Journal of Computing and Security

October 2015, Volume 2, Number 4 (pp. 293–302)

http://www.jcomsec.org

Simulating the Resource Freeing Attack Using Cloudsim Simulator

Shakiba Nayebalsadr [a], Morteza Analoui [a,*]

[a] *School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.*

ARTICLE INFO.

Article history:
Received: 25 August 2015
Revised: 10 January 2016
Accepted: 06 March 2016
Published Online: 26 August 2016

Keywords:
Security, Virtualization, Cloudlet, Resource Freeing Attack, Cloudsim

ABSTRACT

Virtualization is a technique to make the cloud computations secure. This technique can reduce the costs and increase the reliability of systems by means of sharing the physical resources between several VMs (Virtual Machines). However, virtualized environments are vulnerable to some security issues such as lack of performance isolation. These days, using the virtualization as a technique is growing slowly in large organizations. Despite these advantages, virtualizations have created several security challenges. So, the main concern of virtualization service providers and their customers is how to detect and encounter these security challenges. In this research, the resource freeing attack is selected among all available virtualization attacks because of its simplicity and its drastic effects. Resource freeing attack causes an unfair resource distribution among VMs. We use Cloudsim as a powerful cloud simulation toolkit to implement resource freeing attack. After simulating the attack, the CPU's efficiency, response time diagrams, and the obtained bandwidth are measured. The simulation results show the significant changes in VM behavior after occurring the attack in the virtualized environment. These changes help cloud providers to detect resource freeing attack in the cloud.

1 Introduction

Virtualization and security are among the most important issues in cloud computation. Thus, paying attention to the virtualization threats and attacks is necessary. The virtualization techniques enable several virtual machines to share the physical resources of the system [1]. This technology was introduced by IBM in the late sixties. In spite of several offered advantages for users, there are several security challenges for VMs using this technology [2]. Most of the security challenges existing for VMs are similar to physical machines. However, there are some security challenges and attacks which belong specifically to virtualization. One of these challenges is the resource freeing attack which is discussed in this research. This challenge has no solution yet, because of the following reasons:

- It is appeared recently.
- Its nature is mostly like the virtualization.

In this research, we introduced the resource freeing attack, the conceptual design of the attack and evaluated the performance of the attack. We simulated the attack using Cloudsim simulator and obtained the desirable results. We used the average values between 30 simulations in order to plot the corresponding diagrams. The CPU utilization and bandwidth diagrams before and after the attack were plotted and the variation of the output bandwidth over time was discussed.

* Corresponding author.
Email addresses: shnsadr@alumni.iust.ac.ir (S. Nayebalsadr), analoui@iust.ac.ir (M. Analoui)

ISSN: 2322-4460

2 Related Works

Virtualization has been one the interesting topics for the researchers during the last decade. There are many works which are related to this area. Zhang et al. (2011) described Co-Residency Detection in the Cloud environment using Side-Channel attack Analysis [3]. Side-channel attack happens when the attacker is placed on the same server as the victim. In this way, the attacker can violate the isolation of the victim VM and extract desired information. Wang et al. [4] proposed an intrusion detection system in cloud environment. Liu et al. [5] designed a scheduling system that protects against private channels in resources such as memory bus in a cloud environment. The system controlled the overlapping execution of different VMs and injected noise on the memory bus to prevent the extraction of confidential information by an adversarial user. There are numerous other researches which discussed these concepts in cloud [6–8].

A virtualized environment has a lot of variables which makes it difficult for the researchers to study its vulnerabilities. Therefore, a tool which could provide a repeatable and controllable test bed and is free of cost was valuable to the researchers. Calheiros et al.[9] studied on Cloudsim as a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. They developed the CloudSim toolkit for modeling and simulating extensible Clouds. This tool provided a desirable test bed for modeling the cloud environment. Studying the history of simulating an attack using Cloudsim makes it clear that the only attack which is simulated in the cloud environment using CloudSim has been the DDOS attack. These days, there are some defense mechanisms in order to diagnose and reduce the DDOS attack threats in cloud computations. For example, Palvinder et al.[10] developed an algorithm based on analytical procedures in order to confront the DDOS attacks over the cloud. Their results make it clear that performing the simulations using Cloudsim tool could be effective in confronting the DDOS attack.

DDOS attacks over the World Wide Web are one of the favorite topics for the computer researchers in Iran University of science and technology. DDOS attacks cannot happen easily and identifying and confronting them are challenging tasks. These features have made DDOS attacks very attractive for the hackers. Since the DDOS attacks don't have a familiar formation, usual methods cannot identify them completely and high skills are needed. Regarding the performed researches, the only simulated attack using Cloudsim simulator is the DDOS attack and there isn't any suggested procedure for performing the resource freeing attack using this simulator up to now.

In this research, we simulated the resource freeing attack using Cloudsim simulator. Generally, having two running cloudlets, while having no interference between their performances, makes the resource freeing attack happen. In the Resource Freeing Attack, VMs are placed in the same physical machines, which is the Co-Residency concept in Virtualization. Side-channel attack also belongs to the resource freeing attacks family.

2.1 Security in Virtualization

Nowadays, virtualization is known as a basic technology in the cloud computation structure and has made it possible to run a software independent of the hardware in cloud environment. Furthermore, the flexibility achieved by this technique makes it is possible to assign resources such as percentage of the active processor or active memory to the computations with respect to the user demand. So, making the virtualization secure will assure the data and service's security. Since virtualization is able to reduce the costs and increase the reliability of systems by sharing the physical resources between several VMs, its usage has grown slowly in large organizations. Despite these advantages, virtualizations have made several security challenges too. So, the main concern of virtualization service providers and their costumers is how to detect and encounter these security challenges [11, 12]. In this research the resource freeing attack is selected among all available virtualization based attacks because of its simplicity in performing and its drastic effects. We simulated the attack using Cloudsim simulator.

2.2 Threats and Vulnerabilities in the Virtual Environment

As an important technology, Virtualization has been faced with security threats before the advent of cloud computations and has transferred these threats to the cloud environment itself. In hypervisors all users know their own systems as isolated ones, even if they are feeding by the same machine. In this structure a VM is an operating system which is controlled via a control program through a sub-layer [11]. The existing threats in the virtualization environment could be listed as bellow [13].

(1) Observing the VM through the host machine
(2) Observing the VM through a different VM
(3) Back doors in VM

3 Cloudsim Simulator

Cloudsim is a simulation tool which can perform modeling, simulation, testing the cloud computing sub-

layers and services for the new users. Cloudsim tool models the behavior of system components such as data centers, virtual machines and resource preparation policies and can implement some common techniques by simplifying and limiting their tasks. Actually, Cloudsim tool supports modeling and simulating the cloud computational environment in isolated clouds or inside the network. The cloud federation and cloud users interfaces lay down some policies in order to assign the VMs in cloud computational scenarios inside the network. Many researchers in HP institutions in USA use Cloudsim tool as a cloud resource provider and an effective energy controller [9, 14]. There are a lot of classes in Cloudsim environment which are used in network, power, scheduling, etc. regarding the demands of the user. Some of the important classes which are used in simulating the attack are scheduling timeshared and scheduling space shared classes which are discussed in the next section.

When we have several cloudlets on a VM, the challenge is to schedule the CPU and bandwidth. There are two methods for scheduling as follows:

(1) Space-shared scheduling: It means that the CPU and bandwidth are shared among all VMs equally.
(2) Time-shared scheduling: The total capacity of the CPU and bandwidth is assigned to a cloudlet in a specified time interval and after that in the next time interval all these capacities are assigned to another cloudlet.

Cloudsim provides two types of scheduling for VMs: host level and VM level. For simulating the attack, we have applied the space-shared scheduling in the VM level [13].

4 Resource Freeing Attack

Resource-freeing attack (RFA) is a new type of attack in the Cloud. This attack was reported for the first time in 2012 and was related to two different virtual machines over the same physical machine. This attack occurs when a virtual machine needs more resources than the shared portion and changes the other machine's workload. In this way, hypervisor is convinced to assign a larger portion of resources to the attacker [15]. Generally, hypervisor fairly allocates the resources to VMs. If the processor is free, hypervisor allocates it to the running VM. This mode is named as work-conserving. The next mode is non-work-conserving. It happens when the hypervisor allocates a maximum limit to the VMs. The former causes the VMs to perform more efficient but be less isolated and the latter causes the VMs to run more isolated but be less efficient. Performance Isolation means the performance of a VM doesn't affect another VM's performance. The resource freeing attack happens in work-conserving mode. The first step is to increase the usage of the resource by the customer until it reaches a bottleneck. The next step is to lead the customer VM to use the bottleneck resource. In this way, other resources would be accessible to be used by the attacker.

4.1 Conceptual Scheme of the Attack

The attack consists of two parts: beneficiary and helper. The beneficiary is the process or parameter which the attacker wants to improve it. The helper is the process or program which helps the attacker to change the loading of the victim. This program could either reside in the attacker machine or in another physical machine. In this attack the beneficiary and the attacker would help each other in order to change the loading of the victim in such a way that the utilization of an unwanted resource is increased in the victim machine. In this way, the utilization of the target resource is decreased in the victim VM and could be used by the attacker.

In order to implement RFA, the attacker needs to increase the consumption of a resource in the victim machine in such a way that it becomes a bottleneck of the victim machine. In this way, it is impossible for the victim machine to utilize the target resource. In this research, target resources are: Bandwidth and CPU.

4.2 Performing the Attack

The attacker changes the victim's resource consumption pattern in two ways:

(1) Changing the victim's resource consumption pattern inside the victim machine. This case happens when the attacker wants to increase its resources by gaining the victim's portion from the resource.
(2) Changing the victim's resource consumption pattern outside the victim machine. This case happens when an outer helper helps the attacker to change the victim's portion of the resource.

5 Simulating the Attack Using Cloudsim Simulator

In order to simulate the attack in Cloudsim simulator toolkit, we defined a simulation time which was equal to 24 hours. Since we simulated the attack using simulator tool, the real elapsed time was about a few seconds.

In this research, a data-center was created in

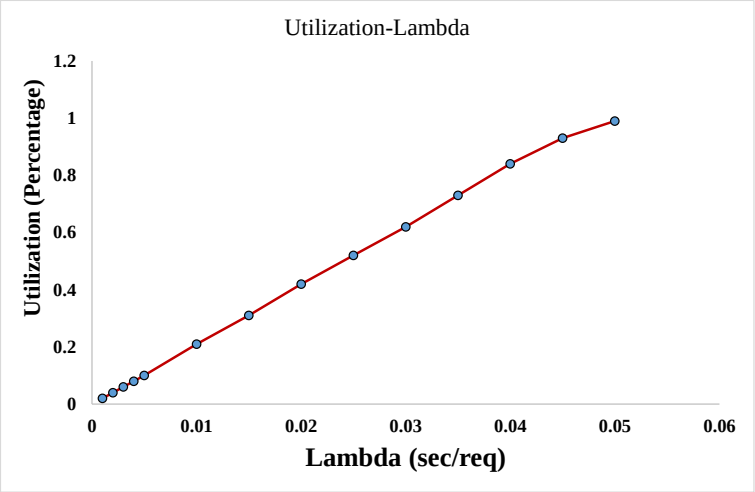

Figure 1. The diagram of CPU performance (measured) versus λ

Cloudsim including a host which contained two different VMs. The properties of these objects were defined as follows.

A. Object properties:
The host and VM properties are as follows:

- Arch = "x64";
- OS = Linux;
- VMM = "Xen";
- RAM = 64GB;
- Host with 12 core and each core 2400 MIPS;

B. The VM properties:

- RAM = 2GB;
- 1 core of CPU;
- Utilization Model of CPU and BW and RAM = Full;

We considered 1000 cloudlets to run on our 2 VMs. Cloudlets requests were sent to VMs and made the CPU busy. We didn't consider cloudlets number 1 to 100 and 900 to 1000, in order to prevent the measuring errors. In this way, our simulator had enough time to get biased. The requests were sent to the VMs with exponential distribution with the rate of 100 requests per seconds. The basic concept of the attack was that the attacker's cloudlets use the cloudlets of the victim machine. In this way, the attacker machine was reinforced and the victim machine was weakened.

The measurements were performed over the first machine firstly and after that the second machine was added.

Note that, the target resources were Bandwidth and CPU. We recognized the attack by analyzing the consumption of these two resources in each VM. An increase in the usage of each of these resources in one of the VMs and a simultaneous decrease in the same resource in the other VM show that RFA attack has been occurred.

6 Implementation & Results

The diagrams which are obtained at the end of the CPU measurements in different states are as follows. The applied data values in diagrams were based on averaging of 30 simulation measurements. The reason of averaging was the random nature of generated values. We measured the utilization in 5-minutes intervals (300 seconds). In the following diagrams the performance of the processor is plotted in terms of lambda (λ). We aimed to measure the CPU utilization and bandwidth in the whole elapsed time and evaluate their accuracy by means of the following formula.

$$R = \frac{CPU\_Utilization * \lambda}{1 - CPU\_Utilization} \quad (1)$$

where, R is the response time per second and $CPU\_Utilization$ is the Efficiency of the processor versus bandwidth in percent and λ is the rate of requests. The $CPU\_Utilization$ is obtained as follows:

$$CPU\_Utilization = S * \lambda \quad (2)$$

where, S is the total response time to the requests. Using the obtained results from the simulations, we plotted the diagrams.

Figure 1 shows the CPU performance (measured) versus different λ. In this figure the measurement is based on the portion of the service time to the total time. The x axis represents λ and the y axis is the average performance.

During the simulation, the time step between two sequential requests is measured and averaged. So, the following diagram is the performance changes

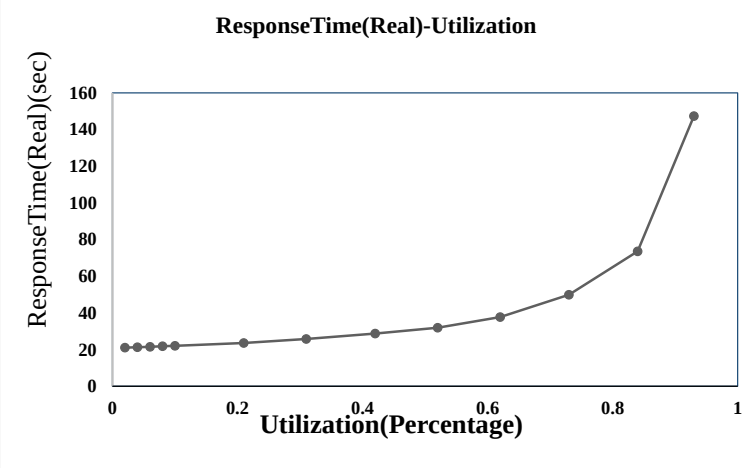

Figure 2. The diagram of response time (measured) versus CPU performance (measured).

versus the inverse of the average. The performance measure in this diagram is calculated by multiplying the service time to the inverse of the average. The x axis is the representative of 1/mean and y axis is the representative of average performance.

Figure 2 shows the response time (measured) versus CPU performance (measured).

Following are the measurements of CPU utilization and bandwidth in 5-minutes periods. This way, the intervals related to the attack could be diagnosed more precisely. These diagrams are covering the measurements both before and after the attack starts and will be discussed in the following subsection.

6.1 Diagrams of BW and CPU Variations Before and After the Attack

Using the exponential distribution and the rate of $1/\lambda$, random numbers were generated. The created exponential number was the distance between two submission times which was the time for sending the cloudlets to the VM. In other words, we considered a time step based on exponential distribution. In this way, the corresponding created cloudlets were based on the exponential distribution.

The main objective of this article is to detect the CPU bottleneck with respect to the behavior of the cloudlets. Note that, in this research we considered the cloudlet corresponding to each attack. Furthermore, it should be noted that the present research is about VMs and the cloudlets which are assigned to them. When cloudlets are running, each one occupies a portion of the CPU and when the process is finished the next cloudlet runs. In other words, cloudlets share CPU with each other. In this research, we used the spaceshared method. In this way, each cloudlet occupies the whole CPU each time and after finishing the process next cloudlet occupies the whole CPU immediately. Therefore, some cloudlets may either be in queue and wait for the processing time or be in the system and run on the processor immediately after the CPU is freed. Actually, we can say the start time will be equal to the submission time if the waiting time is zero. It means that:

$$WaitingTime + SubmitionTime = Start\,Time \quad (3)$$

Based on the provided explanations we would discuss the diagrams of the variations of output (delta throughput) versus time for each bandwidth and the processor utilization versus time before and after the resource freeing attack and would compare these two states.

In this research we used Microsoft Excel in order to plot the diagrams at first. An example of these diagrams is depicted in Figure 3 as the variations of output bandwidth before and after the attack.

As it is shown in Figure 3, there are a lot of time varying details which could not be easily extracted from this type of diagram. So, we used another type of diagrams which are named as Fast Fourier Transform (FFT) which could describe the time varying phenomenon as a function of frequencies of the changes and corresponding power densities. When trying to interpret time-sequence data from a transient solution, it is often useful to look at the data's spectral (frequency) attributes. To interpret some of time dependent data, we need to perform Fourier transform analysis. In essence, the Fourier transform enables us to take any time dependent data and resolve it into an equivalent summation of sine and cosine waves. We have used the fluent software in order to plot FFT diagrams. The x axis of the diagrams is the representative of the frequency (Hz) and the y axis is the representa-

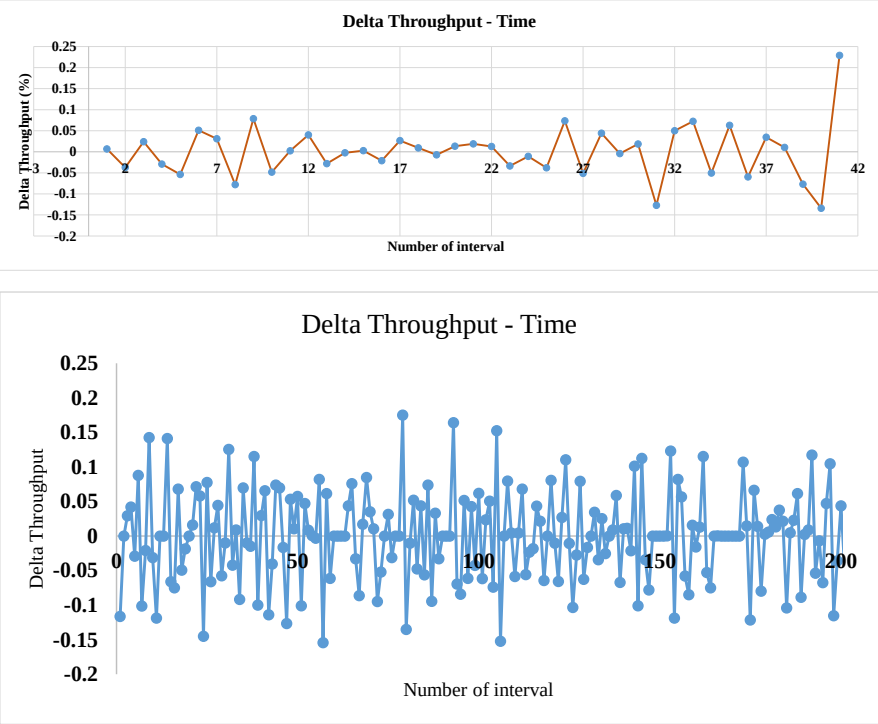

Figure 3. Variations of output bandwidth before (up) and after (down) the attack

tive of signal value squared (Power Spectral Density).

6.2 Comparing the Output Bandwidth Variations Versus Time

In this research the bandwidth throughput and CPU utilization diagrams are plotted using FFT technique. Figure 4 represents the diagram of the output bandwidth variations versus time before and after the attack. The sharp breaks in these diagrams are indicative of the overload and shortage of the load over the victim machine.

The attack scenario was in such a way that when all cloudlets were using the same percentage of the CPU and bandwidth, one of the cloudlets either gained other cloudlets portion of the CPU or the bandwidth, and the victim cloudlets were left with the remained portion of CPU or bandwidth.

Regarding the Figure 4, the frequencies equal to 0.194, 0.306, 0.389 & 0.472 Hz are representatives of the overload situations before the attack. In other words, cloudlets have the maximum value of the bandwidth at these frequencies and either have normal or shortage of the load at other points. After performing the attack, the cloudlet's bandwidth reaches its maximum loading at frequencies equal to 0.139, 0.222, 0.278, 0.361 & 0.472 Hz. According to the critical frequencies, it is seen that the frequency equal to 0.472 Hz is common before and after the attack. So, we could say this frequency is not affected by the attack. This way, it is concluded that the attack has destroyed the frequencies equal to 0.194, 0.389 & 0.306 hertz and has created the new ones equal to 0.139, 0.222, 0.278 & 0.361 hertz.

Furthermore, regarding the Figure 4, the frequencies equal to 0.0883, 0.278, 0.361, 0.444 & 0.5 Hz are representatives of the shortage of the load situations before the attack. In other words, cloudlets have the minimum value of the bandwidth at these frequencies and either have normal load or are overloaded at other points. After performing the attack, the cloudlet's bandwidth reaches its minimum loading at frequencies equal to 0.0883, 0.194, 0.25, 0.333, 0.444 & 0.5 Hz. According to the critical frequencies, it is seen that the frequencies equal to 0.0883, 0.444 & 0.5 Hz is common before and after the attack. So, we could say these frequencies are not affected by the attack. This way, it is concluded that the attack has destroyed the frequencies equal to 0.278 & 0.361 hertz and has created the new ones equal to 0.194, 0.25 & 0.333 hertz.

According to the given explanations, the attack has changed the behavioral pattern of the bandwidth of the cloudlets.

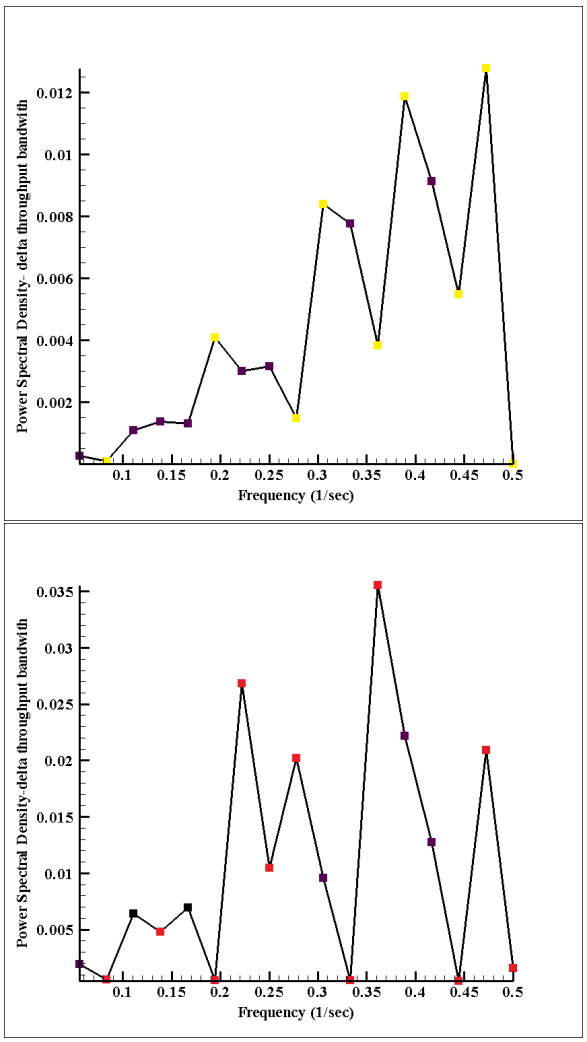

Figure 4. Variations of output bandwidth before (up) and after (down) the attack

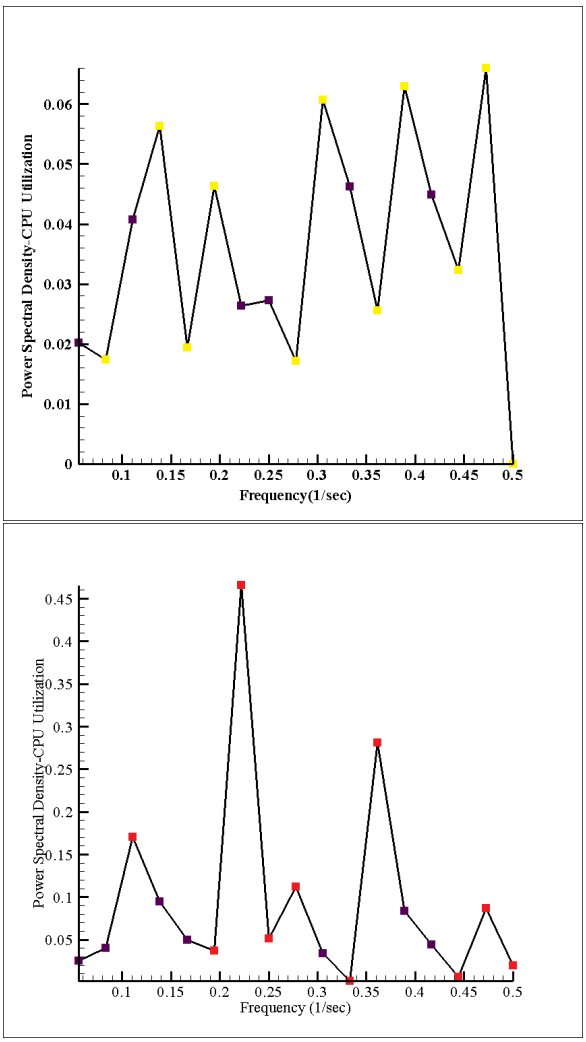

Figure 5. The diagram of CPU utilization by the victim cloudlet before (up) and after (down) the attack

6.3 Comparing the CPU Output Variations Versus Time

Figure 5 shows the CPU utilization by the victim cloudlet before and after the attack.

Regarding the Figure 5 the frequencies equal to 0.139, 0.294, 0.305, 0.388 & 0.472 Hz are representatives of the overload situations before the attack. In other words, cloudlets have the maximum value of the CPU utilization at these frequencies and either have normal or shortage of the utilization at other points. After performing the attack, the cloudlet's CPU utilization reaches its maximum loading at frequencies equal to 0.111, 0.222, 0.277, 0.361 & 0.472 Hz. According to the critical frequencies, it is seen that the frequency equal to 0.472 Hz is common before and after the attack. So, we could say this frequency is not affected by the attack. This way, it is concluded that the attack has destroyed the frequencies equal to 0.139, 0.194, 0.305 & 0.388 hertz and has created the new ones equal to 0.111, 0.222, 0.277 & 0.361 hertz.

Furthermore, regarding the Figure 5, the frequencies equal to 0.0883, 0.167, 0.277, 0.361, 0.444 & 0.5 Hz are representatives of the shortage of the load situations before the attack. In other words, cloudlets have the minimum value of the CPU utilization at these frequencies and either have normal load or are overloaded at other points. After performing the attack, the cloudlet's CPU utilization reaches its minimum loading at frequencies equal to 0.0883, 0.194, 0.25, 0.333, 0.444 & 0.5 Hz. According to the critical frequencies, it is seen that the frequencies equal to 0.0883, 0.444 & 0.5 Hz is common before and after the attack. So, we could say these frequencies are not affected by the attack. This way, it is concluded that the attack has destroyed the frequencies equal to 0.167, 0.278 & 0.361 hertz and has created the new ones equal to 0.194, 0.25 & 0.333 hertz.

According to the given explanations, the attack has

changed the behavioral pattern of the CPU utilization of the cloudlets.

The frequency in the presented diagrams is representative of the time step between dominant cloudlet's behavioral patterns. For example the frequency equal to 0.139 hertz for the overload situation before the attack, shows that the cloudlet's bandwidth utilization reached its maximum every (1/0.139) 2.12 seconds.

7 A Method for Detecting Resource Freeing Attack

In the previous section, it was shown that the resource freeing attack would not be simply detected using traffic or higher order information. In this section, we express an efficient way in order to detect the resource freeing attack using gathered information from after- and before-attack states. In this research, the algorithm presented in [16] for detecting the resource freeing attack using the virtualization environment, was simulated using Cloudsim. Note that, the extracted results were well matched to the experiments. Regarding the experiments, it seems that the time series models are useful in detecting the attack. Therefore, the next number algorithm was evaluated in this research. In order to forecast the next values of measured quantities, Equation (4) was used.

$$Y_i = (1-\alpha)Y_{i-1} + \alpha X_{i-1} \tag{4}$$

Where Y is the estimation value, X is the real value of the quantity and i is the indexer. Similarly, the variations of the variables were calculated using Equation (5).

$$\Delta X_{i+1} = (1-\beta)\Delta X_i + \beta(X_i - Y_i) \tag{5}$$

Regarding the above equations, a next number could be acceptable if it is well matched to Equation (6).

$$Y_{i+1} + \gamma \Delta X_{i+1} \leq \Delta X_{i+1} \tag{6}$$

Otherwise, the mentioned number is assumed as an abnormal one or it could be concluded that an attack was occurred. In this research, the values of α, β and γ were extracted as Table 1.

Regarding simulations and experiments, it was observed that the most appropriate values for the mentioned parameters were as follows:

$$\alpha = 0.25, \beta = 0.0625, \gamma = 2$$

Note that, decreasing γ caused a decrease in the sensitivity of the algorithm to the variations. Therefore the safe data would be considered as attacked ones which is a great mistake. If $\gamma = 2$, the possibility of the mistake would be near zero. It can be concluded that using $\alpha = 0.25, \beta = 0.0625, \gamma = 2$ the best answer will be achieved. That's because by using these values the possibility of mistake is minimum. Accepting the mistake means accepting the error in detecting the attack. The main advantage of this method is the minimum reaction of the system to sudden changes. In this way, the attack would be detected truly. On the other hand, the main disadvantage of this method is its need to frequently update in order to find the next number.

Table 1. Performance comparison of the next number algorithm for different parameters

	α	β	γ	Wrong	Accept
1	0.250242	0.250242	4	167	17
2	0.250242	0.125110	4	130	5
3	0.250242	0.062513	4	88	2
4	0.125110	0.250242	4	148	14
5	0.062513	0.250242	4	122	15
6	0.062513	0.125110	4	81	6
7	0.250242	0.062513	2	60	0

8 Conclusion

The main objectives of this research were simulating the resource freeing attack using Cloudsim simulator and interpreting the behavior of victim and attacker cloudlets. Using simulation we could study the attacker cloudlet's behavior and their interactions with other cloudlets. Based on the explanations and data before and after the attack, we could find some properties of the attacker VM as follows:

(1) Occurrence of the attack changes the consumption pattern of the victim resource.
(2) Occurrence of the attack changes the consumption pattern of at least one resource.
(3) After the attack, one of the resources cannot provide services to its user.
(4) The attack either changes the patterns of two resources or make one of the resources unavailable.

Furthermore, a method was introduced in order to detect resource freeing attack. In future researches we will attempt to present a method in order to prevent this attack and some other methods for detecting and confronting it.

Acknowledgements

We would like to thank Mr. Saber Mohammadi who helped out in many aspects of this project.

References

[1] Vmware, August 2015. URL http://www.vmware.com/.

[2] Daniel A. Menascé. Virtualization: Concepts, applications, and performance modeling. In *31th International Computer Measurement Group Conference, December 4-9, 2005, Orlando, Florida, USA, Proceedings*, pages 407–414. Computer Measurement Group, 2005. URL http://www.cmg.org/?s2member_file_download=/proceedings/2005/5189.pdf.

[3] Yinqian Zhang, Ari Juels, Alina Oprea, and Michael K. Reiter. Homealone: Co-residency detection in the cloud via side-channel analysis. In *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA*, pages 313–328. IEEE Computer Society, 2011. doi: 10.1109/SP.2011.31. URL http://dx.doi.org/10.1109/SP.2011.31.

[4] Huaibin Wang, Haiyun Zhou, and Chundong Wang. Virtual machine-based intrusion detection system framework in cloud computing environment. *JCP*, 7(10):2397–2403, 2012. doi: 10.4304/jcp.7.10.2397-2403. URL http://dx.doi.org/10.4304/jcp.7.10.2397-2403.

[5] Fei Liu, Lanfang Ren, and Hongtao Bai. Mitigating cross-vm side channel attack on multiple tenants cloud platform. *JCP*, 9(4):1005–1013, 2014. doi: 10.4304/jcp.9.4.1005-1013. URL http://dx.doi.org/10.4304/jcp.9.4.1005-1013.

[6] Fei Liu, Lanfang Ren, and Hongtao Bai. Mitigating cross-vm side channel attack on multiple tenants cloud platform. *JCP*, 9(4):1005–1013, 2014. doi: 10.4304/jcp.9.4.1005-1013. URL http://dx.doi.org/10.4304/jcp.9.4.1005-1013.

[7] Yuval Yarom and Katrina Falkner. FLUSH+RELOAD: A high resolution, low noise, L3 cache side-channel attack. In *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014.*, pages 719–732. USENIX Association, 2014. URL https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/yarom.

[8] Yinqian Zhang and Michael K. Reiter. Düppel: retrofitting commodity operating systems to mitigate cache side channels in the cloud. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 827–838. ACM, 2013. doi: 10.1145/2508859.2516741. URL http://doi.acm.org/10.1145/2508859.2516741.

[9] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw., Pract. Exper.*, 41(1):23–50, 2011. doi: 10.1002/spe.995. URL http://dx.doi.org/10.1002/spe.995.

[10] R.Kanniga Devi and S.Sujan. A survey on application of cloudsim toolkit in cloud computing. *International Journal of Innovative Research in Science,Engineering and Technology*, 3(6):13146–13153, 2014. ISSN 23198753.

[11] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Timothy L. Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles 2003, SOSP 2003, Bolton Landing, NY, USA, October 19-22, 2003*, pages 164–177. ACM, 2003. doi: 10.1145/945445.945462. URL http://doi.acm.org/10.1145/945445.945462.

[12] Z. Xiao and Y. Xiao. Security and privacy in cloud computing. *IEEE Communications Surveys Tutorials*, 15(2):843–859, Second 2013. ISSN 1553-877X. doi: 10.1109/SURV.2012.060912.00182.

[13] L. M. Kaufman. Data security in the world of cloud computing. *IEEE Security Privacy*, 7(4):61–64, July 2009. ISSN 1540-7993. doi: 10.1109/MSP.2009.87.

[14] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, April 2010. ISSN 0001-0782. doi: 10.1145/1721654.1721672. URL http://doi.acm.org/10.1145/1721654.1721672.

[15] Venkatanathan Varadarajan, Thawan Kooburat, Benjamin Farley, Thomas Ristenpart, and Michael M. Swift. Resource-freeing attacks: Improve your cloud performance (at your neighbor's expense). In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 281–292, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1651-4. doi: 10.1145/2382196.2382228. URL http://doi.acm.org/10.1145/2382196.2382228.

[16] Mojabi.R. Designing and implementing a method to detect resource freeing attack in xen virtualized environment. Master's thesis, Iran University of Science and Technology, Iran, 2015.

Shakiba Nayebalsadr received her B.S. (2013) from Khaje Nasir Toosi University of Technology in Software Engineering and M.S.C (2015) from , Iran University of Science and Technology in.currently she work's in a computer company. Her research interests are cloud computing security and vulnerabilities in cloud.

Morteza Analoui is Associate Professor in the Department of Computer Engineering at Iran University of Science & Technology, where he is also director of the Networking Laboratory. He received a B.Sc. degree in electrical engineering from Iran University of Science & Technology and a Ph.D. degree in telecommunication from Okayama University, Japan. Dr Analoui has been an Assistant Professor of Electrical and Electronic Engineering at the Okayama University, Japan, and at the Electrical & Computer Engineering Department of Tarbiat Modares University, Tehran, Iran. His research interests include modeling and performance evaluation, network protocols and architecture, network measurement, virtualization and cloud computing.