INTRUSION DETECTION SYSTEM USING SELF ORGANIZING MAP

Liberios VOKOROKOS, Anton BALÁŽ, Martin CHOVANEC
Technical University of Košice, Faculty of Electrical Engineering and Informatics,
Department of Computers and Informatics, Letná 9, 042 00 Košice, Slovak Republic,
E-mail: liberios.vokorokos@tuke.sk, anton.balaz@stuba.sk, martin.chovanec@tuke.sk

SUMMARY

The goal of the article is to presents intrusion detections systems and design architecture of intrusion detection based on neural network self organizing map. In the report is described base problematic of neural network and intrusion detection system. The article further deals with specific design of intrusion detection architecture based on user anomaly behavior. A core of the designed architecture represents neural network SOM, which classifies monitored user behavior and determines possible intrusion of monitored computer system. Result of the designed architecture is simulations in real conditions. Acquired results of simulation assign expediencies of using neural network SOM in the intrusion detection systems.

Keywords: *Intrusion detection system, misuse behavior, anomaly behavior, self organizing map, learning phase, neural network*

1. INTRODUCTION

The goal of intrusion detection is to discover unauthorized use of computer systems. Existing intrusion detection approaches can be divided into two classes - anomaly detection and misuse detection. Anomaly detection approaches the problem by attempting to find deviations from the established patterns of usage. Misuse detection, on the other hand, compares the usage patterns to known techniques of compromising computer security. Architecturally, an intrusion detection system can be categorized into three types - host-based IDS, network-based IDS and hybrid IDS. Host-based IDS, deployed in individual host-machines, can monitor audit data of a single host. Network-based IDS monitors the traffic data sent and received by hosts. Hybrid IDS uses both methods.

Self-Organizing Map has been successfully applied in complex application areas where traditional method has failed. Due to their inherently non-linear nature, they can handle much more complex situations than the traditional methods. One of those problems represents intrusion detection by intrusion detection systems. These systems deal with high dimension data on the input, which is needed to map to 2-dimension space. Designed architecture of the intrusion detection system is application of neural network SOM in IDS systems, developed on the Department of Computer and Informatics, Technical University of Košice supported by VEGA 1/1064/04.

2. SELF ORGANIZING MAP

The Self-Organizing Map [9] is a neural network model for analyzing and visualizing high dimensional data. It belongs to the category of competitive learning network. The SOM Fig. 1 defines a mapping from high dimensional input data space onto a regular two-dimensional array of neurons.

In designed architecture is input vector with six input values and output is realized to 2 dimension space. Every neuron i of the map is associated with an n-dimensional reference vector $m_i\left[m_1,......,m_n\right]^T$, where n denotes the dimension of the input vectors. The reference vectors together form a codebook. The neurons of the map are connected to adjacent neurons by a neighborhood relation, which dictates the topology, or the structure, of the map. Adjacent neurons belong to the neighborhood Ni of the neuron i. In the SOM algorithm, the topology and the number of neurons remain fixed from the beginning. The number of neurons determines the granularity of the mapping, which has an effect on the accuracy and generalization of the SOM. During the training phase, the SOM forms an elastic net that is formed by input data. The algorithm controls the net so that it strives to approximate the density of the data. The reference vectors in the codebook drift to the areas where the density of the input data is high. Eventually, only few codebook vectors lie in areas where the input data is sparse.

The learning process of the SOM goes as follows:

1. One sample vector x is randomly drawn from the input data set and its similarity (distance) to the codebook vectors is computed by using Euclidean distance measure:

$$\left\|x - m_c\right\| = \min_i\left\{\left\|x - m_i\right\|\right\}$$

2. After the BMU has been found, the codebook vectors are updated. The BMU itself as well as its topological neighbors are moved closer to the input vector in the input space i.e. the input vector attracts them. The magnitude of the attraction is governed by the learning rate. As the learning proceeds and new input vectors are given to the map, the learning rate gradually decreases to zero according to the specified learning rate function type. Along with the

learning rate, the neighborhood radius decreases as well. The update rule for the reference vector of unit i is the following:

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)\left[x(t) - m_i(t)\right], i \in N_c(t) \\ m_i(t), i \notin N_c(t) \end{cases}$$

3. The steps 1 and 2 together constitute a single training step and they are repeated until the training ends. The number of training steps must be fixed prior to training the SOM because the rate of convergence in the neighborhood function and the learning rate are calculated accordingly.

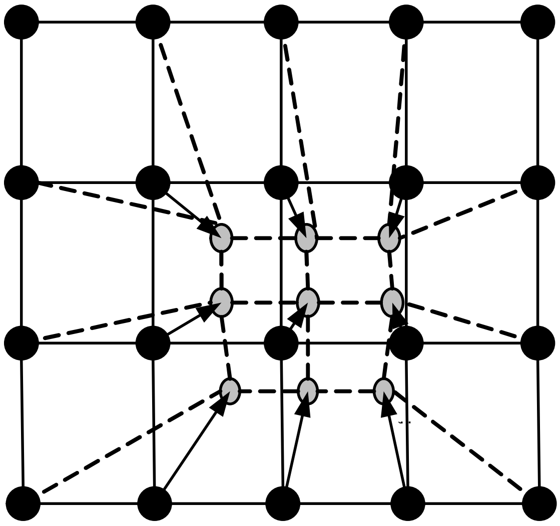

Fig. 1 General SOM topology

After the training is over, the map should be topologically ordered. This means that n topologically close input data vectors map to n adjacent map neurons or even to the same single neuron.

2.1. Mapping precision

The mapping precision measure describes how accurately the neurons respond to the given data set. If the reference vector of the BMU calculated for a given testing vector xi is exactly the same xi, the error in precision is then 0. Normally, the number of data vectors exceeds the number of neurons and the precision error is thus always different from 0. A common measure that calculates the precision of the mapping is the average quantization error over the entire data set:

$$E_q = \frac{1}{N}\sum_{i=1}^{N} \left\| x_i + m_c \right\|$$

2.2. Topology preservation

The topology preservation measure describes how well the SOM preserves the topology of the studied data set. Unlike the mapping precision measure, it considers the structure of the map. For a strangely twisted map, the topographic error is big even if the mapping precision error is small.

A simple method for calculating the topographic error:

$$E_q = \frac{1}{N}\sum_{i=1}^{N} u(x_x)$$

where $u(x_k)$ is 1 if the first and second BMUs of x_k are not next to each other. Otherwise $u(x_k)$ is 0.

3. INTRUSION DETECTION SYSTEMS

Today, there are generally two types of intrusion detection systems [1]: anomaly detection and misuse. Anomaly detection approaches attempt to detect intrusions by noting significant departures from normal behavior. Misuse detection techniques attempt to model attacks on a system as specific patterns, and then systematically scan the system for occurrences of these patterns Fig. 2. This process involves a specific encoding of previous behaviors and actions that were deemed intrusive or malicious. It is important to establish the key differences between anomaly detection and misuse detection approaches. The most significant advantage of misuse detection approaches is that known attacks can be detected fairly reliably and with a low false positive rate. However, the key drawback of misuse detection approaches is that they cannot detect novel attacks against systems that leave different signatures. So while the false positive rate can be made extremely low, the rate of missed attacks false negatives can be extremely high depending on the ingenuity of the attackers. As a result, misuse detection approaches provide little defense against novel attacks, until they can learn to generalize from known signatures of attacks. Anomaly detection techniques, on the other hand, directly address the problem of detecting novel attacks against systems. This is possible because anomaly detection techniques do not scan for specific patterns, but instead compare current activities against models of past behavior. One clear drawback of anomaly detection is its inability to identify the specific type of attack that is occurring. However, probably the most significant disadvantage of anomaly detection approaches is the high rates of false alarm. Because any significant deviation from the baseline can be flagged as an intrusion, it is likely that non-intrusive behavior that falls outside the normal range will also be labeled as an intrusion - resulting in a false positive. Another drawback of anomaly detection approaches is that if an attack occurs during the training period for establishing the baseline data, then this intrusive behavior will be established as part of the normal baseline. In spite of the potential drawbacks of anomaly detection, having the ability to detect novel attacks makes anomaly detection a requisite if future, unknown, and novel attacks against computer systems are to be detected.

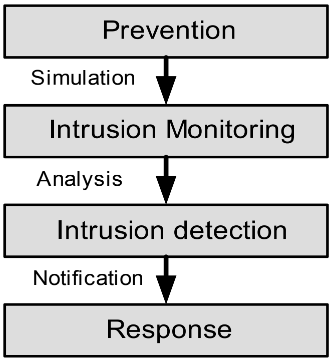

Fig. 2 Intrusion detection system activities

4. STRUCTURE AND ARCHITECTURE OF DESIGNED INTRUSION DETECTION SYSTEM

The designed intrusion detection system consists from the several components Fig. 3, proposed and implemented within research on the Department of Computers and Informatics, Technical university of Košice. The system has its core element – a sensor that is responsible for detecting intrusions. This sensor contains decision-making mechanisms regarding intrusions. Sensors receive raw data from three major information sources Fig. 3: own IDS knowledge base, syslog and audit trails. The syslog includes, configuration of file system, user authorizations and others. This information creates the basis for a further decision-making process. The sensor is integrated with the component responsible for data collection Fig. 4 an event generator. The collection manner is determined by the event generator policy that defines the filtering mode of event notification information. The collection manner is determined by the event generator policy that defines the filtering mode of event notification information.

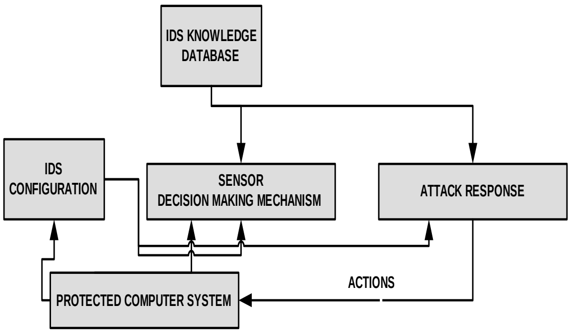

Fig. 3 Intrusion detection system

The collection manner is determined by the event generator policy that defines the filtering mode of event notification information. The event generator (operating system, network, application) produces a policy-consistent set of events that are a log or audit of system events, or network packets. This, set along with the policy information is stored in the protected system. The role of the sensor is to filter information and discard any irrelevant data obtained from the event set associated with the protected system, thereby detecting suspicious activities. The analyzer uses the detection policy database for this purpose. The latter comprises the following elements: attack signatures, normal behavior profiles, and necessary parameters. In addition, the database holds IDS configuration parameters, including modes of communication with the response module. The sensor also has its own database containing the dynamic history of potential complex intrusions (composed from multiple actions).

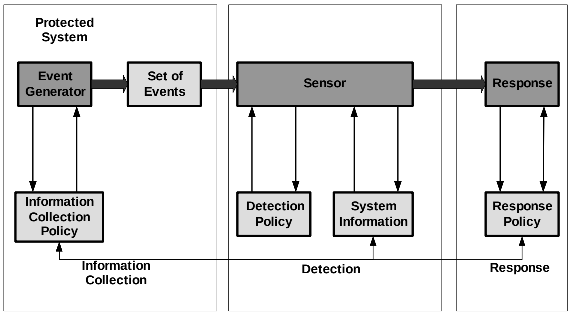

Fig. 4 Host based IDS

Intrusion detection systems can be arranged as either centralized or distributed. A distributed IDS consists of multiple Intrusion Detection Systems (IDS) over a large network, all of which communicate with each other. More sophisticated systems follow an agent structure principle where small autonomous modules are organized on a per-host basis across the protected network [8]. The role of the agent is to monitor and filter all activities within the protected area and depending on the approach adopted make an initial analysis and even undertake a response action. The cooperative agent network that reports to the central analysis server is one of the most important components of intrusion detection systems. DIDS can employ more sophisticated analysis tools, particularly connected with the detection of distributed attacks [9]. Another separate role of the agent is associated with its mobility and roaming across multiple physical locations. In addition, agents can be specifically devoted to detect certain known attack signatures. This is a decisive factor when introducing protection means associated with new types of attacks. IDS agent-based solutions also use less sophisticated mechanisms for response policy updating. The designed system IDS architecture is host based. In this case all components of the system are applied directly on the secured computer system.

5. SOM IMPLEMENTATION TO INTRUSION DETECTION SYSTEM

The goal of the proposed architecture is to investigate effectiveness of application a neural network SOM Fig. 6 at modeling user behavioral patterns so they can distinguish between normal and abnormal behavior. In order to model user behavior identified and isolated the system logs that were required as sources of information for the networks. These logs being common log data provided the required user activity information from where system derived the following behavioral characteristics which typifies users on the system:

- User activity times - The time at which a user is normally active.
- User login hosts - The set of hosts from which a user normally logs in from.
- User foreign hosts - The set of hosts which a user normally accesses via commands on the system (FTP hosts).
- Command set - The set of commands which a user normally uses.
- CPU usage - The typical CPU usage patterns of a user.
- Memory usage – The typical usage of memory for a user.

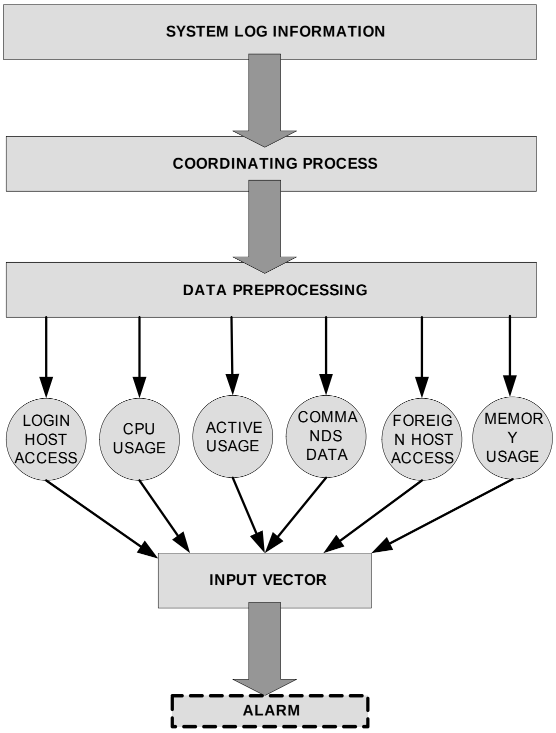

Fig. 5 Structure of an Automated User Behavior Anomaly Detection System

Figure 5 illustrates how a complete system for the detection of user behavioral anomalies is structured. The coordination process is responsible for channeling system information to the neural networks. Each of the behavioral characteristics are both modeled by a SOM network, as well as checked by a limited static rule filter for easy breaches of security. Data acquired from the system logs is required to filter through input data preprocessor Fig. 5, separating selected data from audit data. The input to the neural network Fig. 6 represents data vector consisting from data controlled on the monitored system. Before input vector processing it is needed to normalize input data. The input to neural network is data vector, which consists from six properties representing User activity times, User login hosts, User foreign hosts, Command set CPU usage and Memory Usage. According to large numbers of variations of this data it is necessary to normalize every input vector to be value in range of values [-1, 1]. This range comes out from the previous applications of neural network to system IDS realized within research activity on the Department of Computers and Informatics in Košice [11]. This normalization is more suitable for implementation in proposed SOM network. The architecture uses normalization given by:

$$nv[i] = \frac{v[i]}{\sqrt{\sum_K v[k]^2}}$$

Where *nv[i]* is the normalized value of feature *(i)*, *v[i]* is the feature value of *i*, and *K* is the number of features in a vector. The processing realized by the SOM network consequently produces results for every user characteristic gives as input to the SOM network. Expected network reply is the value close to-for user, which behavior does not divert from normal behavior.

If the value for given user exceeds specified threshold value obtained through the SOM network representing its intrusion behavior denotes raising alarm. If the output value of network is above specified threshold value, alarm is raised. It is necessary to remark that basic request for this detection mechanism is to setup threshold value to specific system whereby make it possible to adapt sensitivity directly to computer system.

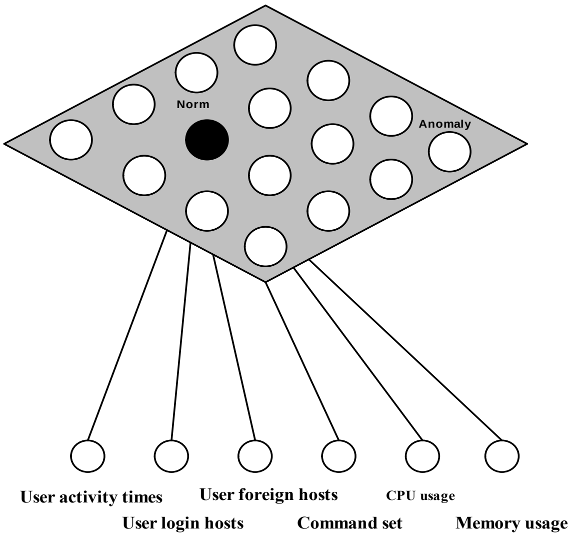

Fig. 6 Form of the designed SOM architecture

6. DESIGNED SYSTEM SIMULATION

The results were obtained on the department server KPI Technical university of Košice. This computer system represents system with average 32 users. The goal of the simulation proposed architecture was to verify application possibility of neural network SOM in the IDS systems. Collecting of essential information from single controlled points lasts 2 days. As data basis was used operating system Linux with modified system of audit records. This information serves as base point for following user behavior classification on the monitored system. Next the neural network SOM was created and trained, which serves as the core of the IDS system. The results Fig. 7 show input vectors classification, which represents behavior and its mapping to particular neurons, which form single possible user behavior states. Form states as intrusion – Intrusion, possible intrusion – Intrusion?, normal – Norm. From the test result SOM network represents suitable core for IDS systems.

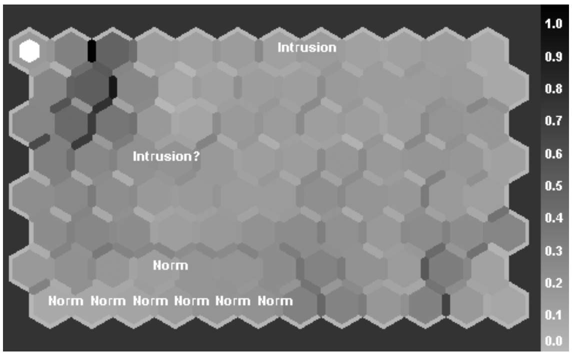

Fig. 7 Classification of user behavior

7. CONCLUSION

The goal of this article was to design architecture of the system detecting intrusion based on user anomaly behavior. Classification module of the proposed architecture is self organizing map. Input to network represents multidimensional input vector describing actual system state and activities taken on the controlled computer system. Neural network output is value, which represents possible state of system intrusion. The proposed architecture is developed within project VEGA 1/1064/04 on the Department of computers and informatics, Technical university of Košice.

Supported by VEGA 1/1064/04

REFERENCES

[1] Wang, J.: A network intrusion detection system based on the artificial neural networks, Shanghai ACM 2004, ISBN 1-58113-955-1

[2] Anup, K. Ghosh: Learning Program Behavior Profiles for Intrusion Detection, Proceedings 1st Workshop on Intrusion Detection and Network Monitoring, 1999.

[3] Vokorokos, L.: Data Flow computer architecture principles. Monograph. Copycenter, s.r.o, Košice, 2002. ISBN 80-7099-824-5

[4] Bishop, C. M. 1995. Neural Networks for Pattern Recognition. Oxford: Clarendon-Press.

[5] Axelsson S., “Intrusion Detection Systems: A Survey and Taxonomy”. Technical report 99-15, Department of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, March 2000.

[6] Debar H., Becker M., Siboni D., “A Neural Network Component for an Intrusion Detection System”. Proceedings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA May 1992

[7] Rhodes, B., Mahaffey, J., Cannady, J.: “Multiple Self-Organizing Maps for Intrusion Detection”. Proceedings of the NISSC 2000 conference.

[9] Kohonen, T. 1995. Self-Organizing Maps, volume 30 of Springer Series in Information Sciences. Berlin, Heidelberg: Springer. (Second Extended Edition 1997).

[10] Lane, T., and Brodley, C. E. 1999. Temporal sequence learning and data reduction for anomaly detection. ACM Transactions on Information and System Security 2(3):295—331.

[11] Baláž, A., Ádám N.: Intrusion Detection System Using Multilayer Perceptron, 6th PhD Student Conference and Scientific and Technical Competition of Students of Faculty of Electrical Engineering and Informatics Technical University of Košice.

[12] Vokorokos, L, Ádám, N., Baláž, A.: Flexible Platform for Neural Network Based on Data Flow Principles, 6th International Symposium of Hungarian Researchers on Computational Intelligence, Budapest, November 18-19, Budapest Tech, 2005, ISBN 963 7154 43 4

[13] Vokorokos, L., Baláž, A., Ádám, N., Petrík, S.: Dataflow Distributed Database Systems, The 16th international DAAAM symposium, Intelligent Manufacturing & Automation, 19-22nd October 2005, Opatija, Croatia, SSN 1726-9679

BIOGRAPHIES

Liberios Vokorokos (prof., Ing., PhD.) was born on 17.11.1966 in Greece. In 1991 he graduated (MSc.) with honours at the department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at Technical University in Košice. He defended his PhD. in the field of programming device and systems in 2000; his thesis title was "Diagnosis of compound systems using the Data Flow applications". He was appointed professor for Computers Science and Informatics in 2005. Since 1995 he is working as an educationist at the Department of Computers and Informatics. His scientific research is focusing on parallel computers of the Data Flow type. In addition to this, he also investigates the questions related to the diagnostics of complex systems. Currently he is a member of the State Examination Committee in the field Computing engineering and Informatics. His other professional interests include the membership on the Advisory Committee for Informatization at the faculty and Advisory Board for the Development and Informatization at Technical University of Košice.

Anton Baláž was born in Sobrance, Slovakia, in 1980. He received the engineering degree in Informatics in 2004 from Faculty of Electrical Engineering and Informatics, Technical University of Košice. Since 2004 he is PhD. student at the Department of computers and informatics FEI TUKE and his scientific research is focused on intrusion detection systems.

Martin Chovanec was born in Lučenec, Slovakia, in 1982. He received the engineering degree in Informatics in 2005 from Faculty of Electrical Engineering and Informatics, Technical University of Košice. Since 2005 he is PhD. student at the Department of computers and informatics FEI TUKE and his scientific research is focused on network security and encryption algorithms.