

Intrusion Detection Using Radial Basis Function Network on Sequences of System Calls

Arvind Rapaka, Alexander Novokhodko, Donald Wunsch
Applied Computational Intelligence Laboratory
University of Missouri-Rolla
Department of Electrical and Computer Engineering
1870 Miner Circle
Rolla, MO 65409-0040
{anrb6b, ayn, dwunsch}@umr.edu

Abstract— Over the past few years, security has been an increasing concern, with the growth of network and technological development. An intrusion detection system is a critical component for secure information management. Unfortunately, present IDS's falls short of providing protection required for growing concern. Creation of an IDS to detect anomaly intrusions, in a timely and accurate manner, has been an elusive goal for researchers.

This paper describes a host-based IDS model, utilizing a Radial Basis Function neural network. It functions as a combined anomaly/misuse detector that helps to overcome most of the limitations in existing models. Rather than creating user profiles or behavioral characteristics, we trained our network using session data in the identification and tested experimentally on different attack/normal sessions. These results suggest that training the IDS on session data is not only effective in detecting intrusions, but also accurate and timely.

I. INTRODUCTION

As modern networks continue to grow, evaluating and detecting vulnerabilities to malicious attacks becomes more critical for e-driven-business models. Our network platforms, diverse applications, technologies and packages are growing increasingly complex. Inevitably our systems are insecure, flawed in practice and vulnerable. Given the seriousness of the problem, we need both deterrence, and termination of attacks that could prove detrimental to civilization. The security assessment problem stems from constantly changing behavior of software. Unfortunately, many of these changes happen without warning.

Therefore, we need to depend on secondary measures and tools such as IDS (Intrusion Detection System). The IDS approach to security is based on the assumption that a system will not be secure, but the violations of security policy (intrusions) can be detected by monitoring and analyzing system behavior [1].

Within the past few years, there has been steady improvement in both research and demand for effective IDS. Many research groups have proposed different methods of implementations, which proactively monitor behaviors

defining the benchmark for the system. Though it is critical to profile normalcy and anomaly, it often proves detrimental to the health of the system, keeping in mind the complexity and the dynamic nature of today's computer systems.

Our results focus on generalizing the attacks for effective, fast and high detection rate with low false alarms. We first introduce the problem of creating an intrusion detection system. In section III, we review related work in intrusion detection systems. In section IV, we discuss the need and effective methods for data reduction and refinement, and our approach in data processing. In section V, we briefly discuss an implementation model, justifying the choice. Finally, we show our results obtained with the application.

II. INTRUSION DETECTION SYSTEM

Intrusion Detection Systems are normally categorized into misuse detection and anomaly detection. In *misuse detection* systems, it refers to known attacks that exploit the system. They can match the pattern on single events or multiple combinations of events. In single event pattern matching, each event is compared with a known signature in the databank. In multiple event pattern matching, it doesn't have a uniform abstract algorithm, because they do not propose the same operators to combine events. The inherent disadvantage is inability to detect an attack that is deviated or unknown to the databank. Another reproach to misuse detecting systems is their limited scope, adding attack signatures and maintaining the attack database. But low positive /negative false alarm has led to its existence [2, 3].

Anomaly detection refers to statistical knowledge about normal activity. Intrusions correspond to deviations from the normal activity of system. These anomaly detection IDS were bogged down by the difficulty in defining the normal activity because of the high variability in nominal usage. As a result, the false positive/ negative alarm rate is high, compared to misuse detection systems. However, it is more effective in detecting new attacks or deviation from the nominal usage.

Furthermore, the IDS is classified based on the data source: Network IDS (NIDS) and Host-based IDS (HIDS) systems. The NIDS watch network traffic usually from one location or network interface. Therefore, NIDS can detect probes, scans, malicious and anomalous activity across the

Partial support for this research from the National Science Foundation, from Sandia National Laboratories, and from the M.K. Finley Missouri endowment, is gratefully acknowledged.

whole subnetwork. It is also effective in identifying general traffic patterns for network and troubleshooting network problems. Its susceptibility to generate false alarms, as well as its inability to detect false negatives is its inherent weakness. However, HIDS technology does not have the benefits of watching the network to identify patterns like NIDS does. Instead, it watches the traces to access servers through the log data. A recommended combination of host and network intrusion detection systems, in which a NIDS is placed at the network entry point and an HIDS at critical servers, is the best way to significantly reduce risk.

Current intrusion detection systems are futile to cope with new, elegant and structured attacks, due to sever practical and theoretical limitations. These limitations have lead many researchers to apply different machine learning approaches such as neural networks [4, 5, 6, 7, 8].

We implement a neural network approach, using Radial Basis Functions, to detect novel attacks, reducing false positive and negative alarms. Our research aims at a data-driven view, to consider our system as a data analysis engine, fast and accurate enough for a real time model.

III. RELATED WORK

There are many different architectural designs for Misuse/Anomaly IDS using program and user profile behavior. Program and user profiles are built by capturing the system events under normal operational condition [1, 9, 10]. Once these profiles are created, provided they represent the nominal behavior signature, they are used to classify the deviation from the corresponding nominal behavior. Many research groups developed profile based classification for anomaly detection. Others used TCP/IP network data, collected by the network sniffer, to monitor the potential attack [10, 11].

Different techniques were proposed to classify the data based on statistical approach, machine learning, simple comparison based technique, integrity checking, data mining, state machine analysis [1, 9, 10, 11, 12, 13]. But one of the key drawbacks is their inability to generalize the anomaly from the data collection process. Hence most of the techniques suffered from the high false alarms due to low threshold for tolerance of anomalous behavior [11].

In host based systems, despite IDS being the major consumer of audit trail, it is apparent that no major operating system supports the all essential needs described above for development of IDS. Many audit data contains redundant, complex and irrelevant data, which requires vigorous screening before inputting to IDS. And also if the operating system will not provide protection to audit trails, the intruder can erase traces. A major loophole in BSM Audit Trail is inability to correlate the events; in turn the intruder would be successful in developing stealth attacks by hiding footprints. Thus we need to develop an intelligent screen activity to avoid stealth attacks. So these limitations make severe dents in the development of an effective IDS, despite elegant

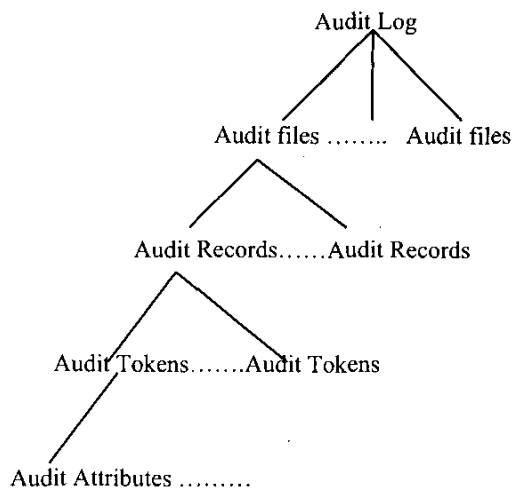


Fig. 1. Structure of Sun BSM Audit Trail

techniques for classification, and also requires effective data processing of existing audit records.

IV. DATA ANALYSIS

We have collected our BSM raw data from MIT's Lincoln Laboratory under DARPA 1998 Intrusion Detection Evaluation Program [14].

The Sun's Solaris Basic Module provides features in compliance with the TCSEC C2 trusted system rating. The BSM data include logs and events specific to users and the system.

The BSM audit trails, a binary file, include detail information about the system events attributable to a user. BSM recognizes more than 240 built-in system signals [13, 15]. In host based IDS design BSM audit trails been extensively used in both commercial and research development. "Praudit", a software tool, translates the binary BSM audit trail to readable format. Once the audit file has been translated we can parse, and can use expert knowledge to extract meaningful features and shed which do not make any difference rather consume processing time.

Audit records are described as either kernel level or user level generated records, depending on the nature of the event described in the event. The user level generated audit records are created by the applications that operate outside the kernel, unlike kernel level generated records. Each user-level and kernel-level events contain header, subject and return tokens whereas the trailer, group and sequence tokens are optional depending on the audit policy.

The event in Fig. 2 is logged when an intruder wants to execute malicious script as "root" through "ps" setuid command in UNIX model kernel. Each audit entry is encapsulated by a header and a trailer, the header line contains token id (header), the byte count (140), the version number (2), the event type (execve), the event modifier (blank), the time of the record, the milliseconds of time. The

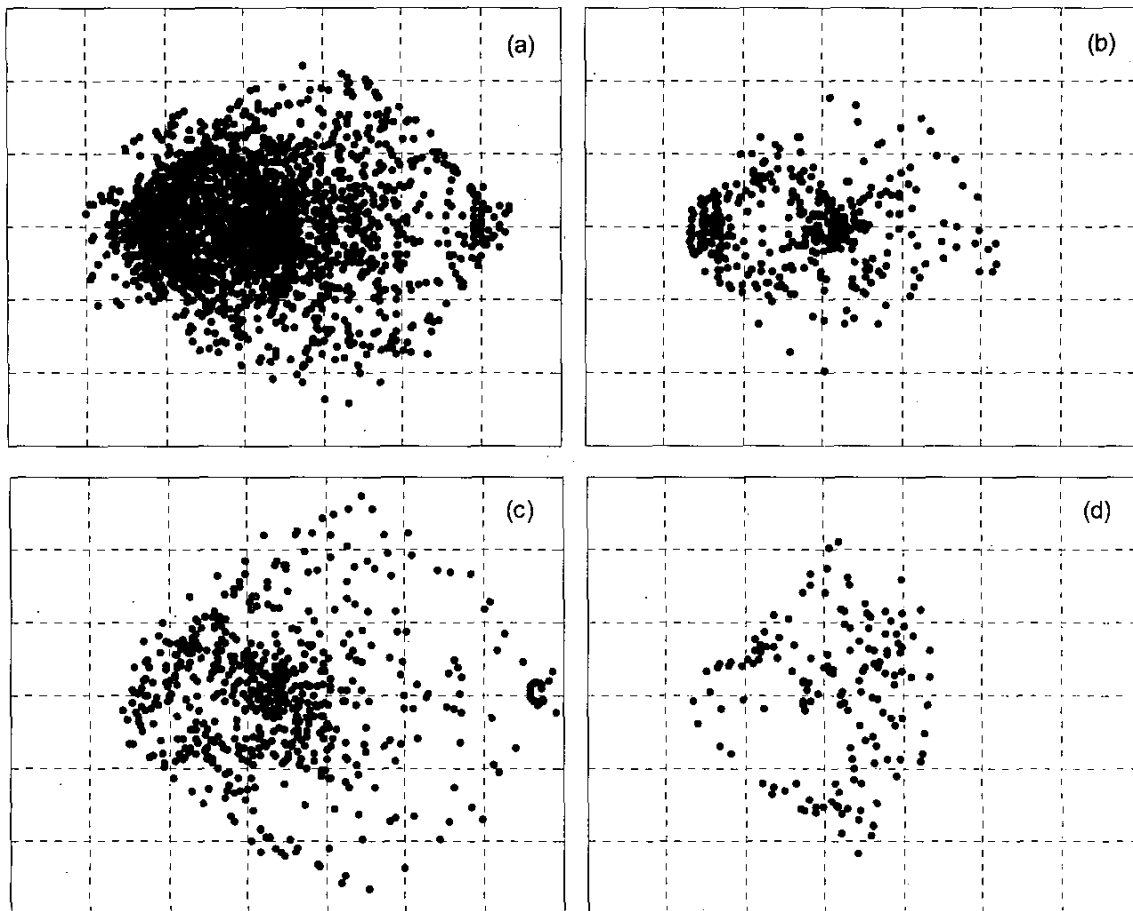


Fig. 3. System call sequences projected on the first two principal components.
 (a) sequences appearing only in "clean" sessions; (b) sequences appearing in both "clean" sessions and sessions with attacks;
 (c) attacks in the first week; (d) attacks in the second week.

V. NEURAL NETWORKS AS CLASSIFIERS

The ability of Artificial Neural Network to classify even if the data is complex, non-deterministic makes an effective implementation model for IDS. The sophisticated new attacks require a flexible and adaptable system that is capable to detect and generalize on the previous learning. A neural network model could potentially address such problems where unstructured data is needed to be classified. On the other hand, building neural network based IDS requires input and output data set and an appropriate architecture and training algorithm. The complexity and effectiveness is determined by the number of neurons in the network and functional relationship for interconnection. The ability of classify in response to the multifarious training data increases with the number of available neurons, where each neuron represents a computational degree of freedom available to the network.

Due to the characteristics of temporal clustered data, we have chosen a radial-basis function network [18]. The radial-basis function network involves three layers with different roles. The input layer is made up of source nodes that connect the network to its data processing unit. The output of the first-layer neurons, each of which represents the radial-basis function, is determined by the distance between the network input and the center of the basis function. The second layer applies a nonlinear transformation from the input space to the hidden space. The output layer is linear and produces a weighted sum of the outputs of the second layer.

An important feature of RBF network is that it is capable to construct local approximations to nonlinear input-output mappings. Thus, an RBF-based IDS can learn multiple local clusters of known attacks and clean data, and perform both misuse and anomaly detection at once. Given an input pattern, the network can check it against patterns it learned before, and make an educated guess as to where this input combination belongs.

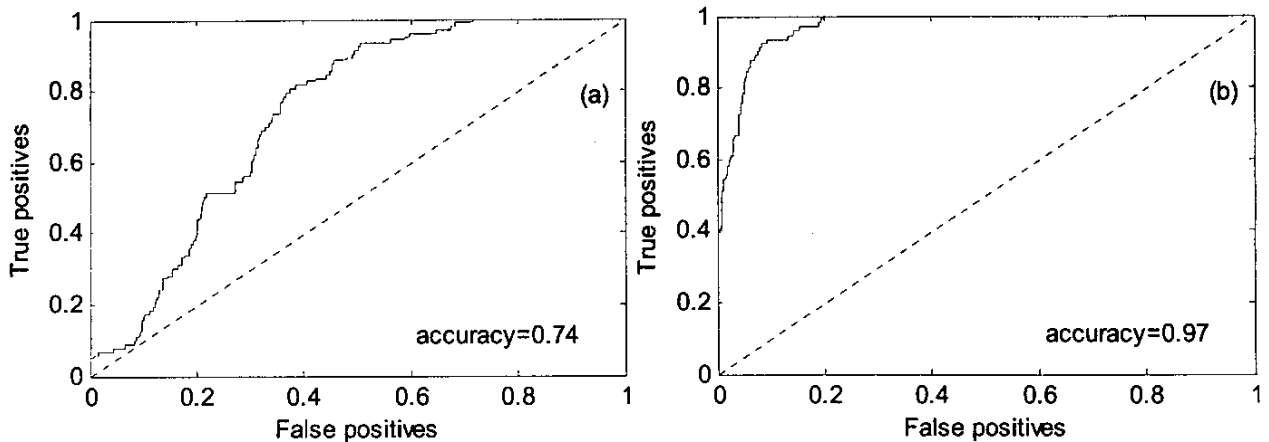


Fig. 4. Receiver operating characteristic curves for IDS trained on (a) the first week data only, and (b) the first week data and the second week "clean" data (2000 centers, spread 0.16).

To model the network we used function 'newrb' from MATLAB Neural Network toolbox [19]. This function builds the network incrementally, adding one center at a time, until the specified performance goal or number of the centers is reached. The centers are located at the positions of input vectors. Each iteration adds a center for the input with the greatest error. We trained from 32 to 2000 kernels with spreads between 0.06 and 8. The first experiment used the week one data for training and the week two data for testing; that is, the network was tested on the intrusion and clean data it had never seen. This allows to evaluate generalization and anomaly detection capabilities of the design. The ROC accuracy for different number of kernels and spreads is summarized in Table 1. An ROC curve for the best result we obtained is shown in Fig. 4(a). In the second experiment we amended the training set with the normal data from the week two; thus, in the test the network had to recognize only the unknown attack among the known normal data. As a result, the detection accuracy improved dramatically (Fig. 4(b)).

TABLE I
ROC ACCURACY FOR DIFFERENT SPREADS AND CENTER NUMBERS

Centers	Spread									
	0.06	0.13	0.16	0.19	0.25	0.5	1	2	4	8
2000	0.69	0.73	0.74	0.73	0.72	0.64	0.55	0.51	0.54	0.50
1500	0.66	0.72	0.73	0.72	0.71	0.65	0.57	0.56	0.57	0.52
1000	0.62	0.68	0.69	0.70	0.68	0.65	0.58	0.53	0.53	0.49
500	0.59	0.68	0.67	0.67	0.65	0.67	0.58	0.49	0.50	0.45
250	0.56	0.63	0.63	0.63	0.67	0.70	0.61	0.62	0.63	0.62
125	0.55	0.63	0.62	0.65	0.65	0.68	0.68	0.68	0.70	0.70
64	0.53	0.62	0.63	0.62	0.64	0.68	0.70	0.67	0.69	0.67
32	0.53	0.63	0.63	0.60	0.67	0.64	0.65	0.64	0.65	0.62

VI. CONCLUSION

The key feature in this experiment was successful detection, using session data, to achieve high detection rate and still timely. The method, using system calls feature vectors, proved successful. Our training was not based on behavioral analysis, but rather, was to generalize the anomaly.

Due to temporally local characteristics, we have chosen RBF for our application. This prototype successfully demonstrated the capability as a data filter that highlights both anomalous intrusions and normal data on the learning patterns. Moreover, the deviations from normal behavior seem to be diagnosed quickly by our prototype. This capability is important, since our goal was to detect intrusion timely.

Experiments provide evidence that a careful audit reduction on session data would enable a high detection rate that could fit into a real time model.

The next step forward for this research would be to extend the feature vector to make IDS robust. As we are more interested in analysis of session data and results, there needs to be a thorough analysis on optimal window size. Also a hybrid network/host-based approach would be highly recommended, thereby increasing detection rate and decreasing the audit reduction processing.

ACKNOWLEDGMENT

We are thankful to Dr. Tim Draelos, Sandia National Laboratories, for helpful discussions.

REFERENCES

- [1] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of Computer Security*, vol. 6, no. 3, pp. 151–180, 1998.

- [2] J. Cannady, "The Application of Artificial Neural Networks to Misuse Detection: Initial Results," *Proceedings of the Recent Advances in Intrusion Detection '98 Conference (RAID'98)*, pp. 31-47, 1998.
- [3] J. Cannady, "Artificial Neural Networks for Misuse Detection," *Proceedings of the 1998 National Information Systems Security Conference (NISSC'98)* October 5-8 1998. Arlington, VA, pp. 443-456, 1998.
- [4] H. Debar, B. Dorizzi, "An application of a recurrent network to an intrusion detection system," *Proceedings of the 1992 IEEE International Conference on Neural Networks*, Vol. 2, pp. 478-483, 1992.
- [5] J. Ryan, M. J. Lin, and R. Mjikkulainen, "Intrusion detection with neural networks," *Proceedings of the 10th Advances in Neural Information Processing Systems Conference*, Denver, CO, 1998.
- [6] L. Didaci, G. Giacinto, and Fabio Roli, "Ensemble Learning for Intrusion Detection in Computer Networks," *Proceedings of VIII Conference of AIIA, Siena, Italy, Sep. 10-13, 2002*. <http://www-dii.ing.unisi.it/aiaa2002/paper/APAUT/didaci-aiaa02.pdf>
- [7] A. K. Ghosh and A. Schwartzbard, "A Study in Using Neural Networks for Anomaly and Misuse Detection," *Proceedings of the 8th USENIX Security Symposium*, Washington, D.C., USA, August 23-26, 1999.
- [8] S. Mukkamala, G. Janoski, A. Sung, "Intrusion detection using neural networks and support vector machines," in *Proceedings of the 2002 International Joint Conference on Neural Networks, 2002, (IJCNN '02)*, vol. 2, pp. 1702-1707, 2002.
- [9] A. K. Ghosh, J. Wanken and F. Charron, "Detecting anomalous and unknown intrusions against programs," *Proceedings of the 1998 Annual Computer Security Applications Conference (ACSAC'98)*, pp. 259-267, 1998.
- [10] R. Sekar, Y. Guang, S. Verma, and T. Shanbhag, "A High-Performance Network Intrusion Detection System," in *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pp. 8-17, Nov. 2-4, 1999.
- [11] S. C. Lee and D. V. Heinbuch, "Training a neural-network based intrusion detector to recognize novel attacks," *Part A, IEEE Transactions on Systems, Man and Cybernetics*, Vol. 31-4, pp. 294-299, 2001.
- [12] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," in *Proceedings of the 7th USENIX Security Symposium (SECURITY'98)*, January 26-29, pp. 79-94, 1998.
- [13] D. Endler, "Intrusion detection: Applying machine learning to Solaris audit data," *Proceedings of the 1998 Annual Computer Security Applications Conference (ACSAC'98)*, pp. 268-279, Los Alamitos, CA, December 1998.
- [14] R. K. Cunningham, R. P. Lippmann, D. J. Fried, S. L. Garfinkle, I. Graf, K. R. Kendall, S. E. Webster, D. Wyschogrod, M. A. Zissman, "Evaluating Intrusion Detection Systems without Attacking your Friends: The 1998 DARPA Intrusion Detection Evaluation", SANS 1999.
- [15] *SunSHIELD Basic Security Module Guide*, Sun Microsystems, Inc., Palo Alto, 2000.
- [16] K. J. Das. "Attack development for Intrusion Detection Evaluation". Master Thesis, MIT, Cambridge, MA, June 2000.
- [17] J. Korba, "Windows NT Attacks for the Evaluation of Intrusion Detection Systems," BS/ME Thesis, MIT, 2000.
- [18] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Upper Saddle River: Prentice Hall, 1999, pp. 256-317.
- [19] H. Demuth, M. Beale, *Neural Network Toolbox User's Guide*, ver. 4, release 13, The MathWorks, Inc., Natick, MA, 2002, p. 7-7.