International Journal of Grid and Distributed Computing
Vol. 9, No. 1 (2016), pp.117-126
http://dx.doi.org/10.14257/ijgdc.2016.9.1.12

An Improved Dendritic Cells Algorithm for Detecting P2P Bots

Shoubao Su[1], Yu Su[2], Mingjuan Xu[3*] and Xianjin Fang[4]

[1]*School of Computer, Jinling Institute of Technology, Nanjing 211169, China;*
[2]*School of Software, Nanjing University, Nanjing 210093, China;*
[3]*School of Information Engineering, West Anhui University, Lu'an 237012, China;*
[4]*School of Computer Science & Engineering, Anhui University of Science and Technology, Huainan 232001, China*
shoubo@jit.edu.cn, suyu01@163.com, xmj8217@wxc.edu.cn, xjfang@aust.edu.cn

Abstract

Presently P2P-controlled bots has become an increasing threat to our network security due to the fact that P2P bots do not have a centralized point to shutdown or trace back, thus making the detection of P2P bots is very difficult. To enhance the detection rate, a new model to detect P2P bots on an individual host is proposed by improving the dendritic cells algorithm (IDCA). In the proposed approach, the raw data for P2P bot detection is obtained via APITrace tool. The processes ID are mapped into the antigens, and the behavioral data created by the processes are mapped into the signals, which are the time series input data of IDCA, are used to implement data fusion and correlation. The test experimental results show that the proposed method is effective to detect P2P-controlled bots on the host with low false positives.

Keywords*: P2P-controlled bots, dendritic cells algorithm (DCA), data fusion, anomaly coefficient*

1. Introduction

Botnet is one of the most considerable issues in the world. As P2P (Peer-to-Peer) sharing software is widely applied, it causes some serious problems, such as network traffic congestion, unwanted file sharing by computer viruses that abuse P2P software, and so on[1-2]. The bot program using protocol can be categorized into four kinds of type, including IRC bots, P2P bots, DNS bots and HTTP bots. The attackers start to use P2P networks in order to control their botnets. By using this approach, the bots can contact other bots without having a centralized point for their command and control (C&C) structure. In a P2P network, each node acts as client-server which provides bandwidth, storage and computational power[3]. Using this approach, bots are able to communicate with other bots by downloading files or commands from other bots' machines and performing different activities. In comparison to IRC structures, everyone can join a P2P network, thus, the more peers acting as bots, the more powerful the botmaster can be. In addition, it will be hard to detect and shut down the botnet when security people would need to isolate each machine[4].

Detecting bot behaviors in networks is an important topic in network security area. The problem of detection of P2P botnets has been denounced as one of the most difficult ones. The majority of the detection proposals available at present are based on monitoring network traffic to determine the potential existence of command-and-control communications (C&C) between the bots and the botmaster. Some traditional methods using payload analysis or signature-based detection scheme are undesirable in large

* Corresponding author: M.J. Xu (xmj8217@wxc.edu.cn), Tel: +86-564-330 7529, Fax: +86-564-330 7139.

ISSN: 2005-4262 IJGDC

amount of traffic. Also there is a privacy issue with looking into payloads. Schoof [5] analyze different peer-to-peer bots such as Sinit and Nugache to examine the behavior of these bots. In their analysis, they note that some peer-to-peer bots communicate on a fixed port. They also discover that some of these bots generate a large number of destination unreachable error messages (DU) and connection reset error messages while trying to connect to other peers. Wang [6] explain some of the features and challenges when dealing with the Nugache P2P botnet. Hammadi [7] conclude that there is no static IDS that will detect Nugache traffic. Huang [8] present a method to analyze and mitigate P2P botnet. They develop ways to mitigate storm worm and introduce an active measurement technique to enumerate the number of infected hosts.

Other researchers analyze different peer-to-peer bots such as Storm bot (Peacomm) where a number of emails are spammed to many accounts holding an executable attachment[1,3]. Houmansadr [9] and Shin[10] try to locate the zombie nodes activities in P2P network by their retrieval of hashes and the control of a large group of network computers. They claimed that if the client within the controlled network searches for hash used by malware, it must be a zombie node. A detailed description of Peacomm is presented by Ref[11]. They also investigate how to detect the Storm bot by using a BotHunte, which tracks the two-way communication flows between internal and external entities to find the infected host[12].

The immune system is a key component in the maintenance of host homeostasis, in which cells known as dendritic cells (DCs) are playing in key roles that are responsible for the initial detection of invading microorganisms and policing the tissue and organs for potential invaders in the form of pathogens. Inspired from the abstract model of natural DCs behaviors, a new artificial immune system based the functions of DCs, called the Dendritic Cell Algorithm (DCA) [9], was proposed by incorporating the principles of immunology with the “danger theory”[16]. The DCA has been successfully applied to many fields, particularly in computer security related, intrusion detection, port scan, botnet detection[15], and a classifier for robotic security[13,17], data analysis and plants recognition[18], and so on. M. Mokhtar [13] have modified the DCA to detect and circumvent the errors in a resource constrained micro-controller onto the robotic unit during operation. Rodríguez-Gómez[13] introduced a detection scheme based on modeling the evolution of the number of peers sharing resources over time, which allows to detect abnormal behaviors associated to parasite P2P botnet resources in a P2P network.

Nowadays, P2P-controlled bots has become an increasing threat to our network and information security. Detecting and controlling traffic of bots is an important issue to solve these problems. In this paper, we improve the dendritic cells algorithm(DCA)[13] to detect P2P bots on the infected machine by data fusion and correlating bots’ behavioral attributes. A Peacomm (Storm P2P bot) is used as a case study. The data fusion refers to fusing all kinds of time series data produced by bot process, correlation refers to correlating results of data fusion with bot process. This algorithm does not need a pre-defined bots’ signatures in order to detect this kind of bots.

The rest of this paper is organized as follows. Next section gives some brief descriptions of the DCA algorithm. Then an improved DCA is proposed to detect P2P controlled bots in section 3, followed by the experimental tests and analysis in section 4. Finally, the conclusions are given in section 5.

2. Dendritic Cell Algorithm (DCA)

The purpose of a DC algorithm is to correlate disparate data-streams in the form of antigen and signals and to label groups of identical antigen as ‘normal’ or ‘anomalous’. The algorithm provides information representing how anomalous a group of antigen are, not simply if a data item is anomalous or not. This is achieved through the generation of

an anomaly coefficient value, termed MCAV (*Mature Context Antigen Value*). The labelling of antigen data with a MCAV coefficient is performed through correlating a time-series of input signals with a group of antigen. To initiate maturity, a DC must have experienced signals, and in response to this express output signals. As the level of input signal experienced increases, the probability of the DC exceeding its lifespan also increases. The level of signal input is mapped as a CSM (C*ostimulatory output signal*). Once CSM reaches a 'migration' threshold value, the cell ceases signal and antigen collection and is removed from the population for analysis. The generic representation of the DCA is rewritten as shown in Figure 1.

Algorithm 1: Pseudocode of the DCA algorithm

```
input : Sorted antigen and signals (PAMP,DS,SS)
output: Antigen and their context (0/1)
Initilize DC;
while CSM output signal < migration threshold do
   get antigen;
   store antigen;
   get signals;
   calculate interim output signals;
   update cumulative output signals;
end
cell location update to lymph node;
if semi-mature output > mature output then
   cell context is assigned as 0;
else
   cell context is assigned as 1;
end
kill cell;
replace cell in population
```

Figure 1. The Pseudocode of the DCA Algorithm

The primary components of a DC-based algorithm are as follows[11,16]:

1) Individual DCs with the capability to perform multi-signal processing.
2) Antigen collection and presentation.
3) Sampling behavior and state changes.
4) A population of DCs and their interactions with signals and antigen.
5) Incoming signals and antigen, with signals pre-categorized as a PAMP (*Pathogenic Associated Molecular Pattern*), danger, safe, or inflammation.
6) Multiple antigen presentation and analysis using "types" of antigen.
7) Generation of anomaly coefficient for various different types of antigen.

3. Improved DCA for Detection of P2P Bots

DCA's input is the time series data consisting of signals and antigens. It performs the functionality of data fusion and causal correlation. The scheme for detection of P2P bots using DCA is illustrate in Figure 2. This process includes four phases: data collection, signal mapping, data fusion, correlation and analysis.

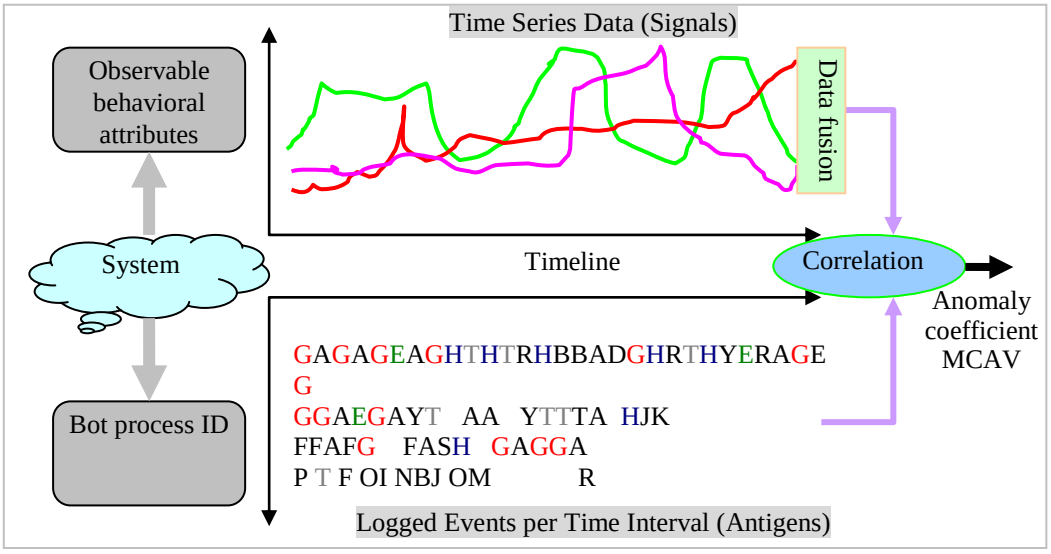


Figure 2. The Principle of IDCA for Detecting P2P Bots

3.1 Data Collection

Signals and antigen are passed as an input to the DCA. To collect signals and antigen, an APITrace[10] is used. We assume that the Peacomm bot is already installed on the victim host. We use an interception program APITrace to record the required behavioral attributes and to intercept and capture specified function calls executed by the monitored processes. These data are then processed, normalized and streamed for signal mapping. In terms of the function calls intercepted, different types of function calls are used as an input to the algorithm. These function calls include Communication functions (*e.g.* send, recv), File access functions (*e.g* ReadFile, WriteFile), Registry access functions (*e.g.* RegOpenKey, RegQueryValue) and Keyboard status functions (*e.g.* GetKeyboardState, GetAsynKeyStat).

3.2 Signal and Antigen Mapping

In order to detect P2P bots using DCA, the behavioral attributes of the bots in a monitored system must be mapped into the three signal categories, that is, PAMP, DANGER and SAFE.

3.2.1 PAMP Mapping: For P2P-based bot detection, the rate of change of invocation of selected API function calls used for keylogging activity is not enough to represent the measure of a monitored system status. According to the preliminary observation of P2P bots, the PAMP signal should be related to the rate of change of three fields of network status. These fields are the value of destination unreachable (DU), failed connection attempts (FCA) and reset connections (RST). So PAMP signal is mapped as:

$$PAMP = DU + FCA + RST \quad (1)$$

3.2.2 DANGER Mapping: DANGER signal is derived from the rate of change of number of packets send per second (pkts/sec). We use a logarithmic scale when using DANGER and is derived according to the following formula:

$$DANGER = 25 * \log_{10}(X), 1 \le X \le 10000 \quad (2)$$

where X is the rate of change of the number of packets sent per second. If X exceeds 10000, the value of DANGER is mapped to 100.

3.2.3 SAFE mapping: SAFE signal is derived from the time difference between two outgoing consecutive communication functions such as [(send, send), (sendto, sendto), (socket, socket)]. It is calculated from the following formula:

$$SAFE = 62.41965 * \log_{10}(Y) \quad (3)$$

Where Y is the time difference between two outgoing consecutive communication functions.

Antigen represents potential culprits that are responsible for changes in the status of the monitored system. The need for correlation between antigen and signals is required to define which processes are active when the signal values are modified. The more active the process, the more antigen it generates. Once the function calls are intercepted by APITrace, they are stored and assigned the value of the process ID to which the function calls belong and the time at which they were invoked. After a certain period of time, both signal and antigen logs are combined and sorted based on time. The combined log files are parsed and the logged information is sent to the DCA for processing and analysis.

We use selected API function calls described in section 3.1 with their process ID to represent antigen. These processes are Peacomm bot, Firefox web browser, IceChat (an IRC client). These signals and antigens form the time series input data of dendritic cell (DC).

3.1 Data Fusion

The DCA is an algorithm based on dendritic cells population in which each DC Performs signal processing, calculates *CSM* and *K* with weights. Each DC's outputs in dendritic cells algorithm are defined as:

$$\begin{aligned} CSM &= w_{11} * PAMP + w_{12} * DANGER + w_{13} * SAFE \\ K &= w_{21} * PAMP + w_{22} * DANGER - 2 * w_{23} * SAFE \end{aligned} \quad (4)$$

where $W = \{w_{ij} \mid w_{ij} \in R, 1 \le i \le 2, 1 \le j \le 3\}$ is the weight matrix of signal transformation of DCA.

The mechanism of signal processing in original DCA can be shown as Figure 3. In DC initialization phase, Each DC in DC population is assigned lifespan. During signal processing, the DC's lifespan is repeatedly subtracted by *CSM* signal, while its signal profile is repeatedly increased by the *K* signal until the termination condition, lifespan<=0, is reached.

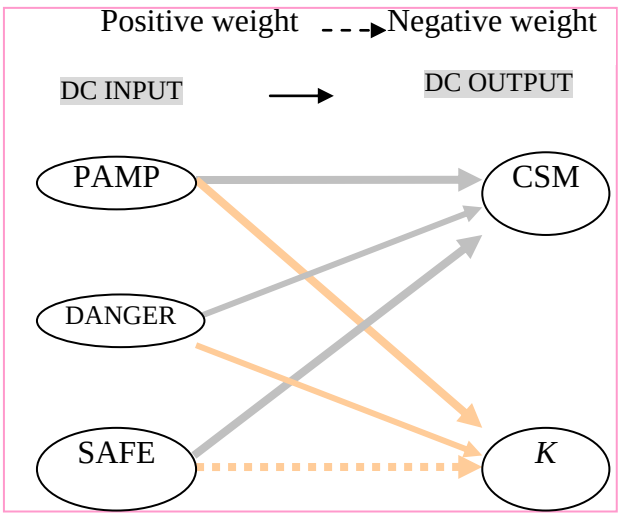


Figure 3. The Mechanism of Signal Processing in Original DCA

3.2 Correlation and Analysis

The *CSM* signal reflects the amount of information a DC has processed, *i.e*. when to make decisions, while the *K* signal is a measure indicating the polarisation towards anomaly or normality, *i.e*. how to make decisions. *CSM* and *K* are used to evaluate the status of the system monitored by the analysis component of the algorithm. This analysis component is also called causal correlation.

The output of each DC_i is store as a pair $(a_{ik}, r_i) \in Antigen \times R$ in a list *LST*, where r_i is the signal profile of a DC when it reaches a termination condition. We also define π1 and π2 as projection functions to obtain the first and second elements of a pair respectively.

Definition 1 (*signal profile update*). The signal profile update function $G : Time \times Population \to R$ is defined as

$$G(t,i) = \begin{cases} 0, & if\ t = 1 \\ \pi 2(Output(t)), if\ \text{satisfying terminati on condition} \\ G(t-1,i) + \pi 2(Output(t)), & otherwise \end{cases} \quad (5)$$

Where *Output(t)* represents the DC's output at a time point *t*. G is repeatedly increased by the *K* signal value until the termination condition is reached.

Definition 2 (*antigen counter*). The antigen counter function $C : N \times Antigen \to \{0,1\}$ is defined as

$$C(j,\alpha) = \begin{cases} 1, if\ \pi 1(LST(j)) = \alpha \\ 0, otherwise \end{cases} \quad (6)$$

Where N=|DC population|, $\alpha \in$ Antigen is an antigen type. The function *C* is used to count the number of instances of antigen type α.

Definition 3 (*signal profile abstraction*). The signal profile abstraction function $R : N \times Antigen \to R$ is defined as

$$R(j,\alpha) = \begin{cases} \pi 2(L(j)), if\ \pi 1(LST(j)) = \alpha \\ 0, otherwise \end{cases} \quad (7)$$

The function ***R*** is used to calculate the sum of all ***K*** values associated with antigen type α.

Definition 4 (*anomaly coefficient calculation*). Given the number of input instances is equal to *n*, the anomaly coefficient calculation function is defined as.

$$\beta = \sum_{j=1}^{n} C(j,\alpha), \gamma = \sum_{j=1}^{n} R(j,\alpha) \Rightarrow K_\alpha = \frac{\gamma}{\beta} \quad (8)$$

A threshold ε can be applied for further classification. An antige n type (*i.e*. process ID in this paper) is classified as anomalous if $K\alpha > \varepsilon$, and normal otherwise.

4. Experiments and Analysis

The aim of experiments is to use the DCA to perform detection of P2P bot which uses different C&C communication protocols.

4.1 Experimental Setup

The experimental environment for the Peacomm bot detection is illustrated in Figure 4. In the experiment two scenarios (S1 and S2) are used as follows:

(1) Inactive (S1): In this session, the Peacomm bot is executed and runs on a monitored host. Other normal applications are also running during this session but there are no activities from the user such as browsing or chatting.

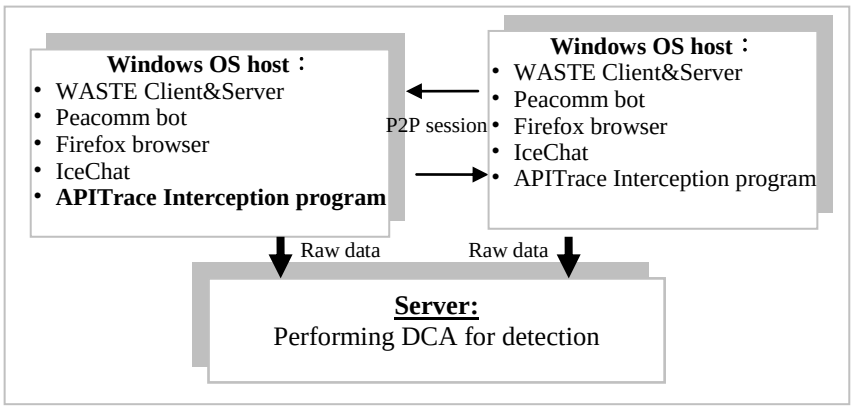


Figure 4. Experimental Environment for Peacomm Detection

(2) Active (S2): In this session, the Peacomm bot is executed and runs on a monitored host. In contrast to (S1), the user uses Firefox for browsing and checking emails as normal activities and uses IceChat for having conversation with other users. In both cases, the APITrace is running to collect the raw data.

4.2 Experimental Results

To evaluate effectiveness of the improved DCA method (IDCA) for detecting bots in a P2P network, we compare the $K\alpha$ (α is an antigen, that is, a process) values of Peacomm with other benign programs (Firefox, IceChat), and the experimental results is listed in table 1.The means Kα value for Peacomm, Firefox and IceChat are respectively shown in Figure 5-7, significant differences are shown for all experiments, where the $K\alpha$ value for the Peacomm bot is higher than the $K\alpha$ values of Firefox and iceChat. If the threshold ε is set 0.5, these processes can be judge as normal or anomaly according to their mean $K\alpha$ values.

Table 1. The Results of the Kα Values Generated from DCA

Experiment scenario	process	Mean Processed Antigens	Mean Kα
PmE1	Peacomm	134985.2	0.5651
	Firefox	2000	0.3737
	icechat	47.2	0.1067
PmE2	Peacomm	136421.6	0.5014
	Firefox	2695.3	0.3996
	icechat	1421.9	0.3372

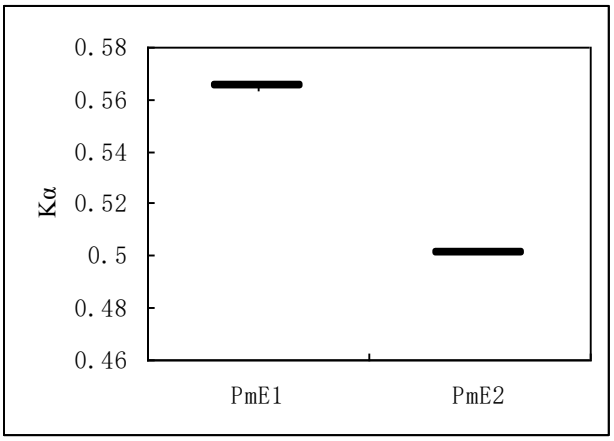


Figure 5. The mean Kα Value of Peacomm Bot via the DCA

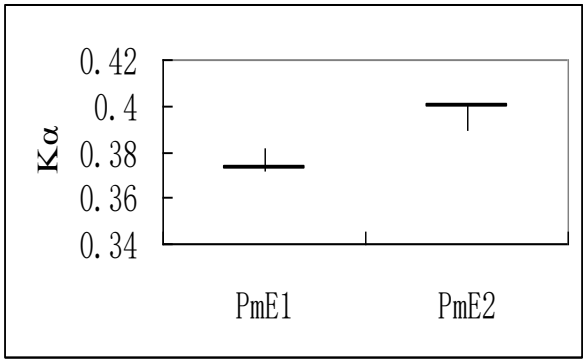


Figure 6. The Mean Kα Value of Firefox Bot via the DCA

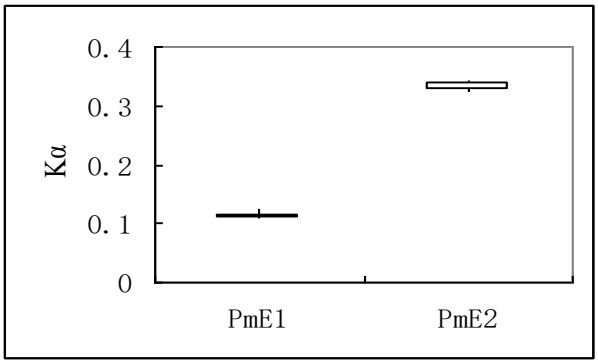


Figure 7. The mean Kα Value of IceChat Bot via the DCA

5. Conclusions

In comparison to IRC bots, P2P bots are more difficult to monitor, detect or shut down as there is no central command and control structure and most of the traffic is encrypted. One way to detect such bots is by monitoring and correlating different activities on a machine. In this paper, we use the IDCA as an intelligent correlation algorithm to correlate different behaviors of normal processes and P2P bots. This correlation of behaviors is based on specifying signals combined with antigen. Peacomm detection case studies is used to measure the performance of IDCA, the results show that the IDCA is

able to classify bots as abnormal processes in comparison to benign processes by generating significant differences in the $K\alpha$ values for both normal and abnormal processes.

Acknowledgements

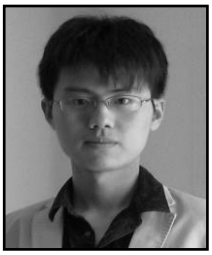
This work is supported by the National Natural Science Foundation of China (Nos.61375121,61402012,61572034), the Provincial Projects of Natural Scientific Research Fund from the Bureau of Education of Anhui Province, China (Nos.KJ2011Z401 for M.J. Xu, a Lecturer of Computer Science, West Anhui University, corresponding author for this article), Anhui Provincial Science Funds for Excellent Youths of Universities (No.2011SQRL150), and sponsored by JIT Scientific Research Program for Introducing Talents.

References

[1] S.C. Sérgio Silva, M.P Rodrigo, Silva, C.G Raquel. Pinto and M Ronaldo, Salles, "Botnets: a survey, Computer Network"s, vol.57, no.2, **(2013)**, pp.378-403.
[2] E.E.A. Ahmed, M.A. Maarof and B.I.A. Barry, "Improving the detection of malware behaviour using simplified data dependent API call graph", International Journal of Security and Its Applications, vol.7, no.5, **(2013)**, pp.29-42.
[3] C.M. Chen, H.C. Lin, "Detecting botnet by anomalous traffic, Journal of Information Security and Applications", **(2014)**, doi: 10.1016/j.jisa.2014.05.002.
[4] D. Dittrich, S. Dietrich, "P2P as botnet command and control: a deeper insight", Proceedings of International Conference on Malicious and Unwanted Software-Malware, pp.213-222, **(2008)**, October 7-8; Alexandria, Virginia, USA.
[5] R. Schoof, R. Koning, "Detecting peer-to-peer botnets", University of Amsterdam, Amsterdam, (2011).
[6] K.C. Wang, C.Y. Huang, S.J. Lin, and Y.D. Liu, "A fuzzy pattern-based filtering algorithm for botnet detection, Computer Networks", vol.55, no.15, **(2011)**, pp.3275-286.
[7] A.L. Hammadi, "Behavioural correlation for malicious bot detection". University of Nottingham, UK, **(2010)**.
[8] C.Y. Huang, Effective bot host detection based on network failure models, Computer Networks, vol.57, no.4, **(2013)**, pp.514-525.
[9] A. Houmansadr, N. Borisov, "BotMosaic: Collaborative network watermark for the detection of IRC-based botnets", Journal of Systems and Software, vol.86, no.3, **(2013)**, pp.707-715.
[10] S. Shin, Z.Y. Xu, G.F. Gu, "EFFORT: A new host–network cooperated framework for efficient and effective bot malware detection", Computer Networks, vol.57, no.13, **(2013),** pp.2628-642.
[11] J. Greensmith, "The Dendritic Cell Algorithm", PhD thesis, University Of Nottingham, Nottingham, UK, **(2007)**.
[12] G.H. Yan, D.T. Ha, S. Eidenbenz, AntBot: Anti-pollution peer-to-peer botnets, Computer Networks, vol.55, no.8, **(2011)**, pp.1941-1956.
[13] R.A. Rodríguez-Gómez, G. Maciá-Fernández, P. García-Teodoro, M. Steiner, D. Balzarotti, Resource monitoring for the detection of parasite P2P botnets. Computer Networks, vol.70, nol.5, **(2014)**, pp.302-311.
[14] A. Castiglione, R.D. Prisco, A.D. Santis, U. Fiore, F. Palmieri, "A botnet-based command and control approach relying on swarm intelligence", Journal of Network and Computer Applications, vol.38, no.1, **(2014**), pp.22-33.
[15] W.M. Li, M. Chen, F. Liu, *et al*, "Analysis on the time-domain characteristics of botnets control traffic", The Journal of China Universities of Posts and Telecommunications, vol.18, no.2, **(2011)**, pp.106-113.
[16] F. Gu, J.L. Greensmith, U. Aicklein, "The dendritic cell algorithm for intrusion detection", in Biologically Inspired Networking and Sensing, Edited L. Pietro and V. Dinesh, **(2012)**, pp.236-248.
[17] J.L. Greensmith, U. Aickelin, "The deterministic dendritic cell algorithm", Proceeding of International Conference on Artificial Immune Systems, Phuket, Thailand, **(2008)**, pp.126-137.
[18] Foster Oates R., "The suitability of the dendritic cell algorithm for robotic security applications", University of Nottingham, Nottingham, UK, (2010).
[19] K. Kumari, J. Anurag, S. Dongre, J. Aakriti, "Improving dendritic cell algorithm by Dempster belief theory", International Journal of Computer Engineering and Technology, vol.3, no.2, **(2012)**, pp.415-423.
[20] E. Bendiab, M.K. Kholladi, "Recognition of plant leaves using the dendritic cell algorithm", International Journal of Digital Information and Wireless Communications, vol.1, no.1, **(2011)**, pp.284-292.

Authors

Shoubao Su, He has severed as a postdoctoral fellow at Harbin Institute of Technology (HIT) for years. He received his Ph.D. and M.S. degrees in computer application technology both from Anhui University, China. Now he is a Professor of School of Computer Engineering at Jinling Institute of Technology, China. His research interests focus on applied swarm intelligent computing to the cybernetics security.



Yu Su, He received his B.A. degree in software engineering from University of Electronic Science and Technology of China in 2015. Currently he is a pre-Master student majoring in software engineering, School of Software, Nanjing University, Nanjing, China. His research interests are swarm intelligence and complex network computing.

Mingjuan Xu, She received her B.A. degree and M.A. degree both in computer science from Anhui University of Technology in 2003, and Anhui University in 2009, respectively. She is presently a Lecturer in the School of Information Engineering, West Anhui University, Lu'an, China. Her chief research interests include intelligent computing, data mining, and cyberspace security.

Xianjing Fang. Prof. Fang received his Ph.D. and M.S. degrees in computer application technology both from Anhui University, China. Currently he is a Professor of School of Computer Science and Engineering at Anhui University of Science and Technology, China. His researches include Intelligent Computing and Network Security.