Artificial Neural Networks for Misuse Detection

James Cannady

School of Computer and Information Sciences
Nova Southeastern University
Fort Lauderdale, FL 33314
cannadyj@scis.nova.edu

Abstract

Misuse detection is the process of attempting to identify instances of network attacks by comparing current activity against the expected actions of an intruder. Most current approaches to misuse detection involve the use of rule-based expert systems to identify indications of known attacks. However, these techniques are less successful in identifying attacks which vary from expected patterns. Artificial neural networks provide the potential to identify and classify network activity based on limited, incomplete, and nonlinear data sources. We present an approach to the process of misuse detection that utilizes the analytical strengths of neural networks, and we provide the results from our preliminary analysis of this approach.

Keywords: Intrusion detection, misuse detection, neural networks, computer security.

1. Introduction

Because of the increasing dependence which companies and government agencies have on their computer networks the importance of protecting these systems from attack is critical. A single intrusion of a computer network can result in the loss or unauthorized utilization or modification of large amounts of data and cause users to question the reliability of all of the information on the network. There are numerous methods of responding to a network intrusion, but they all require the accurate and timely identification of the attack.

This paper presents an analysis of the applicability of neural networks in the identification of instances of external attacks against a network. The results of tests conducted on a neural network, which was designed as a proof-of-concept, are also presented. Finally, the areas of future research that are being conducted in this area are discussed.

1.1 Intrusion Detection Systems

1.1.1 Background

The timely and accurate detection of computer and network system intrusions has always been an elusive goal for system administrators and information security researchers. The individual creativity of attackers, the wide range of computer hardware and operating systems, and the ever-changing nature of the overall threat to target systems have contributed to the difficulty in effectively identifying intrusions. While the complexities of host computers already made intrusion detection a difficult endeavor, the increasing prevalence of distributed network-based systems and insecure networks such as the Internet has greatly increased the need for intrusion detection [20].

There are two general categories of attacks which intrusion detection technologies attempt to identify - anomaly detection and misuse detection [1,13]. Anomaly detection identifies activities that vary from established patterns for users, or groups of users. Anomaly detection typically involves the creation of knowledge bases that contain the profiles of the monitored activities.

The second general approach to intrusion detection is misuse detection. This technique involves the comparison of a user's activities with the known behaviors of attackers attempting to penetrate a system [17,18]. While anomaly detection typically utilizes threshold monitoring to indicate when a certain established metric has been reached, misuse detection techniques frequently utilize a rule-based approach. When applied to misuse detection, the rules become scenarios for network attacks. The intrusion detection mechanism identifies a potential attack if a user's activities are found to be consistent with the established rules. The use of comprehensive rules is critical in the application of expert systems for intrusion detection.

1.1.2 Current Approaches to Intrusion Detection

Most current approaches to the process of detecting intrusions utilize some form of rule-based analysis. Rule-Based analysis relies on sets of predefined rules that are provided by an administrator, automatically created by the system, or both. Expert systems are the most common form of rule-based intrusion detection approaches [8, 24]. The early intrusion detection research efforts realized the inefficiency of any approach that required a manual review of a system audit trail. While the information necessary to identify attacks was believed to be present within the voluminous audit data, an effective review of the material required the use of an automated system. The use of expert system techniques in intrusion detection mechanisms was a significant milestone in the development of effective and practical detection-based information security systems [1, 8, 19, 21, 24, and 28].

An expert system consists of a set of rules that encode the knowledge of a human "expert". These rules are used by the system to make conclusions about the security-related data from the intrusion detection system. Expert systems permit the incorporation of an extensive amount of human experience into a computer application that then utilizes that knowledge to identify activities that match the defined characteristics of misuse and attack.

Unfortunately, expert systems require frequent updates to remain current. While expert systems offer an enhanced ability to review audit data, the required updates may be ignored or performed infrequently by the administrator. At a minimum, this leads to an expert system with reduced capabilities. At worst, this lack of maintenance will degrade the security of the entire system by causing the system's users to be misled into believing that the system is secure, even as one of the key components becomes increasingly ineffective over time.

Rule-based systems suffer from an inability to detect attacks scenarios that may occur over an extended period of time. While the individual instances of suspicious activity may be detected by the system, they may not be reported if they appear to occur in isolation. Intrusion scenarios in which multiple attackers operate in concert are also difficult for these methods to detect because they do not focus on the state transitions in an attack, but instead concentrate on the occurrence of individual elements. Any division of an attack either over time or among several seemingly unrelated attackers is difficult for these methods to detect.

Rule-based systems also lack flexibility in the rule-to-audit record representation. Slight variations in an attack sequence can effect the activity-rule comparison to a degree that the intrusion is not detected by the intrusion detection mechanism. While increasing the level of abstraction of the rule-base does provide a partial solution to this weakness, it also reduces the granularity of the intrusion detection device.

A number of non-expert system-based approaches to intrusion detection have been developed in the past several years [4, 5, 6, 9, 15, 25, and 26]. While many of these have shown substantial promise, expert systems remain the most commonly accepted approach to the detection of attacks.

1.2 Neural Networks

An artificial neural network consists of a collection of processing elements that are highly interconnected and transform a set of inputs to a set of desired outputs. The result of the transformation is determined by the characteristics of the elements and the weights associated with the interconnections among them. By modifying the connections between the nodes the network is able to adapt to the desired outputs [9, 12].

Unlike expert systems, which can provide the user with a definitive answer if the characteristics which are reviewed exactly match those which have been coded in the rulebase, a neural network conducts an analysis of the information and provides a probability estimate that the data matches the characteristics which it has been trained to recognize. While the probability of a match determined by a neural network can be 100%, the accuracy of its decisions relies totally on the experience the system gains in analyzing examples of the stated problem.

The neural network gains the experience initially by training the system to correctly identify pre-selected examples of the problem. The response of the neural network is reviewed and the configuration of the system is refined until the neural network's analysis of the training data reaches a satisfactory level. In addition to the initial training period, the neural network also gains experience over time as it conducts analyses on data related to the problem.

1.3 Neural Network Intrusion Detection Systems

A limited amount of research has been conducted on the application of neural networks to detecting computer intrusions. Artificial neural networks offer the potential to resolve a number of the problems encountered by the other current approaches to intrusion detection. Artificial neural networks have been proposed as alternatives to the statistical analysis component of anomaly detection systems, [5, 6, 10, 23, and 26]. Statistical Analysis involves statistical comparison of current events to a predetermined set of baseline criteria. The technique is most often employed in the detection of deviations from typical behavior and determination of the similarly of events to those which are indicative of an attack [14]. Neural networks were specifically proposed to identify the typical characteristics of system users and identify statistically significant variations from the user's established behavior.

Artificial neural networks have also been proposed for use in the detection of computer viruses. In [7] and [9] neural networks were proposed as statistical analysis approaches in the detection of viruses and malicious software in computer networks. The neural network architecture which was selected for [9] was a self-organizing feature map which use a single layer of neurons to represent knowledge from a particular domain in the form of a geometrically organized feature map. The proposed network was designed to learn the characteristics of normal system activity and identify statistical variations from the norm that may be an indication of a virus.

2. Application of Neural Networks in Misuse Detection

While there is an increasing need for a system capable of accurately identifying instances of misuse on a network there is currently no applied alternative to rule-based intrusion detection systems. This method has been demonstrated to be relatively effective if the exact characteristics of the attack are known. However, network intrusions are constantly changing because of individual approaches taken by the attackers and regular changes in the software and hardware of the targeted systems. Because of the infinite variety of attacks and attackers even a dedicated effort to constantly update the rulebase of an expert system can never hope to accurately identify the variety of intrusions.

The constantly changing nature of network attacks requires a flexible defensive system that is capable of analyzing the enormous amount of network traffic in a manner which is less structured than rule-based systems. A neural network-based misuse detection system could potentially address many of the problems that are found in rule-based systems.

2.1 Advantages of Neural Network-based Misuse Detection Systems

The first advantage in the utilization of a neural network in the detection of instances of misuse would be the flexibility that the network would provide. A neural network would be capable of analyzing the data from the network, even if the data is incomplete or distorted. Similarly, the network would possess the ability to conduct an analysis with data in a non-linear fashion. Both of these characteristics is important in a networked environment where the information which is received is subject to the random failings of the system. Further, because some attacks may be conducted against the network in a coordinated assault by multiple attackers, the ability to process data from a number of sources in a non-linear fashion is especially important.

The inherent speed of neural networks is another benefit of this approach. Because the protection of computing resources requires the timely identification of attacks, the processing speed of the neural network could enable intrusion responses to be conducted before irreparable damage occurs to the system.

Because the output of a neural network is expressed in the form of a probability the neural network provides a predictive capability to the detection of instances of misuse. A neural network-based misuse detection system would identify the probability that a particular event, or series of events, was indicative of an attack against the system. As the neural network gains experience it will improve its ability to determine where these events are likely to occur in the attack process. This information could then be used to generate a series of events that should occur if this is in fact an intrusion attempt. By tracking the subsequent occurrence of these events the system would be capable of improving the analysis of the events and possibly conducting defensive measures before the attack is successful.

However, the most important advantage of neural networks in misuse detection is the ability of the neural network to "learn" the characteristics of misuse attacks and identify instances that are unlike any which have been observed before by the network. A neural network might be trained to recognize known suspicious events with a high degree of accuracy. While this would be a very valuable ability, since attackers often emulate the "successes" of others, the network would also gain the ability to apply this knowledge to identify instances of attacks which did not match the exact characteristics of previous intrusions. The probability of an attack against the system may be estimated and a potential threat flagged whenever the probability exceeds a specified threshold.

2.2 Disadvantages of Neural Network-based Misuse Detection Systems

There appear to be two primary reasons why neural networks have not been applied to the problem of misuse detection in the past. The first reason relates to the training requirements of the neural network. Because the ability of the artificial neural network to identify indications of an intrusion is completely dependent on the accurate training of the system, the training data and the training methods that are used are critical. The training routine requires a very large amount of data to ensure that the results are statistically accurate. The training of a neural network for misuse detection purposes may require thousands of individual attacks sequences, and this quantity of sensitive information is difficult to obtain.

However, the most significant disadvantage of applying neural networks to intrusion detection is the "black box" nature of the neural network. Unlike expert systems which have hard-coded rules for the analysis of events, neural networks adapt their analysis of data in response to the training which is conducted on the network. The connection weights and transfer functions of the various network nodes are usually frozen after the network has achieved an acceptable level of success in the identification of events. While the network analysis is achieving a sufficient probability of success, the basis for this level of accuracy is not often known. The "Black Box Problem" has plagued neural networks in a number of applications [11]. This is an on-going area of neural network research.

2.3 Potential Implementations

There are two general implementations of neural networks in misuse detection systems. The first involves incorporating them into existing or modified expert systems. Unlike the previous attempts to use neural networks in anomaly detection by using them as replacements for existing statistical analysis components, this proposal involves using the neural network to filter the incoming data for suspicious events which may be indicative of misuse and forward these events to the expert system. This configuration should improve the effectiveness of the detection system by reducing the false alarm rate of the expert system. Because the neural network will determine a probability that a particular event is indicative of an attack, a threshold can be established where the event is forwarded to the expert system for additional analysis. Since the expert system is only receiving data on events which are viewed as suspicious, the sensitivity of the expert system can be increased, (typically, the sensitivity of expert systems must be kept low to reduce the incidence of false alarms). This configuration would be beneficial to organizations that have invested in rule-based expert system technology by improving the effectiveness of the system while it preserves the investment that has been made in existing intrusion detection systems. The disadvantage of this approach would be that as the neural network improved its ability to identify new attacks the expert system would have to be updated to also recognize these as threats. If the expert system were not updated then the new attacks identified by the neural network would increasingly be ignored by the expert system because its rule-base would not be capable of recognizing the new threat.

The second approach would involve the neural network as a standalone misuse detection system. In this configuration, the neural network would receive data from the network stream and analyze the information for instances of misuse. Any instances which are identified as indicative of attack would be forwarded to a security administrator or used by an automated intrusion response system. This approach would offer the benefit of speed over the previous approach, since there would only be a single layer of analysis. In addition, this configuration should improve in effectiveness over time as the network learns the characteristics of attacks. Unlike the first approach, this concept would not be limited by the analytical ability of the expert system, and as a result, it would be able to expand beyond the limits of the expert system's rule-base.

3. Initial Analysis of Approach

In an effort to determine the applicability of neural networks to the problem of misuse detection we conducted an analysis the approach utilizing simulated network traffic. The experiment was designed to determine if indications of attack could be identified from typical network traffic, but it was not intended to completely resolve the issue of applying neural networks to misuse detection. The analysis did not address the potential benefit of identifying *a priori* attacks that may be possible through the use of neural networks. However, determining if a neural network was capable of identifying misuse incidents with a reasonable degree of accuracy was considered to be the first step in applying the technology to this form of intrusion detection.

3.1 Neural Network Description

The first prototype neural network was designed to determine if a neural network was capable of identifying specific events that are indications of misuse. Neural networks had been shown to be capable of identifying TCP/IP network events in [27], but our prototype was designed to test the ability of a neural network to identify indications of misuse. The prototype utilized a MLP architecture that consisted of four fully connected layers with nine input nodes and two output nodes. While there are a number of architectures that could be used to address this problem ([12]) a feed-forward neural network architecture was selected based on the flexibility and applicability of the approach in a variety of problems.

The number of hidden layers, and the number of nodes in the hidden layers, was determined based on the process of trial and error. Each of the hidden nodes and the output node applied a Sigmoid transfer function (**1/(1 + exp (-x))**) to the various connection weights. The neural network was designed to provide an output value of 0.0 and 1.0 in the two output nodes when the analysis indicated no attack and 1.0 and 0.0 in the two output nodes in the event of an attack.

Data for training and testing the prototype was generated using the RealSecure™ network monitor from Internet Security Systems, Inc. RealSecure™ is designed to be used by network security administrators to passively collect data from the network and identify indications of attacks. RealSecure™ uses an expert system that includes over 360 attack signatures that it compares with current network activity to identify intrusions. The RealSecure™ monitor was configured to capture the data for each event which would be consistent with a network frame, (e.g., source address, destination address, packet data, etc.), and the results of the RealSecure™ analysis of each event.

In addition to the "normal" network activity that was collected as events by RealSecure™, the host for the monitor was "attacked" using the Internet Scanner™ product from ISS, Inc, and the Satan scanner. These applications were used because of their ability to generate a large number of simulated attacks against a specified network host. The scanners were configured for a variety of attacks, ranging from denial of service attacks to port scans. Approximately 10000 individual events were collected by RealSecure™ and stored in a Microsoft Access™ database, of which approximately 3000 were simulated attacks.

Three levels of preprocessing of the data were conducted to prepare the data for use in the training and testing of the neural network. In the first round of preprocessing nine of the event record data elements were selected from the available set. The nine elements were selected because they are typically present in network data packets and they provide a complete description of the information transmitted by the packet:

- **Protocol ID** - The protocol associated with the event, (TCP = 0, UDP = 1, ICMP = 2, and Unknown = 3).
- **Source Port –** The port number of the source.
- **Destination Port –** The port number of the destination.
- **Source Address** - The IP address of the source.
- **Destination Address** - The IP address of the destination.
- **ICMP Type –** The type of the ICMP packet (Echo Request or Null).
- **ICMP Code –** The code field from the ICMP packet (None or Null).
- **Raw Data Length –** The length of the data in the packet.
- **Raw Data** - The data portion of the packet.

The second part of the preprocessing phrase consisted of converting three of the nine data elements (ICMP Type, ICMP Code and Raw Data) into a standardized numeric representation. The process involved the creation of relational tables for each of the data types and assigning sequential numbers to each unique type of element. This involved creating DISTINCT SELECT queries for each of the three data types and loading those results into tables that assigned a unique integer to each entry. These three tables were then joined to the table that contained the event records. A query was then used to select six of the nine original elements (ProtocolID, Source Port, Destination Port, Source Address, Destination Address, and Raw Data Length) and the unique identifiers which pertain to the remaining three elements (ICMP Type ID, ICMP Code ID, and Raw Data ID). A tenth element (Attack) was assigned to each record based on a determination of whether this event represented part of an attack on a network, (Table 1). This element was used during training as the target output of the neural network for each record.

Protocol ID	Source Port	Destination Port	Source Address	Destination Address	ICMP Type ID	ICMP Code ID	Raw Data Length	Data ID	Attack
0	2314	80	1573638018	-1580478590	1	1	401	3758	0
0	1611	6101	801886082	-926167166	1	1	0	2633	1

Table 1: Sample of pre-processed events query

The third round of data preprocessing involved the conversion of the results of the query into an ASCII comma delimited format that could be used by the neural network (Table 2).

```
0,2314,80,1573638018,-1580478590,1,1,401,3758,0
 0,1611,6101,801886082,-926167166,1,1,0,2633,1
```

Table 2: Sample of ASCII comma-delimited input strings

The preprocessed data was finally loaded into the DataPro utility provided by Qnet 97.01, (Table 3). Qnet uses this application to load data into the neural network during training and testing.

Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Input 9	Output 1
0	2314	80	1573638018	-1580478590	1	1	401	3758	0
0	1611	6101	801886082	-926167166	1	1	0	2633	1

Table 3: Sample of DataPro input to neural network

3.2 Results

The training of the neural network was conducted using a backpropagation algorithm for 10,000 iterations of the selected training data. Like the feed-forward architecture of the neural network, the use of a backpropagation algorithm for training was based on the proven record of this approach in the development of neural networks for a variety of applications [12]. Of the 9,462 records which were preprocessed for use in the prototype, 1000 were randomly selected for testing and the remaining were used to train the system.

The training/testing iterations of the neural network required 26.13 hours to complete. At the conclusion of the training the following results were obtained:

- Training data root mean square error = 0.058298
- Test data root mean square error = 0.069929
- Training data correlation = 0.982333
- Test data correlation = 0.975569

The figures matched very closely with the desired root mean square (RMS) error of 0.0 and the desired correlation value of 1.0.

After the completion of the training and testing of the MLP neural network the various connection weights were frozen and the network was interrogated. Three sample patterns containing “normal” network events and a single simulated attack event (e.g., ISS scans, Satan scans, SYNFlood, etc.) were used to test the neural network. The MLP was able to correctly identify each of the imbedded attacks in the test data, (Figures 1-3).

While this prototype was not designed to be a complete intrusion detection system, the results clearly demonstrate the potential of a neural network to detect individual instances of possible misuse from a representative network data stream.

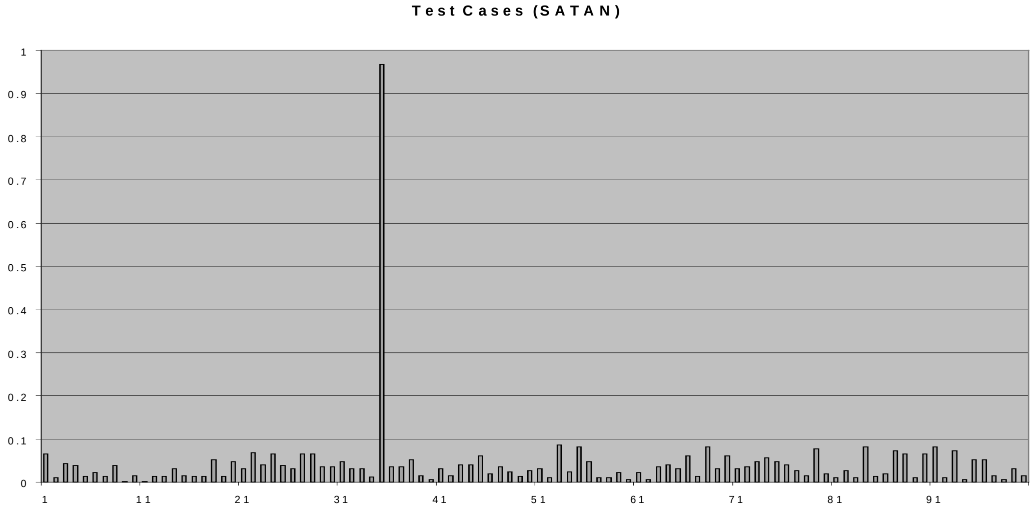

Figure 1 : SYNFlood Attack Test

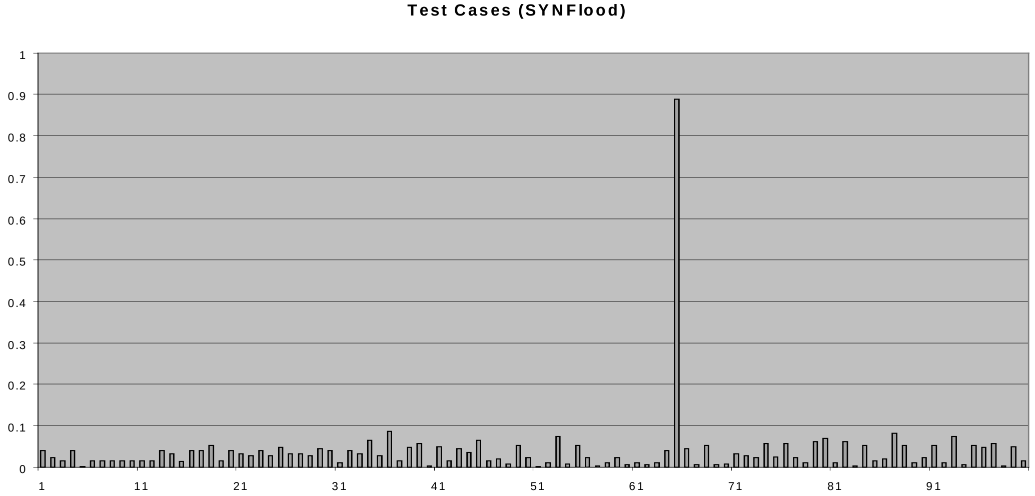

Figure 2 : SATAN Attack Test

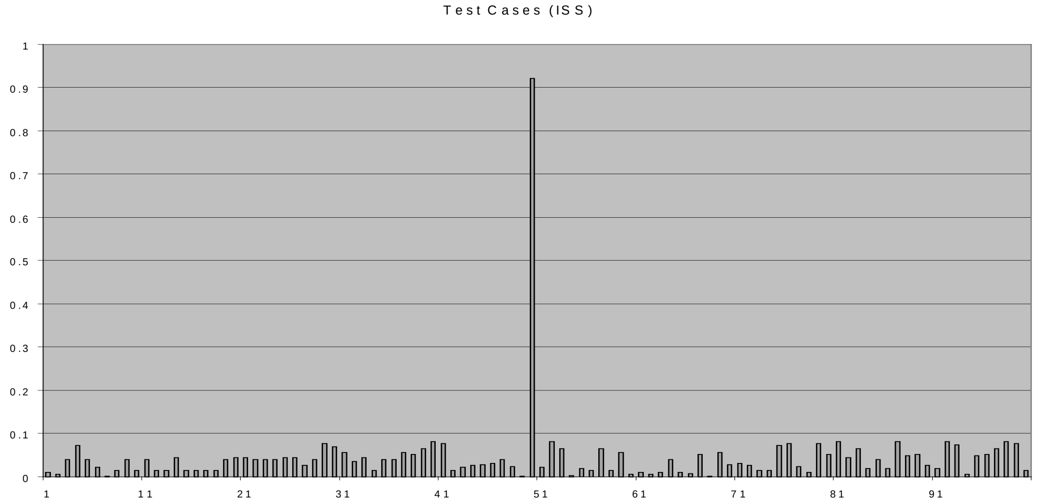

Figure 3 : ISS Scan Attack Test

4. Further Work

The preliminary results from our experimental feed-forward neural network give a positive indication of the potential offered by this approach, but a significant amount of research remains before it can function as an effective intrusion detection system. A complete system will require the ability to directly receive inputs from a network data stream. The most difficult component of the analysis of network traffic by a neural network is the ability to effectively analyze the information in the data portion of an IP datagram. The various commands that are included in the data often provide the most critical element in the process of determining if an attack is occurring against a network.

The most effective neural network architecture is also an issue that must be addressed. A feed-forward neural network that used a backpropagation algorithm was chosen because of its simplicity and reliability in a variety of applications. However, alternatives such as the self-organizing feature map also possess advantages in misuse detection that may promote their use.

In addition, an effective neural network-based approach to misuse detection must be highly adaptive. Most neural network architectures must be retrained if the system is to be capable of improving its analysis in response to changes in the input patterns, (e.g., “new” events are recognized with a consistent probability of being an attack until the network is retrained to improve the recognition of these events). Adaptive resonance theory ([2]) and self-organizing maps ([16]) offer an increased level of adaptability for neural networks, and these approaches are being investigated for possible use in an intrusion detection system.

Finally, regardless of the initial implementation of a neural network-based intrusion detection system for misuse detection it will be essential for the approach to be thoroughly tested in order to gain acceptance as a viable alternative to expert systems. Work has been conducted on taxonomies for testing intrusion detection systems ([3, 22]) that offer a standardized method of validating new technologies. Because of the questions that are certain to arise from the application of neural networks to intrusion detection, the use of these standardized methods is especially important.

5. Conclusion

Research and development of intrusion detection systems has been ongoing since the early 1980's and the challenges faced by designers increase as the targeted systems because more diverse and complex. Misuse detection is a particularly difficult problem because of the extensive number of vulnerabilities in computer systems and the creativity of the attackers. Neural networks provide a number of advantages in the detection of these attacks. The early results of our tests of these technologies show significant promise, and our future work will involve the refinement of this approach and the development of a full-scale demonstration system.

6. References

[1] Anderson, D., Frivold, T. & Valdes, A (May, 1995). Next-generation Intrusion Detection Expert System (NIDES): A Summary. *SRI International Technical Report SRI-CSL-95-07*.

[2] Carpenter, G.A. & Grossberg, S. (1987). A Massively Parallel Architecture for a Self-Organizing Neural pattern Recognition Machine. *Computer Vision, Graphics and Image Processing* 37, 54-115.

[3] Chung, M., Puketza, N., Olsson, R.A., & Mukherjee, B. (1995) Simulating Concurrent Intrusions for Testing Intrusion Detection Systems:Parallelizing. In *NISSC*. pp. 173-183.

[4] Cramer, M., et. al. (1995). New Methods of Intrusion Detection using Control-Loop Measurement. In *Proceedings of the Technology in Information Security Conference (TISC) '95.* pp. 1-10.

[5] Debar, H., Becke, M., & Siboni, D. (1992). A Neural Network Component for an Intrusion Detection System. In *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy.*

[6] Debar, H. & Dorizzi, B. (1992). An Application of a Recurrent Network to an Intrusion Detection System. In *Proceedings of the International Joint Conference on Neural Networks.* pp. (II)478-483.

[7] Denault, M., Gritzalis, D., Karagiannis, D., and Spirakis, P. (1994). Intrusion Detection: Approach and Performance Issues of the SECURENET System. In *Computers and Security Vol. 13, No. 6, pp. 495-507*

[8] Denning, Dorothy. (February, 1987). An Intrusion-Detection Model. *IEEE Transactions on Software Engineering, Vol. SE-13, No. 2.*

[9] Fox, Kevin L., Henning, Rhonda R., and Reed, Jonathan H. (1990). A Neural Network Approach Towards Intrusion Detection. In *Proceedings of the 13th National Computer Security Conference.*

[10] Frank, Jeremy. (1994). Artificial Intelligence and Intrusion Detection: Current and Future Directions. In *Proceedings of the 17th National Computer Security Conference.*

[11] Fu, L. (1992). A Neural Network Model for Learning Rule-Based Systems. In *Proceedings of the International Joint Conference on Neural Networks.* pp. (I) 343-348.

[12] Hammerstrom, Dan. (June, 1993). Neural Networks At Work. *IEEE Spectrum*. pp. 26-53.

[13] Helman, P., Liepins, G., and Richards, W. (1992). Foundations of Intrusion Detection. In *Proceedings of the Fifth Computer Security Foundations Workshop* pp. 114-120.

[14] Helman, P. and Liepins, G., (1993). Statistical foundations of audit trail analysis for the detection of computer misuse, *IEEE Trans. on Software Engineering*, 19(9):886-901.

[15] Ilgun, K. (1993). USTAT: A Real-time Intrusion Detection System for UNIX. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*. pp. 16-28.

[16] Kohonen, T. (1995) *Self-Organizing Maps*. Berlin: Springer.

[17] Kumar, S. & Spafford, E. (1994) A Pattern Matching Model for Misuse Intrusion Detection. In *Proceedings of the 17th National Computer Security Conference,* pages 11-21.

[18] Kumar, S. & Spafford, E. (1995) A Software Architecture to Support Misuse Intrusion Detection. Department of Computer Sciences, Purdue University; CSD-TR-95-009

[19] Lunt, T.F. (1989). Real-Time Intrusion Detection. *Computer Security Journal Vol. VI, Number 1.* pp. 9-14.

[20] Mukherjee, B., Heberlein, L.T., Levitt, K.N. (May/June, 1994). Network Intrusion Detection. *IEEE Network.* pp. 28-42.

[21] Porras, P. & Neumann, P. (1997). EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In *Proceedings of the 20th NISSC.*

[22] Puketza, N., Chung, M., Olsson, R.A. & Mukherjee, B. (September/October, 1997). A Software Platform for Testing Intrusion Detection Systems. *IEEE Software, Vol. 14, No. 5*

[23] Ryan, J., Lin, M., and Miikkulainen, R. (1997). Intrusion Detection with Neural Networks. *AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAI Workshop (Providence, Rhode Island),* pp. 72-79. Menlo Park, CA: AAAI.

[24] Sebring, M., Shellhouse, E., Hanna, M. & Whitehurst, R. (1988) Expert Systems in Intrusion Detection: A Case Study. In *Proceedings of the 11th National Computer Security Conference.*

[25] Staniford-Chen, S. (1995, May 7). Using Thumbprints to Trace Intruders. UC Davis.

[26] Tan, K. (1995). The Application of Neural Networks to UNIX Computer Security. In *Proceedings of the IEEE International Conference on Neural Networks, Vol.1* pp. 476-481.

[27] Tan, K.M.C & Collie, B.S. (1997). Detection and Classification of TCP/IP Network Services. In *Proceedings of the Computer Security Applications Conference*. pp. 99-107.

[28] White, G.B., Fisch, E.A., and Pooch, U.W. (January/February 1996). Cooperating Security Managers : A Peer-Based Intrusion Detection System. *IEEE Network*. pp. 20-23.