A Flow Based Horizontal Scan Detection Using Genetic Algorithm Approach

BARATI, M.[1*,†], HAKIMI, Z.[1*], JAVADI, A.H.[2]

[1]Department of Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[2]Institute of Behavioral Neuroscience, University College London, London, UK
[*]These authors contributed to the same extend
m.barati@qiau.ac.ir

Abstract: An attacker has to 'scan' susceptible points of a network before attacking. There are several methods of detection of such behavior which are mostly based on thresholding. As the performance of these methods is highly dependent on the value of threshold, it is crucial to adjust this value appropriately. This adjustment is not always trivial. In this study we proposed a new method to optimize the parameters of the system using genetic algorithms (GA) based on network flows. Subsequently we compared our method with Snort. The results showed a superior performance as measured by the sensitivity index of d'.
[BARATI, M., HAKIMI, Z., JAVADI, A.H. **A Flow Based Horizontal Scan Detection Using Genetic Algorithm Approach**. *LIFE SCI J* 2013;10(8S): 331-335] (ISSN:1097-8135). http://www.lifesciencesite.com. 52

Keywords: genetic algorithm, horizontal scan attack, intrusion detection, flow, threshold

1. Introduction

With rapid increase of number of computer networks and growth of Internet, network and data security is becoming more important every day. Every day a large number of computers are attacked and their information is stolen. In order to attack a computer, attacker needs to find vulnerable points of the system, which is called 'scanning'. Scanning is done by sending small packets to target systems. Scanning is categorized into two types: horizontal and vertical scan. In horizontal scan an attacker scans a specific port on many destination hosts. Whereas in vertical scan an attacker scans several ports on a single destination host (Sperotto et al., 2010).

To secure networks against these attacks, one needs to use security tools such as intrusion detection systems (IDS). An IDS monitors all network traffic and sends an alarm to network administrator once it detects a suspicious activity. One important parameter in IDSs is detection accuracy. To achieve high performance the following two rates must be low, 'false positives/false alarms' (a normal behavior is detected as an attack) and 'false negatives/misses' (an attack is detected as a normal behavior).

Traditional IDS systems were packet-based. These systems inspect contents of each packet to find suspicious activities. But, due to growing speed of networks to multiple gigabits per second (Gbps), inspecting contents of every packet is very time demanding and even not feasible in most of the cases. For high speed networks, we need another solution that reduces the amount of processing load. Flow-based IDS is an option. In flow-based IDS systems, the patterns of communication within the network (grouping several packets into one pattern) are analyzed, rather than the contents of individual packets (Sperotto et al., 2010).

One of the scan detection methods is Thresholding (Grégr, 2010, Sekar et al., 2006, Moon et al., 2010). These approaches consider some sorts of parameters for detecting scan attacks. If these parameters go beyond a certain threshold, the system sends an alarm to the administrator. Performance of these systems highly depends on the Threshold value (Moon et al., 2010). A frequently used parameter for scan attacks is number of connections between a single source and a destination (Bhuyan et al., 2011).

In this paper, we proposed an approach based on genetic algorithm (GA) to identify horizontal scan attacks within a flow-based IDS. This approach is highly effective in high-speed networks as compared to methods based on packets.

Application of GA in IDS research has begun since 1995 (Crosbie and Spafford, 1995). GA is beneficial due to several reasons: (1) GA optimizes multiple members in each run, in another words it is highly parallel. This way, it avoids being trapped in local minimums and it extends its search domain. (2) GA is highly adaptive; Therefore it can adapt easily with changes in the network, such as changes in the extension and needs (Bankovic et al., 2009).

In our approach we define two parameters: (i) number of destinations that each source visited, and (ii) number of scanned ports on that destination. GA is used to optimize weights of these parameters to achieve a performance close to optimum.

Finally we compared our method with Snort. Next section reviews the previous studies on attacks detection methods based on scanning using GA. Methodology section gives a brief explanation of genetic algorithm, Snort and DARPA 1999 database

and Implementation of our method. Results section presents the results and finally conclusions in the last section.

Related Work

Jung et al. (2004) introduced a new method called threshold random walk (TRW) in which the scanner does not have any knowledge about hosts and their ports' status on the target network. Therefore it is more likely to send packets to the hosts that are not available or do not have requested services. While a legitimate remote host has not such behavior. TRW is an online detection algorithm that is based on count of connection requests for each IP address. This count is a small number for successful connections and increases for unsuccessful connections. Once the count exceeds a specific threshold, the host is marked as 'victim' and an alert is generated (Grégr, 2010).

Sekar et al. (2006) introduced a novel scan detection algorithm called multi-resolution detection system (MRDS). Due to limitations that a single detection threshold value imposes to the system, such as high ratio of false alarms and missing low rate scans, MRDS uses multiple monitoring windows with different thresholds, i.e. a large threshold for detecting slow scan attacks and a small threshold for detecting fast scan attacks. The weakness of this algorithm is that it needs to allocate huge amount of memory for holding multiple monitoring window.

Srinivasa et al. (2011) proposed a GA based IDS (IGIDS). The aim of IGIDS is to improve performance of network IDS by evolving rule set. In this system, each rule is modeled as a chromosome. Then GA operators are applied to create new rules from initial population. The advantage of this system is its integration with any intrusion detection system due to adaptability of their rule set.

Gong et al. (2005) introduced a GA approach in which a set of classification rules excluded from network audit data. Using these rules they are able to detect intrusions in real-time. The main advantage of this approach is its simple implementation and flexibility in detection of network intrusions.

Hoque et al. (2012) proposed an IDS that uses GA to improve detection rate. Their system operated within two phases: precalculation and detection phase. In precalculation phase, according to training data, some chromosomes are created that are used in the next phase in calculation of fitness.

Methodology

Tools

DARPA

IDS evaluation data set of 1999 DARPA (www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/) is a simulated network traffic in a hypothetical military network (Lippmann et al., 2000). Traffic generated by this network is over 5 weeks. The first 3 weeks are for training purposes and the last two weeks are for testing. It has a master identification file that includes information related to attacks. This dataset is used by many researchers for evaluation of their intrusion detection methods (Srinivasa et al., 2011, Srinivasa, 2012).

Snort

Snort (www.snort.org) is a signature based network IDS that is widely used for IDS purposes (Riquet et al., 2012, Sridharan et al., 2006). An alarm is generated whenever there is a match between the rules and the packet. Portscan detection in Snort is supported by a preprocessor plugin. It relies on a set of simple rules such as: (i) Number of TCP/UDP packets sent to destination/port by a special source. (ii) TCP packets with unusual TCP flags set or with any flags set (Staniford et al., 2002). In this study we compared the performance of our system with Snort. For this simulation, we enabled scan rules of Snort and included only IPSweep attacks only.

Genetic Algorithms (GA)

Genetic Algorithm (GA) is an evolutionary algorithm. Its main concept is "survival of the fittest" (Goldberg, 1989). GA creates new population of individuals (generation) in each run. Each individual (member) represents some parameters (chromosomes) that are used in optimization. Given the parameters and weights, each individual receives a fitness value. Fitness is a criterion that shows the goodness of individuals. Individuals with better fitness are theoretically better members of the generation, achieving better performance in the system. Optimization algorithm continues the cycle of creation and evaluation of generations, until a termination condition is met. In each generation, with respect to fitness values, 'selection' operator selects individuals and after application of 'crossover' and 'mutation' operators, next generation is created. An advantage of the GA is that each member of each generation represents a solution to the system under consideration, although not necessarily an optimum solution. To reach an optimum solution the algorithm must be run for many repetitions (*Fig. 1*).

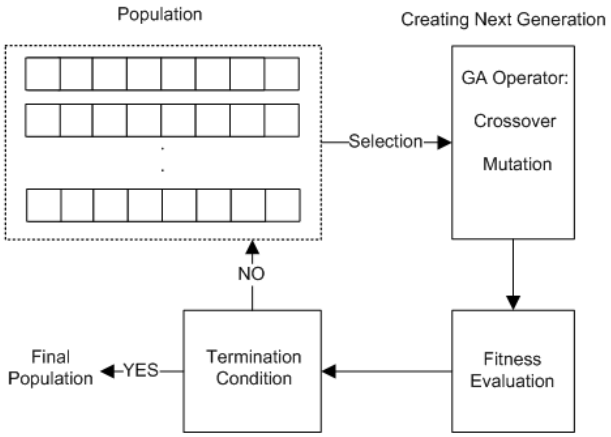

Figure 1. *Flowchart of a typical genetic algorithm (GA)*

Implementation

In this study we propose a method for detection of horizontal scans based on network flows using GA. Two parameters are used for each member of the GA: (a) total number of connections between each source and connected destinations ('Des_{count}'). (b) Total number of ports of an IP a source is connected to ('Prt_{count}'). These two parameters are optimized using GA.

Additionally we had an inclusion criterion for the number of packets that are sent from a source to a destination ('Pkt_{count}'). When a source is scanning a destination port, it sends small number of packets, therefore we included only those flows with less than 8 packets to reduce the number of false-alarms.

We used DAPRA 1999 dataset to simulate network traffic. To read the data out of DARPA and to create the flows, we used Softflowd (www.mindrot.org/projects/ softflowd/). Softflowd is an open-source software that can analyze network traffic. It listens to a network interface and tracks network flows. More importantly it creates the flows. Each flow contains source IP, source port, destination IP, destination port and the protocol. We parse the information stored in each flow to extract the parameters that we are interested in: which destinations a source is connected to ('Des_{count}') as well as to which ports ('Prt_{count}') and how many packets are transferred in this connection ('Pkt_{count}'). The Pkt_{count} is used for inclusion criterion. Also, we defined an array called 'Sourceipflows' to detect horizontal scans. Each row of this array contains information regarding to a specific source such as Pkt_{count}, Prt_{count}, Des_{count}. Using a weighting formula ('$p = \alpha.Des_{count} + \beta.Prt_{count}$') we evaluate the flow. The α and β are two genes constructing the chromosome that are randomly initialized. We categorise the flow as an attack if it exceeds a certain threshold ($p > Atk_{thr}$). We calculated the fitness of each row of the array using sensitivity index (d') in order to have a measure in which hit and false-alarm rates are considered in one measure, as both higher rate of false alarm and lower rate of hit hinders the fitness similarly (Green and Swets, 1966, Macmillan and Creelman, 2004). A higher *d'* corresponds to a better performance.

GA aims to optimise the two scaling factors α and β. α and β values for the best member of the last generation is considered for real applications. The GA is run using the settings mentioned in *Table 1*.

Table 1. *Settings of the genetic algorithm (GA) in our setup*

Parameter	Value
Members in each generation	100
Number of generations	100
Rate of cross-over	20%
Rate of mutation	30%
Survival of best members	10%
Survival of random members	20%
Generation of new members	20%

Results

The optimization performance of the GA is showed in *Fig. 2.* It shows how *d'* is optimized through the generations. *Table 2.* shows the comparison of our method with Snort. The scaling values for the best member of the last generation are used for this comparison.

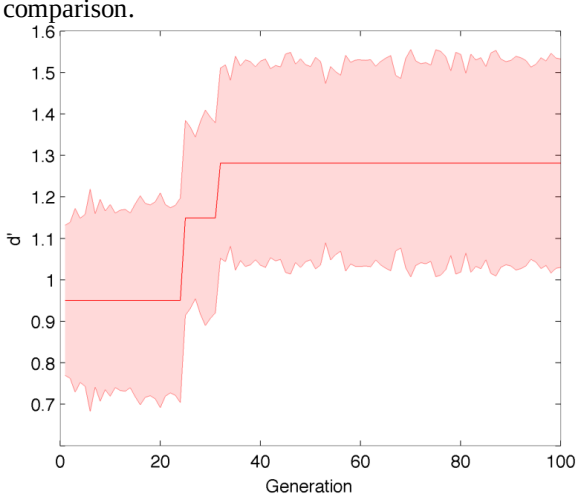

Figure 2. *Optimization using genetic algorithm (GA). The shaded are represents one standard deviation of d'*

Table 2. *Comparison between our method and Snort*

	GA	Snort
Hit rate	32.03%	34.02%
False alarm rate	20.03%	46.07%

Conclusions

Before committing an attack, the attacker must identify susceptible points of the target network. This is done using 'scanning'. Most of the scanning detection methods are based on thresholding (Grégr, 2010, Moon et al., 2010, Sekar et al., 2006). The challenge facing thresholding methods is proper adjustment of the threshold value to achieve an optimum performance. A large threshold leads to large misses and a small threshold leads to large false-alarms (Sekar et al., 2006). We used GA to optimize two scaling values to adjust them appropriately according to a fixed threshold to achieve a close to optimum performance. This way there is no need to have a variable threshold, as its variance can be reflected in α and β.

Comparison of our method with Snort showed that hit rates are comparable while our method achieved significantly lower rate of false alarms, leading to a better performance.

One draw-back of our method is that we need a quite huge amount of storage capacity to store information of flows. Perhaps application of some sort of information coding can be beneficial in this matter.

References

1. Bankovic, Z., Moya, J. M., Araujo, Á., Bojanic, S., Nieto-Taladriz, O. (2009): A Genetic Algorithm-based Solution for Intrusion Detection, Journal of Information Assurance and Security, 4: 192-199.
2. Bhuyan, M. H., Bhattacharyya, D., Kalita, J. (2011): Surveying port scans and their detection methodologies, The Computer Journal, 54: 1565-1581.
3. Crosbie, M., Spafford, G. (1995): Applying genetic programming to intrusion detection, Working Notes for the AAAI Symposium on Genetic Programming, Cambridge, MA: MIT Press, 1-8.
4. Goldberg, D. E. (1989): Genetic algorithms in search, optimization, and machine learning.
5. Gong, R. H., Zulkernine, M., Abolmaesumi, P. (2005): A software implementation of a genetic algorithm based approach to network intrusion detection, Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN), IEEE, 246-253.
6. Green, D. M., Swets, J. A. (1966): Signal detection theory and psychophysics, Wiley New York.
7. Grégr, M. (2010): Portscan detection using NetFlow data, Proceedings of the 16th Conference Student EEICT, Faculty of Information Technology BUT, 229-233.
8. Jung, J., Paxson, V., Berger, A. W., Balakrishnan, H. (2004): Fast portscan detection using sequential hypothesis testing, IEEE Symposium on Security and Privacy, IEEE, 211-225.
9. Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., Das, K. (2000): The 1999 DARPA off-line intrusion detection evaluation, Computer Networks, 34: 579-595.
10. Macmillan, N. A., Creelman, C. D. (2004): Detection theory: A user's guide, Lawrence Erlbaum.
11. Moon, H., Yi, S., Cho, K. (2010): A Modified Multi-Resolution Approach for Port Scan Detection, Global Telecommunications Conference (GLOBECOM), IEEE, 1-5.
12. Riquet, D., Grimaud, G., Hauspie, M. (2012): Large-scale coordinated attacks: Impact on the cloud security, Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), IEEE, 558-563.
13. Sazzadul Hoque, M., Mukit, M. A., Abu Naser Bikas, M. (2012): An Implementation of Intrusion Detection System Using Genetic Algorithm.
14. Sekar, V., Xie, Y., Reiter, M. K., Zhang, H. (2006): A multi-resolution approach forworm detection and containment, International Conference on Dependable Systems and Networks, IEEE, 189-198.
15. Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., Stiller, B. (2010): An overview of IP flow-based intrusion detection, Communications Surveys & Tutorials, IEEE, 12: 343-356.

16. Sridharan, A., Ye, T., Bhattacharyya, S. (2006): Connectionless port scan detection on the backbone, 25th IEEE International Performance, Computing, and Communications Conference (IPCCC), IEEE.
17. Srinivasa, K. (2012): Application of Genetic Algorithms for Detecting Anomaly in Network Intrusion Detection Systems, Advances in Computer Science and Information Technology, 582-591.
18. Srinivasa, K., Chandra, S., Kajaria, S., Mukherjee, S. (2011): IGIDS: Intelligent intrusion detection system using genetic algorithms, World Congress on Information and Communication Technologies (WICT), IEEE, 852-857.
19. Staniford, S., Hoagland, J. A., McAlerney, J. M. (2002): Practical automated detection of stealthy portscans, Journal of Computer Security, 10: 105-136.

4/2/2013