A Survey of Methodologies Available to Detect Security Attacks On Cloud Infrastructure

Sri Kiran Jyothi B R

ME – 2nd Year, Department of Computer Science
S A Engineering College, Anna University
Thiruverkadu, Chennai, Tamil Nadu 600077

Srikiran.jyothi@yahoo.com

Priya V

ME – 2nd Year, Department of Computer Science
S A Engineering College, Anna University
Thiruverkadu, Chennai, Tamil Nadu 600077

priyavira@gmail.com

C. BALAKRISHNAN

Associate Professor, Department of Computer Science
S A Engineering College, Anna University
Thiruverkadu, Chennai, Tamil Nadu 600077

balakrishnan@saec.ac.in

Received 15 February 2019 Received in revised form 19 February 2019 Accepted 20 February 2019

ABSTRACT

Security has always played an important part in an organization and securing access to data / network has become tantamount because of the volume / criticality of data that each organization stores and maintains, which can include data such as health records, credit card information, bank account information, SSN's, passwords etc. Normally such critical data would be stored in a local network but with the growth of the volume of data and the velocity at which this data is being generated everyday it has become impossible to store and maintain this data. Then came the concept of Cloud Computing and with it came the sudden explosion of Private Clouds and IAAS (Infrastructure as a Service). Companies prefer Cloud networks rather than having to invest in purchasing hardware and hence go for IAAS where they can lease hardware on the go on the cloud. IAAS services are provided by major players such as AWS (Amazon Web Services), GCP (Google Cloud Platform) or Microsoft Azure. Though Clouds solve the problem of storage along with them comes the problem of securing these virtual machines which make up the nodes of a Cloud Network which is why it makes it easy to grow a cloud network on the fly. Virtual Machines or VM's are just software and can be attacked using Malware / Virus / backdoors which will allow hackers to gain access to these machines and the data on them. Let us understand some of the different types of attacks and the techniques available to detect and stop them.

Keywords:Virtual Machines, Cloud Security, Big Data, Anomaly Detection, Intrusion Detection, Network Security

I. INTRODUCTION:

Cloud Computing [1] is a technology which allows consumers access to a broad range of computing resources, products and stored information whenever they need them, where ever they need them, using a variety of devices. Cloud Computing services are marketed as a utility in a similar manner to traditional electricity, gas, water and telephony provision. The simplicity and scalability that cloud computing offers has attracted the attention of both private citizens and enterprises. Virtualisation is the fundamental technology that enables cloud computing and differentiates it from traditional IT deployments by dramatically improving machine utilisation and reducing overall total cost of ownership.

Virtualisation is the emulation of the software and/or hardware platforms upon which other software and operating systems run. Ideally, virtualisation allows us to

build an environment that enables one computer to perform the tasks of multiple diverse computing platforms, by sharing the resources of a single hardware platform across multiple virtual systems. An emulated system is called a virtual machine. The operating system installed in a virtual machine is called a guest operating system. The guest operating systems on a host are managed by either a hypervisor or a Virtual Machine Monitor. This additional software layer controls the flow of instructions between the guest operating systems and the physical hardware e.g. the CPU, disk storage, memory, and network interface cards.

II. TYPES OF SECURITY ATTACKS

There are many different types of security attacks possible on a computer / network such as the below

- *Botnets [3]:*A botnet is a network of compromised hosts (bots) which are controlled by an attacker also called the botmaster. The botmaster sends commands via a C&C (Command and Control) channel. Although initial botnets relied on a central design with a core network of few interconnected IRC (Internet Relay Chat) servers, current botnets are based on P2P technologies; In a P2P design, each bot acts as a client and a server. Hence, for triggering a command to some bots, other bots are involved. Therefore, P2P bots are well interconnected
- *Spamming [2]*: Spamming is sending unsolicited bulk messages to multiple recipients. By 2015, the spam volume is forecasted to be 95% of all email traffic. Munging, access filtering and content filtering are some of the vital anti-spam techniques. Munging makes email addresses unusable for spammers, e.g., cde@email.co munges as “cde at email dot co”. Access filtering detects spam based on IP and email addresses while content filtering recognizes predefined text patters in emails to detect spam.
- *Malware:*Malware is software programmed to perform and propagate malicious activities, e.g., viruses, worms and Trojans. Viruses need human intervention for propagation, worms are self-propagating, while Trojans are non-self-replicating. Damage from malware includes corruption of information or OS, installation of spyware, stealing personal credentials or hard disk space etc
- *Phishing*: Phishing fraudulently acquires confidential user data by mimicking e-communication, mainly through email and web spoofing. In email spoofing, fraudulent emails direct users to fraudulent web pages which lure to enter confidential data. In internet spoofing, fraudulent websites imitate legitimate web pages to deceive users into entering data. Many anti-phishing solutions are deployed in company use to counteract this threat.
- *Search Poisoning*: Search poisoning is the dishonest use of Search Engine Optimization techniques to falsely improving the ranking of a webpage. Typically, frequent search keywords are exploited to illicitly direct users towards temporary websites. The first poisoning case was reported in 2007 followed by many others.
- *Denial of Service (DoS):*ADoS attack makes a system or any other network resource inaccessible to its intended users. It is launched by a large number of distributed hosts, e.g., bot net. Many defensive techniques such as intrusion detection systems, puzzle solution, firewalls etc. have been developed to prevent DoS attacks.
- *The HeartBleed Vulnerability [4]:*The Heartbleed vulnerability took the Internet by surprise in April 2014. The vulnerability, one of the most consequential since the advent of the commercial Internet, allowed attackers to remotely read protected memory from an estimated 24–55% of popular HTTPS sites. Heartbleed permits attackers to access sensitive memory from vulnerable servers, potentially together with cryptographical keys, login credentials, and other personal information.

III. TECHNIQUES TO COMBAT ATTACKS

Just as there are many ways to attack a system there are also many ways to detect and prevent such attacks as well which we will see below

Malware Detection [5]

One of the most important challenges in the development of resilient and secure cloud-oriented mechanisms correlates to the adequate identification and detection of malware. This is because of the very fact that, in the majority of cases, malware is the first point of initiation for large-scale Distributed Denial of Service (DDoS) attacks, phishing and email spamming, mainly through the deployment of botware. Current strategies of identifying attacks on cloud infrastructures or the VMs resident among them don't sufficiently address cloud specific problems. Despite the huge efforts employed in past studies regarding the behaviour of certain types of malware in the Internet, so far little has been done to tackle malware presence in clouds. Some studies have aimed to adjust the performance of traditional Intrusion Detection Systems (IDS) under signature-based techniques that employ Deep Packet Inspection (DPI) on network packets. There has also been research done on system-related features on monitored VMs by employing Virtual Machine Introspection (VMI) methods in order to detect threats on a given VM's Operating System (OS).

Malware Detection using Static Analysis [6]

A static analyser inspects a software app by simply disassembling and, de-compilation without actually running it, hence does not infect the device. Since it analyzes an app's whole source code or recovered code, the analyzer can achieve high code coverage. Static analysis lacks the particular execution path and relevant execution context. Moreover, there exist challenges due to the presence of code obfuscation in addition as dynamic code loading. All those approaches lack the ability to analyze code that is obfuscated or loaded dynamically at runtime, , unless they are complemented by some form of dynamic analysis, as recently proposed in StaDynA

Malware Detection using Dynamic Analysis

Static analysis and detection approaches though are fast, they fail against the encrypted, polymorphic and code transformed malware. In order to overcome the shortcomings of static analysis, some dynamic analysis-based methods have been proposed. Dynamic analysis is conducted by executing an app, on either a real or virtual execution environment such as the Android Virtual Device (AVD), and observing the app during its execution.

The analysis system TaintDroid (Enck et al. 2010) and DroidScope (Yan and yin 2012) are said to be the most notably, which enable dynamically monitoring applications in a protected environment.

- TaintDroid focuses on taint analysis and
- DroidScope make introspection at different layers of the platform.

Although each systems give elaborate info concerning the behavior of apps, they need too many resources to deploy on Smartphones directly. Although dynamic analysis surpasses the static analysis in several aspects, dynamic analysis also has some drawbacks.

- Firstly, dynamic analysis requires too many resources relative to static analysis, which hinders it from being deploying on resource constraint smartphone.
- Secondly, dynamic analysis is subject to low code coverage. Sasnauskas and Regehr (2014) mentioned that producing extremely structured inputs that get high code coverage is an open scientific challenge.
- Thirdly, recently malware attempts to detect the emulator and other dynamic analysis systems, avoiding launching their payloads. Thus, some dynamic analysis systems are susceptible to analysis evasion.

Anomaly Detection in Clouds [5]

Anomaly detection has been an energetic research space for quite a number of years. Numerous techniques for different scenarios and application domains have been developed for] the prediction, detection and forecasting

accuracy of anomaly detection in a number of disciplines. This paper proposes a anomaly detection technique to detect intrusions at different layers of the cloud. There is aalso a multi-level approach, which provides fast detection of anomalies discovered in the system logs of each guest OS. One of its drawbacks is that the apparent lack of performance and scalability since it needs a lot of} more resources under high system load. Further, it's designed to classify text-based log information, which can not manifest the consequences of malware.

In-VM and outside-VM interworking approach to malware detection[7]

In-VM and outside-VM interworking detection consists of an in-VM agent running within the guest VM, and a remote scrutiny server monitoring the VM's behaviour. When a potential malware execution is detected the in-VM agent sends the suspicious executable to the scrutiny server, which then uses the signature database to verify malware presence or otherwise and then informs the in-VM agent of the results.

- *CloudAV,*a cloud-based malware detection system featuring multiple antivirus engines, employs in-VM and outside-VM interworking approach to protect the guest VMs against attacks. Apparently, the effectiveness of this scheme depends on the frequency at which the virus signatures are updated by the antivirus vendors.
- *CuckooDroid*also uses the in-VM and outside-VM interworking approach to detect mobile malware presence on Android devices. It consists of an in-device agent which scans executables on the device and sends any suspicious executable to a remote scrutiny server which runs a hybrid of anomaly-based and signature-based malware detectors. The scheme first extracts malware features by using static as well as dynamic analysis on malware apps. The obtained features are then used to train a one-class SVM (Support Vector Machine) classifier for anomaly-based detection.

Hypervisor-assisted malware detection

Hypervisor-assisted malware detection, on the other hand, uses the underlying hypervisor to detect malware within the guest VMs. The solution installs a network sniffer on the hypervisor to monitor external traffic as well as inter-VM traffic. A hypervisor-assisted detection scheme is proposed in this paper using guest application and network flow characteristics.

This paper first uses LibVMI to extract key process features from the processes running within VMs and then uses tcpdump together with the CoralReef network packet analysis tool from CAIDA (Center for Applied Internet Data Analysis) to extract network flow features. The obtained features are then used to train one-class SVM classifiers to detect malware presence within guest VMs.

Data Mining based Intrusion Detection [8]

Intrusion Detection is used to protect attacks on systems and to maintain the data integrity thus ensuring system stability. In order to detect intrusions, we need to use Data Mining techniques. Data Mining is required to analyze the data collected from all of the different VM's in a network which can include User logs or Application Logs etc to detect intrusions. There are many techniques that can be used for Intrusion Detection. Any such technique would involve the below 3 components – Source of the Information, Analysis of the Information and Providing a response based on the analysis. The response here in this case would be to detect intrusion, let us understand a few techniques in use today.

Clustering:Clustering is the process of labeling data and assigning it into groups. Clustering algorithms will cluster new information instances into similar groups. Clustering techniques may be categorised into the below classes: pairwise clustering technique and central clustering technique. Pairwise clustering technique i.e., similarity-based clustering) unifies similar data instances based on a data-pairwise distance measure. On the other hand, Central clustering, also called centroid-based or model-based clustering, models each cluster by its "centroid". In terms of runtime efficiency, centroid-based clustering algorithms are more efficient than similarity-based clustering algorithms. Clustering discovers complicated

intrusions occurred over extended periods of time and totally different areas, correlating totally independent network events.

Classification: Classification is similar to clustering in that it also partitions customer records into distinct segments called classes. Classification categorizes the information records in a preset set of categories used as attribute to label each record; identifying elements belonging to the normal or abnormal category. This technique has been common to discover individual attacks. As compared to the Clustering technique, classification technique is less popular in the domain of intrusion detection. The main reason for this development is that the great amount of information required to be collected to use classification.

OutlierDetection: An outlier is an infrequent observation that immensely deviates from the characteristic distribution of other observations. Outlier detection has several applications, such as data cleaning, fraud detection and network intrusion. The existence of outliers indicates that elements or groups that have very much different behavior from most of the elements of the dataset. Since an outlier could also be described as an information point that is completely different from the rest of the information, we will use many outlier detection schemes for intrusion detection that are based upon mathematical and statistical measures, clustering strategies and data processing strategies. Commonly used outlier techniques in intrusion detection area unit Mahalanobis distance, detection of outliers using Partitioning around medias (PAM), and Bay's algorithm for distance-based outliers. Outlier detection approaches are useful for discovering any unknown attacks. This is the most important reason that makes the outlier detection a well-liked approach for intrusion detection systems.

The VMI IDS [9]

The VMI IDS is accountable for implementing intrusion detection policies by analyzing machine state and machine events through the VMM interface. The VMI IDS is split into 2 components, the OS interface library and the policy engine. The OS interface library's job is to supply an OS-level representation of the virtual machine's state so as to facilitate simple policy development and implementation. The policy engine's job is only to execute IDS policies by exploiting the OS interface library and also the VMM interface.

The OS Interface Library

VMMs manage state strictly at the hardware level, but choose to reason regarding intrusion detection in terms of OSlevel language. A VMM will give us access to any page of physical memory or disk block of a virtual machine, but discovering the contents of sshd's code segment requires answering queries regarding machine state within the context of the OS running within the VM. We need to supply some ways of deciphering low level machine state from the VMM in terms of the upper level OS structures. The OS interface library solves this drawback by exploiting information regarding the guest OS implementation to interpret the VM's machine state, that is exported by the VMM. The policy engine is supplied with an interface for creating high-level queries concerning the OS of the monitored host. The OS interface library should be matched with the guest OS. Completely different guest OSes can have different OS interface libraries. The OS interface library additionally facilitates queries at the level of kernel code.

The Policy Engine

This part interprets system state and events from the VMM interface and OS interface library and decides if or not the system has been compromised. If the system has been compromised, the policy engine is responsible for responding in an appropriate manner. For example, in case of a break-in, the policy engine can suspend or reboot the virtual machine and report the break-in.

VMM Interface

The VMM interface provides a channel for the VMI IDS processes to interact with the VMware VMM process. This interface is composed of two parts: first, a Unix domain socket that allows the VMI IDS to send

commands to, and receive responses and event notifications from, the VMM; and second, a memory-mapped file that supports efficient access to the physical memory of the monitored VM.

Security Analytics for Threat Detection [2] :

The largest application of security analytics is in threat observation and incident investigations, which is of major concern to both financial and defense institutions. The focus is identifying hidden threats faster, track down attackers and predict future attacks with increasing accuracy (minimum false positive rate). Below are some of the channels that need to be monitored for threats

• *Network Traffic*: Monitoring, detecting and predicting suspicious sources and destinations, along with abnormal traffic patterns.

• *Web Transactions*: Monitoring, detecting and predicting abnormal user access patterns, particularly in the usage of critical resources or activities.

• *Network Servers*: Monitoring, detecting and predicting abnormal patterns related to server manipulation, e.g., abnormal or sudden configuration changes, non-compliance with pre-defined policy etc.

• *Network Source*: Monitoring, detecting and predicting abnormal usage patterns of any machine, e.g., related to the type of data the source transmits, processes and receives.

• *User Credentials*: Monitoring, detecting anomalies with regard to a user, or a group of user, not compliant with its inherent access behavior, e.g., abnormal access time or transaction amount.

These activities have brought a revolutionary change within the domains of security management, identity and access management, fraud detection, prevention and governance, risk and compliance, e.g., through centralization of threat data and alert management system, correlating hundred thousands of network events per second, almost realtime continuous assessment of risk, distinction between legitimate and abnormal activities, in conjunction with appropriate prioritization of risks.

Cyber-Targeted Attack Response System [10]

This detects abnormal behavior by analyzing the correlation between security events occurring in the existing security equipment, network traffic and statistical information, and all behavior information occurring at the host to detect the cyber-targeted attack. In addition, the system provides the function traces back the attack source based on the analysis results and security visualization for administrator to see and understand the information conveniently. Data agent in the data and event sensor layer is installed in the target system for collecting the data; it senses the data generated by each system and transfers the data to the data collection layer. On the other hand, the data collection layer receives the data from data sensors, saves large-scale accumulated data in the Hadoop file system (HDFS: Hadoop Distributed File System);, and transfers the real-time processing data to the real-time big data processing part. The large-scale data storage/processing layer is composed of HDFS and Hbase (NoSQL) to save the large-scale data, and MySQL Cluster (in-memory DBMS for real-time processing). The large-scale accumulated data and real-time processing data will be processed by MapReduce and Esper/Storm, respectively. The targeted attack analysis layer consists of real-time data analysis, accumulated data analysis, and correlation analysis..

AccessMiner [11]:

AccessMinerisa system-centric behavioral malware detector designed to model the general AccessMiner is a system-centric behavioral malware detector designed to model the general interactions between benign programs and the underlying operating system (OS). In this method, AccessMineris able to capture which, and how, OS resources are used by normal applications and detect anomalous behavior in real-time. It does not require to be trained on malicious samples, and therefore it is able to provide a general detection solution that can be used to protect against both known and unknown

malware. To make the system a lot more resilient against change of state from advanced attackers, AccessMiner is developed as a custom hypervisor that sits below the software. The main idea behind the AccessMiner approach is that, given enough training, it is possible to identify common patterns in the way benign applications interact with the operating system resources. For instance, while normal programs typically write only to their own directories (and to temporary directories), malware often attempt to tamper with other applications and critical system settings, often residing outside the normal application "scope". As a result, special access activity models can be derived by AccessMiner only by looking at the execution of a broad set of benign applications.

Malware Detection System for Virtual Environments [1]:

The Malware Detection System for Virtual Environments (MDSVE) observes network traffic and identifies any patterns and trends that indicate activities which can potentially have malicious effects on the virtualised environment. The overall functionality of MDSVE is to capture the network traffic of each virtual machine and build useful contextual information to aide traffic analysis. Detected threats will precipitate security alerts or direct action on the VMs involved.

Tasks performed by MDSVE are:

1. Track virtual machine lifecycle
2. Monitor virtual machine communications i.e. internal communications between VMs in the same physical host, and external communications traversing a network interface card on the physical host
3. Capture malware activities
4. Match any suspicious activities with the corresponding virtual machine.
5. Inform management console about suspicious virtual machine instances.

The MDSVE is made up of four basic functional blocks:

- *Network Sniffer- (NS)*captures all of the network packets in real-time and performs flow classification. That is, all packets relating to a logical session are linked together and offered up for further analysis as a contiguous flow of traffic.
- *Malware Trait Detector- (MTD),*analyses the complete flow looking for series of events and features which indicate the presence of specific malware.
- *Virtual Machine Information Collector- (VMIC)*monitors the virtual machine life cycle and captures the virtual machine status along with basic parameters so as to match the virtual machine with the suspicious bot as analysed by the malware analyser and alerts the virtualisation security manager.
- *Virtualisation Security Manager- (VSM)*acts as a security console displaying the alerts and information provided by the VMIC. If malware is detected on any of the virtual machines, then according to the severity of the threat VSM can take action accordingly. For example, inform host based anti-virus system, sinkhole traffic destined to the virtual machine, or even suspend the virtual machine

CONCLUSION

With the proliferation of Cloud Computing and with many organizations starting to store data in the cloud rather than building their own infrastructure, it becomes more imperative to protect these systems and the data they store from attacks. Even though we make advancements in the fields of malware detection, intrusion detection, anti virusesetc so will hackers find more and more new ways of overcoming these obstacles and getting access to what they want. So the attack detection and prevention technologies of the future should be faster, lighter, distributed, portable, self learning and self repairing because of the amount of time and effort it would take for human man power to even monitor / detect such attacks.

RFERENCES

- [1] Pushpinder Kaur Chouhan, Matthew Hagan, Gavin McWilliams, and SakirSezer "Network Based Malware Detection within Virtualised Environments"

- Centre for Secure Information Technologies, Queens University of Belfast, Northern Ireland, UK
- [2] Dr. Tariq Mahmood, Uzma Afzal “"Security Analytics: Big Data Analytics for Cybersecurity", 2013 2nd National Conference on Information Assurance (NCIA)
- [3] Jerome Francois, Shaonan Wang, Walter Bronzi, Radu State, Thomas Engel "BotCloud: Detecting Botnets Using MapReduce", University of Luxembourg – Interdisciplinary Center for Security, Reliability and Trust
- [4] ZakirDurumeric , James Kasten , David Adrian , J. Alex Halderman , Michael Bailey, Frank Li, Nicholas Weaver,JohannaAmann, Jethro Beekman, Mathias Payer, Vern Paxson "The Matter of Heartbleed" - IMC '14 Proceedings of the 2014 Conference on Internet Measurement Conference
- [5] Michael R. Watson, Noor-ul-hassanShirazi, Angelos K. Marnerides, Andreas Mauthe and David Hutchison "Malware Detection in Cloud Computing Infrastructures" - IEEE Transactions on Dependable and Secure Computing (Volume: 13 , Issue: 2 , March-April 1 2016)
- [6] Wang, Xiaolei& Yang, Yuexiang& Zeng, Yingzhi. (2015). Accurate mobile malware detection and classification in the cloud. SpringerPlus. 4
- [7] T. Y. Win, H. Tianfield and Q. Mair, "Big Data Based Security Analytics for Protecting Virtualized Infrastructures in Cloud Computing," in *IEEE Transactions on Big Data*, vol. 4, no. 1, pp. 11-25, 1 March 2018.
- [8] C. -. Lu, A. P. Boedihardjo and P. Manalwar, "Exploiting efficient data mining techniques to enhance intrusion detection systems," *IRI -2005 IEEE International Conference on Information Reuse and Integration, Conf, 2005.*, Las Vegas, NV, USA, 2005, pp. 512-517.
- [9] Garfinkel, Tal & Rosenblum, Mendel. (2003). A Virtual Machine Introspection Based Architecture for Intrusion Detection. NDSS. 3.
- [10] Kim, Hyunjoo& Kim, Ikkyun& Chung, Tai-Myoung. (2014). Abnormal Behavior Detection Technique Based on Big Data. Lecture Notes in Electrical Engineering. 301. 553-563.
- [11] Fattori, A., Lanzi, A., Balzarotti, D., &Kirda, E. (2015). Hypervisor-based malware protection with AccessMiner. Computers & Security, 52, 33-50.