A Novel Search Engine-Based Method for Discovering Command and Control Server

Xiaojun Guo[1, 2, 3, 4], Guang Cheng[1, 2],Wubin Pan[1, 2], Truong Dinhtu[1, 2], Yixin Liang[1, 2]

[1]School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
[2]Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing 211189, China
[3]School of Information Engineering, Xizang Minzu University, Xianyang, 712082, China
[4]XiZang Key Laboratory of Optical Information Processing and Visualization Technology

xjguo@njnet.edu.cn

Abstract. To solve the problem of getting command and control (C&C) server address covertly for malware of Botnet or advanced persistent threats, we propose a novel C&C-server address discovery scheme via search engine. This scheme is com-posed of five modules. The botmaster uses publish module to issue C&C-server IPs in diaries of several free blogs on Internet firstly. Then these diaries could be indexed by search engine (SE). When the infected terminal becomes a bot, it uses keyword production module to produce search keyword and submits some or all these keywords to SEs to obtain the search engine result pages (SERPs). For items in SERPs, the bot uses filtering algorithm to remove noise items and leave valid items whose abstract contain C&C-server IPs. Lastly the bot utilizes extraction and conversion module to extract these C&C-server IPs and translates them into binary format. The experimental results show that our proposed scheme is fully able to discover and obtain C&C-server IPs via various search engines. Furthermore, if we set proper threshold value for SE, it can extract C&C-server IPs accurately and efficiently.

Keywords: Top-*K* Algorithm; Search Engine; Command and Control Server; Botnet; Advanced Persistent Threat (APT)

1 Introduction

Botnets have become one of the biggest threats to Internet security. A typical botnet[1] is a highly controlled platform which consists of many compromised terminals (called *bots*)like smartphone, tablet , or personal computer etc.

The controller of botnet (called *botmaster*) can send commands to these bots through Command & Control (C&C) servers to launch various of network attacks, such as Phishing fraud, E-mail bombing Session Hijacking and DDoS attack[2-4].

Therefore, these C&C servers are the rendezvous points of bots and botmaster. Only if the bots find C&C-server address information (eg. IP address, domain name, URL) can they be controlled and managed by botmaster. Otherwise, these bots have

no threat and practical value[5]. So how to find and get C&C-server addresses for bots is the first step to ensure the whole botnet to work correctly.

There already exist several finding C&C-server address methods. The most common method is to directly hardcode C&C server address (i.e. static IP/domain name) into malware binary code of botnet[6]. But these hardcoded C&C server address will be easily analyzed through reverse analysis, sandbox, etc. The stealthy of this meth-od will become worse as the time goes on. For further improving stealthy, another finding C&C-server address method by means of dynamic DNS service is introduced, such as Domain-Flux[7] and Fast-Flux[11]. The genuine C&C-servers address can be shield by mixed into many seemingly legal but nonexistent domain names or IPs. Unfortunately, the procedure of accessing C&C-servers address will produce a lot of failure DNS queries which seriously expose the bot behavior and cause the detection of C&C domains easily.

Finding C&C-server address method is so critical for botnet that a more stealthy and secure finding scheme is really needed for botmaster. In this paper, inspired by the item structure in Search Engine Result Pages (SERP) as shown in Fig.1, we propose a novel C&C-server Address Finding scheme based on Search Engine, named CAFSE, to satisfy the latency and stealth of C&C-servers address finding process. CAFSE is mainly composed of Publish Module (PM), Keyword Production Module (KPM), Search Module (SM), Noise Item Filter Module (NIFM) and Extraction and Conversion Module (ECM). This paper will describe the mechanism of CAFSE in detail and present the simulation test results.

The remainder of this paper is organized as follows: Section II presents the mechanism of CAFSE. In Section III, we present the simulation test results of our proposed scheme. Finally, Section IV draws the conclusion.

Fig. 1. The valid item and noise item in Google's SERP

2 The Proposed Scheme

The proposed CAFSE scheme is depicted in Fig.2. The botmaster firstly uses PM to issue some C&C-server IP addresses as the text-format content in diaries of several Internet free blogs. These diaries can be indexed by search engine. When the infected terminal becomes a bot, it uses KPM to produce the keyword list and submits some or all keywords in this list to search engines to obtain the SERPs. Then the NIFM removes the noise items and leaves the valid items whose abstract part contain C&C-server IP addresses. Lastly the ECM extracts these C&C-server IP addresses and translates them into binary format. Thus, the bots can directly communicates with C&C-servers indicated by these binary-format IP addresses.

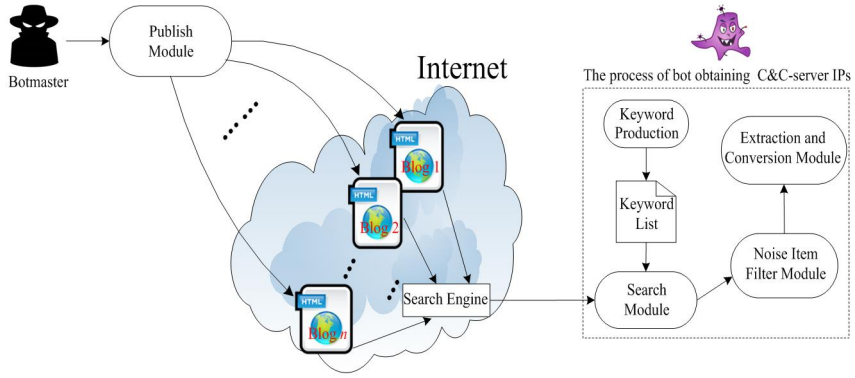

Fig. 2. The mechanism of CAFSE

2.1 Publish Module

In order to provide C&C-server IP addresses for bots, the botmaster uses this module to issue the newest C&C-server IP addresses on the webpage indexed by SE. So how to achieve this function is the main task of PM.

Here we utilize blog, the most common thing on Internet, to implement the Publish Module. The botmaster can use one account name to register N (here, N =10) different free blogs on Internet in advance. When needing to update the C&C-server IP addresses, the botmaster will issue new IPs on the diaries of these registered N Blogs. There are three points that need to be noticed as follows:

Point1: For the convenience of bots to get the C&C-server IPs via SE, the botmaster uses keyword and new C&C-server IP addresses as the title and content of the diary respectively when publishing these new C&C-server IPs in Blog, as shown in Fig.3.

Point2: The number of C&C-server IPs in content part of the diary should be limited, because the abstract of valid item in SERP only can display a small section of the content text of the corresponding diary, not all of it (see Fig.1). This is decided by the abstract generation algorithm of SE. Here, the number is set less than 10.

Point3: If there are a lot of C&C-server IPs (≥10) to be published, the botmaster can divide these IPs into several groups, each of which only contains less than 10 IPs , and then issue each group in one of Blog1~BlogN via the method described in Point1.

Fig. 3. The issued C&C-server IP addresses via the diary of free Blog

2.2 Keyword Production Module

The keyword is the query string that bots submit to SE to get the items including C&C-server IPs from SERPs. It is also the title of the diary where the botmaster issues C&C-server IPs. The produced keyword by KPM should be unique in order to reduce the number of noise items in SERP as far as possible. Meanwhile the produced keyword should be randomness enough to avoid being detected and tracked. Moreover, the time-space cost of KPM should be low enough to do execution because of the limited resource for bots. Here, we employ the Message-Digest Algorithm 5 (MD5)[24] as the KPM and use the MD5 value of date (YYYY-M-D, e.g. "2014-12-6") as the keyword, as shown in Algorithm 1.

Algorithm 1 KeywordProduction ( )

```
1: String Klist[13];  //Keyword List
2: String Year, Day, DataStr;
3: Year Ŷ get the year of current date of victim system;
4: Day Ŷ get the day of current date of victim system;
5: for ( int i=1 ; i <13; i ++)
6: DateStr Ŷ Year+"-"+itoa(i)+ "-"+Day;
       // date format is: YYYY-M-D
7: Klist[i] Ŷ MD5(DataStr);
8: end for
```

In order to avoid producing overmany keywords in a short time, KPM only produces 12 keywords for each day. When bots need to search, they will submit the entire MD5 value of date (i.e. keyword) to SE not a part of it (see Section III. B).

In order to avoid producing overmany keywords in a short time, KPM only produces 12 keywords for each day. When bots need to search, they will submit the entire MD5 value of date (i.e. keyword) to SE not a part of it (see Section III. B).

2.3 Search Module

For different SEs, Search Module constructs retrieval URL (Uniform Resource Locator) with each keyword in keyword list *Klist*[]. Then it submits these URLs to corresponding SE to get the SERPs and extracts each item in SERPs to form item dataset Ω.

Although the retrieval URL (rURL) may include many various parameters for different SEs, the basic format of rURL is nearly the same except the slight difference in parameters name[25]. For example, the Google's basic rURL is "*http://www.google.*com/***search***?**q***=*Keyword&num*=20 ", while which of Baidu is "*http://www.baidu.com/***s***?**word***= *Keyword&rn*=20". From the comparison of these two rURLs, we can see that the parameters "***search***", "***q***" in Google's rURL, "***s***","***word***" in Baidu rURL have the same meanings. They both represent using text search function of Google and Baidu with query entry "*Keyword*". The rURL for other SEs also has the same basic format. In addition, after getting SERPs from SEs, the extraction of each item can be achieved by Jsoup Library[26].

2.4 Noise Item Filter Module

The item in Ω obtained from Search Module can be classified into two types: valid item and noise item, as seen in Fig.1. The valid item is the item whose abstract includes C&C-server IPs. The noise items means the rest items in q except valid items . For example in Fig.1, the first and fourth item are valid items, while the rest items are noise items.

In SERPs, the valid items and noise items appear randomly without any regularity in order. So the NIFM should eliminate the noise items and gather valid items together as much as possible, which can help bots extract C&C-server IPs effectively. Here we use the Top-K query algorithm[27] to implement NIFM to deal with item dataset Ω. The process is as follows:

Firstly, compute the score for each item in Ω. Given set $R=\{I_i:1\leqslant i\leqslant n\}$, where I_i is score vector for i-th item in Ω and $I_i=(s_1, s_2, s_3)$, n is the amount of item in Ω. Because each item is composed of 3 parts: title, link and abstract (see Fig.1), here s_1, s_2, s_3 indicates the score of title, link and abstract of i-th item respectively. As shown in Fig.1, we can find that in the valid item, the keyword may appear in both title and abstract, while the string "*blog*" may appear mostly in link. This feature is obvious for valid item, but not for noise item. Therefore, the score s_1 is the length of keyword included by title, s_2 is the length of string "*blog*" included by link and s_3 is the length of keyword included by abstract.

$$\sum_{e=1}^{v} w_e = 1 \tag{1}$$

Secondly, set the weight vector w. Here, $w=(w_1, w_2, \ldots, w_v)$, $w_e\in[0,1]$, $1\leqslant e\leqslant v$, v is dimension of I_i, and v=3. According to the observation of the items in SERPs, we found that the number of times that keyword and string "*blog*" appears in abstract and

link part of valid item is more than that keyword appears in title. Hence the relation for the corresponding weight of s_1, s_2, s_3 is satisfied $w_1 < w_2 = w_3$.

Lastly, execute the Top-K query procedure on R. For any $e(1 \leqslant e \leqslant v)$, if $I_i[e] \geqslant I_j[e]$, the query function f must be satisfied $f(I_i) \geqslant f(I_j)$. Thus it can be conclude that f should be a monotone increasing function. Here we let f be a weighted-sum function, as shown in formula (2):

$$f(I_i) = \sum_{e=1}^{v} w_e \cdot I_i[e] \tag{2}$$

When executing Top-K query algorithm on R, the top k values among $f(I_1) \sim f(I_n)$ will be returned. So the k items corresponded with these top k values have higher possibility of being the valid items. And the noise items can be filtered through this Top-K query procedure.

2.5 Extraction and Conversion Module

This module uses pattern matching algorithm to find and extract the IP pattern strings from the abstract part of the selected top k items by Top-K query procedure. Then it verifies if IP pattern strings are right or not.

The right IP pattern strings will be converted into binary format IPs in order to facilitate bots to directly access the C&C-servers represented by these IPs.

3 Simulation Results And Discussion

In this section, we present some simulations to evaluate the performance of proposed CAFSE scheme.

Table 1. 10 Free Blogs Used To Issue C&C-server IPs

Number	Blog URL
1	http://blog.163.com/gxjjxg_0617/
2	http://gxjjxg0617.blog.sohu.com/
3	http://blog.csdn.net/gxjjxg_0617
4	http://www.guokr.com/i/0977717367/
5	http://hexun.com/gxjjxg0617/default.html
6	http://blog.tianya.cn/blog-5204529-1.shtml
7	http://gxjjxg0617.blog.51cto.com/
8	http://my.oschina.net/u/2254035/blog
9	http://bbs.chinabyte.com/space-uid-469064.html
10	http://blog.chinaunix.net/uid-29961317-id-4702765.html

We firstly register 10 free blogs on Internet, as shown in Table I. To implement the function of Publish Module, we utilizes the Chrome-v39.0.2171.95m and Plug-in Tab-Snap-v1.2.9[28] to open and login on these 10 free registered blogs, then use the MD5 values of three dates as the diary's titles, several C&C-server IP addresses (each address format is "IP:Port Number" and separated by space) as the diaryôs content and publishes these diaries on 10 Blogs listed in Table I respectively.

The KPM , SM, NIFM and ECM are implemented in Java language and tested on a PC with an Intel Pentium G640 CPU of 2.8GHz, DDR3 SDRAM of 4 GB and Windows 7 (32bit).

3.1 Indexing Time and Quantity

Indexing Time(IT) is period which starts from C&C-server IPs issued by Publish Module until the first valid item appears in SERPs. Indexing Quantity (IQ) denotes the number of valid items in SERPs of one SE. Here we use this two metrics to evaluate the search results generated by the SEs for the 10 Blogs in Table I which contain the issued C&C-server IPs.

For each MD5 value of three dates, we use Search Module to search in Google, Baidu, Bing and Haosou respectively and record the number of valid items in their SERPs at a fixed time in a day. This record process lasts 30 days. The IQ here was the average value of the number of valid items for this 30 days, as shown in Fig.4.

From Fig.4, we can find that the IT of Haosou is 0 day, i.e. the first valid item containing C&C-server IPs is successfully indexed as the same day as when C&C-server IPs were published. The IT of Haosou is the shortest one of 4 SEs. The IT of Google and Baidu is 1day, the worst is Bing whose IT is 6 day. This 4 SEs have big difference in indexing time, but most of them cost more than 1 day. Compared with the four existing C&C-server IP finding schemes for bots (see Section I), in our proposed method the appearance of C&C-server IPs in SERPs of 4 SEs is so slow that bots can't find and obtain the issued C&C-server IPs immediately through SEs. Therefore, this feature is consistent with the requirement that APT[29] and Botnet need malware to possess latency in order to increase their covertness.

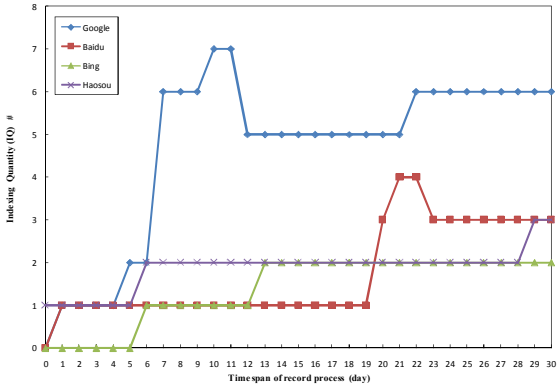

Fig. 4. Valid item number in SERPs of 4 SEs varies with time

In Fig.4, we can also see that the IQ of 4 SEs increase overall as time goes on, especially for Google and Baidu. And the IQ of 4 SEs is tending towards stability. At the time when C&C-server IPs are issued 30 days, the IQ of Google is stable at 6, which means Google presents better indexing effects. By contrast, the IQ of the other three SEs, i.e. Baidu, Haosou and Bing, decreases fewer. The indexing effect of this

three SEs is worse, but the bot still can get the C&C-server IPs from the these few valid items.

3.2 Keyword Length and Direction Impact on Search Effect

We select the first d characters of each keyword as the new keywords to do search test in 4 SEs, record IQs for these new keywords and compute corresponding average value of IQs for the same length new keywords, as shown in left part of Table 2. Note that this test is conducted at the time when C&C-server IPs were issued 30 days.

It can been found that, as the new keyword length d increasing, the IQ of 4 SEs is also increasing. When using entire length of keyword, the IQ achieves the maxi-mum. Therefore, only using the whole MD5 value of date as search keyword can the best search result be presented.

Table 2. Selected direction and different length of new keyword impact on search effect

Search Engines	from beginning to end				from end to beginning			
	***d*=8**	***d*=16**	***d*=24**	***d*=32**	***d*=8**	***d*=16**	***d*=24**	***d*=32**
Google	0	0	0	6	0	0	0	6
Baidu	0	1	1	3	1	1	0	3
Bing	0	0	0	1	0	0	0	1
Haosou	1	3	4	3	2	4	1	3

In addition, changing selection direction of new keyword can't improve the search result. The right part of Table 2 presents test result for different length new keywords which are selected in reverse direction (i.e. from end to beginning). Obviously, if d is 8, 16 and 24, the total number of valid items obtained from Google, Baidu and Bing is 2, which is the same as in right part of Table 2. Meanwhile, the number of valid items obtained from Haosou is decreased. Therefore, compared with left part of Table 2, the keyword selected in reverse direction can't help improve search effect.

3.3 Different K Impact on Filtering Noise Item

Although the number of valid items increases as the time goes on, the number of noise items also increases. We use Top-*K* query algorithm to make valid items to appear in the first ranking *k* items as far as possible, so that the noise items can be filtered efficiently, which is convenient for bots to extract C&C-server IPs.

Fig.5 presents different *k* values impact on the accuracy achieved by using Top-*K* algorithm on the SERPs of 4 SEs ($w=(w_1, w_2, w_3)=(0.2, 0.4, 0.4)$). Here, accuracy means the percentage of valid items that appear in the top *k* items obtained from using Top-*K* algorithm on SERPs. We can find that the accuracy is rather sensitive to the variation of *k* values. For example, the range of *k* for Google varies in $1 \leqslant k \leqslant 6$, which for other SEs is $k \leqslant 3$. As a whole, the accuracy for 4 SEs shows a decreased tendency

along with increment of k , which means that the percentage of noise items in the selected top k items becomes higher. Therefore, different k values should be set for different SEs due to their difference in search ability and indexing mechanism.

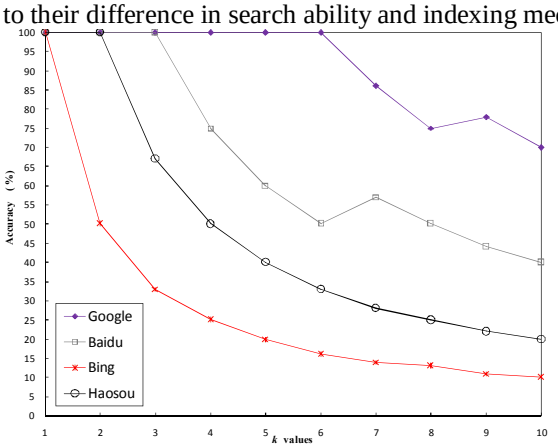

Fig. 5. Different K values impact on filtering noise items

4 Related Work

From the currently published literatures about this topic, the existing methods for finding C&C-server address can be grouped into four categories:

- Static IPs /domain name: The IPs/domain names of C&C-servers are hardcoded into malware beforehand. When the terminal is infected by this malware and become a bot, it directly communicates with C&C-servers represented by these IPs/domain names and joins into corresponding botnet. The typical malwares are Merga-D and Rustock[6]. The big disadvantage of this approach is that the hard-coded IPs or domain names in malware can be obtained by reverse engineering analysis. So that the corresponding C&C-servers are easy to be tracked and shut down.
- Domain-Flux: Bots use a special domain name generation (DGA) algorithm to produce a number of bogus domain names but some of which represent real C&C-servers. Then the bots attempt to send DNS query for each bogus domain name, try to find out those ones who receive successful DNS response, and communicate directly with them. The typical malwares are Conficker, Pushdo and Bobax[7]. Although Domain-Flux method is more invisible and robust , the DNS query packets still present obvious features which can provide a way to detect and block this method in local network[8-10].
- Fast-Flux: Some of bots who own public IP addresses have been disguised "proxy", other bots can communicate with C&C servers only via these proxy bots[11]. To enhance the flexibility and invisibility, the IP addresses of proxy bots are always changing. But there already exist some detection schemes against Fast-Flux method and achieve good effect[12][13].

- P2P-based method: P2P bots (e.g. Phatbot、Nugache[14]) utilize some inherent dynamic discovery mechanism of P2P protocol to find C&C-servers[15], such as Chord, Symphony, Kelips and so on. Once the bot of this type is identified, the C&C-servers may be exposed in its distributed hash table record[16]. Based on this point, researchers have already proposed some effective detection schemes for this C&C-servers finding process[17-23].

5 Conclusion

In this paper, inspired by item structure in search engine result pages, we provide a novel C&C-server IP addresses finding scheme based on search engine, named CAFSE, for solving the problem that how to find and obtain the C&C-server IP addresses for malware in APT and Botnet. CAFSE mainly consists of several modules as PM, KPM, SM, NIFM and ECM. The main advantage of this scheme is enhancing the stealth of acquiring the C&C-server addresses procedure for bots, due to using public search engine service and blog. The experimental results show that our proposed scheme can find and extract C&C-server IP addresses via various search engines accurately. The future research work will be launched in the following aspects: (1) how to increase the amount of valid item in SERPs. (2) for keeping better security, consider how to issue the C&C-server IP addresses in covert ways rather than issuing the plaintext of C&C-server IP addresses directly in abstract part of valid item. (3) further improve the Top-K algorithm to filter noise items as far as possible.

Acknowledgments This work is completed under the support of the Scientific Research Innovation Projects for General University Graduate of Jiangsu province (KYLX_0141); the Fundamental Research Funds for the Central Universities; the National High Technology Research and Development Program ("863"Program) of China (2015AA015603); Jiangsu Future Networks Innovation Institute: Prospective Research Project on Future Networks (BY2013095-5-03); Six talent peaks of high level Talents Project of Jiangsu province (2011-DZ024); Natural Science Foundation of Tibet Autonomous Region of China (2015ZR-13-17, 2015ZR-14-18).

References

1. Khattak S, Ramay N R, Khan K R, et al.: A taxonomy of botnet behavior, detection, and defense. IEEE Communications Surveys & Tutorials. 16(2), 898-924(2014)
2. Chen P, Desmet L, Huygens C.: A study on advanced persistent threats. In Proceedings of the 15th IFIP TC 6/TC 11 International Conference On Communications and Multimedia Security, Aveiro, Portugal, pp. 63-72.(2014)
3. Juels A, Ting Fang Y. : Sherlock Holmes and the case of the advanced persistent threat. In Proceedings of the 5th USENIX conference on Large-Scale Exploits and Emergent Threats. San Jose, CA, USA, pp. 63-72.(2012)
4. Rafael A R G, Gabriel M F, Pedro G T. : Survey and taxonomy of botnet research through life-cycle. ACM Computing Surveys. 45(4):1-33.(2013)

5. Zand A, Vigna G, Yan X, et al. : Extracting probable command and control signatures for detecting botnets. In Proceedings of the 29th Annual ACM Symposium on Applied Computing. ACM, pp. 1657-1662(2014)
6. Ken C , Levi L. : A case study of the rustock rootkit and spam bot. In Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets (2007)
7. Damballa. Top-5 Most Prevalent DGA-based Crimeware Families, https://www.damballa.com/downloads/r_pubs/WP_ DGAs-in-the-Hands-of-Cyber-Criminals.pdf.
8. Yadav S. , Reddy A.K.K. , Reddy A.L.N. et.al. : Detecting Algorithmically Generated Domain-Flux Attacks With DNS Traffic Analysis. IEEE/ACM Transactions on Networking. 20(5):1663-1677(2012)
9. Antonakakis M., Perdisci R., Nadji Y. et.al. : From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In Proceedings of the 21st USENIX Security Symposium (2012)
10. Bilge L., Kirda E., Kruegel C. et.al. : EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In Proceedings of the 2011 Symposium on Network and Distributed System Security (2011)
11. Riden J. Know your Enemy: Fast-Flux service networks, The Honeynet Project, http://www.honeynet.org/book/export/html/130.
12. Holz T, Gorecki C, Rieck K, Freiling FC. : Measuring and detecting fast-flux service networks. In Proceedings of the 15th Annual Network and Distributed System Security Symp. (2008)
13. Nazario J, Holz T. : As the net churns: Fast-Flux botnet observations. In Proceedings of the 3rd International Conference on Malicious and Unwanted Software, pp.24ī 31 (2008).
14. S.Stover, D.Dittrich, J.Hemandez, et.al. : Analysis of the Storm and Nugache Trojans:P2P is here. In proceedings of USENIX , pp. 8-27(2007)
15. Dittrich D. , Dietrich S. : P2P as botnet command and control: A deeper insight. In Proceedings of the 3rd International Conference on Malicious and Unwanted Software, pp.41-48. (2008.)
16. Thorsten H. , Moritz S., Frederic D. et.al. : Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm. In Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats,pp.1-9(2008)
17. Su Chang, Thomas E. Daniels. : P2P botnet detection using behavior clustering & statistical tests. In Proceedings of the 2nd ACM workshop on Security and Artificial Intelligence, pp. 23-30 (2009)
18. Zhang J.J., Perdisci R. , Lee W.K. et.al. : Detecting stealthy P2P botnets using statistical traffic fingerprints. In Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks , pp. 121-132, (2011)
19. David Zhao, Issa Traore, Ali Ghorbani, et.al. : Peer to Peer Botnet Detection Based on Flow Intervals. Information Security and Privacy Research. 376:87-102(2012)
20. Kamaldeep S., Sharath C.G.b, Abhishek T., et.al.: Big Data Analytics framework for Peer-to-Peer Botnet detection using Random Forests. Information Sciences(Online Press) (2014)
21. Zhao D, Traore I, Sayed B, et.al. : Botnet detection based on traffic behavior analysis and flow intervals. Computers & Security. 39: 2-16(2013)
22. Stevanovic M, Pedersen J M. : An efficient flow-based botnet detection using supervised machine learning. In Proceeding of the 2014 IEEE International Conference on Computing, Networking and Communications,pp. 797-801 (2014)

23. Garg S, Sarje A K, Peddoju S K.: Improved Detection of P2P Botnets through Network Behavior Analysis[M].Recent Trends in Computer Networks and Distributed Systems Security. Berlin :Springer Heidelberg, pp. 334-345(2014)
24. The MD5 Message-Digest Algorithm, https://tools.ietf.org /html/rfc1321.
25. Jung Hoon O, Seung Bong L, , Sang Jin L. : Advanced evidence collection and analysis of web browser activity. In Proceedings of 11th Annual Digital Forensics Research Conference New Orleans, LA, USA , pp. S62-S67(2011)
26. Jonathan Hedley. Jsoup HTML Parser, http://jsoup.org/.
27. He Z, Lo E.: Answering why-not questions on top-k queries. IEEE Transactions on Knowledge and Data Engineering. 26(6):300-1315(2014)
28. Timothy Jones. Tab-Snap, https://github. com/tdj28?tab=repositories.
29. Brewer R. : Advanced persistent threats: minimising the damage. Network Security. (4): 5-9(2014)