Intrusion Detection System using K2 Self Learning Algorithm and Open Attacking Platform

Md Tarik, Amandeep Singh

Abstract—The goal of a this IDS is to identify malicious behaviour that targets a network or a host 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 here used is a graphical modeling tool which used to model decision problems containing uncertainty. BN and K2 learning along with open attacking system is used here to make an automatic self-learning intrusion detection system based on signature recognition. But here is the goal to detect not only signature of attack also identifying the new pattern of new attack and storing its signature to database. Also here a host based IDS attached to backside of the network based IDS to provide security not only from outside but also from insiders.

Keywords—Intrusion Detection; IDS; Network Security; Bayesian Network; K2 Learning; Network Based; Host Based; Anomaly; Hacking.

1. Introduction

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

An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station. IDS come in a variety of "flavours'" and approach the goal of detecting suspicious traffic in different ways. There are network based (NIDS) and host based (HIDS) intrusion detection systems. Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPSes for other purposes, such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. IDPSes have become a necessary addition to the security infrastructure of nearly every organization

2.1. Network Intrusion Detection Systems

Network Intrusion Detection Systems (NIDS) are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. It performs an analysis of passing traffic on the entire subnet, works in a promiscuous mode, and matches the traffic that is passed on the subnets to the library of known attacks. Once an attack is identified, or abnormal behavior is sensed, the alert can be sent to the administrator. An example of an NIDS would be installing it on the subnet where firewalls are located in order to see if someone is trying to break into the firewall. Ideally one would scan all inbound and outbound traffic, however doing so might create a bottleneck that would impair the overall speed of the network. OPNET and NetSim are commonly used tools for simulation network intrusion detection systems maintaining the Integrity of the Specifications

2.2. Host Intrusion Detection Systems

Host Intrusion Detection Systems (HIDS) run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator if suspicious activity is detected. It takes a snapshot of existing system files and matches it to the previous snapshot. If the critical system files were modified or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission critical machines, which are not expected to change their configurations.

Intrusion detection systems can also be system-specific using custom tools and honeypots.

3. K2 Learning

K2 is an algorithm for constructing a Bayes Network from a database of records. "A Bayesian Method for the Induction of Probabilistic Networks from Data", Gregory F. Cooper and Edward Herskovits, Machine Learning 9, 1992

3.1. K2 algorithm: a heuristic search method

Use the following functions:

$$g(i,\pi_i)=\prod_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\prod_{k=1}^{r_i}N_{ijk}!$$

Where the N_{ijk} are relative to π_i being the parents of x_i and relative to a database D

$$Pred(x_i)=\{x_1,\ldots x_{i-1}\}$$

It returns the set of nodes that precede x_i in the node ordering.

Input: A set of nodes, an ordering on the nodes, an upper bound u on the number of parents a node may have, and a database D containing m cases

```
Procedure K2
For i:=1 to n do
 π_i = ϕ;
P_old = g(i, π_i );
OKToProceed := true
while OKToProceed and | π_i |<u do
        let z be the node in Pred(x_i)- π_i that
maximizes g(i, π_i ∪{z});
        P_new = g(i, π_i ∪{z});
        if P_new > P_old then
          P_old := P_new ;
          π_i :=π_i ∪{z} ;
        else OKToProceed := false;
end {while}
write("Node:", "parents of this nodes :", π_i
);
end {for}
end {K2}
```

Output: For each nodes, a printout of the parents of the node

4. Bayesian Network

A Bayes network B = (B_S, B_P). A Bayes Network structure B_S is a directed acyclic graph in which nodes represent random domain variables and arcs between nodes represent probabilistic independence. B_S is augmented by conditional probabilities, B_P, to form a Bayes Network B.

- B_S of Bayes Network: the structure

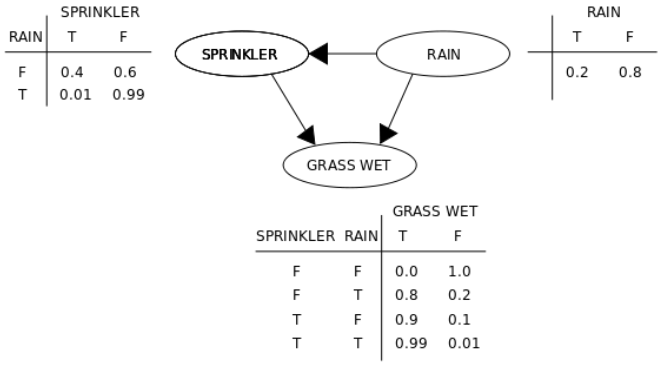

Figure 1 Bayes Network: the structure example

Inference and learning

There are three main inference tasks for Bayesian networks.

4.1. Inferring unobserved variables

Because a Bayesian network is a complete model for the variables and their relationships, it can be used to answer probabilistic queries about them. For example, the network can be used to find out updated knowledge of the state of a subset of variables when other variables (the evidence variables) are observed. This process of computing the posterior distribution of variables given evidence is called probabilistic inference. The posterior gives a universal sufficient statistic for detection applications, when one wants to choose values for the variable subset which minimize some expected loss function, for instance the probability of decision error. A Bayesian network can thus be considered a mechanism for automatically applying Bayes' theorem to complex problems.

The most common exact inference methods are: variable elimination, which eliminates (by integration or summation) the non-observed non-query variables one by one by distributing the sum over the product; clique tree propagation, which caches the computation so that many variables can be queried at one time and new evidence can be propagated quickly; and recursive conditioning and AND/OR search, which allow for a space-time trade-off and match the efficiency of variable elimination when enough space is used. All of these methods have complexity that is exponential in the network's tree width. The most common approximate inference algorithms are importance sampling, stochastic MCMC simulation, mini-bucket elimination, loopy belief propagation, generalized belief propagation, and variational methods.

4.2. Parameter learning:

In order to fully specify the Bayesian network and thus fully represent the joint probability distribution, it is necessary to specify for each node X the probability distribution for X conditional upon X's parents. The distribution of X conditional upon its parents may have any form. It is common to work with discrete or Gaussian distributions since that simplifies calculations. Sometimes only constraints on a distribution are known; one can then use the principle of maximum entropy to determine a single distribution, the one with the greatest entropy given the constraints. (Analogously, in the specific context of a dynamic Bayesian network, one commonly specifies the conditional distribution for the hidden state's temporal evolution to maximize the entropy rate of the implied stochastic process.)

Often these conditional distributions include parameters which are unknown and must be estimated from data, sometimes using the maximum likelihood approach. Direct maximization of the likelihood (or of the posterior probability) is often complex when there are unobserved variables. A classical approach to this problem is the expectation-maximization algorithm which alternates computing expected values of the unobserved variables conditional on observed data, with maximizing the complete likelihood (or posterior) assuming that previously computed expected values are correct. Under mild regularity conditions this process converges on maximum likelihood (or maximum posterior) values for parameters.

A more fully Bayesian approach to parameters is to treat parameters as additional unobserved variables and to compute a full posterior distribution over all nodes conditional upon observed data, then to integrate out the parameters. This approach can be expensive and lead to large dimension models, so in practice classical parameter-setting approaches are more common.

4.3. Structure learning:

In the simplest case, a Bayesian network is specified by an expert and is then used to perform inference. In other applications the task of defining the network is too complex for humans. In this case the network structure and the parameters of the local distributions must be learned from data.

Automatically learning the graph structure of a Bayesian network is a challenge pursued within machine learning. The basic idea goes back to a recovery algorithm developed by Rebane and Pearl (1987) and rests on the distinction between the three possible types of adjacent triplets allowed in a directed acyclic graph (DAG):

1. $X \rightarrow Y \rightarrow Z$
2. $X \leftarrow Y \rightarrow Z$
3. $X \rightarrow Y \leftarrow Z$

Type 1 and type 2 represent the same dependencies (X and Z are independent given Y) and are, therefore, indistinguishable. Type 3, however, can be uniquely identified, since X and Z are marginally independent and all other pairs are dependent. Thus, while the skeletons (the graphs stripped of arrows) of these three triplets are identical, the directionality of the arrows is partially identifiable. The same distinction applies when X and Z have common parents, except that one must first condition on those parents. Algorithms have been developed to systematically determine the skeleton of the underlying graph and, then, orient all arrows whose directionality is dictated by the conditional independencies observed.

An alternative method of structural learning uses optimization based search. It requires a scoring function and a search strategy. A common scoring function is posterior probability of the structure given the training data. The time requirement of an exhaustive search returning a structure that maximizes the score is super exponential in the number of variables. A local search strategy makes incremental changes aimed at improving the score of the structure. A global search algorithm like Markov chain Monte Carlo can avoid getting trapped in local minima. Friedman et al. discuss using mutual information between variables and finding a structure that maximizes this. They do this by restricting the parent candidate set to k nodes and exhaustively searching therein.

Another method consists of focusing on the sub-class of decomposable models, for which the MLE have a closed form. It is then possible to discover a consistent structure for hundreds of variables.

A Bayesian network can be augmented with nodes and edges using rule-based machine learning techniques. Inductive logic programming can be used to mine rules and create new nodes. Statistical relational learning (SRL) approaches use a scoring function based on the Bayes network structure to guide the structural search and augment the network. A common SRL scoring function is the area under the ROC curve.

5. Problem Description

Here almost all IDS for WLAN are mostly based on signature detection of only known patterns and either host based or network. Also there are a series of false positive alarm issue present in existing models. Most of the IDS fails

where there is a new type of attack or intrusion happens because of the IDS haven't any knowledge of that type of attack in their knowledgebase.

A false positive alarm is an issue when normal system behaviour is alarmed as abnormal or intrusion. So all the above scenario lead me to make a WIDS which have

learning environment. Here the learning dataset is transferred as an update to all local hosts in the protected network for internal intrusion detection (i.e. for internal attack).

Learning dataset contains all the new attack patterns and also some new type of normal connection or access to the system to

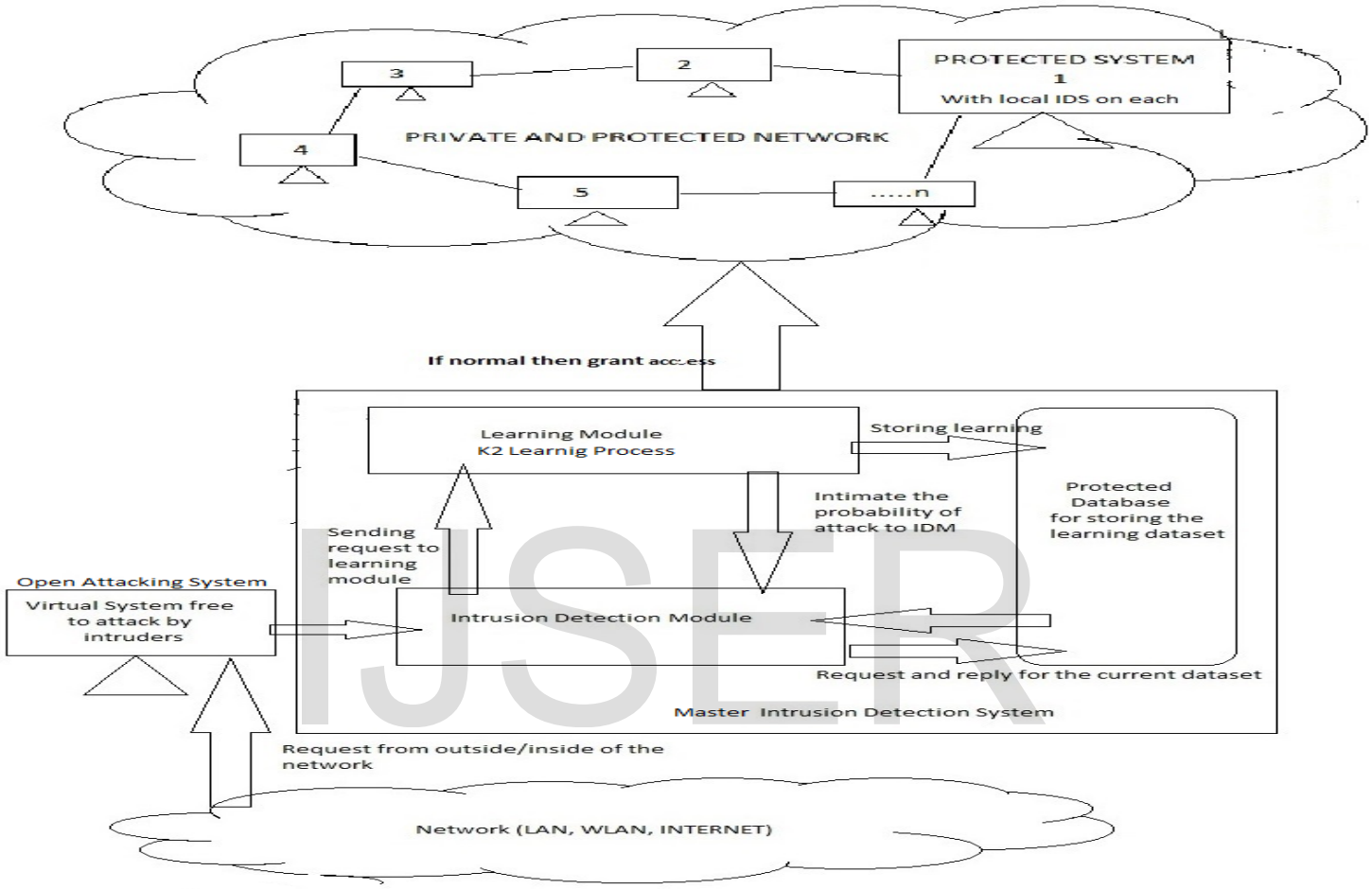

Figure 2 Framework for Smart IDS

possess very low false positive alarms and also very low false negative alarm as well.

To solve that type of limitations here the IDS must be capable of self-learning means adaptive in nature by means of self-learning algorithm like K2.

The IDS must have the goal to recognize the signature of known attacks as well retrain itself to the types or unknown types of attack patterns and inform the administration as soon as possible. Here the main difficulty is to provide the training of system for the changing nature of attack signature day by day by the advancement of technology over the time. The goal of this paper is to provide a framework for both a host based as well as a network based system along with the self-learning scheme.

6. Framwork For Intrusion Detection

Here in this field I propose a framework that work on both network and individual hosts in a network with self-reduce false alarms.

Here the virtual system is made free to attack it just like fishing for the attacker to utilize their skills which is monitored by Intrusion detection super system and all the abnormal behavior is noted as a signature into the database and propagated to all the local IDS on each host in the network.

The master IDS have a capability of learning by means of K2 learning algorithms. If in any rule or protocol set in the master IDS is violated then the IDS start recording the signature of the connection and all request patterns from the requester. In other hand if everything is normal as per the protocol boundary the user granted to access the system.

7. Open Attacking System

It is the very start point and the most important part of the whole IDS as it if made free to attack. There is a question may arise that why I made this system free to attack or why we need this? The answer is here it is a virtual system which seems to be the real system which is request by the user and user haven't any knowledge about its actuality and this may be the same for the attackers and the attackers try to steal information from that system by applying their skills on this virtual system which is monitored by the master IDS. This virtual system has dummy information but looks like the protected host to outside the world. Also there is a mirror copy or image of this virtual system present in the database of master IDS. If any file or any change happened on the virtual system start monitoring the connection for learning by master IDS.

Also this virtual system gives us some time to take decisions and to recognize the request as well.

8. Conclusion

In this paper we have tried to make a WIDS which work on both network and host based system which train itself to detect both normal and abnormal connection

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

Acknowledgment

This work is dedicated to my parents and all my friends who support me. A special thanks to my guide Dr. Ruchi Agarwal, and my friend Mr. Amandeep Singh who help me a lot in my work with their precious time and knowledge.

References

[1] More, S. ; Matthews, M. ; Joshi, A. ; Finin, T. "A Knowledge-Based Approach to Intrusion Detection Modeling" IEEE Symposium on Security and Privacy Workshops (SPW), 2012

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

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

[4] DARPA. Knowledge Discovery in Databases, 1999. DARPA archive. Task Description http://www.kdd.ics.uci.edu/databases/kddcuip99/task.htm

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

[6] 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.

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

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

[9] 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.

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