Analysis of Rogue Anti-Virus Campaigns Using Hidden Structures in k-Partite Graphs*

Orestis Tsigkas and Dimitrios Tzovaras

Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
{torestis,tzovaras}@iti.gr
http://www.iti.gr

Abstract. Driven by the potential economic profits, cyber-criminals are on the rise and use the Web to exploit unsuspecting users. Indeed, a real underground black market with thousands of collaborating organizations and individuals has developed, which brings together malicious users who trade exploits, malware, virtual assets, stolen credentials, and more. Among the various malicious activities of cyber-criminals, rogue security software campaigns have evolved into one of the most lucrative criminal operations on the Internet. In this paper, we present a novel method to analyze rogue security software campaigns, by studying a number of different features that are related to their operation. Contrary to existing data mining techniques for multivariate data, which are mostly based on the definition of appropriate proximity measures on a per-feature basis and data fusion techniques to combine per-feature mining results, we take advantage of the structural properties of the k-partite graph formed by considering the natural interconnections between objects of different types. We show that the proposed method is straightforward, fast and scalable. The results of the analysis of rogue security software campaigns are further assessed by a visual analysis tool and their accuracy is documented.

Keywords: unsupervised learning, security, k-partite graphs.

1 Introduction

Over the last decade, there has been a significant shift in the nature of cybercrime, from server-side to client-side attacks and from mainly destructive (e.g. fast spreading worms) to omnivorously profit-oriented activities like identity theft, fraud, spam, phishing, online gambling, extortion [1]. It is now evident that cybercriminals become increasingly collaborative and organized, changing

* This work has been partially supported by the European Commission through project FP7-ICT-257495-VIS-SENSE funded by the 7th framework program. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

J. Pieprzyk, A.-R. Sadeghi, and M. Manulis (Eds.): CANS 2012, LNCS 7712, pp. 114–125, 2012.

the ways that cybercrimes are committed. Individuals with different skill-sets join in ephemeral relationships to commit a common act and to reproduce their skills and knowledge. All the facts and figures presented in public threat reports are certainly valuable and help to shed some light on those cyber-criminal phenomena, but a lot of unknowns remain.

Among the various malicious activities of cyber-criminals, the spreading of fake antivirus (AV) programs stands out. Fake AV software has been utilized to defraud millions of computer users into paying as for services that the never actually receive. Rogue security software is actually the most common form of scam software, also called *scareware*, which makes use of social engineering to exploit a computer user's fear of revealing sensitive information, losing important data, and/or causing irreversible hardware damage. Therefore, a fake AV program impersonates an antivirus scanner and displays misleading or fraudulent alerts in an attempt to dupe a victim into purchasing a license for a commercial version that is capable of removing non-existent security threats. However, users not only do they never receive what they have paid for, but, to make things worse, their machines get compromised by the installed software, offering new attack opportunities to cyber-attackers. As a result, fake AV software has evolved into one of the most lucrative criminal operations on the Internet.

Moreover, as cyber crime is becoming more organized, new crime mechanisms utilise all available means to automate their malicious activities. This leads to patterns or fingerprints in relevant datasets that are valuable if identified. Such identification within a large set of heterogeneous data is a very difficult and time-consuming task, particularly across layers (network transport, service, transaction). Furthermore, Internet criminals have become adept at modifying their strategies and tactics as new methods are developed to combat their activities. As such, the tools used to identify and characterise their activities must be able to cope with fast-changing requirements. In order to be successful, the techniques used to commit crimes need to be as automated as possible and, of course, stealthy. This automation, by definition, leaves fingerprints that, if found, offer valuable information for the implementation of new detection strategies or for forensic purposes. The problem is that these fingerprints are, a priori, unknown and hidden in a massive amount of data. However, current analysis techniques do not allow us to automatically discover new relevant knowledge about attack phenomena, certainly not from a strategic viewpoint.

Consequently, many open issues remain. Who is behind the deployment of rogue AV websites, how many organized communities are responsible for them, where do they originate, what are the emerging strategies used in cybercrime and how do they evolve over time? Are cyber-criminals able to coordinate their actions? All previously described issues are related to a common security problem often referred to as "attack attribution" [1]. In this paper, we present an unsupervised method for root cause analysis of rogue AV campaigns, by studying a number of different features that are related to their operation and by ascribing large-scale attack phenomena to the same group of individuals or communities.

Contrary to existing data mining techniques for multivariate data, which are mostly based on the definition of appropriate proximity measures on a per-feature basis and data fusion techniques to combine per-feature mining results, we take advantage of the structural properties of the k-partite graph formed by considering the natural interconnections between objects of different types.

The rest of the paper is structured as follows. In Section 2, we provide an overview of the background work in analysis of security software campaigns. Section 3 presents the developed method for unsupervised learning on k-partite graphs, while Section 4 provides an overview of the analysis results. Finally, Section 5 concludes the paper.

2 Background Work

The spreading of rogue security software has been observed as early as in 2005 [2]. A thorough description of various instances of rogue software is presented in security industry reports [3] [4]. These studies aim to shed light on the strategies of cyber-criminals, the prevalence of rogue AV software and its distribution mechanisms. In [5], [6] and [7], the authors provide a study of malicious websites and their underground economy. Last, in the their seminal work in [8] and [9], Cova et al. present a methodology for ascribing rogue security software websites to the same campaign. The proposed methodology requires the definition of proximity measures and clustering on a per-feature basis. Then, the per-feature clustering results are combined using a data fusion technique based on mutli-criteria decision analysis. While the presented technique yields meaningful results, its accuracy largely depends on the security analyst who has to parametrize it at various steps.

3 Clustering Analysis Using k-Partite Graphs

The proposed clustering algorithm aims at identifying the structural properties of graphs by applying dynamic methods based on the class of flow-based graph clustering algorithms represented by Markov Clustering (MCL) [10]. MCL offers several advantages in that it is an elegant approach based on the natural phenomenon of flow, or transition probability, in graphs [11]. Contrary to clustering techniques that are based on the selection of appropriate distance or dissimilarity metrics and on fusion of per-feature results, flow-based graph clustering algorithms take advantage of the similarities of data instances as these are reflected on the structural properties of the graph (e.g. common connections). Therefore, we extend flow-based graph clustering algorithms to search for natural groups of rogue websites (common campaign) in k-partite graphs, and we present a number of enhancements to improve their performance, as well as to enhance the meaningfulness of results. Their operation relies on an iterative process which applies three operators - *expansion*, *inflation* and *pruning* - on an initial transition matrix $\mathbf{P}$, in alternation, until convergence.

The merit of the proposed method compared to existing distance-based clustering methods is threefold. First, defining appropriate distance measures is a not straightforward procedure and different distance measures result in different clustering results. On the contrary, our proposed method searches for similarities rather than dissimilarities between data objects, by considering their interconnections with respect to different features. Second, in many cases defining an appropriate distance metric may not be feasible. For example, one cannot define an appropriate distance metric for two blocks of IP addresses that belong to the same ISP but are located far apart in the IP space. The proposed method does not take into account the actual IP address, but the interconnections of rogue websites with different IP addresses and, therefore, it can capture the case of different IP blocks belonging to the same ISP. Last, our method can work with categorical, numerical, ordinal and binary data without requiring complex data transformations which usually depend of the security analyst's knowledge and expertise.

The security problem of rogue AV websites analysis involves data objects of multiple types that are related to each other, which can be naturally formulated as a k-partite graph. For example, rogue websites are related to malware types, geolocation of the websites, IP address of the website and the nameserver, etc. However, the research on mining the hidden structures from a k-partite graph is still limited and preliminary. Therefore, our research work aims at proposing a principal framework for unsupervised learning on k-partite graphs of various structures. Under this model, we derive a novel algorithm to identify the hidden structures of the graph by identifying strongly connected nodes, using neighbourhood information. The strength of our approach resides in its ability to incorporate multiple features, searching for clusters in the multidimensional space.

3.1 Problem Definition

A k-partite graph is a graph where nodes can be divided in k disjoint groups $(V_0, \ldots, V_{k-1})$, such that no edge connects the vertices in the same group. More formally, a k-partite graph G is defined as $G = \langle V_0 \cup \ldots \cup V_{k-1}, E \rangle$, where $V_l = \{n_i | 1 \leq i \leq N_l\}$, $\forall l \in [0, k-1]$, and $E \subset \bigcup_{l=1}^{k-1} \{V_0 \times V_l\}$ as shown in Fig. 1.

We assume an edge-weighted directed k-partite graph. Moreover, nodes in V_0 (white circles in Fi. 1) correspond to rogue security software websites, while nodes in $V_l, l \in [1, k-1]$ (coloured circles) correspond to feature values of a specific feature. Nodes in $V_{l \neq 0}$ can have connections only to nodes in V_0.

Given a query node n_i in V_l, $l \in [0, k-1]$, our clustering algorithm computes an *attractor* node. All nodes that are attached to the same *attractor* node belong to a single cluster. Based on the graph structure, the *attractor* node can be either a rogue website or a feature value of any given feature. Nodes that belong to the same group V_l have the same type; it is the connections between the k types of

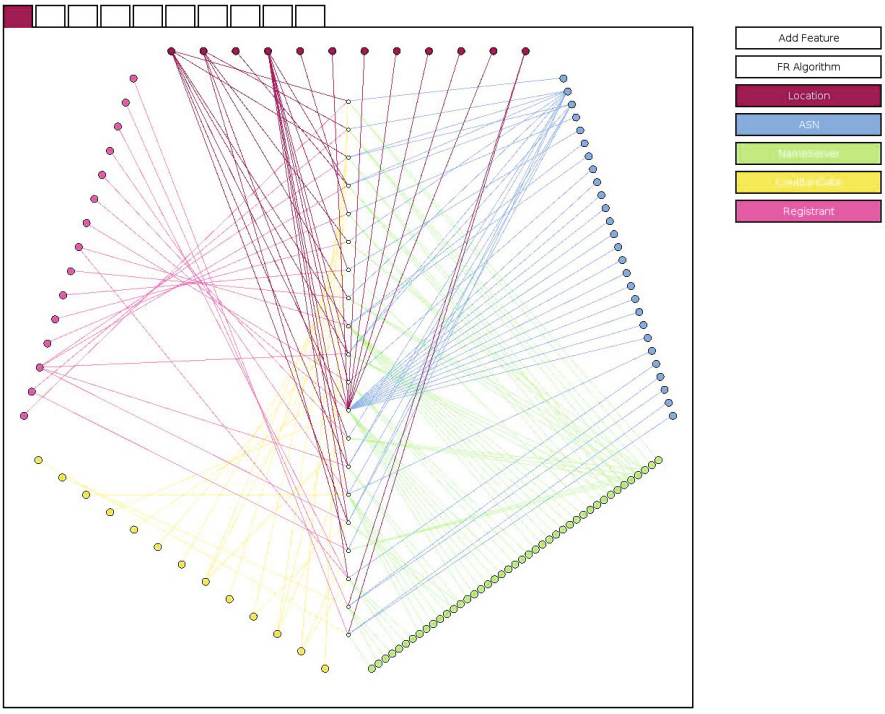

Fig. 1. A 6-partite graph. White circle nodes in the middle represent rogue security software websites. The nodes corresponding to the feature values of 5 different features are placed on the sides of a pentagon using different colouring schemes. Feature values can be connected only to rogue websites and not with each other.

objects that hold the key to mining the hidden structures in the k-partite graph. Given the natural inter-group connections (between V_l and V_m), our objective is to discover the intra-group relationships, such as the clusters within the group. An effective mining algorithm should thus be able to utilize these links across the $(k-1)$ natural groups that are formed by considering the connections between rogue websites and each of the $(k-1)$ features.

3.2 Building the *k*-Partite Graph

The subgraph G_l formed by considering nodes only in V_0 and V_l, $l \in [1, k-1]$, can be conceptually stored in a N_0-by-N_l matrix $\mathbf{M_l}$, where $M_l(i,j)$ is the weight of the edge $\langle i,j \rangle$. The nodes in $V_0(V_l)$ are called row (column) nodes. Note that a column node links to a row node if the corresponding matrix element is not zero. Moreover, row node n_i connects to another row node n_j if there is a column node c linking to both n_i and n_j. We call that path a connection between n_i and n_j through c. Nodes n_i and n_j can have multiple connections via different column nodes.

For each subgraph G_l, $l \in [1, k-1]$, we can construct the adjacency matrix $\mathbf{A_l}$ of G_l using $\mathbf{M_l}$:

$$\mathbf{A_l} = \begin{pmatrix} \mathbf{0} & \mathbf{M_l} \\ \mathbf{M_l^T} & \mathbf{0} \end{pmatrix}$$

In particular, $A_l(i,j)$ denotes the element at i-th row and j-th column in $\mathbf{A_l}$. Suppose we want to traverse the subgraph starting from the row node n_i. Then, we have to transform matrix $\mathbf{A_l}$ into a transition matrix $\mathbf{P_l}$, such that the sum of the probabilities of taking an edge $\langle i,j \rangle$, starting from the row node n_i, does not exceed 1. Therefore, the most common approach is that, for each row node n_i, the normalization of the weight of any edge $\langle i,j \rangle$ is proportional to the edge weight over all the outgoing edges from n_i. More formally:

$$P_l(i,j) = \frac{A_l(i,j)}{\sum_{m=1}^{N_l} A_l(i,m)}$$

and

$$\mathbf{P_l} = \begin{pmatrix} \mathbf{0} & \mathbf{M'_l} \\ \mathbf{M'^T_l} & \mathbf{0} \end{pmatrix}$$

Then, by considering the transition matrices $\mathbf{P_l}$ corresponding to each subgraph G_l, we can construct the transition matrix $\mathbf{P}$ of G as follows :

$$\mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{M'_1} & \mathbf{M'_2} & \dots & \mathbf{M'_{k-1}} \\ \mathbf{M'^T_1} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{M'^T_2} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{M'^T_{k-1}} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{pmatrix}$$

3.3 Clustering Formation

To find the hidden clusters in a graph we make use of Markov Clustering (MCL) algorithm which is based on (stochastic) flow simulation. This algorithm shares the ideas behind random walks. However, unlike random walks which compute a relevance score from a given node in a group to any other node in the group, MCL aims at calculating an "*attractor*" node, by which all nodes belonging to the same cluster will be attracted.

The MCL algorithm is an iterative process of applying three operators - *expansion*, *inflation* and *pruning* - on an initial transition matrix P, in alternation, until convergence. Each of these steps is defined below:

The *expansion* step requires that matrix $\mathbf{C_{N \times N}}$, which will finally hold the *attractor* nodes for each node n_i, is multiplied with the transition matrix $\mathbf{P}$:

$$\mathbf{C} = \mathbf{P} \cdot \mathbf{C} \tag{1}$$

The ith row of matrix $\mathbf{C}$ can be interpreted as the final distribution of a random walk of length 1 starting from node n_i, with the transition probabilities of the random walk given by $\mathbf{P}$.

The *inflation* step requires raising each entry in the matrix $\mathbf{P}$ to the power r and then normalizing the rows to sum to 1.

$$C(i,j) = \frac{C(i,j)^r}{\sum_{m=1}^{N} C(i,m)^r} \tag{2}$$

The inflation step has the effect of strengthening intra-cluster flow and weakening inter-cluster flow, by reducing the probability of visiting nodes that do not belong to the same cluster. This is due to the fact that there are more paths between two nodes that are in the same cluster than between those in different clusters and, therefore, there is a higher probability of visiting the inter-cluster nodes.

Last, the *prune* step removes the entries below a threshold q:

$$C(i,j) = \begin{cases} 0 & \text{, if } C(i,j) \leq q \cdot \max\limits_{j=1}^{n}\{C(i,j)\} \\ C(i,j) & \text{, otherwise} \end{cases} \tag{3}$$

Then, the retained entries are rescaled to have the row sum to 1. This step is primarily meant to reduce the number of non-zero entries in the matrix and hence save memory.

4 Experimental Results

The set of studied rogue AV domains is built by aggregating information from a number of different sources [8]. The considered dataset consists of $5,852$ DNS entries, collected in July and August 2009, pointing to $3,581$ distinct IP addresses hosting rogue AV servers. It is worth noting that at least 45% of these domains were registered through just 29 out of several hundred existing domain registrars.

To study the dynamics of rogue domains and their relation with the associated web servers, we make use of the data collected by HARMUR (HARMUR, a Historical ARchive of Malicious URLs), which enables us to study the relation between client side threats and the underlying server infrastructure, and their evolution over time [12]. The HARMUR dataset was developed by Symantec in the context of the WOMBAT EU-FP7 project [13] and extended in the framework of the VIS-SENSE EU-FP7 project [14] where Symantec is also involved in as a key partner. Instead of developing new detection technologies (e.g., based on honeyclients, or special web crawlers), HARMUR integrates multiple information sources and takes advantage of various data feeds that are dedicated to detecting Web threats. By doing so, HARMUR aims at enabling the creation of a "big picture" of the client-side threat landscape and its evolution.

4.1 Feature Selection

HARMUR collects a number of features associated with each rogue AV domain described in the following list:

- *Geolocation* (F_{Geo}). The country in which the web server of the domain is located.
- *ASN* (F_{ASN}). The number of the Autonomous System associated with the web server IP address.
- *Registrant email address* (F_{Regn}). The email address provided upon registration of the domain.
- *Registrar* (F_{Regr}). The organization which registered the domain.
- *Creation date* (F_{CD}). The date that the domain was registered.
- *Web Server IP address* (F_{IP}). The IP address associated with the domain.
- *Class C* (F_{IP_C}) *and Class B* (F_{IP_B}) *subnets of Web Server IP addresses.* To allow the identification of servers belonging to the same infrastructure, the /24 and /16 network prefix of each IP address is extracted.
- *Web Server version* (F_{Ver}). The version of the web server of the domain.
- *Nameserver IP address* (F_{NS}). The IP address of the authoritative nameserver(s).
- *Registered domain name* (F_{Dom}). The domain name can reveal common naming schemes.

Among the different information tracked through HARMUR, we select a number of features that we believe to be likely to reveal the organized operation of one specific individual or group. Therefore, we define the following feature set: $\mathcal{F} = \{F_{Regn}, F_{NS}, F_{IP}, F_{IP_C}, F_{IP_B}\}$, which will be used by the proposed method to link rogue domains to the same campaign.

Moreover, the set $\mathcal{F}' = \{F_{Geo}, F_{ASN}, F_{Regr}, F_{CD}, F_{Ver}, F_{Dom}\}$ of the remaining features is used to validate the accuracy of our results. Indeed, rogue domains that are grouped in a single cluster should exhibit high homogeneity in terms of their location, the associated AS number, the registrar of the domain and the version of the webserver they are running on. Moreover, rogue domains that are linked to the same campaign are probably registered on the same dates and the domain names should follow similar patterns. This is due to the fact that, cyber-criminals registering a high number of rogue domains try to automate their methods in order to save time and increase their revenue.

4.2 Cluster Analysis

The 12-partite graph that is constructed by considering each of the 11 features is shown in Fig. 2. The graph is positioned using a force-directed algorithm. Force-directed algorithms aim at positioning the vertices of a graph in such a way that preserves the structure of the high-dimensional data as possible in the 2-dimensional space. Therefore, two nearby vertices on the 2-dimensional space have highly similar feature vectors, whereas two distant points should have nothing in common. This allows us to visualize the high-dimensional data set, but also to assess the consistency of the obtained clustering results.

However, the visual clusters formed by the force-directed algorithm should not be considered as indicative of the actual clusters in the dataset for a number

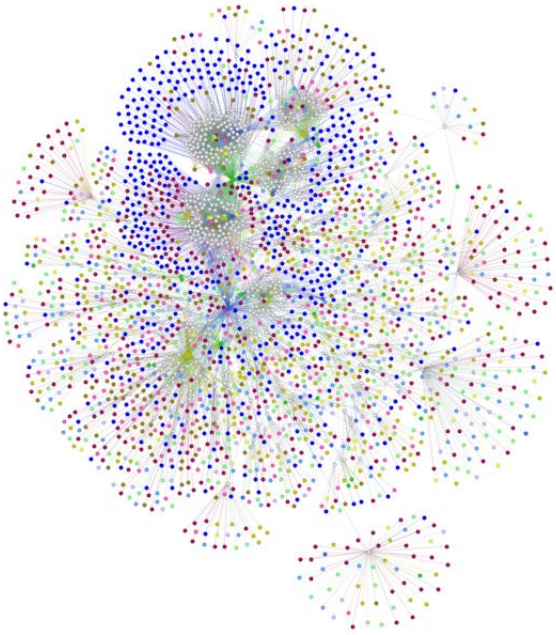

Fig. 2. The k-partite graph comprising of nodes corresponding to both websites and feature values positioned with a force-directed layout algorithm

of reasons. First, the force-directed algorithm takes into account the connections to all features and not only to the features in set $\mathcal{F}$. Second, the force-directed algorithm does not take into account the weight assigned to each feature. Last, force-directed algorithms are known to converge to local minima, which results in sub-optimal positioning of the vertices in a graph.

The weights assigned to each feature in set $\mathcal{F}$ is given by vector $\mathbf{w}$:

$$\mathbf{w} = [0.35, 0.2, 0.2, 0.15, 0.10]$$

In our discriminant analysis, we assign a higher weight to features F_{Regn}, F_{NS} and F_{IP}, since these specific features will yield a high probability that correlated rogue sites are likely due to the same campaign. On the other hand, by assigning lower weight to features F_{IP_C}, F_{IP_B} we give them a little less confidence, since these features are redundant with feature F_{IP}. The inflation parameter r and the cutting threshold q were set equal to 1.15 and 0.01 respectively.

Fig. 3 shows the results of the clustering analysis, where a different colour is used to represent a single cluster. The comparison of clusters corresponding to the results of the force-directed algorithm (position in the 2D space) with clusters corresponding to the results of our cluster analysis (colouring scheme) validate the accuracy of our method. The colour mapping allows us to have a clear overview of the coherency and high homogeneity of each cluster. Moreover,

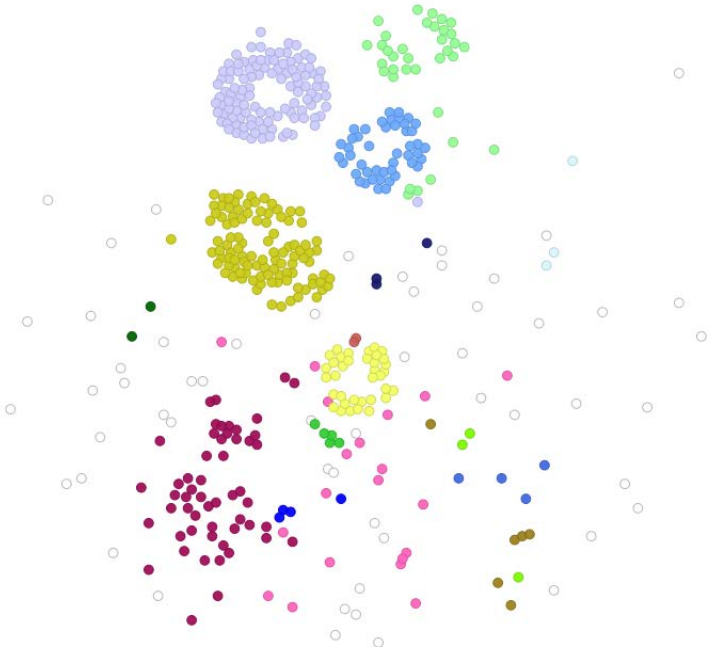

Fig. 3. Results of the clustering analysis. Only nodes corresponding to websites are depicted.

to gain insight into the root causes of each rogue AV campaign, we have to look at the contribution of each separate feature in the formation of clusters.

Indeed, from our clustering analysis, it is evident that, as far as the features in set $\mathcal{F}$ are concerned, one or a few clusters of the separate features contribute to the formation of a single cluster in the big graph. This is not always the case with features in set $\mathcal{F}'$, where a single cluster of the separate features is related to multiple clusters in the big graph. For example, rogue websites located in the USA belong to many different clusters, meaning that many different rogue campaigns are hosted in the USA. By paying special attention to the contribution of each separate feature in the formation of clusters, our clustering analysis allows us to make an interesting observation. For a specific campaign, although the rogue websites address Chinese people, as it is made obvious by the ".cn" extension of their domains, the websites themselves are hosted either in the USA or in Germany.

5 Conclusion

In this paper, we presented an unsupervised method for learning on k-partite graphs for the analysis of rogue AV campaigns. The proposed method takes

advantage of the structural properties of the k-partite graph formed by considering the natural interconnections between objects of different types. We showed that the proposed method is straightforward, fast and scalable. The results of the analysis of rogue security software campaigns were further assessed by a visual analysis tool where their validity was documented.

Acknowledgments. The authors wish to thank Symantec's senior researchers Dr. Marc Dacier, Dr. Olivier Thonnard and Dr. Corrado Leita for providing us with the HARMUR dataset and for their helpful comments that shed light into various aspects of the dynamics of rogue anti-virus campaigns.

References

1. Thonnard, O.: A multi-criteria clustering approach to support attack attribution in cyberspace. PhD thesis, École Doctorale d'Informatique, Télécommunications et Électronique de Paris (March 2010)
2. Wang, Y.M., Beck, D., Jiang, X., Roussev, R.: Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In: NDSS (2006)
3. Fossi, M., Turner, D., Johnson, E., Mack, T., Adams, T., Blackbird, J., Low, M.K., McKinney, D., Dacier, M., Keromytis, A., Leita, C., Cova, M., Overton, J., Thonnard, O.: Symantec report on rogue security software. Technical report, Symantec (October 2009)
4. Rajab, M.A., Ballard, L., Mavrommatis, P., Provos, N., Zhao, X.: The Nocebo Effect on the Web: An Analysis of Fake Anti-Virus Distribution. In: Workshop on Large-Scale Exploits and Emergent Threats (April 2010)
5. Zhuge, J., Holz, T., Song, C., Guo, J., Han, X., Zou, W.: Studying Malicious Websites and the Underground Economy on the Chinese Web. In: 2008 Workshop on the Economics of Information Security, WEIS 2008 (2008)
6. Franklin, J., Paxson, V.: An inquiry into the nature and causes of the wealth of internet miscreants. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS 2007, pp. 375–388. ACM, New York (2007)
7. Stone-Gross, B., Abman, R., Kemmerer, R., Kruegel, C., Steigerwald, D., Vigna, G.: The Underground Economy of Fake Antivirus Software. In: Proceedings of the Workshop on Economics of Information Security, WEIS (2011)
8. Cova, M., Leita, C., Thonnard, O., Keromytis, A.D., Dacier, M.: An Analysis of Rogue AV Campaigns. In: Jha, S., Sommer, R., Kreibich, C. (eds.) RAID 2010. LNCS, vol. 6307, pp. 442–463. Springer, Heidelberg (2010)
9. Cova, M., Leita, C., Thonnard, O., Keromytis, A., Dacier, M.: Gone Rogue: An Analysis of Rogue Security Software Campaigns. In: Proceedings of the 2009 European Conference on Computer Network Defense, EC2ND 2009, pp. 1–3. IEEE Computer Society (2009)
10. Dongen, S.V.: Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht (2000)
11. Satuluri, V., Parthasarathy, S.: Scalable graph clustering using stochastic flows: applications to community discovery. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 737–746. ACM, New York (2009)

12. Leita, C., Cova, M.: HARMUR: storing and analyzing historic data on malicious domains. In: Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, BADGERS 2011, pp. 46–53. ACM, New York (2011)
13. The WOMBAT Project, http://www.wombat-project.eu
14. The VIS-SENSE Project, http://www.vis-sense.eu/