Application of Genetic Algorithms for Detecting Anomaly in Network Intrusion Detection Systems

K.G. Srinivasa

Machine Learning Applications Laboratory,
Department of Computer Science and Engineering,
M.S. Ramaiah Institute of Technology, Bangalore-560 054, India
kgsrinivas@msrit.edu

Abstract. Intrusion Detection System (IDS) can handle intrusions in computer environments by triggering alerts to help the analysts for taking actions to stop the possible attack or intrusion. But, the IDS make the job of analyst more difficult by triggering thousands of alerts for any suspicious activity. In this paper, an anomaly based network intrusion detection system using a genetic algorithm approach is adopted. The proposed method is efficient with respect to good detection rate with low false positives. The experimental results demonstrate the lower execution time of the proposed algorithm *GANIDS* (Genetic Algorithms based Network Intrusion Detection System) when compared with PAYL [1]. The proposed payload based IDS uses an adaptive genetic algorithm for both learning and detection. The proposed *GANIDS* is benchmarked with PAYL [1] using the 1999 DARPA IDS dataset.

Keywords: Intrusion Detection Systems, Genetic Algorithms, Anomaly Detection.

1 Introduction

An intrusion detection system is used to detect many types of malicious activities in network traffic and computer usage. Typically, the types of attacks include network attacks against vulnerable services, data driven attacks on applications, host based attacks such as privilege escalation, unauthorized logins and access to sensitive files. An IDS monitors network traffic and monitors for suspicious activity and alerts the system or network administrator. In some cases the IDS may also respond to anomalous or malicious traffic by taking action such as blocking the user or source IP address from accessing the network. IDS come in a variety of *flavors* and approach the goal of detecting suspicious traffic in different ways. There are IDS that detect based on looking for specific signatures of known threats similar to the way antivirus software typically detects and protects against malware; there exist IDS that detect based on comparing traffic patterns against a baseline and looking for anomalies [2-5].

NIDS: Network Intrusion Detection Systems are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. Ideally you would scan all inbound and outbound traffic; however doing so might create a bottleneck that would impair the overall speed of the network [9].

N. Meghanathan et al. (Eds.): CCSIT 2012, Part I, LNICST 84, pp. 582–591, 2012.

HIDS: Host IDS are run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator of suspicious activity is detected. The other classifications of intrusion detection systems are signature based and anomaly based [13].

Signature based systems: A signature based IDS work with an intrusion database populated offline by knowing of the characteristics of the attack. Thus the IDS have to compare the input and classify it into normal and abnormal categories. A signature based IDS will monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware. The problem is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to the IDS under consideration. During that lag time the IDS would be unable to detect the new threat.

Anomaly Based Systems: An Intrusion Detection System (IDS) which is anomaly based will monitor network traffic and compare it against an established baseline. The baseline identifies normality for that network with respect to the sort of bandwidth to be generally used, the protocols to be used, the ports and devices to connect to each other, and finally alerts the administrator or user when the traffic is detected with anomaly, or significantly different than the baseline. Anomaly based systems have only the normal behaviors in their profiles and any deviation above a threshold is signaled as an anomaly. Unlike signature based systems which give low detection rates and low false positive rates anomaly based system suffer from high false-positive rates; however they have a good detection rate [14].

Signature based systems cannot detect new attacks until they are known and added to the database. This results in lower detection rates. Signature based systems are preferable to detect attacks on the operating systems. However, anomaly based systems have the ability to detect zero-day worms. And hence are preferable for network related attacks. Most of the systems used till now are predominantly signature based however a considerable amount of research is going on for reducing the false positive rates and increasing the detection rates in anomaly based systems. An anomaly based system can classify the input based on either the header information or the payload. In this paper we describe a payload based IDS with applied Genetic Algorithms.

Contribution: In this paper, a genetic algorithm based approach to network intrusion detection system is adopted. The literature demonstrates that the Genetic Algorithms provide better and faster classification than any neural network architectures, and also takes less time for training and gives detection rate. Since GANIDS is payload based, it uses only the destination address and the service port numbers for building profiles and all the other header information is ignored. Further, it uses a single tier architecture where a GA is used for both classification and detection. We have benchmarked our system with respect to PAYL using the 1999 DARPA IDS dataset. On this dataset the proposed system shows a reasonable detection rate with low false positives and a faster running time than PAYL.

2 Related Works

Genetic Algorithms belong to the evolutionary algorithms and is very efficient in machine learning. Genetic algorithms are search procedures often used for optimization problems. The genetic algorithm works by slowly evolving a population of chromosomes that represent better and better solutions to the problem. It has emerged to be a very effective tool in data mining applications. Since in an IDS the incoming packet needs to be classified into normal and abnormal categories a GA functions best in this job since it can classify with a higher accuracy than any other methods for example Neural Networks etc. The objective of using a GA is to obtain a better classification of the input data resulting in higher detection rates with lesser false positives, which is a major concern for an Anomaly Based IDS. In addition to that genetic algorithms are relatively faster than neural networks and requires less time for training and hence the performance of the system increases considerably [15]. Neural networks are trained to detect intrusion systems. An n-layer network is constructed and abstract commands are defined in terms of sequence of information units, the input to the neural in the training data. Each command is considered with pre-defined *w* commands together to predict the next coming command expected from the user. After training, the system will have the profile of the user. At the testing step, an anomaly is said to occur as the user deviates from the expected behavior [16]. Evolving fuzzy classifiers have been studied for possible application to the intrusion detection problem. System audit training data is used to extract rules for each normal and abnormal behavior by the genetic algorithm. Rules are represented as complete expression tree with identified operators, such as conjunction, disjunction and not [8].

An efficient and biologically inspired learning model for anomaly intrusion detection in the multi-agent IDS is designed for decentralized intrusion detection and prevention control in large switched networks. The proposed model called Ant Colony Clustering Model improves the existing ant-based clustering approach in searching for near-optimal clustering heuristic. The multiple agent technology and Genetic programming (GP) are used to detect network attempts. Each agent monitors one parameter of the network packet and GP is used to find the set of agents that collectively determine anomalous network behaviors. This method has the advantage of using many small autonomous agents, but the communication among them is still a problem. Also the training process can be time consuming if the agents are not appropriately initialized [7, 11]. Researchers in [6, 7, 8,9,10, 12] have proposed paradigm consist from; neuro-fuzzy network, fuzzy inferences, and GA to detect intrusion activities in networks. This method firstly used a set of parallel nero-fuzzy classifiers (five layers 4- for type of attack, and one for normal). Then fuzzy inference used the output from classifiers to take a decision whether the current action is normal or not. The role of GA was used to optimize the classifier engine to give the right decision. This Method also used the same data KDD CUP 99 for training and for testing the system.

3 Architecture

The architecture of GANIDS is as shown in the Figure 1. An initial population of chromosomes is generated randomly where each chromosome represents a possible solution to the problem (an set of parameters).The incoming traffic is first captured using a packet capture engine which is then used to extract the payload by removing all the header information present in the packet and the payload is given as input to the genetic algorithm which in the training phase uses it to build profiles.

Two Point Crossover: In our system we use a two point crossover scheme where the two parents crossover at two different points producing a total of eight off springs out of which two are replicas of the parents itself which are discarded. The remaining two are then tested for fitness. If they are fit enough then they are added to the population else they are not. Selection of the parents for crossover is done by finding the fittest chromosome from the existing population and the input data forms the other parent. Since the input data is used to construct profiles the network behavior will be mapped on to the profiles efficiently.

Replacement Strategy: There are mainly two types of replacement techniques that are widely used viz. *Complete Replacement* and *Partial Replacement*. Complete replacement though easy to implement lose some of the fittest members in the population. However it is desirable for some of the chromosomes that are fit to survive in the population, hence we use a partial replacement technique where only some of the members are replaced and the rest are retrieved as it is. In our system we use a *steady state replacement* technique which is a partial replacement technique where the off springs replace the parents in the population. Also in our system parents that are unable to produce an offspring that is fit enough to be added to the population will also be removed from the population.

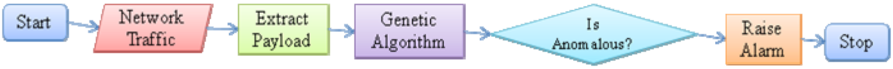

Fig. 1. GANIDS Architecture

Mutation: Mutation is very necessary in a genetic algorithm because it enables the algorithm to explore the search space more effectively and hence produces better results. In our system we perform mutations based on a mutation probability which varies dynamically during the course of execution. The algorithm used in the proposed system is presented below.

Problem Definition: Let x_i be the input payload at time instance *i,* then the problem is to find the Chromosome *c* which yields the lowest value for the computation *manhattan_distance* (*c.weight*[ ] , x_i)

Pseudo code: Here we give the pseudo code for crossover and mutation functions and finally the pseudo code for the genetic algorithm that we have used. If *pm* is the mutation probability and *G* the number of generations and nc_i the number of crossover points is two.

The algorithm given below is used during the training phase i.e. the machine learning phase of the IDS. In the training phase, the input from the training data is used to build profiles. The machine learning phase functions as follows. First the input payload is used to find the fittest chromosome. Then the fittest chromosome and the payload itself are crossed to produce a total of eight offsprings out of which two of them are the replicas of the parents itself which are discarded. The remaining six children are checked for fitness and checked against a threshold value. Only children which are fit enough are added to the population and others are discarded. Also if none of the six children are fit enough to be added to the population then even the parents are also removed from the population.

```
CROSSOVER
Input:
   $x_i$ - payload at time i.
  fittest – fittest chromosome
Output:
  children created and added if fit.
begin
  Children[6]=Cross(fittest,$x_i$)
  for all c ε Children
        find the fitness of c
        if fitness(c) > threshold
              add_to_population(c)
  remove(fittest)
end
```

```
MUTATION
Input:
        c – Chromosome
Output:
        $c^1$ – mutated chromosome
begin
  r=random()
  if  r > pm then
              mutate(c)
end
```

```
Genetic Algorithm:
Input:
        $x_i$ – payload at time i
Output:
        fittest – the fittest chromosome
begin
  for i=0 to G do
              min_dist = INFINITY
              fittest = 0
              for every c ε Chromosome do
                    dist = manhattan_distance( c , $x_i$ )
                    if $dist \le min\_dist$ then
                          fittest = i
                          min_dist = dist
              crossover( fittest , $x_i$ )
              for every c ε  Chromosome
                    mutation(c)
end
```

Then mutation is applied in order to explore the search space better. Mutation is done as follows; a random number *r* is generated for every chromosome in the population. If the value of *r* is greater than the mutation probability then some random numbers of weights are changed to some random values. In the testing phase the fittest chromosome is found as in the algorithm but the crossover and mutation operations are not performed. Instead when the fittest chromosome is found, the minimum distance obtained is checked against a threshold and if it is higher than the threshold then it is flagged off as an anomaly.

4 Performance Analysis

The two architectures GANIDS and PAYL are benchmarked using the same data used by PAYL, the DARPA 1999 data set. This standard data set is used as reference by a number of researchers and offers the possibility of comparing the performance of various IDS. This data set has been criticized because of the environment in which data were collected, but it is possible to tune an IDS in such a way that it scores particularly well on this particular data set: some attributes – specifically: remote client address, TTL, TCP options and TCP window size – have a small range in the DARPA simulation, but have a large and growing range in real traffic. IDS which take into account the above-mentioned attributes are likely to score much better on the DARPA set than in real life. Since our system does not consider these attributes, we can legitimately expect that the system in real life performs as well as it does on the DARPA benchmark. The GANIDS is trained using internal network traffic of week 1 and week 3. Then, the same data is used to build PAYL models taking advantage of the classification given by the neural network. After this double training phase, it is possible to use the testing weeks (4 and 5) to benchmark the network intrusion detection algorithm. This data contains several attack instances (97 payload-based attacks are detectable applying the same traffic filter mentioned above), as well as legal traffic, directed against different hosts of the internal network: the attack source can be situated both inside and outside the network. Figure 2 shows the graph of percentage of false positive packets versus percentage of instances of detected attacks on FTP packets on port 21 of DARPA. The percentage of true negatives in case of GANIDS is almost 10% less on average when compared to PAYL. Similar graph in Figure 3 shows better performance of GANIDS when compared to PAYL when test on TELNET packets on Port 23. False Positive attack instances was found to be linearly increasing with increase in detected attack instances on the application on GANIDS which was an improvisation over the existing performance.

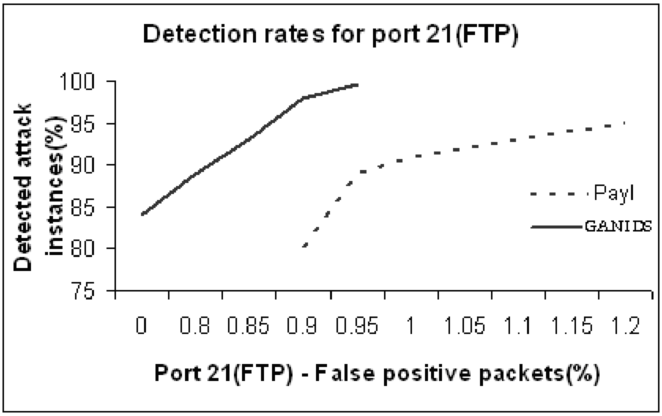

Fig. 2. A comparison of PAYL and GANIDS in terms of percentage of true negatives (reported on y axis) w.r.t the percentage false positives(x axis)

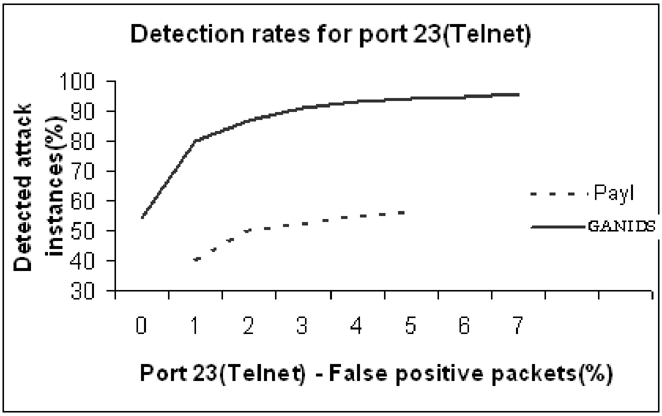

Fig. 3. A comparison of PAYL and GANIDS in terms of percentage of true negatives (reported on y axis) w.r.t the percentage false positives(x axis)

The experiments on SMTP and HTTP packets on Ports 25 and 80 also demonstrates better performance of GANIDS when compared to PAYL as shown in Figure 4 and Figure 5.

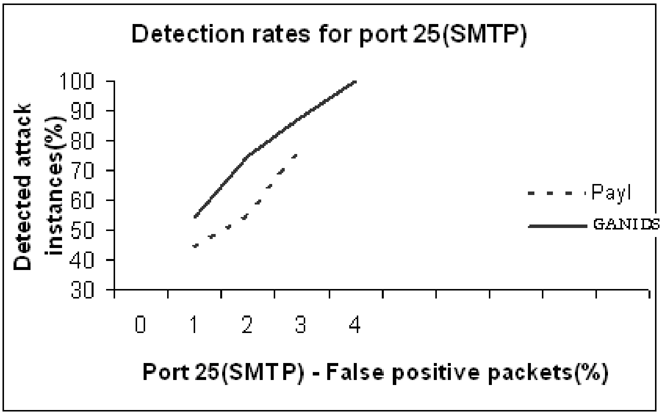

Fig. 4. A comparison of PAYL and GANIDS in terms of percentage of true negatives (reported on y axis) w.r.t the percentage false positives(x axis)

Table 1 reports a summary of these results: the first column reports PAYL's statistics and the second column reports the result of GANIDS. It is possible to observe that GANIDS overcomes PAYL on every benchmarked protocol: there is a remark about FTP protocol. During FTP protocol benchmarks we found a high rate of false positives both with PAYL and with GANIDS: all these packets are sent by the same source host, which is sending FTP commands in a way that is typical of the Telnet protocol (one character per packet, with the TCP flag PUSH set). These packets are marked as an attack because the training model does not contain this kind of traffic over the FTP control channel port, although it is normal traffic. During our experiments with PAYL we found the same behavior.

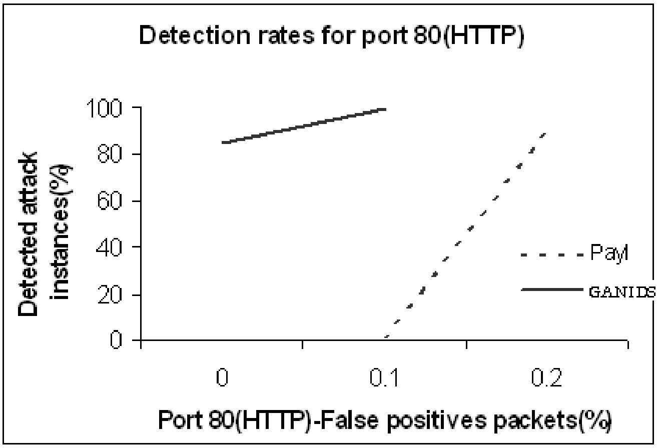

Fig. 5. A comparison of PAYL and GANIDS in terms of percentage of true negatives (reported on y axis) w.r.t the percentage false positives(x axis)

We trained our intrusion detection models, i.e., the base models and the meta-level classifier; using the 7 weeks of labeled data, and used them to make predictions on the 2 weeks of unlabeled test data (i.e. we were not told which connection is an attack). The test data contains a total of 38 attack types, with 14 types in test data only (i.e., our models were not trained with instances of these types). The reason for high false positive rate in *GANIDS* using was due to the obsolete nature of the DARPA 1999 dataset.

Table 1. Comparison between PAYL and GANIDS; DR stands for detection rate, while FP is the false positive rate

Architecture Used		PAYL	GANIDS
HTTP	DR FP	89.00% 0.17%	95.00% 0.01%
FTP	DR FP	95.50% 1.23%	98.00% 1.00%
Telnet	DR FP	54.17% 4.71%	85.12% 6.72%
SMTP	DR FP	73.34% 3.08%	95.00% 3.69%

5 Conclusions

It is often difficult to know which items from an audit trail will provide the most useful information for detecting intrusions. The process of determining which items are most useful is called feature selection in the machine learning literature. We have conducted a set of experiments in which we are using genetic algorithms both to select the measurements from the audit trail that are the best indicators for different classes of intrusions and to "tune" the membership functions for the fuzzy variables. *GANIDS* is a Genetic Algorithm based approach for anomaly based Network Intrusion Detection systems. The experiments on the DARPA set show that this approach reduces the number of profiles used by PAYL (payload length can vary between 0 and 1460 in a Local Area Network, while the proposed approach considers less than one hundred nodes). The experiments show that PAYL three times more the profiles as with the GANIDS. We benchmark *GANIDS* extensively against the PAYL algorithm and performance analysis shows a higher detection rate and lower false positives rate.

Acknowledgement. This project is partially funded by Research Grants of DRDO ER&IPR for the project titled *Machine Learning Techniques for Data Mining Based Intrusion Detection Systems,* to Dr. Srinivasa K G, Professor, Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bangalore for the year 2010-12.

References

[1] Wang, K., Stolfo, S.J.: Anomalous Payload-Based Network Intrusion Detection. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 203–222. Springer, Heidelberg (2004)

[2] Bolzoni, D., Etalle, S., Hartel, P.: POSEIDON: a 2-tier anomaly-based network intrusion detection system. In: Fourth IEEE International Workshop on In Information Assurance, IWIA 2006 (2006)

[3] Zhang, L.-H., et al.: Intrusion detection using rough set classification. Journal of Zhejiang University Science 5(9), 1076–1086 (2004)

[4] Zhao, J.-L., Zhao, J.-F., Li, J.-J.: Intrusion Detection Based On Clustering Genetic Algorithm. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, August 18-21 (2005)

[5] Lunt, T.: Detecting intruders in computer systems. In: Proceedings of Auditing and Computer Technology Conference, pp. 23–30 (1999)

[6] Ryan, J., Lin, M., Miikkulainen, R.: Intrusion detection with neural networks. In: Advances in Neural Information Processing Systems, vol. 10. MIT Press (1998)

[7] Crosbie, M.: Applying genetic programming to intrusion detection. In: Proceedings of AAAI Fall Symposium Series, pp. 45–52 (1995)

[8] Gomez, J., Dasgupta, D., Nasraoui, O.: Complete expression trees for evolving fuzzy classifiers systems with genetic algorithms and application to network intrusion detection. In: Proceedings of the NAFIPS-FLINT Joint Conference, pp. 469–474 (2002)

[9] Heady, R., Luger, G., Maccabe, A., Servilla, M.: The architecture of network level intrusion detection system, Technical Report, Department of Computer Science, University of New Mexico (1990)

[10] Ozyer, T., Alhaji, R., Barker, K.: Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule prescreening. Journal of Network and Computer Applications, 99–113 (2007)

[11] Crosbie, M., Spafford, E.: Applying genetic Programming to Intrusion Detection. In: Proceedings of the AAAI Fall Symposium (1995)

[12] Toosi, N., Kahani, M.: A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. Computer Communications 30, 2201–2212 (2007)

[13] Vokorokos, L., Balaz, A.: Host-based intrusion detection system, Technical University of Koaice, Department of Computers and Informatics, Slovak Republic (2010)

[14] Depren, O., Topallar, M., Anarim, E., Kemal Ciliz, M.: An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Bogazici University, Electrical and Electronics Engineering Department, Information and Communications Security (BUICS) Lab, Bebek, Istanbul, Turkey (2007)

[15] Li, W.: Using Genetic algorithms for Intrusion Detection System, Department of Computer Science and Engineering Mississippi State University, Mississippi State (2004)

[16] Ryan, J., Lin, M.-J., Miikkulainen, R.: Intrusion Detection with Neural networks. The University of Texas, Austin (1998)