International Journal of Education and Science Research Review
E-ISSN 2348-6457
Volume-9, Issue-4 July-August-2022
P-ISSN 2349-1817
www.ijesrr.org
Email- editor@ijesrr.org

PERCEPTION OF MALWARE CRITICIZE ON VIRTUAL MACHINES FOR A SELF-HEAL APPROACH IN CLOUD COMPUTING USING VM SNAPSHOTS

Phadtare Tushar Tulsidas ,
Research Scholar,
School of Science & Technology, Glocal University Saharanpur (U.P)

Dr. Praveen Kumar,
Research Supervisor,
School of Science & Technology, Glocal University Saharanpur (U.P)

ABSTRACT

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 selfheal. The self-healing approach with machine learning algorithms can determine new threats with some prior knowledge of its functionality. 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.

Key words—Cloud Computing; VM Snapshots; Machine Learning Algorithms; API Calls; Self-Healing.

INTRODUCTION

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 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.

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.

METHODOLOGY

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.

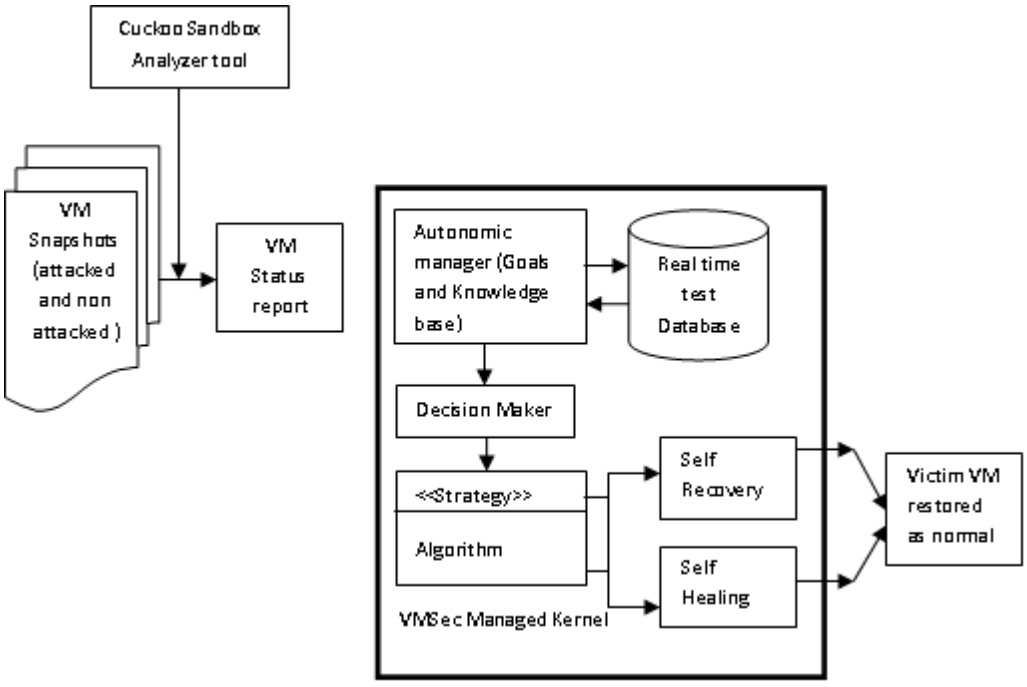

Fig. 1. VMSec Managed Architecture

TABLE I

SVMPARAMETER 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,

$$Femp(x, t) = 1/n\Sigma\, n/1\ (f(xi\ ,\ t),\ yi) = training\ failures \quad (1)$$

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

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

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:

$$1n\Sigma\ (w.\ \alpha\ ni{=}n\ xi + h,\ y\ ) + ||w||2\ ,\ w.r.t\ mini|w.xi| = 1 \quad (3)$$

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

$$P\ (X|Y) = (Y|X)P(X)P(Y)\ldots\ldots(4)$$

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.

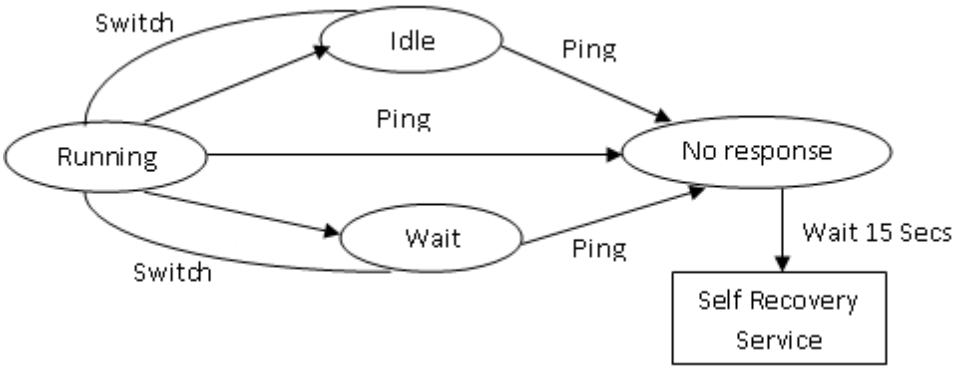

Fig. 2. State diagram of a typical Virtual Machine

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,VM1, VM2…….VMn(VMware environmental setup)
3. For i = 1 to n
4. Ping each VM continuously, say some ms.

(i) Monitor each VM Status VMi 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 {s1, s2…sn} 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 {s1, s2…sn},

(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

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	**512MB**
Hard disk	**40GB**
Network	1 Gbps Ethernet Interface
Operating System	16.4 Desktop Ubuntu
Qemu KVM	2.4.50
VM Manager	libvirt 1.2.20
Malware AnalysisTool	Cuckoo Sandbox
Penetration TestingSoftware	Metasploit framework
Memory SnapshotFeature 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 .

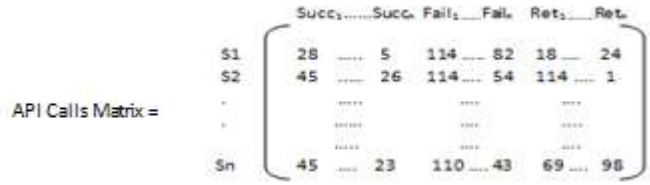

Fig.3. API Calls Matrix

Here, in the API calls matrix, the rows represent the VM snapshot samples, the columns Succ1…Succn represent the number of times each API call was made in [Succ1….Succn]. The total number of API calls made is given by 'n'. Similarly failed API calls are given as fail1…..a failn 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 Ret1….Retn [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

$$A = Total\ number\ of\ VM\ snapshots\ correctly\ predicted / Total\ number\ of\ VM snapshots \quad (7)$$

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.

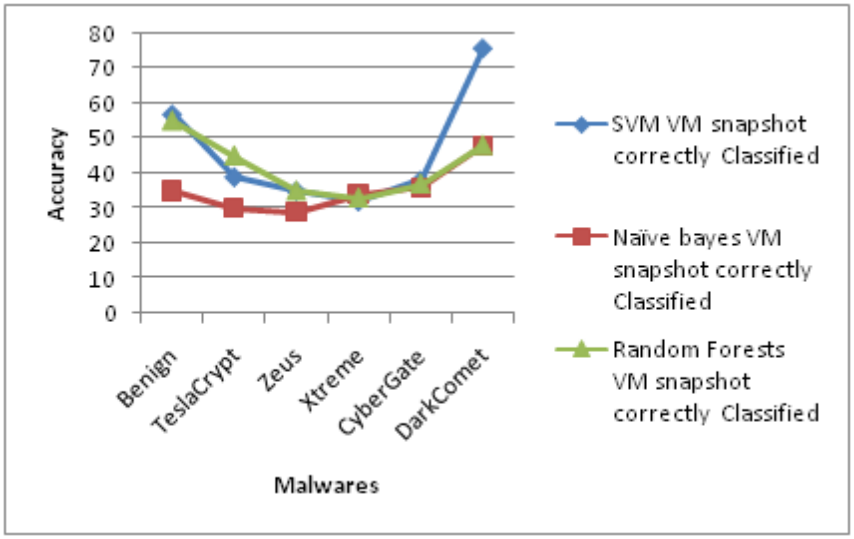

Fig. 3. Comparative analysis of correctly classified snapshots (training)

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

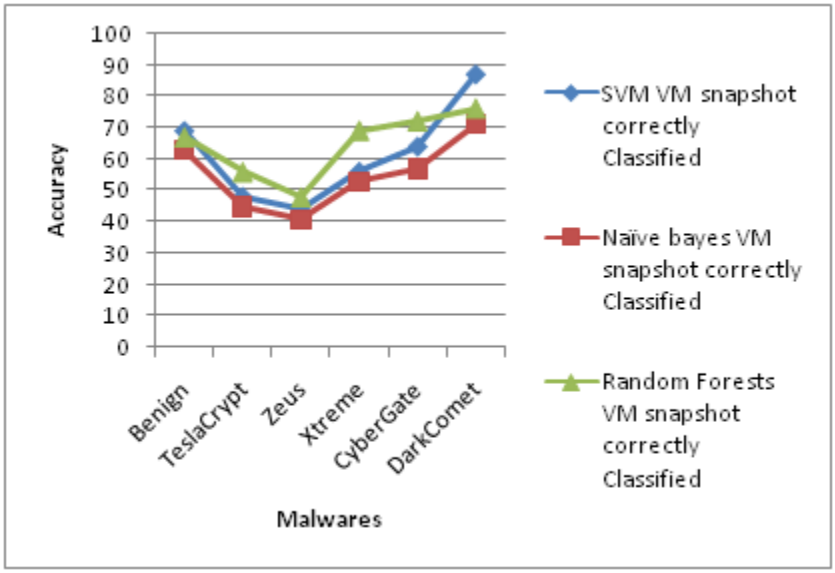

Fig. 4. Comparative analysis of correctly classified snapshots (testing)

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.

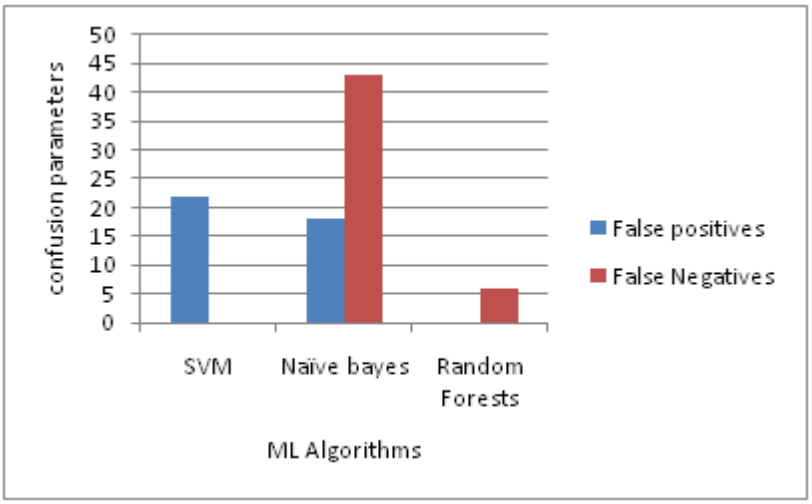

Fig. 5. Comparative analysis of confusion matrix

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

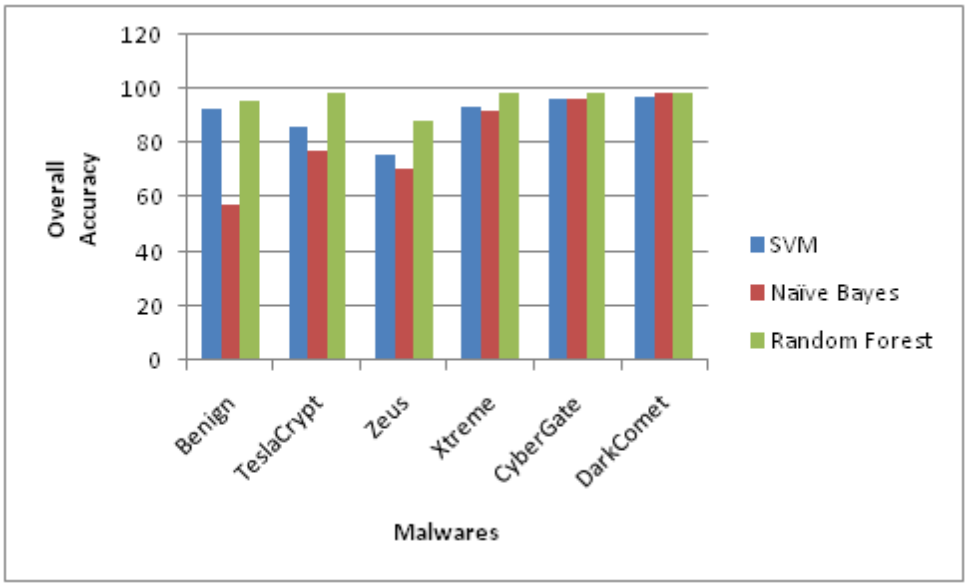

Fig. 6. Comparative analysis of overall accuracy of machine learning Algorithms

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 Coresident 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.