International Journal of Innovative Research in Engineering
Volume 3, Issue 2 (March-April 2022), PP: 99-102.
www.theijire.com **ISSN No: 2582-8746**

Malware Detection Techniques for Cloud Infrastructure Using Recurrent Neural Networks

NALLUSAMY P[1], GOWTHAM A[2] , LIBIN TITUS T[3] , GUNASEKARAN G[4], DHIVAGAR P[5]
[1,2,3,4,5]*Dhanalakshmi Srinivasan Engineering college, Perambalur / Anna University, Tamilnadu, India.*

How to cite this paper:
NALLUSAMY P1, GOWTHAM A2 , LIBIN TITUS T3 , GUNASEKARAN G4, DHIVAGAR P5 "Malware Detection Techniques for Cloud Infrastructure Using Recurrent Neural Networks" IJIRE-V3I02-99-102.

Abstract: *Several organizations are utilizing cloud technologies and resources to run a spread of applications. These services facilitate businesses save on hardware management, measurability and maintainability concerns of underlying infrastructure. Key cloud service suppliers (CSPs) like Amazon, Microsoft and Google provide Infrastructure as a Service (IaaS) to fulfill the growing demand of such enterprises. This increased utilization of cloud platforms has created it a beautiful target to the attackers, thereby, making the security of cloud services a high priority for CSPs. during this respect, malware has been recognized collectively of the most dangerous and harmful threats to cloud infrastructure (IaaS). during this paper, we tend to study the effectiveness of repeated Neural Networks (RNNs) primarily based deep learning techniques for police work malware in cloud Virtual Machines (VMs). we tend to specialize in 2 major RNN architectures: Long Short Term Memory RNNs (LSTMs) and bidirectional RNNs (BIDIs). These models learn the behavior of malware over time supported run-time fine-grained processes system options like central processor, memory, and disk utilization. we tend to measure our approach on a dataset of 50,480 malicious and benign samples. The method level options were ollected exploitation real malware running in associate degree open on-line cloud surroundings with no restrictions, that is very important to emulate practical cloud supplier settings and additionally capture verity behavior of concealment and complicated malware. Both our LSTM and BIDI models come through high detection rates over ninety nine for various analysis metrics.In addition, associate degree analysis study is conducted to know the importance of input file representations. Our results counsel that especially cases, input ordering will have some have an effect on on the performance of the trained RNN models.*

Key Words: *Deep learning, recurrent neural network, cloud IaaS, online malware detection, long short term memory RNNs, bidirectional RNNs*

I. INTRODUCTION

A heterogeneous cloud may be a advanced platform requiring substantial security infrastructure. in step with the National Institute of Standards and Technology , a cloud platform ought to have essential characteristics not restricted to on-demand self service, broad network access, and resource pooling. These options have helped shaping cloud computing into a customary for each non-public and public sectors. As such, several organizations square measure utilizing the cloud procedure power for various tasks to satisfy growing business wants. Typically, a cloud service supplier (CSP) offers Infrastructure as a Service (IaaS) wherever purchasers square measure allowed to 'rent' house within the sort of virtual machines (VMs) among an information center to facilitate completely different operational jobs. purchasers have the power to spawn several of those virtual machines ondemand. Such a convenient means of utilizing procedure resources comes from the outlined cloud essential characteristics. Recently, the quantity of cloud services, especially VMs, being offered yet because the range of purchasers demanding the employment of those services has raised dramatically. This increase has created the cloud a really fascinating target for attackers since these resources, if exploited, may be recruited to launch giant scale cybersecurity attacks . Cloud malware is one among the foremost common and growing threats wherever a malicious software system is intentionally designed to attack VMs running on a cloud IaaS. though malware may be a well researched challenge , it's impact magnifies in cloud settings because of many underlying reasons: (i) the high demand of cloud resources usage yet because the increase within the range of purchasers considerably broaden the attack vector, (ii) many purchasers lack the power to properly secure their nonheritable resources, and (iii) the increase of automatic configuration tools (e.g., Puppet,1 Chef,2 etc.) any adds to the list of security vulnerabilities. If a VM is spawned employing a script that contains a configuration vulnerability (a flaw in security settings, like failing to auto-encrypt files or amendment a default image root password) it can be left liable to attacks. Further, any VM spawned mistreatment a similar script can possibly have a similar weakness. this can be significantly true in cases wherever a consumer is deploying a large-scale system on the cloud. as an example, deploying internet|an internet|an online} Service utilized by many users can generally embrace multiple web, application, and info servers, that in most cases can all be deployed mistreatment a similar configuration script. The redundant use of

configuration scripts across the servers that conjure internet|an internet|an online} service might permit malware to simply propagate to every server within the web service. Consequentially, police investigation cloud malware in a very real time, online, and effective manner is an important task for CSPs. to deal with these challenges, various malware detection approaches are planned and square measured principally classified into static analysis , dynamic analysis and on-line malware detection . Static analysis works via analysing viables by code examination and making a signature for the executable if it's flagged as a malware, whereas, dynamic Associate in Nursingalysis works by running an viable in a very closed setting (e.g., sandbox) and observation its behavior. on-line malware detection ways specialise in perpetually observation hosts by analyzing traditional and malicious behaviors in the least times. Static and dynamic analysis ways square measure well understood in literature and each have their shortcomings . Static approach falls short against polymorphic malware, that perpetually changes its identifiable options, and zero-day malware. Such subtle malware will evade detection by applying packing and crypting ways to alter the means it's. Dynamic analysis will mitigate the constraints of static analysis since it's supported the behavior of the malware throughout execution; but, good malware will observe the presence of sandboxes and stop malicious activities to avoid detection.

II. OVERVIEW OF DETECTING MALWARE

Detecting malware in a very speedy and effective manner has become a necessity. As such, researchers have utilized machine learning (ML) as a mature and reliable method for static, dynamic and on-line malware detection. during this paper, we introduce associate degree approach of on-line cloud malware detection using deep learning (DL). particularly, we have a tendency to demonstrate the effectiveness of victimisation continual Neural Networks (RNNs) for on-line malware detection by utilizing processes system features of VMs in cloud IaaS environments. Our work is driven by the belief that a lot of VMs running on the cloud square measure mechanically provisioned to try and do a particular task. In turn, such VMs can contain a set set of processes to achieve this task. Note that processes square measure dynamic in nature,so different sudden processes can perpetually be created and deleted. However, an oversized range of the running processes belong to the fastened set. for instance, one VM organized to host net|an internet|an online} service can usually have web server processes (e.g, Apache), information processes (e.g. MySQL), etc. that can be depicted as a sequence. every method during this sequence is depicted as a vector of the utilised system options. Towards this finish, we have a tendency to use RNN to be told the sequence of processes running in a very VM and the way the presence of malware can disrupt this sequence. We conducted associate degree analysis for the malware samples that showed that the bulk of the malware was ready to modification their method names to a legitimate system method. Malware was additionally capable of attaching itself to a legitimate method and, owing to these 2 reasons, typical whitelisting ways aren't effective, thence a lot of subtle ways square measure needed. In our previous work , we have a tendency to used easy shallow CNN model that well-tried effective however with a restricted detection accuracy. This was used as a baseline for our a lot of sophisticated RNN approach.

The main contributions during this paper square measure as follows:

• we have a tendency to introduce a unique approach of detective work cloud malware victimisation RNNs by utilizing processes system options.We demonstrate that the set of processes running in a very VM may be depicted as a sequence of system options.

Further, we have a tendency to highlight that RNNs will effectively discover the presence of malware processes at intervals the benign processes sequence.

• we offer a comparative analysis of Long Short Term Memory (LST M) and Bidirectional (BIDI) models in terms of analysis metrics, along side coaching and detection time.

• we offer associate degree analysis on the result of victimisation totally different input representations. Our experiments counsel that each LSTM and BIDI models achieved high performance no matter the order of system options, whereas, the order of processes at intervals the input sequences compact the performance by a spread of 1-2%.

III. ONLINE MALWARE DETECTION

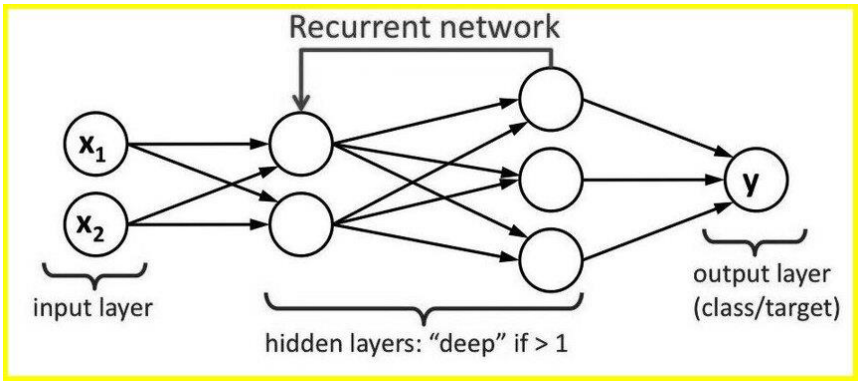
The advantages of on-line malware detection approaches are:(1) they don't deem a closed setting, (2) they Continuously monitor the VMs, as opposition dynamic analysis approaches wherever once associate workable is deemed be it freely runs on the system, and (3) they take into account the whole VM behavior as opposition simply associate workable behavior.The utilize performance counters for on-line malware detection, whereas planned the use of memory features; but, these works used ancient mil algorithms and targeted ancient host-based systems. so as to reinforce the accuracy of malware detection in cloud, a lot of cloud-specific techniques area unit planned. planned associate anomaly detection for VMs in cloud setting exploitation system calls. They used associate ensemble of Bayesian predictors and call trees. Similarly, planned associate intrusion detection system exploitation system calls and used ancient mil algorithms together with KNN and bunch. Further, used API calls captured through the hypervisor and used a non linear phase-space algorithmic rule to sight abnormal behavior. Other works have centered on exploitation options which will solely be fetched through the hypervisor. on condition that several experimental setups area unit run inside the context of a hypervisor, it is common to examine options collected from the hypervisor. Also, such techniques area unit appropriate to be enforced by the CSP since they are doing not need within visibility to the [VMs] utilised performance metrics which will be fetched from the hypervisor so as to sight malware. This paper utilised a 1 category SVM for malware detection; but, they centered on

malware that's known-to-be as highly-active malware. Similarly, demonstrated a recorder primarily based approach to sight malware. This work uses VM-level system and resource utilization features. This worked well in detection extremely active malware with high resource utilization options however wasn't as effective in detection malware that hide itself with low utilization. Beside the works that used ancient mil algorithms, others centered on exploitation deep learning algorithms for on-line malware detection. The extended their add and introduced a detection method that uses a CNN model with the goal of characteristic low profile malware. This technique achieved ≈90% accuracy using resource utilization options and was ready to determine multiple low-profile malware since it centered on per-process level performance metrics. One limitation of this work is that the authors used a shallow CNN model and didn't offer an associate analysis on exploitation numerous CNN models. during this regards, provided a baseline analysis of exploitation stateof-the-art CNN models together with multiple models. we have a tendency to extend this work by providing associate analysis on using RNN.

IV.METHODOLOGY

The projected System options area unit collected from all processes running in a VM at bound time. With several short lived processes (i.e. being created and destroyed quickly among every VM) as well as having their IDs reassigned by the software, it may be dishonest and troublesome to find out their behavior. As such, we tend to outline ''unique processes'' to scale back such dynamism. in contrast to ancient software method that is known by a ''pid'', a unique process is a lot of involved regarding the behavior of a method and is known by a tuple of 2 parts method name and also the command accustomed run the method.
An software processes regenerate to distinctive processes. Processes sharing a similar 2-tuple (e.g., forked processes) are aggregate by taking the typical of their measures. This approach conjointly helps in reducing the amount of processes in an exceedingly single sample.The collected distinctive processes' options are going to be depicted as knowledge samples to be used as input to the RNN models, where every knowledge sample may be a sequence of distinctive processes.Typically, a malware infects a VM and creates one or more processes which is able to disrupt the benign sequence of processes. counting on the malware, it will attach itself to another method and stop its own main method to avoid detection which can flip some existing distinctive processes behavior to malicious.A malware method will hide among the big range of running methodes by renaming its process to some normally used names. However, victimisation the conception of distinctive method makes it tougher for the malware method to cover as a result of the number of distinctive processes is considerably smaller. Further, a malware method are going to be a lot of visible since it'll be thought-about a novel method. Our aim is to find out from the sequence of processes (including benign processes that a malware connected to) in an exceedingly given sample and to spot it as malicious or benign.

Fig.1 Working Model Of Recurrent Neural Network



V. RNN SENSITIVITY TO DIFFERENT INPUT REPRESENTATIONS

Each sample consists of a sequence of distinctive processes. However, it's not clear whether the order of distinctive processes and options in an exceedingly single sample would have an effect on the RNN models' ability to find out and generalize effectively. sterilization the ordering of the input file can typically reveal insights on a way to best train bound models.
For instance, the provided associate analysis on the effects of input ordering once victimization CNN models. They used similar method system options for malware detection victimization CNN models and studied the results of processes and options
ordering within the input, delineate as a picture (denoting processes × features). The authors performed experiments with CNN models by generating totally different sets of a similar input data with totally different orderings. during this study, the authors were able to enhance the accuracy of detection malware from ninetieth to 98%. As such, it had been complete that bound orderings of input data will if truth be told improve the performance of the models and should be made properly. In this section, we offer associate analysis on whether or not the order of sequence in an exceedingly single sample (denoted by row models) as well because the order of options (denoted by gap models) would affect the results of the RNN models. The key intuition of this analysis lies within the undeniable fact that some distinctive processes may be closely connected, and together with them in shut proximity in the input

sequences may facilitate the models to simply draw andlearn such correlations. as an example, contemplate 2 distinctive processes of the FastCGI method Manager (FPM) php-fpm: master and also the its forked pool of processes php-fpm: pool . Similarly, some system options may well be connected. For example, options of hardware usage like cpu_user, cpu_sys, cpu_num, and cpu_percent square measure closely connected. As per totally different row orderings square measure created by indiscriminately changing the sequence of distinctive processes for all samples. Similarly, totally different column orderings square measure created by indiscriminately changing the order of the options that belong to every distinctive process. This will increase the percentages of preventing connected processes or options from showing in shut positions in an exceedingly given sample.

VI. EXPERIMENTAL SETUP

All experiments resulted in 50,480 information samples collected.This is as a result of the information we have a tendency to area unit aggregation represent the behavior of all processes within the virtual machine, not simply the actual malware executables. Models coaching was performed on a high performance computing center (HPC) with four Dell PowerEdge R730 servers, every with one NVIDIA Tesla K80 GPU. The RNN models were engineered and tested by Python scripts mistreatment Keras7 API that is made on high of Tensorflow.8 , As explicit in Section III-D, the input to the RNN models is a sequence of vectors, every denoting the options for a particular distinctive method. In our experiments, the Almost number of distinctive processes in any experiment is one hundred twenty, hence,all sequences area unit soft to be of identical length. The system options collected for every distinctive method are preprocessed by changing categorical string options to one-hot vectors and standardizing the information values.

VII. CONCLUSION

we analyzed the impact of input representations on our models by conducting random ordering experiments with relevance distinctive processes and options (i.e., col and row experiments). On one hand, the results showed that the order of the options doesn't impact the models performance, whereas, it impacts the coaching time of the models. On the other hand, the order of distinctive processes impacts the performance yet because the coaching time of the models. In the future, we have a tendency to decide to increase the dimensions of our experiments by victimisation thousands of malware samples as well as a lot of malware families. in addition, we have a tendency to decide to study the impacts of malware propagation to equally designed VMs in an exceedingly cloud surroundings on the hardiness of our detection models. we have a tendency to additionally decide to study the impacts of multiple malware infections to an equivalent VM.

References

[1] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," IEEE Security & Privacy, vol. 9, 2011.

[2] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacono, "On technical security issues in cloud computing," in IEEE CLOUD, 2009.

[3] N. Gruschka and M. Jensen, "Attack surfaces: A taxonomy for attacks on cloud services," in IEEE CLOUD, 2010, pp. 276–279.

[4] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," IEEE Communications Surveys & Tutorials, vol. 15, no. 2, 2013.

[5] K. Dahbur, B. Mohammad, and A. B. Tarakji, "A survey of risks, threats and vulnerabilities in cloud computing," in ISWSA, 2011.

[6] A. Gholami and E. Laure, "Security and privacy of sensitive data in cloud computing: a survey of recent developments," arXiv preprint arXiv:1601.01498, 2016.

[7] M. Abdelsalam, R. Krishnan, and R. Sandhu, "Clustering-based IaaS cloud monitoring," in 10th IEEE CLOUD. IEEE, 2017.

[8] J. Demme and et al., "On the feasibility of online malware detection with performance counters," in ACM SIGARCH Computer Architecture News, vol. 41, no. 3. ACM, 2013.

[9] G. Tahan, L. Rokach, and Y. Shahar, "Mal-ID: Automatic malware detection using common segment analysis and meta-features," Journal of Machine Learning Research, vol. 13, no. Apr, 2012.

[10] J. Z. Kolter and M. A. Maloof, "Learning to detect and classify malicious executables in the wild," Journal of Machine Learning Research, vol. 7, no. Dec, 2006.

[11] T. Abou-Assaleh and et al., "N-gram-based detection of new malicious code," in COMPSAC, vol. 2. IEEE, 2004.

[12] A. Shabtai and et al., "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey," information security technical report, vol. 14, no. 1, 2009.

[13] B. Athiwaratkun and J. W. Stokes, "Malware classification with LSTM and GRU language models and a character-level cnn," in ICASSP. IEEE, 2017.

[14] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in 10th MALWARE. IEEE, 2015.

[15] S. Seok and H. Kim, "Visualized malware classification based-on convolutional neural network," Journal of the Korea Institute of Information Security and Cryptology, vol. 26, no. 1, 2016.