An Efficient Co-Resident Dos Attack Defense Mechanism for Cloud Computing using Two-Player Security Game Approach

Rethishkumar S., R. Vijayakumar

Abstract: ***For cloud computing systems, Virtual Machines (VM) were conceived as the basic component. However, VMs give effective computing resources; they were prone to lots of security threats. Whereas few threats can be easily rectified, but few attacks like co-resident attacks were tedious process to identify. So, to reduce the co-resistance DOS attacks by creating it as tedious for attackers to initiate attacks, two-player game approach based defense mechanism is suggested in our work. The attacker behavior variations among the attacker and normal users under PSSF VM allocation policy, is examined initially in the proposed mechanism. EDBSCAN (Enhanced Density-based Spatial Clustering of Applications with Noise), is utilized to do the clustering analysis process. Based on the clustering algorithm, the Partial labeling is performed, to partially comprehend the users as legal or malicious. In order to classify the nodes, the semi-supervised learning using Deterministic Annealing Semi-supervised SVM (DAS3VM) optimized by branch and bounds method is performed. The two-player security game approach helps to raise the cost of introduction new VMs therefore reducing the probability of initiating co-resident DOS attack, once after the user accounts were classified. Therefore, the security threats can be avoided effectively with the help of the proposed defense mechanism. Experimental result confirms that the suggested co-resident DOS attack defense mechanism makes a desirable involvement to avoid the security threats.***

Keywords: Co-Resident DOS Attack, PSSF, EDBSCAN, DAS3VM, Branch and Bound Method.

I. INTRODUCTION

Cloud computing has prepared the way for "the long-held dream of computing as a utility" [1]. To prevent the business over provisioning their own resources and to give precise amount of computing, Commercial third-party is utilized. As a key, Virtualization model work here. The cloud providers can be able to profitably leverage economies of scale and statistical multiplexing of computing resources, by placing various virtual hosts on a single physical machine. Various cloud computing models exist, among that, the Infrastructure-as-a-Service (IaaS) model were utilized by providers like Amazon's Elastic Compute Cloud (EC2) service, to provide a set of virtualized hardware configurations for customers [2]. The sharing of a common physical platform between the various virtual hosts, nevertheless, it brings a new dispute to security, as a customer's virtual machine (VM) may be co-located with unknown and un-trusted parties. Placement on a common platform requires the sharing of physical resources and drops.

Revised Version Manuscript Received on December 29, 2017.

Rethishkumar S, Research Scholar, Department of Computer Sciences, Mahatma Gandhi University, Kottayam (Kerala), India.

Dr. R. Vijayakumar, Professor, Department of Computer Sciences, Mahatma Gandhi University, Kottayam (Kerala), India.

The sensitive data processed in a cloud potentially vulnerable to proceed the malicious co-residents sharing the physical machine. Researchers previously explained the methods of bypassing co-resident isolation in virtualization middleware, specifically through the through cache-based side channels [3, 4, 5]. Their output explains that the hypervisors provides a novel attack surface through which privacy and isolation assured to be agreed. Nevertheless, in the academic literature [6, 7], defenses against such vulnerabilities are already being proposed. In cloud computing environments, Virtual machines (VM) area commonly used resource. VMs assist to raise the usage rate of the underlying hardware platforms, for the cloud providers. The -demand resource scaling, and outsources enables the maintenance of computing resources, for the cloud customers. Nevertheless, apart from entire merits, it provides a new security threat [8]. In theory, VMs running on the same physical server (i.e., co-resident VMs) are logically separated from another [9]. Practically, malicious users can construct different side channels to circumvent the logical isolation, and to acquire the sensitive information from co-resident VMs, ranging from the coarse-grained, e.g., workloads and web traffic rates, to the fine-grained, e.g., cryptographic keys. Harmless details like workload statistics can be helpful for clever attackers. For instances, those data can be utilized to recognize, when the system is in danger, the time to begin further attacks, like Denial-of-Service attacks. A two-player security game approach is proposed in our work, to protect the VMs against co-resident DOS attacks. Game based security approaches have broadly enforced as an efficient scheme for defense mechanisms. In [10] a game theoretical approach was established to protect against co-resident attacks. Nevertheless, this approach doesn't conceive the force of the datacenter size in attack defense. Similarly, the semi-supervised learning based classification approach doesn't give optimal solutions. So, to rectify these restrictions, the suggested two-player security game approach based defense mechanism conceives the data center size. The Enhanced DBSCAN (EDBSCAN) follows the DAS3VM with branch and bound methods to identify the globally optimal solutions, as the DBSCAN clustering approach acquires time complexity. At last, the two-player game model is utilized to raise the cost and complexity of the launching attacks, therefore forcing the attackers to behave as normal users. The rest of this work is organized as follows: Section 2 explains different past researches based on this research work. Section 3 describes.

The proposed defense mechanism. Section 4 indicates the experimental results of the proposed model while the section 5 gives the conclusion about this research work.

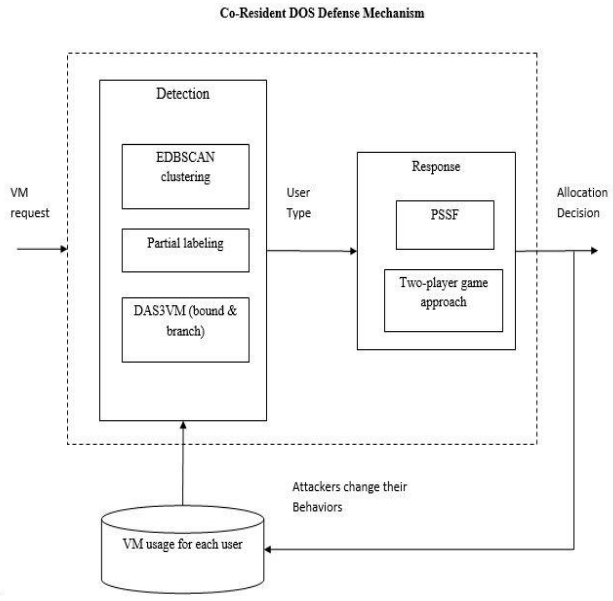
II. RELATED WORKS

Yinqian Zhang et al [11] explain the Home Alone, which is system that allows a tenant to check its VMs' exclusive use of a physical machine. This system's key idea is to reverse the usual application of side channels. Instead of enforcing a side channel as a vector of attack, this system utilizes a side-channel (in the L2 memory cache) as novel, defensive detection tool. By examining cache utilization at the time where "friendly" VMs synchronize to eliminate the portions of the cache, a tenant with the help of Home Alone can identify the activity of a co-resident "foe" VM. The idea of Home Alone involves classification techniques to examine the cache usage and guest operating system kernel modifications which reduce the performance impact of friendly VMs sidestepping monitored cache portions. The execution of Home Alone on Xen-PVM doesn't require any changes to the current hypervisors and any special action or cooperation by a cloud provider. Co-resident watermarking, a traffic analysis attack which permits a malicious co-resident VM to insert a watermark signature into the network flow of a target instance is explained by Adam Bates et al [12]. In order to ex-filtrate and transmit co-residency data from the physical machine, compromising isolation without reliance on internal side channels, this watermark technique is utilized. But this approach is very complex to protect without costly underutilization of the physical machine. Moreover, it computes the co-resident watermarking under a large variety of conditions, system loads and hardware configurations, from a local lab environment to production cloud environments (Future grid and the University of Oregon's ACISS). The capability to begin a covert channel of 4 bits per second is explained, and it proved that the co residency with a target VM instance in less than 10 seconds. It also provides that passive load measurement of the target and following behavior profiling is feasible with this attack. The final investigation explains the requirement for the careful design of hardware, which is utilized in cloud. Yi Han et al [13] focus mainly on the co-resident attack, where malicious users targets to co-locate their virtual machines (VMs) with target VMs on the same physical server, and then accomplish a side channels to gather the private information from the victim. Our earlier work explains how to avoid or mitigate the threat of side channels. Nevertheless, the given solutions were unrealistic for the existing commercial cloud platforms. This method explains the issue from various views, and examines how to reduce the attacker's possibility of co-locating their VMs with the targets, by managing the fulfilled workload balance and low power consumption for the system. Particularly, it brings-in a security game model to distinguish it from other VM allocation policies.

Likewise, we have enough techniques for the security of cloud computing. Nevertheless, various approaches concentrate on avoiding the side channels for preventing the attacks. The clever attackers rectify these hurdles and initiate the attacks. Therefore more advanced defense systems were required to be involved in this research.

III. PROPOSED METHODOLOGY

The examination of the attacker behavior is performed and distinguished with that of the legitimate user, in the initial stage of the suggested methodology. The examining process is done in three different scenarios, such as under no security mechanism, with presence of PSSF VM allocation policy and the proposed defense mechanism. The nature of the normal users can be identified in under no defense mechanism. No matter how they begin VMs, it is obviously preferable that they can extend their VMs across a wider range of servers, to raise the probability of co-locating with their targets, from the attacker's view. The new VMs will be allotted to lightly loaded servers, under PSSF policy. Furthermore, entire servers were controlled in groups, to reduce the usage of power, and a new VM won't be allotted to one server in a new group, till entire servers in the existing active groups were completely used. PSSF doesn't distinguish among the users, and treats entire VM requests uniformly. AS a result, if the attacker maintains generating a new accounts, everything begins one and only one VM, PSSF becomes less effective even though in this worst case. The only option for an attacker is to generate the new accounts. So, the suggested defense mechanism concentrates mainly on generating life difficult for the attacker by maximizing the cost for generating the new accounts. This, in turn, gives the possibility of initializing new VMs minimum and so, it forces the attacker to behave as normal user. Figure 1 shows the overall architecture of the proposed defense mechanism.



The defense mechanism comprises of two parts: the detection module and the response module. When a user requests to begin a new VM, the former module loads the historical data from the database, segregates the user into one of the three types: low/medium/high risk, and sends the result to the response module. The latter module restricts the available servers,

Based on the output, so, VMs of any user only co-locate with VMs which is initiated by users of the similar type. Furthermore, the response module fix the parameter in the VM allocation policy PSSF (also based on the classification result), which chooses a server within the restricted set. It is probable that the attackers will adjust with the way they initiate the VMs, based on the allocation decisions. In the meantime, the defense system gets updated with the database of VM usage regularly, that modifies the allocation decision. Hence, to examine the optimization step for this adversarial learning issue, we model the issue as a two-player security game (G) between the attacker (A) and the defender (D). Before examining the security model, let us know about co-resident and DOS attacks.

A. Co-Resident Attacks

The co-resident threat conceives the malicious and motivated adversary, which isn't associated with the cloud provider, in the cloud computing environment. Victims were lawful cloud customers, which introduces the Internet-facing samples of the virtual servers to proceed with their business. The adversary, who is possibly a business competitor, desires to make use of the novel capabilities, which is approved to him by the cloud co-residency to identify the precious details regarding his target's business. They involve reading private data or compromising a victim machine. They also involve more subtle attacks like performing load measurements on the victim's server or introducing a denial-of-service attack. The adversary is free to launch and control an arbitrary number of cloud instances in the Masquerading, which is another legitimate cloud customer. The cloud infrastructure is a trusted component, as it is required for the normal utilization of any third-party cloud. Ristenpart et al. [3] states that, Co-residency detection through virtualization side channels is a tedious process. This work spends strategies for enforcing the instance placement routines of the Amazon EC2 cloud infrastructure, to probabilistically accomplish the co-location with a target instance. From there, co-residency can be identified with the help of a cross-VM covert channel as a ground truth. While more advanced methods of victorious placement are outlined, namely abusing temporal locality of instance introducing, it is shown that a brute force approach is also modestly successful. An attacker has the ability to introduce various instances, to do the co-residency check, terminate, and repeat till the appropriate placement is acquired, which is a Masquerading. Several cross-VM information leakage attacks were defined, like the load profiling and keystroke timing attacks. Nevertheless, independent results assure that various approaches in earlier work, like the utilization of naive network probes, are no longer relevant on the EC2.

B. DOS Attacks

DOS attack aims to provide the services which are unavailable to its users. The attack makes use of huge system resources like processing power, memory, and bandwidth. This consumption will provide the service inaccessible to the users or impossibly slow. DOS attacks, and their variant Distributed Denial of Service (DDOS), capture huge media attention mostly due to its magnitude. In 1988, only six DDOS attacks occur, as the reports show. DDOS attacks targets huge websites such as CNN, Yahoo, and Amazon in the year 2000 with an attack rate of about 1GBps. DDOS attacks attains the rates of 70GBps in the year 2007. In 2013, a huge attack occurs on Spamhaus spam detection service that attains the greatest rate during 300 GBps [14]. Lately, in February 2014, the highest DDOS attack known until now with the rate of 400GBps took place. This attack aims a public cloud service provider termed as Cloud Flare. Attacks of that magnitude affect not only their targets, but influences the entire Internet in the area. As described in [15], regular Internet users experienced noticeable slowness in their Internet services. DOS attacks can be managed on multiple layers in the network. An attacker can work as a DOS attack at the network level to provides the entire server unreachable. This is performed by acquiring the Network Interface Card (NIC) of the server entirely occupied with hopeless packets in such a way that no more bandwidth is accessible for legitimate users. A DOS attack can be introduced in the transport layer with the help of the very old, but still efficient, SYN Flood technique. In a SYN flood attack the attacker transmits a flood of TCP SYN requests that acquires the server busy without actually finishing the three-way handshake procedure which is utilized in the setup of TCP sessions. DOS attacks can also be introduced at the application level by transmitting the fake requests to the application layer protocol to use the servers' memory and processing power. Transmitting a flood of fake SMTP requests to an electronic-mail server is a clear sample. Various security appliances are at present has the ability to identify the simple DOS attacks that come from an individual attacking node. Hence, DOS attacks have developed into a more complicated attack termed as Distributed DOS (DDOS). In DDOS, the DOS attack is proceeded from various sources around the Internet such that it would be tougher to trace and block the attacker. More details regarding DDOS in cloud computing can be identified in [16]. Although DDOS take major concentration in the media, it is not the only threatening variety of DOS attacks. Another type of DOS is Asymmetric application-level DOS attacks that develop the susceptibility in web-servers, databases, also other cloud resources. This variety of attack permits a malicious attacker to slow down an application with the help of very small attack payload, at times as small as 100 bytes. In [17], a method that protects on covariance-matrix was utilized to identify the DOS attacks. This method was confirmed to be highly efficient in identifying the DOS attacks that are based on flooding. A new DOS attack along with its counter-measure was launched in [18]. This attack works on the application level to identify a network bottle-neck in one of the links and concentrates on flooding this link.

C. Co-Resident DOS Attacks

Co-Resident DOS attacks is a special type of DOS attacks, which is a combination of co-resident and DOS attack. One research direction has enforced the Game Theory defense mechanisms in protecting the cloud infrastructure against Co-Resident DOS attack.

In co-resident DOS attack, the attacker services a VM inside the public cloud and handles the DOS from the serviced VM onto another VM within the same node. The attacker makes use of simple tools (like nmap and hping) to work out the perfect location of the VM in the cloud and perform a DOS attack on the bottle-neck network channel shared between the VMs. In [19] a detection method based on the game theory defense mechanisms was launched. A model was recommended to avoid the flooding attacks which were launched in [20]. This model can give the foundation of further survey in the topic of DOS flooding attacks prevention.

D. Defense Action Set

From the attacker perspective, it is proved that the attacker has the ability of triggering their targets to introduce the new VMs, capable of compromising a low risk account, and only begins same type of VMs. Thus, according to these attacker behaviors, the defense process is determined. The defense process initiates with defining the attacker behavior. Then the user accounts were gathered (as cluster) with the help of EDBSCAN and then labeled partially as low, medium or high risks. At last, the accounts will be divided with the help of semi-supervised learning- DAS3VM with branch and bound method. According to these results, the cost for the initialization of new VMs through new accounts is increased making it complex for the attacker to co-locate the target VMs.

E. EDBSCAN Clustering

DBSCAN clustering was utilized for clustering the user accounts earlier. We have two parameters in the DBSCAN algorithm, ε and MinPts, where ε is the maximum distance among the two neighbors, and MinPts is the minimum number of points in a cluster. Nevertheless, once MinPts is set, ε can be defined by drawing a k-distance graph (k = MinPts) as described in [21]. Alternatively, ε can be conceived as a function of MinPts. MinPts shouldn't be too small; else noise in the data will result in spurious clusters. According to these conditions, the clustering process is proceeded. Nevertheless the problem with the DBSCAN clustering process is the time complexity for the entire process. So, the Enhanced DBSCAN is suggested to rectify the time complexity issue. In EDBSCAN, the parameters Eps and MinPts are defined in a different way from DBSCAN. The procedure is provided in the following algorithm:

Input: List of points pointList and depth

Output: KD Tree

Function kdtree(pointList, depth)

Step 1: Select axis based on depth so that axis cycles through all valid values (axis=depth mod k)

Step 2: Sort point list and choose median as pivot element

Step 3: Create node and construct sub-trees

node location := median;

leftChild := kdtree(points in pointList before median, depth+1);

rightChild := kdtree(points in pointList after median, depth+1);

Step 4: Repeat Steps 1 - 3 till pointList is empty.

This algorithm is utilized in the EDBSCAN algorithm in order to minimize the time complexity issue. Let'd' be the distance of a point 'p' to its kth nearest neighbor, then the d-neighborhood of 'p' contains exactly k+1 points for almost all points 'p'. The d-neighborhood of 'p' comprises of more than k+1 points only if several points have accurately the same distance 'd' from 'p' which is quite doubtful. Moreover, modifying 'k' for a point in a cluster doesn't result in a huge changes of'd'. This only occurs if the kth nearest neighbors of p for k = 1, 2, 3, were pin-pointed approximately on a straight line which in general isn't true for a point in a cluster. For the provided k, a function k-dist from the database D is determined by mapping every point to the distance from its kth nearest neighbor. When categorization the points of the database in descending order of their k-dist values, the graph of this function provides some hints regarding the density distribution in the database. This graph is known as the sorted k-dist graph. If an arbitrary point 'p' is selected, set the parameter Eps to k-dist(p) and set the parameter MinPts to k, entire points with an equal or smaller kdist value will be core points. Nevertheless, as represented in k-dist graphs for k > 4 doesn't considerably vary from the 4-dist graph and they required being significantly more in computations. The appropriateness of value 4 to MinPts was further confirmed by various proposals [22]. Hence, the parameter MinPts is set to 4 at the time of experimentations. The 4-dist value of the threshold point is utilized as the Eps value for DBSCAN. These computed values were provided as input to the DBSCAN algorithm. The time requirement of DBScan algorithm is O (n log n) where n is the size of the dataset and since it is not an appropriate one to work with huge datasets. When it is linked with k-distance graph to automatically choose the MinPts and Eps values, it raised to O (n2 log n). The current research work makes use of a KD Tree (space partitioning tree) to minimize the time complexity to O (log n) time. With the help of KD-Tree finding k nearest neighbors for every n data point the complexity is O (kn log n). The k value is insignificant and hence it doesn't make much variations and so the time complexity becomes O (n log n). As of now the MinPts and Eps parameters were defined, they were used in the clustering of the user accounts.

Partial Labeling: After acquiring the cluster list, the next step is to distinguish then with the attacker's potential behaviors, and mark those clusters that are extremely possible to be malicious or legal. Once after completing the labeling process, the semi-supervised learning method is enforced for user account classification.

F. DAS3VM

The final step of classification process is to enforce semi-supervised learning techniques on the partially labeled dataset which is acquired from the previous step. We have three parameters in this algorithm. The regularization parameters λ handles the tradeoff among the maximizing the hyper plane margin, and minimizing the misclassification rate.

The second parameter λ' manages the power of unlabeled data, and give back the confidence in the cluster assumption. The third parameter r is computed according to the clustering result. So, to classify every node into one of the three types (low, medium or high risk), the "one-vs-all" approach is selected:

- For every three types of labels − H1, H2 and L, we construct the respective SVM, and then make use of it to test entire nodes.
- Every node is provided with three scores −SH1, SH2 and SL. A node is finally labeled as H1 (H2, L) if and only if SH1 (SH2, SL) is positive while the other two scores are negative. Else, the node is labeled as M (medium risk).
- Nodes labeled as H1 and H2 are combined to H.

This process of account classification is an effective way, as the account classification is highly accurate. But it can't be termed as 100% accurate, as the clever attackers will adjust their behaviors correspondingly. Nevertheless, it should be observed that the target of the suggested classification approach isn't perfectly labels the high risk accounts, but moderately selects the hyper plane according to L, so that it is a tedious process or expensive process for attackers to be classified as low risk. Once after accomplish this target, the two-player security game approach is computed. Nevertheless, the classification scheme gives the sub-optimal solutions which are required to be improved to prevent the maximum activities of clever attackers. So, the semi-supervised learning method is enforced with branch and bound method [23] for acquiring the globally optimal solutions.

G. Branch and Bound for DAS3VM.

The classification function f with respect to L over the space χ, where χ is generally a discrete form and it has to be reduced. A branch and bound algorithm has two main ingredients:

Branching: The region χ is iteratively split into smaller sub-regions. This earns a tree structure where every node corresponds to a sub-region.

Bounding: Let us assume two (disjoint) sub-regions (i.e. nodes) A and $B \subset \chi$. Deduce that an upper bound on the best value of f over A is known and a lower bound (say b) on the best value of f over B is known and that a < b. Then, there is an element in the subset A that is good when compared with entire elements of B. So, when searching for the global minimize, we nned to safely get rid of the elements of B from the search: the sub-tree corresponding to B is pruned.

The objective is to reduce the classification function over complete 2^u possible choices for the vector y_u, which comprises of the set χ which is launched above. The binary search tree has the following structure. Any node corresponds to a partial labeling of the data set and its two children correspond to the labeling of some unlabeled point. Then, one can correlate with any node a labeled set L comprising of both the original labeled examples and a subset S of unlabeled examples $\{(xj, yj)\}_{j \in S \subset [l+1 \ldots n]}$ to which the labels yj have been allotted. One can also correlate an unlabeled set U = [l + 1 . . . n] \ S respective to the subset of unlabeled points which isn't allotted to a label yet. The size of the sub-tree rooted at this node is therefore $2^{|U|}$. The tree's root has only the original set of labeled examples correlated with it, i.e. S is empty. The leaves in the tree match to a complete labeling of the dataset, i.e. U is empty. Rest of the nodes corresponds to partial labeling.

With reference to the upper bound, it is determined to have the following simple strategy: for a leaf node, the upper bound is nothing but simply the value of the function; for a non-leaf node, there is no upper bound. Alternatively, the upper bound is the best objective function which has been identified so far. The set A is the leaf corresponding to the best solution which has been identified so far and the set B is the sub-tree that is considering to be analyzed. Because of this choice for the upper bound, a natural way to discover the tree is a depth first search. Certainly, it is significant to go to the leaves as frequently as possible, to have a tight upper bound and therefore it perform aggressive pruning. Let $D(\alpha, yU)$ be the dual objective function, where y_U corresponds to the labels of the unlabeled points which isn't allotted a label yet,

$$D(\alpha, yU) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(K(x_i, x_j) + \frac{\delta_{ij}}{2C} \right)$$

The dual feasibility is $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i \, y_i = 0$

The vector $\alpha(yU)$ must be found that fulfills, because of dual is maximized.

$$D(\alpha(yU), yU) \leq \max D(\alpha, yU) = J)yU)$$

The target is to identify a choice for $\alpha(yU)$ so that a lower bound on Q can be estimated effectively. The choice corresponding to the lower bound presented above is the following. Train an SVM on the labeled points, acquire the vector α and finish it with zeros for the unlabeled points. Then Q (yU) is the same for entire possible labeling of the unlabeled points and the lower bound is the SVM objective function on the labeled points.

Then the branching process is done. Let s(L) be the SVM objective function trained on the labeled set.

$$s(L) = \min \frac{1}{2} w^2 + C \sum_{(x_i, y_i) \in L} \max(0, 1 - y_i(w.x_i + b))^2$$

The lower bound is s (L) is enforced for the branching strategy it comprises of choosing the following point in U,

$$\arg \max_{x \in U, y \in \pm 1} s(L \cup \{x, y\})$$

The algorithm is executed iteratively. Initially, the upper bound can either be set to +1 or to a solution found by another algorithm. The bound and branching is given as follows:

Algorithm 2: Branch and bound for DAS3VM

Function: (Y *, v) DAS3VM(Y, ub)

Input: Y: a partly labeled vector (0 for unlabeled)

ub: an upper bound on the optimal objective value.

Output: Y *: optimal fully labeled vector

v: corresponding objective function.

If $\sum \max(0, Y_i) > ur$ then

return

end if

$v \leftarrow SVM(Y)$ // Compute the SVM objective function on the labeled points

if v > ub then

return // lower bound is higher than the upper bound

end if

if Y is fully labeled then

Y*← Y

Return

end if

Find index i and label y // Find next unlabeled point to label

$Y_i \leftarrow -y$ // Start first by the most likely label

(Y*, v) ←DAS3VM(Y, ub)//Find (recursively) the best solution

$Y_i \leftarrow -Y_i$ // Switch the label

(Y *, v) ←DAS3VM (Y, min(ub, v)) // Explore other branch with updated upper-bound

if $v_2 < v$ then

$Y^* \leftarrow Y^*_2$ and $v \leftarrow v_2$ //Keep the best solution

End if

Therefore, the optimal solutions for the DAS3VM classification were acquired and so the user accounts are classified appropriately.

Two-Player Security Game Approach

As addressed in [10], after dividing the user accounts, the game approach is enforced for improving the defense mechanism. The attacker's nature, as identified earlier, when the attacker begins their first VM (there is no incentive for attackers to begin more than one VM initially, as none of them will co-locate with the targets), they are labeled as medium risk. So, to be reclassified as low risk, the attacker has to maintain the first VM running before initiating more VMs. This is termed as the initial cost. After being labeled as low risk, the attacker can generate as many VMs as required. Nevertheless, they have to cautiously manage the pace, so that they won't be reclassified as medium or even high risk. When it becomes more expensive to maintain the existing account which is being conceived as low risk rather than generating a new account (i.e., pay the initial cost again), the attacker will get rid of the exiting account. Table 1 indicates the comparison of the VM allocation policies with defense mechanisms with respect to attacker's overall cost.

Table.1. Attacker's overall cost comparison

Methods	Difficulty of achieving co-residence	Attacker's overall cost
Existing VM allocation policies	Low-Medium	Low
PSSF only	Medium-high	Medium
Defense mechanism with PSSF	High	High
Proposed Two player game approach	High	Very high

Under most of the current VM allocation policies, it is comparatively very simple for the malicious users to co-locate with their targets, and the entire cost is also affordable (quite low). If the PSSF policy is enforced, it will be complex to accomplish the co-residence. Nevertheless, since increasing the attacker's cost isn't one of the targets when the design of the policy, there is still room for enhancing this aspect. Under the defense mechanism [10], entire accounts were divided into various categories, and attackers are required to behave as normal users. So, their overall cost will be maximized desirably. Furthermore, the integration of the PSSF policy assures that the low probability of co-location. The two player game based defense mechanism is suggested in our work it further put an effort to raise the overall cost with optimal solutions for the huge data center. Hence, this defense mechanism is an efficient and practical countermeasure against the co-resident DOS attack.

IV. PERFORMANCE EVALUATION

The performance of the proposed two-player game approach based defense mechanism (referred as Two-player approach in graphs) is computed in CloudSim. The results are distinguished with the co-location resistant (CLR) algorithm proposed in [24], PSSF based game theoretical approach (referred as PSSF in graphs) [10] with respect to the account classification accuracy, precision, recall and attackers overall cost.

A. Classification Accuracy

Accuracy is the percentage with respect to the perfectly completed classification of user accounts as legal and malicious users.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FN + FP + TN)} * 100$$

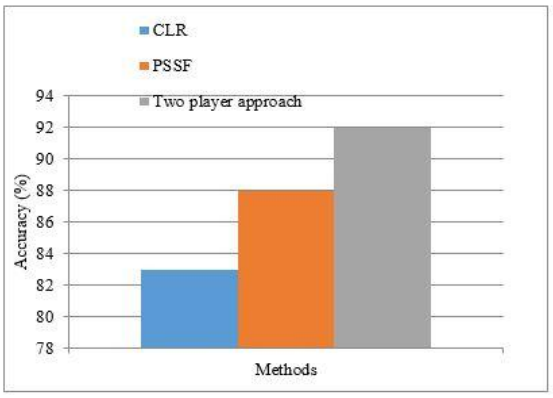


Figure.2. Accuracy comparison

Figure 2 indicates the comparison of the defense mechanisms with respect of accuracy. From the graph, it can be identified that the proposed Two-player approach gives highly accurate classification. This is due to the fact that the optimal solutions for DAS3VM were acquired by bound and branch method which raises the accuracy of classification. In the proposed research work, classification accuracy is enhanced by partially labeling the VM with the help of clustering technique before preceding the classification process. This can improve the classification accuracy with partially labeled information. The proposed research method shows 4.55 % increased accuracy when compared with the PSSF approach and 10.84 % than the CLR approach.

B. Precision

Precision is the exactness of the classification of the user accounts.

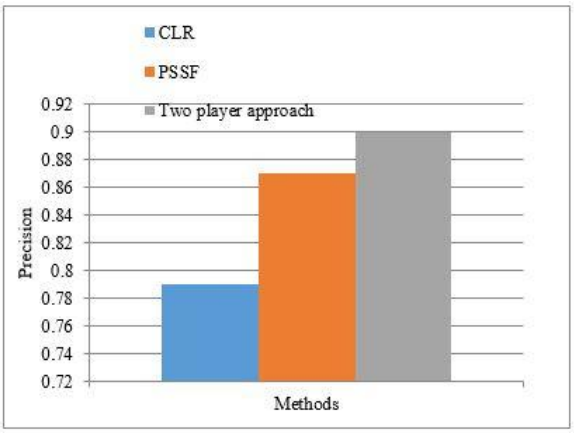


Figure.3. Precision comparison

Figure 3 indicates that the comparison of the defense mechanisms with respect to precision. From the graph, it can be identified that the proposed Two-player approach gives précised classification thanks to the optimal solutions for the huge data center. It is confirmed that the proposed defense mechanism can get rid of the possibility of attacks with mush probability. In the proposed research work, precision is unprepared by classifying the VM from the known partially labeled information. This can prevent the inaccurate prediction result because of the available knowledge. The proposed research method seems to give 3.4 % enhanced precision result when compared with the PSSF method, and 13.9 % enhanced precision result when compared with the CLR method.

C. Recall

Recall is the comprehensiveness of the classification performed in the cloud user accounts.

$$\text{Recall} = \frac{TN}{(TP + FN)} * 100$$

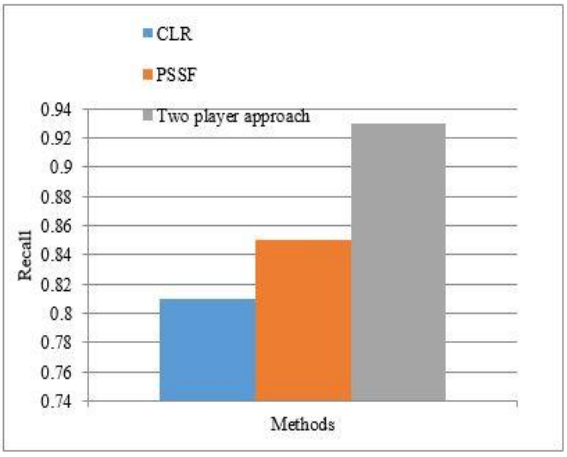


Figure.4. Recall Comparison

Figure 4 indicates that the comparison of the defense with respect to recall. From the graph, it can be identified that the proposed Two-player approach gives classification with high recall. The proposed approach raises the defense against co-resident DOS attacks with higher accuracy. Recall also optimized by classifying the VM as malicious or genuine with the help of the partially labeled information. Recall of the proposed research method is enhanced to 8.6% better when compared with the PSSF method and 14.1% better when compared with the CLR method.

D. Attacker's overall cost

This parameter assist in computing the cost acquired for introducing an attack i.e. generating a new account for starting a new VM. The cost is indicated in US dollars ($) for common cost evaluation.

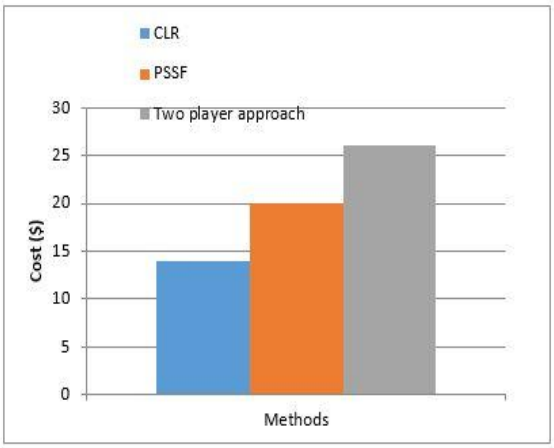


Figure.5. Attacker's overall cost comparison

Figure 4 provides the comparison of the attacker's overall cost for starting a new VM for initiating a co-resident DOS attack in the existence of the computed defense mechanisms. From the graph, the cost acquired by the attacker to begin a co-resident DOS attack is greater in proposed two-player approach when compared with the other methods. It is confirmed that the proposed defense mechanism makes it a tedious process for an attacker by creating an attack process highly expensive. Therefore, it forces the attacker to minimize risks and act as a normal user. The proposed research method indicates 30% enhanced performance when compared with the PSSF method and 85.7% enhanced performance when compared with the CLR method.

V. CONCLUSION

An efficient two-player game based defense mechanism is given in our work to reduce the attackers from introducing the resident DOS attacks. Even though various game based methods have been established earlier, most of them concentrate much on avoiding the side channels to get rid of the attacks. But they worn protect efficiently against the clever attackers. So, two-player security game concept was enforced, where the user accounts' clustering is performed by EDBSCAN technique, whereas the optimized classification is accomplished by DAS3VM with bound and branch method. This approach conceives huge size data centers, while assigning the VMs with the help of PSSF policy. This methodology reduces the probability of malicious users from initiating the new accounts for co-resident DOS attack launch effectively and the users were enforced to make use of only one account at a time as it became expensive. This is confirmed from the performance analysis results with respect to accuracy, precision, recall and attacker's overall cost. Therefore, the proposed defense mechanism raises the security in cloud computing environment.

REFERENCES

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., et al.: Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, University of California, Berkeley (2009)
2. Amazon. Amazon Elastic Compute Cloud (EC2). http://aws. amazon.com/ec2/
3. Ristenpart, T., Tromer, E., Shacham, H., Savage, S.: Hey, You, Get off of my cloud: exploring information leakage in third-party compute clouds. In: CCS'09: Proceedings of 16th ACM Conference on Computer and Communications Security, Chicago (2009)
4. Xu, Y., Bailey, M., Jahanian, F., Joshi, K., Hiltunen, M., Schlichting R.: An exploration of L2 cache covert channels in virtualized environments. In: Proceedings of 3rd ACM Workshop on Cloud Computing, Security (CCSW'11) (2011)
5. Zhang, Y., Juels, A., Oprea, A., Reiter, M.K.: HomeAlone: Co-Residency Detection in the Cloud via Side-Channel Analysis. In: Proceedings of 2011 IEEE Symposium on Security and Privacy, Berkeley (2011)
6. Keller, E., Szefer, J., Rexford, J., Lee, R.B.: Eliminating the hypervisor attack surface for a more secure cloud. In: Proceedings of ACM Conference on Computer and Communications, Security (CCS'11) (2011)
7. Raj, H., Nathuji, R., Singh, A., England, P.: Resource management for isolation enhanced cloud services. In: Proceedings of 2009 ACM Workshop on Cloud Computing Security, CCSW '09, Chicago (2009)
8. Bier, V. M., & Azaiez, M. N. (Eds.). (2008). Game theoretic risk analysis of security threats (Vol. 128). Springer Science & Business Media.
9. Han, Y., Chan, J., Alpcan, T., & Leckie, C. (2014, June). Virtual machine allocation policies against co-resident attacks in cloud computing. In 2014 IEEE International Conference on Communications (ICC) (pp. 786-792). IEEE.
10. Han, Y., Alpcan, T., Chan, J., Leckie, C., & Rubinstein, B. I. (2016). A game theoretical approach to defend against co-resident attacks in cloud computing: Preventing co-residence using semi-supervised learning. IEEE Transactions on Information Forensics and Security, 11(3), 556-570.
11. Yinqian Zhang, Ari Juels, Alina Oprea (2011) "Home Alone: Co-Residency Detection in the Cloud via Side-Channel Analysis" 2011 IEEE Symposium on Security and Privacy.
12. Adam Bates, Benjamin Mood, Joe Pletcher, Hannah Pruse, Masoud Valafar (2010) "Detecting Co-Residency with Active Traffic Analysis Techniques".
13. Han Y, Tansu Alpcan, Jeffrey Chan, Christopher Leckie, (2011) "Security Games for Virtual Machine Allocation in Cloud Computing".
14. Yu, S.: Distributed Denial of Service Attack and Defense. Springer, 2014.
15. Lenon, M.: Cloudare infrastructure hit with 400gbs ntp-based ddos attack, 2014. http://www.securityweek.com/cloudflare-infrastructure-hit-400gbs-ntp-based-ddos-attack
16. Kumar, N., Sharma, S.: Study of intrusion detection system for ddos attacks in cloud computing. In: Wireless and Optical Communications Networks (WOCN), 2013 Tenth International Conference on, pp. 1{5. IEEE, 2013.
17. Ismail, M.N., Aborujilah, A., Musa, S., Shahzad, A.: Detecting ooding based dos attack in cloud computing environment using covariance matrix approach. In: Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, p. 36. ACM, 2013.
18. Liu, H.: A new form of dos attack in a cloud and its avoidance mechanism. In: Proceedings of the 2010 ACM workshop on Cloud computing security workshop, pp. 65{76. ACM, 2010.
19. Bedi, H.S., Shiva, S.: Securing cloud infrastructure against co-resident dos attacks using game theoretic defense mechanisms. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics, pp. 463{469. ACM, 2012.
20. Zunnurhain, K.: Fapa: a model to prevent ooding attacks in clouds. In: Proceedings of the 50th Annual Southeast Regional Conference, pp. 395{396. ACM, 2012.
21. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf. Knowl. Discovery Data Mining (KDD), 1996, pp. 226–231.
22. Phung, D., Adams, B., Tran, K., Venkatesh, S. and Kumar, M. (2009) High Accuracy Context Recovery using Clustering Mechanisms, In proceedings of the seventh international conference on Pervasive Computing and Communications, PerCom Galveston, USA, Pp. 122-130
23. Chapelle, O., Sindhwani, V., & Keerthi, S. S. (2006). Branch and bound for semi-supervised support vector machines. In Advances in neural information processing systems (pp. 217-224).
24. Y. Azar, S. Kamara, I. Menache, M. Raykova, and B. Shepard, "Co-location-resistant clouds," in Proc. 6th ACM Workshop Cloud Comput. Secur, 2014, pp. 9–20.