Detection of Malware Attacks on Virtual Machines for a Self-Heal Approach in Cloud Computing using VM Snapshots

Linda Joseph and Rajeswari Mukesh

Original scientific paper

***Abstract*—Cloud Computing strives to be dynamic as a service oriented architecture (SoA). The services in the SoA are rendered in terms of private, public and in many other commercial domain aspects. These services should be secured and thus are very vital to the cloud infrastructure. In order, to secure and maintain resilience in the cloud, it not only has to have the ability to identify the known threats but also to new challenges that target the infrastructure of a cloud. In this paper, we introduce and discuss a detection method of malwares from the VM memory snapshot analysis and the corresponding VM snapshots are classified into attacked and non-attacked VM snapshots. As snapshots are always taken to be a backup in the backup servers, this approach could reduce the overhead of the backup server with a self-healing capability of the VMs in the local cloud infrastructure itself without any compromised VM in the backup server. A machine learning approach is projected here to classify the attacked and non attacked snapshots. The features of the snapshots are gathered from the API calls of VM instances. Our proposed scheme has a high detection accuracy of about 93% while having the capability to classify and detect different types of malwares with respect to the VM snapshots. Finally the paper exhibits an algorithm using snapshots to detect and thus to self-heal. The self-healing approach with machine learning algorithms can determine new threats with some prior knowledge of its functionality.**

***Index Terms*—Cloud Computing; VM Snapshots; Machine Learning Algorithms; API Calls; Self-Healing.**

I. INTRODUCTION

The advent of the cloud enables it to be used as a service oriented architecture with its many services ranging across private, public and hybrid clouds. Most of the leading companies have resorted to cloud providers with services such as pay-as-you-go and on-demand of the virtual resources. This brings in numerous cost savings and benefits for the companies to achieve higher levels of reliability, scalability and availability. Cloud services are divided mainly as SaaS, PaaS and IaaS, of which the IaaS component has evolved to contain most of the challenges due to its much flexibility to the end users. Infrastructure as a Service (IaaS) is where the customers have the most of the control. It enables virtual machines (VM's) to be deployed as resources in the form of services. The different services provided by IaaS with virtual machines are print services, web services, mail services and so on. Software as a Service (SaaS) enables the customers to access applications on demand. Platform as a Service enables the customers to access the required platforms to develop and code.

In spite of all the security measures cloud computing is not far from traditional and newer generation security threats. Vulnerabilities of cloud computing includes threats from core technologies of web applications and services, cryptography and virtualization. Access vulnerabilities include unauthorized access, Internet protocol vulnerabilities, and loopholes in security controls and in authentication schemes. Numerous attacks on virtual machines are the flooding attacks, backdoor attacks, user to root attacks such as the root kit attacks, inter communication attacks such as the internal denial of service attacks on the hypervisor such as the VM escape attacks and so on.

The key component of cloud computing is the virtualization technology. Virtual machines use the concepts of virtualization technology to enable multiple operating system environments in the virtual machine instances, in a single physical machine/server. The required number of resources are scheduled and deployed with the expectation of the property of isolation, that is, each virtual machine deployed has to work without any connection with the other virtual machines. The hypervisors are solely responsible for providing the virtualized environment by managing the physical machines. It should also provide the virtual devices to the VMs which are in isolation to each other with fairness. Thus the hypervisor has to improve the overall performance of all the virtual machines with the available physical resources.

The need of the hour is that, these VMs have to be not only secured but also should be made resilient, that is able to identify the malware threats and should avoid these types of threats from entering the VM's and to restore its functionalities immediately. In spite of the cloud having its benefits of each VM created in isolation, it has number of vulnerabilities in terms of security breaches. It is presumed that the current cloud infrastructures are prone to various types of attacks which may be extravagant in nature.

Manuscript received April 6, 2018; revised July 30, 2018. Date of publication September 17, 2018. Prof. Nikola Rožić has been coordinating the review of this manuscript and approved it for publication.

Authors are with the Hindustan Institute of Technology and Science, Chennai, Tamil Nadu, India.

E-mails: {lindaj, rajeswarim}@hindustanuniv.ac.in

Digital Object Identifier (DOI): 10.24138/jcomss.v14i3.537

Studies indicate that signature based techniques make usage of the deep packet inspection (DPI). DPI indicates that the packets in the network are monitored by evaluating the contents of the packet. It basically uses the rules or policies that are already defined. Intrusion detection systems with signature based more particularly employ DPI for network packets. The approach taken by the paper [1] discusses a scheme which uses per-flow metastatics of a packet and its volumetric information. This scheme works with signature based anomaly detection on an online basis.

For this reason, this component of cloud has to be more secured from malwares and vulnerabilities. We consider the IaaS layer of cloud computing, as this layer is the most sensitive layer of cloud and prone to various types of attacks. Attacks may be oriented towards resource scheduling, VM live migration, network connectivity. The elements that make up this layer comprises of the

1. Cloud Nodes that serves as hardware servers running a hypervisor to host number of VM's.
2. The network infrastructure components that provides network connectivity within the cloud structure and thus to the users connected with that particular cloud node. The VMs from a cloud node may be given to the requesting users by the service providers.
3. The Scheduling and provisioning on demand component of IaaS layer of cloud.

II. Related Work

This paper presents a cloud security mechanism wherein, a number of snapshots are given to the training phase of the machine learning algorithms and once when such a snapshot of a given VM is compared with the trained set, the VM identifies and is saved from further attacks. The machine learning algorithm is able to identity these set of snapshots and thus able to save the created VM from further attacks.

A. Virtualization and its aspects on cloud

Virtualization is considered to be a large and a more dynamic field of research, with utmost new technologies and threats coming out frequently with vivid solutions and fully complete. A VM should be secured from attacks from within itself such as malware [10, 3], guest OS root kits [11] or from other co-resident VM's on the same physical machine. Threats can affect the virtual machine manager, the virtual machines itself, the operating systems in the VM instances, the applications running on these Oss and the network. Security refers to sensitive data free from disclosure and alteration of data. According to [8], botnets have become of a greater concern in the cloud paradigm, where these botwares are capable of sneaking into the cloud environment. In [10], [11]. Yi han has authored on co residence attacks in VMs which are members on the same host allocated to different users in the cloud environment which is a security concern. They have proposed game theoretical approaches to control the co resident attacks on the neighboring VMs.

Also, the concept of malicious VM migration has enabled more side effects on the security of the cloud infrastructure. Migration is needed as per of the customers or due to the load balancing policy made by the provider. This leads to the presence of the VMs in potential risk when it comes in contact with huge targets of threats in the cloud infrastructure. More specifically automation has made vulnerabilities and treats to be propagated to large scale cloud infrastructures, where malwares easily sneak into the VM configurations. One such tool is the Ansible, an automation engine where cloud provisioning, application deployment, configuration management and many other IT needs are automated. These automation tools have paved the creation of new VMs from clones or snapshots, thus resulting in the collection of servers all with the same functionality prone to vulnerabilities and threats [1].

Data to be secured should have the properties of confidentiality, integrity and availability of (i) Data in memory, on disk or in any other form of storage (ii) The workflow state such as resource allocation and scheduling levels, the execution paths etc (iii) The network and control levels. Security has to be pertained to agents such as the VMM, VM instances, OSs in VM instances and the applications running in VMs.

B. Security in Virtual Machines

The foremost and the biggest challenges of cloud computing is the security and health of the virtual machines in the IaaS cloud environment, where a customer has to be provided with the virtual machines(VM) in the cloud system. The requested VM with its mentioned specifications has to be placed in the cloud server. Previous research indicates security-as-a service that could be provided and provisioned to the customers according to their security specification and needs [7]. The health of the VM has to also monitor whether the VM which is secured meets the requirements of a healthy VM. Any VM thus created must satisfy the properties of security viz confidentiality, integrity, availability and authenticity. A VM's security health [12] is monitored for any malwares or guest OS root kits, any co resident VMs on the same server [5, 6, 10, 11].

Past research shows that in order to eliminate side channel attacks, number of solutions have been discussed in [7,10,11].Study indicates that an attacker can achieve co-residency with an estimate of about 40% that could mean that almost 4 attacker VM's out of 10 could co locate a VM victim. The authors, Yi Han et al, have efficiently tried to minimize the number of attacker VM to be co resident with the target VM. They have discussed a secure policy which could increase the difficulty levels of the attackers trying to achieve co resident attacks. In the paper, [12] the authors have brought about ways to monitor the health of a VM. The health of the VM in terms of security is solely dependent on the VM's interactions with its counterparts in its environment.

C. Anomaly Detection and Defense Techniques in Clouds

Anomaly detection techniques have been always an ongoing research particularly for the security concerns in either a normal network topology or let it be a cloud architecture. Researchers are more concerned for any unwanted events in the form of malwares which are not expected in conformation to an expected pattern in the data pertaining to the network traffic ie data in transit, data in storage or data in use. The authors in [1] have precisely worked out with their survey on

the different anomalies for predicting, detecting and forecasting. Cloud specific threats have been focused by the authors of [18], [19] that required prior knowledge, thus making them to be unsuitable for online detection in the cloud environments. Anomaly detection at different layers of cloud was proposed by guan et al. Their approach could not be demonstrated in dynamic cloud infrastructures. Anomaly detection methods can be analyzed from the system logs of each VM and was brought about by [20].This method lacked to detect the malwares under a text based log data. The authors of [21] were able to propose a prototype in which anomaly detection was based on the performance metrics of CPU utilization, memory management, hardware software components in a cloud context.

The authors in [21] were able design an online adaptive anomaly detection (AAD) architecture which proposes to detect anomalies through the analysis of runtime and execution metrics. The authors have used the one class SVM (Support Vector Machines) algorithm. This mechanism could not discover the early attack attempts or unwanted events in the cloud environment. The authors Watson et al have proposed an adaptive detection approach which says, that they can respond to new threats in online and with minimal computation cost. Accordingly they have achieved a detection accuracy rate of above 90%.Our method closely resembles the work Watson et al which aims to self heal and recover the VMs which has been prone to attacks. These victim VMs can specifically be able to retrieve back to the state before being attacked and come back to the normal mode through the snapshots.

Previous studies have proposed various defense mechanisms for securing the virtual machines in the cloud environment. The Virtual machine introspection (VMI) [22] proposes to protect the virtualization environment by providing a novel virtualization security by monitoring the state of a guest VM with its own benefits such as higher level privileges, strong isolation and so on. Intrusion detection in the VMs is used with the help of the VMI technique by the authors of [23]. Moreover memory forensics [24] with the help of virtual introspection has been proposed by B.Hay et al. The problems faced by using the VMI is the semantic gap challenged by the authors of [2].The semantic gap problem is defined as the problem of procuring high-level OS semantics from low level bits and bytes in physical memory involving data structures and its process.

The authors Muhammad et al in [31] have described an machine to machine reputation system in which the collaborating systems monitors the communication behavior of other systems in the network. Specifically, a trustworthy score is computed by these reputation systems based on the past behavior so as to identify malicious nodes in the communication patterns. Similarly in [32], authors have proposed a collaborative spit detection system that enables the service providers to be in collaboration with the end users without any transfer of private data. They argue that the proposed system maintains the privacy of the users with the resultant of high true positive rate and ensuring small false positive rate. The PrivBox [33] is one such reputation systems which can ensure the rating by the users of cloud in an encrypted manner. Thus a trusted environment for the cloud users is provided by these methods in an untrusted overview of booming cloud technologies.

D. Decision making and Predictive Analysis

Decision making can be mainly focused as a rule based technique. It can also be implemented as reinforcement learning, queuing theory, control theory and time series analysis according to the authors in [25]. The rule based technique is a form of pure decision maker while other techniques take the both the forms of predictor and decision making. Any decision making algorithm can be organized as self-configuring, self-healing, self-optimizing and self-protecting. Self-configuring approaches dynamically get adapted to changing environments by using a set of already given policies in the knowledge base. Such changes could include any new inclusions in the VM instances, or a removal of any data in the VM instances, or any dramatic changes in the behavior of the given VM instance. Thus self-configuring approach can enable continuous progression and productivity of VM workloads in a secure manner.

A self-healing approach can discover and diagnose any VM instance malfunctioning. It could enable a corrective action model from a given set of policies in the knowledge base. Thus the corrective action model can alter its own state without affecting the other VM's in the environment. A self-optimizing approach could monitor and thus reallocate VM instances to improve the overall utilization to avoid unnecessary resource provisioning [25]. A self-protecting approach could detect, identify, anticipate any threats, may be a hostile behavior of a VM instance, which could include unauthorized access, Denial of service attacks, Side channel attacks, confidentiality breaches and so on. A self-protecting capability could enforce security and privacy policies in the cloud environment [26].

An autonomic secure predictive model is developed by using the algorithms of machine learning for training the dataset of the virtual machine. It composes of the autonomic manager, which in turn consists of the goals, feedback and policies, which could be used for the purpose of training data. Machine learning algorithms are classified as supervised and unsupervised learning algorithms. For our purpose of study, the supervised modeling is opted, where the datasets are provided to train the autonomic component and thus to get the desired output. Specifically, predictive models use statistics to predict the suited security deliverance of the underlying VM at work.

The definition of any machine learning process, according to [25], is it could be divided as three major components: (a) A random generator (b) A supervisor function (c) A machine learning component. Such a secure predictive model can be modeled as the machine learning element. The objective of the predicting component is to obtain the best approximate value of the supervisor's response. The supervisor's response is the selection of the best approximation of the given set of functions, based on a training set of 't' independent value sets, given as $(x_1,y_1)\ldots\ldots\ldots(x_t,y_t)$.

The predictive modeling makes use of the statistics and also classifiers to predict the necessary outcomes. In order to experiment with the predictive modeling, two criteria can be followed:

1. A well-defined parameterized model should be considered for the data, so that the learning algorithm does over fit.
2. Any variable of the data should be predictable as accurately as possible.

E. Challenges for inter virtual machine communication

The paper[29] talks about the inter virtual machine communications, which is defined as two processes on the same physical machine, but in different VMs, which wants to exchange data in some form. These two processes in each of the VM have to communicate through the standard network interface, the two VMs being on the same physical host. The major challenge in this approach is that data has to be encapsulated, addressed, transmitted and verified in the network stack and also through the virtualization layer. Study shows that in order to address the issue of IVMC, a shared memory can be made use of to tunnel the isolation boundaries. Isolation is an important property of virtual machines for securing VM's in remaining intact. This would in turn be a overhead, if the VM's need to communicate among themselves overriding the isolation property. The authors in [15] proposed the development of an autonomic resource provisioning for SaaS applications hosted in the clouds. They have paved the development of an autonomic management system with QoS requirements to maximizing the efficiency with the minimization of the cost of services.

A snapshot can be defined as the ability to have a recorded state of a running VM at any given moment of time. These snapshots can then be used for restoring back the VM, either from a stop mode or when it fails from any attacks. A snapshot copy can be done instantly and can be retrieved efficiently. Snapshot technologies are most commonly used to protect the data and to reduce the recovery time. In order to effectively make use of the snapshots for autonomic recovery and self-healing of a given VM, a thorough understanding of the snapshots was performed. A VM snapshot is typically formed using either a system or a disk formats, most generally used as backup for the VM restoration purposes. Such a backup copy is used to create an entire architectural copy as the restoration point of a system under attack or failure.

The normal size of each VM takes around 40GB and each snapshot may amount of 8GB of storage space [28]. So, in order to reduce the storage capacity of the snapshots only the differences in the state changes are maintained. The snapshots thus taken are to be stored on the backup server during the night hours, which can amount to lot of storage spaces, in spite of the less storage capacity. Our model proposes a novel architecture where instead of saving the snapshots in the backup server and restoring the VM under attack, the snapshot taken at the local system itself would retrieve back the VM once it is identified as a victim VM autonomically. Overhead of creating backup is avoided and the model is proved to be efficient. Thus, the VM snapshot retrieval is a lightweight version than any modifications to be done at the hypervisor level.

III. METHODOLOGY

In this paper, we present a way to retrieve the VMs under attack by detecting the anomalies and also discuss a mechanism to avoid these anomaly patterns again by using the machine learning algorithms of SVM, the Naïve Bayes and the decision tree algorithms. More specifically, we evaluate these algorithms for the different anomaly types. The malware samples used for this purpose are TeslaCrypt, DarkComet, Xtreme, CyberGate, and Zeus. The main contributions of this paper are

1. Experiments in this work are all done for autonomic prediction architecture.
2. Estimating the accuracy of the time-series prediction algorithm's with respect to the different snapshots taken from virtual machines.
3. Investigating the aspect of malware detection in the cloud oriented scenarios on the generated snapshots.
4. Introduction to the self-healing capability of the virtual machine under study.

The overall architecture of the proposed method is depicted in Fig (1), which is named as the VMSec Managed Architecture. It consists of the (i) VMSec Agent, which monitors and sends the status report to the autonomic manger. For this purpose, the nitro monitoring system is used. (ii) The autonomic manager which is enabled with the knowledge base of the behavioral analysis of a particular VM. It is a rule based system, which is enabled with all the policies. The pattern of the attack is identified and any mismatch data is present between the knowledge base and the monitored data from the status report generated by the VM monitor is taken into account by the autonomic manager. (iii) The decision maker has to now decide based on the output obtained by comparing the VM status report and the knowledge base. Detecting whether the VM is under attack has to be determined. Here, we make use of the machine learning algorithms, Naïve Bayes, the SVM and the Random Forests. (iv) If the VM is detected to be under attack and based on the severity of the attack, a self healing algorithm is used to recover the VM under attack.

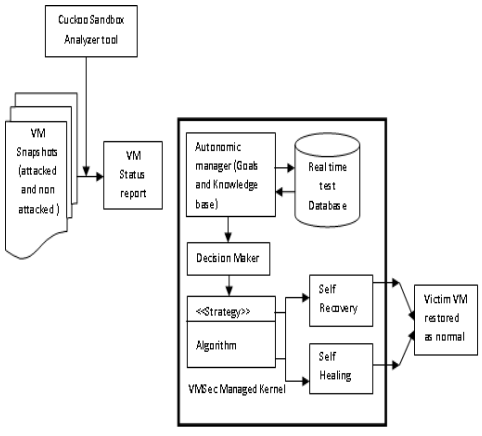

Fig. 1. VMSec Managed Architecture

TABLE I
SVM PARAMETER SETUP

C(Complexity)	KERNEL	regOptimizer
1.0	RBF	RegSMOimproved

This algorithm tries to retrieve back the VM to the most recent snapshot before the attack had taken place. This ensures that the VM does not have any trace of the attack. The machine learning algorithm may be embedded as an application on the hypervisor level to the running instances of the VMs. The proposed novel architecture uses the machine learning algorithms to classify the attacked and non-attacked snapshots.

Any machine learning algorithm has to approximate the best approximation of the autonomic manager's response. The failure to detect and classify has to be minimized and is given by loss, L (f(x, t), y), where t is the parameter of the classification function, given the input x. Therefore, the expected number of failures to be minimized is given by the empirical failure risk,

$$F_{emp}(x, t) = \frac{1}{n}\sum_{1}^{n} L(f(x_i, t), y_i) = training\ failures \quad (1)$$

In order to improve the overall accuracy of the machine learning algorithm, the overall failures pertaining to testing also has to minimized and is given by

$$F(x, t) = \int L(f(x, t), y)dP(x, y) = testing\ failures \quad (2)$$

Here, P(x, y) is the probability of the joint distribution function such that P (y|x) P(x) is the unknown data in the training dataset. A set of hyperplanes is defined to minimize the training failures and the complexity features, defined as f(x) = (w. α(x)) + h:

$$\frac{1}{n}\sum_{i=n}^{n} L(w.\alpha\, x_i + h, y_i) + ||w||^2 \text{ , } w.r.t\ min_i|w.x_i| = 1 \quad (3)$$

Where, w is a set of weights, h is the threshold value and α is the kernel function used in the SVM algorithm. We take the SVM machine learning algorithm as it performs well with good accuracy and is also more effective. However with large training set, the time taken for training the data may be quite high.

The naïve bayes algorithm uses the concept of class probabilities and conditional probabilities. The probabilities is calculated as probability of a randomly selected data that belongs to a class with the bayes theorem indicated as

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \quad (4)$$

The class with the highest probability is selected as the result by comparing the probabilities that belongs to all the class.

The time taken to restore the victim VM from the condition of under attack has to be performed from a working point in time when the VM data is consistent and thus to ensure that whatever applications that are used could communicate with each and running. This can be given by the metric recovery time objective (RTO) that defines the maximum amount of time required to restore a VM after a crash.

$$RTO = MTTD + MTTR \quad (5)$$

Where MTTD is the Mean Time to Detect, defined as the time taken to detect the malwares quickly and early, so that the victim VM could be fixed as soon as they occur, thus preventing the system failures. The MTTR (Mean Time to recover) is the time taken to predict the victim VM under attack and thus taking the preventive actions.

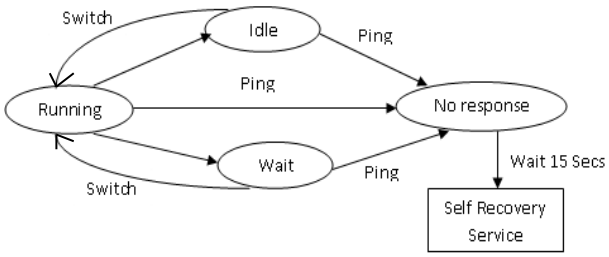

Fig. 2. State diagram of a typical Virtual Machine

A VM snapshot operation creates files .vmdk, -delta.vmdk, .vmsd, and .vmsn files. For the feature extraction phase of the snapshot dataset the delta files .vmdk are taken and the API calls are considered which are taken as states of the VM during the execution of the malware. The API calls reflect any state changes that happen in the operating system, files, registry, mutexes and processes. Each unique API call can be represented as numeric values, so amounts as a good criterion for the feature set.

A. Self-Recovery Algorithm

1. Start
2. Input: Input: VM's in a physical machine,VM_1, VM_2…….VM_n(VMware environmental setup)
3. For i = 1 to n
4. Ping each VM continuously, say some ms.
   (i) Monitor each VM Status VM_i for any response.
5. If no response from the VM,
   (i) Generate a status report from the cuckoo sandbox (malware detection).
   (ii) Find out the victim VM.
   (iii) Power off the running VM under attack.
   (iv) Obtain the VM snapshots of this particular victim VM.
6. From all the delta snapshot files {s_1, s_2…s_n} from the victim VM, generate the API Calls return codes.
7. Input to a machine learning algorithm and classify the attack files from the benign files.
8. Make an alert the Virtual Machine is under attack.
9. End

B. Self- Heal Algorithm

1. Start

2. From the set of non attacked VM snapshots delta files $\{s_1, s_2 \ldots s_n\}$,
   (i) Select the most appropriate snapshot, i.e. the first snapshot that was taken just before the malware attack, with respect to the VM system time.
3. Roll back to this selected snapshot instantaneously.
4. Power on the VM and resume the process autonomically from this selected snapshot.
5. End

IV. Experimental Setup

In this section we present the detailed workflow of the proposed architecture. Each of the virtual machine created in the VMware workstation has the following specifications,

TABLE II
VM Specifications

Parameter	Specifications
CPU	1 virtual CPU core 3.2 GHz
Memory	512 MB
Hard disk	40 GB
Network	1 Gbps Ethernet Interface
Operating System	16.4 Desktop Ubuntu
Qemu KVM	2.4.50
VM Manager	libvirt 1.2.20
Malware Analysis Tool	Cuckoo Sandbox
Penetration Testing Software	Metasploit framework
Memory Snapshot Feature Extraction	DECAF

A metasploit framework was used to penetrate attacks into the VM and the attacked VM snapshots were generated. For the unattacked VM snapshots the VM was restored back to the main base saved state, which is before the penetration of the attack. A careful malware analysis was done on the VMs. In order to extract the features from the VM snapshots to be given as input to the machine learning algorithms, the API calls are used as the features to be given as input. API calls form one of the features of the cuckoo sandbox among many others such as mutexes, registry keys, files, IP addresses and the DNS queries. These API calls are represented as a combination matrix consisting of the frequency of the failed APIs, successful APIs and the response return codes [34].

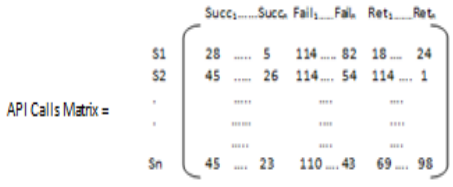

Fig.3. API Calls Matrix

Here, in the API calls matrix, the rows represent the VM snapshot samples, the columns $Succ_1 \ldots Succ_n$ represent the number of times each API call was made in $[Succ_1 \ldots . Succ_n]$. The total number of API calls made is given by 'n'. Similarly failed API calls are given as $fail_1 \ldots .$a $fail_n$ column which indicates the number of times the API calls failed and the number of response return codes of the API Calls is represented as $Ret_1 \ldots . Ret_n$ [34]. The VM snapshot images were analyzed from DECAF (Dynamic Executable Code Analysis Framework), which is a binary analysis framework based on qemu [30]. The API calls are obtained from the API tracer plug-in of DECAF. The data consists of two classes: attacked snapshots and the unattacked snapshots features. The number of features generated was too large. Thus from almost 4578 features, some 206 features were selected by a wrapper selection feature method. The boruta package was used for this purpose. A combination matrix was represented with all the features. In order to evaluate our algorithm, the VM snapshot dataset is randomly spilt up in 2/3 ratio of the collected data as the training data and the testing data.

Overall Accuracy: The overall accuracy A can be measured as the percentage of the correctly classified predictions of normal snapshots to the total number of snapshots. It is given as

$$A = \frac{Total\ number\ of\ VM\ snapshots\ correctly\ predicted}{Total\ number\ of\ VM\ snapshots} \quad (7)$$

V. Results and Discussions

We discuss the results of the assessment of the three implemented machine learning algorithms, namely the support vector machine (SVM), the naïve bayes algorithm, the random forests. As previously mentioned, we have spilt the features from the API calls of the memory snapshot features in to the training dataset and the testing dataset.

TABLE III
Analysis of Machine Learning Algorithms in the Training Phase

Parameters/ Malware Samples	SVM		Naïve bayes		Random Forests	
	VM snapshot correctly Classified	VM snapshot incorrectly Classified	VM snapshot correctly Classified	VM snapshot incorrectly Classified	VM snapshot correctly Classified	VM snapshot incorrectly Classified
Benign	57	8	35	27	55	10
TeslaCrypt	39	10	30	15	45	2
Zeus	35	14	29	8	35	10
Xtreme	32	5	34	7	33	5
CyberGate	38	2	36	2	37	3
DarkComet	76	4	48	2	48	4

Evaluated training phase: The table shows the results generated for the error rates pertaining to the training phase of the machine learning algorithms and a comparative graph showing the correct classifications of the snapshot files.

Evaluated testing phase: The table shows the results generated for the error rates pertaining to the testing phase of the machine learning algorithms.

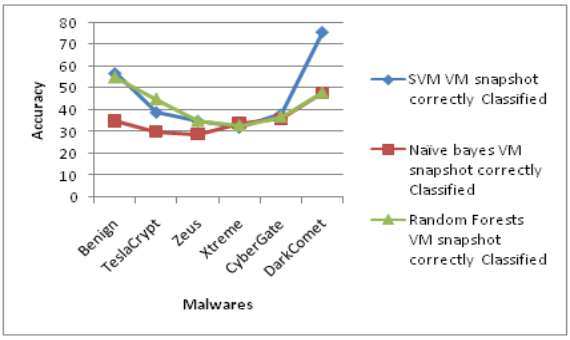

Fig. 3. Comparative analysis of correctly classified snapshots (training)

TABLE IV
ANALYSIS OF MACHINE LEARNING ALGORITHMS IN THE TESTING PHASE

Parameters/ Malware Samples	SVM		Naïve bayes		Random Forests	
	VM snapshot correctly Classified	VM snapshot incorrectly Classified	VM snapshot correctly Classified	VM snapshot incorrectly Classified	VM snapshot correctly Classified	VM snapshot incorrectly Classified
Benign	69	5	63	21	67	8
TeslaCrypt	48	7	45	13	56	0
Zeus	44	10	41	12	48	7
Xtreme	56	3	53	4	69	2
CyberGate	64	3	57	1	72	0
DarkComet	87	2	71	0	76	2

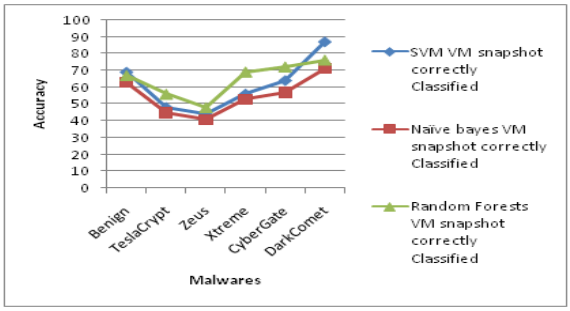

Fig. 4. Comparative analysis of correctly classified snapshots (testing)

TABLE V
CONFUSION MATRIX

	False positives	False Negatives
SVM	22	0
Naïve bayes	18	43
Random Forests	0	6

Results comparing the overall accuracy of the machine learning algorithms to detect the un- attacked and the attacked snapshots correctly.

From the results, we find that on an average the random forests have responded well to the snapshot data in classifying the malwares from the benign samples. This algorithm resulted in a high accuracy with a good performance, but as can be noticed from the number of false negatives which has obtained to be 0, whereas the random forests have resulted in 6 false negatives.

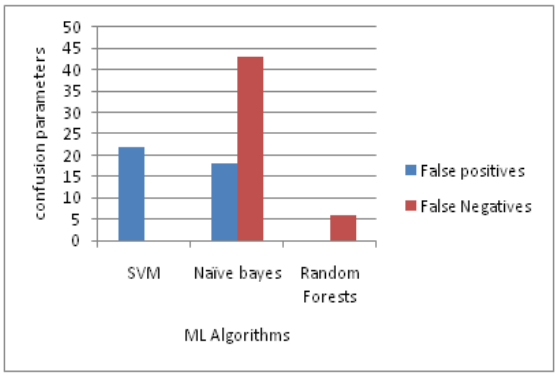

Fig. 5. Comparative analysis of confusion matrix

TABLE VI
OVERALL ACCURACY OF MACHINE LEARNING ALGORITHMS

ML Algorithms/ Samples	SVM	Naïve Bayes	Random Forest
Benign	92.9	57.6	96
TeslaCrypt	86	77	99
Zeus	75.5	71	88.5
Xtreme	93.2	92.1	99
CyberGate	96.4	96.8	99
DarkComet	97	99	99

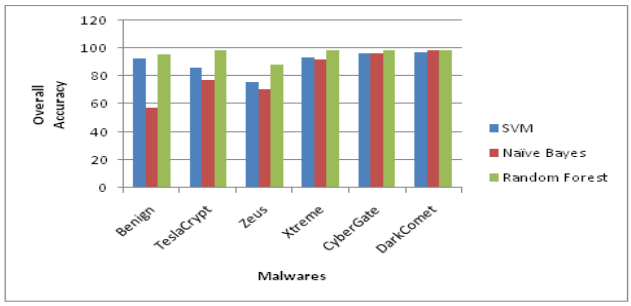

Fig. 6. Comparative analysis of overall accuracy of machine learning algorithms

VI. CONCLUSIONS AND FUTURE WORK

This paper proposes the self-recovery and self-healing of a Virtual Machine under attack, with machine learning algorithms to classify and identify the attacks under different malware conditions. Some files were introduced as benign, simple .exe files with the malware infected files using the metasploit penetration software. From the samples of snapshot delta files, the API calls features were extracted and given as input to the SVM, the naïve bayes and the random forests algorithm. The API calls are considered because of their actual behavior in the respective files. The algorithms classified the dataset and the performances between the different algorithms are plotted with respect to the attacked VM snapshots and the non-attacked VM snapshots. From the generated results and the confusion matrix, it was found that the SVM out performs due to the reduction in the generation of the false negatives. The lowest accuracy was achieved by the naïve bayes algorithm (82.25 %), followed by SVM (90.16 %) and the random forests (96.75 %). Based on the generated results, it is recommended that we make use of the random forests

algorithm as it showed a higher accuracy scope, nevertheless it generated in 6 false negatives. The SVM algorithm generates 0 false negatives, so it recommended in implementing this algorithm for further analysis and future work. A state diagram of the virtual machine with respect to the response time is depicted for restoration. In order to retrieve the VM from the local system itself and to avoid the over head of the backup server in a cloud scenario, this approach could save time and the network congestion caused in the backup servers.

The paper demonstrated results based on the design concept, for our future work, several improvements could be brought out related to the practical implementation of the project.

1. From the classified set of the non-attacked VM snapshots, the best approximation VM snapshot can be found be identified with respect to time.
2. Restoration to be made possible with rollback to this identified non attacked VM snapshot and run autonomically.
3. A wider dataset could be proposed with all possible types of malwares.
4. This approach to be implemented in real time and to know the running status of the retrieved VM under attack.

REFERENCES

[1] Michael R.Watson, Noor-ul-hassan Shirazi, Angelos K.Marnerides, Andreas Mauthe and David Hutchison "Malware Detection in Cloud Computing Infrastructures", IEEE Transactions on Dependable and Secure Computing,PP.192-205,2016, DOI: 10.1109/TDSC.2015.2457918.

[2] Bryan.D.Payne,Martim Carbone, Monirul I.Sharif,Wenke Lee, "Lares: An architecture for secure active monitoring using virtualization",Proc 29th IEEE symposium security privacy PP.233-247,May 2008, DOI:10.1109/SP.2008.24

[3] Ristenpart Thomas, Eran Tromer, Hovav Shacham, Stefan Savage "Hey,you Get off my cloud: Exploring Information leakage in third party compute clouds," proc.16 ACM conf. Computer and Communication security, PP 199-212,2009.

[4] Zhang Y, Ari Juels, Michael K.Reiter, Thomas Ristenpart,"Cross-VM side Channels and their use to extract private keys," proc,DOI:10.1145/2382196.2382230.

[5] Marnerides A.K, M.R.Watson, N.Shirazi, A.Mauthe and D.Hutchison," Malware analysis in cloud computing: Network and system characteristics," IEEE Globecom, PP.305-316, 2013, DOI: 10.1109/GLOCOMW.2013.6825034.

[6] Jamkhedar P, J Szefer, D Perez Botero,"A framework for realizing security on demand in cloud computing," proc. IEEE 5th Conf.Cloud Computing technology and science, PP.371-378, 2013.

[7] Win T.Y, H.Tianfield, Q.Mair," Virtualization Security Combining mandatory access control and virtual machine introspection", proc. IEEE/ACM 7th Intl. Conf. Utility Cloud Computing (UCC), PP.1004-1009, Dec 2014,DOI: 10.1109/UCC.2014.165.

[8] Gruschka N and M.Jensen,"Attack surfaces: A taxonomy for attacks on cloud services", in Cloud Computing (CLOUD), IEEE 3rd International Conference, PP. 276-279, 2010, DOI: 10.1109/CLOUD.2010.23.

[9] Christodorescu M, R.Sailer, D.L.Schales, D.Sgandurra, and D.Zamboni," Cloud security is not (just) virtualization security: A short paper,"ACM workshop on cloud computing security, ser. CCW", New York, NY, USA, PP.97-102, 2009.

[10] Yi Han, Tansu Alpcan, Jeffrey Chan, Christopher Leckie," Using Virtual Machine Allocation Policies to Defend against Co-Resident Attacks in Cloud Computing", IEEE Transactions on Dependable and Secure Computing, Vol 14 Issue 1,PP 98-107, 2017, DOI: 10.1109/TDSC.2015.2429132.

[11] Yi Han, Tansu Alpcan, Jeffrey Chan, Christopher Leckie and Benjamin I.P.Rubinstein," A game Theoretical Approach to defend against Co-resident Attacks in Cloud Computing: Preventing Co-residence Using Semi-Supervised Learning", IEEE Transactions on Information Forensics and Security,Vol.11,no.3,PP 556-570,2016, DOI: 10.1109/TIFS.2015.2505680.

[12] Tianwei Zhang, Ruby B. Lee, Princeton University", Monitoring and Attestation of virtual machine security health in cloud computing, Journal IEEE Micro, Vol 36, Issue 5, PP 28-37, 2016, DOI: 10.1109/MM.2016.86.

[13] Preeti Mishraa, E.S.Pillai, Vijay Varadharajan, Udaya Tupakula"Intrusion Detection techniques in cloud environment: A survey", Journal of Network and Computer Applications, Vol.77, PP: 18-47,Jan 2017,DOI:10.1016/j.jnca.2016.10.015.

[14] Brian Hay, Kara Nance, "Forensics examination of volatile system data using virtual introspection", ACM SIGOPS Operating System Review, Vol 42, No3 PP.74-82, 2008, DOI: 10.1145/1368506.1368517.

[15] Rajkumar Buyya, Rodrigo N. Calheiros and Xiaorong Li "Autonomic cloud computing: Open Challenges and architectural elements", International conference of emerging Applications of Information technology PP 3-10, 2012,DOI:10.1109/EAIT/2012.6407847.

[16] Chandola.V, A.Banerjee, and V.Kumar "Anomaly detection: A Survey," ACM Computing Surveys (CSUR), Vol.14, no.3, p.15, 2009, doi:10.1145/1541880.1541882.

[17] Jicheng Shi, Xiang Song, Haibo Chen, binyu Zhang ,"Limiting Cache based side-Channel in multi-tenant cloud using dynamic page coloring" Proc.41st Annual IEEE/IFIP international conference on dependable systems and network workshops(DSN-w 2011) pp.194-199,2011, DOI: 10.1109/DSNW.2011.5958812.

[18] Qiang Guan and Song Fu,"Adaptive anomaly identification by exploring metric subspace in cloud computing infrastructures," in Reliable distributed Systems(SRDS), 2013 IEEE 32nd International Symposium on IEEE,2013,pp. 205-214, DOI: 10.1109/SRDS.2013.29.

[19] Bahl P, R.Chandra, A. Greenberg, S.Kandula, D.A.Maltz, and M.Zhang," Towards highly reliable enterprise network services via inference of multi-level dependencies", in ACM SIGCOMM Computer Communication Review, Vol. 37,no 4 ACM,2007,pp.13-24, DOI:10.1145/1282427.1282383.

[20] Lee J.H, M.W.Park, J.H.Eom and T.M.Chung," Multi-level intrusion detection system and log management in cloud computing", in Advanced Communication Technology (ICACT), 2011 13th international conference on IEEE, 2011, pp. 552-555.

[21] Pannu H.S, J.Liu, and S.Fu," AAD: Adaptive anomaly detection system for cloud computing infrastructures," Reliable Distributed Systems, IEEE Symposium on Vol.0,pp.396-397,2012, DOI: 10.1109/SRDS.2013.29.

[22] Win T.Y, H.Tianfield, Q.Mair," Virtualization Security Combining mandatory access control and virtual machine introspection", proc.IEEE/ACM 7th Intl. Conf. Utility Cloud Computing(UCC),PP.1004-1009,Dec 2014, DOI: 10.1109/UCC.2014.165.

[23] Garfunkel T and Mendel Rosenblum,"A VM introspection based architecture for intrusion detection", Proc.18th annual Network Distribution systems Secure symposium, PP 191-206, 2003.

[24] Hay B, K. Nance, and M. Bishop, "Live Analysis: Progress and Challenges," Security & Privacy, IEEE, vol. 7, no. 2, pp. 30–37, 2009, DOI: 10.1109/MSP.2009.43.

[25] Ali Y.Nikravesh, Samuel A.Ajila,Chung Horng Lung," An autonomic prediction suite for cloud resource provisioning", Journal of cloud computing advances, Systems and applications, 2017, DOI:10.1186/s13677-017-0073-4.

[26] Zizhong Chen, Member, Jack Dongarra, "Highly Scalable Self-Healing algorithms for High performance scientific Computing" IEEE Transactions on Computers, Vol.58, No.11, November 2009, DOI: 10.1109/TC.2009.42.

[27] Radu S.Pirscoveanu, Steven S.Hansen, Thor.M.T.Larsen, Matija Stevenovic, Jens Myrup Pedersen, Alexandre Czech "Analysis of Malware behavior: Type classification using machine learning" IEEE International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), PP 1-7, 2015, DOI: 10.1109/CyberSA.2015.7166115.

[28] Yaodong Yang, Bo Mao, Hong Jiang, Yuekun Yang, Hao Luo, and Suzhen Wu "SnapMig: Accelerating VM Live Storage Migration by Leveraging the Existing VM Snapshots in the Cloud", Transactions on Parallel and Distributed Systems, pp 437-441, 2018, DOI:10.1145/1655008.1655022.

[29] Carl Gebhardt and Allan Tomlinson, "Challenges for Inter Virtual Machine Communication", Royal Holloway, University of London, Technical Report, 2010.

[30] Aravind Prakash, Eknath Venkataramani, Heng Yin "On the Trustworthiness of Memory Analysis—An Empirical Study from the

Perspective of Binary Execution", IEEE Transactions on Dependable and Secure Computing, Volume: 12, Issue: 5, PP 557-570, 2015, DOI: 10.1109/TDSC.2014.2366464.

[31] Muhammad Ajmal Azad, Samiran Bag, Feng Hao,"M2M-REP: Reputation of Machines in the Internet of Things", ACM, ARES '17, Reggio Calabria, Italy, 2017, DOI: 10.1145/3098954.3098976.

[32] Muhammad Ajmal Azad, Ricardo Morla," Rapid detection of spammers through collaborative information sharing across multiple service providers", Future Generation of Computer, Elsevier, 2018, DOI:10.1016/j.future.2017.12.026.

[33] Muhammad Ajmal Azad, Samiran Bag, Feng Hao,"PrivBox: Verifiable Decentralized Reputation System for Online Marketplaces", Elsevier FGCS, 2018, DOI:10.1016/j.future.2018.05.069.

Linda Joseph is an assistant professor at Hindustan Institute of Technology and Science, Chennai, Tamil Nadu, India. She did her masters at Noorul Islam College of Engineering,Anna University, Tamil Nadu. She is now with the Department of Computer Science and Engineering. Her research interests are network security, Artificial Intelligence, Cloud Computing and security.

Dr Rajeswari Mukesh, Professor and head, is with the Department of Computer Science and Engineering, Hindustan Institute of Technology and Science, Chennai, Tamil Nadu, India. She did her master's from Madras University and completed her doctorate from Jawaharlal Nehru Technological University. She has published about 25 research papers.A total of 40 citations have been cited for her research papers.Her research interests include cyber security,Cloud security,Theoritical Computer Science and IoT.