ANOMALY DETECTION ON RUN TIME MACHINES USING OC-SVM

Lavanya N, Deepika N

Student, Sr. Assistant Professor
Computer Science and Engineering,
New Horizon College of Engineering, Bangalore, India

Abstract : Intrusion Detection Systems (IDSs) be worn to recognize and report unauthorized otherwise suspicious computer or network actions. Host-based IDSs, the thought of this paper, are planned to observe the host system deeds, while network-based IDSs check network traffic in favor of multiple hosts. Agreeing to their detection techniques, IDSs can also be off the record into misuse detection or anomaly detection conditional to whether the intrusion patterns are recognized or not through the design phase [10].

***IndexTerms* - Intrusion Detection System, Host based IDSs, Network based IDSs, Misuse Detection, Anomaly Detection.**

I. INTRODUCTION

Intrusion Detection Systems (IDSs) be worn to recognize and report unauthorized otherwise suspicious computer or network actions. Host-based IDSs, the thought of this paper, are planned to observe the host system deeds, while network-based IDSs check network traffic in favour of multiple hosts. Agreeing to their detection techniques, IDSs can also be off the record into misuse detection or anomaly detection conditional to whether the intrusion patterns are recognized or not through the design phase [4]. Misuse detection approach glimpse meant for predefined patterns or signatures linked to acknowledged attacks, and consequently they are able to accomplish a high level of detection truthfulness. Even if, misuse detection techniques cannot determine unnamed attacks for which signatures have not been standing apart yet (zero-day attacks) or well-known actions, which are able to distinction their signatures with every execution (polymorphic tacks) [1].

Anomaly detection measures are artistic of detecting novel attacks, despite the fact that they are lying face down to make a large number of false alarms due regularly to the dilemma in procurement a illuminating account of normal accomplish of the system [5]. The anomaly detectors will consequently make an treacherous number of false alarms (by misclassifying rare normal events as anomalous), which could not make the grade the fidelity of the anomaly detection system, mainly that the base-rate of normal minutes be in command of the anomalous ones. Host-based anomaly detection systems normally monitor for vital conflicts in operating system calls, as they offer a entry between user and kernel modes [8]. Understandings accessible that the historical order of system calls delivered by a process to request kernel services is real in valuable normal process behavior. There are many details that make intrusion detection the key parts in the whole attack system. First, many of the old-style organisms and desires have been built and developed without taking safety particularly into account. Second, computer systems and applications may have errors or bugs in their plan that could be donations by burglars to attack the systems or applications. Hence, the deterrent skill may not be as valuable as expected [11].

II. LITERATURE SURVEY

Quite a lot of unsupervised anomaly detection dealings have been useful to intrusion detection to progress IDSs concert in all levels such as in clustering, features selection and classifications. Erected on the prior figure of the a variety of unsupervised anomaly detection systems. The dissimilarity reviews the pros and cons of each one [6].

Concerning machine learning skills for intrusion detection can repetitively shape the model based on the training data set, which holds data instances that can be labelled by revenue of a usual of attributes (features) and associated labels. The attributes can be of limitless sorts such as categorical or continuous.

The fragility of knowledge base detection modus operandi. Anomaly detection comprehends supervised techniques and unsupervised techniques [7]. Many procedures were worn to become conscious superior outcomes for these techniques. This paper suggests a notion of machine learning techniques for anomaly detection. The trials conventional that the supervised learning methods knowingly exceed the unsupervised ones if the test data contains no unidentified doses. Among the supervised ways and means, the best concert is completed by the non-linear methods, such as SVM, multi-layer perceptron and the rule-based means. Modus operandi for unsupervised such as K-Means, SOM, and one class SVM achieved well again recitation over the other skills even though they fluctuate in their competences of detecting all attacks classes expertly [5].

III. ANOMALY DETECTION TECHNIQUES

Concerning machine learning skills for intrusion detection can repetitively shape the model based on the training data set, which holds data instances that can be labelled by resources of a typical of attributes (features) and allied labels. The attributes can be of immeasurable sorts such as categorical or continuous [9].

Intrusion detection systems are habitually used still with other defense systems such as loom control and validation as a second expose line to defend information systems. There are many niceties that make intrusion detection the key parts in the entire attack system. First, many of the old-style organisms and desires have been built and residential without taking wellbeing extremely into account. Second, computer systems and applications may have errors or bugs in their plan that could be assistance by burglars to attack the systems or applications. Hence, the deterrent skill may not be as valuable as estimated [11].

3.1Nature of Input Data

A decisive facade of any anomaly detection technique is the nature of the input data. Input is on average a collection of data instances. Each data instance can be described by means of a set of attributes. The attributes can be of tainted types such as binary, categorical or continuous. To every data instance bravery necessitate of only one attribute (univariate) or multiple attributes (multivariate) [3].

In the occurrence of multivariate data cases, all attributes aptitude be of same type or might be a unify of different data types. Input data can also be categorized based on the relationship at hand surrounded by data instances.
Supreme of the existing anomaly detection techniques pact by record data (or point data), in which no relationship is unspoken among the data instances [7].

3.2 Type of Anomaly

A significant facade of an anomaly detection technique is the nature of the much loved anomaly. Anomalies can be classified into following three categories:

1) Point Anomalies: If an detached data instance be capable of be cautious as anomalous with respect on the way to the rest of data, then the instance is dubbed as a point anomaly. This is the humblest type of anomaly and is the emphasis of majority of research on anomaly detection.

2) Contextual Anomalies: If a data instance in a exact context, then it is named as a contextual anomaly or conditional anomaly [1].

Intrusion Detection Systems (IDSs) are worn to distinguish and legend unauthorized or suspicious computer or network measures. Host-based IDSs, the awareness of this paper, are projected to supervise the host system actions, at the same time as network-based IDSs observes network traffic for several hosts. Allowing to their detection techniques, IDSs can in addition be categorized into misuse detection or anomaly detection depending on whether the intrusion patterns are acknowledged or not all over the design phase [3]. Misuse detection approaches momentary look for predefined patterns or signatures associated to customary attacks, and therefore they are able to achieve a high level of detection accuracy. Despite the fact that, misuse detection techniques cannot ascertain un-identified attacks for which signatures have not been standing apart yet (zero-day attacks) or well-known actions, which are able to discrepancy their signatures with each implementation (polymorphic tacks) [10].

Relating machine learning skills for intrusion detection can repetitively shape the model based on the training data set, which holds data instances that can be labelled by means of a normal of attributes (features) and associated labels. The attributes can be of immeasurable sorts such as categorical or continuous [12].

IV. Implementation

4.1 Description of Strategies

A recognized portrayal of both term vector weighting strategies. The wished-for approach for efficient drawing out of feature vectors based on variable length n-grams is described next. Let T = o_1, o_2 , . . . , o_L be a trace of system call annotations (o_i) of length L , generated by a process with an alphabet of size m = |∑ | (unique) system calls. The anthology of K traces that are generated by the progression (or method) of importance and then provided for deceitful the anomaly detection system is denoted by T = {T_1 , . . . , T_K} [1].

The binary term vector, φ(T), maps each trace T ∈ T into a vector of size m system calls, T → $\varphi(T)_{o \in \sum}$, where each term or system call $o_i \in \sum$ in the vector is assigned a binary flag depending on its manifestation (one) or not (zero) in the trace T . The term vector can be weighted by the term frequency (tf):

$$f_{tf}(o,T) = \text{freq}(o_{i,}); i = 1,\ldots\ldots, m \qquad (1)$$

where freq be the number of times system call oi appears within T , normalized by L (the whole number of system calls within T).

The inverse document frequency (idf) is projected near increase (or decrease) the weights of terms so as to are rare (or common) across every bit of documents. The term vector weighted beside the tf.idf is consequently given by:

$$\phi_{tf.idf}(o,T,T) = K\,[df(O_\iota)]\ \text{freq}(o_\iota); \iota = 1, \ldots, \mu \qquad (2)$$

where the document frequency df (o_i) be the number of traces T_k be the anthology T of size K so as to contains system call o_i [5].

For a definite process, the size of the dictionary (D), which is the size of the feature vectors, depends on the alphabet size m , the sliding window size N and the uniformity of the process. The feature vectors embrace the n-grams extracted from the sliding window of size N, weighted by their frequencies of occurrences in the trace. The value of N is a user- defined parameter with the goal of influences the detection power and the size of the feature vectors. A small N value is at all times desirable since it results in smaller feature vectors, and thus allows faster detection and response during operation. Besides, it can in actual fact tell apart the anomalies from the normal sequences of the attack trace [14].

4.2 ADFA-LD Dataset

The making of system call datasets for manipulative and evaluating host-based ADSs is classically performed in two phases. Normal system call traces are opening unruffled all the way through normal operation of the process or system within a secured environment. These traces are assumed attack-free and worn for training the anomaly detectors. The testing traces are generated near collecting the system calls beginning the same host as being under attacks. These attack traces include both normal and anomalous sequences for testing, though it is difficult to cut off the sign of an attack inside the trace. Therefore, during testing, the complete attack trace or the collection of attack traces is well thought-out as one anomaly [5].

	Number of traces	Number of system calls
Training data (Normal)	833	308,077
Testing data (Normal)	4373	2,122,085
Attack data (Anomalous)	746	3,17,388
Total	5,952	2,747,550

The ADFA-LD dataset be generated by means of a modern operating system and servers attacked near exploiting a variety of (publicly known) security vulnerabilities. As described with the authors, the ADFA- LD be generated with a flattering patched Ubuntu Linux 11.04 operating system by an Apache 2.2.17 web server, PHP 5.3.5 server side scripting engine, TikiWiki 8.1 content management system, FTP server, MySQL 14.14 database management system and an SSH server [5]. Ordinary system call traces be generated as of the host system through normal user activities, such since web browsing and Latex document preparation. These traces be collected using Linux audit daemon (auditd 5), an auditing framework designed for collecting and tracking security audit trails. The ADFA-LD dataset comprises 833 normal traces for training and 4373 normal traces for testing.

The dataset comprises 746 traces generated on or after 60 different attacks, belonging to six types of attack vectors. These attacks are launched next to a certified penetration tester next to the system, by modern penetration testing tools like Metasploit 6 framework [2]. The attack framework includes client and server side attacks in addition to social engineering techniques. The client side attacks purposeful on initiating connections from the target machine, for example by a exploiting poisoned executable or Trojaned programs. The TikiWiki vulnerability is one more vector of attack worn to upload a copy of connection to the attacking system by the Meterpreter, an improved functionality command shell provided by the Metasploit framework [9].

4.3 OC-SVM based detection approach

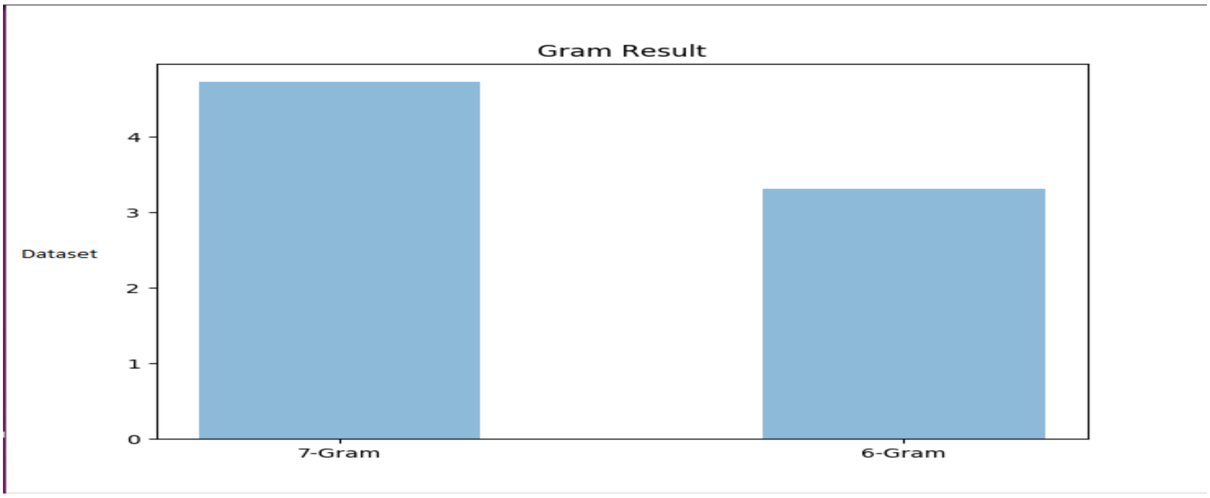
One-Class Support Vector Machine (OC-SVM) is a influential and frequently used machine learning scheme in various domains. This formulation is comparable to that of the two-class SVM, on the other hand the OC-SVM considers to the normal training data are far as of the origin, as the anomalous data lie in the neighbourhood of the origin. Readily available is an equivalent formulation that uses a hypersphere to portray the data in the feature space, and tries to locate the smallest hypersphere that contains most of the data. The hyperplane that separates normal from anomalous data corresponds to the classification rule $f(v) = w^T \cdot v + b$*[3]*, which represents the dot product of the normal vector (w) and a bias term (b).

The optimization problem consists consequently of verdict the rule f with a maximal geometric margin. This classification rule can be old to classify a test input v *test* as anomalous if $f(v^{test}) < 0$; or normal otherwise. In practice, there is forever a trade-off flanked by maximizing the distances of the hyperplane from the origin and the number of normal data points controlled in the other region unconnected by the hyperplane. The detachment from the hyperplane to a test example v^{test} could also be worn as score or degree of membership to the normal or anomalous class; these scores are then use to generate the ROC curves. Our proposed anomaly detection system is based on OC-SVMs detectors trained using our sets of feature vectors (*VN*3, and *VN*6) extracted as of the (normal) training traces. Training OC-SVMs is done using LIBSVM, 9 a library for support vector machines. In our experiments, we have qualified and compared the performance of OC-SVMs using two commonly used kernels: linear kernel and Gaussian kernel.

V. RESULTS AND DISCUSSION

The results obtained by the projected ADS using OC-SVMs with linear and Gaussian kernels, skilled on our feature vectors (VN3, and VN 6) for detecting system call anomalies in the ADFA-LD datasets. As assured beforehand, the results of OC-SVMs trained using the term vector (with tf and tf.idf weights) are also accessible for comparison [7].

Initially we need to upload the dataset which needs to be tested against the training dataset and the anomaly dataset. As the feature will be extracted and fed to detector, once we upload the dataset from the file it will fed and the results will be displayed [11].



Comparison between 6 gram and 7 gram dataset

The above graph represents the time taken for the previous algorithm that is by taking the 6 grams as the input as well as the current enhancement that is by taking 7 grams as input. The amount of dataset considered by the 7 grams are in large numbers when compared to the previous algorithms [12].

REFERENCES

[1] D.E. Denning, An intrusion detection model, in: Proceedings of the Seventh IEEE Symposium on Security and Privacy, 1986, pp. 119–131.

[2] J. McHugh, A. Christie, J. Allen, Defending yourself: the role of intrusion detection systems, IEEE Softw. 17 (5) (20 0 0) 42–51, doi: 10.1109/52.877859.

[3] S. Axelsson, Intrusion detection systems: a survey and taxonomy, in: Tech. Rep., Chalmers University, 2000, pp. 99–115.

[4] H.-J. Liao, C.-H.R. Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: a comprehensive review, J. Netw. Comput. Appl. 36 (1) (2013) 16–24, doi: 10.1016/j. jnca.2012.09.004.

[5] G.F. Cretu, A. Stavrou, M.E. Locasto, S.J. Stolfo, A.D. Keromytis, Casting out demons: sanitizing training data for anomaly sensors, in: IEEE Symposium on Security and Privacy, 2008. SP 2008., IEEE, 2008, pp. 81–95 .

[6] C. Gates , C. Taylor , Challenging the anomaly detection paradigm: a provocative discussion, in: in: Proceedings of the 2006 Workshop on New Security Paradigms, NSPW ’06, ACM, New York, NY, USA, 2006, pp. 21–29.

[7] R. Sommer, V. Paxson, Outside the closed world: On using machine learning for network intrusion detection, 2010. 0, 305–316.

[8] S. Forrest , S.A. Hofmeyr, A. Somayaji, T.A. Longstaff, A sense of self for Unix processes, in: Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, 1996, pp. 120–128 .

[9] Abdullah, B., Abd-algafar I., Salama G. I. and Abd-alhafez A. Performance Evaluation of a Genetic Algorithm Based Approach to Network Intrusion Detection System, Proceedings of 13th International Conference on Aerospace Sciences and Aviation Technology (ASAT-13), Military Technical College, Cairo, Egypt, 2009;1-5.

[10] Anderson, J. P. Computer security threat monitoring and surveillance. Technical Report, Fort Washington, PA, USA.,1980;9-11.

[11] Anderson, D., Frivold, T. and Valdes, A. Next-generation intrusion detection expert system (NIDES): A summary Technical Report SRI–CSL–95–07,Computer Science Laboratory,SRI International, May 1995.

[12] Beghdad, R. Critical study of neural networks in detecting intrusions. Computers and Security, 27(5-6): 2008;168–175.

[13] Devikrishna, K. S. and Ramakrishna , B. B. .An Artificial Neural Network based Intrusion Detection System and Classification of Attacks", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Jul-Aug 2013, 3(4): 1959-1964.

[14] Denning, D. E.. An intrusion detection model, IEEE Transactions on Software Engineering, CA,. IEEE Computer Society Press;1987.