Designing Anomaly Detection System for Cloud Servers by Frequency Domain Features of System Call Identifiers and Machine Learning

Waqas Haider, Jiankun Hu, and Nour Moustafa(✉)

School of Engineering and Information Technology,
UNSW Canberra, Canberra, Australia
nour.moustafa@unsw.edu.au

Abstract. The protection of operating systems from the current cyber threats has paramount importance. This importance is reflected by the functional dependency of any known or unknown cyber-attack upon the machines operating system. In order to design an anomaly detection system to protect an operating system from unknown attacks, acquiring comprehensive information related to running activities is the first crucial step. System call identifiers are one of the most reflective logs related to running activities in an operating system. Number of system call identifiers based host anomaly detection systems have been presented from the last two decades by using logs as raw system call identifiers. However, due to the stealth and penetration power of the unknown attacks, there is a need of acquiring and investigating more possible logs from machines operating system for the reliable protection. In this paper, firstly we apply the sine and Fourier transformation to the short sequence of system call identifiers, in order to model the frequency domain feature vector of any running activity at the cloud server. Second, different machine learning algorithms are trained and tested as anomaly detection engine using frequency domain transformed feature vectors of the short sequence of system call identifiers. The proposed work is evaluated using recently released intrusion detection systems data-set i.e., NGIDS-DS alongside two other old data-sets for comparative purposes. The experimental results indicate that the frequency domain feature vectors of short sequence of system call identifiers have comparatively superior performance than raw short sequence of system call identifiers, in detecting anomalies and building normal profile.

Keywords: HIDS · HADS · Operating system security
Intrusion detection

1 Introduction

Although firewall technology [1] and access control mechanisms [2,3] can provide strong cybersecurity protection, the wide spread of advanced hacking tools

J. Hu et al. (Eds.): MONAMI 2017, LNICST 235, pp. 137–149, 2018.
https://doi.org/10.1007/978-3-319-90775-8_12

plus the daunting number of combinations of vulnerable points from software, operating systems and networking protocols has rendered it impossible to prevent all cyberattacks, in particular zero-day attacks [4,5,5,6]. Today hacking groups which may be sponsored by the governments or individuals can design and launch the type of cyber-attacks which are capable of penetration through network defense zone [7–12]. Such type of attacks are only visible at machines operating system while performing the malicious tasks. The global cyber threats reports alarming the fact that, the target of these attacks are critical machines. For example, storage and processing servers in the cloud computing environment are prime targets, because at present corporate enterprises utilize cloud computing infrastructure for data to analyze, interpret and to make proactive decisions to keep the business competitive [13]. Further, most of the storage and processing servers in cloud computing infrastructure are comprised of Linux and Unix based operating systems [14]. During operation, the patterns of any legitimate or anomalous events in these operating systems are present at the kernel level system call identifiers sequences. Each system call identifiers sequence represents the relation of activity resource consumption at the software level with the time [15].

Detecting anomalous behavior in critical cloud servers has been observed to be a serious problem for the cloud computing service providers, due to the following two major reasons: (i) During the last two decades, number of system call identifiers based host intrusion detection systems are presented [16,17]. In these systems the researchers suggested to log raw system call identifiers as data source or spatial and domain knowledge based transformation of these identifiers as features. The spatial transformation means that, the length, data values, frequency and range of data values in a system call identifiers sequence [17], whereas domain knowledge based transformations means, transforming a raw system call identifier by considering its relation with activity purpose and resource. As the traditional components of an intrusion detection system are data source, feature construction and decision engine [16]. Critical cloud servers defense based on just raw or spatial representation of system call identifiers may results in the exclusion of other useful features in the final defense mechanism; and (ii) there is a trend in hacking industry to learn the state of the art defense mechanism and then design the attacks to break them [18]. In this regard, designing and developing cyber defense systems is observed to be an ongoing process [19]. Therefore, depending on just one type of logs i.e. raw or spatial representation of system call identifiers, can minimize the reliability factor.

In this paper, the two main contributions are as follows: (i) In order to explore the new features in the theory of host based anomaly detection systems, the short sequence raw system call identifiers are transformed into frequency domain by applying sine and Fourier transformation, and (ii) To evaluate and compare the capability of proposed frequency domain feature vectors as comprehensive reflection of normal activities including discrimination power for classifying normal and attack feature vectors, different machine learning algorithms as anomaly detection engine and recently released intrusion detection system

data-set i.e. NGIDS-DS [20] are used. The considered machine learning algorithms include, SVM with linear and radial base kernels, KNN and ELM. Although anomaly intrusion detection is virtually a classical classification problem where there exist many powerful machine learning algorithms [21–23], our focus is on the construction of new features as features play a critical role. The rest of the paper is organized as follows: the literature review is given in Sect. 2; the proposed work is given in Sect. 3; experimental results and discussion are provided in Sect. 4; and the concluded remarks are given in Sect. 5.

2 Literature Review

In this section, the existing host based anomaly detection systems based on system calls are analyzed and classified. The classification of these systems are based on how the feature vectors are constructed by the spatial transformation or domain knowledge based manipulation of raw system calls identification. For instance, pioneer researchers of this domain utilized the raw short sequences of system call identifiers as feature vectors [24]. Later, some researchers utilized the spatial transformation of raw system call identifiers sequence i.e. considering just most frequent, less frequent, maximum and minimum system call identifiers as feature vector [16,17,25]. In addition some researchers have utilized the domain knowledge to manipulate raw system call identifiers in order to construct feature vectors for the host activities [26–28].

The raw short sequence of system call identifiers based host anomaly detection techniques build a model for the sub-sequences of the normal traces, and in decision engine a test occurrence opposing considerably from the model established will be reflected as abnormal. For example, in pioneer host intrusion detection works by Forests [24,29], the feature matrix is constructed by sliding window of fixed length across the normal traces and at decision engine a trial trace comprising a percentage of mismatch away from a threshold is considered as abnormal. Tackling the long traces, Kosoresow et al. modified the look-ahead algorithms by calculating the divergences within small, fixed-length sectors of the traces [30]. Furthermore, at decision engine of the short sequence based techniques, statistical learning notions are widely adopted to predict the behavior (normal or abnormal) by summarizing the intrinsic associations concealed behind the normal traces. The example includes, artificial neural network (ANN) [31,32], SVM [33], hidden Markov model (HMM) [34,35] and semantic data mining [26].

In contrast with the raw short sequence of system call identifiers as features, in [16,17,25] the spatial transformation of raw system call identifiers are presented to construct feature vectors for the host activities. For example, in [16,17] a feature vector of a trace of the raw system call identifiers is constructed by just considering most frequent, less frequent, minimum, maximum and even/odd count of system call identifiers in terms of integer data. Similarly, in [25] for windows operating system, the count of key dll calls is used to construct the feature vectors of host activities. Moreover, in [26–28,36] the domain knowledge is considered to construct feature vectors by using system call identifiers. For example,

in [27] the authors suggested the criteria for the selection of a few system calls to conduct audit, which is based on attack domain knowledge. This approach has considered only those system call identifiers, which are assumed to be involved in privileged transition flows, during attack and normal behavioral scenarios. Similarly, in [28] the traces of system call identifiers are suggested to be represented with eight kernel modules. Further, in [36], a model is presented based on system calls arguments (i.e. execution path) and sequences incorporated with clustering. The key characteristic is the consideration of different ways of using a system call in a specific process as ingredient to construct a feature vector.

In the above discussion, it can be observed that, in order to ensure the reliability of host defense by countering the current threats there is still a need to investigate the novel and hidden features and as a addition in this work we proposed the sine and Fourier transformation at the raw system call identifiers to extract frequency domain features from host operating system. To the best of our knowledge, this work can open a new way may to investigate the application of frequency domain transformation to system call identifiers in building host defense.

3 Proposed Work

In this section, the proposed work is elaborated in terms of training and testing framework given in Fig. 1, for host based anomaly detection systems (HADS). The key contribution is the application of sine and Fourier transformation in the feature construction phase of the proposed HADS. At the feature construction phase of the existing HADSs, the spatial and domain knowledge transformations have been applied, therefore it is intended to investigate the applicability of sine and Fourier transformation as feature construction and later its impact in detecting host anomalies. Further, for performance comparison, at the decision

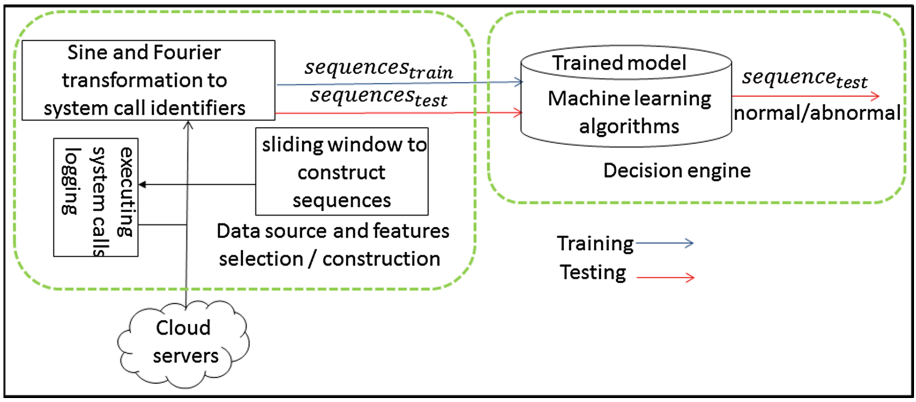

Fig. 1. Proposed HADS framework

engine of the proposed HADS different machine learning algorithms are configured independently such as SVM, KNN and ELM.

An online HADS starts its operation by first logging the comprehensive data from machines operating system. In the proposed HADS we are dealing with Linux or UNIX based operating systems where the system calls calling are considered to be the comprehensive audit data [13] that can be logged in an online manner as shown in Fig. 1. In addition most of the critical machines around the globe are comprised of these operating systems [37]. Once the data unit is logged, then a traditional HADS feature selection or construction mechanisms are triggered [38]. In the proposed HADS, we adopted first time the sine and Fourier transformation to transform the raw system call identifiers time domain signal (i.e. shown in Fig. 2) into frequency domain signal by utilizing the scheme in [39]. In order to log a unit data and to apply sine and Fourier transformation according to time, a sliding window with 1 s length is adopted i.e. each one second signal of system call identifiers is transformed in to frequency domain. Further, once the incoming system call identifier signal is transformed into the frequency domain, the feature vector is constructed. The formal description of the transformation process is elaborated as follows. First the input time domain signal of system call identifiers in 1 s is represented with sine transformation that is defined in Eq. (1). In Eq. (1), the variable x shows the system call identifier and $t = 1$ to T and T can be 1 s.

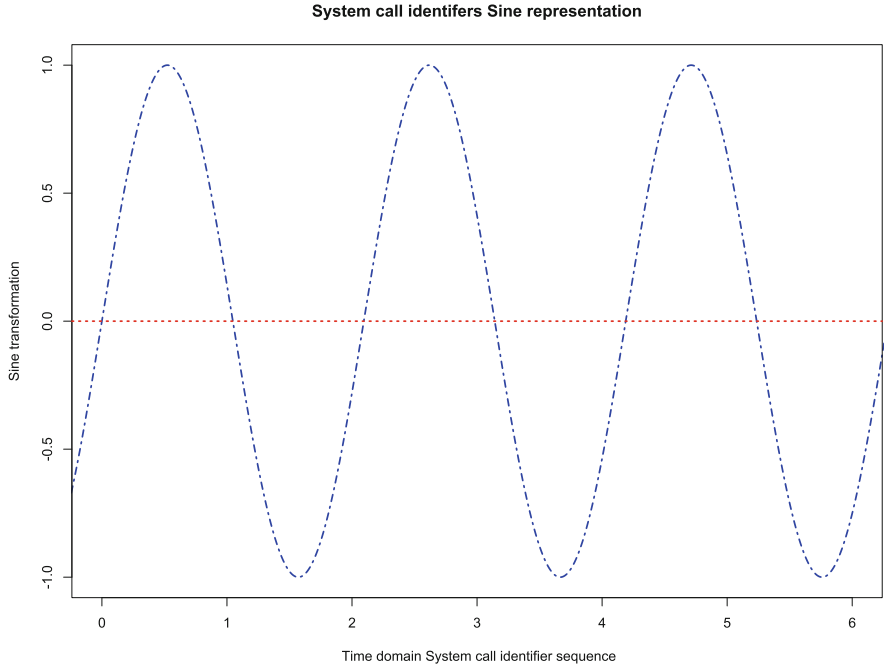

Fig. 2. Time domain representation of system call identifier sequence with sine

$$f(x) = sin(x)_t \tag{1}$$

For instance, the system call identifiers range is from 0 to 350 in considered version of the kernel of the Linux operating system (i.e. Ubunto 14.04), however this range can be vary depending the version of the kernel. The first 6 system call identifiers sine transformation is elaborated in Fig. 2. In the frequency transformation process, after sine conversion to input signal then the Fourier transformation is applied on sine transformed signal that is defined in Eq. (2) where for any real number ξ (i.e. the sine transformed values of system call identifiers), the independent variable x represents time (with SI unit of seconds) and the transform variable ξ represents frequency (in hertz). In Fig. 3, the Fourier transformation for the first 6 system call identifiers sine transformed signal is shown. Further, in frequency transformation process, the Fourier transformed components of the input signal are treated as the sequence or feature vector of the host activity in one second. These feature vectors are further utilized to train and test the adopted machine learning algorithms as anomaly detection engines for the host. The decision engine of the proposed HADS is configured with three machine learning algorithms respectively i.e. SVM, KNN and ELM. The purpose is to evaluate the performance in terms of accuracy and error, of frequency domain feature vectors in building normal profile and to classify anomalous feature vectors. The parameters of the selected machine learning algorithms which

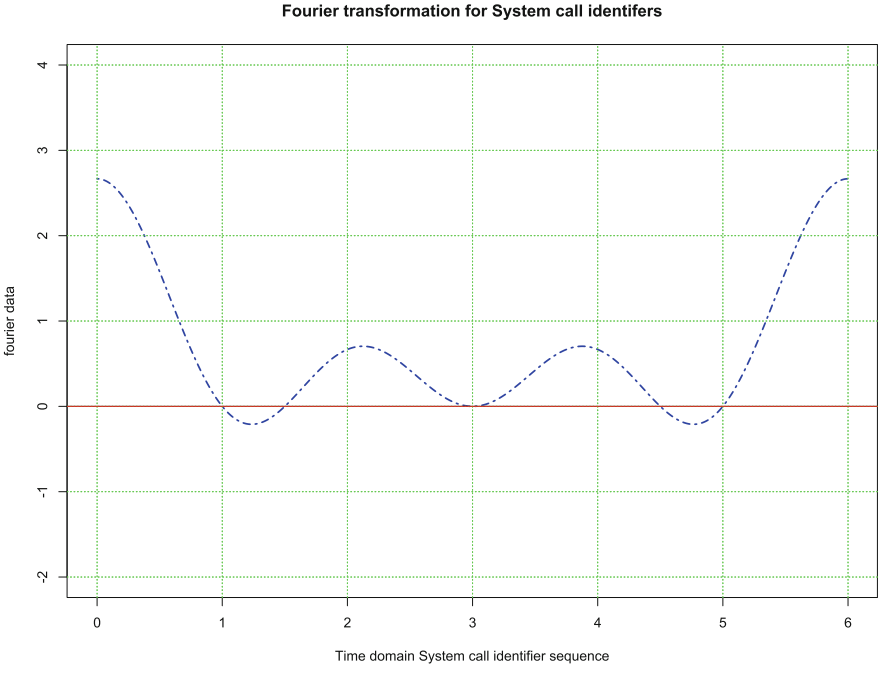

Fig. 3. Fourier transformation of system call identifiers sine representation

are empirically observed optimum are as follows. SVM (rbf) [16,17] is configured with the parameters n = 10 (cross validation value), s = 0 (default type of SVM), d = 5 (degree in kernel function) and rest all on default values. KNN [40] is configured with k = 10 (e.g. k-fold cross validation). ELM [41] with number of hidden neurons = 50, activation function = radbas, sigmoid, sin and all data points of the feature matrix are normalized between the scale 1 and −1.

$$f(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi \xi^{\xi}} dx \tag{2}$$

Algorithm 1. Anomaly Detection: Training and Testing

Require: system call identifiers
Ensure: test sequence is normal or abnormal
Training
1:Sine transformation to input data using eq(1)
2:Fourier transformation to data collected from step 1 and using eq(2)
3:Train [SVM or KNN or ELM] ⟵ sequences from step 2
Testing
4: Repeat step 1 and 2 respectively
5: Predict a sequence as normal or abnormal ⟵ Trained [SVM or KNN or ELM]
6: Go to step 4 until the last test sequence
7: End

In order to automate the above discussed framework of proposed HADS, Algorithm 1 is developed. In Algorithm 1, at the training, the normal or abnormal input signal (i.e., in the case of experimenting with labeled host IDS data-set) of system call identifiers are fragmented according to 1 s of sliding window and transformed into the frequency domain as discussed above. The reason to adopt 1 s length of sliding window is to extract frequency domain information from short sequence of system calls while short sequence of system calls are acknowledged as the good discriminator between normal and abnormal [24]. Further, the adopted machine learning algorithms are trained respectively with input frequency domain feature vectors. In Algorithm 1, at testing the trained machine learning models can predict/classify the input test frequency domain feature vector for normal or abnormal.

4 Experiments and Results

The proposed HADS given in Fig. 1 is evaluated using the criteria given in [16, 17]. The purpose of this section is to answer the following questions: (i) How can the accuracy of the HADS be improved by employing frequency domain transformation to system call identifiers? (ii) Is it possible to minimize the error in detecting host anomalies while adopting frequency domain feature vectors for the host activities? And (iii) what is the impact of host activities frequency domain information in building normal profile and in detecting anomalies?

In our experiment we utilized three IDS data-sets namely, ADFA-LD [42], KDD 98 [43], and NGIDS-DS [20]. ADFA-LD is a small data-set with fewer

attacks and normal data collection, whereas KDD 98 is outdated in-terms of modern attacks and normal computer activities foot prints. However, both these data-sets are utilized to compare the performance of proposed Algorithm 1. The training and testing traces of both these data-sets are acquired from [17]. Further, the modern IDS data-set (i.e., NGIDS-DS) which is generated with the maximum possible quality of realism, in the next generation cyber range infrastructure of the Australian Centre for Cyber Security (ACCS) at the Australian Defence Force Academy (ADFA), Canberra, which is designed according to the guidelines provided in [44]. The key advantage of this infrastructure is the availability of the IXIA Perfect Storm hardware. The combination of a network traffic-generation appliance and virtual cyber range provides both legitimate traffic and host-based connectivity. The IXIA Perfect Storm tool provides four major capabilities. Firstly, it can produce a mixture of modern normal and unknown abnormal cyber traffic. Secondly, it can generate the maximum number and type of zero-day attacks with different dynamic behaviors based on packs that exploit known Common Vulnerability Exposures (CVE). Thirdly, it can establish profiles of the cyber traffic of multiple enterprises. Fourthly, it can generate ground truth automatically. Moreover, the composition of all three data-sets is given in Table 1, where roughly 1:5 training to testing ratio is adopted with normal data as suggested in [38].

Table 1. Data-sets composition for training and testing Algorithm 1

Data-sets	Normal training data	Normal validation data	Test attack data
NGIDS-DS records	17,758,345	71,033,389	1,262,426 records
ADFA-LD traces	833	4372	746
KDD 98 traces	1076	4305	465

The accuracy and error comparison of three machine learning algorithms which are adopted in Algorithm 1 for three data-sets, is given in Table 2. According to [16,17] DR is calculated at testing phase of the Algorithm 1 by dividing the number of detected abnormal sequences to the total number of abnormal sequences. Further, for FAR, first false positive and negative rates (i.e., FPR and FNR) are calculated at the testing phase of Algorithm 1 respectively. FPR is calculated by dividing the detected normal sequences as abnormal to the total number of normal attacks, whereas FNR is measured by dividing the number of abnormal sequences detected as normal to the total number of abnormal sequences. Lastly, the FAR is calculated as average joint error which is defined as $\mathrm{FAR} = \mathrm{FNR} + \mathrm{FPR}/2$.

It can be observed from Table 2 that, by transforming the raw system call identifiers sequences into frequency domain have significant impact on the accuracy of proposed HADS. For instance, there is a significant increase in the DR and decrease in the FAR at each machine learning algorithm using frequency transformed sequences. The major reason for this accuracy improvement is in fact the extraction of hidden features (i.e., frequency of amplitudes in a time) from the raw system call identifiers and then the inclusion of these features as

Table 2. Accuracy and error comparison of system call identifiers raw and frequency domain sequences with multiple data-sets

Data-sets	Algorithms	Raw				Transformed			
		DR%	FAR%	FNR%	FPR%	DR%	FAR%	FNR%	FPR%
NGIDS-DS	SVM(rbf)	5	50	99	0.2	7.2	49	99	0.2
	KNN	8	50	99	0.6	10.5	48	99	0.5
	ELM	75	19	18	21	81	14	13	15
ADFA-LD	SVM(rbf)	70	20	30	10	75	17	26	9
	KNN	60	20	39.2	2	67	16.6	33	0.9
	ELM	88	17	12	23.7	95	11.47	5	16
KDD 98	SVM(rbf)	44	55	57	52	61	46	30	61.8
	KNN	34	68	65.1	70	48	53	52	49.7
	ELM	91	5	8.09	3	97	2	3	0.56

pre-classification assistance to machine learning algorithms. Also, all three algorithms performances are low upon NGIDS-DS data-set. The reasons behind this fact are: (i) both ADFA-LD depicts less complex data-set with small number of attacks and normal activities footprints; (ii) Kdd 98 is outdated and less complex, with inclusion of small number of high foot print attacks and differentiable normal computer activities reflection; and (iii) NGIDS-DS is complex data-set with inclusion of huge number of modern low foot print attacks and normal computer activities [13].

Further, it can be observe from Fig. 4 and Table 2 that, SVM and KNN performances are low as compared to ELM upon NGIDS-DS. The reasons for this aspect are as follows: (i) As the data-set NGIDS-DS [20] is recently released and it reflects modern sophisticated ways of conducting attacks that constitutes low foot print upon host logs i.e., system call identifier sequences of the processes. Due to this, in the data set the normal to attack records ratio is about 90:1. Hence, it is observed complex for SVM and KNN to distinguish the data points in two classes where the one class is the majority class [45,46]; (ii) system call identifiers sequence actually represents any type of activity (e.g. legitimate or illegal) that occurred at the host but from machine learning classifier point of view it constitutes a high similarity between the data points for normal and attack sequences. Hence it is challenging for the selected SVM and KNN versions to distinguish the sequences or vectors having similar data values [45,46]; (iii) in ELM, a single hidden layer feed-forward NN selects randomly hidden layers and determine the output weight (e.g. weight times feature vectors) for fitting the target output about any feature vector in a feature matrix. The hidden layers do not need to be tuned iteratively as compared to traditional ANN and the activation functions are adoptable [41] and (iv) in ELM, the data points of a feature vector are transformed into another domain or extended dimension using activation functions as kernels such as sigmoid, sin, and raidbas. As a result,

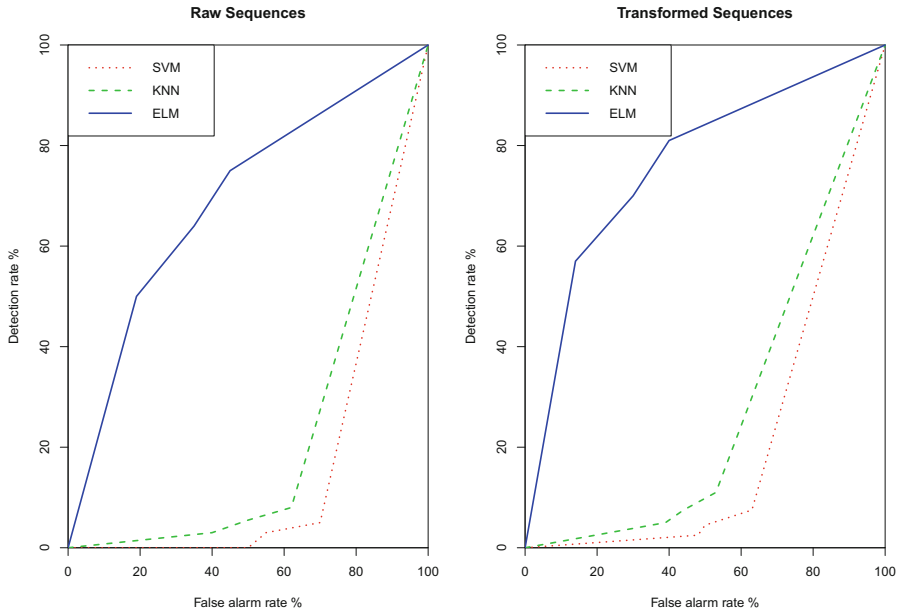

Fig. 4. Using latest IDS data-set (NGIDS-DS), raw and frequency transformed system call identifiers sequences comparison with multiple machine learning algorithms in terms of anomaly detection accuracy and error via ROC curves

ELM classifier is able to discriminate between the feature vectors of different classes by learning the natural hidden patterns which are not visible with the data points in raw domain [47].

5 Conclusion

It is vital to protect machines operating systems in the current and future era of cyber threats, where attacks saturation power is observed able to penetrate the network defense zones. To deal with this, an anomaly detection mechanism for cloud servers is proposed and investigated in this paper. In the proposed host based anomaly detection system, first, the audit data from LINUX/UNIX based cloud servers (i.e., system call identifiers) is transformed into frequency domain by sine and Fourier transformation from time domain, in order to extract frequency domain feature vectors of running activities at the host. Second, different machine learning algorithms are trained and tested with these frequency domain feature vectors as anomaly detection engine. Results, demonstrate that, these frequency domain features of host activities identification, are capable of detecting host anomalies with minimum error. In future, it is intended to transform the other types of audit data from machines such as CPU power and memory consumption, in order to design more reliable anomaly detection system for machines operating system.

References

1. Pabla, I., Khalil, I., Hu, J.: Intranet security via firewalls. In: Stavroulakis, P., Stamp, M. (eds.) Handbook of Information and Communication Security, pp. 207–219. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-04117-4_11
2. Wang, H., Zhang, Y., Cao, J.: Access control management for ubiquitous computing. Future Gener. Comput. Syst. **24**(8), 870–878 (2008)
3. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Military Communications and Information Systems Conference (MilCIS), pp. 1–6. IEEE (2015)
4. Wang, Y., Wen, S., Xiang, Y., Zhou, W.: Modeling the propagation of worms in networks: a survey. IEEE Commun. Surv. Tutor. **16**(2), 942–960 (2014)
5. Moustafa, N., Slay, J.: The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems. In: 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), pp. 25–31. IEEE (2015)
6. Cesare, S., Xiang, Y., Zhou, W.: Malwisean effective and efficient classification system for packed and polymorphic malware. IEEE Trans. Comput. **62**(6), 1193–1206 (2013)
7. Rudd, E., Rozsa, A., Gunther, M., Boult, T.: A survey of stealth malware: attacks, mitigation measures, and steps toward autonomous open world solutions. IEEE Commun. Surv. Tutor. **19**(2), 1145–1172 (2017)
8. Moustaf, N., Slay, J.: Creating novel features to anomaly network detection using DARPA-2009 data set. In: Proceedings of the 14th European Conference on Cyber Warfare and Security, p. 204. Academic Conferences Limited (2015)
9. Ficco, M., Palmieri, F.: Introducing fraudulent energy consumption in cloud infrastructures: a new generation of denial-of-service attacks. IEEE Syst. J. **11**(2), 460–470 (2017)
10. Kumarage, H., Khalil, I., Tari, Z., Zomaya, A.: Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling. J. Parallel Distrib. Comput. **73**(6), 790–806 (2013)
11. Kumarage, H., Khalil, I., Tari, Z.: Granular evaluation of anomalies in wireless sensor networks using dynamic data partitioning with an entropy criteria. IEEE Trans. Comput. **64**(9), 2573–2585 (2015)
12. Alabdulatif, A., Kumarage, H., Khalil, I., Yi, X.: Privacy-preserving anomaly detection in cloud with lightweight homomorphic encryption. J. Comput. Syst. Sci. **90**, 28–45 (2017)
13. Haider, W., Hu, J., Xie, Y., Yu, X., Wu, Q.: Detecting anomalous behavior in cloud servers by nested arc hidden SEMI-Markov model with state summarization. IEEE Trans. Big Data (2017)
14. Rittinghouse, J.W., Ransome, J.F.: Cloud Computing: Implementation, Management, and Security. CRC Press, Boca Raton (2016)
15. Zissis, D., Lekkas, D.: Addressing cloud computing security issues. Future Gener. Comput. Syst. **28**(3), 583–592 (2012)
16. Haider, W., Hu, J., Xie, M.: Towards reliable data feature retrieval and decision engine in host-based anomaly detection systems. In: 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), pp. 513–517. IEEE (2015)
17. Haider, W., Hu, J., Yu, X., Xie, Y.: Integer data zero-watermark assisted system calls abstraction and normalization for host based anomaly detection systems. In: 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud), pp. 349–355. IEEE (2015)

18. Taddeo, M., Glorioso, L.: Ethics and Policies for Cyber Operations: A NATO Cooperative Cyber Defence Centre of Excellence Initiative, vol. 124. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-45300-2
19. Herpig, S.: Anti-war era: the need for proactive cyber security. In: Felici, M. (ed.) CSP 2013. CCIS, vol. 182, pp. 165–176. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41205-9_14
20. Haider, W., Hu, J., Slay, J., Turnbull, B., Xie, Y.: Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling. J. Netw. Comput. Appl. **87**, 185–192 (2017)
21. Toh, K.-A., Tan, G.-C.: Exploiting the relationships among several binary classifiers via data transformation. Pattern Recogn. **47**(3), 1509–1522 (2014)
22. Toh, K.-A.: Training a reciprocal-sigmoid classifier by feature scaling-space. Mach. Learn. **65**(1), 273–308 (2006)
23. Tran, Q.-L., Toh, K.-A., Srinivasan, D., Wong, K.-L., Low, S.Q.-C.: An empirical comparison of nine pattern classifiers. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **35**(5), 1079–1091 (2005)
24. Hofmeyr, S.A., Forrest, S., Somayaji, A.: Intrusion detection using sequences of system calls. J. Comput. Secur. **6**(3), 151–180 (1998)
25. Haider, W., Creech, G., Xie, Y., Hu, J.: Windows based data sets for evaluation of robustness of host based intrusion detection systems (IDS) to zero-day and stealth attacks. Future Internet **8**(3), 29 (2016)
26. Creech, G., Hu, J.: A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns. IEEE Trans. Comput. **63**(4), 807–819 (2014)
27. Cho, S.-B., Park, H.-J.: Efficient anomaly detection by modeling privilege flows using hidden Markov model. Comput. Secur. **22**(1), 45–55 (2003)
28. Murtaza, S.S., Khreich, W., Hamou-Lhadj, A., Gagnon, S.: A trace abstraction approach for host-based anomaly detection. In: IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), pp. 1–8. IEEE (2015)
29. Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.: A sense of self for unix processes. In: Proceedings of 1996 IEEE Symposium on Security and Privacy, pp. 120–128. IEEE (1996)
30. Kosoresow, A.P., Hofmeyer, S.: Intrusion detection via system call traces. IEEE Softw. **14**(5), 35–42 (1997)
31. Moustafa, N., Slay, J., Creech, G.: Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. IEEE Trans. Big Data (2017)
32. Ghosh, A.K., Schwartzbard, A., Schatz, M.: Learning program behavior profiles for intrusion detection. In: Workshop on Intrusion Detection and Network Monitoring, vol. 51462, pp. 1–13 (1999)
33. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. In: Barbará, D., Jajodia, S. (eds.) Applications of Data Mining in Computer Security, vol. 6, pp. 77–102. Springer, Boston (2002). https://doi.org/10.1007/978-1-4615-0953-0_4
34. Hoang, X., Hu, J.: An efficient hidden Markov model training scheme for anomaly intrusion detection of server applications based on system calls. In: Proceedings of 12th IEEE International Conference on Networks, (ICon 2004), vol. 2, pp. 470–474. IEEE (2004)

35. Hu, J., Yu, X., Qiu, D., Chen, H.-H.: A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection. IEEE Netw. **23**(1), 42–47 (2009)
36. Maggi, F., Matteucci, M., Zanero, S.: Detecting intrusions through system call sequence and argument analysis. IEEE Trans. Dependable Secure Comput. **7**(4), 381–395 (2010)
37. Silic, M., Back, A.: Open source software adoption: lessons from linux in munich. IT Prof. **19**(1), 42–47 (2017)
38. Creech, G.: Developing a high-accuracy cross platform host-based intrusion detection system capable of reliably detecting zero-day attacks. Ph.D. dissertation, University of New South Wales, Canberra, Australia (2014)
39. Bracewell, R.N., Bracewell, R.N.: The Fourier Transform and Its Applications, vol. 31999. McGraw-Hill, New York (1986)
40. Moustafa, N., Creech, G., Slay, J.: Big data analytics for intrusion detection system: statistical decision-making using finite dirichlet mixture models. In: Palomares Carrascosa, I., Kalutarage, H.K., Huang, Y. (eds.) Data Analytics and Decision Support for Cybersecurity. DA, pp. 127–156. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59439-2_5
41. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1), 489–501 (2006)
42. Creech, G., Hu, J.: Generation of a new IDS test dataset: time to retire the KDD collection. In: 2013 IEEE Wireless Communications and Networking Conference (WCNC), pp. 4487–4492. IEEE (2013)
43. KDD98 (1988). http://www.ll.mit.edu/mission/communications/
44. Davis, J., Magrath, S.: A survey of cyber ranges and testbeds. Defence Science and Technology Organisation Edinburgh (Australia) Cyber and Electronic Warfare Division, Technical report (2013)
45. Xing, Z., Pei, J., Keogh, E.: A brief survey on sequence classification. ACM SIGKDD Explor. Newsl. **12**(1), 40–48 (2010)
46. Justino, E.J., Bortolozzi, F., Sabourin, R.: A comparison of SVM and HMM classifiers in the off-line signature verification. Pattern Recogn. Lett. **26**(9), 1377–1385 (2005)
47. Vong, C.-M., Ip, W.-F., Wong, P.-K., Chiu, C.-C.: Predicting minority class for suspended particulate matters level by extreme learning machine. Neurocomputing **128**, 136–144 (2014)