Indian Journal of Science and Technology, *Vol 9(45), DOI: 10.17485/ijst/2016/v9i45/106506, December 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

HTTP Botnet Defense Mechanism using System Dynamics based Genetic Algorithm

Seena Elizebeth Mathew* and A. Pauline

Department of Computer Science and Engineering, MVJ College of Engineering, Near ITPB, Whitefield, Bangalore-560 067, Karnataka, India; Seena.em@mvjce.edu.in

Abstract

Objectives: The system which is under the control of Bot master is called Bot. Botnet refers to the network of bots. Hypertext Transfer Protocol (HTTP) Botnet use HTTP protocol for communication. Findings: HTTP Botnet is difficult to detect since their features are somewhat similar to normal HTTP traffic[1]. Genetic algorithm Based detection method results in better analysis of botnet attacks. However, it sets the initialization pool by picking the values randomly and can assure only less false positive rate. **Novelty:** This paper proposes System Dynamics (SD) based Genetic Algorithm for improving the efficiency of Genetic algorithm and hence the botnet detection.

Keywords: Genetic Algorithm, HTTP Botnet, Layered Detection, System Dynamics

1. Introduction

Nowadays cyber-attacks are increasing day by day and it is caused by Botnet. Botnet is a collection of compromised systems under the control of Bot master.

A representation of history of malicious botnets is shown in Figure 1[2]. The proposed system uses SD based genetic algorithm for defending HTTP Botnet attacks. The system consist of a packet capturing module, layered detection module, genetic algorithm module, firewall filtering module and a SD module. The incoming packets from external network are captured by packet capturing module. Then packet capturing module will give these packets to layered detection module. A Layered detection module consists of different layers. The function of each layer to identify attack based on threshold values. Genetic Algorithm is used to calculate the threshold values for each layer.

In Automated Layered Detection, manual threshold values drive the system. Efficiency of the system depends on the manual threshold value[3].

Initialization, selection, mutation, crossover and best n selection are different phases of Genetic algorithm.

Degree of periodic repeatability can be used to check whether the respective sender is bot or not, but the time complexity will be more[7–9]. Choi proposed detection of botnet by analysing traffic and made the comparison between Bot traffic and legitimate traffic[10–12].

Genetic algorithm generates the individuals fitting for the current envornment[14–16]. It generates the results with high false positive rate. Various types of Genetic algorithms are there. Depending up on the application type of genetic algorithm will be selected[18,19]. Simple genetic algorithm, Parallel Genetic Algorithm (PGA), Distributed Genetic Algorithm (DGA), adaptive genetic algorithm, messy genetic algorithm and so on[20,21]. Fuzzy knowledge base can be used to calculate such as population size, crossover rate and mutation rate[22].

2. Proposed System

2.1. Problem Statement

HTTP botnet use HTTP protocol for communication. Different steps in Botnet attack are given in Figure 2. But this HTTP traffic features are somewhat similar to ordinary traffic (features of tcp packets). Hence it is very difficult to detect.

**Author for correspondence*

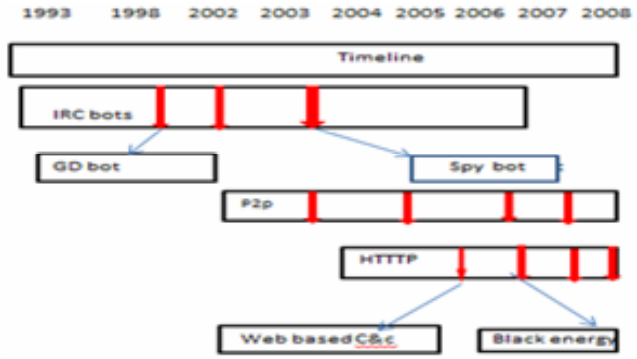

Figure 1. History of malicious bots mathew.

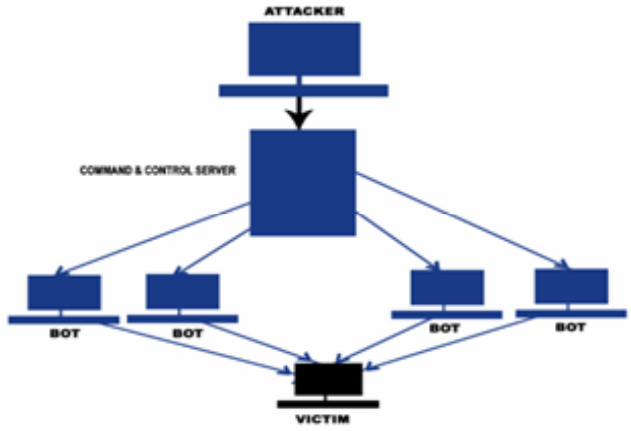

Figure 2. Botnet attack mathew.

2.2. Solution Methodology

The proposed system makes use of SD based genetic algorithm approach for HTTP botnet defence.

In Genetic Algorithm based Layered detection initialization pool is set up picking the values randomly. Hence it can assure only less false positive rate. So a system for setting the initialization pool is highly required.

The SD is a method invented by American Massachusetts Institute of Technology's Professor J.W. Forrester.

Different steps in SD are[23,24]:

1. Generate the casual loop diagram by analyzing the problem.
2. Defining the system variables and establishing the stock and flow diagram.
3. Designing SD equation
4. Setting model parameter and start the simulation

2.3. Schematic Model

A block diagram model of SD based Genetic Algorithm shown in Figure 3.

2.3.1 Packet Capturing Module

When the packets arrive at the system these packets are captured by packet capturing module based on the firewall lists and given to detection module.

2.3.2 Layered HTTP Botnet Detection

The proposed system uses 4 layers. Each layer maintains a specific genetic threshold value. Packet count is compared against threshold value. Attack is reported if count is larger than the threshold value. Flow chart is given in Figure 4.

2.3.3 Genetic Algorithm Module

The step of Genetic Algorithm is shown Figure 5. Different steps are.

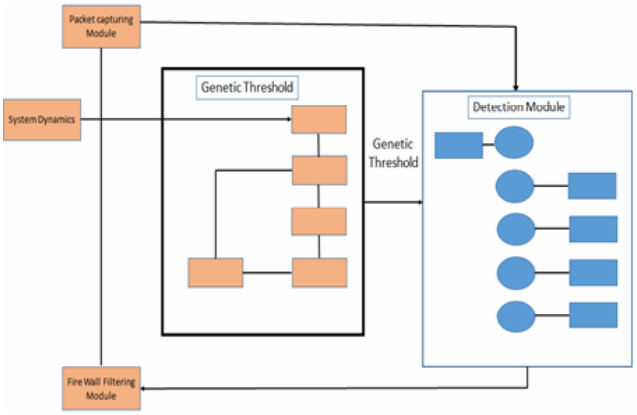

Figure 3. SD based genetic algorithm for HTTP botnet detection mathew.

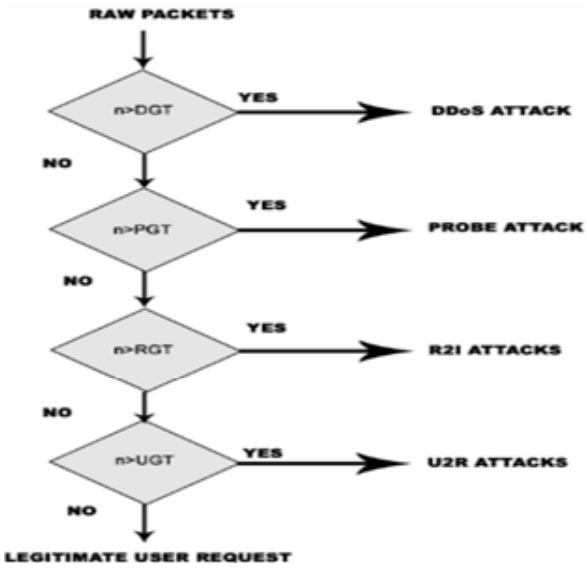

Figure 4. Layered detection module mathew.

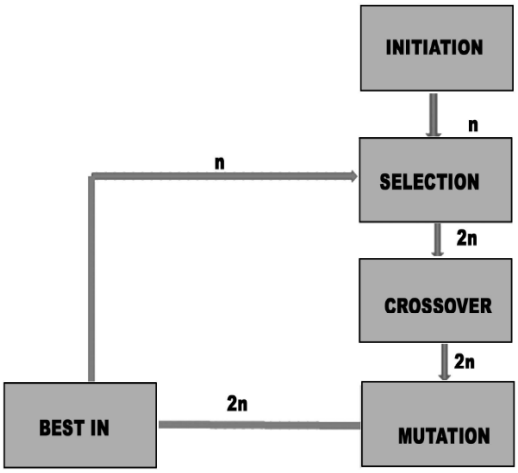

Figure 5. Genetic algorithm flowchart mathew.

2.3.3.1 Initialization

Initialization population is created by randomly picking set on n threshold values for each attack or it can be taken from previous attack history.

2.3.3.2 Selection

For the first iteration selection pool is generated from the initialization pool. It will copy n individuals from the initialization pool and calculate each row's rank and the one with high fitness is set as $n+1^{th}$ row. From the second iteration onwards it takes the population from best n selection phase.

2.3.3.3 Cross Over

Cross over phase first copies first n rows from selection phase. Then it selects 2 rows randomly from the current population and crosses over the features of these selected rows and set it as the $n+1^{th}$ row. Like this cross over phase generate 2n rows.

2.3.3.4 Mutation

Mutation phase first copies first n rows from crossover phase. Then it selects 1 row randomly from the current population and mutates any one or number of the features of this selected row and set it as the $n+1^{th}$ row. Like this mutation phase generate 2n rows

2.3.3.5 Best N Selection

This phase takes input from crossover phase. Calculate the fitness of each row using the fitness calculating equation and assign ranks to these rows. Then sort these rows according to their ranks.

These four steps will be executed for n number of times. Result will be more optimized as we increase the number of iterations.

2.4 SD Module

In this system, SD module is used to set the initialization pool. Steps are:

2.4.1 Causal Loop Diagram

Casual loop diagram of botnet attack detection is given in Figure 6. Success refers to the probability of getting correct result using the particular threshold. Failure means the probability of getting incorrect results. Rank is calculated using the equation.

$$Rank/Weight = \frac{(Success - Failure)}{n} \quad (1)$$

2.4.2 Stock and Flow Diagram

The stock and flow diagram of bot attack is shown in Figure 7. There are three state variables in all. SUCCESS, FAILURE, TOTAL .RANK/WEIGHT, an auxiliary variable, as per Eqn.1. SCT is the success rate. FLT is the failure rate.

2.5 Firewall Filtering Module

The Firewall performs filtering operation based on black list and Grey list.

2.5.1 Grey List

If attack is detected from an IP address for the first time, we can't predict whether it is attacker or legitimate user.

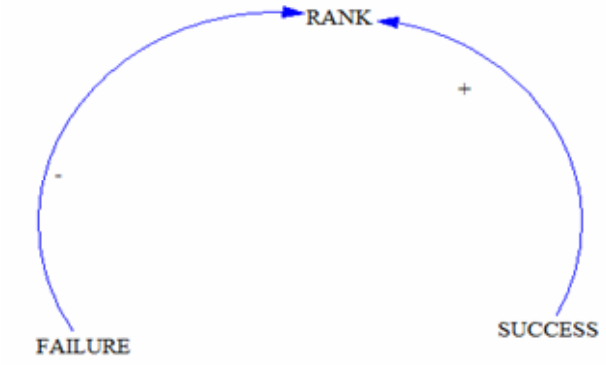

Figure 6. Casual loop diagram mathew.

So we will add this to Grey list and starts monitoring the activities of that particular IP.

2.5.2 Black List

If attack is detected from an IP in the grey list then we can confirm the presence of the attacker. So that IP will be added to the Black list.

3. Comparison with other Systems

3.1 Fuzzy Based Genetic Algorithm

Dynamic parametric genetic algorithm is given in the Figure 9.

Time complexity for this method will be more.

3.2 Bernoulli's Randomised Genetic Algorithm

Here initialization pool is designed by picking the values randomly. With 4 iterations set of threshold values are given in Figure 10.

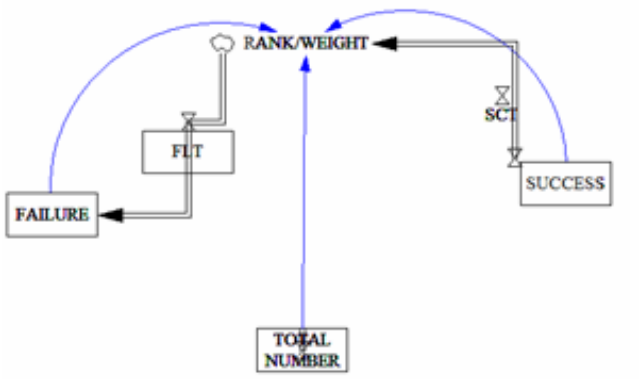

Figure 7. Stock and flow diagram mathew.

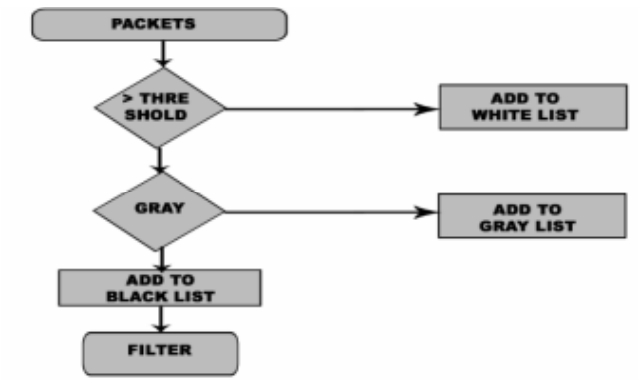

Figure 8. Firewall filtering mathew.

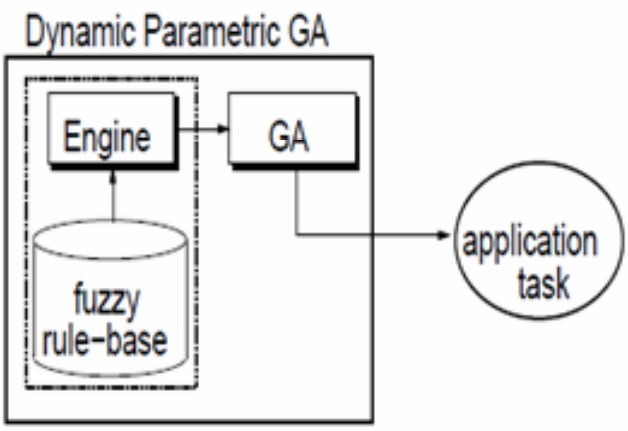

Figure 9. Dynamic parametric genetic algorithm mathew.

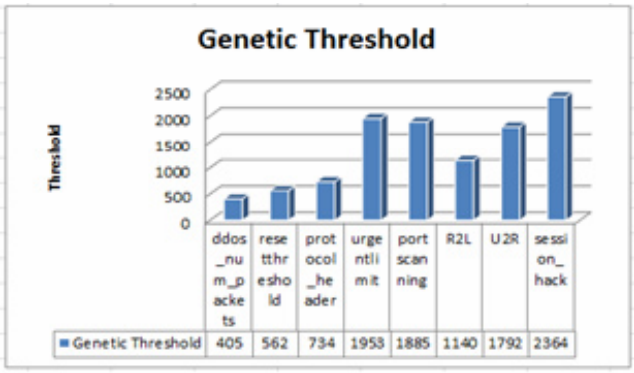

Figure 10. A set of genetic threshold values with 4 iterations mathew.

So as we increase the number of iterations we will get more optimized threshold values. But as the number of iterations increases time complexity will be increased.

4. Conclusion

System Dynamic Based Genetic algorithm method to detect and filter http botnet attack has been studied.

HTTP Botnet defence Mechanism Based on System Dynamics and Genetic Algorithm is for defending HTTP Botnet attacks. In Randomized Genetic Algorithm Based method[1], initialization pool is designed randomly that will affect the system efficiency. Conversely, In SD Based genetic algorithm, Initialization pool is designed using a system dynamic's concept. So that dynamicity of the attacker will be considered. Hence it will produce efficient detection.

It is further opined that rather than using simple GA, other types of Genetic algorithm and enhanced convergence can lead to more optimized results.

5. Acknowledgment

The authors wish to thank the CSE department for their support and help in completing this work.

6. References

1. Koo TM, Chang HC, Wei GQ. Construction p2p firewall HTTP-botnet defence mechanis. Proceedings of IEEE International Conference on Computer Science and Automation Engineering, IEEE Xplore Digital Library; 2011 Jul 14.
2. Mathew SE, Ali A, Stephen J. Genetic algorithm based layered detection and defence of HTTP botnet. ACEEE International Journal on Network Security. 2014 Jan; 5(1).
3. Mathew SE, Ali A. Automated layered HTTP botnet defence mechanism. International Journal of Scientific and Engineering Research. 2013 Aug.
4. Abdullah B, Alghafar IA, Salama GI, Alhafez AA. Performance evaluation of a genetic algorithm based approach to network intrusion detection system. International Conference on Aerospace Sciences and Aviation Technology; 2009 May.
5. Bankovic Z, Stepanovic DA, Bojanic S, Taladriz ON. Improving network security using genetic algorithm approach. Computers and Electrical Engineering. 2007; 33:438–51.
6. Li W. A genetic algorithm approach to network intrusion detection. SANS Institute, USA; 2004.
7. Taylor BN, Kuyatt CE. Guidelines for evaluating and expressing the uncertainty of NIST measurement results. National Institute of Standards and Technology. 1994 Sep. p. 1–20.
8. Dallal GE. Degree of feedom [Internet]. 2007 [cited 2007 May]. Available from: h t t p : / / www.tufts.edu/-gdallaIl-dof.htmL.
9. NIST/SEMATECH, e-handbook of statistical methods; 2003 Jun.
10. Jones. Botnets: detection and mitigation. Federal Computer Incident Response Center (FEDCIRC); 2003 Feb.
11. Cooke E, Jahanian F, Pherson DC. The zombie roundup: understanding, detecting, and disturbing botnets. In the 1st workshop on steps to reducing unwanted traffic on the internet (SRUTI '05); 2005 Jul.
12. Barford P, Yegneswaran V. An inside look at botnets. Special Workshop on Malware Detection, Advances in Information Security, Springer Verlag; 2006.
13. Choi H, Lee H, Lee H, Kim H. Botnet detection by monitoring group activities in DNS traffic. 7th IEEE International Conference on Computer and Information Technology (ICCIT); 2007. p. 715–20. Young M. The Technical Writer's Handbook. Mill Valley, CA: University Science; 1989.
14. Mitchell M. An introduction to genetic algorithms. Massachusetts Institute of Technology (MIT) Press; 1996.
15. Goldberg DE, Smith RE. Nonstationary function optimization using genetic algorithms with diploidy and dominance. In Grefenstette JJ, editor, Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Erlbaum Associates; 1987. p. 59–68.
16. Koza JR. Genetic programming: on the programming of computers by means of natural selection. MA:MIT Press; 1992; Cambridge.
17. Harik GR. (1995). Finding multimodal solutions using restricted tournament selection. In Eshelman LJ (ed.). Proceedings of the Sixth International Conference on Genetic Algorithms. 1995. p. 24–31; San Mateo. CA:Morgan Kaufmann Publishers.
18. Whitley D. The GENITOR algorithm and selection pressure. In Schaffer JD, editor. Proceedings of the Third International Conference on Genetic Algorithms; 1989. p. 161–21; San Mateo. Morgan Kaufmann; 1989.
19. Blickle D, Thiele L. A comparison of selection schemes used in genetic algorithm [Computer Engineering and Communication Networks Lab TIK thesis]. Gloriastresse 35, 8092 Zurich ,Switzerland, Swiss Federal Institute of Technology (ETH); 1995 Dec.
20. Sivanandam, Deepa SN. Introduction to genetic algorithms. Springer-Verlag Berlin Heidelberg; 2008.
21. Whitley D. (1988). GENITOR: a different genetic algorithm. In Proceedings of the Rocky Mountain Conference on Artificial Intelligence. Denver Colorado; 1988. p. 118–30.
22. Lee MA, Takagi H. Dynamic control of genetic algorithms using fuzzy logic techniques. Proceeding of 5th Internation Conference on Genetic Algorithms (ICGA'93), Urbana-Champaign, IL; 1993 Jul 17–21. p. 76–83.
23. Shan KH, Qing ZM, Jun T, Yuan LC. The research of simulation for network security based on system dynamics. Fifth International Conference on Information Assurance and Security; 2009.