International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:4, 2010

Scaling up Detection Rates and Reducing False Positives in Intrusion Detection using NBTree

Dewan Md. Farid, Nguyen Huu Hoa, Jerome Darmont, Nouria Harbi, Mohammad Zahidur Rahman

***Abstract*—**In this paper, we present a new learning algorithm for anomaly based network intrusion detection using improved self adaptive naïve Bayesian tree (NBTree), which induces a hybrid of decision tree and naïve Bayesian classifier. The proposed approach scales up the balance detections for different attack types and keeps the false positives at acceptable level in intrusion detection. In complex and dynamic large intrusion detection dataset, the detection accuracy of naïve Bayesian classifier does not scale up as well as decision tree. It has been successfully tested in other problem domains that naïve Bayesian tree improves the classification rates in large dataset. In naïve Bayesian tree nodes contain and split as regular decision-trees, but the leaves contain naïve Bayesian classifiers. The experimental results on KDD99 benchmark network intrusion detection dataset demonstrate that this new approach scales up the detection rates for different attack types and reduces false positives in network intrusion detection.

***Keywords*—**Detection rates, false positives, network intrusion detection, naïve Bayesian tree.

I. Introduction

AN intrusion detection system (IDS) is a security tools used to detect unauthorized activities of a computer system or network. In other words, intrusion detection is the process of identifying actions that attempt to compromise the confidentiality, integrity or availability of a computer system or network. It was first introduced by James P. Anderson in 1980 [1]. In his report, Anderson presents a threat model that classifies intrusions to develop a security monitoring surveillance system based on detecting anomalies in user behavior. Later in 1986, Dr. Dorothy Denning proposed several models for IDS based on statistics, Markov chains, time-series, etc [2]. IDS were first implemented for host-based that located in servers to examine the internal interfaces [3], but with the evolution of computer networks the focus gradually shifted toward network-based. Network-based intrusion detection system (NIDS) performs packet logging, real-time traffic analysis of IP network, and tries to discover if an intruder is attempting to break into the network. Normally, IDS are classified into three systems such as misuse-based system, anomaly-based system, and hybrid system. Misuse-based IDS performs simple pattern matching techniques to match an attack pattern corresponding to known attack patterns in the database and produces very low false positives (FP). It requires regular updates of rules or signatures and not capable to detects unknown attacks. Anomaly-based IDS identifies new attacks by analyzing the anomalous behaviors from normal behaviors [4], and achieves high detection rates (DR) for both known as well as unknown attacks, but produces many false positives (FP). Anomaly-based IDS generate rules by observing collected audit data. Audit data is the records of activities generated by the operating system that are logged to a file in chronologically sorted order. On the other hand, a hybrid IDS combines the techniques of both misuse-based and anomaly-based detection systems. Currently adaptive intrusion detection aims to solve the problems of analyzing the huge volumes of audit data and realizing performance optimization of detection rules.

Anomaly network intrusion detection based on data mining techniques such as decision tree (DT), naïve Bayesian classifier (NB), neural network (NN), support vector machine (SVM), k-nearest neighbors (KNN), fuzzy logic model, and genetic algorithm have been widely used by researchers to improve the process of intrusion detection [5]-[11]. However, there exist various problems that induce the complexity of detection systems. Some of these problems are low detection accuracy, unbalanced detection rates for different attack types, and high false positives. In this paper, a new learning algorithm for anomaly based network intrusion detection using improved self adaptive naïve Bayesian tree is presented, which scales up the balance detections for different attack types and keeps the false positives at acceptable level. The experimental results by using on KDD99 benchmark network intrusion detection dataset prove that the proposed algorithm has achieved both high detection rates (DR) for different attacks, and the significant reduction of false positives (FP) in comparison with existing methods.

The remainders of the paper are organized as follows. Section II presents some anomaly based intrusion detection methods. The proposed algorithm is introduced in Section III. Then, the experimental results are expressed in Section IV. Finally, our conclusions and future works are mentioned in Section V.

Dewan Md. Farid is with the ERIC Laboratory, University Lumière Lyon 2 – 5 av. Pierre Mendes, France – 69676 BRON Cedex, France (phone: +33 0648882531; fax: +33 478772375; e-mail: dewanfarid@gmail.com).

Nguyen Huu Hoa is with the Labratoire ERIC, University Lumière Lyon 2 –France (e-mail: nhhoa@eric.univ-lyon2.fr).

Jerome Darmont is with the ERIC Laboratory, University Lumière Lyon 2– France (e-mail: jerome.darmont@univ-lyon2.fr).

Nouria Harbi is with the ERIC Laboratory, University Lumière Lyon 2– France (e-mail: nouria.harbi@univ-lyon2.fr).

Mohammad Zahidur Rahman is with the Department of Computer Science and Engineering, Jahangirnagar University, Bangladesh (e-mail: rmzahid@juniv.edu).

II. Anomaly Based Intrusion Detection Techniques

A. Statistical Anomaly Detection

In statistical based intrusion detection, the IDS capture the network traffics or system activities, and generate profiles to represent their behavior. These profiles is based on metrics such as the traffic rate, the number of packets for each protocol, the rate of connections, the number of different IP addresses, etc. Typically, two types of profiles are considered during the intrusion detection process: the currently observed profile over time, and the previously trained or stored profile. As the network or system event occur, the IDS updates the profile and periodically calculates an anomaly score by comparing the current profile with the stored profile using a function of abnormality of all measures within the profile. If the anomaly score is higher than a certain threshold the IDS generates an alert. Statistical based IDS do not require prior knowledge of attacks and capable to detect the very latest attacks, but skilled attackers can train a statistical anomaly based IDS to accept abnormal behavior as normal. Also it is difficult to determine the threshold point that balances the likelihood of false positives (no attack but alarm raised by IDS) with the likelihood of false negatives (attack occur but no alarm rose by IDS).

In the early 1980's, an Intrusion Detection Expert System (IDES) was developed by Stanford Research Institute (SRI) that continuously monitored user behavior and detected suspicious events [12]. Later SRI developed an improved version of IDES called the Next-Generation Intrusion Detection Expert System (NIDES) [13], [14] that could operate in real time for continuous monitoring of user activity or could run in a batch mode for periodic analysis of the audit data. NIDES enable the system to compare the current activities of the user/system/network with the audited intrusion detection variables stored in the profile and then raise an alarm if the current activity is sufficiently far from the stored audited activity. In 1988, a statistical anomaly-based IDS was proposed by Haystack [15], which used both user and group-based anomaly detection strategies. In this system, a range of values were considered normal for each attribute and during a session if an attribute fell outside the normal range then an alarm raised. It was designed to detect six types of intrusions: attempted break-ins by unauthorized users, masquerade attacks, penetration of the security control system, leakage, denial of service, and malicious use. Statistical Packet Anomaly Detection Engine (SPADE) [16] is a statistical anomaly detection system that is available as a plug-in for SNORT. SNORT is an open source network intrusion detection and prevention system (NIDPS) developed by Sourcefire [17], [18]. SNORT performs protocol analysis, content searching/matching, and commonly blocks a variety of intrusions such as buffer overflows, stealth port scans, web application attacks, SMB probes, and OS fingerprinting attempts.

B. Data Mining Based Anomaly Detection

Recently, to build an effective and efficient real time IDS researchers are increasingly looking at using of data mining algorithms for adaptive intrusion detection [19], [20]. Some of the data mining algorithms are cited below.

1. Naïve Bayesian Classifier: NB classifier provides a probabilistic approach for performing supervised learning. It provides an optimal way to predict the class of an unknown example and widely used in many field of data mining. In NB classifier class conditional probabilities for each attribute value are calculated from the given dataset and then these probabilities are used to classify the known or unknown examples. Several researchers have adapted ideas from NB classifier to create models for anomaly detection [21], [22]. Valdes et al. [23] developed an anomaly detection system that employed NB classifier to perform intrusion detection.

2. Decision Tree: DT is powerful and popular tools for classification and prediction [7]. It can be constructed from dataset with many attributes. A decision tree has three main components: nodes, leaves, and edges. Each node is labeled with an attribute by which the data is to be partitioned. Each node has a number of edges, which are labeled according to possible values of the attribute. An edge connects either two nodes or a node and a leaf. Leaves are labeled with a decision value for categorization of the data. To make a decision using a decision Tree, start at the root node and follow the tree down the branches until a leaf node representing the class is reached. Each decision tree represents a rule set, which categorizes data according to the attributes of dataset.

3. Neural Network: NN based IDS focus on detecting deviations in program behavior as a sign of an intrusion. NN learns to predict the behavior of the various users in the computer system. Ghosh et al. used the feed-forward back propagation algorithm for classifying system-call sequence to detect anomalies and misuses [24]. In another paper, Ramadas et al. [25] present the Anomalous Network-Traffic Detection with Self Organizing Maps (ANDSOM) an anomaly intrusion detection model for the network based IDS that creates a two dimensional Self Organizing Map for each network service. In this paper, neurons are trained with normal network traffic during the training phase to capture characteristic patterns. When real time data is fed to the trained neurons, then an anomaly is detected if the distance of the incoming traffic is more than a preset threshold.

4. K Nearest Neighbors: KNN is a classification algorithm based on the use of distance measures. It finds k examples in dataset that are closest to the classifying example and assigns the most frequent label among these examples to the new example. When a classification is to be made for a new example, its distance to each attribute in the dataset must be determined. Only the k closest examples in the dataset are considered further. The new example is then placed to the class that contains the most examples from this set of K closest examples.

III. Proposed NBTree for Intrusion Detection

Naïve Bayesian tree (NBTree) algorithm is similar to the classical recursive partitioning schemes, except that the leaf nodes created are naïve Bayesian classifier instead of node predicting a single class [26]. NBTree is a hybrid approach that attempts to utilize the advantage of both decision trees and naïve Bayesian classifier. It splits the dataset by applying entropy based algorithm and used standard naïve Bayesian classifiers at the leaf node to handle attributes. NBTree applies strategy to construct decision tree and replaces leaf node with NB classifier.

A. Improved Self Adaptive Naïve Bayesian Tree

In a given training data, $D = \{A_1, A_2, \ldots, A_n\}$ of attributes, where each attribute $A_i = \{A_{i1}, A_{i2}, \ldots, A_{ik}\}$ contains attribute values and a set of classes $C = \{C_1, C_2, \ldots, C_n\}$, where each class $C_j = \{C_{j1}, C_{j2}, \ldots, C_{jk}\}$ has some values. Each example in the training data contains weight, $W = \{W_1, W_2 \ldots, W_n\}$. Initially, all the weights for examples of training data have equal unit value that set to $W_i = 1/n$. Where n is the total number of the training examples. Estimates the prior probability $P(C_j)$ for each class by summing the weights and how often each class occurs in the training data. For each attribute, A_i, the number of occurrences of each attribute value A_{ij} can be counted by summing the weights to determine $P(A_{ij})$. Similarly, the conditional probability $P(A_{ij} \mid C_j)$ can be estimated by summing the weights how often each attribute value occurs in the class C_j in the training data. The conditional probabilities $P(A_{ij} \mid C_j)$ are estimated for all values of attributes. The algorithm then uses the prior and conditional probabilities to update the weights. This is done by multiplying the probabilities of the different attribute values from the examples. Suppose the training example e_i has independent attribute values $\{A_{i1}, A_{i2}, \ldots, A_{ip}\}$. We already know the prior probabilities $P(C_j)$ and conditional probabilities $P(A_{ik} \mid C_j)$, for each class C_j and attribute A_{ik}. We then estimate $P(e_i \mid C_j)$ by

$$P(e_i \mid C_j) = P(C_j) \prod_{k=1 \to p} P(A_{ij} \mid C_j) \quad (1)$$

To update the weight of training example e_i, we can estimate the likelihood of e_i for each class. The probability that e_i is in a class is the product of the conditional probabilities for each attribute value. The posterior probability $P(C_j \mid e_i)$ is then found for each class. Then the weight of the example is updated with the highest posterior probability for that example and also the class value is updated according to the highest posterior probability. Now, for each attribute A_i, evaluate the utility, $u(A_i)$, of a spilt on attribute A_i. Let $j = argmax_i(u_i)$, i.e., the attribute with the highest utility. If u_j is not significantly better than the utility of the current node, create a NB classifier for the current node. Partition the training data D according to the test on attribute A_i. If A_i is continuous, a threshold split is used; if A_i is discrete, a multi-way split is made for all possible values. For each child, call the algorithm recursively on the portion of D that matches the test leading to the child. The main procedure of proposed improved self adaptive naïve Bayesian algorithm is described as follows.

Algorithm Improved Self-adaptive NBTree (ISANBT)
Input: a training dataset D of labeled examples.
Output: a hybrid decision tree with naïve Bayesian classifier at the leaves.
Procedure:

1. Initialize all the weights in D, $W_i = 1/n$.
2. Calculate the prior probabilities $P(C_j)$ for each class C_j in D. $P(C_j) = \dfrac{\sum_{Ci} W_i}{\sum_{i=1}^{n} W_i}$
3. Calculate the conditional probabilities $P(A_{ij} \mid C_j)$ for each attribute values in D. $P(A_{ij} \mid C_j) = \dfrac{P(A_{ij})}{\sum_{C_i} W_i}$
4. Calculate the posterior probability for each example in D. $P(e_i \mid C_j) = P(C_j) \prod P(A_{ij} \mid C_j)$.
5. Update all the weights in D with Maximum Likelihood (ML) of posterior probability $P(C_j \mid e_i)$; $W_i = P_{ML}(C_j \mid e_i)$ and change the class value of examples associated with maximum posterior probability, $C_j = C_i \rightarrow P_{ML}(C_j \mid e_i)$.
6. For each attribute A_i, evaluate the utility, $u(A_i)$, of a spilt on attribute A_i.
7. Let $j = argmax_i(u_i)$, i.e., the attribute with the highest utility.
8. If u_j is not significantly better than the utility of the current node, create a naïve Bayesian classifier for the current node and return.
9. Partition the training data D according to the test on attribute A_i. If A_i is continuous, a threshold split is used; if A_i is discrete, a multi-way split is made for all possible values.
10. For each child, call the algorithm recursively on the portion of D that matches the test leading to the child.

IV. Experimental Analysis

A. Intrusion Detection Data Stream

The KDD cup 1999 dataset was used in the 3rd International Knowledge Discovery and Data Mining Tools Competition for building a network intrusion detector, a predictive model capable of distinguishing between intrusions and normal connections [27]. In 1998, DARPA intrusion detection evaluation program, a simulated environment was set up to acquire raw TCP/IP dump data for a local-area network (LAN) by the MIT Lincoln Lab to compare the performance of various intrusion detection methods. It was operated like a real environment, but being blasted with multiple intrusion attacks and received much attention in the research community of adaptive intrusion detection. The KDD99 dataset contest uses a version of DARPA98 dataset. In KDD99 dataset, each example represents attribute values of a class in the network data flow, and each class is labeled either normal or attack. The classes in KDD99 dataset categorized

into five main classes (one normal class and four main intrusion classes: probe, DOS, U2R, and R2L).

1) Normal connections are generated by simulated daily user behavior such as downloading files, visiting web pages.

2) Denial of Service (DoS) attack causes the computing power or memory of a victim machine too busy or too full to handle legitimate requests. DoS attacks are classified based on the services that an attacker renders unavailable to legitimate users like apache2, land, mail bomb, back, etc.

3) Remote to User (R2L) is an attack that a remote user gains access of a local user/account by sending packets to a machine over a network communication, which include send-mail, and Xlock.

4) User to Root (U2R) is an attack that an intruder begins with the access of a normal user account and then becomes a root-user by exploiting various vulnerabilities of the system. Most common exploits of U2R attacks are regular buffer-overflows, load-module, Fd-format, and Ffb-config.

5) Probing (Probe) is an attack that scans a network to gather information or find known vulnerabilities. An intruder with a map of machines and services that are available on a network can use the information to look for exploits.

In KDD99 dataset these four attack classes (DoS, U2R, R2L, and probe) are divided into 22 different attack classes that tabulated in Table I.

TABLE I.
DIFFERENT TYPES OF ATTACKS IN KDD99 DATASET

4 Main Attack Classes	22 Attack Classes
Denial of Service (DoS)	back, land, neptune, pod, smurt, teardrop
Remote to User (R2L)	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster
User to Root (U2R)	buffer_overflow, perl, loadmodule, rootkit
Probing	ipsweep, nmap, portsweep, satan

There are 41 input attributes in KDD99 dataset for each network connection that have either discrete or continuous values and divided into three groups. The first group of attributes is the basic features of network connection, which include the duration, prototype, service, number of bytes from source IP addresses or from destination IP addresses, and some flags in TCP connections. The second group of attributes in KDD99 is composed of the content features of network connections and the third group is composed of the statistical features that are computed either by a time window or a window of certain kind of connections. The list of the input attributes in KDD99 dataset for each network connections is shown in the Table II.

TABLE II.
INPUT ATTRIBUTES IN KDD99 DATASET

No	Input Attribute	Type	No	Input Attribute	Type
1	Duration	Con.	22	is_guest_login	Dis.
2	protocol_type	Dis.	23	Count	Con.
3	Service	Dis.	24	srv_count	Con.
4	Flag	Dis.	25	serror_rate	Con.
5	src_bytes	Con.	26	srv_serror_rate	Con.
6	dst_bytes	Con.	27	rerror_rate	Con.
7	Land	Dis.	28	srv_rerror_rate	Con.
8	wrong_fragment	Con.	29	same_srv_rate	Con.
9	Urgent	Con.	30	diff_srv_rate	Con.
10	Hot	Con.	31	srv_diff_host_rate	Con.
11	num_failed_logins	Con.	32	dst_host_count	Con.
12	logged_in	Dis.	33	dst_host_srv_count	Con.
13	num_compromised	Con.	34	dst_host_same_srv_rate	Con.
14	root_shell	Con.	35	dst_host_diff_srv_rate	Con.
15	su_attempted	Con.	36	dst_host_same_src_port_rate	Con.
16	num_root	Con.	37	dst_host_srv_diff_host_rate	Con.
17	num_file_creations	Con.	38	dst_host_serror_rate	Con.
18	num_shells	Con.	39	dst_host_srv_serror_rate	Con.
19	num_access_files	Con.	40	dst_host_rerror_rate	Con.
20	num_outbound_cmds	Con.	41	dst_host_srv_rerror_rate	Con.
21	is_host_login	Dis.	-	-	-

Table III shows the number of examples of 10% training data and 10% testing data in KDD99 dataset. There are some new attack examples in testing data, which is no present in the training data.

TABLE III.
NUMBER OF EXAMPLES IN TRAINING AND TESTING KDD99 DATA

Attack Types	Training Examples	Testing Examples
Normal	97277	60592
Denial of Service	391458	237594
Remote to User	1126	8606
User to Root	52	70
Probing	4107	4166
Total Examples	494020	311028

B. Experimental Analysis

In order to evaluate the performance of proposed algorithm for network intrusion detection, we performed 5-class classification using KDD99 network intrusion detection benchmark dataset. All experiments were performed using an Intel Core 2 Duo Processor 2.0 GHz processor (2 MB Cache, 800 MHz FSB) with 1 GB of RAM. The results of the comparison of proposed improved self adaptive naïve Bayesian tree algorithm (ISANBT) with C4.5 and with naive Bayesian classifier (NB) are tabulated in Table IV and Table V.

TABLE IV.
COMPARISON OF THE RESULTS USING 41 ATTRIBUTES

Method	Normal	Probe	DOS	U2R	R2L
ISANBT (DR %)	99.76	99.21	99.65	99.11	99.16
ISANBT (FP %)	0.07	0.44	0.05	0.12	6.85
NB (DR %)	99.27	99.11	99.69	64.00	99.11
NB (FP %)	0.08	0.45	0.05	0.14	8.02
C4.5 (DR %)	98.73	97.85	97.51	49.21	91.65
C4.5 (FP %)	0.10	0.55	0.07	0.14	11.03

TABLE V.
COMPARISON OF THE RESULTS USING 19 ATTRIBUTES

Method	Normal	Probe	DOS	U2R	R2L
ISANBT (DR %)	99.79	99.65	99.76	99.43	99.25
ISANBT (FP %)	0.06	0.48	0.04	0.10	6.32
NB (DR %)	99.65	99.35	99.71	64.84	99.15
NB (FP %)	0.06	0.49	0.04	0.12	6.87
C4.5 (DR %)	98.81	98.22	97.63	56.11	91.79
C4.5 (FP %)	0.08	0.51	0.05	0.12	8.34

We tested the performance of ISANBT algorithm using the reduced dataset of 12 and 17 attributes in KDD99, which increase the detection rate that are summarized in Table VI.

TABLE VI.
PERFORMANCE OF PROPOSED ALGORITHM USING REDUCED DATASET

Class Value	12 Attributes	17 Attributes
Normal	99.89	99.82
Probe	99.42	99.39
DoS	99.79	99.78

U2R	99,38	99.44
R2L	99.23	99.29

V. Conclusion

This paper introduced a new learning algorithm for anomaly based network intrusion detection using improved self adaptive naive Bayesian tree, which analyzes the large volume of network data and considers the complex properties of attack behaviors to scaling up detection rates and reducing false positives in intrusion detection. In this paper, we have concentrated on the development of the performance of decision tree and naïve Bayesian classifier for network intrusion detection. It has been successfully tested that proposed ISANBT algorithm maximized the balance detection rates on the 5 classes of KDD99 benchmark network intrusion detection dataset, as well as minimized false positives at acceptable level. The attacks of KDD99 dataset detected with 99% accuracy using proposed algorithm. The future work focus on apply this detection model into real computer network.

Acknowledgment

Support for this research received from ERIC Laboratory, University Lumière Lyon 2 – France, and Department of Computer Science and Engineering, Jahangirnagar University, Bangladesh.

References

[1] James P. Anderson, "Computer security threat monitoring and surveillance," Technical Report 98-17, James P. Anderson Co., Fort Washington, Pennsylvania, USA, April 1980.
[2] Dorothy E. Denning, "An intrusion detection model," IEEE Transaction on Software Engineering, SE-13(2), 1987, pp. 222-232.
[3] D. Y. Yeung, and Y. X. Ding, "Host-based intrusion detection using dynamic and static behavioral models," Pattern Recognition, 36, 2003, pp. 229-243.
[4] Lazarevic, A., Ertoz, L., Kumar, V., Ozgur,. A., Srivastava, and J., "A comparative study of anomaly detection schemes in network intrusion detection," In Proc. of the SIAM Conference on Data Mining, 2003.
[5] Barbara, Daniel, Couto, Julia, Jajodia, Sushil, Popyack, Leonard, Wu, and Ningning, "ADAM: Detecting intrusion by data mining," IEEE Workshop on Information Assurance and Security, West Point, New York, June 5-6, 2001.
[6] Lee W., Stolfo S., and Mok K., "Adaptive Intrusion Detection: A data mining approach," Artificial Intelligence Review, 14(6), December 2000, pp. 533-567.
[7] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naïve Bayes vs. decision trees in intrusion detection systems," In Proc. of 2004 ACM Symposium on Applied Computing, 2004, pp. 420-424.
[8] Mukkamala S., Janoski G., and Sung A.H., "Intrusion detection using neural networks and support vector machines," In Proc. of the IEEE International Joint Conference on Neural Networks, 2002, pp.1702-1707.
[9] J. Luo, and S.M. Bridges, "Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection," International Journal of Intelligent Systems, John Wiley & Sons, vol. 15, no. 8, 2000, pp. 687-703.
[10] YU Yan, and Huang Hao, "An ensemble approach to intrusion detection based on improved multi-objective genetic algorithm," Journal of Software, vol. 18, no. 6, June 2007, pp. 1369-1378.
[11] Shon T., Seo J., and Moon J., "SVM approach with a genetic algorithm for network intrusion detection," In Proc. of 20th International Symposium on Computer and Information Sciences (ISCIS 2005), Berlin: Springer-Verlag, 2005, pp. 224-233.
[12] Dorothy E. Denning, and P.G. Neumann "Requirement and model for IDES- A real-time intrusion detection system," Computer Science Laboratory, SRI International, Menlo Park, CA 94025-3493, Technical Report # 83F83-01-00, 1985.
[13] D. Anderson, T. Frivold, A. Tamaru, and A. Valdes, "Next generation intrusion detection expert system (NIDES)," Software Users Manual, Beta-Update Release, Computer Science Laboratory, SRI International, Menlo Park, CA, USA, Technical Report SRI-CSL-95-0, May 1994.
[14] D. Anderson, T.F. Lunt, H. Javitz, A. Tamaru, and A. Valdes, "Detecting unusual program behavior using the statistical component of the next generation intrusion detection expert system (NIDES)," Computer Science Laboratory, SRI International, Menlo Park, CA, USA, Technical Report SRI-CSL-95-06, May 1995.
[15] S.E. Smaha, and Haystack, "An intrusion detection system," in Proc. of the IEEE Fourth Aerospace Computer Security Applications Conference, Orlando, FL, 1988, pp. 37-44.
[16] N. Ye, S.M. Emran, Q. Chen, and S. Vilbert, "Multivariate statistical analysis of audit trails for host-based intrusion detection," IEEE Transactions on Computers 51, 2002, pp. 810-820.
[17] Martin Roesch, "SNORT: The open source network intrusion system," Official web page of Snort at http://www.snort.org/
[18] L. C. Wuu, C. H. Hung, and S. F. Chen, "Building intrusion pattern miner for sonrt network intrusion detection system," Journal of Systems and Software, vol. 80, Issue 10, 2007, pp. 1699-1715.
[19] W. Lee, R.A. Nimbalkar, K.K. Yee, S.B. Patil, P.H. Desai, T.T. Tran, and S.J. Stolfo, "A data mining and CIDF based approach for detecting novel and distributed intrusions," In Proc. of the 3rd International Workshop on Recent Advances in Intrusion Detection (RAID 2000), Toulouse , France, 2000, pp. 49-65.
[20] W. Lee, and S.J. Stolfo, "Data mining approach for intrusions detection," In Proc. of the 7th USENIX Security Symposium (SECURITY-98), Berkeley, CA, USA, 1998, pp. 79-94.
[21] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Bayesian event classification for intrusion detection," In Proc. of the 19th Annual Computer Security Applications Conference, Las Veges, NV, 2003.
[22] N. Ye, M. Xu, and S.M. Emran, "Probabilistic networks with undirected links for anomaly detection," In Proc. of the IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point, NY, 2000.
[23] A. Valdes, and K. Skinner, "Adaptive model-based monitoring for cyber attack detection," In Recent Advances in Intrusion Detection Toulouse, France, 2000, pp. 80-92.
[24] A.K. Ghosh, and A. Schwartzbart, "A study in using neural networks for anomaly and misuse detection," In Proc. of the Eighth USENIX Security Symposium, Washington, DC, 1999, pp. 141-151.
[25] M. Ramadas, and S.O.B. Tjaden, "Detecting anomalous network traffic with self-organizing maps," In Proc. of the 6th International Symposium on Recent Advances in Intrusion Detection, Pittsburgh, PA, USA, 2003, pp. 36-54.
[26] R. Kohavi, "Scaling up the accuracy of naïve Bayes classifiers: A Decision Tree Hybrid," In Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA:AAAI Press/MIT Press, 1996, pp. 147-149.
[27] The KDD Archive. KDD99 cup dataset, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html