Progress in Command and Control Server Finding Schemes of Botnet

Xiaojun Guo[1,2,3], Guang Cheng[1,3+], Yifei Hu[1,3], Mian Dai[1,3]
[1]*School of Computer Science and Engineering*
Southeast University, Nanjing 210096, China
[2]*School of Information Engineering*
Xizang Minzu University, Xianyang 712082, China
[3]*Key Laboratory of Computer Network and Information Integration*
Southeast University, Nanjing 210096, China
+gcheng@njnet.edu.cn

***Abstract*—Botnets have become one of the most serious threats to current Internet and future network security. Only finding and connecting command and control(C&C) servers can bots join and work for botnet, hence how to find C&C servers is critical to botnet management and running. In this paper, we preliminarily summarize and classify the currently typical C&C server finding schemes as three types: dedicated IP address, Internet infrastructure and third-party application/service from a new perspective. And we compare these three types on four aspects. It's seen that third type presents better than other two types on complexity, flexibility, traffic covertness and scale.**

Keywords—Future network; Botnet; command and control; domain generation algorithm; third-party application/service

I. Introduction

Botnets have become one of the biggest threats to Internet security. A typical botnet is a highly controlled platform which consists of many compromised terminals (called *bots*) like smartphone, tablet, or personal computer etc. The botnet manager (called *botmaster*) can send commands to these bots through C&C servers to launch various of network attacks, such as Phishing fraud, E-mail bombing and Session Hijacking[1]. Also, botnet can cause serious threats for future network such as DDoS attack in SDN network[2], Interest flooding attacks in CCNx[3] and so on.

Once a victim terminal is compromised to become a bot (eg. a home PC with a broadband connection), it needs to join to the botnet that the botmaster is creating so that it may be controlled. So the first work for bot is to find C&C-server address information (eg. IP address, domain name, URL) in some way. After getting the C&C server address, the bot can build connection with them and register for further instructions. Then through C&C server, the botmaster also can communicate (issue commands and receive information) with these bots over a command and control channel.

Therefore, these C&C servers are the rendezvous points of bots and botmaster. Only when the bots find C&C-server address can they be controlled and managed by botmaster. Otherwise, these bots have no threat and practical value[4]. So how to find and get C&C-server addresses for bots is the first step to ensure the whole botnet to work correctly.

This survey mainly focuses on the existing methods for bots how to find their C&C Servers, because these methods play very important roles in botnets' working process and activities. Our contribution is classifying these methods into three categories based on underlying Internet services or applications. And each category has been described in details. To the best of our knowledge, this is the first survey that summarizes current types of C&C Severs finding schemes from a new perspective--services or applications. We also give a comparison of these three categories on complexity, flexibility, traffic covertness and scale, as shown in Table 1.

II. Classification

Each bot instance communicates with botmaster via C&C servers in botnet. Therefore, how to seek or find C&C servers is extremely critical for botnet to work normally. In addition, to avoid detection, most botmasters would like the ability to rapidly send instructions to bots without being detected so that the source of those commands could stay unrevealed. This impels botmaster to use all possible means as C&C servers finding schemes. So this section summarizes traditional and newly emerging types of C&C servers finding schemes, as shown in Fig.1.

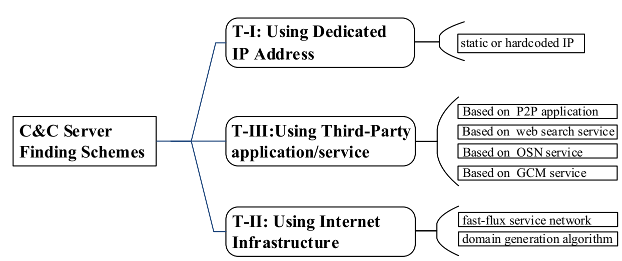

Fig. 1. The Classification of C&C Server Finding Schemes

A. T-I: Using Dedicated IP Address

In this type, the IPs 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

2324-9013/16 $31.00 © 2016 IEEE
DOI 10.1109/TrustCom.2016.262

C&C-servers represented by these IPs and joins into corresponding botnet. The typical malwares are Merga-D, Rustock[5] , ZeroAccess clickbot[6] etc. For example, the C&C server's IP addresses of ZeroAccess clickbot are encoded and stored in the resource section of its DLL file. These IP addresses can be decoded using a simple XOR algorithm with one specific key. Fig. 2 gives an example of one decoded C&C server IP. However, the big disadvantage of this approach is that the hardcoded 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. David *et al*.[7] evaluate how well different IP blacklists detect botnets through two metrics: completeness and responsiveness. They successfully apply to a set of IP blacklists in order to evaluate if these are able to detect Zeus-related infections. Anirudh *et al*.[8] utilize the persistence and distribution feature of each spamming IP address to form Behavioral Blacklisting, which is used to filter spam bot instead of IP-based blacklist.

```
009600c8   39783d74 7a6f6662 37767669 39727930   t=x9bfozivv70yr9
009600d8   776e766f 78307a72 32663979 306d6c64   ovnwrz0xy9f2dlm0
009600e8   38637537 39727877 6d393668 7826676f   7uc8wxr9h69mog&x
009600f8   2633323d 26303d79 72616573 623d6863   =23&y=0&search=b
00960108   682b726b 746f7079 6b656568 0000d200   kr+hypotheek....
00960118   74746800 2f2f3a70 322e3539 312e3131   .http://95.211.1
00960128   312e3339 633f2f36 3d64696c 31613868   93.16/?clid=h8a1
00960138   34327370 307a7168 00000000            ps24hqz0....
```

Fig. 2. A decoded C&C server IP of ZeroAccess clickbot

B. T-II: Using Internet Infrastructure

There are some core systems to guarantee normal working of Internet, named Internet Infrastructure, such as BGP, DNS, PKI, CDN etc. They can be often used to help bots find C&C servers, especially DNS. Because DNS is in charge of correspondence between domain name and IPs, many botnets utilize DNS to change C&C servers's domain names frequently in order to avoid detection. The typical methods are Fast-Flux Service Network (FFSN) and Domain generation algorithm (DGA).

FFSN is a distributed proxy network composed of compromised machines (i.e. flux-agents) which can direct incoming DNS requests to the botnet's desired address in use. These flux-agents own public IP addresses and have been disguised as "proxy", other bots can communicate with C&C servers only via these proxies[9]. There are two types of FFSN. One is Single-Flux in which flux-agents serve as " alleged DNS Servers" and send the often changed IPs as response to bots' C&C Server DNS request. Another is Double-Flux in which some flux-agents act as redirectors and only forward bots' C&C Server DNS request to other flux-agents that act as "alleged DNS Servers"[10]. Although FFSN technique has good flexibility and invisibility, there already existing some detection schemes against it and achieve good effect. For example, Holz et al.[11] design a linear decision function composed of the number of A record, NS record and unique ASN to detect fast-flux domains. Huang et al.[12] present spatial snapshot Fast-flux detection system. It maps all of IP addresses in a DNS response packet into geographic coordinate system. Then it uses spatial distribution estimation and spatial service relationship evaluation for identifying FFSN. In addition, some authors use other features or metrics to identifying the fast-flux domain name[13][14].

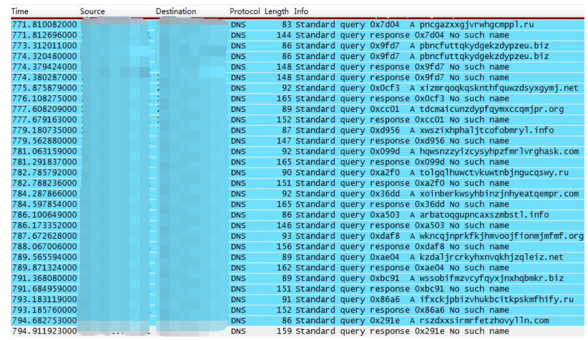

Fig. 3. Network traffic produced by domain-flux bot

```
suffix = ["anj", "ebf", "arm", "pra", "aym", "unj",
    "ulj", "uag", "esp", "kot", "onv", "edc"]

def generate_daily_domain():
    t = GetLocalTime()
    p = 8
    return generate_domain(t, p)

def scramble_date(t, p):
    return (((t.month ^ t.day) + t.day) * p) +
        t.day + t.year

def generate_domain(t, p):
    if t.year < 2007:
        t.year = 2007
    s = scramble_date(t, p)
    c1 = (((t.year >> 2) & 0x3fc0) + s) % 25 + 'a'
    c2 = (t.month + s) % 10 + 'a'
    c3 = ((t.year & 0xff) + s) % 25 + 'a'
    if t.day * 2 < '0' || t.day * 2 > '9':
        c4 = (t.day * 2) % 25 + 'a'
    else:
        c4 = t.day % 10 + '1'
    return c1 + 'h' + c2 + c3 + 'x' + c4 +
        suffix[t.month - 1]
```

Fig. 4. Torpig daily domain generation algorithm[16]

To evade network detection and mitigation techniques, bots can use an algorithm to periodically generate any number of domain names and contact a few of them every day, receiving updates and actions to be executed. This algorithm is called DGA. With the help of DGA, bots produce a number of bogus domain names(see Fig. 3) at one moment 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 DGA is more robust and not easy to eliminate, because its generated domains can change frequently based on time, such as current date, hour , even minute. The typical botnets using DGA algorithm are Conficker, Pushdo, Bobax[15] and Torpig[16]. Fig. 4 presents one typical DGA pseudocode used by Torpig bots[16]. Although DGA method is more invisible and secretly, the DNS query packets still present obvious features which can provide a way to detect and block this method in local network[17][18]. Antonakakis *et al.*[19] first extract the n-gram feature, entropy-based feature and structural domain feature of Non-Existent Domain produced by bots, and then use Hidden Markov Models to identify the domain names representing C&C servers. *Yadav et al*.[20] utilize the failures around successful

DNS queries and the entropy of the domains belonging to such queries, for detecting botnets with lower latency.

C. T-III: Using Third-Party application/service

As the Internet is spreading and communication is improving, many new applications/services have sprung up like mushrooms, such as P2P applications, web search engine (WSE) service and so on. Especially as mobile devices/smartphones become widespread, some new types applications/services, e.g. online social networks(OSN), cloud messaging, have been emerging and popular. All these applications/services provide new fields for C&C servers construction, which produces different C&C servers finding methods for bots.

The P2P applications, like PPlive, eMule, Skype and KuGoo, are very popular on Internet. These P2P applications can be used to construct P2P botnet(e.g. Phatbot, Nugache[21]) by attacker. In P2P botnet, bots are able to utilize some inherent dynamic discovery mechanism of P2P protocol to find C&C-servers[22], such as Chord, Symphony, Kelips and so on. Some researchers have already provided more detailed description of P2P botnets[23-25]. Once a P2P bot is identified, the C&C-servers may be exposed in its distributed hash table record[26]. Based on this point, researchers have already proposed some effective detection schemes for this C&C-servers finding process[27-29]. For the sake of brevity, here we will not restate related work in this article.

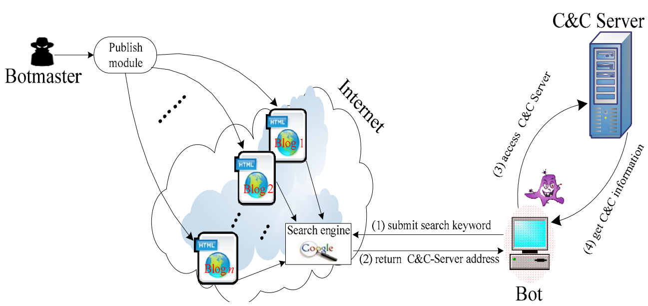

Fig. 5. The C&C mechanism of CAFSE.

WSE services (e.g. Bing, Google web search) have already been an integral part of user's daily Internet behaviors. People can use WSE easily and conveniently to seek online information they need[30]. There are enough data to prove that WSE has become more and more important basic service on Internet[31][32]. Unfortunately, WSE also can be used to find C&C Servers by bots. Guo *et al*.[33] show such a C&C-server addresses finding scheme, called CAFSE, based on WSE and experimented on some frequently-used search engine like Google, Baidu, Bing and Haosou, as shown in Fig. 5. In CAFSE, the botmaster uses publish module to issue C&C-server IPs in diaries of several free blogs on Internet whose title are MD5 values of date. These diaries can be indexed by search engine(SE). When the infected terminal becomes a bot, it uses keyword production module to produce the same MD5 values of date as the keywords and submits some or all keywords to search engines to obtain the search engine result pages(SERPs). Then, for items in SERPs, the bot uses Top-K algorithm to remove noise items and leave valid items whose abstract part contain C&C-server IPs. Lastly the bot exploits pattern matching method to extract these C&C-server IPs and translates them into binary format.

With the rapid development in mobile computing technology, mobile devices (e.g. smartphones, tablets) have evolved to offer sophisticated functionalities at lower costs. However, mobile devices become an important target of attackers for establishing mobile botnets because their computing capabilities become higher and they are consistently connected to the Internet via Wi-Fi or cellular networks. Cui[34]designed a mobile botnet called Andbot which exploited a novel command and control strategy named URL Flux, as shown in Fig. 6. The proposed Andbot had desirable features including being stealthy, resilient and low-cost (i.e., low battery power consumption, low traffic consumption and low money cost) which promised to be appealing for botmasters.

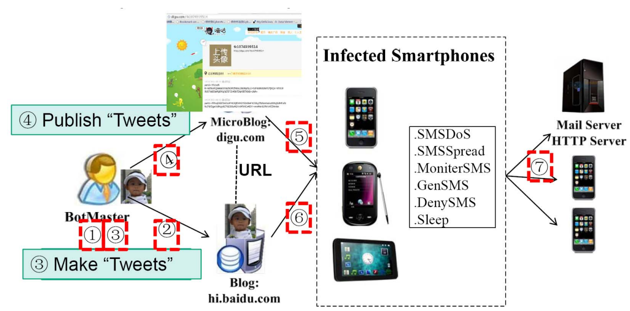

Fig. 6. The C&C Architecture of Andbot

Lee *et al*.[35] explored a new C&C channel for mobile botnets(see Fig. 7) that was based on the push notification service (PNS) of Android: Google Cloud Messaging for Android (GCM). They found two vulnerabilities: the registration process of the GCM only checked the validity of Gmail address and applications hid received push messages from users. They evaluated the feasibility of the push notification service-based mobile botnet (Punobot) in several aspects and showed that Punobot is stealthy, energy-efficient, and dangerous.

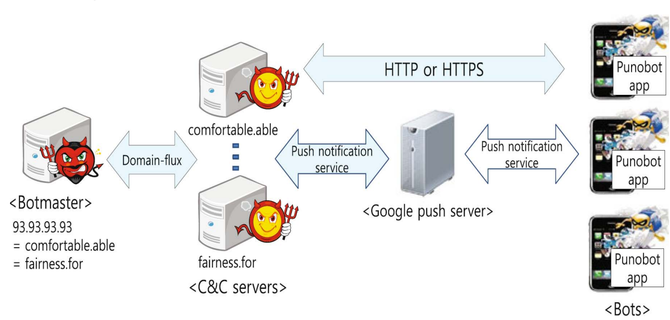

Fig. 7. Architecture of Punobot

OSN play a huge part in current Internet and people's lives. The sheer volume of social network traffic and the ability to easily host information within a social network page for little to no cost have made OSN a very attractive tool to botmaster. Bots can easily find and communicate with C&C servers through OSN. Shishir *et al.*[36] proposed a new generation botnet -Stegobot which was based on a model of covert communication over a social-network overlay. Stegobot used image steganography to hide the presence of C&C communication within image sharing behavior of user interaction. Both analysis and experiments indicated that Stegobot's network was not only stealthy but also functionally powerful in transferring sensitive data from its victims to the botmaster. Boshmaf *et al.* [37]designed and analyzed a socialbot based on Facebook(see Fig. 8). This botnet had one botmaster, 102 bots, and ran eight weeks. These bots can seek and connect C&C servers through OSN platform itself channel like in literature[36] or socialbot-OSN channel which carried only OSN-specific API calls and normal HTTP traffic. Furthermore, they concluded that socialbots could be profitable and could cause serious privacy breaches. Therefore, socially-aware software security could be at risk.

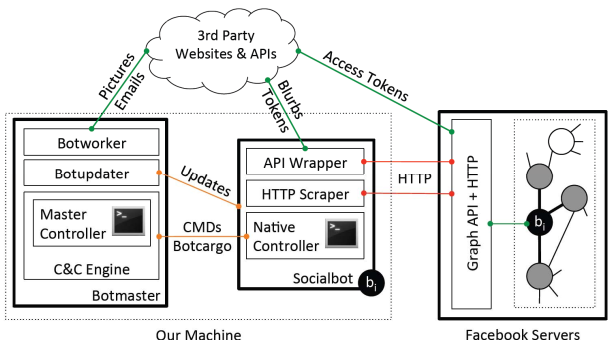

Fig. 8. The Facebook Socialbot Network

III. COMPARISON

2.4 Comparison

We compare three types C&C server finding schemes at four aspects: complexity, flexibility, traffic covertness and scale, as shown in Table1. Complexity means the difficulty for botmaster to design, implement or spread the C&C server finding scheme. Flexibility indicates how easily the C&C servers are built or changed. Traffic covertness means if the network traffic produced during the process of finding C&C servers presents obvious features compared to other application network traffic. Scale denotes the number of C&C servers used to connect bots.

From Table1, we can see that T-I and T-II is more complex than T-III. T-I and T-II will fully designed and implemented in their malicious code, while T-III only needs a small amount of code with the help of third-Party application/service. T-II and T-III is better than T-I in flexibility, because the C&C server IPs are hardcoded in bot code in T-I, which is not easily to be changed. Moreover, the network traffic produced by T-III during the process of finding C&C servers is difficult to identify because it's hidden in the traffic of third-Party application/service it relies on. And the number of C&C servers in T-III can be easily controlled according to the bot number in botnet built on third-Party application/service.

TABLE I. C&C SERVER FINDING SCHEMES COMPARISON

Types	Complexity	Flexibility	Traffic Covertness	Scale
T-I	★★★★☆	★★☆☆☆	★★☆☆☆	★☆☆☆☆
T-II	★★★★★	★★★★★	★☆☆☆☆	★★★☆☆
T-III	★★★☆☆	★★★★★	★★★★★	★★★★★

IV. CONCLUSION

C&C server finding scheme is very important for botnet. This paper mainly surveyed the typical schemes into three types and compare them in four aspects. With the new kinds of Internet or future network application appearing, we believe that more and more new types of C&C server finding schemes will arise or be exposed. For the future work , we will continuously focus on this topic and summarize the new schemes at proper time.

ACKNOWLEDGMENT

This work was supported by Natural Science Foundation of Tibet Autonomous Region of China(2015ZR-13-17), Scientific Research Innovation Projects for General University Graduate of Jiangsu province(KYLX_0141), Fundamental Research Funds for the Central Universities, National High Technology Research and Development Program("863"Program) of China (2015AA015603), Six talent peaks of high level Talents Project of Jiangsu province (2011-DZ024).

REFERENCES

[1] S. Khattak, N. R. Ramay, K. R. Khan, et al. A taxonomy of botnet behavior, detection, and defense. IEEE Commun. Surveys & Tutorials, 16(2): 898-924, 2014.

[2] S. Lim, J. Ha, H. Kim, et al. A SDN-oriented DDoS blocking scheme for botnet-based attacks. In Proceedings of 6th International Conference on Ubiquitous and Future Networks, 63-68, New Jersey, NJ, 2014, IEEE.

[3] M. Virgilio, G. Marchetto and R. Sisto.Interest Flooding Attack Countermeasures Assessment on Content Centric Networking. In Proceedings of 12th International Conference on Information Technology-New Generations, 721-724, New Jersey, NJ, 2015, IEEE.

[4] A. Zand , G. Vigna, X. Yan, et al. Extracting probable command and control signatures for detecting botnets. In Proceedings of the 29th Annual ACM Symposium on Applied Computing,1657-1662, New York, NY, 2014, ACM.

[5] C. Ken and L. Levi. A case study of the rustock rootkit and spam bot. In Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets, 2007, USENIX.

[6] L. Wayne . A deeper look into the ZeroAccess clickbot. https://www.virusbulletin.com/virusbulletin/2013/04/deeper-look-zeroaccess-clickbot , April 2013.

[7] O. David, L. Jesus, F. Toni, et al. Benchmarking IP blacklists for financial botnet detection. In Proceedings of 6th International Conference on Information Assurance and Security, 62-67, New Jersey, NJ, 2010, IEEE.

[8] R. Anirudh, F. Nick and V. Santosh. Filtering spam with behavioral blacklisting. In Proceedings of the 14th ACM Conference on Computer and Communications Security, 342-351, New York, NY, 2007, ACM.

[9] S. William and D. Robert . Know your Enemy: Fast-Flux service networks. The Honeynet Project. http://www.honeynet.org/book/export/html/130. July 2007.

[10] A. Caglayan, M. Toothaker, D. Drapeau, et al. Real-time detection of fast flux service networks. In Proceedings of the IEEE Cybersecurity Applications & Technology Conference for Homeland Security, 285–292, Washington, DC, USA, 2009 , IEEE.

[11] T. Holz, C. Gorecki, K. Rieck, et al.. Measuring and detecting fast-flux service networks. In Proceedings of the 16th Annual Network and Distributed System Security Symposium , San Diego, CA, 2008, Internet Society.

[12] S. Y. Huang, C. H. Mao and H. M. Lee. Fast-flux service network detection based on spatial snapshot mechanism for delay-free detection. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, 101-111, New York, NY, 2010 ACM.

[13] E. Passerini, R. Paleari , L. Martignoni , et al. Fluxor: Detecting and monitoring fast-flux service networks. Lecture Notes in Computer Science. 5137: 186-206, Berlin Heidelberg, 2008, Springer.

[14] C. H. Hsu, C. Y. Huang, K. T. Chen. Fast-flux bot detection in real time. In Proceedings of 2010 conference on Recent Advances in Intrusion Detection, 464-483, Berlin Heidelberg, 2010, Springer.

[15] Damballa. DGAs in the Hands of Cyber-Criminals, https://www.damballa.com/downloads/r_pubs/WP_DGAs-in-the-Hands-of-Cyber-Criminals.pdf , February 2014

[16] B. Stone-Gross, C. Marco, C. Lorenzo, et al. Your botnet is my botnet: analysis of a botnet takeover. In Proceedings of the 16th ACM Conference on Computer and Communications Security, pp.635-647, New York, NY, 2009, ACM.

[17] S. Yadav, A.K.K. Reddy, A.L.N. Reddy, et al. Detecting Algorithmically Generated Domain-Flux Attacks With DNS Traffic Analysis. IEEE/ACM Trans. on Networking, 20(5):1663-1677, 2012.

[18] L. Bilge, E. Kirda, C. Kruegel, et.al. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In Proceedings of the 2011 Symposium on Network and Distributed System Security, San Diego, CA, 2011, Internet Society..

[19] M. Antonakakis, R. Perdisci, Y. Nadji, et.al. From throw-away traffic to bots: detecting the rise of DGA-based malware. In Proceedings of the 21st USENIX Security Symposium, 24-24, USENIX, 2012 .

[20] S. Yadav and A. L. N. Reddy. Winning with DNS failures: strategies for faster botnet detection. In Proceedings of 7th International ICST Conference on Security and Privacy in Communication Networks, 446-459, Berlin Heidelberg, 2012, Springer.

[21] S. Stover, D. Dittrich, J. Hemandez, et.al. Analysis of the Storm and Nugache Trojans:P2P is here. Magazine of USENIX & SAGE , 32(6): 18-27, 2007.

[22] D. Dittrich and S. Dietrich. P2P as botnet command and control: A deeper insight. In Proceedings of the 3rd International Conference on Malicious and Unwanted Software, 41-48, New Jersey, NJ, 2008, IEEE.

[23] A. H. Lashkari, S. G. Ghalebandi, and M. R. Moradhaseli. A Wide Survey on Botnet. In Proceedings of the International Conference on Digital Information and Communication Technology and Applications , 445-454, Berlin Heidelberg, 2011, Springer.

[24] G. Starnberger, C. Kruegel and E. Kirda. Overbot: a botnet protocol based on Kademlia. In Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, New York, NY, 2008, ACM.

[25] A. Dennis, R. Christian and B. Herbert. Reliable Recon in Adversarial Peer-to-Peer Botnets. In Proceedings of the 2015 ACM Conference on Internet Measurement. 129-140, New York, NY, 2015, ACM.

[26] H. Thorsten , S. Moritz, D. Frederic, 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, 1-9, 2008, USENIX.

[27] S. Kamaldeep, C.G. Sharath, T. Abhishek, et.al. Big Data Analytics framework for Peer-to-Peer Botnet detection using Random Forests. Information Sciences(Online Press), 2014.

[28] D. Zhao, I. Traore, B. Sayed, et.al. Botnet detection based on traffic behavior analysis and flow intervals. Computers & Security, 39: 2-16, 2013.

[29] M. Stevanovic and J. M. Pedersen . An efficient flow-based botnet detection using supervised machine learning. In Proceeding of the 2014 IEEE International Conference on Computing, Networking and Communications, 797-801, New Jersey, NJ, 2014, IEEE.

[30] A. Hannak, P. Sapiezynski, K. A. Molavi, et al. Measuring personalization of web search. In Proceedings of the 22nd international conference on World Wide Web, 527-538, New York, NY, 2013, ACM.

[31] Statistic Brain Research Institute. Google Annual Search Statistics. http://www.statisticbrain.com/google-searches/, June 2015.

[32] CNNIC. The 36th Statistical Report of China Internet Developing Status. https://www.cnnic.cn/hlwfzyj/hlwxzbg/hlwtjbg/201507/P020150723549500667087.pdf , July 2015.

[33] X. J. Guo , G. Cheng, W. B. Pan, et al. A Novel Search Engine-Based Method for Discovering Command and Control Server. In Proceedings of 15th International Conference on Algorithms and Architectures for Parallel Processing, 311-322, New Jersey, NJ, 2015, IEEE.

[34] X. Cui , B.X. Fang , L.H. Yin, et al. Andbot: towards advanced mobile botnets. In Proceedings of the 4th USENIX conference on Large-cale Exploits and Emergent Threats, California, CA, 2011, USENIX.

[35] L. Hayoung, K. Taeho, L. Sangho , et al. Punobot: Mobile Botnet Using Push Notification Service in Android. In Proceedings of 14th International Workshop on Information Security Aplications, 124-137, Berlin Heidelberg, 2013, Springer.

[36] N. Shishir, H. Amir, P. Pratch, et al. Stegobot: a covert social network botnet. In Proceedings of 13th International Conference on Information Hiding, 299-313, Berlin Heidelberg, 2011, Springer.

[37] Y. Boshmaf, I. Muslukhov, K. Beznosov, et al. Design and analysis of a social botnet. Computer Networks, 57(2): 556–578, 2013.