

# A Reference Based Analysis Framework for Analyzing System Call Traces

Varun Chandola<sup>\*</sup>  
Oak Ridge National  
Laboratory  
chandolav@ornl.gov

Shyam Boriah  
University of Minnesota  
sboriah@cs.umn.edu

Vipin Kumar  
University of Minnesota  
kumar@cs.umn.edu

## ABSTRACT

Reference based analysis (RBA) is a novel data mining tool for exploring a test data set with respect to a reference data set. The power of RBA lies in its ability to transform any complex data type, such as symbolic sequences and multivariate categorical data instances, into a multivariate continuous representation. The transformed representation not only allows visualization of the complex data, which cannot be otherwise visualized in its original form, but also allows enhanced anomaly detection in the transformed feature space. We demonstrate the application of the RBA framework in analyzing system call traces and show how the transformation results in improved intrusion detection performance over state of art data mining based intrusion detection methods developed for system call traces.

## Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Data Mining*

## General Terms

Algorithms

## Keywords

Intrusion Detection, Reference Based Analysis, Anomaly Detection

## 1. INTRODUCTION

System call trace monitoring has been the focus of many research articles over the past decade [7, 8, 4, 7, 12, 10, 9, 6, 5, 13]. The focus of these techniques is to apply data mining and machine learning based solutions to detect intrusions in the computer system behavior by analyzing the system call traces.

<sup>\*</sup>This work was done when author was at University of Minnesota.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. CSIRW '10, April 21-23, Oak Ridge, Tennessee, USA Copyright © 2010 ACM 978-1-4503-0017-9 ... \$5.00

A key challenge associated with analysis of system call traces is that each trace is a long sequences of discrete events, where each event is a system call. Each trace is list of system calls issued by a single process in an operating system. Depending on the nature of the program that calls the process, the traces can have varying lengths. For example, a single daemon process that runs continuously results in very long traces. Moreover, different executions of the same program can result in traces with widely varying lengths. This, along with the fact that the system calls are discrete values with no relationship to each other, makes visual analysis of trace data a challenging task. For example, consider a hypothetical set of traces shown in Figure 1. It is clear that it is not possible to visualize the long traces and identify which traces contain intrusions in the raw form.

```
open, read, mmap, mmap, open, read, mmap ...
open, mmap, mmap, read, open, close ...
open, close, open, close, open, mmap, close ...
```

Figure 1: A sample data set comprising of three operating system call traces.

Recently, we proposed a data driven analysis framework, called *Reference Based Analysis* (RBA) [2, 1], that can be used to analyze a given data set, with respect to a reference data set. The strength of the RBA framework is that it can be used to analyze complex types of data, for which limited analysis techniques exist. The key property of RBA is that it maps data instances to a multi-dimensional space such that the data instances that are similar to the reference set and data instances that are different from the reference set are easily distinguishable in the transformed space.

In this paper, we show how RBA can be used to analyze system call traces. We use the analysis for two tasks: (i) visualizing normal and intrusive test traces with respect to a reference set of normal traces, and (ii) using the RBA induced transformation as features to be used within an anomaly detection algorithm to detect intrusions. We present our results on benchmark data sets and show that the RBA based visualization is a novel way of analyzing system call traces and identifying various properties of a database of system call traces. Moreover, we also show that when the RBA induced features are used within a nearest neighbor based anomaly detection algorithm, it outperforms all state of art intrusion detection methods.

## 2. REFERENCE BASED ANALYSIS FOR DIS-CRETE SEQUENCES

The key aspect of RBA is that it maps data into a multivariate continuous space. This mapping is done with respect to a reference data set. The reference data set is assumed to contain instances of one class, also known as the *reference* class, while the test data set can contain instances belonging to reference class as well as other classes. For this paper the reference class is also referred to as *normal* class. The test data is assumed to contain normal traces as well as *anomalous* (or *intrusive*) class.

Let the reference data set be denoted with  $\mathbf{S}$ , such that  $\mathbf{S}$  is a collection of traces,  $S_1, S_2, \dots, S_m$ . The test data set is denoted with  $\mathbf{T}$ , such that  $\mathbf{T}$  is a collection of traces,  $T_1, T_2, \dots, T_n$ .

For a given trace,  $T$ , one can define a set of features as follows. We first extract all  $k$ -windows<sup>1</sup> from  $T$ . For each  $k$ -window,  $w_i$ , we compute the number of times it occurs (as a substring) in all sequences in the reference set  $\mathbf{S}$ , denoted as  $f_{w_i}$ . Thus for a sequence  $T$  of length  $l$ , one can compute  $(l - k + 1)$  frequencies. For each  $k$ -window we also compute the number of times its  $(k - 1)$  length prefix occurs in all sequences in the reference set  $\mathbf{S}$ , denoted as  $f'_{w_i}$ . Thus each window is characterized by two frequencies,  $f_{w_i}$  and  $f'_{w_i}$ . Each window is then binned into a two dimensional histogram, with  $p$  equi-width bins along each dimension, based on the two frequencies. The bin values are then normalized by dividing each bin value with the total sum of all bins. Note that  $f_{w_i} \leq f'_{w_i}$ , since the frequency of a window is always upper bounded by the frequency of its prefix. Thus only half of the 2-D histogram can be non-zero. The 2-D histogram is then “flattened” to construct a  $\frac{p(p+1)}{2}$  vector. This vector is used as the multivariate continuous representation of the original test trace. In a similar manner, each trace in the reference data set can also be converted into a multivariate continuous representation.

## 3. INTRUSION DETECTION BENCHMARK DATA SETS

The validation data sets were collected from two repositories of benchmark data generated for evaluation of intrusion detection algorithms. One repository was generated at University of New Mexico<sup>2</sup>. The reference sequences consisted of sequence of system calls generated in an operating system during the normal operation of a computer program, such as sendmail, ftp, lpr etc. The anomalous sequences consisted of sequence of system calls generated when the program is run in an abnormal mode, corresponding to the operation of a hacked computer. We experimented with a number of data sets available in the repository but are reporting results on two data sets, viz, *snd-unm* and *snd-cert*. For each of the two data sets, the original size of the reference as well as anomaly data was small, so we extracted sliding windows of length 100, with a sliding step of 50 from every sequence to increase the size of the data sets. The duplicates from the anomaly data set as well as sequences that also existed in the reference data set were removed. The data sets from these

repositories have been used in many papers to evaluate the proposed anomaly detection techniques [7, 8, 4, 7, 12, 10, 9, 6, 5].

The other intrusion detection data repository was the *Basic Security Module* (BSM) audit data, collected from a victim Solaris machine, in the DARPA Lincoln Labs 1998 network simulation data sets [11]. The repository contains labeled training and testing DARPA data for multiple weeks collected on a single machine. For each week we constructed the reference data set using the sequences labeled as normal from all days of the week. The anomaly data set was constructed in a similar fashion. The data is similar to the system call data described above with similar (though larger) alphabet. The three data sets thus created are called *bsm-week1*, *bsm-week2*, and *bsm-week3*.

The details of the validation data sets are provided in Table 1.

Source	Data Set	$ \Sigma $	$\hat{l}$	$ \mathbf{S}^N $	$ \mathbf{S}^A $	$ \mathbf{S} $	$ \mathbf{S}^T $
UNM	snd-cert	56	803	1811	172	811	1050
	snd-unm	53	839	2030	130	1030	1050
DARPA	bsmweek1	67	149	1000	800	10	210
	bsmweek2	73	141	2000	1000	113	1050
	bsmweek3	78	143	2000	1000	67	1050

**Table 1: System call data sets used for experimental evaluation.  $\hat{l}$  – Average Length of Sequences,  $\mathbf{S}^N$  – Reference Data Set,  $\mathbf{S}^A$  – Anomaly Data Set,  $\mathbf{S}$  – Training Data Set,  $\mathbf{S}^T$  – Test Data Set.**

## 4. USING RBA FEATURES FOR SYSTEM CALL TRACE ANALYSIS AND INTRUSION DETECTION

The RBA framework is used to convert each system call trace into a multidimensional vector of continuous values. Figures 2 and 3 show the two dimensional plots using top two principal components of the transformed multidimensional data for *snd-cert* and *bsm-week3* data sets, respectively. It is evident from both figures that RBA mapping results in separation of the reference and intrusive sequences. The two figures reveal several insights regarding the two data sets, which are not apparent in the raw data. First, is that for *snd-cert* data set, the two types of sequences are different, with little overlap, while for the *bsm-week3* data set, the overlap between reference and intrusive sequences is more. This is the reason why the RBA based technique performs relatively poorly on *bsm-week3* (See Table 2). Second insight is that both data sets exhibit clusters for the reference sequences. For the *snd-cert* data set, the clusters are fewer and well separated, while for the *bsm-week3* data set, the clusters are more in number as well as closer to each other

For intrusion detection, we used the RBA features in a nearest neighbor based anomaly detection technique [14] and compared the performance against several existing techniques. We used the precision on the anomaly class as our evaluation metric; higher value indicates better performance. For more details about the existing techniques and the evaluation metric the reader is referred to our earlier work [3].

<sup>1</sup>Substrings of length  $k$  that occur in the sequence

<sup>2</sup><http://www.cs.unm.edu/~immsec/systemcalls.htm>

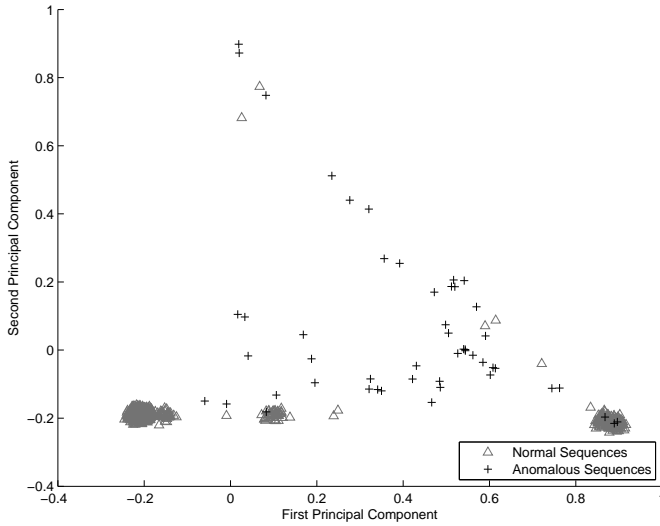


Figure 2: Reference vs. Intrusive Sequences for the *snd-cert* Data Set using RBA Mapping.

The comparative results for the precision metric are provided in Table 2. The results show that the RBA based technique either outperforms or is equal to best among all other techniques across all data sets.

	UNM		DARPA			Avg
	snd-unnm	snd-cert	bsm-week1	bsm-week2	bsm-week3	
tstd	0.58	0.64	0.20	0.36	0.60	0.63
fsa	0.82	0.88	0.40	0.52	0.64	0.70
fsaz	0.80	0.88	0.50	0.56	0.66	0.73
pst	0.28	0.10	0.00	0.10	0.34	0.31
rip	0.72	0.70	0.20	0.18	0.50	0.48
hmm	0.00	0.00	0.00	0.02	0.20	0.07
rba	<b>0.84</b>	<b>0.88</b>	<b>0.50</b>	<b>0.60</b>	<b>0.66</b>	<b>0.78</b>

Table 2: Comparing precision of RBA based technique against existing techniques.

## 5. CONCLUSIONS

In this paper, we have shown how the RBA framework can be used to analyze system call traces. Visualization of long traces is a challenge especially to highlight the presence of traces corresponding to different behaviors, such as normal vs. intrusive. RBA provides an informative visualization which not only captures the relationship between the normal and intrusive traces, but also allows one to explore other characteristics of the data, such as presence of multiple modes, separability of two types of traces, etc. Besides the visualization capabilities, RBA also allows one to transform data into a feature space in which the difference between normal and intrusive traces is highlighted and results in enhanced intrusion detection.

## 6. REFERENCES

- [1] V. Chandola. *Anomaly Detection for Symbolic Sequences and Time Series Data*. PhD thesis, University of Minnesota, Sept. 2009.
- [2] V. Chandola, S. Boriah, and V. Kumar. A framework for exploring categorical data. In *Proceedings of the ninth SIAM International Conference on Data Mining*, 2009.

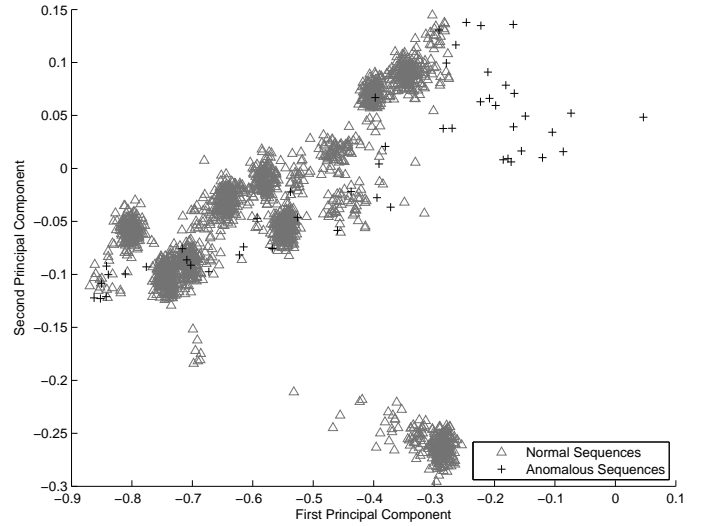


Figure 3: Reference vs. Intrusive Sequences for the *bsm-week3* Data Set using RBA Mapping.

- [3] V. Chandola, V. Mithal, and V. Kumar. A comparative evaluation of anomaly detection techniques for sequence data. In *Proceedings of International Conference on Data Mining*, 2008.
- [4] S. Forrest, C. Warrender, and B. Pearlmuter. Detecting intrusions using system calls: Alternate data models. In *Proceedings of the 1999 IEEE ISRSP*, pages 133–145, Washington, DC, USA, 1999. IEEE Computer Society.
- [5] B. Gao, H.-Y. Ma, and Y.-H. Yang. Hmms (hidden markov models) based on anomaly intrusion detection method. In *Proceedings of International Conference on Machine Learning and Cybernetics*, pages 381–385. IEEE, 2002.
- [6] F. A. Gonzalez and D. Dasgupta. Anomaly detection using real-valued negative selection. *Genetic Programming and Evolvable Machines*, 4(4):383–403, 2003.
- [7] S. A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. *Journal of Computer Security*, 6(3):151–180, 1998.
- [8] T. Lane and C. E. Brodley. Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information Systems and Security*, 2(3):295–331, 1999.
- [9] W. Lee and S. Stolfo. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, 1998.
- [10] W. Lee, S. Stolfo, and P. Chan. Learning patterns from unix process execution traces for intrusion detection. In *Proceedings of the AAAI 97 workshop on AI methods in Fraud and risk management*, 1997.
- [11] R. P. Lippmann and et al. Evaluating intrusion detection systems - the 1998 darpa off-line intrusion detection evaluation. In *DARPA Information Survivability Conference and Exposition (DISCEX) 2000*, volume 2, pages 12–26. IEEE Computer Society Press, 2000.
- [12] C. C. Michael and A. Ghosh. Two state-based approaches to program-based anomaly detection. In *Proceedings of the 16th Annual Computer Security Applications Conference*, page 21. IEEE Computer Society, 2000.
- [13] N. Nguyen and P. Reiher. Detecting insider threats by monitoring system call activity. In *IEEE Information Assurance Workshop*, pages 18–20, 2003.
- [14] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 427–438. ACM, 2000.