Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.790 – 795

RESEARCH ARTICLE

IMPLEMENTATION OF IDS USING SNORT ON BAYESIAN NETWORK

[1]M.Chandu Jagan Sekhar, [2]K.Tulasi, [3]V.V.Amulya, [4]D.Ravi Teja, [5]M.Santhosh Kumar

[1]Assistant Professor, Dept of CSE, VITS College of Engineering, Visakhapatnam

[2345]B.Tech Student, VITS, Visakhapatnam

mchandujagansekhar@gmail.com, tulasivits20@gmail.com, vvamulya@rediffmail.com,
ravitejadaley9@gmail.com, santhoshmarrapu@gmail.com

ABSTRACT

The goal of a network-based intrusion detection system (IDS) is to identify malicious behavior that targets a network and its resources. In present day technology we have got many preferences but no technology is so helpful to diminish the existing problem completely. 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.

***Keywords**: Bayesian Network, IDS, Signature Based.*

I. INTRODUCTION

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 "flavors" 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 IDPS for other purposes, such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. IDPS have become a necessary addition to the security infrastructure of nearly every organization.

IDPS typically record information related to observed events, notify security administrators of important observed events and produce reports. Many IDPS can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g. reconfiguring a firewall) or changing the attack's content.

An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system.

There are several ways to categorize an IDS:

Misuse detection vs. anomaly detection: in misuse detection, the IDS analyzes the information it gathers and compares it to large databases of attack signatures. Essentially, the IDS looks for a specific attack that has already been documented. Like a virus detection system, misuse detection software is only as good as the database of attack signatures that it uses to compare packets against. In anomaly detection, the system administrator defines the baseline, or normal, state of the networks traffic load, breakdown, protocol, and typical packet size. The anomaly detector monitors network segments to compare their state to the normal baseline and look for anomalies.

Network-based vs. Host-based systems:

In a network-based system, or NIDS, the individual packets flowing through a network are analyzed. The NIDS can detect malicious packets that are designed to be overlooked by a firewalls simplistic filtering rules. In a host-based system, the IDS examines at the activity on each individual computer or host.

Passive system vs. Reactive system:

In a passive system, the IDS detects a potential security breach, logs the information and signals an alert. In a reactive system, the IDS responds to the suspicious activity by logging off a user or by reprogramming the firewall to block network traffic from the suspected malicious source.

Though they both relate to network security, an IDS differs from a firewall in that a firewall looks out for intrusions in order to stop them from happening. The firewall limits the access between networks in order to prevent intrusion and does not signal an attack from inside the network. An IDS evaluates a suspected intrusion once it has taken place and signals an alarm. An IDS also watches for attacks that originate from within a system.

It generally deals with two sort of problems. They are

1. Anomaly based
2. Signature based

II. Related Work

Existing System

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.

Proposed System:

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.

SNORT**:**

Snort is a free and open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS) created by Martin Roesch in 1998. Snort is now developed by Source fire, of which Roesch is the founder and CTO. In 2009, Snort entered InfoWorld's Open Source Hall of Fame as one of the "greatest [pieces of] open source software of all time". Snort's open source network-based intrusion detection system (NIDS) has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching, and content matching. These basic services have many purposes including application-aware triggered quality of service, to de-prioritize bulk traffic when latency-sensitive applications are in use.

The program can also be used to detect probes or attacks, including, but not limited to, operating system fingerprinting attempts, common gateway interface, buffer overflows, server message block probes, and stealth port scans.

Snort can be configured in three main modes: sniffer, packet logger, and network intrusion detection. In sniffer mode, the program will read network packets and display them on the console. In packet logger mode, the program will log packets to the disk. In intrusion detection mode, the program will monitor network traffic and analyze it against a rule set defined by the user. The program will then perform a specific action based on what has been identified.

PACKET CAPTURE:

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, color-coding and other features that let you dig deep into network traffic and inspect individual packets. You can use Wireshark to inspect a suspicious program's network traffic, analyze the traffic flow on your network, or troubleshoot network problems.

III. MODULES

1. K2 Algorithm
2. Bayesian Recognition
3. Junction Tree Inference

MODULES DESCRIPTION - K2 Algorithm:

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.

Algorithm K2 used in learning step needs:

1. A given order between variables
2. 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 cannot be added to the node parent set.

K2 Algorithm:

```
procedure K2;
{Input: A set of n 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.}
{Output: For each node, a printout of the parents of the node.}
for i:= 1 to n do
    $\pi i := \emptyset$;
    Pold := f(i, $\pi i$); {This function is computed using Equation 20.}
    OKToProceed := true;
    While OKToProceed and $|\pi i| < u$ do
        let z be the node in Pred(xi) - $\pi i$ that maximizes f(i, $\pi i \cup \{z\}$);
        Pnew := f(i, $\pi i \cup \{z\}$);
        if Pnew > Pold then
            Pold := Pnew;
            $\pi i := \pi i \cup \{z\}$;
```

```
else OKToProceed := false;
end {while};
write('Node: ', xi, ' Parent of xi: ',πi);
end {for};
end {K2};
```

Bayesian Recognition:

Bayesian methods 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 application og greedy heuristic, combined with techniques to avoid local maxima in the posterior density (e.g., greedy search with random restarts or best first searches).

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.

Bayesian recognition classify the systems behavior in to two classes

1. Normal
2. Anomaly

Normal indicates that system is protected from any kind of attacks, and anomaly means that something happen wrong with system it means some kind of attack is made by intruders. It also provide some parametric value for all features that are present in intrusion dataset, these values are very helpful in determining to which extent attack is made. The result of Bayesian recognition is a input to construct a junction tree.

Junction Tree Inference:

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.

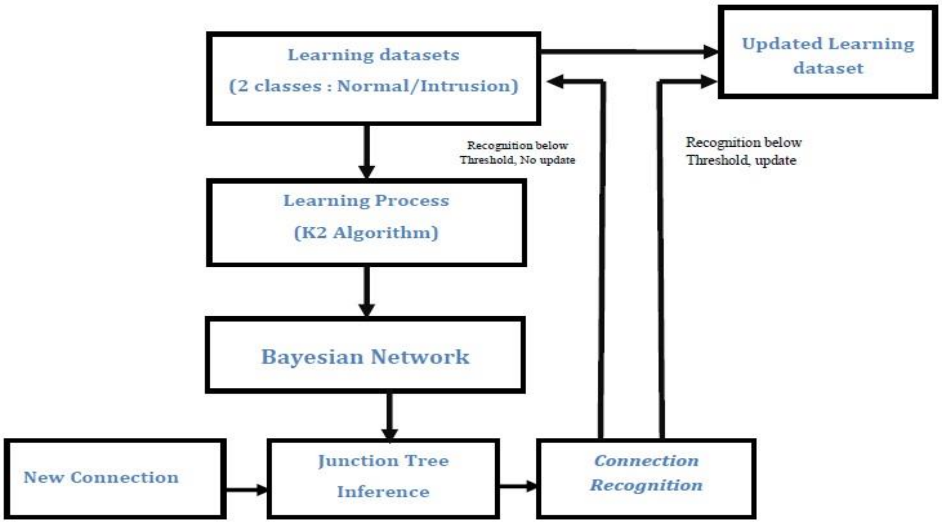
The first step of JT algorithm creates an undirected graph from an input DAG through a procedure called moralization. Moralization keeps the same edges, but drops the direction, and then connects the parents of every child.

The junction tree algorithms take as input a decomposable density and its junction tree. They have the same distributed structure:

- Each cluster starts out knowing only its local potential and its neighbors.
- Each cluster sends one message (potential function) to each neighbor.
- By combining its local potential with the messages it receives, each cluster is able to compute the marginal density of its variables.

Junction tree produce a final outcome which contain the information about the all connection or nodes respective with their Id no and it list the difference between actual and predicted class help us in ensuring whether the prediction was made correct or incorrect. By this means we can able to determine that attack is made or not. It is the last module of our system which starts working by constructing a network from DAG i.e. created by K2 learning process and also get the results of Bayesian recognition to determine the unauthorized access or applications.

Proposed Architecture:



IV. CONCLUSION

In this paper, we outlined a framework for an adaptive intrusion detection system using Bayesian network with a tool named SNORT . Bayesian networks provide automatic detection capabilities, they learn from audit data and can detect both normal and abnormal connections. Snort is a network monitoring tool and can also be used for capturing the packets that generally move in a network. 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.

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 vl.4b, 2003.
3. DARPA. Knowledge Discovery in Databases, 1999. DARPA archive. Task Description http://www.kdd.ics.uci.edu/databases/kddcuip99/task.htm