Intelligence Intrusion Detection Prevention Systems using Object Oriented Analysis method

S.MURUGAN MCA.,MPHIL.,CGT.,MISTE.,(MS).,
ACTS,CDAC KNOWLEDGE PARK, NO 1 OLD MADRAS ROAD,BANGALORE

DR.K.KUPPUSAMY,DIRECTOR I/C
COMPUTER CENTRE,ALAGAPPA UNIVERSITY,KARAIKUDI

Abstract

This paper is deliberate to provide a model for "Intelligence Intrusion Detection Prevention Systems using Object Oriented Analysis method ", It describes the state's overall requirements regarding the acquisition and implementation of intrusion prevention and detection systems with intelligence (IIPS/IIDS). This is designed to provide a deeper understanding of intrusion prevention and detection principles with intelligence may be responsible for acquiring, implementing or monitoring such systems in understanding the technology and strategies available.

With the need for evolution, if not revolution, of current network architectures and the Internet, autonomous and spontaneous management will be a key feature of future networks and information systems. In this context, security is an essential property. It must be thought at the early stage of conception of these systems and designed to be also autonomous and spontaneous.

Future networks and systems must be able to automatically configure themselves with respect to their security policies. The security policy specification must be dynamic and adapt itself to the changing environment. Those networks and systems should interoperate securely when their respective security policies are heterogeneous and possibly conflicting. They must be able to autonomously evaluate the impact of an intrusion in order to spontaneously select the appropriate and relevant response when a given intrusion is detected.

Autonomous and spontaneous security is a major requirement of future networks and systems. Of course, it is crucial to address this issue in different wireless and mobile technologies available today such as RFID,Wifi, Wimax, 3G, etc. Other technologies such as ad hoc and sensor networks, which introduce new type of services, also share similar requirements for an autonomous and spontaneous management of security.

Intelligence Intrusion Prevention Systems (IIPS) are designed to aid in preventing the compromise of information systems and thus help preserve the basic triad of all security, confidentiality, Integrity and availability (CIA), not only of information but the infrastructures that store and transmit it as well.

Intelligence Intrusion detection systems (IDS) refer to any technology or strategy that allows us to detect the attempted compromise of our systems and information, and as before, preserve the CIA of the information and infrastructures.

In many cases these two systems work together and with the networking infrastructure to do their jobs. As IIPS/IIDS technology has improved over the last few years, prevention and detection have been consolidated into one network device, or as it is commonly referred to, one "appliance." In other cases the IPS is a separate technology, usually a software package or "agent" that runs on a desktop or host to detect attempted compromise.

Keywords: IIDPS using OODA; Intelligence Intrusion Detection Prevention ; Unknown Malware attack Prevention by using OODA.

1.Introduction:

One of the biggest challenges in the network intrusion detection field is the limitation imposed by the use of well-known attack signatures that disable the previous detection of new attacks. This work presents a packet analysis methodology for detecting anomalous behaviors, based on attack signatures, but on verifying whether the network protocols are being violated, and on the content of the respective headers. The biggest benefit of this methodology is the possibility of detecting anomalies or inadequate behaviors that can correspond, totally or partially, to variations on well-known and unknown attacks.

Due to the unknown network attacks are hardly be detected and the early warning and response mechanism cannot be established, many of intrusion detection prevention systems (IDPS) are only effective in detecting known Network attacks and cannot evaluate the risk of Network service. In order to conquer

these limitations and inspired by intelligence principles, this presents an intelligence based active defense model for Network attacks which is on the basis of mathematical or artificial intelligence techniques. The risk of Network attacks are quantitatively analyzed on the relationship between the antibody concentration .Proposed Theoretical analysis and experimental evaluation demonstrate that the model is more suitable for detecting unknown attacks, and provides an active defense mechanism for detecting network anomalies also.

In response to the threat posed by malicious code and by potential system and network intruders, the information assurance community continues to work on countervailing techniques and tools. Designers must test their detection algorithms under realistic conditions; however, traffic collected from real networks may contain any number of unknown attacks whose presence may compromise test results. Ideally researchers and developers would be able to introduce specific attacks into an isolated test-bed network that includes realistic attack-free background traffic. We present such a system for generating synthetic network traffic based on models of behavior observed in real networks.

2.Multi-Method Attack Detection Review:

IDP Feature Brief Accurate Attack Protection Today's complex attacks manifest themselves differently in different customer environments. Networks IDP's advanced attack protection and customization helps accurately detect attacks and drop them from the network to prevent any damage.

Networks IDP's advanced attack protection combines the following features:· **Multiple detection methods** that include compound signatures, **Stateful signatures**, protocol anomaly and backdoor detection. · An open signature format that allows administrators to view how the attack string matches the attack signature and edit the signature as needed. This level of understanding and customization allows administrators to tailor the attack signature to address unique attack requirements.· Extensive signature customization improves the ability to detect unique attacks by providing an additional level of control needed to tailor the signature specific requirements.· Compound signatures: Improves speed of detection by proving the ability to combine Stateful signatures and protocol anomalies into a single attack object to detect complex attacks within a single session. · Over 400 customization parameters and Perl style regular expressions: Provides the ability to tailor signatures or create completely custom signatures .

Multi-Method Attack Detection Networks' Multi-Method Detection (MMD™) combines multiple detection mechanisms in a single product for comprehensive coverage. Because different types of attacks require different methods to identify them, products using only a few detection methods are incapable of detecting all attacks. Networks IDP's Multi-Method Detection maximizes the types of attacks detected, ensuring critical threats do not go undetected. The detection methods in MMD include: Mechanism Description Stateful Signatures Detect known attack patterns only in relevant traffic. Protocol Anomaly Detect unknown or permutated attacks. Backdoor Detection Detect unauthorized interactive backdoor traffic. Traffic Anomaly Detect attacks spanning multiple sessions and connections. Network Honeypot Detect attackers that are impersonating network resources and tracking attacks against them.Layer-2 Detection Detect layer-2 (ARP) attacks. DOS Detection Detect certain Denial of Service attacks. Spoofing Detection Detect IP spoofing attacks. Compound Signatures Combines stateful signatures and protocol anomalies to detect complex attacks in a single session. Stateful Signature Detection There are certain attacks that can be identified using an attack signature, is the pattern of the attack that can be found in the network traffic. Stateful Signatures were developed to significantly increase detection performance and reduce the false alarms associated with signature-based intrusion detection systems currently on the market. Networks IDP tracks the state of a connection and looks for attack patterns in only the relevant portions of the traffic where these attacks can be perpetrated. Traditional signature-based intrusion detection systems look for attack patterns arbitrarily in the traffic stream, resulting in higher rates of false alarms.

IDP Feature Brief For example, to determine if someone is attempting to login to a server as a root user, a traditional signature-based IDS would send an alarm any time the word "root" appears in the transmission, generating false alarms. Networks IDP,using Stateful Signature Detection, would only look

for the string "root" in the login sequence, which is a way to accurately detect the attack. All of Networks IDP's signatures are accessible via a simple, graphical user interface, making it easy to understand what the system is looking for in traffic. Additionally, an open signature format and signature editor provides the ability to quickly modify or add signatures. Both of these capabilities allow users to determine what is important to their environments and make sure the system identifies it.

Let's take a look at the Send mail Wiz Attack, which tries to gain root access to a SMTP server. To perpetrate, an attacker initiates an SMTP connection and sends "wiz" during the control connection. An intrusion detection system needs to identify the "wiz" pattern to detect the attack. Most IDSes look for attacks using packet-signature detection, which means they look for a pattern match in each and every packet, without regard to the state of the communication. As shown in the diagram above, a packet signature would look for the "wiz" pattern in the 3-way handshake, the command mode, the data transmission mode, and the 4-way close. Because the packet signature looks for the attack in all of this unrelated traffic, it wastes resources processing unnecessary information and generates lots of false alarms. This is because SMTP data transmissions are random and use a base64 encoding mechanism that results in the manifestation of the "wiz" pattern at least once in every 32,000 characters.

A few IDS products have tried to improve their packet signature detection implementation by looking for patterns at fixed offsets within each packet. This method is sometimes referred to as context-based signature detection. Systems that use this method still do not understand the communication flows and continue to generate false alarms, as seen in the diagram, by either picking up the pattern across all packets or missing attacks because the attack has permutated the attack pattern. However, Stateful Signatures track and understand the state of the communication and, therefore, narrow down the pattern matching to the exact location (communication mode and flow direction - meaning client to server or server to client traffic flow) where the attack can be perpetrated. As seen in the diagram, Stateful Signatures only perform signature pattern matching on relevant traffic where an attack can be perpetrated. As a result, performance is greatly improved and the occurrence of false positives significantly reduced.

IDP Feature Brief Protocol Anomaly Detection Attackers are constantly evolving, launching new or sophisticated attacks that don't follow a pattern. Protocol anomaly detection can be used to identify the attacks that deviate from the protocols that "normal" traffic follows. For instance, it would identify attacks that use ambiguous traffic to try to evade detection and compromise a network and/or host. Using a buffer overflow as an example (below), an attacker gains full access of a machine by making the server run the attacker’s code under the server’s permissions.

IDP Feature Brief Protocol Anomaly Detection compares the amount of data allowed by the buffer with the amount of data sent and alarm traffic sent in excess of the allowed amount. Protocol anomaly detection is as effective as the number of protocols it supports. If a protocol is not supported, then attacks exploiting that protocol will go undetected in the network. Networks IDP is the first to support such a broad range of protocols, including SNMP (to protect against more than 60,000 vulnerabilities) and SMB (to protect against Windows-based vulnerabilities running on internal systems). Backdoor Detection Networks was the first product capable of identifying and protecting against backdoor attacks. Backdoor attacks enter a network and allow an attacker to take complete control of a system, often resulting in a loss of data. For example,an attacker can exploit a vulnerability to load a Trojan onto a network resource, and then interact with that system to control it. The attacker continues can then try different commands in an effort to launch attacks from that system or compromise other systems. Networks IDP identifies the unique characteristics of the interactive traffic and sends an alarm for unexpected activity. Example: Trojan Attack Backdoor Detection is the only way to detect Worms and Trojans· Looks for interactive traffic · Detects unauthorized interactive traffic, based on what the administrator defines is allowed · Detects virtually any backdoor, even if the traffic is encrypted and the protocol is unknown

IDP Feature Brief Traffic Anomaly Detection Some attacks are not contained within a single session, rather they span a number of connections. Often these attacks are reconnaissance missions, gathering information on the network for future attacks. Traffic anomaly detection can identify this activity by comparing incoming traffic to "normal" traffic patterns and identifying deviations. This method allows Networks IDP to detect intrusion attempts that span multiple connections, by defining and applying

thresholds and triggers. Network probes and port scans are examples of attacks that can be detected by traffic anomaly detection. In the Reconnaissance Attack example below, an attacker scans for open ports on the network and then returns to exploit any discovered vulnerabilities. While not an attack in and of itself, network and port scans are indicative of someone trying to get the information they need to launch a future attack. If an administrator is aware of network and port scans, then they can be on the look out for future attacks from that IP address. Network Honeypot There are a lot of attackers out there exploring networks to see what they can get away with. A Network Honeypot is a good way to filter out some of the "noise" created by the less sophisticated attackers. By impersonating services that don't exist,the Network Honeypot sends fake information to people scanning the network to try an entice attackers to access the non-existent services. It identifies the attacker when they attempt to connect to the service. There is no reason for legitimate traffic to access these resources because they don't exist; therefore any attempt to access them constitutes an attack. The Network Honeypot impersonates services, sending fake information in response to scans to try and entice attackers to access the non-existent services.

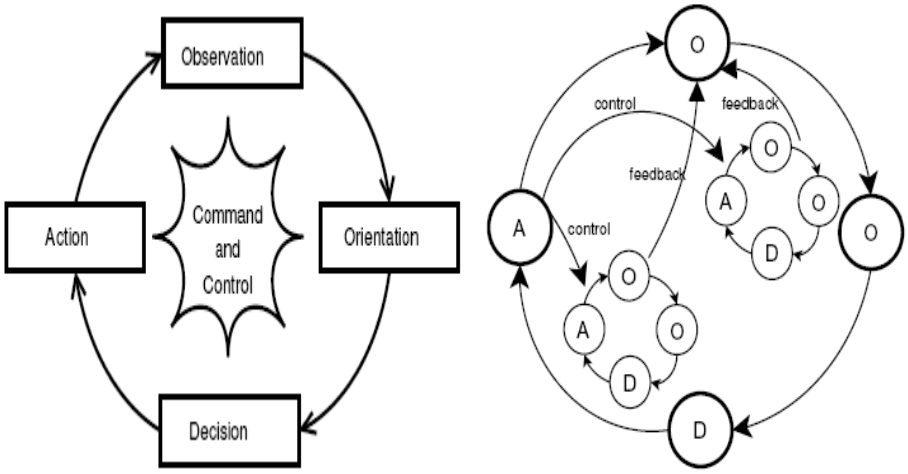
An attack is identified when the attacker returns and tries to access the impersonated resources. There is no reason for legitimate traffic to access these resources because they don't exist, so any attempt to connect constitutes an attack. This is a good way to stop the "noise' created by "script kiddies" and unsophisticated attackers.

	attack models	programming guidelines	pattern languages	evaluation criteria	vulnerability databases
vuln. avoidance:	⊕⊕	⊕	⊕	⊖	⊖
vuln. discovery:	⊕⊕	⊕	⊖⊖	⊕	⊖
sec. evaluation:	⊕	⊖	⊖⊖	⊕⊕	⊕⊕
mistakes:	⊕⊕	⊕⊕	⊕	⊖⊖	⊕
measures:	⊖⊖	⊕	⊕⊕	⊕⊕	⊖⊖
general:	⊕⊕	⊕	⊕⊕	⊕⊕	⊖⊖
system specific:	⊕⊕	⊖	⊖	⊖⊖	⊕⊕
design-level:	⊕	⊖	⊕⊕	⊕⊕	⊖⊖
code-level:	⊕⊕	⊕⊕	⊖	⊖⊖	⊖⊖

3.OODA principles used for finding Unknown attack :

Worms are on the top of malware threats attacking computer system although of the evolution of worms detection techniques. Early detection of unknown worms is still a problem. This paper produce a method for detecting unknown worms based on local victim information. The proposed system uses Artificial Neural Network (ANN) for classifying worm/ nonworm traffic and predicting the percentage of infection in the infected network. This prediction can be used to support decision making process for network administrator to respond quickly to worm propagation in an accurate procedure.

A method of detecting and blocking unknown attack in Network during unpacking of file after the code and data contained in the file are unpacked is described. The method includes inserting a ANN methods into one or more un assessed process running in the Network Host. ANN is then placed one or more calls carried out by the one or more un accessed processes ;the one or more calls determining an optimal time period in which to detect unknown attack in the unassisted processes. During the optimal time period the one or more calls carried out by one or more un-assessed processes or suspended and attributes of one or more un-assesses processes are detected and the likely unknowingness of the one or more unassisted processes is determined from the attributes.



Motivated by OODA principles, we put forth that the analysis of performance should always take into consideration these three aspects:

1) characteristics of the malware or malware attack
2) the particular performance criterion being considered
3) the environment within which they are operating.

Thus, typical statements about malware performance could take one of the forms below:

$$\textit{FixedE, Fixed P} \Rightarrow \textit{P(C1)} \leq \textit{P(C2)} \quad (1)$$
$$\textit{Fixed C, Fixed P} \Rightarrow \textit{P(E1)} \leq \textit{P(E2)} \quad (2)$$
$$\textit{Fixed C} \Rightarrow \textit{P1(E)} = f(\textit{P2(E)}) \quad (3)$$
$$\textit{Fixed E} \Rightarrow \textit{P1(C)} = g(\textit{P2(C)}) \quad (4)$$

Equation 1 describes the situation where, for example, a malware programmer wants to optimize the choice of characteristics (C1 or C2) with respect to the required performance criterion P and when the environment characteristics E are fixed.

Equation 2 describes the variation of a given performance criterion P of a malware (attack) with fixed characteristics C but under different operating environments E1 and E2.

Finally, Equations 3 and 4 represent tradeoffs between two different performance criteria P1 and P2. Equation 3 describes how different performance criteria of a given malware might interact as the operating environment changes; this is useful for malicious attackers trying to understand how different objectives might conflict as the characteristics of the target vary. On the other hand, understanding the trade-offs in Equation 4 might help "defenders" of a particular system

4.For effective prevention must have :

In-line operation - only by operating in-line can an IPS device perform true protection, discarding all suspect packets immediately and blocking the remainder of that flow.

Reliability and availability - should an in-line device fail, it has the potential to close a vital network path and thus, once again, cause a DoS condition.An extremely low failure rate is thus very important in order to maximise uptime, and if the worst should happen, the device should provide the option to fail open or support fail-over to another sensor operating in a fail -over group (see below). In addition, to reduce downtime for signature and protocol coverage updates, an IPS must support the ability to receive these updates without requiring a device re-boot. When operating inline, sensors rebooting across the enterprise effectively translate into network downtime for the duration of the reboot.

Resilience - as mentioned above, the very minimum that an IPS device should offer in the way of High Availability is to fail open in the case of system failure or power loss (some environments may prefer this default condition to b e "fail closed" as with a typical firewall, however - the most flexible products will allow this to be user-configurable). Active -Active stateful fail-over with cooperating in-line sensors in a fail-over group will ensure that the IPS device does not become a single point of failure in a critical network deployment

Low latency - when a device is placed in-line, it is essential that its impact on overall network performance is minimal. Packets should be processed quickly enough such that the overall latency of the device is as close as possible to that offered by a layer 2/3 device such as a switch, and no more than a typical layer 4 device such as a firewall or load-balancer.

High performance - packet processing rates must be at the rated speed of the device under real-life traffic conditions, and the device must meet the stated performance with all signatures enabled. Headroom should be built into the performance capabilities to enable the device to handle any increases in size of signature packs that may oc cur over the next three years. Ideally, the detection engine should be designed in such a way that the number "signatures" (or "checks") loaded does not affect the overall performance of the device.

Unquestionable detection accuracy - it is imperative that the quality of the signatures is beyond question, since false positives can lead to a Denial of Service condition. The user MUST be able to trust that the IDS is blocking only the user selected malicious traffic. New signatures should be made available on a regular basis, and applying them should be quick (applied to all sensors in one operation via a central console) and seamless (no sensor reboot required).

Fine-grained granularity and control - fine grained granularity is required in terms of deciding exactly which malicious traffic is blocked. The ability to specify traffic to be blocked by attack, by policy, or right down to individual host level is vital. In addition, it may be necessary to only alert on suspicious traffic for further analysis and investigation.

Advanced alert handling and forensic analysis capabilities - once the alerts have been raised at the sensor and passed to a central console, someone has to examine them, correlate them where necessary, investigate them, and eventually decide on an action. The capabilities offered by the console in terms of alert viewing (real time and historic) and reporting are key in determining the effectiveness of the IPS product.

We can identifies the unknown attack by using below queries ,it will result us to get more effectively the unknown attack.

1. Are non-worms, like worms, temporally consistent? If so, we must identify properties that distinguish the two.

2. Can we detect processes with similar behavior on multiple hosts? If so, we can detect the outbreak of a worm, the behavior of which across hosts is likely to be similar: bounded by time as are fast-spreading worms by definition, there are only so many ways for them to achieve some effect on a host quickly. A worm's filename and executable, by contrast, can be too easily altered during propagation (as through metamorphosis) and are, thus, less reliable than more dynamic techniques. Presumably, the more "worm-like" a process (i.e., the more temporally consistent), the more likely we are to detect it if running on multiple hosts.

3. Can we avoid mistaking popular non-worms for worms?

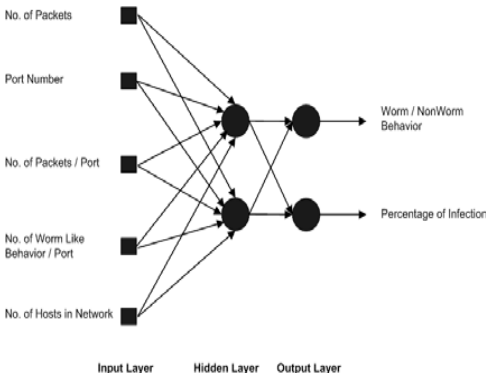
We cannot assume that processes common to many hosts (e.g., explorer.exe) are necessarily worms, lest we infer incorrectly that an attack is in progress. And we should not confuse a non-worm running on one host with a worm running on another, even if behaving similarly, lest we over- state an outbreak's severity.

5.Used ALGORITHM:

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

Classification / Prediction Combined Model (CPC Model)

In this model, shown in below, the idea was to use one ANN to produce the two desired outputs.



CPC Model.

6.Developing ANNM with AIS :

Data are collected using a developed application called (Worm Detection Traffic Analyzer). Input Data includes 5 features elected from many features that help in identifying worm behavior from non-worm network behavior. All of these inputs are numerical values. The normalization of value from 0 to 1 is done by the tool used for building the ANN model which is Neuro Solution.

The Dataset consists of 5430 exemplars: 4230 examplers are used for training (% 78), 1200 examplers are used for testing (% 22). Collected data are gathered from different experiments, each experiment produces many exemplars, and the sequence within the same experiment is matter. When this application is used in the real life application, it means that the sequence of traffic is producing information in the network traffic; data are related in this way. But every experiment is completely separated from other experiments. Then experiments orders are randomized to accomplish different conditions randomly, but within the same experiments result, data can't be randomized.

After testing different topologies, the best result was chosen to produce the model. The Network Training Paradigm is Supervised Learning implemented using Multilayer Perceptron (MLP) which is feed forward neural network that uses Back propagation algorithm. After testing some activation functions, best results are produced by Sigmoid Function. Also Momentum at step size 100 produces the best results as learning rule

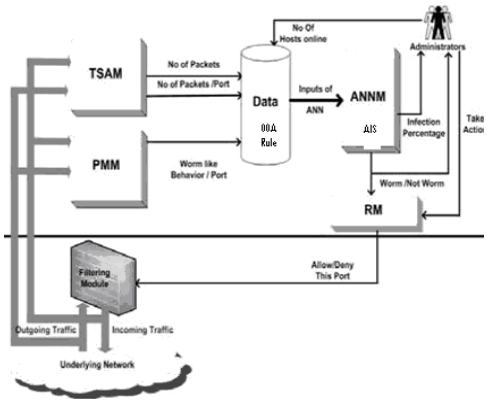
There are two methods for updating weights that can be used: On-Line, Batch learning. After testing both of them, best results are produced by online updating.

The proposed model produces good results in worm detection. The advantage of the ANN method over other techniques is its ability to classify correctly a worm not used in the training. The proposed system produces perfect result with accuracy of %99.96 in detecting the presence of worm in the network even for unknown worms.

An artificial immune network model is connectionist in nature but it follows an evolutionary-like learning algorithm, that is the immune clonal selection principle (Burnet, 1959).

The motivation for the development of the immune network model, named aiNet, was stressed and it was theoretically and empirically compared with artificial neural networks. The theoretical comparisons allow us to conclude that there are many similarities between aiNet and ANNs and, certainly one area has much to gain from the other.

Empirical results demonstrated the aiNet capability of solving non-linearly separable problems through the minimal spanning tree (MST) of a trained network, as far as there is a relative separation between the clusters. Otherwise, alternatives were given such that its performance could be evaluated and compared to other ANNs from the literature. It could be verified that, although it is a self-organizing strategy, its results were comparable to those obtained by one class of supervised learning algorithm for ANNs in classification problems. It is not difficult to speculate that powerful hybrid algorithms, combining immune network models and artificial neural networks, might soon arise. Additionally, we can propose as future extensions of this paper some enhancements and analysis of the aiNet model, like: 1) the development of a general framework to visualize a trained aiNet for network antibodies of length $L \geq 4$; 2) a detailed convergence analysis of the algorithm; and 3) the inclusion of an affinity threshold ε.



The proposed model used for detecting worm behavior using ANN/AIS.
(1) Traffic Statistical Analyzer Module (TSAM).
(2) Port Matching Module (PMM).
(3) Artificial Neural Network Module (ANNM/AIS).
(4).DATA Module OOA Rule mining.
(5) Response Module (RM).

For evaluation purposes, the True Positive Rate (TPR), which is the number of positive instances classified correctly and shown in Eq. (1), is measured, and the False Positive Rate (FPR), which is the number of negative instances misclassified and shown in Eq. (2), is measured. The Total Accuracy measures the number of absolutely correctly classified instances, either positive or negative, divided by the entire number of instances shown in Eq. (3). The absolute error of Prediction can be calculated using Eq. (4). Mean Squared Error (MSE) is the average of the square of the difference between the desired response and the actual system output (the error), the formula for the mean squared error is shown id Eq. (5).

TPR= TP / TP +FN----------1
FPR= FP/ FP +TN----------2
TOTAL ACCURACY= [TP+TN] / [TP +FP+TN +FN]----------3
Absolute Error of prediction ε =P- P`----------4

$$MSE = \frac{\sum_{j=0}^{P}\sum_{i=0}^{N}(d_{ij} - y_{ij})^2}{N\,P} \qquad (5)$$

Mean Squared Error - (MSE)

Where P = number of output processing elements, N is number of exemplars in the data set, yij is network output for exemplars i at processing element j, dij is desired output for exemplars i at processing element j.

7.Conclusion:

Provided is an attack classification method for computer network security. In the attack classification method, attacks are classified depending on vulnerability abused by an attack, attack propagation skills, and attack intentions. The classification results are arranged in the order of the vulnerability abused by an attack, the attack propagation skills, and the attack intentions. The arranged classification results with unknown attack prevention are output. Accordingly, it is possible to easily detect an attack flow where an attack

An attack is identified when the attacker returns and tries to access the impersonated resources. There is no reason for legitimate traffic to access these resources because they don't exist, so any attempt to connect constitutes an attack. This is a good way to stop the unknown attackers.

Reference:

[1] Rodrigues Gomes and Luiz Antonio da Frota Mattos Attacks Detection Based on IP and TCP Protocols Violations Normal
[2] Liang Guangmin, "Modeling Unknown Web Attacks in Network Anomaly Detection," iccit, vol. 2, pp.112-116, 2008 Third International Conference on Convergence and Hybrid Information Technology, 2008
[3] Stig Andersson, Andrew Clark, George Mohay, Bradley Schatz and Jacob Zimmermann "A Framework for Detecting Network-based Code Injection Attacks Targeting Windows and UNIX"
[4] Nenad DULANOVIĆ,Dane HINIĆ,Dejan SIMIĆ "AN INTRUSION PREVENTION SYSTEM AS A PROACTIVE SECURITY MECHANISM IN NETWORK INFRASTRUCTURE"
[5] Ibrahim A. Farag Mohammed A. Shouman Tarek S. Sobh Heba Z. El-Fiqi "Intelligent System for Worm Detection"
[6] Tala Tafazzoli and Seyed Hadi Sadjadi "Malware fuzzy ontology for semantic web"
[7] Leandro N. de Castro & Fernando J. Von Zuben "Immune and Neural Network Models: Theoretical and Empirical Comparisons**"**

Author Biography:

Prof. Dr K.KUPPUSAMY is working as an Associate Professor, 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 22 years of teaching experience at PG level in the field of Computer Science. He has presented many papers in the National and International conferences. His areas of research interests include Information/Network Security, Algorithms, Neural Networks, Software Engineering & Testing and Optimization Techniques.

Mr S.MURUGAN is Working as ACTS_Coordinator, CDAC ,Bangalore.He received BSc in Physics from Madurai Kamaraj University ,Madurai, in 1989 and MCA degree in Computer Applications from Alagappa University,Karaikudi,Tamilnadu ,India and MPhil(CS) from Manonmaniam Sundaranar University,Tirunelveli,Tamilnadu,India . He has 18 years of teaching and admin experience at PG level in the field of Computer Science. He has published 6 papers in the National conferences and 2 in International conference. His research interests include: Intelligence Network Security Algorithms, Malware prevention and Detection mechanism and algorithm. He has published 8 books and courseware in the field of Computer Science.