360° Anomaly Based Unsupervised Intrusion Detection

Stefano Zanero
Dipartimento di Elettronica e Informazione
Politecnico di Milano Technical University
via Ponzio 34/5 20133 Milano Italy

February 3, 2007

Abstract

This paper is meant as a reference to describe the research conducted at the Politecnico di Milano university on unsupervised learning for anomaly detection. We summarize our key results and our ongoing and future work, referencing our publications as well as the core literature of the field to give the interested reader a roadmap for exploring our research area.

1 Introduction

Intrusion Detection Systems are often blamed for being ineffective security measures. However, since networked computer systems are prone to be attacked and compromised, we need to monitor them for signs of intrusions: the development of good intrusion detectors is therefore a necessity.

It is well known that two complementary approaches exist in intrusion detection: in *misuse detection* systems attacks are directly defined by the means of signatures; in *anomaly detection* systems instead normality is described and deviations are consequentially flagged.

Misuse detectors are simpler to design and build, and therefore most of today's commercial IDS products are substantially misuse-based. Misuse detectors however are effective only against commonly known attack tools, for which such a signature can be available. They cannot detect "zero-day" attacks, they suffer from evasion techniques [1] and polymorphism of attacks [2]. Additionally, they are useless against insider abuse or other security violations that do not make use of exploits, such as social engineering attack effects.

An obvious solution would be to switch to an anomaly detection approach, modeling what is *normal* instead of what is *anomalous*. Not needing a database of "known" attacks, such systems can potentially detect unknown techniques and insider abuses. A number of host-based anomaly detection systems have been proposed in academic projects, but they have failed to turn into real world

systems (with a few exceptions). This is mainly due to the presence of "false positives", or false alerts. While in misuse detectors a proper configuration and tuning can avoid most "noncontextual" and irrelevant alerts, in anomaly detectors false positives can be reduced, but not totally eliminated. In second place, most anomaly detectors have a "normal/alert" outcome which does not actually tell the user what is wrong, but just alert him/her when the "abnormality" of the situation goes beyond predefined thresholds. This makes them less user friendly, and ultimately unusable for automated response and intrusion prevention.

Due to space limitation, we cannot and will not attempt to review all the previous literature on intrusion detection or to go more in depth than this. We refer however the curious reader to [3] for a more comprehensive and taxonomical review.

Our research work focuses on the analysis and development of anomaly based intrusion detection systems based on *unsupervised learning algorithms*. In this paper, we will briefly summarize the main results of our research, leading to the development of a complete, integrated suite of tools for anomaly based intrusion detection at both host and network levels. Our key original contributions have been published in international conferences [4–7], submitted to scientific journals, and have been the core of a doctoral thesis [8]. We refer the reader to such publications for further information on our research work

2 Network Intrusion Detection

Network Intrusion Detection is a particularly challenging field for the application of unsupervised learning algorithms. In particular, the varying size of the payloads of the datagrams, and their heterogeneous nature which defies a compact representation as a single feature, are the hardest problems to solve. Most existing researches on this topic avoid this problem altogether by discarding the payload and retaining only the information in the packet header, or by tracking connection-wide variables instead of analyzing single packets [9–13].

In previous works [5–8] we proposed a novel network based anomaly detection system which uses a two-tier architecture to overcome dimensionality problems and apply unsupervised learning techniques to the payload of packets, as well as to the headers. The overall architecture is shown in Fig. 1. In the first tier of the system, a Self Organizing Map (SOM) [14] operates a basic form of pattern recognition on the payload of the packets, observing one packet payload at a time and "compressing" it into a byte of information (a "payload class" value) [5, 7]. We considered performance issues and proposed improvements and heuristics to increase the throughput of SOMs by almost three times, with marginal misclassification rates, to reach a speed which is suitable for online Intrusion Detection purposes [6].

This classification is then added to a subset of the information decoded from the packet header and passed on to the second tier algorithm, which is an unsupervised algorithm for outlier detection in multivariate time series based

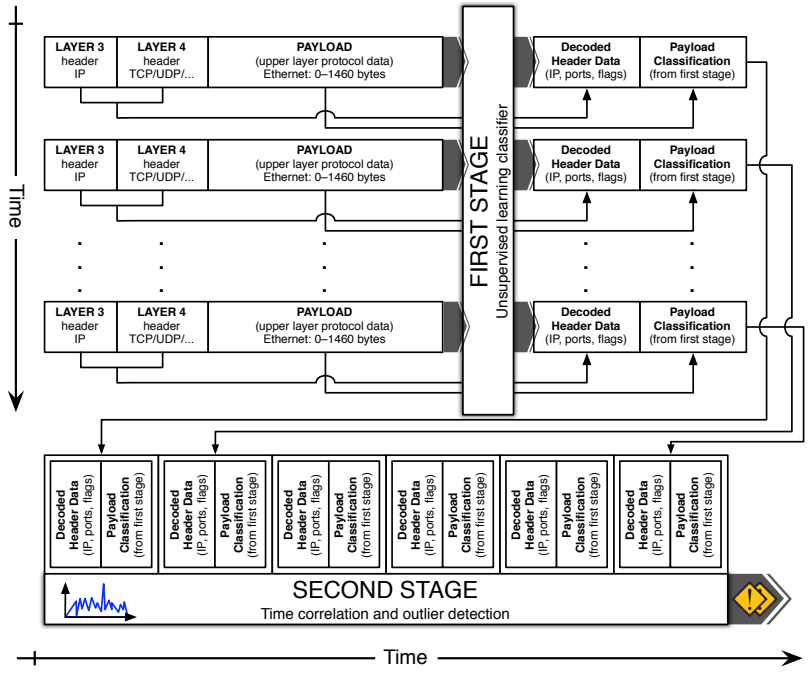


Figure 1: The overall architecture of the two-stage network-based IDS prototype.

on discounting learning (Smart Sifter [15]). The output of SmartSifter is a value expressing the statistical distance of the new observation from the former ones. In order to automatically tune the threshold beyond which a data vector is considered an outlier, we modified SmartSifter by introducing a training phase during which the distribution of the anomaly scores is approximated, and an estimated quantile of the distribution is also computed. In this way we can directly set the IDS sensitivity as the percentage of packets we want to consider as outliers [8] .

We ran the prototype over various days of the 1999 DARPA dataset. The average results are reported in Table 1. The first column contains the sensitivity threshold of the algorithm, that is, the target percentage of packets to be classified as outliers. It is also a good predictor of the False Positive Rate (FPR), if the attack rate is not too high. For a comparison, the authors of SmartSifter claim a 18% detection rate, with a 0.9% false positive rate. Our algorithm can instead reach a 92% detection rate with a 0.17% false positive rate, thus demonstrating a highly superior performance. PAYL [16] is the only other prototype we are aware of, which uses part of the payload of packets. The best overall results for PAYL show a detection rate of 58.7%, with a false positive rate that is between 0.1% and 1%. Our architecture can reach the same detection rate with a false positive rate below 0.03%, thus an order of magnitude better than

Threshold	Detection Rate	False Positive Rate
0.03%	66.7%	0.031%
0.05%	72.2%	0.055%
0.08%	77.8%	0.086%
0.09%	88.9%	0.095%

Table 1: Detection rates and false positive rates for our prototype

PAYL, or on the other hand it can reach a 88.9% detection rate with no more than a 1% rate of false positives.

3 System Call Anomaly Detector

Host based anomaly detection has been widely studied in literature. The seminal work of Denning [17], followed by others [18, 19], used purely statistical approaches, sometimes with good results. Most of these works, however, do not take into account sequential events, just system-wide variables (and they are also the works that come up to mind to most people when "anomaly detection" is named). Other studies focus on the analysis of user sessions to find masqueraders [4, 20–22]. Nowadays however interactive console access to systems is less and less used.

The first mention of intrusion detection through the analysis of the sequence of syscalls from system processes is in [23], where "normal sequences" of system calls are considered (without paying any attention to the parameters of each invocation). Variants of [23] have been proposed in [24–28]. An inductive rule generator called RIPPER [29,30] has been also proposed for analyzing sequences of syscalls and extracting rules [31] that can then be enforced for intrusion prevention purposes [32, 33].

The use of Markov chains as a simple, short range correlation model was also proposed, e.g in [34–36]. In [4] we proposed a bayesian framework for *behavior detection* using Markov models.

Alternatively, other authors proposed to use static analysis, as opposed to dynamic learning, to profile a program normal behavior [37, 38].

Curiously enough, none of these methods analyzes either the arguments of the system calls. This is due to the inherent complexity of the task, in a similar way to what we saw before for Network IDS and packet payloads. Two recent research works began to focus on this problem. In [39] a number of models are introduced to deal with the most common arguments. This is the work we discuss in depth and extend in our paper. In [40] an alternative framework is proposed, using the LERAD algorithm (Learning Rules for Anomaly Detection) which mines rules expressing "normal" combinations of arguments. Strangely, neither work uses the concept of sequence analysis. A concept named "Resilience" has also recently been introduced [41], involving the mapping of arguments of system calls as multidimensional data points. However, this approach is still in the

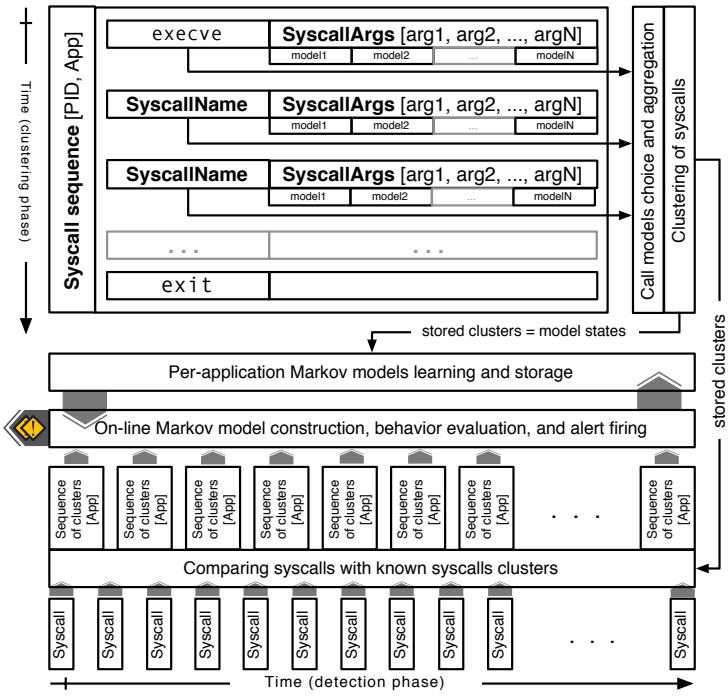


Figure 2: The overall architecture of the host-based IDS prototype.

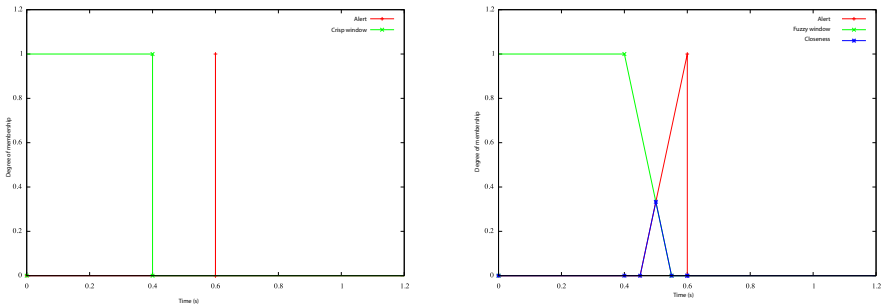
early stages of development.

In [8] we described a tool that can detect anomalies by analyzing system call arguments and sequences. The system is an almost complete re-engineering of SyscallAnomaly [39]. In particular, our prototype implements some of the ideas of SyscallAnomaly along with Markovian based modeling, clustering and behavior identification outperforming the original application with both an increased DR and a reduced FPR.

The overall architecture is drawn in Fig. 2: a hierarchical clustering algorithm is used to identify groups of similar syscalls (for details see [8]); the resulting clusters become the nodes of a Markov chain built to characterize the behavior of each application on the system in terms of syscall sequences. Anomaly thresholds are also learned directly from the training data.

During the detection phase, each system call is associated to a cluster, and the likelihood of its arguments is calculated against the models of that cluster. The probability of the last transition and the cumulative probability of the sequence are calculated using the Markov model, implementing a correction algorithm in order to avoid the assignment of low probabilities. Model probabilities are calculated on-line, and compared to stored thresholds.

Calls whose *arguments* are anomalous, or whose *presence* is anomalous in that position; and sequences that are *overall unlikely* are flagged with alerts.



(a) (b)

Figure 3: Comparison of crisp (a) and fuzzy (b) time-windows

4 Correlation and aggregation approaches

We propose also (in a work still under review for publication, and which cannot therefore be explained thoroughly here) a process of *alert fusion* suitable for our anomaly detectors.

Alert fusion is the *correlation* of *aggregated* streams of alerts. The *aggregation* of two alerts, reported by the same IDS, is the grouping of alerts that have similar features and are close in time. On the other hand, alert *correlation* means recognizing logically linked alerts. The desired output of an *alert fusion* process is a compact, high-level view of what is happening into the network as a whole. In the security field, such a process is also known as *security information monitoring*. We refer the reader to the model in [42] for a more detailed taxonomy.

Time-distance aggregation might be able to catch simple scenarios like remote attacks against remote applications (e.g., web servers) vulnerabilities. This kind of anomalies have evidence in both network and host activities. This also helps to reduce the number of redundant alerts, since many IDS report the same attack raising more than one alert, depending on the specific analyzer implementation. We propose to use fuzzy measures [43] and fuzzy sets [44] to design more robust aggregation algorithms. The use of fuzzy sets allows us to precisely define a time-distance criterion, which in addition can handle unavoidable errors such as delayed detections.

As can be seen in Fig. 3, using fuzzy sets delayed detections can be modeled using a triangle-shaped set instead of a singleton. We can also use a trapezoidal fuzzy set instead of a crisp window, resulting in a more robust distance measurement and time window definition. In the example, simple triangles and trapezoids have been used: however, more accurate/complex membership functions could be used as well.

If we have two alert streams (network and host alerts) in input, the output should be one stream of uncorrelatable alerts (i.e., already aggregated alerts plus the ones that are not correlated). In a first version of the post-processing

procedure, we chose to classify uncorrelated alerts (network alerts without host verification and vice-versa) as false-positives and discard them. Experimentally it can be shown that this approach is too pessimistic and could lead to discarding true positive. We used the deviation of the anomaly value from the threshold as an indication of "belief" that an attack took place, and the false positive rate as a measure of systematic "disbelief" [43, 44] of the detector. Using them in an appropriate scaling function, we can keep alerts that, albeit uncorrelated, have a strong belief supporting them.

Since alert correlation is a relatively new problem, evaluation techniques are limited to a few approaches [45]. Thus the development of solid testing methodologies is needed from both the theoretical and the practical points of view. To evaluate our approach, we used the following metrics: since the main goal is to reduce the amount of alerts without discarding true positives, the DR should ideally not decrease while the FPR should be reduced as much as possible. From our experiments, the fuzzy approach with the belief/misbelief correction shows the best performances.

The **correlation phase** is even more challenging, because it commonly needs *a priori* knowledge which we do not have; for instance, precise information about attacks names, division of attacks into classes, and alert priorities. Most previous approaches use formalizations and/or signatures [46, 47].

Statistical techniques have been also proposed. The current version of EMERALD [48] implements a so-called probabilistic alert correlation engine. Described in details in [49], the approach relies on the definition of some similarity metrics between alerts; the correlation phase calculates a weighted similarity value and finds "near" alerts to be fused together. The features used include the source IDS identifiers, timestamps, the alert thread, source and destination addresses and ports. Association rule mining techniques have been used [50–52] in order to learn recurrent alert sequences for unsupervised alert scenario identification.

Classic time-series modeling and analysis have been also applied. The approach detailed in [53] constructs alert time-series counting the number of events occurring into fixed-size sampling intervals; authors then exploits trend and periodicity removal techniques in order to filter out predictable components and leave *real* alerts only as the output. The main shortcoming of this approach is the need for *long* alert streams in order to be effective.

Other authors proposed [54] to implement a Granger statistical causality test. Without going into details, the test is based on a *causality statistic* which quantifies how much of the history of a given set of alerts is needed to explain the evolution of another set of alerts. Repeating the procedure for each couple of set allows to identify "causally related" events and to reconstruct scenarios in an unsupervised fashion. However, our experiments showed how the Granger causality test is heavily dependent on the tuning of configuration parameters, making it quite unreliable. We are working on improved tests for detecting correlation in an unsupervised manner among sets of alerts.

5 Conclusions and future works

In this paper we summarized the core results of the research conducted at the Politecnico di Milano university on unsupervised learning for anomaly detection. We also described part of our ongoing and future work, referencing our own publications as well as the reference literature of the field.

We described a network intrusion detection system totally based on unsupervised learning, which uses unsupervised payload clustering and classification techniques that enables an effective outlier detection algorithm to flag anomalies. We also described a host anomaly detector, that exploits the analysis of both system calls arguments and behavior,

Finally we showed how we are working to integrate both prototypes to create a fully unsupervised, integrated intrusion detection environment. We analyzed previous literature about alert fusion (i.e., aggregation and correlation), and found that effective techniques have been proposed, but they are not really suitable for anomaly detection, because they require a priori knowledge (e.g., attack names or division into classes) to perform well. To overcome this we successfully exploited fuzzy sets and measures to aggregate alerts reported by the two IDS. Our experiments showed that the proposed fuzzy aggregation approach is able to decrease the FPR at the price of a small reduction of the DR (a necessary conseguence).

We also showed preliminary results on the use of the Granger causality test to recognize logically linked alerts, also giving a statistical quantification of the degree of "causality". Even if the method does not require a priori knowledge, we identified a significant issue in the fact that the statistical test relies on non-obvious configuration parameters which values significantly affect the final result.

Future extensions of this work will investigate better algorithms for correlation of anomaly based alerts. We are also trying to allow a human expert to refine the training of the system, with a "semi-supervised" approach. Additionally, we need to enhance the amount of information a human operator can get from the system, and to make it more user friendly and actionable. Another possible extension of this work is the investigation of algorithms and criteria to correlate anomaly and misuse-based alerts together, in order to bridge the gap between the existing paradigms of intrusion detection.

Finally, we noted throughout our works that the evaluation of an intrusion detection system is a difficult and open research topic [55]. It is very difficult to plan tests for the different performance metrics of an IDS system (such as throughput, detection capabilities, etc.), and it is even more difficult to combine these tests in a meaningful, overall evaluation. The only available dataset for IDS evaluation, the so-called "DARPA IDS Evaluation dataset", has a number of known shortcomings in the network data samples [56, 57]. In [8] we also make it evident that the host based traces suffer from similar issues. Furthermore, the dataset is outdated and the attack scenarios are too simple. Therefore we think it is high time to study and create a more sound methodology for evaluating and testing intrusion detection systems. We are designing a toolset

for generating synthetic traffic and superimposing attacks, and we will try to develop a methodology for evaluation which is both scientifically repeatable and sound with respect to real world usage requirements.

Personal and Formal Acknowledgments

Prof. Sergio Savaresi, Dr. Matteo Matteuci and Federico Maggi worked with me on most of this research, and I need to thank them gratefully. Most of this work was supported by the Italian Ministry of Education and Research under the FIRB Project "Performance evaluation for complex systems", in the research unit led by my advisor, prof. Giuseppe Serazzi, whose continual support I gratefully acknowledge. A huge number of people contributed a comment, an idea, or worked on the systems during the years: without making a very long list, I need to thank each one of you. You know who you are.

References

[1] Thomas H. Ptacek and Timothy N. Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical Report T2R-0Y6, Secure Networks, Calgary, Canada, 1998.

[2] G. Vigna, W. Robertson, and D. Balzarotti. Testing Network-based Intrusion Detection Signatures Using Mutant Exploits. In *Proceedings of the ACM Conference on Computer and Communication Security (ACM CCS)*, pages 21–30, Washington, DC, October 2004.

[3] Rebecca Gurley Bace. *Intrusion detection*. Macmillan Publishing Co., Inc., Indianapolis, IN, USA, 2000.

[4] Stefano Zanero. Behavioral intrusion detection. In Cevdet Aykanat, Tugrul Dayar, and Ibrahim Korpeoglu, editors, *Proceedings of ISCIS 2004*, volume 3280 of *Lecture Notes in Computer Science*, pages 657–666, Kemer-Antalya, Turkey, October 2004. Springer.

[5] Stefano Zanero. Analyzing tcp traffic patterns using self organizing maps. In Fabio Roli and Sergio Vitulano, editors, *13th International Conference on Image Analysis and Processing - ICIAP 2005*, volume 3617 of *Lecture Notes in Computer Science*, pages 83–90, Cagliari, Italy, September 2005. Springer.

[6] S. Zanero. Improving self organizing map performance for network intrusion detection. In *SDM 2005 Workshop on "Clustering High Dimensional Data and its Applications"*, 2005.

[7] Stefano Zanero and Sergio M. Savaresi. Unsupervised learning techniques for an intrusion detection system. In *Proc. of the 2004 ACM Symposium on Applied Computing*, pages 412–419. ACM Press, 2004.

[8] Stefano Zanero. *Unsupervised Learning Algorithms for Intrusion Detection*. PhD thesis, Politecnico di Milano T.U., Milano, Italy, May 2006.

[9] M.V. Mahoney and P.K. Chan. Detecting novel attacks by identifying anomalous network packet headers. Technical Report CS-2001-2, Florida Institute of Technology, 2001.

[10] Calvin Chow. Parzen-Window network intrusion detectors. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 4*, pages 385–388, Washington, DC, USA, aug 2002. IEEE Computer Society.

[11] K. Labib and R. Vemuri. NSOM: A real-time network-based intrusion detection system using self-organizing maps. Technical report, Dept. of Applied Science, University of California, Davis, 2002.

[12] M. V. Mahoney and P. K. Chan. A machine learning approach to detecting attacks by identifying anomalies in network traffic. Technical Report CS-2002-08, Florida Institute of Technology, 2002.

[13] M. V. Mahoney. Network traffic anomaly detection based on packet bytes. In *Proceedings of the 19th Annual ACM Symposium on Applied Computing*, 2003.

[14] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 3 edition, 2001.

[15] K. Yamanishi, J.-I. Takeuchi, G. J. Williams, and P. Milne. Online unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Knowledge Discovery and Data Mining*, 8(3):275–300, 2004.

[16] Ke Wang and Salvatore J. Stolfo. Anomalous payload-based network intrusion detection. In *RAID Symposium*, September 2004.

[17] D. E. Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, February 1987.

[18] Mark Burgess, Hârek Haugerud, Sigmund Straumsnes, and Trond Reitan. Measuring system normality. *ACM Trans. Comput. Syst.*, 20(2):125–160, 2002.

[19] N. Ye and Q. Chen. An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Quality and Reliability Engineering International*, 17(2):105–112, 2001.

[20] Jake Ryan, Meng-Jang Lin, and Risto Miikkulainen. Intrusion detection with neural networks. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[21] H. Debar, M. Becker, and D. Siboni. A neural network component for an intrusion detection system. In *Proc. IEEE Symposium on Research in Computer Security and Privacy*, 1992.

[22] M. Theus and M. Schonlau. Intrusion detection based on structural zeroes. *Statistical Computing & Graphics Newsletter*, 9:12–17, 1998.

[23] Stephanie Forrest, Steven A. Hofmeyr, Anil Somayaji, and Thomas A. Longstaff. A sense of self for Unix processes. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, Washington, DC, USA, 1996. IEEE Computer Society.

[24] Stephanie Forrest, Alan S. Perelson, Lawrence Allen, and Rajesh Cherukuri. Self-nonself discrimination in a computer. In *SP '94: Proceedings of the 1994 IEEE Symposium on Security and Privacy*, page 202, Washington, DC, USA, 1994. IEEE Computer Society.

[25] J. B. D. Cabrera, L. Lewis, and R.K. Mehara. Detection and classification of intrusion and faults using sequences of system calls. *ACM SIGMOD Record*, 30(4), 2001.

[26] G. Casas-Garriga, P. Díaz, and J.L. Balcázar. ISSA: An integrated system for sequence analysis. Technical Report DELIS-TR-0103, Universitat Paderborn, 2005.

[27] Intrusion Detection Using Sequences of System Calls. S. hofmeyr and s. forrest and a. somayaji. *Journal of Computer Security*, 6:151–180, 1998.

[28] Anil Somayaji and Stephanie Forrest. Automated response using system–call delays. In *Proceedings of the 9th USENIX Security Symposium*, Denver, CO, August 2000.

[29] William W. Cohen. Fast effective rule induction. In Armand Prieditis and Stuart Russell, editors, *Proc. of the 12th International Conference on Machine Learning*, pages 115–123, Tahoe City, CA, Jul 1995. Morgan Kaufmann.

[30] Y. Chevaleyre, N. Bredeche, and J. Zucker. Learning rules from multiple instance data : Issues and algorithms. In *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU02)*, Annecy, France, 2002.

[31] Wei Fan, Matthew Miller, Salvatore J. Stolfo, Wenke Lee, and Philip K. Chan. Using artificial anomalies to detect unknown and known network intrusions. In *ICDM*, pages 123–130, 2001.

[32] N. Provos. Improving host security with system call policies. Technical Report 02-3, CITI, November 2002.

[33] Suresh N. Chari and Pau-Chen Cheng. Bluebox: A policy-driven, host-based intrusion detection system. *ACM Trans. Inf. Syst. Secur.*, 6(2):173–200, 2003.

[34] Andrew P. Kosoresow and Steven A. Hofmeyr. Intrusion detection via system call traces. *IEEE Softw.*, 14(5):35–42, 1997.

[35] Christina Warrender, Stephanie Forrest, and Barak A. Pearlmutter. Detecting intrusions using system calls: Alternative data models. pages 133–145, 1999.

[36] S. Jha, K. Tan, and R. A. Maxion. Markov chains, classifiers, and intrusion detection. In *CSFW '01: Proceedings of the 14th IEEE Workshop on Computer Security Foundations*, page 206, Washington, DC, USA, 2001. IEEE Computer Society.

[37] David Wagner and Drew Dean. Intrusion detection via static analysis. In *SP '01: Proceedings of the 2001 IEEE Symposium on Security and Privacy*, page 156, Washington, DC, USA, 2001. IEEE Computer Society.

[38] Jonathon T. Giffin, David Dagon, Somesh Jha, Wenke Lee, and Barton P. Miller. Environment-sensitive intrusion detection. In *RAID*, pages 185–206, 2005.

[39] C. Kruegel, D. Mutz, F. Valeur, and G. Vigna. On the Detection of Anomalous System Call Arguments. In *Proceedings of the 2003 European Symposium on Research in Computer Security*, Gjøvik, Norway, October 2003.

[40] G. Tandon and P. Chan. Learning rules from system call arguments and sequences for anomaly detection. In *ICDM Workshop on Data Mining for Computer Security (DMSEC)*, pages 20–29, 2003.

[41] Dave Aitel. Resilience. `http://www.immunitysec.com/resources-papers.shtml`, February 2006.

[42] Fredrik Valeur. A comprehensive approach to intrusion detection alert correlation. *IEEE Trans. Dependable Secur. Comput.*, 1(3):146–169, 2004. Member-Giovanni Vigna and Member-Christopher Kruegel and Fellow-Richard A. Kemmerer.

[43] Zhenyuan Wang and George J. Klir. *Fuzzy Measure Theory*. Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[44] George J. Klir and Tina A. Folger. *Fuzzy sets, uncertainty, and information*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987.

[45] Joshua Haines, Dorene Kewley Ryder, Laura Tinnel, and Stephen Taylor. Validation of sensor alert correlators. *IEEE Security and Privacy*, 01(1):46–56, 2003.

[46] S. Eckmann, G. Vigna, and R. Kemmerer. STATL: An attack language for state-based intrusion detection. In *Proceedings of the ACM Workshop on Intrusion Detection*, Atene, November 2000.

[47] Steven J. Templeton and Karl Levitt. A requires/provides model for computer attacks. In *NSPW '00: Proceedings of the 2000 workshop on New security paradigms*, pages 31–38, New York, NY, USA, 2000. ACM Press.

[48] P. A. Porras and P. G. Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In *Proc. 20th NIST-NCSC Nat'l Information Systems Security Conf.*, pages 353–365, 1997.

[49] Alfonso Valdes and Keith Skinner. Probabilistic alert correlation. In *RAID '00: Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, pages 54–68, London, UK, 2001. Springer-Verlag.

[50] Klaus Julisch and Marc Dacier. Mining intrusion detection alarms for actionable knowledge. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 366–375, New York, NY, USA, 2002. ACM Press.

[51] O. Dain and R. Cunningham. Fusing heterogeneous alert streams into scenarios. In *Proc. of the ACM Workshop on Data Mining for Security Applications*, pages 1–13, November 2001.

[52] Hervé Debar and Andreas Wespi. Aggregation and correlation of intrusion-detection alerts. In *RAID '00: Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, pages 85–103, London, UK, 2001. Springer-Verlag.

[53] Jouni Viinikka, Hervé Debar, Ludovic Mé;, and Renaud Séguier. Time series modeling for ids alert management. In *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 102–113, New York, NY, USA, 2006. ACM Press.

[54] Xinzhou Qin and Wenke Lee. Statistical causality analysis of infosec alert data. In *RAID*, pages 73–93, 2003.

[55] Stefano Zanero. My ids is better than yours... or is it ? In *Blackhat Federal 2006 Briefings*, 2006.

[56] John McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. on Information and System Security*, 3(4):262–294, 2000.

[57] M. V. Mahoney and P. K. Chan. An analysis of the 1999 DARPA / Lincoln laboratory evaluation data for network anomaly detection. In *Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003)*, pages 220–237, Pittsburgh, PA, USA, September 2003.