Developing an Algorithm to Implement Efficient Intrusion Detection System using Soft Computing

A. S. Zadgaonkar, Pooja Agrawal, Anju Lata Rathore

Abstract: ***During last decades information technologies based on the computer networks play an important role in various spheres of human activity. Information has become the organizations most precious asset. Organizations have become increasingly dependent on the information since more information is being stored and processed on network-based systems. The widespread use of e-commerce has increased the necessity of protecting the system to a very high extent. Problems of great importance are entrusted on them, such as keeping, transmission and automation of information processing. The security level of processed information can vary from private and commercial to military and state secret.***

Index Terms—Information, Organizations, Commercial, Security, Internet, Automation.

I. INTRODUCTION

Herewith the violation of the information confidentiality, integrity and accessibility may cause the damage to its owner and have significant undesirable consequences. Thus the problem of information security is concerned to many organizations and companies for development of security facilities that require significant contributions. To protect computer systems such accustomed mechanisms as identification and authentication mechanisms of the delimitation and restriction of the access to information and cryptographic methods are applied. With the advent of soft computing, intrusion detection has become an integral part of the security process.

An intrusion detection system can be described at a very macroscopic level as a detector that processes information coming from the system to be protected. It uses three kinds of information: long-term information related to the technique used to detect intrusions configuration information about the current state of the system, and audit information describing the events that are happening to the system.

II. NETWORKING ATTACKS

This section is an overview of the four major categories of networking attacks. Every attack on a network can comfortably be placed into one of these groupings.

1. Denial of Service (DoS): A DoS attack is a type of attack in which the hacker makes a computing or memory resources too busy or too full to serve legitimate networking requests and hence denying users access to a machine e.g. apache, smurf, neptune, ping of death, back, mail bomb, UDP storm etc. are all DoS attacks.

2. Remote to User Attacks (R2L): A remote to user attack is an attack in which a user sends packets to a machine over the internet, which s/he does not have access to in order to expose the machines vulnerabilities and exploit privileges which a local user would have on the computer e.g. xlock, guest, xnsnoop, phf, sendmail dictionary etc.

3. User to Root Attacks (U2R): These attacks are exploitations in which the hacker starts off on the system with a normal user account and attempts to abuse vulnerabilities in the system in order to gain super user privileges e.g. perl, xterm.

4. Probing: Probing is an attack in which the hacker scans a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited so as to compromise the system. This technique is commonly used in data mining e.g. saint, portsweep, mscan, nmap etc.

III. PROBLEM STATEMENT

The rapid development of computer networks and mostly of the Internet has created many stability and security problems such as intrusions on computer and network systems. Further the dependency of companies and government agencies is increasing on their computer networks and the significance of protecting these systems from attacks is serious because a single intrusion can cause a heavy loss or the consistency of network becomes insecure. During recent years number of intrusions has dramatically increased. Therefore it is very important to prevent such intrusions. The hindrance of such intrusions is entirely dependent on their detection that is a key part of any security tool such as Intrusion Detection System (IDS), Intrusion Prevention System (IPS), Adaptive Security Alliance (ASA), checkpoints and firewalls. Hence accurate detection of network attack is imperative. A variety of intrusion detection approaches are available but the main problem is their performance, which can be enhanced by increasing the detection rates and reducing false positives.

Manuscript received on May 3, 2013

Dr. A.S. Zadgaonkar, Vice Chancellor, Dr. C.V.Raman University, Kargiroad, Kota, Bilaspur(C.G.), India
Pooja Agrawal, Asst. Professor, Dr. C.V.Raman University, Kargiroad, Kota, Bilaspur, India
Anju Lata Rathore, Research Scholar, Dr. C.V.Raman University, Kargiroad, Kota, Bilaspur(C.G.), India

IV. INTRUSION DETECTION SYSTEM

Intrusion detection is defined as the process of intelligently monitoring the events occurring in a computer system or network, analyzing them for signs of violations of security policy. The primary aim of Intrusion Detection System (IDS) is to protect the availability, confidentiality and integrity of critical networked information systems. Intrusion Detection Systems are an important component of defensive measures protecting computer systems and networks from abuse. When an IDS is properly deployed it can provide warnings indicating that a system is under attack.

It is critical for intrusion detection in order for the IDS to achieve maximal performance.

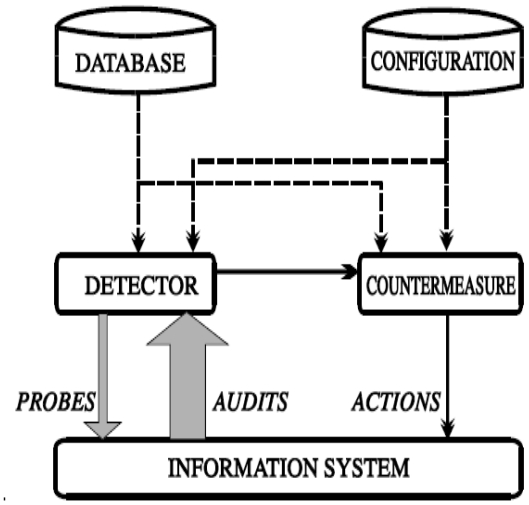

Fig 1. Intrusion Detection System

An intrusion detection system can be described at a very macroscopic level as a detector that processes information coming from the system to be protected. This detector can also launch probes to trigger the audit process, such as requesting version numbers for applications. It uses three kinds of information: long-term information related to the technique used to detect intrusions (a knowledge base of attacks for example), configuration information about the current state of the system, and audit information describing the events that are happening to the system. The role of the detector is to eliminate unneeded information from the audit trail. It then presents either a synthetic view of the security-related actions taken during normal usage of the system, or a synthetic view of the current security state of the system. A decision is then taken to evaluate the probability that these actions or this state can be considered as symptoms of an intrusion or vulnerabilities. A countermeasure component can then take corrective action to either prevent the actions from being executed or change the state of the system back to a secure state.

1. Classification of Intrusion Detection: Intrusions Detection can be classified into two main categories. They are as follow:

A. ***Host Based Intrusion Detection*:** HIDSs evaluate information found on a single or multiple host systems, including contents of operating systems, system and application files.

B. ***Network Based Intrusion Detection*:** NIDSs evaluate information captured from network communications, analyzing the stream of packets which travel across the network.

2. Components of Intrusion Detection System: An intrusion detection system normally consists of three functional components [23]. The first component of an intrusion detection system, also known as the event generator, is a ***data source***. Data sources can be categorized into four categories namely Host-based monitors, Network-based monitors, Application-based monitors and Target-based monitors. The second component of an intrusion detection system is known as the ***analysis engine***. This component takes information from the data source and examines the data for symptoms of attacks or other policy violations. The analysis engine can use one or both of the following analysis approaches:

A. Misuse/Signature-Based Detection: This type of detection engine detects intrusions that follow well-known patterns of attacks (or signatures) that exploit known software vulnerabilities [24][25]. The main limitation of this approach is that it only looks for the known weaknesses and may not care about detecting unknown future intrusions [26].

B. Anomaly/Statistical Detection: An anomaly based detection engine will search for something rare or unusual [26]. They analyses system event streams, using statistical techniques to find patterns of activity that appear to be abnormal. The primary disadvantages of this system are that they are highly expensive and they can recognize an intrusive behavior as normal behavior because of insufficient data

C. The third component of an intrusion detection system is the ***response manager***. In basic terms, the response manager will only act when inaccuracies (possible intrusion attacks) are found on the system, by informing someone or something in the form of a response.

V. RELATED WORKS

Some import applications of soft computing techniques for Network Intrusion Detection is described in this section. Several Genetic Algorithms (GAs) and Genetic Programming (GP) has been used for detecting intrusion detection of different kinds in different scenarios. Some uses GA for deriving classification rules . Gas used to select required features and to determine the optimal and minimal parameters of some core functions in which different AI methods were used to derive acquisition of rules . There are several papers related to IDS which has certain level of impact in network security.

The effort of using GAs for intrusion detection can be referred back to 1995, when Crosbie and Spafford applied the multiple agent technology and GP to detect network anomalies . For both agents they used GP to determine anomalous network behaviors and each agent can monitor one parameter of the network audit data. The proposed methodology has the advantage when many small autonomous agents are used but it has problem when communicating among the agents and also if the agents are not properly initialized the training process can be time consuming.

Li described a method using GA to detect anomalous network intrusion. The approach includes both quantitative and categorical features of network data for deriving classification rules. However, the inclusion of quantitative feature can increase detection rate but no experimental results are available.

Goyal and Kumar described a GA based algorithm to classify all types of smurf attack using the training dataset with false positive rate is very low (at 0.2%) and detection rate is almost 100% .

Lu and Traore used historical network dataset using GP to derive a set of classification [19]. They used support-confidence framework as the fitness function and accurately classified several network intrusions. But their use of genetic programming made the implementation procedure very difficult and also for training procedure more data and time is required.

Xiao et al.used GA to detect anomalous network behaviours based on information theory. Some network features can be identified with network attacks based on mutual information between network features and type of intrusions and then using these features a linear structure rule and also a GA is derived. The approach of using mutual information and resulting linear rule seems very effective because of the reduced complexity and higher detection rate. The only problem is it considered only the discrete features.

Gong et al. presented an implementation of GA based approach to Network Intrusion Detection using GA and showed software implementation. The approach derived a set of classification rules and utilizes a support-confidence framework to judge fitness function.

Abdullah et al.showed a GA based performance evaluation algorithm to network intrusion detection. The approach uses information theory for filtering the traffic data.

VI. GENETIC ALGORITHM OVERVIEW

A Genetic Algorithm (GA) is a programming technique that mimics biological evolution as a problem-solving strategy. It is based on Darwinian's principle of evolution and survival of fittest to optimize a population of candidate solutions towards a predefined fitness.

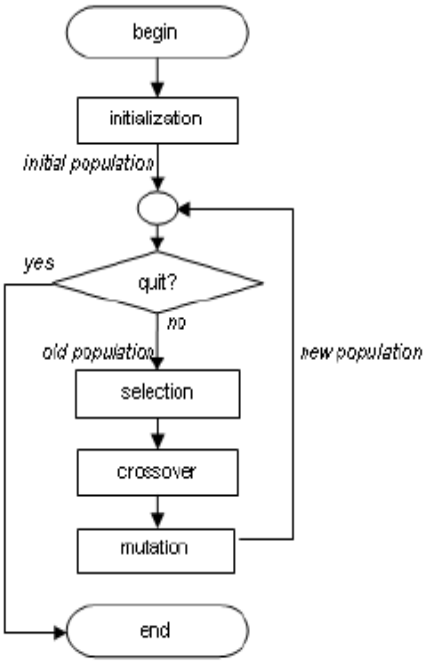

Fig2: the operation of a generic GA.

GA uses an evolution and natural selection that uses a chromosome-like data structure and evolve the chromosomes using selection, recombination and mutation operators [6]. The process usually begins with randomly generated population of chromosomes, which represent all possible solution of a problem that are considered candidate solutions. From each chromosome different positions are encoded as bits, characters or numbers. These positions could be referred to as genes. An evaluation function is used to calculate the goodness of each chromosome according to the desired solution; this function is known as "Fitness Function". During the process of evaluation "Crossover" is used to simulate natural reproduction and "Mutation" is used to mutation of species [6]. For survival and combination the selection of chromosomes is biased towards the fittest chromosomes.

When we use GA for solving various problems three factors will have vital impact on the effectiveness of the algorithm and also of the applications . They are: i) the fitness function; ii) the representation of individuals; and iii) the GA parameters. The determination of these factors often depends on applications and/or implementation.

VII. PROPOSED METHODOLOGY

Our system can be divided into two main phases: the precalculation phase and the detection phase. Listing 1 depicts major steps in precalculation phase, where a set of chromosome is created using training data. This chromosome set will be used in the next phase for the purpose of comparison.

Listing 2 depicts major steps of detection phase, where a population is being created for a test data and going through some evaluation processes (selection, crossover, mutation) the type of the test data is predicted.

The precalculated set of chromosome is used in this phase to find out fitness of each chromosome of the population.

1. In the precalculation phase, we have made 23 groups of chromosomes according to training data. There were 23 (22+1) groups for each of attack and normal types presented in training data.

2. Number of chromosomes in each group is variable and depends on the number of data and relationship among data in that group. Total number of chromosomes in all groups were tried to keep in reasonable level to optimize time consumption in testing phase.

3. In the testing / detection phase, for each test data, an initial population is made using the data and occurring mutation in different features.

4. This population is compared with each chromosomes prepared in training phase. Portion of population, which are more loosely related with all training data than others, are removed. Crossover and mutation occurs in rest of the population which becomes the population of new generation.

5. The process runs until the generation size comes down to 1 (one). The group of the chromosome which is closest relative of only surviving chromosome of test data is returned as the predicted type.

VIII. CONCLUSION

This research is centered on "Developing an algorithm to implement efficient intrusion detection system". The outcome of the proposed work will likely to yield expected result and fulfill the following objective:

1. Algorithm to implement Intrusion detection system based on Soft Computing Technique.
2. Reduce number of false positive alarms.

REFERENCES

1. M. Botha, R. Solms, "Utilizing Neural Networks For Effective Intrusion Detection", ISSA, 2004.
2. R. Graham, "FAQ: Network Intrusion Detection Systems". March 21, 2000.
3. D. Zamboni, "Using Internal Sensors For Computer Intrusion Detection". Center for Education and Research in Information Assurance and Security, Purdue University. August 2001.
4. K. Scarfone, P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)". Computer Security Resource Center (National Institute of Standards and Technology). February 2007.
5. A. Chittur, "Model Generation for an Intrusion Detection System Using Genetic Algorithms". January 2005.
6. W. Li, "Using Genetic Algorithm for Network Intrusion Detection". "A Genetic Algorithm Approach to Network Intrusion Detection". SANS Institute, USA, 2004.
7. W. Lu, I. Traore, "Detecting New Forms of Network Intrusion Using Genetic Programming". Computational Intelligence, vol. 20, pp. 3, Blackwell Publishing, Malden, pp. 475-494, 2004.
8. M. M. Pillai, J. H. P. Eloff, H. S. Venter, "An Approach to Implement a Network Intrusion Detection System using Genetic Algorithms", Proceedings of SAICSIT, pp:221-228, 2004.
9. S. M. Bridges, R. B. Vaughn, "Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection", Proceedings of 12th Annual Canadian Information Technology Security Symposium, pp. 109-122, 2000.
10. J. Gomez, D. Dasgupta, "Evolving Fuzzy Classifiers for Intrusion Detection", Proceedings of the IEEE, 2002.
11. M. Middlemiss, G. Dick, "Feature selection of intrusion detection data using a hybrid genetic algorithm/KNN approach", Design and application of hybrid intelligent systems, IOS Press Amsterdam, pp.519-527, 2003.
12. Srinivas Mukkamala, Andrew H. Sung, Ajith Abraham, "Intrusion detection using an ensemble of intelligent paradigms", Journal of Network and Computer Applications, Volume 28, Issue 2, April 2005, Pages 167-182
13. S. Peddabachigari, Ajith Abraham, C. Grosan, J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems", Journal of Network and Computer Applications, Volume 30, Issue 1, January 2007, Pages 114–132
14. M. Saniee Abadeh, J. Habibi, C. Lucas, "Intrusion detection using a fuzzy genetics-based learning algorithm", Journal of Network and Computer Applications, Volume 30, Issue 1, January 2007,Pages 414-428
15. Tao Peng, C. Leckie, Kotagiri Ramamohanarao, "Information sharing for distributed intrusion detection systems", Journal of Network and Computer Applications, Volume 30, Issue 3, August 2007, Pages 877–899
16. M. Crosbie, E. Spafford, "Applying Genetic Programming to Intrusion Detection", Proceedings of the AAAI Fall Symposium, 1995.
17. T. Xia, G. Qu, S. Hariri, M. Yousif, "An Efficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm", Proceedings of the 24th IEEE International Performance Computing and Communications Conference (IPCCC '05), Phoenix, AZ, USA. 2005.
18. Anup Goyal, Chetan Kumar, "GA-NIDS: A Genetic Algorithm based Network Intrusion Detection System", 2008.
19. R. H. Gong, M. Zulkernine, P. Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection", 2005.
20. B. Abdullah, I. Abd-alghafar, Gouda I. Salama, A. Abd-alhafez, "Performance Evaluation of a Genetic Algorithm Based Approach to Network Intrusion Detection System", 2009.
21. A. Sung, S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks" in Symposium on Applications and the Internet, pp. 209– 216. 2003.
22. J. P. Planquart, "Application of Neural Networks to Intrusion Detection", SANS Institute Reading Room.
23. R. G. Bace, "Intrusion Detection", Macmillan Technical Publishing. 2000.
24. S. Kumar, E. Spafford, "A Software architecture to Support Misuse Intrusion Detection" in The 18th National Information Security Conference, pp. 194-204. 1995.
25. K. Ilgun, R. Kemmerer, P. A. Porras, "State Transition Analysis: A Rule-Based Intrusion Detection Approach", IEEE Transaction on Software Engineering, 21(3):pp. 181-199. 1995.
26. S. Kumar, "Classification and Detection of Computer Intrusions", Purdue University, 1995.
27. V. Bobor, "Efficient Intrusion Detection System Architecture Based on Neural Networks and Genetic Algorithms", Department of Computer and Systems Sciences, Stockholm University / Royal Institute of Technology, KTH/DSV, 2006.
28. KDD-CUP-99 Task Description; http://kdd.ics.uci.edu/databases/kddcup99/task.html
29. KDD Cup 1999: Tasks; http://www.kdd.org/kddcup/index.php?section=1999&method=task
30. KDD Cup 1999: Data; http://www.kdd.org/kddcup/index.php?section=1999&method=data
31. Results of the KDD'99 Classifier Learning Contest; http://cseweb.ucsd.edu/~elkan/clresults.html
32. H. G. Kayacık, A. N. Zincir-Heywood, M. I. Heywood, "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets", May 2005.
33. G. Folino, C. Pizzuti, G. Spezzano, "GP Ensemble for Distributed Intrusion Detection Systems". ICAPR 54-62, 2005.

34. Sectools.Org: 2006 Results; http://sectools.org/tools2006.html
35. SecTools.Org: Top 125 Network Security Tools; http://sectools.org/tag/ids/
36. Snort (software); http://en.wikipedia.org/wiki/Snort_%28software%29
37. InfoWorld, The greatest open source software of all time, 2009; http://www.infoworld.com/d/open-source/greatest-open-source-software-all-time- 776?source=fssr
38. Suricata (software); http://en.wikipedia.org/wiki/Suricata_(software)
39. [39] The Bro Network Security Monitor; http://bro-ids.org/