Correlating File-Based Malware Graphs Against the Empirical Ground Truth of DNS Graphs

Jukka Ruohonen
University of Turku, Finland
juanruo@utu.fi

Sanja Šćepanović
Aalto University, Finland
sanja.scepanovic@aalto.fi

Sami Hyrynsalmi
University of Turku, Finland
sthyry@utu.fi

Igor Mishkovski
Aalto University, Finland
& University of Ss. Cyril and Methodius – Skopje, Macedonia
igor.mishkovski@finki.ukim.mk

Tuomas Aura
Aalto University, Finland
tuomas.aura@utu.fi

Ville Leppänen
University of Turku, Finland
ville.leppanen@utu.fi

ABSTRACT

This exploratory empirical paper investigates whether the sharing of unique malware files between domains is empirically associated with the sharing of Internet Protocol (IP) addresses and the sharing of normal, non-malware files. By utilizing a graph theoretical approach with a web crawling dataset from F-Secure, the paper finds no robust statistical associations, however. Unlike what might be expected from the still continuing popularity of shared hosting services, the sharing of IP addresses through the domain name system (DNS) seems to neither increase nor decrease the sharing of malware files. In addition to these exploratory empirical results, the paper contributes to the field of DNS mining by elaborating graph theoretical representations that are applicable for analyzing different network forensics problems.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: [Invasive software]

Keywords

DNS mining, shared hosting, network forensics, complex network analysis, ground truth problem, cyber security

1. INTRODUCTION

This empirical paper investigates the concept of sharing. The interest is to explore whether the sharing of unique malware files correlates with the sharing of normal, non-malware files between Internet domains, and whether analogous correlations are present between the sharing of malware files and the sharing of IPv4 addresses. Shared web hosting provides a motivation for hypothesizing about the presence of the latter type of correlations. In particular, the traditional shared IP hosting (a.k.a. name-based virtual hosting) has not lost its popularity for servicing web pages from multiple virtual hosts (or, in this paper, domains) but from a single IPv4 address. Although the historical arrival of virtual private servers – and cloud computing in general – have presumably decreased the overall popularity of this kind of web hosting, the associated security issues have not disappeared from the Internet [9]. Then, to hypothesize, domains that have shared both IPv4 addresses and unique malware files might well conceive a cluster of domains that have been compromised through a common compromised infrastructure. Another plausible option would be a set of popular domains belonging to a large content delivery network with a fixed IPv4 pool. Likewise, malware is commonly associated with different benign and non-compromised file-sharing services and related cloud computing infrastructures [7, 14], which may also explain the sharing of IPv4 addresses.

These loose hypotheses align with the adopted graph theoretical approach elaborated in the opening Section 2. At the risk of overgeneralization, the background can be summarized by pointing out two popular graph theoretical approaches for studying malware in the Internet. A traditional approach has built a directed or undirected malware graph based on a bipartite structure comprised of malware files and machines [11]. Here, the term machine can be understood as an Internet host, which, in terms of DNS, resolves either to a fully qualified domain name or an IP address. These two basic building blocks, in turn, have provided the basis for the so-called DNS graphs with which the placement of edges is often done in terms of the labeled domain and address vertices [17]. By positing DNS graphs as a comparative "ground truth" representation (cf. [19]), this paper compares the two graph representations empirically.

For comparing the file-based and address-based representations in Section 3, the paper uses simple correlation analysis with a malware dataset that is primarily based on the infrastructure maintained by the security company F-Secure. The empirical analysis also utilizes a further subgraph representation that explicitly combines the two graph representations for sharing of files and IPv4 addresses. That is, this combined graph representation defines a group of domains that have shared both malware and IPv4 addresses. Although the paper finds no strong correlations between the two main graph representations, the elaborated subgraph representation is still applicable as a simple data reduction technique in network forensics, as concluded in Section 4.

ECSA, November 28-December 02, 2016, Copenhagen, Denmark

DOI: http://dx.doi.org/10.1145/2993412.2993414

2. MOTIVATION AND SETUP

The motivation is done by postulating a high-level network forensics scenario through graph theoretical reasoning.

2.1 Theoretical Representation

A number of relational representations can be considered for modeling malware in a graph $G_t = (V_t, E_t)$ comprised of finite sets of vertices V_t and edges E_t. In this paper, all vertices represent Internet domain names, while the placement of edges vary. As a sign of this variance, the subscript t is used to emphasize that G_t is a theoretical representation.

By a theoretical representation, it is meant that a given graph aims to model hypothetical connections between vertices. For instance, it can be hypothesized that an edge between two Internet domains, $v_t, u_t \in V_t$, may reflect a common malware orchestrator in case the two domains have both made the same unique malware file available for download at some point in time. The hypothesized relations are only statistical approximations; nothing can be concluded about the reason why the unique malware file was downloadable from v_t and u_t. Often, the sharing of a unique malware between two domains, $(v_t, u_t) \in E_t$, merely means that the malware has been "dropped" to some compromised or otherwise innocent domains v_t and u_t, for instance.

In this paper, each vertex represents more specifically a domain aggregated to the second highest level in the hierarchical notation for fully qualified domain names. For instance, v_t might denote `linuxmint.com`, where the top-level domain (TLD) is `com` and the second-level domain (2-LD) is `linuxmint`. While this aggregation is generally rather encompassing, it is also a commonly done simplification in empirical DNS mining [17]. Furthermore, it does not affect the theoretical representation as such.

Thus, to briefly continue the example, the domain `linuxmint.com` was recently (in early 2016) compromised with an attack that involved a payload of a backdoor'ed Linux installation image [6]. While technical details were not disclosed to the public, it is clear that the attack could have been severe already theoretically due to the way Linux operating systems are distributed in the Internet. Then, to theorize, a hypothetical edge ($\texttt{linuxmint.com}_t$, u_t) $\in E_t$ to another aggregated domain, $u_t \neq \texttt{linuxmint.com}_t$, would mean that the same unique malicious image would have been shared by two domains; hence, there would be a weak signal for hypothesizing about a common orchestrator behind the attack. Though, the file might as well having been automatically propagated to some download mirror u_t, and, thus, the word hypothetical should be boldfaced throughout the paper. This boldfacing is related to the lack of solid ground truths in computer, network, and information security.

2.2 Ground Truth Representation

The second graph representation follows the commonly used approach (cf. [12, 19]) of defining an explicit ground truth graph against which a theorized representation can be compared. In this representation, each edge $(v_a, u_a) \in E_a$ between two aggregated domains implies that the two domains have shared at least one IPv4 address (i.e., A record) in their DNS records at some point in time. While the operationalization details may vary [17], G_a is still a typical example of a DNS graph. Thus, to again continue with the earlier example, the hypothetical edge ($\texttt{linuxmint.com}_t$, u_t) $\in E_t$ might also show as ($\texttt{linuxmint.com}_a$, u_a) $\in E_a$, which would mean that the malicious Linux image shared between the two domain vertices would have been actually shared also from the same IPv4 address(es) at some point in time. This particular, imaginary example notwithstanding, such a direct correspondence would be a fairly good indication about a name-based hosting solution whereby services provided by different domains are served from a single IPv4 address.

It should be understood that also this IP address representation leads to a number of problems related to the ground truths in computer networking. To give an example: the hypothetical presence of a theoretically defined $e_t = (v_t, u_t)$ could be probed via different network sensor solutions for detecting the transmission of malware files or the network behavior of a particular malware. In this case, deep packet inspection (layer-7) would be a typical empirical ground truth against which more efficient but coarse packet-based (layer-3) techniques could be contrasted [2, 5]. In this paper, however, G_a is constructed by resolving the $|V_t| = |V_a| = n$ domains through live DNS a few times.

Consequently, DNS-specific ground truth problems [17] are present. For instance, due to the Akamai-style content delivery networks, round-robin DNS, and the so-called (malicious) fast flux networks [15], it is often difficult to make robust mappings between domains and their addresses. The caches for live DNS resolving might be even poisoned [18], which would be a particularly illuminating case on how a network security ground truth assumption might be actively compromised. While keeping these points in mind, the analytical construction of G_t and G_a is subsequently elaborated.

2.3 Graph Construction

The theoretical representation G_t can be further divided into two spanning subgraphs that represent the sharing of malware files and clean files, G_t^m and G_t^c, respectively. Both contain the exact same domain name vertices, which are all also present in the IPv4 address representation G_a. Thus, only the arrangement and amount of edges vary within the set $\{G_t^c, G_t^m, G_a\}$. This feature of vertex stability is ensured during the construction of the three observed (sub)graphs, which are summarized in Table 1.

The relational representations are constructed in two steps. In the initial vertex construction step all qualified domains are added as vertices to all of the observed graphs. To qualify for the addition, (a) a sampled host must be a valid 2-LD-TLD, which implies that sampled hosts represented as IPv4 addresses are excluded by design. Moreover, (b) each semantically valid 2-LD-TLD must also be a valid Internet domain at the time of the construction. Here, validity means that at least one live DNS resolving round provided one or more A records for mapping the aggregated domain name to one or more IPv4 addresses. Finally, (c) any host that passes the two preceding conditions must have also shared at least one file with another valid 2-LD-TLD for which at least one IPv4 address was available from the live DNS database. The last criterion implies that the characteristics of G_a cannot be directly used to quantify name-based shared hosting. That is to say, G_a proxies the sharing of addresses only among those domains that have shared clean or unclean files.

Edges are added to the initialized graphs in the second step. This addition requires making a tricky ground truth assumption to determine whether a given $e_t \in E_t$ should be added to the "malware graph" G_t^m or the "clean graph" G_t^c. In other words, it must be determined whether the theoret-

Table 1: Three Observed Graph Representations

	Representation		
	A. Theoretical (Section 2.1)		B. "Ground Truth" (Section 2.2)
Graph	1. $G_t^c = (V_t^c, E_t^c)$	2. $G_t^m = (V_t^m, E_t^m)$	3. $G_a = (V_a, E_a)$
Edges	Clean files: if $v_t^c, u_t^c \in V_t^c$ have distributed the same clean file, $(v_t^c, u_t^c) \in E_t^c$	Malware files: if $v_t^m, u_t^m \in V_t^m$ have distributed the same malware file, $(v_t^m, u_t^m) \in E_t^m$	Addresses: if $v_a, u_a \in V_a$ have resolved to the same IPv4s, $(v_a, u_a) \in E_a$
Vertices	Domains (2-LD-TLDs)	Domains (2-LD-TLDs)	Domains (2-LD-TLDs)

ically defined edge e_t is unclean or clean; whether the underlying shared files are malware or not. Following previous research [14], this choice is made based on a large number of malware (anti-virus) detection engines used in the contemporary security industry. Specifically, any given edge is defined to be malware in case

$$\sum_{i=1}^{k} \sum_{j=1}^{d} g_j(f_i) > 0, \tag{1}$$

where the first summation is over the $f_1, \ldots, f_k$ unique files shared by any aggregated domains v_t and u_t. The second summation aggregates results from d malware detection engines, each one of which is defined to output

$$g_j(x) = \begin{cases} 0 & \text{if the file } x \text{ is "clean", or} \\ 1 & \text{if the file } x \text{ is "unclean".} \end{cases} \tag{2}$$

This operationalization is deterministic, strict, and encompassing. It is deterministic because, in theory, the choice could be also (better) determined by a probabilistic machine learning model. It is strict because only in case all d detection engines classified a file clean, the file is defined to be clean. It is encompassing because an edge is defined to be clean only in case all of the files are clean. In other words, even one malware file out of a thousand clean files, for instance, classifies an edge as malware, and thus, the edge would be added to G_t^m. From a purely empirical perspective, the utilized operationalization balances the desirable goal of having a large $|E_t^m|$ over the increase of false positives. This said, the operationalization would be easy to alter for instance by using thresholds, which would allow to vary the confidence placed on the ground truth assumption.

Also the addition of edges to G_a requires a noteworthy ground truth assumption. In particular, the domain-to-IP mappings require a predefined learning period [17]. To elaborate this learning, consider resolving a domain $v_a \in V_a$ for r times, such that the domain is associated with addresses

$$A = \{(a_{1_1}, \ldots, a_{k_1}), \ldots, (a_{1_r}, \ldots, a_{k_r})\}, \tag{3}$$

where each resolving round results $k \geq 0$ addresses through the global domain name system. Due to the qualification requirement (b), $|A| > 0$ for any $v_a \in G_a$. Then, during the construction of G_a, an edge is placed between domains v_a and u_a in case any IPv4 address a_{i_j} in $\alpha(v_a) = A$ is present also in $\alpha(u_a)$. The ground truth problem relates to the integer r for which no objective value can be given; when $r \to \infty$, the number of false positives (i.e., addresses that are no longer associated with a domain) increases due to the dynamic nature of the global domain name system. In this paper, all domains were resolved five times, which took several days due to the large amount of sampled hosts.

2.4 Metrics

The three graphs are classical in terms of their underlying characteristics: G_t^c, G_t^m, and G_a are undirected, unweighted, and unlabeled graphs without self-loops or multiple edges. The vertex stability characteristic also allows relatively easy comparisons of the three graphs – at least when compared to the case of different vertex sets with different sizes.

Seven graph metrics are used in the empirical experiment, and six of these are well-established in applied research. The first two reflect the structure of the whole graphs. These are the (1) graph *density* (i.e., the number of observed edges to all potential edges) and the (2) *global clustering coefficient*. In general, the latter metric approximates the overall tendency of a graph to cluster through "triadic transitivity", whereas low density values indicate a sparse graph. The subsequent three metrics are all vertex-specific. These are: the (3) *degree*, (4) *closeness*, (5) *betweenness*, and (6) *transitivity*, as defined in terms of the *local clustering coefficient*. No scaling or normalization is done for any of the metrics.[1]

From the last four vertex metrics, degree is the only one with an unambiguous interpretation. In G_t^c and G_t^m, the number of adjacent domains (that is, the degree) of any vertex proxies the between-domain "popularity" of the files made available for download from the domain vertex. Note also that due to the qualification criterion (c) discussed in Section 2.3, the degree is always positive in G_t from which the two spanning subgraphs are constructed; $d(v_t) > 0$ such that either $d(v_t^c) \geq 0$ or $d(v_t^m) \geq 0$, but $d(v_t^c) + d(v_t^m) > 0$. If the degree of a v is high in G_a, on the other hand, the domain is hosted from an IPv4 address to which also many other A records are pointing to – given a sampled pool of domains that have shared unique files (i.e., given V_t in G_t).

However, the metrics (4) and (5) are based on topological distances, which are not straightforward to interpret in the present context. The metrics (3), (4), (5), and, to a lesser extent (6), are also so-called centrality metrics. Consequently, as these measure the same theoretical phenomenon, the metrics are often correlated. These correlations apply also to local clustering coefficients (transitivity), which often correlate particularly with degrees [10]. The interpretation given for the metric (6) is somewhat opaque, however. The lo-

[1] The graph density is defined as $2 \times m_i \;/\; n(n-1)$, where $m_i \in \{|E_t^c|, |E_t^m|, |E_a|\}$ and $n = |V_t^c| = |V_t^m| = |V_a|$, while the global graph metric (2) is based on a ratio of "three times the number of triangles to connected triples" (for formal definitions and general discussion see, e.g., [10]; and also the documentation provided for the implementation used [4]).

cal clustering coefficient is computed by querying for the distinct pairs of vertices that are neighbors of a given vertex, counting the pairs that are connected within this set, and finally dividing by the total number of pairs [10]. In terms of G_a, then, this local transitivity metric approximates the average probability that a domain's name-based hosting neighbors are themselves connected through shared IPv4 addresses. As this reasoning seems relatively logical, it can be expected that the values for metric (6) are higher in G_a compared to G_t^c and G_t^m for which the values are rather difficult to interpret. Nonetheless, the local clustering coefficient reflects general clustering tendencies, which are also the context for the last metric related to a theoretically defined malware cluster based on an edge-induced subgraph.

The last (7) custom metric is defined as the number of distinct unordered edge pairs in E_t^m that are present also in E_a, scaled by $|E_t^m|$. If G_{ta}^m denotes the underlying edge-induced subgraph, this representation mixes the theoretical and computer network representations by defining a partitioning of domains that share both malware and IPv4 addresses. Although a more thorough analysis could be carried out with algorithms related to graph isomorphisms [8], the subgraph G_{ta}^m provides an interesting relational grouping for actual network forensics and qualitative interpretation.

3. EXPERIMENTAL RESULTS

The empirical experiment is presented by illustrating a few descriptive correlations computed from the subsequently introduced malware dataset.

3.1 Data

The dataset is based on a snapshot from two continuously updated collections. Most of the files for G_t^c and G_t^m are based on (i) a web crawling infrastructure maintained by F-Secure for security intelligence purposes [14]. This primary data source is further augmented (ii) from a commonly used (e.g., [1]) "Clean MX" open data feed [3]. Both sources were further passed through the (iii) the aggregation site VirusTotal [13] in order to obtain the necessary information for carrying out the arithmetic in (1) and (2). In total, the raw snapshot covers over 3.4 million (M) files. From these, only about 0.1 M passed the qualification criteria noted in Section 2.3. That is, these are shared files distributed from valid domains that still resolved through live DNS in early May 2016. As can be concluded from the brief numerical summary in Fig. 1, the between-domain sharing of unique files (whether clean or malware) is rather uncommon in general.

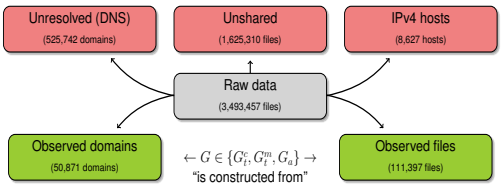

Figure 1: Sample Characteristics

The resolver (at `208.67.222.222`) provided by OpenDNS (Cisco) was used to resolve the A records required for the construction of the computer network representation G_a. As said, the size of (3) was restricted by fixing the limiting integer to $r = 5$. Because this restriction is slightly less than what is often used for DNS graph mining [15, 17], the mappings can be evaluated to be only moderate for ensuring robust placement of edges in the address graph G_a.

3.2 Results

All three graphs are rather sparse, but G_t^c is denser than G_t^m and G_a, as exemplified by the graph densities for the three graphs: $< 10^{-2}$, $< 10^{-4}$, and $< 10^{-4}$, respectively. In other words, the sharing of normal, non-malware files is generally more common than the sharing of malware files. The malware graph G_t^m attains also much lower (0.61) global clustering coefficient compared to G_t^c (0.97) and G_a (0.93). For further examining the structural (dis)similarity of the three graphs, the four vertex-based metrics (see Section 2.4) can be correlated because $|V_t^c| = |V_t^m| = |V_a|$. These correlations can be disseminated in two steps: (a) by considering "between-metric correlations" within each graph, and (b) by computing "between-graph correlations" for each metric.

1. As can be observed from Table 2, many of the vertex-based metrics correlate in all graphs, although the reported correlation coefficients are not uniform. The highest coefficients (around 0.9) are seen between degree and closeness scores, whereas betweenness shows the smallest but still visible coefficients with the other three vertex metrics. The coefficients are also higher in the malware graph G_t^m compared to the "clean" G_t^c.

2. When comparing the between-metric correlations (Table 2) to the between-graph correlations shown in the subsequent Table 3, the coefficients are much lower in the latter case. In particular, for all four metrics, the coefficients are small or even negligible between G_t^m and G_a. If a domain has shared malware files with other domains, it thus seems unlikely that this group of domains would be hosted from common addresses.

Table 2: Between-Metric Correlation Coefficients[a]

G	#	Metric	1. Deg.	2. Clo.	3. Bet.	4. Tra.
G_t^c	1.	Degree		0.894	0.205	0.641
	2.	Closeness			0.208	0.536
	3.	Betweenness				−0.305
	4.	Transitivity				
G_t^m	1.	Degree		0.998	0.434	0.793
	2.	Closeness			0.428	0.777
	3.	Betweenness				0.454
	4.	Transitivity				
G_a	1.	Degree		0.999	0.285	0.702
	2.	Closeness			0.285	0.699
	3.	Betweenness				0.294
	4.	Transitivity				

[a] Spearman rho; underlined $p < 0.05$.

The last custom metric was defined as the number of distinct edges that are present in E_t^m and E_a, scaled by $|E_t^m|$. The value obtained is 0.029, which means that only about three percent of the edges in G_t^m are present in the computer network representation G_a. Thus, the domains that have shared unique malware files have only seldom been hosted

Table 3: Between-Graph Correlation Coefficients[a]

Metric		Graph		
		G_t^c	G_t^m	G_a
1. Degree	Clean G_t^c		-0.234	-0.038
	Malware G_t^m			-0.006
	Address G_a			
2. Closeness	Clean G_t^c		-0.217	-0.133
	Malware G_t^m			-0.007
	Address G_a			
3. Betweenness	Clean G_t^c		0.231	0.256
	Malware G_t^m			0.122
	Address G_a			
4. Transitivity	Clean G_t^c		-0.232	0.146
	Malware G_t^m			0.021
	Address G_a			

[a] Spearman rho; underlined $p < 0.05$.

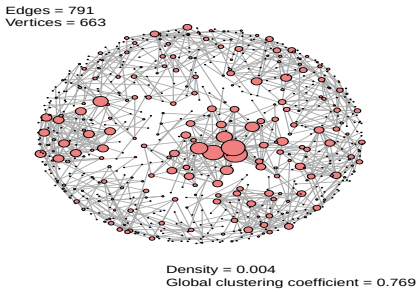

Figure 2: Malware-Address Cluster (G_{ta}^m)

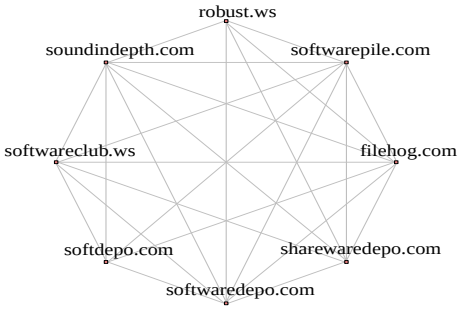

Figure 3: Domains in the Malware-Address Cluster

from the same unique IPv4 addresses. This said, the underlying malware-address subgraph, G_{ta}^m, still contains many malware-sharing domains with shared IPv4 addresses, as can be observed from Fig. 2 within which the size of the vertices is proportional to the degree of the vertices.

Supposedly, the shown cluster may have also something other in common besides the sharing of unique malware as well as IPv4 addresses. That is, this theoretically connected knit would offer a good case for further forensics related to network blocks, hosting providers, geographic origins, technical servicing solutions, and related characteristics that may explain the context behind the relational grouping. To demonstrate the potential at this front, Fig. 3 illustrates a few domains extracted from G_{ta}^m. Given that all shown domains have shared a malware file from the same IPv4, the illustration may reflect a common orchestrator, a common compromised hosting service, or some other abstraction that may explain the placement of the edges within the subgraph.

4. DISCUSSION

This exploratory empirical paper correlated a theoretically defined graph representation against a so-called ground truth representation based on DNS graphs. The questions explored were related to the sharing of unique malware files and IPv4 addresses on one hand, and the sharing of malware and clean files on the other hand. Based on corresponding file-sharing and address-sharing graph representations, the examined dataset indicated no visible evidence that the sharing of addresses would increase the between-domain sharing of malware. Although neither the dataset nor the presented evidence allows to draw definite conclusions, it seems that sharing of IPv4 addresses is unlikely to generally explain the sharing of files, whether these are malware or normal files. This said, the elaborated construction of the subgraphs allows to define theoretical groupings that also cluster empirically. These analytical groupings can be used in further data mining and more practical network forensics tasks that operate with DNS and related protocols.

A few limitations should be also remarked. First, the operationalization and graph construction (see Section 2.3) is statistically prone to include false positives due to the strict choice in (1). As this shortcoming is empirically testable, the observations should be assessed also in terms of varying detection rates, among other related quantities. Second, the file-sharing representation did not utilize edge weights in terms of the number of shared files; hence, nothing can be concluded about the actual volume of unique files shared between domains. Third, the sharing of distinct files was operationalized according to MD5 hashes, which leads to problems due to the continuous evolution of malware and software in general. The fourth concern relates to the deliberate absence of a rigorous longitudinal evaluation; only a data snapshot was evaluated. This concern extends to the dynamic mappings between domain names and addresses, although such dynamics are a lesser concern because most of the observed domains have only one A record on average (see Fig. 4). Thus, rather, the longitudinal concern relates to the time gap between the actual data collection and the post-collection use of data from the live DNS database.

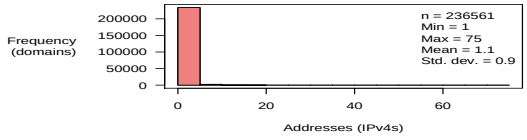

Figure 4: Unique Addresses per Domain

The last and arguably most important concern relates to the theorization and interpretation of what is being observed (a.k.a. construct validity). Although the paper was loosely motivated by security issues in virtual hosting [9], the observed dataset is insufficient for observing the actual network infrastructures and hosting solutions behind the sharing of IPv4 addresses. Nor is it possible to speculate whether and

how the distribution of malware is associated with the likelihood that a site has been compromised. To some extent, malware distribution is likely to increase the likelihood, although many malware-distributing domains are entirely legitimate, benign, and secure [14]. These fundamental issues provide also worthwhile topics for further research.

In terms of further scholarly research directions in the field of DNS graph mining, on the other hand, the presented graph representation is beneficial over the labeled variants [11, 12, 15, 17] due to the vertex stability feature. For instance, many dynamic models for graph evolution build on this particular feature [16]. It would be also interesting to pursue further the question of how a file-based or a DNS-based graph evolves through the addition, deletion, and placement of edges between a fixed sample of vertices.

Acknowledgements

The authors gratefully acknowledge Tekes – the Finnish Funding Agency for Innovation, DIMECC Oy, and the Cyber Trust research program for their support. The authors would also like to thank F-Secure and the group behind "Clean MX" for supplying malware data, and Rotarua Limited (d.b.a. VirusTotal) for enabling further data processing.

5. REFERENCES

[1] M. Akiyama, T. Yagi, K. Aoki, T. Hariu, and Y. Kadobayashi. Active Credential Leakage for Observing Web-Based Attack Cycle. In S. J. Stolfo, A. Stavrou, and C. V. Wright, editors, *Proceedings of the 16th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2013), Lecture Notes in Computer Science (Volume 8145)*, pages 223–243, Rodney Bay, 2013. Springer.

[2] M. Alsaleh and P. C. van Oorschot. Evaluation in the Absence of Absolute Ground Truth: Toward Reliable Evaluation Methodology for Scan Detectors. *International Journal of Information Security*, 12(2):97–110, 2013.

[3] Clean MX. Realtime Database. Data feed available online in April 2016: http://support.clean-mx.de/clean-mx/viruses, 2016.

[4] G. Csárdi and T. Nepusz. The igraph Software Package for Complex Network Research. InterJournal, Complex Systems CX.18. Available online in June 2014: http://www.interjournal.org/manuscript_abstract.php?361100992, 2006.

[5] M. Dusi, F. Gringoli, and L. Salgarelli. Quantifying the Accuracy of the Ground Truth Associated with Internet Traffic Traces. *Computer Networks*, 55(5):1158–1167, 2011.

[6] K. Fiveash. Linux Mint Hit by Malware Infection on Its Website, Forum After Hack Attack: "We Don't Know Motivation Behind This", Says Distro Creator. Ars Technica, February 22, 2016, available online in May 2016: http://bit.ly/24moogm, 2016.

[7] X. Han, N. Kheir, and D. Balzarotti. The Role of Cloud Services in Malicious Software: Trends and Insights. In M. Almgren, V. Gulisano, and F. Maggi, editors, *Proceedings of the 12th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2015), Lecture Notes in Computer Science (Volume 9148)*, pages 187–204, Milan, 2015. Springer.

[8] J. Kinable and O. Kostakis. Malware Classification Based on Call Graph Clustering. *Journal of Computer Virology*, 7(4):233–245, 2011.

[9] S. A. Mirheidari, S. Arshad, S. Khoshkdahan, and R. Jalili. A Comprehensive Approach to Abusing Locality in Shared Web Hosting Servers. In *Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2013)*, pages 1620–1625, Melbourne, 2013. IEEE.

[10] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, Oxford, 2010.

[11] E. Nissan. An Overview of Data Mining for Combating Crime. *Applied Artificial Intelligence*, 26(8):760–786, 2012.

[12] B. Rahbarinia, R. Perdisci, and M. Antonakakis. Efficient and Accurate Behavior-Based Tracking of Malware-Control Domains in Large ISP Networks. *ACM Transactions on Privacy and Security*, 19(2):4:1–4:31.

[13] Rotarua Limited (d.b.a. VirusTotal). VirusTotal. Available online in April 2016: https://virustotal.com/, 2016.

[14] J. Ruohonen, S. Šćepanović, S. Hyrynsalmi, I. Mishkovski, T. Aura, and V. Leppänen. A Post-Mortem Empirical Investigation of the Popularity and Distribution of Malware Files in the Contemporary Web-Facing Internet. In *Proceedings of the European Intelligence and Security Informatics Conference (EISIC 2016)*, Uppsala, 2016. IEEE.

[15] J. Ruohonen, S. Šćepanović, S. Hyrynsalmi, I. Mishkovski, T. Aura, and V. Leppänen. The Black Mark Beside My Name Server: Exploring the Importance of Name Server IP Addresses in Malware DNS Graphs. In *Proceedings of the Third International Symposium on Social Networks Analysis, Management and Security (SNAMS 2016)*, Vienna, 2016. IEEE.

[16] T. A. B. Snijders, G. G. van de Bunt, and C. E. G. Steglich. Introduction to Stochastic Actor-Based Models for Network Dynamics. *Social Networks*, 32(1):44–60, 2010.

[17] M. Stevanovic, J. M. Pedersen, A. D'Alconzo, S. Ruehrup, and A. Berger. On the Ground Truth Problem of Malicious DNS Traffic Analysis. *Computers & Security*, 55:142–158, 2015.

[18] S. Tzur-David, K. Lashchiver, D. Dolev, and T. Anker. Delay Fast Packets (DFP): Prevention of DNS Cache Poisoning. In M. Rajarajan, F. Piper, H. Wang, and G. Kesidis, editors, *Proceedings of the 7th International ICST Conference (SecureComm 2011), Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (Volume 96)*, London, 2012. Springer.

[19] J. Yang and J. Leskovec. Structure and Overlaps of Ground-Truth Communities in Networks. *ACM Transactions on Intelligent Systems and Technology*, 5(2):26:1–26:35, 2014.