Artificial Neural Network for Anomaly Intrusion Detection

Lixin Wang
9982688
lwan060@ec.auckland.ac.nz

Abstract

Since the advent of intrusion detection system (IDS) in the early 1980s, IDS has been suffering many problems until now. The detection of novel attacks and lower rate of false alarms must be realized in successful IDS. Misuse detection compares data against predefined patterns usually collected by an IDS signature database. It is hard for misuse detection to detect even slightly variation of known attacks. Anomaly detection finds attacks by using deviations from the normal behavior. Although anomaly detection can detect novel attacks, it cannot identify specific type of attack and entail the high rates of false alarms. In this paper, we point out the weakness of the previous methods such as statistical analysis and rule-based system in intrusion detection. In order to detect unknown attacks and avoid malicious hiding intrusion, artificial neural network for anomaly detection was introduced.

1 Introduction

In this paper, we introduce the intrusion classification based on its manifestation and its location. We focus on the study of anomaly detection. Process-based anomaly detection has more advantages over single user based anomaly detection by generalizing and abstracting each user's individual behaviors. Hiding intrusion is a huge threat to intrusion detection technology. The intrusion can be hidden by maliciously fabricating the attack's manifestation to be normal. Artificial neural network (ANN) was introduced to defeat malicious hiding intrusion by monitoring the program's internal states. The black box nature of ANN increase the instability the training process of ANN, but at the same time the black box of ANN can approximate the program's internal states even if the program code is unavailable. However, under such conditions the internal states of the program are extremely difficult to fabricate. ANN can generalize from

previous behavior to recognize future unseen behavior and the classification ability of ANN can be used to detect even slightly different intrusions.

2 Intrusion and intrusion detection

In order to detect intrusion, data collection plays an important role. On the one hand, information can be collected through operating system or applications. On the other hand, network based data collection is to detect intrusions by monitoring network traffic. Intrusion detection can also be classified into two fields: misuse detection and anomaly detection.

2.1 Intrusion manifestation

An intrusion is an unauthorized attempt to access, manipulate, modify, or destroy information, or to render a system unreliable or unusable [1]. In order to detect intrusions some source of information in which the intrusions is manifest must be observed. Intrusion can manifest themselves in many ways. Intrusive activities can manifest either from the operating systems or from the network traffic. In this sense, intrusion can be classified into host-based intrusion and network-based intrusion. On the other hand, intrusion manifestations can be regarded as the signals to be detected no matter where they are from. Intrusion detectors justify the intrusive behaviors based on either signal or noise characterizations. Again, intrusion can be classified into anomaly-based intrusion and signature-based intrusion. In [1, 2, 5 and 6], their classifications define misuse intrusion as a replacement of signature-based intrusion. Intrusion classifications are based on the intrusion manifestation and its data collection locations.

2.2 Host-based detection vs. network-based detection

Intrusion detection tools can be classified into network-based or host-based intrusion detection. Host-based systems analyze data from the operating system or applications subject to attack. Network-based systems look for sign of intrusions from network traffic being monitored.

2.2.1 Host-based detection

Modern operating systems provide auditing, logging and performance monitor to detect intrusion. Most host-based systems collect data continuously as the system is operating, but periodic snapshots of the system state can also provide data that has the potential to reveal unexpected changes [4]. Anyway, host-based detection can not select auditing to detect intrusion owing to the lack of necessary information about the operating system. Unselective logging of messages actually may incur extra auditing overhead and analysis burdens. And selective logging is hard to determine without required knowledge and computation.

2.2.2 Network-based detection

Network-based data collection has the advantage that a single sensor, properly placed, can monitor a number of hosts and can look for attacks that target multiple hosts [4]. With the ease of construction, network monitoring is introduced in many commercial intrusion detection systems.

2.3 Anomaly detection vs. misuse detection

Anomaly detection is based on the assumption that misuse or intrusive behavior deviates from normal system use. Misuse detection seeks to discover intrusions by precisely defining the signatures ahead of time and watching for their occurrence.

2.3.1 Anomaly detection and hiding intrusion

2.3.1.1 Anomaly detection

Anomaly-based intrusion detectors take unusual or abnormal patterns as intrusions. The detectors must baseline the normal pattern of the program being monitored, and then use deviations from this baseline to detect intrusions. Anomaly detection is based on two assumptions: Firstly, the anomaly systems are necessarily different form non-intrusive activities at some level of observation. Secondly, intrusions will be accompanied by manifestations that are sufficiently unusual so as to permit detection [4]. In [6] anomaly detection assumes that intrusions are highly correlated to abnormal behavior exhibited by either a user or an application, so abnormal behaviors

manifest themselves in either user level or application level. One drawback of anomaly detection approaches is if the well-known attacks match the established profile of a user, anomaly detectors may not differentiate these attacks. Another drawback is their vulnerability to an intruder who breaches the system during the training period. Then the misled anomaly detector may interpret intrusive events as normal system behaviors. At the same time it is difficult for an anomaly detector to classify or name specific attacks. Perhaps the most obvious disadvantage of anomaly detection is the high rate of false alarms. A high false positive rate may result from a narrowly trained detection algorithm. On the opposite, a high false negative rate may result from a broadly trained anomaly detection approach.

2.3.1.2 Process-based intrusion detection

Intrusion detection systems can analyze either network traffic or host system logs. As we know, most computer security violations are made by misusing programs. To increase the detection ability and decrease the false alarm rate, process-based intrusion detection is introduced in the field of intrusion detection. Process-based intrusion detection focuses on system process because attacks against computer systems are in fact attacks specific software programs [6]. Process-based detectors analyze the behavior of executing processes for possible intrusive actions. When a program is misused its behavior will deviate from its normal state. By analyzing the usage or misusage of specific software programs, computer-based intrusion can be tracked at a finer grain of resolution. On the other hand, single user based detector can not detect anomalous behaviors across the entire user group. There exists distinctive difference in the regularity of the data from user to user. Process-based detector adds a layer of abstraction so that anomalous behavior can be detected irrespective of individual user's behavior. Most process-based intrusion detectors are based on anomaly detection. A specific process profile is built during the training phase of intrusion detectors by capturing the process's system calls. An intruder can mislead the training process in a user

profiling system, but the intruder can actually be defeated by a process-based detector. Because the process-based detector has the ability to generalize from each user's behavior, any specific behavior under a specific user can be summarized to a high process level. Two possible approaches to monitoring process behavior are: instrumenting programs to capture their internal states or monitoring the operating system to capture external system calls made by a program. The latter option is more attractive in general because it does not require access to source code for instrumentation [1].

2.3.1.3 Hiding intrusion

Tan et al in [7] demonstrate convincingly that their attacks can be hidden and then they extend their argument by saying: "We speculate that similar attacks are possible against other anomaly based IDS and that results have implications for other areas of information hiding". This paper addresses the assumption of anomaly detectors that intrusions cause anomalous manifestations. The authors believe that this assumption has caused a severe consequence that is the underlying causes and characteristics of the anomalous behaviors can not be justified in a correct way. A modified intrusive activity with anomalous manifestations can be undifferentiated from arguably normal activities. This paper identified a weakness (blind spot) in Stide and exploited it using several simple and well described attacks downloaded from the Internet. The intrusion then can be hidden by either making the attack's manifestations appear normal or finding a blind spot to hide it in. It can be argued that their result is convincible to us. This novel method for hiding intrusion has several limitations as below: Firstly, Stide is an open source anomaly detector but not all other IDSs are. Their approach requires the attacker to understand intimately the weaknesses of Stide. Secondly, the attacked programs (lpr and sendmail) are also open source. And only the programs' external system calls are captured. This is not true for most of commercial programs in which source code can never be available to the public. On the other hand, the programs' internal states by code instrumentation can also be utilized to prevent the hiding

intrusions. It becomes more difficult for the intruder to manipulate the manifestations of both the external system calls and internal states to avoid being detected. Even if the source code of the monitored program is not readily available, neural network can be trained to approximate the internal states of the monitored program for detection of misuse. (See Figure 1)

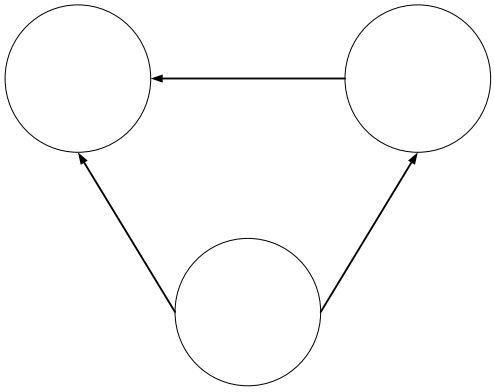

Figure 1

2.3.2Misuse detection

The significant advantage of misuse detection is that known attacks can be detected with a lower false positive rate. Obviously, misuse detection cannot detect novel attacks against systems with different signatures. If an appropriate abstraction can be found, signature-based systems can identify previously unseen attacks that are abstractly equivalent to known patterns [4]. To detect unseen attacks, DIDAFIT in [8] uses a fingerprint technique that can summarize signatures to help identify illegitimate SQL statements. Signature is called fingerprint in DIDAFIT. Over-summarization may result in high rate false positives. As new security vulnerabilities in software are discovered and exploited every day, misuse detection methods is not flexible for defeating malicious attacks.

3 Current approaches to anomaly intrusion detection

3.1 Statistical anomaly detection

An anomaly detector based on statistical methods is constantly monitoring the deviance of the current behavior profile from the normal behavior profile. Only normal training data is learned by the anomaly detector. The anomaly detector extrapolates anomalous behaviors through the low probability of the behaviors. False positives and negatives can be generated due to the inadequacy or insensitivity of the statistical measures chosen. Another concern on statistical method is whether enough

training data can be collected or not. Part of the problem is that neither the noise characteristics (normal usage) nor the signal characteristics (intrusions) have been adequately studied by using statistical methods [4].

3.2 Expert system shells anomaly detection

Most current methods to the process of detecting intrusions utilize some form of rule-based analysis [2]. An expert system is a computer system that emulates the decision-making ability of a human expert. Expert system is a branch of AI that makes extensive use of specialized knowledge to solve problems at the level of a human expert. Expert systems are the most common form of rule-based intrusion detection approaches. The internal structure of rule based system is described in Figure 2.

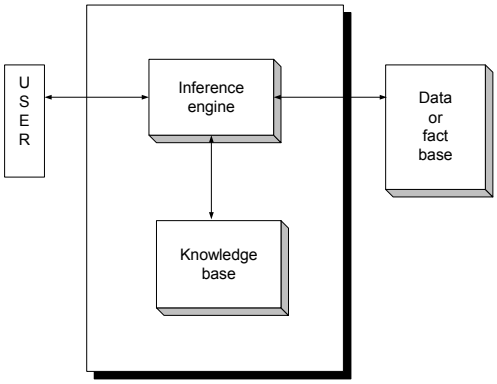

Figure 2

The user supplies facts or other information to the expert system and receives expert advice or expertise in response. Internally, the expert system consists of two main components. The knowledge base contains the knowledge with which the inference engine draws conclusions. The inference engine consists of a set of rules that encode the knowledge of a human expert. These rules are used by the system to draw conclusions about the anomalous intrusions. These conclusions are the expert system's responses to the user's queries for expertise. 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 attack. Expert systems are especially suitable for the problems which require significant human expertise for their solution in a specific domain. Not everyone in that domain is an expert. An expert system performs better than a human. An expert system will never get tired.

3.2.1 Disadvantages of expert system for intrusion detection

Knowledge acquisition bottleneck is a major obstacle for constructing a robust expert system. The expert systems need to be maintained frequently so as to update its ability to detect new intrusions. This updates include both the rule-base and knowledge-base. The lack of maintenance will severely degrade the performance of expert systems. When trying to improve the detection ability of expert systems, increasing the level of abstraction of the rule-base may result in high rate of false negatives. This trade-off can never be avoided by expert systems. A win-win solution may be needed. Rule-based systems only focus on the occurrence of individual intrusions. The state transitions in a sequential attack cannot be detected by a rule-based system. In [4] the author did not directly state this point. But in the taxonomy of anomaly-based intrusion detector table, we notice that rule-based system was classified into the category which can only cope with non time series attacks. This limitation prevents rule-based system from detecting a series of sequential attacks with time feature. This also makes hiding intrusions possible under rule-based system. Heuristics knowledge can provide valuable shortcuts that can reduce time and cost. But sometimes it is not that intelligent for expert systems because expert system cannot generalize knowledge to improve detection ability. We may need a self-learning system which can generalize the past behaviors of the system.

4 Artificial neural network

The goal of ANN for intrusion detection is to be able to generalize form incomplete data and to be able to classify online data as being normal or intrusive [1]. Neural networks can learn from an environment by adjusting their internal structure through a training process. The neural network uses non-linear regression to abstract information from the abnormal training cases to predict future attacks [5].

4.1 What is ANN?

Originally, artificial neural networks (ANNs) were mainly inspired by the observation from human nervous system with the complex webs of interconnected neurons built in. Artificial neural

networks are based on a densely interconnected set of simple units. Each unit takes a number of real-valued inputs and produces a single real-valued output. ANN is composed of simple processing units and connections between them. The connection between any two units has some weight, which is used to determine how much one unit will affect the other. A subset of the units of the network acts as input nodes, and another subset acts as output nodes. By assigning a value, or activation, to each input node, and allowing the activations to propagate through the network, a neural network performs a functional mapping from one set of values to another set of values. The mapping itself is stored in the weight of the network [1].

The eight network inputs are connected to three hidden units, which are in turn connected to the eight output units. This structure can be seen in Figure 3. Because of this structure, the three hidden units will be forced to re-represent the eight input values in some way that captures their relevant features, so that this hidden layer representation can be used by the output units to compute the correct target values.

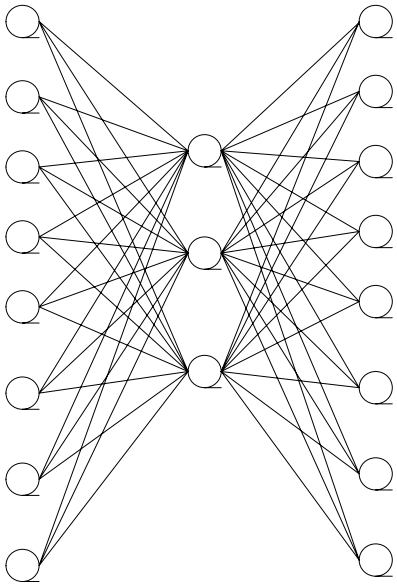

Figure 3

4.2 Neural network training

During the training phase, the network is trying to learn a relationship between the inputs and outputs. The neural network gains the experience initially by training the system to correctly identify training data set 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. Overfitting the training data is an important issue in ANN learning. Overfitting results in networks that generalize poorly to new data despite

excellent performance over the training data. Cross-validation methods can be used to estimate an appropriate stopping point for gradient descent search and thus to minimize the risk of overfitting.

4.3 Advantage of ANN based anomaly detection

The first advantage of using neural network would be the ability to generalize from past behavior to detect novel attacks. The accuracy of classification by ANN benefits from its classifier algorithm. The classifier algorithm determines the best solution by trying to minimize the number of incorrectly classified cases during the training process. A neural network might be trained to recognize known suspicious behaviors with a high degree of accuracy. ANN learning can solve problems with the noisy and complicated training data. ANN learning is robust to errors in the training dataset. ANN can also detect time series attacks with the help of its strong classification ability. Time series of the sequential attacks can be treated as one ANN's input node for analysis. An ideal application in intrusion detection will be to gather sufficient normal and abnormal audit data for a user or a program and then apply a classification algorithm to learn a classifier that can label or predict new unseen audit data as belonging to the normal class or the abnormal class [4]. With an input layer, a hidden layer and an output layer a neural network can be constructed any arbitrarily complex function [1]. Anomaly detection based on ANN can detect novel and modified attacks, decrease the rates of false positives and even detect hiding intrusion by ANN's intrinsic black-box feature. The another reason for using ANN is that the black box of ANN can approximate the internal states of commercial software even if the source code is unavailable. Intrusion becomes so difficult to simulate that hiding intrusion on the program's internal states is impractical for an attacker. The advantages of ANN can be summarized in Table 1.

Detection capability	Simple statistics	Rule-based system	ANN
Generalization	Worse	N/A	Better
Classification	Worse	N/A	Better
Detecting novel attacks	Worse	Worse	Better

Detecting hiding intrusions	N/A	N/A	Better
Detecting time series attacks	N/A	N/A	Better
Detecting non time series attacks	Worse	Worse	Better
Self learning	Worse	Worse	Better

Table 1

4.4 Disadvantage of ANN based anomaly detection

The ability of ANN to detect anomalous intrusion depends upon the accurate training of the IDS. The training data and the training methods are critical. The training process needs large amount of data to avoid overfitting. Overfitting is especially dangerous because it can easily lead to predictions that are far beyond the range of the training data. Overfitting can also produce wild predictions in multilayer perceptrons even with noise-free data. Data collection process is also difficult. The second disadvantage of applying networks to intrusion detection is the black box nature of the neural network. Neural networks have been viewed as a black box that cannot explain how they actually model data. Neural networks adapt the analysis of data in response to the training 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. Everything has its two sides. On the other hand, the advantage of the black box is that it can be used to detect malicious attacks against commercial software where the source code is unavailable [6].

5 Conclusion

This paper began with a classification of intrusions based on their manifestations on the specific locations. Process-based anomalous intrusion detection is our major concern and discussion point in this paper. The process's internal states and external states can be both captured to detect anomalous intrusions. Expert system is proved to be not a suitable detection tool for anomalous intrusions owing to its many limitations. In

conclusion, ANNs may be the most suitable technology for anomaly intrusion detection. ANNs' generalization and classification ability performs much better than expert system shells. Although the black box nature of ANNs makes the training process unpredictable, it can be used to defeat hiding intrusions.

Acknowledgements. I would like to thank Professor Clark Thomborson for his academic advises during the period of writing this term paper. Without his precious instructions, I can not find my way of thinking my own critical comments from different reference papers. The way of critical thinking mainly benefits from his teaching in the software security class in the first semester of 2003.

Reference

[1] A study in using neural networks for anomaly and misuse detection. Anup K. Ghosh & Aaron Schwartzbard

[2] Artificial Neural Networks for Misuse Detection. James Cannady

[3] A Survey and Analysis of Neural Network Approaches to Intrusion Detection. Hussam O. Mousa

[4] Intrusion and intrusion detection. John McHugh

[5] Intrusion Detection Applying Machine Learning to Solaris Audit Data. David Endler

Ideas:

[6] Detecting Anomalous and Unknown Intrusions Against Programs. Anup K. Ghosh, James Wanken, & Frank Charron

[7] Hiding Intrusions: From the Abnormal to the Normal and Beyond. Kymie Tan, John McHugh, and Kevin Killourhy

[8] Learning Fingerprints for a Database Intrusion Detection System. Sin Yeung Lee, Wai Lup Low, and Pei Yuen Wong