Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.026360
Article

Tech Science Press

Randomized MILP framework for Securing Virtual Machines from Malware Attacks

R. Mangalagowri[1,*] and Revathi Venkataraman[2]

[1]Computer Science Department, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, 603203, India
[2]School of Computing, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, 603203, India
*Corresponding Author: R. Mangalagowri. Email: mangalar@srmist.edu.in
Received: 23 December 2021; Accepted: 23 February 2022

Abstract: Cloud computing involves remote server deployments with public network infrastructures that allow clients to access computational resources. Virtual Machines (VMs) are supplied on requests and launched without interactions from service providers. Intruders can target these servers and establish malicious connections on VMs for carrying out attacks on other clustered VMs. The existing system has issues with execution time and false-positive rates. Hence, the overall system performance is degraded considerably. The proposed approach is designed to eliminate Cross-VM side attacks and VM escape and hide the server's position so that the opponent cannot track the target server beyond a certain point. Every request is passed from source to destination via one broadcast domain to confuse the opponent and avoid them from tracking the server's position. Allocation of SECURITY Resources accepts a safety game in a simple format as input and finds the best coverage vector for the opponent using a Stackelberg Equilibrium (SSE) technique. A Mixed Integer Linear Programming (MILP) framework is used in the algorithm. The VM challenge is reduced by a firewall-based controlling mechanism combining behavior-based detection and signature-based virus detection. The proposed method is focused on detecting malware attacks effectively and providing better security for the VMs. Finally, the experimental results indicate that the proposed security method is efficient. It consumes minimum execution time, better false positive rate, accuracy, and memory usage than the conventional approach.

Keywords: Virtualization technology; security; cross-VM channel attack; VM-escape; R-VM-MILP algorithm (randomized VM allocation of security resources); Mixed Integer Linear Programming (MILP); SSE strategy; firewall-based monitoring method

1 Introduction

Cloud computing and cloud applications have become widespread within a decade. Most IT services and popular consumer computing apps like Dropbox and Netflix rely on infrastructure provided by Cloud Service Providers (CSPs) like Amazon and Google [1]. Data security and personal privacy are the most significant barrier to greater cloud adoption for businesses and individuals. Cloud-based systems employ libraries

developed for single-user server environments, in which adversaries can only communicate over well-defined network interfaces [2]. In the cloud, however, malicious programs and the crypto stack may operate on a single system, separated by a VMM (virtual machine manager). Virtualization-based technologies include simple frameworks that are available with scalability making their safety a genuine concern [3]. These technologies came into existence in the early 2000s or the late 1990s. The first commercial version of virtualization based on x86 architecture became famous and added to the availability of high-speed networks cloud computing services were born. Examples of this type of cloud can be Salesforce.com (1999) and Amazon Web Services (2002). Virtualizations, an essential element of cloud service approaches like infrastructure as a Service (IaaS), were first seen in 2006 with Amazon's EC2 [4]. A VMM or a hypervisor is a software component that maintains the accessibility of host resources to a group of VMs. Each VM can run an Operating System (OS) separately x`, but with common shares on memory and I/O devices [5]. This research work aims to classify vulnerabilities of virtualized systems from the angle of platform security and based on attack routes, including origins and destinations. This work can be instrumental in exploring virtualization boundaries and their vulnerability to threats.

(1) Guest VM to VMM-An attacker handled by a guest VM can assault the VMM since VMM ensures isolation for the entire environment. This is the most dangerous attack. As virtualization becomes more widespread, this type of attack will become more frequent, making it critical to protect against it. (2) Guest to Host—In Type II hypervisor, hosts are primary OS that runs VMM where they may also include specific guest VMs like Dom0 in Xen hypervisors. Guest VMs avoid hypervisors while executing their code as confidential guests in hosts [6] and are similar to CMM assaults from guests. (3) Host to VMM–These intruders control VMMs through attacks on hosts where most of them may be privileged users. (4) Guest to Guest-Guest VM attacks can interrupt VMMs aggregations and extract information or resources of other guest VMs controlling VMMs [7]. (5) Guest to Self and Host to Self–These operations enhance privileges under similar circumstances using guest VMs or host OSs. The threats had existed before virtualizations became famous and have been reviewed on standalone systems. Generally, guests are frequent sources for VM's vulnerabilities as VMs have separate privileges for guests in operations, leading to their compromises. Guest to VMM, Guest to Host, and Guest to Guest are grouped into a single class in classifications [8]. Moreover, sandboxes achieve process segregations, vital security factors in VMMs. These elements restrain information leakage in side-channel shares when multiple programs run on the same physical hardware. Various effective side-channel attacks targeting the software layer are solely developed in non-virtualized systems and obtain sensitive data effectively [9]. Until recently, physical attacks were widely discarded in cloud settings to recognize a successful physical attack. With the implementation of virtualization, a new range of security concerns has emerged:

- In cloud platforms, multiple customers may share a single physical host, making separation even more essential. The guest VM's data must be safeguarded against threats posed by other guests [10]. DoS (Denial of Service) attacks that are executed against hosts at VMs need to be controlled, as they affect entire VMs servers.
- VMM is one of the most significant software components in virtualization techniques, and it can be affected by software defects, resulting in security vulnerabilities.
- Novel types of malware have emerged that target virtualization, and because they function at high ability than the OS, they are susceptible to regular anti-virus software.
- When virtualization is used, several standard security measures for isolated systems are degraded or made ineffective [11]. Traditional security techniques are not always valid in virtualized environments (i.e., VMs may be sharing their physical hosts with other malicious VMs).

Conventional techniques have a significant disadvantage in that they manage multiple securities by identifying all conceivable resource assignment combinations. The main problem of the research work is Cross-VM side attacks in cloud computing. Although several research and methodologies were introduced, the security is not ensured significantly. The existing approaches have drawbacks with attacks and security factors. Also, the existing works have problems with an error rate, memory storage, and execution time. A novel randomized security resource allocation approach has developed that scale to challenge orders of magnitude larger than previous techniques. Section 2 explains the benefits and drawbacks of some of the most modern solutions for cloud computing security challenges. The proposed methodology is discussed in Section 3. The results and discussion are detailed in Section 4. The conclusion and future work are discussed in Section 5.

2 Literature Review

This section covers virtualization's advantages and disadvantages in cloud computing. Irazoqui et al. [12] proposed the framework to guard VMs in Xen and VMware against Cross-VM cache attacks. The Advanced Encryption Standard (AES) of various prominent cryptographic libraries, such as Open SSL, Polar SSL, and Libgcrypt, have non-constant execution timings and are subject to Bernstein's correlation attack. The vulnerability exists even if the VMs operate on multiple levels on the same computer. Experiments on Amazon EC2 and Google Compute Engine demonstrate the vulnerability's practical implications. The findings of this investigation reveal that AES implementations in popular libraries and data encrypted with AES on primary cloud services still pose a security concern.

Saeed et al. [13] presented the zero-day cross-VM network channel attacks in their study where simulated internal cloud virtual network. Malicious VMs reroute network traffic to target VMs in the initial stages of attacks, and promptly, like Air cracks, use this opportunity to extract decrypted information of target VMs. Though restricted in their access rights, the attackers execute ROPs (Return-Oriented Programs), exploit networks to connect to root domains and obtain tool stacks (root domain) that have no direct access. Countermeasures to these types of attacks are also discussed.

Kumar et al. [14] suggested utilizing a broadcast domain to hide the server's location so that the opponent can't track it beyond a specific point. The OpenStack platform is used to construct a cloud computing environment in this work. This approach has proved its capability of safeguards against most known attacks. This approach's broadcasting domain ranges are based on the availability of bandwidths. They thus could be made a part of cloud computing platforms where virtual resources dynamically get added or eliminated.

Zhang et al. [15] developed a security framework for an access-driven side-channel attack. A malicious machine obtains fine-grained data from a victim VM on the same computer. This is the first attack against a symmetric multiprocessing system virtualized with a contemporary VMM (Xen). Nowadays, such methods are very prevalent, ranging from virtualized desktops to sandbox applications or OS compromises to clouds that co-locate the activities of equally distrusting clients. Constructing a side-channel overcomes obstacles like core migration, channel noise, and the difficulties of preventing victims with enough frequency to extract fine-grained data.

Liu et al. [16] presented a covert channel-aware scheduler that prioritizes security to prevent side-channel attacks. The scheduler may handle the execution time of simultaneously executed VMs, and it can also introduce noise regularly to reduce the risk of side channels. Also, a prototype that allows for overlapping control and noise injection is created. However, the new scheduler will enable users to dynamically change scheduling factors to respond to various situations and create a balance between performance and security.

Yang et al. [17] established unified realistic information leakage systems that depicted parameters influencing side channels and the impacts of threats on side-channels due to migrations that restrict co-residency lengths. The study used Integer Linear Programming (ILP) to identify the best migration strategies due to migrations' high computing costs. Their basic genetic algorithm improved the scalability and optimality of migrations, where their experimental results showed that migration-based defenses could provide higher security assurances at minimal performance costs.

Wang et al. [18] proposed novel physical memory side-channels that considered balloon driver's errors, a standard mechanism that balances multiple VM's physical memories. The study's scheme showed security against side-channel assaults, frequent in IaaS clouds. Further, it could transfer more fine-grained information than conventional cross-VM side channels. By utilizing Xen, it is demonstrated how to transmit data through a side channel.

Using the BLP (Bell-La Padula) paradigm created by Bell and LaPadula, Wu et al. [19] presented an access control technique called preventing VM escape (PVME). As a result, the PVME idea was implemented using thorough virtualization architecture with just 4% to 8% latency overhead.

Alnaim et al. [20] examined privilege escalations and avoidance of VMs in their study. Their misuse patterns were a part of an ongoing library, the first step in creating security reference designs for NFVs (Network Function Virtualizations).

PVMH (Prevent VM hopping) is an access control solution proposed by Dong et al. [21] in their study to protect VMs against hopping attacks. Their scheme based on the Biba and BLP approaches ran on Xen platforms, where their experiments demonstrated their approach's effectiveness in eliminating VM hopping attacks with increased accuracy.

Lv et al. [22] suggested a security evaluation method for cloud services based on stochastic game nets. Cloud services' virtualization security risk paradigm is properly defined using graphical tools, and virtualization security risk elements are accurately assessed. The experimental results show that this approach effectively replicates cloud services' complex and dynamic safety problems. From the foregoing discussion, it is evident that the existing approach offers numerous advantages. Only a substantial Stackelberg equilibrium exists for all Stackelberg games, not a weak Stackelberg equilibrium.

Furthermore, the leader can frequently produce strongly favorable equilibrium by arbitrarily adopting a plan that induces the follower to strongly choose the required technique. The drawback is that existing techniques manage various security by identifying all resource assignment possibilities. The goal of this project is to enhance the security level.

The existing methods have issues with security, and the number of attacks is still not handled in the previous work. Still, difficulties are occurred in preventing victims with enough frequency to extract fine-grained data. Also, the existing techniques had a problem with performance cost and communication overhead. Hence, the overall attack detection accuracy is lower, and security is not ensured.

3 Proposed Methodology

This research paper introduces a secured framework for mitigating the Cross-VM Side Channel attack and VM Escape. This paper proposes utilizing a unique approach to hide the server's location so that the opponent cannot track the target server after a certain point. This will handle safe cache management; however, there are various other techniques to reduce the risk of a cross-VM attack. Fig. 1 shows the overall process of the proposed methodology.

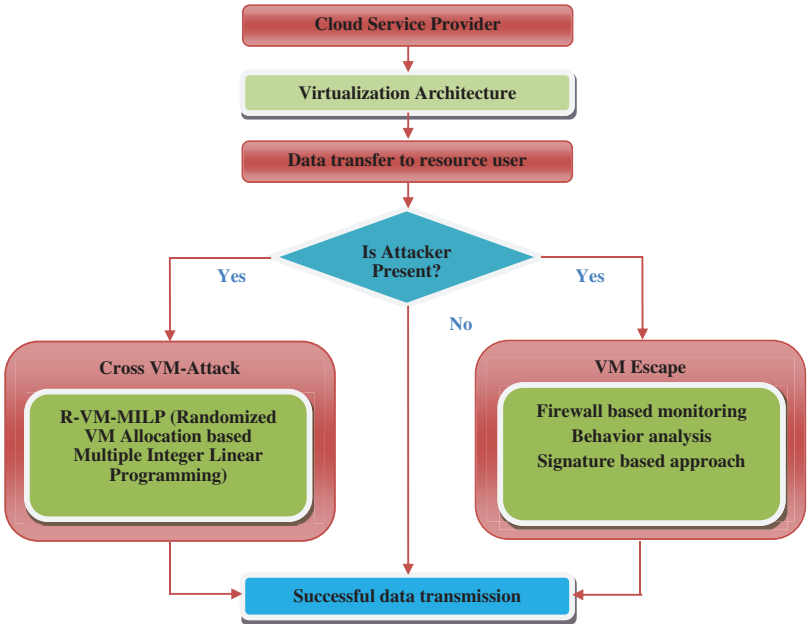

Figure 1: The overall process of the proposed methodology

The proposed strategy assumes that an attacker can use any of the ways to identify the target server, establish a VM on the target server, and attack the co-located VMs. As a solution to this issue, the servers are grouped in such a way that the opponent is unable to track the target. All requests must pass through a broadcast domain on their way from source to destination, which disrupts the opponent's path and prevents the opponent from detecting the target beyond this point.

- Initially, an R-VM-MILP algorithm (Randomized VM Allocation of Security Resources) has been used to mitigate the cross-VM channel assault. This algorithm takes a safety game in compact form as input and resolves effective coverage vector matching to an SSE strategy for a defender. A Mixed Integer Linear Programming is used in MILP.
- The VM-escape problem is then reduced by a Firewall-based monitoring mechanism that is a hybrid of behavior-based detection and signature-based virus detection.

3.1 Cross VM Side-Channel Attack

Attackers create covert channels on shared hardware resources and use the same for gathering the necessary information. When victims execute tasks, important information, including cryptographic keys or other sensitive information, is extracted. Side-channel attacks have existed for a long time, but their impact has grown significantly due to the introduction of cloud computing. While the phenomena that are reliably connected to software function vary depending on the hardware's specific qualities, that phenomenon is employed as a side-channel. Two common examples are the timing of certain hardware functions and the hardware device's acoustic properties. High-rate hardware functions are frequently used as side-channels since they convey data quicker, providing conditional information of operating programs.

3.1.1 Compact Security Game Model

Many security domains, including Los Angeles International Airport (LAX) and Federal Air Marshal Service (FAMS), assign resources to cover a wide range of potential targets. During their combinatorial explosions in strategy spaces and payoff representations [23]. Succinct representations of resources are

developed for algorithms before they are used. The proposed technique is similar to existing compact representations of games but angling more towards the security domain. Let $T = \{t_1, \ldots, t_n\}$ denote set of to-be-attacked targets. R = $\{r_1, \ldots, r_m\}$ denote the resources used by the defender to cover these targets (for example, in FAMS domain targets are flights and air marshals are resources). Consider that all resources are similar and that any target can be given to them. As indicated in Tab. 1, every target has four payoffs indicating possible consequences of an attack on the target. Based on whether the defender covers the target or not, the payoff for the unsecured attack is denoted as $U^u_{\ominus}(t)$, and the payoff for a concealed attack is $U^c_{\ominus}(t)$. Likewise, the attacker's payoffs are $U^u_{\psi}(t)$ and $U^c_{\psi}(t)$.

Table 1: Example payoffs for an attack on a target

	Covered	Uncovered
Defender	5	−20
Attacker	−10	30

The challenging characteristic of this approach is to identify that payoffs are solely dependent on assaulted target and regardless of the defender's concealments. For identifying every concealed target, ct, relevant components of the defender's methods are integrated into coverage vectors C. Corresponding attack vectors A denote assault on single targets with a probability of one. The projected reward offered to defenders for defending assaults and coverage vectors is shown in Eq. (1) while the anticipated payoff for an attack on target t, namely C, is computed using Eq. (2). The follower uses the same notation as the leader by displacing Θ with Ψ.

$$U_{\ominus}(C, A) = \sum\nolimits_{t \epsilon T} a_t.(c_t.U^u_{\ominus}(t) + (1 - c_t)U^u_{\ominus}(t)) \tag{1}$$

$$U_{\ominus}(t, C) = c_t U^c_{\ominus}(t) + (1 - c_t)U^u_{\ominus}(t) \tag{2}$$

$$\Gamma(C) = \{t : U_{\psi}(t, C) \geq U_{\psi}(t', C) \forall \in t' T\}. \tag{3}$$

In strong SSE, the attacker chooses target in the attack group with maximum payoff for the defender. Let t^* denote optimal target. The expected SSE payoff for defender is $\widehat{U_{\ominus}}(C) = U_{\ominus}(t^*, C)$, and for attacker is $\widehat{U_{\psi}}(C) = U_{\psi}(t^*, C)$.

Some small-format security game is also defined in a standard format. The attack vector A matches the attacker's clear strategies by one strategy per target. Every possible resource allocation relates to a strategy based on its standard form for the defender. Every accessible resource is mapped to a target in a resource allocation; thus, there are n targets. Eq. (4) shows how a coverage vector relates to a mixed approach for two resources and four targets. The probability of a strategy covering targets I and j is δ⊖^(i,j). The probability of covering target 1 is the total probabilities assigned to clean strategies that contain 1.

$$\frac{\delta^{1,2}_{\ominus} + \delta^{1,3}_{\ominus} + \delta^{1,4}_{\ominus} = c_1}{\delta^{1,2}_{\ominus} + \delta^{2,3}_{\ominus} + \delta^{2,4}_{\ominus} = c_2 \delta^{1,3}_{\ominus} + \delta^{2,3}_{\ominus} + \delta^{3,4}_{\ominus} = c_3 \delta^{1,4}_{\ominus} + \delta^{2,4}_{\ominus} + \delta^{3,4}_{\ominus} = c_4} \tag{4}$$

The defender's payoff function establishes a payoff for every resource allocation schedule and target combination. The value is U ⊖^u if the allocation conceals the target, and U ⊖^c if it is not, which is equal for also the attacker payoff function. Every strategy is described by n continuous variables in the compact form and payoff function by 4n variables. In normal form, the defender's procedure requires n Choose m variables, whilst the attacker's procedure remains unchanged.

3.1.2 R-VM-MILP Algorithm (Randomized VM Allocation of Security Resources)

Safety games are inputs, while Randomized VM Allocations of Security Resources are based on Mixed Integer Linear Programming (MILP), which resolves best coverage vectors, similar to defender SSE approaches. Eqs. (5)–(11) explain the proposed MILP, while Eqs. (6) and (7) are used to generate assault vectors that have single targets with a probability of 1. Eq. (8) restricts coverage vectors to possibilities in the interval [0,1], while Eq. (9) limits coverage to available resources. Eq. (10) shows the scheduled payoff of the defender, based on the attacked target in A. An upper bound of $U_{\Theta}(t, C)$ is arranged on d, only for the attacked target. For all other targets, RHS is indiscriminately high. As objective maximizes d, for any optimal solution $d = U_{\Theta}(C, A)$. Likewise, Eq. (11) makes the attacker choose a plan to attack the group of C. The initial portion of constraint implies that $-U_{\Psi}(t, C) \geq 0$, such that k must be the maximal payoff for attacking some target. The second portion forces $k - U_{\Psi}(t, C) \leq 0$ for any attacked target in A. If attack vector specifies a target that is not maximal, this constraint is degraded. The objective and Eqs. (10)–(11) show that C and A are mutual best-responses in an optimal solution.

$$max\ d \tag{5}$$

$$a_t \in \{0, 1\} \forall_t \in T \tag{6}$$

$$\sum\nolimits_{t \in T} a_t = 1 \tag{7}$$

$$c_t \in [0, 1] \qquad \forall_t \in T \tag{8}$$

$$\sum\nolimits_{t \in T} c_t = m \tag{9}$$

$$d - U_{\Theta}(t, C) \leq (1 - a_t).Z\ \forall_t \in T \tag{10}$$

$$0 \leq k - U_{\psi}(t, C) \leq (1 - a_t).Z\ \forall_t \in T \tag{11}$$

An optimal solution to R-VM-MILP approach is depicted. Initially, legal coverage vectors administered by mixed strategies are explained, and later how a full SSE is built from optimal solution is depicted at last.

Algorithm 1: The proposed R-VM-MILP algorithm

Input: Coverage vectors,

Output: Optimized payoff function

targets ← T sorted by $U_{\psi}^{u}(t)$

payoff[t] ← $U_{\psi}^{u}(t)$, coverage [t] ← 0

left ← m, next ← 2

while next ≤ n do

addedCov[t] ← $\dfrac{payoff\ [next] - U_{\psi}^{u}(t)}{U_{\psi}^{u}(t) - U_{\psi}^{u}(t)} - coverage\ [t]$

if coverage[t] + addedCov[t] ≥ 1 then

end if

coverage [t] += addedCov[t]

left- = $\sum\limits_{t \in T} addedCovt[t]$

next ++

end while

3.2 VM ESCAPE Attack Model

Assuming cloud service providers or their infrastructures can be trusted in executions virtual machines escape attacks while allowing malicious persons to operate on VMs [24]. Since the cloud service providers deliver services through multiplexed physical infrastructures, attackers' machines are assumed to be functioning on the same systems but as potential threats to VMs. Thus, making them compromise Hypervisors or host machines by manipulating shared resources like network intrusion message cues, file systems, CPU cache etc. Fig. 2 shows the virtual machine (VM) escape attack model.

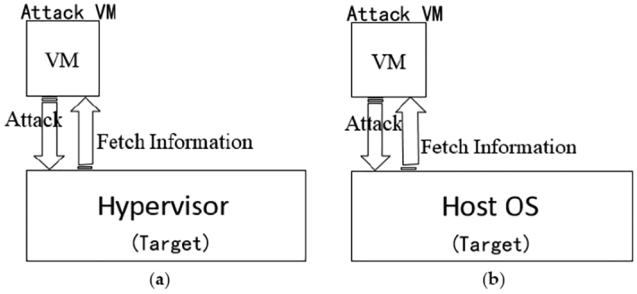

Figure 2: Virtual machine (VM) escape attack model

The following two major assault models are discussed here: As depicted in (a), an attacker conducts overflow assaults on physical resources such as the CPU, RAM, disc, and others using analog commands such as ioctl and virtio. The attackers use direct contacts with the Host OS to interfere with the host's privileged commands and resources, as depicted in (b). The following are steps in VM escape attack:

1. Placement Malicious attackers arrange their malicious VMs on a similar physical system as the target Hypervisor or hosts, which are known as placement. This is a probability and relatively difficult process that necessitates the utilization of a variety of co-residency identification approaches.

2. Extracting Information After arranging, the attacker tries to access Hypervisor's or hosts' key permissions and then collects data from the host's and Hypervisor's other virtual machines.

3.2.1 Firewall Based Monitoring

The virtual firewall is a virtualized environment comprising network firewall services. Furthermore, by monitoring and filtering packets that operate with both virtual machines and distributed virtual switches, this firewall responded to these assaults [25]. Furthermore, the packets entering and exiting the specific network access are monitored, and also destination IP, protocols, and sources of network packets are inspected to recognize whether they should be allowed to pass or not. As a result, after the filtering and matching with their pre-determined procedures and policies, the transferred packets are either accepted or declined. This firewall implements portable security regulations on VMs migrating from one system to another. Furthermore, the need for particular hardware deployment is avoided to reduce costs. The other firewall is familiar with the physical system.

The proposed study framework is depicted in Fig. 3 above. It is made up of two key processes: behaviour detection and malware classification utilizing a proposed signature-based malware approach. It also included a secure environment and database platform to meet the framework's requirements.

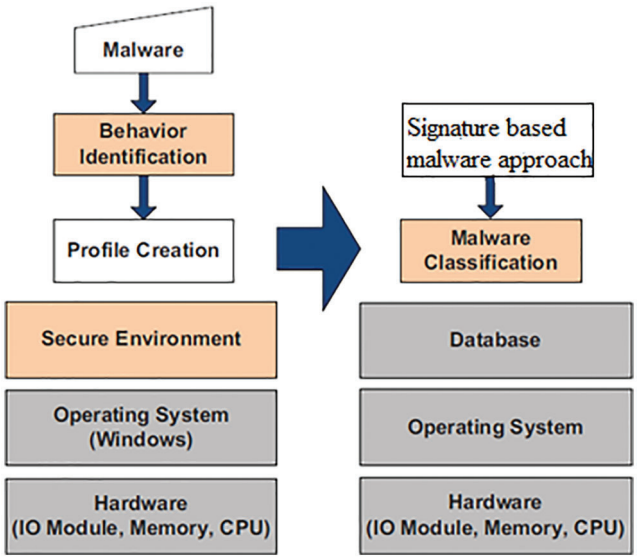

Figure 3: Firewall based monitoring

Behavior Identification

The technique of determining types and characteristics of malicious software is known as malware behaviour analysis which outlines the malware attacks' objective and function. Understanding malware behavior is important so as to utilize it for malware definition and classification. Dynamic analysis is performed to complete this task by carrying out tun time analysis and resource monitoring. A new behavior definition procedure is employed to filter and develop possible malware behaviour.

Malware Detection Using Signature-Based Approach

This method is guided by malware behaviour defined in each sample. Various malware samples will have similar characteristics and behaviors because it was created using polymorphic and metamorphic techniques. Related classification approaches will be used to detect the shared behavior of each malware family. The malware classification in this study is done utilizing a signature-based approach.

Signature Extraction: Signatures are formed by extracting a sequence of 2 to 5 system calls till tracking ceases, as shown in Fig. 4. Signatures are a series of system calls made in certain orders, and duplicates are removed to identify specific executables [26]. This procedure is repeated for malware and benign traces, and subsequently, on receipt of all malware signatures, they are sorted by their frequency of appearances. This procedure is also used for benign executables. Instead of categorizing signatures, this work compiles a master list of all benign signatures. Hence, the ultimate objective of the proposed system is to strike a balance between minimal malware signatures and high detection accuracies where the generated lists reduce iterations of malware detection algorithms. Quantitative measures are used to assess the capabilities of the proposed approach's malware behavior investigations where the corresponding variables emerge from behavior analysis. Previously proposed static and dynamic methods compare the generated variables.

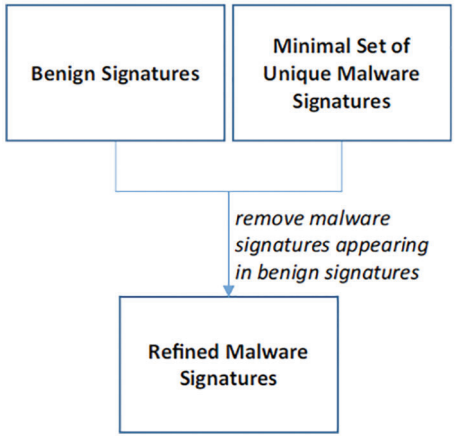

Figure 4: Refining malware signature extraction

Advantages of the proposed system

The proposed system is focused on increasing security and handling the number of attacks effectively in this work. It dealt the difficulties more accessible to prevent victims with enough frequency to extract fine-grained data. The performance cost and communication overhead is reduced considerably. Hence the overall attack detection accuracy is improved and security is ensured importantly. Also it provides better performance metrics in terms of improved execution time, memory storage and error rates.

4 Results and Discussion

Experiments were conducted out on a computer with dual Xeon 3.2 GHz processors and 2 GB of RAM, operating RHEL 3. To resolve MILPs, CPLEX 9.0.0 with default system parameters is utilized. For making a game with a certain number of targets and resources, randomly choose four integer payoffs for each target. Uniform [0,100] is used for U Θ^c and U Ψ^u, while Uniform [100,0] is used for U Θ^u and U Ψ^c. The parameters listed below are used to evaluate the security performance of the new method.

In this work, the Cross-VM side attacks is evaluated in a cloud setting, each of which possesses two CPUs, co-reside on a single-socket quad-core processor, specifically an Intel Core 2 Q9650 with an operating frequency of 3.2 GHz [27,28].

Tab. 2 illustrates the attack detection rate of the proposed and existing methods.

Table 2: Performance comparison table for attack detection rate of proposed method

No. of VMs	PVME	ERASER	R-VM-MILP
10	67.23	74.12	90.6
20	69.56	76.25	91.82
30	73.24	78.51	92.25
40	75.41	81.78	93.41
50	78.89	87.05	94.05

Fig. 5 illustrates the relation of Attack detection rate of conventional and newly designed methods. If there are 50 VMs, then the new attack detection rate is high as 94.05%, whereas the existing method ERASER has 87.05% and the PVME method has 78.89%. Signature extraction is applied for malware and benign traces, and subsequently, on receipt of all malware signatures, they are sorted by their frequency of appearances.

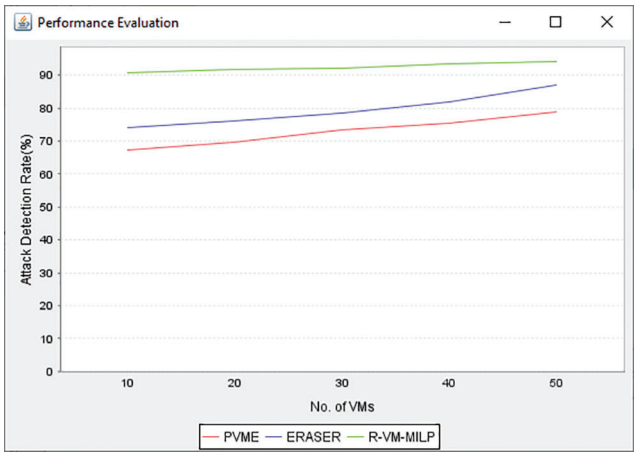

Figure 5: Attack detection rate for the proposed and existing security-based framework

Tab. 3 illustrate the False Negative rate of new and existing methods.

Table 3: Performance comparison table for false-negative rate of the proposed method

No. of VMs	PVME	ERASER	R-VM-MILP
10	16.25	12.57	9.35
20	18.52	16.56	11.27
30	20.56	19.05	13.97
40	23.25	22.43	16.23
50	26.52	25.69	18.43

In Fig. 6 False Negative rate of conventional and newly designed methods is compared. It is evident from the experiments that the new R-VM-MILP approach has less False Negative rate compared to the existing security model. The ultimate objective of the proposed R-VM-MILP system is to strike a balance between minimal malware signatures and high detection accuracies. Hence it is focused on reducing the false-negative rates efficiently.

Tab. 4 illustrate the False positive rate of novel and existing methods.

In Fig. 7, False-positive rate of conventional and newly designed methods is compared. It shows that the R-VM-MILP approach has less false positive rate than the traditional security model. This firewall implements portable security regulations on VMs migrating from one system to another. Furthermore, the need for particular hardware deployment is avoided so as to reduce costs.

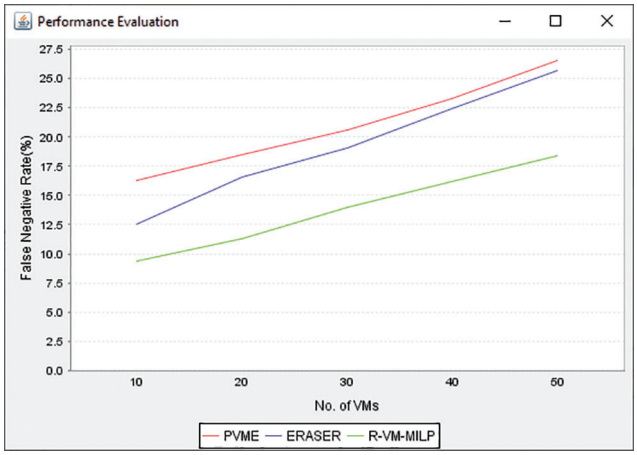

Figure 6: False negative rate of new and existing security-based framework

Table 4: Performance comparison table for false-positive rate of the proposed method

No. of VMs	PVME	ERASER	R-VM-MILP
10	15.35	12.19	9.12
20	17.55	13.72	10.35
30	19.62	15.22	11.41
40	22.52	17.51	12.32
50	25.76	19.42	14.68

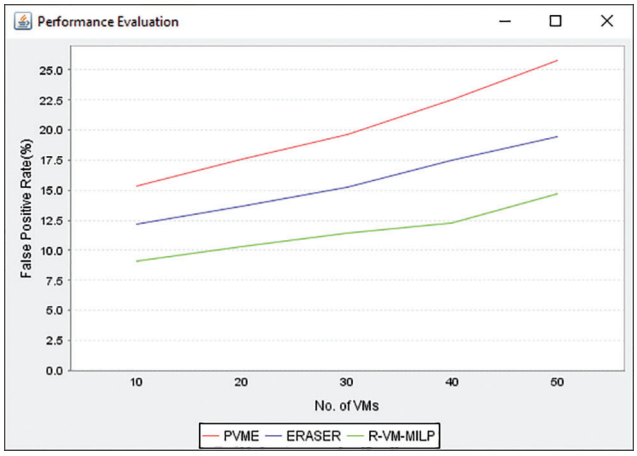

Figure 7: False positive rate of proposed and existing security-based framework

Tab. 5 illustrate the Performance comparison table for memory usage of new approach.

Fig. 8 illustrates the performance comparison of Memory usage of the proposed and existing approaches. Here the number of VMs is in increasing order for evaluating the new method's performance. If there are 50 VMs, then the memory usage of the proposed (R-VM-MILP) algorithm is less (8.81 MB), whereas the existing method ERASER has 10.72 MB and PVME has 13.81 MB.

Table 5: Performance comparison table for memory usage (MB) of proposed method

No. of Cloudlets	PVME	ERASER	R-VM-MILP
10	9.56	7.05	5.25
20	10.78	7.58	5.92
30	11.56	8.41	6.49
40	12.48	9.15	7.56
50	13.81	10.72	8.81

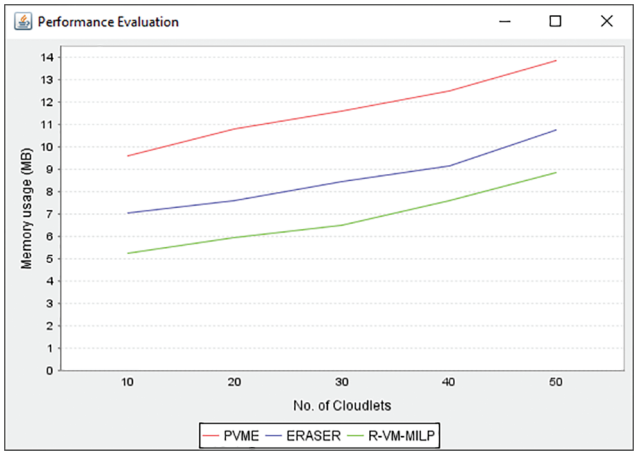

Figure 8: Memory usage of proposed and existing security based framework

Tab. 6 Illustrate the performance comparison table for Execution time (sec) of novel and traditional methods.

Table 6: Performance comparison table for Execution time (sec) of the proposed and existing method

No. of Cloudlets	PVME	ERASER	R-VM-MILP
10	3.15	2.89	2.15
20	4.90	4.12	3.56
30	5.50	4.65	3.89
40	6.35	5.12	4.32
50	7.11	6.45	5.69

In Fig. 9, the performance of Execution time (sec) of newly designed and existing approaches is compared. Here, the number of VMs is increasing order for evaluating performance in the new method. When there are 50 VMs, the execution time of the proposed (R-VM-MILP) algorithm is less as 5.69 (s), whereas the existing method ERASER has 6.45 (s) and the PVME method has 7.11 (s). In this proposed research work, optimal resources are selected, and therefore this proposed R-VM-MILP algorithm increases the response time speed hence it maximizes the overall VM performance.

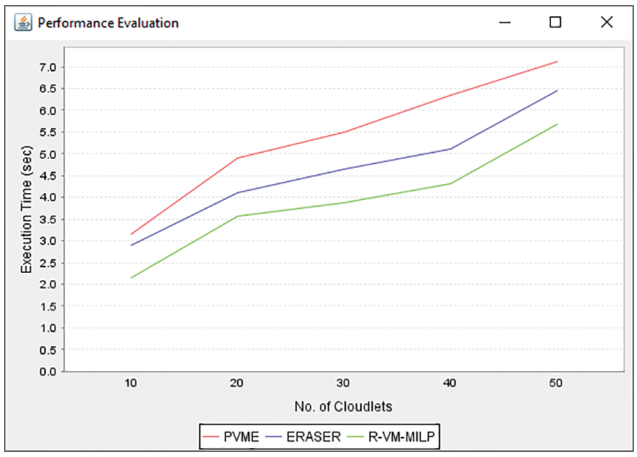

Figure 9: Execution time (sec) of proposed and existing security based framework

Fig. 10 illustrates the comparison between the PVME, ERASER and R-VM-MILP techniques for the accuracy metric. It shows that the existing PVME, ERASER methods provide lower accuracy whereas the proposed R-VM-MILP scheme provides higher accuracy. The proposed method improves the data transmission speed by allocating the optimal resources over cloud environment. The VM attacks observe the system metrics through cloud requests and detect any possible misuse trends. Thus it is used to increase the throughput and accuracy efficiently in the cloud environment.

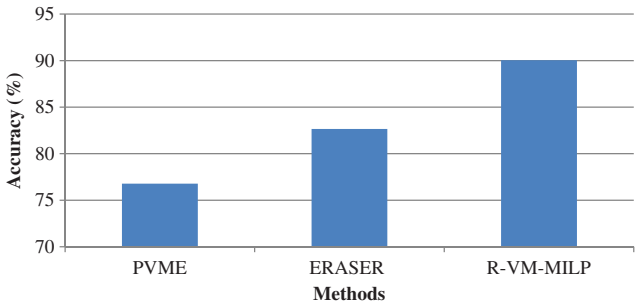

Figure 10: Accuracy

5 Conclusion

In many security domains, allocating limited resources is a major issue. The essential concept is to employ a compact model of security games that allows for exponential memory and runtime. Solving Stackelberg games in general results in developing much faster algorithms for payoff-constrained safety games, which are common aspects of security. In this work, R-VM-MILP method is introduced for calculating optimum security game solutions that can scale up to massive games with large pools of resources and targets. To combat cross-VM channel assaults, an R-VM-MILP method (Randomized VM Allocation of Security Resources) is designed. This method takes a compact safety game as input and finds an optimum coverage vector for the defender, similar to an SSE approach. As a result, the developed model is more effective than the previous approach at mitigating the attack. Memory utilization is reduced using this proposed approach, improving network performance. The VM-escape problem is then reduced by a Firewall-based monitoring mechanism that is a hybrid of behavior-based detection and signature-based virus detection. R-VM-MILP algorithm is proposed to detect and discard

the VM attacks and other considerably using a firewall-based process. Hence it increases the security higher, and various attacks are discarded. Compared to the existing security model, the proposed R-VM-MILP algorithm takes less time to execute (5.69 s), accuracy by 90.04%, and uses less memory (8.81 MB). This also focuses on advanced security measures to improve the solution.

Acknowledgement: The Author with a deep sense of gratitude would thank the supervisor for his guidance and constant support rendered during this research.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. Sierra-Arriaga, R. Branco and B. Lee, "Security issues and challenges for virtualization technologies," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–37, 2020.

[2] S. Anwar, Z. Inayat, M. F. Zolkipli, J. M. Zain, A. Gani *et al.,* "Cross-VM cache-based side channel attacks and proposed prevention mechanisms: A survey," *Journal of Network and Computer Applications*, vol. 93, no. 9, pp. 259–279, 2017.

[3] M. A. Shahid and M. Sharif, "Cloud computing security models, architectures, issues and challenges: A survey," *The Smart Computing Review*, vol. 5, no. 6, pp. 602–616, 2015.

[4] F. Bazargan, C. Y. Yeun and M. J. Zemerly, "State-of-the-art of virtualization, its security threats and deployment models," *International Journal for Information Security Research (IJISR)*, vol. 2, no. 34, pp. 335–343, 2012.

[5] J. S. Reuben, "A survey on virtual machine security," *Helsinki University of Technology*, vol. 2, pp. 1–5, 2017.

[6] Y. Shoaib and O. Das, "Pouring cloud virtualization security inside out," *arXiv preprint arXiv:1411.3771*, pp. 1–13, 2014.

[7] M. Kazim, R. Masood, M. A. Shibli and A. G. Abbasi, "Security aspects of virtualization in cloud computing," in *Proc. IFIP Int. Conf. on Computer Information Systems and Industrial Management*, Krakow, Poland, pp. 229–240, 2013.

[8] B. Grobauer, T. Walloschek and E. Stocker, "Understanding cloud computing vulnerabilities," *IEEE Security &Privacy*, vol. 9, no. 2, pp. 50–57, 2010.

[9] A. R. Riddle and S. M. Chung, "A survey on the Security of hypervisors in cloud computing," in *Proc. IEEE 35th Int. Conf. on Distributed Computing Systems Workshops*, Columbus, OH, USA, pp. 100–104, 2015.

[10] A. Nezarat and Y. Shams, "A game theoretic-based distributed detection method for VM-to-hypervisor attacks in cloud environment," *The Journal of Supercomputing*, vol. 73, no. 10, pp. 4407–4427, 2017.

[11] V. Shrivastava and D. S. Bhilare, "SAFETY: A framework for secure IaaS clouds," *International Journal of Advanced Networking and Applications*, vol. 6, no. 6, pp. 2549–2555, 2015.

[12] G. Irazoqui, M. S. Inci, T. Eisenbarth and B. Sunar, "Fine grain cross-vm attacks on xen and vmware," in *Proc. 2014 IEEE Fourth Int. Conf. on Big Data and Cloud Computing*, Sydney, NSW, Australia, pp. 737–744, 2014.

[13] A. Saeed, P. Garraghan, B. Craggs, D. van der Linden, A. Rashid, and S. A. Hussain, "A cross-virtual machine network channel attack via mirroring and tap impersonation," *In IEEE 11th Int. Conf. on Cloud Computing (CLOUD)*, Francisco, CA, USA, pp. 606–613, 2018.

[14] B. Kumar, K. Abhishek, A. Kumar and M. P. Singh, "System and method for mitigating cross vm attacks in cloud computing by securing the network traffic," in *Proc. 2015 IEEE Symp. on Computer Applications & Industrial Electronics (ISCAIE)*, Langkawi, Malaysia, pp. 221–225, 2015.

[15] Y. Zhang, A. Juels, M. K. Reiter and T. Ristenpart, "Cross-VM side channels and their use to extract private keys," in *Proc. 2012 ACM Conf. on Computer and Communications Security*, New York NY, United States, pp. 305–316, 2012.

[16] F. Liu, L. Ren and H. Bai, "Mitigating cross-VM side channel attack on multiple tenants cloud platform," *Journal of Computers*, vol. 9, no. 4, pp. 1005–1013, 2014.

[17] C. Yang, Y. Guo, H. Hu, W. Liu and Y. Wang, "An effective and scalable VM migration strategy to mitigate cross-VM side-channel attacks in cloud," *China Communications*, vol. 16, no. 4, pp. 151–171, 2019.

[18] Z. Wang, R. Yang, X. Fu, X. Du and B. Luo, "A shared memory based cross-VM side channel attacks in IaaS cloud," in *Proc. IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*, San Francisco, CA, pp. 181–186, 2016.

[19] J. Wu, Z. Lei, S. Chen and W. Shen, "An access control model for preventing virtual machine escape attack," *Future Internet*, vol. 9, no. 2, pp. 20, 2017.

[20] A. K. Alnaim, A. M. Alwakeel and E. B. Fernandez, "Threats against the virtual machine environment of NFV," in *Proc. 2019 2nd Int. Conf. on Computer Applications & Information Security (ICCAIS)*, Riyadh, Saudi Arabia, pp. 1–5, 2019.

[21] Y. Dong and Z. Lei, "An access control model for preventing virtual machine hopping attack," *Future Internet*, vol. 11, no. 3, pp. 82, 2019.

[22] J. Lv and J. Rong, "Virtualization security risk assessment for enterprise cloud services based on stochastic game nets model," *IET Information Security*, vol. 12, no. 1, pp. 7–14, 2017.

[23] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordónez *et al.,* "Computing optimal randomized resource allocations for massive security games," in *Proc. 8th Int. Conf. on Autonomous Agents and Multiagent Systems*, Budapest, Hungary, vol. 1, pp. 689–696, 2009.

[24] J. Wu, Z. Lei, S. Chen and W. Shen, "An access control model for preventing virtual machine escape attack," *Future Internet*, vol. 9, no. 2, pp. 1–19, 2017.

[25] X. Ma, X. Fu, B. Luo, X. Du and M. Guizani, "A design of firewall based on feedback of intrusion detection system in cloud environment," in *Proc. 2019 IEEE Global Communications Conf. (GLOBECOM)*, Waikoloa, HI, USA, pp. 1–6, 2019.

[26] B. I. Santoso, M. R. S. Idrus and I. P. Gunawan, "Designing network intrusion and detection system using signature-based method for protecting openstack private cloud," in *Proc. 2016 6th Int. Annual Engineering Seminar (InAES)*, Yogyakarta, Indonesia, pp. 61–66, 2016.

[27] A. Shahzad and A. Litchfield, "Virtualization technology: Cross-VM cache side channel attacks make it vulnerable," in *Proc. Australasian Conf. on Information Systems, ACIS 2015*, Adelaide, Australia, 2015.

[28] Y. Zhang, A. Juels, M. K. Reiter and T. Ristenpart, "Cross-VM side channels and their use to extract private keys," in *Proc. 2012 ACM Conf. on Computer and Communications Security*, New York, NY, United States, pp. 305–316, 2012.