The Utilization of Artificial Intelligence in a Hybrid Intrusion Detection System

MARTIN BOTHA, ROSSOUW VON SOLMS, KENT PERRY, EDWIN LOUBSER AND GEORGE YAMOYANY
Port Elizabeth Technikon

Computer security, and intrusion detection in particular, have become increasingly important in today's business environment, to ensure safe and trusted commerce between business partners as well as effective organisational functioning. Various approaches to intrusion detection are currently being utilized, but unfortunately in practice these approaches are relatively ineffective and inefficient. New means and ways that will minimize these shortcomings must, therefore, continuously be researched and defined. This paper will propose a proactive and dynamic model, based on trend analysis, fuzzy logic and neural networks that could be utilized to minimise and control intrusion in an organisation's computer system.

Categories and Subject Descriptors: D.1 [**Software**]: Programming Techniques; I.2 [**Computing Methodologies**]: Artificial Intelligence; I.5 [**Computing Methodologies**]: Pattern Recognition

General Terms: Security, Algorithms

Additional Key Words and Phrases: Intrusion Detection, Intrusion Detection Systems, Fuzzy Logic, Neural Network, Pattern Recognition

1. INTRODUCTION

Over the last few decades, information has become an organisation's most precious asset and everything an organisation does involves using information in some way or another [Peppard 1993; Von Solms 1993]. Organisations have therefore become increasingly dependent on the rapid access and management of information since more information is being stored and processed on network-based computers than ever before. The increase in connectivity not only provides access to larger and varied resources of data more quickly than ever before, it also provides an access path to the data from virtually anywhere in the network-based environment [Seleznyov 2001].

One example that will support last mentioned statement is the introduction of e-commerce. E-commerce is a relatively new discipline in information technology that was developed to support the need of business to complete electronic transactions (which consist of information) over a network environment. Many organisations have already moved towards e-commerce technology and it is expected that many more will do the same over the next few years [Berst 1999].

The introduction of technologies such as e-commerce will not only increase the preciousness of information, but will also increase security requirements of those organizations which are intending to utilize such technologies. Evidence of these requirements can be seen in the 2001 CSI/FBI Computer Crime and Security Survey [Power 2001]. According to this source the annual financial losses caused through security breaches in 2001 have increased by 277% when compared to the results from 1997.

Information is currently protected through a process of identifying, implementing, managing and maintaining a set of information security controls or countermeasures [GMITS 1998]. These security controls can be of a physical (for example door locks), a technical (for example passwords) and/or a procedural nature (for example to make back-up copies of critical files)[Stallings 1995].

Intrusion detection has become an integral part of the information security process since it can implement and manage the identified information security controls [Brace 2000]. Intrusion detection is implemented by an intrusion detection system and today there are many commercial intrusion detection systems available. These commercial implementations are generally restricted in their monitoring functionality [Dowland 2000] and more research is currently being performed to improve this functionality.

In this paper, a new intrusion detection model will be proposed and described that can be utilized to improve the intrusion monitoring functionality in an intrusion detection system. The model is based on the assumption that the intruder's behaviour can be grouped into common generic intrusion phases and that all users' actions on the system can be monitored in terms of these phases.

Author Addresses:
M Botha., Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000; bothamar@saps.org.za
R von Solms, Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000; rossouw@petech.ac.za.

The first part of the paper will attempt to provide an overview on intrusion detection systems (IDS). The overview will commence with a discussion on the most important definitions and components of intrusion detection systems as well as shortcomings currently encountered with misuse intrusion detection systems. This section will conclude with a short discussion on how the shortcomings can be addressed.

The second part of the paper will spell out a strategy that can be used to address the current shortcomings. It will commence with a discussion on how this is accomplished by briefly describing a generic hybrid intrusion identification strategy. This will be followed by a discussion on each of the three components, with special references to the misuse component.

The final part of the paper will propose a novice model that can be used to improve the restricted intrusion monitoring functionality of current intrusion detection systems, and will commence with the identification of nine major components of the model. It will then discuss each component in detail with specific reference to the main functions. This section will conclude with a discussion on how the model will be implemented in the Windows environment.

2. OVERVIEW OF CURRENT INTRUSION DETECTION SYSTEMS

The monitoring functionality of current intrusion detection systems was highlighted as one of the main problem areas that need to be researched. In order to understand this problem one has to investigate the terms intrusion detection, intrusion detection system and intrusion detection analysis approach as well as problems encountered with commercial intrusion detection systems.

Bace [2000] defines intrusion detection as the process of intelligently monitoring the events occurring in a computer system or network, analysing them for signs of violations of the security policy. The intrusion detection process is performed by an intrusion detection system and is defined as a software or hardware product that monitors the events occurring in a computer system or network and analyses them for signs of intrusions, which are defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a host or network .

An intrusion detection system consists of three functional components [Bace 2000, p.27), namely:

1. An information source that provides a stream of event records;
2. An analysis engine that finds signs of intrusions; and
3. A response component that generates reactions based on the outcome of the analysis engine.

The first component of an intrusion detection system is a data source. This element can also be considered as an event generator. Data sources can be categorized into four categories, namely:

1. Host-based monitors: This monitor collects data from sources internal to a computer, usually at the operating system level;
2. Network-based monitors: This monitor collects network packets;
3. Application-based monitors: This monitor collects data from running applications; and
4. Target-based monitors: This monitor uses cryptographic hash functions to detect alterations to system objects and then compares these alterations to a policy.

The second component of an intrusion detection system is an analysis engine. This component takes information from the data source and examines the data for symptoms of attacks or other policy violations. The analysis engine can use one or both of the following analysis approaches:

1. *Misuse detection*: An analysis engine that implements this strategy will search for something defined to be 'bad'. This type of detection engine detects intrusions that follow well-known patterns of attacks (or signatures) that exploit known software vulnerabilities [Kumar and Spafford 1995; Ilgun and Kemmerer 1995]. The primary limitation of this approach is that it looks only for known weaknesses, and may not be of much use in detecting unknown future intrusions [Seleznyov 2000].
2. *Anomaly detection*: An anomaly based detection engine will search for something rare or unusual. They analyse system event streams, using statistical techniques to find patterns of activity that appear to be abnormal. The main problems of anomaly based intrusion detection systems are that they tend to be computationally expensive because several metrics are often maintained that need to be updated against every system activity and they may be gradually trained incorrectly to recognize an intrusive behaviour as normal in the future due to insufficient data [Seleznyov 2000].

Current commercial intrusion detection systems that must combat intrusion attacks are mostly based on the misuse detection approach. This means that these systems will only be able to detect known intrusion attacks and in most cases in a reactive manner which make them highly ineffective for the current status of information security. The ineffectiveness is mainly due to the following:

1. Normally, intruders perform non-stereo type attacks which do not match the known attack patterns of the commercial intrusion detection systems;
2. The process of developing new attack patterns is time consuming; and
3. It is difficult to identify new effective sensors or signatures that can be used by the intrusion detection systems to identify intrusion attacks due to insufficient attack data [Mc Hugh 2001].

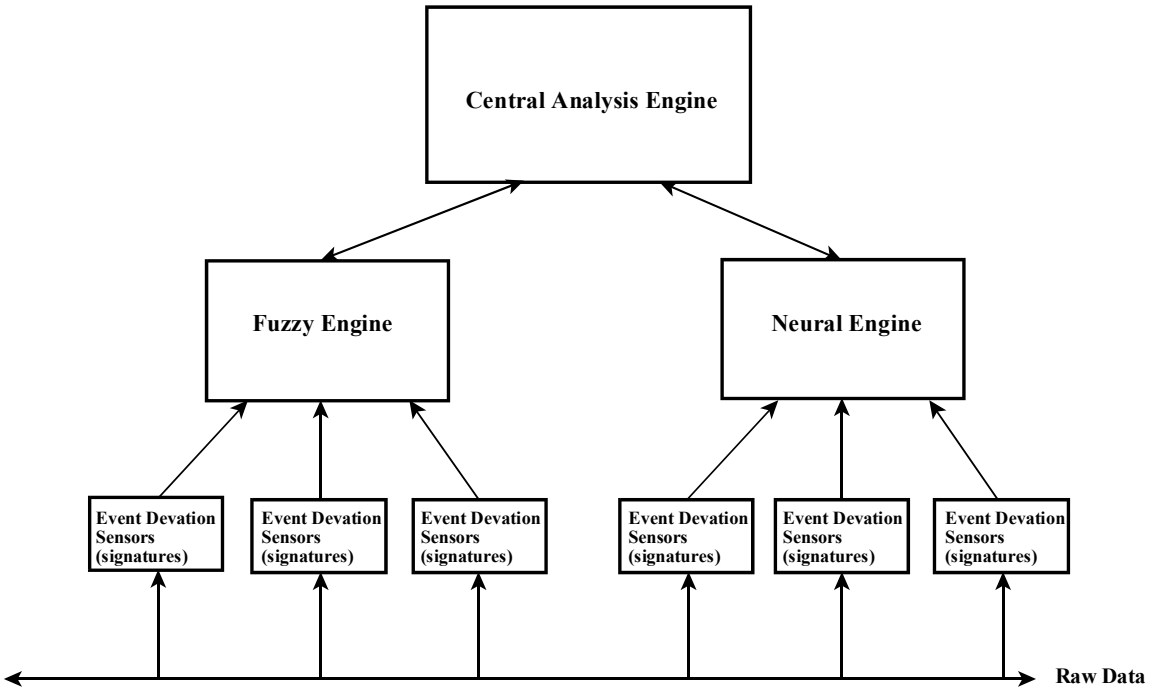


Figure 1: Generic Hybrid Intrusion Identification Strategy

The monitoring functionality of current intrusion detection systems can be improved by utilizing a hybrid intrusion detection system which consists of both anomaly and misuse analysis strategies. The anomaly strategy will be used to detect new or unknown attacks, or other scenarios of concern, while misuse strategy will be used to detect known attack signatures and protects the integrity of the anomaly strategy by ensuring that a patient adversary cannot gradually change behaviour patterns to re-train the anomaly detector to accept attack behaviour as normal. This strategy will be discussed in more detail in the next section.

3. GENERIC HYBRID INTRUSION IDENTIFICATION STRATEGY

The previous section concluded with a description of a hybrid intrusion detection system. This section will explain how this hybrid system *idea* can be used to improve the monitoring functionality of current intrusion detection systems.

The hybrid system *idea* will be implemented through a generic hybrid intrusion identification strategy. This strategy consists of three independent computational components. These components are:

1. a central analysis engine,
2. a fuzzy engine, and
3. a neural engine.

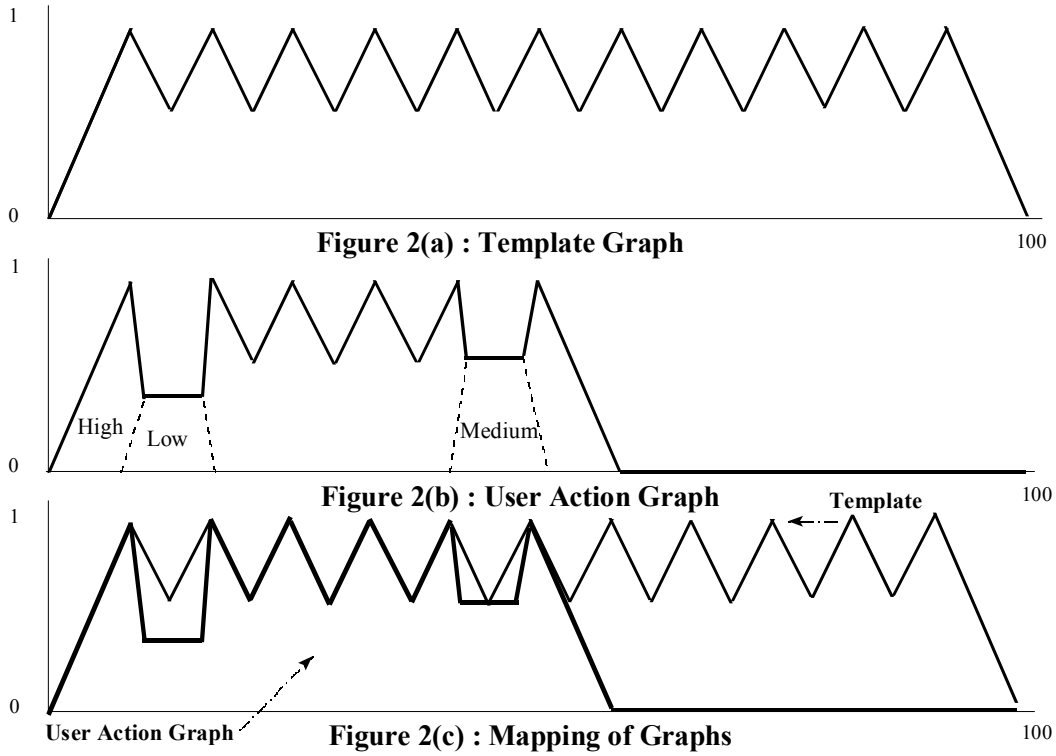
The hierarchical hybrid strategy is illustrated in Figure 1. The figure demonstrates a streamlined, centralised intrusion detection design that consists of both the misuse detection approach and the anomaly detection approach.

The fuzzy engine component implements the misuse detection approach and is based on fuzzy logic and a set of generic intrusion phases. The engine will provide a more effective monitoring functionality because it will not require regular updates on new intrusion attacks. The generic intrusion phases refer to the steps followed by an intruder when conducting an intrusion attack. During each step, system objects and resources are misused by the intruder. Through previous research performed on generic intrusion phases [Nomad 1999; Cooper 1995), it was possible to define the following phases:

1. Probing phase;
2. Initial access phase;
3. Super-user access phase;
4. Hacking phase;
5. Covering phase; and
6. Backdoor phase.

Fuzzy logic utilized in the fuzzy engine, is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth, thus truth values which are between completely true and completely false [Kosko

1993]. The objective of the engine is to compare (i) the generic intrusion phases to (ii) the actions of a user or intruder thereby predicting



patterns of misuse. This will be done by developing two graphs using fuzzy logic. The first graph will be called the template and will represent the six generic intrusion phases, and the second graph will represent the actual actions of the user/intruder and will be called the user action graph. By mapping the two graphs, it is possible to determine patterns of misuse. The output of this mapping process is a numeric value that will be used by the central strategy engine to determine if a 'user' is carrying out an intrusion attack. Figure 2(a) represents the template, 2(b) represents the user action graph and 2(c) shows how the two graphs are mapped.

The second component is the neural engine and operates at the same time as the fuzzy engine. This component is based on an artificial neural network that allows the engine to exploit the advantage of not defining statistical assumptions that might influence the accuracy of the engine. The engine analyses the commands used by users on the system to perform user behaviour recognition, thus performing the anomaly detection approach. The output of this behaviour recognition process is a numeric value and will also be utilized by the central strategy engine to determine if a 'user' is carrying out an intrusion attack.

The last component is the central engine. This engine is based on a statistical measure that will convert the values received from both the fuzzy and neural engines into a probability value. This is done by calculating the mean for every group of individual values and then determining the confidence level that the user is performing an intrusion attack. The engine will use this value to trigger an alarm in the form of a response or countermeasure.

The generic hybrid intrusion identification strategy was briefly explained in this section. The next section will discuss how this strategy can be utilized to develop an intrusion detection system that can be used to monitor and detect intrusions dynamically and in real-time.

4. THE CONCEPTUAL MODEL: NEGPAIM

Having introduced the generic hybrid intrusion identification strategy, it is now possible to define how this strategy can be implemented in an intrusion detection system. The strategy will be implemented through a model called Next Generation Proactive Identification Model (NeGPAIM). NeGPAIM is an improved version of PAIM that was developed as a misuse based intrusion detection system. NeGPAIM is based on a three-tier architecture that will maximise the performance of the intrusion detection system. Figure 3 shows the various components of NeGPAIM. The core of NeGPAIM consists of nine major components. These components are called: Information Provider,

Collector, Coupler, Information Refiner, Fuzzy Engine, Neural Engine, Central Analysis Engine, Responder and Manager. The various components of NeGPAIM will be discussed briefly in the sections below.

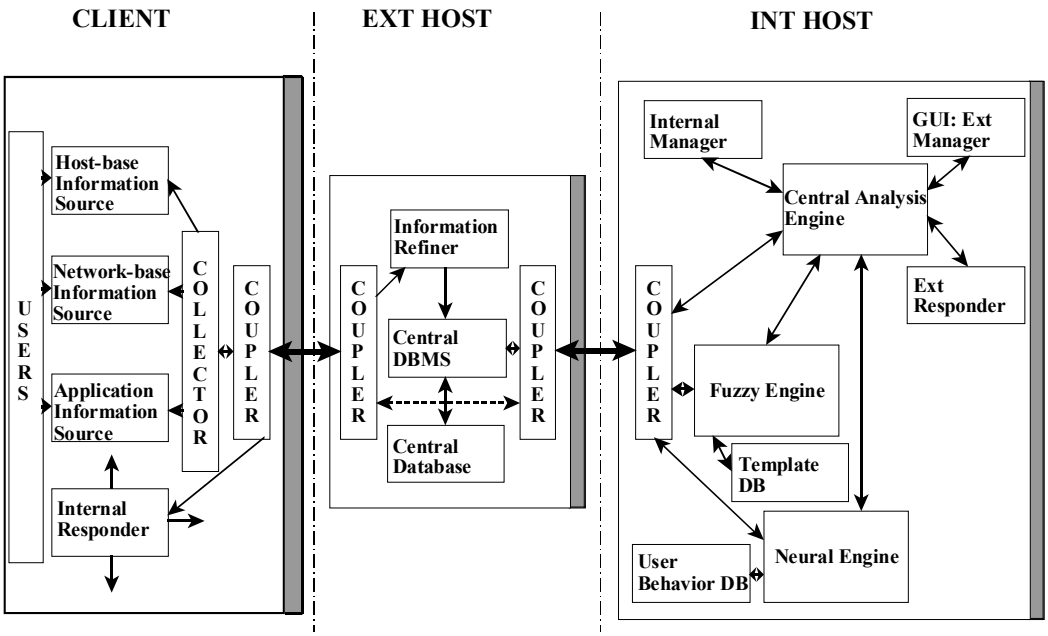


Figure 3: General Representation of NeGPAIM

4.1 Information Provider

The information provider component refers to the various sources that will provide input data to the intrusion detection system. These sources can be grouped as host-based sources, network-based sources and application-based sources. It must be emphasized that NeGPAIM does not control this component, but will only read information (input data) from it.

4.2 Collector

The collector is a software component that reads the predefined information from the information provider component. The component functions independently from the information provider in order not to decrease the performance level of the system applications. This component will be implemented at the operating system level and will read the input data directly form the various audit trails records such as system level audit log, application level audit log, user level audit log and firewall audit log. The collector will read the various audit logs in real-time and in chronological order. After the collector has collected input data on an event, it will send it to the information refiner. This process is performed on a continuous basis.

4.3 Coupler

The coupler component is an interface unit that provides network communication between the local client(s), external host and the internal host. The coupler's principal function in NeGPAIM is to ensure a consistent, reliable and secure exchange of data between the local client(s) and external and internal host.

4.4 Information Refiner

The information refiner is a software component that will be implemented on the external host. This component has two major functions:

1. To structure the collected input data.
2. To integrate the collected input data.

The information refiner will have to change the existing audit log entries into a standard format that can be used by the fuzzy engine and the neural engine. This will be achieved by defining an audit log structure for every type of audit log. The information refiner will use two types of structures, namely hard structure and soft structure. The hard structure defines nine fields for each record while the soft structure defines only five fields. The information refiner

will perform the second function by sorting the input data into a chronological order. The input data will be ordered according to a standard time reference such as Network Time Protocol [Bace 2000]. The sorted data will then be stored in the central information source by the central database management system.

4.5 Fuzzy Engine

The fuzzy engine is one of the two low-level processing units of NeGPAIM and will process the input data. The engine will compute a template and user action graph for each user and map the two graphs to determine whether a user (intruder) is performing an intrusion attack. If the user (intruder) is conducting an intrusion attack, the engine will notify the central analysis engine with an intrusion probability value. This process is performed on a continuous basis.

4.6 Neural Engine

The neural engine is the second low-level processing unit and will also process the input data. This engine will process the data by searching for patterns of user behaviour that appear to be abnormal. The neural engine processing involves two stages namely:

1. In the first stage, the engine is populated by a training set of historical sample data that is representative of the user behaviour. This training process is only performed once for each user on the system and thereafter the historical behaviour pattern (reference) is updated periodically.
2. In the second stage, the engine accepts input data (event data) and compares it to historical behaviour references, determining similarities and differences. This process will be conducted on a continuous basis. The engine will report any abnormal behaviour to the central analysis engine by way of an intrusion probability value.

4.7 Central Analysis Engine

The central analysis engine is a high-level processing unit. The objective of this engine is to analyse the possibility that a user (intruder) is conducting an intrusion attack. The engine will achieve this by performing the following functions:

1. To calculate the statistical means for both the probability values received from the low-level components for each user on a continuous basis.
2. To convert this statistical means to an intrusion probability level that will be used to combat intrusion proactively and dynamically on a continuous basis.
3. To ensure accurate intrusion identification by implementing threshold detection.

The output of the central analysis engine is the generation of an alert signal that will be sent to the responder.

4.8 Responder

The responder component consists of two modules namely the internal responder and the external responder. The internal responder is resided on the client(s) and will protect the system by issuing tasks such as an explicit challenge to the intruder for further authentication, locking the intruder's terminal and terminating the intrusion session [Dowland 2000]. The tasks will be determined by the level of the alert signal issued by the central analysis engine.

The external responder is resided on the internal host and will perform a notification function. The function will include:

1. generating alert messages according to alert signals; and
2. notifying the security officer via pop-up messages, e-mail messages, cellular phone calls and conventional phone calls.

4.9 Manager

The manager component consists of two modules namely, internal manager and external manager. The manager will address the following management aspects of NeGPAIM:

1. Configuration management: Configuration management includes management of detection functions and corresponding response mechanisms used. Configuration of the detection function involves setting the criteria for user behaviour characteristics, frequency of user behaviour refinement, fuzzy rules, threshold levels and level of data collection. The management of the response function determines the system behaviour when generating an alert signal. This may include settings such as level of response required for each alert signal.
2. Security services management: Security services management involves managing the security services that are part of NeGPAIM. It involves controlling user credentials, confidentiality, integrity and access control services.

The nine major components of NeGPAIM were briefly introduced and discussed. These components are resided on a three-tier architecture, namely client, external host and internal host. By utilizing this architecture, it is not only possible to improve both the security and the performance of NeGPAIM, but it will also provide the security officer with an effective mechanism to identify and monitor intrusion attacks. The implementation of NeGPAIM will be discussed next.

5. PRACTICAL IMPLEMENTATION OF NEGPAIM

NeGPAIM has been proposed and described in the previous section. The model is generic and can be implemented in many different computing environments. In order to test the feasibility of the model, an initial prototype called Implementing Fuzzy Engine Prototype (IFEP) was developed. This prototype has only implemented the fuzzy engine since it was unknown if fuzzy logic can successfully be utilized to performing misuse detection. However, research has proved that neural networks can successfully be utilized to perform anomaly detection and was therefore not included in the prototype.

The initial prototype was developed by employing CLIPS developing software. The prototype was tested by way of several independent case studies. When evaluating IFEP, it was concluded that the prototype was successful in performing misuse detection.

Work is currently being conducted to develop a full implementation of NeGPAIM for a Windows platform. The implementation is called Hybrid Intrusion Detection System (HIDS). HIDS will be implemented on a server (host) as a software suite. The software suite will consist of an GUI, communication interface, retrieving unit and processing units The graphical user interface modules are developed in Visual Basic, while the communication interface and the processing modules (fuzzy engine, neural engine and central analysis engine) are developed in Visual C++. The retrieving unit is developed through a Perl script.

The results of the tests that were performed on HIDS indicate that the prototype will be implemented successfully. When completed, the prototype will be tested in a live environment to determine if NeGPAIM can overcome the shortcomings of current commercial intrusion detection systems.

6. CONCLUSION

A new model, named NeGPAIM, which will monitor and identify intrusion attacks in a proactive and dynamic manner is proposed in this paper. The model will identify intrusion attacks proactively by not allowing the attacker to perform his attack objective. For example, if the attacker has the objective of stealing credit card information, the model will identify the attack at an early stage and activate a response that will disconnect the attack session. The model will be dynamic, because it does not require regular updates of new intrusion patterns. The model will achieve these features by implementing a hybrid intrusion detection system. This system consists of two low level components and one high level component. The first low-level component is called the fuzzy engine and implements the misuse detection approach. The fuzzy engine differs from current misuse detection system in the sense that it does not search for a particular pattern of attack, but searches for general misuse of resources and objects. The second low level component is called the neural engine and implements the anomaly detection approach. The central analysis engine is the high level component that will analyse the outputs of the two low level engines.

Finally, although the new model will be very effective, it must never be used to replace the responsibility of the information security officer. It should only assist the information security officer in the protection of the valuable information of the organization and associated resources, and thus improve the effectiveness of existing security controls.

REFERENCES

BACE, R.G. (2000). *Intrusion Detection*. Macmillan Technical Publishing, Indianapolis, In USA. [ISBN: 1-57870-185-6].

BERKAN, R.C.(1997). *Fuzzy System Design Principles*(1st ed.). United States of America : IEEE Press Marketing.

BERST, J. (1999). *Hacking 101 : Why Should You Know What They Know* [on-line]. Zdnet. Available from Internet : URL http://www.zdnet.com/anchordesk/story/story 3163.html.

COOPER, F.J., GOGGANS, C., HALVEY, J.K., HUGHES, L., MORGAN, L., SIYAN, K., STALLINGS, W., & STEPHENSON, P. (1995). *Implementing Internet Security*. Indianapolis : New Riders. [ISBN: 1-562-05471-6].

DOWNLAND, P.S & FURNELL, S.M.(2000). *A Conceptual Intrusion Monitoring Architecture and Thoughts on Practical Implementation*. IFEP `2000.

GMITS. 1997. *Guidelines for the management of IT security (GMITS). Part 2 : Managing and Planning IT Security*, TR 13335-2, ISO/IEC, JTC, 1/SC 27

ILGUN, K. & KEMMERER, R. (1995). *State Transition Analysis: A Rule-Based Intrusion Detection Approach*. IEEE Transaction on Software Engineering, 21(3):pp. 181-199.

KOSKO, B. (1993). *Fuzzy Thinking*(1st ed.). New York, U.S.A.: Hyperion Publishing.

KUMAR, S. & SPAFFORD, E. (1995). *A Software Architecture to Support Misuse Intrusion Detection*. In: The 18th National Information Security Conference, pp. 194-204.

NOMAD, S. (1999). *The Hack FAQ* [on-line]. Available from Internet : URL: http://www.nmrc.org/faqs/hack faq/hack FA~1.html

PEPPARD, J. (1993). Information, Technology and Strategy. In: J. Peppard (Ed.), *I.T. Strategy for Business* (pp. 1-25). London : Pitman Publishing. [ISBN: 0-273-60024-9].

POWER, R. (2001). 2001 CSI/FBI Computer Crime and Security Survey, *Computer Security Issues and Trends*, Vol.VII, No.1.

SELEZNYOV, A. (2000). *A Hybrid Model for Intrusion Detection Based on Temporal Probabilistic Networks*, IFIP `2000.

SELEZNYOV, A. (2001). *A Methodology to Detect Temporal Regularities in User Behaviour for Anomaly Detection*, IFIP `2001.

VON SOLMS, R. (1993*). A Process Approach to Information Security Management*, IFIP'93, WG 11.1, Toronto, Canada, 1993.