Journal homepage www.jzs.univsul.edu.iq
Journal of Zankoy Sulaimani
Part-A- (Pure and Applied Sciences)

Innovative Cloud-based Snooping Detection model in both Public and Private Infrastructure

Mustafa Ibrahim Khaleel[1]

1College of Science – Computer Department. University of Sulaimani, Sulaimani, Iraq

E-mail: mustafa.khaleel@univsul.edu.iq

Article info

Original: 30 March 2018
Revised: 1 June 2018
Accepted: 11 June 2018
Published online: 20 June 2018

Key Words:
Inter-VM traffic
RFMON mode
PrCN Network
Cloud Schedular

Abstract

Network access technologies, including Wi-Fi and 4G LTE, are becoming more and more popular in Cloud computing infrastructure due to their increased performance, reliabilities, and ease of development. Cloud consumers can collect and process real time and continuous sets of massive assets. The large data size and security concerns have resulted in an ever-increasing need for efficient paradigm concept to integrate the functionalities of data monitor, analysis and anomalous traffic behavior detection. The procedure of intercepting traffics assigned by Cloud consumers and passing through Cloud scheduler to the Cloud infrastructure data centers has been known as wireless packet sniffer. This could capture the entire packets and analyze the contents in both Private Cloud Network (PrCN) and Public Cloud Network (PuCN) in the RFMON (Radio Frequency MONitor) mode. After buffering the entire Cloud consumer's images in the Cloud scheduler, further interpretation of the packets can be carried out to distinguish malicious from beneficial packets. We designed and developed an intrusion detection model, namely Cloud Snooping Disclosure (CSD) to monitor the Cloud consumer's image traffic loads, detect the anomalous traffic behaviors, and block the malicious intrusion. Our heuristic is based on two major steps, Forward and Backward scanning process. The step includes the initialization process and installing the security parameters for both sides, Cloud users and Cloud scheduler, while the second one relates to capturing anomalous inter-VM traffics. Furthermore, our algorithm incorporates *pcap* library into Cloud scheduler so that any incongruous traffic behaviors can be reported and saved. Our system was inspired by some existing researches that applied sniffer software such as Ethereal, Tcpdump, and Snort. The simulation results indicate that the effectiveness of our heuristic had the ability to detect and eliminate approximately 107 anomalous traffic behaviors from five case trials that have been generated by CloudSim framework.

Introduction

Today, Cloud-based networking has become very complex to meet various application needs and performance requirements. As a result, troubleshooting and maintaining a reliable network connection among Cloud service models has become cumbersome which calls for some novel specialized monitoring and analysis tools. Such software allows cloud providers to evaluate and examine the data stream constantly flowing through the network by keeping track of network metrics and identifying certain faults. In addition, Cloud network Sniffers can help identifying network attacks and detect security threats; they can be used in intrusion detection, load balancing, and packet error correction. Furthermore, they can be used to help understand packets' structure and traffic patterns generated by common network applications. Cloud network monitoring can also be used to evaluate protocol performance and assist in protocol development in each of Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) models [1,2]. Despite their usefulness, sniffers in Cloud network infrastructure may be harmful when used by intruders. With network sniffers, intruders can intercept consumer's images, steal and intentionally modify

information from targeted networks. A secure network monitoring for a corporate network is a critical IT function that can improve network performance, enhance productivity and infrastructure cost overruns [3]. It can also find and help resolve snail-paced webpage downloads, lost-in-space e-mail, questionable user activity and file delivery caused by overloaded, crashed servers, dicey network connections or other devices. As a matter of fact, monitoring cloud network can be achieved by using various software or a combination of plug-and-play hardware and software appliances. Deciding what to monitor on your network is affected by network topology map which should be accurate and up-to-date. The network topology map includes information such as number and types of servers, applications, and O.S. etc. [4].

In this paper, we present a paradigm that can be deployed in both public and private Cloud infrastructure system for monitoring and security purpose. Some architecture questions need to be considered: how Cloud consumer's images are assigned over Cloud scheduler? And how the analyzation process for these image's content will be established? We propose the implementation of a model, namely Cloud Snooping Disclosure (CSD), running under both platforms Microsoft and Linux systems. This paradigm is based on two algorithms. The first one, Forward Scanning, captures the entire data link layer frames passing through the Cloud scheduler to the Cloud infrastructure passively while the second algorithm, Backward Scanning, disclosures the anomalous inter-VMs traffic behaviors. Figure (1) illustrates our Cloud-based network system architecture while figure (2) depicts Cloud consumer's image flow from application layer to the network interface card.

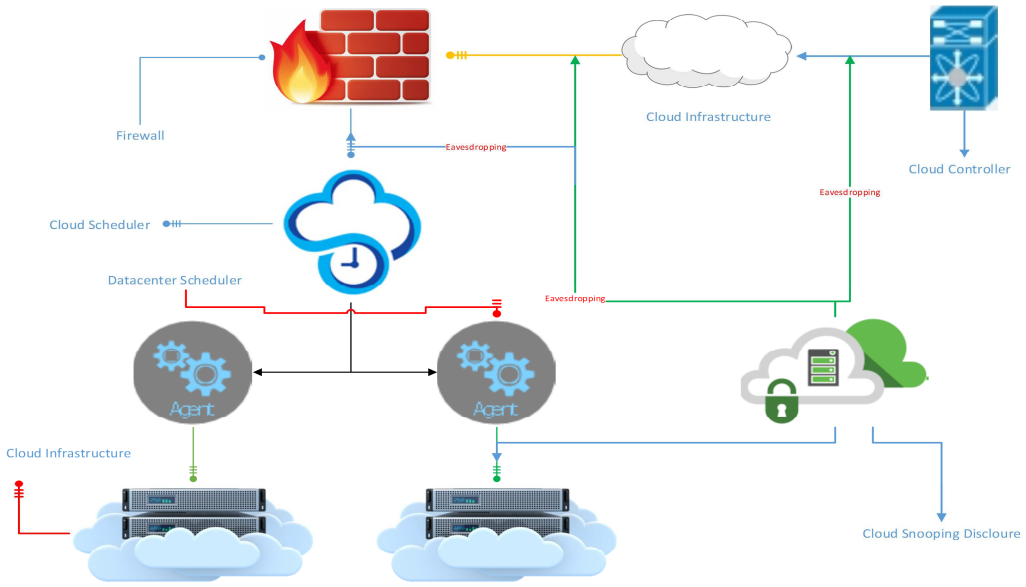


Figure (1): Cloud-Based Network System Architecture

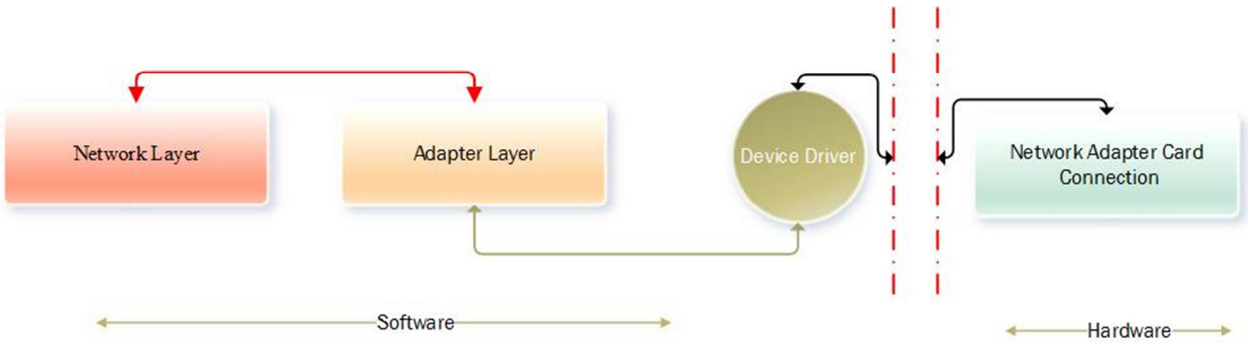


Figure (2): Cloud Images Flow

Related Works

Nowadays, tracking Cloud module applications is playing an ever-increased important role for making the Cloud consumer's assignments more secure. These challenges are motivating researches alike to formulate new approaches and build innovative paradigms. As a result, developing a new model that can behave as a helpful filter that allows network administrator to monitor the systems and detect the anomalous network behaviors through capturing and analyzing the packet's content that have been scheduled over Cloud resources. For decades, several researches have been devoted to optimizing Cloud security protocols with the objective of decreasing the incongruous network traffic behaviors. Chen et all. [5] proposed an anomalous detection model based on Cloud system. Their intention was to develop a solution to redirect the traffic using Software-Defined Networks (SDN). Their heuristic had the ability to capture inter-VM traffic, detect both known and unknown anomalous network behaviors, adopt hybrid techniques to analyze VM network behaviors, and control network systems. McKeown et al. [6] developed the control unit which redirects the separation architecture of OpenFlow. This technique is based on an Ethernet internal flow-table switch. By operating this parading, researchers can evaluate their experimental over heterogenous switches in a uniform way at line-rate with high port-density. Grobauer et al. [7] defined four main indicators related to the Cloud computing vulnerabilities. The authors well-explained the fact that how Cloud computing influences each risk that may affect Cloud consumer's images. Recently, many researches have discussed the network traffics and intrusion detections in Cloud infrastructure. The one such as [8], programmed an algorithm to detect intrusion issues under Linux operating system. This also including the traffic bottleneck analyzing issue using packet sniffer. Oktay and Sahingoz [9] generated different attack types that affects each of availability, confidentiality and integrity of Cloud resources. They also came up with a model to prevent such behavior using intrusion detection model. Xing et al. [10] investigated SnortFlow heuristic. This model based on both the OpenFlow and Snort based IPS. They applied this method to expose and detect intrusions which made easy for them to establish countermeasures through reconfiguring Cloud network model on-the-fly. Zissis and Lekkas [11] introduced a Trusted Third Party that relates to Cloud security characteristics. Their solution model based on cryptography where both SSO and LDAP are applied based on Public Key Infrastructure. This step was to ensure the integrity, authentication, and confidentiality of assigned Cloud module applications.

Cloud System Flow Chart and Analytical Models

A. Cloud System Flowchart

Cloud Snooping Disclosure heuristic initiates with two essential procedure functions: the first one used for emulating the available addresses inside the system, and the second one is for defining the mapping ports between the Cloud consumers and the execution-based units (VMs) in Cloud infrastructure through the Cloud meta-mapper. Depending on IOCTL driver, Cloud Snooping Disclosure model gets the assigned applications into scheduler's buffer and arrange them in decreasing deadline's order to be dispatched over Cloud resources. This will be accomplished after checking the integrity of each packet. Flow Charts (1-a) and (1-b) explain both the basic Cloud Snooping Disclosure flow and system monitor flow.

B. Analytical Models

Inspired by previous work [9], we can define three cases for both VM-based execution units and infrastructure connections anomalous detection behaviors.

Definition Case (Normal Equilibrium-Type 0): vulnerabilities are available, but they have not been exploited yet by intruders as illustrates in figure (3) case 0.

Definition Case (Anomalous Equilibrium-Type 1): vulnerabilities are exploited but have not yet been breached as depicts in figure (3) case 1.

Definition Case (Anomalous Equilibrium-Type 2): Cloud infrastructure are exploited, and the system is continuous cyber-attacked as explains in figure (3) case 2.

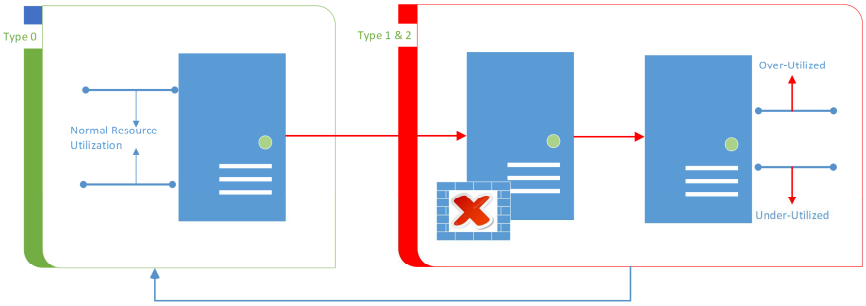


Figure (3): Case 0-2 Normal and Anomalous behavior

Problem Formulation

Assuming that the Cloud system infrastructure deviates from type-0 [Normal Equilibrium] to either type-1 or type-2 [Anomalous Equilibriums], it is our paradigm's objective to analyze the entire passing data link layer frames and block the malicious intrusion so that the system returns to its normal equilibrium operation.

$$\underset{\forall passing\ images}{analyze}\ (\sum_{i=1}^{con} \mathcal{DL}(\mathcal{F}_i)) \quad \text{----------} \quad (1)$$

$$Subject \begin{cases} inf. \in Type-0 \\ inf. \notin Type-1\&2 \end{cases}$$

Where *con* is the inter connections between Cloud consumers and Cloud infrastructure while *DL(F)* is the data link frames that are assigned through Cloud scheduler to be executed over Cloud-based execution units (VMs). *Inf* is Cloud infrastructure hardware.

Model Methodology

Anomalous behavior detection heuristic in Cloud infrastructure designate of hybrid modules which detect the VM's behavior and decide what type a VM belongs.

A. Cloud Snooping Disclosure model

One of the major objectives of Cloud Snooping Disclosure model is to dissect module applications assigned by Cloud consumers in real-time matter and redirect the anomalous traffic behaviors into scheduler's buffer. It is the kernel's system duty to pass the entire received traffics to the central processing units in Cloud data centers rather than just those packets addressed to it, which is a feature normally used for packet sniffing. Intrusion Prevention System (IPS) has been conjoined with an OpenFlow protocol to be consolidated with our algorithm model. This allow us to simulate a flow table to deploy new protocols without changing any networking devices and detect any malicious behaviors such as sending responses to packets even though they are addressed to other machines. The mathematical model for Cloud Snooping Disclosure and behavior detection are given in equation (2) and (3).

$$wrfmon = \int_{t_0}^{t_1} \mathcal{B}(\mathcal{N}_\mathcal{J} + \mathcal{A}_\mathcal{J})\, dt \quad \text{----------} \quad (2)$$

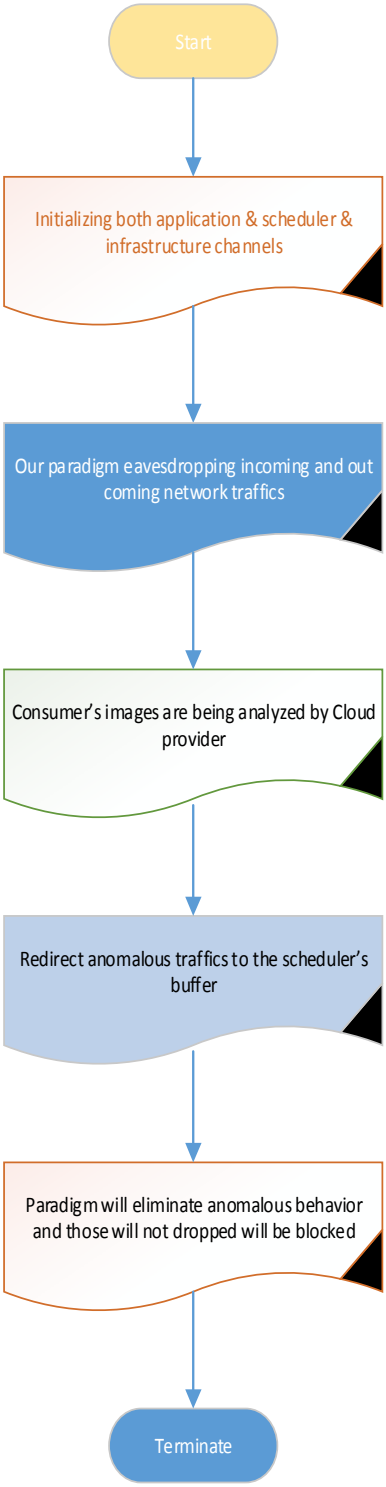
$$\mathcal{B}(\mathcal{N}_\mathcal{J} + \mathcal{A}_\mathcal{J}) = \sum_{j=1}^{\mathcal{VM}} \mathcal{VM}_j + \mathcal{CON}_{\mathcal{J},\mathcal{J}++} \quad \text{----------} \quad (3)$$

Where N_J and A_J are normal and anomalous behavior of virtual machine J during period of time while $CON_{J, J+1}$ is the connection detection between the source and destination Cloud resources including the entire virtual machines that are fully operated for execution Cloud consumer's assignments.

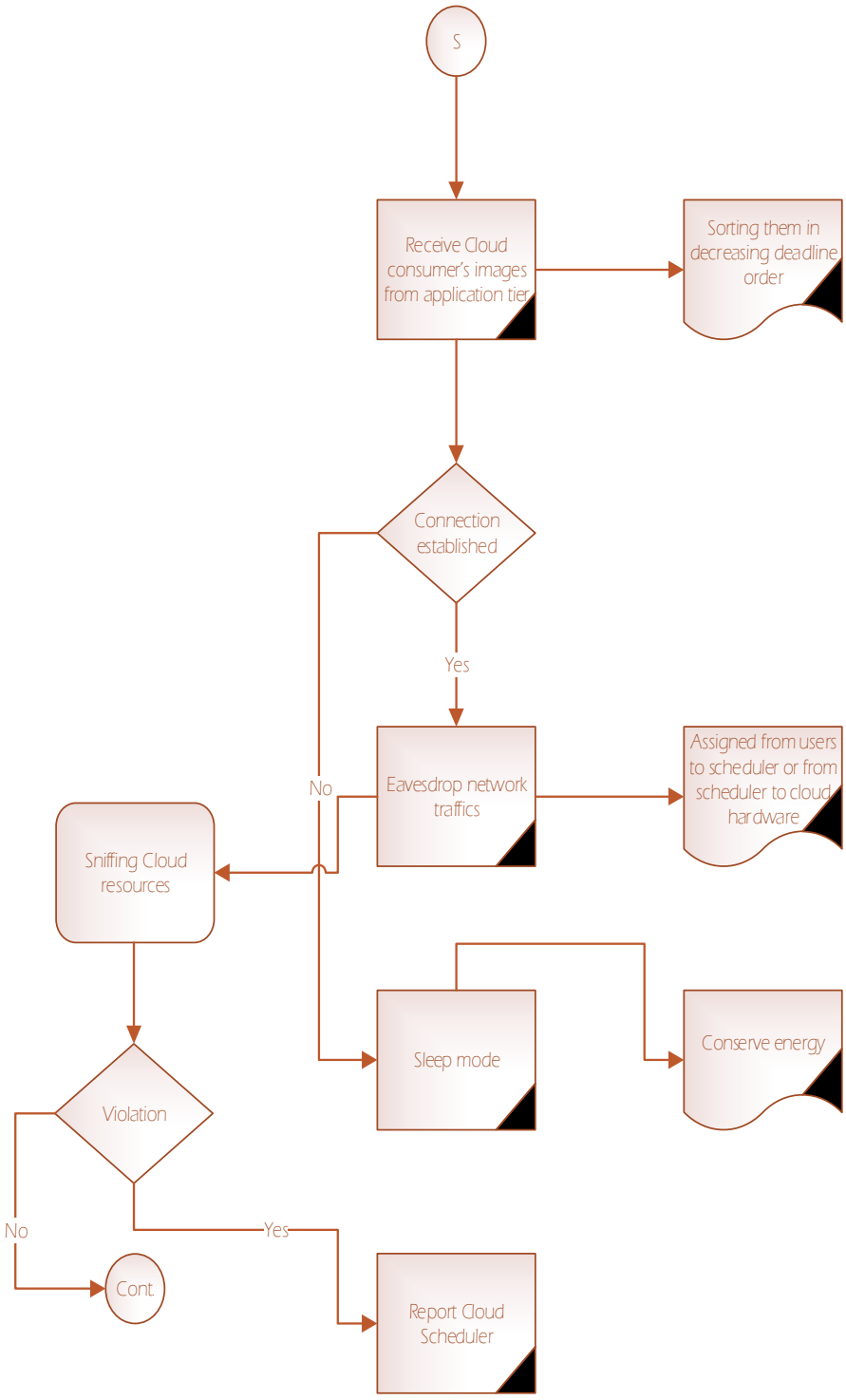
B. LIBPCAP Library model

Pcap is an open source library for capturing images and network analysis. It includes a kernel-level packet filter, a low-level dynamic link library (packet.dll), and a high-level system-independent library (pcap.dll). Most Cloud applications access the network through widely used operating system primitives such as

sockets. It is easy to access data on the network within this approach since the operating system directly communicates with the low-level system (protocol handling, packet reassembly, etc.) through an easy to use interface. The pcap library model can be used by many types of network tools for analysis, troubleshooting, security and monitoring [7]. Figure (4) depicts the architecture of pcap library components. Particularly, some existing classical tools that incorporate pcap library model in their kernels include protocol analyzer, network monitors, traffic loggers, network intrusion detection systems (NIDS) and security tool.



Flow chart (1-a): Basic CSD Flow Chart



Flow chart (1-b): System Monitor Flow

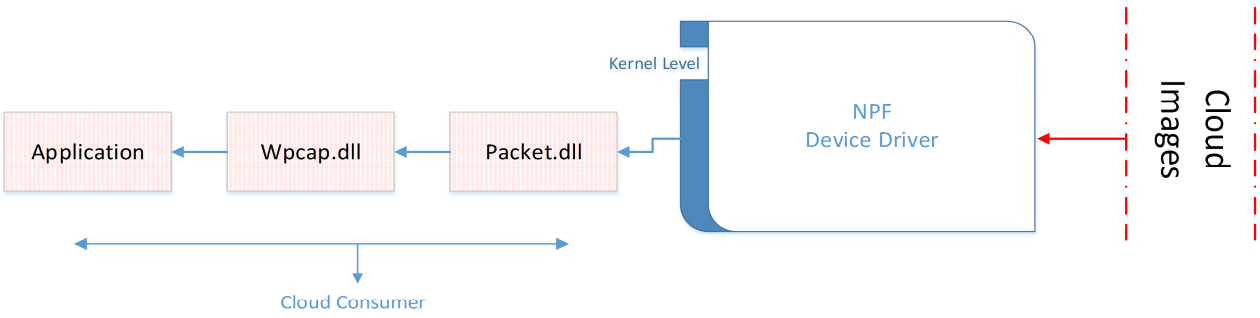


Figure (4): Pcap Library Component

C. Traffic Filtration model

Filtering Platform (FP) is another pattern that integrated into our CSD heuristic with the objective function of increasing the reliability of the Cloud infrastructure servers. This platform comprises set of services and application interfaces that allows module applications to perform packet processing and filtration pipeline. However, it includes features that can be configured for invoking processing logic on a per-application basis. It is commonly used by firewalls and other packet-processing or connection monitoring components. Shims, filter engine, and callout are the main missions of filtration platform model. The mathematical formula for the traffic filtrating function is given in equation (4). Where $pass_1$ and $pass_i$ are the initial and descending trials of the consumer's image filtration process. TC_i is the total connections to reach module application (i).

$$pass_1(image_i) = \frac{\sum_{i=1}^{CON}(image)_i}{TC_i} \quad \text{----------} \quad (4)$$

$$pass_2(image_i) = \left(\sum_{i=1}^{ant}(image)_i\right) \Big/ \left(\sum_{i=1}^{CON}(image)_i\right) \quad \text{----------} \quad (5)$$

- **Shims**: The structure of Cloud image's properties is being disclosed with this filter which provides various types of filtration for protocols at different layers. For instance, Application Layer Enforcement (ALE) is a type of shim that can be established for Cloud consumers as in-built service. However, users can apply different shim filter's platform for different protocols. Other shims include Network Layer Module (NLM) shim, RPC Runtime shim (which resolves the runtime incompatibility), Internet Control Message Protocol (ICMP) shim and Stream shim.
- **Filter Engine**: The packet filtration can be done using filter engine by verifying the data against the specified set of rules. It incorporates both modes (kernel and consumer) to extend the basilar of filtration capabilities. The cloud module applications that have been exposed by the shims will be matched against the aforementioned filtering rules. Then, based on the result, it decides whether the packet is admissible or negligible. For further action necessities, a *callout* function model will be requisitioned.
- **Callout:** As the filtration rules for the protocols are being registered, the Cloud provider will specify the callout function. This will allow the system to apply further filtration process other than just default block/allow states. The callout function will be invoked when the return result is matched which handles the filtering operation.

D. Cloud Scheduler Buffer model

Cloud meta-mapper and Cloud local schedulers incorporate pcap_setbuff function to resize the buffer array for the kernel ring through mini port network driver to receive consumer's images. This allows the administrator to read the operations periodically and drain Cloud image packets out of the ring buffer up to user mode [9]. However, pcap_mintocopy is another module applied to our CSD heuristic to control the signaling mechanism. Basically, the user mode code does not get notified immediately when a packet is

available in the kernel ring buffer. The user mode code gets signal of available packets when at least "mintocopy" bytes are available in the kernel buffer. This mechanism is used to avoid transferring one single packet from kernel mode to user mode at a time.

Performance Evaluation

For this experiment, we have used Open Source Java-based CloudSim toolkit to simulate a cloud infrastructure that comprises 90 heterogeneous computing nodes. Similar to previous work [9]. We have incorporated KDD Cup-99 dataset [12], Intrusion Detection System (IDS) Evaluation dataset, to setup network intrusion detection. It is the intrusion detection system's incipient to concentrate on discovering the identity for each packet sender, information exchanged, and threats posed on documents. This allow us to distinguish "anomalous connection behaviors" from "trusted connection behaviors". However, five different trial cases have been simulated using CloudSim framework to intense our outcome results. Each case has different workloads and interconnection edges as shown in table [1]. To make our view more precise, we define a default threshold for the entire cases. Surpassing this value will redirect our paradigm to become unsettle. As it is not practical to draw and illustrate the total of 17 anomalous traffic behaviors, a fraction of the results with the full coverage of the scenario has been plotted in figures [5-7]. Figure (5-a) depicts the normal network traffic which is the initialization assignment of Cloud consumer's images. According to the plot, the system has normal range workloads between 78 and 115. On the other hand, we generated anomalous network attack with probability rate of 15% in figure (5-b). This increased the normal traffic rate about 7% of the total system value. For this one, the range manipulate between 85 and 123. Also, we can observe that the curve is heading to become closer to the system threshold. Since our prime objective of Cloud Snooping Disclosure model is to detect anomalous network traffics as many as possible, we increased the probability rate of the incongruous network traffics by (35, 60, 75%) as shown in figures (6) and (7). The sequential values range between 136 and 186. However, figure (6-a) illustrates that even within probability rate of 35%, the anomalous network traffics reached the system threshold. For both figures (6-b) and (7-a), the anomalous traffics surpass the threshold vale which makes the system unstable and invade the most part of the bandwidth. To get back the system to the normal behavior, we applied our algorithm to detect and eliminate these anomalous traffics as explained in figure (7-b). within our paradigm, the entire probability rates returned under the threshold value. As mentioned before, we covered only five trial cases. Our model had the ability to disclose five incongruous network traffics in the initialization case and 41 incongruous network traffics in the last case. Moreover, the heuristic also eliminated approximately 107 anomalous behavior attacks when Cloud consumer's images are assigned over both Cloud scheduler and Cloud infrastructure data centers.

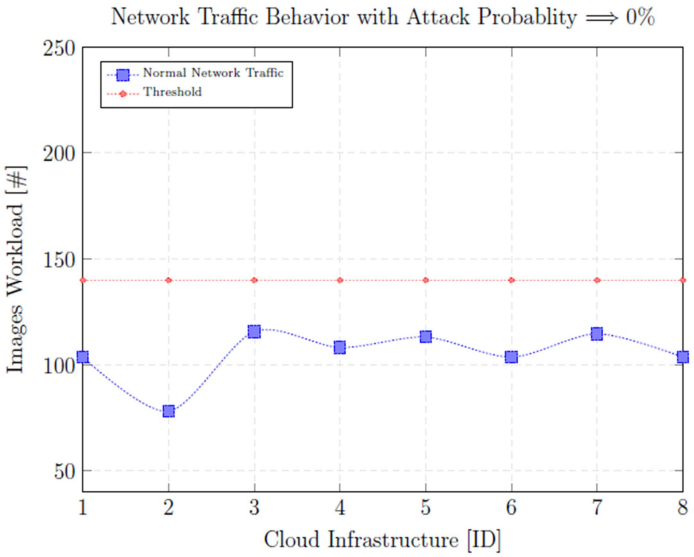


Figure (5-a): Probability Attack = 0%

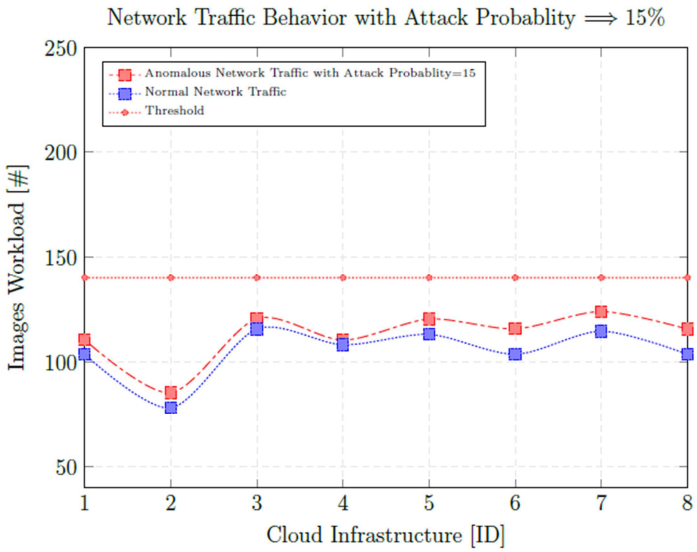


Figure (5-b): Probability Attack = 15%

Table (1): Five Case Trails Anomalous Detection Results

Table 1: Anomalous Detection Applied for Five Trials			
Number of Case Trials	Consumer Image Workloads	Interconnection Edge $\|E\|$	Anomalous Detection $\|N\|$
Trial 1	50	85	0
Trial 2	65	130	5
Trial 3	75	156	23
Trial 4	85	167	36
Trial 5	120	230	41

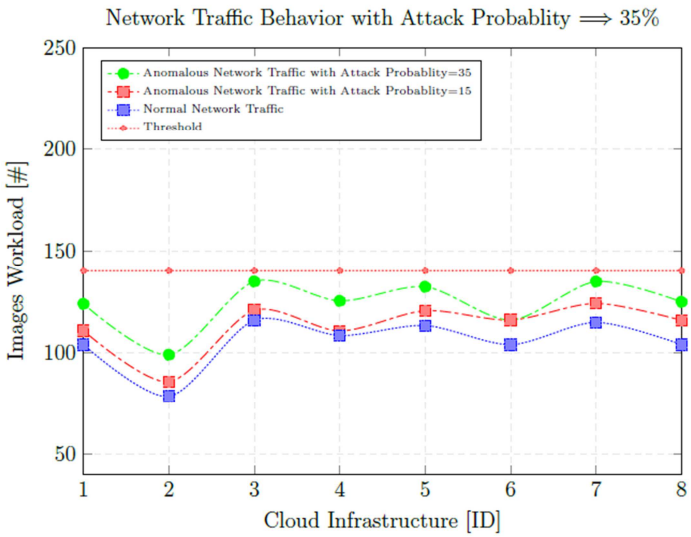


Figure (6-a): Probability Attack = 35%

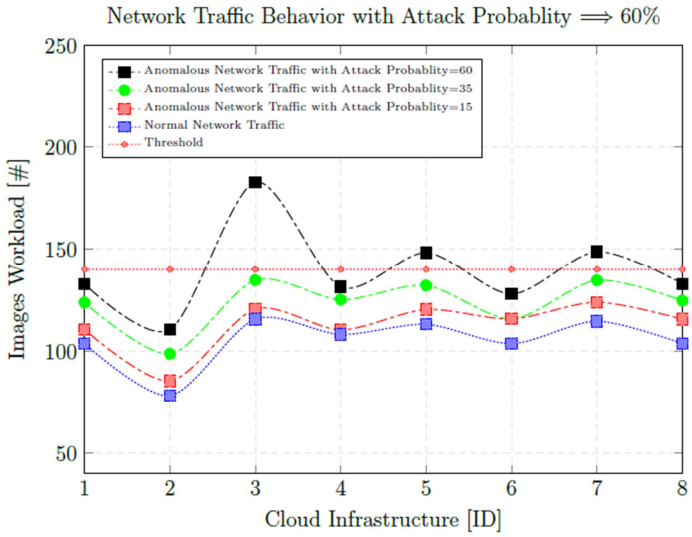


Figure (6-b): Probability Attack = 60%

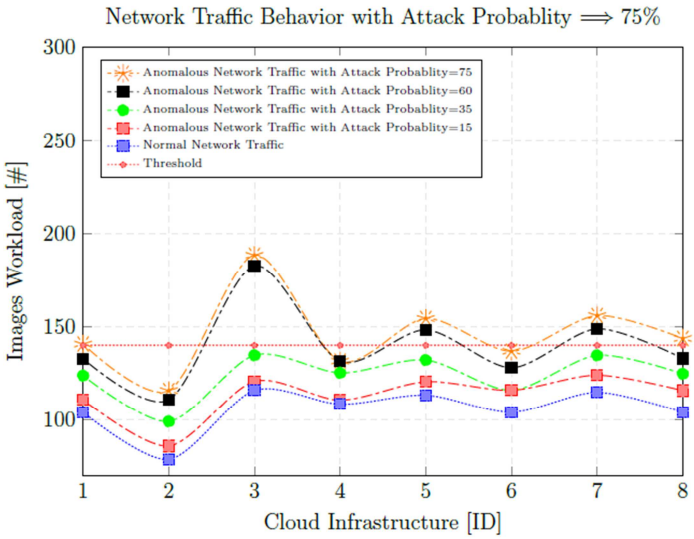


Figure (7-a): Probability Attack = 75%

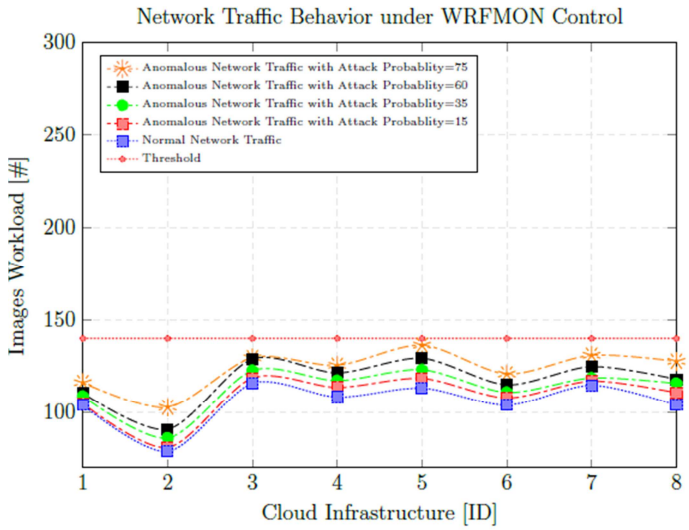


Figure (7-b): Recovering Attack Process

Conclusion

Establishing a secure Cloud computing infrastructure challenges engineers and managers to seek for near optimal solutions due to the fact that cloud environment is a hybrid of both physical and virtual overlaid networks. It is not uncommon that both the Cloud provider and Cloud consumer desire to satisfy their necessities in terms of protecting preinstalled VMs and prescheduled module applications against any eccentric behaviors. Consequently, an anomalous traffic behavior detection model has been formulated in this experiment to evaluate the performance of both assigned Cloud images and inter-VM traffics. This paradigm is programmed based on two main levels. The first one captures and analyzes the consumer's application content to detect any attached malicious behaviors so that it can initialize the interconnection between Cloud application tier and Cloud scheduler tier. Our heuristic's main duty is to observe any monetized attempts through incongruous network traffics. The last and perhaps the most significant step is to monitor the Cloud execution resources and report any anomalous behaviors to the Cloud scheduler to cope with (either eliminating or obstructing them). This will give the Cloud providers a tremendous benefit to satisfy each of confidentiality, integrity, and availability of the system. The evaluation results demonstrate that our algorithm had the ability of satisfying the three aforementioned factors after reaching the probability rate of anomalous traffic behaviors to 75%.

References

[1] Liviu Ciovica, Marian P. Cristescu, and Lucian A. Fratila, *"Cloud Based Business Processes Orchestration",* 21st International Economic Conference IECS, Sibiu, Romania, pp.592-596, (2014).

[2] Karandeep Kaur, *"A Review of Cloud Computing Service Models",* International Journal of Computer Applications, Vol. 140, No. 7. (2016).

[3] Victor A. Clincy and Nael Abu-Halaweh, *"A Taxonomy of Free Network Sniffers for Teaching and Research",* Journal of Computing Sciences in Colleges, USA, pp. 64-75. (2005).

[4] H. Wang and Y. Chen, *"Network topology description and visualization",* 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), China, (2010).

[5] X. Ye, X. Chen, H. Wang, X. Zeng, G. Shao, X. Yin and C. Xu, *"An anomalous behavior detection model in cloud computing",* Tsinghua Science and Technology, China, pp. 322-332. (2016).

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner*," OpenFlow: Enabling innovation in campus networks",* Computer Communication Review, Turkey, pp. 69–74. (2008).

[7] B. Grobauer, T. Walloschek, and E. Stocker, *"Understanding cloud computing vulnerabilities",* IEEE Security & Privacy, pp. 50–57. (2011).

[8] M. A. Qadeer, M. Zahid, A. Iqbal, and M. Siddiqui, *"Network Traffic Analysis and Intrusion Detection using Packet Sniffer"*, 2nd International Conference on Communication Software and Networks, (2010).

[9] U. Oktay and O. K. Sahingoz, *"Attack types and intrusion detection systems in cloud computing",* 6th International Information Security & Cryptology Conference, pp. 71–76. (2013).

[10] T. Xing, D. Huang, L. Xu, C. J. Chung, and P. Khatkar, *"Snortflow: A openflow-based intrusion prevention system in cloud environment"*, Research and Educational Experiment Workshop (GREE), Second GENI, pp. 89–92. (2013).

[11] Dimitrios Zissis and Dimitrios Lekkas, *"Addressing cloud computing security issues",* Future Generation Computer Systems, Netherlands, pp. 583–592. (2012).

[12] Pavan Kaur and Dinesh Kumar, *"A Study on Intrusion Detection based on KDDCUP'99 Benchmark Dataset"*, International Journal of Engineering Research and Management (IJERM), Vol. 2, No.5. (2015).