International Journal of Embedded Systems and Emerging Technologies
Vol. 1: Issue 1
www.journalspub.com

Functioning of Intelligent Intrusion Detection and Prevention System (IIDPS)

S. Murugan*
Department of Computer Science and Engineering, Alagapa University, Karaikudi (Tamil Nadu), India

Abstract

In this paper, the architecture of Intelligent Intrusion Detection and Prevention System is proposed. The intelligence element was introduced using Artificial intelligence techniques. The development of IIDPS was done as a protection from cyber-attacks. The TSAM is decision maker agent and PMM is the main controller agent. Although, the proposed system is a combination of different types of intelligent agents, hybrid architecture under real time constraints.

Keywords*: Intelligent Intrusion Detection and Prevention System, TSAM, PMM*

****Author for Correspondence****: Email ID: murugan.sethu@gmail.com*

INTRODUCTION

Artificial Intelligence (AI) can be used for construction of intelligent models to enhance the information security management, incursion detection and prevention capabilities, efficiency of security event management, and decision making. Intelligent systems also known as intelligent assistants assist users in decision making process for forming and monitoring specific metrics, faults and events association that which could help in investigation of the attack and prevention from 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[1-4].

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 will combine AI techniques with other techniques such as conventional programs, statistical packages and object based and rule based data mining creating hybrid intelligent system architecture. 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[2-6]. 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-resolution hierarchy of computational loops yields deep insights into the phenomena of behavior, perception, cognition, problem solving, and learning.

DESIGN ISSUES

A noteworthy choice to be made amid the structural plan is the thing that operators ought to be incorporated. 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. Roles in some methodologies are things the agents that will perform by looking at 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 real time constraints. Intelligent agents help 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 [5-8]. The outline and programming of specialists ought to be centered on boosting their execution measure which encapsulates the rule for accomplishment of an operator's conduct. 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[6-9].

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 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 is the mobility which is the degree to which the agents travel through the network [7-10].

The input data to the models for learning and outputs of the models plays 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 happens as the operator watches its collaborations with the earth and its own basic leadership forms. 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[9-10].

The classification, association and prediction of upcoming cyber-attacks is supported by Artificial Neural Networks (ANN) through learning, adapting from past, current data and events.

Fuzzy logic permits dispensation of qualitative variables and imprecise reasoning when the suggestions are not exact and abrupt. A possible avenue for integrating data mining, neural networks, and fuzzy expert systems 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 then 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-based incorporates knowledge for the security domain such as raw data and events, performance measures, patterns, policies, and decisions.

ARCHITECTURE OF IIDPS

The following Figure 1 shows the architecture of the proposed Intelligent 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.

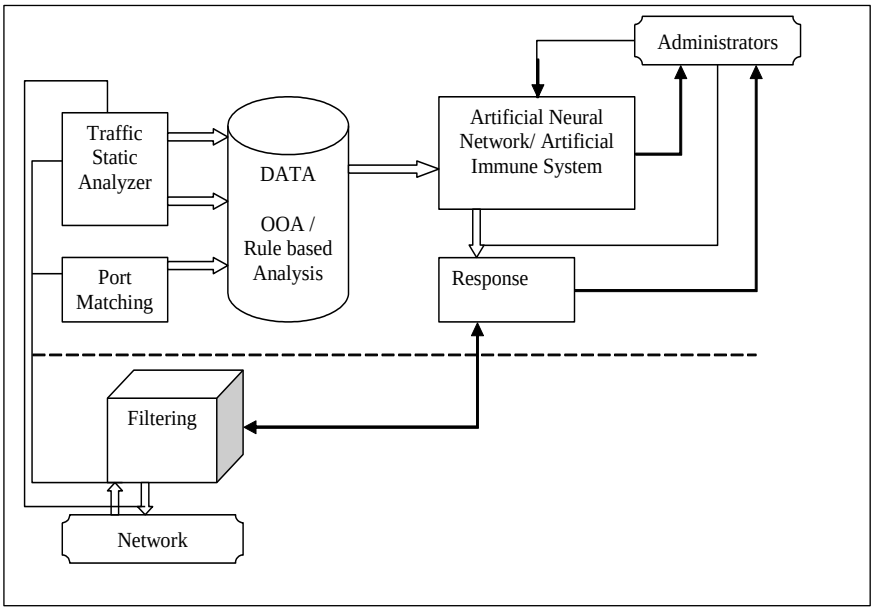

Fig. 1: *Architecture of IIDPS.*

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:

1. Traffic Statistical Analyzer Module
2. Port Matching Module
3. Artificial Neural Network Module
4. Filtering Module
5. 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.

Traffic Statistical Analyzer Module (TSAM)

This module is taking care for calculating statistical values based on the analysis of incoming and outgoing traffic while 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, 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 number of packets and number of packets per port that are used as 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, 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 need flexible tools that rapidly investigate several traffic properties from looking into headers and payload, that reporting concise information on hosts, flows etc. From this investigation it is conceivable to pick on which viewpoint to center the portrayal and to consider methodologies for knowledge follow refinement to expel false information.

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

1. The reports are analyzed with flows, immediately locate unknown malware behaviors looking for flows with more than one packet;

2. 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 compare 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 has previously been implemented for the traffic generated by legitimate applications. Independent from application-level protocol, it can be likewise implemented to different kinds of traffic. Moreover, symbolizing statistical properties of traffic at packet-level assists in construction of analytical and empirical models can be utilized for traffic generation and simulation, which represent another way to better consider the influence 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 to traffic profiling for 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.

1. The selection of potential targets
2. Random or deterministic scanning
3. Preference for the local subnet
4. The generation of scanning traffic
5. Protocol on the transport or network layer
6. Port number

7. 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.

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 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.

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.

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.

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 1: *Scanning Traffic.*

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 2 shows the Malware Packet Propagation with various protocols and ports connections to identify the known malware. Protocol column gives the name of network protocol used by a malware 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 therefore try to connect to different ports. The Connection column in the table regards to the quantity of packets, through which 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.

Table 2: *Malware Packet Propagation.*

Known Malware	Protocol	Port	Connection
Sasser B	TCP	445	128 in parallel
Welchia A	ICMP TCP	- 80 &135	As many packets as possible n/a
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(DNS MX) TCP (SMTP)	53 25	n/a n/a

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 real way to know whether a bundle source is casualty or slave aggressor, every record is being analyzed as though it is from the casualty or from slave assailant. 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.

The following Figure 2 shows the working of Port Matching Module.

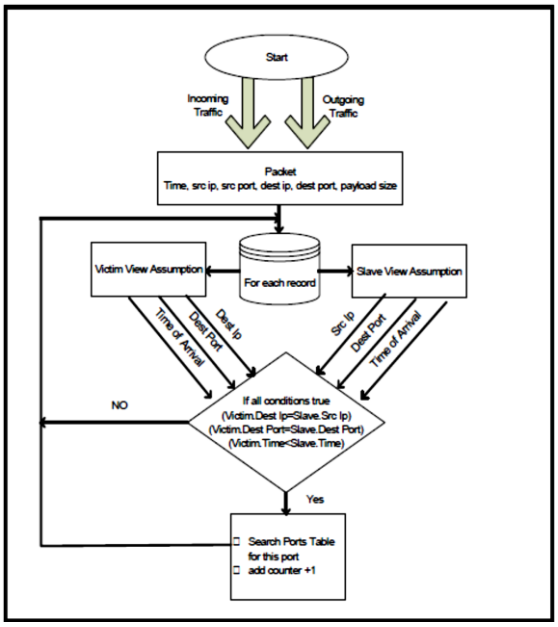

Fig. 2: *Port Matching Module.*

Artificial Neural Network Module (ANNM)

An artificial neural net is an electrical analogue of a biological neural net the cell body in an artificial neural net is modeled by a linear activation function. The

activation function, in general, attempts to enhance the signal contribution received through different dendrons. The action is assumed to be signal conduction through resistive devices. The synapse in the artificial neural net is modeled by a non-linear inhibiting function, for limiting the amplitude of the signal processed at cell body. The most common non-linear functions used for synaptic inhibition are Sigmoid function, Signum function, Step function.

Sigmoid and tan hyperbolic (tanh) functions are grouped under soft nonlinearity, whereas signum and step functions are under hard type non-linearity. Artificial neural nets have been successfully used for recognizing objects from their feature patterns. For classification of patterns, the neural networks should be trained prior to the phase of recognition process. The process of training a neural net can be broadly classified into Supervised learning, Unsupervised learning, Reinforcement learning. In this work, the Back Propagation Network (BPN) training algorithm has been used to train the ANN.

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 name Classification, Prediction Combined (CPC) model and Classification, Prediction Separated (CPS) models are designed using ANN to classify and predict the behavior class of incoming malware.

In CPC, one ANN is used to produce results by combining classification and prediction of malware behavior classes. In CPS, two ANNs are used to produce results separately one for classification and another for prediction of malware behavior classes. The description of each model follows.

Classification Prediction Combined Model (CPC Model)

In this model, as shown in Figure 3, 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. 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 overfitting will occur.

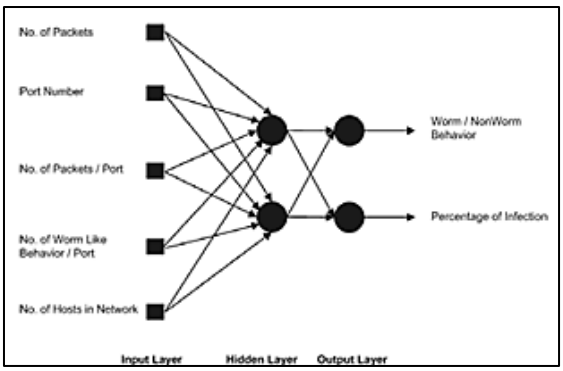

Fig. 3: *Neural Network.*

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 producing 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 value ranges or category distributions. To solve prediction problem this ANN produces one output: percentage of infection in the network.

AIS Module

Artificial Immune System is a self-adjusting technique for malware recognition. Still, the scalability and coverage problems reduce the recognition efficiency of an Artificial Immune System. Keeping in mind the end goal to comprehend these challenges, Collaborative Artificial Immune System, autonomous safe bodies in various PCs were set up by a mimicked structure called Immune Collaborative Body. Invulnerable bodies could impart indicators to each other, keeping in mind the end goal to enhance the discovery proficiency. A collaborative module was added in every immune body for communication and coordination.

AIS are outline constructed based on a set of general-purpose algorithms and models to generate nonfigurative constituents of the immune system. The diversity and self-adaptive characteristics of AIS make it extraordinary in irregularity detection. Especially it has the ability to detect unknown intrusions. As another approach in Computational Intelligence, AIS has some frail focuses, the most common two are adaptability and scope. The issues prompt to low effectiveness and high false negative. Keeping in mind the end goal to enhance scope rate, countless is required, and this will bring about unfortunate time cost. In the meantime the quick advancement of PC system makes the spread of malevolent programming (malware) much speedier. Vulnerabilities in a specific sort of programming make the interruption of malware simple. The feeble purposes of AIS keep it from exhibiting effective security. Then again, if the Artificial Immune Systems in PCs with comparable situations share their lymphocytes, the contention between proficiency of AIS and spread speed of malware will be enhanced. 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 incompact 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 remained are mature detectors which are able to detect unknown non self is not only 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 a 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.

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.

Response Module

This module is in charge of applying the activity prescribed by the overseer. It can be intended to make programmed move. Its goal is to reconfigure the bundle firewall to piece activity on the suspected port(s) that is utilized as a part of malware proliferation. 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.

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.

Filtering Module

This is the phase where fine tunings is done, in light of the past utilization and distinguished interruptions. 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.

PROPOSED ALGORITHM

The algorithm creates and updates a malware signature database. The database is updated every time a new malware

signature found or an existing signature is found in clean program.

The algorithm can be described as:

```
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
Go to 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
```

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

Table 3: *Truth Table for Algorithm-Clean Files.*

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

Table 4*: Truth Table for Algorithm-Malware.*

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

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

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

The scenarios described in the Table 4, 5, are explained below.

Scenario 1: The record being prepared as a perfect document and no coordinating mark is found in the spotless mark database and no coordinating mark is found in the malware signature database. This is defined as a false negative clean (FNC) and a *true* negative malware (TNM) situation.

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.

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 as a *True Positive Clean* (TPC) and a *true negative malware* (TNM) situation.

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 as a *True Positive Clean* (TPC) and a *False Positive Malware* (FPM) situation.

Scenario 5: The file being processed as a malware file and no matching signature is found in the clean signature database and no matching signature is found in the malware signature database. This as a *True Negative Clean* (TNC) and a *False Negative Malware* (FNM) situation.

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 as a *True Negative Clean* (TNC) and a *True Positive Malware* (TPM) situation.

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 as a *False Positive Clean* (FPC) and a *False Negative Malware* (FNM) situation.

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 as 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 flaws 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 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 combination of scores and counts.

Table 6: *Experimental Results For New And Unknown Malware Using Automatically Extracted Signatures By Applying Association Mining.*

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

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 flaws 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. Other than numbers of the marks, the mark databases additionally conveyed the scores for every mark that depicted the likelihood of finding. Table 5 Experimental results for new and unknown malwares using automatically extracted signatures by applying sequential association mining signature in the entire program collection. Besides count of the signatures, used these scores also to reach a final class verdict. The last measure that used was a combination of scores and counts.

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. Zolkipli M.F., Jantan A. An Approach for Malware Behavior Identification and Classification. *Proceeding of 3rd International Conference on Computer Research and Development*. Shanghai: 2011 March 11-13; 191–4p.
2. Chou T.S., Yen K.K. Fuzzy Belief k-Nearest Neighbors Anomaly Detection of User to Root and Remote to Local Attacks. *The 2007 IEEE Workshop on Information Assurance*. United States Military Academy, West Point, NY. 207–13p.
3. Wang J. Internet Worm Early Detection and Response Mechanism. *The Journal of China Universities of Posts and Telecommunications*. 2007:14(3).
4. Wagner D., Soto P. Mimicry attacks on host-based intrusion detection systems. *In Proceedings of the 9th ACM Conference on Computer and communications Security, ACM.* New York: 2002; 255–64p.
5. Tian R., Batten L., Versteeg S. Function Length as a Tool for Malware Classification. *Proceedings of the 3rd International Conference on Malicious and Unwanted Software.* Fairfax: 2008 October 7-8; 57–64p.
6. Petroni, Hicks M., Automated detection of persistent kernel and control-flow attacks. *In Proceedings of the 14th ACM Conference on Computer and Communications Security*. *ACM*. 2007.
7. Nataraj L., Yegneswaran V., Porras P. *et al*. A Comparative Assessment of Malware Classification Using Binary Texture Analysis and Dynamic Analysis. *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence.* 2011: 21–30p.
8. Rieck K., Trinius P., Willems C. *et al.* Automatic Analysis of Malware Behavior Using Machine Learning. *Journal of Computer Security*. 2011: 19; 639–68p.
9. Park Y., Reeves D., Mulukutla V. *et al.* Fast Malware Classification by Automated Behavioral Graph Matching. *Proceedings of the 6th Annual Workshop on Cyber Security and Information Intelligence Research.* 2010: 45.
10. Martignoni L., Christodorescu M., Jha S. Fast, generic, and safe unpacking of malware. *In Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC)*. 2007: 431–41p.