Large Scale Activity Monitoring for distributed honeynets

Jerome Francois, Radu State, Olivier Festor

To cite this version:

Jerome Francois, Radu State, Olivier Festor. Large Scale Activity Monitoring for distributed honeynets. The Second International Conference on Internet Monitoring and Protection - ICIMP 2007, 2007, San Jose, United States. 10.1109/ICIMP.2007.24 . inria-00172053

HAL Id: inria-00172053

https://hal.inria.fr/inria-00172053

Submitted on 13 Sep 2007

Large Scale Activity Monitoring for distributed honeynets

Jerome François, Radu State, Olivier Festor
email (state, jerome.francois, festor)@loria.fr
Madynes research group
INRIA-LORIA
615, rue du jardin botanique
54600 Villers-les-Nancy
Nancy, France

***Abstract*— This paper proposes a new distributed monitoring approach based on the notion of centrality of a graph and its evolution in time. We consider an activity profiling method for a distributed monitoring platform and illustrate its usage in two different target deployments. The first one concerns the monitoring of a distributed honeynet, whilst the second deployment target is the monitoring of a large network telecope. The central concept underlying our work are the intersection graphs and a centrality based locality statistics. These graphs have not been used widely in the field of network security. The advantage of this method is that analyzing aggregated activity data is possible by considering the curve of the maximum locality statistics and that important change point moments are well identified.**

I. Introduction

The motivations of this paper are twofolds. The first motivation of our work is related to the conceptual approaches and algorithms required to perform distributed monitoring. If we consider a distributed monitoring platform for a given target deployment (please see figure 1), several questions must be addressed.

– Do all management agents observe the same type of events ? If no, how can we correlate a distributed view and aggregate the commonly observed evidence ?
– Can we discover a temporal behavior of the whole platform ? Do some agents tend to observe the same type of behavior during a particular time of the day, while others remain to hold a localized and very isolated observation behavior ?

A second motivation of our work came from a very realistic requirements. We are part of a large honeyney distributed over the Internet. Each individual honeypot monitors backscatter packets and incoming attacks. When working on the resulted datasets, we were challenged by the lack of methods capable to compare such a distributed platforms and to detect temporal/spatial trends in the observed traffic patterns.

Our paper is structured as follows : In section 2, a generic method for analyzing a distributed monitoring platform is described. This method uses graph intersections in order to model the distributed platform and to follow their temporal evolution. Section 3 shows how this method can be used for monitoring a large honeynet. An analysis concerning IP related headers is done for the two data sources and additional results concerning differences and analogous behavior between these two are presented. Section 4 presents related works and finally section 5 concludes the paper.

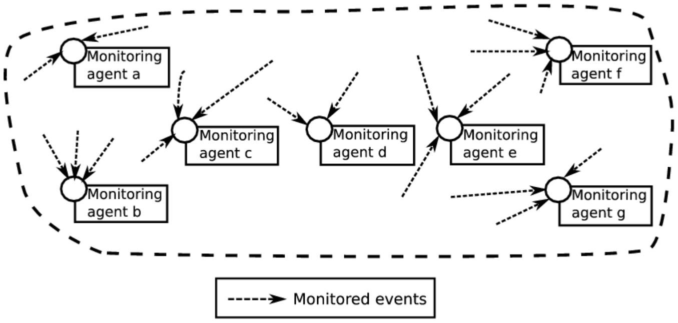


Fig. 1. Distributed monitoring model

II. Intersection graphs

The method based on intersection graphs has been introduced in [1] for profiling communications patterns between the users of a high profiled enterprise.

A. Graphs and activity profiling

A graph is composed of several nodes and arcs. Two nodes are linked if there is a relation between them. A relation can be : similarity, difference, or communication exchanges. The relation will be formally defined for each deployment target in the following sections. We consider that arcs are not directed and that the graph is an undirected graph. The adjacency matrix of a graph is a boolean square matrix where each line and each column represents a node. It is defined as :

$$A_{ij} = 1 \; if \; an \; arc \; between \; i \; and \; j \; exists, \; 0 \; else$$

$$where \; i \; and \; j \; are \; 2 \; vertices \; of \; the \; graph$$

Since we consider a undirected graph :

$$A_{ij} = A_{ji}(symmetrical\ matrix)$$

If we consider the figure 2, the corresponding matrix is :

$$A = \begin{array}{cccccccc} & a & b & c & d & e & f & g \\ a & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ b & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ c & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ d & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ e & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ f & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ g & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

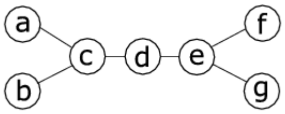


Fig. 2. An undirected graph

B. Central node

Generally, a central node is interesting because it has multiple direct or indirect relations. Using the most central node we can evaluate the centrality of the graph by counting the number of relations (arcs). A simple method to detect this node could be to get the node which has the maximum number of neighbors.

For example, in figure 2 the most connected nodes are c and e with 3 neighbors. However, if we consider the node d, this one seems to be also well connected, although it has only 2 neighbors. In fact, if a node has only few relations but these relations lead to nodes that are well connected, then the original node is interesting and central. Therefore, we can consider not only the direct neighbors but a subgraph of all nodes which are located in an area defined by the distance from the evaluated node. The centrality is the number of arcs of the subgraph.

In figure 2, considering a distance k = 2, nodes c and e have a centrality of 4. For the node d, the associated value is 6. Based on on this method, the central node is d.

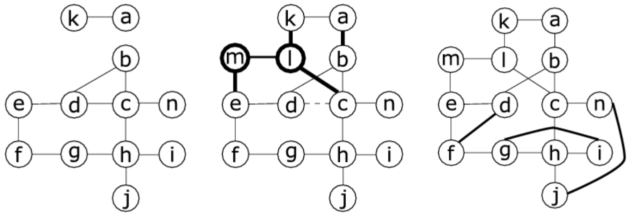
Another way to get the central nodes is to use the eigenvalues and eigenvectors, as proposed in [2].

C. Locality statistics

A graph can vary over the time and thus we need to somehow capture and describe variations in the centrality. The main idea is to consider at each time instant the central node and the associated centrality and to analyse the temporal behavior of these two entities. The intuition behind is that when major graph changes occur in the topologies of a graph, the relations between nodes change and this will be reflected by a change in the centrality too. Moreover, the central node which is responsible of the maximal centrality can be detected and the appearance or disappearance of a node implies that its relationships increased or respectively decreased.

Consider the example of the evolution of a graph which is described below :

- t = 1 : 10 nodes, 11 arcs
- t = 2 : node and arcs added but with isolated node
- t = 3 : increase of number of arcs
- t = 4 et 5 : 5 arcs added
- t = 6 : 5 nodes removed, about linear graph
- t = 7 : increase of nodes and arcs
- t = 8 : remove only one node which was isolated
- t = 9 : increase of nodes and arcs
- t = 10 : 5 nodes removed, non linear but scattered graph



(a) t = 1 (b) t = 2 (c) t = 4 et t = 5

Fig. 3. Graph time series (bold line : adding, dashed line : removing)

The following formula describes formally the locality statistics, described in the previous paragraph :

$$\psi_k(v) = \ number\ of\ arcs\ of\ the\ subgraph$$

$$of\ k\ nearest\ neighbors\ of\ v$$

$$M_k = \max_{v \in nodes} \psi_k(v) \qquad (1)$$

Figure 4 presents the result of this formula with different values of k = 1..4. For k = 0, the value is always 0 which is normal because in this case no neighbors are concerned and only the current node composes the subgraph.

The values for k = 3 and k = 4 are identical and that means that for k less than 3 it's possible to find a node having the associated subgraph of neighbors covering the total graph. This observation shows that the choice of k is important. k must not be too small because important information might not be revealed. If k is to large, all the graph is covered. In our case, the value of k = 2 seems to be a good choice.

In the figure 4, the plot for k = 2 increases up to 5 because the graph has more and more nodes and arcs. We can also observe that due to the linearity of the graph, the locality statistics decreases. The locality statistics allowed to observe this evolution. Large values of this statistics are to be associated with major changes in the inter-node relationships.

It is also important to observe the responsible nodes associated to the peaks of the locality statistics (maximum centrality). In the previous example, node c is always central.

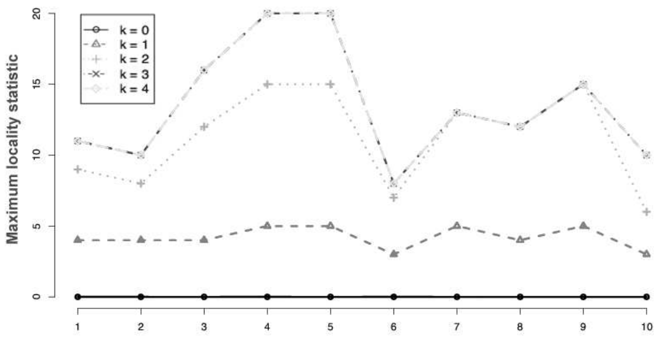


Fig. 4. Locality statistics according to time

The major goal is not only to show the evolution of the topology of the graph but in fact to discover new nodes that might become important. For instance, for time instants 3 and 4, node c is the only central node. This centrality is equal to 12 and respectively 15. The same analysis for the node g shows that its values goes from 6 to 12. In all cases, its centrality is lower that the one of c, but the evolution of g is more interesting. This type of behavior can be put into evidence by a standardized locality statistics :

$$\tilde{\psi}_{k,t}(v) = \frac{(\psi_{k,t}(v) - \hat{\mu}_{k,t,\tau}(v))}{\max(\hat{\sigma}_{k,t,\tau}(v), 1)}$$

$$\hat{\mu}_{k,t,\tau}(v) = \frac{1}{\tau} * \sum_{t'=t-\tau}^{t-1} \psi_{k,t'}(v)$$

$$\hat{\sigma}_{k,t,\tau}(v) = \frac{1}{\tau - 1} \sum_{t'=t-\tau}^{t-1} (\psi_{k,t'}(v) - \hat{\mu}_{k,t,\tau}(v))^2$$

$$\tilde{M}_{k,t} = \max_{v \in nodes} \tilde{\psi}_{k,t}(v) \qquad (2)$$

In fact, in the formula 2, the centrality is standardized with respect to previous values of a sliding window. The size of the window is τ. Nodes which tend to remain constant will have a low value. In figure 5, the interesting plot for k = 2 shows that for example between time instants 4 and 5 when the graph does not change, the associated value decreases quickly. This is due to the low value of $\tau = 5$.

When central nodes are extracted, node g becomes the only central node at time 4, showing that node c was only central at the beginning. Thus, the importance of c is lowered over time and a new node g can become an important node.

D. *From graphs to network monitoring*

If we consider a distributed monitoring platform, we can use a graph model to represent the relationships among the monitoring agents. Each agent is represented by a node in the graph. A relationship between two agents is given by the similarity in the observed data and is thus domain specific. The major idea however is to consider an arc between two nodes, if and only if the associated agents have observed a common activity. To illustrate this idea, if we consider different honeypots of a honeynet and each honeypot monitors commonly used parameters like source IP addresses, source ports, destination ports, an arc between two nodes exists if both agents have a significant overlap in the observed parameters.

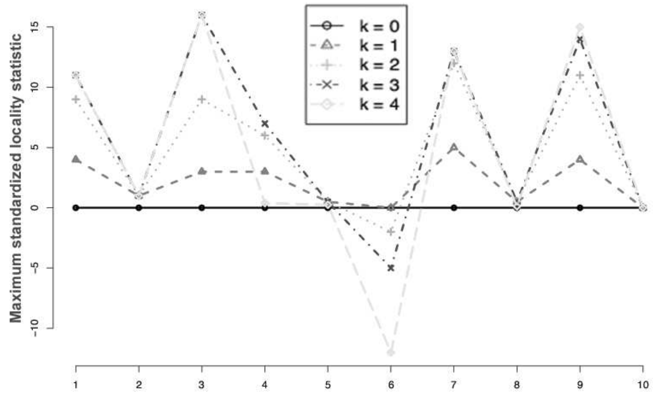


Fig. 5. Locality statistics according to time

III. Honeynet and intersection graphs

A. *Honeypot and honeynet*

A honeypot is described in [3] as an environment where vulnerabilities are deliberately introduced. Malicious intruders are lured into attacking such a system and providing useful information to security officers and researchers. Such information typically includes details about the source of the attack, temporal patterns in this activity and the tools used during and after an attack.

However, only one honeypot is not sufficient for a sound analysis at a Internet scale level. Several honeypots can be grouped into a network which is called an honeynet. In this case, all honeypot share their informations with others and they are dispersed over all the Internet.

For our work, the honeynet of the Leurre.com project was used. This network consists of 129 individual systems run by 43 honeypots. Each individual honeypot uses 3 distinct IP addresses and emulates 3 different operating systems (one operating system per address : Windows NT server, Windows 98, and Linux Red Hat 7.3). Data is collected locally and centralized in a database.

The period of our study covers the data from May to December 2004 and includes more than 11 millions IP packets. The period is sliced into weeks. The table I gives the exact details about the analyzed data.

		Honeypot
#monitored addresses		129
Number of incoming packets	05	475 519
	06	1 211 820
	07	1 495 525
	08	1 821 534
	09	1 371 280
	10	2 317 525
	11	2 292 083
	12	1 451 770
Number of unique source IP addresses	05	18 392
	06	39 419
	07	34 011
	08	49 076
	09	60 666
	10	77 032
	11	84 485
	12	82 500
Size of data	05	69 MB
	06	176 MB
	07	217 MB
	08	264 MB
	09	199 MB
	10	337 MB
	11	333 MB
	12	211 MB

TABLE I

Global information about the honeynet data. The months are represented in number (05, 06, 07...)

B. Applications

1) Source IP addresses: The goal of our first analysis was to analyse the distributed views of the honeypots with respect to the source IP addresses and identify the ones that stand out of the crowd, ie that capture suspect source addresses that are not captured by other honeypots.

Nodes represent the different honeypot platforms. For each nodes, the sets with captured source addresses are compared. Two nodes are linked only if the intersection between the corresponding sets represents less than a threshold of the union of addresses. If nodes were really distinct, there would be more and more arcs and the locality statistic would increase. The normalized locality statistic permits to detect where and when the topology changes significantly and to detect the honeypots which are responsible for the new maximal locality. These central honeypots could be considered as interesting because they detects particular source IP addresses

Figure 7 shows the plots of the simple locality statistics, using several threshold percentages. For small thresholds, the plots tend to overlap, a good setting of this value is 0.25%, where only few points are not overlapped.

On the average, there are one or more nodes having high centrality values, because these nodes are linked to all other nodes. The figure 6 shows the number of this type of nodes as well as the respective honeypot. The first value is less important because in this case all nodes are isolated. The number of honeypots that are significantly different (0.25%) decreases too. The figure 7 represents the standardized locality with $\tau = 5$ weeks. Using the method of the intersection graph, we can observe that when the value of the maximum standardized locality statistics is low, the topology of the graph is constant, while high values indicate major topology changes. The plots are generally overlapping and there are 8 peaks. The concerning central nodes have been extracted and some nodes (6) appear several time. Therefore, the 6 honeypots corresponding to these nodes are very different with respect to the remaining ones.

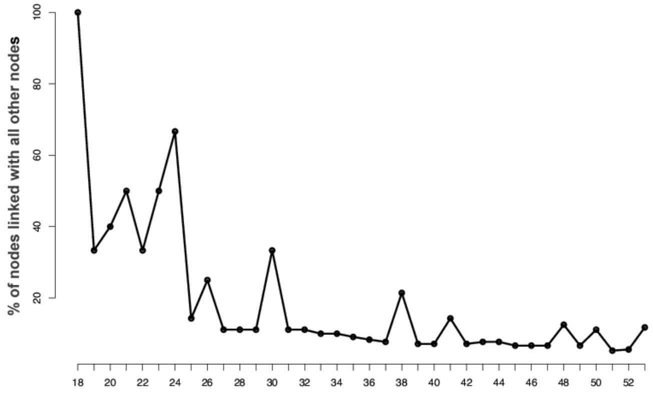


Fig. 6. Number of central nodes for the honeynet

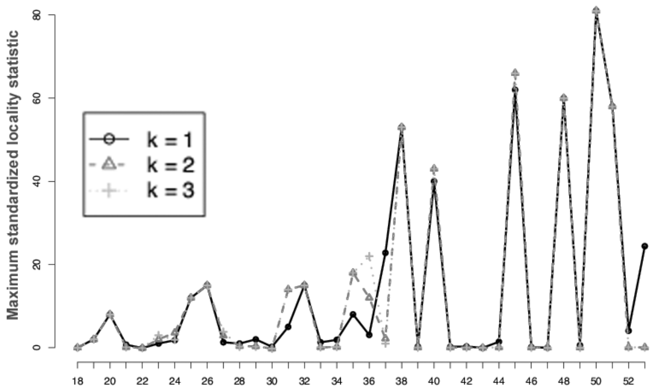


Fig. 7. Honeynet source IP addresses analysis - Standardized locality with $\tau = 5$ (shared addresses $\leq 0.25\%$)

2) Source ports: A second goal was to detect honeypots which observe port source addresses that other honeypots have not observed. Only packets with both flags SYN and ACK were considered. This kind of packets are in fact backscatter packets. In this particular case, the perceived source ports are in fact ports which have been attacked with IP spoofed packets. Thus, this study is relevant to attacked ports.

A node in the graph is a honeypot platform and similar to the previous case, an arc links 2 nodes if the set intersection of their source ports is lower than a threshold of the union of the source ports. Therefore, if honeynets were different, the locality statistic of these nodes would increase and the plots of the maximal locality statistic would show it. The plots corresponding to the unnormalized maximal locality statistic are represented in figure 8 (for a threshold of 10%) and respectively in figure 9 for a threshold of 25%. A threshold of 25% implies that the number of arcs is higher and the different plots are not overlapping. However, the aim of our work was to detect platforms that are different and a 25% threshold means that we consider 2 honeypots different even if they share one quarter of their source ports. If we consider both thresholds 10% and 25% we observe that the peaks in both plots are located at the same time instants and such the threshold of 10% is sufficient for detecting topology changes. The plots of the maximal centralized locality statistic with a sliding window size of 5 look like the figure 8 and 9.

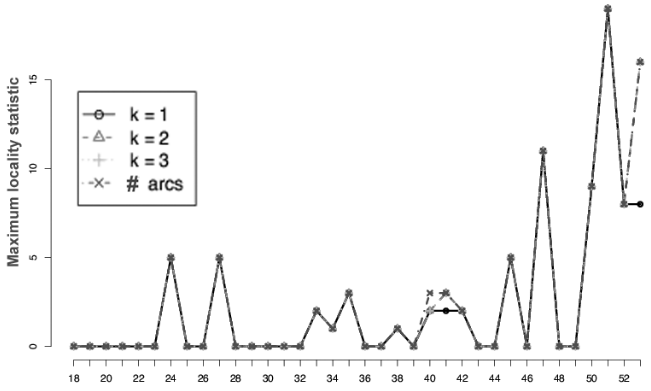


Fig. 8. Honeynet source ports analysis - locality statistic (shared ports $\leq$ 10%)

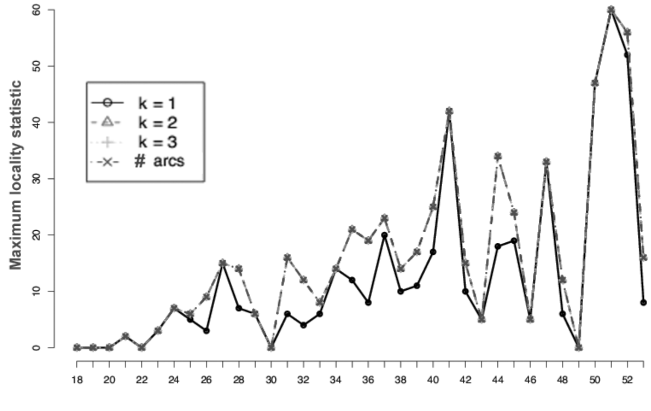


Fig. 9. Honeynet source ports analysis - locality statistic (shared ports $\leq$ 25%)

If we consider now the plots for a threshold of 10%, at many time instants the number of arcs is 0. In these cases the honeypots share more than 10% of the detected attacked ports. The ports are coded with 2 bytes in the TCP header and so 2^{16} ports are theoretically possible. However only few ports out of this large pool are really used and correspond to known deployed services.

Although several peaks are visible, the maximum locality is not very high and it's probably due to the low quantity of data at the honeynet. For instance,if the ports detected would be completely different between the 43 honeypots, the number of arcs would be : $\sum_{i=43-1}^{1} i = 946$.

3) Attack tools: A TCP session is established thanks to the 3-way handshake. First the initiator sends a packet with flag SYN and a random sequence number (also called Initial Sequence Number -ISN). The correspondent acknowledges the packets with an acknowledgment number equal to the previous sequence number + 1. Finally the initiator acknowledges this reply. Some attack tools use always the same sequence number or do not use a good (high entropy) random number generator. Consequently, the acknowledgment numbers are either always the same, or depend on the use of a specific exploit code. We looked if the same attack tool was used to attack different computers and for this work we considered also the the backscatter packets (replies of attacks).

In this case, the construction of the graphs consists in considering nodes as honeypots and two nodes will be linked if they share more than a threshold of the union of their observed acknowledgment numbers. Using a threshold of 90% the plots are given in figure 10. In general the acknowledgment numbers are different between platforms because the number of arcs is low. This is due to the diversification of the attack tools.

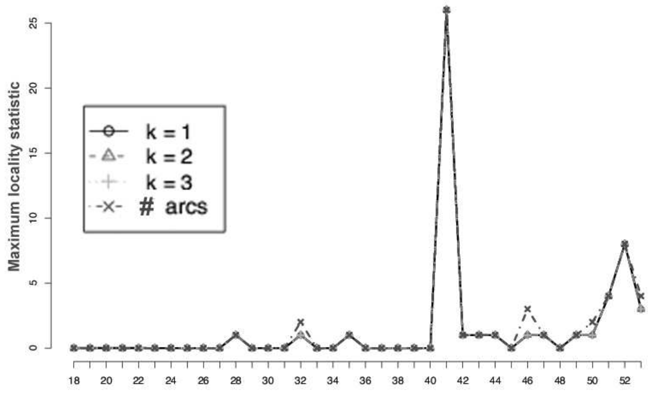
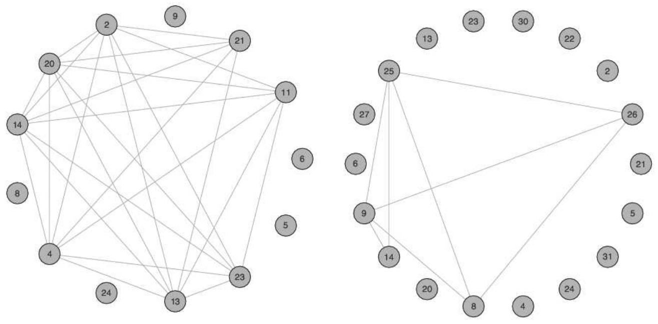


Fig. 10. Honeynet acknowledgment numbers analysis - locality statistic (shared acknowledgment numbers $\geq$ 90%)

Two peaks are clearly visible and in these case the plots are overlapping. This shows the presence of one or central honeypot linked with all others. Using the standardized locality statistic with a sliding window size of 5, the obtained plots are similar because the standardization is made thanks to previous values, which are mostly equal to 0. The figure 11 presents the graphs of weeks 41 and 52 corresponding to the peaks. In the figure 11(a), many nodes are linked with many others. A lot of honeypots have detected about the same acknowledgment numbers (threshold $\geq 90\%$) and the use of the same attack tools is undeniable. However for the second peak in week 52, (shown in the figure 11(b)) the picture is totally different and only some honeypots are concerned. In this case, this is probably due to a same attack tool with a bad random numbers generator which implies that the same generated number is used several times and detected by different honeypots.



(a) week 41

(b) week 52

Fig. 11. Intersection graphs for acknowledgment numbers shown by the honeynet

IV. Related works

The honeypots and honeynets are presented in [3] where general definitions and platform description are are given. That reference containts also results about the localization of the attacks or the observation of worm spreading in the context of the Leurre.com project.

In [4], the same authors propose a more elaborated method to study the data of the honeynet. In fact, the authors clusterize the different captured network packets using the Levenshtein distance in order to group packets which are due to the same attack.

In [5], the goal of the paper is to determine the degree of the interaction of a honeypot needed to collect useful data, while in the same time avoiding to collect too much useless data. Even if it seems that a low level interaction honeypot is sufficient, the use of a high level of interaction degree is needed to correctly configure the low level interaction.

The reference book in system administration [9] includes several examples on the use of graphs and the centrality of a node by using eigen vectors. The first work applying these techniques to security monitoring is [1], where the email exchanges in the enron database is analyzed in order to prove that that some employees had inside level information on the fraudulos management. The same method was applied to network security in [10] for end user level activity profiling. The goal was to detect if the websites visited by employes can be associated to a normal type of behavior and how malware spreading can be detected if anormal activity is observed.

V. Conclusions

In the work presented in this paper we were challenged by several research questions. Firstly, we needed a generic method to analyze large scale honeynet data.

The central concept underlying our work are the intersection graphs. These graphs have not been used widely in the field of network security. The advantage of this method is that analyzing aggregated data is possible by considering the curve of the maximum locality statistic and the maximum standardized locality statistics. This is possible because these plots are closely related to the trend of the variation in the topology of a graph. This method allows also to identify the nodes, which are important in the graph. Importance can be assimilated with monitoring agents that observe unusual network activities.

Acknowledgment We thank Fabien Pouget and Marc Dacier from the Leurre.com project for their collaboration in the honeypot project.

References

[1] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park, “Scan statistics on enron graphs,” *Comput. Math. Organ. Theory*, vol. 11, no. 3, pp. 229–247, 2005.

[2] M. Burgess, *Analytical network and system administration.* John Wiley & Sons Ltd., 2004.

[3] F. Pouget, M. Dacier, and H. Debar, “Attack processes found on the Internet,” in *NATO Research and technology symposium IST-041/RSY-013 ”Adaptive Defence in Unclassified Networks”, 19 April 2004, Toulouse, France*, Apr 2004.

[4] F. Pouget and M. Dacier, “Honeypot-based forensics,” in *AusCERT2004, AusCERT Asia Pacific Information technology Security Conference 2004, 23rd - 27th May 2004, Brisbane, Australia*, May 2004.

[5] F. Pouget and T. Holz, “A pointillist approach for comparing honeypots,” in *DIMVA 2005, Conference on Detection of Intrusions and Malware & Vulnerability Assessment, July 7-8, 2005, Vienna, Austria - Also published in LNCS Volume 3548*, Jul 2005.

[6] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, “Inferring internet denial-of-service activity,” *ACM Trans. Comput. Syst.*, vol. 24, no. 2, pp. 115–139, 2006.

[7] K. E. Giles, D. J. Marchette, and C. E. Priebe, “On the spectral analysis of backscatter data,” in *Hawaii International Conference on Statistics and Related Fields*, 2004.

[8] V. Yegneswaran, P. Barford, and D. Plonka, “The design and use of internet sinks for network abuse monitoring,” 2004.

[9] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, *Internet Denial of Service : Attack and Defense Mechanisms*, ser. Radia Perlman Computer Networking and Security. Prentice Hall PTR, december 2004.

[10] D. J. Marchette, “Statistical opportunities in network security,” in *35th Symposium on the interface*, 2003.