FUNCTIONING OF INTELLIGENCE INTRUSION MULTI DETECTION PREVENTION SYSTEMS (IIMDPS)

By

S. MURUGAN * K. KUPPUSAMY **

* Research Scholar, Department of Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu, India.
** Professor, Department of Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu, India.

ABSTRACT

This paper focuses on functioning of Intelligence Intrusion Multi Detection Prevention Systems (IIMDPS). It describes the prevention of unknown malware with the help of mathematical scheme and few models with newly designed algorithm. This is designed to provide a deeper understanding of existing intrusion detection principles with intelligence strategies, that will be responsible for acquiring unknown malware, which compare the false positive rate and the false negative rate. That will be proven by conducting different experiments with WEKA simulation.

Keywords: Intelligence Intrusion Detection Prevention System (IIDPS), Unknown Malware, Intelligence Intrusion Multi Detection Prevention Systems (IIMDPS).

INTRODUCTION

AI techniques can be used in building intelligent models to improve the information security management, intrusion detection and prevention capabilities, efficiency of security event management, and decision making (Hentea, 2003, 2004, 2005b, 2005c, 2006). Intelligent systems (Meystel & Albus, 2002) called intelligent assistants help the users in decision making process for configuring and monitoring specific metrics, faults and events correlation that could lead to the reconnaissance of the attack and prevention of the cyber attack. Efficient security management requires an intelligent system that supports security event management approach with enhanced real-time capabilities, adaptation, and generalization to predict possible attacks and to support human's actions.

The proposed IIDPS architecture includes elements of intelligence to create functional relationships and malware information flow between different subsystems. The elements of intelligence are based on components using one or more AI techniques like artificial neural networks, fuzzy logic. In addition to the development of intelligent system, it combines some AI techniques with other techniques such as conventional programs, statistical packages and object based and rule based data mining creating hybrid intelligent system architecture (Hentea, 1999).

The IIDPS architecture is based on the Real Control System (RCS) techniques. Intelligence in systems is created by a definite architecture that organizes joint functioning of Traffic Static Analyzer Model (TSAM), Port Matching Model (PMM), Filtering Model (FM), Artificial Neural Network Model (ANNM) and Artificial Immune System (AIS). All elements of intelligence are based on elementary functioning loop (self containing agent) which allows creating functional relationships and information flows. The cyber security of an enterprise is observed, controlled and it serves as a medium for elementary functioning loop activities.

At each level, plans are made and updated with different planning horizons. At each level, short term memory traces sensory data over different historical data intervals by using event log. At each level, feedback control loops have a characteristic. This model of a multi-resolutional hierarchy of computational loops yields deep insights into the phenomena of behavior, perception, cognition, problem solving, and learning.

1. Design Issues

A major decision to be made during the architectural design is what agents should be included. Several types of agents can be designed to support IIDPS. In the proposed system, TSAM and PMM as key agents should be the decision maker agent and controller agent.

An intelligent agent ANNM and AIS is viewed as a

combination of functionalities and intelligent capabilities. That ability to act in an uncertain environment, learning, adaptability, probability of success. Roles in some methodologies are things the agents that will perform by looking at the combinations of functionalities. Major contributor to the field of autonomous agents is artificial intelligence. The proposed system is based on the integration of different types of intelligent agents, hybrid architecture under realtime constraints. Intelligent agents helps in automating various tasks such as, gathering malware information, filtering, and using it for decision support and can help to improve the productivity of the administrator. The design and programming of agents should be focused on maximizing their performance measure which embodies the criterion for success of an agent's behavior. Other important issues that are required include portability, stability, resilience, and security of the agents and system. The interface should exhibit intelligent features that assist the user in decision making and taking actions to control the security process.

The design phase has to identify the type of feedback available for learning because, it is usually the most important factor in determining the nature of the learning problem that the agent faces. The field of machine learning usually distinguishes the cases of supervised and unsupervised learning. The scope of IIDPS is broad and requires using a single or a combination of both forms for getting the best results. Another characteristic that should be considered in the mobility is the degree to which the agents travel through the network.

The input data to the models for learning and outputs of the models play an important role in the design. Principal factor in the design will consider the availability of prior knowledge for some tasks of IIDPS. The majority of learning will begin with no knowledge at all about what the agent is trying to learn. Learning takes place as the agent observes its interactions with the environment and its own decision making processes. Learning is a process of self improvement and thus, an important feature of intelligent behaviors. The order of implementation of the models is dependent on the resources and needs.

Data mining supports automated analysis and interpretations of the data and events collected from different sources as well as discovery of associations among data and events and feedback to human user.

Artificial neural networks support classification, association, and prediction of future cyber attacks by learning, adapting from past, current data and events. For example, investigation patterns can be classified using neural networks based on unsupervised learning.

Fuzzy logic allows processing of qualitative variables and approximate reasoning when the propositions are inexact and vague. One model is used for risk assessment. However, a synergy between different approaches can serve to enhance and highlight the qualitative aspects of each model, thus creating knowledge and intelligence for assisting the human to make decisions. A possible avenue for integrating data mining, neural networks, and fuzzy expert systems (Hentea, 1997) in addressing the intrusion attempts would be to use the object and rule based data mining and neural network to discover and to classify the reconnaissance patterns and its attributes. This information can be communicated to fuzzy expert system that could return advice to human to take actions based on the status of intrusion attempts. Further, neural networks can recognize patterns and predict possible cyber attacks. Also, neural networks can draw conclusions from fuzzy or uncertain data about a given situation. The knowledge-base incorporates knowledge for the security domain such as, raw data and events, performance measures, patterns, policies, and decisions.

2. Architecture of IIDPS

Figure 1 shows the architecture of the proposed Intelligent

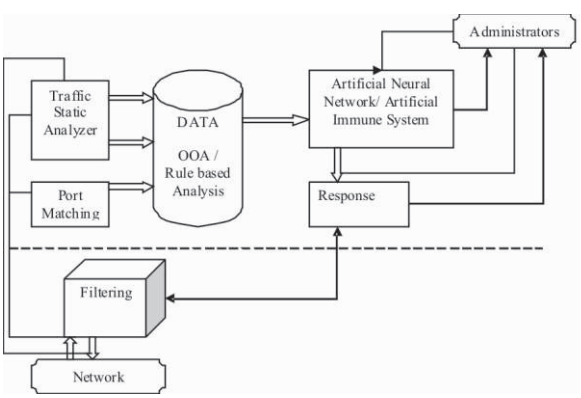

Figure 1. Architecture of IIDPS

Intrusion Detection and Prevention System. This system is used to identify malware traffic from normal traffic; also it can predict the infection percentage in the network, which can be used by the administrator to take the appropriate action.

This system depends only on the data that collected from the local victim malware information. As seen in Figure, the system consists of 5 functioning modules:

- Traffic Statistical Analyzer Module
- Port Matching Module
- Artificial Neural Network Module
- Filtering Module
- Response Module

Function of the proposed IIDPS starts with monitoring the incoming and outgoing traffic using sniffing tool. The network traffic is used by TSAM to calculate network traffic statistics. The monitored traffic is used as input to the PMM, which use the idea of infection-like-behavior in malware spreading to identify suspected malware traffic. Then administrators apply the number of hosts online as an input to ANNM, which uses the data that collected from other modules to classify the traffic into malware traffic or normal traffic, and to predict the percentage of infection in the network.

2.1 Traffic Statistical Analyzer Module (TSAM)

This module is taking care for calculating statistical values based on the analysis of incoming and outgoing traffic for a new unknown or known malware entrant. It captures the traffic for finding known and unknown malware packets, calculates the number of packets per time unit, and the number of packets produced by each source/destination port in time unit. It produces number of packets per protocol in a time unit. But only the number of packets and number of packets per port that are used as an input to the data set for ANN. This module analyzes statistical properties of traffic generated by known and unknown malwares. Analyzing properties of aggregate traffic and separating it into streams are called as sessions - by source hosts, or by flows, etc. and considering not only sessions related variables as arrival times, size, duration but also packet-level variables inside sessions: Inter-Packet Times (IPT) and Packet Sizes (PS). Compare findings with other categories of unknown malware traffic. This is based on the observation of known and unknown traffic both traversing backbone links and captured by network telescopes. Because of this work, the issues involved in performing such kind of analysis, as the lack of useful traffic traces and the need for data refinement.

The process used can be synthetically sketched into a number of sequential steps depicted. After the traffic trace acquisition, human interaction is usually necessary to inspect the trace. The type of traffic captured is a first fundamental step before performing a detailed statistical analysis. To do this, it need an flexible tools that rapidly investigate several traffic properties from looking into headers and payload, that reporting concise information on hosts, flows, etc. From this analysis, it is possible to choose on which aspect to focus the characterization and to conceive strategies for intelligence trace refinement to remove bogus data.

As known malware send a single UDP packet to each victim host:

- The reports are analyzed with flows, immediately it locate unknown malware behaviors looking for flows with more than one packet;
- Malware after isolation in traffic, the software tool extracts measurements data from the traffic trace and it may also be able to perform a preliminary analysis. Finally, the data sets obtained can be loaded into statistical analysis software and analyzed, looking at marginal distributions, time dependence, correlations, etc.

While analyzing the data, look for repeating behaviors and, by applying the same analysis to malware and legitimate applications, aim at sketching similarities and differences. The overall traffic is compared before and during malware propagation may allow inferring information about the impact of malwares on links and nodes. The results are basically related to aggregate traffic and to the analysis of host-based sessions, focusing on packet-level variables.

A packet level analysis already adopted for the traffic

generated by legitimate applications. Being independent of the application-level protocol, it can be equally applied to different kinds of traffic. Furthermore, characterizing statistical properties of traffic at packet-level can help in building analytical and empirical models to be used for traffic generation and simulation, which represent another mean to better assess the impact of malware traffic on links and nodes. Finally, traffic at packet level remains observable after encryption made by SSL or IPSec, making packet-level traffic modeling a robust approach for traffic profiling of an anomaly detection and traffic classification.

This gives a classification of malware based on their traffic. There are two main areas which can be distinguished and each area can be subdivided into further criteria.

- The selection of potential targets
- Random or deterministic scanning
- Preference for the local subnet
- The generation of scanning traffic
- Protocol on the transport or network layer
- Port number
- Number of parallel connections respectively sending rate

After a malware establishes a connection to a victim host, it tries to propagate over the network by sending its code. This malware propagation phase is not considered by this classification, In case of UDP, the scanning traffic includes the propagation of the malware.

2.1.1 Potential Infection Victims Selection

Malware differ in their selection of possible targets gives an overview of the malwares and the nature of their IP selection mechanism (Wang et.al., 2007) as there are more techniques a malware could use. However, for further considerations permutation scanning can be assumed as some sort of random scanning. A hit-list indeed has an influence on the impact of a malware, but the traffic of such a malware does not differ very much from a malware without hit-list. Therefore, permutation scanning and hit-lists are not discussed in detail any further.

Email malware behave, as mentioned above, in a completely different manner. They can be seen as a separate class of malwares except for the known malware which only have parts behaving like an email malware. Email malwares do not need to choose any IP addresses.

2.1.2 Random

The column "random" tells if a random process is involved in the generation of IP addresses. The selection of IP addresses is often only partially random. Only SQL Slammer and Code Red I create completely random IP addresses. Mostly, the first 8 to 16 bits are taken from the own IP address and the remaining bits are generated at random.

Some malwares do a sequential scanning in combination with random scanning. They count up starting at a randomly calculated IP address.

2.1.3 Local Subnet

As mentioned above, an IP address is often created by taking the upper bits from the actual local address of the infected host. For example, if only the last 8 bits are chosen at random, this means that, the scanned machines are located in the same subnet with the subnet mask 255.255.255.0.

A scan of an IP in the local subnet does not pass a router at the boarder of this subnet. Therefore, only a scan of an external IP address can be observed at the outgoing link of a local subnet.

Most malware cover multiple scanning mechanisms. Often they use some kind of probabilistic function to decide between the implemented possibilities. Not only the scanning mechanism can depend on a probabilistic function, sometimes the choice between several vulnerabilities is also done in this way.

2.1.4 Generation of Scanning Traffic

This shows a classification based on the scanning traffic of malware. The scanning traffic consists of the first packets sent by a malware when trying to establish a connection to a potential victim. Table 1 gives an overview of the first packets sent by the various known malwares.

Table 2 shows the Malware Packet Propagation with

various protocols and ports connections to identify the known malware. Protocol column gives the name of the network protocol used by a malware which is often given by the exploited vulnerability. It is observed from the table that, the malwares used TCP and UDP over IP, except for the Welchia malware, which first sends an ICMP request. The Port column states the port number used by the protocol to identify whether the port open or closed. Some malwares like Nimda or Welchia make use of multiple vulnerabilities and try to connect with different ports. The Connection column in the table regards to the quantity of packets, through which the malware tries to establish connections. In TCP, a clever malware would use multiple threads to open many connections and wait for many answers at a time. UDP is not connection-oriented and therefore, a UDP malware can send as many packets as possible. The precise quantity can be given as packets sent per second.

2.2 Port Matching Module (PMM)

Being fully automated, a malware's behaviors usually repetitious and predictable, making it possible to be detected. After a vulnerable host is infected by a malware on a port I (i.e., the host is the destination of an early malware attack), the infected host will send out scans to other hosts targeting at the same port I in a short time. This module uses this idea to produce the number of packets per port that match the malware infection behavior. Since there is no way to know if a packet source is a victim or slave attacker, each record is being examined as if it is from the victim or from slave attacker. Then in a selected unified time interval, if a packet is sent from a slave to a victim on specific port, followed by a packet is sent from this victim IP address to the same destination port, thus is counted as malware-like behavior on that port. A dynamic table is made to produce number of occurrence for this malware like behavior per each used port. Figure 2 shows the working of Port Matching Module.

Known Malware	Random	Local subnet
Sasser B	X	X
Welchia A	X	X
Blaster A	X	X
SQL Slammer	X	
Code Red Iv2	X	
Nimda A	X	X
Morris	X	X

Table 1. Scanning Traffic

2.3 Artificial Neural Network Module (ANNM)

A supervised ANN can be trained to take the values that represent the current behavior of the network under non-malware traffic and malware traffic. After sufficient number of iterations, it can be used as a control unit in the proposed system to identify the malware traffic.

In this phase, two models named Classification, Prediction Combined (CPC) model and Classification,

Known Malware	Protocol	Port	Connection
Sasser B	TCP	445	128 in parallel
Welchia A	ICMP	-	As many packets as possible n/a
	TCP	80 &135	
Blaster A	TCP	135	20 in parallel
SQL Slammer	UDP	1434	As many packets as possible
Code Red Iv2	TCP	80	99 in parallel
Nimda A	TCP	80 &137-139/445	n/a
Netsky D	UDP(DNSMX)	53	n/a
	TCP (SMTP)	25	n/a

Table 2. Malware Packet Propagation

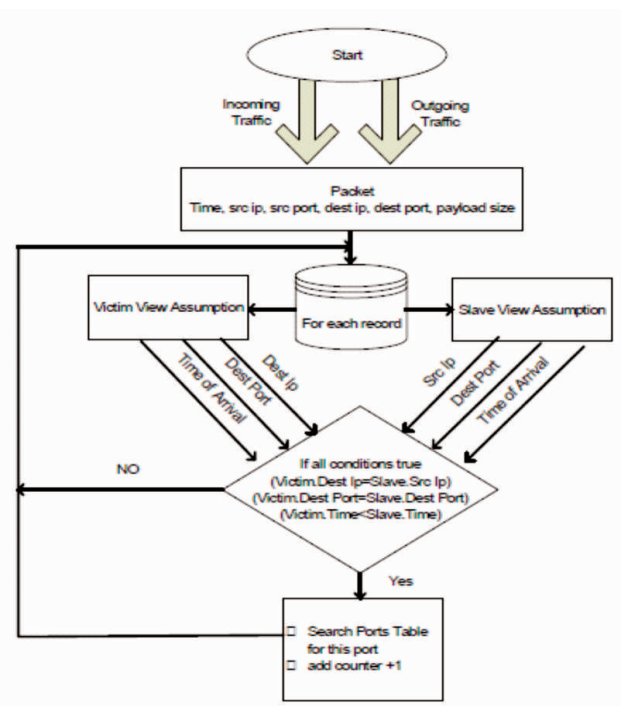

Figure 2. Port Matching Module

Prediction Separated (CPS) models are designed using ANN to classify and predict the behaviour class of incoming malware. In CPC, one ANN is used to produce results by combining classification and prediction of malware behaviour classes. In CPS, two ANNs are used to produce results separately one for classification and another for prediction of malware behaviour classes. The description of each model follows.

2.3.1 Classification Prediction Combined Model (CPC Model)

In this model, the ANN produces two desired outputs. Classification describing a set of predetermined classes. Each tuple is assumed to belong to a predefined class as determined by the class label attributes. The set of tuples used for model construction as training set. The model is represented as classification rules or mathematical formulae. The model used for classifying future or unknown malware. The known malware are compared with the classified result from the model. Test set is independent of training set otherwise over fitting will occur.

2.3.2 Classification Prediction Separated Model (CPS Model)

In this model, two ANNs are used to solve the classification and prediction problem. The first ANN produces two outputs: malware behavior class, and normal behavior class. The result is produced to any class the traffic belongs. Second ANN is a Prediction model, which produce continuous valued functions i.e predicts unknown values or missing values, regression analysis used for prediction. Predict data values or construct generalized linear models based on database data. One can only predict the value ranges or category distributions. To solve prediction problem, this ANN produces one output: percentage of infection in the network.

2.4 Artificial Immune System (AIS) Module

Artificial Immune System is a self-adaptive method for malware detection. However, the scalability and coverage problems reduce the detection efficiency of an Artificial Immune System. In order to solve these problems, the authors proposed a model called Collaborative Artificial Immune System, independent immune bodies in different computers were organized by a virtual structure called Immune Collaborative Body. Immune bodies could share detectors with each other, in order to improve the detection efficiency. A collaborative module was added in every immune body for communication and coordination.

Artificial Immune System (AIS) is a framework based upon a set of general-purpose algorithms and models to create abstract components of the immune system. The diversity and self-adaptive characteristics of AIS make it remarkable in anomaly detection. Especially, it has the ability to detect unknown intrusions. As a new approach in Computational Intelligence, AIS has some weak points, the most typical two are scalability and coverage. The problems lead to low efficiency and high false negative. In order to improve coverage rate, a huge number of lymphocytes is needed, and this will cause intolerable time cost. At the same time, the speedy development of computer network makes the spread of malicious software (malware) much faster. Vulnerabilities in a certain kind of software make the intrusion of malware easy. The weak points of AIS prevent it from presenting efficient protection. On the other hand, if the Artificial Immune Systems in computers with similar environments share their lymphocytes, the conflict between efficiency of AIS and spread speed of malware will be ameliorated. Such a model for lymphocytes sharing is called Collaborative Artificial Immune System is implemented by the structure called Immune Collaborative Body. In order to keep the diversity of immune system, Immune Bodies can join an Immune Collaborative Bodies freely, and only some efficient memory lymphocytes can be shared. Immune Collaborative Body is an in compact Coupling which is organized by a set of similar computers without a center node. The self-adaptive characteristic of AIS offers it the ability of detecting unknown malware.

The classical arithmetic in AI is Negative Selection Algorithm (NSA). The principle of NSA is that, normal states are defined as self, and self is encoded according to some formats such as binary string or real vector. Based

on the self code, immature detectors are generated randomly or semi randomly. The immature detectors are trained by known self set, those who match self are deleted and the remaining are mature detectors which are able to detect unknown non-self is not only an unknown abnormity. However non-self is not only unknown but also borderless, the mature detectors can only cover a small area of it, and here comes the coverage problem. In order to reach the level of practical applications, a huge number of detectors are needed. Training these detectors needs long time, and this is the scalability problem.

In fixed detection rate, there is an exponential relationship between the amount of candidate detectors and the amount of protected selves. In order to use the AIS in real environment the scalability problem must be solved. At present, the general method is to change the coding and generation of detectors so as to improve training efficiency, such as dynamic clonal selection algorithm improved r-chunk matching algorithm and negative selection mutation. However, the evolution ability of a single artificial system is far from the requirement. In order to reduce evolutionary time and increase resistance of the population rapidly, antibody is used in social anti epidemic system. if someone has certain resistance, these ability can be spread to other individuals via anybody. Social anti-epidemic system can improve group defense ability rapidly. Considering the similarity between human being and network inspired from the social anti epidemic system of human being building a collaborative artificial immune system is a feasible way.

2.4.1 Coverage Problem

Another problem of AIS is coverage problem. Possible threats and intrusions are unknown and variation, detectors trained by NSA can cover only a small part of non-self. The direct result of low coverage is high false negative, that is to say a large number of unknown intrusion cannot be detected in time. The limitations of NSA is the expression of binary string. To verify this argument, presented two dimensional real values to encode detector. This methods improved individual detection rate.

The diversity of AIS makes different immune body different detection ability. In order to share those differences and improve the coverage and scalability of AIS. This idea is inspired from the social anti-epidemic system.

The main function of ICB is to share efficient detectors in a certain range. This is based on the typical phenomenal, that a certain kind of malware always intrude the similar computer system for they might have the same vulnerability. Share detectors will reduce the training time of individual immune system, and improve coverage for epidemical malware rapidly. Furthermore, the algorithm of collaboration is expressed as pseudocode, including the phase of join collaborative body, collaboration, and quit collaborative body.

2.5 Response Module

This module is responsible for applying the action recommended by the administrator. It can be designed to take automatic action. Its objective is to reconfigure the packet firewall to block traffic on the suspected port(s) that is used in malware propagation. Administrator can then take an appropriate action based on the company/ organization security policy. Using this system, administrator knows if the monitored network is infected or not, and in case of infection, the percentage of infection is known.

The module includes a user interface based on multimedia for supporting network administrator's operations, and a knowledge base for maintaining trustworthiness as systems change and adapt. This knowledge base must be adaptive and shared via web. The validation of the computer generated decisions can be performed by comparing with the decision of experts. Automated methods for knowledge discovery allows building knowledge base for decision making and taking actions using human experience and judgment.

2.6 Filtering Module

This is the stage where fine tunings is done, based on the previous usage and detected intrusions. This helps in reducing false positive levels and to have more security tools, that help with the refining stage by actually making

sure that an alert is valid by checking whether vulnerable to the attack or not. Rule based detection, even known as signature detection, pattern matching and misuse detection.

3. Proposed Algorithm

The algorithm creates and updates a malware signature database. The database is updated everytime a new malware signature found or an existing signature is found in clean program. The algorithm can be described below in Algorithm 1.

```
Input : A Collection of Files
Output : Malware and Clean Signature Databases
Read a malware file;
Run ANN-AIS assoc analysis on the file with 0% support;
Output the generated rules to create the malware signature database;
Read a clean file;
Run ANN-AIS assoc analysis on the file with 0% support;
Output the generated rules to create the clean signature database;
for each file do
Read the file;
Run ANN_AIS assoc analysis on the file with 0% support;
Output the generated rules;
Search the malware signature database for the generated rules;
Search the clean signature database for the generated rules;
        if True Positive or True Negative then
  Goto next file;
      end
      else if False Positive then
         if Subject File is a Malware then
  Remove the matching signatures from the malware signature database;
         end
         else if Subject File is a Clean File then
      Remove the matching signatures from the clean signature database;
         end
      end
      else if False Negative then
if Subject File is a Malware then
           Add the new signatures to the malware signature database;
      end
else if Subject File is a Clean File then

Add the new signatures to the clean signature database;

end
end
end
```

Algorithm 1

Malware	Found	Verdict
0	0	FN
0	1	TP
1	0	TN
1	1	FP

Table 3. Truth Table for Algorithm–Clean Files

Malware	Found	Verdict
0	0	TN
0	1	FP
1	0	FN
1	1	TP

Table 4. Truth Table for Algorithm-Malware

The main update is the step where the algorithm searches the signature database for the generated rules. Various scenarios arises from this situation described as truth table given in Tables 3, 4 and 5.

The scenarios described in the Table 5, are explained below,

3.1 Scenario 1

The file being processed as a clean file and no matching signature is found in the clean signature database and no matching signature is found in the malware signature database. This is defined as a False Negative Clean (FNC) and a True Negative Malware (TNM) situation.

3.2 Scenario 2

The file being processed as a clean file and no matching signature is found in the clean signature database and matching signatures are found in the malware signature database. This is a False Negative Clean (FNC) and a False Positive Malware (FPM) situation.

3.3 Scenario 3

The file being processed as a clean file and matching signatures are found in the clean signature database and no matching signature is found in the malware signature database. This is a True Positive Clean (TPC) and a True Negative Malware (TNM) situation.

3.4 Scenario 4

The file being processed as a clean file and matching signatures are found in the clean signature database and matching signatures are found in the malware signature database. This is a True Positive Clean (TPC) and a False Positive Malware (FPM) situation.

3.5 Scenario 5

The file being processed as a malware file and no

Malware	Found in clean	Found in Malware	Clean	Malware
0	0	0	FNC	TNM
0	0	1	FNC	TNM
0	1	0	FNC	TNM
0	1	1	FNC	TNM
1	0	0	FNC	TNM
1	0	1	FNC	TNM
1	1	0	FNC	TNM
1	1	1	FNC	TNM

Table 5. Truth Table for Algorithm-Malware and Clean Programs

matching signature is found in the clean signature database and no matching signature is found in the malware signature database. This is a True Negative Clean (TNC) and a False Negative Malware (FNM) situation.

3.6 Scenario 6

The file being processed as a malware file and no matching signature is found in the clean signature database and matching signatures are found in the malware signature database. This is a True Negative Clean (TNC) and a True Positive Malware (TPM) situation.

3.7 Scenario 7

The file being processed as a malware file and matching signatures are found in the clean signature database and no matching signature is found in the malware signature database. This is a False Positive Clean (FPC) and a False Negative Malware (FNM) situation.

3.8 Scenario 8

The file being processed as a malware file and matching signatures are found in the clean signature database and matching signatures are found in the malware signature database. This is a False Positive Clean (FPC) and a True Positive Malware (TPM) situation.

This algorithm remedied the problem of basic standard algorithm (not able to stand the test for new and unknown malwares and a high false positive rate of around 30%). This indicates that, the flaw in the filtering method as only those signatures were filtered out, that were found in the training data. A high number of signatures were found in the clean programs in the test dataset. Assigning the final class outcome based upon the majority vote for malware and clean signatures in a file, decreased the false positive rate significantly.

Table 6 shows the Experimental results for new and unknown malwares using automatically extracted signatures by applying ANN-AIS signature in the entire program collection. Besides count of the signatures, these scores also to reach a final class verdict. The last measure is a combination of scores and counts.

Method	Detection Rate	Accuracy
Count	73.1%	85.9%
Score	85.9%	89.6%
Combined	86.5%	92.5%

Table 6. Experimental Results for New and Unknown Malware using Automatically Extracted Signatures by Applying Association Mining.

The basic algorithm was not able to stand the test for new and unknown malwares and gave a high false positive rate of around 30%. This indicated the flaw in the filtering method as only those signatures were filtered out, that were found in the training data. A high number of signatures were found in the clean programs in the test dataset. The modified algorithm remedied this problem.

Assigning the final class outcome based upon the majority vote for malware and clean signatures in a file decreased the false positive rate significantly. Besides counts of the signatures, the signature databases also carried the scores for each signature that described the probability of finding.

Conclusion

In this paper, the architecture of newly constructed Intelligent Intrusion Detection and Prevention System has been described. The design issues of IIDPS have been discussed. The functions of various modules in the IIDPS have been explained. Besides, the algorithm used in ANN-AIS module for capturing unknown malware signatures has briefed.

References

[1]. Meystel, A.M. &Albus, J.M., (2002). *Intelligent Systems Architecture, Design, and Control*. New York, New York, John Wiley & Sons, Inc.

[2]. Hentea, M. (1997). "Architecture and design issues in a hybrid knowledge-based expert system for intelligent quality control". PhD Thesis, Illinois Institute of Technology, Chicago, Illinois.

[3]. Hentea, M. (1999). "Intelligent approach for network management system: Architecture and design issues for ATM computer networks". *Proceedings of 1999 Advanced Simulation Technologies Conference*, San Diego, California.

[4]. Hentea, M. (2003). "Intelligent model for cyber attack detection and prevention". *Proceedings of the ISCA 12th International Conference Intelligent and Adaptive*

Systems and Software Engineering, San Francisco, California, pp. 5-10.

[5]. Hentea, M. (2004). "Data mining descriptive model for intrusion detection systems". *Proceedings of the 2004 Information Resources Management Association International Conference*, New Orleans, Louisiana, pp. 1118-1119.

[6]. Hentea, M. (2005a). "Improving intrusion awareness with a neural network classifier". *Proceedings of the ISCA 14th International Conference Intelligent and Adaptive Systems and Software Engineering*, Toronto, Canada, 163-168

[7]. Hentea, M. (2005b). "Use of reconnaissance patterns for intelligent monitoring model". *Proceedings of the 2005 Information Resources Management Association International Conference*, San Diego, California, pp. 160-163.

[8]. Hentea, M. (2006). "Enhancing information security risk management with a fuzzy model". *Proceedings of 19th International Conference on Computer Applications in Industry and Engineering*, Las Vegas, Nevada, pp. 32-139.

[9]. Wang, W. (2005). *The Intelligent Proactive Information Assurance and Security Technology*. Retrieved on January 5, 2005.

[10]. Wang, J., et al., (2007). "Internet Worm Early Detection and Response Mechanism". *The Journal of China Universities of Posts and Telecommunications*, Vol. 14, No. 3,

ABOUT THE AUTHORS

Dr. S. Murugan is working as a Freelance Content Writer and Adjunct Faculty. He has completed his PhD from School of Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu, India. He received BSc in Physics from Madurai Kamaraj University, Madurai, and MCA Degree from School of Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu., India in 1992 and MPhil (CS) from Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India in 2002. He has 24 years of teaching and admin experience at UG & PG level in the field of Computer Science. He has published 10 papers in the National Level and 30 papers in International Level. Key research interests include Software Agents, Data Mining and Knowledge Management, Big data, Autonomic Computing, Data Mining, Human Resource Management, Education Intelligence Network Security Algorithms, Malware Prevention and Detection Mechanism and Algorithm, HCI and Cryptography. He has published various Books and Courseware in the field of Computer Science and Engineering.

Dr. K. Kuppusamy is currently working as a Professor and Chair in the Department of Computer Science and Engineering, Alagappa University, Karaikukdi, Tamilnadu, India. He has received his Ph.D in Computer Science and Engineering from Alagappa University, Karaikudi, Tamilnadu in the year 2007. He has 30 years of teaching experience at PG level in the field of Computer Science. He has published many papers in the International and National Journals and presented many papers in the International and National Conferences. His areas of research interests include Information/Network Security, Algorithms, Neural Networks, Software Engineering and Testing and Optimization Techniques, Big Data Analytics, Data Mining ,Text mining, Web Mining, Grid Computing, Open Source Code, and also Human Resource Management Education. He has published various Books and Courseware in the field of Computer Science and Engineering.