Hiding Fast Flux Botnet in Plain Email Sight

Zhi Wang, Meilin Qin, Mengqi Chen, and Chunfu Jia(✉)

Nankai University, Tianjin, China
cfjia@nankai.edu.cn

Abstract. Fast flux and domain flux are widely used as evading techniques to conceal botnet C&C server. But nowadays, more and more machine learning schemes are introduced to recognize and detect fluxing botnet automatically and effectively. In this paper, we propose a novel fluxing scheme to hide C&C server in plain email sight. Email flux tries to blend in with normal email communication. With the excellent reputation of email servers, the malicious activity is more likely to get lost in the normal email crowd. Therefore, DNS-based botnet detection schemes are difficult to detect the email flux botnet. Comparing to the cost of registering a public IP address or a domain, the cost of registering an email account is much less, and email account reveals less geolocation information. And we introduce asymmetric encryption strategy to fortify DGA, preventing adversaries from taking down the botnet by registering email account before bot master. We also discuss possible countermeasures in the future to mitigate email flux.

Keywords: Fast flux · Domain flux · Botnet
Command and control channel · Evasion technique

1 Introduction

Botnet is a network of compromised computers, known as bots or zombies, that could be remotely controlled by an attacker in the Internet, so-called botmaster. Currently, botnets are the main platform for attackers to carry out large scale cyber crimes, such as sending spams, phishing, launching distributed denial of service (DDoS) attacks. According to Symantec report, in 2016 there are 98.6 million hosts controlled in botnets, which is an increase of 6.7 million over last year [1]. There were at least 255,065 unique phishing attacks worldwide, which represents an increase of over 10% from the 230,280 attacks identified in 2015 [2]. The number of DDoS attacks per day ranged from 131 to 904 in the second quarter in 2017 [3]. Hence, botnet is one of the most significant threats to the Internet.

To build a complete botnet, a stealthy command and control (C&C) channel must be built between the botmaster and bots, through which the botmaster can send commands to all bots. Due to host-based detection such as reverse

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2018
X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 182–197, 2018.
https://doi.org/10.1007/978-3-319-78816-6_14

engineering is hard, most defenders focus on detecting the C&C channel, trying to cut off the communication and shut down the botnet. Therefore, the botmaster will make every effort to conceal the C&C channel to decrease the risk of detection. For example, [4,5] exploit social network to construct botnet, [6] uses email protocol as C&C channel and [7] hides the commands in SMS message.

To make the C&C channels more stealthy, there are many evading techniques such as fast flux and domain flux. With fast flux, the bots would query a certain domain that is mapped onto a set of IP addresses that change frequently [8]. However, fast flux uses only one single domain name, which will lead to a single point of failure. In domain flux, the botmaster associates one or more IP with several domains to avoid being easily blocked by blacklisting.

Although the fast flux and domain flux techniques can hide botnet C&C server behind a set of IP addresses or randomly generated domain names, the defenders can also identify the botnets through DNS traffic analysis. The fast flux and domain flux rely on DNS service, and there are some significant difference between fluxing botnet DNS traffic and normal DNS traffic, such as the number of different IP addresses resolved from the same domain, the length of each packet and so on. Many machine learning models are trained to recognize suspicious fluxing DNS communication automatically. Email is not a kind of IP-based C&C delivery, thus the email sent by bot will have similar features in DNS traffic with normal users. We propose an email flux method that can bypass the existing machine learning detection techniques against fast flux and domain flux.

We summarize the contributions of this paper as follows:

- We present the email flux, which is a different from fast flux and domain flux. It applies randomly generated email addresses to establish the fluxing C&C channel so that email flux could evade traditional machine learning methods using DNS traffic analysis.
- We enhance the traditional domain generation algorithm(DGA) used in domain flux, preventing adversaries from controlling the automatically generated email accounts in advance.
- We analyze the traffic, cost and reputation of email flux and prove its availability and concealment.
- We discuss the possible countermeasures in the future for mitigating email flux and other possible new fluxing channels.

2 Related Works

Botnet is a network of compromised computers, known as bots or zombies, that could be instructed by a controller in the Internet, so-called botmaster. Botnets can be used to perform DDoS attack, steal data and send spam. In the face of potential attacks, the Intrusion Detection System (IDS) is an important defensive mechanism to defend against these possible attacks.

The common types of IDS techniques include: signature based detection, anomaly detection, artificial neural network (ANN) based IDS and fuzzy logic based IDS [9]. The signature based detection can detect the malicious traffic by using a set of rules and known signature attack stored in a knowledge database [10]. However, the disadvantage is that it could not detect unknown new attacks. The anomaly detection could detect abnormal system behavior and malicious traffic, which needs to be specified a baseline by the security researcher. The ANN based IDS detection utilizes ANN as a pattern recognition technique. The fuzzy logic based on rule can detect the intrusion behavior of the traffic [9].

The core of the botnet is its C&C, many attackers use fast flux and domain flux methods to hide their C&C channels [11,12]. By exploiting fast flux technique, the botmaster hides the real IP addresses that belong to his C&C servers. Each bot can use the same domain name to connect with C&C servers, while the IP addresses resolved are constantly changing.

There are many approaches to detect fast flux such as active or passive DNS traffic monitoring. [13] uses a combination of passive DNS monitoring and active DNS probing to detect botnets, which based on a cluster analysis of the features obtained from the payload of DNS-messages and uses active probing analysis. [14] is based on large-scale passive analysis of DNS traffic generated by hundreds of local recursive DNS (RDNS) servers located in different networks and scattered across several different geographical locations, to detect and track malicious flux networks. They clarify four characteristics of flux domain names: (1) short TTL; (2) high frequency of change of the set of resolved IPs returned at each query; (3) large overall set of resolved IPs acquired by querying the same domain; (4) the resolved IPs are scattered across many different networks. Then they utilize these features to filter flux domains. Even though the fast flux seems to be a fine evading technique, it has a single-point-of-failure problem. If a security researcher discovers a botnet's domain name, he will blacklist this domain name and block the communication of botnet.

To avoid this issue, attackers utilize domain flux method to hide C&C servers of botnet such as Tropig [8] and Conficker [15]. By using domain flux technique, the botmasters can frequently change domain names mapped to a single IP address. Each bot generate a list of domain names by running the same DGA and then tries to connect to the domain names in the list until the success of finding C&C servers. Generally, the inputs (or seeds) of the DGA are the current date information and some numeric parameters. Unlike fast flux, domain flux is more resilient to avoiding take-down attempts. More specifically, even if the current domain is blocked, the botmaster only need to register the next domain to control his botnet again.

Because the domain names change frequently, blacklisting domain name is not effective. In order to detect domain flux, many approaches have been proposed. [16] uses DNS query data and analyzes the network and zone features of domains to build a dynamic reputation system. [17] monitors DNS traffic and presents 15 behavioral features used in the identification of malicious domains. [18] detects malware-related domains based on DNS resolution patterns by monitoring DNS traffic from the upper DNS hierarchy. [19] preserves the privacy of the users of the network and only uses the DNS replies to detect domain flux.

[20] proposes a combination of clustering and classification algorithm that relies on the similarity in characteristic distribution of domain names to remove noise and group similar domains. However, it can only detect centralized botnet, not P2P botnet.

The biggest difference between email flux and traditional flux is that email flux uses email as C&C channel. In order to make the botnet hard to be shut down, the fast flux and domain flux botnets sacrifice their concealment for robustness. Whether it is fast flux or domain flux will generates anomaly DNS traffic, and the existing detection methods for fast flux and domain flux can be simply summarized by monitoring DNS traffic. In contrast, email flux do not rely on IP-based C&C delivery. Although the emails sent by bot have a few differences to normal email communication, it is almost the same from the point of DNS monitoring. Thus the existing DNS-based detection method cannot catch the email flux.

There are literatures that select other channels for C&C communication. [21] utilizes the URL shortening service. The botmaster hides the IP address of C&C server into URLs, and change URL into automatically generated alias. However, it is still an IP-based C&C delivery. Visiting websites that do not exist will result in Name Error DNS responses, which is suspicious and has been the target for many detection methods like [22]. Besides, the algorithm that generates alias is similar to DGA, so that the whole botnet may be taken over by the defenders through pre-calculating and registering the alias [8]. We imporved the DGA and use push mechanism to send commands directly to each bot in case of being controlled by the defenders.

[7] selects SMS as C&C channel. Due to SMS message has to be sent by one phone number which is in use by the owner of the compromised phone, it is easy to be aware of. Email account does not combined with the compromised host, and the botmaster can register and allocate one email account to each bot. [6] firstly presents the feasibility of email-based botnet, we have made some development on its basis. [6] does not present a complete botnet, it just proves the feasibility that one bot can execute the command embed in email. It demonstrates the difficulty for the defenders to crack the encoded commands from the point of view of cryptography, however, the defenders can block the suspicious email account without knowing the content. We propose a more specific and practical email-based botnet, and introduce flux method to improve the robustness and resilience of botnet.

3 Designing of Email Flux

We propose an email flux method to hide C&C channels. Our design derives from traditional domain flux, but there are numerous differences between email flux and traditional flux method. First of all, the C&C channel is different. Domain flux is IP-based, thus a bot will generate anomaly DNS traffic, while email flux is similar to normal email communication. Second, in traditional domain flux, bots request commands from the generated domains, which is called pull mode.

In our work, the commands are sent directly from botmaster to bot email account, which is called push mode. Third, the email that embedded with commands is stored in email server. Standing in the DNS perspective, the communication is set between the compromised host and an email server, whose reputation is much higher than the domain used in domain flux.

The process of a C&C communication is simple but efficient. The botmaster embeds commands in emails with the asymmetric encryption algorithm and sends the private key to bots directly. A bot extracts the command by its decryption key and responds to it. The response will be sent to email accounts which are automatically generated and physically controlled by botmaster. Due to the bots just send their response to those email accounts rather than get commands from here, the botnet is impossible to be controlled by the defenders.

The email flux botnet uses a email addresses generation algorithm that generates a set of random email addresses composed of alphabet letters and digits. The inputs of the algorithm will be the current date information and a customized string. That is to say, there are 2 parameters which can determine the output email address. The date will be automatically changed while the string is determined by botmaster. The botmaster can change the generated email address list at any time through sending a new customized string to bots. For each round, such as a day, week, or the month, the email flux botnet generates k (e.g., 1000) different email addresses through the algorithm.

The botmaster and each bot have to share an email addresses generation algorithm, therefore, the botmaster and each bot will independently generate the same lists of email addresses periodically. The botmaster can also ask bots to change the customized string in order to get a new list of addresses. The botmaster periodically registers certain email addresses in the list in advance. Then, the bot contacts email addresses in the list in order until one succeeds–the botmaster will reply the response to show its validation.

Email flux can be simply classified into two stages: registering the email addresses to email providers and connecting to these email addresses via email. Essentially, email flux refers to periodically changing and registering the email addresses to bypass detection and blacklisting. There are many high reputation email providers we can choose, such as gmail, outlook, yahoo, and so on. Hence, in order to improve the resilience of email flux against take-down attempts, we can frequently change the email service provider.

3.1 Registration Stage

The botmaster has to register the email accounts at first and then check the response sent by bots. The botmaster needs to execute an email addresses generation algorithm to obtain k email addresses. The input of the algorithm will be the current date and a customized string. After acquiring a list of email addresses, the botmaster will select several addresses on the top of the list and register them. In general, at the beginning of each day's communication, the botmaster can only receive message in the first email account on the list.

However, as Fig. 1 shows, this address may be blocked by defenders because of a large number of suspicious email communication.

Here, we take registering outlook mailbox as an example to explain the registration process in detail. Step one, log in to www.outlook.com. Step two, fill in contact information and user information. Step three, enter an email address generated by the algorithm. If this email address has been registered, the system will prompt you to re-enter a new one. Step four, input validation information, to confirm that the account created is a real person. Other mailboxes may need to be verified by SMS.

Finally, the botmaster will receive a notification from the mail service provider if the registration is successful. Then, newly registered email address is available for email flux.

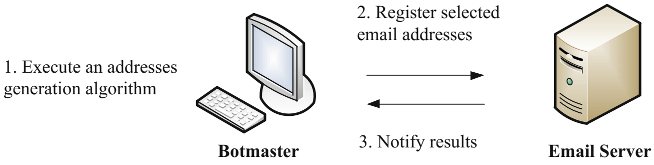
1. Execute an addresses generation algorithm
2. Register selected email addresses
3. Notify results
Botmaster
Email Server

Fig. 1. Email flux registration

3.2 Connection Stage

Each bot will periodically try to connect with email addresses to send message, as shown in Fig. 2. First, each bot independently executes an email addresses generation algorithm to get k email addresses. The input to the algorithm is also the current date a customized string. That is to say, the inputs of bot and botmaster are exactly the same, to make sure the lists of email addresses are the same as what botmaster generates.

Next, the bot attempts to send message to the email addresses in the list in order until one succeeds. These email accounts play the role of C&C servers because each bot will contact them. After sending its response, the bot will receive two kinds of response: a confirmed message and undelivered message. When the bot receives a confirmed message, it indicates that bot has successfully connected to the C&C server. Due to the botmaster physically controls the C&C servers, he can send email to bots to indicate he has received its response. The confirmed messages are also be encrypted.

When the bot receive a undelivered message, it means the bot attempted to contact an email address that had not yet been registered by the botmaster or had been blocked by the email provider. In this case, the bot needs to connect to the next email address in the list. If these email addresses all failed, bot will contact the email addresses hard-coded in its configuration file.

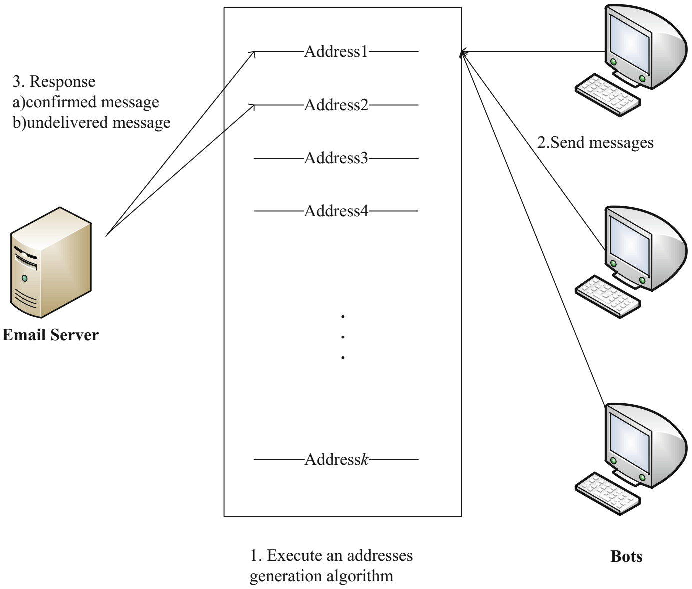

Fig. 2. Email flux connection

3.3 Improvement of Algorithm

There is a weakness in traditional domain flux that use DGA to generate domain names. Once the defenders know the DGA algorithm through reverse engineering, they can forecast and pre-register the next round of domain lists. [8] shows how the defenders take control of a botnet through forecast and pre-register the automatically generated domains. In order to make sure the communication can be carried out successfully, in our work, the botmaster should register a number of email accounts in advance and then pass the customized string he uses to bots. Since the customized string is one of the parameters that determine the generated email addresses, in this case, even though the defenders caputre the bot and know the email address generation algorithm, they still cannot preempt the email address on the top of the list. Those email accounts pre-registered by botmaster should belong to different email operators so that it is impossible to block them all in a short time. That gives the botmaster enough time to judge whether the defenders know the email address list and deal with it.

4 Analysis

In this section, we use quantified data to analyze the feature, or advantage, of email flux in detail. We set each list generated by bot with k=1000 email addresses. The botmaster registers top 5 email addresses in the list. The email service providers in the experiment we select are Gmail, Outlook, Sina, Foxmail and 163 email. First, we describe the traffic of email flux, and then we discuss the reputation of email. Finally, we evaluate the costs of botmaster managing a email flux botnet.

4.1 Traffic

In order to combat spam, email providers limit the amount of mails that each user can send. These limits restrict the number of messages sent per day and the number of recipients per message. After a user reaching the limits, he can't send new messages for up to 24 hours. However, they can still receive incoming email. As shown in Table 1, there are some major limitations set by popular email providers.

Table 1. Daily sending limit

Email provider	Message sent per day
Gmail	500
Outlook	100
163	50
Foxmail	50
Sina	30

From the Table 1, we draw a conclusion that the number of messages sent by each bot per day should not exceed the minimum 30. Thus, we set the number of messages to n=20 in our experiment. We assume the botnet consists of 5,000 bots, thus the total volume of emails per day is T= 100,000.

As shown in Fig. 3, we get the total number of global emails in June 2017 from https://talosintelligence.com. There are totally 59,209 available email server in the world. Thus, each email server will receive an average of more than 6 million emails every day. Supposing that all the bots use the same email server, the percentage of malicious email is only about 1.6%. If the defender use traditional detection method to locate bot members, they can only find the email server as the C&C server of the botnet because each bot member sends message to the mail server. If the defender blocks the detected email server, 98.4% innocent user will be implicated. However, the bot will use different email service providers and servers for communication. The effect of shutting down the email server will be even worse. Thus, email flux is feasible, and traditional detection and blocking method is useless for our email flux botnet.

4.2 Reputation

Reputation is an important factor in botnet detection. There are certain detection methods based on the reputation of domain [16,23,24]. Each domain has a

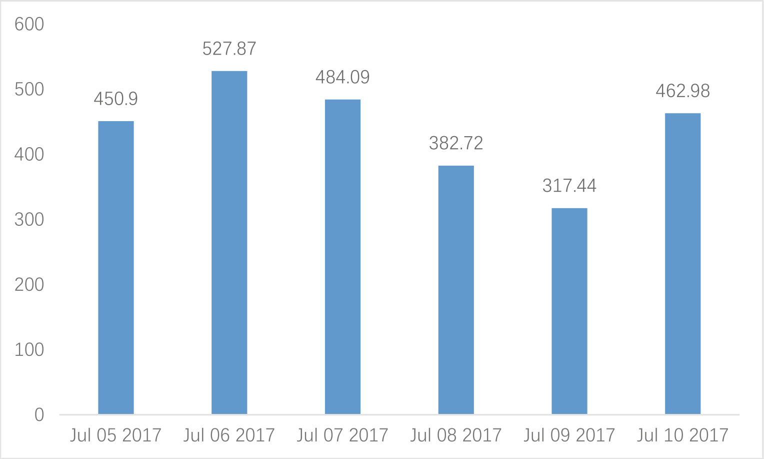

Fig. 3. Total number of global emails (billions)

reputation score upon registration. We select email as C&C channel, the email account itself does not have reputation based on the traditional reputation computing algorithm. However, the email provider does. Table 2 is the global rank of major email providers we get from Alexa. It shows that the reputation of the email providers that we use in the email flux botnet is extremely high. Thus they are not easy to cause suspicion, which indicates the superiority of email flux compared with other domain flux.

Table 2. Alexa rank of email providers

Site	Global rank	Rank in country (CN)
163.com	375	64
Sina.com	4,568	375
Outlook.com	5,014	9,088
Gmail.com	11,564	15,985
Foxmail.com	32,936	3,089

4.3 Costs

To compare with the costs of domain flux, we collect the price of registering a domain as follows:

As shown in Table 3, the money spend on registering domains are very high. Besides, the process of registering a new domain is cumbersome. The registrant has to fill in their personal information such as real name, phone number and identify card number. After submission, the relevant department will take a phone call for verification. Thus, domain registration not only cost a lot but also hard to fake. Traditional domain flux needs numerous of new domains every day, which may bring a heavy burden for botmaster.

On the contrary, the email registration is far more convenient. First of all, there is no fee for email registration. Then, the email account does not bind with personal information except for phone number, and there is no verification from email providers. Due to it is easy to register an email address, the price of buying a email account is low. Thus, the botmaster can acquire email accounts by manual registration on his own, automatically registration through certain program or buying online.

5 Potential Countermeasures

In this paper, we classify detection of email flux botnets into three types: hosts, DNS traffic and email providers. At hosts, security researchers attempt to detect and analyze malware by monitoring system statues. However, malware can use complex and advanced technology to conceal itself and increase the difficulties of analysis. Network monitors usually monitor DNS traffic to detect botnet. Since email services are very popular and have heavy usage volume, it is unlikely to be noticed. Besides, all email flux traffic is encrypted by email service providers automatically, and we also enhance the encryption to make it difficult for defenders to investigate it.

5.1 Detection in Hosts

If security researchers can detect malware for botnets on hosts, then they will know email addresses generation algorithms or addresses lists through a series of analyzing. The security researchers may distribute these email addresses to email provider for blacklisting. However, as email addresses are just a fraction of large set of email accounts used for actual communicating, blacklisting techniques is

Table 3. The cost of registering domains.

Types	Captions	First year (CNY)	Renewal (CNY)
.com	Global registration volume first	60	78
.net	The most popular domain name	65	78
.cn	The most popular for Chinese people	35	68
.top	To show one's personality	9	34
.cc	Competitive domain name	38	60
.org	Trusted domain name	70	78
.shop	For e-commerce	49	188
.me	For personal use	28	160

ineffective in countering such email fluxing. Reverse engineering of bot executables is a time-consuming process, and during this time the botmaster may send commands to the bot to change the algorithm. Also, there are many evasion techniques make reverse engineering difficult to be implemented. For example, malware authors can utilize emulation technology to obfuscate malware [25]. They also can use code protection tools to protect malicious code.

5.2 Detection in DNS Traffic

Analyzing IP Addresses. Fast flux detection schemes typically analyze the IP address diversity via monitoring DNS traffic. [26] analyzes traffic characteristics and introduces dynamic whitelisting to differentiate between FFSN and CDN. [27] develops a automated identification of fast flux domains by IP address diversity and flux-agent. Email flux is different from fast flux and do not need to frequently change IP addresses. Therefore, email flux can not be detected by analyzing IP address diversity.

Analyzing Group Activities. There are some methods that focus on group activities for DNS requests. [28,29] based on features extracted from groups of domains, which has to consider a problem of how to group these domains. The authors chose *random* groups of domains to overcome this problem. But there is no rigorous way to test and verify the validity of these hypotheses. [30] also considered the history of suspicious domain group activities, at the same time, they still analyzed suspicious failures in DNS traffic. In email flux, the bot do not generate a large of domains. Therefore, the above detection methods based on DNS group activities are invalid for our botnet.

Analyzing Failures Resolutions. Many domains generated by DGA need to be resolved via DNS, but the botmaster usually pre-registers only a small part of domains. Thus it will result in failure resolutions traffic by queries of bots. [22] presented a technique to efficiently analyze streams of unsuccessful domain name resolutions to automatically identify DGA-based botnet by using a combination of clustering and classification algorithms. Such failures domain resolutions also called Non-Existent Domain (NXDomains). [31] utilized the failures around successful DNS queries and the entropy of the domains belonging to such queries, for detecting the botnet. [32] also proposed a light-weight anomaly detection approach based on failed DNS queries, with a novel tool DNS failure graphs. The graphs captures the interactions between hosts and failed domain names. One of methods in [33] is identifying randomly name failed DNS requests. These detection approaches mainly analyze failures resolutions, which is not applicable in our case. Since email flux generates the email addresses instead of domain names, they can not detect it.

Analyzing Individual Domains. [34] presented a DGA classifier to classify individual domains. They used two basic linguistic features named meaningful characters ratio and n-gram normality score to tell DGA and non-DGA-generated domains. [35] also focus on detecting domains on a per-domain basis. They leveraged a random forest classifier to classify single domains. Similarly, because email flux do not generate domain names, such methods also can not detect it.

5.3 Detection in Email Providers

Limiting Registrations. In email service, there must be two email accounts in one communication process. The IP addresses of each host used to access domain is automatically assigned when connecting to the Internet. However, the botmaster have to register an email account first and then use it. Although we have improved the traditional DGA algorithm that do not need numerous new email accounts every day, the botnet still needs a large number of email accounts according to its size. Such a large scale is almost impossible for manual registration, thus the defenders only need to prevent automatic registration.

Nowadays, registering an email account only need to identify a common letter-based CAPTCHA. Obviously, they are insufficient because there have been many ways to crack it [36–38]. There are many improved forms of CAPTCHA and the email providers can update it.

Besides, if an email address is comprise of a series of random letters and is not 'human-pronounceable', it is probably automatically registered for malicious intent. The email provider can set more limitation for them. For example, require them to fill out their detailed personal information.

Restricting Newly Registered Accounts. The email flux botnet requires the botmaster to register several newly generated email addresses and put them into use every day. Generally speaking, the newly registered email account do not receive a lot of emails because few people know its address. Thus, the email providers can increase the security level of newly registered accounts, for example, filter out all the emails whose contents or addresses are comprise of a series of random letters and are not 'human-pronounceable'.

Some email providers, such as QQ, prohibit the newly registered email accounts from using third-party clients to send and receive emails. It is a good idea because the ability of dealing with the communication in a large-scale botnet is beyond the scope of human being, and a small-scale botnet can only result in limited impact.

Broadening the Detection Focus. As email service providers are mostly private enterprises, they need to pay special attention to the privacy of their customers. Thus, the cooperation between the defenders and email service providers are limited and only the email service providers can come into contact with the email content. As shown in Table 4, it is true that the spam is the biggest threat in email field. But in fact, the email service providers only focus on checking if an email is a spam with machine learning algorithms. That is to say, if the botmaster can make the emails embedded with commands different from spam, e.g. the frequency of sending, the format and length, it is highly possible to run the botnet well.

Therefore, the email service provider should broaden their detection on spam detection. For instance, if one email account often sends or receives emails with the same content, it is probably controlled by a member of a botnet. Besides, the cooperation with defenders should be strengthened within the range of privacy protection.

Table 4. The percentage of legitimate emails and spams

Email type	Average volume (billions)	Percentage
Legitimate	63.79	14.45%
Spam	65	85.55%

6 Conclusion

As far as we known, we first proposed the email flux botnet. The bot can automatically generate a list of random email addresses for covert email-based C&C communication. The email flux botnet can obviously bypass traditional DNS-based detection methods against fast flux and domain flux, because email flux exploits the communication with good reputation email servers to build stealthy botnet C&C channel without suspicious DNS traffic. And we enhance the traditional DGA algorithm used in domain flux, preventing adversaries from taking down or taking over botnet by registering C&C email account in advance. We discuss the potential countermeasures in the future to mitigate the threat of email flux botnet.

Acknowledgements. The authors acknowledge the National Natural Science Foundation of China (Grant: 61300242, 61772291), the Tianjin Research Program of Application Foundation and Advanced Technology (Grant: 15JCQNJC41500, 17JCZDJC30500), the Open Project Foundation of Information Security Evaluation Center of Civil Aviation, Civil Aviation University of China (Grant: CAAC-ISECCA-201701, CAAC-ISECCA-201702), Project supported by the National Key Basic Research Program of China (Grant: 2013CB834204).

References

1. Symantec: internet security threat report for 2016 (2017). https://www.symantec.com/zh/cn/security-center/threat-report?inid=globalnav_scflyout_istr
2. APWG: global phishing survey for 2016 (2017). https://apwg.org/apwg-news-center/APWG-News/
3. Kaspersky: DDoS attacks in Q2 2017 (2017). https://securelist.com/ddos-attacks-in-q2-2017/79241/
4. Kartaltepe, E.J., Morales, J.A., Xu, S., Sandhu, R.: Social network-based botnet command-and-control: emerging threats and countermeasures. In: Zhou, J., Yung, M. (eds.) ACNS 2010. LNCS, vol. 6123, pp. 511–528. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13708-2_30
5. Yin, T., Zhang, Y., Li, S.: DR-SNBot: a social network-based botnet with strong destroy-resistance. In: IEEE International Conference on Networking, Architecture, and Storage, pp. 191–199 (2014)
6. Singh, K., Srivastava, A., Giffin, J., Lee, W.: Evaluating email's feasibility for botnet command and control. In: IEEE International Conference on Dependable Systems and Networks with Ftcs and DCC, pp. 376–385. IEEE, Anchorage, June 2008
7. Zeng, Y., Shin, K.G., Hu, X.: Design of SMS commanded-and-controlled and P2P-structured mobile botnets. In: Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks, WISEC 2012, pp. 137–148, ACM, New York (2012)
8. Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., Kruegel, C., Vigna, G.: Your botnet is my botnet: analysis of a botnet takeover. In: ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, , pp. 635–647, November 2009
9. Iqbal, S., Kiah, M.L.M., Dhaghighi, B., Hussain, M., Khan, S., Khan, M.K., Choo, K.-K.R.: On cloud security attacks: a taxonomy and intrusion detection and prevention as a service. J. Netw. Comput. Appl. **74**, 98–120 (2016)
10. Osanaiye, O., Choo, K.-K.R., Dlodlo, M.: Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud ddos mitigation framework. J. Netw. Comput. Appl. **67**, 147–165 (2016)
11. Ollmann, G.: Botnet communication topologies. Retrieved September, vol. 30, p. 9 (2009)
12. Salusky, W., Danford, R.: Know your enemy: fast-flux service networks. Honeynet Proj., pp. 1–24 (2007)
13. Pomorova, O., Savenko, O., Lysenko, S., Kryshchuk, A., Bobrovnikova, K.: Anti-evasion technique for the botnets detection based on the passive DNS monitoring and active DNS probing. In: Gaj, P., Kwiecień, A., Stera, P. (eds.) CN 2016. CCIS, vol. 608, pp. 83–95. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39207-3_8
14. Perdisci, R., Corona, I., Giacinto, G.: Early detection of malicious flux networks via large-scale passive dns traffic analysis. IEEE Trans. Dependable Secure Comput. **9**, 714–726 (2012)
15. Porras, P., Di, H., Yegneswaran, V.: A foray into conficker's logic and Rendezvous points. In: USENIX Conference on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More, p. 7 (2009)

16. Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N.: Building a dynamic reputation system for DNS. In: Proceedings of the 19th USENIX Conference on Security, USENIX Security 2010, p. 18. USENIX Association, Berkeley (2010)
17. Bilge, L., Kirda, E., Kruegel, C., Balduzzi, M.: Exposure: Finding malicious domains using passive dns analysis. In: Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, February 2011
18. Antonakakis, M., Perdisci, R., Lee, W., Nikolaos Vasiloglou, I., Dagon, D.: Detecting malware domains at the upper DNS hierarchy. In: USENIX Conference on Security, p. 27 (2011)
19. Guerid, H., Mittig, K., Serhrouchni, A.: Privacy-preserving domain-flux botnet detection in a large scale network. In: International Conference on Communication Systems and Networks, pp. 1–9 (2013)
20. Nguyen, T.-D., CAO, T.-D., Nguyen, L.-G.: DGA botnet detection using collaborative filtering and density-based clustering. In: Proceedings of the Sixth International Symposium on Information and Communication Technology, SoICT 2015, pp. 203–209. ACM, New York (2015)
21. Lee S., Kim, J.: Fluxing botnet command and control channels with URL shortening services. Elsevier Science Publishers B. V. (2013)
22. Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Abu-Nimeh, S., Lee, W., Dagon, D.: From throw-away traffic to bots: detecting the rise of DGA-based malware. In: USENIX Conference on Security Symposium, p. 24 (2011)
23. Yahyazadeh, M., Abadi, M.: BotGrab: a negative reputation system for botnet detection. Comput. Electr. Eng. **41**(6), 68–85 (2015)
24. Sharifnya, R., Abadi, M.: A novel reputation system to detect dga-based botnets. In: International Econference on Computer and Knowledge Engineering, pp. 417–423 (2013)
25. Sharif, M., Lanzi, A., Giffin, J., Lee, W.: Automatic reverse engineering of malware emulators. In: 2009 30th IEEE Symposium on Security and Privacy, pp. 94–109, May 2009
26. Campbell, S., Chan, S., R. Lee, J.: Detection of fast flux service networks. In: Australasian Information Security Conference, pp. 57–66 (2011)
27. Holz, T., Gorecki, C., Rieck, K., Freiling, F.C.: Measuring and detecting fast-flux service networks. In: Network and Distributed System Security Symposium, NDSS 2008, San Diego, California, USA, pp. 487–492, February 2008
28. Yadav, S., Reddy, A.K.K., Reddy, A.L., Ranjan, S.: Detecting algorithmically generated malicious domain names. In: ACM SIGCOMM Conference on Internet Measurement 2010, Melbourne, Australia, pp. 48–61, November 2010
29. Yadav, S., Reddy, A.K.K., Reddy, A.L.N., Ranjan, S.: Detecting algorithmically generated domain-flux attacks with dns traffic analysis. IEEE/ACM Trans. Netw. **20**(5), 1663–1677 (2012)
30. Sharifnya, R., Abadi, M.: Dfbotkiller: domain-flux botnet detection based on the history of group activities and failures in DNS traffic. Digit. Invest. **12**(12), 15–26 (2015)
31. Yadav, S., Reddy, A.L.N.: Winning with DNS failures: strategies for faster botnet detection. In: Rajarajan, M., Piper, F., Wang, H., Kesidis, G. (eds.) SecureComm 2011. LNICST, vol. 96, pp. 446–459. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31909-9_26
32. Jiang, N., Cao, J., Jin, Y., Li, L.E., Zhang, Z.L.: Identifying suspicious activities through DNS failure graph analysis. In: The 18th IEEE International Conference on Network Protocols, pp. 144–153, October 2010

33. Gavrilut, D.T., Popoiu, G., Benchea, R.: Identifying DGA-based botnets using network anomaly detection. In: 2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 292–299, September 2016
34. Schiavoni, S., Maggi, F., Cavallaro, L., Zanero, S.: Phoenix: DGA-based botnet tracking and intelligence. In: Dietrich, S. (ed.) DIMVA 2014. LNCS, vol. 8550, pp. 192–211. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08509-8_11
35. Anderson, H.S., Woodbridge, J., Filar, B.: DeepDGA: adversarially-tuned domain generation and detection. In: ACM Workshop on Artificial Intelligence and Security, pp. 13–21 (2016)
36. Golle, P.: Machine learning attacks against the Asirra CAPTCHA. In: ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, pp. 535–542, October 2008
37. Yan, J., El Ahmad, A.S.: A low-cost attack on a microsoft CAPTCHA. In: Proceedings of the 15th ACM Conference on Computer and Communications Security, CCS 2008, pp. 543–554. ACM, New York (2008)
38. Zhu, B.B., Yan, J., Li, Q., Yang, C., Liu, J., Xu, N., Yi, M., Cai, K.: Attacks and design of image recognition CAPTCHAS. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, pp. 187–200. ACM, New York (2010)