A Framework for an Adaptive Intrusion Detection System using Bayesian Network

Conference Paper · June 2007

DOI: 10.1109/ISI.2007.379535 · Source: IEEE Xplore

3 authors:

Farah Jemili
University of Sousse
48 PUBLICATIONS **360** CITATIONS
SEE PROFILE

Montassar Zaghdoud
Ecole Nationale des Sciences de l'Informatique
4 PUBLICATIONS **133** CITATIONS
SEE PROFILE

Mohamed Ben Ahmed
Université de la Manouba
197 PUBLICATIONS **1,082** CITATIONS
SEE PROFILE

A FRAMEWORK FOR AN ADAPTIVE INTRUSION DETECTION SYSTEM USING BAYESIAN NETWORK

Farah Jemili
Laboratoire RIADI, ENSI, Manouba University
Manouba 2010, Tunisia
Jmili_farah@yahoo.fr

Montaceur Zaghdoud
Laboratoire RIADI, ENSI, Manouba University
Manouba 2010, Tunisia
Montaceur.zaghdoud@ensi.rnu.tn

Mohamed Ben Ahmed
Laboratoire RIADI, ENSI, Manouba University
Manouba 2010, Tunisia
Mohamed.benahmed@riadi.rnu.tn

KEYWORDS

Adaptive intrusion detection, bayesian network, learning algorithm, learning dataset, inference.

ABSTRACT

The goal of a network-based intrusion detection system (IDS) is to identify malicious behavior that targets a network and its resources. Intrusion detection parameters are numerous and in many cases they present uncertain and imprecise causal relationships which can affect attack types. A Bayesian Network (BN) is known as graphical modeling tool used to model decision problems containing uncertainty. In this paper, a BN is used to build automatic intrusion detection system based on signature recognition. The goal is to recognize signatures of known attacks, match the observed behavior with those known signatures, and signal intrusion when there is a match. A major difficulty of this system is that intrusions signatures change over the time and the system must be retrained. An IDS must be able to adapt to these changes. The goal of this paper is to provide a framework for an adaptive intrusion detection system that uses Bayesian network.

1. INTRODUCTION

Intrusion detection can be defined as the process of identifying malicious behavior that targets a network and its resources [1].

Malicious behavior is defined as a system or individual action which tries to use or access to computer system without authorization (i.e., *crackers*) and the privilege excess of those who have legitimate access to the system (i.e., the *insider threat*).

The proliferation of heterogeneous computer networks has serious implications for the intrusion detection problem. Foremost among these implications is the increased opportunity for unauthorized access that is provided by the network's connectivity.

Intrusion detection is not an easy task due to the vastness of the network activity data and the need to regularly update the IDS to be adapted to unknown attack methods.

Nowadays, completely protect a network from attacks is being a very hard task. Even heavily protected networks are sometimes penetrated, and an Adaptive Intrusion Detection System seems to be essential and is a key component in computer and network security.

2. INTRUSION DETECTION SYSTEM

There are two general methods of detecting intrusions into computer and network systems: anomaly detection and signature recognition [2]. Anomaly detection techniques establish a profile of the subject's normal behavior (norm profile), compare the observed behavior of the subject with its norm profile, and signal intrusions when the subject's observed behavior differs significantly from its norm profile. Signature recognition techniques recognize signatures of known attacks, match the observed behavior with those known signatures, and signal intrusions when there is a match.

An IDS installed on a network is like a burglar alarm system installed in a house. Through various methods, both detect when an intruder/burglar is present. Both systems issue some type of warning in case of detection of presence of intrusion/burglar.

Systems which use misuse-based techniques contain a number of attack descriptions, or 'signatures', that are matched against a stream of audit data looking for evidence of the modeled attacks. The audit data can be gathered from the network, from the operating system, or from application log files [2]. Experimentation conducted in this research work is based on DARPA KDD'99 data set.

3. DARPA'99 DATA SET

MIT Lincoln Lab's DARPA intrusion detection evaluation datasets have been employed to design and test intrusion detection systems. The KDD 99 intrusion detection datasets are based on the 1998 DARPA initiative, which provides designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies [3][12][13].

To do so, a simulation is made of a factitious military network consisting of three 'target' machines running various operating systems and services. Additional three machines are then used to spoof different IP addresses to generate traffic. Finally, there is a sniffer that records all network traffic using the TCP dump format. The total simulated period is seven weeks [13]. Packet information in the TCP dump file is summarized into connections. Specifically, "a connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows from a source IP address to a target IP address under some well defined protocol" [13].

DARPA KDD'99 dataset represents data as rows of TCP/IP dump where each row consists of computer connection which is characterized by 41 features.

Features are grouped into four categories:

- ***Basic Features:*** Basic features can be derived from packet headers without inspecting the payload.
- ***Content Features*****:** Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts;
- ***Time-based Traffic Features*****:** These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval;
- ***Host-based Traffic Features*****:** Utilize a historical window estimated over the number of connections – in this case 100 – instead of time. Host based features are therefore designed to assess attacks, which span intervals longer than 2 seconds.

In this study, we used KDD'99 dataset which is counting almost 494019 of training connections. Based upon a discriminate analysis, we used data about only important features (the 9^{th} first features):

- ***Protocol type:*** type of the protocol, e.g. tcp, udp, etc.
- ***Service:*** network service on the destination, e.g., http, telnet, etc.
- ***Land****:* 1 if connection is from/to the same host/port; 0 otherwise.
- ***Wrong fragment****:* number of ``wrong'' fragments.
- ***Num_failed_logins:*** number of failed login attempts.
- ***Logged_in:*** 1 if successfully logged in; 0 otherwise.
- ***Root_shell:*** 1 if root shell is obtained; 0 otherwise.
- ***Is_guest_login:*** 1 if the login is a ``guest'' login; 0 otherwise.

To these features, we added the "attack_type". Indeed each training connection is labelled as either normal, or as an attack with specific type.

DARPA'99 base counts 38 attacks which can be gathered in four main categories:

- ***Denial of Service (dos):*** Attacker tries to prevent legitimate users from using a service.
- ***Remote to Local (r2l):*** Attacker does not have an account on the victim machine, hence tries to gain access.
- ***User to Root (u2r):*** Attacker has local access to the victim machine and tries to gain super user privileges.
- ***Probe:*** Attacker tries to gain information about the target host.

4. BAYESIAN NETWORK

A Bayesian network is a graphical modeling tool used to model decision problems containing uncertainty. It is a directed acyclic graph where each node represents a discrete random variable of interest. Each node contains the states of the random variable that it represents and a conditional probability table (CPT) which give conditional probabilities of this variable such as realization of other connected variables, based upon Bayes rule:

$$P(B/A) = \frac{P(A/B)P(B)}{P(A)}$$

The CPT of a node contains probabilities of the node being in a specific state given the states of its parents. The parent-child relationship between nodes in a Bayesian network indicates the direction of causality between the corresponding variables. That is, the variable represented by the child node is causally dependent on the ones represented by its parents [4][14].

Several researchers have been interested by using Bayesian network to develop intrusion detection systems. Axelsson in [5] wrote a well-known paper that uses the Bayesian rule of conditional probability to point out the implications of the *base-rate fallacy* for intrusion detection. It clearly demonstrates the difficulty and necessity of dealing with false alerts.

Kruegel in [1] presented a model that simulates an intelligent attacker using Bayesian techniques to create a plan of goal-directed actions. An event classification scheme is proposed based on Bayesian networks. Bayesian networks improve the aggregation of different model outputs and allow one to seamlessly incorporate additional information.

Johansen in [6] suggested that a Bayesian system which provides a solid mathematical foundation for simplifying a seemingly difficult and monstrous problem that today's Network IDS fail to solve. He added that Bayesian Network IDS should differentiate between attacks and the normal network activity by comparing metrics of each network traffic sample.

5. BAYESIAN NETWORK LEARNING ALGORITHM

Methods for learning bayesian graphical models can be partitioned into at least two general classes of methods: constraint-based search and Bayesian methods.

The constraint-based approaches [7][8] search the data for conditional independence relations from which it is in principle possible to deduce the Markov equivalence class of the underlying causal graph. Two notable constraint based algorithms are the PC algorithm which assumes that no hidden variables are present and the FCI algorithm which is capable of learning something about the causal relationships even assuming there are latent variables present in the data [7].

Bayesian methods [9] utilize a search-and-score procedure to search the space of DAGs, and use the posterior density as a scoring function. There are many variations on Bayesian methods, however, most research has focused on the application of greedy heuristics, combined with techniques to avoid local maxima in the posterior density (e.g., greedy search with random restarts or best first searches).

Both constraint-based and Bayesian approaches have advantages and disadvantages. Constraint-based approaches are relatively quick and possess the ability to deal with latent variables. However, constraint-based approaches rely on an arbitrary significance level to decide independencies.

Bayesian methods can be applied even with very little data where conditional independence tests are likely to break down. Both approaches have the ability to incorporate background knowledge in the form of temporal ordering, or forbidden or forced arcs. Also, Bayesian approaches are capable of dealing with incomplete records in the database. The most serious drawback to the Bayesian approaches is the fact that they are relatively slow.

In this paper, we are dealing with incomplete records in the database so we opted for the Bayesian approach and particularly for the K2 algorithm.

K2 learning algorithm showed high performance in many research works. The principle of K2 algorithm, proposed by Cooper and Herskovits, is to define a database of variables: X1,..., Xn, and to

build an acyclic graph directed (DAG) based on the calculation of local score [10]. Variables constitute network nodes. Arcs represent "causal" relationships between variables.
Algorithm K2 used in learning step needs:

- A given order between variables
- and the number of parents, u of the node.

K2 algorithm proceeds by starting with a single node (the first variable in the defined order) and then incrementally adds connection with other nodes which can increase the whole probability of network structure, calculated using the *g* function. A requested new parent which does not increase node probability can not be added to the node parent set.

$$g(x_i, pa_i(x_i)) = \prod_{j=1}^{qi} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{ri} N_{ijk}!$$

where, for each variable *xi; ri* is the number of possible instantiations; *N* is the number of cases in the database; *wij* is the *j*-th instantiation of **pa***i* in the database; *qi* is the number of possible instantiations for **pa***i*; *Nijk* is the number of cases in *D* for which *xi* takes the value *xik* with **pa***i* instantiated to *wij* ; *Nij* is the sum of *Nijk* for all values of *k*.
Execution time is in the order $O(Nu^2n^2r)$ with r being the maximum value for ri [10].

K2 Algorithm

Input: A set of variables $x_1,\ldots, x_n$, a given order among them, an upper limit *u* on the number of parents for a node, a database on $x_1,\ldots, x_n$.
***Output*:** a DAG with oriented arcs.

For i := 1 to n do
$pa_i(x_i) = \emptyset$; *OK : = true ;*
$P_{old} := g(x_i, pa_i(x_i))$;
While OK and $|pa_i(x_i)| < u$ *do*
Let z be the node in the set of predecessors of xi that does not belong to $pa_i(x_i)$ *which maximizes* $g(x_i, pa_i(x_i) \cup \{z\})$;
$P_{new} := g(xi, pai(xi) \cup \{z\})$;
If $P_{new} > P_{old}$ *Then*
$P_{old} := P_{new}$;
$pa_i(x_i) := pa_i(x_i) \cup \{z\}$;
Else OK := false ;

We ordered network variables as follows: *protocole_type, sevice, land, wrong_fragment, num_failed_logins, logged_in, root_shell, is_guest_login, attack_type.*

We had chosen the number 4 as the upper limit of node parents. Learning result is shown in Figure1:

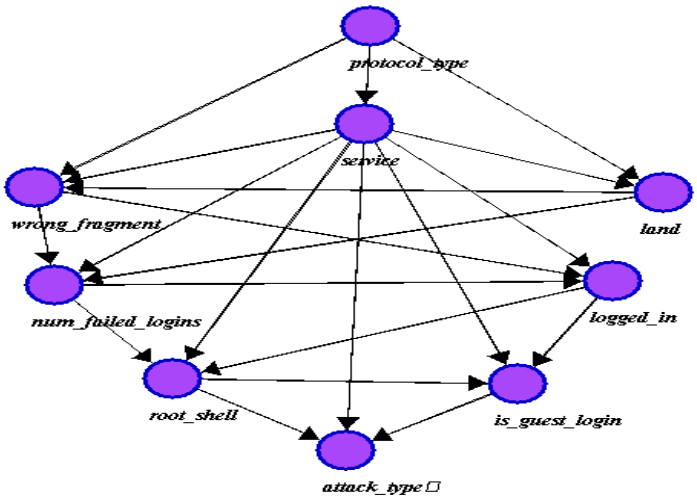

Figure 1 : K2 Bayesian Network

6. JUNCTION TREE INFERENCE ALGORITHM

The most common method to perform discrete exact inference is the Junction Tree algorithm developed by Jensen [11].

The idea of this procedure is to construct a data structure called a junction tree which can be used to calculate any query through message passing on the tree.
The first step of JT algorithm creates an undirected graph from an input DAG through a procedure

called moralization. Moralization keeps the same edges, but drops the direction, and then connects the parents of every child. Junction tree construction follows four steps:

- **JT Inference Step1:** Choose a node ordering. Note that node ordering will make a difference in the topology of the generated tree. An optimal node ordering with respect to the junction tree is NP-hard to find.
- **JT Inference Step2:** Loop through the nodes in the ordering. For each node Xi, create a set Si of all its neighbours. Delete the node Xi from the moralized graph.
- **JT Inference Step3:** Build a graph by letting each Si be a node. Connect the nodes with weighted undirected edges. The weight of an edge going from Si to Sj is $|Si \cap Sj|$.
- **JT Inference Step4**: Let the junction tree be the maximal-weight spanning tree of the cluster graph.

7. PROBLEM DESCRIPTION

A major shortcoming of current IDSs that employ Bayesian network is that they can give a series of false alarms in cases of a noticeable systems environment modification. There can be two types of false alarms in classifying activities in case of any deviation from normal patterns: false positives and false negatives.

False positive alarms are issued when normal behaviors are incorrectly identified as abnormal and false negative alarms are issued when abnormal behaviors are incorrectly identified as normal. Though it is important to keep both types of false alarm rates as low as possible, the false negative alarms should be the minimum to ensure the security of the system.

To overcome this limitation, an IDS must be capable of adapting to the changing conditions typical of an intrusion detection environment. It is important that an IDS should have automatic adaptability to new conditions. Otherwise, an IDS may start to lose its edge. Such adaptability can be achieved by using incremental dataset which is automatically updated.

The goal of an IDS based on signature recognition is to recognize signatures of known attacks, match the observed behavior with those known signatures, and signal intrusion when there is a match. A major difficulty of this system is that intrusions signatures change over the time and the system must be retrained. An IDS must be able to adapt to these changes. The goal of this paper is to provide a framework for an adaptive intrusion detection system that uses Bayesian network.

8. A FRAMEWORK FOR AN ADAPTIVE INTRUSION DETECTION SYSTEM

In this section we propose a framework for an adaptive intrusion detection system using bayesian network. The learning dataset can be updated by adding new intrusions signatures.

Learning dataset contains signatures of normal connections and signatures of several types of known attacks. The framework process begins with classifying connections of the learning dataset into 2 classes: Normal/Intrusion by using the associations' rules. This will accelerate both the learning process from the dataset and the deduction process about any new connection (normal or intrusion).

In the case of a connection which is not registered in the learning dataset, the system will find low probability degrees (below threshold) for this connection to be either normal or intrusion. Normal connections signatures being registered in the dataset, we are going to consider intrusion every connection that is not labeled as normal in the dataset. Thus, the suspect connection will be added to the learning dataset and be labeled as intrusion.

The class Intrusion of the learning dataset is therefore an incremental class, automatically updated by our system.

In a second step, intrusions of the updated learning dataset can be classified into different classes. Classes of known attacks and another class that will be defined for the connections not initially registered in the learning dataset and have been recognized by the system as intrusions.

Therefore, the system will also be able to provide us with the attack types of known intrusions. This is very important in order to be able to take the necessary security measures.

Figure 2 presents an architecture for our framework:

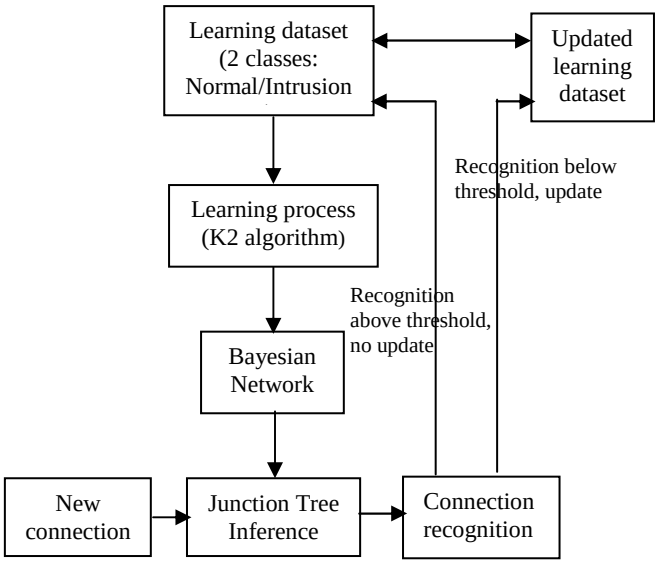

Figure 2: The framework for the adaptive intrusion detection system

9. EXPERIMENTATION RESULTS

The dataset used for experimentation is DARPA KDD'99 which contains signatures of normal connections and signatures of 38 known attacks gathered in four main classes: dos, r2l, u2r and probe. The class "other" contains detected intrusions that are not initially registered in KDD'99 dataset.

The main criterion that we have considered in the experimentation of our system is the detection rate. Detection rate is defined as the number of examples correctly classified by our system divided by the total number of test examples.

1. ***First step***: Connections in DARPA'99 are classified into 2 classes; each connection of the DARPA'99 dataset is labeled as either normal or intrusion.

Table 1: Detection Rate (1)

Connection	**Detection**
Normal (58714)	87.68 %
Intrusion (61960)	88.64%

2. ***Second Step***: Intrusions of the DARPA'99 dataset can be classified in 4 known classes: DOS, Probing, R2L and U2R. The class "other" is defined for the connections not initially registered in DARPA'99 dataset and have been recognized as intrusions by the system.

Table 2: Detection Rate (2)

Intrusion	**Detection**
DOS (61960)	88.64%
Probing (827)	99.15%
R2L (30)	20.88 %
U2R (15)	6.66%
Other (200)	66.51%

Table 1 shows a high performance of our system in detection of Normal and Intrusion connections.

Table 2 shows a high performance of our system in detection of DOS, Probing and Other connections. The low performance of our system in detection of R2L and U2R connections may be explained by the low proportions of the R2L and U2R training connections.

10. CONCLUSION

In this paper, we outlined a framework for an adaptive intrusion detection system using bayesian network. Bayesian networks provide automatic detection capabilities, they learn from audit data and can detect both normal and abnormal connections.

Our system demonstrated a high performance when detecting Intrusions. This system can be improved by integrating an expert system which is able to provide recommendations based on attack types.

Another alternative consists of using possibilistic networks rather than bayesian networks to better qualitatively represent intrusion risk evaluation.

REFERENCES

[1] Kruegel Christopher, Darren Mutz William, Robertson Fredrik Valeur. Bayesian Event Classification for Intrusion Detection Reliable Software Group University of California, Santa Barbara, , 2003.

[2] Brian C. Rudzonis. Intrusion Prevention: Does it Measure up to the Hype? SANS GSEC Practical v1.4b, 2003.

[3] DARPA. Knowledge Discovery in Databases, 1999. DARPA archive. Task Description http://www.kdd.ics.uci.edu/databases/kddcup99/task.htm

[4] Jensen F. Bayesian Networks and Decision Graphs. Springer, New York, USA, 2001.

[5] Axelsson S. The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection. In 6th ACM Conference on Computer and Communications Security, 1999.

[6] Johansen Krister and Lee Stephen. Network Security: Bayesian Network Intrusion Detection (BNIDS) May 3, 2003.

[7] Peter Spirtes, Clark Glymour, and Richard Scheines. Causation, Prediction, and Search. Springer Verlag, New York, 1993.

[8] Thomas S. Verma and Judea Pearl. Equivalence and synthesis of causal models. In P.P. Bonissone, M. Henrion, L.N. Kanal, and J.F. Lemmer, editors, Uncertainty in Artificial Intelligence 6, pages 255-268. Elsevier Science Publishers B.V. (North Holland), 1991.
.

[9] Gregory F. Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. Machine Learning, 1992.

[10] Sanguesa R., Cortes U. Learning causal networks from data: a survey and a new algorithm for recovering possibilistic causal networks. AI Communications 10, 31–61, 1997.

[11] Jensen Frank, Jensen Finn V. and Dittmer Soren L. From influence diagrams to junction trees. Proceedings of UAI, 1994.

[12] ISTG. The 1998 intrusion detection off-line evaluation plan. MIT Lincoln Lab., Information Systems Technology Group, 1998. http://www.11.mit.edu/IST/ideval/docs/1998/id98-eval-11.txt

[13] Kayacik, G. H., Zincir-Heywood, A. N. Analysis of Three Intrusion Detection System Benchmark Datasets Using Machine Learning Algorithms, Proceedings of the IEEE ISI 2005 Atlanta, USA, May, 2005.

[14] Peter Spirtes, Thomas Richard-son, and Christopher Meek. Learning Bayesian networks with discrete variables from data. In Proceedings of the First International Conference on Knowledge Discovery and Data Mining, pages 294-299, 1995.

[15] A. Valdes and K. Skinner. Adaptive, Model-based Monitoring for Cyber Attack Detection. In Proceedings of RAID 2000, Toulouse, France, October 2000.

[16] David Maxwell Chickering. A transformational characterization of equivalent Bayesian network structures. In Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-95), pages 87-98, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[17] David Maxwell Chickering. Learning equivalence classes of Bayesian network structures. In Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI-96), pages 150-57, Portland, Oregon, 1996.

[18] David. Heckerman, Dan Geiger, and D. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning, 1995.

[19] David Heckerman. Bayesian networks for data mining. Data Mining and Knowledge Discovery, 1998.

[20] David Madigan, Steen A. Anderson, Michael D. Perlman, and Chris T. Volinsky. Bayesian model averaging and model selection for Markov equivalence classes of acyclic digraphs. Technical report, see http://www.stat.washington.edu/bayes/papers.html, University of Washington, November 1995.

[21] Moninder Singh and Marco Valtorta. An algorithm for the construction of Bayesian network structures from data. In Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI-93), pages 259-265, San Francisco, CA, 1993. Morgan Kaufmann Publishers.

[22] Steen Anderson, David Madigan, and Michael Perlman. A characterization of Markov equivalence classes for acyclic digraphs. Technical report 287, see http://www.stat.washington.edu/tech.reports, University of Washington, 1995.