ENERGY EFFICIENT INTRUSION DETECTION FOR NFV CLOUD INFRASTRUCTURES USING MACHINE LEARNING TECHNOLOGY

[1]**Nageswara Rao Rayapati,** Assistant Professor, Dept. of C.S.E, ACE Engineering College, Ankushapur, Ghatkesar, Hyderabad**.**
[2]**Mukkamala Anantha Lakshmi,** Lecturer in Computer Science, Sri Majety Guravaiah Degree College, Guntur.

ABSTRACT: Recent times have seen a steady shift of technology from traditional software models to the cloud. Widespread deployments of Network Function Virtualization (NFV) technology will replace many physical appliances in telecommunication networks with software executed on cloud platforms. Setting compute servers continuously to high-performance operating modes is a common NFV approach for achieving predictable operations. Intrusion detection systems are one of the most suitable security solutions for protecting cloud based environments. The Dynamic Voltage-Frequency Scaling (DVFS) technology available in Intel processors is a known option for adapting the power consumption to the workload, but it is not optimized for network traffic processing workloads. This paper proposes energy efficient intrusion detection for NFV Cloud infrastructures using Machine learning technology. Attack presence is confirmed by performing Machine learning technique with belief propagation. Performance measures such as Detection Rate and Accuracy are used to evaluate the performance of the approach. From results obtained accuracy is 94.11% and achieved detection rate as 70 sec. transmission rate of proposed method is better than compared to setting the maximum frequency or using the ondemand governor.
KEYWORDS: Network Function Virtualization, Intrusion detection, Machine learning, Detection Rate and Accuracy.

I. INTRODUCTION

Cloud computing is the environment which provides people platform to share resource, service, and information [1]. It provides companies with a flexible architecture which in turn provides an efficient framework for computing. Due to the adoption of the cloud computing environment by a large number of organizations, there are a lot of risks and challenges which have come with it. Cloud computing servers as a utility available on the internet so the issues like user privacy, data leakage, and authentication remain one of the biggest challenges to cloud computing environment.

Network Function Virtualization (NFV) will enable economies of scale by replacing countless physical appliances with software running on distributed cloud platforms [2]. Central Office locations and base station points-ofpresence in telecommunication networks are in the same class of sites that, when housing small distributed data centers, were found to consume 95% of the total data center power consumption in the USA. As they are equipped with commodity servers for NFV edge clouds, every Watt that could be saved matters tremendously [3].

Central Processing Units (CPUs) are the major source of power consumption in servers, in particular for telecommunications-oriented equipment. Recent high-end server-grade CPUs have as many as 24 cores and a data sheet nominal power consumption of over 450 Watts. The Dynamic Voltage and Frequency Scaling (DVFS) capability of the CPU is widely used in enterprise environments for meeting performance requirements while reducing the power consumption. The CPU capabilities are complemented at the operating system level, for example through software in the Linux operating system known as DVFS governors.
Intrusion Detection Systems (IDS) are used for detecting cyber attacks on virtualized environments [4]. Several approaches exist for intrusion detection systems which can be broadly classified as Malware detection techniques and security analytics. While malware detection

assures high accuracy levels, it is vulnerable to zero-day attacks. Zero day attacks are those that have been live for less than a day. Security analytics involves analyzing network and application logs. While zero-day attacks can be detected by analytic methods, it cannot detect attacks in real time.

Machine learning techniques along with big data analytics can be used for real time detection of attacks in virtualized environments. Big data analytics on network and application logs can be used to identify potential attack paths [5]. The possibility of attacks can be confirmed by using machine learning techniques on collected log features to classify the paths as malicious or benign. The problem of attack detection using machine-learning techniques is not new to literature. While signature detection techniques can detect attacks based on signatures of already learnt attacks, anomaly detection techniques learn network traffic from a baseline profile and detect anomalies as ones that deviate significantly from the baseline profile. Signature detection techniques are effective against known attacks while anomaly detection has the ability to detect unknown and new attacks (zero-day).

II. LITERATURE SURVEY

Omar et.al proposes a model that specializes on detecting botnet intrusions [6]. A randomized data partitioning learning model is employed for botnet intrusion detection. The approach involves four consecutive stages- obtaining a dataset for training, developing an efficient and scalable machine learning algorithm that can deal with large scale network traffic, data reduction to reduce the number of data samples and finally, a learning model of multiple randomized trees. The Information Security and Object Technology (ISOT) dataset was used for training. Features that describe the network traffic characteristics such as the Source and destination ports, payload length, packet size etc were used for training.

Rossi et al. [7] verified that when threads are fairly distributed through the cores of a CPU, DVFS could provide a significant reduction in the energy consumption. However, while being able to save only about 10% energy, stock DVFS policies implemented in Linux increased response time of a web server by about 70%.

Controlling DVFS in response to network traffic was presented by Kuan et al. in [8]. They proposed a system that obtained packet inter-arrival rates and used the information in a DVFS governor. A similar solution was proposed and characterized in a virtual router environment. A model of the network congestion was used for predicting whether it would be possible to reduce the CPU frequency while maintaining an acceptable delay.

Jie-Hao et al. [9] used Artificial Neural Networks(ANN) to detect DDoS attacks where they compared the detection outcome with decision tree, ANN, entropy and Bayesian. The authors identified users requests to a specific resource and their communicative data. Then samples of such requests are sent to the detection systems to be judged for abnormalities. Detection accuracy can be improved by combining SVM with other techniques. Li et al. [10], designed an intelligent module for network intrusion prevention system with combination of SNORT and configurable firewall. The SVM classifier is also used with SNORT to reduce false alarm rate and improve accuracy of Intrusion Prevention System (IPS).

III. INTRUSION DETECTION FOR NFV CLOUD INFRASTRUCTURES

The framework of energy efficient intrusion detection for NFV Cloud infrastructures using Machine learning technology is represented in below Fig. 1.

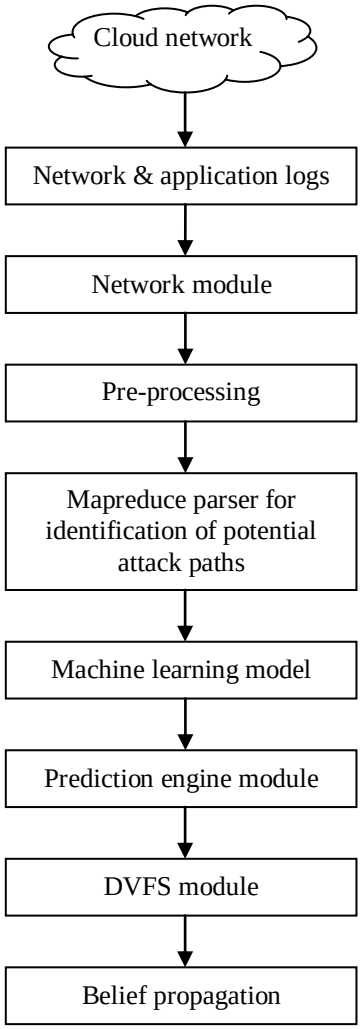

Fig. 1: INTRUSION DETECTION FOR NFV CLOUD INFRASTRUCTURE ARCHITECTURE

The Amazon Web Services (AWS) public cloud is used as the cloud environment. The AWS Elastic Compute Cloud (EC2) service is used to create virtual machines (instances, as they are identified in AWS). The first step in the approach is to collect attack features. TShark is used to obtain the network logs containing the traffic flows of the guest VMs. Source and destination IPs along with port numbers are collected.

The Network Module was inserted in the network path between the network interface card and the NFV. It was responsible for extracting features in real-time by examining the packets that were in transit towards the NFV. The Network Module was implemented in C using the Pcap library [25]. We are aware of the significant overhead introduced by this implementation choice. However, it provided a familiar API that allowed parsing the content of incoming packets, selecting only those addressed to the Snort instance and extracting a number of 53 features corresponding to different protocols present in the packets up to Layer 7 in the OSI stack.

Pre-processing module processes captured packets in a specific format by removing redundant information that has very low correlation with detection. A number of 5300 data samples in the training stage for each frequency, out of which 5000 data samples were used for training, and 300, for the validation step during offline learning process.

Generally, a cyber attack leaves its footprints in network as well as application logs. Hence, a more than usual number of communication logs from a particular source to a destination could be indication of an attack. In order to efficiently sift through the huge amounts of logs generated, MapReduce technique can be used. MapReduce is run over the correlation logs to find their frequency of occurrence. Those paths which have occurrence count greater than one is considered a threat.
The Machine Learning Module was active only during the offline training phase. It performed the training of the neural network. System implementation the real-time operation. Once attack features are collected, the next step is to determine the presence of an attack. Machine learning techniques are used to confirm whether the correlation paths flagged as threats in the previous phase actually do belong to attacks.

The Prediction Module executed the classification algorithm based on the trained network and determined the frequency to be set on the processor. The Prediction Module was implemented in Python using the well-known scikit-learn library. The Prediction Engine Module with neural network consisted of three layers with 20 neurons in the hidden layer nodes is used for the training.

Dynamic Voltage and Frequency Scaling (DVFS) is a generic name for techniques that dynamically change the voltage and operating frequency of a processor in order to adapt the energy consumption to the momentary workload. The Frequency Setting (DVFS) Module is the component that actually configured the desired frequency in the processor registers, triggering the frequency change during realtime operation. This DVFS module, also implemented in C, determined the correct frequency by performing a lookup in a pre-configured table and configured the frequency value in the CPU using the interface of the user space Linux power capping governor.

Belief propagation is used to calculate the probability of attack presence based on the earlier mentioned four attributes. Conditional probabilities of each of the individual attributes are then combined together to form the final probability of the attack. If the obtained probability is higher than a particular threshold, then an attack presence is confirmed.

IV. RESULT ANALYSIS

This section presents the measurement results considering the setup presented in Section III. The Amazon Web Services (AWS) public cloud is used as the cloud environment. The proposed solution was implemented and tested with malware samples collected from Packetstorm. The system correctly identifies all the malware samples in real time. When normal (non-malicious) applications are run, the system does not raise any alarm. Hence, cases of false positives are eliminated.

The average detection time and accuracy of the described method as follows:

The time taken by the system to detect malware samples is called detection time. The average detection time is calculated as the result of dividing the sum of detection times for all the samples analyzed by the total number of samples. The system achieves an average detection time of 70 Seconds.

$$\text{Avg. Detection time} = \frac{\sum \text{Detection time for individual malware samples}}{\text{Total number of samples analysed}} \ldots (1)$$

Accuracy is the measure of correct classification.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \ldots\ldots (2)$$

True Positive (TP) refers to malicious activity that occurred and was correctly predicted. False Positive (FP) refers to malicious activity that did not occur but was wrongly predicted. True Negative (TN) refers to malicious activity that did not occur and was correctly predicted. False Negative (FN) refers to malicious activity that occurred but was wrongly predicted.

The obtained results are represented in below Table 1 as:

Table 1: RESULT ANALYSIS

True Positive (T P)	10
True Negative (T N)	6
False Positive (F P)	0
False Negative (F N)	1

Therefore obtained accuracy is

Accuracy= (10+6) / (10+0+1+6) = 0.9411

Fig. 2 presents the energy variation according to the transmission rate for three scenarios: DVFS Prediction Engine with 200 neurons, maximum frequency and ondemand governor. As expected, the energy consumption in the maximum frequency and ondemand scenarios are rather constant.

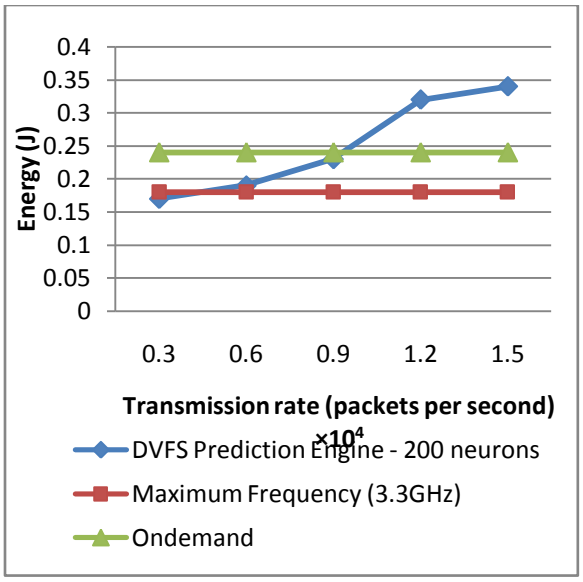

Fig. 2: ENERGY CONSUMPTION

The use of DVFS Prediction Engine, on the other hand, causes a raise on energy consumption in some cases due to the computational overload introduced by the neural network computation. Indeed, the DVFS Prediction Engine curves surpass the maximum frequency when the transmission rate was over 8, 000 pps for 200 neurons.

The correlation of network logs and application logs in the approach enables the user to not only detects anomalous network connections, but also helps the user to identify the source of the connection. On scaling the system to include multiple virtual machines, the network and user logs are extended to incorporate the ID of the virtual machine from which a particular log entry is obtained from. Hence, when an anomaly is detected, users can easily track back to identify the virtual machine as well as the application and user ID from which the suspicious activity has originated from. In particular when the smaller network was used, our method was able to provide better results in an NFV cloud compared to both the industry best practice and to the widely available open source alternative.

V. CONCLUSION

In this paper, Energy efficient intrusion detection for NFV Cloud infrastructures using Machine learning technology is described. Network Function Virtualization (NFV) technology will replace many physical appliances in telecommunication networks with software executed on cloud platforms. Our method employs a pre-trained machine learning model to dynamically determine the best CPU frequency to be used for processing packets. Machine learning algorithms along with big data capabilities can be combined to develop an effective approach for identifying cyber threats and security analytics in virtualized environments. Performance measures such as Detection Rate and Accuracy are used to evaluate the performance of the approach. Potential attack paths are identified from the correlations of packets by using Map Reduce technique. Then, belief propagation is used to confirm the presence of attack. From results obtained accuracy is 94.11% and achieved detection rate as 70 sec. transmission rate of

proposed method is better than compared to setting the maximum frequency or using the ondemand governor.

VI. REFERENCES

[1] Ta Nguyen Binh Duong, Nguyen Quang Sang, "Distributed Machine Learning on IAAS Clouds", 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), Year: 2018

[2] Michel Gokan Khan, Javid Taheri, Andreas Kassler, Marian Darula, "Automated Analysis and Profiling of Virtual Network Functions: the NFV-Inspector Approach", 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Year: 2018

[3] Sooeun Song, Jong-Moon Chung, "Sliced NFV service chaining in mobile edge clouds", 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), Year: 2017

[4] A.M. Riyad, M.S. Irfan Ahmed, R.L. Raheemaa Khan, "Multi agent based intrusion detection architecture for the IDS adaptation over time", 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), Year: 2017

[5] Atiku Abubakar, Bernardi Pranggono, "Machine learning based intrusion detection system for software defined networks", 2017 Seventh International Conference on Emerging Security Technologies (EST)
Year: 2017

[6] O.Y Al-Jarrah, O. Alhussein, P.D. Yoo, S. Muhaidat, K. Taha and K. Kim, Data randomization and cluster-based partitioning for botnet intrusion detection, IEEE Tansactions on Cybernetics, vol. 46, no.8, pp. 1796- 1806, Oct. 2016

[7] F. D. Rossi, M. G. Xavier, E. D. Conte, T. Ferreto, and C. A. F. D. Rose, "Green software development for multicore architectures," in 2014 IEEE Symposium on Computers and Communications (ISCC), June 2014, pp. 1–6.

[8] J. Kuang, L. Bhuyan, and R. Klefstad, "Traffic-aware power optimization for network applications on multicore servers," in DAC Design Automation Conference 2012, June 2012, pp. 1006–1011.

[9] J. Li, Y. Liu, and L. Gu, "Ddos attack detection based on neural network," in 2nd International Symposium on Aware Computing (ISAC),. IEEE, 2010, pp. 196–199

[10] H. Li and D. Liu, "Research on intelligent intrusion prevention system based on snort," in International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE), vol. 1. IEEE, 2010, pp. 251–253.