Evaluating Host-Based Anomaly Detection Systems: Application of the Frequency-Based Algorithms to ADFA-LD

Miao Xie[1], Jiankun Hu[1], Xinghuo Yu[2], and Elizabeth Chang[1]

[1] UNSW Canberra, Canberra, ACT 2612, Australia
[2] RMIT University, Melbourne, VIC 3001, Australia

Abstract. ADFA Linux data set (ADFA-LD) is released recently for substituting the existing benchmark data sets in the area of host-based anomaly detection which have lost most of their relevance to modern computer systems. ADFA-LD is composed of thousands of system call traces collected from a contemporary Linux local server, with six types of up-to-date cyber attack involved. Previously, we have conducted a preliminary analysis of ADFA-LD, and shown that the frequency-based algorithms can be realised at a cheaper computational cost in contrast with the short sequence-based algorithms, while achieving an acceptable performance. In this paper, we further exploit the potential of the frequency-based algorithms, in attempts to reduce the dimension of the frequency vectors and identify the optimal distance functions. Two typical frequency-based algorithms, i.e., k-nearest neighbour (kNN) and k-means clustering (kMC), are applied to validate the effectiveness and efficiency.

Keywords: host-based intrusion detection system (HIDS), Unix system call.

1 Introduction

In the area of host-based anomaly detection [1], most of the existing benchmark data sets, such as UMN [2] and DARPA [3] intrusion detection data sets, are compiled a decade ago and have failed to reflect the characteristics of modern computer systems. To fill this gap, ADFA-LD [10] [11] is released recently, which is generated from a Linux local server configured to represent a contemporary computer system. This server provides a range of services such as file sharing, database, remote access and web server, with the operating system of fully patched Ubuntu 11.04 (Linux kernel 2.6.38). The FTP, SSH and MySQL 14.14 are enabled with their default ports. Apache 2.2.17 and PHP 5.3.5 are installed for web-based services. In addition, TikiWiki 8.1 is installed as a web-based collaborative tool. During a given sampling period, the system calls invoked by each specific process is collected from this server in the form of a trace and, for simplicity, the index of the system call is recorded rather than its name. 833 and 4373 normal traces are captured respectively for the purposes of training and

M.H. Au et al. (Eds.): NSS 2014, LNCS 8792, pp. 542–549, 2014.

validation, during which no attacks occur against the host and a variety of legitimate applications are operated as usual. Subsequently, six types of cyber attack [10], i.e., **Hydra-FTP**, **Hydra-SSH**, **Adduser**, **Java-Meterpreter**, **Meterpreter** and **Webshell**, are launched in turn, each of which generates $8 \sim 20$ attack traces. Table 1 summarises the composition of ADFA-LD, in according with type, number and label.

Table 1. Composition of ADFA-LD

Training	Validation	Hydra-FTP	Hydra-SSH	Adduser	Java-Meterpreter	Meterpreter	Webshell
833	4373	162	148	91	125	75	118
normal	normal	attack	attack	attack	attack	attack	attack

There are two common categories of technique to detect intrusions/anomalies using system call traces: short sequence-based and frequency-based [6]. Short sequence-based techniques tend to mine patterns from subsequences of system call traces and a decision is often made through a comparison to the model of the normal patterns [4] [5] [7] [8] [9] [11]. Although this category of techniques are able to generate an accurate normal profile, the learning procedures are extremely time-consuming. Frequency-based techniques, on the contrary, are much cheaper in terms of computation, since they reorganise the system call traces into equal-sized vectors based on the concept of 'frequency' and deal only with the resulting frequency vectors [12] [13] [14]. However, their accuracies in modelling a normal profile may be deteriorated due to the loss of positional information.

Previously, we have conducted a preliminary analysis of ADFA-LD and shown that most of the intrusions/anomalies presented in ADFA-LD can be identified by the frequency-based kNN algorithms [15]. In this paper, we intend to further exploit the potential of the frequency-based algorithms against ADFA-LD. First, it attempts to map the original $n-$dimensional space of frequency vectors into a lower $p-$dimensional space by using principal component analysis (PCA). Second, various distance functions are attempted separately to validate their effectiveness. In different settings, two typical frequency-based algorithms, i.e., kNN and kMC, are tested respectively. Detection accuracy (ACC) and false positive rate (FPR) are employed as the performance metrics, which are given in the form of RoC curve.

The rest of this paper is organised as follows. Section 2 introduces how to reduce the dimension of the frequency vectors and the distance functions. Section 3 details the kNN and kMC algorithms and presents their performances obtained from ADFA-LD and, finally, section 4 summarises this paper.

2 Model, Dimension Reduction and Distance Functions

In this section, we define the model by which, as previously mentioned, the system call traces can be transformed into equal-sized frequency vectors. Then, we discuss how to reduce the dimension of the frequency vectors, as well as various

distance functions which will be used for measuring the similarity between two frequency vectors.

A system call trace is a discrete sequence, with a variant length and the elements ranging from 1 to n (the maximal index of a system call). The indexes of the system calls and n are determined by the operating system; for example, Linux kernel 2.6.38 provides a total of 325 system calls [16] such that $n = 325$. Let s denote a system call trace, $|s|$ its length and f_i the number of occurrence of the system call indexed by i, where $i = 1, 2, \cdots, n$. The element of the frequency vector can be defined as

$$\bar{f}_i = \frac{f_i}{|s|}.$$

Although the system call traces can be transformed into shorter and equal-sized frequency vectors according to the above model, while operating these n dimensional vectors, the computational cost is still considerable. As most of the frequency vectors are sparse, intuitively, the dimension can be largely reduced and a comparable performance can be achieved as long as most of the variance is retained. Let m denote the total number of the training system call traces. The training data set, say $\mathbf{T}$, can be organised in the form of a $m \times n$ matrix by which we can reduce the dimension using PCA [17] [18]. If the sample covariance matrix of $\mathbf{T}$ is denoted by Q, using eigen decomposition, Q can be factorised as $Q = W \Lambda W'$ where $\Lambda = diag\,(\lambda_1, \lambda_2, \cdots, \lambda_n)$ is a diagonal matrix with respect to the descending eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and W is the $n \times n$ orthogonal matrix that contains the eigenvectors, i.e., $W = [w_1\ w_2\ \cdots\ w_n]$. By specifying r, $0 < r < 1$, it can obtain a subset of the eigenvectors from W, i.e., $\bar{W} = [w_1\ w_2\ \cdots\ w_p]$ where $p < n$ and

$$r \leq \sum_{i=1}^{p} \lambda_i.$$

For any frequency vector s, the dimension can be reduced according to

$$\bar{s} = s\bar{W}.$$

Let $\mathbf{V}$ and $\mathbf{A}$ denote the validation and attack data sets respectively. After the dimension is reduced, the training, validation and attack data sets are denoted by $\bar{\mathbf{T}}$, $\bar{\mathbf{V}}$ and $\bar{\mathbf{A}}$ respectively, where $\bar{\mathbf{T}} = \mathbf{T}\bar{W}$, $\bar{\mathbf{V}} = \mathbf{V}\bar{W}$ and $\bar{\mathbf{A}} = \mathbf{A}\bar{W}$.

Let $\bar{Q}$ and $\bar{\Lambda}$ denote the sample covariance matrix of $\bar{\mathbf{T}}$ and the diagonal matrix obtained from the eigen decomposition of $\bar{Q}$ respectively. Given any two p dimensional frequency vectors, denoted by x and y for short, some distance functions are defined for measuring their similarity, as shown in Table 2.

3 The Frequency-Based Algorithms

Application of the frequency-based algorithms to ADFA-LD is presented in this section. We specify $r = 0.8$, which indicates that 80% variance of the raw data

Table 2. Distance functions

Distance/Metric	$distance(x, y)$	Distance/Metric	$distance(x, y)$
Euclidean	$(x-y)(x-y)^{'}$	Minkowski	$\left\{\sum_{i=1}^{p}\|x_i - y_i\|^q\right\}^{\frac{1}{q}}$
Standardised Euclidean	$(x-y)\bar{A}^{-1}(x-y)^{'}$	Cosine	$1 - \frac{xy^{'}}{(xx^{'})^{\frac{1}{2}}(yy^{'})^{\frac{1}{2}}}$
Mahalanobis	$(x-y)\bar{Q}^{-1}(x-y)^{'}$	Correlation	$1 - \frac{1}{n}\frac{(x-\mu_x)(y-\mu_y)^{'}}{\sigma_x\sigma_y}$

is retained; as such, $p = 9$. By testing a range of each parameter, the ACC and FPR of each algorithm against each type of attack are given in the form of a RoC curve. In particular, ACC is the number of successfully detected abnormal traces (attack involved) dividing by the total number of abnormal traces and FPR is the number of normal traces which are identified as abnormal dividing by the length of the validation data set.

3.1 kNN

kNN is the most widely used algorithm in the area of anomaly detection [18] [19] [20] [21]. Based on the kNN algorithm, we detect a system call trace by searching its p dimensional frequency vector's k nearest neighbours within a radius of d from $\bar{\mathbf{T}}$ in terms of a certain distance function. That is, for any $y \in \bar{\mathbf{V}} \cup \bar{\mathbf{A}}$ and all $x \in \bar{\mathbf{T}}$, if

$$\#\left(distance(x, y) \leq d\right) \geq k,$$

y is normal; otherwise abnormal.

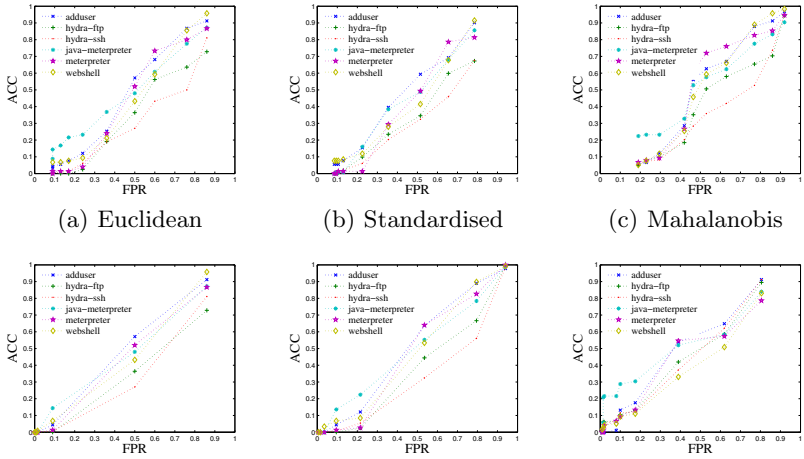
Table 3. Parameters of kNN

Distance/Metric	d	step width	Distance/Metric	d	step width
Euclidean	$[0.01, 0.1]$	0.01	Minkowski $q = 2$	$[0.1, 1]$	0.1
Standardised Euclidean	$[1, 10]$	1	Minkowski $q = 3$	$[0.05, 0.5]$	0.05
Mahalanobis	$[0.5, 5]$	0.5	Cosine	$[0.05, 0.5]$	0.05

There are two parameters d and k to be specified in the algorithm, where d varies according to the distance function adopted and $\frac{k}{m}$ indicates a small probability. We fix $k = 20$ empirically, i.e., the small probability is equal to 0.024, and test a range of d for each distance function separately. The parameters are summarised in Table 3, with the results shown in Figure 1.

3.2 kMC

kMC algorithm is originated from signal processing [22] and has been widely used for the problems of anomaly detection [23] [24]. It aims to partition the given observations into k clusters in which each observation belongs to the cluster in terms of the nearest mean. Then, an observation is detectable according to its distances to the clusters.



(a) Euclidean (b) Standardised (c) Mahalanobis

(d) Minkowski $q = 2$ (e) Minkowski $q = 3$ (f) Cosine

Fig. 1. The results from kNN

Given the observations $\{x_1, x_2, \cdots, x_m\}$, i.e., the set of the p dimensional training frequency vectors ($\bar{\mathbf{T}}$), the kMC algorithm partitions the observations into k clusters $\mathbf{C} = \{C_1, C_2, \cdots, C_k\}$ by minimising

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{k} \sum_{x_j \in C_i} \| x_j - c_i \|$$

where c_i is the centre of C_i. Although solving the problem is computationally difficult, there are a number of efficient heuristic algorithms which are usually similar to the idea of the expectation-maximisation (EM) algorithm. For example, Lloyd's algorithm [25] is able to reach the local optimum by an iterative process which, in particular, alternates between two steps: assignment and update. In an assignment step, each observation is assigned to the cluster whose mean yields the least within-cluster sum of squares and, in an update step, the centres of the observations in the new clusters are calculated.

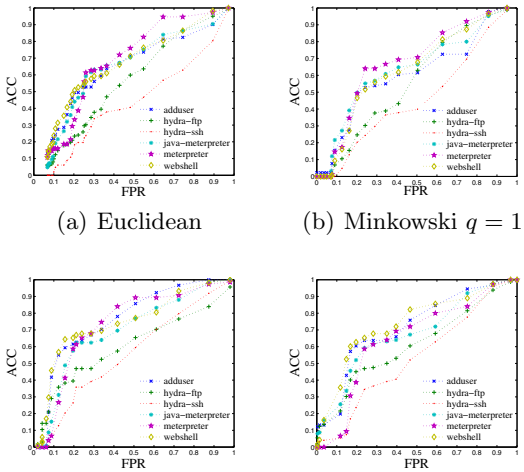
Table 4. Parameters of kMC

Distance/Metric	τ	step width	Distance/Metric	τ	step width
Euclidean	$[0.005, 0.15]$	0.005	Cosine	$[0.025, 0.5]$	0.025
Minkowski metric $q = 1$	$[0.15, 0.9]$	0.05	Correlation	$[0.025, 0.5]$	0.025

When $\{c_1, c_2, \cdots, c_k\}$ are ready, the frequency vector of a system call trace, say y, $y \in \bar{\mathbf{V}} \cup \bar{\mathbf{A}}$, can be detected through the following inequation,

$$\min_{i=1,2,\cdots,k} distance(c_i, y) \leq d.$$

If this inequation is true, the system call trace is identified as normal; otherwise abnormal. There are also two parameters k and d to be specified in the KMC algorithm. k is related to the distribution of the given observations and, by manually adjusting, it is fixed to 5. d is not easy to empirically specify as it varies according to the distance function. As a result, we employ the maximum of the within-cluster distances obtained from $\bar{\mathbf{T}}$ as a scale d^*, and test d by multiplying a range of coefficient τ and this scale, i.e., $d = \tau d^*$. All the parameters are given in Table 4 and the results are shown in Figure 2.



(a) Euclidean (b) Minkowski $q = 1$

(c) Cosine (d) Correlation

Fig. 2. The results from kMC

3.3 Evaluation

In this subsection, we evaluate the results according to three aspects: (1) the performances of the two frequency-based algorithms against ADFA-LD, (2) the performances against each type of attack, and (3) the correlation between the performances and the distance functions.

The kNN algorithm fails to effectively detect the attacks with the low dimensional frequency vectors no matter what distance function is used and its performance is much worse than that of the original frequency vectors. However, the kMC algorithm is able to achieve an ACC of higher than 60% with a FPR of lower than 20% for most types of attack. In addition, the kMC algorithm is much more efficient than the kNN algorithm in terms of computation, as each time of detection requires only computing the distances to the k centres. Thus, it can conclude that the kMC algorithm outperforms the kNN algorithm when the dimension of the frequency vector is reduced.

As far as the performances against each type of attack, **Java-Meterpreter** is the easiest type to detect using both the algorithms. **Hydra-FTP** and **Hydra-SSH**, on the contrary, can not be effectively addressed by the frequency-based

algorithms for which, basically, an ACC of 50% will incur an FPR of 50%. This result indicates that a frequency-based algorithm is not versatile against any type of attack.

Finally, we look at how the performance is relating to the distance function. The result from cosine distance is the best which, in particular, achieves an ACC of 60% with a FPR of around 10% except for **Hydra-FTP** and **Hydra-SSH**, when the kMC algorithm is employed. Correlation distance is the second choice, by which the performance is comparable with that of cosine distance. Although Euclidean and Mahalanobis distances are most commonly used distance metrics, their performances, in this case, are not impressive. In short, distance function is not a crucial factor to performance.

4 Conclusion

In this paper, following the preliminary analysis, we applied two typical frequency-based algorithms to ADFA-LD. After transforming the system call traces into the frequency vectors, in order to further reduce the computational cost, we attempted to reduce the dimension of the frequency vectors using PCA, and the subsequent analysis was conducted in a lower dimensional space. The results shown that the kNN algorithm is ineffective against the attacks, and the kMC algorithm can detect most types of attack effectively. In the future, we will continue to study the characteristics of ADFA-LD and attempt to design more efficient and effective algorithms for detecting the attacks.

References

1. Stavroulakis, P., Stamp, M.: Handbook of information and communication security. Springer (2010)
2. `http://www.cs.unm.edu/~immsec/systemcalls.htm`
3. `http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/`
4. Forrest, S., Hofmeyr, S., Somayaji, A., Longstaff, T.A.: A sense of self for Unix processes. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy, pp. 120–128 (1996)
5. Kosoresow, A.P., Hofmeyer, S.A.: Intrusion detection via system call traces. IEEE Software 14, 35–42 (1997)
6. Forrest, S., Hofmeyr, S., Somayaji, A.: The Evolution of System-Call Monitoring. In: Annual Computer Security Applications Conference, ACSAC 2008, pp. 418–430 (2008)
7. Eskin, E., Wenke, L., Stolfo, S.J.: Modeling system calls for intrusion detection with dynamic window sizes. In: Proceedings of the DARPA Information Survivability Conference Exposition II, DISCEX 2001, pp. 165–175 (2001)
8. Hoang, X.D., Hu, J.: An efficient hidden Markov model training scheme for anomaly intrusion detection of server applications based on system calls. In: Proceedings of the 12th IEEE International Conference on Networks (ICON 2004), pp. 470–474 (2004)

9. Hoang, X.D., Hu, J., Bertok, P.: A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference. Journal of Network and Computer Applications 32, 1219–1228 (2009)
10. 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 (2013)
11. Creech, G., Hu, J.: A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguous and Discontiguous System Call Patterns. IEEE Transactions on Computers 63, 807–819 (2014)
12. Liao, Y., Vemuri, V.R.: Use of K-nearest neighbor classifier for intrusion detection. Computers & Security 21, 439–448 (2002)
13. Chen, W.-H., Hsu, S.-H., Shen, H.-P.: Application of SVM and ANN for intrusion detection. Computers & Operations Research 32, 2617–2634 (2005)
14. Sharma, A., Pujari, A.K., Paliwal, K.K.: Intrusion detection using text processing techniques with a kernel based similarity measure. Computers & Security 26, 488–495 (2007)
15. Xie, M., Hu, J.: Evaluating host-based anomaly detection systems: A preliminary analysis of ADFA-LD. In: 2013 6th International Congress on Image and Signal Processing (CISP), pp. 1711–1716 (2013)
16. `http://osinside.net/syscall/system_call_table.htm`
17. Jolliffe, I.: Principal component analysis. Wiley Online Library (2005)
18. Xie, M., Han, S., Tian, B.: Highly Efficient Distance-Based Anomaly Detection through Univariate with PCA in Wireless Sensor Networks. In: 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 564–571 (2011)
19. Xie, M., Hu, J., Tian, B.: Histogram-Based Online Anomaly Detection in Hierarchical Wireless Sensor Networks. In: 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 751–759 (2012)
20. Xie, M., Hu, J., Han, S., Chen, H.-H.: Scalable Hypergrid k-NN-Based Online Anomaly Detection in Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems 24, 1661–1670 (2013)
21. Hu, J., Gingrich, D., Sentosa, A.: A k-Nearest Neighbor Approach for User Authentication through Biometric Keystroke Dynamics. In: IEEE International Conference on Communications, ICC 2008, pp. 1556–1560 (2008)
22. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A k-means clustering algorithm. Applied Statistics, 100–108 (1979)
23. Mahmood, A.N., Hu, J., Tari, Z., Leckie, C.: Critical infrastructure protection: Resource efficient sampling to improve detection of less frequent patterns in network traffic. Journal of Network and Computer Applications 33, 491–502 (2010)
24. Xi, K., Tang, Y., Hu, J.: Correlation keystroke verification scheme for user access control in cloud computing environment. The Computer Journal 54, 1632–1644 (2011)
25. Lloyd, S.: Least squares quantization in PCM. IEEE Transactions on Information Theory 28, 129–137 (1982)