Computer Communications 49 (2014) 33–47

Contents lists available at ScienceDirect

Computer Communications

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

GMAD: Graph-based Malware Activity Detection by DNS traffic analysis

Jehyun Lee, Heejo Lee *

Division of Computer and Communication Engineering, Korea University, Seoul 136-713, Republic of Korea

ARTICLE INFO

Article history:
Received 4 August 2013
Received in revised form 18 February 2014
Accepted 28 April 2014
Available online 9 May 2014

Keywords:
Malware domain name
DNS
Botnet
Sequential correlation
Graph clustering

ABSTRACT

Malicious activities on the Internet are one of the most dangerous threats to Internet users and organizations. Malicious software controlled remotely is addressed as one of the most critical methods for executing the malicious activities. Since blocking domain names for command and control (C&C) of the malwares by analyzing their Domain Name System (DNS) activities has been the most effective and practical countermeasure, attackers attempt to hide their malwares by adopting several evasion techniques, such as client sub-grouping and domain flux on DNS activities. A common feature of the recently developed evasion techniques is the utilization of multiple domain names for render malware DNS activities temporally and spatially more complex. In contrast to analyzing the DNS activities for a single domain name, detecting the malicious DNS activities for multiple domain names is not a simple task. The DNS activities of malware that uses multiple domain names, termed multi-domain malware, are sparser and less synchronized with respect to space and time.

In this paper, we introduce a malware activity detection mechanism, ***GMAD***: *Graph-based Malware Activity Detection* that utilizes a sequence of DNS queries in order to achieve robustness against evasion techniques. *GMAD* uses a graph termed *Domain Name Travel Graph* which expresses DNS query sequences to detect infected clients and malicious domain names. In addition to detecting malware C&C domain names, *GMAD* detects malicious DNS activities such as blacklist checking and fake DNS querying. To detect malicious domain names utilized to malware activities, *GMAD* applies domain name clustering using the graph structure and determines malicious clusters by referring to public blacklists. Through experiments with four sets of DNS traffic captured in two ISP networks in the U.S. and South Korea, we show that *GMAD* detected thousands of malicious domain names that had neither been blacklisted nor detected through group activity of DNS clients. In a detection accuracy evaluation, *GMAD* showed an accuracy rate higher than 99% on average, with a higher than 90% precision and lower than 0.5% false positive rate. It is shown that the proposed method is effective for detecting multi-domain malware activities irrespective of evasion techniques.

1. Introduction

Malicious software, namely, malware, is one of the most significant sources of Internet intrusion. Uncountable malicious activities caused by malwares, including information theft, DDoS attack, and spamming, are the most critical problems in our information life. According to the report of an AV vendor, Kaspersky, over 1.5 billion attacks by malwares were observed in 2012, and 6.5 million unique malicious domain names were used for these attacks [1]. In comparison with 2011, the increment of malicious domain names in 2012 was 2.5 million. Another estimation result for 2013 reports that thousands of new malware web-sites have appeared each day during the last several years [2].

From the efforts of anti-virus vendors and researchers to combat malwares, one of the most effective ways to reduce malware activity is to block network access to a remote control server, generally called a command and control (C&C) server. This point of view, of course, includes the malwares that are not remotely controlled but send stolen information to their remote server, as well as botnet, which are controlled through the C&C servers. In several security reports [3,4], taking down malwares by blocking the C&C servers has been shown to achieve an immediate reduction in malicious activities.

Among the several approaches for detecting malicious remote servers, Domain Name System (DNS) traffic monitoring has been employed in many previous studies [5–12] because of its efficiency and effectiveness. The DNS is a centralized network point that is essential for using Internet services including the malicious network activities. As malwares use domain names instead of

* Corresponding author.
E-mail addresses: arondit@korea.ac.kr (J. Lee), heejo@korea.ac.kr (H. Lee).

http://dx.doi.org/10.1016/j.comcom.2014.04.013
0140-3664/

static IP addresses to access their remote servers, DNS traffic monitoring is indispensable for detecting malicious servers.

Despite the many studies on malware and malicious domain name detection, recent intelligent malwares can survive even after their C&C domain names are blocked. The main reason for this malware survivability is that the malwares have numbers of alternative C&C domain names. Traditional single domain malwares have suffered from *one-point failure problem*. However, the recent malwares [13–16] overcome the problem by using redundant domain names for continuing to update the malware binaries when their C&C domain names are detected and blocked. Furthermore, recent sophisticated malwares not only replace their C&C domain names after blocking occurs, but also use various evasion techniques such as sub-grouping, fake-query and one-time domain, through the multiple domain names. In this paper, we call the malware that utilizes multiple domain names for malicious activities a multi-domain malware. In order to take down multi-domain malwares effectively, it is important to detect the domain names in their entirety against the evasion. Unfortunately, the methods proposed in previous studies considered a single domain name and limited patterns of multi-domain such as random domain names, so that they are not robust enough to combat evasion techniques that use multi-domains.

In this paper, we propose a malware activity detection mechanism, ***GMAD***: *Graph-based Malware Activity Detection*, which uses the sequential correlation between domain names. *GMAD* detects malicious domain names used for malicious activities. Sequential correlation is a spatial property among domain names, caused by the query patterns of DNS clients. We express the sequential correlation of domain names on a graph structure called ***DNTG***: *Domain Name Travel Graph*, which was proposed in our preliminary work [17]. Using the graph expression, the proposed mechanism can find suspicious domain names that have a high sequential correlation with a known malicious domain name. In this way, our mechanism detects malware domain names that have not yet been detected and are not included in domain black lists.

Sequential correlation, which is the DNS behavior property used in *GMAD*, is a spatial property that is robust against the malwares that evade legacy detection methods. Malware activity detection mechanisms can be classified into two categories according to the kind of properties of malware behaviors: temporal properties and spatial properties. Temporal properties, such as DNS query timing synchronicity, have a weakness when combatting the evasion techniques mentioned above. On the other hand, compared with other approaches that use a spatial property, sequential correlation is much less influenced by the number of queries, infected hosts, and activities of the legitimate users. This advantage allows the malicious domains that are queried at a low rate by less infected hosts that have not been covered by legacy detection methods to be detected. Several evasion techniques, such as sub-grouping [16] and one-time domain using domain generation [13–15], can be used for bypassing legacy detection methods through a low query rate during the monitoring time and few on-line hosts. Consequently, our mechanism has the advantage that it can take down malwares more effectively by detecting the malware domains that, due to temporal and spatial evasion techniques, are not covered by legacy malware domain detection approaches.

In our experiments with real network DNS traffic, *GMAD* shows its ability to detect thousands of malware domain names. The DNS traffic is gathered from DNS servers of ISPs in Korea and the U.S. In detection accuracy evaluation, *GMAD* shows an accuracy rate higher than 99% on average with a precision rate higher than over 90% and a false positive rate lower than 0.5%. In terms of detection sensitivity, *GMAD* detects twenty-eight times more malicious domain names than one of the most sophisticated malicious domain detection methods, *BotGAD* [8,9], in experiments using the same data set. The source of *GMAD*'s superior performance is in that it detects sparse and low-rate malware activities that have not been detected by the previous approaches.

Through *GMAD*, we achieve a sensitive detection that has not been achieved in previous studies. Against the temporal and spatial evasion techniques of the recent intelligent malwares, *GMAD* accurately detects malicious domain names that are used at a low rate and by a few infected machines. Its detection sensitivity allows of *GMAD* to respond effectively to rapidly increasing numbers of malicious domain names and malwares. In terms of efficiency, *GMAD* achieves scalable malware detection in huge network environments. Through our experiments with real-world ISP level DNS data, we evaluated that *GMAD* successfully detects malwares in the wild that use various DNS query patterns and evasion techniques and efficiently works in large and complex network environments.

The rest of this paper is as follows. In Section 2, we review the previous efforts for combating malwares by detecting their inherent features of network behaviors, and we point out advantage of our proposal. Before the explanation for *GMAD* mechanism, we introduce our behavior property, i.e., sequential correlation, and how sequential correlation is used for solving several faced problems in comparison with previously proposed behavior properties, in Section 3. We explain detailed processes of *GMAD* with algorithms in Section 4. In Section 5, we evaluate the performance of *GMAD* in terms of accuracy and sensitivity using real world DNS traffic. After that, we analyze the effects of heuristic clustering metrics and data dependent features in experiment results for the best configuration, and scalability of *GMAD* in Section 6. In Section 7, we discuss several issues we should consider. Finally, we conclude this paper in Section 8.

2. Related work

The DNS has been considered a monitoring place to detect malware activity. As compared with other approaches, DNS monitoring has advantages when faced with encrypted protocols and change of traffic behavior. Previous work can be classified into two categories; DNS monitoring approaches for malware detection and graph-based approaches for malware and malicious domain name/URL detection.

DNS monitoring approaches, such as *BotGAD* [8], *Pleiades* [11], *Bayesian DNS traffic similarity based detection* [18], *BotSniffer* [19], and the black list extension mechanism using DNS queries [10], have been studied in several ways. DNS-based approaches share the advantages of robustness against encrypted communications and the efficiency of centralized detection. Choi et al. [8,9] distinguished group activities in DNS traffic from legitimate users activities in a study using the concept of the client set of a domain name. They measured the similarity of the DNS clients of each domain name using quantitative likelihood. This approach is countered by recent multi-domain malwares that separate their activities into multiple domain names.

The most recent study of the *BotGAD* [9] and a DGA detection approach of Yadav et al. [12] respond to the multi-domain malware problem by grouping domain names based on the lexical similarity among the domain names and network features, such as corresponding IP addresses. However, these approaches still have limitations when faced with the multi-domain malwares that do not use DGA and have little lexical similarity in their domain names.

Antonakakis et al.'s botnet C&C detection system [11] used NX domain names for detecting DGA domain names. Use of DGA is one of the important features of recent malwares. Their DGA

models are efficient to classify newly generated domain names. However, it could not respond to most of malware DNS activities except DGA-based C&C queries, because the clustering method of *Pleiades* was dependent on lexical and structural features of domain names. Because DNS activities of malwares are not limited on C&C communication, detecting DGA that are used only for C&C domain names has relatively small detection coverage.

Villamarín-Salomón and Brustoloni [18] proposed a Bayesian method for detecting botnet-based DNS traffic. Known botnet, mutations of the botnet, and their fluxed domain names can be detected by the similarity of their DNS traffic by using this method. However, their study results can be affected by traffic noise from background traffic because there is a high possibility of a similarity of DNS traffic in famous domain names. In our study, we applied the sequence of queries to the concept of DNS traffic similarity. If the concept of sequence is applied to previously proposed methods, it may enhance their performance.

Guofei et al. [19]'s work focused on the spatial–temporal correlation and the similarity of botnet activities for detecting C&C servers. The approach was also based on the automated and repeated activities of bots. Analysis of the temporal property of bots is effective for detecting many traditional botnets, but not modern botnets that use avoidance methods, such as fake queries and asynchronous DNS queries. The spatial correlation which they considered is a repeated similar pattern, and it can be evaded by the scattered and randomized query patterns used by recent malware.

Blacklist extension using the co-occurrence relationship between DNS queries was proposed by Ishibashi et al. [10]. The major difference of our method is that a cascading co-occurrence of DNS queries raises more than only at suspicion about the domain names. The infected heavy user problem addressed in this study is reduced, because the co-occurrence of domain names queried with an irregular sequence carries a low weight in our work.

Several studies have applied the graph structure to botnet detection. Shishir et al. [20], attempted to distinguish botnet communications. In their graph approach, they represented communication relationships between bot hosts on a graph. Their approach shares several concepts with this study. However, the type of botnets covered by their method is mainly P2P due to their focus and target data.

Yamada et al.'s study [21] applied a temporal relationship and graph structure to URLs. Our work uses the topological feature of sub graphs, which is hard to express at the degree of a node, as compared to analyzing link features. However, the learning approach that uses link features, which they applied, may be useful for expressing node characteristics and automating our detection mechanism.

Jiang et al. [22] proposed a graph-based suspicious DNS activity detection method from failed DNS queries. Using failed DNS queries is effective to detect abnormal DNS activities, such as domain fluxing and spamming, but it has limitation for detecting malicious domain names which are actually working. In addition, the trinonnegative matrix factorization algorithm that they used for extracting distinct DNS activities needs too large size of matrix to analyze much larger DNS traffic which includes valid domain names efficiently.

John et al. [23] reported a valuable analysis result using their own system. Their approach and concrete analysis of active botnet are applicable, since they provide data basis for proving the properties of botnet, although the observed results may have a limitation in the case of artificially generated malware traffic.

Lastly, our preliminary work [17] proposed a graph structure for detecting multi-C&C botnets by clustering domain names according to the number of clients and DNS query density. It showed an ability to tracking continuously generated botnet domain names which are hard to detect through temporal or spatial similarity. In *GMAD*, we attempts to extend the detection coverage from naive botnet activity to complex malware activity which is more irregular and sparser. *GMAD* enhances the detection coverage by adopting domain clustering with client sharing in addition to the simple graph filtering. As a result, we detected more than ten times of malicious domain names in the same data set, as compared with the preliminary work which detected less than five hundreds of botnet domain names.

Thus, previous studies on detecting malware including botnet are dependent on the temporally and spatially similar DNS activity patterns, but recent intelligent malware processes no longer work identically. A malware activity detection method needs to consider that different activity patterns generated by the same malware need to be detected. Another important limitation is detection sensitivity. The detection ability of statistical approaches using graph analysis or machine learning techniques is limited to distinguishing dense and regular activities from the legitimate activities. The malwares that show similar or even sparser DNS activities with a few infected machines should be considered.

Our mechanism uses the graphs constructed from DNS monitoring to detect malware activities. Because most of large scale malicious activities have been occurred by remotely controlled groups of malwares, i.e., botnet, previous studies for combatting to botnet share the fundamental problems and response approaches with our work.

As compared to the previous studies, our work considers temporally and spatially irregular DNS query patterns generated by evasion techniques and adapts a robust property, sequential correlation. Moreover, our graph structure and clustering process with an increasing threshold provide a wide detection spectrum from low-rate malware activity to C&C servers with hundreds of multi-domain malware.

3. Malware DNS behavior properties and problems

3.1. Proposed DNS behavior property: sequential correlation

Sequential correlation is the correlation between two domain names that are queried after or before each other. The degree of the sequential correlation is determined by the client sharing ratio (*CSR*) between the connected domain names. The client sharing is estimated using the Jaccard similarity of query source IP addresses. The *CSR* formula $CSR(v_i, v_j)$ is defined as Eq. (1), where C_{v_i} is the set of IP addresses querying a domain v_i.

$$CSR(v_i, v_j) = \frac{|C_{v_i} \cap C_{v_j}|}{|C_{v_i} \cup C_{v_j}|} \quad (1)$$

The higher the *CSR* the higher the dependency between two domain names, but the opposite is not always true. In many practical cases observed in our experiments, even if the *CSR* between two domain names are low, the *CSR* among the entire domain members of a malware is observed to be high. For example, accessing *www.google.com* through an Internet browser automatically causes DNS queries to the domain names of the statistics and image server of *google*. Moreover, in many countries other than the U.S., a DNS query to *www.google.com* is redirected to a local domain such as *google.ca* or *google.co.kr*. These DNS query patterns are forced by the system and have high sequential dependency. The sequence of the domain names is hardly changed and *CSR* is high. We categorize the causes of sequential DNS queries into three relationships: server-driven, client-driven, and accidental. The intentional sequential correlation is caused by either the server-driven or client-driven relationship.

- The server-driven relationship constitutes those cases where the subsequent domain names to be queried are determined by the response of the prior query, or the content of the accessed server. For example, imported images give rise to other DNS queries if they are below different domain names. Another important case is the pre-defined domain names for load balancing or contents distribution. These types of sequential query are performed regardless of the user's will and occur systematically with a high probability. Consequently, the sequential relationships among the domain names have statistically high regularity with few exceptions.
- The client-driven relationship constitutes the cases where the sequence and list of querying domain names are determined by the process or by a human on the client side. The programs that try to connect some domain names periodically also cause periodic DNS queries. In many cases, the programs have a static list and the order of querying domain names including the query strategy such as when and how many times they are queried. In this context, it is hard for human users to generate the program-like query pattern and vice versa.

3.2. Previous malware behavior properties

The properties previously proposed for distinguishing malware DNS activities from legitimate DNS activities can be classified into two classes: temporal and spatial. Fig. 1 illustrates the conceptual comparison between previously proposed properties and our property for malicious DNS activity detection. A rectangle on the figure represents a DNS client c_i, and a circle represents a domain name d_i. An arrow from a client to a domain name represents a DNS query for the domain name d_i from the clients at time t_i. The arrows that have the same query time on the first row of the figure, a temporal property approach, represent simultaneously generated DNS queries within a time slot. On the other hand, in the case of one of the most recent approaches using a spatial property, i.e., DNS co-occurrence, the arrows do not have a query time because the approach does not consider the time and order of the DNS queries. Finally, the proposed property, sequential correlation, considers the order and the list of queried domain names, irrespective of the regularity of the time intervals between DNS queries and time synchronicity of each client. These two properties have been used for detecting malwares from their DNS activities. The pros and cons of the properties are as follows.

- The temporal property is an inherent feature of malware due to the centralized, automated, and time synchronized C&C structure characteristics, such as DNS queries at the same time and in a regular time period to a same domain name. The temporal property is effective for distinguishing distinguish human user activities from those of traditional IRC botnets and simple malware sending messages to the C&C server periodically. However, this property is also seen in legitimate background processes such as daemons communicating with their update or remote support servers. Moreover, many sophisticated malwares that do not use IRC as the C&C protocol rarely show the temporal property.
- Conversely, spatial properties are manifested by a set of malwares that share an identical and pre-defined domain list. A malware has a list of domain names for malicious activity and its C&C servers, and the domain names in the list are queried to a DNS server by the same malwares on the Internet. The spatial property is robust against intentionally and unintentionally unsynchronized DNS activities and changes in the member clients of malwares. Like the temporal property, this property is also evident in the legitimate processes that share centralized servers. Thus, several advanced approaches that use spatial properties adopt a knowledge-base such as a domain or IP blacklist, to estimate the credit of domains and clients. One of the recent approaches that use a spatial property, *DNS co-occurrence* [10], achieved black list enhancement by tracking domain names queried from the DNS clients that had queried a known malicious domain name.

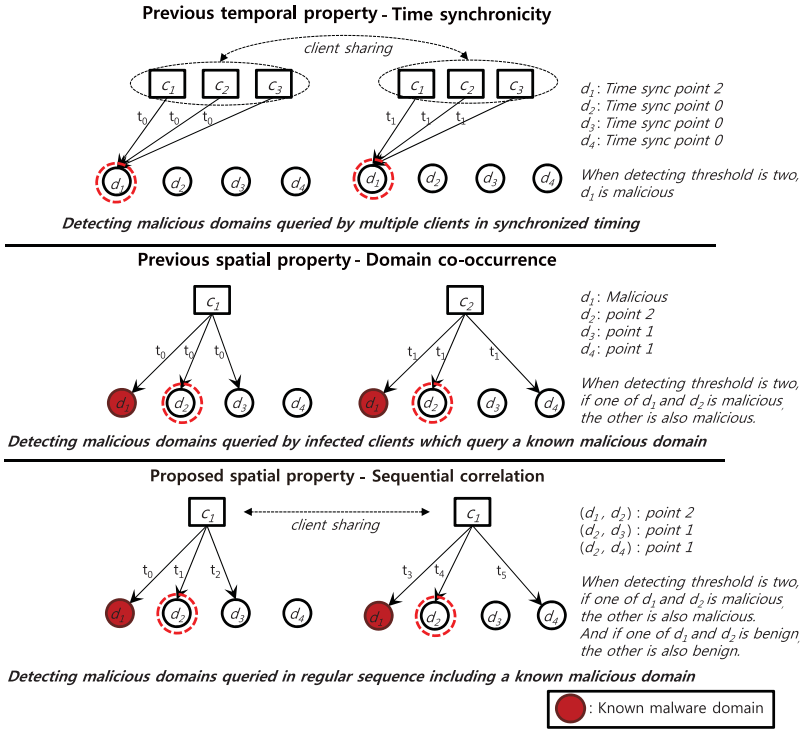

Fig. 1. Sequential correlation in compared with previous malware detection properties.

The most significant advantages of our property, sequential correlation are twofold. The first advantage is the sensitivity of its detection, which is achieved by gathering the scattered and individualized behaviors achieved by evasion techniques. In spite of the various approaches that have been developed, the evasion techniques of recent malwares have succeeded in taking a step forward in the race between detection and avoidance. Recent malwares no longer repeat their DNS behaviors infinitely without hibernation. Moreover, the behaviors of the infected clients are not temporally synchronized. This means that we cannot find vertical (repeated similar patterns of a client) [24] and horizontal (shared similar patterns between clients) [8,19] correlations in the DNS behaviors of infected clients. However, *GMAD* gathers the temporally scattered behaviors of a client and unsynchronized behaviors between clients on a graph structure, *DNTG*.

The second advantage of sequential correlation is the accuracy of its detection, which is achieved by distinguishing the noise DNS behavior toward legitimate domain names. Temporal properties, such as the repeated query patterns of a client [24] and the similarity of the patterns between numbers of clients [8], are also shown for the legitimate domain names and provide few additional evidence about their reputation because their reputation is evaluated only by the behavior of their clients. Spatial properties, such as lexical similarity among domain names and spatial similarity between DNS clients [19], are hardly shared between legitimate and malicious domain names, but they are satisfied only by limited old types of malwares. Methods that use other spatial properties, such as co-occurrence [10], are practical against the evasion techniques by adopting obvious evidence, i.e., domain black lists. However, they are not strict enough to distinguish intentional fake and casually occurred noise legitimate queries from the infected machines. Several sophisticated malwares hide the spatial properties of their DNS activities in noise by using fake-DNS queries generating numbers of legitimate or invalid DNS queries. However, sequential correlation allows us to cluster legitimate domain names to form a legitimate domain group even though the domain names are intentionally queried by infected clients. The sequential correlation of the domain names working together, which are supported by legitimate clients, leads a domain name to belong to the corresponding group of domain names that should be queried together in a legitimate access case.

3.3. Robustness problem against evasion techniques

Given the escalation in the evasion techniques of malwares, a detection method should consider robustness against these evasion techniques. The key idea of known evasion techniques is to hide the group activity of the malware processes on the Internet. Malware authors construct their malwares such that they do not generate the same DNS queries at the same time, in order to impersonate a normal user. However, it is significantly difficult to make numbers of malwares on remote hosts work individually without any shared domain names and common query patterns.

An evasion technique is used for hiding the infected clients and malicious domain names from detection. The evasion techniques against detection mechanisms that use DNS analysis can be classified into two classes, in the same way as the detection approaches, i.e., temporal and spatial evasion. Temporal evasion techniques drive the DNS queries from infected machines to temporally scattered timing. Given that the C&C protocol of malwares has moved from IRC to HTTP and custom protocols, malwares no longer need to be temporally synchronized. Even during malicious activity causing DNS queries, the attackers, generally called bot-masters in much of the research papers, use only a part of their infected machines in order to avoid exposing all the members. Concrete examples of the temporal evasion techniques that have been observed in malware in the wild are "client sub-grouping", "domain sub-grouping", "query timing randomization" and so on. Sub-grouping means dividing the resources, such as infected clients and C&C servers, for malicious activity into several small groups to avoid detection of all the resources and to hide their group activities. Consequently, the DNS activities that adopt a temporal evasion technique shows different client sets at each time slot for a domain name. For instance, malicious domain d_1 d_4 which are queried by infected client c_1 and c_2 in Fig. 2, evade the detection based on DNS group activity by using temporal evasion, querying at each different time slot. Previous detection approaches using temporal properties, such as time synchronized group activity from a static and exclusive client set of a domain name, are easily evaded by these techniques.

Spatial evasion techniques drive the DNS queries from infected machines to complex target domains and query orders. Static or

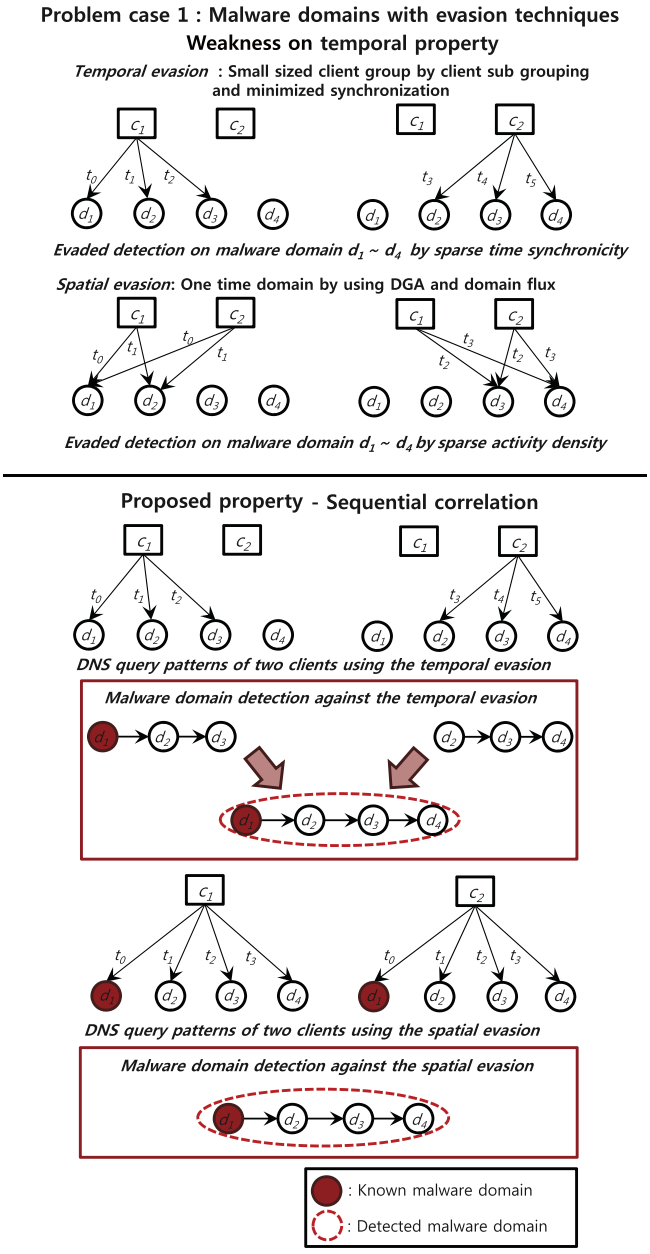

Fig. 2. Problem case 1: temporal and spatial evasion techniques and the countermeasure using sequential correlation.

random order query patterns with multiple C&C domain names are well known spatial evasion techniques. The most sophisticated spatial evasion technique uses domain names that are continuously generated by the domain generation algorithm (DGA). The tremendous numbers of domain names are used for redundant C&C domain names and fake domain names to hide the real C&C domain names in the domain crowds. In terms of DNS analysis, the spatial evasion techniques show us different domain names for each query from an infected client. The spatial evasion technique in the lower part of Fig. 2 using DGA and domain fluxing also evades the detection system based on temporal properties.

In contrast, the sequential correlation on a graph allows us to track the newly generated domain names and represents the connectivity between the old and new domain name. Even if each client makes queries to each different subset of the entire domain set, as in the temporal evasion case shown in Fig. 2, the sequential correlation on a graph organizes an accumulated graph of the entire domain set as in the framed example against the temporal evasion techniques. In more severe cases, against the spatial evasion techniques used by several known malwares, such as *Conficker* [14] and *Torpig* [15], the domain names are distinguished as an independent domain group, because the randomized sequential correlation has exclusiveness with the sequential correlation of the other non-randomized domain names. *GMAD* solves this problem of domain clustering through the DNTG as in the example in the lowest frame of Fig. 2.

3.4. False detection problem for legitimate domain names

Another problem that has been a challenge for the DNS monitoring approaches is distinguishing the DNS queries for legitimate domain names from infected clients. Popular domain names that are frequently queried by the users of infected machines cause false detection and noise in order to confuse the temporal and spatial malware query patterns. Even these DNS queries for legitimate domain names are utilized intentionally. Several well-known malwares that have a large number of infected machines insert legitimate or invalid domain names into their DNS query lists. These fake DNS queries make the process of finding real C&C domain names inefficient and labor-intensive by inserting false detection and garbage domain names into their DNS behaviors.

To resolve the false detection problem, previous studies have used a white list of well-known and popular legitimate domain names collected from statistics services, such as *Alexa* [25], or a popularity metric using the number of querying clients [26]. However, this approach is easily evaded by using legitimate domain names that are not sufficiently popular for being distinguished to the malicious domain names, as in the case illustrated in Fig. 3. In contrast, sequential correlation allows legitimate domain names to be distinguished from malicious domain names. Because legitimate domain names also have a server-driven and client-driven sequential correlation, like d_3 and d_4 in the figure, even if the clients are automated processes, the legitimate domain names are clustered into a domain group separate from the malicious domain group. The exceptional case is when a domain name is queried by only infected clients that query a known malicious domain name, or when a dominant portion of the clients are infected. These domain names would be considered as domain names used by malwares. In contrast, if a legitimate domain name has a client set that is completely different from that of the other domain names in a domain group, it would be separated by our mechanism, even if several infected clients query legitimate domain names. As summarized in Table 1, detection systems that use the temporal and spatial properties among malware processes and the density and size of activities, i.e., the numbers of queries and on-line memberships in DNS, are effective for detecting traditional

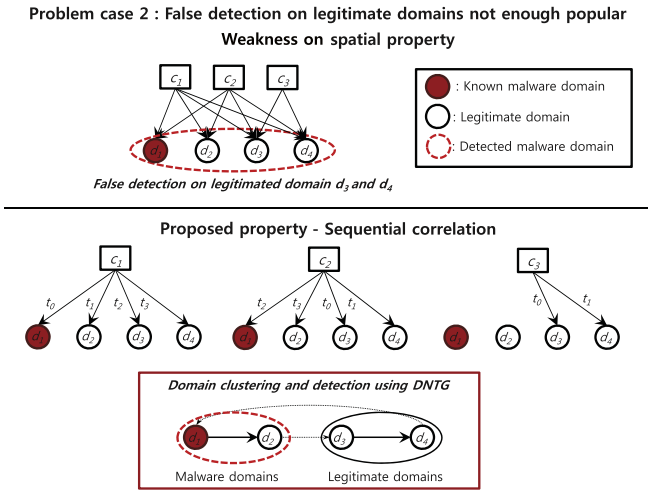

Fig. 3. Problem case 2: legitimate domain queries from infected machines causing false detection on the detection mechanisms using spatial property, and the countermeasure using sequential correlation.

Table 1
Comparison for legacy and proposed malware detection properties on DNS.

DNS behavior property	Traditional malware	Users	Intelligent malware
DNS query volume	High	Low	Various
Temporal property	High	Low	Low
Spatial property	High	Low	Various
Sequential correlation	High	Low	High

malware such as IRC botnets, but have been evaded by the intelligent query techniques of recent malwares. Sequential correlation is sufficiently discriminatory to meet the challenges of malware activity detection.

4. Malware activity detection using DNTG

In this section, we introduce a malware activity detection mechanism called *GMAD*: Graph based Malware Activity Detection. *GMAD* detects malware activity by analyzing DNS traffic using a graph expression named ***DNTG***: *Domain Name Travel Graph*. *DNTG* is a graph that expresses a domain name as a node and the sequential correlation among the domain names as a directed edge, an arc. Sequential correlation is a relationship between two domain names that are continuously queried at a DNS. And a graph structure is suitable to represent the sequential correlation. The sequential correlation has increasing and equivalent entities, i.e., domain names, and their relationship that has direction and loop. A loop is an indispensable property to express repeated DNS activities without obvious start and end node. *DNTG* satisfies these required conditions.

GMAD constructs *DNTG* from DNS traffic and clusters domain names that are queried in a patterned order and by similar client sets. This clustering allows malware domain names to be distinguished from the legitimate domain names that are casually or intentionally queried. Lastly, *GMAD* determines the maliciousness of each domain cluster using a domain blacklist. In this section, we explain in detail the algorithms for graph construction from DNS traffic, graph clustering and malware activity detection starting with brief overview of the mechanism.

4.1. Mechanism overview

GMAD detects malware activities in DNS traffic through three processes: *P1 – graph construction*; *P2 – graph clustering*; and

P3 – malware activity detection, as shown in Fig. 4. As its results, it reports malware domain names that are used not only for malware C&C servers but also for malicious activities, such as malware dropping, update, spamming, and blacklist check. The summarized functionality of each process is as follows.

- *P1 – graph construction*. DNS query traffic captured in front of a DNS server is converted to a *DNTG*. Through *P1*, the mechanism extracts the sequential correlation of domain names as a pre-processing step for graph clustering. *P1* extracts four information from the DNS traffic: (1) A list of queried domain names as the nodes of a graph; (2) sequential correlations as the edges of a graph; (3) a table of query source IP addresses for each domain name; and (4) the number of queries of each domain name. The domain names that are queried only by one client during a monitoring time are removed.
- *P2 – graph clustering*. This process is a key part of the mechanism. *P2* groups intentionally related domain names to detect malware domain names that are working together. *P2* does not make decisions about the maliciousness of each domain name and each domain group, but it separates domain names that are used by different processes and services. According to a clustering feature, client sharing, *P2* clusters the domain names queried by the same process or service.
- *P3 – malware activity detection*. This process detects malware domain names based on a domain black list. As mentioned above, domain names on a blacklist are usually C&C domain names or the infection domain names of a Trojan; however, the detection covers the domain names for many other malicious purposes, as well as C&C domain names.

4.2. Process 1: graph construction

Algorithm 1. P1: Graph construction

Input: Query set $Q = \{q\}$
Output: Domain name travel graph $G = (V, E)$
$V \leftarrow \phi$;
$E \leftarrow \phi$;
$q_c \leftarrow GetFirstQuery(Q)$;
while q_c is not the end of Q **do**
  $c \leftarrow GetClientIP(q_c)$;
  $v_i \leftarrow GetLastDomain(c)$;
$v_j \leftarrow GetDomain(q_t)$;
$V \leftarrow V \cup v_j$;
  if v_i is exist **then**
    $E \leftarrow E \cup Edge(v_i, v_j)$;
    $e \leftarrow GetEdgeIndex(Edge(v_i, v_j))$;
    // Weight of the edge e
    $W[e] \leftarrow W[e] + 1$;
    // Set of IP addresses contribute to the edge e
    $C_e \leftarrow C_e \cup c$;
  else
    $SetLastDomain(c, v_j)$;
  end if
  // Set of IP addresses which queried domain v_j
  $C_{v_j} \leftarrow C_{v_j} \cup c$;
  $q_c \leftarrow GetNextQuery(Q)$;
end while

The construction of the domain name travel graph is the first step in constructing a graph from given DNS traffic. Converting network traffic or logs to the graph allows sequence of queries from the DNS traffic to be traced. In the *DNTG* structure, each domain name is assigned to a node on a graph with a label, an ordered query from an identical client to two domain names is expressed as an edge. In the *Google* example mentioned at the start of the section on sequential correlation 3.1, domain *www.google.com* and *www.google.co.kr* are connected by an edge from *www.google.com* to *www.google.co.kr*, and *www.google.co.kr* and *www.gstatic.com* are connected in the same manner. This graph construction is performed on *GMAD* using Algorithm 1.

One of the key features of *DNTG* is that the DNS query activities from the DNS clients are accumulated on a graph. Graph operations on a single graph make the mechanism scalable in temporally and spatially. As a construction example, in Fig. 5, DNS queries of two DNS clients are recorded on a graph. The order and timing of the DNS queries of each client do not affect those of the other clients. In the example case, a DNS query for *a.com* from *Client 2* is not connected with *c.name* queried by *Client 1*.

4.3. Process 2: graph clustering

Algorithm 2. P2: Graph clustering

Input: Initial domain name travel graph $G_i = (V_i, E_i)$
Output: Clustered domain name travel graph $G_c = (V_c, E_c)$
$V_c \leftarrow V_i$;
$E_c \leftarrow E_i$;
$t_w \leftarrow GetEdgeWeightThreshold()$;
$t_{css} \leftarrow GetClientSetSizeThreshold()$;
$t_{csr} \leftarrow GetClientSharingThreshold()$;
$e \leftarrow GetFirstEdgeIndex(V_i)$;
while e is not the end of E_i **do**
  $v_{src} \leftarrow GetSrcNode(e)$;
  $v_{dst} \leftarrow GetDstNode(e)$;
  // Weight of the edge e
  $w \leftarrow W[e]$;
  // The number of distinct clients of the edge e
  $c \leftarrow GetSetSize(C_e)$;
  $s \leftarrow ClientSharingRatio(C_{v_{src}}, C_{v_{dst}})$;
  if $w < t_w$ or $c < t_{css}$ or $s < t_{csr}$ **then**
    // Edge cut
    $E_c \leftarrow E_c - e$;
    $C_e \leftarrow \phi$;
    $W[e] \leftarrow 0$;
  end if
endwhile

In the graph clustering step, domain groups are extracted from the entire set of domain names. The process makes an initial *DNTG* for the numbers of components by cutting edges. A component represents a domain group. The clustering process removes the edges whose *CSR*, number of clients, or number of queries are less than each threshold. If an edge connecting two domain names has a lower *CSR* than the other edges, *GMAD* considers that the domain name connected with the edge is not utilized by the same client set. The number of clients and queries represent the degree of regular and intended behaviors. As a result of edge removal, the nodes connected by the edges lose their connection. The nodes that lose their connection form a connected-component. The edges that separate a connected-component from the initial graph are called the cut-edges in graph theory fields. According to the definition of the *DNTG* structure, a component that is an isolated set of nodes represents an isolated domain group.

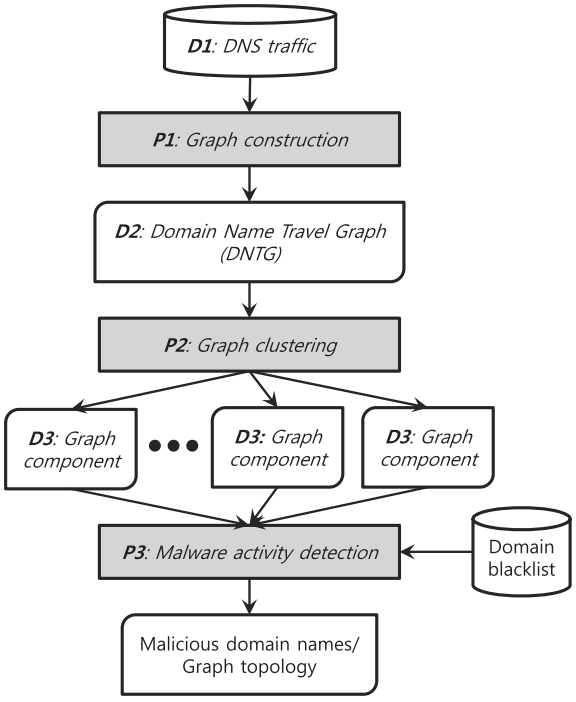

Fig. 4. Process flow diagram of the malware activity detection mechanism using DNTG.

GMAD adopts increasing thresholds and iterated cutting in the clustering process. Because every potential component has a different cut-edge value, the clustering needs to be performed with various t_{csr} which is an upper bound threshold for edge cut. For instance, a malware whose query domain list and member clients are static produces edges that have a *CSR* that is close to 100%. On the other hand, a malware that queries a sub-set of the entire query domain list at a monitoring time and is not time synchronized shows a relatively low level of client sharing. If a clustering uses a t_{csr} of 90% as an edge-cut threshold, it acquires the former component connected with the edges whose *CSR* values are over 90%, but it loses the edges and nodes of the latter case. To obtain the maximized member domains of each component, the clustering process needs to adopt a minimized clustering threshold for a graph. To adopt the minimized threshold for the components, *GMAD* performs a clustering process on the initial graph iteratively with a threshold that increases from a low starting value. We set the starting threshold value as 0.2 in our experiments. The definition of the clustering process is described as Algorithm 2.

4.4. Process 3: malware activity detection

The malware activity detection process determines the malicious domain names that are used for malware activities. The domain names that belong to a clustered graph are evaluated according to whether the domain name set of the graph includes any known malicious domain names. Fig. 6 shows an example of the overall detection process using the graph. The graph clustering process classifies the domain names on a graph into malicious and legitimate. A clustered graph includes the domain names and their sequential correlation information, and a clustered graph classified as a malicious graph provides a list of malicious domain names working together. As in the example, one clustered graph represents one malware activity. Yet, if more than two malwares share their clients, a graph represents all these malwares on a single graph. This exceptional case can be observed when the malwares download another malware on infected machines, which are installed by a same dropper or from a same infection route.

Even though the detection is performed on a graph structure, no graph traversal or comparison operation is required. Because a clustered graph is a closed graph, the detection process only reads

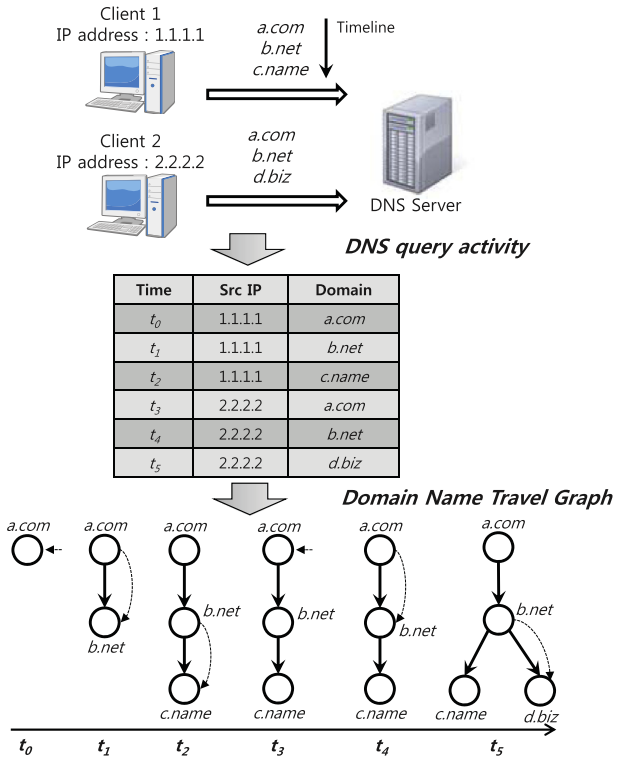

Fig. 5. Example for DNTG construction from DNS traffic.

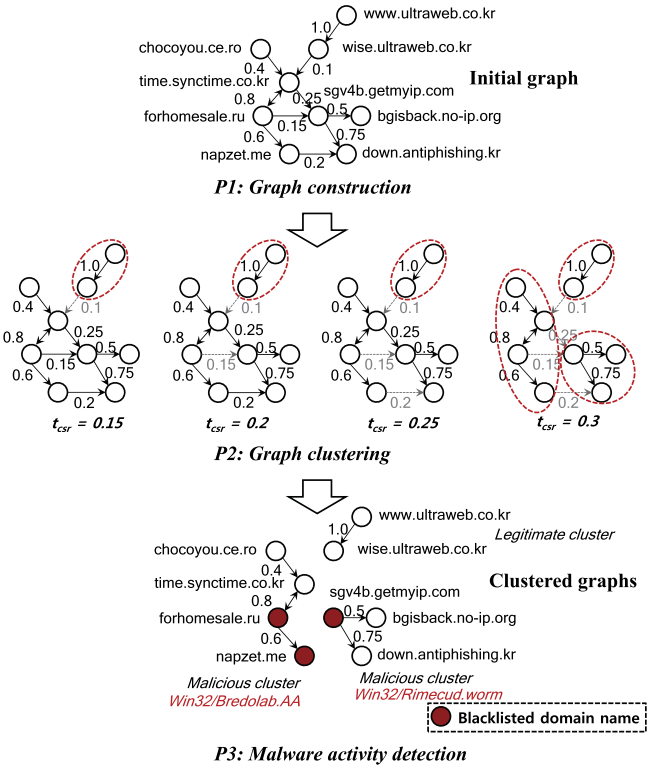

Fig. 6. A real-case example of detection using *GMAD* process.

a list of node labels and finds each label on a black list using a hash-map based search algorithm that takes constant time. We will discuss the time complexity of each process in the analysis section.

5. Experiments

In this section, we show the results of experiments on the DNS traffic of real networks, including the tracking ability of our mechanism. We implemented *DNTG* constructor and clustering processes using *C++* for experiments. A graph tool, *Pajek* [27], was used for visualization. The *Kamada–Kawai Algorithm* [28], included in *Pajek* was used as a graph layout algorithm. The experiments were performed on a desktop PC with a 3.30 GHz Intel quad core CPU, 8 GB main memory and Microsoft Windows 7 64 Bit. The mechanism is not yet fully automated and does not necessarily work in real time, but the automated parts take only a few minutes to process two hours of ISP level DNS traffic.

5.1. Data sets

We performed an experiment with DNS traffic captured on real networks to show the effectiveness of *GMAD*. We captured DNS query traffic in front of DNS servers in large ISP networks. We used four trace sets extracted from the traffic for our experiments and detailed analysis. These were captured around midnight and in the afternoon in the U.S. and South Korea. In this part, we call the data sets *US*1, *US*2, *KR*1 and *KR*2. Each data set has two hours of DNS traces. *US*1 and *KR*1 are afternoon traces, and *US*2 and *KR*2 are middle of the night traces.

As shown in Table 2, *US*1, and *US*2 have a similar traffic volume, but *KR*1 has twice as many query and domain numbers and three times as many clients as *KR*2. In terms of malicious activity, *KR*2 shows 80% of blacklist domain names as compared with *KR*1, even though it has only half of the query volume of *KR*1. It can be reasoned that several malwares in *KR*2 are active at night time.

5.2. Detection result and clustered graphs

The malicious domain names detected by *GMAD* are structured as domain name clusters, and a cluster is represented as a graph. The clustered graphs in Fig. 7 are the graphs of domain name clusters detected from *KR*1. In order to show graph topologies, we hid the domain names on the nodes. The brightness of the edges represents the *CSR* value of the edges: the darker edge represents a higher *CSR*, and the brighter edge represents a lower *CSR*. The clustered graphs that have more than two malware labels are graphs that contain the malicious domain names of plural malwares. These cases are observed when some clients are infected by plural malwares, or when the malwares share a domain name.

The number of detected domain names is affected mainly by the malwares using DGA. The multi-domain malwares using DGA generate thousands of domain names per hour. In our experiments, the malicious domain names generated by DGA had various patterns, including fully randomized domain names. Several malwares make hundreds of domain names for querying their C&C domain names at the public DNS blacklist services to check whether their C&C domain names are listed. Another malware activity that uses the numbers of domain names is attack target scanning for finding mail servers and vulnerable web and database servers. The scanning activity queries hundreds of domain names in the dictionary order. In this special case, although the scanning activities were detected by *GMAD*, we excluded the detected domain names from the statistics because the target domain names of the DNS activities were legitimate.

Another important contributions of our work to detect multi-domain malware activity is detecting domain name groups which are not share lexical similarity as well as obviously randomly generated domain names. In several previous approaches combatting the multi-domain malwares such as Yadav's work [12], their target was detecting randomly generated domain names by DGA. But, according to the reports analyzing recent malwares, many top ranked large scale malwares have the domain names which are not randomly generated. In practice, tens of malicious domain name groups which are detected by *GMAD* such as the malicious domain names on Table 3, have little lexical similarity among their member domain names. These types of malwares are hard to detect through their repeated patterns or probabilistic distribution of letters.

5.3. Detection accuracy

We gathered 13,392 distinct malicious domain names for performance evaluation from public black list: *DNS-BH* [29], *MalwareDomainList.com* [30] and *malc0de.com* [31]. Because *GMAD* detects malicious domain names which are not in the black list, we manually investigated non-listed domain names to determine their maliciousness. For the manual investigation, we used DNS activity analysis reports posted on public malware and malicious domain name information sites, *Threat Expert* [32], *Microsoft Malware Protection Center* [33], *Symantec Threat Explorer* [34], *Site Advisor* [35], *MalwareURL.com* [36] and SURBL [37]. Some of the information sites used for manual investigations provide their own black lists, but they are not fully public.

Table 4 shows the evaluation results using accuracy metrics. *GMAD* shows over 80% precision and a false positive rate lower than 0.5% for the four data sets. The precision is rate of correctly detected domain names among all detected domain names. The false positive rate (FPR) is rate of incorrectly detected legitimate domain names among all legitimate domain names. The definitions of the precision and false positive rate are in Eqs. (2) and (3), respectively. In the equations and tables in the experiments and analysis section, we abbreviate the true positive, true negative, and false positive as *TP*, *TN* and *FP*, respectively.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$FPR = \frac{FP}{TN + FP} \tag{3}$$

According to Table 4, *GMAD* showed up to seven hundreds of false positive cases in two hour data sets. False positive of *GMAD* is mainly caused by clustering failure. In several cases, sequential correlations which are yielded by a too few clients are hard to

Table 2
DNS trace sets for analysis and evaluation.

Data set	Date	Time	Query	Domains	Clients	Blacklisted
*US*1	June 22, 2010	13:00–15:00	1713 K	228 K	16 K	0.2 K (0.09%)
*US*2	June 22, 2010	22:00–24:00	1737 K	221 K	15 K	0.2 K (0.10%)
*KR*1	January 22, 2010	13:00–15:00	8661 K	1,151 K	183 K	1.0 K (0.09%)
*KR*2	January 22, 2010	22:00–24:00	4210 K	614 K	52 K	0.8 K (0.17%)

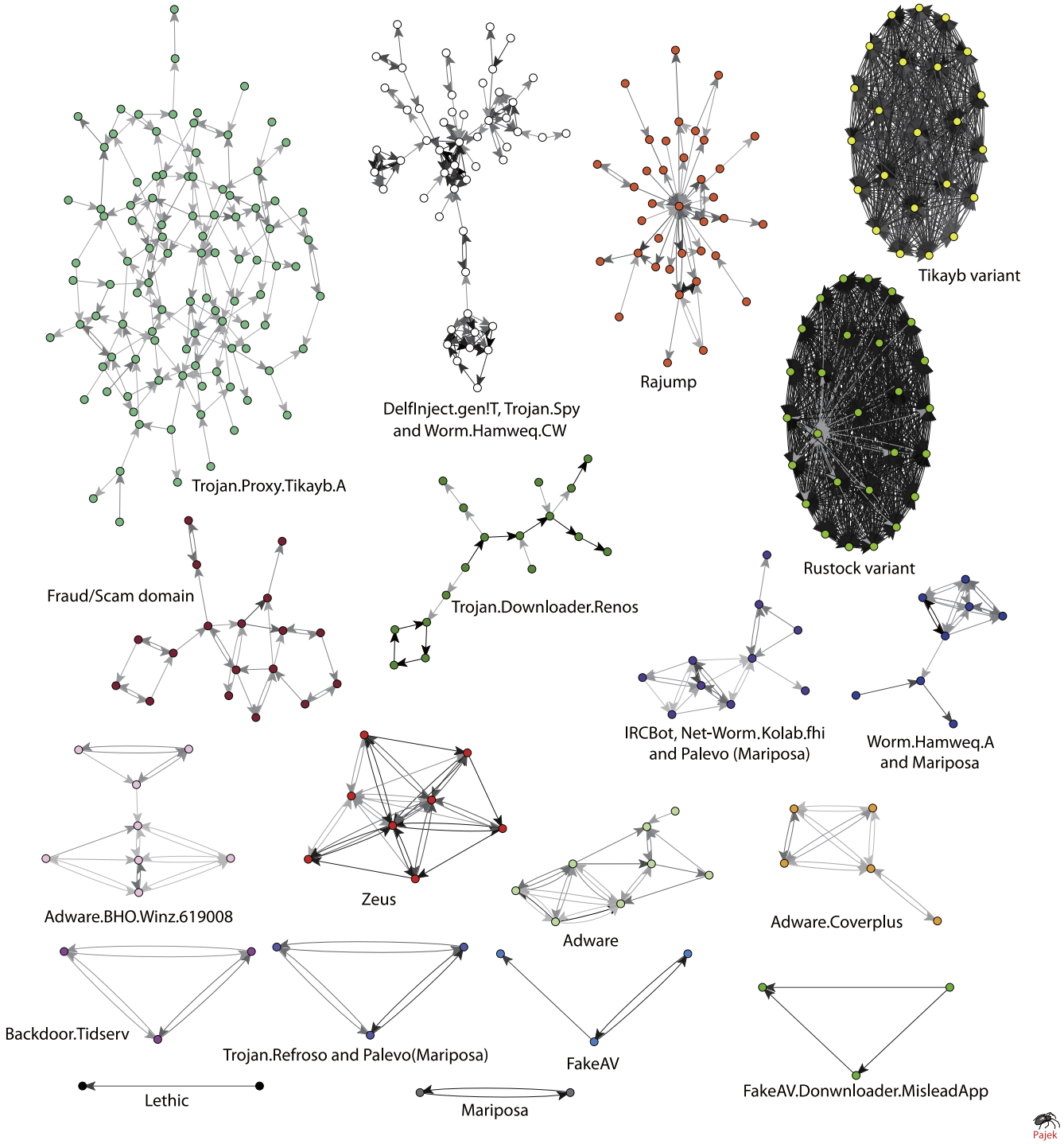

Fig. 7. Samples of malicious domain name clusters detected from *KR*1.

distinguish from incident correlations. In our experiments, when a legitimate domain name that was only queried by one or two clients had sequential correlation with a known malicious domain name, the legitimate domain name was false detected. This type of false detection cases can be reduced by setting a higher t_{css} for considering the behaviors that are caused by sufficiently many clients only. According to our analysis about the effect of t_{css}, in clustering the higher t_{css} were used, the higher detection precision was.

Another false detection case is caused by flawed knowledge-base. A legitimate domain name which was temporarily exploited for malicious activities, such as redirection host and DNS for C&C, is possibly black-listed, and it leads all the other domain names in a legitimate cluster to false detection. However, using these temporary black domain names to track the other malicious domain names and infected clients is useful and superior detection ability in compared with the previous methods. Detecting malicious activities using temporary black domain names without manual analysis and extracting only malicious activities from a long-term activities with lower false detection are our open problem.

5.4. Detection sensitivity

One of the most significant challenges in responding to malware is to detect the entirety of malicious domain names. The

Table 3
Examples of detected malware domain names on *KR*1.

Idx	Domain names	Group size
Redirection URLs and various malwares	megaparty.ws	27
	playnewforex.info	
	fotovideo2009.biz	
	totalinfluence.biz	
	ixxlkg.cn	
	totalinfluence.info	
	8ciehnvy.com	
	playnewforex.ru	
	djsu.info	
	...	
DelfInject.gen and Hamweq.CW	thunder.ircdevils.net	51
	4949.zerx-virus.biz	
	thunder.helldark.biz	
	yhtjanj.com	
	yhtjanj.biz	
	udptanj.com	
	udptanj.net	
	...	
Tikayb.A.	gotoplaywithme.name	101
	ac.ultima2009.info	
	ae.ultima2009.info	
	ec.gotoplaywithme.name	
	hh.gotoplaywithme.name	
	fk.gotoplaywithme.name	
	...	
IRCBot, Kolab.fhi and Palevo	mails.pes2009.biz	10
	serv1.alwaysproxy8.info	
	tes.enterhere2.biz	
	tes.stuckin.org	
	tes.memehehz.info	
	ninja.ibedyou2.com	
	idfc2.info	
	host.idfc2.info	
	...	

Table 4
Detection performance for real-world DNS data sets.

Data set	TP	TN	FP	Precision (%)	FPR (%)
*US*1	4917	223,175	588	89.31	0.26
*US*2	3650	217,211	185	95.17	0.08
*KR*1	5932	1,144,832	626	90.45	0.05
*KR*2	3705	517,891	687	84.36	0.13

redundancy provided by multiple domain names gives more resiliency and functionality to the malwares. It is difficult to predict the domain names that have not yet been used, but a detection method should detect at least newly generated and rarely used malicious domain names sensitively. Detection sensitivity is how many malicious domain names are detected in the sparser DNS query density and the less active machines. In our experiment, we compared the degree of the detection sensitivity using the number of detected malicious domain names over the same data set.

In this section, we describe our experimental comparison of the detection sensitivity of *GMAD* with one of the most evolved malware activity detection method, *BotGAD* [8]. *BotGAD* uses temporal group activity in malware DNS queries, i.e., time-synchronized DNS queries on a domain name, and similarity among querying client sets, along with time slots. *BotGAD* is an advanced method as compared with the other previous methods in that it detects malware domain names without any knowledge-base and considers several evasion techniques, such as hibernation and random interval activity. In addition, in the most recent study of *BotGAD* [9], the related malicious domain names are detected as a domain group using lexical similarity and network features, such as corresponding IP addresses among the domain names. However, its effectiveness is limited in the case of multi-domain malwares which have little lexical and networking similarity in their domain names.

In terms of detection sensitivity, *GMAD* shows about a 28-fold improvement over *BotGAD* on average over the four data sets. Detailed performance comparisons according to the detected malicious domain names along with the number of query clients for each domain name are illustrated in Fig. 8. Detecting the more number of malicious domain names that are queried by a few clients represents the more sensitive detection. Fig. 8 represents the number of domain names detected by *GMAD* and *BotGAD* in the data sets *US*1 (upper-left), *US*2 (upper-right), *KR*1 (lower-left) and *KR*2 (lower-right). *BotGAD* responds to the multi-domain malwares partly by grouping related domain names. However, it does not show a sufficient performance as compared with *GMAD*, even though the detection coverage of *GMAD* is dependent on a blacklist.

The better domain clustering ability of *GMAD* gives advantage to the detection sensitivity too. If a detection method cannot group the related domain names, this means that the density of the malicious activities is too sparse to allow their maliciousness to be determined using that method. The source of the major quantitative difference between *GMAD* and *BotGAD* is the detection of malicious domain names that are queried by fewer than two infected hosts. This result shows the importance of detection sensitivity and the number of malicious domain names that are undetectable by previous detection methods because of the density and size of the activity.

According to our analysis of the detection results, another reason for *GMAD*'s superiority is that *BotGAD* detects only C&C domain names that are regularly accessed. In contrast, the domain names detected by *GMAD* include the domain names which are used for malicious DNS activities in a wider sense, such as blacklist checking, spamming, domain scanning, fake C&C, as well as C&C domain names in sparse communication density. The fact that domain names are not detected by *BotGAD* as shown in Fig. 8, even though queried by more than three infected machines, means that the malwares evaded *BotGAD* using evasion techniques to combat detection methods using temporal property and lexical similarity.

On the other hand, compared with the Jiang et al.'s work [22] which has the most similar target to our work and also takes graph-based clustering, *GMAD* shows better detection sensitivity. In the data sets *US*1 and *US*2 which are smaller but have similar query-per-client rate to the data set that are used in [22], *GMAD* shows 60% and 20% more domain detection than all of the suspicious domain names of their work. An obvious difference in detection ability is the detection of valid domain names which was not considered at the Jiang et al'.s proposal. In their work, only invalid domain names are considered, which have no corresponding resource records. It means their method does not cover the malicious domain names that are working actively, as well as active C&C.

In detail, the data set used in Jiang et al.'s experiments has 20% more queries and clients than our data set, i.e., *US*1 and *US*2. In the Jiang et al.'s experiment, near 3000 invalid domain names were considered as a suspicious domain. The actual number of domain names which were verified as the malicious were not concretely disclosed. However, in our result, *GMAD* detected 4917 and 3650 domain names as the verified malicious domain names, i.e., *TP* on Table 4, from *US*1 and *US*2, respectively. Although this comparison is not based on equivalent data set, the result obviously shows the difference on detection performance caused by the detection ability for the valid domain names and coverage.

6. Analysis

In this section, we analyze the effect of the metrics that represent the degree of sequential correlation on detection performance.

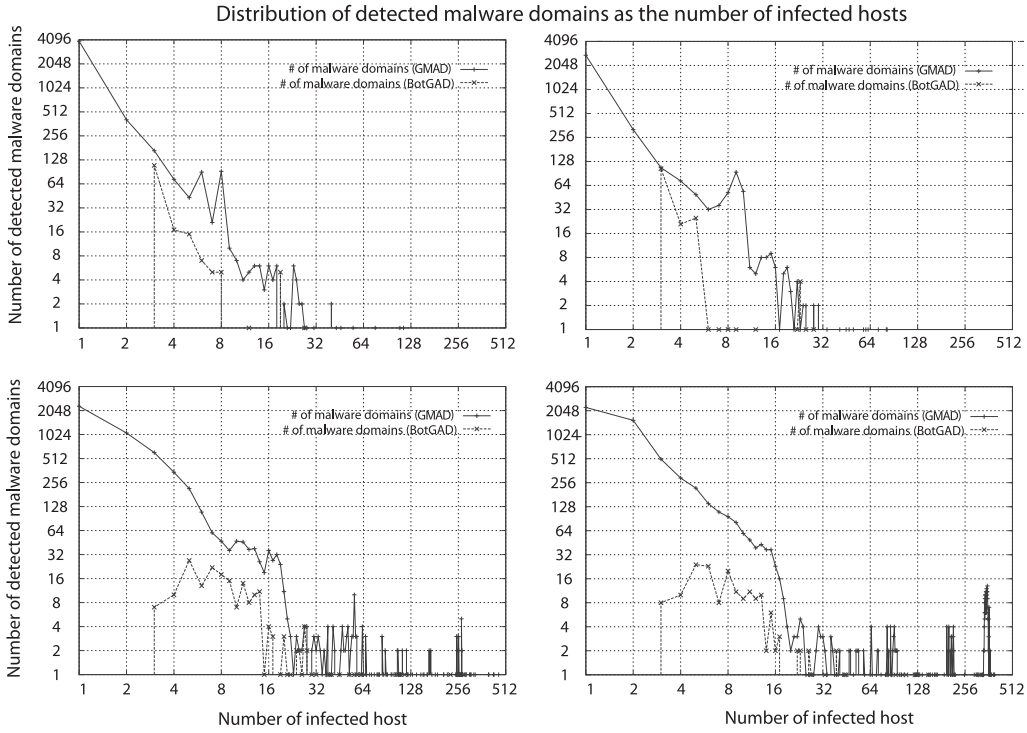

Fig. 8. Malware domain detection result comparing with *BotGAD*.

Through the analysis, we attempt to find the trade-off between, as well as the most beneficial configuration of, detection coverage and accuracy to detect real world malwares. The analysis showed that the different configurations make it possible to detect different malware activities which cannot be detected by the other configurations. This result indicates that we should adopt an optimized detection strategy along with the operation purpose and policy. Finally, we performed a scalability analysis in terms of time complexity to show the practicality and efficiency of *GMAD*.

6.1. Clustering accuracy as minimum valid DNS clients

GMAD effectively detects malicious domain names that are used by only a few on-line infected hosts. However, most of the properties estimated from DNS behavior, including that used in our proposed method, sequential correlation, are more reliable when there are more behavior subjects, i.e., DNS clients. This means that there can be a trade-off between the minimum number of infected clients for estimating the maliciousness of a domain name and false detection. In other words, including DNS queries to a domain name queried by fewer DNS clients for estimating a property render a detection mechanism more sensitive but less accurate. To maximize the effectiveness of our mechanism, we analyzed the effect on the detection accuracy of *GMAD* when the minimum number of valid DNS clients, denoted by t_{css} representing an edge-cut threshold for client set size (*CSS*), is applied in the clustering algorithm in Section 4.3. t_{css} causes *GMAD* to ignore sequential correlation generated from clients whose number is less than t_{css}.

According to the results of experiments to analyze the effect of sensitive detection, shown in Table 5, a higher t_{css} yields fewer detected domain names but a higher accuracy level. A higher t_{css} yields more true positive malicious domain names, quantitatively, but it does not include the result of the lower t_{css} cases. Where the set of true positive malicious domain names that are detected when t_{css} is k is expressed as $TP_{t_{css}=k}$. We compared the results while increasing t_{css} from one to five. In the comparison between $TP_{t_{css}=1}$ and $TP_{t_{css}=5}$, the number of domain names detected only in

Table 5
Detection performance as filtering threshold t_{css} on *KR*1.

t_{css}	TP	TN	FP	Precision (%)	FPR (%)
1	5932	1,144,832	626	90,45	0.05
2	3946	1,147,066	378	91.26	0.03
3	3248	1,148,049	63	98.10	0.01
4	4901	1,146,445	44	99.11	0.00
5	5946	1,145,426	18	99.70	0.00

$t_{css} = 1$ is represented as $|(TP_{t_{css}=1} - TP_{t_{css}=5})|$. In the data set *KR*1, $|(TP_{t_{css}=1} - TP_{t_{css}=5})|$ was 2120 domain names and the opposite case, $|(TP_{t_{css}=5} - TP_{t_{css}=1})|$, was 2134. The commonly detected domain names in both $TP_{t_{css}=1}$ and $TP_{t_{css}=5}$ are 3812 domains. This means that a different t_{css} leads to the detection of different types of malicious activities.

In terms of detection sensitivity, the number of detected domain names decreased with a higher t_{css} as there was more loss of connectivity by the edge-cut using t_{css}. However, the number of true positive cases was increased. This means that the malicious domain names that were not detectable using a clustering based on *CSR* value can be detected a clustering using *CSS*. According to our empirical analysis, a group of domain names which has a larger *CSS* possibly has low *CSR* and vice versa. A malicious activity whose clients share a similar querying domain name set has high *CSR* among the domain names, and it should be dealt with clustering using *CSR*. On the other hand, a malicious activity performed by numbers of clients shows spatially scattered behavior unless the numbers of clients are finely synchronized. A clustering using *CSS* is proper to this type of activity.

6.2. Client sharing ratio of malicious domain clusters

The *CSR* of each malicious domain name cluster represents the malwares' client–server structure and C&C strategy. As discussed in the explanation of the graph clustering process, the *CSR* among malicious domain names is dependent on the communication

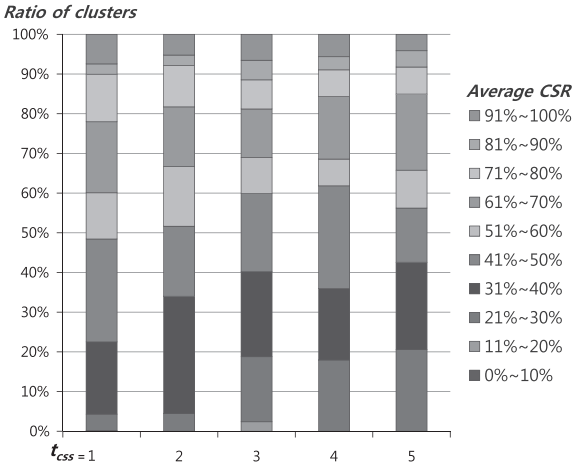

Fig. 9. The distribution of average *CSR* on the detected domain clusters as t_{css}.

strategy between malwares and malicious domain names. A finely synchronized group of machines infected by a malware that has a static domain names list shows a high *CSR* in its DNS activities. In contrast, the malwares operated in sub-grouping or sparse communication generate DNS queries only for a part of their domain names at one time.

Within the malicious domain name clusters detected from *KR*1, the domain names in more than one third of the clusters were connected with edges that had an average *CSR* from 30% to 50%. Fig. 9 shows the distribution of the average *CSR* values of the malicious clusters. Most clusters had an average *CSR* of 30–40% and less than 10% of the clusters were connected with fully shared clients. In terms of the number of infected clients, the malicious domain names that had more clients showed a lower *CSR*. This can be interpreted in two ways. The first possibility is that it is relatively hard to synchronize the larger malwares, which have more infected clients, and the clients of each malicious domain name also may not be synchronized. The second possibility is that the malwares are using evasion techniques to hide their group activities, intentionally. In conclusion, if we consider that the malwares that have more on-line clients are more critical, even if their DNS activities are sparse and scattered, we should focus on detecting domain names that have large client sets and a low *CSR*. A low *CSR* means that the entire number of infected clients is much greater than that observed. On the other hand, the largest portion of detected malwares consists of the small scale malwares which have only one to three malicious domain names and clients within our monitoring time slot. Estimating the optimal time and resource consumption for revealing all the infected clients of a malware can be another research topic.

6.3. Scalability analysis

GMAD has a practical performance that is robust against the huge volume of input data, i.e., the DNS query traces. According to time complexity analysis, where n is the number of the input DNS queries, *GMAD* has $O(n)$ time complexity through the entire detection process. In the worst cases, the time complexity of each step is $O(n)$, and the reasoning is as follows.

- *Graph Construction*: twice of map searches which takes a constant time [38] for n DNS queries.
- *Graph Clustering*: $n-1$ times of comparison operation for $n-1$ arcs.
- *Weak-connected component separation*: finding weak-connected component [39] from n and $n-1$ nodes and edges respectively.

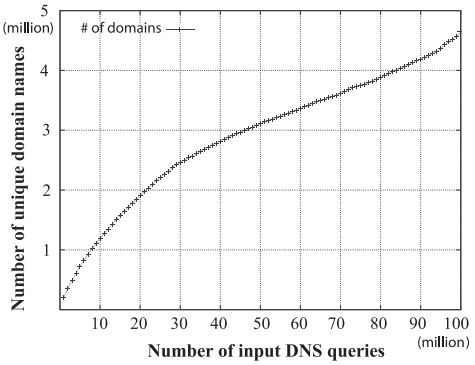

Fig. 10. The number of unique domain names as the increase of input DNS queries.

- *Blacklist matching*: n times hash-map search, which takes a constant time.

In the worst case, the number of nodes and edges in the DNTG is as many as the number of DNS queries, however, in fact, they do not linearly increase with the size of the DNS queries. The time and space complexity are dependent on the number of unique domain names, and the growth of unique domain names slows down as illustrated in Fig. 10 following *Heaps' law* [40]. In our empirical analysis, the Heaps' law formula for the number of unique domain names is $D(n) = 0.28 \cdot n^{0.6}$. As an illustration, the size of DNTG constructed from eight million DNS queries for two hours is only 21 MB. Computation overhead per each arc is also slight. Three processes except the graph clustering within the four processes only need a couple of integer comparison and hashing per each arc. The graph clustering has the heaviest computation overhead because it compares sets of IP address. However, the computation time and space consumption is minimized by hashing to the IP addresses of the clients which are the input data of estimating the *CSR*. The hash matching is much faster than original 32 bit IP address matching.

On the other hand, *GMAD* is practically scalable to large scale data, as compared with other previous studies [8,9,19,20,22] which compute information of the clients and domains at once. Concretely, *GMAD* minimized the data which must be analyzed simultaneously. The topology of a graph only includes the connectivity information and their arc weight. In the same DNS query data size, the size of the topology information is much smaller than the timing data of the previous temporal approaches must process simultaneously. The spatial approaches based on the similarity of plural DNS clients at once, not one by one, also have same memory overhead problem to the temporal approaches. In contract, the estimation of arc weight in *GMAD* can be processed independently. It means that the estimation procedure only needs to load the information of two nodes on the memory at once, and it can be processed in parallel.

7. Discussion

In this section, we discuss about considerable drawbacks and questions, i.e., the limitation of sequential correlation against the traditional single domain malware, and the universality of *GMAD*.

7.1. Drawback of sequential correlation approach

Though *GMAD* mainly focuses on the multi-domain malicious activities and evasion technique problems, *GMAD* covers a part of single domain names which show the sequential correlation to themselves. But the coverage is relatively limited. The sequential correlation has a limitation on to detect malicious single domain

activities which are different to the normal single domain activities only on their temporal patterns. As a sequential correlation, those two activities are considered as the same patterns. In contrast to the multiple domain activities, the single domain activities have to investigated by their strict regularity if they have, because temporally irregular queries to single domain name have little difference to the normal activities. However, the lack of robustness caused by the strictness is a well-known limitation of previous temporal approaches. To overcome this trade-off may need to use another property and data sources in addition to the DNS query pattern. It can be a topic for our future work.

7.2. Locality of experimental data sets and universality

A malicious activity detection method may show different performance to the different data sets which have their characteristic locality. In many cases, domain black lists have been managed suitable to each ISP. We observed that the kind of malwares were slightly different between data sets gathered from U.S and South Korea. But *GMAD* has enough universality to this difference. As shown on Table 5, it does not affect the detection performance. *GMAD* does not shows significant difference on accuracy for the four different data sets. The number of detected domain names can be different due to the number of domain flux malwares with the DGA. In another point of view, even though *KR*1 has three times of queries and twice of domain names compared with *KR*2, it does not affect to the accuracy.

8. Conclusion

Malwares on the Internet are becoming intelligent and complex, and therefore they can evade the legacy detection methods. DNS activity analysis, which has been one of the most effective response methods, also faces the evasion problems incurred by the malware's use of multiple domain names. In this paper, we proposed a malware activity detection mechanism named *GMAD* that uses graph expression and a robust DNS behavior property, sequential correlation. *GMAD* finds malicious domain names in DNS traffic, which are utilized by the malwares as C&C servers, DNSs, update servers, etc. Through the graph representing the sequential correlation and graph clustering, *GMAD* reveals the malicious domain names, even though the DNS activities for the domain names are being adopted temporal or spatial evasion techniques. In our experimental evaluation using DNS traffic data gathered from two ISP DNSs in the U.S. and South Korea, *GMAD* showed superior detection accuracy and sensitivity as compared with the previous DNS analysis approaches. The major contribution of this study is to reveal the malicious domain names that have not been detected by the legacy DNS analysis methods using temporal and spatial behavior properties. We extend the detection coverage to all the sequentially correlated domain names utilized for malware activities, even though the domain names are not regularly and periodically queried as are the legacy C&C domain names. We believe that the contributions of our work will facilitate the prevention of damage from malware infection and malicious activities on the Internet more effectively.

Acknowledgements

This research was supported by the Public Welfare & Safety Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2012M3A2A1051118).

References

[1] D. Maslennikov, Y. amestnikov, Kaspersky Security Bulletin 2012, The overall statistics for 2012, Tech. rep., Kaspersky Lab, December 2012.
[2] Symantec MessageLabs, Intelligence report, Tech. rep., Symantec Corporation, January 2013.
[3] Symantec MessageLabs, Intelligence report, Tech. rep., Symantec Corporation, March 2011.
[4] Symantec MessageLabs, Intelligence report, Tech. rep., Symantec Corporation, June 2011.
[5] R. Perdisci, I. Corona, D. Dagon, W. Lee, Detecting malicious flux service networks through passive analysis of recursive DNS traces, in: Proc. of the Annual Computer Security Applications Conf. (ACSAC), IEEE Computer Society, 2009, pp. 311–320.
[6] A. Ramachandran, N. Feamster, D. Dagon, Revealing botnet membership using dnsbl counter-intelligence, in: Proc. of the 2nd Conf. on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), USENIX Association, 2006, pp. 49–54.
[7] J.A. Morales, A. Al-Bataineh, S. Xu, R. Sandhu, Analyzing DNS activities of bot processes, in: Proc. of the 4th IEEE Intl. Conf. on Malicious and Unwanted Software (MALWARE), IEEE, 2009, pp. 98–103.
[8] H. Choi, H. Lee, H. Kim, BotGAD: detecting botnets by capturing group activities in network traffic, in: Proc. of the 4th Intl. ICST Conf. on COMmunication System softWAre and middlewaRE (COMSWARE), ACM, 2009, pp. 1–8.
[9] H. Choi, H. Lee, Identifying botnets by capturing group activities in DNS traffic, Comput. Netw. 56 (1) (2012) 20–33.
[10] K. Ishibashi, T. Toyono, H. Hasegawa, H. Yoshino, Extending black domain name list by using co-occurrence relation between DNS queries, IEICE Trans. Commun. 95 (3) (2012) 794–802.
[11] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, D. Dagon, From throw-away traffic to bots: detecting the rise of DGA-based malware, in: Proc. of the 21st USENIX Security Symposium, USENIX Association, 2012.
[12] S. Yadav, A.K.K. Reddy, A.L.N. Reddy, S. Ranjan, Detecting algorithmically generated domain-flux attacks with DNS traffic analysis, IEEE/ACM Trans. Netw. 20 (5) (2012) 1663–1677.
[13] J. Wolf, Technical Details of Srizbi's Domain Generation Algorithm, Tech. rep., FireEye Malware Intelligence Lab, 2008.
[14] P. Porras, H. Saidi, V. Yegneswaran, Conficker C Analysis, Tech. rep., SRI International, 2009.
[15] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, G. Vigna, Your botnet is my botnet: analysis of a botnet takeover, in: Proc. of the 16th ACM Conf. on Computer and Communications Security (CCS), ACM, 2009, pp. 635–647.
[16] P. Lin, Anatomy of the mega-D takedown, Network Secur. 2009 (12) (2009) 4–7.
[17] J. Lee, J. Kwon, H.-J. Shin, H. Lee, Tracking multiple C&C botnets by analyzing DNS traffic, in: Proc. of the 6th IEEE Workshop on Secure Network Protocols (NPSec), IEEE, 2010, pp. 67–72.
[18] R. Villamarín-Salomón, J.C. Brustoloni, Bayesian bot detection based on DNS traffic similarity, in: Proc. of the 2009 ACM Symposium on Applied Computing (SAC, ACM, 2009, pp. 2035–2041.
[19] G. Guofei, Z. Junjie, W. Lee, BotSniffer: detecting botnet command and control channels in network traffic, in: Proc. of the 15th Annual Network and Distributed System Security Symposium (NDSS), ISOC, 2008.
[20] N. Shishir, M. Prateek, H. Chi-Yao, C. Matthew, B. Nikita, BotGrep: finding p2p bots with structured graph analysis, in: Proc. of the 19th USENIX Security Symposium, USENIX Association, 2010, pp. 95–110.
[21] A. Yamada, H. Masanori, Y. Miyake, Web tracking site detection based on temporal link analysis, in: Proc. of Intl. Conf. on Advanced Information Networking and Applications Workshops, IEEE Computer Society, 2010, pp. 626–631.
[22] N. Jiang, J. Cao, Y. Jin, L.E. Li, Z.-L. Zhang, Identifying suspicious activities through DNS failure graph analysis, in: Proc. of the 18th IEEE Intl. Conf. on Network Protocols (ICNP), IEEE, 2010, pp. 144–153.
[23] J.P. John, A. Moshchuk, S.D. Gribble, A. Krishnamurthy, Studying spamming botnets using botlab, in: Proc. of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI), USENIX Association, 2009, pp. 291–306.
[24] G. Gu, P. Porras, V. Yegneswaran, M. Fong, W. Lee, Bothunter: detecting malware infection through ids-driven dialog correlation, in: Proc. of 16th USENIX Security Symposium, USENIX Association, 2007, pp. 1–16.
[25] Alexa Internet Inc, Alexa top sites. <http://www.alexa.com>.
[26] J. Zhang, Y. Xie, F. Yu, D. Soukal, W. Lee, Intention and origination: an inside look at large-scale bot queries, in: Proc. of the 20th Annual Network and Distributed System Security Symposium (NDSS), ISOC, 2013.
[27] V. Batagelj, A. Mrvar, M. Zaversnik, Pajek. <http://pajek.imfm.si>.
[28] T. Kamada, S. Kawai, An algorithm for drawing general undirected graphs, Inf. Process. Lett. 31 (1989) 7–15.
[29] DNS-BH project team, DNS-BH. <http://www.malwaredomains.com>.
[30] MalwareDomainList, Malwaredomainlist.com. <http://www.malwaredomainlist.com>.
[31] Malc0de, malc0de.com. <http://malc0de.com/database>.
[32] Threat Expert Ltd., Threat expert. <http://www.threatexpert.com>.

[33] Microsoft Corporation, Microsoft malware protection center. <http://www.microsoft.com/security/portal>.
[34] Symantec Corporation, Symantec threat explorer. <http://us.norton.com/securityresponse/threatexplorer/index.jsp>.
[35] McAfee Inc, Site advisor. <http://www.siteadvisor.com>.
[36] The MalwareURL Team, MalwareURL. <http://www.malwareurl.com>.
[37] SURBL, SURBL: URI Reputation Data. <http://www.surbl.org>.
[38] Microsoft Corporation, ATL collection classes. <http://msdn.microsoft.com/en-us/library/vstudio/15e672bd(v=vs.100).aspx>.
[39] R.E. Tarjan, Depth-first search and linear graph algorithms, SIAM J. Comput. 1 (2) (1972) 146–160.
[40] H. Heaps, Information Retrieval, Computational and Theoretical Aspects, Library and Information Science, Academic Press, 1978.