Privacy Preserving Of Cloud Storage From Multimalware Using SVM

P. Prathap[1], V. Sivasangar[2], M. Venkadesan[3]
[1,2,3]Student Members, Department of Computer Science
Arasu Engineering College, Kumbakonam, Tamil Nadu.

Abstract— *Malware is a pervasive problem in distributed computer and network systems. Cloud services are prominent within the private, public and commercial domains. Many of these services are expected to be always on and have a critical nature; therefore, security and resilience are increasingly important aspects. In order to remain resilient, a cloud needs to possess the ability to react not only to known threats, but also to new challenges that target cloud infrastructures. In this paper we introduce and discuss an online cloud anomaly detection approach, comprising dedicated malware detection components. Analyzing and restricting files with malware code into cloud will greatly reduce cloud service attacks. This flexible detection system capable of detecting new malware strains with no prior knowledge of their functionality or their underlying instructions. Simply, this system detects malware attacks at upload/download and service level.*

Keywords—*Svm; anomaly; Kmeans; Recovery; Malware; uplod/download;*

I. INTRODUCTION

Cloud Computing refers to manipulating, configuring, and accessing the applications online. It offers online data storage, infrastructure and application. Cloud data enters are beginning to be used for a range Of always-on services across private, public and commercial domains These need to be secure and resilient in the face of challenges that include cyber attacks as well as component failures and mis-configurations. Viruses used to be the province of hackers whose aim was to demonstrate their technical prowess by defacing web sites. Today, security attacks are becoming much more sophisticated and infinitely more dangerous. Increasingly, they are being utilised by criminal gangs and state-sponsored organisations to steal information, secrets, intellectual property and money… as well as to cause disruption to business operations and industrial processes. Everybody in business uses documents of one form or another to share information with colleagues, customers and business partners… but not nearly enough people know that documents are a major source of infection. Professional, ruthless and well-organised criminal gangs are now targeting the actual documents used by businesses, organisations and governments. This 'New World' needs a new approach because hackers are taking advantage of three self-evident truisms.

II. BACKGROUND& RELATED WORK

A. *Cloud Computing*

Cloud computing offers different service models that allow customers to choose the appropriate service model that fits their environment needs, Cloud service models are software as a service (SaaS), Platform as a service (PaaS), and Infrastructure as a service (IaaS)

Software-as-a-service (SaaS): The consumer uses the provider's applications, which are hosted in the cloud. For example, Salesforce.com CRM Application.

Platform-as-a-service (PaaS): Consumers deploy their own applications into the cloud infrastructure. Programming languages and application development tools used must be supported by the provider. For example, Google Apps.

Infrastructure-as-a-service (IaaS): Consumers are able to provide storage, network, processing, and other resources, and deploy and operate arbitrary software, ranging from applications to operating systems.

B. *Anomaly Detection in Clouds*

Anomaly detection (also outlier detection) is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset. Typically the anomalous items will translate to some kind of problem such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are also referred to as outliers, novelties, noise, deviations and exceptions. An anomaly-based intrusion detection system is an intrusion detection system for detecting both network and computer intrusions and misuse by monitoring system activity and classifying it as either normal or anomalous. The classification is based on heuristics or rules, rather than patterns or signatures, and attempts to detect any type of misuse that falls out of normal system operation. This is as opposed to signature-based systems, which can only detect attacks for which a signature has previously

been created. In order to positively identify attack traffic, the system must be taught to recognize normal system activity. The two phases of a majority of anomaly detection systems consist of the training phase (where a profile of normal behaviors is built) and testing phase (where current traffic is compared with the profile created in the training phase).

C. Malware Detection in Clouds

Malware is a pervasive problem in distributed computer and network systems. Identification of malware variants provides great benefit in early detection. Cloud services are prominent within the private, public and commercial domains. Many of these services are expected to be always on and have a critical nature; therefore, security and resilience are increasingly important aspects. In order to remain resilient, a cloud needs to possess the ability to react not only to known threats, but also to new challenges that target cloud infrastructures. Nowadays computer systems and communication infrastructures are likely to be influenced by different types of attacks so there is need to put further efforts for improving the software trust. Therefore, there will be increase in necessity in the coming time, as the number of software developers and applications will likely grow very significantly. However there is more need to adopt some better techniques which can ensure the malware code detection efficiently by testing method over a large set of malicious executables. Malware the malicious software is used to gather sensitive information, disrupt computer operation or to have access to secure computer systems. It can be appear in the form of coding, scripts, active contents and other software. Malware is the term used to refer a variety of forms of intrusive software. Detecting new and unknown malware is a major challenge in today's software security profession. A lot of approaches for the detection of malware using data mining techniques have already been proposed. Majority of the works used static features of malware. However, static detection methods fall short of detecting present day complex malware. Although some researchers proposed dynamic detection methods.

III. **ALGORITHMS**

Classification and Clustering

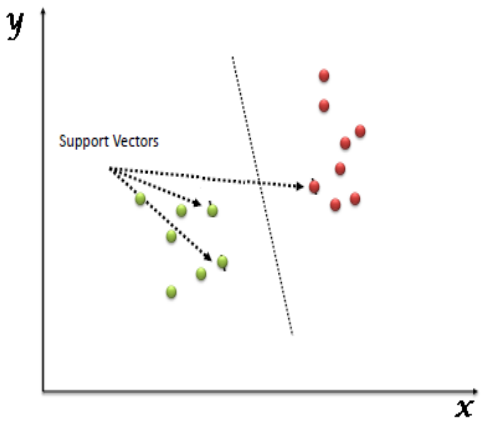
In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering and is often used in industrial applications either when data are not labeled or when only some data are labeled as a preprocessing for a classification pass.

SVM for classification

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well. Support Vectors are simply the co- ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper- plane/ line).In machine learning and statistics, **classification** is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. An example would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition. Classification is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance. An algorithm that implements classification, especially in a

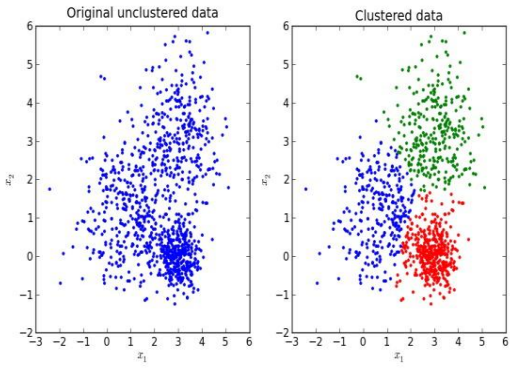
concrete implementation, is known as a **classifier**. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category.

Terminology across fields is quite varied. In statistics, where classification is often done with logistic regression or a similar procedure, the properties of observations are termed explanatory variables (or independent variables, repressors, etc.), and the categories to be predicted are known as outcomes, which are considered to be possible values of the dependent variable. In machine learning, the observations are often known as *instances*, the explanatory variables are termed *features* (grouped into a feature vector), and the possible categories to be predicted are *classes*. SVM has a technique called the **kernel trick**. These are functions which takes low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called kernels. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then find out the process to separate the data based on the labels or outputs you've defined. It works really well with clear margin of separation. It is effective in high dimensional spaces. It is effective in cases where number of dimensions is greater than the number of samples. It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. Support vectors are the data points that lie closest to the decision surface. SVMs maximize the margin around the separating hyper plane.



K-means for clustering

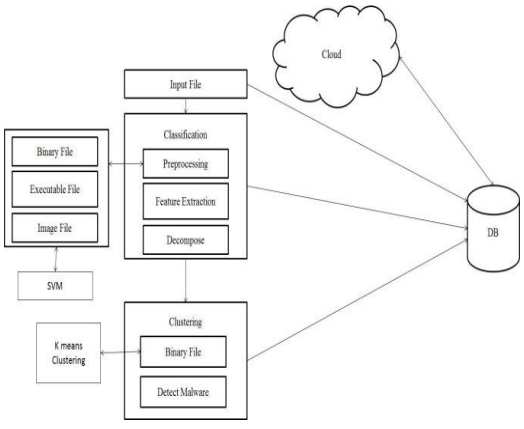
Clustering is a process of partitioning a set of data (or objects) into a set of meaningful sub-classes, called clusters. Help users understand the natural grouping or structure in a data set. Used either as a stand-alone tool to get insight into data distribution or as a preprocessing step for other algorithms'-**means clustering** is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. *K-means* clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Verona cells. K-means (Macqueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroid, one for each cluster. This centroid should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. The K-means algorithm involves randomly selecting initial centroid where **K** is a user **defined** number of desired clusters. Each point is then assigned to a closest centroid and the collection of points close to a centroid form a cluster. **Cluster** analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (**clusters**). This clustering algorithm separates data into the best suited group based on the information the algorithm already has. Data is separated in different clusters, which are usually chosen to be far enough apart from each other spatially, in Eucledian Distance, to be able to produce effective data mining results. Each cluster has a center, called the **centroid**, and a data point is clustered into a certain cluster based on how close the features are to the centroid.



IV. MALWARE DETECTION IN CLOUD

This project focus on detecting malwares while upload or download a file to cloud and detect and remove attacks that affect cloud services. It analyze the document and recover or restrict document with malware. This system works like prevention mechanism. It can be appear in the form of coding, scripts, active and other software. We can detect malicious executable by looking at the frequency analysis of byte code in a file. Finally implement classification and clustering algorithm , which uses byte sequences in a file as features. Simply, this system detects malware attacks at upload/download and service level. This project focus on the malware code is embedded into a document files like word,pdf,text files. The malware code can be detected by using the machine learning algorithms. To use svm for classifying the input file into a normal file or malware affected file.The malware file is detected to apply the clustering algorithm to find the type of the malware code in the file.The file is identified as malware file to use recovery mechanism to remove the malware code in the file to give a original file.The file as contains maximum threshold level of malware code than to reject the document. The document is recovered to be uploaded or downloaded. The executable file is converted into a document format and then to be uploaded to the cloud. This project is overcome this problem to check the uploaded document is the original document or not. These systems detect the malware code in the upload and download level.

System Architecture



V. RESULTS

The experiments we present in this section test the detection aspects of the System and Network Analysis Engines. Given the fact that both engines perform online anomaly detection under the one-class SVM formulation we initially present our results related to the computational cost of the online training and testing of the algorithm, sincethey affects the overall response of the real-time detection process. We subsequently present our assessment on detecting the malware strains as well as the DDoS attacks. In addition, we further present a comparison between the detection accuracy obtained when using a joint dataset (i.e. composed of both system and network features) with a feature set that strictly considers network-based features.

Training and Classification Cost Analysis

The required time for training the one-class SVM classifier on various sizes of training datasets. For the sake of completeness we have experimented with a range of sizes having, as a maximum, a large dataset consisting up to 80,000 rows. This was in order to demonstrate the extremely small impact that training and classification have in our actual experimental conditions. The dataset used in the experiments was around 200 samples, which resulted in a training time of between 2 and 10ms, which is not possible to measure reliably using our tools. Hence, the dataset was extrapolated up to 80,000 entries in order to produce an observable trend. Considering feature extraction takes in the order of seconds to complete,17 the time taken to train the classifier is negligible, especially since it is only required to take place once during the lifetime of the classifier. In scenarios where the role of a server changes significantly and frequently the classifier would need to be retrained in order to produce a model of normal behaviour that sufficiently characterises the new normal behaviour patterns. Though, in our experience, in such cases it is more usual to replace a VM with the new version by swapping one for the other, rather than altering it in place.

This allows the new image to be profiled and a more complete model of the new normal to be established before deployment.Classification could also potentially hold up the process of obtaining a class for a particular vector and, like training, is dependent on dataset size. However the time taken to produce a class is also negligible with respect to the time taken to obtain the feature vector itself, despite the fact that classification is carried out on every sample vector.

VI. CONCLUSION

Malware can be classified according to similarity in its flow graphs. This analysis is made more challenging by packed malware. In

this paper we proposed fast algorithms to unpack malware using application level emulation, and perform malware classification using the edit distance between structured control flow graphs. We implemented and evaluated a prototype. It was demonstrated that the automated unpacking system was fast enough for desktop integration. The automated unpacking was also demonstrated to work against a promising number of synthetic samples using known packing tools, with high speed. To detect the completion of unpacking, we proposed and evaluated the use of entropy analysis. It is shown that our system can successfully identify variants of malware.

VII. FUTURE ENHANCEMENT

The proposed approach uses single-linkage agglomerative algorithm to classify malware. However, when a new malware is captured, cluster system has tore- run clustering algorithm using whole samples. The future work might be developing an incremental clustering algorithm to improve the efficiency of cluster system. The proposed approach is not to replace classification of binary malware. It is a complementary approach to understand whole behavior of attackers. In future work must overcome the classification of binary malware.

VIII. REFERENCES

[1]. Marnerides, C. James, A. Schaeffer, S. Sait, A. Mauthe, andH. Murthy, "Multi-level network resilience: Traffic analysis, anomaly detection and simulation," ICTACT J. Commun. Technol., Special Issue Next Generation Wireless Netw. Appl., vol. 2, pp. 345– 356, Jun. 2011.

[2]. J. P. G. Sterbenz, D. Hutchison, E. K. C, etinkaya, A. Jabbar, J. P. Rohrer,M. Sch€oller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," Comput.Netw., vol. 54, no. 8, pp. 1245–1265, Jun. 2010.

[3]. A. K. Marnerides, M. R. Watson, N. Shirazi, A. Mauthe, and D. Hutchison, "Malware analysis in cloud computing: Network and system characteristics," in Proc. IEEE Globecom Workshop, 2013, pp. 482–487.

[4]. M. R. Watson, N. Shirazi, A. K. Marnerides, A. Mauthe, and D. Hutchison, "Towards a distributed, self- organizing approach to malware detection in cloud computing," in Proc. 7th IFIP/IFISC IWSOS, 2013, pp. 182–185.

[5]. M. Garnaeva. Kelihos/Hlux Botnet Returns with New Techniques. Securelist [Online]. Available: http://www.securelist.com/en/blog/655/Kelihos_Hlux_bot net_returns_with_new_techniques, Feb. 2012.

[6]. H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang, "On the analysis of the zeus botnet crimeware toolkit," in Proc. 8th Annu. Int. Conf. Privacy Security Trust, Aug. 2010, pp. 31–38.

[7]. T. Brewster. (2014, Jul. 11). GameOver Zeus returns: Thieving malware rises a month after police actions, Guardian Newspaper [Online]. Available: http://www.theguardian.com/technology/2014/jul/11/game over-zeus-crimina l-malware-police-hacking

[8]. A.K. Marnerides, P. Spachos, P. Chatzimisios, and A. Mauthe,"Malware detection in the cloud under ensemble empirical model decomposition," in Proc. 6th IEEE Int. Conf. Netw. Comput., 2015, pp. 82–88.

[9]. L. Kaufman, "Data security in the world of cloud computing," IEEE Security Privacy, vol. 7, no. 4, pp. 61– 64, Jul. 2009.

[10]. M. Christodorescu, R. Sailer, D. L. Schales, D. Sgandurra, and D. Zamboni, "Cloud security is not (just) virtualization security: A short paper," in Proc. ACM Workshop Cloud Comput. Security, New York, NY, USA, 2009, pp. 97–102.

[11]. N. Gruschka and M. Jensen, "Attack surfaces: A taxonomy for attacks on cloud services," in Proc. IEEE 3rd Int. Conf. Cloud Comput., Jul. 2010, pp. 276–279.

[12]. Y. Chen, V. Paxson, and R. H. Katz. (2010, Jan.). Whats new about cloud computing security?. EECS Department, Univ. of California. Berkeley, Tech. Rep. UCB/EECS-2010-5. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS- 2010-5.html

[13]. G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "Bothunter: Detecting malware infection through ids-driven dialog correlation," in Proc. 16th USENIX Security Symp. USENIX

[14]. Security Symp., Berkeley, CA, USA, 2007, pp. 12:1– 12:16.

[15]. M. Bailey, J. Oberheide, J. Andersen, Z. Mao, F. Jahanian, and J. Nazario, "Automated classification and analysis of internet malware," in Proc. 10th Int. Conf. Recent Adv. Intrusion Detection, 2007, vol. 4637, pp. 178–197.

[16]. Mazzariello, R. Bifulco, and R. Canonico, "Integrating a network ids into an open source cloud computing environment," in Proc. 6th Int. Conf. Inf. Assurance Security, Aug. 2010, pp. 265–270.

[17]. S. Roschke, F. Cheng, and C. Meinel, "Intrusion detection in thecloud," in Proc. 8th IEEE Int. Conf. Dependable, Autonomic Secure Comput., Dec. 2009, pp. 729–734.