performs worst among all the algorithms. BF offers better efficiency as compared to PSSF and SC-PSSF. BF is the second efficient algorithm in terms of CU. The PSCF shows the maximum CU efficiency of 82.63%, which is best compared to BF, PSSF, SC-PSSF, and WF. Among all the algorithms, PSCF has presented a good balance between the CU and CR.

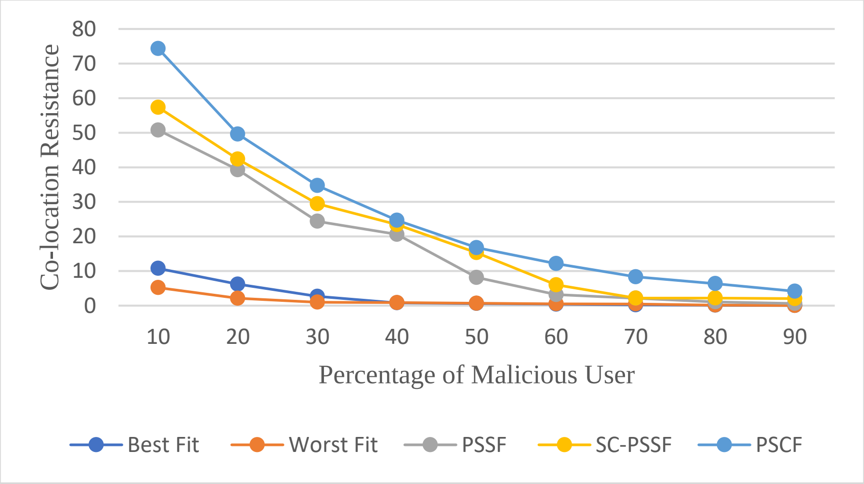

Fig.2. Co-location resistance

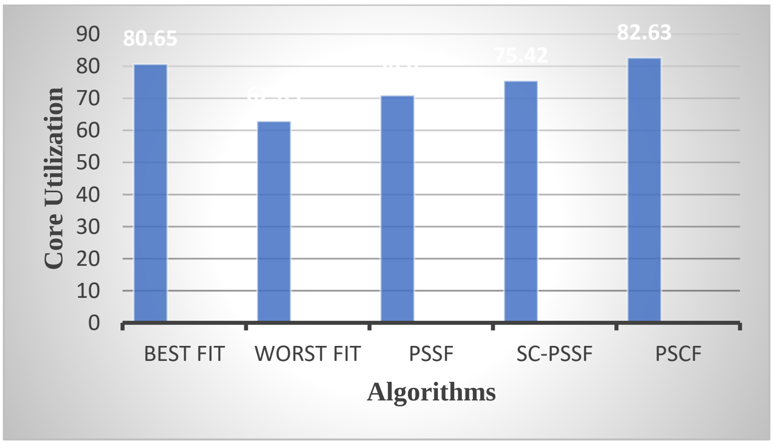

Fig.3. Core utilization

4.3. *Load Balancing*

The load on the machines greatly influences the performance of the cloud system. The prime responsibility of CSP is to distribute the load properly among the machines. Load balancing is the distribution of VMs among PMs. Let A_u is the set of users such that $A_u = \{A_1, A_2, A_3, \dots \dots A_k\}$ and $|VM(A_u)|$ is the set of VMs requests placed by users A_u. Let $T_{servers}$ is the total number of servers. The standard deviation of VMs in hosts is evaluated as:

$$L = \frac{\sum_{i=1}^{k} |VM(A_i)|}{T_{servers}} \tag{13}$$

From fig. 4 it is observed that when the VMs are less, the deviation is less, but as the number of VMs increases, the deviation is more. The WF has shown the maximum deviation, while BF has shown a deviation less than WF, PSSF, and SC- PSSF. SC-PSSF has shown less deviation than PSSF. The least deviation is shown by the PSCF technique, which is worth expecting.

4.4. *Energy Consumption*

Energy has a vital role in cloud computing. The CPU consumes 90% of the energy in a cloud center [26], and if a server is idle, it will consume 70% of the maximum energy. The energy can be evaluated as:

$$E = 70\% \times E_{max} + (1 - 70\%) \times E_{max} \times R \tag{14}$$

Where, E_{max} is the maximum energy consumption of a server, and R represents the CPU ratio of the host.

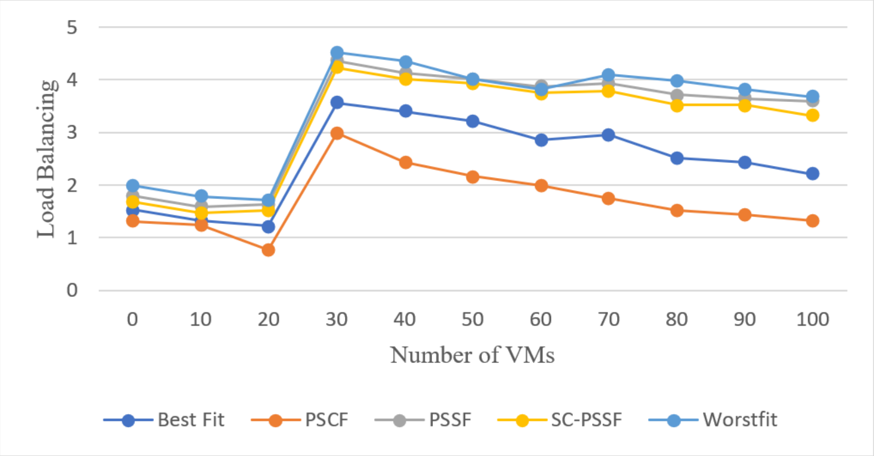

Fig.4. Load balancing

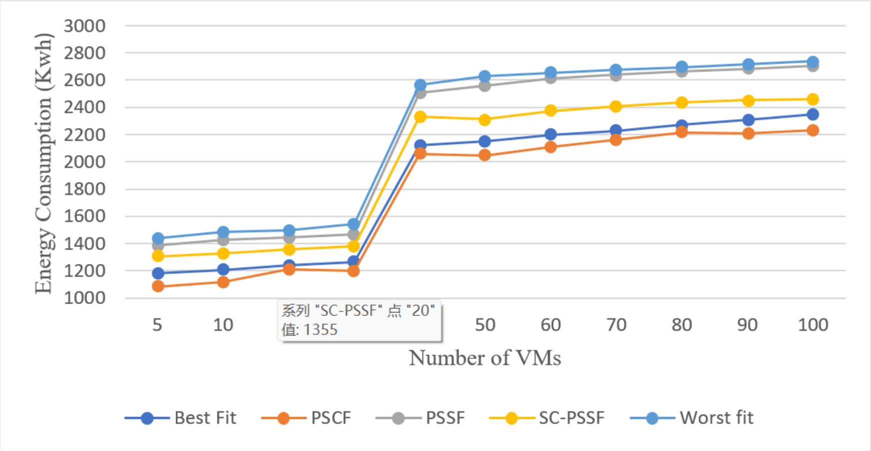

Fig.5. Energy consumption

A detailed analysis has been performed on the energy consumption by the data centre upon the implementation of the proposed algorithm. Fig. 5 shows the variation in energy consumption concerning the VMs of different algorithms. It is observed that below 30 VMs there is not much variation in energy consumption. The sudden rise in the energy can be seen when the number of VMs goes beyond 30. Energy consumed in the case of PSSF and the WF is quite comparable. In some cases, the energy utilized by PSCF and BF is at power when the number of VMs is 20, 40, and 70 and in the rest of the scenario, PSCF takes the lead position against BF. Still, the performance of PSCF is better in terms of energy consumption than other algorithms.

5. Conclusions

This paper described a secured VM placement technique, PSCF, which addresses the security threats associated with co-tenancy in public clouds. Instead of providing the solution after the attack, the proposed work minimizes the probability of attackers getting co-located with authentic users. It utilizes the CU and CR metrics to display the technique's efficiency against security concerns associated with the placement of the VMs. It considers the servers previously used for the placement of the VMs of the old user, and it also considers the co-located users with the given user while placing its VMs on the PMs. The proposed algorithm is simulated on the Cloudsim, and the result is compared against some well-known algorithms, BF, WF, PSSF, and SC-PSSF. It is observed from the analysis of the result that PSCF has performed better, showing a maximum CR of 74.32% and CU of 82.63%. The energy consumption of PSCF has reduced by 18%, 17%, and 10% as compared to Worst Fit, PSSF, and SC-PSSF, respectively. In addition, load balancing metrics are also studied to evaluate the technique's performance. From the analysis, it is concluded that PSCF offers good resistance against co-location attacks. The PSCF has performed better than WF and BF as they are not providing any resistance against the co-location resistance. PSSF offers security against the co-location resistance considering only the previously selected server, while SC-PSSF works for the reduction in energy consumption in PSSF. PSCF is better than the latter two as it considers both the previously used server and co-tenant for allocating VMs. It enhances security and reduces the number of active servers and thus reducing energy consumption. In the future, the work can be extended to consider the live migration of the VM, as it plays a crucial role in the VM placement. The work can be tested in a real cloud scenario with multiple data centers to evaluate its efficiency. Further, more factors affecting the security of the cloud can be undertaken, such as traffic, scheduling, scaling, provisioning, and many more.

References

[1] Chuka-Maduji N. and Anu V., “Cloud Computing Security Challenges and Related Defensive Measures: A Survey and Taxonomy,” SN Computer Science, Vol. 2, No. 4, pp 1-17, 2021. doi:10.1007/s42979-021-00732-3.

[2] Srivastava A. and Kumar N, “Resource management techniques in cloud computing: A State of Art,” ICIC Express Letters, Vol. 14, No. 9, pp. 909–916, 2020. doi:10.24507/icicel.14.09.909.

[3] Kumar P and Kumar Bhatt A, “Enhancing multi-tenancy security in the cloud computing using hybrid ECC-based data encryption approach,” IET Communications, Vol. 14, No. 18, pp. 3212-3222, 2020. doi:10.1049/iet-com.2020.0255.

[4] Umar Sayibu, Frimpong Twum, Issah Baako, "Delivering a Secured Cloud Computing Architecture and Traditional IT Outsourcing Environment via Penetration Tools in Ghana", International Journal of Computer Network and Information Security Vol.11, No.11, pp.46-59, 2019.

[5] Compastié M, Badonnel R, Festor O, He R, “From virtualization security issues to cloud protection opportunities: An in-depth analysis of system virtualization models,” Computers & Security, Vol. 97, pp. 101905, 2020. doi:10.1016/j.c,ose.2020.101905.

[6] Hasan M M and Rahman M A, “A signaling game approach to mitigate co-resident attacks in an IaaS cloud environment,” Journal of Information Security and Applications, Vol. 50, pp. 102397, 2020. doi:10.1016/j.jisa.2019.102397

[7] Thirumalai C, Mohan S, Srivastava G., “An efficient public key secure scheme for cloud and IoT security,” Computer Communications, Vol. 150, pp. 634-643, 2020. doi:10.1016/j.comcom.2019.12.015.

[8] Narayana K E and Jayashree K, “Survey on cross virtual machine side channel attack detection and properties of cloud computing as sustainable material,” Materials Today: Proceedings, Vol. 45, pp. 6465-6470, 2021. doi:10.1016/j.matpr.2020.11.283

[9] Saxena D, Gupta I, Kumar J, Singh A K, Wen X, “A secure and multiobjective virtual machine placement framework for cloud data center,” IEEE Systems Journal, Vol. 16, No. 2, pp. 3163-3174, 2021. doi:10.1109/JSYST.2021.3092521.

[10] Hansraj, Tiwari P K, Chaudhary A, “Secure VM placement analysis against co-location based attack in cloud,” Journal of Discrete Mathematical Sciences and Cryptography, Vol. 24, No. 5, pp. 1457-1465, 2021. doi:10.1080/09720529.2021.1945215.

[11] Azizi S., Zandsalimi M. H, Li, D., “An energy-efficient algorithm for virtual machine placement optimization in cloud data centers,” Cluster Computing, Vol. 23, No. 4, pp. 3421-3434, 2020. doi:10.1007/s10586-020-03096-0

[12] Tao X, Wang L, Xu Z, Xie R.,”Secure and Efficient Allocation of Virtual Machines in Cloud Data Center,” In 2021 IEEE Symposium on Computers and Communications (ISCC), 2021. doi:10.1109/ISCC53001.2021.9631399

[13] Narayana K E and Jayashree K., “Survey on cross virtual machine side channel attack detection and properties of cloud computing as sustainable material,” Materials Today: Proceedings, Vol. 45, pp. 6465-6470, 2021. doi: 10.1016/j.matpr.2020.11.283

[14] Tao X, Wang L, Xu Z, Xie R., “SCAMS: A Novel Side-Channel Attack Mitigation System in IaaS Cloud,” In MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM), 2021. doi:10.1109/MILCOM52596.2021.9652991

[15] Xiao Y, Liu L, Ma Z, Wang Z, Meng W., “Defending co-resident attack using reputation-based virtual machine deployment policy in cloud computing,” Transactions on Emerging Telecommunications Technologies. Vol. 32, No. 9, pp. e4271, 2021. doi:10.1002/ett.4271

[16] Saxena S, Sanyal G, Srivastava S, Amin R., “Preventing from Cross-VM Side-Channel Attack Using New Replacement Method,” Wireless Personal Communications, Vol. 97, No. 3, pp. 4827–4854, 2017. doi:10.1007/s11277-017-4753-7

[17] Kulah Y, Dincer B, Yilmaz C, Savas E., “SpyDetector: An approach for detecting side-channel attacks at runtime,” International Journal of Information Security, Vol. 18, No. 4, pp. 393-422, 2019. doi:10.1007/s10207-018-0411-7

[18] Z. Wang, J. Wu, Z. Guo, G. Cheng, H. Hu, “Secure virtual network embedding to mitigate the risk of covert channel attacks,” In Computer Communications Workshops (INFOCOM WKSHPS) IEEE, 2016, pp. 144–145. Doi: 10.1109/INFCOMW.2016.7562061

[19] Chhabra S and Singh A K., “A secure VM allocation scheme to preserve against co-resident threat,” International Journal of Web Engineering and Technology, Vol. 15, No. 1, pp. 96-115,2020.

[20] Wang, X., Wang, L., Miao, F., & Yang, J., “Svmdf: A secure virtual machine deployment framework to mitigate co-resident threat in cloud,” In 2019 IEEE Symposium on Computers and Communications (ISCC), pp.1-7, 2019. 10.1109/ISCC47284.2019.8969721.

[21] Y. Han, J. Chan, T. Alpcan, and C. Leckie., “Using virtual machine allocation policies to defend against Co-resident attacks in cloud computing,” IEEE Transactions on Dependable & Secure Computing, Vol. 14, No. 1, pp. 95–108, 2017, 10.1109/TDSC.2015.2429132

[22] Jia, H., Liu, X., Di, X., Qi, H., Cong, L., Li, J., & Yang, H, “Security strategy for virtual machine allocation in cloud computing,” Procedia computer science, Vol. 147, pp. 140-144, 2019. doi: 10.1016/j.procs.2019.01.204

[23] Thabet, M., Hnich, B., & Berrima, M., “A sampling-based online Co-Location-Resistant Virtual Machine placement strategy,” Journal of Systems and Software, Vol. 187, pp. 111215, 2022. doi:10.1016/j.jss.2022.111215

[24] "CloudSim," http://www.cloudbus.org/cloudsim/.

[25] Sankaran, L., & Subramanian, S. J., “CloudSim Exploration: A Knowledge Framework for Cloud Computing Researchers,” In Applied Soft Computing and Communication Networks, Springer, Singapore, pp. 107-122, 2021. doi:10.1007/978-981-33-6173-7_8

[26] Mishra, S. K., Puthal, D., Sahoo, B., Jayaraman, P. P., Jun, S., Zomaya, A. Y., & Ranjan, R. “Energy-efficient VM-placement in cloud data center,” Sustainable computing: informatics and systems, Vol. 20, pp. 48-55, 2018. doi: 10.1016/j.suscom.2018.01.002

Authors' Profiles

Ankita Srivastava, a Ph. D. scholar in the Department of Computer Science at Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow, India, pursued her B. Tech (IT) in 2011 and M. Tech (CSE) in 2016 from Dr. A P J Abdul Kalam Technical University, Lucknow, India. Her research interests are meta-heuristics, optimization, cloud computing, and security.

Narander Kumar (Dr. Narander Kumar) received his Post Graduate degree and Ph. D. in CS & IT from the Department of Computer Science and Information Technology, Faculty of Engineering and Technology, M.J.P. Rohilkhand University, Bareilly, Uttar Pradesh, India, in 2002 and 2009 respectively. His research interest includes Quality of Service (QoS), Computer Networks, resource management mechanism, networks for multimedia applications, and performance evaluation. His current research area is Cloud Computing Environment.

How to cite this paper: Ankita Srivastava, Narander Kumar, "A Secure VM Placement Strategy to Defend against Co-residence Attack in Cloud Datacentres", International Journal of Computer Network and Information Security(IJCNIS), Vol.16, No.2, pp.55-64, 2024. DOI:10.5815/ijcnis.2024.02.05