Some Evaluations of the Effectiveness of Anomaly Based Intrusion Detection Systems Based on the Junction Tree Algorithm

Evgeniya NIKOLOVA*
Faculty for Computer Science and Engineering, Burgas Free University
Burgas, Bulgaria

Veselina JECHEVA*
Faculty for Computer Science and Engineering, Burgas Free University
Burgas, Bulgaria

ABSTRACT

The aim of this paper is to present some evaluations of the effectiveness of IDS based on the junction tree algorithm (JTA). We stop our attention to two statistical methods - sensitivity and specificity which are functions of the true positive rate, the false negative rate and the false positive rate, the true negative rate respectively. Relationship between them is given by the receiver operating characteristic curve, which is one graphically method for estimation. For achieving a balance between the false positive rate and the false negative rate are used the crossover error rate.

Keywords: intrusion detection, anomaly based intrusion detection, junction tree algorithm**.**

1. INTRODUCTION

Intrusion detection is the activity of monitoring the events occurring in a computer network and detecting inappropriate, incorrect, or anomalous activity. An intrusion detection system (IDS) is a system that automates the intrusion detection process and monitors system data (network or host) to distinguish intrusions and attacks or normal user activity. [1]. Based on the intrusion detection method IDS can be divided into two main categories: misuse based and anomaly based. Misuse detection IDSs generate the alarms based on specific attack signatures. Despite of the fact they achieve a high level of accuracy, their major drawback is the little possibility of detecting novel attacks [11].

In this paper we direct our attention to the anomaly based IDS. An anomaly-based IDS examines ongoing traffic, activity, transactions, or behavior for anomalies on networks or systems that may indicate attack. The basic principle is the assumption that “attack behavior” differs enough from “normal user behavior” that it can be detected by cataloging and identifying the differences involved [2]. By creating baselines of normal behavior, anomaly-based IDS systems can observe when current behavior deviates statistically from the norm. This capability theoretically gives anomaly-based IDSs abilities to detect new attacks that are neither known nor for which signatures have been created.

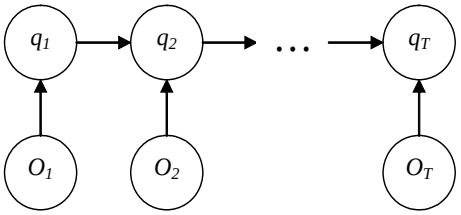
Analyzing program behavior profiles for intrusion detection has emerged as a viable alternative to user-based approaches to intrusion detection [5]. Program behavior profiles are built by capturing system calls made by the program under normal operational conditions.

In this paper we are interested in finding methodology for the attacks recognition during the normal activities in the system, i.e. to discern the normal activity patterns from abnormal ones. For that purpose we apply the junction tree algorithm (JTA). The algorithm takes the form of message passing on a graph referred to as a junction tree, whose nodes are clusters of variables. Each cluster starts with one potential of the factorized density. By combining this potential with the potentials it receives from its neighbors, it can compute the marginal over its variables. Except finding marginal probabilities, JTA helps to find the most likely state of the distribution and to define the anomalies in the obtained sequence. In Section 2 we give a brief exposition of the system model and the applied methodology. Finally, some of our experimental results and some evaluations of the method effectiveness are provided in Section 3.

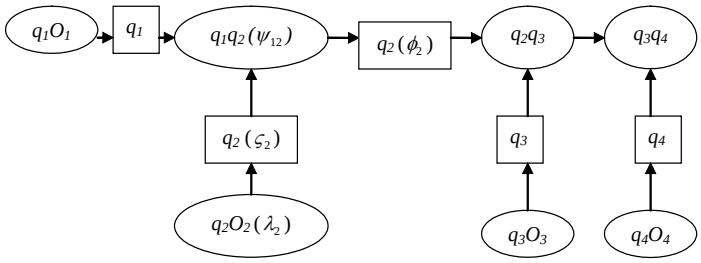
2. THE SYSTEM MODEL

The system model can be represented graphically as shown below:

* Supported by Burgas Free University under Project 5/2008.



where the vector $O=(O_1, O_2, \ldots, O_T)$ is the examined observation sequence and $Q=(q_1, q_2, \ldots, q_T)$ is the state sequence at the moments $t=1, 2, \ldots, T$. Each q_t is among the elements of the set $S = \{1, \ldots, N\}$. Consider the following junction tree:



We applied the junction tree algorithm (JTA) ([8], [9]) to this graphical model. Its purpose is to define a potential representation of the graph such that, as a result the marginals of individual or groups (cliques) could be obtained directly from the modified potentials. Except finding marginal probabilities, from the JTA we obtain the most likely state sequence during the examined period of time. Finally we compute the state transition probabilities for the result state sequence.

More details not introduced here could be found in [10].

3. SIMULATION EXPERIMENTS

The Experiment Data and Detection Results

The experiment data were obtained from the Immune Systems Project, performed by the researches in the Computer Science Department, University of New Mexico [12]. The data include normal user activity traces of some privileged processes that run with administrative rights (synthetic ftp, synthetic lpr, synthetic sendmail, login, named and xlock) and their child processes patterns, as well as intrusion data. Process behavior profiles are built by capturing system calls from Unix-based and Sun SPARCstations system examination during some period of time. The input data files are sequences of ordered pairs of numbers, where the first number is the process ID (PID) of the process executed, and the second one is the system call number. Forks are considered as separate processes and their execution results are examined as normal user activity.

The methods for pattern generation are described in [3] and [4]. They have proved that short sequences of system call traces produced by the execution of the privileged processes are a good discriminator between the normal and abnormal operating characteristics of programs. The experimental results have also confirmed that short sequences of system call traces are stable and consistent during program's normal activities.

Experiments based on the described methodology, were accomplished using corresponding software in C++. Based on the normal user activity patterns we evaluated the state transition probabilities during the system work in attack absence. The experimental data also contain abnormal user activity patterns, generated by some common intrusion tools (sunsendmailcp, lprcp, buffer overflow, some Trojans, etc.). We considered these data as current system activity data, which are examined by the JTA. We used a slide window with length T =10 to cross the traces of current user activity. Given an unknown observation sequence, we find the most likely state sequence, as well as its state transition probabilities. Comparing the obtained state transition probabilities to the state transition probabilities of the normal user activity patterns we detect the intrusions presence. Some of the results from this comparison for the enumerated processes are summarized in Figures 1, 2, 3, 4 and 5.

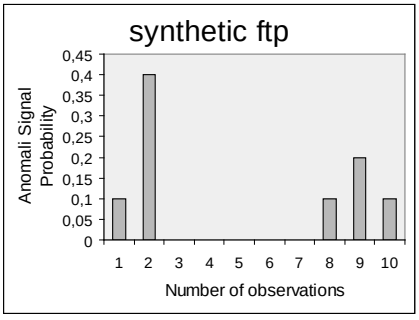


Figure 1. Anomaly Signal Probability for synthetic ftp

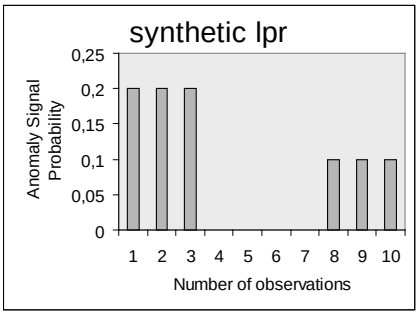


Figure 2. Anomaly Signal Probability for synthetic lpr

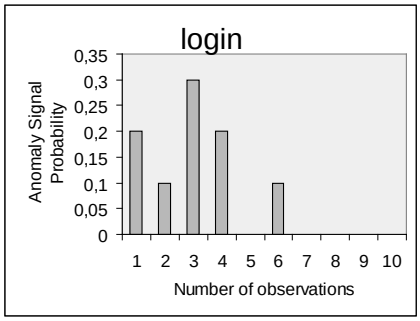


Figure 3. Anomaly Signal Probability for login

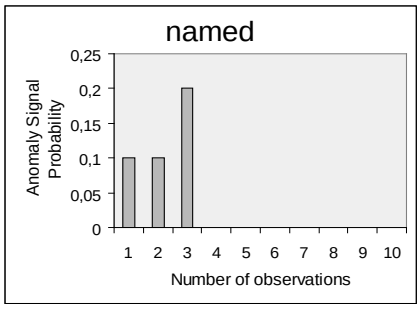


Figure 4. Anomaly Signal Probability for named

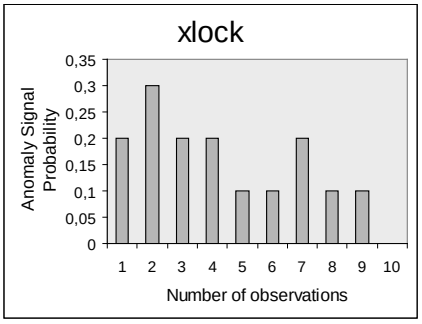


Figure 5. Anomaly Signal Probability for xlock

Some statistical methods for evaluating the effectiveness of IDS based on the JTA

Two statistical methods for evaluating the effectiveness of IDS are sensitivity and specificity. *Sensitivity* is defined as the true positive rate (TPR), i.e. intrusion correctly detected. Mathematically, sensitivity is expressed as

$$\frac{TPR}{TPR+FNR},$$

where the false negative rate (FNR) represents undetected attacks on a system. Therefore, the FNR is equal to one minus the sensitivity. The true negative rate (TNR) represents IDS that is correctly reporting that there are no intrusions. *Specificity* is expressed as

$$\frac{TNR}{TNR+FPR},$$

where the false positive rate (FPR) is the frequency with which the IDS reports malicious activity in error. Hence, the FPR is equal to one minus the specificity. Figures 6 and 7 contain graphs of the sensitivity and the specificity for the processes synthetic lpr and synthetic ftp respectively.

The true danger of a high FPR lies in fact that it may cause to ignore the system's output when legitimate alerts are raised. FNR changes in an inverse proportion to FPR as we see in Figures 8 and 9, which contain graphs of the FPR and FNR for the processes login and synthetic sendmail respectively.

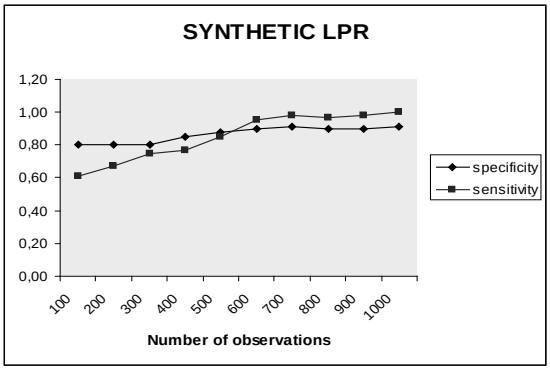


Figure 6. The specificity and the sensitivity for the process synthetic lpr

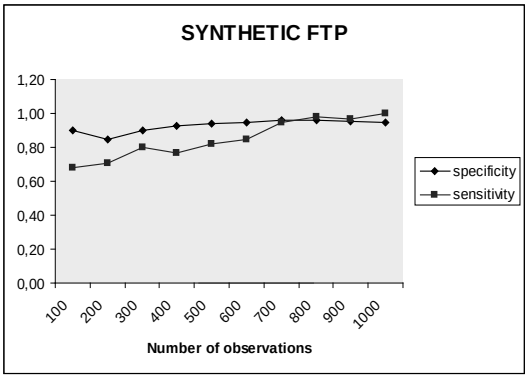


Figure 7. The specificity and the sensitivity for the process synthetic ftp

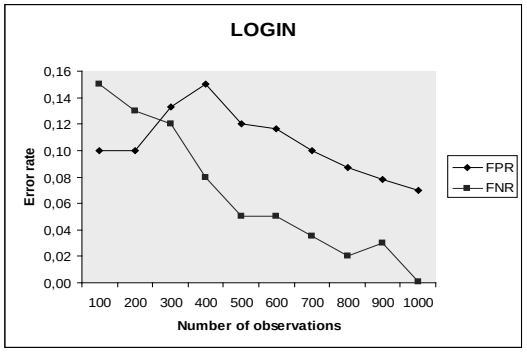


Figure 8. The false positive rate and the false negative rate for the process login

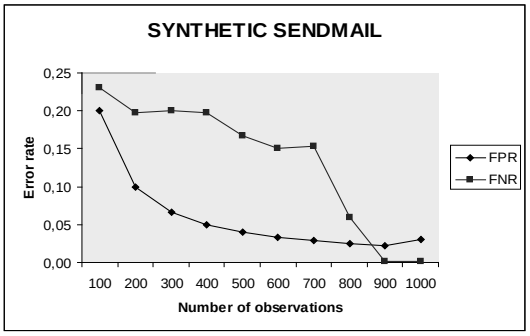


Figure 9. The false positive rate and the false negative rate for the process synthetic sendmail

The *receiver operating characteristic* (ROC) curve is a method of graphically demonstrating the relationship between sensitivity and specificity. An ROC space is defined by FPR and TPR as x and y respectively, which depicts relative trade-offs between true positive and false positive. Each sensitive value can be plotted against its corresponding specificity value to create the diagram for the processes login, synthetic sendmail, synthetic ftp and synthetic lpr at Figures 10, 11, 12 and 13 respectively. The decision threshold divides the normal activity traces into a true negative and a false positive group, and the abnormal activity traces into a true positive and a false negative group. As the decision threshold moves to the right along the x-axis, sensitivity ranges from one, when all tests are read as abnormal (no false negatives), to 0, when all are normal (no true positives). Maximal sensitivity is realized when all tests are reported as abnormal. Specificity moves in concert from 0 (no true negatives) to one (no false positives). Maximal specificity is achieved by reporting all tests as normal. The best possible prediction method would yield a point in upper left corner (0,1) of the ROC space, representing 100% sensitivity (all true positives are found) and 100% specificity (no false positives are found). This point is called a *perfect classification*. In our case the points in upper left corner of the ROC space for the processes login, synthetic sendmail, synthetic ftp and synthetic lpr are (0,07; 0,97), (0,16; 0,94), (0,05; 0,99) and (0,09; 0,96) respectively. The diagonal line (from the left bottom to the right corner) divides the ROC space in areas of good and bad classification. Points above this line indicate good classification results, while points below the line indicate wrong results. As we see at Figures 11 and 12 our method gives us satisfactory classification results.

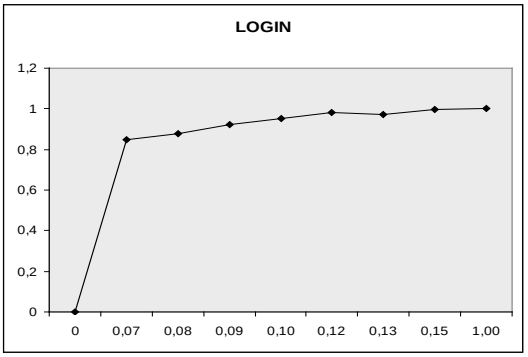


Figure 10. The ROC curve for the process login

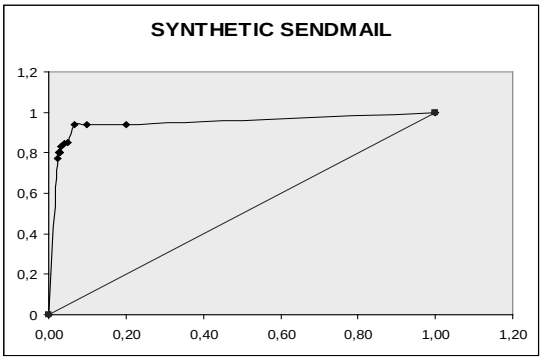


Figure 11. The ROC curve for the process synthetic sendmail and the diagonal line

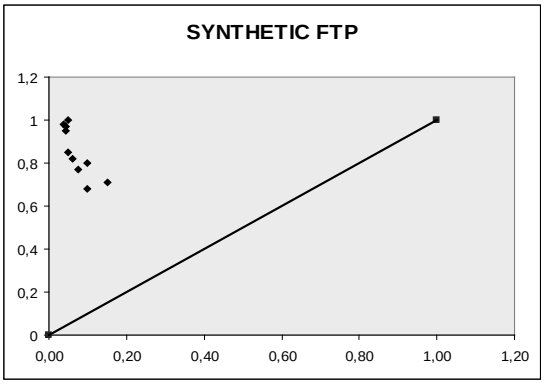


Figure 12. The ROC curve for the process synthetic ftp and the diagonal line

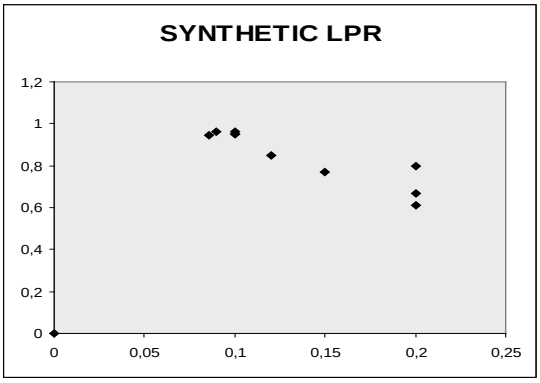


Figure 13. The ROC curve for the process synthetic lpr

The *crossover error rate* (CER) is defined as the value for which the FPR and the FNR are equal. In order to achieve a balance between FPR and FNR, we may select the IDS with the lowest CER. Figures 8 and 9 contain graphs of the FNR, fraction of intrusions incorrectly not detected, and the FPR, fraction of non-intrusions incorrectly detected, for the same input. The crossover error rates for the examined processes are represented in the Table 1. From the figures we see that the CER for the processes synthetic ftp and synthetic sendmail are relatively low, so we could conclude the method produces more satisfactory results for these processes.

Process	CER
synthetic ftp	0,04
synthetic lpr	0,11
synthetic sendmail	0,03
login	0,12

Table 1. The crossover error rate

The best results were 88,4% accurate, i.e. this anomaly detection method could accurately verify a given unknown sequence to be normal or anomalous with 88,4% accuracy. The remaining 11,6% are accounted for the FPR (8,6%) and the FNR (3%). Table 2 contains the values of the FPR, the FNR and the accuracy for the processes synthetic ftp, synthetic lpr, synthetic sendmail and login. From the table we see the accuracy reaches its lowest value for synthetic sendmail. So our future research will be directed towards investigating the reasons for this relatively low-grade result for this particular process.

Process	FPR	FNR	Accuracy
synthetic ftp	5%	1%	94%
synthetic lpr	9%	4%	87%
synthetic sendmail	16%	6%	78%
login	7%	3%	90%

Table 2. The false alarms rate and the algorithm accuracy

4. DISCUSSIONS

This paper investigated the applicability of the JTA in anomaly based intrusion detection and the emphasis has been on determining the success of the proposed methodology. The approach aims at building privileged process profiles and to detect anomalies by measuring deviations from the created profile. As a result of the JTA we found the joint distribution on the non-obvious variables in a clique with all the obvious variables clamped in their obvious states. Then calculating the likelihood is not difficult since we just sum out over the non-obvious variables of any converged potential. From the received sequence it is easy to determine if it is anomalous, comparing to the state transition probability.

The major advantage of graphical representation models, such as the one in JTA, over many other types of predictive models, such as neural networks and decision trees, is that unlike those "black box" approaches, the graph structure represents the inter-relationships among the data set attributes. Human experts can easily understand the network structures and if necessary can easily modify them to obtain better predictive models. Other advantages of graphical representation models include explicit uncertainty characterization, fast and efficient computation, and quick training. They are highly adaptive and easy to build, and provide explicit representation of domain specific knowledge in human reasoning frameworks. Moreover, graphical representations offer good generalization with limited training data and easy maintenance when adding new features or new training data. Another advantage of the anomaly-based approach in general is its potential to detect novel attacks, insider attacks or account theft.

A disadvantage of the proposed method is its considerable computational price, since it has $O(TM^2)$ complexity. Since M^2 is the cardinality of the clique state space, its value could be significant in the case of a large system. Another disadvantage of the anomaly based IDS in general is the creation of the database containing the user profiles, which could be a task of considerable difficulty and requires some period of time the system is unprotected.

5. CONCLUSIONS AND FUTURE WORK

The present paper proposes a method of intrusion access recognition in the anomaly based IDS. We applied a graphical representation model, precisely the Junction tree algorithm, as well as some probabilistic methods to distinguish the normal user activity from the intrusion one. Some experimental simulations were accomplished as well. The approach could be extended to other processes and for different types of intrusions. The purpose of future work could be the algorithm optimization, as well as some different probabilistic method applications and the comparison of the obtained results.

REFERENCES

[1] Abraham, A., J. Thomas, Distributed Intrusion Detection Systems: A Computational Intelligence Approach, **Applications of Information Systems to Homeland Security and Defense**, Idea Group Inc. Publishers, USA, Chapter 5, 2005, pp. 105-135.

[2] Chan P., M. Mahoney, M. Arshad, Learning Rules and Clusters for Network Anomaly Detection, Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, George Mason University, **Technical Report CS-2003-06**, 2003.

[3] Forrest S., S.A. Hofmeyr, A. Somayaji, T.A. Longtaff, A Sense of Self for Unix Processes, In

Proceedings of the 1996 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Los Alamitors, CA, pp.120-128.

[4] Forrest S., S.A. Hofmeyr, A. Somayaji, Intrusion detection using sequences of system calls, **Journal of Computer Security**, Vol. 6, 1998, pp. 151-180.

[5] Ghosh A.K., A. Schwartzbard, M. Schatz, Learning Program Behavior Profiles for Intrusion Detection, In **Proceedings of the 1st Workshop on Intrusion Detection and Network Monitoring**, pp. 51–62, 1999.

[6] Hoang X.D., J. Hu, P. Bertok, A Multi-layer Model for Anomaly Intrusion Detection Using Program Sequences of System Calls, In **Proceedings of the 11th IEEE International Conference on Networks (ICON 2003)**, Sydney, Australia, 2003.

[7] Jensen F.V., F. Jensen, Optimal Junction Trees, In **Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence**, San Francisco, CA, USA, pp. 360-366, 1994.

[8] Lauritzen S.L., **Graphical Models**, Oxford Science Publications, 1996.

[9] Lauritzen, S. L., D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems (with discussion)", **Journal of Royal Statistical Society**, Series B, 50(2), 1988, pp. 157–224.

[10] Nikolova E., V. Jecheva, Anomaly Based Intrusion Detection Systems Based on the Junction Tree Algorithm, In **Proceedings of the International Multiconference on Computer Science and Information Technology**, ISSN 18967094, Wisla, Poland, 2007, pp. 465 – 471.

[11] Tran T.P., T. Jan, A. J. Simmonds, A Multi-Expert Classification Framework for Network Misuse Detection, In **Proceeding of Artificial Intelligence and Soft Computing**, ISBN 0-88986-610-4, 2006.

[12] University of New Mexico, Computer Immune Systems Project, http://www.cs.unm.edu/~immsec/systemcalls.htm.