INTRUSION DETECTION SYSTEM BASE ON NEURAL NETWORK

[*] SALAM A. ISMAEEL, [**] WESAM A. SHOKER, and [***] THAHA A. TAHA
[*] Computer Man College for Computer Studies, Khartoum / Sudan salam.ismaeel@gmail.com
[**]wesamabed79@yahoo.com [***]thaha.taha@gmail.com

ABSTRACT

An anomaly based intrusion detection systems needs to be able to learn user's or system's behavior because users and systems behavior changes over time in today's dynamic environment. In this research experimenting with user's behavior will used as parameters in anomaly intrusion detection. The proposed intrusion detection system is uses a back propagation neural network to learn user's behavior.

The neural network will check if it able to classify normal behavior correctly, and detect known and unknown attacks without using a huge amount of training data.

The experiments were separated into three parts. The first preliminary experiment was conducted to see when the neural network was properly trained to classify sessions correctly. In this experiment, both known and unknown attacks were used. The next experiment was conducted to test the neural network with a small traffic, known and unknown attacks. Unknown attacks are the most threatening attacks, because these attacks are not known or not expected. In the final experiment, the classification rate was 82% on known attacks.

Keywords: *Intrusion detection; Anomaly IDS; Back propagation training algorithm; Neural Network*

1. INTRODUCTION

The intrusion threats are bigger than ever. This is because of the fact that there are applications available on the Internet that give people with almost no computer experience the possibility to break into computer systems. Because of this, attacks against computer systems and networks have increased significantly in recent years [4].

Most of the prevalent Internet attacks today can be stopped or mitigated proactively with little fear of false attacks. But what about the new and unknown attacks? It is hard to protect yourself against something you do not know anything about [1].

The aim of this work is to design an efficient system to detect intrusions caused by outer and insider intruders in host-based system and network-based system. The proposed system monitors all clients' behavior in the network and tests them to check if they are normal or abnormal. The proposed system consists of two stages, the first stage is monitoring all events that happen and analyzing them, the second stage is to detect intrusions. The detection operation combines anomaly intrusion detection and misuse intrusion detection to detect more types of intrusions.

Back propagation neural network used to learn the normal network traffic and detect the abnormal traffic. Misuse detection matches the current traffic with several signatures.

2. IDS ARCHITECTURE

The proposed system is independent of any particular target system, thereby providing a design for specified-purpose intrusion detection system. This system combines the two distinct intrusion detection approaches, anomaly and misuse. Combining these two approaches enables bypassing drawbacks that appear in each approach.

The architecture of the proposed intrusion detection system (IDS) is shown in Fig. 1 and is stated in the following points [1, 3]:

1. An audit record is created. This occurs when an action happens, such as open file.
2. The detection engine searches for anomaly behavior comparing the current data with the historical data.
3. Anomaly record is generated when abnormal behavior is observed. The anomaly record is forwarded to a number of different sub-systems for security officer, response, and storage.
4. The security officer is notified through visual method such as showing the message as alarm.
5. A response is generated. Responses include actions such as shutdown the target, restart the target or notify the security officer.
6. Store the anomaly record to use it in expecting new type of intrusion.
7. Reports can be generated and forwarded to the security officer.

As any IDS, the proposed system can be divided into three parts; monitoring system, detection model, and alerting model. Each part of the proposed system will be discussed in detail in the following subsections.

3. MONITORING SYSTEM

A trace policy describes the valid operation sequences of a single program execution, multiple program executions, a user, a group of users, a host … etc. The entity or entities are collectively called a monitored subject, or simply a subject. Therefore, a subject could be a distributed process, a group of distributed processes. A host refers to all processes running on the host. A user refers to all processes that are owned by the user. Monitoring a subject means analyzing the sequence of operations performed by the subject [5].

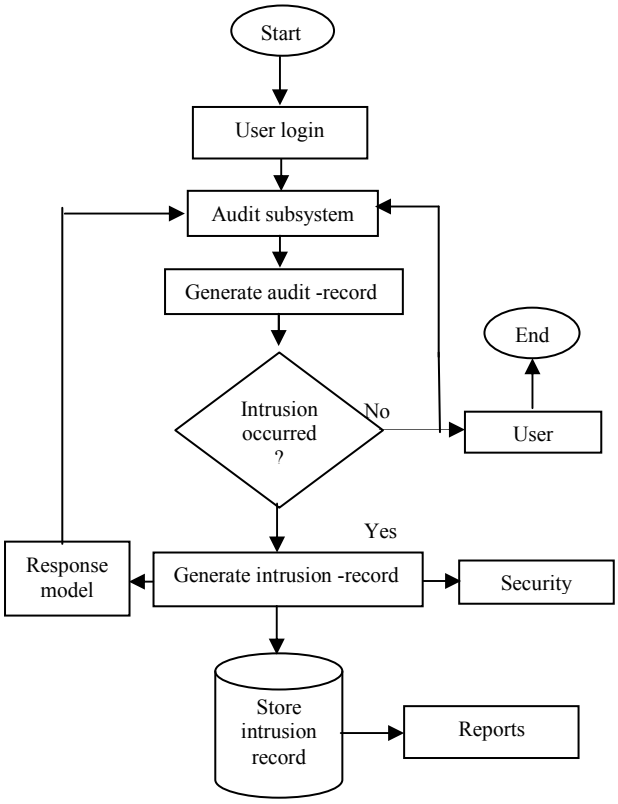

Figure (1): The architecture of the proposed system

The execution trace of a subject is the time ordered sequence of operations performed by the processes forming the subject. Each process has its own traces and the subject trace is the merge of the individual traces.

The goal is to motivate the specification language for specifying trace policies. The following four aspects of program behavior are security-relevant considerations:

1. Accesses of system object (e.g., files, folders).
2. Sequencing should do operations in some particular order.
3. Synchronization among processes of a parallel program or among processes of related programs, access to shared data object.
4. Race conditions.

The proposed monitoring system is based on the hypothesis that security violations can be detected by monitoring a system's audit records for abnormal patterns of system usage. The model includes profiles for representing the behavior of subjects with respect to objects.

Host based monitor activities are normally executed only by an administrator. Operating systems log any event user accounts are added, deleted, and renamed. Host-based IDS can detect an improper change as soon as it is executed. Host-based IDS can also audit policy changes that affect what systems track in their log.

The proposed system does not contain any special features for dealing with complex actions that exploit a known or suspected security flaw in the target system; indeed, it has no knowledge of the target system's security mechanisms or its deficiencies. By detecting the intrusion, however, the security officer may be better able to locate vulnerabilities.

The proposed system was build as an administrator for a LAN network with a two personal computer connected to server. These two computers will operate under windows, the first step is start up the server, no network services without server. Each client registers himself as an employee. The employee has to fill registration form which asks him about many personal aspects. This personal information is decided previously.

After that, employee returns to his office to get himself ready for the next step through which the system learns more about him, as shown in Fig. 2.

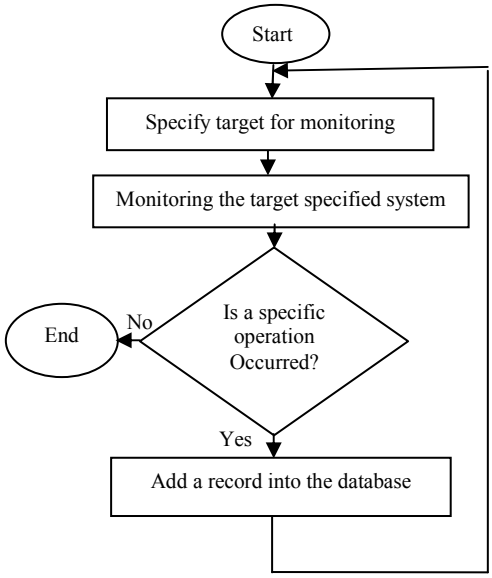

Figure (2): Proposed monitoring system

Monitoring Algorithm can be characterized as follows:

Algorithm 1:

```
1. Start when the computer is turned-on.
2. Specify the target system that wants to
   monitor it.
3. Monitoring the specified target system.
4. Check if any operation of (delete, create,
   rename, and open) on the file or folder has
   occurred then add a record in database, go to
   step 3
5. End
```

4. DETECTION MODEL

There are two categories of intrusion detection system. They are called network-based IDS and host-based IDS. Networks have made computer systems easily accessible from remote location. This allows legitimate users to access computer systems and information on those computer systems and this may lead to a large number of intrusions.

A single intrusion of a computer network can result in the loss or unauthorized utilization or modification of a large amount of data and cause users to question the reliability of all the information on the network. An important shortcoming of IDS is that can detect when an event is unusual, which may or may not indicate an attack.

This model acts the heart of the proposed system, it is consists of two phases: Encoding phase and Detection phase.

4.1 ENCODING PHASE

The main function of this phase is reading the records from monitoring system database and encoding it. The input files to the neural network must be encoded. These files must be in decimal format, therefore all log files must be converted to decimal format, as shown in the following algorithm.

```
Algorithm 2:
1.  Start while database (DB) contain
    operation record
2.  If  rec. no  from MSDB  larger  than
    zero, then go to step 3
        Else go to step 11.
3.  Read record from MSDB
4.  Set coding  = 0 , ×=0
5.    × = ×+1
6.   If × <  field no  then go to step 7
        Else go to step 2
7. Execute Query
  Query = "select target field, code, ser,
    max (ser). As m ser from code table where
    target field = record, field (x)"
8. Check if rec. on. Of Query larger then
    zero then go to step 9
        Else   go to   step 10
9.  Coding  (× + ser ) = Query (code )
10.  x = × + Query (m ser ) and go to step 5
11. End
```

4.2 DETECTION PHASE

The purpose of this phase is to detect intrusions by using artificial neural network. In this phase, two kinds of intrusions will be detected, anomaly and misuse intrusion.

A. ANOMALY DETECTION

This is the first detection mechanism in the proposed system. The structure of the anomaly detection is shown in Fig.3.

- **Machine learning component:** This part uses back propagation neural network to learn normal patterns of system behavior. This normal behavior is stored in profiles. This allows the system to adapt to new environment.
- **Anomaly intrusion detection module:** This module extracts patterns of an observed audit trail and compares these new patterns with the normal patterns. If the similarity of the sets of patterns is below a specified threshold, the system alarms of an intrusion.
- **Decision- making module:** This will decide whether or not to activate anomaly intrusion detection and integrates evaluation results provided by the anomaly intrusion detection.
- **Communication module:** It is the bridge between the intrusion detection and decision- making module.
- **Intrusion detection sentries:** pre-process audit data and send results to the communicate module.

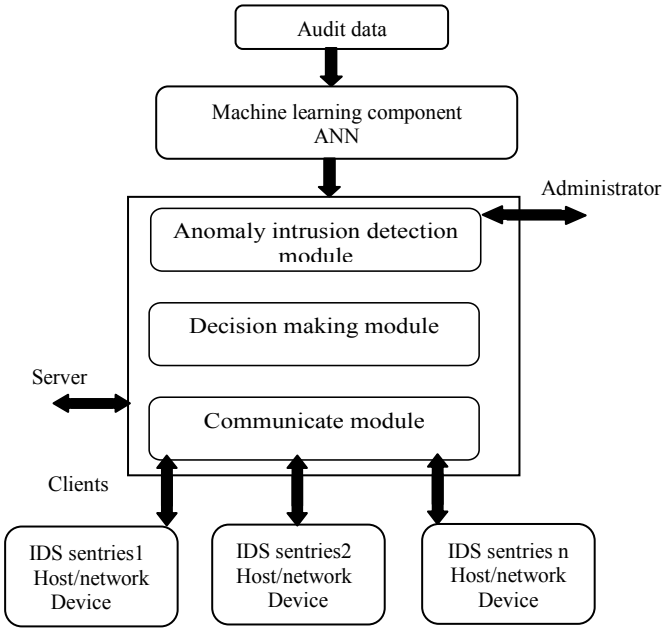

Figure (3): Structure of the anomaly detector

Algorithm (3) shows the steps of anomaly detector.

Algorithm 3:

```
Begin
 Monitoring system turn on.
Select specific fields from monitoring system
data base and arrange them in vectors
While .T. DO
  Begin
  Read vector from monitoring system data base.
  Apply encoding algorithm.
  Input vector to the designed NN
  If  the  output  of  NN  is  (1)  then  Mark  the
    specified Vector as normal
  Else   mark the specified vector as abnormal
 If no more vector found then
Exit
End (while)
End
```

Figure 4 shows the designed neural network (NN) that has been used in the proposed system. Inputs values to the NN are coded as x_1, x_2… x_9. Where:

x_1: the destination name.
x_2: create event.
x_3: delete event.
x_4: rename event.
x_5: open event.
x_6: the attribute of folder (hidden or not).
x_7: the file

x_8: the folder.
x_9: the event in time.

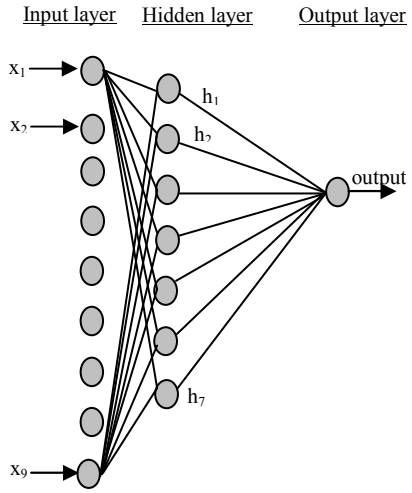

Figure (4): Neural network architecture used for anomaly detector

The main task of this phase is to identify intrusion patterns by considering the threshold that was computed in the neural network with back propagation training algorithm. The detector was designed using neural network, which utilizes back propagation architecture that consists of (9) input nodes, the hidden layer consist of (7) nodes and one output node.

The NN is designed to provide an output value (0) to indicate a misuse attack, and if output value is larger than error then anomaly intrusion has occurred. If output value is smaller than error then behavior is normal.

The number of hidden layers and the number of nodes in hidden layer is determined based on the process of trial and error. Algorithm (3) shows the steps of anomaly detection.

The hidden layer consists of seven nodes, because this number of nodes gives good rate of accuracy in learning operation. If the number of nodes in hidden layer is increased then the accuracy will be increased but this number of nodes in hidden layer needs more time because of the computations between the nodes.

The output layer consists of one node, because we need either yes or not. It means intrusion or not-intrusion.

B. MISUSE DETECTOR

This type of detection involves a comparison of user's activities with the known behaviors of attackers. This detection mechanism matches the current behavior with a signature of known attacks. A signature is a pattern that we want to look at in event records. The structure of the misuse detector is shown in Fig. 5.

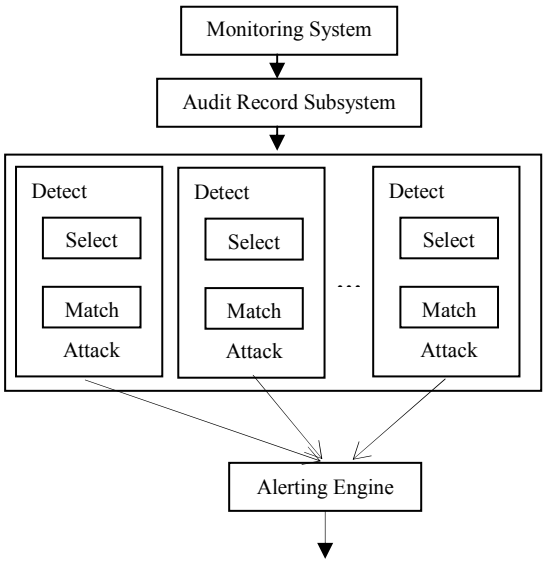

Figure (5): The structure of the misuse detector

The algorithm of the misuse detector is illustrated in algorithm (4).

Algorithm 4:

```
Begin
    Monitoring system turn on.
   Select specific fields from monitoring
system data base and arrange them in    vectors
    While .T. DO
       Begin
          Select specific features from records
            in MSDB depending on the signature
            of the attack
          Match the selected features with the
            features of the attack signature
         If the two sets of features match then
             Misuse attack is found
         Else
              Mark the event as normal
         If   no more attacks found then
              Exit
         Else
             Check another attack type
      End
End
```

5. ALERTING MODEL

This is the last stage in the proposed system. This stage is concerned with deciding if an event or a set of events is an intrusion or not. It receives the output of anomaly detection or misuse detection part and then gives the result in a report. This report shows if the event is an intrusion if it is normal. The report also specifies the source that causes the abnormal behavior and the data and time of it. This model is important, since it is used to stop an intruder. Alerting model has many levels of reactions against the intruder, using any one of them depends on the nature of attack.

This model may be enough in showing the warning messages or locking some bottoms in the keyboard or freezing the IP that causes that attack.

6. EXPERIMENT RESULTS

The experiments were conducted in two parts, the preliminary experiment and final experiment:

The preliminary experiment: In this experiment, the number of hidden units was determined. Where, both known and unknown attacks were used in the same file. Simulation results show that with 5 hidden units, and 100 iterations, the detection rate was 86%. The results from this experiment gave the background for choosing the number of hidden units and iterations used for the training of the neural network in the final experiment.

This test was just a brief testing from several tests and for each test changed the number of iterations for the neural network with 100 iterations.

The final experiment: The testing was divided into normal traffic, known attacks, and unknown attacks. The classification rate of normal is 82%, but for the known attacks is 82% and the classification rate of unknown attacks is 80%.

The reaction is locking the IP that causes the abnormal behavior, this IP can not do any thing because the locking enforced on it. Locking operation performed by master user or administrator. The final report can display the number of occurrence of this attack.

The final report can be divided into four types of reports (IDS general report, IDS monitoring report, IDS general attack forms report, and IDS summary report).

1. **IDS General Report:** Contains the IP of attacker, type of attack, date, time, type of event, class of user, file or folder, category, and diagnosis percentage.
2. **IDS Monitoring Report:** Contains the file or folder name that acts the destination of event, type of event, time of event, and IP of attacker.
3. **IDS General Attack Forms Report:** Contains the IP of attacker (source of attack), the IP of destination attack, type of attack, attack form, date of attack, and time of attack.
4. **IDS Summary Report:** Contains the IPs of attackers, number of misuse intrusions, number of anomaly intrusions, number of create event, number of delete event, number of rename event, number of open event, type of destination attack(file or folder), first date, last date, first time, last time, and type of extension file (.txt, .sys, .avi, and custom) .

5. CONCLUSIONS

From the excremental results, the following conclusions are drawn:

1. Using the machine learning component of IDS architecture allows the system to adopt into new environments. This makes the proposed system able to detect both of attack anomalies and misuse attack.
2. The system allows intrusions to be detected without knowledge about these flaws in the target system that allowed the intrusion to take place, and without necessarily observing the particular action that exploits the flaw.
3. The modular characteristics of the architecture allow it to be easily extended, configured and modified, either by adding new features, or by replacing features when they need to be updated.
4. Using neural network in anomaly detection shows that neural network can learn the characteristics of normal behavior and identify instances that are unlike normal behaviors that are anomalies. The training of the neural network requires a very large amount of data to ensure that the results are accurate.

REFERENCES

[1] Berkeley R.," A Neural Network Based Intelligent Intrusion Detection System," *M.Sc. Thesis, Information Technology Dept. Griffith University, June 2003.*

[2] Escamilla T., *Intrusion Detection: Network Security Beyond the Firewall,* John Wily and Sons, Ins. 1998.

[3] Proctor P. E., *The Practical Intrusion Detection Handbook,* Prentice-Hall, Inc 2001.

[4] Stallings W., *Cryptography and Network Security: Principles and Practices*, 3rd Edition, Prentice Hall, 2003.

[5] Stallings W., *Networks Security Essentials: Application and Standard,* Prentice-Hall, Inc.2000.