Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis

*Tzy-Shiah Wang, Hui-Tang Lin *, Wei-Tsung Cheng, Chang-Yu Chen*

Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, Tainan City, Taiwan

ARTICLE INFO

Article history:
Received 18 January 2016
Received in revised form 15 August 2016
Accepted 3 October 2016
Available online 5 October 2016

Keywords:
Domain generation algorithm
Botnet detection mechanism
Name error response
Traffic analysis
Network security

ABSTRACT

Botnets are one of the leading threats to network security nowadays and are used to conduct a wide variety of malicious activities, including information theft, phishing, spam mail distribution, and Distributed Denial of Service (DDoS) attacks. Among the various forms of botnet, DGA-based botnets, which utilize a Domain Generation Algorithm (DGA) to avoid detection, are one of the most disruptive and difficult to detect. In such botnets, the DGA is used to generate a huge list of candidate Command and Control (C&C) server domains, and the bot then attempts to connect to an active C&C server by querying each DNS server in turn. DGA-based botnets are highly elusive and difficult to detect using traditional defensive mechanisms and therefore have a high survivability. Accordingly, this study proposes a DGA-based botnet detection scheme designated as DBod based on an analysis of the query behavior of the DNS traffic. The proposed scheme exploits the fact that hosts compromised by the same DGA-based malware query the same sets of domains in the domain list and most of these queries fail since only a very limited number of the domains are actually associated with an active C&C. The feasibility of the proposed method is evaluated using the DNS data collected from an education network environment over a period of 26 months. The results show that DBod provides an accurate and effective means of detecting both existing and new DGA-based botnet patterns in real-world networks.

1. Introduction

The rapid development of the Internet has brought about many changes in our daily lives. While many of these changes are beneficial in that they facilitate a more convenient and enjoyable lifestyle, the increasing reliance on the Internet and associated network services exposes users to the risk of malicious attacks by third parties intent on causing disturbance or more severe long-term damage. Among the various network security concerns, botnets are regarded as one of the most critical. In a typical botnet attack, a set of compromised hosts known as zombies or bots are manipulated by a controller (referred to as a botmaster) through a Command and Control (C&C) server to launch a range of illegal activities such as spam distribution, information theft, Distributed Denial of Service (DDoS) attacks, and so on (Goebel and Holz, 2007). Typically, the bots are distributed over multiple networks and are therefore difficult to detect (Salusky and Danford, 2007). Consequently, botnets have high survivability and the potential to cause massive disruption and damage.

Various approaches for detecting botnets have been proposed over the last decade (Abu Rajab et al., 2006; Gu et al., 2007, 2008; Stone-Gross et al., 2009). However, these

* *Corresponding author.*
Email addresses: tcwang@nsda.ee.ncku.edu.tw (T.-S. Wang), htlin@mail.ncku.edu.tw (H.-T. Lin), wtcheng@nsda.ee.ncku.edu.tw (W.-T. Cheng), markisgood@nsda.ee.ncku.edu.tw (C.-Y. Chen).
http://dx.doi.org/10.1016/j.cose.2016.10.001
0167-4048/

countermeasures have simply prompted botmasters to develop more sophisticated attack technologies in response. Most current botmasters use P2P-based C&C structures such as Zeus (Macdonald and Manky, 2009; ZeuS Gets More Sophisticated Using P2P Techniques, 2011), Waledac (Williams, 2010) and Storm (Stewart, 2008). P2P-based botnets are difficult to deploy and maintain. However, they provide a robust C&C structure, which avoids the single point of failure (SPOF) problem (Zeidanloo and Manaf, 2009) and hence improves the resilience of the bot toward traditional defense mechanisms. To conceal their malicious activities and reduce the risk of detection, attackers have recently developed a new type of botnet known as a DGA-based botnet, which avoids blocking by dynamically migrating the C&C server. Many such botnets have been identified, including Kraken (Yadav et al., 2012), Srizbi (Yadav et al., 2012), Mjuyh (Yadav et al., 2012), Conficker-A/B (Porras, 2009), Conficker-C (Porras et al., 2009), Murofet (Shevchenko, 2010) and Torpig (Stone-Gross et al., 2009). In a typical attack, each bot periodically executes a Domain Generation Algorithm (DGA) to produce a list of candidate C&C domains based on a random seed (e.g., popular search strings, the current date or time, and so on). The seed used to generate the list of candidate domains varies from one botnet to another. For example, Kraken chooses a random string and then combines this string with a randomly chosen suffix, such as -able, -ment or -ly (Yadav et al., 2012). Mjuyh uses the domain name "Mjuyh.com" as the top-level-domain (TLD) and second-level-domain (SLD) and chooses random alphanumeric characters as the fourth-level-domain (Yadav et al., 2012). Torpig uses the current week and year (Stone-Gross et al., 2009). By contrast, Conficker uses the current Coordinated Universal Time (UTC) (Porras, 2009; Porras et al., 2009). Having obtained a list of domains using the chosen seed, the bot attempts to resolve the domain names by sending DNS queries continuously until one of the domains is found to be active and mapped to an existing C&C server. The DGA-based approach greatly increases the botnet survivability since each bot can easily obtain the IP address of another active C&C server in the event that the current server is blocked by the defender. In other words, the DGA strategy combines the benefits of a centralized botnet structure with the survivability of a P2P structure and is thus highly resistant toward traditional detection mechanisms (Antonakakis et al., 2012; Sharifnya and Abadi, 2013).

Although DGA-based botnets use multiple domain names to increase their survivability, they still leave behind several clues which can be exploited to observe and detect their presence. For example, the bots generate a large number of identical DNS queries since the domains generated by the different bots using the same DGA algorithm are the same. Secondly, the bots generate numerous Name Error responses (Mockapetris, 1987), also known as NXDOMAIN responses, since most of the generated domain names have no actual IP address (Stone-Gross et al., 2009). (Note that for convenience, these domain names are referred to simply as NXDomains in the remainder of this study.) DNS-based detection is a valuable tool for botnet detection since it provides particular advantages in regard to botnet-employed encrypted protocols and changes in traffic behavior (Lee et al., 2010). Several methods have been proposed for detecting botnets based on an observation of the C&C communication pattern (Antonakakis et al., 2012; Bilge et al., 2011). However, these approaches require prior training using traffic obtained from known DGA-based botnets. The training process potentially degrades the accuracy of the detection results if, for example, the botnet behavioral features change while the system is still undergoing training. Moreover, the detection performance is reliant on the availability of a training sample with a sufficient size.

Accordingly, the present study proposes a new DGA-based botnet detection scheme known as DBod. DBod is deployed "below" the local recursive DNS (RDNS) server and is used to monitor the DNS query/response messages from/to the hosts within the network. Importantly, DBod monitors only the DNS traffic and is thus more practical and efficient than detection schemes which monitor all of the network traffic. Furthermore, as with the Pleiades system (Antonakakis et al., 2012), DBod leverages only the unsuccessful DNS resolutions to perform botnet detection (i.e., DBod acts on only a subset of the DNS data rather than all the data). DBod includes both a clustering algorithm and a group detection algorithm designed to identify clusters comprised of bot-compromised hosts. The proposed scheme is motivated by the fact that bots belonging to the same DGA-based botnet (i.e., using the same domain generation algorithm) query the same domain list and most of these domains are queried with only a low likelihood by normal (i.e., non-malicious) users.

Notably, DBod has the ability to perform botnet detection without the need for prior training or the large-scale deployment of DNS analysis tools. Moreover, DBod captures the clusters comprising bot-compromised hosts using a statistical behavioral analysis approach. Consequently, it has the ability to identify novel DGA-based botnets without requiring any prior sample of the related malware family. The main contributions of this study can be summarized as follows:

- A DGA-based botnet detection mechanism designated as DBod is proposed based on an analysis of the unsuccessful domain name resolutions. More specifically, DBod distinguishes bot-compromised clusters from normal clusters on the basis of the difference in their query behaviors of bots and normal users, respectively. DBod provides an efficient means of identifying bot-compromised hosts and is scalable to large ISP networks.
- DBod has the ability to detect DGA-based botnets without the need for prior training or knowledge of the related botnet behavior. As a result, it is capable not only of identifying existing DGA-based botnet patterns, but also new patterns not previously reported in the literature.
- The detection performance of DBod is evaluated using both emulated DNS traffic and traffic collected from a real-world network over a period of 26 months. The latter analysis facilitates the derivation of a novel metric which can be used to identify DGA-based botnet patterns.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work in the field. Section 3 provides an overview of the proposed DBod scheme. Section 4 introduces the detailed DBod algorithm and associated assumptions. Section 5 presents and discusses the performance results obtained for the DBod algorithm using emulated network data and real-world DNS traffic data. Finally, Section 6 presents some brief concluding remarks.

2. Related work

Botnets are a particularly virulent form of cyber-attack with the potential to cause massive and long-lasting disruption. Accordingly, the problem of developing effective schemes for detecting botnet activity has attracted significant attention in the network security field. Depending on the particular method used, Botnet detection schemes can be broadly classified as either *signature-based* or *anomaly-based* (Silva et al., 2013). Schemes of the former type extract pattern information from the monitored traffic, mark the signature, and register it on a blacklist. For example, Kugisaki et al. (2007) analyzed the behavior of bots using Internet Relay Chat (IRC) to communicate and then used the analysis results (e.g., the connection distribution and interval) to derive the signature required to identify the associated botnet behavior. Wurzinger et al. (2009) used the network traffic traces associated with known bot instances to generate a botnet detection model which exploited the fact that all bots receiving a command from the botmaster respond to the command in a particular way. Signature-based methods provide a relatively straightforward means of identifying botnet activity. However, since they are based on historical botnet patterns, they invariably fail to detect new botnets with a different communication behavior. While this problem can be resolved by constantly updating the signature database, the associated data analysis task greatly increases the processing cost and reduces the overall performance efficiency (Liu et al., 2009).

Anomaly-based techniques perform botnet detection by identifying network traffic anomalies. Goebel and Holz (2007) presented a technique for detecting IRC-based botnets based on an evaluation of well-known IRC channel name patterns. Lu and Ghorbani (2008) developed an IRC botnet detection scheme based on anomalies in the request response time. Gu et al. (2007) proposed a passive botnet detection system designated as BotHunter, in which networking activities were associated with particular state-based infection models in accordance with their dialogue behavior. However, since the models are predefined, BotHunter is unable to detect malwares whose behaviors differ from those of existing bots. BotSniffer (Gu et al., 2008) exploits the fact that all bots belonging to the same malware respond to a command from the botmaster in the same way. BotMiner (Gu et al., 2008) improves the detection performance of BotSniffer by employing a cross-cluster correlation technique to identify hosts which share both similar communication patterns and similar malicious activities.

However, anomaly-based detection methods fail if the communications are encrypted (Silva et al., 2013). Many researchers have thus proposed botnet detection schemes based on an analysis of the DNS traffic. For example, Stalmans and Irwin (2011) proposed a scheme for detecting Fast Flux (FF) botnets through an analysis of the DNS queries, while Choi et al. (2009) presented an online unsupervised botnet detection scheme referred to as BotGAD inspired by the fact that hosts compromised by the same malware exhibit similar DNS traffic activities. However, BotGAD incurs a high processing cost due to the need to monitor a huge volume of network traffic (Feily et al., 2009). Villamarín-Salomón and Brustoloni (2009) presented two Bayesian-based approaches for DNS botnet detection based on looking for either domain names with abnormally high or temporally-concentrated query rates or abnormally-recurring DDNS replies, respectively. Stone-Gross et al. (2009) used a DNS-based detection approach to identify a new class of botnet known as Domain Flux botnets, in which the bots generate numerous domain names using a DGA and then attempt to establish communications with a subset of them. The same authors used a similar approach to identify Torpig attacks, in which the bots take common words used in Twitter searches as seeds to generate new domain names every day. Yadav et al. (2010) presented an efficient approach for identifying botnets based on the temporal correlation of successful and failed DNS queries and the entropy of the domain names belonging to each query (Yadav and Reddy, 2012). However, malicious traffic may egress from the network before the system has the chance to classify a domain as possibly malicious (Stalmans and Irwin, 2011).

Most of the methods described above rely on a reverse engineering of the botnet malware executables. However, in many practical cases, such knowledge is not available in advance. To address this problem, Bilge et al. (2011) proposed a method designated as Exposure for detecting DGA-based botnets, phishing sites and FF domains based on four pre-defined categories of features. Antonakakis et al. (2012) presented a method known as Pleiades for detecting randomly-generated domains based on an analysis of the NXDomain responses. However, both detection methods (Antonakakis et al., 2012; Bilge et al., 2011) require prior training using traffic obtained from known DGA-based botnets. Furthermore, Exposure requires a training period of almost seven days to guarantee an adequate detection capability. The long training process potentially degrades the accuracy of the detection results if, for example, the botnet behavioral features change while the system is still undergoing training. Moreover, the detection performance is reliant on the availability of a training sample with a sufficient size. Zhou et al. (2013) introduced a botnet detection scheme based on the rationale that an NXDomain queried by a group of hosts in a certain period of time is statistically unlikely to be queried again in the following period of time. Accordingly, the proposed scheme clusters the requested domains in accordance with their top-level-domain and then inspects the clusters for similar lifetime patterns. However, not all DGA botnets generate candidate C&C domain lists with the same top-level-domain (e.g., the domains generated by Conficker (Porras et al., 2009) choose a top-level domain at random, while those generated by Mjuyh (Yadav et al., 2012) utilize various third-level-domains (3LD)). Schiavoni et al. (2014) presented an approach known as Phoenix for identifying DGA domains based on an inspection of the IP features and certain linguistic features. Phoenix consists of three modules, namely a discovery module, a detection module and a module for intelligence and insights. The detection module detects domains whose names are generated automatically, and are thus unreadable or meaningless. However, some DGA-based botnets, such as Torpig (Stone-Gross et al., 2009), Kraken (Yadav et al., 2012) and Srizbi (Yadav et al., 2012), are designed to generate human-readable domain names. Moreover, benign domain names which are not obviously human-understandable confuse the scheme and reduce its detection accuracy (Mowbray and Hagen, 2014). Sharifnya and Abadi (2015) presented a negative reputation

system, referred to as DFBotKiller, which considers the history of both suspicious group activities and suspicious domain failure traffic. The proposed method identifies bot-infected hosts utilizing a statistical approach and works online. However, it suffers a high false alarm rate if a sufficient history of suspicious domain activities is not considered (Sharifnya and Abadi, 2015).

3. System overview

The DNS is a critical component of the Internet infrastructure with the responsibility of translating the domain name to the corresponding IP address whenever a host tries to query a domain. Because a DGA bot must query a DNS server before connecting to the C&C server, it inadvertently leaves a trace in the DNS traffic. Thus, in the DGA-based botnet detection scheme proposed in this study, designated as DBod, compromised and normal users are differentiated based on an analysis of the DNS traffic. This section provides a high-level overview of the proposed scheme. As shown in Fig. 1, DBod consists of a Filtering module, a Clustering module and a Group Identification module. The details of each module are briefly discussed in the following.

3.1. *Filtering module*

As shown in Fig. 1, DBod commences by discarding the active domains in the DNS traffic. In other words, DBod operates on the rationale that most of the domains in the candidate list generated by the DGA have no actual IP address and therefore result in an NXDomain response when queried. Accordingly, DBod distinguishes bot-compromised hosts from normal users on the basis of differences in the query behaviors of the NXDomains. In practice, NXDomains are generated not only by compromised hosts (i.e., bots), but also by legitimate users under certain circumstances. The normal NXDomains cause noise in the detection process, and therefore increase both the number of false alarms and the calculation time. Accordingly, in DBod, the detected NXDomains are passed to the Filtering module such that "normal" NXDomains can be identified and removed. Most of the normal NXDomains are the result of misconfigurations, or the output of legitimate applications such as Bit Torrent and Spamhaus (The Spamhaus Project). Thus, they can be easily identified since they have the same TLD and SLD when queried. For example, when using Spamhaus to check whether "example.net" is malicious, the host queries the domain "example.net.sbl.spamhaus.org" and obtains an NXDomain response. NXDomains may also be legitimately produced when high-traffic-volume websites such as Facebook, for example, go offline temporarily as a result of technical issues. It is difficult to establish a simple rule for filtering out such NXDomains. However, the noise is typically short lived and appears with only a low probability. As a result, it has a limited impact on the detection performance over the long-term, and can thus be effectively ignored.

The final cause of legitimate NXDomains is that of typos. As described above, bots compromised by the same DGA-based malware tend to query the same list of NXDomains. The probability that two different users make the same typo when spelling a domain name within the same time frame is very small. Consequently, it is reasonable to infer that NXDomains which are queried by only one user within a certain period of time are most likely normal domains. Thus, the filtering module removes any NXDomains queried by a single user where this user exhibits only one query behavior within the considered time window.

Using the filtering rules described above, DBod "passes" only those NXDomains which have been generated by bots, or are produced as a result of habitual typos or the temporary closure of legitimate websites. As described in the following section, the filtered NXDomains are input to the Clustering module for further processing. Notably, the filtering process reduces the volume of data to be processed and therefore simplifies both

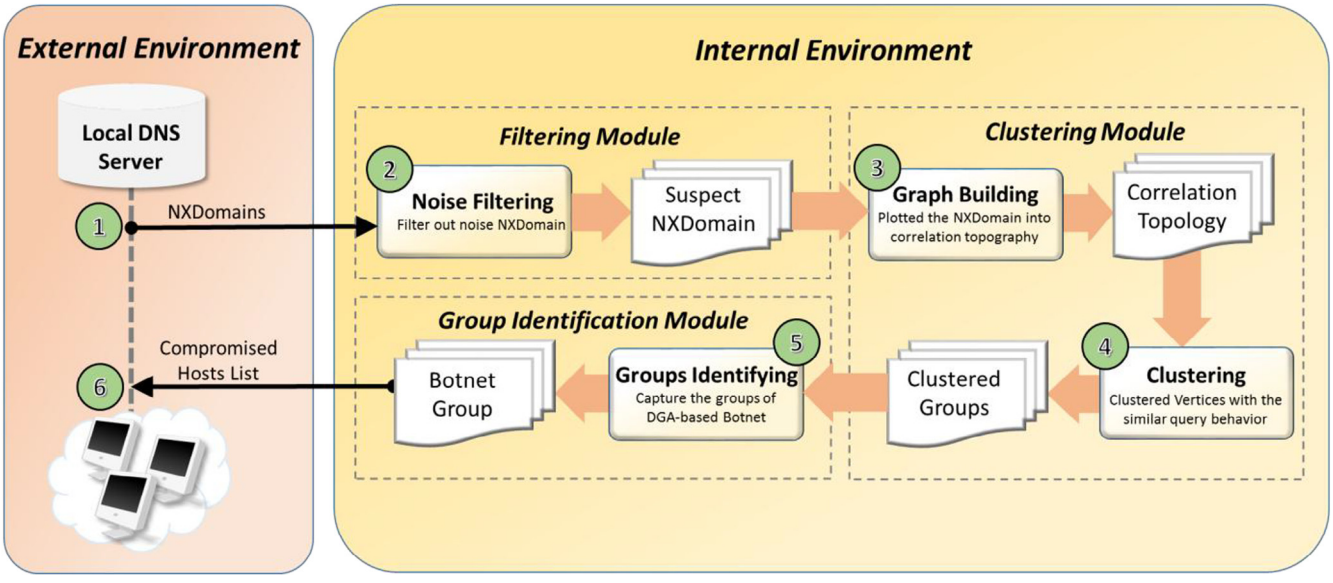

Fig. 1 – System architecture of DBod.

the clustering process and the other downstream fine-grained analysis tasks.

3.2. Clustering module

The aim of the Clustering module is to group the hosts compromised by the same DGA algorithm. To achieve this goal, the query behaviors of the NXDomains are first translated to a correlation topology in accordance with their relationship intensities. (Note that the relationship intensity represents here the similarity of the query behaviors of two hosts; where a more similar behavior indicates a more intense, i.e., close relationship.) The clustering process is then performed based on an inspection of the relationship intensities among the various hosts. Generally speaking, the NXDomains generated by hosts belonging to the same DGA-based botnet have a high degree of similarity during a given epoch. In other words, the query behaviors of the hosts compromised by the same DGA-based botnet have a similar scope. Furthermore, the NXDomains generated by the DGA algorithm are queried only once by each bot in most cases. By contrast, the NXDomains generated as a result of typos or temporary website closures, for example, are generally queried multiple times since most users either retype the domain name (in the event of a typing error) or attempt to reconnect to the chosen site (in the event of website closure). As a result, the hosts compromised by the same DGA-based botnet tend to exhibit a similar "distribution" of their query domains, whereas those of normal users do not. Consequently, in producing the correlation topology, the Clustering module considers the similarity in both the scope of the query behaviors of the hosts and the distributions of their query behavior. In DBod, the clustering process is performed using the Chinese Whispers (CW) algorithm (Biemann, 2006) since it is time-linear with respect to the number of edges and is thus capable of handling very large graphs in a reasonable time. Having performed the clustering process, the clustering results are output to the Group Identification module for further processing.

3.3. Group identification module

In the Group Identification module, the bot-compromised hosts are distinguished from normal hosts using a supervised statistical algorithm based on two behavioral characteristics, namely (i) the query time distribution, and (ii) the query count distribution. Intuitively, for a certain group of hosts, the probability of two normal users querying the same NXDomai n at the same time is very low. However, for hosts which are compromised by the same DGA-based botnet, the probability is much higher since both hosts attempt to connect to a C&C server using the same list of candidate domains. Consequently, the similarity in the query times of two different hosts provides a useful behavioral characteristic for detecting DGA-based botnets. The query count distribution of the hosts also provides a useful indication of DGA-based botnet activity since each bot usually queries each domain on the candidate list just once. Thus, clusters in which the hosts query the same NXDomain repeatedly have a greater likelihood of being normal clusters, while those in which the hosts query the same NXDomain only once have a greater probability of being compromised clusters.

4. DBod design

4.1. Definitions and notations

For each time window T of a day, DBod receives an input stream of DNS messages from the local recursive DNS server within the monitored network. To extract the features of interest from the DNS traffic, it is necessary to develop data structures which provide access to the involved fields as required. From the discussions in the previous section, DBod requires only information relating to the query domains, the hosts and the query times. Consider a network consisting of N domains queried by M hosts. The DNS traffic required by DBod, referred to as a *flow*, can be modeled as:

$$f = \{D, H, T\}, \tag{1}$$

where $D = \{d_u\}_{u=1\ldots N}$ is the set of domains; $H = \{h_i\}_{i=1\ldots M}$ is the set of hosts; and $T = \{t_{iu}(\alpha) \mid 1 \leqq i \leqq M, 1 \leqq u \leqq N\}$ is the time at which host h_i queries domain d_u for the α_{th} time within the current time window. (Note that $\alpha > 1$ if h_i queries d_u repeatedly within the time window. Specifically, α is equal to the number of occasions on which h_i queries d_u in the time frame. Furthermore, $t_{iu}(\alpha)$ denotes the time at which each query occurs. Thus, if h_i queries d_u twice within the time window, then $|t_{iu}(\alpha)| = 2$ and two time records are obtained, i.e., $t_{iu}(1) = 10{:}32$ and $t_{iu}(2) = 10{:}38$ for example.) Thus, the query behavior of the hosts can be modeled using a query matrix Q in which the elements have an entry of $|t_{iu}(\alpha)|$ if the host queries the domain within the considered time window T, i.e.,

$$Q(i,u) = \begin{cases} |t_{iu}(\alpha)|, & \text{if} \quad h_i \text{ ever queried } d_u \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let the subset of domains queried by host h_i be denoted as D_i. The relationship intensity between any two hosts h_i and h_j, where $i \neq j$, can then be evaluated using the following correlation matrix:

$$R(i,j) = \begin{cases} 1 - \sum_{u=1}^{N} \frac{|Q(i,u) - Q(j,u)|}{|D_i \cap D_j| \cdot \max\{Q(i,u), Q(j,u)\}}, & \text{if} \quad D_i \cap D_j \notin \varnothing \\ 0, & \text{if} \quad D_i \cap D_j \in \varnothing. \end{cases} \tag{3}$$

where $R(i,j) \in [0,1]$ satisfies the following properties:

1. $R(i,j) = R(j,i) \geq 0$.
2. $R(i,i) = 0$
3. $R(i,j) = 0$ if i does not query the same domain as j.

4.2. Filtering module

The Filtering module in DBod uses a whitelist of normal NXDomains to filter the domain information received from the DNS server so as to reduce the calculation time of the detection process and minimize the number of false alarms. More specifically, the filtering process is performed using four common block list datasets, namely Spamhaus (The Spamhaus Project), BRBL (Barracuda Reputation Block List), SpamCop (SpamCop) and AHBL (The Abusive Hosts Blocking List). (It is

noted that while AHBL is no longer in operation, some mail servers still attempt to use its services.) NXDomains which appear on these four datasets are filtered out since they are regarded to be the result of normal query behavior. By contrast, the remaining NXDomains are regarded as "candidate bot-generated NXDomains" and are thus passed to the Clustering module for further analysis.

4.3. *Clustering module*

As described in Section 3, the filtered hosts are grouped in accordance with similarities in their query behavior using the CW algorithm. The clustering process is performed using the correlation topology $G = (V, E)$, in which V is the set of vertices (where $V = H$) and E is the set of edges (where $E = \{(h_i, h_j, R(i,j)) | h_i, h_j \in H\}$). The output of the Clustering module has the form $C = \{c_k\}_{k=1\ldots K}$, where C is a set of K clusters.

4.4. *Group identification module*

The Group Identification module receives set C from the Clustering module and, for each candidate cluster c_k in C, identifies whether or not the cluster represents a DGA-based botnet group by performing a statistical analysis of the query time distribution and query count distribution of the group, as described in Section 3.3. To facilitate the discussions, let the flow f to which the hosts in group c_k belong be denoted as f^k, where $f^k = \{D^k, H^k, T^k\}$. Here, $H^k = \{h_i\}_{i=1\ldots M^k}$ is the set of hosts belonging to cluster c_k; $D^k = \{d_u\}_{u=1\ldots N^k}$ is the set of NXDomains queried by H^k; and $T^k = \{t^k_{iu}(\alpha) | 1 \leqq i \leqq M^k, 1 \leqq u \leqq N^k\}$ is the α_{th} time at which host h_i queries NXDomain d_u. Let the following score function $S(k)$ for cluster c_k be introduced:

$$S(k) = w_T \cdot S_T(k) + w_C \cdot S_C(k), \tag{4}$$

where $w_T, w_C \geqq 0$ are weighting coefficients with $w_T + w_C = 1$. Furthermore, $S_T(k)$ and $S_C(k)$ are the query time distribution score and query count distribution score for cluster c_k, respectively. $S_T(k)$ is computed as

$$S_T(k) = \frac{1}{N^k} \sum_{u=1}^{N^k} \sqrt{\frac{1}{\sum_{i=1}^{M^k} Q(i,u)} \sum_{i=1}^{M^k} \sum_{\alpha=1}^{Q(i,u)} \left(\frac{t^k_{iu}(\alpha)}{m^k_T(u)} - 1 \right)^2}, \tag{5}$$

where $m^k_T(u)$ is the mean query time of all the hosts belonging to cluster c_k which queries NXDomain d_u in the considered time frame, i.e.,

$$m^k_T(u) = \frac{\sum_{i=1}^{M^k} \sum_{\alpha=1}^{Q(i,u)} t^k_{iu}(\alpha)}{\sum_{i=1}^{M^k} Q(i,u)}. \tag{6}$$

From inspection, the value of $S_T(k)$ ranges from 0 to 1, where a value closer to 0 implies a greater similarity between the query times of the hosts belonging to the same group (indicating that the group is more probably a botnet group), and *vice versa*.

The distribution of the query count within cluster c_k is evaluated by performing a random walk in the bipartite graph consisting of the hosts belonging to cluster c_k and the NXDomains queried by these hosts. Let the undirected bipartite graph be denoted as $G^k = (V^k, E^k)$, in which V^k is the set of vertices (where $V^k = \{H^k \cup D^k\}$) and $E^k = \{e_{iu}\}$ is the set of edges (where e_{iu} equals $Q(i,u)$ if host h_i queries NXDomain d_u and 0 otherwise). For a graph G^k, the random walk starts at an arbitrary vertex and takes random steps through the graph with length $\lambda \cdot (M^k + N^k)$, where λ is a constant. (In general, a higher value of λ yields a more stable performance. In the current study, a preliminary investigation showed a value of $\lambda = 100$ to be suitable.) In performing the random walk, the probability of taking a step from vertex h_i belonging to H^k to vertex d_u belonging to D^k is given by:

$$P_u(i) = \frac{Q(i,u)^{-1}}{\sum_{i=1}^{M^k} Q(i,u)^{-1}}. \tag{7}$$

By contrast, the probability of taking a step from vertex d_u belonging to D^k to vertex h_i belonging to H^k is equal to:

$$P_i(u) = \frac{Q(i,u)^{-1}}{\sum_{u=1}^{N^k} Q(i,u)^{-1}}. \tag{8}$$

Equations (7) and (8) follow intuitively since the greater the number of times that host h_i queries domain d_u (i.e., the higher the value of $Q(i,u)$), the less likely it is that domain d_u is a C&C domain. Having completed the random walk through G^k, the number of times each node has been stepped on can be calculated and denoted as x_u and y_i for vertexes d_u and h_i, respectively. The query count distribution score $S_C(k)$ can then be obtained as:

$$S_C(k) = \sqrt{\frac{1}{N^k M^k} \sum_{u=1}^{N^k} \left(\frac{x_u}{\bar{x}} - 1 \right)^2 \sum_{i=1}^{M^k} \left(\frac{y_i}{\bar{y}} - 1 \right)^2}, \tag{9}$$

where $\bar{x}$ and $\bar{y}$ are the mean values of x_u and y_i, respectively. By definition, $S_C(k)$ has a value in the interval of [0,1]. A value close to 0 implies that at least one of the following cases occurs: i) the query behaviors of all the hosts belonging to the same group have a high similarity; and ii) the query behaviors of the associated NXDomains have a high similarity.

Having determined the value of $S_C(k)$, the score function $S(k)$ for cluster c_k can be obtained from Eq. (4). Here, a lower score implies that the cluster is, with a higher probability, malicious (i.e., bot-compromised) group. However, the score does not in itself indicate whether or not the group is truly a bot group. Thus, a final identification result is obtained using a prespecified threshold parameter, θ. More specifically, the final identification result for each cluster c_k is obtained in accordance with:

$$\beta_k = \begin{cases} 0, & \text{if } S(k) < \theta \\ 1, & \text{otherwise.} \end{cases} \tag{10}$$

Here, $\beta_k = 0$ indicates that cluster c_k is judged to be malicious, while $\beta_k = 1$ indicates that the cluster is benign. Clearly, the identification outcome depends on the value assigned to the threshold parameter, θ. In the present study, a threshold value of $\theta = 0.2$ was found to yield an effective detection performance (based on an observation period of 26 months).

5. Evaluation

It is difficult to evaluate the performance of any botnet detection scheme in a live operational environment since the ground truth regarding the nature of the identified data is not easily confirmed. For example, given a host cluster c_k identified as a bot-compromised cluster by DBod, the cluster may actually fall into three categories: i) the hosts of c_k or their query domains are listed in a known blacklist (i.e., the result is true positive); ii) c_k represents legitimate host behavior (i.e., the result is false positive); or iii) the real nature of c_k is unknown (i.e., no prior information regarding the cluster is available in any security data source). Thus, confirming the correctness of the detection result in real-world live networks is difficult and usually requires extensive manual analysis. Accordingly, in this study, the detection performance of DBod was evaluated through two different sets of experiments. In the first series of experiments, the sensitivity of DBod to the botnet parameters was evaluated in an emulated network environment consisting of computer-generated traffic produced using a DNS cache probing technique. Notably, by using computer-generated traffic, it is easy to confirm whether a cluster identified by DBod as malicious is truly a bot-compromised cluster. In the second series of experiments, DBod was applied off-line to the DNS traffic data collected over a 26-month period from an education network in Tainan City, Taiwan. In both sets of experiments, DBod was run on an Ubuntu server with an Intel Core i5 3.1 GHz processor and 12 GB RAM.

5.1. *Overview of Tainan city education network dataset*

The real-world DNS traffic was generated by more than 10,000 users (including students, teachers and employees) of the Education Network of Tainan City (http://cc.ncku.edu.tw) over a period extending from May 2013 to June 2015. As described in Section 1, the monitoring point was "below" the DNS servers; thus providing visibility of the NXDomains generated by the individual hosts. Notably, no interaction with the users took place during the collection period, and hence the data can be regarded as reliable ground truth data. Fig. 2 shows the average hourly number of queries generated by the hosts in the monitored environment. It is seen that the total number of hourly queries is around 2.8 million on weekdays and 1.6 million on weekend days. Moreover, it is observed that the query behavior of the hosts is strongly correlated to the daily routines of the users. In other words, the bulk of the queries are generated during normal working hours (8 am ~ 5 pm) or during the evening.

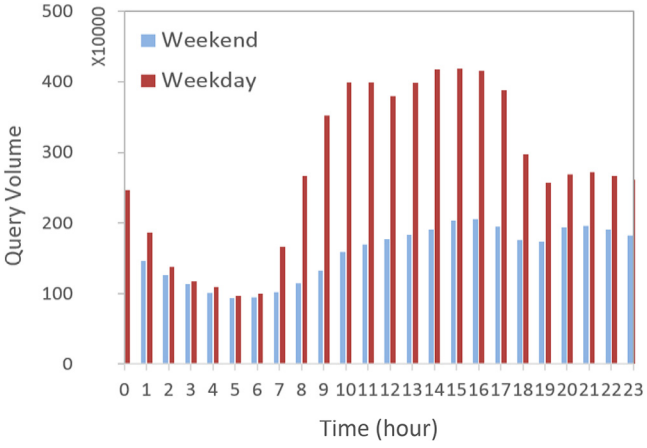

Fig. 2 – Average daily DNS traffic on weekdays and weekends.

On average, around 600 NXDomains per hour were collected from the network for analysis purposes. That is, botnet detection was performed using around 0.02% of the total number of queries. Following the filtering process, the raw traffic volume was reduced by approximately 10% on average. Therefore, the average size of the correlation matrix R used to perform clustering was 500×500. In other words, the scale of the detection process was reduced to a manageable size compared to that of a scheme based on all the network traffic.

5.2. *Performance evaluation using computer-generated malicious traffic*

This section evaluates the performance of the proposed DBod scheme in the emulated computer network given various settings of the botnet parameters.

5.2.1. *Evaluation dataset and setup*

100 hosts were selected from the Education Network described above, and the DNS queries generated by these hosts were collected over a period of five days. On each day, the length of the time window T was set as 1 hour. On average, each host was found to generate about 6 NXDomains per hour (with a total of 70,035 NXDomains). Malicious DNS traffic was generated in accordance with four different botnet patterns obtained from the Offensive Computing (Open Malware – Community Malicious Code Research and Analysis) and Virus Share (Virus Share) websites (i.e., Kraken (MD5: 04966960f3f5ed32ae479079a1bcf6e9), Conficker (MD5: c065a52047091953edf00aca21d3e726), Cycbot (MD5: 0007016380ffa951d661882762f3b59c), and Murofet (MD5: 19c50a3d1f41f62d7838cbdc989e4b49)). (Notably, the chosen botnet patterns are classified as bot malware by more than three-quarters of the antivirus programs on the Virus Total website (Virus Total).) For each botnet pattern, malicious DNS traffic was generated in a controlled LAN network with virtual machines. More precisely, each botnet sample was run on a virtual machine (where each machine had 2-core virtual processors, each with 4 GB of virtual memory) connected directly to the Internet and recording only DNS traffic while denying all other types of traffic. Four datasets of domains were thus obtained, which resembled (and in some cases were equivalent to) the domains generated by actual DGA-compromised hosts. The four datasets comprised 9280 (Kraken), 12,450 (Conficker), 4205 (Cycbot) and 7624 (Murofet) distinct domains, respectively. In real-world network environments subject to DGA-based botnet attacks, the bot-compromised hosts are owned and operated by a normal user, and thus continue to exhibit a normal query behavior in addition to a malicious botnet behavior. To emulate this situation, the malicious traffic generated by the virtual machines was added to that of a

Table 1 – Detection results obtained for different botnet patterns.

Accuracy	Botnet pattern			
	Kraken	Conficker	Cycbot	Murofet
TP_{rate} (%)	99.30	99.8	98.52	99.51
FP_{rate} (%)	0.16	0.18	0.23	0.39
ACC (%)	99.79	99.75	99.65	99.60
FDR (%)	1.43	1.67	2.06	3.41

certain fraction of benign hosts (the 100 hosts selected from the Education Network). The selected hosts were treated as bot-compromised hosts rather than new hosts in the dataset. Thus, the emulated network consisted of a total of 100 hosts, where the number of benign and bot-compromised hosts varied in accordance with the fraction of hosts to which malicious traffic was added.

5.2.2. *Detection evaluation results given fixed fraction parameter*

Table 1 presents the detection results obtained for the four DGA-based botnet patterns. Note that the number of bot-compromised hosts is equal to 10 in every case (i.e., the fraction of hosts assumed to be bot-compromised is set to 10% for each botnet pattern). It is seen that DBod achieves a minimum true positive rate (TP_{rate}) of 98.52%. In other words, DBod provides an effective means of identifying bot-compromised hosts in a typical network containing both normal and malicious users. Notably, DBod detects compromised hosts based on the query behaviors of the clustered groups rather than their entropy. This strategy significantly improves the robustness of the detection process in the face of different malware patterns. For example, as shown in Table 1, DBod achieves an accuracy of more than 99.6% over the four considered DGA patterns.

5.2.3. *Detection evaluation results given adjustable parameters*

The detection performance of DBod is constrained by the outcome of the CW clustering process. For example, the maximum possible detection accuracy is limited to 80% if the clustering algorithm achieves only an 80% clustering accuracy. It is thus necessary to investigate the effectiveness of the DBod Clustering module in separating the bot-compromised hosts from the normal hosts during the CW clustering process. In this study, the performance of the Clustering and Group Identification modules was evaluated using two adjustable parameters, namely i) the volume of NXDomains generated per hour, and ii) the fraction of bot-compromised hosts in the network. In analyzing the performance of the Clustering module, the malicious traffic was generated using the Conficker botnet pattern. However, the domain name generation mechanism of Conficker was modified so as to permit the number of NXDomains generated by each bot to be varied as required. Fig. 3(a) shows the results obtained for the clustering accuracy of the Clustering module as a function of the number of NXDomains generated by each bot-compromised host. (Note that the fraction of bot-compromised hosts was set equal to 10% in every case.) It is seen that the clustering accuracy increases to almost 100% as the number of NXDomains increases. This result is intuitive since the query behaviors of the bots become more similar with an increasing number of NXDomains. In other words, the correlation matrix R approaches a value of 1 as the number of NXDomains increases, and hence all of the bots tend to be clustered into the same group. Fig. 3(b) shows the variation of the clustering accuracy with the fraction of bot-compromised hosts in the network. (Note that the number of NXDomains generated by each compromised host is set equal to 10 in every case. This volume is relatively small compared to that typically generated under the Conficker pattern, i.e., Conficker-A generates 250 candidate C&C domains each time (Porras, 2009; Porras et al., 2009).) It is seen that the clustering accuracy is reduced as the fraction of bot-compromised hosts increases. This finding is again intuitive since the CW clustering algorithm assigns nodes to a particular group in accordance with the distributions of their neighbors. Thus, as the number of bot-compromised hosts increases, the likelihood of a normal host being misclassified as a botnet also increases.

To analyze the effectiveness of the Group Identification module, Fig. 4(a) shows the false alarm rate of the DBod system as the number of NXDomains generated by the bot-compromised hosts increases. It is seen that both the false positive rate and the false negative rate approach 0 as the number of NXDomains increases beyond 5. In other words, the detection accuracy of DBod increases as the volume of malicious activity increases. Fig. 4(b) shows that the bot-compromised hosts in the network

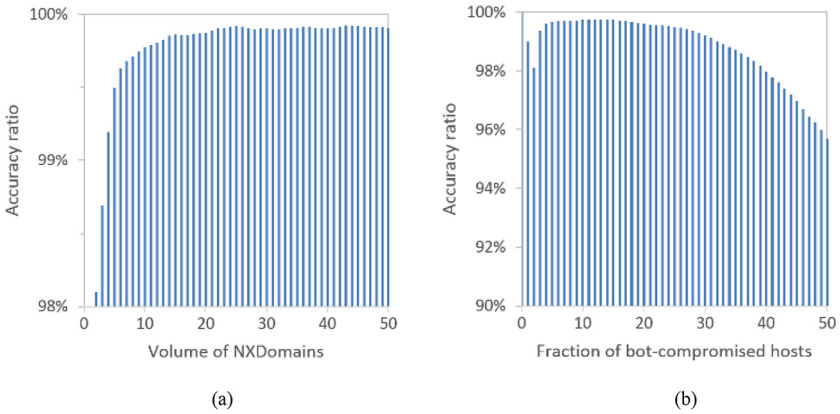

Fig. 3 – Variation of clustering accuracy as function of: (a) volume of NXDomains; and (b) fraction of bot-compromised hosts.

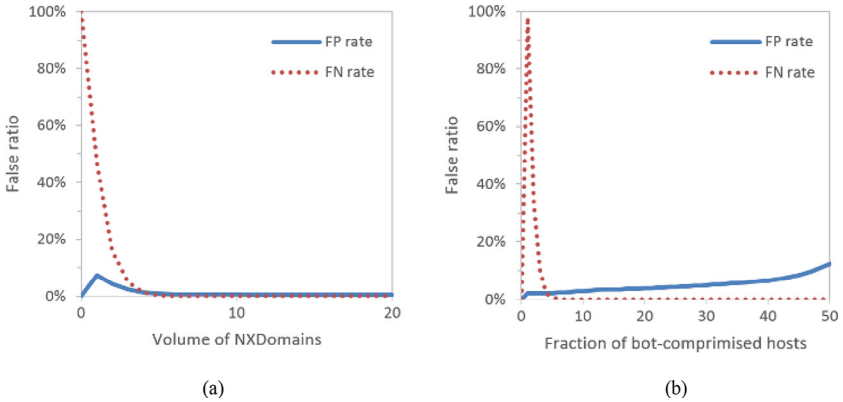

Fig. 4 – Variation of false alarm ratio as function of: (a) volume of NXDomains; and (b) fraction of bot-compromised hosts.

are more reliably detected as the number of compromised hosts increases. However, Fig. 4(b) also shows that the clustering accuracy is reduced with an increasing number of malicious hosts (i.e., the FP_{rate} increases). This finding is reasonable since, as discussed above, normal hosts are assigned with an increasing probability to a malicious cluster as the number of compromised hosts increases.

In general, the results presented in Figs. 3 and 4 show that the volume of malicious queries generated by each compromised host and the proportion of compromised hosts within the network both play a critical role in determining the detection performance of the DBod scheme. However, overall, the results confirm that DBod detects DGA-based botnets with only a low false alarm rate.

5.3. *Performance evaluation using real-world DNS traffic*

This section describes the detection performance of DBod when applied to the DNS traffic data obtained from the Tainan City Education Network described in Section 5.1.

5.3.1. *Pre-analysis on ground truth*

Fig. 5(a) shows the cumulative distribution function (CDF) for the number of hosts querying an NXDomain each day. The results show that the majority of NXDomains were queried by less than 50 hosts per day. Fig. 5(b) shows the CDF for the volume of NXDomain queries per hour (solid line) and per day (dotted line). As shown, the vast majority (~99%) of the NXDomains were queried less than 600 times per hour.

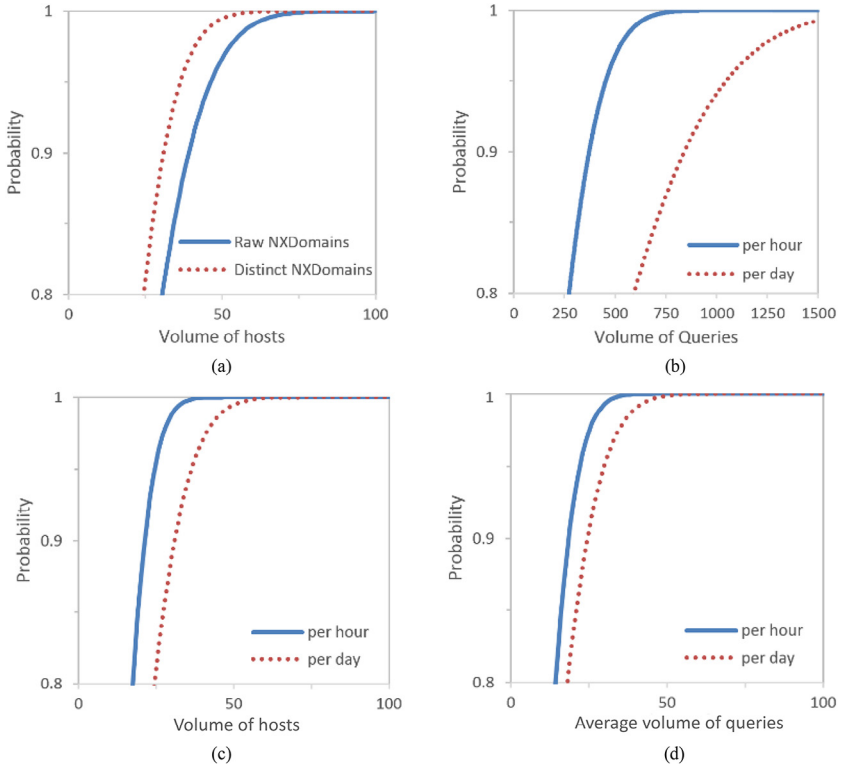

Fig. 5 – CDF for various DNS parameters.

However, around 20% of the NXDomains were queried more than 600 times per day. In other words, around 20% of the NXDomains were queried repeatedly over a period of multiple hours. Fig. 5(c) shows the CDF for the number of hosts querying an NXDomain per hour (solid line) and per day (dotted line). Notably, the gap between the two lines represents the number of hosts which repeatedly query the same NXDomains. According to the characteristic properties of DGA-based botnets described in Section 1, a smaller gap infers a greater likelihood that the monitored network contains DGA-based bots. Fig. 5(d) shows the CDF for the average number of queries generated by those hosts which query an NXDomain. Again, the gap between the two lines indicates the extent of the repetitive queries on the NXDomains. In other words, a smaller gap implies a greater likelihood of DGA-based botnet activity in the network. By contrast, a larger gap implies that some of the NXDomains are queried frequently by hosts in multiple hours, which suggests the occurrence of a temporary shutdown of a high-volume website (e.g., Facebook) or the presence of some other form of botnet activity. However, we focus on defending attack of DGA-based botnet here and left the issue for further research.

The Education Network dataset was further analyzed by calculating the correlation coefficients between the query volume, the number of NXDomains and the number of active domains. The correlation coefficient between the query volume of the active domains per hour and the number of distinct active domains was found to be 0.931. In other words, the query volume of the active domains was correlated almost perfectly with the number of active domains. This finding is consistent with conventional wisdom that, for normal users, the query volume scales proportionally with the number of queried domains. However, the correlation coefficient between the query volume of the NXDomains and the number of distinct NXDomains was found to have a value of just 0.384. In other words, the two variables are not strongly related to one another, which suggests the existence of non-human generated NXDomains in the monitored network. Intuitively, it is reasonable to expect the volume of NXDomains to increase with an increasing number of active domains. However, the correlation coefficient between the volume of NXDomains and the number of active domains was found to be –0.015. In other words, the volume of NXDomains and the number of active domains are not in fact directly related.

Overall, the correlation coefficient results suggest that for the considered environment, most of the NXDomains may be the result of either illegal activity or legal activity, i.e., the NXDomains cannot be definitively assigned to one type of activity or the other. As a result, further analysis is required to determine which of the NXDomains are actually candidate C&C domains generated by a DGA algorithm and which are generated by a legitimate user.

5.3.2. *Evaluation of DBod with ground truth*

In evaluating the DNS data obtained from the Education Network environment, the time window T was set as 1 hour. In other words, the detection process was performed using the DNS data collected each hour of every day over the 26-month observation period. Executing the clustering scheme described in Section 4.3, we found that the NXDomain traffic obtained in each time window comprise 26.7 clusters on average. The clustering result was roughly the same for all of the detection periods in each day (i.e., daytime, night or early morning). Thus, the inference above that most of the NXDomains were created by both legal and illegal applications was further confirmed. Having performed the clustering process, the Group Identification module was therefore applied to classify each cluster as either malicious or normal, as described in Section 4.4.

An average of 4.3 malicious clusters was detected per day over the experimental observation period. Among the detection results, one of the most significant findings was that of five Conficker clusters. Notably, even though all five clusters belong to the same DGA malware family, no association was found between them. Thus, it was inferred that the botnet attacks were performed using different versions of the Conficker malware. Among the five clusters, the largest cluster consisted of 31 bot-compromised hosts, while the smallest contained just 2 hosts. Fig. 6 shows a sample of the NXDomains generated by the largest Conficker cluster over a period of 1 h on a typical day. It is observed that the NXDomains consist of a random string and various TLDs (i.e., net, com, cc, cn, ws, info, biz and org).

Observing the query behaviors of the Conficker botnets over the long term, we found that some of the hosts became "dormant" when other bot-compromised hosts belonging to the same cluster attempted to connect to the C&C server. For example, Table 2 shows the query distributions of the NXDomains generated by the Conficker cluster consisting of five bot-compromised hosts over a 24-h observation period. (Note that due to the Computer-Processed Personal Data Protection Law, the IP addresses in Table 2 are replaced by simple letters (A ~ E) for designation purposes.) It is seen that in the event that the compromised hosts generate NXDomain queries, the number of queries generated by each host is exactly the same. In addition, it is observed that two of the bot-compromised hosts (i.e., A and B) attempt to connect to the C&C server once every 3 h, while the other hosts (C, D and E) attempt to generate connections less than twice in the considered 24-h period. (Note that a similar tendency was observed over the entire experimental period.) The host behavior shown in Table 2 suggests that the botmaster deliberately maintains most of the bots in a dormant state for most of the time in order to minimize the likelihood of their detection. Nonetheless, the results presented in Table 2 confirm that DBod has

hdphedb.info	kgtgucfb.info	ufoxnqxlp.org	bqanjba.cc	jtduhxty.biz
wrtgt.ws	qcfqua.cc	tkoutqkl.net	edesuo.cn	awzbubrkel.cc
nvfkwvxvt.ws	dfammobs.cc	ocvctrvrupz.org	ufhkyrs.biz	xrvffxmo.org
hszge.cn	ixmohorpj.net	qlsrfhfexse.com	gxthfdmcoy.ws	lkzerb.biz
gajluj.cn	ohvdtj.ws	mzevydxb.com	ppzuk.biz	siztaozlw.cn
flnsscrbhcy.ws	ggoqllntum.cc	zvqnnaqn.net	yudjixzw.net	lbqfewpkaoc.org
tipmwndl.biz	tgvwx.net	objptmr.biz	yhzyvnlbs.cn	nfehpyti.ws
dpbqcjbu.org	nejmnjmrr.cn	xhtanl.com	jrxqcvzewp.cc	cgqojxjmvz.ws
wuvnxxxadyb.ws	rqoie.info	pgqzdzp.com	nxtixgvo.info	llfnstyslay.net
vkztrund.biz	pctczal.org	mwtepmy.net	ukdmubey.com	nkagaixqp.biz
kzybhtao.com	oxvzm.biz	rdlysfdwbrf.cc	hpqflrewanu.org	
piwgyhhqjt.net	ipygev.org	ombhzylh.info	ukdmubey.com	

Fig. 6 – Sample of NXDomains generated by Conficker cluster of 31 hosts in a typical hourly period of one day.

Table 2 – Query distribution of NXDomains generated by Conficker cluster with 5 bot-compromised hosts in 24-h observation period.

Time	Bot-compromised hosts				
	A	B	C	D	E
12 am–1 am	68	68	0	0	0
1 am–2 am	0	0	0	0	0
2 am–3 am	0	0	0	0	0
3 am–4 am	87	87	0	0	0
4 am–5 am	0	0	0	0	0
5 am–6 am	0	0	0	0	0
6 am–7 am	78	78	0	0	0
7 am–8 am	0	0	0	0	0
8 am–9 am	0	0	0	0	0
9 am–10 am	63	63	63	0	0
10 am–11 am	0	0	0	0	0
11 am–12 pm	0	0	0	0	0
12 pm–1 pm	72	72	0	72	0
1 pm–2 pm	0	0	0	0	0
2 pm–3 pm	0	0	0	0	0
3 pm–4 pm	77	0	0	0	0
4 pm–5 pm	0	0	0	0	0
5 pm–6 pm	0	0	0	0	0
6 pm–7 pm	68	68	68	0	68
7 pm–8 pm	0	0	0	0	0
8 pm–9 pm	0	0	0	0	0
9 pm–10 pm	70	70	0	0	0
10 pm–11 pm	0	0	0	0	0
11 pm–12 am	0	0	0	0	0

the ability to detect DGA-based botnets even when the botmaster adopts such a strategy.

It will be recalled that the Mjuyh botnet (Yadav et al., 2012) uses the domain name "Mjuyh.com" as the SLD and TLD, and takes random alphanumeric characters as the third-level-domain (3LD). In the present experiments, it was found that some of the NXDomains had a similar domain characteristic as that of Mjuyh bots, but varied in their subdomains. For example, as shown in Fig. 7, the NXDomains generated by the Mjuyh-like hosts comprised a random string with various 3LDs, e.g., hqrqbezayfz.liebiao.800fy.com and xmndo.www.lxt998.com. Among the various subdomains listed in Fig. 7, "liebiao.800fy.com" was previously identified in a systematic DDoS attack detected by Nominum in 2014 (Bruce). The present results therefore suggest the possible emergence of a new Mjuyh-like DGA-based botnet behavior, in which the botnets use multiple subdomains when generating a list of candidate domains rather than the fixed string of "Mjuyh.com" used in conventional Mjuyh botnets. To differentiate between the two types of botnet, the new form of Mjuyh-like botnet (which generates candidate C&C domains with the same 3LD, SLD and TLD, but a random fourth-level-domain) is referred to hereafter as the derivative Mjuyh botnet.

In the present study, the derivative Mjuyh botnet behavior was first detected in the data collected during March 2014, and was observed in a cluster consisting of four hosts. The derivative Mjuyh bots generated NXDomains with the forms shown in Fig. 7. A further inspection of Fig. 7 shows that many of the detected domains are either current website domains (e.g., 23us.com, 500sf.com, 521wooolf.com, qiniudn.com, liebiao.com, baidu.com, qiniudn.com, bsrcsc.com, piaopiao.com, wzgjjt.com, jrfgy.com, lyaux.com, and yuerengu.com.cn) or previous domains (e.g., lxt998.com, qilinchuanqi.com, wuyangairsoft.com, zhaobjl.com). This finding suggests that the botmaster attempted to hide the footprints of the malicious C&C connections among those of legitimate domains in order to increase their likelihood of being classified (erroneously) as normal query domains. Moreover, most of the subdomains are registered in China (most are Chinese gaming sites), or based on a Chinese word (e.g., junyu, qiyue, wuyang, and so on).

Analyzing the query behavior of the derivative Mjuyh botnet over a longer term, we deduce two interesting observations. First, the bot-compromised hosts were active every hour (in contrast to the Conficker bots, which entered a dormant state after establishing a connection). Thus, the hosts were more easily detected since they generated NXDomains continuously over the entire day. Second, the query volume showed an obvious change at certain periods of every day. More specifically, the number of NXDomain queries increased markedly between 11 am and 12 pm each day and then reduced significantly between 10 pm and 11 pm (see Fig. 8). As discussed earlier in relation to Fig. 2, DNS traffic tends to follow the daily routine of real-world users. Since for most individuals, the workday tends to start around 8 ~ 10 am, the results presented in Fig. 8 suggest that the time zone of the botmaster may be two or three hours earlier than that of Taiwan (with a

777.521wooolf.com
a978sf1111.qiniudn.com
dlq.523176.com
dlq.jinyu521.com
dw.yefb.com
ini.egkj.com
liebiao.800fy.com
wuyangairsoft.com
www.23us.com
www.500sf.com
www.baidusf999.com
www.bflm.cc
www.bsrcsc.com
www.lxt998.com
www.piaopiao.com
www.qilinchuanqi.com
www.qiyue98.net
www.uc711.com
www.wzgjjt.com
www.yuerengu.com.cn
www.zhaobjl.com
xin.jrfgy.com
xin.lyaux.com

Fig. 7 – Typical subdomains generated by cluster of derivative Mjuyh bots.

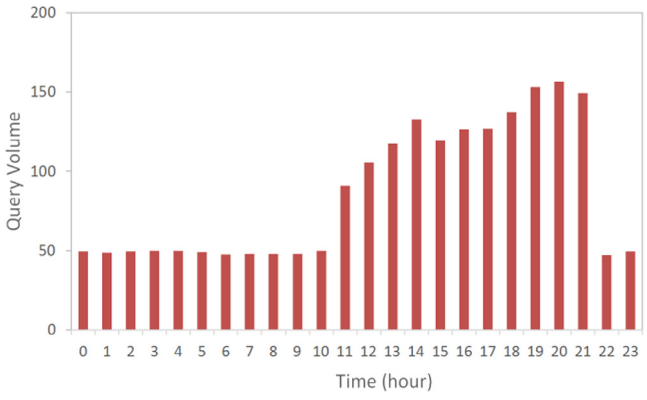

Fig. 8 – Hourly query distribution of number of NXDomains generated by cluster of derivative Mjuyh botnet on average.

UTC offset of +08:00 hours). In other words, assuming that the botmaster wrote the DGA algorithm using the time zone in which he or she resides, the botmaster may be located in a geographical region with a +5 or +6 UTC offset, e.g., Russia, Kazakhstan, Pakistan, India, and so on. Clearly, the time zone may have been set arbitrarily by the botmaster. However, the UTC offset suggested by the detection results nonetheless represents an interesting first line of investigation in searching for the botnet source.

5.3.3. *False reports of DBod*

For the 26-month evaluation period considered in this study, four categories of cluster were falsely reported as botnet groups. The first cluster was triggered by queries to the domain name "imgcdn.ptvcdn.net", associated with a content delivery network (CDN) service provided by Pandora TV in South Korea. The hosts who queried the domain "imgcdn.ptvcdn.net" were triggered to do so by a program called Pandora service, which is initiated automatically whenever users install a multimedia software system known as KMPlayer. If the domain is unavailable at the time of installation, an NXDomain is produced, and hence the users are identified as members of a DGA-based botnet group even though they are in fact legitimate users.

The second cluster comprised NXDomains generated by queries to a market research website with the name "b.scorecardresearch.com" (ScorecardResearch). Users joining this site provide demographic and Internet usage information by web tagging, where this information is sent to the site periodically. During the DBod observation period, the domain went offline for some unknown reason. Consequently, the legitimate requests of users seeking to connect to the domain during this period were falsely interpreted as malicious queries.

The third cluster comprised NXDomains generated by McAfee, a well-known anti-virus application (McAfee secure). McAfee functions by coding the suspect file name, combining it with "avts.mcafee.com" or "avqs.mcafee.com", and then querying the DNS server. The server response includes a reputation value, which provides an indication as to whether the file is most likely malicious or benign. If the file has no previous record in the server, an NXDomain is generated. If multiple users attempt to verify the same file within the same time frame, multiple NXDomains composed of a random string and a fixed subdomain are produced. Consequently, the query behavior resembles that of a DGA-based botnet, and hence a misclassification problem occurs.

The final erroneous cluster was triggered by queries to the domain "denis.stalker.h3q.com", a large-scale (>10 million peers) tracker used to find torrents for peer-to-peer file sharing. The domain was offline during the entire data collection period. Thus, for any BitTorrent application which did not update its tracker list during this period, querying the domain resulted in an NXDomain response despite the legitimacy of the application.

It is worth noting that the NXDomains associated with the four DGA clusters described above contain fixed subdomains. Consequently, the misclassification problem can be avoided by manually adding the affected NXDomains to a whitelist, and then using this whitelist as an input in the initial DBod filtering process.

5.3.4. *Analysis of DGA-based botnet*

Having detected a DGA-based botnet, we found it interesting to investigate the difference in the query behaviors of the normal and bot-compromised hosts. In practice, this analysis relies on a volume analysis. More specifically, for each NXdomain d_u, the correlation between the query volume and the number of hosts (IPs) can be evaluated by the ratio:

$$\log(u) = \log\left(\frac{\sum_{i=1}^{M} Q(i,u)}{\sum_{i=1}^{M} \delta(i,u)}\right), \tag{11}$$

where $\delta(i,u)$ has a value of 1 if host h_i queries NXdomain d_u within the considered time window, and a value of 0 otherwise.

For a DGA-based bot, the correlation between the query volume and the number of IPs is very strong since each domain on the candidate C&C list is usually queried only once by each bot. In other words, the query volume for each NXdomain d_u is directly associated with the number of hosts which query d_u, and thus $\log(u)$ tends toward 0. However, for normal NXDomains, i.e., those NXDomains generated by legitimate applications or by the temporary shutdown of reputable websites, a larger positive correlation value is obtained since many users query the same domain repeatedly. Fig. 9(a) shows the correlation between the query volume and the number of IPs for

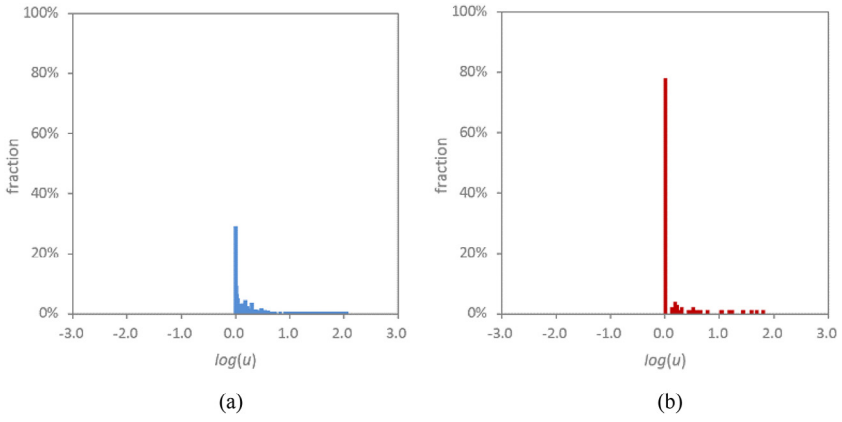

Fig. 9 – Correlation between query volume and number of IPs for: (a) normal traffic and (b) malicious traffic.

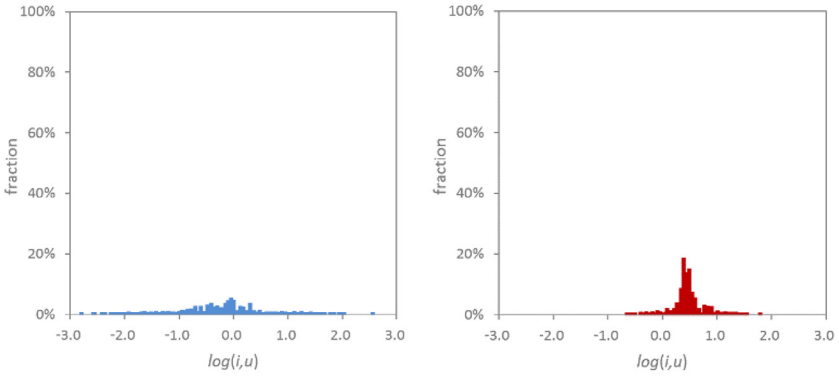

Fig. 10 – Correlation between total query volume of each NXDomain and total query volume of each IP for: (a) normal traffic and (b) malicious traffic.

the case of normal NXDomain traffic. Fig. 9(b) shows the equivalent results for the case of NXDomains generated by hosts identified by DBod as DGA-based bots. As shown, the correlation ratio log(*u*) has a value very close to 0 for most (approximately 80%) of the NXDomains. In general, the results presented in Fig. 9 confirm that the query behavior of the bot-compromised hosts identified by DBod is truly different from that of normal users.

In addition to analyzing the correlation between the query volume and the number of hosts, it is also interesting to examine the correlation between the total query volume of each host querying the NXDomain and the total query volume of the NXDomain. For each host h_i and NXdomain d_u, this correlation can be evaluated as:

$$\log(i,u) = \log\left(\frac{\sum_{u=1}^{N} Q(i,u)}{\sum_{i=1}^{M} Q(i,u)}\right), \tag{12}$$

For example, consider a time window with a duration of 1 hour and assume that two hosts (A and B) query NXDomain "abc.com" 10 and 15 times, respectively. In other words, "abc.com" is queried a total of 25 times within the time window. Assume that hosts A and B also query other NXDomains during the same time window, and have total query volumes of 100 and 20, respectively. Hence, for domain "abc.com", two values of log(*i*,*u*) are obtained, namely log(100/25) = 0.6 for host A and log(20/25) = –0.097 for host B.

As shown in this illustrative example, log(*i*,*u*) has a negative value if the total query volume of NXDomain d_u is greater than the total query volume of host h_i within the time window, and *vice versa*. Fig. 10(a) and (b) shows the correlation results obtained for normal NXDomains and malicious NXDomains, respectively. For the normal NXDomains, as shown in Fig. 10(a), the values of log(*i*,*u*) are aligned with the origin and uniformly distributed between the positive and negative ranges. Moreover, the average value of log(*i*,*u*) is equal to –0.13, i.e., close to 0 (as expected). However, for malicious NXDomains, the total query volume of the NXDomain is often far less than the total query volume of the bots. Hence, as shown in Fig. 10(b), the distribution of log(*i*,*u*) is shifted in the positive direction and is no longer aligned with the origin (i.e., the average value of log(*i*,*u*) is equal to 0.48). Moreover, the standard deviations of log(*i*,*u*) for the normal and malicious NXDomains are 0.718 and 0.169, respectively. In other words, the correlation values for the malicious NXDomains are more closely concentrated than those for the normal NXDomains.

The above analysis provides a novel approach for investigating the difference in the query behaviors of normal and bot-compromised hosts, respectively. Moreover, the difference in the correlation value (i.e., log(*u*) and log(*i*,*u*)) of the normal and bot-compromised hosts provides a useful metric for distinguishing the existence of DGA-based botnets.

5.4. *Limitation of DBod and comparison with existing botnet detection schemes*

In general, the evaluation results presented in the preceding sections confirm the ability of DBod to detect DGA-based botnets in real-world network environments. Notably, DBod can detect botnets even when the botnet behavior is different from that of previously-reported botnets. However, DBod has several practical limitations. For example, as described in Section 3.2, the hosts are clustered in accordance with the similarity of their query behaviors during the detection time window. Thus, DBod is unable to detect bot-compromised hosts if the hosts have never been attacked before or have never attempted to connect to the C&C server. Unfortunately, the analysis results obtained in this study have shown that some bot-compromised hosts remain dormant for hours or even days after connecting to the server. In theory, this problem can be mitigated by extending the observation window. However, this incurs a greater processing cost since the data volume is correspondingly increased. Notably, however, the dormant nodes can be detected by DBod as soon as they wake up and recommence malicious activity. Thus, an observation time frame of 1 h is still deemed to be acceptable for the proposed approach.

In general, it is difficult to conduct a fair comparison among different botnet detection schemes due to the different evaluation environments considered in every case (e.g., the volume of hosts, the execution time, the DGA malware used, and so on). Thus, rather than attempting to perform a detailed quantitative comparison between DBod and other well-known botnet detection schemes, Table 3 presents a qualitative comparison

Table 3 – Comparison of DBod with other well-known botnet detection schemes.

Botnet detection scheme	Characteristics						
	DGA-based Botnet detection	Non-existing Domains queries	Robust to encrypted communication	No need for *a priori* knowledge	Robust to unknown DGA	Low false alarm rate	On-line Detection
Rishi (Goebel and Holz, 2007)	No	No	No	No	No	No	Yes
BotHunter (Gu et al., 2007)	No	No	No	No	No	Yes	Yes
BotSniffer (Gu et al., 2008)	No	No	No	Yes	No	Yes	Yes
BotMiner (Gu et al., 2008)	Yes	No	No	Yes	Yes	Yes	No
Pleiades (Antonakakis et al., 2012)	Yes	Yes	Yes	No	Yes	Yes	No
EXPOSURE (Bilge et al., 2011)	Yes	No	Yes	No	No	No	No
Wurzinger et al., 2009	No	No	No	No	No	Yes	Yes
Stalmans and Irwin, 2011	Yes	No	Yes	No	No	No	Yes
BotGAD (Choi et al., 2009)	Yes	No	Yes	No	Yes	No	Yes
Yadav et al., 2010	Yes	No	Yes	No	Yes	No	Yes
Zhou et al., 2013	Yes	Yes	Yes	No	No	Yes	Yes
Phoenix (Schiavoni et al., 2014)	Yes	No	Yes	No	Yes	Yes	Yes
DFBotKiller (Sharifnya and Abadi, 2015)	Yes	Yes	Yes	Yes	Yes	No	Yes
DBod	Yes	Yes	Yes	Yes	Yes	Yes	Yes

of the various schemes. (Note that the existing schemes presented in Table 3 are previously reviewed in Section 2.) Notably, DBod has the ability to detect DGA-botnets even when the C&C communications are encrypted by the botmaster. Furthermore, DBod distinguishes botnet clusters from normal clusters using a supervised statistical algorithm based on the difference in behavioral characteristics of the two types of cluster. Accordingly, DBod can detect unknown DGA-based botnets without *a priori* knowledge of the botnet behavior or a botnet sample for training purposes. Finally, the DBod clustering algorithm (Chinese Whispers) is low cost, and therefore facilitates the identification of bot-infected hosts in real-time.

6. Conclusion

New generation botnets commonly use some form of Domain Generation Algorithm (DGA) to improve their survivability and locate the C&C server. Accordingly, this study has proposed a defense mechanism designated as DBod for detecting DGA-based botnets in real-world networks. In the proposed scheme, the hosts are grouped into clusters in accordance with the relationship intensity between them, and each cluster is then identified as either malicious or benign depending on its query time distribution and query count distribution characteristics.

The performance of the proposed DBod scheme has been evaluated using both emulated DNS traffic and the traffic obtained over a 26-month collection period from a real-world network consisting of more than 10,000 users. The results have confirmed the ability of DBod to detect DGA-based botnets in realistic network environments. Notably, a new Mjuyh-like botnet behavior has been identified, in which the botnet uses multiple subdomains when generating a list of candidate domains. In other words, DBod provides the ability to defend not only against existing DGA-based botnet patterns, but also against new and emerging botnet patterns.

Acknowledgments

This work was supported in part by Taiwan Information Security Center (TWISC), Academia Sinica, under Grant No. MOST 103-2221-E-006-147-MY3, and Ministry of Science and Technology, R.O.C., under Grant No. MOST 104-2218-E-001-002.

REFERENCES

Abu Rajab M, Zarfoss J, Monrose F, Terzis A. A multifaceted approach to understanding the botnet phenomenon, in Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, 2006, pp. 41–52.

Antonakakis M, Perdisci R, Nadji Y, Vasiloglou Ii N, Abu-Nimeh S, Lee W, et al., From throw-away traffic to bots: detecting the rise of DGA-based malware, in USENIX Security Symposium, 2012, pp. 491–506.

Barracuda Reputation Block List, [Online]. <http://www.barracudacentral.org/>.

Biemann C. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems, in Proceedings of the first workshop on graph based methods for natural language processing, 2006, pp. 73–80.

Bilge L, Kirda E, Kruegel C, Balduzzi M. EXPOSURE: finding malicious domains using passive DNS analysis, in NDSS, 2011.

Bruce VN. DNS-Based DDoS: Diverse Options for Attackers. [Online]. <http://www.circleid.com/posts/20150415_dns_based_ddos_diverse_options_for_attackers/> [accessed 15.04.15].

Choi H, Lee H, Kim H. BotGAD: detecting botnets by capturing group activities in network traffic, in Proceedings of the Fourth International ICST Conference on communication System software and middleware, 2009, p. 2.

Feily M, Shahrestani A, Ramadass S. A survey of botnet and botnet detection, in 2009 Third International Conference on Emerging Security Information, Systems and Technologies, 2009, pp. 268–273.

Goebel J, Holz T. Rishi: Identify bot contaminated hosts by IRC nickname evaluation, in Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, 2007.
Gu G, Porras PA, Yegneswaran V, Fong MW, Lee W. Bothunter: Detecting malware infection through IDS-driven dialog correlation, in Usenix Security, 2007, pp. 1–16.
Gu G, Zhang J, Lee W. BotSniffer: Detecting botnet command and control channels in network traffic, 2008.
Gu G, Perdisci R, Zhang J, Lee W. BotMiner: clustering analysis of network traffic for protocol-and structure-independent botnet detection, in USENIX Security Symposium, 2008, pp. 139–154.
Kugisaki Y, Kasahara Y, Hori Y, Sakurai K. Bot detection based on traffic analysis, in The 2007 International Conference on Intelligent Pervasive Computing, IPC, 2007, pp. 303–306.
Lee J, Kwon J, Shin H-J, Lee H. Tracking multiple C&C botnets by analyzing DNS traffic, in 2010 6th IEEE Workshop on Secure Network Protocols (NPSec), 2010, pp. 67–72.
Liu J, Xiao Y, Ghaboosi K, Deng H, Zhang J. Botnet: classification, attacks, detection, tracing, and preventive measures. EURASIP J Wirel Commun Netw 2009;2009:1184–7.
Lu W, Ghorbani A. Botnets detection based on IRC-Community, in Global Telecommunications Conference (GLOBECOM 2008), pp. 1–5, 2008.
Macdonald D, Manky D. Zeus: God of DIY botnets, FortiGuard Center Threat Research and Response, 2009.
McAfee secure, [Online]. <http://www.mcafee.com/>.
Mockapetris PV. Domain names – concepts and facilities, 1987.
Mowbray M, Hagen J. Finding domain-generation algorithms by looking at length distribution, in 2014 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pp. 395–400, 2014.
Open Malware – Community Malicious Code Research and Analysis, [Online]. <http://www.offensivecomputing.net/>.
Porras P. Inside risks reflections on Conficker. Commun ACM 2009;52:23–4.
Porras P, Saidi H, Yegneswaran V. An analysis of Conficker's logic and rendezvous points, Computer Science Laboratory, SRI International, Tech. Rep, 2009.
Salusky W, Danford R. Know your enemy: fast-flux service networks, The Honeynet Project, pp. 1–24, 2007.
Schiavoni S, Maggi F, Cavallaro L, Zanero S. Phoenix: DGA-based botnet tracking and intelligence, in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, 2014, pp. 192–211.
ScorecardResearch, [Online]. <https://www.scorecardresearch.com/>.
Sharifnya R, Abadi M. A novel reputation system to detect DGA-based botnets, in 2013 3th International eConference on Computer and Knowledge Engineering (ICCKE), 2013, pp. 417–423.
Sharifnya R, Abadi M. DFBotKiller: domain-flux botnet detection based on the history of group activities and failures in DNS traffic. Digit Invest 2015;12:15–26.
Shevchenko S. Domain name generator for murofet. [Online]. <http://blog.threatexpert.com/2010/10/domain-name-generator-for-murofet.html>; 2010.
Silva SS, Silva RM, Pinto RC, Salles RM. Botnets: a survey. Comput Netw 2013;57(2):378–403.
SpamCop, [Online]. <https://www.spamcop.net/>.
Stalmans E, Irwin B. A framework for DNS based detection and mitigation of malware infections on a network, in Information Security South Africa (ISSA), 2011, pp. 1–8.
Stewart J. Inside the storm: Protocols and encryption of the storm botnet, Black Hat, USA, 2008.
Stone-Gross B, Cova M, Cavallaro L, Gilbert B, Szydlowski M, Kemmerer R, et al., Your botnet is my botnet: analysis of a botnet takeover, in Proceedings of the 16th ACM conference on Computer and communications security, 2009, pp. 635–647.
The Abusive Hosts Blocking List, [Online]. <http://www.ahbl.org/node>.
The Spamhaus Project, [Online]. <http://www.spamhaus.org/>.
Villamarín-Salomón R, Brustoloni JC. Bayesian bot detection based on DNS traffic similarity, in Proceedings of the 2009 ACM symposium on Applied Computing, 2009, pp. 2035–2041.
Virus Share, [Online]. <https://virusshare.com/>.
Virus Total, [Online]. <https://www.virustotal.com/>.
Williams J. What we know (and learned) from the Waledac takedown. [Online]. <http://blogs.technet.com/b/mmpc/archive/2010/03/15/what-we-know-and-learned-from-the-waledac-takedown.aspx>; 2010 [accessed 15.03.10].
Wurzinger P, Bilge L, Holz T, Goebel J, Kruegel C, Kirda E. Automatically generating models for botnet detection. In: Backes M, Ning P, editors. Computer security – ESORICS 2009, vol. 5789. Lecture Notes in Computer Science. Berlin/Heidelberg: Springer; 2009. p. 232–49.
Yadav S, Reddy ALN. Winning with DNS failures: strategies for faster botnet detection. In: Security and privacy in communication networks. Springer; 2012. p. 446–59.
Yadav S, Reddy AKK, Reddy ALN, Ranjan S. Detecting algorithmically generated malicious domain names, in Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, 2010, pp. 48–61.
Yadav S, Reddy AKK, Ranjan S. Detecting algorithmically generated domain-flux attacks with DNS traffic analysis. IEEE/ACM Trans Netw 2012;20:1663–77.
Zeidanloo HR, Manaf AA. Botnet command and control mechanisms, ICCEE'09. Second International Conference on Computer and Electrical Engineering, 2009, pp. 564–568.
ZeuS Gets More Sophisticated Using P2P Techniques. [Online]. <http://www.abuse.ch/?p=3499>; 2011 [accessed 10.10.11].
Zhou Y-L, Li Q-S, Miao Q, Yim K. DGA-based botnet detection using DNS traffic. J Internet Serv Inf Secur 2013;3:116–23.

Tzy-Shiang Wang received his B.S. degree in Department of Applied Mathematics from National Dong-Hua University, Taiwan, in 2005, and his M.S. degree in Department of Information Management in Kun Shan University, Taiwan, in 2007. He is currently working toward a Ph.D. degree in the Institute of Computer and Communication Engineering at National Cheng-Kung University. His current research interests include clustering algorithms, social network and network security.

Hui-Tang Lin received B.S. degree in Control Engineering from National Chiao Tung University, Taiwan, in 1989, the M.S. and the Ph.D. degrees both in Electrical Engineering from Michigan State University, East Lansing, MI, in 1992 and 1998, respectively.

He is currently a full professor at the Department of Electrical Engineering and the Institute of Computer and Communication Engineering of National Cheng-Kung University, Taiwan. His research interests include QoS of high-speed networks, wireless networks, optical networks, social networks and network security. He is a member of IEEE.

Wei-Tsung Cheng received his M.S. degree from National Cheng-Kung University, Tainan, Taiwan, in 2013. His current research interest includes network security.

Chang-Yu Chen received his M.S. degree from National Cheng-Kung University, Tainan, Taiwan, in 2014. His current research interests include the area of clustering algorithms and network security.