

Chapter 2

Security-Aware Virtual Machine Placement in Cloud Data Center

Abstract Infrastructure as a Service (IaaS) facilitates the provisioning of virtual machines (VMs) in cloud computing platform for disjoint customers in a highly scalable, flexible, and cost-efficient fashion. However, introducing new VMs to a physical server where vulnerable VM already exists could lead to potential security risks to the new ones. Furthermore, even the physical server itself could be compromised by attackers through one of these vulnerable VMs. Therefore, VM placement could bring great impact over the security level of the whole cloud. In this chapter, we first quantify the security risks of cloud environments based on virtual machine vulnerabilities and placement schemes. Based on our security evaluation, we present a novel VM placement algorithm that can minimize the cloud's overall security risks by considering the connections among VMs. According to the experimental results, our approach can greatly improve the survivability of most VMs and the entire cloud. The computing costs and deployment costs of our techniques are also practical.

2.1 Introduction

Infrastructure as a Service (IaaS) such as Amazon Web Services (AWS) has been attracting more and more customers due to the ability to perform virtual machine (VM) provisioning and thereby offering the highest level of flexibility and scalability. One of the most challenging tasks in IaaS is the placement of virtual machines. There have been many kinds of strategies developed for the VM placement problem based on different principles. For example, network-aware VM placement tries to place VMs with large communication requirements close to each other in order to minimize the overall network costs in the cloud [1, 2], while energy-aware VM placement tries to find an optimal placement of VMs with the goal of minimizing the energy consumption for the cloud [3, 4].

This chapter includes copyrighted materials, which were reproduced with permission of IEEE and the authors. The original article is: Xuebiao Yuchi and Sachin Shetty, "Enabling security-aware virtual machine placement in IaaS clouds," *IEEE Military Communications Conference (Milcom 2015)*, pp.1554–1559, 26–28 Oct. 2015, ©IEEE. Reprinted by permission.

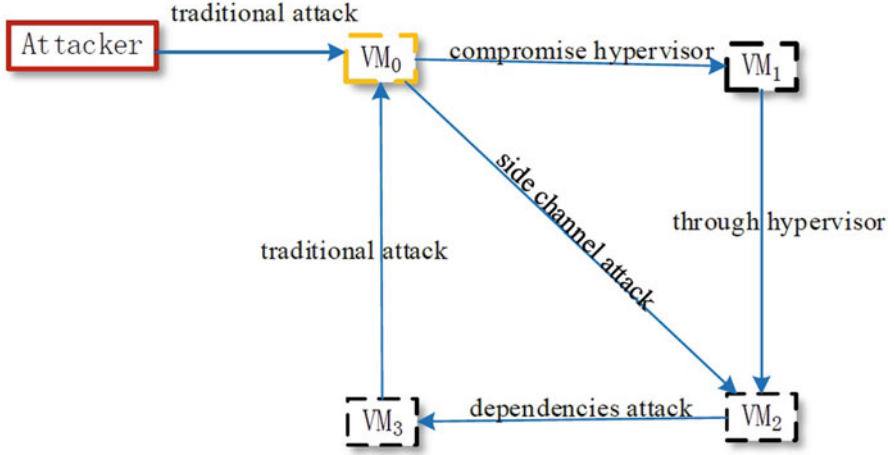


Fig. 2.1 Example of VM attack graph

However, most of these VM placement strategies do not take security risk factor into consideration. Security risk is one of the most major obstacles that affects the acceptance of cloud-based services. Meanwhile, prevalent known vulnerabilities are found to be very common in public VM images that are provisioned by cloud service providers [5, 6]. After scanning a number of public VM images, researchers [6] found that image publishers may leave some unwanted information (e.g., passwords, keys, and other credentials) in their images. A prevalent VM image with known vulnerabilities can be instantiated by a large number of users in cloud, therefore it may generate a large number of security vulnerabilities for attackers. As shown in Fig. 2.1, introducing a new VM image with known vulnerabilities (*vm0*) to a physical server can lead to security risks to the co-locating VMs. This is due to the fact of the 1 to n mapping relationship between the physical server and VMs, which makes vulnerabilities propagate rapidly across the entire IaaS infrastructure. Even the physical server itself could be taken over by attacker if the VM hypervisor (*vm1*) is compromised through one of the VMs. For example, in the case of side-channel based attacks (towards *vm1*) through VM co-location which is one of the common attacks suffered by IaaS platform, the adversaries can map the internal VM placement of the cloud and mount cross-VM side-channel attacks by placing malicious VMs on the victim's physical machine. Therefore, with the new computing model of public cloud, it is easier for attackers to launch attacks through prevalent vulnerabilities.

Obviously, those attacks are VM placement-based and their success largely depends on the placement strategies of the cloud. Therefore, VM placement could cause great impact on the cloud's overall security condition. In order to minimize the security risks of the cloud and alleviate the customer's security concerns, it is necessary to develop security-aware VM placement strategies in which VMs

with high risks will be separated from VMs with low risks, and the possibility of allocating a vulnerable VM or “bad neighbor” on a physical machine where VMs with higher security level already exist will be reduced.

In this chapter, we first conduct VM security evaluation based on their vulnerabilities. Then we try to quantify the security risks of physical machines based on their current VMs placement as well as dependencies among these VMs. Finally, based on our security evaluation, we develop a novel security-aware VM placement algorithm which can minimize the security risks for the cloud. Experimental results show that our solution can greatly reduce the overall security risks of the cloud.

The rest of this chapter is organized as follows. We discuss related work in Sect. 2.2. Section 2.3 describes the metrics we use for evaluating the security risks of both VMs and physical machines in our model. Section 2.4 presents the details of our VM placement algorithm we developed. Section 2.5 presents the experimental results of our algorithm. We discuss our work in Sect. 2.6. Finally we conclude the chapter in Sect. 2.7.

2.2 Related Work

Various VM placement strategies have been proposed for cloud data centers to reduce their network overhead [1, 2] or energy consumption [3, 4]. However, there are very few efforts on VM placement strategies to minimize the security risks for the cloud platform. In [7], researchers propose a VM placement algorithm based on incompatibilities between users. Each cloud user can submit a list of adversary users with whom it does not want to share a PM. Next, the lists of adversary users are merged to create incompatible groups that are taken into account when placing a VM. This work provides an interesting solution to improve the security of cloud computing by performing isolation between users. This placement algorithm does not take into account user’s security preferences because it does not incorporate any security metrics. In [8], the authors develop VM migration techniques based on Markov chain analysis, which aims to minimizing the security risks considering the connections among virtual machines and improving the survivability of the whole cloud. However, the problem of initial VM placement problem is not considered in their effort. In [9], the authors propose a system of security metrics specific to the cloud computing and use the metrics to develop virtual machines placement algorithms. However, the security metrics ignore the security risks caused due to co-resident VMs. In [10], Saeed et al. present a security-aware approach for resource allocation framework in clouds that allows for effective enforcement of defense-in-depth for cloud VMs. They model the cloud provider’s constraints and customer’s requirements as a constraint satisfaction problem (CSP), which can be solved using Satisfiability Modulo Theories (SMT) solvers to reduce risk and improve manageability in cloud. However, the authors formulate the problem as a satisfiability problem and not as an optimization problem. As a result, the solution is not optimal and only satisfies the input constraints.

2.3 Security Evaluation

The security evaluation procedure consists of evaluating the risks of both VMs and physical machines in the cloud. First, we quantify each VM's vulnerabilities based on the US National Vulnerability Database (NVD) [11], in which all vulnerabilities are scored according to the Common Vulnerability Scoring System (CVSS) [12]. We then calculate the probability of risk for each VM by exploring dependency relations with VMs in the cloud. Finally, the security score for each physical machine will be inferred based on the risk of the hosted VMs.

2.3.1 VM Vulnerability Identification

In our work, we use NVD to identify the vulnerabilities of the VMs. To fully understand the use of NVD in the calculation of security risks, it is necessary to understand the concepts of CVSS and Common Vulnerabilities and Exposures (CVE). CVSS provides an open framework to estimate and quantify the software vulnerabilities of various vendors. CVSS is adopted by several organizations such as CERT, IBM, and Cisco to prioritize the response to the vulnerabilities they encounter in their day to day activities. CVSS is currently maintained by the Forum of Incident Response and Security Teams (FIRST). CVE is a dictionary that assigns unique identifiers for all the security vulnerabilities that are publicly known. CVE is used as the industry standard for vulnerability and exposures names. Once vulnerability is discovered, it is assigned a unique CVE Identifier (e.g., CVE-2012-0015), brief description, and references such as advisories or vulnerability reports.

The NVD is the repository which provides CVSS scores for all CVE vulnerabilities. At present, NVD contains information about over 74,000 CVE vulnerabilities (as of 12/31/2015). NVD was created by the government of USA to help the Department of Homeland Security to warn public about common computer vulnerabilities. These vulnerabilities now include the latest attacks on cloud computing environments. The NVD website provides XML feeds for all the CVE vulnerabilities with CVSS metrics for all the years from 2002 to present which can be downloaded. We can use the data obtained from NVD to identify the vulnerabilities of the VMs based on the cloud operating system version running on them. The CVSS metrics obtained from the NVD are used to calculate the risks. There are several attributes that describe the nature and specifics of the CVE vulnerability. The CVSS quantifies the severity of vulnerabilities according to various attributes. The typical attributes required to compute security metrics include CVE identifier, CVSS score, CVSS vector, CVSS exploit score, CVSS impact score, vendor, product name, versions affected, and description.

2.3.1.1 VM Security Evaluation

The CVSS base score is the primary metric and describes the severity of the vulnerability. The base score uses an interval scale of (0,10) to measure the severity of vulnerabilities, which corresponds to three discrete states: low severity, medium severity, and high severity. First we check with NVD to collect potential vulnerabilities for OS and software in VM. Vulnerability scanner tools, such as Nessus and Qualys, are available to conduct this job. Since it's possible for a VM to have more than one vulnerabilities, it is usually desirable to aggregate the scores of individual vulnerabilities for each VM [13]. Here we simply choose the single most critical vulnerability (as designated by the CVSS Base Score) of each VM as this VM's vulnerability score, with the assumption that the vulnerable level of a VM is not higher than the weakest vulnerability of that VM. However, some other assessment models using CVSS values can also be used for our VM vulnerability quantification, such as Time-to-Compromise [14], Vulnerability Exposure [15], etc.

We now have quantified value for each VM's vulnerability. Then we need to map the quantified vulnerability to the possibility of compromise for each VM by exploring dependency relations among all VMs. There are already many research efforts on how to discover dependency relations between VMs. This, identifying the dependency of VMs, is out of the chapter's scope. In this chapter, we simply use the network topological structure information like IP address and network port numbers generated by the common network statistics tool netstat as their dependency relations. After all dependency relations are obtained, we can construct the VM Dependency Graphs as shown in Fig. 2.2.

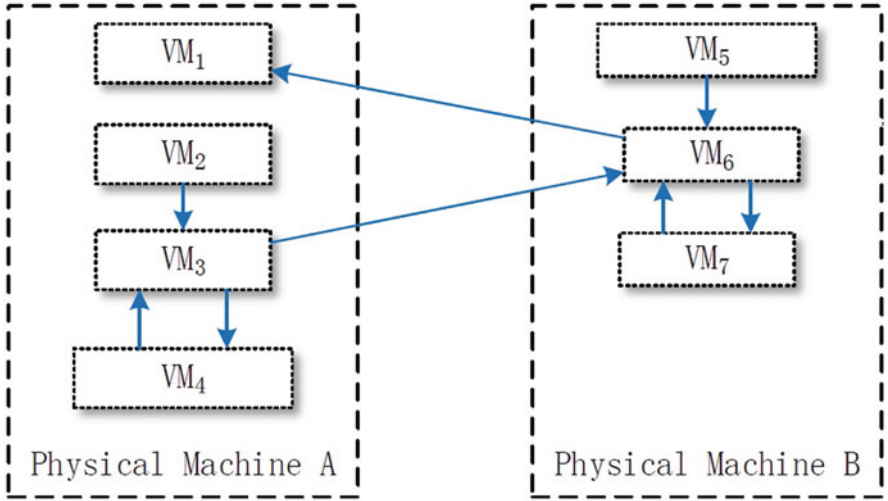


Fig. 2.2 VM dependency relations example

Here we use a linear mapping function to map the quantified vulnerability to the possibility of compromise for each VM according to their dependency relations as revealed in Fig. 2.2. Given a VM (denoted as VM_a) and the vulnerability score of VMs connected with VM_a is V_1, V_2, V_m , then the possibility of compromise for VM_a , as denoted by C_{VM_a} , is given by $C_{VM_a} = \frac{V_a}{\sum_{i=1}^m V_i}$.

The probability of exploitation for a VM quantifies the security level of this VM, which can correspond to three discrete states: low compromise, medium compromise, and high compromise.

2.3.2 Physical Machine Security Evaluation

We have computed the probability of exploitation of VM in the cloud. Next, we need to calculate the probability of survivability for each physical machine based on the security level of VMs hosted. Note that if any one of the VMs presented on the physical machine is compromised by adversaries, then the physical machine will be compromised with high probability. In other words, the survival possibility of a physical machine is the possibility that all owned VMs can survive in the attack. Given a physical machine PMA and a set of $VMs = VM_1, VM_2, VM_n$ which currently located at PMA , and the compromised probabilities for these VMs are $C_{VM_1}, C_{VM_2}, C_{VM_n}$, then the survivability score for the physical machine PMA , as denoted by S_{PMA} , is given by $S_{PMA} = \prod_{i=1}^n (1 - C_{VM_i})$.

The survivability score quantifies the security level of physical machines, which can also correspond to three discrete states: low survivability, medium survivability, and high survivability. Now that we have obtained both risk probability score for each VM and survivability score for each physical machine, we present our VM placement scheme based on these scores.

2.4 Secure Aware VM Placement

From the previous discussion, we can learn that the success of attacks highly depends on the placement strategy of the cloud. Thus, our approach is to find a systematic solution to place VM which can reduce security risks for both VMs to be placed and physical machines in the cloud as much as possible. For a VM whose risk probability score is low, it's infeasible for it to be placed on a physical machine with low survivability, which would result in the VM's risk probability increasing. On the other hand, for a physical machine with high survivability, it's also not feasible to place VMs with high risk probability, which would lead to great negative impact to the survivability level of this physical machine. Therefore, in a secure aware VM placement manner, both the VM's risk and physical machine's survivability should

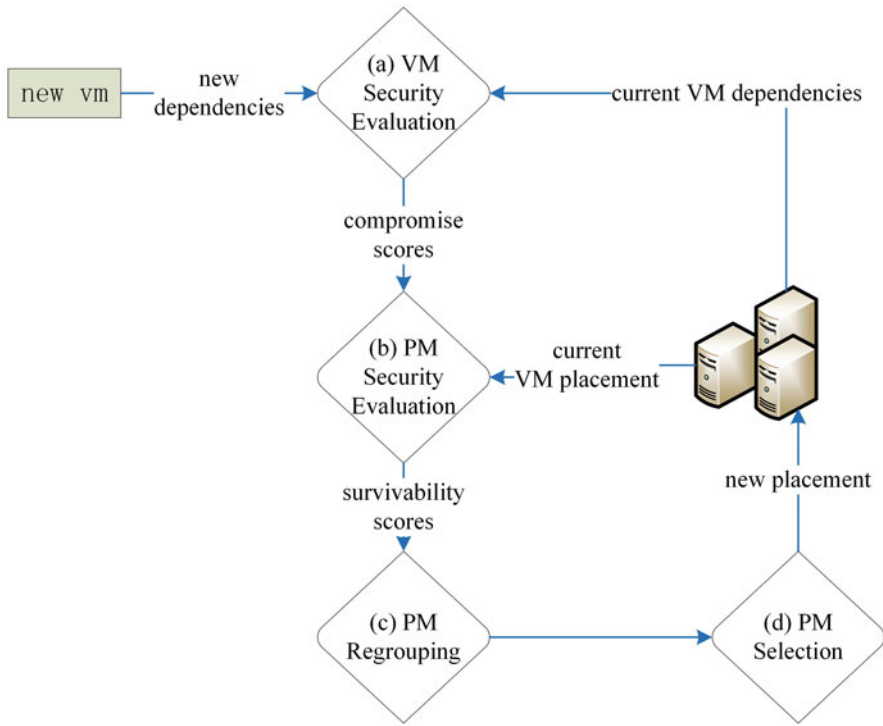


Fig. 2.3 Overview of the VM placement procedure

be considered simultaneously. For example, it is reasonable for a VM with low risk probability to be placed on physical machine whose survivability is high.

Figure 2.3 illustrates the VM placement scheme. When a new VM arrives, we first need to update the compromise score C_{vm_j} for each of those presented VMs (phase a), based on possible dependency relationships introduced by the new VM (in practice, this is usually given in terms of VM connections by users). Note that for each of those presented VMs, its C_{vm_j} will remain the same if there exists no connection between this VM and the new vm. With all presented VM's C_{vm_j} updated, we can then recalculate the survivability score S_{pm_i} for all presented physical machines (phase b). Similarly, for a single physical machine, its S_{pm_i} will keep the same if there is no connection between any of its presented VMs and this new vm.

Then we need to group (or regroup) those physical machines based on their updated S_{pm} , as some of them might be degraded to the other groups due to their connections with vm (phase c). In this chapter, those physical machines will be identified into one of the three separate groups (namely, low survivability, medium survivability, and high survivability). On the other hand, the new vm will also be marked with one of the three labels (low compromise, medium compromise, and

Table 2.1 Groups for VM and physical machine with different security levels

Group #	C_{VM}	S_{PM}
1	Low	High
2	Medium	Medium
3	High	Low

high compromise) according to its C_{VM} value. Here we suppose this new vm is labeled with medium compromise, which means it is going to be placed into some physical machine in the group medium survivability (see Table 2.1). In practice, in case there are no more physical machines left in this group for placing more VMs, the new vm will be otherwise introduced to those physical machines in other groups, based on specified regulations customized by the cloud provider. Here we assume that the overall physical machine capacities for each group are unlimited for new VM placement.

Next, we need to decide which PM within this group we should choose to place the vm (phase d). Obviously, unless the vm is compromise free, introduction of vm into any physical machine will definitely bring additional survivability loss to it, with the loss rate being $(1 - C_{VM})$. Here we choose to place the vm onto the physical machine whose survivability loss will be of the least when introducing the vm. We can easily conclude that we should choose the physical machine whose S_{PM} is of the lowest to place the vm. Below is the algorithm for the overall VM placement procedure we have described (Fig. 2.4).

2.5 Simulation Results

We conducted simulation studies to evaluate the performance of the proposed scheme. We generated 253 new VM requests to a data center including 120 physical machines. Each physical machine can hold up to ten VMs at most. The number of VMs that already presented on each physical machine is random set between (1, 10). Each VM (both new and existed) will have up to 8 randomly chosen dependencies with the other VMs. For the simplicity, here we assume the compromise possibilities for both new VMs and existed VMs are randomly distributed between (0, 1).

We compared the new placement algorithm with the random one to investigate the improvement of security levels. First, we provide the before and after comparison of the survivability possibility for all physical machines in Fig. 2.5. With the new placement algorithm, 95.0 % physical machines obtained improved survivability. The maximum survivability improvement is 81.7 % and the average improvement of survivability is 25.7 %.

Table 2.2 gives the distribution of physical machines in different groups before and after each placement, showing that the number of physical machines with high S_{PM} reduces much less under the new placement algorithm. Therefore, the new placement could greatly reduce the overall security loss of the cloud while reasonably allocating new VMs based on their corresponding security levels.

Fig. 2.4 VM placement algorithm**Algorithm 1** VM placement (vm, PM, VM)

```

1: Input:  $vm$ : new VM being placed
            $PM$ : list of PMs available in cloud
            $VM$ : list of VMs presented on each PM
2: Output:  $pm_k$ : the PM selected for the new VM
3: for each  $pm_i \in PM$ 
4:   for each  $vm_j \in pm_i$ 
5:      $C_{vm_j} \leftarrow C'_{vm_j}$ 
6:   end for
7:    $S_{pm_i} \leftarrow S'_{pm_i}$ 
8:    $PM_{group\#} \ll = pm_i$ 
9: end for
10:  $k \leftarrow 0, r \leftarrow 1$ 
11: for each available  $pm_i \in PM_{group(vm)}$ 
12:   if  $S_{pm_i} < r$ 
13:      $k \leftarrow i, r \leftarrow S_{pm_i}$ 
14:   end if
15: end for
16:  $S_{pm_k} \leftarrow S_{pm_k} * (1 - C_{vm})$ 
17: return  $pm_k$ 

```

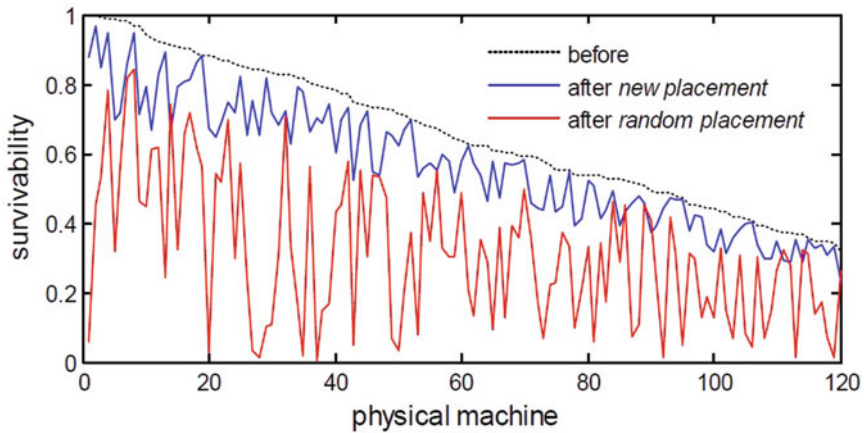
**Fig. 2.5** Comparison of survivability

Table 2.2 Number of physical machines within each group before and after placement

Group #	$S_P M$	Before placement	After random placement	After proposed placement
1	High	51	6	29
2	Medium	55	31	65
3	Low	14	83	26

2.6 Discussion

The new placement procedure is straightforward without drawing in any overweight models, which makes it much easier to deploy in practice. In addition, there also exists no specific restriction in the total number of groups, which enables this procedure with great flexibility. As such, the number of groups could be customized independently by the cloud provider without sacrificing any performance issues.

The placement of new VMs could lead to extra loss of survivability for a physical machine, while the new placement algorithm tends to choose physical machines with less survivability within each group to place the new VMs. Thus, the new placement procedure is naturally compatible with the real world where physical machines with less survivability will be given larger chance for placing new VMs, until they are ultimately filled up or just degraded to the other groups.

Note that we locate new VMs successively by treating the presence of new VMs as time discrete. In practice, multiple new VMs may arrive at the same time. As such, their final placement in the cloud may vary due to different allocating order, which can further lead to different impact to the cloud's security status. Analysis of multiple new VMs allocation procedure will be part of our future work. Moreover, the current version of our algorithm only considers security factor. However, network, energy saving, load balancing, and other factors should also be considered in realistic applications. The demands in terms of resources such as CPU, memory, and computation duration are not taken into account either. The integration of these factors is a work in progress.

2.7 Conclusion

The usage of appropriate VM allocation strategies is critical to minimize overall security risks for the clouds against potential attacks, since the success of attacks highly depends on the placement strategy of the cloud. The main contribution of this chapter is the development of algorithms for security-aware allocation of VMs in cloud systems. Based on the security evaluation for both VMs and physical machines, we developed novel VM placement scheme in which new VMs can be

placed optimally in security-aware manner while the security risks for the cloud platform can be greatly reduced.

However, the current version of our algorithm does not take into account the constraints in terms of resources such as CPU, memory, and computation duration. The integration of these resources is a work in progress. Our final idea is to consider multiple objectives such as minimizing energy consumption and network cost, while respecting all the security constraints.

Acknowledgements This work is based on research sponsored by the Office of the Assistant Secretary of Defense for Research and Engineering (OASD(R&E)) under agreement number FAB750-15-2-0120. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Office of the Assistant Secretary of Defense for Research and Engineering (OASD(R&E)) or the US Government. This work is also supported in part by an ARO grant W911NF-12-1-0055, National Science Foundation (NSF) Grant HRD-1137466, Department of Homeland Security (DHS) SLA grant 2010-ST-062-0000041 and 2014-ST-062-000059.

References

1. M. Alicherry and T. Lakshman, "Optimizing data access latencies in cloud systems by intelligent virtual machine placement," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 647–655.
2. H. Maziku and S. Shetty, "Network aware vm migration in cloud data centers," in *Research and Educational Experiment Workshop (GREE), 2014 Third GENI*. IEEE, 2014, pp. 25–28.
3. A. Hameed, A. Khoshkbarforoushha, R. Ranjan, P. P. Jayaraman, J. Kolodziej, P. Balaji, S. Zeadally, Q. M. Malluhi, N. Tziritas, A. Vishnu *et al.*, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," *Computing*, pp. 1–24, 2014.
4. D. Hatzopoulos, I. Koutsopoulos, G. Koutitas, and W. Van Heddeghem, "Dynamic virtual machine allocation in cloud server facility systems with renewable energy sources," in *Communications (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4217–4221.
5. S. Zhang, X. Zhang, and X. Ou, "After we knew it: empirical study and modeling of cost-effectiveness of exploiting prevalent known vulnerabilities across iaas cloud," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*. ACM, 2014, pp. 317–328.
6. S. Bugiel, S. Nürnberger, T. Pöppelmann, A.-R. Sadeghi, and T. Schneider, "Amazonia: when elasticity snaps back," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 389–400.
7. Z. Afoulki, A. Bousquet, and J. Rouzard-Cornabas, "A security-aware scheduler for virtual machines on iaas clouds," *Report 2011*, 2011.
8. M. Li, Y. Zhang, K. Bai, W. Zang, M. Yu, and X. He, "Improving cloud survivability through dependency based virtual machine placement," in *SECRYPT*, 2012, pp. 321–326.
9. E. Caron, A. D. Le, A. Lefray, and C. Toinard, "Definition of security metrics for the cloud computing and security-aware virtual machine placement algorithms," in *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference on*. IEEE, 2013, pp. 125–131.

10. S. Al-Haj, E. Al-Shaer, and H. V. Ramasamy, "Security-aware resource allocation in clouds," in *Services Computing (SCC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 400–407.
11. NVD, "US National Vulnerability Database," <https://nvd.nist.gov>.
12. NVD, "US National Vulnerability Database," <https://nvd.nist.gov>.
13. H. Holm, M. Ekstedt, and D. Andersson, "Empirical analysis of system-level vulnerability metrics through actual attacks," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, no. 6, pp. 825–837, 2012.
14. D. J. Leversage and E. James, "Estimating a system's mean time-to-compromise," *Security & Privacy, IEEE*, vol. 6, no. 1, pp. 52–60, 2008.
15. S. Frei, M. May, U. Fiedler, and B. Plattner, "Large-scale vulnerability analysis," in *Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense*. ACM, 2006, pp. 131–138.

Moving Target Defense for Distributed Systems

Shetty, S.; Yuchi, X.; Song, M.

2016, XVII, 76 p. 36 illus., 28 illus. in color., Hardcover

ISBN: 978-3-319-31031-2