Digital Investigation 12 (2015) 15–26

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

DFBotKiller: Domain-flux botnet detection based on the history of group activities and failures in DNS traffic

Reza Sharifnya, Mahdi Abadi*

Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

ARTICLE INFO

Article history:
Received 28 April 2014
Received in revised form 17 November 2014
Accepted 27 November 2014
Available online 4 January 2015

Keywords:
Botnet detection
Domain-flux botnet
Negative reputation system
Domain group activity
Domain failure

ABSTRACT

Each botnet needs an addressing mechanism to locate its command and control (C&C) server(s). This mechanism allows a botmaster to send commands to and receive stolen data from compromised hosts. To maximize the availability of the C&C server(s), botmasters have recently started to use domain-flux techniques. However, domain-flux botnets have some important characteristics that we can use to detect them. They usually generate a large number of DNS queries resolved to the same IP address and they often generate many failures in DNS traffic. The domain names in the DNS queries are randomly or algorithmically generated and their alphanumeric distribution is significantly different from legitimate ones. In this paper, we present DFBotKiller, a negative reputation system that considers the history of both suspicious group activities and suspicious failures in DNS traffic to detect domain-flux botnets. Our main goal is to automatically assign a high negative reputation score to each host that is involved in these suspicious domain activities. To identify randomly or algorithmically generated domain names, we use three measures, namely the Jensen-Shannon divergence, Spearman's rank correlation coefficient, and Levenshtein distance. We demonstrate the effectiveness of DFBotKiller to detect hosts infected by domain-flux botnets using multiple DNS queries collected from our campus network and a testbed network consisting of some bot-infected hosts. The experimental results show that DFBotKiller can make a good trade-off between the detection and false alarm rates.

Introduction

A *botnet* is a network of compromised hosts, also known as *bots* or *zombies*, which are remotely under control of a botmaster through one or more C&C servers. The botmaster can use the botnet for sending spam, conducting distributed denial of service (DDoS) attacks, stealing personal information, or other malicious activities (Silva et al., 2013). Nowadays, Botnets are considered as one of the serious threats to Internet security, since they facilitate large-scale coordinated attacks using multiple infected hosts.

The bots within a botnet usually employ DNS queries to locate the botnet's C&C server(s). This allows the botmaster to change the real location of the C&C server(s) without reconfiguring its bots. A growing number of new generation botnets use a technique known as *domain-fluxing* as either their primary or secondary evasion strategy. For example, Conficker.C, Kraken, Cycbot, and Murofet employ domain-fluxing as their primary evasion strategy. Meanwhile, Zeus variants are utilizing domain-fluxing as their backup strategy to locate C&C server(s). The main goal of this technique is to generate a large number of domain names for a C&C server, such that more resiliencies against takedown attempts and filtering technologies are provided. Indeed, domain-flux botnets combine the facility of centralized C&C server(s) with the power of P2P structures

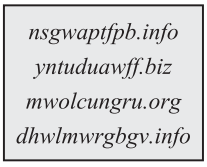
* Corresponding author. Tel.: +98 21 82884935.
E-mail addresses: reza.sharifnyay@modares.ac.ir (R. Sharifnya), abadi@modares.ac.ir (M. Abadi).

http://dx.doi.org/10.1016/j.diin.2014.11.001
1742-2876/

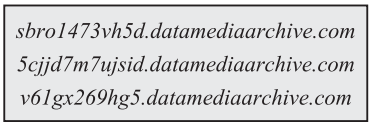
to make their C&C communications more resistant to botnet detection systems or other security measures (Antonakakis et al., 2012).

A domain name is an easy-to-remember identification string for a particular web site, application, or service on the Internet. Actually, it is an alias for an IP address. Each domain name consists of one or more parts, technically called *labels*, which are conventionally concatenated and delimited by dots. The right-most label is known as the *top level domain* (TLD) (Ruan et al., 2013). For example, the label *com* is the TLD of the domain name *example.com*. The second and third labels from the right are known as the *second level domain* (SLD) and the *third level domain* (3LD), respectively. The hierarchy of domain names descends from right to left. Each label to the left specifies a subdomain of the domain name to the right. This tree of subdomains may have up to 127 levels. Hereinafter, we refer to the labels of domain names as *domain labels*.

A domain-fluxing bot dynamically generates a unique list of multiple domain names based on domain wildcarding or a domain generation algorithm (DGA). Domain wildcarding abuses native DNS functionality to wildcard a registered domain name such that all of its subdomains are resolved to the same IP address. The wildcarded information that appears random is used by the botmaster to uniquely identify a victim or bypass anti-spam technologies (Ollmann, 2009). Domain generation algorithms generate a large number of domain names based on an initial seed. For example, Conficker.C generates 50,000 domain names every day by using the current date and time at UTC as the seed. Also, Kraken generates specific English-language alike words and combines each of them with a randomly chosen suffix, such as -able, -dom, -ment, -ship, or -ly (Yadav et al., 2012). Fig. 1 shows some domain names generated by Conficker.C and Cycbot. The domain names are then resolved by sending DNS queries as the bot tries to locate the C&C server(s) in hopes that the botmaster has registered at least one or more of them. Since the domain names are dynamically generated in volume and typically have a short time-to-live (TTL), it is very difficult to filter out them. To predict future domain names, a security vendor has to reverse engineer the generation algorithm, which is a time-consuming process and during this time the botmaster may command his bots to change the algorithm.



(a) Conficker.C



(b) Cycbot

Fig. 1. Domain names generated by domain-flux botnets.

Domain-flux botnets can be identified by the following characteristics (Yadav and Reddy, 2012; Choi and Lee, 2012): (1) the bots generate a large number of DNS queries, (2) the domain names in the DNS queries are generated randomly or algorithmically and their alpha-numeric distribution is significantly different from human-generated ones, (3) the generated domain names are often mapped to the same IP address or having the same TLD and SLD, and (4) many of the DNS queries are failed as many of the generated domain names may not be registered.

During the past few years, some efforts have been focused on the detection of suspicious domain activities (Stalmans and Irwin, 2011; Yadav and Reddy, 2012; Bilge et al., 2014), but none of them consider the history of these activities in the monitored network. This makes the detection system has a potentially high false alarm rate. To address this shortcoming, we present DFBotKiller, an online negative reputation system that considers the history of both suspicious domain group activities and suspicious domain failures to automatically assign a high negative reputation score to each host infected by domain-flux botnets.

In general, reputation systems represent a significant trend in supplementary decision making services. The basic idea is to calculate a reputation score for each object within a community or domain, based on a set of opinions held about it. The systems have been extensively used in many applications, such as multiagent systems (Sabater and Sierra, 2001), P2P networks (Kamvar et al., 2003), mobile ad-hoc networks (Ibrohimovna and Heemstra de Groot, 2010), and so on, but few studies to date have applied them for botnet detection.

We consider suspicious activities in DNS traffic for three reasons: (1) DNS traffic is a small percentage of the network traffic. Hence, its monitoring has less overhead than monitoring the whole network traffic. (2) In domain-flux botnets, a large number of DNS queries are daily generated to locate C&C server(s). Because the domain names in these queries are randomly or algorithmically generated, the probability of detecting bot-infected hosts is increased. (3) Domain-flux botnets often generate many failed DNS queries in early stages of their life-cycles. Therefore, we can quickly detect bot-infected hosts before performing any malicious activity.

The remainder of this paper is organized as follows. We first describe domain-flux botnets in Section “Domain-flux botnets” and then briefly review related work in Section “Related work”. In Section “DNS-based negative reputation system”, we present DFBotKiller and evaluate it in Section “Experimental results”. Finally, we give some conclusions in Section “Conclusion”.

Domain-flux botnets

Domain-flux botnets are a new generation of botnets that generate a large number of domain names, randomly or algorithmically, to protect their C&C infrastructures from takedowns (Bilge et al., 2014). They often use domain wildcarding or domain generation algorithm (DGA) to constantly change and allocate multiple domain names to their C&C server(s). This allows them to bypass the domain