Graph-Theoretic Characterization of Cyber-threat Infrastructures

By

Amine Boukhtouta, Djedjiga Mouheb, Mourad Debbabi, Omar Alfandi, Farkhund Iqbal and May El Barachi

Contents lists available at ScienceDirect

Digital Investigation

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

DFRWS 2015 US

Graph-theoretic characterization of cyber-threat infrastructures

Amine Boukhtouta [a, *], Djedjiga Mouheb [a], Mourad Debbabi [a], Omar Alfandi [b, c], Farkhund Iqbal [b], May El Barachi [b]

[a] *Computer Security Laboratory, Concordia University, NCFTA-Canada, Montréal, Québec, H3G 1M8, Canada*
[b] *Zayed University, Khalifa City B, P.O. Box 144534, Abu Dhabi, United Arab Emirates*
[c] *University of Göttingen, Wilhelmsplatz 1, 37073 Göttingen, Germany*

Keywords:
Malware Analysis
Cyber-threat infrastructure
Graph theory
Graph fingerprinting
Cyber-threat Characterization

A B S T R A C T

In this paper, we investigate cyber-threats and the underlying infrastructures. More precisely, we detect and analyze cyber-threat infrastructures for the purpose of unveiling key players (owners, domains, IPs, organizations, malware families, etc.) and the relationships between these players. To this end, we propose metrics to measure the badness of different infrastructure elements using graph theoretic concepts such as centrality concepts and Google PageRank. In addition, we quantify the sharing of infrastructure elements among different malware samples and families to unveil potential groups that are behind specific attacks. Moreover, we study the evolution of cyber-threat infrastructures over time to infer patterns of cyber-criminal activities. The proposed study provides the capability to derive insights and intelligence about cyber-threat infrastructures. Using one year dataset, we generate notable results regarding emerging threats and campaigns, important players behind threats, linkages between cyber-threat infrastructure elements, patterns of cyber-crimes, etc.

Introduction

Nowadays, many cyber-criminals make use of network resources to conduct their malicious activities. For instance, they set up networks known as botnets, to perpetrate attacks against corporations or Internet users. A leading server, known as botmaster, instructs malware-infected machines to steal data, perform reconnaissance, launch DDoS attacks, etc. Some malware steal sensitive information and send it to depots of stolen information managed by cyber-criminals. The collected information is used by criminals for their own benefit, such as using credit card numbers to purchase goods for reselling. To ensure the stealth of botnet activities, criminals make use of DNS protocol. They use fast-fluxing and dynamic DNS techniques to change domain names and resolving IPs. Thus, they manage to avoid malicious IPs and domains being blacklisted. These techniques allow cyber-threat infrastructures to expand through many ISPs and registrars, resulting in a platform to orchestrate malicious activities.

In this paper, we investigate cyber-threats and the underlying infrastructures. In particular, we answer the following questions: (1) What are the elements of a cyber-threat infrastructure and their inter-relationships? (2) What are the infrastructures used by cyber-criminals to perpetrate attacks? (3) What are the most important players in terms of organizations and people that are contacts for registered malicious domains and IP addresses? (4) How cyber-threat infrastructures evolve over time? To

* Corresponding author.
E-mail address: a_boukh@encs.concordia.ca (A. Boukhtouta).

http://dx.doi.org/10.1016/j.diin.2015.05.002

answer the aforementioned questions, we analyze and correlate the different sources of malware, threat and network data such as dynamic malware analysis reports, passive DNS, Whois and antivirus engines databases. We extract and gather various threat-related information such as malicious domains/IPs, owners, organizations, registrars, etc. This information is used to conduct an in-depth analysis of cyber-threat infrastructures by employing graph theory concepts. The use of graph theory is of crucial importance since it offers the characterization of the interaction between the infrastructure actors and the identification of influential elements or groups. In addition, it provides a structure for studying how graphs evolve over time.

We identify cyber-threat infrastructures by examining the connectivity of graph components. In addition, we discover the most important players in cyber-threat infrastructures using metrics that measure the badness of different infrastructure elements through the computation of centrality concepts (e.g., degree centrality, betweenness centrality, closeness centrality) and node influence concepts, namely Google PageRank (Brin and Page (1998)). This is achieved by assigning a score and rank to each element within the infrastructure. Moreover, we study the evolution of cyber-threat infrastructures over time by computing graph similarities. To this end, we use the so-called graph kernels (Ralaivola et al. (2005); Vishwanathan et al. (2010); Gärtner (2003)) and min-hash fingerprinting (Teixeira et al. (2012)), which is an effective technique for comparing large-scale graphs. The main contributions of our work are the following. First, we define the different elements of cyber-threat infrastructures and their relationships. Second, we put forward a methodology to investigate cyber-threat infrastructures. More precisely: (1) We identify the most important players (e.g., owners, domains, IPs, organizations) by measuring the badness of different infrastructure elements. (2) We quantify the sharing of infrastructure elements between different malware samples and families. The sharing lies within intra- and inter-malware families. This is of paramount importance, as it helps to identify the groups that are behind the appearance of malware samples. (3) We study the evolution of cyber-threat infrastructures over time to infer patterns.

The proposed methodology provides the capability to derive important intelligence about cyber-threat infrastructures. We have applied our methods on one-year dataset to generate several statistics and insights regarding emerging threats and campaigns, important players behind cyber-threats, linkages between cyber-threat infrastructure elements, patterns of cybercrimes, etc. These insights are relevant to law enforcement agencies to establish a situational awareness of cyber-threat infrastructures, devise appropriate strategies and set priorities for takedowns.

The remainder of this paper is organized as follows: In Approach Section, we describe our approach to investigate cyber-threat infrastructures. In Experimental Results Section, we provide statistics and insights generated from the analysis of cyber-threat infrastructures. Related works are introduced in Related Work Section. Finally, we conclude with a discussion and future works in Conclusion Section.

Approach

In this work, we develop a framework to collect insights and intelligence out of dynamic malware analysis. Malware samples tend to exhibit a cooperative strategy with remote malicious domains and IPs to perpetrate malicious activities, e.g., stealing credentials, spam propagation, advanced DDoS attacks, etc. In the light of these facts, we design and integrate an approach to generate cyber-threat intelligence for the purpose of identifying the infrastructures used by malware to threaten the cyber-space. Our approach to generate cyber-intelligence is depicted in Fig. 1. The approach falls into: (1) data collection, (2) cyber-threat graph generation, (3) descriptive statistics, (4) badness scoring and (5) patterns inference.

Data collection

We collect malware samples on a daily basis from a trusted third party ThreatTrack (2015). These malware samples are analyzed through a sandbox technology to monitor malware behavior on either physical or virtual machines. The malware behavior is stored in XML reports. We usually manage to get an average of 45,000 malware reports per day. For each report, we extract the domains visited by malware samples, IPs resolving to these domains and IPs directly connected by malware through FTP, SMTP, IRC servers as well as plain UDP and TCP connections. Furthermore, we use VirusTotal malware naming schema to get malware family information out of 54 anti-virus engines. In addition, we use Whois database to get domains and IPs records. The intent is to gather domains' owners, organizations, registrars, physical addresses, networks and name-servers.

Cyber-threat graph generation

Li (2014) introduced the intelligence collection process based on different data sources. Being inspired by this concept, we use data collected from dynamic malware analysis and Whois database to define a cyber-threat infrastructure, which is composed as a set of components involved in malware activities (Fig. 2). The components of a cyber-threat infrastructure are: malware, domains, IP addresses, FTP servers, SMTP servers, IRC channels, timestamps, organizations, registrars, technical people, administrative people and domain owners. The interaction between the infrastructure components can be stated as follows: A malware tends to visit domains, which can be command and control servers (C&Cs) or redirections of legitimate domains to malicious proxies. These domains are resolved to IPs. They also usually have second-level domains. On the other hand, malware can connect to FTP servers to upload stolen information or download other malware binaries. Malware can also connect to SMTP servers to conduct spamming activities or IRC channels to interact with IRC botnets. They can also connect directly through non-conventional TCP and UDP protocols for the purpose of cooperating with infected machines or C&Cs. FTP/SMTP servers and IRC channels can be hosted within a second-level domain server. The latter is registered within

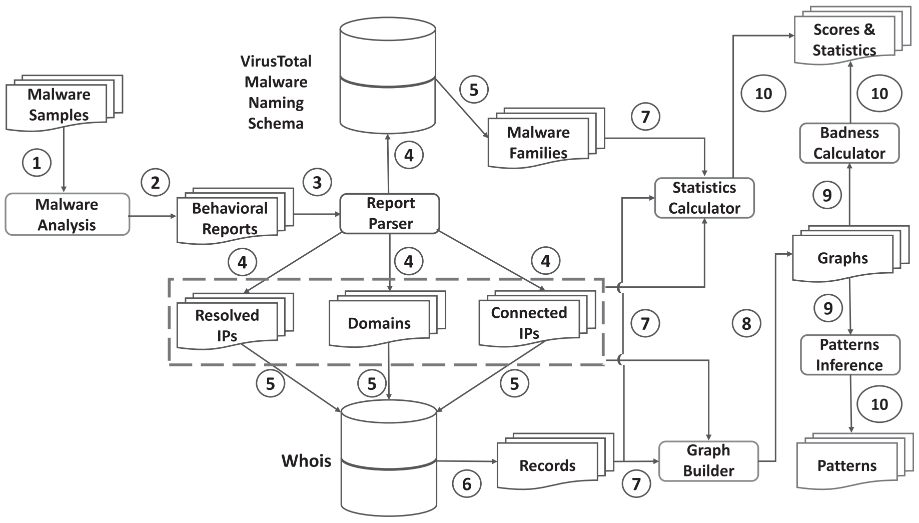

Fig. 1. Approach overview.

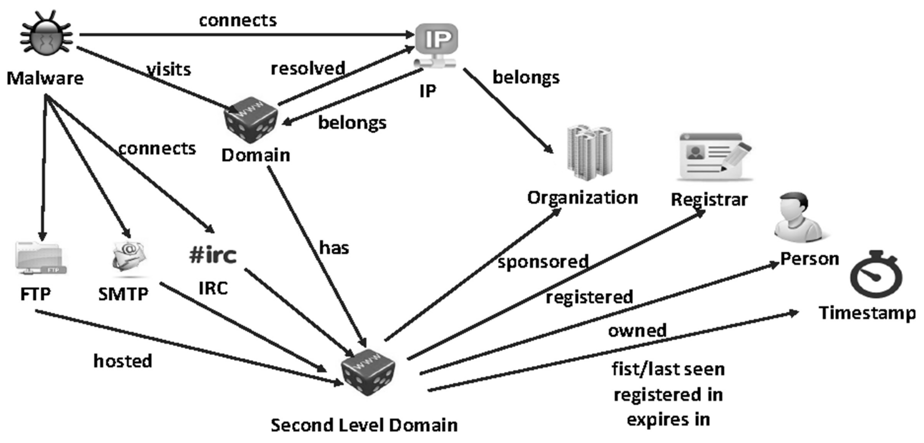

Fig. 2. Cyber-threat infrastructure schema.

a registrar and sponsored by an organization. It can have an administrative contact, a technical support contact and an owner. Each domain or a second-level domain has a creation timestamp, expiration timestamp and passive DNS first/last seen timestamps. An organization can have more than one IP block and be located in different countries.

We represent cyber-threat infrastructures as a complex network of *directed graphs*. The vertices of the graph represent components of cyber-threat infrastructures, i.e., malware, domains, IPs, FTP/SMTP servers, IRC channels, organizations, registrars, technical/administrative people, domain owners and physical addresses. Collection timestamps are properties of malware nodes, whereas first/last seen timestamps, creation and expiration timestamps are properties of second-level domains.

Fig. 3 illustrates a directed graph representing a cyber-threat infrastructure. The red vertices represent malware samples connecting to domains (blue vertices). Both of these domains resolve to the same IP address (yellow vertex)(in the web version). Owners, organizations and registrars are represented by green vertices.

The increasing number of vertices appearing in cyber-threat infrastructures make their analysis a complex task. Fig. 4 depicts the evolution of five days cyber-threat infrastructures. To overcome the complexity of cyber-threat graphs, we use a graph abstraction technique, where we decompose heterogeneous directed graphs (vertices representing many types) into homogenous weighted graphs. To illustrate the abstraction, we consider the case of malware samples sharing two domains. Initially, each vertex v_i, representing a malware sample, is linked to two vertices, v_{d1} and v_{d2}, representing visited domains. This sub-graph is abstracted to two linked vertices v_{d1} and v_{d2}, representing domains. The edge between v_{d1} and v_{d2} is labeled with the number of malware shared by these domains. Fig. 5 depicts the abstraction of cyber-threat infrastructure graphs. By performing abstraction, we generate the following sub-graphs: (1) *Domain-Malware graph*: Domains are linked if

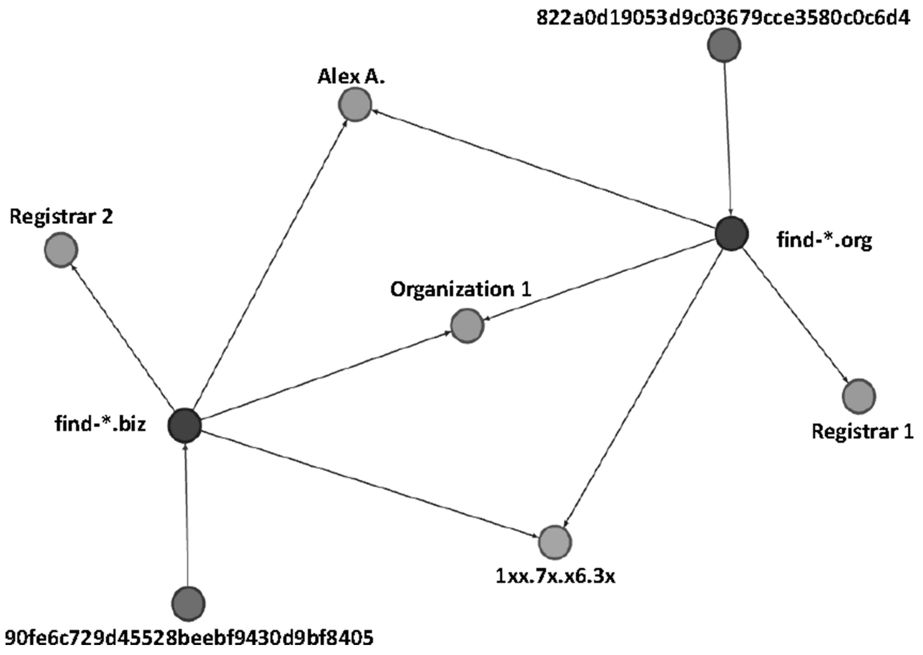

Fig. 3. Example of a cyber-threat infrastructure.

they are visited by shared malware samples. (2) *Domain-IP graph*: Domains are linked if they resolve to shared IPs. (3) *IP-Malware graph*: IPs are linked if shared malware samples connect to. (4) *Owner-Malware graph*: Owners are linked if they own domains that are visited by shared malware samples. (5) *Owner-Physical address graph*: Owners are linked if they register different domains with the same physical addresses. (6) *Organization-Malware graph*: Organizations are linked if they have IPs connected by shared malware samples.

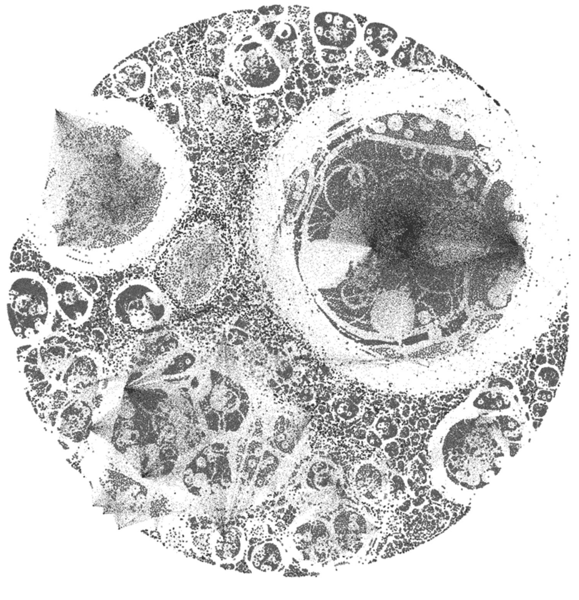

Fig. 4. Components representing Cyber-Threat Infrastructures.

Badness scoring

In this research effort, we put an emphasis on finding *what are the key players in cyber-threat infrastructures*. The importance of vertices in a network graph is known as *vertex's centrality*. The latter represents a real-valued function produced to provide a ranking, which identifies the most important nodes (Borgatti (2005)). Despite the fact that different centralities, namely, degree centrality (Sabidussi (1966)), closeness centrality (Stephenson and Zelen (1989); Dangalchev (2006)), betweenness centrality (Freeman (1977)), and Eigenvector centrality (Bonacich (2007)), are widely used in the analysis of different social networks, we are mostly interested in evaluating the importance or influence of different actors in cyber-threat infrastructures. For this purpose, some algorithms have been defined, such as, Hypertext Induced Topic Search (HITS) algorithm (Kleinberg (1999)) and Google's PageRank algorithm (Brin and Page (1998)). In our approach, we adopt Google's PageRank algorithm due to its efficiency, feasibility, less query time cost, and less susceptibility to localized links (Grover and Wason (2012)). In the sequel, we briefly introduce the PageRank algorithm and the random-surfer model.

Definition 1. (PageRank). Let $I(v_i)$ be the set of vertices that link to a vertex v_i and let $deg_{out}(v_i)$ be the out-degree centrality of a vertex v_i. The PageRank of a vertex v_i, denoted by $PR(v_i)$, is provided in Eq. (1):

$$PR(v_i) = d\left[\sum_{v_j \in I(v_i)} \frac{PR(v_j)}{deg_{out}(v_i)}\right] + (1-d)\frac{1}{|D|} \tag{1}$$

*I*n the aforementioned formula, the constant d is called damping factor. Its value is generally assumed to be set to

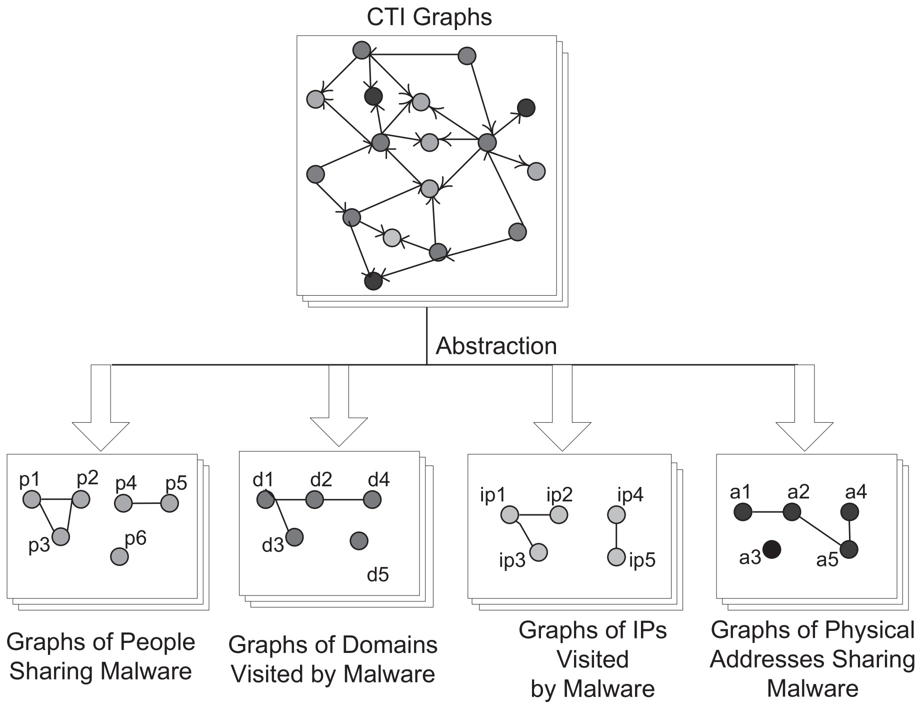
CTI Graphs
Abstraction
p1
p2
p3
p4
p5
p6
d1
d2
d3
d4
d5
ip1
ip2
ip3
ip4
ip5
a1
a2
a3
a4
a5
Graphs of People Sharing Malware
Graphs of Domains Visited by Malware
Graphs of IPs Visited by Malware
Graphs of Physical Addresses Sharing Malware

Fig. 5. CTI graphs abstraction.

0.85 (Brin and Page (1998)). From Eq. (1), we can have one equation per vertex v_i with an equal number of unknown $PR(v_i)$ values. Assuming that the PageRank values $PR(v_i)$ sum up to 1 ($sum_{i=1}^{n} PR(v_i) = 1$), then the PageRank algorithm tries to find out iteratively different PageRank values. This algorithm has been developed intuitively considering a user surfing the Web, starting from a web page and randomly visiting another web page through a link. If the user is on page v_j with a probability d (damping factor), then the probability for this user to visit another page v_i is equal to $1/deg_{out}(v_j)$. With a probability of $1-d$, the user will stop following links and pick another random page in V. Since the web-surfing process shows randomness, the authors of the PageRank algorithm claim that the PageRank values can be computed through a stochastic process. Thus, a stochastic transition matrix W is defined. The vertices ranking values are computed as expressed in Eq. (2):

$$\overrightarrow{PR} = d\left[W.\overrightarrow{PR}\right] + (1-d)\frac{1}{|D|}\overrightarrow{1} \tag{2}$$

The stochastic matrix W is defined as follows:

$$w_{ij} = \frac{1}{deg_{out}(v_j)} \text{if a vertex is linked to } v_i$$
$$w_{ij} = 0 \text{ otherwise}$$

The notation R→ stands for a vector where its i_{th} element is $PR(v_i)$ (PageRank of v_i). The notation $\overrightarrow{1}$ stands for a vector having all elements equal to 1. The computation of PageRank values is done iteratively by defining a convergence stopping criterion ε. At each computation step t, a new vector $(\overrightarrow{PR}, t)$ is generated based on previous vector values $(\overrightarrow{PR}, t-1)$. The algorithm stops computing values when the condition $\left|(\overrightarrow{PR}, t) - (\overrightarrow{PR}, t-1)\right| < \varepsilon$ is satisfied. In our case, since graphs are abstracted to weighted undirected graphs, the out-degree centrality of a vertex v_i is similar to the degree centrality. However, the weights of edges for each vertex are normalized with values between 0 and 1. The definition of the stochastic matrix W is slightly changed to:

$$w_{ij} = e_{ij} \times \frac{1}{deg_{out}(v_j)} \text{ if a vertex is linked to}$$
$$w_{ij} = 0 \text{ otherwise}$$
$$e_{ij} : \text{edge}(v_i, v_j) \text{ (normalized weight value)}$$

The reason behind using PageRank algorithm to compute badness of vertices lies in: (1) Scores are computed through a stochastic approach, which reflects randomness in the evolution of a model. With respect to cyber-threat infrastructures, we assume that there exists a random evolution, on a daily basis, in the appearance of malware samples, domains, IPs, servers, organizations, owners and registrars. Such appearance of new vertices impacts the evolution of badness scores. (2) The random web-surfer model illustrates how web pages can be accessed with a probability value (damping factor). In analogy with cyber-threat infrastructures, the probabilistic approach is interesting since it reflects potential actions done through infected machines: A malicious domain can be visited through an infected machine, an IP address can be connected by infected machines or resolved to a malicious domain or a server, an FTP server can be used to upload stolen information, an SMTP server can be used to launch spam or phishing campaigns, an IRC channel can be used to instruct bots to launch DDoS attacks, malware propagation or other malicious activities.

Patterns inference

Here, we closely study *how cyber-threat infrastructures evolve over time*. To this end, we target the identification of discernible regularities and irregularities in such infrastructures by isolating observable patterns in the generated graphs. In cyber-threat infrastructures, a pattern is associated with possible relationships between domains, IPs, owners and organizations. To infer patterns, we compute similarities between graphs collected on a daily basis.

Computation of similarity between graphs is a challenging task especially when dealing with large-scale evolving graphs. To overcome this challenge, we resort to the so-called graph kernels (Ralaivola et al. (2005); Vishwanathan et al. (2010); Gärtner (2003)). A graph kernel is a function that computes the similarity between graphs using linear methods. However, for large-scale graphs, graph kernel methods generate vectors with high dimensions that are not easy to handle. To address this issue, graph kernel methods require an important process known as *fingerprinting* (Ralaivola et al. (2005)). The latter consists of producing compact representations, known as fingerprints, for graph structures based on the generated vectors. Graph similarities are then computed using these fingerprints. To generate graph fingerprints, we use the *min-hashing* technique. Our approach of computing graph similarities is inspired by the work published by (Teixeira et al. (2012)) who introduced a fingerprinting technique for graph kernels based on min-hashing. In the sequel, we introduce our methodology used to observe the presence of patterns and how they are inferred. First, we present the different steps to compute similarities between graphs, as illustrated in Fig. 6. These steps are: (1) decomposition, (2) vector generation, and (3) fingerprinting.

Decomposition

During this step, we decompose each graph, obtained from the abstraction process, into a set of substructures. These substructures may be obtained based on paths, cycles, trees, etc. In our approach, we decompose graphs based on paths between graph vertices. In other words, two vertices v_1 and v_2 form a substructure if there is an edge between v_1 and v_2.

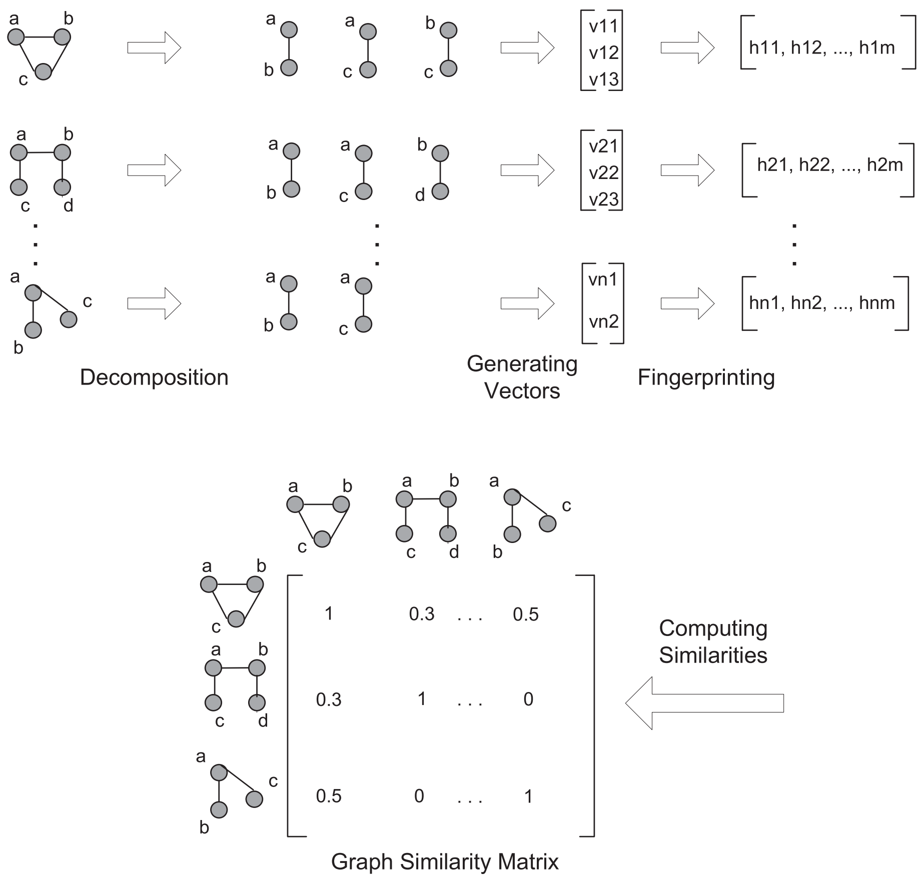

Fig. 6. Fingerprinting approach.

Vector generation

This step consists of mapping graph substructures into vectors. Algorithm 1 takes as input, a set of substructures $S(G)$ obtained from decomposing a graph G. For each substructure, we convert it into an integer value using the formula in line 3, where α is a random number, P is a big prime number such that $0<\alpha<P$, and $\mathscr{L}$ is the labelling function that returns the number of malware shared between v_i and v_j.

```
Algorithm 1: Vector Generation
Data: Set of substructures $S(G)$ from a graph $G$;
Result: Vector $V$;
1 $i := 1$;
2 for each substructure $(v_i, v_j)$ in $S(G)$ do
3     $x := (\alpha\ \mathcal{L}(v_i, v_j))\ (\text{mod P})$ ;
4     $V[i] := x$;
5     $i{+}{+}$;
6 end
```

Fingerprinting

The vectors generated from the graph substructures might have high dimensionalities for large-scale graphs. To reduce the dimensionality of a vector, we use fingerprinting to produce a compact representation that is easy to handle. To this end, we adopt the ideas presented in (Teixeira et al. (2012)) to fingerprint graphs based on min-hashing. The fingerprinting method is presented in Algorithm 2. It takes as inputs, a vector generated from the substructures of a graph G and a set of m hash functions $h_1, h_2, \ldots, h_m$. It produces as output, a fingerprint of graph G, which is a compact representation of the input vector. The fingerprint consists of a vector of m min-hash values, where m is the number of hash functions.

```
Algorithm 2: Fingerprinting
Data: Vector $V$ of a graph $G$;
      Set $H$ of hash functions $h_1, h_2, ..., h_m$;
Result: Graph fingerprint $F$;
1 for $i := 1$; $i \leq |H|$; $i++$ do
2     $F[i] := min(h_i(V))$;
3 end
```

Graph similarity computation

Graph similarities are represented through a matrix, where each value is proportional to the number of values, in the min-hash vectors, that are shared between graph pairs (Fig. 6). This matrix is important for grouping graphs with proportional similarities into groups that can be good candidates to detect patterns between corresponding min-hash values. The graph similarity matrix provides a big picture of the evolution of cyber-threat infrastructures over time. However, it needs to be leveraged to extract patterns effectively. To this end, we propose an algorithm to infer patterns as explained hereafter.

Pattern time-based inference

In order to extract patterns, we elaborate a time-based inference algorithm (Algorithm 3). This algorithm takes, as inputs, a similarity matrix, an analysis period (in terms of days), a time window (usually one day), and a density threshold to filter days where we have low similarities in the matrix. The algorithm collects common patterns, by sliding the time window through the analysis period, and checks if the similarity value is higher or equal to the density threshold. If so, it computes the intersection between patterns found on days representing the row and column index in the similarity matrix. The collected patterns are stored in a list structure that we sort at the end to collect the most or least occurring patterns.

```
Algorithm 3: Pattern Inference
Data: Similarity Matrix $M$;
      Analysis Period $t$;
      Time Window $w$;
      Threshold $th$;
Result: List of patterns $P$;
1  for $i := 1$; $i \leq t$; $i++$ do
2      for $j := i+1$; $j \leq i+w$; $j++$ do
3          if $M[i][j] \geq th$ then
4              $I := Patterns[i] \cap Patterns[j]$;
5              $P$.append($I$);
6              clear($I$);
7          end
8      end
9      sort($P$)
10 end
```

Experimental results

In the sequel, we present our analysis results of cyber-threat infrastructures. The results include descriptive statistics, badness ranking and patterns inference. It is important to mention that due to the sensitivity of the collected data, we have anonymized domains, IPs, organizations, owners and addresses. For domain names, we remove some characters at the beginning or at the middle of domain names and replace them with "*"character. For IP addresses, we replace some digits with "x" character. Regarding organizations, for each country, we replace organization names with indexed codes; for example, *ORG1(CHINA)-GD* represents an organization which has a network located in China, Guangdong city. For owners, we replace names with initials. For physical addresses, we mask some digits and letters with "*" character.

Dataset description

Our dataset consists of one year malware data collected from 25th August 2013 to 25th August 2014. Table 1 presents some statistics regarding the collected data. It is important to mention that the domains are filtered and do not contain legitimate domains that are listed in the top one million Alexa whitelist domains (Alexa (2015)).

Table 1
Dataset description.

Collected data	Statistics
Malware samples	4,717,628
Domains	9,303,378
Second-level domains	151,757
IPs that domains resolve to	240,174
IPs that malware connect to	118,270
Domain Whois records	110,414
IP Whois records	287,005

Descriptive statistics

In this section, we present some statistics that we generated from our analysis of cyber-threat infrastructures.

Domains & resolving IPs

Table 2 lists the top-10 most visited second-level domains by malware. We notice that five out of ten of these domains are legitimate. This can be explained by the fact that malware samples tend to test connectivity by visiting legitimate domains or redirecting access to legitimate domains to fake webpages. Malware also connect to legitimate domains to download vulnerable patches of operating systems or software to exploit vulnerabilities and perpetrate malicious activities.

Table 3 lists the top-10 fast-fluxing domains (domains that resolve to many IPs). We observe the presence of domains that have the same second-level domain, *nb*.com*, and resolve to many IP addresses. A main observation from this table is that fast flux and dynamic generated domain names are widely used by malware.

Table 4 illustrates the most shared resolving IPs between domains. We observe the presence of resolving IPs belonging to the same IP space. There are two IP spaces (IPs starting with "184" and "216") containing many resolving IP addresses. The presence of common IP spaces implicitly infers that cyber-criminals are prone to use an IP infrastructure to perpetrate malicious activities, or infect vulnerable IP spaces to let them be part of their botnets. The IP address "184.1xx.xxx.x6" represents the most resolved IP. By tracking associated domains, we observe that it resolves to domains dynamically generated and belonging to the same second-level domain. The domain generator associated with this IP generates a set of letters and digits. The length of the chars sequence is 30. The second-level domain is **eker.com*. Malware families associated with the top listed IP are: *antiav*, *barys*, *graftor*, *injector*, *ramnit*, *sality*, *slugin*, *swisyn*, *symmi*, *vbinject*, *virut* and *zusy*.

Table 5 lists the number of resolving IPs per network name belonging to organizations. We observe that China has the highest number of resolving IPs since 9 out of 10 network names spread throughout different Chinese regions. There is only one American network name that is present in the top-10 of networks containing resolving IPs.

Connected IPs

Table 6 lists the number of malware that connect to IPs. Connected IPs are directly accessed by malware through conventional protocols (e.g., FTP, IRC, and SMTP) and unconventional TCP and UDP ports. In contrast to resolving IPs, connected IPs are spread through many IP spaces. We

Table 3
Domain vs. Number of Resolving IPs.

Domain	# IPs
j.nb*.com	23,021
ip*.33*.org	10,779
f.nb*.com	10,313
i.nb*.com	7,130
g.nb*.com	5,825
*sopuli.*to.org	4,300
h.nb*.com	4,232
e.nb*.com	3,963
router.bi*.com	3,573
*lytics.com	3,342

Table 4
Resolving IPs vs. Number of domains.

IP	# Domains
184.1xx.xxx.x6	171,388
199.xxx.xx.xx0	125,454
184.x7x.xxx.xx5	90,766
184.x7x.xxx.xx0	84,296
184.x7x.xxx.xx8	82,104
216.2xx.xxx.x5	21,402
216.2xx.xxx.x1	20,521
46.xx.xxx.x0	13,606
162.xxx.x.xx4	7,410

Table 2
Domains vs. Number of Malware.

2*nd* Level domain	# Malware
*il.ru	252,358
*entre.ru	194,749
*soft.com	190,327
*update.com	166,995
*admr.com	160,123
cloud*.net	137,883
*lytics.com	119,233
*box.net	113,619
*host2.com	110,373
*tal.com	106,817

Table 5
Network name vs. Number of resolving IPs.

Network	# IPs
ORG1(CHINA)-GD	12,880
ORG1(CHINA)-JS	7,245
ORG2(CHINA)-SD	5,113
ORG2(CHINA)-HA	3,745
ORG2(CHINA)-HE	3,597
ORG1(USA)-2011L	3,255
ORG1(CHINA)-SC	3,232
ORG1(CHINA)-FJ	3,001
ORG1(CHINA)-HB	2,844
ORG2(CHINA)-LN	2,815

Table 6
Connected IP vs. Number of Malware.

Domain	# Malware
93.xxx.xx.xx0	11,553
65.xx.xx.xx7	4,497
219.xxx.x.xx7	4,097
113.xx.xxx.xx6	3,223
95.xxx.xx.xx3	2,719
124.xxx.xxx.6	2,609
147.xxx.xxx.x7	2,498
69.xxx.xx.x0	2,429
89.xxx.xx.xx4	2,253
125.xx.xxx.x4	2,139

observe that all the listed IPs are from different IP spaces. This can be explained by the absence of fast-fluxing. Connected IPs are dedicated to be a depot of stolen information, a spamming server, IRC channel, or a nest of other malware samples ready to be downloaded. The top listed connected IP has been associated with a VPN anonymity service. We observe a high interaction with this IP and the following malware families: *antiav*, *barys*, *esfury*, *hype*, *injector*, *navipromo*, *pirminay*, *ramnit*, *slugin*, *swisyn*, *symmi*, *vbinject*, *virut*, *vundo*, and *zbot*.

Table 7 illustrates the number of connected IPs within top-10 network names. We observe that 8 out of 10 network names are located in Asian countries. The number of organizations is 9, among which 3 are Chinese organizations, 2 are Korean and 2 are American organizations. However, top ranking network names belong to a Malaysian and an Indian organization respectively. The Malaysian network name is associated with 3 malware samples, namely, a variant of *zlob*, a variant of *zbot*, and a variant of *proxyTroj*. The latter is a proxy Trojan, which infects computers to play the role of a Command & Control and bot at the same time. Usually, such malware variants tend to communicate and cooperate through infected machines.

Whois information

Table 8 lists the different registrants and the corresponding number of domains. We notice that people behind suspicious domains use privacy services to protect their identities. Thus, there exists a big number of domains that share the same private registrants. In the aforementioned table, 12 out of 15 registrants are protected by privacy services and one registrant has a regular name (*E*l Y.*).

Table 7
Network name vs. Number of connected IPs.

Network	# IPs
ORG1(MALAYSIA)-HSDPA	10,093
ORG1(INDIA)-SouthZone	2,176
ORG1(CHINA)-GD	1,089
ORG1(KOREA)	837
ORG1(USA)-2011L	739
ORG3(CHINA)	681
ORG1(CHINA)-JS	604
ORG2(KOREA)	472
ORG4(CHINA)	358
ORG2(USA)	309

Table 8
Registrant vs. Number of domains.

Registrant	# Domains
Registration private	2,983
Whoisguard protected	744
Domain administrator	632
Domain admin	451
Whois Agent	378
Perfect Privacy LLC	274
E*l Y.	187
Whois Privacy Protection Service	184
Private Registrant	163
Oneandone Private Registration	123
Spy Eye	120
This domain for sale toll free:*-822-*	104
DNS Admin	92
Reactivation Period	92
Domain Manager	75

We observe also the presence of a registrant with the name of a well-known malware family, namely, *Spy Eye* (Sood et al. (2013)). The domains registered with *Spy Eye* have been visited by 830 malware samples, mainly belonging to the following malware families: *conjar*, *fareit*, *nebuler*, *zbot* and *zusy* (*zbot* variant). All these families represent Trojan bots involved in password stealing, downloading other malware samples, modifying system files and registry, adding startup items to systems, etc. We also notice that there are 104 domains registered with a message (*This domain for sale toll free: *-822-**). This phenomenon is known as domains' parking, where blackhat Search Engine Optimization (SEO) people are used to infect machines with malware samples to contact these domains and make them visible for different search engines. A domain is easier to sell, the more visible it is.

Table 9 lists the different physical addresses associated with registered second-level domains. We notice that 708 s-level domains are registered with an address located in Panama. This address corresponds to a privacy service that hides relevant registrants' information. This service is prone to suspicious activities since it serves spamming domains, websites of companies involved in robot-calls and scam abuses.

Badness ranking

In the sequel, we provide a description of different observations related to the computation of badness scores for IPs, domains and owners.

Table 9
Physical Address vs. Number of Registered domains.

Address	# Domains
P.O. Box****–***** Panama	708
***** Northsight blvd****9 USA	379
***** Gran bay parkway w. USA	272
***** P.O. Box** Beach Australia	228
***** Memorial Dr.#935 USA	186
Ilyinka Street** Russia	120
*****24th Street USA	115
*** Lee Road Suite**0 USA	108
*** Main street#*** USA	108
– Boulevard Massena France	87

Table 10
Top-10 domains badness scores.

Domain	Avg score $\times 10^3$
*entre.ru	5.1194617836
*box.net	3.50800756525
*spectr.ru	2.89366998268
*file.ru	2.33333105403
*sung.ru	2.17069956058
*express.ru	2.0137619806
*ldr.ru	1.6519902951
*.elb.*aws.com	1.43752913358
d1sx0cjuasqkw9.*ront.net	1.41139045489
*pro.ru	1.40483436327

Table 11
Top-10 connected IPs badness scores.

IP	Avg score $\times 10^3$
93.xxx.xx.xx0	6.90947359832
46.xxx.xxx.xx9	5.76183602875
113.xx.xxx.xx6	5.1836928519
220.xxx.xxx.7	4.53174275775
125.xx.xxx.x4	4.52513888983
124.xxx.xxx.x1	4.51910871828
221.xxx.xxx.x8	4.31768525507
91.xxx.xx.x0	3.06311160603
239.255.255.250	2.88498391036
89.xxx.xx.xx4	2.5103999827

Table 10 illustrates the top-10 average badness scores for domains observed on one year. We notice that 7 out of 10 domains have "*.ru*" extension. One of the obtained domains is dynamically generated. An interesting fact is that 5 out of 7 "*.ru*" domains are registered with the same information (registrant is known as "Private Person" and the same name servers). The number of associated malware is 830. In addition, these domains share a lot of malware families, spanning over bot Trojans and bitcoiners, mainly *badur*, *bitcoinminer*, *graftor*, *kryptik*, *loadmoney*, *minggy*, *strictor*, *symmi*, and *zusy* (zbot variant). We suspect that people belonging to the same criminal group use the same registrant information and are behind bitcoining campaigns and botnet activities.

Table 11 illustrates the top-10 average badness scores for connected IPs. The top ranked badness IP is the same leading IP "93.xxx.xx.xx0" observed in Table 6. The same observation can be made on IPs "113.xx.xxx.xx6" and "125.xx.xxx.x4", which are present in both Tables. The reason is that these IPs have maintained a badness score throughout the whole year, whereas other IPs, listed in Table 6, have not maintained their badness score as much as the IPs listed in Table 11. It is important to notice the presence of the IP "239.255.255.250", which is SSDP multicast reserved IP. This IP is mainly used by what are called Universal Plug and Play (UP&P) malware families, e.g., *conficker*, *downadup* and their variants. These malware infect other machines through vulnerabilities found in Windows server services (e.g., RPC Handling Remote Code Execution Vulnerability). Aben (2008); Messmer (2009) illustrate how UP&P devices can be used as an infection vector through SSDP protocol. Such vector of infection is still active since we observe a lot of new variants connecting to "239.255.255.250" IP address.

In Fig. 7 and Fig. 8, we present the evolution of the badness scores for the top-5 domains and connected IPs. We observe that the badness of domains has a *periodic badness persistence*. For instance, the domain "**entre.ru*" had high badness scores during the first 60 days and the domain "**box.net*" had high badness scores during the last 70 days. Similarly, the domain "**spectr.ru*" had high badness scores during a period of 40 days, the domain "**file.ru*" had high badness scores during a period of 75 days, and the domain "**sung.ru*" had high badness scores during 30 days. However, we can observe some sporadic changes in scores for all the domains (spikes after observing low badness ranking). For instance, the domain "**entre.ru*" had some score changes at days 95, 149, 334 and 345. Similarly, the domain "**spectr.ru*" had some changes of scores between day 195 and day 210 and the domain **sung.ru*" had changes of scores at days 96, 150, 335 and 346. In contrast to

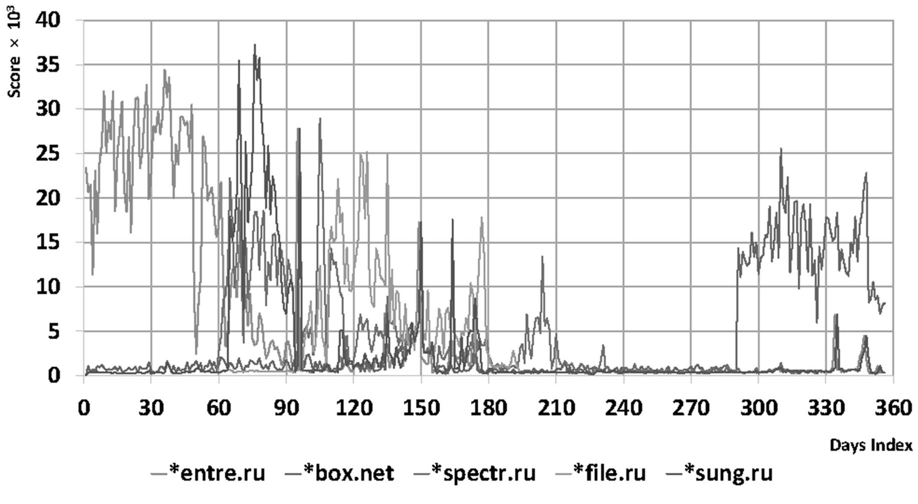

Fig. 7. Top-5 domains badness score.

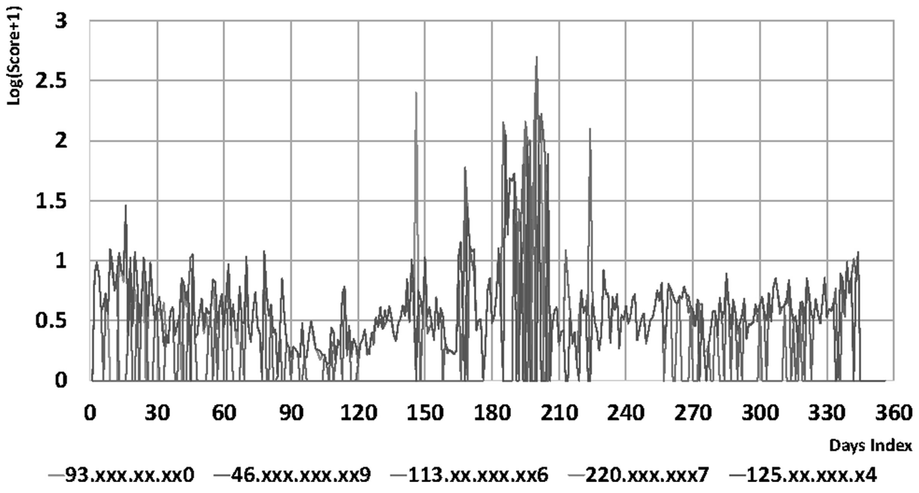

Fig. 8. Top-5 IPs badness score.

domains, connected IPs show more sporadic patterns (abrupt changes of scores). All the observed IPs had idle time periods, where their badness scores were equal to 0. For instance, the IP "93.xxx.xx.xx0" had an abrupt change in day 147 and the IP "113.xx.xxx.xx6" had an abrupt change in day 200.

Fig. 9 illustrates different owners sharing malware samples in July 2014. The colored graph network contains different communities, obtained by applying a community detection algorithm (Blondel et al. (2008)), where each color represents a community. We managed to obtain 23 communities. The graph nodes have been anonymized. We can also notice that the bigger a node is, the higher is its badness score. For instance, the person "A.Di.M" has the highest badness score (0.029), followed by the person "D.VAN.A" (score of 0.028). The third place is shared between three people, namely, "M.K.S.M", "A.K", and "A.D.de.M.S" (score of 0.026).

Patterns inference

Fig. 10 and Fig. 11 illustrate the similarity matrix obtained during one year period. A major observation is that domains have more patterns than connected IPs. Similarly to badness ranking, patterns in domains are more persistent and periodic. We observe high density in the first and last semesters in the domain patterns similarity matrix. Connected IPs show less periodicity than domains. The presence of patterns tends to be ephemeral and the maximum period is commonly in the order of 1–60 days. However, there are some IPs that have some persistence. Such case is illustrated hereafter in the pattern use cases.

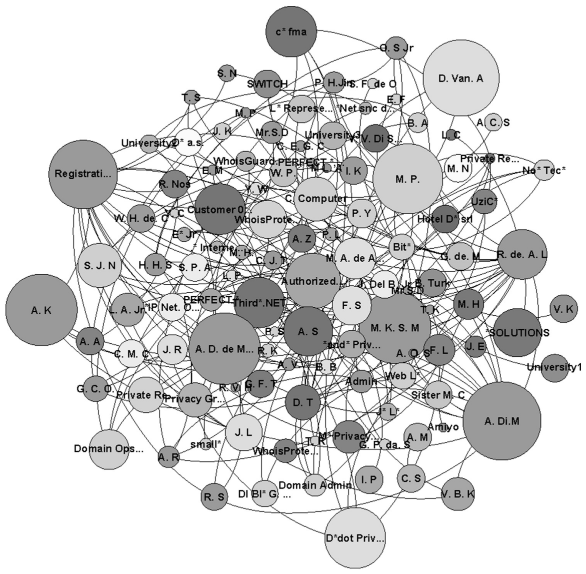

Fig. 9. Registrants communities and badness scores.

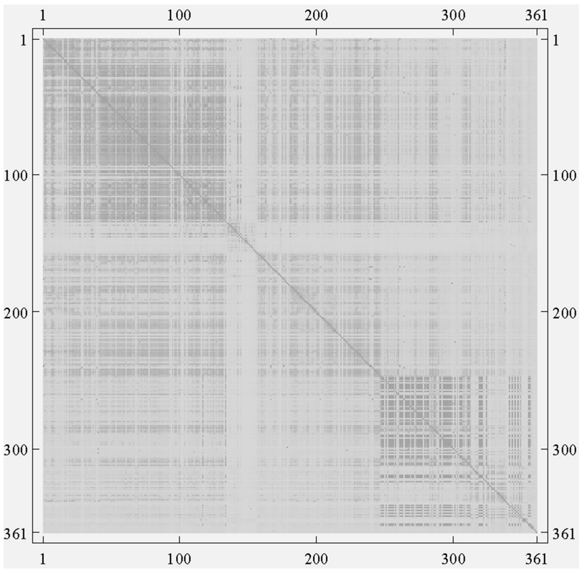

Fig. 10. Domain patterns similarity matrix.

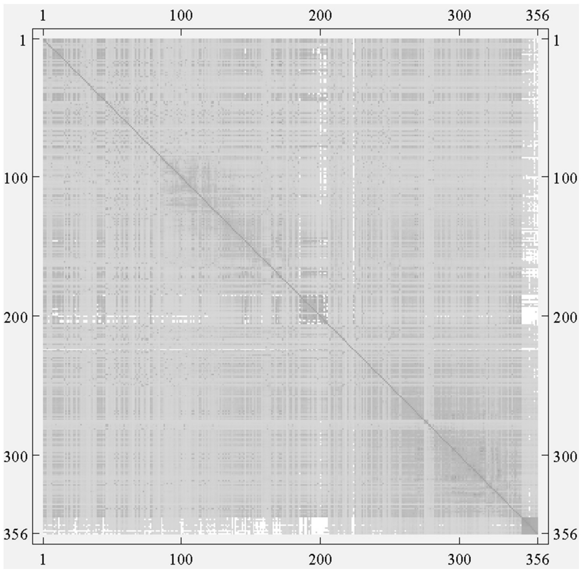

Fig. 11. IP Patterns similarity matrix.

Table 12 shows patterns involving dynamic domains generated in the same way. In this case, we have three domain names with two letters followed by two digits. All these domains share a big number of malware samples (in the order of 6,000 malware samples). In addition, they have the same owner protected by the same privacy service. Such use case acknowledges the observation found in Table 8 and i.e., the hosting companies with a privacy service are nests for suspicious domains.

Table 13 shows IP patterns connected by thousands of malware samples. These IPs are located in China, where we observe 4 organizations with 5 network names. All the patterns have appeared during long time periods: more than 300 days for the first two patterns, 289 days for the third and fourth patterns and 123 days for the last pattern. This use case indicates that there is a cluster of IP patterns that represents a collaborative malware activity in China.

Related work

Various research efforts use graph theory for the purpose of studying social media networks. Java et al. (2007) investigate microblogging phenomena through studying topological and geographical properties of Twitter's social network. Johan Ugander et al. (2011) study the structure of the Facebook social graph using different network features such as degree distribution, path length, clustering, and mixing patterns. Luca Deri and Simone Mainardi represent collected DNS with ".it" suffix data through complex graphs. They found that the Italian DNS ecosystem, represented through domain and resolver degree frequencies, follows power law distributions, and acknowledged the nature of DNS large scale evolution. In another work, Deri et al. (2013) aim to rank Internet domains based on their popularity across resolvers. Regarding threat network analysis, Nadj et al. (2013) conduct an outstanding effort to unveil the structure of criminal networks. They use DNS history of known C&Cs, IPs found in blacklists, and spam URLs to build graphs. They develop a method based on the Eigenvector metric to identify general structural trends and determine which strategy should be adopted for an effective remediation through takedown. The authors show that in many cases, by de-registering five domain names, many criminal networks can be taken down. Moreover, in one highlighted case, disabling 20% of criminal network hosts reduces the volume of successful malicious DNS lookups by 70%. Despite the interesting results shown by Nadj et al., 2013 we provide more insightful information related to cyber-threat infrastructures by including new actors such as malware families, second-level domains, organizations, owners, etc. We also focus on the study of the evolution of cyber-threat infrastructures to understand the scale and forecast the potential evolution in the near future.

Table 12
Domain patterns use case.

Patterns	Days	Mal.	Owners
f*[dd]75.com; a*[dd]75.com	332	6045	Registration Private
f*[dd]75.com; w*[dd]88	329	6046	Registration Private
w*[dd]88.com; a*[dd]75.com	317	5966	Registration Private

Table 13
IP patterns use case.

Patterns	Days	Mal.	Networks
220.xxx.xxx.7; 125.xx.xxx.x4	317	2123	ORG1-BJ; ORG5
220.xxx.xxx.7; 124.xxx.xxx.x1	311	2068	ORG1-BJ; ORG1-HE
221.xxx.xxx.x8; 125.xx.xxx.x4	289	1938	ORG6; ORG5
220.xxx.xxx.7; 221.xxx.xxx.x8	289	1948	ORG1-BJ; ORG6
124.xxx.xxx.x1; 125.xx.xxx.x4	278	1925	ORG1-HE; ORG5
124.xxx.xxx.x1; 221.xxx.xxx.x8	123	1836	ORG1-HE; ORG6

Conclusion

In this paper, we have presented an approach to investigate cyber-threats and the underlying infrastructures. To this end, we use graph-theory concepts to rank the badness of different infrastructure elements. This allowed us to identify key players and quantify the sharing among these players. This is of paramount importance as it helps to unveil potential criminal groups. Moreover, we have presented a methodology to track the evolution of cyber-threat infrastructures over time and infer patterns of cyber—criminal activities. Using one year dataset, we have derived important insights about cyber-threats. As a future work, we plan to design and integrate a near real-time cyber-threat situational awareness dashboard. To this end, we will automate the approach presented in this paper. In addition, based on the observations found in the evolution of badness scores for domains and connected IPs, we aim to assess the empirical periods to consider for domains badness persistence and IPs badness sporadicalness.

References

Aben, E., 2008. Conficker/conflicker/downadup as seen from the ucsd network telescope, http://www.caida.org/research/security/ms08-067/conficker.xml.

Alexa, 2015. https://www.alexa.com.

Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008;10. http://arxiv.org/abs/0803.0476v2.

Bonacich P. Some unique properties of eigenvector centrality. Soc Netw 2007;29(4):555–64. http://dx.doi.org/10.1016/j.socnet.2007.04.002.

Borgatti SP. Centrality and network flow. Soc Netw Jan. 2005;27(1): 55–71. http://dx.doi.org/10.1016/j.socnet.2004.11.008.

Brin S, Page L. The anatomy of a large-scale hypertextual web search engine. Comput Netw ISDN Syst Apr. 1998;30(1–7):107–17. http://dx.doi.org/10.1016/S0169-7552(98)00110-X.

Dangalchev C. Residual closeness in networks. Phys A: Stat Mech its Appl 2006;365(2):556–64. http://dx.doi.org/10.1016/j.physa.2005.12.020.

Deri Luca, Simone Mainardi MMEG. Graph theoretical models of dns traffic. In: In: 9th international wireless communications and Mobile computing conference, IWCMC; Jul 2013. p. 1162–7. http://dx.doi.org/10.1109/IWCMC.2013.6583721.

Deri L, Mainardi S, Martinelli M, Gregori E. Exploiting dns traffic to rank internet domains. In: In: IEEE international conference on communications, ICC'13; 2013. p. 1325–9. http://dx.doi.org/10.1109/ICCW.2013.6649442.

Freeman LC. A set of measures of centrality based on betweenness. Sociometry Mar. 1977;40(1):35–41. http://dx.doi.org/10.2307/3033543.

Gärtner T. A survey of kernels for structured data. SIGKDD Explor Newsl Jul. 2003;5(1):49–58. http://doi.acm.org/10.1145/959242.959248.

Grover N, Wason R. Comparative analysis of pagerank and hits algorithms. ESRSA Publications Int J Eng Res Technol Oct 2012;1:1–15., http://www.ijert.org/view-pdf/1344/comparative-analysis-of-pagerank-and-hits-algorithms.

Java A, Song X, Finin T, Tseng B. Why we twitter: understanding microblogging usage and communities. In: Proceedings of workshop on web mining and social network analysis. New York, NY, USA: ACM; 2007. p. 56–65. http://doi.acm.org/10.1145/1348549.1348556.

Kleinberg JM. Authoritative sources in a hyperlinked environment. J ACM Sep. 1999;46(5):604–32. http://doi.acm.org/10.1145/324133.324140.

Li F. Apt attributon and dns profiling. USA. Las Vegas: Presented at Blackhat; 2014., http://tinyurl.com/l3awua4.

Messmer E. Downadup/conflicker worm: when will the next shoe fall. 2009. http://tinyurl.com/nlnv633.

Nadj Y, Antonakakis M, Perdisci R, Lee W. Connected colors: unveiling the structure of criminal networks. In: Stolfo S, Stavrou A, Wright C, editors. Research in attacks, intrusions, and defenses; 2013. p. 390–410. Vol. 8145 of Lecture Notes in Computer Science. Springer, http://dx.doi.org/10.1007/978-3-642-41284-4_20.

Ralaivola L, Swamidass SJ, Saigo H, Baldi P. Graph kernels for chemical informatics. Neural Netw 2005;18(8):1093–110. http://dx.doi.org/10.1016/j.neunet.2005.07.009.

Sabidussi G. The centrality index of a graph. Psychometrika 1966;31(4): 581–603. http://dx.doi.org/10.1007/BF02289527.

Sood AK, Enbody RJ, Bansal R. Dissecting spyeye–understanding the design of third generation botnets. Comput Netw 2013;57(2):436–50. http://dx.doi.org/10.1016/j.comnet.2012.06.021.

Stephenson K, Zelen M. Rethinking centrality: methods and examples. Soc Netw Mar. 1989;11(1):1–37. http://dx.doi.org/10.1016/0378-8733(89)90016-6.

Teixeira CHC, Silva A, Meira Jr W. Min-hash fingerprints for graph kernels: a trade-off among accuracy, efficiency, and compression. JIDM 2012;3(3): 227–42. https://seer.lcc.ufmg.br/index.php/jidm/article/view/199.

ThreatTrack, 2015. http://www.threattracksecurity.com/.

Ugander Johan, Karrer Brian, Backstrom Lars. The anatomy of the facebook social graph. Comput Res Repos (CoRR) 2011;abs/1111.4503. Cameron Marlow, http://arxiv.org/abs/1111.4503.

Vishwanathan SVN, Schraudolph NN, Kondor R, Borgwardt KM. Graph kernels. J Mach Learn Res Aug. 2010;11:1201–42. http://dl.acm.org/citation.cfm?id=1756006.1859891.